МИКРОСХЕМЫ ИНТЕГРАЛЬНЫЕ
К1921ВГ1Т, К1921ВГ10Я

Руководство пользователя

от 10.06.2026

К1921ВГ1Т, К1921ВГ10Я – микроконтроллеры архитектуры RISC-V SCR5 с 4Мб основной и 512Кб дополнительной Flash-памяти, 1Мб ОЗУ, ADC, DAC, PWM, ACMP, TRNG, CRYPTO, HASH, CRC, QSPI, SPI, UART, I2C, I2S, LIN, CAN, CANFD, USB, Ethernet.

Особенности

- Сброс и питание:
  - напряжение питания: 3,0 В – 3,6 В;
  - температурный диапазон: (-40 – 105) °С;
  - POR, BOR;
- Ядро: RISC-V SCR5 RV32IMAFDCP
  - до 204 МГц;
- Тактовые сигналы:
  - внутренний осциллятор HSE для подключения внешнего резонатора от 2 МГц до 27 МГц;
  - внутренний RC-генератор HSI: 4 МГц;
  - системная PLL: от 200кГц до 204 МГц.
  - внутренний RC-генератор блока RTC: 32кГц.
- 112 выводов I/O.
- Память:
  - по 32 Кб кэш команд и данных L1;
  - 128 Кб общий кэш L2;
  - 16 Кб OTP память;
  - 4 Мб основной Flash-памяти с ECC;
  - 512 Кб дополнительной Flash-памяти с ECC;
  - 1 Мб памяти ОЗУ с ECC;
  - контроллер внешней памяти EMC объемом до 128 Мбайт;
  - контроллер внешней памяти SDRAM объемом до 32 Мбайт.
- Аналоговая периферия:
  - 48-канальный 12-разрядный быстродействующий АЦП с режимами цифрового компаратора;
  - датчик температуры TSENSOR, подключаемый к одному из входов АЦП;
  - два 12-разрядных блока ЦАП;
  - четыре аналоговых компаратора ACMP.
- Контроллер DMA: 32 канала.
- Контроллеры интерфейсов:
  - восемь приемопередатчиков UART;
  - контроллер CAN 2.0b (4 узла);
  - четыре контроллера CAN FD (по 1 узлу в каждом);
  - восемь контроллеров LIN
  - четыре контроллера I2C;
  - два контроллера I2S;
  - восемь контроллеров SPI;
  - два контроллера QSPI;
  - контроллер USB 2.0 (FS/HS) с FS PHY;
  - контроллер Ethernet 10/100/1000 с PHY;
  - контроллер MSC.
- Шестнадцать таймеров 32-разрядных.
- Блок часов реального времени RTC.
- Сторожевой таймер WDT.
- Шестнадцать двухканальных блоков ШИМ (PWM).
- Блок CRYPTO: AES-128; AES-256, Кузнечик, Магма.
- Блок генерации случайных чисел TRNG.
- Блок HASH: SHA-1, SHA-224, SHA-256, MD5.
- Блок подсчета CRC: CRC7, CRC8, CRC16, CRC32.
- Интерфейс отладки: JTAG.

Отладчики: Olimex ARM-USB-OCD-H, J-Link V9, CMSIS-DAP, FT2232.

- Корпус LQFP-208 (К1921ВГ1Т).
- Корпус BGA-216 (К1921ВГ10Я).

Содержание

1 Введение 8
2 Область применения и особенности микроконтроллера 9
3 Краткое техническое описание микроконтроллера 10
3.1 Функциональные параметры 10
3.2 Электрические параметры 47
4 Архитектура изделия 50
4.1 Блок коммутации микроконтроллера 50
4.2 Механизмы функциональной безопасности 51
5 Система питания 53
5.1 Организация системы питания 53
5.2 Супервизор и мониторы питания 53
5.3 Способы снижения энергопотребления 54
6 Блок управления сбросом и синхронизацией RCU 56
6.1 Общая система тактирования 56
6.2 Синтезатор частоты PLL 58
6.3 RC-генератор 61
6.4 Осциллятор HSE 63
6.5 Система слежения за тактовыми сигналами 64
6.6 Сигналы сброса 65
6.7 Прерывания 65
6.8 Тактирование и сброс периферийных блоков 65
7 Блок управления системой SIU 67
7.1 Описание блока 67
7.2 Сборщик неисправностей 68
7.3 Описание работы 68
7.4 Тайм-аут при зависании шин 69
8 Организация памяти 70
8.1 Модуль коррекции ошибок ECC SRAM 73
8.2 Модуль внедрения ошибок EIM 73
9 Контроллер Flash-памяти 76
9.1 FlashM-память 76
9.2 FlashD-память 78
9.3 OTP-память 79
9.4 Сервисный сброс всей Flash-памяти 79
9.5 Модуль коррекции ошибок ECC FLASH 79
10 Контроллер внешней памяти EMC 81
11 Контроллер внешней памяти SDRAM 83
11.1 Особенности контроллера SDRAM 83
11.2 Описание внешних выводов 84
11.3 Описание работы SDRAM 84
11.4 Функциональное описание 86
11.5 Интерфейс SDRAM 87
12 Микропроцессорное ядро SCR5 91
12.1 Структура ядра 92
12.2 Конвейер инструкций 93
12.3 Модуль операций с плавающей запятой 94
12.4 Модуль DSP инструкций 94
12.5 Уровни привилегий 95
12.6 Регистры управления/состояния (CSR) 96
12.7 Система мониторинга производительности ядра 130
12.8 Управление доступом к памяти 137

12.9 Кэш первого уровня (L1) 140
12.10 Операции кэша уровня L1 143
12.11 Кэш второго уровня (L2) 144
12.12 Блок управления памятью (MMU) 160
12.13 Исключения и прерывания 165
12.14 Ввод-вывод с отображением в памяти 168
12.15 Модуль межъядерной связи (ICCM) 169
12.16 Система команд 172
12.17 Режим работы ядер Lockstep 183
13 Конфигурируемый контроллер прерываний уровня платформы PLIC 185
13.1 Структура контроллера PLIC 185
13.2 Определения 186
13.3 Обзор функционирования 186
13.4 Векторы прерываний 186
13.5 Конфигурация прерываний 194
13.6 Программная модель обработки внешних прерываний 196
14 Контроллер прямого доступа к памяти DMA 197
14.1 Режимы работы 199
14.1 Дескрипторы DMA 201
14.2 Цепочки дескрипторов DMA 202
14.1 Запросы от периферийных модулей 202
14.2 Передача между периферийными устройствами 203
14.1 Арбитраж 203
14.2 Глубина прерывания 204
14.3 Программное управление периферийным запросом 204
14.4 Тайм-ауты AXI 205
14.5 Сторожевой таймер 205
14.6 Ограничение тактирования 205
14.1 Ограничение ожидающих выполнения команд AXI 205
14.2 Изменение порядка следования байт 205
14.1 Логгер ошибочных передач 206
14.2 Сценарии настройки 206
15 Порты ввода-вывода GPIO 208
15.1 Функционирование порта 208
15.2 Режим альтернативных функций 209
15.3 Входные фильтры 210
15.4 Прерывания 210
15.5 Генерация аппаратных запросов 211
15.6 Механизм блокировки конфигурации 212
15.7 Механизм маскирования 212
16 Таймеры TMR 214
16.1 Функционирование таймера 215
16.2 Режимы счёта 216
16.3 Блоки захвата/сравнения 220
16.4 Выходной модуль 221
16.5 Прерывания таймера 225
16.6 Взаимодействие с DMA 226
16.7 Генерация запросов внешних событий 226
17 Контроллер интерфейса MSC (Micro Second Channel) 227
17.1 Общие положения 227
17.2 Передающий канал (downstream) 228
17.3 Канал приема (upstream) 236

17.4 Прерывания 241
17.5 Конфигурация входов/выходов 242
18 Блоки ШИМ (PWM) 245
18.1 Таймер 246
18.2 Компаратор 249
18.3 Обработчик событий 250
18.4 Пороговый выключатель 254
18.5 Генератор задержки ШИМ 255
18.6 Фильтр коротких импульсов 257
18.7 Модулятор 257
18.8 Детектор сигнала аварии 259
18.9 Триггер событий 260
18.10 ШИМ высокого разрешения 263
19 Блок АЦП (ADC) 266
19.1 Общее описание 266
19.2 Секвенсор 268
19.3 Модуль АЦП 276
19.4 Цифровой компаратор 280
19.5 Прерывания 283
19.6 Примеры работы блока АЦП 284
20 Блок ЦАП (DAC) 290
20.1 Генератор событий запуска 292
20.2 Запросы от периферийных блоков к ЦАП 292
20.3 Запросы DMA 293
20.4 Формат данных ЦАП 293
20.5 Устройство выходного каскада к физическому блоку ЦАП 294
20.6 Режимы работы 295
20.7 Преобразование 299
21 Датчик температуры TSENSOR 300
21.1 Функциональное описание 300
21.2 Электрические характеристики 301
21.3 Управление датчиком температуры 302
22 Модули аналоговых компараторов ACMP 303
22.1 Описание модуля 303
22.2 Функциональная безопасность 304
23 Блок генератора случайных чисел TRNG 305
23.1 Источник энтропии 305
23.2 Процесс дискретизации 306
23.3 Непрерывное тестирование 307
23.4 Тест на количество повторений 307
23.5 Тест на подсчет пропорций (окно 512-недвоичных выборок) 307
23.6 Тест корреляции 307
23.7 Тест на подсчёт автокорреляции 307
23.8 Тест при запуске 308
23.9 Смешивание XOR 308
23.10 Обработка 308
23.11 FIFO 308
23.12 LFSR 308
23.13 Прерывания 309
23.14 Программирование ИГСЧ 309
23.15 Производительность 312
24 Хеш-процессор (HASH) 313

24.1 Функционирование хеш-процессора 313
24.2 Основные особенности....313
24.3 Описание хеш-процессора 313
24.4 Длительность обработки 314
24.5 Типы данных 314
24.6 Вычисление хеш-суммы сообщения....316
24.7 Автоматическое заполнение сообщения 317
24.8 Вычисление хеш-функции 318
24.9 Вычисление HMAC 318
24.10 DMA интерфейс 320
24.11 Прерывания 320
25 Блок подсчёта циклического избыточного кода CRC 321
25.1 Особенности блока CRC 321
25.2 Функции блока CRC 321
26 Модуль криптографии CRYPTO....324
26.1 Общие положения....324
26.2 Структура блока криптографии....324
26.3 Режимы выполнения криптографических операций....325
26.4 Режимы работы 333
26.5 Экстренное прекращение выполнения операции 344
26.6 Производительность....344
27 Сторожевой таймер WDT 345
28 Блок часов реального времени RTC 347
29 Приемопередатчики UART 350
29.1 Функционирование блока UART 351
29.2 Интерфейс прямого доступа к памяти....354
29.3 Прерывания 355
29.4 Программирование 356
30 Контроллеры интерфейса SPI....357
30.1 Структура контроллера SPI 357
30.2 Интерфейс прямого доступа к памяти....359
30.3 Функционирование....359
30.4 Прерывания 363
31 Контроллеры интерфейса QSPI....365
31.1 Структура контроллера QSPI....365
31.2 Режим работы SPI....366
31.3 Последовательность команд QSPI 367
31.4 Режимы протокола интерфейса сигнала QSPI....368
31.5 Программирование QSPI....371
31.6 Бит занятости QSPI и функция прерывания 372
31.7 Поведение сигнала FSS....372
31.8 Прерывания QSPI....372
32 Контроллеры интерфейса I2S....373
32.1 Режимы работы....373
32.2 Передача и прием данных....373
32.3 Особенности блока I2S....374
32.4 Логика работы блока 374
32.5 Примеры конфигурации блока....377
33 Контроллеры интерфейса LIN....378
33.1 Байтовые поля 378
33.2 Режим сна 381
33.3 Регистры управления и контроля....382

34 Контроллеры интерфейса I2C 384
34.1 Протокол шины 384
34.2 Функциональное описание 391
34.3 Инициализация и функционирование 394
35 Контроллер интерфейса USB 2.0 DC 409
35.1 Общее описание 409
35.1 Операции DC контроллера 409
35.2 Операции DMA 413
35.3 Конечные точки EP 419
35.4 Пример реализации DC контроллера 422
36 Контроллер интерфейса USB 2.0 HC 424
36.1 Общие сведения 424
36.1 Принцип работы 425
36.2 Операции DMA 426
36.3 Поддержка приемо-передатчиков 426
37 Блок физического уровня интерфейса USB 1.1 (PHY) 427
37.1 Назначение блока 427
37.2 Общая структура блока 427
37.3 Описание цифровой части USB PHY 428
37.4 Тактирование блока 444
38 Контроллер интерфейса CAN 2.0B 445
38.1 Протокол CAN 445
38.2 Структура и функционирование контроллера CAN 451
38.3 Узел CAN 458
38.4 Объекты сообщений 464
38.5 Прием и передача сообщений 467
38.6 Фильтрация сообщений 470
38.7 Удаленные запросы 471
38.8 Дополнительные режимы передачи 472
38.9 FIFO структура объектов сообщений 473
38.10 Режим шлюза 476
38.11 Прерывания объектов сообщений 479
38.12 Программирование контроллера CAN 482
38.13 Пример расчета скорости передачи 1Мбит/с 482
39 Контроллер интерфейса CAN FD 484
39.1 Область применения 484
39.2 Основные характеристики 484
39.3 Описание работы CAN 484
39.4 Описание сигналов тактирования и сброса 489
39.1 Структура регистров 491
39.2 Инструкция по программированию 492
40 Контроллер Ethernet 10/100/1000 496
40.1 Общее описание 496
40.2 Функционирование 498
40.3 Интерфейс DMA на передачу 499
40.4 Интерфейс DMA на прием 502
40.5 Интерфейс MDIO 505
40.6 Блок физического уровня интерфейса Ethernet 10/100/1000 (PHY) 505
41 Программно-аппаратные средства отладки 510
42 Электрические параметры микроконтроллера 511
42.1 Предельные нормы параметров 511
42.2 Допустимые нормы параметров 512

42.3 Электрические параметры периферийных блоков 513
43 Заключение 519
Приложение А (обязательное) Регистры микроконтроллера 520
А.1 Регистры блока управления сбросом и синхронизацией RCU 520
А.2 Регистры блока управления системой SIU 553
А.3 Регистры контроллера FlashM-памяти 587
А.4 Регистры контроллера FlashD-памяти 598
А.5 Регистры контроллера памяти SRAM с кодом коррекции Хэмминга 609
А.6 Регистры контроллера внешней памяти EMC 614
А.7 Регистры контроллера SDRAM 615
А.8 Регистры контроллера прямого доступа к памяти DMA 619
А.8.1 Регистры каналов DMA 620
А.8.2 Общие регистры DMA 639
А.9 Регистры портов ввода-вывода GPIO 649
А.10 Регистры 32-разрядного таймера TMR 673
А.11 Регистры блока MSC 683
А.12 Регистры блоков ШИМ 700
А.13 Регистры блока АЦП 729
А.14 Регистры блока ЦАП 752
А.15 Регистры управления аналоговым компаратором 761
А.16 Регистры генератора случайных чисел ИГСЧ 766
А.17 Регистры блока HASH 781
А.18 Регистры блока подсчета циклического избыточного кода CRC 789
А.19 Регистры модуля CRYPTO 792
А.20 Регистры сторожевого таймера WDT 804
А.21 Регистры блока RTC 808
А.22 Регистры приемопередатчика UART 813
А.23 Регистры контроллера интерфейса SPI 825
А.24 Регистры контроллера интерфейса QSPI 831
А.25 Регистры контроллера интерфейса I2S 839
А.26 Регистры контроллера интерфейса LIN 847
А.27 Регистры контроллера интерфейса I2C 856
А.28 Регистры контроллера USB 2.0 DC 865
А.29 Регистры контроллера USB 2.0 HC 871
А.30 Регистры управления USBCTR (PHY) 893
А.31 Регистры контроллера интерфейса CAN 2.0B 897
А.32 Регистры контроллера интерфейса CANFD 924
А.33 Регистры контроллера Ethernet 10/100/1000 944
Приложение Б (обязательное) Коды состояний функционирования блока I2C 954
Приложение В (обязательное) Назначение выводов в корпусе 962
Приложение Г (обязательное) Габаритные чертежи корпусов 967
Приложение Д (справочное) Примеры схем включения микроконтроллеров 969

1 Введение

32-разрядные микроконтроллеры применяются при разработке и изготовлении электронной техники. Высокая вычислительная мощность и при этом относительно низкая стоимость делают эти устройства привлекательными для самого широкого круга разработчиков.

Микросхемы К1921ВГ1Т, К1921ВГ10Я представляют собой СБИС 32-разрядного микроконтроллера на базе двух высокопроизводительных ядер RISC-V, предназначенных для промышленных и потребительских приложений, включая системы дистанционного мониторинга, контрольно-измерительные приборы, сетевые устройства, системы автоматизации производственных процессов, автомобильную электронику, системы управления электродвигателями.

Микросхемы К1921ВГ1Т и К1921ВГ10Я имеют общий кристалл, различие только в корпусах LQFP-208 и BGA-216. Периферийный состав идентичен.

В состав микроконтроллера входит широкий набор цифровой и аналоговой периферии, в связи с чем, он может применяться в различных системах цифровой обработки сигналов, в том числе, требующих точных аналогово-цифровых преобразований, в системах управления и сбора информации.

Наличие в микроконтроллере аппаратных интерфейсов, применяемых для связи, мониторинга и управления модулями автомобильной электроники расширяет его возможности по применению в составе блоков, систем и комплексов автомобильного электрооборудования.

В настоящем техническом описании приведено описание архитектуры, функционального построения и периферии микроконтроллеров К1921ВГ1Т, К1921ВГ10Я. Техническое описание может служить практическим руководством по применению микроконтроллеров для разработчиков систем на их основе и программистов.

2 Область применения и особенности микроконтроллера

Сфера применения микросхем К1921ВГ1Т, К1921ВГ10Я довольно широка: средства измерений, связи, наблюдения, безопасности, автоматизация производства, медицины, энергетики, промышленности, систем электронного оборудования автомобилей.

Микросхемы К1921ВГ1Т, К1921ВГ10Я разработаны на базе двух высокопроизводительных ядер RISC-V SCR5.

Разработанные микроконтроллеры имеют встроенную Flash-память программ объемом 4 Мбайта, которую можно использовать для хранения и загрузки пользовательского программного обеспечения. Также имеется дополнительная Flash-память объемом 512 Кбайт и память данных однократного программирования (OTP) объемом 16 Кбайт.

Система тактирования микроконтроллера позволяет использовать различные источники тактового сигнала, что позволяет расширить набор применений и решаемых задач пользователя. Существует возможность гибкой настройки тактовых сигналов для блоков периферии.

3 Краткое техническое описание микроконтроллера

3.1 Функциональные параметры

Структурная схема микроконтроллеров К1921ВГ1Т, К1921ВГ10Я показана на рисунке 3.1.

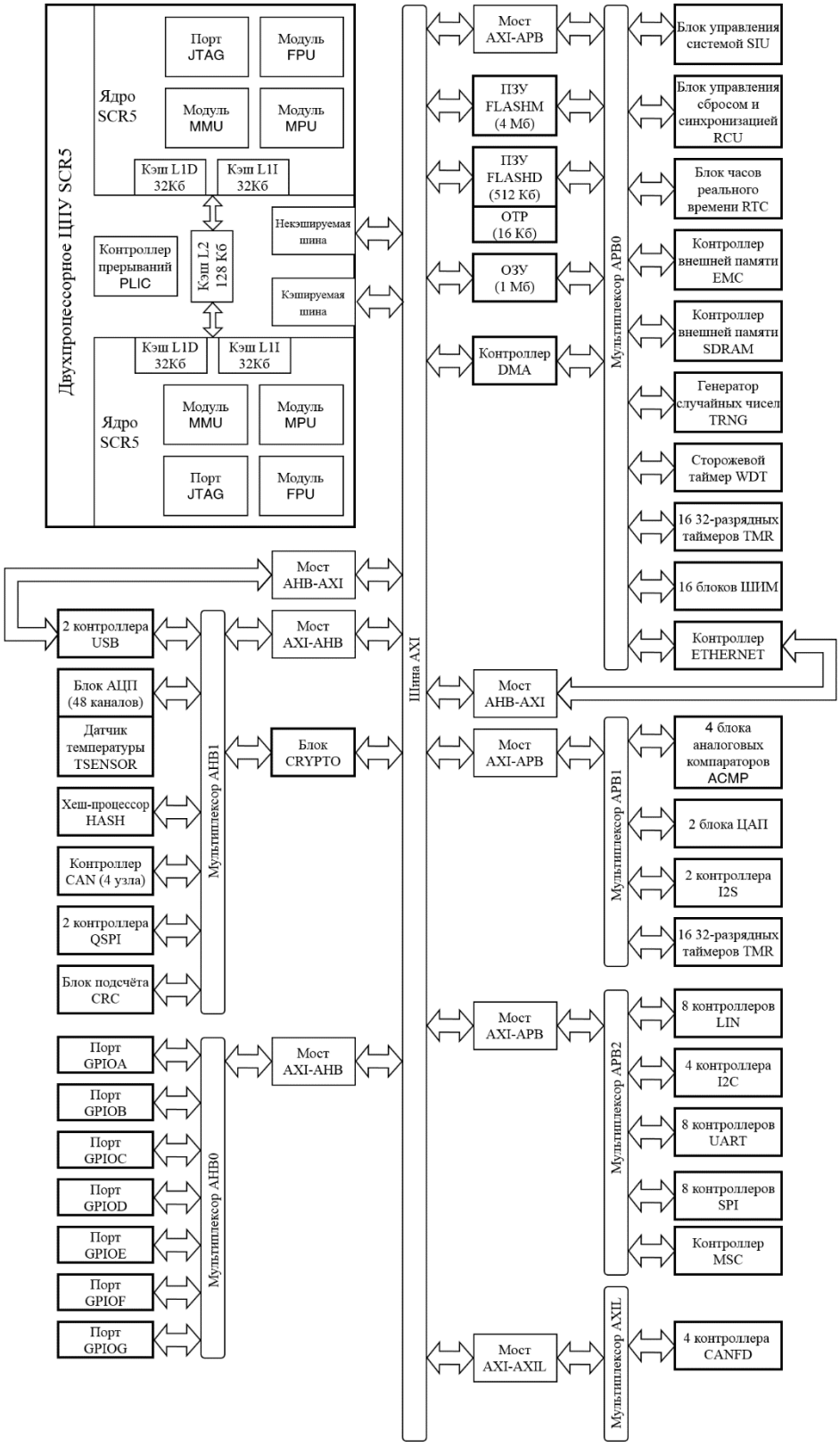


Рисунок 3.1 – Структурная схема микроконтроллеров К1921ВГ1Т, К1921ВГ10Я

Состав микроконтроллеров К1921ВГ1Т и К1921ВГ10Я

В состав микроконтроллеров К1921ВГ1Т, К1921ВГ10Я входят функциональные элементы:

- двухъядерное 32-разрядное ЦПУ с поддержкой DSP инструкций, поддержкой набора одноцикловых команд умножения с накоплением, команд централизованного управления потоком данных, арифметических и логических команд, встроенным модулем обработки команд с плавающей запятой с одинарной точностью FPU, поддержкой отладочного интерфейса JTAG;
- блок управления сбросом и синхронизацией RCU, имеющий в своем составе RC-генератор (4 МГц) и 3 синтезатора частоты PLL;
- блок управления системой SIU;
- основная FlashM-память объемом 4 Мбайт с кодом коррекции ECC для хранения кода программ (FLASHM);
- дополнительная FlashD-память объемом 512 Кбайт с кодом коррекции ECC для хранения кода программ (FLASHD);
- однократно программируемая (OTP) память объемом 16 Кбайт для хранения данных;
- кэш команд и кэш данных первого уровня объемом по 32 Кбайт каждый;
- общий кэш второго уровня 128 Кбайт;
- ОЗУ объемом 1 Мбайт с кодом коррекции ECC;
- контроллер внешней памяти (EMC) объемом до 128 Мбайт, поддерживающий SRAM, ROM и NOR Flash;
- контроллер внешней памяти (SDRAM) объемом до 32 Мбайт;
- 32-канальный контроллер прямого доступа к памяти (DMA);
- семь 16-разрядных портов ввода-вывода A, B, C, D, E, F, G (GPIOA – GPIOG);
- шестнадцать 32-разрядных таймеров с блоком захвата/сравнения и возможностью генерации ШИМ (TMR0 – TMR15);
- шестнадцать двухканальных ШИМ-модуляторов (PWM0 – PWM15);
- интерфейс подключения внешних устройств питания (MSC);
- 48-канальный 12-разрядный быстродействующий АЦП с режимами цифрового компаратора (ADC);
- два 12-разрядных блока ЦАП (DAC0 – DAC1);
- датчик температуры TSENSOR, подключаемый к одному из входов АЦП;
- четыре аналоговых компаратора ACMP (ACMP0 – ACMP3);
- аппаратный истинный генератор случайных чисел (TRNG);
- HASH–процессор (HASH);
- аппаратный блок для подсчёта циклического избыточного кода (CRC);
- модуль поддержки криптографических вычислений (CRYPTO);
- сторожевой таймер (WDT);
- блок RTC с батарейным питанием, тактированием от внешнего генератора 32,768 кГц, контролем генерации и автоматическим переходом на внутренний генератор в случае сбоев (RTC);
- супервизоры (POR/BOR) и мониторы питания;
- контроллеры интерфейсов:
  - восемь приемопередатчиков UART (UART0 – UART7);
  - восемь контроллеров SPI (SPI0 – SPI7);
  - два контроллера QSPI (QSPI0 – QSPI1);
  - два контроллера I2S (I2S0 – I2S1);
  - восемь контроллеров LIN (LIN0 – LIN7);
  - четыре контроллера I2C (I2C0 – I2C3);
  - два контроллера USB 2.0 с интерфейсом UTMI+ (USBDC–Device, USBHC–Host);
  - блок физического уровня USB 1.1 PHY;
  - контроллер CAN 2.0B, содержащий 4 узла (CAN);

- четыре контроллера CANFD (CANFD0 – CANFD3), по одному узлу в каждом;
- контроллер Ethernet 10/100/1000 с блоком физического уровня PHY (ETH).

Особенности выводов микроконтроллеров

Все выводы микроконтроллера по их функциональному назначению, организованы в группы:

- 16-разрядные порты ввода-вывода A, B, C, D, E, F, G;
- порт тестирования JTAG;
- выводы передачи/приема данных Ethernet PHY;
- выводы передачи/приема данных USB PHY;
- выводы сигналов блоков ЦАП;
- питание аналогового домена, включающего блоки:
    - АЦП последовательного приближения (ADC);
    - внутренний датчик температуры (TSENSOR);
    - синтезатор частоты PLL;
    - блок Ethernet PHY;
- питание микроконтроллера.

Каждый из выводов портов ввода-вывода является двунаправленным выводом общего назначения с набором альтернативных функций. Режим работы, альтернативная функция, а также подтяжка (к высокому уровню) и функционирование в режиме выхода с открытым стоком/истоком могут быть заданы для каждого вывода независимо от других.

Примечание – После сброса микроконтроллера выводы портов ввода-вывода конфигурируются как выводы общего назначения и находятся в третьем состоянии.

Порт JTAG, предназначенный для внутрисхемного программирования микроконтроллера, тестирования и отладки программ пользователя, включает в свой состав пять выводов TCK, TMS, TDI, TDO, TRST для подключения JTAG эмулятора.

Вывод RST# является входом сигнала сброса микроконтроллера и должен находиться в состоянии логической единицы. Внешний сброс осуществляется подачей на вывод логического нуля в течение минимум 10 мкс.

Вывод SERVEN запрещает (в состоянии логической единицы) любые операции со всей Flash-памятью, кроме полного его стирания;

Выводы XI_RTC и XO_RTC предназначены для подключения внешнего источника тактового сигнала (с частотой 32 кГц) блока RTC.

Выводы XI_ETH и XO_ETH предназначены для подключения внешнего источника тактового сигнала блока Ethernet.

Выводы XI_OSC и XO_OSC предназначены для подключения внешнего источника тактового сигнала микроконтроллера с частотой (2 – 27) МГц.

Вывод VBAT предназначен для подключения внешнего источника питания (батарейки) блока RTC. Номинальное значение напряжения должно находиться в диапазоне от 1,8 до 3,3 В.

Примечания:

1) На плате вывод питания GND может быть объединен с выводами питания AGND_ETH (при этом должны быть приняты меры для снижения помех по питанию).

2) Выключение питания микроконтроллера подразумевает полное снятие напряжения как с выводов питания VCC, AVCC_ADC, AVCC_ETH, VBAT, так и со всех остальных выводов микроконтроллера.

3) Запрещено подавать напряжение на функциональные выводы при выключенном питании микроконтроллера.

УГО и назначение выводов микроконтроллера К1921ВГ1Т

Условное графическое обозначение микроконтроллера приведено на рисунке 3.2.

Функциональное назначение выводов микроконтроллера приведено в таблицах 3.1 – 3.2, в которых приняты обозначения: I – вход, O – выход, I/O – вход/выход, Z – третье состояние.

Примечание – Альтернативные функции выводов портов A, B, C, D, E, F, G указаны в таблице 3.1.

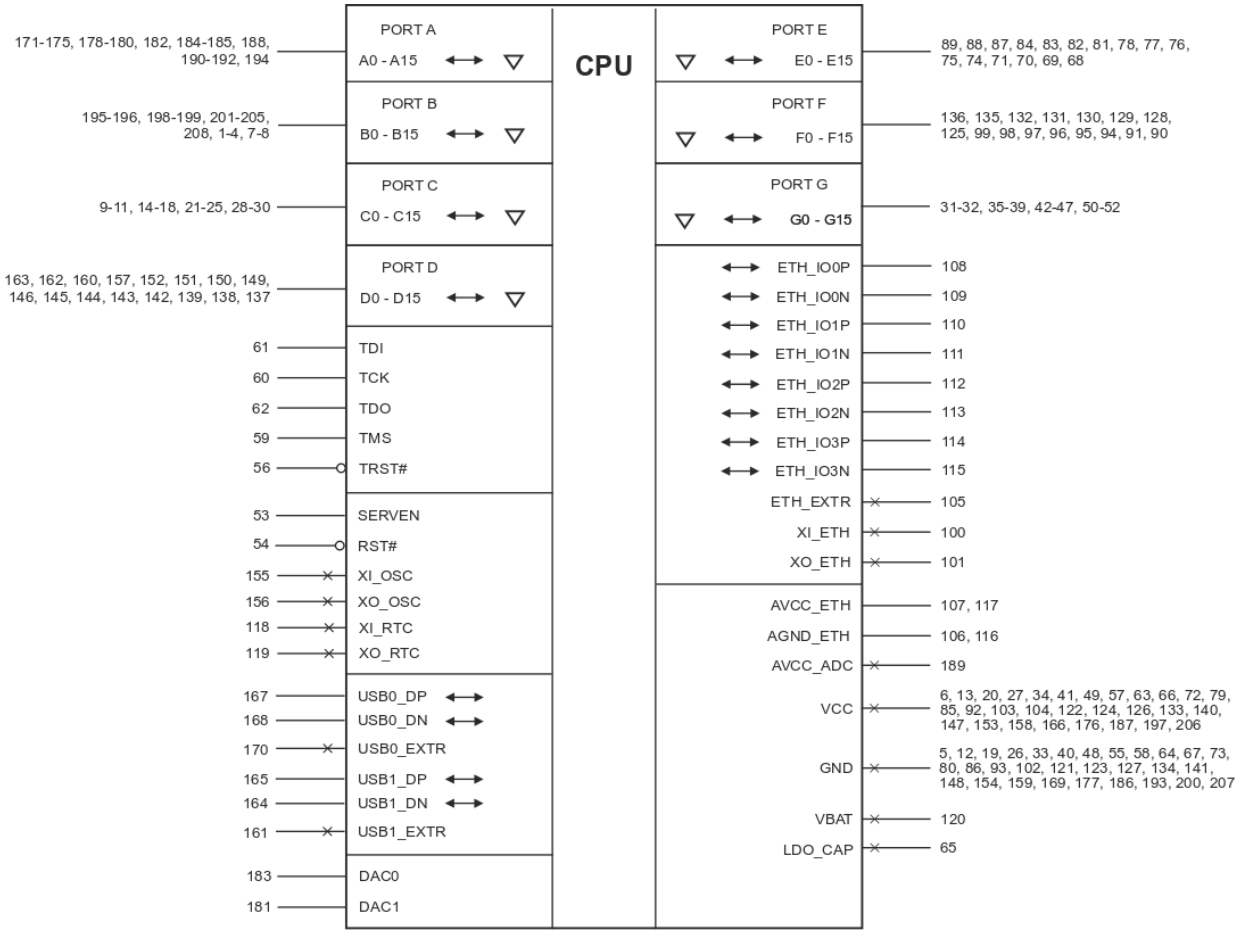


Рисунок 3.2 – Условное графическое обозначение микросхемы

Таблица 3.1 – Функциональное назначение выводов, имеющих альтернативные функции

Номер вывода	Обозначение			Функциональное назначение вывода	Тип вывода
	вывода	Альтернативной функции			
		№	Обозначение		
1	2	3	4	5	6
171	A0			Вход/выход порта A, разряд 0	I/O
		1	SPI0_CLK	Вход/выход синхронизации SPI0	I/O
		2	TMR14_IO	Вход/выход TMR14	I/O
		3	CANFD0_RX	Вход данных CANFD0	I
		4	USB1_RESET	Сигнал сброса PHY USB1	O
		5	TMR0_IO0	Вход/выход 0 TMR0	I/O
			ADC_CH0	Вход АЦП канал 0	I
172	A1			Вход/выход порта A, разряд 1	I/O/Z
		1	SPI0_FSS	Вход/выход выбора устройства SPI0	I/O
		2	TMR15_IO	Вход/выход TMR15	I/O
		3	CANFD0_TX	Выход данных CANFD0	O
		4	USB1_CLK	Вход тактового сигнала PHY USB1	I
		5	TMR0_IO1	Вход/выход 1 TMR0	I/O
			ADC_CH1	Вход АЦП канал 1	I
173	A2			Вход/выход порта A, разряд 2	I/O/Z
		1	SPI0_RX	Вход данных SPI0	I
		2	CLKOUT	Выход тактового сигнала CLKOUT	O
		3	CANFD1_RX	Вход данных CANFD1	I
		4	USB1_XCVRSELECT0	Выход выбора скорости передатчика USB1	O
		5	TMR0_IO2	Вход/выход 2 TMR0	I/O
			ADC_CH2	Вход АЦП канал 2	I
174	A3			Вход/выход порта A, разряд 3	I/O/Z
		1	SPI0_TX	Выход данных SPI0	I
		2	EXTCLK	Вход внешнего тактового сигнала	O
		3	CANFD1_TX	Выход данных CANFD1	I
		4	USB1_XCVRSELECT1	Выход выбора скорости передатчика USB1	O
		5	TMR0_IO3	Вход/выход 3 TMR0	I/O
			ADC_CH3	Вход АЦП канал 3	I
175	A4			Вход/выход порта A, разряд 4	I/O/Z
		1	SPI1_CLK	Вход/выход синхронизации SPI1	I/O
		2	PWM0_A	Выход ШИМ0 канал A	O
		3	CANFD2_RX	Вход данных CANFD2	I
		4	USB1_TERMSELECT	Сигнал выбора подключения FS и HS USB1	O
		5	TMR1_IO0	Вход/выход 0 TMR1	I/O
			ADC_CH4	Вход АЦП канал 4	I
178	A5			Вход/выход порта A, разряд 5	I/O/Z
		1	SPI1_FSS	Вход/выход выбора устройства SPI1	I/O
		2	PWM0_B	Выход ШИМ0 канал B	O
		3	CANFD2_TX	Выход данных CANFD2	O
		4	USB1_SUSPENDM	Сигнал режима Suspend USB1	O
		5	TMR1_IO1	Вход/выход 1 TMR1	I/O
			ADC_CH5	Вход АЦП канал 5	I
179	A6			Вход/выход порта A, разряд 6	I/O/Z
		1	SPI1_RX	Вход данных SPI1	I
		2	PWM1_A	Выход ШИМ1 канал A	O
		3	CANFD3_RX	Вход данных CANFD3	I
		4	USB1_OPMODE0	Сигнал выбора операции USB1, разряд 0	O
		5	TMR1_IO2	Вход/выход 2 TMR1	I/O
			ADC_CH6	Вход АЦП канал 6	I

Продолжение таблицы 3.1

1	2	3	4	5	6
180	A7			Вход/выход порта A, разряд 7	I/O/Z
		1	SPI1_TX	Выход данных SPI1	O
		2	PWM1_B	Выход ШИМ1 канал B	O
		3	CANFD3_TX	Выход данных CANFD3	O
		4	USB1_OPMODE1	Сигнал выбора операции USB1, разряд 1	O
		5	TMR1_IO3	Вход/выход 3 TMR1	I/O
			ADC_CH7	Вход АЦП канал 7	I
182	A8			Вход/выход порта A, разряд 8	I/O/Z
		1	SPI2_CLK	Вход/выход синхронизации SPI2	I/O
		2	PWM2_A	Выход ШИМ2 канал A	O
		3	I2S1_BCLKT	Вход/выход битовой синхронизации передатчика I2S1	I/O
		4	USB1_LINESTATE0	Сигнал состояния шины данных USB1 разряд 0	I
		5	TMR2_IO0	Вход/выход 0 TMR2	I/O
			ADC_CH8	Вход АЦП канал 8	I
184	A9			Вход/выход порта A, разряд 9	I/O/Z
		1	SPI2_FSS	Вход/выход выбора устройства SPI2	I/O
		2	PWM2_B	Выход ШИМ2 канал B	O
		3	I2S1_LRCLKT	Вход/выход выбора канала передатчика I2S1	I/O
		4	USB1_LINESTATE1	Сигнал состояния шины данных USB1 разряд 1	I
		5	TMR2_IO1	Вход/выход 1 TMR2	I/O
			ADC_CH9	Вход АЦП канал 9	I
185	A10			Вход/выход порта A, разряд 10	I/O/Z
		1	SPI2_RX	Вход данных SPI2	I
		2	PWM3_A	Выход ШИМ3 канал A	O
		3	I2S1_BCLKR	Вход/выход битовой синхронизации приемника I2S1	I/O
		4	USB1_TXVALID	Сигнал разрешения передачи данных USB1	O
		5	TMR2_IO2	Вход/выход канала 2 TMR2	I/O
			ADC_CH10	Вход АЦП канал 10	I
188	A11			Вход/выход порта A, разряд 11	I/O/Z
		1	SPI2_TX	Выход данных SPI2	O
		2	PWM3_B	Выход ШИМ3 канал B	O
		3	I2S1_LRCLKR	Вход/выход выбора канала приемника I2S1	I/O
		4	USB1_TXREADY	Сигнал готовности к передаче данных USB1	I
		5	TMR2_IO3	Вход/выход канала 3 TMR2	I/O
			ADC_CH11	Вход АЦП канал 11	I
190	A12			Вход/выход порта A, разряд 12	I/O/Z
		1	SPI3_CLK	Вход/выход синхронизации SPI3	I/O
		2	PWM4_A	Выход ШИМ4 канал A	O
		3	I2S1_SDI	Вход данных I2S1	I
		4	USB1_RXACTIVE	Сигнал активности приема USB1	I
		5	TMR3_IO0	Вход/выход 0 TMR3	I/O
			ADC_CH12	Вход АЦП канал 12	I
191	A13			Вход/выход порта A, разряд 13	I/O/Z
		1	SPI3_FSS	Вход/выход выбора устройства SPI3	I/O
		2	PWM4_B	Выход ШИМ4 канал B	O
		3	I2S1_SDO	Выход данных I2S1	O
		4	USB1_RXVALID	Сигнал индикации получения данных USB1	I
		5	TMR3_IO1	Вход/выход 1 TMR3	I/O
			ADC_CH13	Вход АЦП канал 13	I
192	A14			Вход/выход порта A, разряд 14	I/O/Z
		1	SPI3_RX	Вход данных SPI3	I
		2	PWM5_A	Выход ШИМ5 канал A	O
		3	I2S1_MCLK	Выход основной синхронизации I2S1	O
		4	USB1_RXERROR	Сигнал обнаружения ошибки приема USB1	I
		5	TMR3_IO2	Вход/выход 2 TMR3	I/O
			ADC_CH14	Вход АЦП канал 14	I

Продолжение таблицы 3.1

1	2	3	4	5	6
194	A15			Вход/выход порта A, разряд 15	I/O/Z
		1	SPI3_TX	Выход данных SPI3	O
		2	PWM5_B	Выход ШИМ5 канал B	O
		3	TMR0_IO0	Вход/выход 0 TMR0	I/O
		4	USB1_VBUSVALID	Сигнал допустимого напряжения VBUS USB1	I
		5	TMR3_IO3	Вход/выход 3 TMR3	I/O
			ADC_CH15	Вход АЦП канал 15	I
195	B0			Вход/выход порта B, разряд 0	I/O/Z
		1	CAN0_RX	Вход данных CAN узел 0	I
		2	SPI4_CLK	Вход/выход синхросигнала SPI4	I/O
		3	UART0_RX	Вход данных UART0	I
		4	USB1_CHRGVBUS	Сигнал управления зарядом VBUS USB1	O
		5	TMR4_IO	Вход/выход TMR4	I/O
			ADC_CH16	Вход АЦП канал 16	I
196	B1			Вход/выход порта B, разряд 1	I/O/Z
		1	CAN0_TX	Выход данных CAN узел 0	O
		2	SPI4_FSS	Вход/выход выбора устройства SPI4	I/O
		3	UART0_TX	Выход данных UART0	O
		4	USB1_DISCHRGVBUS	Сигнал управления разрядом VBUS USB1	O
		5	TMR5_IO	Вход/выход TMR5	I/O
			ADC_CH17	Вход АЦП канал 17	I
198	B2			Вход/выход порта B, разряд 2	I/O/Z
		1	CAN1_RX	Вход данных CAN узел 1	I
		2	SPI4_RX	Вход данных SPI4	I
		3	LIN0_RX	Вход данных LIN0	I
		4	USB1_DPPULLDOWN	Сигнал включения pull-down для линии DP USB1	O
		5	TMR6_IO	Вход/выход TMR6	I/O
			ADC_CH18	Вход АЦП канал 18	I
199	B3			Вход/выход порта B, разряд 3	I/O/Z
		1	CAN1_TX	Выход данных CAN узел 1	O
		2	SPI4_TX	Выход данных SPI4	O
		3	LIN0_TX	Выход данных LIN0	O
		4	USB1_DMPULLDOWN	Сигнал включения pull-down для линии DM USB1	O
		5	TMR7_IO	Вход/выход TMR7	I/O
			ADC_CH19	Вход АЦП канал 19	I
201	B4			Вход/выход порта B, разряд 4	I/O/Z
		1	CAN2_RX	Вход данных CAN узел 2	I
		2	SPI5_CLK	Вход/выход синхросигнала SPI5	I/O
		3	UART1_RX	Вход данных UART1	I
		4	USB1_IDPULLUP	Сигнал включения pull-up линии ID USB1	O
		5	TMR8_IO	Вход/выход TMR8	I/O
			ADC_CH20	Вход АЦП канал 20	I
202	B5			Вход/выход порта B, разряд 5	I/O/Z
		1	CAN2_TX	Выход данных CAN узел 2	O
		2	SPI5_FSS	Вход/выход выбора устройства SPI5	I/O
		3	UART1_TX	Выход данных UART1	O
		4	USB1_HOSTDISCONNECT	Сигнал отключения хоста (окончания сессии) USB1	I
		5	TMR9_IO	Вход/выход TMR9	I/O
			ADC_CH21	Вход АЦП канал 21	I
203	B6			Вход/выход порта B, разряд 6	I/O/Z
		1	CAN3_RX	Вход данных CAN узел 3	I
		2	SPI5_RX	Вход данных SPI5	I
		3	LIN1_RX	Вход данных LIN1	I
		4	USB1_DATA0	Шина данных интерфейса USB1, разряд 0	I/O
		5	TMR10_IO	Вход/выход TMR10	I/O
			ADC_CH22	Вход АЦП канал 22	I

Продолжение таблицы 3.1

1	2	3	4	5	6
204	B7			Вход/выход порта B, разряд 7	I/O/Z
		1	CAN3_TX	Выход данных CAN узел 3	O
		2	SPI5_TX	Выход данных SPI5	O
		3	LIN1_TX	Выход данных LIN1	O
		4	USB1_DATA1	Шина данных интерфейса USB1, разряд 1	I/O
		5	TMR11_IO	Вход/выход TMR11	I/O
			ADC_CH23	Вход АЦП канал 23	I
205	B8			Вход/выход порта B, разряд 8	I/O/Z
		1	ACMP0_OUT	Выход аналогового компаратора ACMP0	O
		2	SPI6_CLK	Вход/выход синхросигнала SPI6	I/O
		3	UART2_RX	Вход данных UART2	I
		4	USB1_DATA2	Шина данных интерфейса USB1, разряд 2	I/O
		5	TMR12_IO	Вход/выход TMR12	I/O
			ADC_CH24	Вход АЦП канал 24	I
208	B9			Вход/выход порта B, разряд 9	I/O/Z
		1	ACMP1_OUT	Выход аналогового компаратора ACMP1	O
		2	SPI6_FSS	Вход/выход выбора устройства SPI6	I/O
		3	UART2_TX	Выход данных UART2	O
		4	USB1_DATA3	Шина данных интерфейса USB1, разряд 3	I/O
		5	TMR13_IO	Вход/выход TMR13	I/O
			ADC_CH25	Вход АЦП канал 25	I
1	B10			Вход/выход порта B, разряд 10	I/O/Z
		1	ACMP2_OUT	Выход аналогового компаратора ACMP2	O
		2	SPI6_RX	Вход данных SPI6	I
		3	LIN2_RX	Вход данных LIN2	I
		4	USB1_DATA4	Шина данных интерфейса USB1, разряд 4	I/O
		5	TMR14_IO	Вход/выход TMR14	I/O
			ADC_CH26	Вход АЦП канал 26	I
2	B11			Вход/выход порта B, разряд 11	I/O/Z
		1	ACMP3_OUT	Выход аналогового компаратора ACMP3	O
		2	SPI6_TX	Выход данных SPI6	O
		3	LIN2_TX	Выход данных LIN2	O
		4	USB1_DATA5	Шина данных интерфейса USB1, разряд 5	I/O
		5	TMR15_IO	Вход/выход TMR15	I/O
			ADC_CH27	Вход АЦП канал 27	I
3	B12			Вход/выход порта B, разряд 12	I/O/Z
		1	TMR0_IO0	Вход/выход 0 TMR0	I/O
		2	SPI7_CLK	Вход/выход синхронизации SPI7	I/O
		3	UART3_RX	Вход данных UART3	I
		4	USB1_DATA6	Шина данных интерфейса USB1, разряд 6	I/O
		5	TMR12_IO	Вход/выход TMR12	I/O
			ADC_CH28	Вход АЦП канал 28	I
4	B13			Вход/выход порта B, разряд 13	I/O/Z
		1	TMR1_IO0	Вход/выход 0 TMR1	I/O
		2	SPI7_FSS	Вход/выход выбора устройства SPI7	I/O
		3	UART3_TX	Выход данных UART3	O
		4	USB1_DATA7	Шина данных интерфейса USB1, разряд 7	I/O
		5	TMR13_IO	Вход/выход TMR13	I/O
			ADC_CH29	Вход АЦП канал 29	I
7	B14			Вход/выход порта B, разряд 14	I/O/Z
		1	I2C1_SDA	Вход/выход данных I2C1	I/O
		2	SPI7_RX	Вход данных SPI7	I
		3	LIN3_RX	Вход данных LIN3	I
		4	UART0_CTS	Вход сигнала управления потоком CTS UART0	I
		5	TMR0_IO0	Вход/выход 0 TMR0	I/O
			ADC_CH30	Вход АЦП канал 30	I

Продолжение таблицы 3.1

1	2	3	4	5	6
8	B15			Вход/выход порта B, разряд 15	I/O/Z
		1	I2C1_SCL	Вход/выход синхронизации I2C1	I/O
		2	SPI7_TX	Выход данных SPI7	O
		3	LIN3_TX	Выход данных LIN3	O
		4	UART0_RTS	Выход сигнала управления потоком RTS UART0	O
		5	TMR0_IO1	Вход/выход 1 TMR0	I/O
			ADC_CH31	Вход АЦП канал 31	I
9	C0			Вход/выход порта C, разряд 0	I/O/Z
		1	PWM0_A	Выход ШИМ0 канал A	O
		2	QSPI1_CLK	Вход/выход синхронизации QSPI1	I/O
		3	TMR11_IO	Вход/выход TMR11	I/O
		4	MSC_DO	Выход данных MSC	O
		5	TMR0_IO2	Вход/выход 2 TMR0	I/O
			ADC_CH32	Вход АЦП канал 32	I
10	C1			Вход/выход порта C, разряд 1	I/O/Z
		1	PWM0_B	Выход ШИМ0 канал B	O
		2	QSPI1_FSS	Вход/выход выбора устройства QSPI1	I/O
		3	TMR12_IO	Вход/выход TMR12	I/O
		4	MSC_CLK	Выход тактового сигнала MSC	O
		5	TMR0_IO3	Вход/выход 3 TMR0	I/O
			ADC_CH33	Вход АЦП канал 33	I
11	C2			Вход/выход порта C, разряд 2	I/O/Z
		1	PWM1_A	Выход ШИМ1 канал A	O
		2	QSPI1_IO0	Вход/выход данных QSPI1, разряд 0	I/O
		3	TMR1_IO0	Вход/выход 0 TMR1	I/O
		4	MSC_DI0	Вход данных MSC разряд 0	I
		5	TMR15_IO	Вход/выход TMR15	I/O
			ADC_CH34	Вход АЦП канал 34	I
14	C3			Вход/выход порта C, разряд 3	I/O/Z
		1	PWM1_B	Выход ШИМ1 канал B	O
		2	QSPI1_IO1	Вход/выход данных QSPI1, разряд 1	I/O
		3	TMR2_IO0	Вход/выход 0 TMR2	I/O
		4	MSC_SS0	Выход сигнала выбора ведомого MSC, разряд 0	O
		5	TMR1_IO1	Вход/выход 1 TMR1	I/O
			ADC_CH35	Вход АЦП канал 35	I
15	C4			Вход/выход порта C, разряд 4	I/O/Z
		1	PWM2_A	Выход ШИМ2 канал A	O
		2	QSPI1_IO2	Вход/выход данных QSPI1, разряд 2	I/O
		3	TMR3_IO0	Вход/выход 0 TMR3	I/O
		4	MSC_SS1	Выход сигнала выбора ведомого MSC, разряд 1	O
		5	TMR1_IO2	Вход/выход 2 TMR1	I/O
			ADC_CH36	Вход АЦП канал 36	I
16	C5			Вход/выход порта C, разряд 5	I/O/Z
		1	PWM2_B	Выход ШИМ2 канал B	O
		2	QSPI1_IO3	Вход/выход данных QSPI1, разряд 3	I/O
		3	TMR4_IO	Вход/выход TMR4	I/O
		4	MSC_SS2	Выход сигнала выбора ведомого MSC, разряд 2	O
		5	TMR1_IO3	Вход/выход 3 TMR1	I/O
			ADC_CH37	Вход АЦП канал 37	I
17	C6			Вход/выход порта C, разряд 6	I/O/Z
		1	PWM3_A	Выход ШИМ3 канал A	O
		2	TMR10_IO	Вход/выход TMR10	I/O
		3	ACMP0_OUT	Выход аналогового компаратора ACMP0	O
		4	I2C2_SDA	Вход/выход данных I2C2	I/O
		5	TMR2_IO0	Вход/выход 0 TMR2	I/O
			ADC_CH38	Вход АЦП канал 38	I

Продолжение таблицы 3.1

1	2	3	4	5	6
18	C7			Вход/выход порта C, разряд 7	I/O/Z
		1	PWM3_B	Выход ШИМ3 канал B	O
		2	TMR11_IO	Вход/выход TMR11	I/O
		3	ACMP1_OUT	Выход аналогового компаратора ACMP1	O
		4	I2C2_SCL	Вход/выход синхронизации I2C2	I/O
		5	TMR2_IO1	Вход/выход 1 TMR2	I/O
			ADC_CH39	Вход АЦП канал 39	I
21	C8			Вход/выход порта C, разряд 8	I/O/Z
		1	PWM4_A	Выход ШИМ4 канал A	O
		2	TMR12_IO	Вход/выход TMR12	I/O
		3	SPI0_CLK	Вход/выход синхронизации SPI0	I/O
		4	–	–	–
		5	TMR2_IO2	Вход/выход 2 TMR2	I/O
			ADC_CH40	Вход АЦП канал 40	I
22	C9			Вход/выход порта C, разряд 9	I/O/Z
		1	PWM4_B	Выход ШИМ4 канал B	O
		2	I2S0_BCLKT	Вход/выход битовой синхронизации передатчика I2S0	I/O
		3	SPI0_FSS	Вход/выход выбора устройства SPI0	I/O
		4	–	–	–
		5	TMR2_IO3	Вход/выход 3 TMR2	I/O
			ADC_CH41	Вход АЦП канал 41	I
23	C10			Вход/выход порта C, разряд 10	I/O/Z
		1	PWM5_A	Выход ШИМ5 канал A	O
		2	I2S0_LRCLKT	Вход/выход выбора канала передатчика I2S0	I/O
		3	SPI0_RX	Вход данных SPI0	I
		4	–	–	–
		5	TMR3_IO0	Вход/выход 0 TMR3	I/O
			ADC_CH42	Вход АЦП канал 42	I
24	C11			Вход/выход порта C, разряд 11	I/O/Z
		1	PWM5_B	Выход ШИМ5 канал B	O
		2	I2S0_BCLKR	Вход/выход битовой синхронизации приемника I2S0	I/O
		3	SPI0_TX	Выход данных SPI0	O
		4	–	–	–
		5	TMR3_IO1	Вход/выход 1 TMR3	I/O
			ADC_CH43	Вход АЦП канал 43	I
25	C12			Вход/выход порта C, разряд 12	I/O/Z
		1	PWM_TZ0	Вход сигнала аварии ШИМ, нулевой	I
		2	I2S0_LRCLKR	Вход/выход выбора канала приемника I2S0	I/O
		3	SPI1_CLK	Вход/выход синхронизации SPI1	I/O
		4	–	–	–
		5	TMR3_IO2	Вход/выход 2 TMR3	I/O
			ADC_CH44	Вход АЦП канал 44	I
28	C13			Вход/выход порта C, разряд 13	I/O/Z
		1	PWM_TZ1	Вход сигнала аварии ШИМ, первый	I
		2	I2S0_SDI	Вход данных I2S0	I
		3	SPI1_FSS	Вход/выход выбора устройства SPI1	I/O
		4	USB0_CLK	Вход тактового сигнала интерфейса USB0	I
		5	TMR3_IO3	Вход/выход 3 TMR3	I/O
			ADC_CH45	Вход АЦП канал 45	I
29	C14			Вход/выход порта C, разряд 14	I/O/Z
		1	PWM_TZ2	Вход сигнала аварии ШИМ, второй	I
		2	I2S0_SDO	Выход данных I2S0	O
		3	SPI1_RX	Вход данных SPI1	I
		4	I2C3_SDA	Вход/выход данных I2C3	I/O
		5	TMR4_IO	Вход/выход TMR4	I/O
			ADC_CH46	Вход АЦП канал 46	I

Продолжение таблицы 3.1

1	2	3	4	5	6
30	C15			Вход/выход порта C, разряд 15	I/O/Z
		1	PWM_TZ3	Вход сигнала аварии ШИМ, третий	I
		2	I2S0_MCLK	Выход основной синхронизации I2S0	O
		3	SPI1_TX	Выход данных SPI1	O
		4	I2C3_SCL	Вход/выход синхронизации I2C3	I/O
		5	TMR5_IO	Вход/выход TMR5	I/O
			ADC_CH47	Вход АЦП канал 47	I
163	D0			Вход/выход порта D, разряд 0	I/O/Z
		1	PWM14_A	Выход ШИМ14 канал A	O
		2	I2C1_SDA	Вход/выход данных I2C1	I/O
		3	TMR5_IO	Вход/выход TMR5	I/O
		4	UART1_CTS	Вход сигнала управления потоком CTS UART1	I
		5	TMR6_IO	Вход/выход TMR6	I/O
			ACMP0_INN	Инверсный вход аналогового компаратора 0	I
162	D1			Вход/выход порта D, разряд 1	I/O/Z
		1	PWM14_B	Выход ШИМ14 канал B	O
		2	I2C1_SCL	Вход/выход синхронизации I2C1	I/O
		3	TMR6_IO	Вход/выход TMR6	I/O
		4	UART1_RTS	Выход сигнала управления потоком RTS UART1	O
		5	TMR7_IO	Вход/выход TMR7	I/O
			ACMP0_INP	Прямой вход аналогового компаратора 0	I
160	D2			Вход/выход порта D, разряд 2	I/O/Z
		1	PWM15_A	Выход ШИМ15 канал A	O
		2	I2C0_SDA	Вход/выход данных I2C0	I/O
		3	TMR7_IO	Вход/выход TMR7	I/O
		4	–	–	–
		5	TMR8_IO	Вход/выход TMR8	I/O
			ACMP1_INN	Инверсный вход аналогового компаратора 1	I
157	D3			Вход/выход порта D, разряд 3	I/O/Z
		1	PWM15_B	Выход ШИМ15 канал B	O
		2	I2C0_SCL	Вход/выход синхронизации I2C0	I/O
		3	TMR8_IO	Вход/выход TMR8	I/O
		4	GMII_TXCLK	Вход синхросигнала передачи Ethernet	I
		5	TMR9_IO	Вход/выход TMR9	I/O
			ACMP1_INP	Прямой вход аналогового компаратора 1	I
152	D4			Вход/выход порта D, разряд 4	I/O/Z
		1	PWM_SYNC	Вход синхронизации ШИМ	I
		2	ACMP2_OUT	Выход аналогового компаратора ACMP2	O
		3	TMR9_IO	Вход/выход TMR9	I/O
		4	GMII_GTXCLK	Выход синхросигнала гигабит. передачи Ethernet	O
		5	TMR10_IO	Вход/выход TMR10	I/O
			ACMP2_INN	Инверсный вход аналогового компаратора 2	I
151	D5			Вход/выход порта D, разряд 5	I/O/Z
		1	PWM_TZ4	Вход сигнала аварии ШИМ, четвертый	I
		2	ACMP3_OUT	Выход аналогового компаратора ACMP3	O
		3	TMR10_IO	Вход/выход TMR10	I/O
		4	GMII_RXCLK	Вход синхросигнала приема Ethernet	I
		5	TMR11_IO	Вход/выход TMR11	I/O
			ACMP2_INP	Прямой вход аналогового компаратора 2	I
150	D6			Вход/выход порта D, разряд 6	I/O/Z
		1	EXTCLK	Вход дополнительного тактового сигнала	I
		2	TMR6_IO	Вход/выход TMR6	I/O
		3	QSPI0_CLK	Вход/выход синхронизации QSPI0	I/O
		4	–	–	–
		5	TMR12_IO	Вход/выход TMR12	I/O
			ACMP3_INN	Инверсный вход аналогового компаратора 3	I

Продолжение таблицы 3.1

1	2	3	4	5	6
149	D7			Вход/выход порта D, разряд 7	I/O/Z
		1	CLKOUT	Выход тактового сигнала CLKOUT	O
		2	SDR_CLKOUT	Выход тактового сигнала SDRAM	O
		3	QSPI0_FSS	Вход/выход выбора устройства QSPI0	I/O
		4	–	–	–
		5	TMR14_IO	Вход/выход TMR14	I/O
			ACMP3_INP	Прямой вход аналогового компаратора 3	I
146	D8			Вход/выход порта D, разряд 8	I/O/Z
		1	CANFD0_RX	Вход данных CANFD0	I
		2	MEM_ADDR0	Выход шины адреса внешней памяти, разряд 0	O
		3	QSPI0_IO0	Вход/выход данных QSPI0, разряд 0	I/O
		4	GMII_RXDV	Вход валидности принимаемых данных Ethernet	I
		5	TMR13_IO	Вход/выход TMR13	I/O
145	D9			Вход/выход порта D, разряд 9	I/O/Z
		1	CANFD0_TX	Выход данных CANFD0	O
		2	MEM_ADDR1	Выход шины адреса внешней памяти, разряд 1	O
		3	QSPI0_IO1	Вход/выход данных QSPI0, разряд 1	I/O
		4	GMII_RXER	Вход ошибки приема Ethernet	I
		5	TMR15_IO	Вход/выход TMR15	I/O
144	D10			Вход/выход порта D, разряд 10	I/O/Z
		1	CANFD1_RX	Вход данных CANFD1	I
		2	MEM_ADDR2	Выход шины адреса внешней памяти, разряд 2	O
		3	QSPI0_IO2	Вход/выход данных QSPI0, разряд 2	I/O
		4	GMII_MDIO	Вход/выход управляющих данных Ethernet	I/O
		5	TMR1_IO0	Вход/выход 0 TMR1	I/O
143	D11			Вход/выход порта D, разряд 11	I/O/Z
		1	CANFD1_TX	Выход данных CANFD1	O
		2	MEM_ADDR3	Выход шины адреса внешней памяти, разряд 3	O
		3	QSPI0_IO3	Вход/выход данных QSPI0, разряд 3	I/O
		4	GMII_TXEN	Выход разрешения передачи Ethernet	O
		5	TMR13_IO	Вход/выход TMR13	I/O
142	D12			Вход/выход порта D, разряд 12	I/O/Z
		1	CANFD2_RX	Вход данных CANFD2	I
		2	MEM_ADDR4	Выход шины адреса внешней памяти, разряд 4	O
		3	SPI2_CLK	Вход/выход синхронизации SPI2	I/O
		4	GMII_TXER	Выход ошибки передачи Ethernet	O
		5	TMR0_IO0	Вход/выход 0 TMR0	I/O
139	D13			Вход/выход порта D, разряд 13	I/O/Z
		1	CANFD2_TX	Выход данных CANFD2	O
		2	MEM_ADDR5	Выход шины адреса внешней памяти, разряд 5	O
		3	SPI2_FSS	Вход/выход выбора устройства SPI2	I/O
		4	GMII_CRS	Вход контроля носителя Ethernet	I
		5	TMR0_IO1	Вход/выход 1 TMR0	I/O
138	D14			Вход/выход порта D, разряд 14	I/O/Z
		1	CANFD3_RX	Вход данных CANFD3	I
		2	MEM_ADDR6	Выход шины адреса внешней памяти, разряд 6	O
		3	SPI2_RX	Вход данных SPI2	I
		4	GMII_COL	Вход обнаружения конфликта Ethernet	I
		5	TMR0_IO2	Вход/выход 2 TMR0	I/O
137	D15			Вход/выход порта D, разряд 15	I/O/Z
		1	CANFD3_TX	Выход данных CANFD3	O
		2	MEM_ADDR7	Выход шины адреса внешней памяти, разряд 7	O
		3	SPI2_TX	Выход данных SPI2	O
		4	GMII_MDC	Выход синхросигнала упр. данных Ethernet	O
		5	TMR0_IO3	Вход/выход 3 TMR0	I/O

Продолжение таблицы 3.1

1	2	3	4	5	6
89	E0			Вход/выход порта E, разряд 0	I/O/Z
		1	UART4_RX	Вход данных UART4	I
		2	MEM_ADDR8	Выход шины адреса внешней памяти, разряд 8	O
		3	SPI6_CLK	Вход/выход синхросигнала SPI6	I/O
		4	GMII_TXD0	Выход нулевого бита данных Ethernet	O
		5	TMR1_IO0	Вход/выход 0 TMR1	I/O
88	E1			Вход/выход порта E, разряд 1	I/O/Z
		1	UART4_TX	Выход данных UART4	O
		2	MEM_ADDR9	Выход шины адреса внешней памяти, разряд 9	O
		3	SPI6_FSS	Вход/выход выбора устройства SPI6	I/O
		4	GMII_TXD1	Выход первого бита данных Ethernet	O
		5	TMR1_IO1	Вход/выход 1 TMR1	I/O
87	E2			Вход/выход порта E, разряд 2	I/O/Z
		1	UART5_RX	Вход данных UART5	I
		2	MEM_ADDR10	Выход шины адреса внешней памяти, разряд 10	O
		3	SPI6_RX	Вход данных SPI6	I
		4	GMII_TXD2	Выход второго бита данных Ethernet	O
		5	TMR1_IO2	Вход/выход 2 TMR1	I/O
84	E3			Вход/выход порта E, разряд 3	I/O/Z
		1	UART5_TX	Выход данных UART5	O
		2	MEM_ADDR11	Выход шины адреса внешней памяти, разряд 11	O
		3	SPI6_TX	Выход данных SPI6	O
		4	GMII_TXD3	Выход третьего бита данных Ethernet	O
		5	TMR1_IO3	Вход/выход 3 TMR1	I/O
83	E4			Вход/выход порта E, разряд 4	I/O/Z
		1	LIN4_RX	Вход данных LIN4	I
		2	MEM_ADDR12	Выход шины адреса внешней памяти, разряд 12	O
		3	PWM15_A	Выход ШИМ15 канал A	O
		4	GMII_TXD4	Выход четвертого бита данных Ethernet	O
		5	TMR2_IO0	Вход/выход 0 TMR2	I/O
82	E5			Вход/выход порта E, разряд 5	I/O/Z
		1	LIN4_TX	Выход данных LIN4	O
		2	MEM_ADDR13/	Выход шины адреса внешней памяти, разряд 13/	
			SDR_BA0	Адрес банка SDRAM разряд 0	O
		3	PWM15_B	Выход ШИМ15 канал B	O
		4	GMII_TXD5	Выход пятого бита данных Ethernet	O
		5	TMR2_IO1	Вход/выход 1 TMR2	I/O
81	E6			Вход/выход порта E, разряд 6	I/O/Z
		1	LIN5_RX	Вход данных LIN5	I
		2	MEM_ADDR14/	Выход шины адреса внешней памяти, разряд 14/	
			SDR_BA1	Адрес банка SDRAM разряд 1	O
		3	PWM14_A	Выход ШИМ14 канал A	O
		4	GMII_TXD6	Выход шестого бита данных Ethernet	O
		5	TMR2_IO2	Вход/выход 2 TMR2	I/O
78	E7			Вход/выход порта E, разряд 7	I/O/Z
		1	LIN5_TX	Выход данных LIN5	O
		2	MEM_ADDR15/	Выход шины адреса внешней памяти, разряд 15/	
			SDR_CKE	Включение тактового сигнала SDRAM	O
		3	PWM14_B	Выход ШИМ14 канал B	O
		4	GMII_TXD7	Выход седьмого бита данных Ethernet	O
		5	TMR2_IO3	Вход/выход 3 TMR2	I/O
77	E8			Вход/выход порта E, разряд 8	I/O/Z
		1	UART6_RX	Вход данных UART6	I
		2	MEM_ADDR16/	Выход шины адреса внешней памяти, разряд 16/	
			SDR_WEN	Сигнал WE# SDRAM	O
		3	PWM_TZ5	Вход сигнала аварии ШИМ, пятый	I
		4	GMII_RXD0	Вход нулевого бита данных Ethernet	I
		5	TMR3_IO0	Вход/выход 0 TMR3	I/O

Продолжение таблицы 3.1

1	2	3	4	5	6
76	E9			Вход/выход порта E, разряд 9	I/O/Z
		1	UART6_TX	Выход данных UART6	O
		2	MEM_ADDR17/	Выход шины адреса внешней памяти, разряд 17/	
			SDR_RASN	Сигнал RAS# SDRAM	O
		3	TMR13_IO	Вход/выход TMR13	I/O
		4	GMII_RXD1	Вход первого бита данных Ethernet	I
		5	TMR3_IO1	Вход/выход 1 TMR3	I/O
75	E10			Вход/выход порта E, разряд 10	I/O/Z
		1	UART7_RX	Вход данных UART7	I
		2	MEM_ADDR18/	Выход шины адреса внешней памяти, разряд 18/	
			SDR_CASN	Сигнал CAS# SDRAM	O
		3	TMR14_IO	Вход/выход TMR14	I/O
		4	GMII_RXD2	Вход второго бита данных Ethernet	I
		5	TMR3_IO2	Вход/выход 2 TMR3	I/O
74	E11			Вход/выход порта E, разряд 11	I/O/Z
		1	UART7_TX	Выход данных UART7	O
		2	MEM_ADDR19/	Выход шины адреса внешней памяти, разряд 19/	
			SDR_CSN	Сигнал CS# SDRAM	O
		3	TMR15_IO	Вход/выход TMR15	I/O
		4	GMII_RXD3	Вход третьего бита данных Ethernet	I
		5	TMR3_IO3	Вход/выход 3 TMR3	I/O
71	E12			Вход/выход порта E, разряд 12	I/O/Z
		1	LIN6_RX	Вход данных LIN6	I
		2	MEM_ADDR20/	Выход шины адреса внешней памяти, разряд 20/	
			SDR_DQ0	Маска шины данных SDRAM разряд 0	O
		3	SPI3_CLK	Вход/выход синхронизации SPI3	I/O
		4	GMII_RXD4	Вход четвертого бита данных Ethernet	I
		5	TMR4_IO	Вход/выход TMR4	I/O
70	E13			Вход/выход порта E, разряд 13	I/O/Z
		1	LIN6_TX	Выход данных LIN6	O
		2	MEM_ADDR21/	Выход шины адреса внешней памяти, разряд 21/	
			SDR_DQ1	Маска шины данных SDRAM разряд 1	O
		3	SPI3_FSS	Вход/выход выбора устройства SPI3	I/O
		4	GMII_RXD5	Вход пятого бита данных Ethernet	I
		5	TMR5_IO	Вход/выход TMR5	I/O
69	E14			Вход/выход порта E, разряд 14	I/O/Z
		1	LIN7_RX	Вход данных LIN7	I
		2	MEM_ADDR22/	Выход шины адреса внешней памяти, разряд 22/	
			SDR_DQ2	Маска шины данных SDRAM разряд 2	O
		3	SPI3_RX	Вход данных SPI3	I
		4	GMII_RXD6	Вход шестого бита данных Ethernet	I
		5	TMR6_IO	Вход/выход TMR6	I/O
68	E15			Вход/выход порта E, разряд 15	I/O/Z
		1	LIN7_TX	Выход данных LIN7	O
		2	MEM_ADDR22/	Выход шины адреса внешней памяти, разряд 23/	
			SDR_DQ3	Маска шины данных SDRAM разряд 3	O
		3	SPI3_TX	Выход данных SPI3	O
		4	GMII_RXD7	Вход седьмого бита данных Ethernet	I
		5	TMR7_IO	Вход/выход TMR7	I/O
136	F0			Вход/выход порта F, разряд 0	I/O/Z
		1	QSPI0_CLK	Вход/выход синхронизации QSPI0	I/O
		2	SDR_DATA16	Вход/выход данных SDRAM разряд 16	I/O
		3	LIN4_RX	Вход данных LIN4	I
		4	USB0_DATA0	Шина данных интерфейса USB0, разряд 0	I/O
		5	TMR8_IO	Вход/выход TMR8	I/O

Продолжение таблицы 3.1

1	2	3	4	5	6
135	F1			Вход/выход порта F, разряд 1	I/O/Z
		1	QSPI0_FSS	Вход/выход выбора устройства QSPI0	I/O
		2	SDR_DATA17	Вход/выход данных SDRAM разряд 17	I/O
		3	LIN4_TX	Выход данных LIN4	O
		4	USB0_DATA1	Шина данных интерфейса USB0, разряд 1	I/O
		5	TMR9_IO	Вход/выход TMR9	I/O
132	F2			Вход/выход порта F, разряд 2	I/O/Z
		1	QSPI0_IO0	Вход/выход данных QSPI0, разряд 0	I/O
		2	SDR_DATA18	Вход/выход данных SDRAM разряд 18	I/O
		3	LIN5_RX	Вход данных LIN5	I
		4	USB0_DATA2	Шина данных интерфейса USB0, разряд 2	I/O
		5	TMR10_IO	Вход/выход TMR10	I/O
131	F3			Вход/выход порта F, разряд 3	I/O/Z
		1	QSPI0_IO1	Вход/выход данных QSPI0, разряд 1	I/O
		2	SDR_DATA19	Вход/выход данных SDRAM разряд 19	I/O
		3	LIN5_TX	Выход данных LIN5	O
		4	USB0_DATA3	Шина данных интерфейса USB0, разряд 3	I/O
		5	TMR11_IO	Вход/выход TMR11	I/O
130	F4			Вход/выход порта F, разряд 4	I/O/Z
		1	QSPI0_IO2	Вход/выход данных QSPI0, разряд 2	I/O
		2	EMC_CEN0/	Сигнал включения микросхемы внешней памяти разряд 0/	
			SDR_DATA20	Вход/выход данных SDRAM разряд 20	I/O
		3	UART6_RX	Вход данных UART6	I
		4	USB0_DATA4	Шина данных интерфейса USB0, разряд 4	I/O
		5	TMR12_IO	Вход/выход TMR12	I/O
129	F5			Вход/выход порта F, разряд 5	I/O/Z
		1	QSPI0_IO3	Вход/выход данных QSPI0, разряд 3	I/O
		2	EMC_CEN1/	Сигнал включения микросхемы внешней памяти разряд 1/	
			SDR_DATA21	Вход/выход данных SDRAM разряд 21	I/O
		3	UART6_TX	Выход данных UART6	O
		4	USB0_DATA5	Шина данных интерфейса USB0, разряд 5	I/O
		5	TMR13_IO	Вход/выход TMR13	I/O
128	F6			Вход/выход порта F, разряд 6	I/O/Z
		1	I2C0_SDA	Вход/выход данных I2C0	I/O
		2	EMC_CEN2/	Сигнал включения микросхемы внешней памяти разряд 2/	
			SDR_DATA22	Вход/выход данных SDRAM разряд 22	I/O
		3	UART7_RX	Вход данных UART7	I
		4	USB0_DATA6	Шина данных интерфейса USB0, разряд 6	I/O
		5	TMR14_IO	Вход/выход TMR14	I/O
125	F7			Вход/выход порта F, разряд 7	I/O/Z
		1	I2C0_SCL	Вход/выход синхронизации I2C0	I/O
		2	EMC_CEN3/	Сигнал включения микросхемы внешней памяти разряд 3/	
			SDR_DATA23	Вход/выход данных SDRAM разряд 23	I/O
		3	UART7_TX	Выход данных UART7	O
		4	USB0_DATA7	Шина данных интерфейса USB0, разряд 7	I/O
		5	TMR15_IO	Вход/выход TMR15	I/O
99	F8			Вход/выход порта F, разряд 8	I/O/Z
		1	TMR2_IO0	Вход/выход 0 TMR2	I/O
		2	EMC_CEN4/	Сигнал включения микросхемы внешней памяти разряд 4/	
			SDR_DATA24	Вход/выход данных SDRAM разряд 24	I/O
		3	SPI4_CLK	Вход/выход синхросигнала SPI4	I/O
		4	–	–	–
		5	TMR0_IO0	Вход/выход 0 TMR0	I/O
98	F9			Вход/выход порта F, разряд 9	I/O/Z
		1	TMR3_IO0	Вход/выход 0 TMR3	I/O
		2	EMC_CEN5/	Сигнал включения микросхемы внешней памяти разряд 5/	
			SDR_DATA25	Вход/выход данных SDRAM разряд 25	I/O
		3	SPI4_FSS	Вход/выход выбора устройства SPI4	I/O
		4	USB0_RESET	Сигнал сброса интерфейса USB0	O
		5	TMR0_IO1	Вход/выход 1 TMR0	I/O

Продолжение таблицы 3.1

1	2	3	4	5	6
97	F10			Вход/выход порта F, разряд 10	I/O/Z
		1	TMR4_IO	Вход/выход TMR4	I/O
		2	EMC_CEN6/	Сигнал включения микросхемы внешней памяти разряд 6/	
			SDR_DATA26	Вход/выход данных SDRAM разряд 26	I/O
		3	SPI4_RX	Вход данных SPI4	I
		4	USB0_XCVRSELECT0	Выход выбора скорости передатчика USB0, разряд 0	O
		5	TMR0_IO2	Вход/выход 2 TMR0	I/O
96	F11			Вход/выход порта F, разряд 11	I/O/Z
		1	TMR5_IO	Вход/выход TMR5	I/O
		2	EMC_CEN7/	Сигнал включения микросхемы внешней памяти разряд 7/	
			SDR_DATA27	Вход/выход данных SDRAM разряд 27	I/O
		3	SPI4_TX	Выход данных SPI4	O
		4	USB0_XCVRSELECT1	Выход выбора скорости передатчика USB0, разряд 1	O
		5	TMR0_IO3	Вход/выход 3 TMR0	I/O
95	F12			Вход/выход порта F, разряд 12	I/O/Z
		1	TMR6_IO	Вход/выход TMR6	I/O
		2	EMC_WEN/	Сигнал управления записью во внешнюю память/	
			SDR_DATA28	Вход/выход данных SDRAM разряд 28	I/O
		3	SPI5_CLK	Вход/выход синхросигнала SPI5	I/O
		4	USB0_TERMSELECT	Сигнал выбора подключения FS и HS интерфейса USB	O
		5	TMR1_IO0	Вход/выход 0 TMR1	I/O
94	F13			Вход/выход порта F, разряд 13	I/O/Z
		1	TMR7_IO	Вход/выход TMR7	I/O
		2	EMC_LBN/	Сигнал включения младшего байта внешней памяти/	
			SDR_DATA29	Вход/выход данных SDRAM разряд 29	I/O
		3	SPI5_FSS	Вход/выход выбора устройства SPI5	I/O
		4	USB0_SUSPENDM	Сигнал режима Suspend USB0	O
		5	TMR1_IO1	Вход/выход 1 TMR1	I/O
91	F14			Вход/выход порта F, разряд 14	I/O/Z
		1	TMR8_IO	Вход/выход TMR8	I/O
		2	EMC_UBN/	Сигнал включения старшего байта внешней памяти/	
			SDR_DATA30	Вход/выход данных SDRAM разряд 30	I/O
		3	SPI5_RX	Вход данных SPI5	I
		4	USB0_OPMODE0	Сигнал выбора операции USB0, разряд 0	O
		5	TMR1_IO2	Вход/выход 2 TMR1	I/O
90	F15			Вход/выход порта F, разряд 15	I/O/Z
		1	TMR9_IO	Вход/выход TMR9	I/O
		2	EMC_OEN/	Сигнал включения работы на выход внешней памяти/	
			SDR_DATA31	Вход/выход данных SDRAM разряд 31	I/O
		3	SPI5_TX	Выход данных SPI5	O
		4	USB0_OPMODE1	Сигнал выбора операции USB0, разряд 1	O
		5	TMR1_IO3	Вход/выход 3 TMR1	I/O
31	G0			Вход/выход порта G, разряд 0	I/O/Z
		1	PWM6_A	Выход ШИМ6 канал A	O
		2	MEM_DATA0	Вход/выход шины данных внешней памяти, разряд 0	I/O
		3	CAN0_RX	Вход данных CAN узел 0	I
		4	USB0_LINESTATE0	Сигнал состояния шины данных USB0 разряд 0	I
		5	TMR2_IO0	Вход/выход 0 TMR2	I/O
32	G1			Вход/выход порта G, разряд 1	I/O/Z
		1	PWM6_B	Выход ШИМ6 канал B	O
		2	MEM_DATA1	Вход/выход шины данных внешней памяти, разряд 1	I/O
		3	CAN0_TX	Выход данных CAN узел 0	O
		4	USB0_LINESTATE1	Сигнал состояния шины данных USB0 разряд 1	I
		5	TMR2_IO1	Вход/выход 1 TMR2	I/O

Продолжение таблицы 3.1

1	2	3	4	5	6
35	G2			Вход/выход порта G, разряд 2	I/O/Z
		1	PWM7_A	Выход ШИМ7 канал A	O
		2	MEM_DATA2	Вход/выход шины данных внешней памяти, разряд 2	I/O
		3	CAN1_RX	Вход данных CAN узел 1	I
		4	USB0_TXVALID	Сигнал разрешения передачи данных USB0	O
		5	TMR2_IO2	Вход/выход 2 TMR2	I/O
36	G3			Вход/выход порта G, разряд 3	I/O/Z
		1	PWM7_B	Выход ШИМ7 канал B	O
		2	MEM_DATA3	Вход/выход шины данных внешней памяти, разряд 3	I/O
		3	CAN1_TX	Выход данных CAN узел 1	O
		4	USB0_TXREADY	Сигнал готовности к передаче данных USB0	I
		5	TMR2_IO3	Вход/выход 3 TMR2	I/O
37	G4			Вход/выход порта G, разряд 4	I/O/Z
		1	PWM8_A	Выход ШИМ8 канал A	O
		2	MEM_DATA4	Вход/выход шины данных внешней памяти, разряд 4	I/O
		3	UART4_RX	Вход данных UART4	I
		4	USB0_RXACTIVE	Сигнал активности приема USB0	I
		5	TMR3_IO0	Вход/выход 0 TMR3	I/O
38	G5			Вход/выход порта G, разряд 5	I/O/Z
		1	PWM8_B	Выход ШИМ8 канал B	O
		2	MEM_DATA5	Вход/выход шины данных внешней памяти, разряд 5	I/O
		3	UART4_TX	Выход данных UART4	O
		4	USB0_RXVALID	Сигнал индикации получения данных USB0	I
		5	TMR3_IO1	Вход/выход 1 TMR3	I/O
39	G6			Вход/выход порта G, разряд 6	I/O/Z
		1	PWM9_A	Выход ШИМ9 канал A	O
		2	MEM_DATA6	Вход/выход шины данных внешней памяти, разряд 6	I/O
		3	UART5_RX	Вход данных UART5	I
		4	USB0_RXERROR	Сигнал обнаружения ошибки приема USB0	I
		5	TMR3_IO2	Вход/выход 2 TMR3	I/O
42	G7			Вход/выход порта G, разряд 7	I/O/Z
		1	PWM9_B	Выход ШИМ9 канал B	O
		2	MEM_DATA7	Вход/выход шины данных внешней памяти, разряд 7	I/O
		3	UART5_TX	Выход данных UART5	O
		4	USB0_VBUSVALID	Сигнал допустимого уровня напряжения VBUS USB0	I
		5	TMR3_IO3	Вход/выход 3 TMR3	I/O
43	G8			Вход/выход порта G, разряд 8	I/O/Z
		1	PWM10_A	Выход ШИМ10 канал A	O
		2	MEM_DATA8	Вход/выход шины данных внешней памяти, разряд 8	I/O
		3	CAN2_RX	Вход данных CAN узел 2	I
		4	USB0_CHRGVBUS	Сигнал управления зарядом VBUS USB0	O
		5	TMR4_IO	Вход/выход TMR4	I/O
44	G9			Вход/выход порта G, разряд 9	I/O/Z
		1	PWM10_B	Выход ШИМ10 канал B	O
		2	MEM_DATA9	Вход/выход шины данных внешней памяти, разряд 9	I/O
		3	CAN2_TX	Выход данных CAN узел 2	O
		4	USB0_DISCHRGVBUS	Сигнал управления разрядом VBUS USB0	O
		5	TMR5_IO	Вход/выход TMR5	I/O

Продолжение таблицы 3.1

1	2	3	4	5	6
45	G10			Вход/выход порта G, разряд 10	I/O/Z
		1	PWM11_A	Выход ШИМ11 канал A	O
		2	MEM_DATA10	Вход/выход шины данных внешней памяти, разряд 10	I/O
		3	CAN3_RX	Вход данных CAN узел 3	I
		4	USB0_DPPULLDOWN	Сигнал включения pull-down для линии DP USB0	O
		5	TMR6_IO	Вход/выход TMR6	I/O
46	G11			Вход/выход порта G, разряд 11	I/O/Z
		1	PWM11_B	Выход ШИМ11 канал B	O
		2	MEM_DATA11	Вход/выход шины данных внешней памяти, разряд 11	I/O
		3	CAN3_TX	Выход данных CAN узел 3	O
		4	USB0_DMPULLDOWN	Сигнал включения pull-down для линии DM USB0	O
		5	TMR7_IO	Вход/выход TMR7	I/O
47	G12			Вход/выход порта G, разряд 12	I/O/Z
		1	PWM12_A	Выход ШИМ12 канал A	O
		2	MEM_DATA12	Вход/выход шины данных внешней памяти, разряд 12	I/O
		3	LIN6_RX	Вход данных LIN6	I
		4	USB0_IDPULLUP	Сигнал включения pull-up линии ID USB0	O
		5	TMR8_IO	Вход/выход TMR8	I/O
50	G13			Вход/выход порта G, разряд 13	I/O/Z
		1	PWM12_B	Выход ШИМ12 канал B	O
		2	MEM_DATA13	Вход/выход шины данных внешней памяти, разряд 13	I/O
		3	LIN6_TX	Выход данных LIN6	O
		4	USB0_HOSTDISCONNECT	Сигнал отключения хоста (окончания сессии) USB0	I
		5	TMR9_IO	Вход/выход TMR9	I/O
51	G14			Вход/выход порта G, разряд 14	I/O/Z
		1	PWM13_A	Выход ШИМ13 канал A	O
		2	MEM_DATA14	Вход/выход шины данных внешней памяти, разряд 14	I/O
		3	LIN7_RX	Вход данных LIN7	I
		4	MSC_DI1	Вход данных MSC разряд 1	I
		5	TMR10_IO	Вход/выход TMR10	I/O
52	G15			Вход/выход порта G, разряд 15	I/O/Z
		1	PWM13_B	Выход ШИМ13 канал B	O
		2	MEM_DATA15	Вход/выход шины данных внешней памяти, разряд 15	I/O
		3	LIN7_TX	Выход данных LIN7	O
		4	MSC_SS3	Выход сигнала выбора ведомого MSC, разряд 3	O
		5	TMR11_IO	Вход/выход TMR11	I/O

Примечания:

1) Входы блока АЦП ADC_CH0 – ADC_CH47, а также входы аналоговых компараторов CMPn_INN, CMPn_INP не являются альтернативными функциями и подключены к выводам GPIO постоянно.

2) В графе «Тип вывода» принятые условные обозначения:

- I – вход,
- O – выход,
- Z – третье состояние.

Таблица 3.2 – Функциональное назначение выводов, не имеющих альтернативных функций

Обозначение вывода	Номер вывода	Тип вывода	Функциональное назначение вывода
1	2	3	4
XI_OSC	155	–	Вывод внешнего тактового сигнала / Вывод для подключения кварцевого резонатора
XO_OSC	156	–	Вывод для подключения кварцевого резонатора
XI_RTC	118	–	Вывод для подключения кварцевого резонатора (32 кГц)
XO_RTC	119	–	Вывод для подключения кварцевого резонатора (32 кГц)
RST#	54	I	Вход сигнала сброса
TCK	60	I	Вход тактового сигнала порта JTAG (Pull-down)
TDI	61	I	Вход данных порта JTAG (Pull-up)
TDO	62	O	Выход данных порта JTAG (Pull-up)
TMS	59	I	Вход режима порта JTAG (Pull-up)
TRST#	56	I	Вход сброса порта JTAG (Pull-up)
SERVEN	53	I	Вход сигнала активации сервисного режима (Pull-down)
VBAT	120	–	Вход подключения батарейного питания
LDO_CAP	65	–	Вывод подключения конденсатора емкостью 10 мкФ ESR 50 мОм
AVCC_ADC	189	–	Аналоговое питание АЦП 3,3 В
VCC	6; 13; 20; 27; 34; 41; 49; 57; 63; 66; 72; 79; 85; 92; 103; 104; 122; 124; 126; 133; 140; 147; 153; 158; 166; 176; 187; 197; 206	–	Питание 3,3 В
GND	5; 12; 19; 26; 33; 40; 48; 55; 58; 64; 67; 73; 80; 86; 93; 102; 121; 123; 127; 134; 141; 148; 154; 159; 169; 177; 186; 193; 200; 207	–	Земля

Продолжение таблицы 3.2

1	2	3	4
AVCC_ETH	107; 117	–	Питание Ethernet 3,3 В
AGND_ETH	106; 116	–	Земля Ethernet
ETH_EXTR	105	–	Вывод Ethernet для подключения внешнего резистора номиналом 2,49 кОм 1% на AGND
ETH_IO0P	108	I/O	Вход/выход данных 0 (+) Ethernet
ETH_IO0N	109	I/O	Вход/выход данных 0 (–) Ethernet
ETH_IO1P	110	I/O	Вход/выход данных 1 (+) Ethernet
ETH_IO1N	111	I/O	Вход/выход данных 1 (–) Ethernet
ETH_IO2P	112	I/O	Вход/выход данных 2 (+) Ethernet
ETH_IO2N	113	I/O	Вход/выход данных 2 (–) Ethernet
ETH_IO3P	114	I/O	Вход/выход данных 3 (+) Ethernet
ETH_IO3N	115	I/O	Вход/выход данных 3 (–) Ethernet
XI_ETH	100		Вывод для подключения кварцевого резонатора с частотой 25 МГц к блоку Ethernet
XO_ETH	101		Вывод для подключения кварцевого резонатора с частотой 25 МГц к блоку Ethernet
DAC0	183	O	Выход канала ЦАП0
DAC1	181	O	Выход канала ЦАП1
USB0_DP	167	I/O	Вход/выход «USB0 Dp»
USB0_DN	168	I/O	Вход/выход «USB0 Dn»
USB0_EXTR	170	-	Вывод USB0 для подключения внешнего резистора номиналом 6 кОм 1% на GND
USB1_DP	165	I/O	Вход/выход «USB1 Dp»
USB1_DN	164	I/O	Вход/выход «USB1 Dn»
USB1_EXTR	161	-	Вывод USB1 для подключения внешнего резистора номиналом 6 кОм 1% на GND

Примечания:

1) В графе «Тип вывода» принятые условные обозначения:
   - I – вход,
   - O – выход,
   - Z – третье состояние

2) Сопротивление внутренних резисторов подтяжки Pull-up в диапазоне от 27 кОм до 65 кОм, типовое – 40 кОм. Сопротивление внутренних резисторов подтяжки Pull-down в диапазоне от 31 кОм до 84 кОм, типовое – 48 кОм.

Микросхемы К1921ВГ1Т выполнены в пластиковом корпусе LQFP-208.

Масса микросхемы – не более 2,25 г.

Габаритный чертеж корпуса микросхемы приведен в Приложении Г.

УГО и назначение выводов микроконтроллера К1921ВГ10Я

Условное графическое обозначение микроконтроллера К1921ВГ10Я приведено на рисунке 3.3.

Функциональное назначение выводов микроконтроллера приведено в таблицах 3.3 – 3.4, в которых приняты обозначения: I – вход, O – выход, I/O – вход/выход, Z – третье состояние.

Примечание – Альтернативные функции выводов портов A, B, C, D, E, F, G указаны в таблице 3.3.

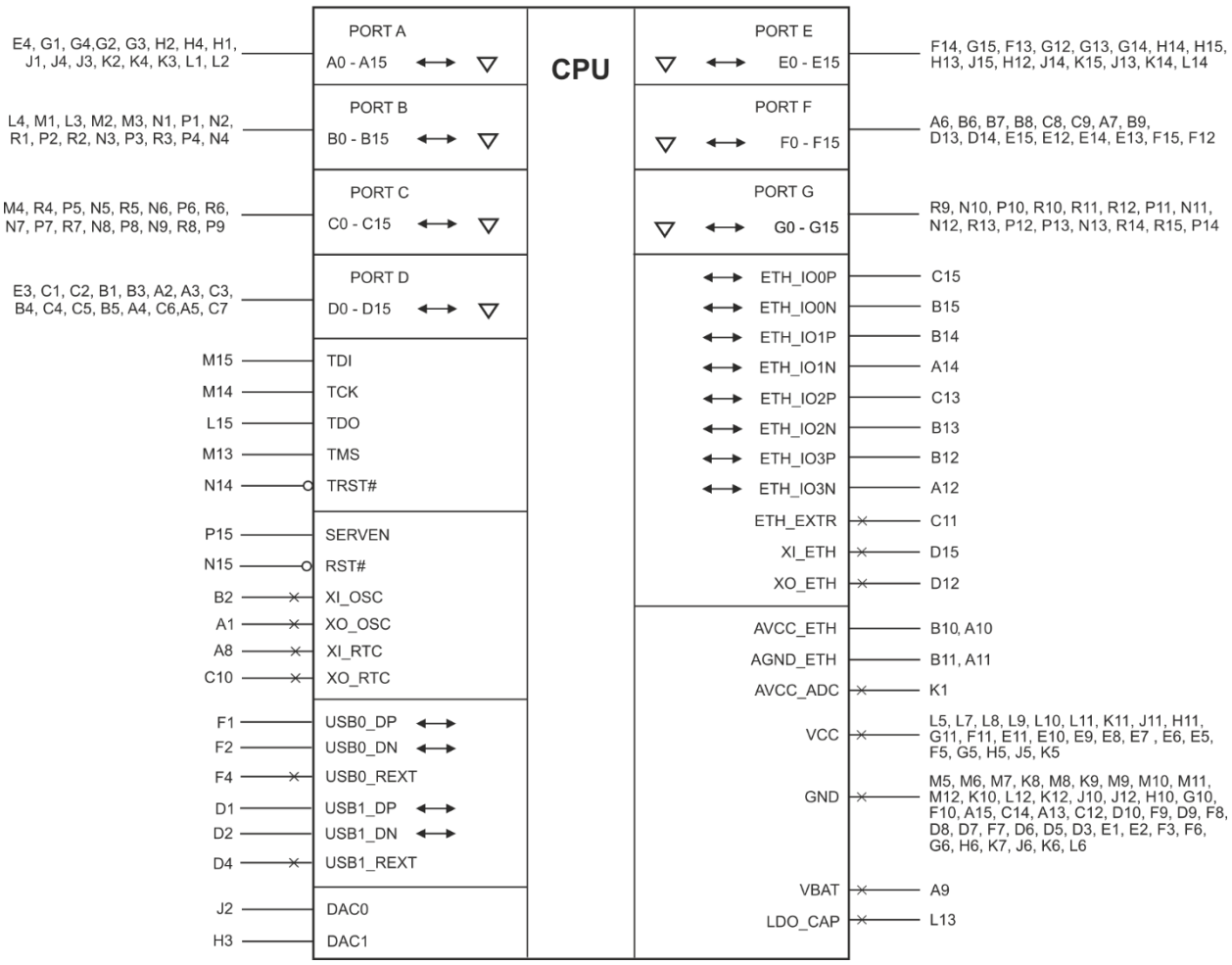


Рисунок 3.3 – Условное графическое обозначение микросхемы К1921ВГ10Я

Таблица 3.3 – Функциональное назначение выводов, имеющих альтернативные функции

Номер вывода	Обозначение			Функциональное назначение вывода	Тип вывода
	вывода	Альтернативной функции			
		№	Обозначение		
1	2	3	4	5	6
E4	A0			Вход/выход порта A, разряд 0	I/O/Z
		1	SPI0_CLK	Вход/выход синхронизации SPI0	I/O
		2	TMR14_IO	Вход/выход TMR14	I/O
		3	CANFD0_RX	Вход данных CANFD0	I
		4	USB1_RESET	Сигнал сброса PHY USB1	O
		5	TMR0_IO0	Вход/выход 0 TMR0	I/O
			ADC_CH0	Вход АЦП канал 0	I
G1	A1			Вход/выход порта A, разряд 1	I/O/Z
		1	SPI0_FSS	Вход/выход выбора устройства SPI0	I/O
		2	TMR15_IO	Вход/выход TMR15	I/O
		3	CANFD0_TX	Выход данных CANFD0	O
		4	USB1_CLK	Вход тактового сигнала PHY USB1	I
		5	TMR0_IO1	Вход/выход 1 TMR0	I/O
			ADC_CH1	Вход АЦП канал 1	I
G4	A2			Вход/выход порта A, разряд 2	I/O/Z
		1	SPI0_RX	Вход данных SPI0	I
		2	CLKOUT	Выход тактового сигнала CLKOUT	O
		3	CANFD1_RX	Вход данных CANFD1	I
		4	USB1_XCVRSELECT0	Выход выбора скорости передатчика USB1	O
		5	TMR0_IO2	Вход/выход 2 TMR0	I/O
			ADC_CH2	Вход АЦП канал 2	I
G2	A3			Вход/выход порта A, разряд 3	I/O/Z
		1	SPI0_TX	Выход данных SPI0	I
		2	EXTCLK	Вход внешнего тактового сигнала	O
		3	CANFD1_TX	Выход данных CANFD1	I
		4	USB1_XCVRSELECT1	Выход выбора скорости передатчика USB1	O
		5	TMR0_IO3	Вход/выход 3 TMR0	I/O
			ADC_CH3	Вход АЦП канал 3	I
G3	A4			Вход/выход порта A, разряд 4	I/O/Z
		1	SPI1_CLK	Вход/выход синхронизации SPI1	I/O
		2	PWM0_A	Выход ШИМ0 канал A	O
		3	CANFD2_RX	Вход данных CANFD2	I
		4	USB1_TERMSELECT	Сигнал выбора подключения FS и HS USB1	O
		5	TMR1_IO0	Вход/выход 0 TMR1	I/O
			ADC_CH4	Вход АЦП канал 4	I
H2	A5			Вход/выход порта A, разряд 5	I/O/Z
		1	SPI1_FSS	Вход/выход выбора устройства SPI1	I/O
		2	PWM0_B	Выход ШИМ0 канал B	O
		3	CANFD2_TX	Выход данных CANFD2	O
		4	USB1_SUSPENDM	Сигнал режима Suspend USB1	O
		5	TMR1_IO1	Вход/выход 1 TMR1	I/O
			ADC_CH5	Вход АЦП канал 5	I
H4	A6			Вход/выход порта A, разряд 6	I/O/Z
		1	SPI1_RX	Вход данных SPI1	I
		2	PWM1_A	Выход ШИМ1 канал A	O
		3	CANFD3_RX	Вход данных CANFD3	I
		4	USB1_OPMODE0	Сигнал выбора операции USB1, разряд 0	O
		5	TMR1_IO2	Вход/выход 2 TMR1	I/O
			ADC_CH6	Вход АЦП канал 6	I

Продолжение таблицы 3.3

1	2	3	4	5	6
H1	A7			Вход/выход порта A, разряд 7	I/O/Z
		1	SPI1_TX	Выход данных SPI1	O
		2	PWM1_B	Выход ШИМ1 канал B	O
		3	CANFD3_TX	Выход данных CANFD3	O
		4	USB1_OPMODE1	Сигнал выбора операции USB1, разряд 1	O
		5	TMR1_IO3	Вход/выход 3 TMR1	I/O
			ADC_CH7	Вход АЦП канал 7	I
J1	A8			Вход/выход порта A, разряд 8	I/O/Z
		1	SPI2_CLK	Вход/выход синхронизации SPI2	I/O
		2	PWM2_A	Выход ШИМ2 канал A	O
		3	I2S1_BCLKT	Вход/выход битовой синхронизации передатчика I2S1	I/O
		4	USB1_LINESTATE0	Сигнал состояния шины данных USB1 разряд 0	I
		5	TMR2_IO0	Вход/выход 0 TMR2	I/O
			ADC_CH8	Вход АЦП канал 8	I
J4	A9			Вход/выход порта A, разряд 9	I/O/Z
		1	SPI2_FSS	Вход/выход выбора устройства SPI2	I/O
		2	PWM2_B	Выход ШИМ2 канал B	O
		3	I2S1_LRCLKT	Вход/выход выбора канала передатчика I2S1	I/O
		4	USB1_LINESTATE1	Сигнал состояния шины данных USB1 разряд 1	I
		5	TMR2_IO1	Вход/выход 1 TMR2	I/O
			ADC_CH9	Вход АЦП канал 9	I
J3	A10			Вход/выход порта A, разряд 10	I/O/Z
		1	SPI2_RX	Вход данных SPI2	I
		2	PWM3_A	Выход ШИМ3 канал A	O
		3	I2S1_BCLKR	Вход/выход битовой синхронизации приемника I2S1	I/O
		4	USB1_TXVALID	Сигнал разрешения передачи данных USB1	O
		5	TMR2_IO2	Вход/выход канала 2 TMR2	I/O
			ADC_CH10	Вход АЦП канал 10	I
K2	A11			Вход/выход порта A, разряд 11	I/O/Z
		1	SPI2_TX	Выход данных SPI2	O
		2	PWM3_B	Выход ШИМ3 канал B	O
		3	I2S1_LRCLKR	Вход/выход выбора канала приемника I2S1	I/O
		4	USB1_TXREADY	Сигнал готовности к передаче данных USB1	I
		5	TMR2_IO3	Вход/выход канала 3 TMR2	I/O
			ADC_CH11	Вход АЦП канал 11	I
K4	A12			Вход/выход порта A, разряд 12	I/O/Z
		1	SPI3_CLK	Вход/выход синхронизации SPI3	I/O
		2	PWM4_A	Выход ШИМ4 канал A	O
		3	I2S1_SDI	Вход данных I2S1	I
		4	USB1_RXACTIVE	Сигнал активности приема USB1	I
		5	TMR3_IO0	Вход/выход 0 TMR3	I/O
			ADC_CH12	Вход АЦП канал 12	I
K3	A13			Вход/выход порта A, разряд 13	I/O/Z
		1	SPI3_FSS	Вход/выход выбора устройства SPI3	I/O
		2	PWM4_B	Выход ШИМ4 канал B	O
		3	I2S1_SDO	Выход данных I2S1	O
		4	USB1_RXVALID	Сигнал индикации получения данных USB1	I
		5	TMR3_IO1	Вход/выход 1 TMR3	I/O
			ADC_CH13	Вход АЦП канал 13	I
L1	A14			Вход/выход порта A, разряд 14	I/O/Z
		1	SPI3_RX	Вход данных SPI3	I
		2	PWM5_A	Выход ШИМ5 канал A	O
		3	I2S1_MCLK	Выход основной синхронизации I2S1	O
		4	USB1_RXERROR	Сигнал обнаружения ошибки приема USB1	I
		5	TMR3_IO2	Вход/выход 2 TMR3	I/O
			ADC_CH14	Вход АЦП канал 14	I

Продолжение таблицы 3.3

1	2	3	4	5	6
L2	A15			Вход/выход порта A, разряд 15	I/O/Z
		1	SPI3_TX	Выход данных SPI3	O
		2	PWM5_B	Выход ШИМ5 канал B	O
		3	TMR0_IO0	Вход/выход 0 TMR0	I/O
		4	USB1_VBUSVALID	Сигнал допустимого уровня VBUS USB1	I
		5	TMR3_IO3	Вход/выход 3 TMR3	I/O
			ADC_CH15	Вход АЦП канал 15	I
L4	B0			Вход/выход порта B, разряд 0	I/O/Z
		1	CAN0_RX	Вход данных CAN узел 0	I
		2	SPI4_CLK	Вход/выход синхросигнала SPI4	I/O
		3	UART0_RX	Вход данных UART0	I
		4	USB1_CHRGVBUS	Сигнал управления зарядом VBUS USB1	O
		5	TMR4_IO	Вход/выход TMR4	I/O
			ADC_CH16	Вход АЦП канал 16	I
M1	B1			Вход/выход порта B, разряд 1	I/O/Z
		1	CAN0_TX	Выход данных CAN узел 0	O
		2	SPI4_FSS	Вход/выход выбора устройства SPI4	I/O
		3	UART0_TX	Выход данных UART0	O
		4	USB1_DISCHRGVBUS	Сигнал управления разрядом VBUS USB1	O
		5	TMR5_IO	Вход/выход TMR5	I/O
			ADC_CH17	Вход АЦП канал 17	I
L3	B2			Вход/выход порта B, разряд 2	I/O/Z
		1	CAN1_RX	Вход данных CAN узел 1	I
		2	SPI4_RX	Вход данных SPI4	I
		3	LIN0_RX	Вход данных LIN0	I
		4	USB1_DPPULLDOWN	Сигнал включения pull-down для линии DP USB1	O
		5	TMR6_IO	Вход/выход TMR6	I/O
			ADC_CH18	Вход АЦП канал 18	I
M2	B3			Вход/выход порта B, разряд 3	I/O/Z
		1	CAN1_TX	Выход данных CAN узел 1	O
		2	SPI4_TX	Выход данных SPI4	O
		3	LIN0_TX	Выход данных LIN0	O
		4	USB1_DMPULLDOWN	Сигнал включения pull-down для линии DM USB1	O
		5	TMR7_IO	Вход/выход TMR7	I/O
			ADC_CH19	Вход АЦП канал 19	I
M3	B4			Вход/выход порта B, разряд 4	I/O/Z
		1	CAN2_RX	Вход данных CAN узел 2	I
		2	SPI5_CLK	Вход/выход синхросигнала SPI5	I/O
		3	UART1_RX	Вход данных UART1	I
		4	USB1_IDPULLUP	Сигнал включения pull-up линии ID USB1	O
		5	TMR8_IO	Вход/выход TMR8	I/O
			ADC_CH20	Вход АЦП канал 20	I
N1	B5			Вход/выход порта B, разряд 5	I/O/Z
		1	CAN2_TX	Выход данных CAN узел 2	O
		2	SPI5_FSS	Вход/выход выбора устройства SPI5	I/O
		3	UART1_TX	Выход данных UART1	O
		4	USB1_HOSTDISCONNECT	Сигнал отключения хоста (окончания сессии) USB1	I
		5	TMR9_IO	Вход/выход TMR9	I/O
			ADC_CH21	Вход АЦП канал 21	I
P1	B6			Вход/выход порта B, разряд 6	I/O/Z
		1	CAN3_RX	Вход данных CAN узел 3	I
		2	SPI5_RX	Вход данных SPI5	I
		3	LIN1_RX	Вход данных LIN1	I
		4	USB1_DATA0	Шина данных интерфейса USB1, разряд 0	I/O
		5	TMR10_IO	Вход/выход TMR10	I/O
			ADC_CH22	Вход АЦП канал 22	I

Продолжение таблицы 3.3

1	2	3	4	5	6
N2	B7			Вход/выход порта B, разряд 7	I/O/Z
		1	CAN3_TX	Выход данных CAN узел 3	O
		2	SPI5_TX	Выход данных SPI5	O
		3	LIN1_TX	Выход данных LIN1	O
		4	USB1_DATA1	Шина данных интерфейса USB1, разряд 1	I/O
		5	TMR11_IO	Вход/выход TMR11	I/O
			ADC_CH23	Вход АЦП канал 23	I
R1	B8			Вход/выход порта B, разряд 8	I/O/Z
		1	ACMP0_OUT	Выход аналогового компаратора ACMP0	O
		2	SPI6_CLK	Вход/выход синхросигнала SPI6	I/O
		3	UART2_RX	Вход данных UART2	I
		4	USB1_DATA2	Шина данных интерфейса USB1, разряд 2	I/O
		5	TMR12_IO	Вход/выход TMR12	I/O
			ADC_CH24	Вход АЦП канал 24	I
P2	B9			Вход/выход порта B, разряд 9	I/O/Z
		1	ACMP1_OUT	Выход аналогового компаратора ACMP1	O
		2	SPI6_FSS	Вход/выход выбора устройства SPI6	I/O
		3	UART2_TX	Выход данных UART2	O
		4	USB1_DATA3	Шина данных интерфейса USB1, разряд 3	I/O
		5	TMR13_IO	Вход/выход TMR13	I/O
			ADC_CH25	Вход АЦП канал 25	I
R2	B10			Вход/выход порта B, разряд 10	I/O/Z
		1	ACMP2_OUT	Выход аналогового компаратора ACMP2	O
		2	SPI6_RX	Вход данных SPI6	I
		3	LIN2_RX	Вход данных LIN2	I
		4	USB1_DATA4	Шина данных интерфейса USB1, разряд 4	I/O
		5	TMR14_IO	Вход/выход TMR14	I/O
			ADC_CH26	Вход АЦП канал 26	I
N3	B11			Вход/выход порта B, разряд 11	I/O/Z
		1	ACMP3_OUT	Выход аналогового компаратора ACMP3	O
		2	SPI6_TX	Выход данных SPI6	O
		3	LIN2_TX	Выход данных LIN2	O
		4	USB1_DATA5	Шина данных интерфейса USB1, разряд 5	I/O
		5	TMR15_IO	Вход/выход TMR15	I/O
			ADC_CH27	Вход АЦП канал 27	I
P3	B12			Вход/выход порта B, разряд 12	I/O/Z
		1	TMR0_IO0	Вход/выход 0 TMR0	I/O
		2	SPI7_CLK	Вход/выход синхронизации SPI7	I/O
		3	UART3_RX	Вход данных UART3	I
		4	USB1_DATA6	Шина данных интерфейса USB1, разряд 6	I/O
		5	TMR12_IO	Вход/выход TMR12	I/O
			ADC_CH28	Вход АЦП канал 28	I
R3	B13			Вход/выход порта B, разряд 13	I/O/Z
		1	TMR1_IO0	Вход/выход 0 TMR1	I/O
		2	SPI7_FSS	Вход/выход выбора устройства SPI7	I/O
		3	UART3_TX	Выход данных UART3	O
		4	USB1_DATA7	Шина данных интерфейса USB1, разряд 7	I/O
		5	TMR13_IO	Вход/выход TMR13	I/O
			ADC_CH29	Вход АЦП канал 29	I
P4	B14			Вход/выход порта B, разряд 14	I/O/Z
		1	I2C1_SDA	Вход/выход данных I2C1	I/O
		2	SPI7_RX	Вход данных SPI7	I
		3	LIN3_RX	Вход данных LIN3	I
		4	UART0_CTS	Вход сигнала управления потоком CTS UART0	I
		5	TMR0_IO0	Вход/выход 0 TMR0	I/O
			ADC_CH30	Вход АЦП канал 30	I

Продолжение таблицы 3.3

1	2	3	4	5	6
N4	B15			Вход/выход порта B, разряд 15	I/O/Z
		1	I2C1_SCL	Вход/выход синхронизации I2C1	I/O
		2	SPI7_TX	Выход данных SPI7	O
		3	LIN3_TX	Выход данных LIN3	O
		4	UART0_RTS	Выход сигнала управления потоком RTS UART0	O
		5	TMR0_IO1	Вход/выход 1 TMR0	I/O
			ADC_CH31	Вход АЦП канал 31	I
M4	C0			Вход/выход порта C, разряд 0	I/O/Z
		1	PWM0_A	Выход ШИМ0 канал A	O
		2	QSPI1_CLK	Вход/выход синхронизации QSPI1	I/O
		3	TMR11_IO	Вход/выход TMR11	I/O
		4	MSC_DO	Выход данных MSC	O
		5	TMR0_IO2	Вход/выход 2 TMR0	I/O
			ADC_CH32	Вход АЦП канал 32	I
R4	C1			Вход/выход порта C, разряд 1	I/O/Z
		1	PWM0_B	Выход ШИМ0 канал B	O
		2	QSPI1_FSS	Вход/выход выбора устройства QSPI1	I/O
		3	TMR12_IO	Вход/выход TMR12	I/O
		4	MSC_CLK	Выход тактового сигнала MSC	O
		5	TMR0_IO3	Вход/выход 3 TMR0	I/O
			ADC_CH33	Вход АЦП канал 33	I
P5	C2			Вход/выход порта C, разряд 2	I/O/Z
		1	PWM1_A	Выход ШИМ1 канал A	O
		2	QSPI1_IO0	Вход/выход данных QSPI1, разряд 0	I/O
		3	TMR1_IO0	Вход/выход 0 TMR1	I/O
		4	MSC_DI0	Вход данных MSC разряд 0	I
		5	TMR15_IO	Вход/выход TMR15	I/O
			ADC_CH34	Вход АЦП канал 34	I
N5	C3			Вход/выход порта C, разряд 3	I/O/Z
		1	PWM1_B	Выход ШИМ1 канал B	O
		2	QSPI1_IO1	Вход/выход данных QSPI1, разряд 1	I/O
		3	TMR2_IO0	Вход/выход 0 TMR2	I/O
		4	MSC_SS0	Выход сигнала выбора ведомого MSC, разряд 0	O
		5	TMR1_IO1	Вход/выход 1 TMR1	I/O
			ADC_CH35	Вход АЦП канал 35	I
R5	C4			Вход/выход порта C, разряд 4	I/O/Z
		1	PWM2_A	Выход ШИМ2 канал A	O
		2	QSPI1_IO2	Вход/выход данных QSPI1, разряд 2	I/O
		3	TMR3_IO0	Вход/выход 0 TMR3	I/O
		4	MSC_SS1	Выход сигнала выбора ведомого MSC, разряд 1	O
		5	TMR1_IO2	Вход/выход 2 TMR1	I/O
			ADC_CH36	Вход АЦП канал 36	I
N6	C5			Вход/выход порта C, разряд 5	I/O/Z
		1	PWM2_B	Выход ШИМ2 канал B	O
		2	QSPI1_IO3	Вход/выход данных QSPI1, разряд 3	I/O
		3	TMR4_IO	Вход/выход TMR4	I/O
		4	MSC_SS2	Выход сигнала выбора ведомого MSC, разряд 2	O
		5	TMR1_IO3	Вход/выход 3 TMR1	I/O
			ADC_CH37	Вход АЦП канал 37	I
P6	C6			Вход/выход порта C, разряд 6	I/O/Z
		1	PWM3_A	Выход ШИМ3 канал A	O
		2	TMR10_IO	Вход/выход TMR10	I/O
		3	ACMP0_OUT	Выход аналогового компаратора ACMP0	O
		4	I2C2_SDA	Вход/выход данных I2C2	I/O
		5	TMR2_IO0	Вход/выход 0 TMR2	I/O
			ADC_CH38	Вход АЦП канал 38	I

Продолжение таблицы 3.3

1	2	3	4	5	6
R6	C7			Вход/выход порта C, разряд 7	I/O/Z
		1	PWM3_B	Выход ШИМ3 канал B	O
		2	TMR11_IO	Вход/выход TMR11	I/O
		3	ACMP1_OUT	Выход аналогового компаратора ACMP1	O
		4	I2C2_SCL	Вход/выход синхронизации I2C2	I/O
		5	TMR2_IO1	Вход/выход 1 TMR2	I/O
			ADC_CH39	Вход АЦП канал 39	I
N7	C8			Вход/выход порта C, разряд 8	I/O/Z
		1	PWM4_A	Выход ШИМ4 канал A	O
		2	TMR12_IO	Вход/выход TMR12	I/O
		3	SPI0_CLK	Вход/выход синхронизации SPI0	I/O
		4	–	–	–
		5	TMR2_IO2	Вход/выход 2 TMR2	I/O
			ADC_CH40	Вход АЦП канал 40	I
P7	C9			Вход/выход порта C, разряд 9	I/O/Z
		1	PWM4_B	Выход ШИМ4 канал B	O
		2	I2S0_BCLKT	Вход/выход битовой синхронизации передатчика I2S0	I/O
		3	SPI0_FSS	Вход/выход выбора устройства SPI0	I/O
		4	–	–	–
		5	TMR2_IO3	Вход/выход 3 TMR2	I/O
			ADC_CH41	Вход АЦП канал 41	I
R7	C10			Вход/выход порта C, разряд 10	I/O/Z
		1	PWM5_A	Выход ШИМ5 канал A	O
		2	I2S0_LRCLKT	Вход/выход выбора канала передатчика I2S0	I/O
		3	SPI0_RX	Вход данных SPI0	I
		4	–	–	–
		5	TMR3_IO0	Вход/выход 0 TMR3	I/O
			ADC_CH42	Вход АЦП канал 42	I
N8	C11			Вход/выход порта C, разряд 11	I/O/Z
		1	PWM5_B	Выход ШИМ5 канал B	O
		2	I2S0_BCLKR	Вход/выход битовой синхронизации приемника I2S0	I/O
		3	SPI0_TX	Выход данных SPI0	O
		4	–	–	–
		5	TMR3_IO1	Вход/выход 1 TMR3	I/O
			ADC_CH43	Вход АЦП канал 43	I
P8	C12			Вход/выход порта C, разряд 12	I/O/Z
		1	PWM_TZ0	Вход сигнала аварии ШИМ, нулевой	I
		2	I2S0_LRCLKR	Вход/выход выбора канала приемника I2S0	I/O
		3	SPI1_CLK	Вход/выход синхронизации SPI1	I/O
		4	–	–	–
		5	TMR3_IO2	Вход/выход 2 TMR3	I/O
			ADC_CH44	Вход АЦП канал 44	I
N9	C13			Вход/выход порта C, разряд 13	I/O/Z
		1	PWM_TZ1	Вход сигнала аварии ШИМ, первый	I
		2	I2S0_SDI	Вход данных I2S0	I
		3	SPI1_FSS	Вход/выход выбора устройства SPI1	I/O
		4	USB0_CLK	Вход тактового сигнала интерфейса USB0	I
		5	TMR3_IO3	Вход/выход 3 TMR3	I/O
			ADC_CH45	Вход АЦП канал 45	I
R8	C14			Вход/выход порта C, разряд 14	I/O/Z
		1	PWM_TZ2	Вход сигнала аварии ШИМ, второй	I
		2	I2S0_SDO	Выход данных I2S0	O
		3	SPI1_RX	Вход данных SPI1	I
		4	I2C3_SDA	Вход/выход данных I2C3	I/O
		5	TMR4_IO	Вход/выход TMR4	I/O
			ADC_CH46	Вход АЦП канал 46	I

Продолжение таблицы 3.3

1	2	3	4	5	6
P9	C15			Вход/выход порта C, разряд 15	I/O/Z
		1	PWM_TZ3	Вход сигнала аварии ШИМ, третий	I
		2	I2S0_MCLK	Выход основной синхронизации I2S0	O
		3	SPI1_TX	Выход данных SPI1	O
		4	I2C3_SCL	Вход/выход синхронизации I2C3	I/O
		5	TMR5_IO	Вход/выход TMR5	I/O
			ADC_CH47	Вход АЦП канал 47	I
E3	D0			Вход/выход порта D, разряд 0	I/O/Z
		1	PWM14_A	Выход ШИМ14 канал A	O
		2	I2C1_SDA	Вход/выход данных I2C1	I/O
		3	TMR5_IO	Вход/выход TMR5	I/O
		4	UART1_CTS	Вход сигнала управления потоком CTS UART1	I
		5	TMR6_IO	Вход/выход TMR6	I/O
			ACMP0_INN	Инверсный вход аналогового компаратора 0	I
C1	D1			Вход/выход порта D, разряд 1	I/O/Z
		1	PWM14_B	Выход ШИМ14 канал B	O
		2	I2C1_SCL	Вход/выход синхронизации I2C1	I/O
		3	TMR6_IO	Вход/выход TMR6	I/O
		4	UART1_RTS	Выход сигнала управления потоком RTS UART1	O
		5	TMR7_IO	Вход/выход TMR7	I/O
			ACMP0_INP	Прямой вход аналогового компаратора 0	I
C2	D2			Вход/выход порта D, разряд 2	I/O/Z
		1	PWM15_A	Выход ШИМ15 канал A	O
		2	I2C0_SDA	Вход/выход данных I2C0	I/O
		3	TMR7_IO	Вход/выход TMR7	I/O
		4	–	–	–
		5	TMR8_IO	Вход/выход TMR8	I/O
			ACMP1_INN	Инверсный вход аналогового компаратора 1	I
B1	D3			Вход/выход порта D, разряд 3	I/O/Z
		1	PWM15_B	Выход ШИМ15 канал B	O
		2	I2C0_SCL	Вход/выход синхронизации I2C0	I/O
		3	TMR8_IO	Вход/выход TMR8	I/O
		4	GMII_TXCLK	Вход синхросигнала передачи Ethernet	I
		5	TMR9_IO	Вход/выход TMR9	I/O
			ACMP1_INP	Прямой вход аналогового компаратора 1	I
B3	D4			Вход/выход порта D, разряд 4	I/O/Z
		1	PWM_SYNC	Вход синхронизации ШИМ	I
		2	ACMP2_OUT	Выход аналогового компаратора ACMP2	O
		3	TMR9_IO	Вход/выход TMR9	I/O
		4	GMII_GTXCLK	Выход синхросигнала гигабитной передачи Ethernet	O
		5	TMR10_IO	Вход/выход TMR10	I/O
			ACMP2_INN	Инверсный вход аналогового компаратора 2	I
A2	D5			Вход/выход порта D, разряд 5	I/O/Z
		1	PWM_TZ4	Вход сигнала аварии ШИМ, четвертый	I
		2	ACMP3_OUT	Выход аналогового компаратора ACMP3	O
		3	TMR10_IO	Вход/выход TMR10	I/O
		4	GMII_RXCLK	Вход синхросигнала приема Ethernet	I
		5	TMR11_IO	Вход/выход TMR11	I/O
			ACMP2_INP	Прямой вход аналогового компаратора 2	I
A3	D6			Вход/выход порта D, разряд 6	I/O/Z
		1	EXTCLK	Вход дополнительного тактового сигнала	I
		2	TMR6_IO	Вход/выход TMR6	I/O
		3	QSPI0_CLK	Вход/выход синхронизации QSPI0	I/O
		4	–	–	–
		5	TMR12_IO	Вход/выход TMR12	I/O
			ACMP3_INN	Инверсный вход аналогового компаратора 3	I

Продолжение таблицы 3.3

1	2	3	4	5	6
C3	D7			Вход/выход порта D, разряд 7	I/O/Z
		1	CLKOUT	Выход тактового сигнала CLKOUT	O
		2	SDR_CLKOUT	Выход тактового сигнала SDRAM	O
		3	QSPI0_FSS	Вход/выход выбора устройства QSPI0	I/O
		4	–	–	–
		5	TMR14_IO	Вход/выход TMR14	I/O
			ACMP3_INP	Прямой вход аналогового компаратора 3	I
B4	D8			Вход/выход порта D, разряд 8	I/O/Z
		1	CANFD0_RX	Вход данных CANFD0	I
		2	MEM_ADDR0	Выход шины адреса внешней памяти, разряд 0	O
		3	QSPI0_IO0	Вход/выход данных QSPI0, разряд 0	I/O
		4	GMII_RXDV	Вход валидности принимаемых данных Ethernet	I
		5	TMR13_IO	Вход/выход TMR13	I/O
C4	D9			Вход/выход порта D, разряд 9	I/O/Z
		1	CANFD0_TX	Выход данных CANFD0	O
		2	MEM_ADDR1	Выход шины адреса внешней памяти, разряд 1	O
		3	QSPI0_IO1	Вход/выход данных QSPI0, разряд 1	I/O
		4	GMII_RXER	Вход ошибки приема Ethernet	I
		5	TMR15_IO	Вход/выход TMR15	I/O
C5	D10			Вход/выход порта D, разряд 10	I/O/Z
		1	CANFD1_RX	Вход данных CANFD1	I
		2	MEM_ADDR2	Выход шины адреса внешней памяти, разряд 2	O
		3	QSPI0_IO2	Вход/выход данных QSPI0, разряд 2	I/O
		4	GMII_MDIO	Вход/выход управляющих данных Ethernet	I/O
		5	TMR1_IO0	Вход/выход 0 TMR1	I/O
B5	D11			Вход/выход порта D, разряд 11	I/O/Z
		1	CANFD1_TX	Выход данных CANFD1	O
		2	MEM_ADDR3	Выход шины адреса внешней памяти, разряд 3	O
		3	QSPI0_IO3	Вход/выход данных QSPI0, разряд 3	I/O
		4	GMII_TXEN	Выход разрешения передачи Ethernet	O
		5	TMR13_IO	Вход/выход TMR13	I/O
A4	D12			Вход/выход порта D, разряд 12	I/O/Z
		1	CANFD2_RX	Вход данных CANFD2	I
		2	MEM_ADDR4	Выход шины адреса внешней памяти, разряд 4	O
		3	SPI2_CLK	Вход/выход синхронизации SPI2	I/O
		4	GMII_TXER	Выход ошибки передачи Ethernet	O
		5	TMR0_IO0	Вход/выход 0 TMR0	I/O
C6	D13			Вход/выход порта D, разряд 13	I/O/Z
		1	CANFD2_TX	Выход данных CANFD2	O
		2	MEM_ADDR5	Выход шины адреса внешней памяти, разряд 5	O
		3	SPI2_FSS	Вход/выход выбора устройства SPI2	I/O
		4	GMII_CRS	Вход контроля носителя Ethernet	I
		5	TMR0_IO1	Вход/выход 1 TMR0	I/O
A5	D14			Вход/выход порта D, разряд 14	I/O/Z
		1	CANFD3_RX	Вход данных CANFD3	I
		2	MEM_ADDR6	Выход шины адреса внешней памяти, разряд 6	O
		3	SPI2_RX	Вход данных SPI2	I
		4	GMII_COL	Вход обнаружения конфликта Ethernet	I
		5	TMR0_IO2	Вход/выход 2 TMR0	I/O
C7	D15			Вход/выход порта D, разряд 15	I/O/Z
		1	CANFD3_TX	Выход данных CANFD3	O
		2	MEM_ADDR7	Выход шины адреса внешней памяти, разряд 7	O
		3	SPI2_TX	Выход данных SPI2	O
		4	GMII_MDC	Выход синхросигнала управляющих данных Ethernet	O
		5	TMR0_IO3	Вход/выход 3 TMR0	I/O

Продолжение таблицы 3.3

1	2	3	4	5	6
E14	E0			Вход/выход порта E, разряд 0	I/O/Z
		1	UART4_RX	Вход данных UART4	I
		2	MEM_ADDR8	Выход шины адреса внешней памяти, разряд 8	O
		3	SPI6_CLK	Вход/выход синхросигнала SPI6	I/O
		4	GMII_TXD0	Выход нулевого бита данных Ethernet	O
		5	TMR1_IO0	Вход/выход 0 TMR1	I/O
G15	E1			Вход/выход порта E, разряд 1	I/O/Z
		1	UART4_TX	Выход данных UART4	O
		2	MEM_ADDR9	Выход шины адреса внешней памяти, разряд 9	O
		3	SPI6_FSS	Вход/выход выбора устройства SPI6	I/O
		4	GMII_TXD1	Выход первого бита данных Ethernet	O
		5	TMR1_IO1	Вход/выход 1 TMR1	I/O
F13	E2			Вход/выход порта E, разряд 2	I/O/Z
		1	UART5_RX	Вход данных UART5	I
		2	MEM_ADDR10	Выход шины адреса внешней памяти, разряд 10	O
		3	SPI6_RX	Вход данных SPI6	I
		4	GMII_TXD2	Выход второго бита данных Ethernet	O
		5	TMR1_IO2	Вход/выход 2 TMR1	I/O
G12	E3			Вход/выход порта E, разряд 3	I/O/Z
		1	UART5_TX	Выход данных UART5	O
		2	MEM_ADDR11	Выход шины адреса внешней памяти, разряд 11	O
		3	SPI6_TX	Выход данных SPI6	O
		4	GMII_TXD3	Выход третьего бита данных Ethernet	O
		5	TMR1_IO3	Вход/выход 3 TMR1	I/O
G13	E4			Вход/выход порта E, разряд 4	I/O/Z
		1	LIN4_RX	Вход данных LIN4	I
		2	MEM_ADDR12	Выход шины адреса внешней памяти, разряд 12	O
		3	PWM15_A	Выход ШИМ15 канал A	O
		4	GMII_TXD4	Выход четвертого бита данных Ethernet	O
		5	TMR2_IO0	Вход/выход 0 TMR2	I/O
G14	E5			Вход/выход порта E, разряд 5	I/O/Z
		1	LIN4_TX	Выход данных LIN4	O
		2	MEM_ADDR13/	Выход шины адреса внешней памяти, разряд 13/	
			SDR_BA0	Адрес банка SDRAM разряд 0	O
		3	PWM15_B	Выход ШИМ15 канал B	O
		4	GMII_TXD5	Выход пятого бита данных Ethernet	O
		5	TMR2_IO1	Вход/выход 1 TMR2	I/O
H14	E6			Вход/выход порта E, разряд 6	I/O/Z
		1	LIN5_RX	Вход данных LIN5	I
		2	MEM_ADDR14/	Выход шины адреса внешней памяти, разряд 14/	
			SDR_BA1	Адрес банка SDRAM разряд 1	O
		3	PWM14_A	Выход ШИМ14 канал A	O
		4	GMII_TXD6	Выход шестого бита данных Ethernet	O
		5	TMR2_IO2	Вход/выход 2 TMR2	I/O
H15	E7			Вход/выход порта E, разряд 7	I/O/Z
		1	LIN5_TX	Выход данных LIN5	O
		2	MEM_ADDR15/	Выход шины адреса внешней памяти, разряд 15/	
			SDR_CKE	Включение тактового сигнала SDRAM	O
		3	PWM14_B	Выход ШИМ14 канал B	O
		4	GMII_TXD7	Выход седьмого бита данных Ethernet	O
		5	TMR2_IO3	Вход/выход 3 TMR2	I/O
H13	E8			Вход/выход порта E, разряд 8	I/O/Z
		1	UART6_RX	Вход данных UART6	I
		2	MEM_ADDR16/	Выход шины адреса внешней памяти, разряд 16/	
			SDR_WEN	Сигнал WE# SDRAM	O
		3	PWM_TZ5	Вход сигнала аварии ШИМ, пятый	I
		4	GMII_RXD0	Вход нулевого бита данных Ethernet	I
		5	TMR3_IO0	Вход/выход 0 TMR3	I/O

Продолжение таблицы 3.3

1	2	3	4	5	6
J15	E9			Вход/выход порта E, разряд 9	I/O/Z
		1	UART6_TX	Выход данных UART6	O
		2	MEM_ADDR17/	Выход шины адреса внешней памяти, разряд 17/	
			SDR_RASN	Сигнал RAS# SDRAM	O
		3	TMR13_IO	Вход/выход TMR13	I/O
		4	GMII_RXD1	Вход первого бита данных Ethernet	I
		5	TMR3_IO1	Вход/выход 1 TMR3	I/O
H12	E10			Вход/выход порта E, разряд 10	I/O/Z
		1	UART7_RX	Вход данных UART7	I
		2	MEM_ADDR18/	Выход шины адреса внешней памяти, разряд 18/	
			SDR_CASN	Сигнал CAS# SDRAM	O
		3	TMR14_IO	Вход/выход TMR14	I/O
		4	GMII_RXD2	Вход второго бита данных Ethernet	I
		5	TMR3_IO2	Вход/выход 2 TMR3	I/O
J14	E11			Вход/выход порта E, разряд 11	I/O/Z
		1	UART7_TX	Выход данных UART7	O
		2	MEM_ADDR19/	Выход шины адреса внешней памяти, разряд 19/	
			SDR_CSN	Сигнал CS# SDRAM	O
		3	TMR15_IO	Вход/выход TMR15	I/O
		4	GMII_RXD3	Вход третьего бита данных Ethernet	I
		5	TMR3_IO3	Вход/выход 3 TMR3	I/O
K15	E12			Вход/выход порта E, разряд 12	I/O/Z
		1	LIN6_RX	Вход данных LIN6	I
		2	MEM_ADDR20/	Выход шины адреса внешней памяти, разряд 20/	
			SDR_DQ0	Маска шины данных SDRAM разряд 0	O
		3	SPI3_CLK	Вход/выход синхронизации SPI3	I/O
		4	GMII_RXD4	Вход четвертого бита данных Ethernet	I
		5	TMR4_IO	Вход/выход TMR4	I/O
J13	E13			Вход/выход порта E, разряд 13	I/O/Z
		1	LIN6_TX	Выход данных LIN6	O
		2	MEM_ADDR21/	Выход шины адреса внешней памяти, разряд 21/	
			SDR_DQ1	Маска шины данных SDRAM разряд 1	O
		3	SPI3_FSS	Вход/выход выбора устройства SPI3	I/O
		4	GMII_RXD5	Вход пятого бита данных Ethernet	I
		5	TMR5_IO	Вход/выход TMR5	I/O
K14	E14			Вход/выход порта E, разряд 14	I/O/Z
		1	LIN7_RX	Вход данных LIN7	I
		2	MEM_ADDR22/	Выход шины адреса внешней памяти, разряд 22/	
			SDR_DQ2	Маска шины данных SDRAM разряд 2	O
		3	SPI3_RX	Вход данных SPI3	I
		4	GMII_RXD6	Вход шестого бита данных Ethernet	I
		5	TMR6_IO	Вход/выход TMR6	I/O
L14	E15			Вход/выход порта E, разряд 15	I/O/Z
		1	LIN7_TX	Выход данных LIN7	O
		2	MEM_ADDR22/	Выход шины адреса внешней памяти, разряд 23/	
			SDR_DQ3	Маска шины данных SDRAM разряд 3	O
		3	SPI3_TX	Выход данных SPI3	O
		4	GMII_RXD7	Вход седьмого бита данных Ethernet	I
		5	TMR7_IO	Вход/выход TMR7	I/O
A6	F0			Вход/выход порта F, разряд 0	I/O/Z
		1	QSPI0_CLK	Вход/выход синхронизации QSPI0	I/O
		2	SDR_DATA16	Вход/выход данных SDRAM разряд 16	I/O
		3	LIN4_RX	Вход данных LIN4	I
		4	USB0_DATA0	Шина данных интерфейса USB0, разряд 0	I/O
		5	TMR8_IO	Вход/выход TMR8	I/O

Продолжение таблицы 3.3

1	2	3	4	5	6
B6	F1			Вход/выход порта F, разряд 1	I/O/Z
		1	QSPI0_FSS	Вход/выход выбора устройства QSPI0	I/O
		2	SDR_DATA17	Вход/выход данных SDRAM разряд 17	I/O
		3	LIN4_TX	Выход данных LIN4	O
		4	USB0_DATA1	Шина данных интерфейса USB0, разряд 1	I/O
		5	TMR9_IO	Вход/выход TMR9	I/O
B7	F2			Вход/выход порта F, разряд 2	I/O/Z
		1	QSPI0_IO0	Вход/выход данных QSPI0, разряд 0	I/O
		2	SDR_DATA18	Вход/выход данных SDRAM разряд 18	I/O
		3	LIN5_RX	Вход данных LIN5	I
		4	USB0_DATA2	Шина данных интерфейса USB0, разряд 2	I/O
		5	TMR10_IO	Вход/выход TMR10	I/O
B8	F3			Вход/выход порта F, разряд 3	I/O/Z
		1	QSPI0_IO1	Вход/выход данных QSPI0, разряд 1	I/O
		2	SDR_DATA19	Вход/выход данных SDRAM разряд 19	I/O
		3	LIN5_TX	Выход данных LIN5	O
		4	USB0_DATA3	Шина данных интерфейса USB0, разряд 3	I/O
		5	TMR11_IO	Вход/выход TMR11	I/O
C8	F4			Вход/выход порта F, разряд 4	I/O/Z
		1	QSPI0_IO2	Вход/выход данных QSPI0, разряд 2	I/O
		2	EMC_CEN0/	Сигнал включения микросхемы внешней памяти разряд 0/	
			SDR_DATA20	Вход/выход данных SDRAM разряд 20	I/O
		3	UART6_RX	Вход данных UART6	I
		4	USB0_DATA4	Шина данных интерфейса USB0, разряд 4	I/O
		5	TMR12_IO	Вход/выход TMR12	I/O
C9	F5			Вход/выход порта F, разряд 5	I/O/Z
		1	QSPI0_IO3	Вход/выход данных QSPI0, разряд 3	I/O
		2	EMC_CEN1/	Сигнал включения микросхемы внешней памяти разряд 1/	
			SDR_DATA21	Вход/выход данных SDRAM разряд 21	I/O
		3	UART6_TX	Выход данных UART6	O
		4	USB0_DATA5	Шина данных интерфейса USB0, разряд 5	I/O
		5	TMR13_IO	Вход/выход TMR13	I/O
A7	F6			Вход/выход порта F, разряд 6	I/O/Z
		1	I2C0_SDA	Вход/выход данных I2C0	I/O
		2	EMC_CEN2/	Сигнал включения микросхемы внешней памяти разряд 2/	
			SDR_DATA22	Вход/выход данных SDRAM разряд 22	I/O
		3	UART7_RX	Вход данных UART7	I
		4	USB0_DATA6	Шина данных интерфейса USB0, разряд 6	I/O
		5	TMR14_IO	Вход/выход TMR14	I/O
B9	F7			Вход/выход порта F, разряд 7	I/O/Z
		1	I2C0_SCL	Вход/выход синхронизации I2C0	I/O
		2	EMC_CEN3/	Сигнал включения микросхемы внешней памяти разряд 3/	
			SDR_DATA23	Вход/выход данных SDRAM разряд 23	I/O
		3	UART7_TX	Выход данных UART7	O
		4	USB0_DATA7	Шина данных интерфейса USB0, разряд 7	I/O
		5	TMR15_IO	Вход/выход TMR15	I/O
D13	F8			Вход/выход порта F, разряд 8	I/O/Z
		1	TMR2_IO0	Вход/выход 0 TMR2	I/O
		2	EMC_CEN4/	Сигнал включения микросхемы внешней памяти разряд 4/	
			SDR_DATA24	Вход/выход данных SDRAM разряд 24	I/O
		3	SPI4_CLK	Вход/выход синхросигнала SPI4	I/O
		4	–	–	–
		5	TMR0_IO0	Вход/выход 0 TMR0	I/O
D14	F9			Вход/выход порта F, разряд 9	I/O/Z
		1	TMR3_IO0	Вход/выход 0 TMR3	I/O
		2	EMC_CEN5/	Сигнал включения микросхемы внешней памяти разряд 5/	
			SDR_DATA25	Вход/выход данных SDRAM разряд 25	I/O
		3	SPI4_FSS	Вход/выход выбора устройства SPI4	I/O
		4	USB0_RESET	Сигнал сброса интерфейса USB0	O
		5	TMR0_IO1	Вход/выход 1 TMR0	I/O

Продолжение таблицы 3.3

1	2	3	4	5	6
E15	F10			Вход/выход порта F, разряд 10	I/O/Z
		1	TMR4_IO	Вход/выход TMR4	I/O
		2	EMC_CEN6/	Сигнал включения микросхемы внешней памяти разряд 6/	
			SDR_DATA26	Вход/выход данных SDRAM разряд 26	I/O
		3	SPI4_RX	Вход данных SPI4	I
		4	USB0_XCVRSELECT0	Выход выбора скорости передатчика USB0, разряд 0	O
		5	TMR0_IO2	Вход/выход 2 TMR0	I/O
E12	F11			Вход/выход порта F, разряд 11	I/O/Z
		1	TMR5_IO	Вход/выход TMR5	I/O
		2	EMC_CEN7/	Сигнал включения микросхемы внешней памяти разряд 7/	
			SDR_DATA27	Вход/выход данных SDRAM разряд 27	I/O
		3	SPI4_TX	Выход данных SPI4	O
		4	USB0_XCVRSELECT1	Выход выбора скорости передатчика USB0, разряд 1	O
		5	TMR0_IO3	Вход/выход 3 TMR0	I/O
E14	F12			Вход/выход порта F, разряд 12	I/O/Z
		1	TMR6_IO	Вход/выход TMR6	I/O
		2	EMC_WEN/	Сигнал управления записью во внешнюю память/	
			SDR_DATA28	Вход/выход данных SDRAM разряд 28	I/O
		3	SPI5_CLK	Вход/выход синхросигнала SPI5	I/O
		4	USB0_TERMSELECT	Сигнал выбора подключения FS и HS USB0	O
		5	TMR1_IO0	Вход/выход 0 TMR1	I/O
E13	F13			Вход/выход порта F, разряд 13	I/O/Z
		1	TMR7_IO	Вход/выход TMR7	I/O
		2	EMC_LBN/	Сигнал включения младшего байта внешней памяти/	
			SDR_DATA29	Вход/выход данных SDRAM разряд 29	I/O
		3	SPI5_FSS	Вход/выход выбора устройства SPI5	I/O
		4	USB0_SUSPENDM	Сигнал режима Suspend USB0	O
		5	TMR1_IO1	Вход/выход 1 TMR1	I/O
F15	F14			Вход/выход порта F, разряд 14	I/O/Z
		1	TMR8_IO	Вход/выход TMR8	I/O
		2	EMC_UBN/	Сигнал включения старшего байта внешней памяти/	
			SDR_DATA30	Вход/выход данных SDRAM разряд 30	I/O
		3	SPI5_RX	Вход данных SPI5	I
		4	USB0_OPMODE0	Сигнал выбора операции USB0, разряд 0	O
		5	TMR1_IO2	Вход/выход 2 TMR1	I/O
F12	F15			Вход/выход порта F, разряд 15	I/O/Z
		1	TMR9_IO	Вход/выход TMR9	I/O
		2	EMC_OEN/	Сигнал включения работы на выход внешней памяти/	
			SDR_DATA31	Вход/выход данных SDRAM разряд 31	I/O
		3	SPI5_TX	Выход данных SPI5	O
		4	USB0_OPMODE1	Сигнал выбора операции USB0, разряд 1	O
		5	TMR1_IO3	Вход/выход 3 TMR1	I/O
R9	G0			Вход/выход порта G, разряд 0	I/O/Z
		1	PWM6_A	Выход ШИМ6 канал A	O
		2	MEM_DATA0	Вход/выход шины данных внешней памяти, разряд 0	I/O
		3	CAN0_RX	Вход данных CAN узел 0	I
		4	USB0_LINESTATE0	Сигнал состояния шины данных USB0 разряд 0	I
		5	TMR2_IO0	Вход/выход 0 TMR2	I/O
N10	G1			Вход/выход порта G, разряд 1	I/O/Z
		1	PWM6_B	Выход ШИМ6 канал B	O
		2	MEM_DATA1	Вход/выход шины данных внешней памяти, разряд 1	I/O
		3	CAN0_TX	Выход данных CAN узел 0	O
		4	USB0_LINESTATE1	Сигнал состояния шины данных USB0 разряд 1	I
		5	TMR2_IO1	Вход/выход 1 TMR2	I/O

Продолжение таблицы 3.3

1	2	3	4	5	6
P10	G2			Вход/выход порта G, разряд 2	I/O/Z
		1	PWM7_A	Выход ШИМ7 канал A	O
		2	MEM_DATA2	Вход/выход шины данных внешней памяти, разряд 2	I/O
		3	CAN1_RX	Вход данных CAN узел 1	I
		4	USB0_TXVALID	Сигнал разрешения передачи данных USB0	O
		5	TMR2_IO2	Вход/выход 2 TMR2	I/O
R10	G3			Вход/выход порта G, разряд 3	I/O/Z
		1	PWM7_B	Выход ШИМ7 канал B	O
		2	MEM_DATA3	Вход/выход шины данных внешней памяти, разряд 3	I/O
		3	CAN1_TX	Выход данных CAN узел 1	O
		4	USB0_TXREADY	Сигнал готовности к передаче данных USB0	I
		5	TMR2_IO3	Вход/выход 3 TMR2	I/O
R11	G4			Вход/выход порта G, разряд 4	I/O/Z
		1	PWM8_A	Выход ШИМ8 канал A	O
		2	MEM_DATA4	Вход/выход шины данных внешней памяти, разряд 4	I/O
		3	UART4_RX	Вход данных UART4	I
		4	USB0_RXACTIVE	Сигнал активности приема USB0	I
		5	TMR3_IO0	Вход/выход 0 TMR3	I/O
R12	G5			Вход/выход порта G, разряд 5	I/O/Z
		1	PWM8_B	Выход ШИМ8 канал B	O
		2	MEM_DATA5	Вход/выход шины данных внешней памяти, разряд 5	I/O
		3	UART4_TX	Выход данных UART4	O
		4	USB0_RXVALID	Сигнал индикации получения данных USB0	I
		5	TMR3_IO1	Вход/выход 1 TMR3	I/O
P11	G6			Вход/выход порта G, разряд 6	I/O/Z
		1	PWM9_A	Выход ШИМ9 канал A	O
		2	MEM_DATA6	Вход/выход шины данных внешней памяти, разряд 6	I/O
		3	UART5_RX	Вход данных UART5	I
		4	USB0_RXERROR	Сигнал обнаружения ошибки приема USB0	I
		5	TMR3_IO2	Вход/выход 2 TMR3	I/O
N11	G7			Вход/выход порта G, разряд 7	I/O/Z
		1	PWM9_B	Выход ШИМ9 канал B	O
		2	MEM_DATA7	Вход/выход шины данных внешней памяти, разряд 7	I/O
		3	UART5_TX	Выход данных UART5	O
		4	USB0_VBUSVALID	Сигнал допустимого уровня VBUS USB0	I
		5	TMR3_IO3	Вход/выход 3 TMR3	I/O
N12	G8			Вход/выход порта G, разряд 8	I/O/Z
		1	PWM10_A	Выход ШИМ10 канал A	O
		2	MEM_DATA8	Вход/выход шины данных внешней памяти, разряд 8	I/O
		3	CAN2_RX	Вход данных CAN узел 2	I
		4	USB0_CHRGVBUS	Сигнал управления зарядом VBUS USB0	O
		5	TMR4_IO	Вход/выход TMR4	I/O
R13	G9			Вход/выход порта G, разряд 9	I/O/Z
		1	PWM10_B	Выход ШИМ10 канал B	O
		2	MEM_DATA9	Вход/выход шины данных внешней памяти, разряд 9	I/O
		3	CAN2_TX	Выход данных CAN узел 2	O
		4	USB0_DISCHRGVBUS	Сигнал управления разрядом VBUS USB0	O
		5	TMR5_IO	Вход/выход TMR5	I/O

Продолжение таблицы 3.3

1	2	3	4	5	6
P12	G10			Вход/выход порта G, разряд 10	I/O/Z
		1	PWM11_A	Выход ШИМ11 канал A	O
		2	MEM_DATA10	Вход/выход шины данных внешней памяти, разряд 10	I/O
		3	CAN3_RX	Вход данных CAN узел 3	I
		4	USB0_DPPULLDOWN	Сигнал включения pull-down для линии DP USB0	O
		5	TMR6_IO	Вход/выход TMR6	I/O
P13	G11			Вход/выход порта G, разряд 11	I/O/Z
		1	PWM11_B	Выход ШИМ11 канал B	O
		2	MEM_DATA11	Вход/выход шины данных внешней памяти, разряд 11	I/O
		3	CAN3_TX	Выход данных CAN узел 3	O
		4	USB0_DMPULLDOWN	Сигнал включения pull-down для линии DM USB0	O
		5	TMR7_IO	Вход/выход TMR7	I/O
N13	G12			Вход/выход порта G, разряд 12	I/O/Z
		1	PWM12_A	Выход ШИМ12 канал A	O
		2	MEM_DATA12	Вход/выход шины данных внешней памяти, разряд 12	I/O
		3	LIN6_RX	Вход данных LIN6	I
		4	USB0_IDPULLUP	Сигнал включения pull-up линии ID USB0	O
		5	TMR8_IO	Вход/выход TMR8	I/O
R14	G13			Вход/выход порта G, разряд 13	I/O/Z
		1	PWM12_B	Выход ШИМ12 канал B	O
		2	MEM_DATA13	Вход/выход шины данных внешней памяти, разряд 13	I/O
		3	LIN6_TX	Выход данных LIN6	O
		4	USB0_HOSTDISCONNECT	Сигнал отключения хоста (окончания сессии) USB0	I
		5	TMR9_IO	Вход/выход TMR9	I/O
R15	G14			Вход/выход порта G, разряд 14	I/O/Z
		1	PWM13_A	Выход ШИМ13 канал A	O
		2	MEM_DATA14	Вход/выход шины данных внешней памяти, разряд 14	I/O
		3	LIN7_RX	Вход данных LIN7	I
		4	MSC_DI1	Вход данных MSC разряд 1	I
		5	TMR10_IO	Вход/выход TMR10	I/O
P14	G15			Вход/выход порта G, разряд 15	I/O/Z
		1	PWM13_B	Выход ШИМ13 канал B	O
		2	MEM_DATA15	Вход/выход шины данных внешней памяти, разряд 15	I/O
		3	LIN7_TX	Выход данных LIN7	O
		4	MSC_SS3	Выход сигнала выбора ведомого MSC, разряд 3	O
		5	TMR11_IO	Вход/выход TMR11	I/O

Примечания:

1) Входы блока АЦП ADC_CH0 – ADC_CH47, а также входы аналоговых компараторов CMPn_INN, CMPn_INP не являются альтернативными функциями и подключены к выводам GPIO постоянно.

2) В графе «Тип вывода» принятые условные обозначения:

- I – вход,
- O – выход,
- Z – третье состояние.

Таблица 3.4 – Функциональное назначение выводов, не имеющих альтернативных функций

Номер вывода	Обозначение			Функциональное назначение вывода	Тип вывода
	вывода	Альтернативной функции			
		№	Обозначение		
1	2	3	4	5	6
B2	XI_OSC	-	-	Вход кварцевого генератора/Вход тактового сигнала	I
A1	XO_OSC	-	-	Выход кварцевого генератора/Выход тактового сигнала	O
A8	XI_RTC	-	-	Вход низкочастотного кварцевого генератора	I
C10	XO_RTC	-	-	Выход низкочастотного кварцевого генератора	O
N15	RST#	-	-	Вход внешнего сигнала сброса	I
M14	TCK	-		Вход тактового сигнала порта JTAG	I
M15	TDI	-		Вход данных порта JTAG	I
L15	TDO	-		Вход режима порта JTAG	O
M13	TMS	-		Вход сброса порта JTAG	I
N14	TRST#	-	-	Вход сброса порта JTAG	I
P15	SERVEN	-	-	Вход сигнала активации сервисного режима	I
A9	VBAT	-	-	Вывод подключения батарейного питания	
L13	LDO_CAP	-	-	Вывод подключения конденсатора для внутреннего LDO	
K1	AVCC_ADC	-	-	Вывод аналогового питания модуля АЦП	
L5, L7, L8, L9, L10, L11, K11, J11, H11, G11, F11, E11, E10, E9, E8. E7, E6, E5, F5, G5, H5, J5, K5	VCC	-	-	Вывод цифрового питания	
M5, M6, M7, K8, M8, K9, M9, M10, M11, M12, K10, L12, K12, J10, J12, H10, G10, F10, A15, C14, A13, C12, D10, F9, D9, F8, D8, D7, F7, D6, D5, D3, E1, E2, F3, F6, G6, H6, K7, J6, K6, L6	GND	-	-	Общий вывод ИС	
B10, A10	AVCC_ETH	-	-	Вывод питания модуля Ethernet 3,3 В	
B11, A11	AGND_ETH	-	-	Общий вывод модуля Ethernet	

Продолжение таблицы 3.4

1	2	3	4	5	6
C11	ETH_EXTR	-	-	Вывод Ethernet для подключения внешнего резистора	
C15	ETH_IO0P	-	-	Вход/выход данных 0 (+) Ethernet	I/O
B15	ETH_IO0N	-	-	Вход/выход данных 0 (–) Ethernet	I/O
B14	ETH_IO1P	-	-	Вход/выход данных 1 (+) Ethernet	I/O
A14	ETH_IO1N	-	-	Вход/выход данных 1 (–) Ethernet	I/O
C13	ETH_IO2P	-	-	Вход/выход данных 2 (+) Ethernet	I/O
B13	ETH_IO2N	-	-	Вход/выход данных 2 (–) Ethernet	I/O
B12	ETH_IO3P	-	-	Вход/выход данных 3 (+) Ethernet	I/O
A12	ETH_IO3N	-	-	Вход/выход данных 3 (–) Ethernet	I/O
D15	XI_ETH	-	-	Вход для подключения кварцевого резонатора Ethernet	I
D12	XO_ETH	-	-	Вывод для подключения кварцевого резонатора Ethernet	O
J2	DAC0	-	-	Выход канала ЦАП0	O
H3	DAC1	-	-	Выход канала ЦАП1	O
F1	USB0_DP	-	-	Вход/выход «USB0 Dp»	I/O
F2	USB0_DN	-	-	Вход/выход «USB0 Dn»	I/O
F4	USB0_REXT	-	-	Вывод USB0 для подключения внешнего резистора	-
D1	USB1_DP	-	-	Вход/выход «USB1 Dp»	I/O
D2	USB1_DN	-	-	Вход/выход «USB1 Dn»	I/O
D4	USB1_REXT	-	-	Вывод USB1 для подключения внешнего резистора	-

Примечания:
1) В графе «Тип вывода» принятые условные обозначения:
- I – вход,
- O – выход,
- Z – третье состояние

2) Выводы D11 и K13 оставлять неподключенными.
3) Сопротивление внутренних резисторов подтяжки Pull up в диапазоне от 27 кОм до 65 кОм, типовое – 40 кОм. Сопротивление внутренних резисторов подтяжки Pull down в диапазоне от 31 кОм до 84 кОм, типовое – 48 кОм.

Микросхемы К1921ВГ10Я выполнены в пластиковом корпусе BGA-216.
Масса микросхемы – не более 3 г.
Габаритный чертеж корпуса микросхемы приведен в Приложении Г.

3.2 Электрические параметры

Значение суммарного максимального тока по выводам портов ввода-вывода A, B, C, D, E, F, G не должно превышать 150 мА.

Микросхемы К1921ВГ1Т, К1921ВГ10Я устойчивы к воздействию статического электричества с потенциалом не менее 2 000 В по ГОСТ РВ 5962-004.

Электрические параметры микросхем при приемке и поставке соответствуют нормам, приведенным в таблице 3.5.

Таблица 3.5 – Электрические параметры микросхем при приемке и поставке

Наименование параметра, единица измерения, режим измерения		Буквенное обозначение параметра	Норма параметра		Температура среды, ºС
			не менее	не более	
1		2	3	4	5
1 Выходное напряжение низкого уровня буферов ввода/вывода, В, I_{OL} = 6 мА, U_{CC1} = 3,3 В		U_{OL}	–	0,4	–40 ± 3 25 ± 10 105 ± 5
2 Выходное напряжение высокого уровня буферов ввода/вывода, В, I_{OH} = –6 мА, U_{CC1} = 3,3 В		U_{OH}	U_{CC1}– 0,4	–	
3 Ток утечки низкого уровня по входам с отключенной схемой «pull-up» и «pull-down», мкА, 0 В < U_I < U_{CC1}		I_{LIL}	–1	–	
4 Ток утечки высокого уровня по выводам с отключенной схемой «pull-up» и «pull-down», мкА, 0 В < U_I < U_{CC1}		I_{LIH}	–	1	
5 Входной ток низкого уровня по выводам с подключенной схемой «pull-up», мкА, 0 В < U_I < U_{CC1}		I_{I1}	–200	–	
6 Входной ток высокого уровня по выводам с подключенной схемой «pull-down», мкА, 0 В < U_I < U_{CC1}		I_{I2}	–	200	
7 Динамический ток потребления по выводу VCC в активном режиме[1)], мА, U_{CC1}=3,6 В, f_{CI} = 204 МГц	T=+25 ºC	I_{OCC1}	–	150	
	T=+105 ºC			250	
8 Динамический ток потребления по выводу VCC в активном режиме [2)], мА, U_{CC1}=3,6 В, f_{CI} = 204 МГц	T=+25 ºC	I_{OCC2}	–	170	
	T=+105 ºC			350	
9 Максимальный ток потребления АЦП по аналоговому питанию AVCC_ADC, мкА, U_{CC1} = U_{CC2} = 3,6 В, F_S=2,5 MSPS		I_{AVCC_ADC}	–	500	

Продолжение таблицы 3.5

1	2	3	4	5
10 Дифференциальная нелинейность АЦП, LSB, $U_{CC1} = U_{CC2} = 3{,}3$ В, $f_{CI} = 204$ МГц	$E_{DNL\ ADC}$	–2,5	2,5	–40 ± 3 25 ± 10 105 ± 5
11 Интегральная нелинейность АЦП, LSB, $U_{CC1} = U_{CC2} = 3{,}3$ В, $f_{CI} = 204$ МГц	$E_{INL\ ADC}$	–5,0	5,0	
12 Дифференциальная нелинейность ЦАП, LSB, $U_{CC1} = U_{CC2} = 3{,}3$ В, $f_{CI} = 204$ МГц	$E_{DNL\ DAC}$	–2	2	
13 Интегральная нелинейность ЦАП, LSB, $U_{CC1} = U_{CC2} = 3{,}3$ В, $f_{CI} = 204$ МГц	$E_{INL\ DAC}$	–4	4	
14 Гистерезис аналогового компаратора, мВ, $U_{CC1} = U_{CC2} = 3{,}3$ В	G	–	120	
15 Функциональный контроль, $U_{CC1} = (3{,}0; 3{,}6)$ В, $f_{CI} = 204$ МГц	ФК	–	–	

_____

[1] Ядра работают в режиме LockStep, используются основная Flash-память и RAM. Задействованы порты ввода-вывода GPIO, остальные периферийные блоки не тактируются и находятся в состоянии сброса.

[2] Ядра работают в режиме LockStep, используются основная Flash-память и RAM. Задействованы порты ввода-вывода GPIO, АЦП, DMA-контроллер, блок криптографии, остальные периферийные блоки тактируются и выведены из состояния сброса.

Необходимо соблюдать следующий порядок подачи напряжения питания: сначала подаются напряжения питания U_{CC1}, U_{CC2}, а затем входные напряжения низкого уровня U_{IL} и высокого уровня U_{IH}.

Порядок снятия напряжений с выводов микросхемы при выключении должен быть обратным подаче: первыми снимаются входные сигналы, а затем напряжения питания.

Допускается возможность одновременной подачи и снятия напряжений питания и напряжений входных сигналов.

Значения предельно допустимых электрических режимов эксплуатации в диапазоне рабочих температур среды соответствуют нормам, приведенным в таблице 3.6.

Таблица 3.6 – Значения предельно допустимых электрических режимов эксплуатации

Наименование параметра режима эксплуатации, единица измерения	Буквенное обозначение	Предельно допустимая норма при эксплуатации		Предельная норма	
		не менее	не более	не менее	не более
1 Напряжение питания цифровой части, В	U_{CC1}	2,5	3,6	–	4,5
2 Напряжение питания аналоговой части, В	U_{CC2}	2,5	3,6	–	4,5
3 Батарейное напряжение питания, В	U_{CC3}	1,6	3,6	–	4,5
4 Входное напряжение низкого уровня, В	U_{IL}	–0,5	$0{,}3U_{CC1}$	–0,6	–

Продолжение таблицы 3.6

Наименование параметра режима эксплуатации, единица измерения	Буквенное обозначение	Предельно допустимая норма при эксплуатации		Предельная норма	
		не менее	не более	не менее	не более
5 Входное напряжение высокого уровня, В	U_{IH}	$0,7U_{CC1}$	U_{CC1}	–	$U_{CC1}+0,6$
6 Выходной ток высокого уровня, мА	I_{OH}	–8,0	–	–10	–
7 Выходной ток низкого уровня, мА	I_{OL}	–	8,0	–	10
8 Частота следования импульсов тактового сигнала, МГц	f_{CI}	0,001	204	–	–
9 Диапазон входной частоты PLL, МГц	f_{PLL}	10	100		
10 Диапазон опорных напряжений ЦАП, В	$U_{REF\ DAC}$	1,5	U_{AVDD}		

4 Архитектура изделия

Микроконтроллеры К1921ВГ1Т, К1921ВГ10Я спроектированы на базе RISC-V двойного процессорного ядра SCR5 с поддержкой набора команд RV32IMAFDC, блоком управления памятью (MMU), кэшем второго уровня L2 и поддержкой межъядерной связи SCRX.

Двойное процессорное ядро микроконтроллера может работать в режиме LOCKSTEP.

4.1 Блок коммутации микроконтроллера

Все устройства микроконтроллера соединены между собой через блок коммутации. На рисунке 4.1 приведена схема соединения основных и периферийных блоков микроконтроллера внутри блока коммутации.

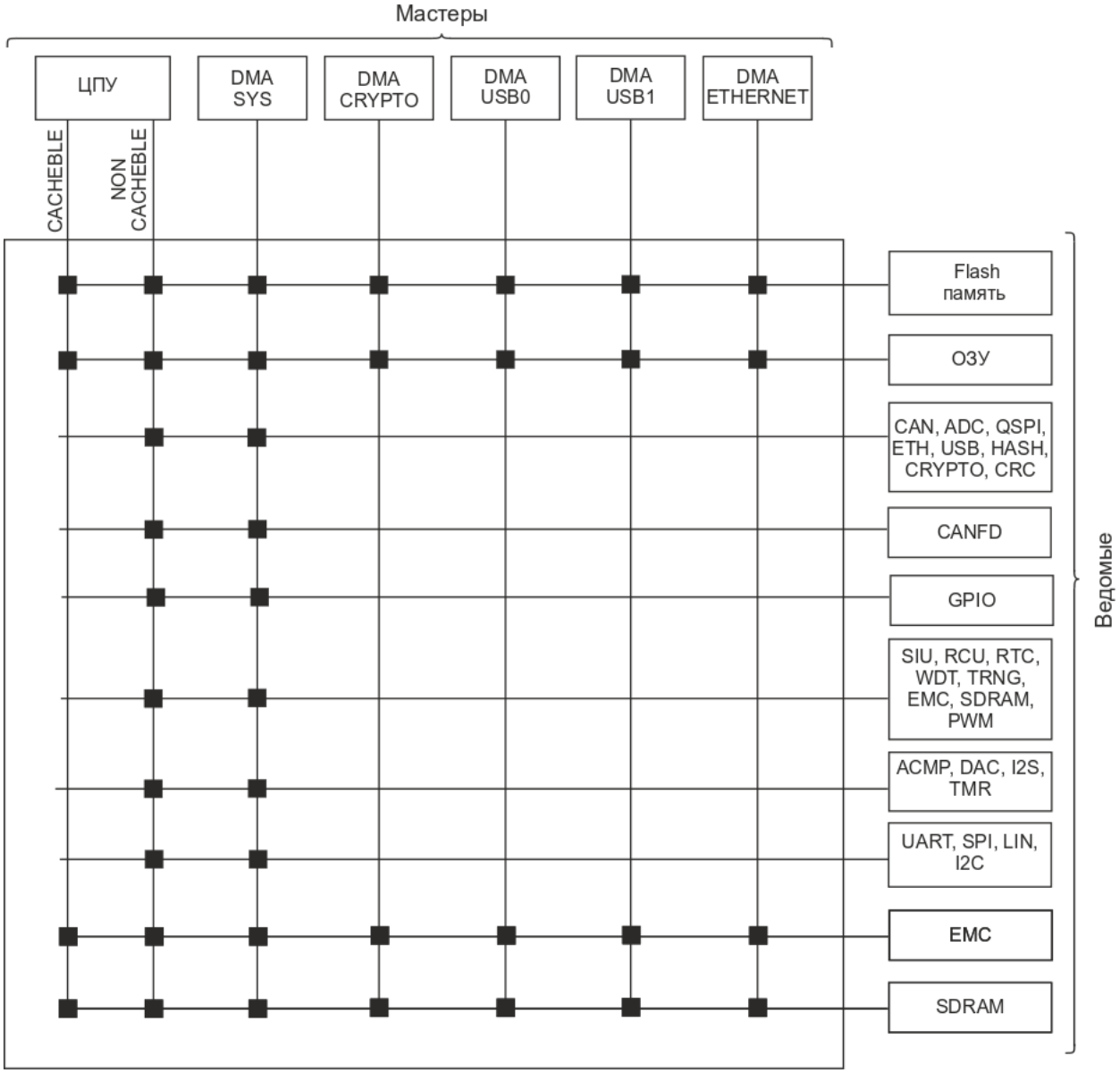


Рисунок 4.1 – Схема соединения блоков внутри блока коммутации микроконтроллера

4.2 Механизмы функциональной безопасности

В микроконтроллере предусмотрены следующие механизмы функциональной безопасности:

1) Режим работы ядер Lockstep

Возможность работы в режиме Lockstep: второе ядро является теневым и дублирует работу первого с задержкой на два такта во избежание одинакового воздействия сбоя на оба ядра. Аппаратный компаратор выравнивает задержки и сравнивает выходные значения обоих ядер. В случае несоответствия подается сигнал тревоги. Компаратор поддерживает аппаратное внесение ошибки для самопроверки.

2) L1I кеш.

Код Хэмминга: коррекция одной ошибки - обнаружение двух
(Single Error Correction - Double Error Detection SECDED).
Данные: 128 бит + 9 изб.
Теги: 19 бит + 6 изб.
Состояния (алгоритм LRU): 10 бит + 5 изб.

3) L1D кеш.

Код Хэмминга: коррекция одной ошибки - обнаружение двух
(Single Error Correction - Double Error Detection SECDED).
Данные: 128 бит + 9 изб.
Теги: 19 бит + 6 изб.
Состояния (алгоритм LRU): 10 бит + 5 изб.

4) L2 кеш.

Код Хэмминга: коррекция одной ошибки - обнаружение двух
(Single Error Correction - Double Error Detection SECDED).
Данные: 128 бит + 9 изб.
Теги: 25 бит + 6 изб.

5) Flash-память.

Операции записи/стирания защищены ключом.
Код Хэмминга: коррекция одной ошибки - обнаружение двух
(Single Error Correction - Double Error Detection SECDED).
Данные: 64 бита + 8 изб.

6) SRAM.

Код Хэмминга: коррекция одной ошибки - обнаружение двух
(Single Error Correction - Double Error Detection SECDED).
Данные: 8 бит + 5 изб. (32-разрядное слово кодируется побайтно).

7) Мониторы питания шины 1.1В / Мониторы питания шины 3.3В.

Ограничитель выходного уровня (Voltage Clamp).

High/Low/Minimal Voltage Detection (HVD/LVD/MVD): при отклонении выходного напряжения на определённый порог выставляется флаг в статусном регистре в блоке SIU и формируется маскируемое прерывание или сброс.

Защита от повышенного выходного напряжения (Upper Voltage Detection - UVD): при более значительном отклонении выходного напряжения (20%) выставляется флаг в статусном регистре в блоке RCU и формируется сброс.

8) Механизм слежения за наличием системного тактового сигнала.

В блоке RCU реализован механизм слежения за системным тактовым сигналом (в случае соответствующих настроек в регистре SYSCLKCFG это выходной сигнал PLL). Системный тактовый сигнал инкрементирует внутренний счетчик. В случае не достижения им заданного в регистре SECCNT0(1) значения формируется сигнал ошибки (прерывание) и система переключается на резервный тактовый сигнал (внутренний RC clock = 4 МГц).

9) Программная проверка АЦП с использованием ЦАП.

Тестовый выход встроенного ЦАП подключен ко входу неиспользуемого канала АЦП, что позволяет осуществить программную проверку работы АЦП и ЦАП.

10) Программная проверка аналоговых компараторов с использованием ЦАП.

Режим сравнения сигналов со встроенного ЦАП.

11) Программно-аппаратная проверка встроенного датчика температуры.

Последовательная проверка в двух режимах (CTAT1 и CTAT2) и анализ результата.

12) Запрет на реконфигурацию блоков RCU, SIU, Watchdog, GPIO без ввода ключа.

Механизм блокировки записи в регистры (настройка регистров и последующая запись флага блокировки в регистр LOCK вместе с ключом).

13) Внутренние мосты интерфейсов и кроссбар.

Механизм формирования ошибки при доступе к областям неразмеченной памяти.

Механизм формирования прерывания по тайм-ауту в случае неполучения сигнала готовности от периферийного устройства за определённое количество тактов.

14) CAN. CRC16 (стандарт).

15) UART. Контроль четности (стандарт).

16) Ethernet. Контроль четности (стандарт).

17) USB. Контроль четности (стандарт).

5 Система питания

5.1 Организация системы питания

Схема организации питания микроконтроллера представлена на рисунке 5.1.

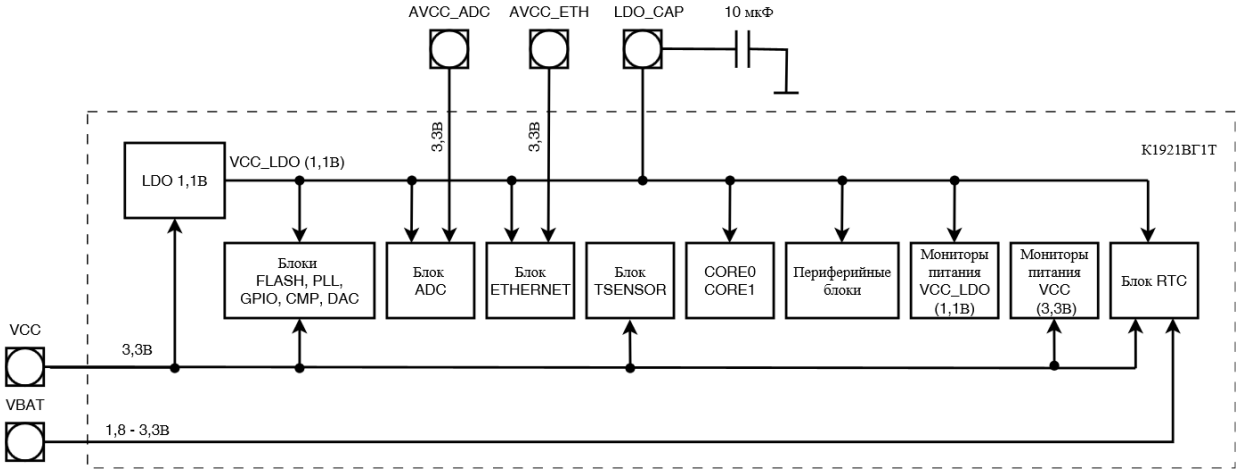


Рисунок 5.1 – Схема организации питания

Периферийные блоки, не указанные отдельно на схеме, подключены к цепи питания VCC_LDO (1,1 В) и включают в себя RCU, SIU, EMC, SDRAM, DMA, TMR, MSC, PWM, TRNG, HASH, CRC, CRYPTO, WDT, UART, SPI, QSPI, I2S, LIN, I2C, USB, CAN, CANFD.

5.2 Супервизор и мониторы питания

Микроконтроллер содержит мониторы питания шин VCC (3,3В) и VCC_LDO (1,1В), а также модули сброса по включению питания (Power-On-Reset – POR) и по просадке напряжения питания (Brown-Out-Reset – BOR). Мониторы шин представлены блоками MVD (Minimal Voltage Detector), LVD (Low Voltage Detector), HVD (High Voltage Detector) и UVD (Upper Voltage Detector). Электрические характеристики указанных блоков представлены в таблицах 5.1 и 5.2.

Таблица 5.1 – Электрические параметры мониторов питания

Наименование параметра, (режим измерения)	Норма, В		
	не менее	номинал	не более
Порог срабатывания блока MVD 1,1 В	–	0,89	–
Порог срабатывания блока LVD 1,1 В	–	0,99	–
Порог срабатывания блока HVD 1,1 В	–	1,21	–
Порог срабатывания блока UVD 1,1 В	–	1,32	–
Порог срабатывания POR 1,1 В (спадающее напряжение питания)	0,5	0,55	0,6
Порог срабатывания POR 1,1 В (нарастающее напряжение питания)	0,70	0,71	0,72
Порог срабатывания блока MVD 3,3 В	–	2,5	–
Порог срабатывания блока LVD 3,3 В	–	2,97	–
Порог срабатывания блока HVD 3,3 В	–	3,63	–
Порог срабатывания блока UVD 3,3 В	–	3,9	–
Порог срабатывания POR 3,3 В (спадающее напряжение питания)	1,6	1,7	1,8
Порог срабатывания POR 3,3 В (нарастающее напряжение питания)	1,90	1,91	1,92

Блок POR генерирует сигнал сброса длительностью не менее 150 мкс при подаче напряжения на порт цифрового питания. При падении напряжения ниже порогового уровня блок генерирует сигнал сброса низкого уровня до тех пор, пока питание не поднимется выше порога. Факт формирования сброса по включению питания фиксируется в поле POR регистра RSTSTAT блока RCU.

Основная функция блока BOR – генерировать сигнал сброса при падении напряжения питания VCC ниже порогового значения. Пороговые значения, при которых генерируется сигнал сброса при нарастающем и спадающем VCC различаются, при этом порог генерации сигнала сброса для спадающего напряжения ниже. Уровни BOR являются программируемыми и задаются значением поля TUNE регистра BORCFG блока RCU (см. таблицу 5.2).

Таблица 5.2 – Пороги срабатывания блока BOR

Значение поля TUNE	Порог срабатывания, В	
	при росте напряжения	при спаде напряжения
00b	3,0	2,9
01b	2,7	2,6
10b	2,2	2,1
11b	Зарезервировано	Зарезервировано

Факт формирования сброса по просадке напряжения питания фиксируется в поле BOR регистра RSTSTAT блока RCU.

В микроконтроллер встроены блоки мониторов питания по основным шинам, каждый из которых настроен на определённый уровень. Индикация ошибок по питанию (выход за границу допустимого диапазона) отображается в регистре FSTAT блока SIU.

Мониторы питания типа UVD при срабатывании формируют сигнал сброса для защиты схемы от повышенного напряжения. Событие, по которому произошёл сброс, фиксируется в полях UVD33/UVD11 регистра RSTSTAT блока RCU.

5.3 Способы снижения энергопотребления

Для снижения энергопотребления микроконтроллера можно воспользоваться следующими возможностями:

– Выбрать в качестве источника системного тактового сигнала внутренний RC-генератор. При этом необходимо отключить сигнал осциллятора путем сброса бита EN регистра HSECFG блока RCU;

– Перевести системную ОЗУ в режим пониженного энергопотребления записью нулевого значения в поле RM и установкой бита RMEN регистра SSRAMMCR блока SIU.

– Отключить неиспользуемые области системной ОЗУ с помощью установки битов LSx (где x – выбранная область) регистра SSRAMSCR;

– Отключить память ОЗУ неиспользуемых периферийных блоков с помощью регистра PSRAMSMCR;

– Перевести блоки кэш памяти в режим пониженного энергопотребления записью нулевого значения в поле RM и установкой бита RMEN регистра CCSMCR блока SIU;

– Отключить блоки памяти кэша инструкций и/или кэша данных, если они не используются, с помощью регистра CCSMCR;

– Перевести ядро микроконтроллера в состояние IDLE командой *asm volatile ("WFI");*.

Примечание – Следует отметить, что отключаться должны только неиспользуемые области системной ОЗУ. Следовательно, при сборке проекта, в скрипте линкера должен быть указан только реально используемый объем ОЗУ, а не полный.

6 Блок управления сбросом и синхронизацией RCU

6.1 Общая система тактирования

В микроконтроллере предусмотрена развитая подсистема управления тактовыми сигналами и сигналами сброса, см. рисунок 6.1.

Управляющие регистры блока RCU приведены в приложении А.1.

Основными тактовыми сигналами, которые используются в микросхеме, являются:

– LSICLK – внутренний тактовый сигнал частотой 32,768 кГц (тактовый сигнал внутреннего RC-генератора);

– LSECLK – внешний низкочастотный тактовый сигнал частотой 32,768 кГц с выводов XI_RTC и XO_RTC;

– EXTCLK – внешний тактовый сигнал с выводов GPIOA3 или GPIOD6;

– HSICLK – внутренний высокочастотный тактовый сигнал частотой 4 МГц (тактовый сигнал внутреннего RC-генератора);

– HSECLK – внешний тактовый сигнал с выводов XI_OSC и XO_OSC;

– PLL0CLK – тактовый сигнал с выхода PLL0 для тактирования ядра и/или периферийных блоков;

– PLL1CLK – тактовый сигнал с выхода PLL1 для тактирования блоков SDRAM, I2S, USB, Ethernet, CANFD;

– PLL2CLK – тактовый сигнал с выхода PLL2 для тактирования блоков SDRAM, I2S, USB, Ethernet, CANFD;

– CLKOUT – выходной тактовый сигнал, источник которого выбирается селектором CLKSEL в регистре CLKOUTCFG;

– SYSCLK – системный тактовый сигнал, определяющий частоту работы процессорного ядра. Источник сигала SYSCLK выбирается битовым полем SRC регистра SYSCLKCFG (см. Приложение А.1);

Линии внешних сигналов EXTCLK, CLKOUT подключены к выводам портов GPIO и управляются с помощью альтернативных функций портов.

CG (Clock Gating) – разрешение подачи тактового сигнала к блокам. Разрешение управляется установкой бит в регистрах CGCFGAHB и CGCFGAPB.

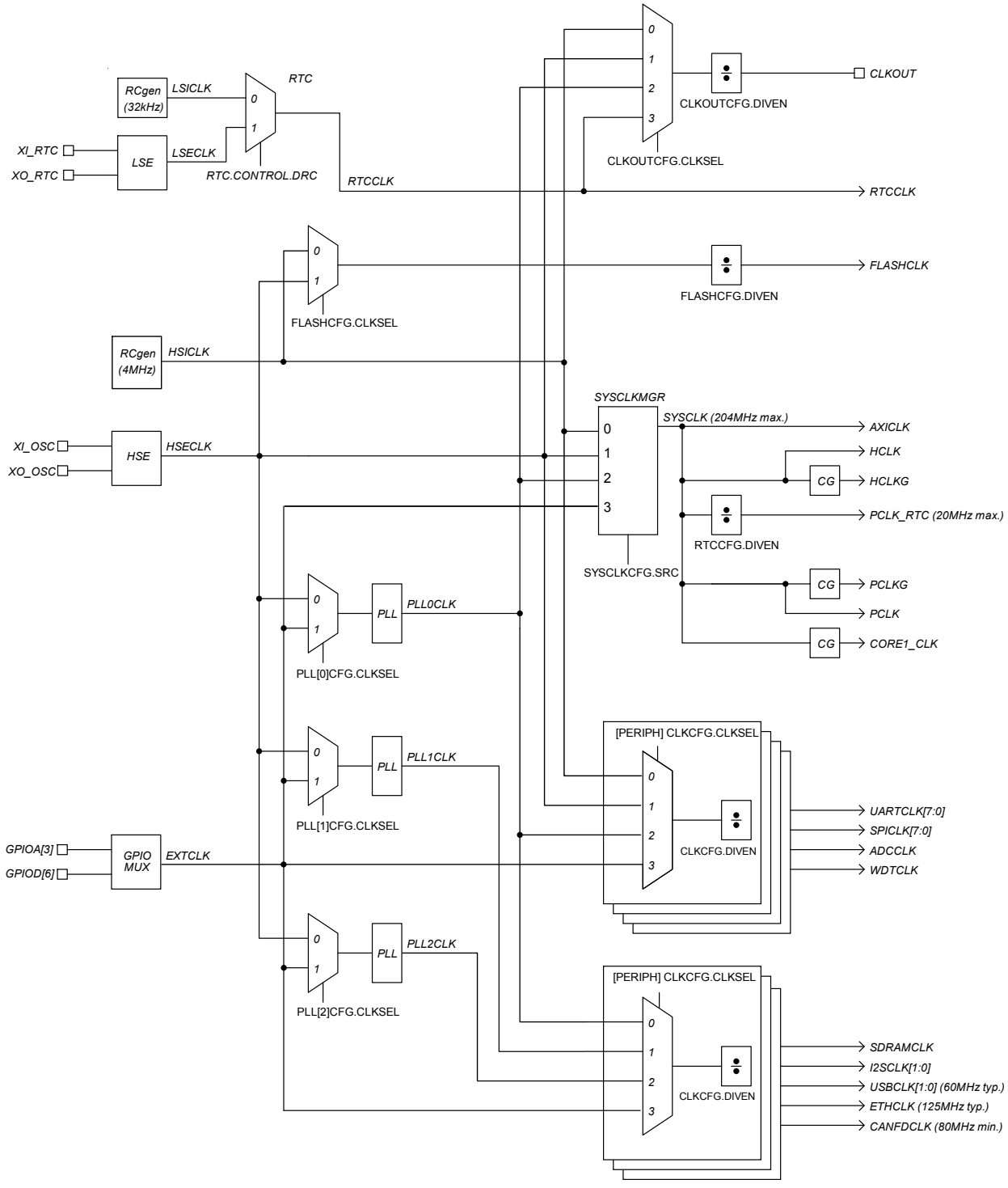


Рисунок 6.1 – Схема синхронизации

6.2 Синтезатор частоты PLL

Блок PLL представляет собой синтезатор частот со встроенным широкополосным ГУН (генератор, управляемый напряжением) на основе дифференциального кольцевого генератора. Блок PLL имеет ультранизкое энергопотребление, позволяет работать в режиме дробного и целочисленного деления частоты, имеет возможность отключения сигнала на выходе, обеспечивает работу в режиме «на проход».

Общая структурная схема блока PLL показана на рисунке 6.2.

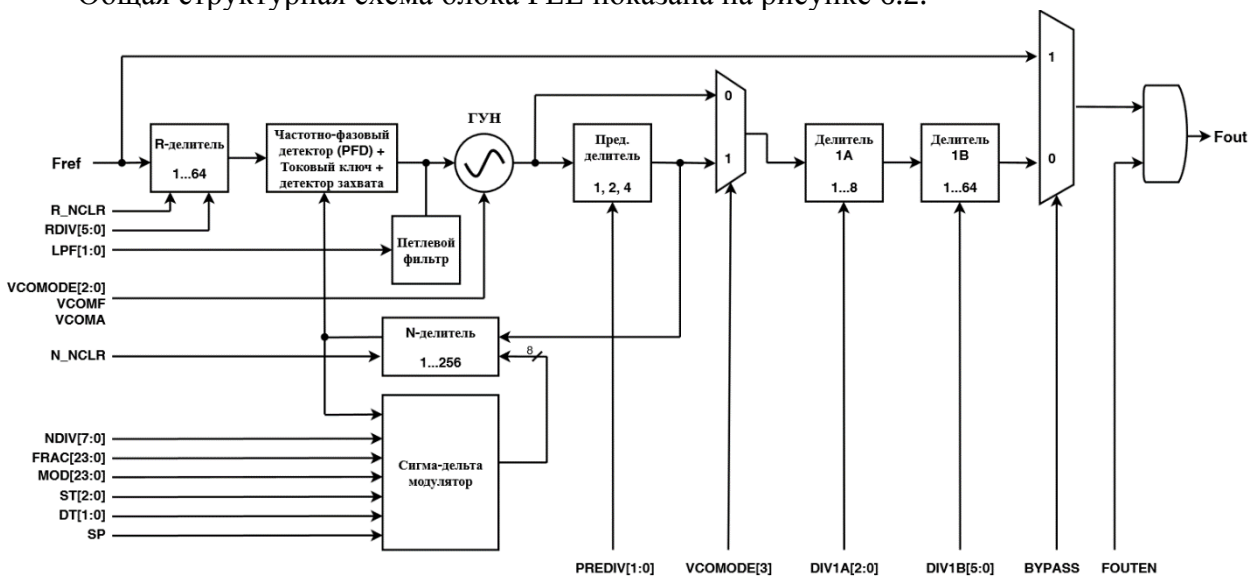


Рисунок 6.2 – Структурная схема блока PLL

PLL имеет два режима работы предварительного делителя частоты:

1) Предварительный делитель исключен из цепи выходного сигнала (бит VCOMODE[3] = 0);

2) Предварительный делитель включен в цепь выходного сигнала (бит VCOMODE[3] = 1).

В случае исключения предварительного делителя из цепи выходного сигнала частота сигнала F_{OUT} на выходе блока PLL определяется по формуле 6.1:

$$Fout = \frac{Fvco}{(DIV1A+1)\times(DIV1B+1)}, \quad (6.1)$$

где *Fvco* – частота сигнала на выходе ГУН, определяемая по формуле 6.3;
DIV1A – коэффициент деления делителя 1A;
DIV1B – коэффициент деления делителя 1B.

В случае включения предварительного делителя в цепь выходного сигнала частота сигнала F_{OUT} на выходе блока PLL определяется по формуле 6.2:

$$Fout = \frac{Fvco}{PR\times(DIV1A+1)\times(DIV1B+1)}, \quad (6.2)$$

где *Fvco* – частота сигнала на выходе ГУН, определяемая по формуле 6.3;
PR – коэффициент деления предварительного делителя (1, 2, 4);
DIV1A – коэффициент деления делителя 1A;
DIV1B – коэффициент деления делителя 1B.

Частота сигнала на выходе ГУН в любом из режимов работы предварительного делителя определяется по формуле 6.3:

$$Fvco = \frac{Fref \times N \times PR}{RDIV+1}, \quad (6.3)$$

где *Fref* – частота опорного сигнала;

N – коэффициент деления N-делителя в цепи обратной связи ГУН (диапазон значений от 1 до 256);

PR – коэффициент деления предварительного делителя (1, 2, 4);

RDIV – коэффициент деления опорного делителя (диапазон значений от 1 до 64).

N-делитель может работать в целочисленном (ST = 0) или дробном режиме (более подробно об установке того или иного режима можно прочесть в описании битового поля ST регистра PLLCFG в Приложении А.1).

В случае целочисленного режима значение N определяется только целой частью: *N = NDIV*.

В случае дробного режима значение N определяется по формуле 6.4:

$$N = NDIV + \frac{FRAC}{MOD}, \quad (6.4)$$

где *NDIV* – целая часть коэффициента деления;

FRAC – числитель дробной части коэффициента деления (диапазон значений от 0 до MOD - 1);

MOD – знаменатель дробной части коэффициента деления (диапазон значений от 2 до 2^{24} - 1).

Формула 6.4, равно как и диапазоны допустимых значений FRAC и MOD, справедливы для всех порядков сигма-дельта модулятора (режим работы сигма-дельта модулятора определяется битовым полем ST регистра PLLCFG).

Коэффициенты, *DIV1A*, *DIV1B*, *RDIV*, *NDIV* задаются соответствующими полями регистра PLLDIV. Коэффициент *PR* определяется полем *PREDIV* регистра PLLDIV. Коэффициент *FRAC* задается соответствующим полем регистра PLLFRAC. Коэффициент *MOD* задается значением регистра PLLMOD.

Существуют следующие ограничения параметров:

1) Значение входной опорной частоты Fref должно находиться в диапазоне от 10 МГц до 100 МГц;

2) Значение частоты на входе частотно-фазового детектора (PFD) должно находиться:
   - В режиме целочисленного деления от 10 МГц до Fvco / 16;
   - В режиме дробного деления от 10 МГц до Fvco / 20;

3) Коэффициент деления делителя обратной связи:
   - В режиме целочисленного деления $1 \leq N \leq 160$;
   - В режиме дробного деления $20 \leq N \leq 160$;

4) Значение частоты на выходе ГУН (*Fvco)* должно находиться в пределах 400 – 1000 МГц;

5) Значение выходной частоты Fout должно находиться в диапазоне от 200 кГц до 204 МГц.

Настройку блока PLL необходимо делать до того, как его тактовый сигнал будет выбран в качестве рабочей частоты системы или одного из блоков. Первоначально необходимо отключить прохождение входного сигнала сквозь блок PLL путём сброса BYPASS бита в регистре PLLCFG, а также вывести блок PLL из спящего режима путем сброса бита PD в регистре PLLCFG. Затем установить делители PREDIV, DIV1A, DIV1B, RDIV, NDIV в регистре PLLDIV соблюдая указанные выше ограничения. Биты RNCLR и NNCLR должны быть установлены либо предварительно, либо одновременно с установкой значения в полях делителей RDIV и NDIV. При использовании дробного режима необходимо выбрать режим работы сигма-дельта модулятора в поле ST регистра PLLCFG, а также записать коэффициенты FRAC и MOD в регистры PLLFRACи PLLMOD. При использовании целочисленного режима поле ST должно иметь нулевое значение.

Следующим шагом следует выбрать источник опорного тактового сигнала с помощью бита CLKSEL, установить режим работы ЧФД и токового ключа в рабочий режим (значение 11b), выбрать поддиапазон ГУН (рекомендуемое значение – 11b) и разрешить генерацию выходной частоты путем установки бита FOUTEN в регистре PLLCFG.

При правильной установке всех значений и выходе блока PLL на рабочий режим будет установлен бит LOCK в регистре PLLSTAT.

6.3 RC-генератор

В состав микроконтроллера входит RC-генератор сигнала прямоугольной формы.

Предусмотрена подстройка частоты с помощью регистра RCCTRL блока SIU. С помощью схемы выбора частоты меняется конфигурация схемы включения время задающей цепи: в общую схему генератора добавляются или исключаются добавочные конденсаторы, что оказывает влияние на время переключения задающего инвертора и на частоту переключения всего генератора в целом.

Частота генерации зависит от значений битового поля TUNE регистра RCCTRL и может изменяться в пределах от 3,894 МГц при TUNE равном 0 до 2,553 МГц при TUNE равном 255 (FF'h) (при напряжении питания 1,1 В и температуре 25°С). По умолчанию, значение TUNE = 66.

В таблице 6.1 представлены соответствия значений регистра RCCTRL частоте генерации (при напряжении питания 1,1 В и температуре 25°С). В таблице 6.2 представлены электрические характеристики RC генератора.

Таблица 6.1 – Соответствие частоты значениям поля TUNE регистра RCCTRL

TUNE	Частота, МГц	TUNE	Частота, МГц	TUNE	Частота, МГц	TUNE	Частота, МГц
0	3,894	32	3,567	64	3,477	96	3,229
1	3,830	33	3,515	65	3,427	97	3,187
2	3,800	34	3,491	**66**	**3,403**	98	3,167
3	3,753	35	3,452	67	3,367	99	3,136
4	3,774	36	3,470	68	3,383	100	3,150
5	3,717	37	3,423	69	3,340	101	3,113
6	3,699	38	3,407	70	3,325	102	3,100
7	3,656	39	3,372	71	3,291	103	3,071
8	3,717	40	3,422	72	3,338	104	3,112
9	3,660	41	3,376	73	3,294	105	3,074
10	3,634	42	3,354	74	3,273	106	3,057
11	3,592	43	3,318	75	3,240	107	3,028
12	3,621	44	3,343	76	3,263	108	3,048
13	3,570	45	3,301	77	3,223	109	3,013
14	3,553	46	3,286	78	3,209	110	3,001
15	3,515	47	3,254	79	3,179	111	2,975
16	3,670	48	3,394	80	3,302	112	3,089
17	3,615	49	3,347	81	3,259	113	3,051
18	3,589	50	3,324	82	3,238	114	3,033
19	3,547	51	3,290	83	3,205	115	3,005
20	3,566	52	3,306	84	3,221	116	3,018
21	3,517	53	3,265	85	3,181	117	2,984
22	3,501	54	3,250	86	3,168	118	2,973
23	3,463	55	3,219	87	3,138	119	2,946
24	3,526	56	3,271	88	3,187	120	2,990
25	3,475	57	3,229	89	3,147	121	2,955
26	3,452	58	3,210	90	3,129	122	2,939
27	3,415	59	3,178	91	3,099	123	2,913
28	3,441	60	3,200	92	3,120	124	2,931
29	3,396	61	3,162	93	3,084	125	2,899
30	3,381	62	3,149	94	3,071	126	2,889
31	3,346	63	3,120	95	3,044	127	2,865

Продолжение таблицы 6.1

TUNE	Частота, МГц	TUNE	Частота, МГц	TUNE	Частота, МГц	TUNE	Частота, МГц
128	3,302	160	3,073	192	3,015	224	2,832
129	3,258	161	3,035	193	2,979	225	2,801
130	3,237	162	3,017	194	2,962	226	2,785
131	3,204	163	2,989	195	2,934	227	2,762
132	3,219	164	3,002	196	2,947	228	2,772
133	3,179	165	2,968	197	2,914	229	2,744
134	3,166	166	2,957	198	2,903	230	2,734
135	3,135	167	2,931	199	2,878	231	2,712
136	3,178	168	2,967	200	2,913	232	2,743
137	3,138	169	2,933	201	2,881	233	2,714
138	3,120	170	2,917	202	2,865	234	2,701
139	3,090	171	2,891	203	2,840	235	2,680
140	3,111	172	2,910	204	2,857	236	2,695
141	3,074	173	2,878	205	2,827	237	2,668
142	3,062	174	2,868	206	2,817	238	2,659
143	3,034	175	2,844	207	2,794	239	2,639
144	3,145	176	2,946	208	2,887	240	2,726
145	3,107	177	2,912	209	2,854	241	2,697
146	3,087	178	2,896	210	2,838	242	2,683
147	3,058	179	2,870	211	2,814	243	2,662
148	3,071	180	2,882	212	2,826	244	2,671
149	3,036	181	2,851	213	2,795	245	2,646
150	3,025	182	2,841	214	2,786	246	2,638
151	2,997	183	2,817	215	2,763	247	2,617
152	3,042	184	2,857	216	2,800	248	2,650
153	3,006	185	2,826	217	2,770	249	2,624
154	2,989	186	2,811	218	2,756	250	2,611
155	2,962	187	2,787	219	2,734	251	2,591
156	2,981	188	2,803	220	2,750	252	2,605
157	2,948	189	2,775	221	2,722	253	2,580
158	2,937	190	2,765	222	2,713	254	2,572
159	2,912	191	2,743	223	2,691	255	2,553

Таблица 6.2 – Электрические характеристики RC-генератора

№ п/п	Наименование параметра	Норма		
		Не менее	Номинал	Не более
1	Диапазон подстроки частоты, %	–	-29 / +12,6	–
2	Нестабильность по температуре, %	–	-1,1 / +2,1	–
3	Ток потребления, мкА	–	7	70
Примечание – Значения при напряжении LDO 1,1 В и температуре 25 °С				

6.4 Осциллятор HSE

В состав микроконтроллера входит внутренний осциллятор HSE с подключаемым внешним кварцевым резонатором частотой от 2 до 27 МГц. Выводы микроконтроллера XI_OSC и XO_OSC служат для подключения кварцевого резонатора.

На рисунке 6.3 показан колебательный контур, подключаемый к осциллятору HSE.

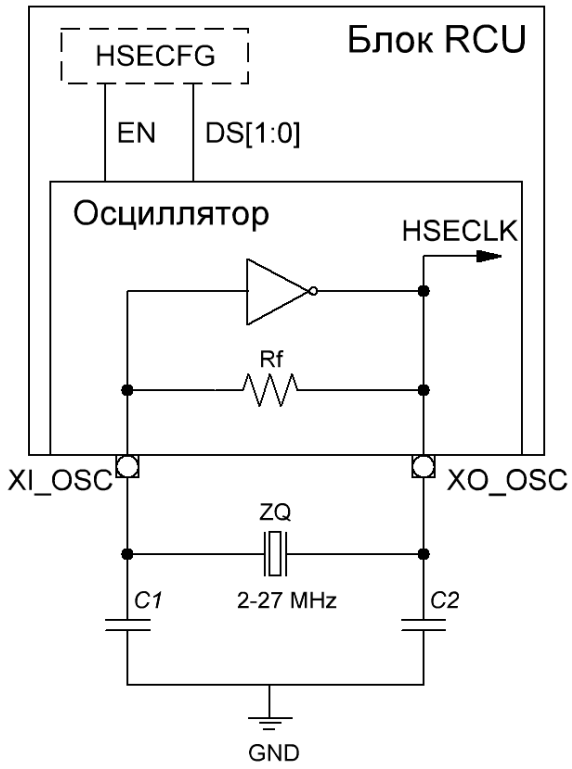


Рисунок 6.3 – Схема колебательного контура осциллятора

Внутренний резистор Rf используется для смещения инвертора в область с высоким коэффициентом усиления.

Конденсаторы C1 и C2 используются для регулировки времени включения, поддержания стабильности и точности работы осциллятора. Рекомендуемые параметры конденсаторов в схеме колебательного контура, в зависимости от генерируемой частоты, приведены в таблице 6.3.

При трассировке печатной платы необходимо устанавливать конденсаторы и резонатор как можно ближе к соответствующим выводам микроконтроллера. Рекомендуется использовать многослойные керамические конденсаторы с диэлектриком типа C0G (NP0), но также допускается использовать X7R с рабочим напряжением не ниже 6.3 В для типоразмеров 0402 или 0603.

Таблица 6.3 – Рекомендуемые параметры конденсаторов

Рекомендуемый параметр	Частота 2 МГц	Частота 27 МГц
Емкость конденсаторов C1, C2, пФ	50	16
Максимальное эквивалентное сопротивление (ESR), Ом	1000	40

Время запуска осциллятора – не более 30 мс.

6.5 Система слежения за тактовыми сигналами

Система слежения осуществляет контроль источников тактовых сигналов и позволяет обрабатывать исключительные ситуации, связанные с их пропаданием (срыв генерации блока PLL, ненадежный контакт с внешним резонатором и т. п.).

Текущий статус тактовых сигналов можно установить, прочитав соответствующие биты CLKGOODx и CLKERRx регистра CLKSTAT, где x – HSICLK, HSECLK, PLL0CLK, EXTCLK. Бит CLKGOODx будет установлен, если соответствующий тактовый сигнал стабильно работает, и будет сброшен при его сбое. Бит CLKERRx имеет обратное поведение – при сбое устанавливается, сбрасывается при нормальной работе тактового сигнала.

Время реакции системы слежения на исчезновение тактового сигнала настраивается с помощью полей регистров SECCNT0, SECCNT1, и по умолчанию равно 65536 тактам сигнала HSICLK. Реакция на любое пропадание тактового сигнала на меньшее время будет отсутствовать.

Минимальная частота, за которой может осуществляться слежение – 800 Гц.

Значение минимально возможного времени реакции вычисляется по формуле (6.5) (результат округляется до целого в большую сторону).

$$SECCNT_MIN = (4 \times T_{REF} + 6 \times T_{CLK})/T_{REF}, \quad (6.5)$$

где T_{REF} – период опорной тактовой частоты сигнала HSICLK;

T_{CLK} – период контролируемой тактовой частоты.

Например, если необходима максимально быстрая реакция на пропадание частоты 100 МГц на выходе блока PLL (T_{CLK} = 10 нс) при частоте сигнала HSICLK 4 МГц (T_{REF} = 250 нс), то необходимо в поле VAL2 регистра SECCNT1 записать значение SECCNT_MIN = 5, вычисленное по формуле (6.5).

Стоит отметить, что SECCNT_MIN является лишь временем детектирования сбоя. Сам переход в аварийный режим (переключение системной частоты на сигнал HSICLK) займет дополнительно не более 14 тактов сигнала HSICLK.

Кроме слежения за каждым из источников тактового сигнала по отдельности, система отслеживания позволяет контролировать текущий системный тактовый сигнал SYSCLK.

Включение контроля сигнала SYSCLK осуществляется установкой бита SECEN регистра SYSCLKCFG.

При сбое во время работы от PLL будет осуществлен аварийный переход системной частоты на сигнал HSICLK и будет вызвано прерывание RCU совместно с установкой флага прерывания PLLFAIL в регистре INTSTAT (флаг сбрасывается записью единицы).

Прямой переход из аварийного состояния к тактированию от вызвавшего сбой, но уже восстановившегося источника – невозможен. Если это необходимо сделать, то сначала нужно перейти на один из стабильных источников (например, сигнал HSICLK), и лишь затем начать переход на бывший «сбойным», но восстановившийся источник.

При переходе микроконтроллера в аварийное состояние по тактированию требуется обязательно перейти на стабильный источник тактирования (бит CLKGOODx для него будет установлен), например, на сигнал HSICLK (SYSSEL = 0). Если аварийный источник возобновил тактирование, то снова перейти на него можно, записав поле SYSSEL регистра SYSCLKCFG на номер соответствующего источника.

6.6 Сигналы сброса

При включении питания сброс контроллера осуществляется с помощью блока POR.

Также сброс может выполняться по следующим событиям:

– Внешним сигналом с вывода RST (активный уровень сигнала сброса – низкий);

– Сигналом от сторожевого таймера WDT;

– Сигналом от блока BOR;

– При смене режимов работы ядра (Lockstep ↔ Dualcore);

– Сигналами от мониторов питания UVD33, UVD11 (см. подраздел 5.2);

– Сигнал таймаута при возникновении неисправности (значение определяется регистром TIMEOUT блока SIU);

– Программный сброс с помощью регистра RSTSYS.

Событие, по которому произошел сброс, можно определить с помощью регистра RSTSTAT.

6.7 Прерывания

Микроконтроллер поддерживает прерывания, связанные с работой блока RCU.

Выбрать источники, влияющие на него можно в регистре INTEN – события появления или пропадания одного из тактовых сигналов, выход блока PLL в рабочий режим, а также ошибка установления тактового сигнала с выхода PLL. Все события сопровождаются флагами в регистре INTSTAT.

6.8 Тактирование и сброс периферийных блоков

Для некоторых периферийных блоков система тактирования предусматривает выбор независимого тактового источника. У блоков UART, SPI, ADC, WDT, I2S, USB, ETH, SDRAM, CANFD есть соответствующие регистры, в которых можно включать тактирование, выбирать его источник и настраивать отключаемый делитель. Источник тактирования выбирается независимо для каждого из блоков с помощью его собственного селектора, показанного на рисунке 6.4.

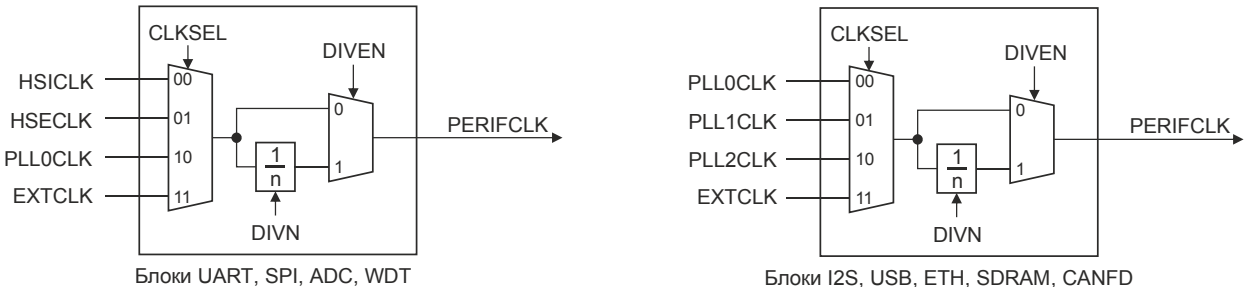


Рисунок 6.4 – Схема селекторов дополнительного синхросигнала блоков

Для остальных блоков, а именно для блоков, относящихся к APB и AHB периферии, тактирование подается установкой соответствующего бита в CGCFGAPB (CGCFGAHB), см. таблицу 6.4.

Таблица 6.4 – Тактовые сигналы периферийных блоков

Периферийный блок	Тактовый сигнал	Делитель	Управляющий регистр
ADC	ADCCLK	1-63	ADCCFG, CGCFGAHB, RSTDISAHB
SPI	SPICLK	1-63	SPICFG, CGCFGAPB2, RSTDISAPB2
UART	UARTCLK	1-63	UARTCFG, CGCFGAPB2, RSTDISAPB2
WDT	WDTCLK	1-63	WDTCFG
I2S	I2SCLK	1-63	I2SCFG, CGCFGAPB1, RSTDISAPB1
FLASH	FLASHCLK	1-63	FLASHCFG
CLOCKOUT	CLKOUT	1-65535	CLKOUTCFG
USB	USBCLK	1-63	USBCFG, CGCFGAHB, RSTDISAHB
ETH	ETHCLK	1-63	ETHCFG, CGCFGAPB0, RSTDISAPB0
CANFD	CANFDCLK	1-63	CANFDCFG, CGCFGAHB, RSTDISAHB
SDRAM	SDRAMCLK	1-63	SDRAMCFG, CGCFGAPB0, RSTDISAPB0
RTC	PCLK	1-63	RTCCFG, CGCFGAPB0, RSTDISAPB0
GPIO	HCLK	–	CGCFGAHB, RSTDISAHB
CAN	HCLK	–	CGCFGAHB, RSTDISAHB
QSPI	HCLK	–	CGCFGAHB, RSTDISAHB
HASH	HCLK	–	CGCFGAHB, RSTDISAHB
CRYPTO	HCLK	–	CGCFGAHB, RSTDISAHB
CRC	HCLK	–	CGCFGAHB, RSTDISAHB
EMC	PCLK	–	CGCFGAPB0, RSTDISAPB0
TRNG	PCLK	–	CGCFGAPB0, RSTDISAPB0
WDT	PCLK	–	CGCFGAPB0, RSTDISAPB0
DMA	PCLK	–	CGCFGAPB0, RSTDISAPB0
TMR	PCLK	–	CGCFGAPB1, RSTDISAPB1
DAC	PCLK	–	CGCFGAPB1, RSTDISAPB1
ACMP	PCLK	–	CGCFGAPB1, RSTDISAPB1
I2C	PCLK	–	CGCFGAPB2, RSTDISAPB2
UART	PCLK	–	CGCFGAPB2, RSTDISAPB2
SPI	PCLK	–	CGCFGAPB2, RSTDISAPB2
LIN	PCLK	–	CGCFGAPB2, RSTDISAPB2

По умолчанию на всех периферийных блоках отключено тактирование, и все они находятся в состоянии сброса. Для начала работы нужно подать тактовый сигнал, а также вывести блок из состояния сброса, осуществив запись единиц в соответствующие биты регистров CGCFGAHB, CGCFGAPB[2-0] и RSTDISAHB, RSTDISAPB[2-0].

7 Блок управления системой SIU

7.1 Описание блока

Блок управления системой содержит набор регистров, позволяющий управлять взаимодействием некоторых модулей на системном уровне, а также настройки отдельных аналоговых блоков.

Управляющие регистры блока SIU приведены в приложении А.2.

Регистры блока управления системой позволяют:

- перейти в режим «dualcore» и обратно в случае разрешения в конфигурационном слове, выполнить настройки второго ядра;
- задать режим низкого энергопотребления для памяти;
- определить находится ли микроконтроллер в сервисном режиме и, если это так, выполнить сервисное стирание памяти;
- выполнить подстройку частоты RC-генератора;
- выполнить подстройку LDO;
- задать режим работы шкалы датчика температуры;
- настроить синхронизацию блоков таймеров и ШИМ;
- выбрать тип контроллера внешней памяти: NOR/NAND Flash (EMC) или SDRAM;
- проанализировать возникшие неисправности;
- узнать о версии микроконтроллера.

В регистре CHIPID содержится идентификатор микросхемы, а его младшие 4 бита указывают номер ревизии.

Регистр SERVCTL управляет стиранием памяти в сервисном режиме (если при старте микроконтроллера вывод SERVEN был в состоянии логической единицы). В данном режиме Flash-память блокируется для чтения и выполнения, доступно только полное стирание всей памяти, включая NVR область. Этот механизм можно использовать, например, для стирания памяти, после того как в конфигурационном слове была запрещена работа JTAG.

С помощью регистра PWMSYNC можно управлять цепочками синхронизации блоков PWM. Подробнее о синхронизации описано в подразделе 18.1 «Синхронизация таймеров блоков ШИМ».

В регистрах PWMEN и TMREN можно одновременно запустить работу нескольких блоков PWM и TMR, обеспечив их синхронный счет.

В регистре TMRCFG можно связать вход внешнего тактирования таймера с выходом предыдущего. Переключение между сигналом с внешней ножки TMR_IO и выходом предыдущего таймера осуществляется с помощью мультиплексора TMRxCLKMUX. Для таймера TMR0 предыдущим является таймер TMR15.

Регистр RCCTRL управляет отключением и подстройкой внутреннего RC-генератора. Подробнее описано в подразделе 6.3.

Регистр LDOCTRL управляет подстройкой выходного напряжения внутреннего регулятора. Таблица коррекции представлена в приложении А.2 в описании регистра.

Регистры SSRAMSCR, SSRAMMCR, PSRAMSMCR, CCSMCR управляют режимами пониженного потребления памяти.

Регистр EMMUXCTRL позволяет выбрать между использованием контроллеров внешней памяти EMC или SDRAM.

7.2 Сборщик неисправностей

Сборщик неисправностей предоставляет систематический подход для определения и управления неисправностями, а также позволяет поместить систему в безопасное состояние. Управление осуществляется через набор регистров:

- INTEN, RIS, MIS, ICLR: позволяют включить прерывания по неисправностям и проанализировать их состояние;
- FRSTEN, FSTAT: регистры разрешают опциональный сброс по тайм-ауту и показывают состояние флагов (см. подраздел 7.3 «Описание работы»);
- TIMEOUT: устанавливает значение тайм-аута для сброса;
- NODESTEST: запускает режим самотестирования узлов lockstep и собирает информацию о результатах;
- NODExECNT, где x = 0..4: счётчик количества ошибок, полученных от узлов lockstep;
- NODExCTRL, где x = 0..4: регистр управления узлом, позволяет настроить лимит ошибок по которому произойдёт прерывание/сброс, посмотреть статус блоков сравнения и общий флаг ошибок.

7.3 Описание работы

При установленных битах в регистре FRSTEN и при возникновении соответствующих условий, а именно понижении/повышении напряжений питания или сбой блоков сравнения узлов, запускается внутренний счётчик и конечный автомат переходит из состояния "NORMAL" в состояние "ALARM". Счётчик считает тактами системной частоты до значения в регистре TIMEOUT. При переполнении счётчика конечный автомат переходит из состояния "ALARM" в состояние "FAULT" и вырабатывается сброс который фиксируется в блоке RCU регистре RSTSTAT поле FAULT.

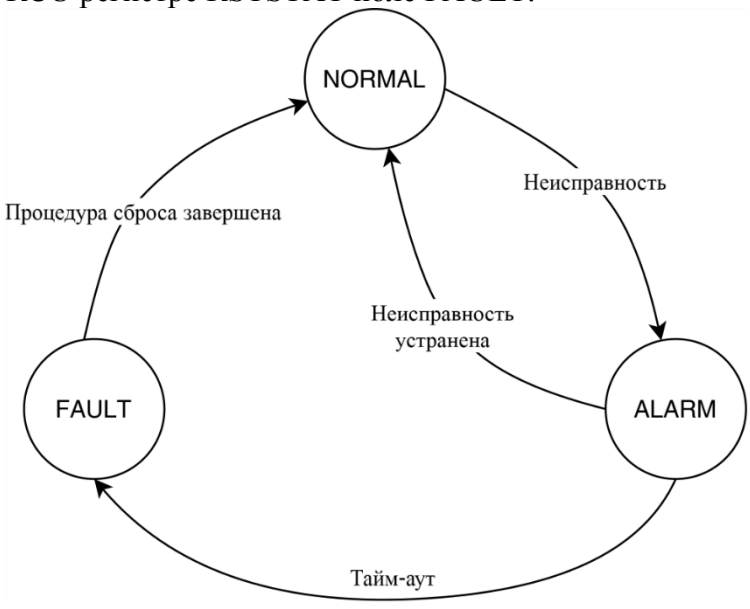


Рисунок 7.1 – Машина состояний сборщика неисправностей

Отменить счёт счётчика и, соответственно, сброс, возможно только приведением системы в нормальное состояние:

- в случае возникновения неисправности по питанию - восстановить номинальные значения (DVCC - 1.1V, AVCC - 3.3V);
- в случае сбоя блоков сравнения узлов (ошибка ядра в режиме «lockstep») необходимо выполнить сброс регистров NODExCTRL и программно проанализировать ситуацию.

В этом случае счёт прекращается и конечный автомат переходит из состояния "ALARM" в состояние "NORMAL".

7.4 Тайм-аут при зависании шин

Ещё одной детектируемой неисправностью является зависание внутренних шин при обращении к периферийному устройству. Если при выполнении операций чтения или записи какой-либо блок не возвращает сигнал готовности за 4095 тактов системной частоты, блок коммутации микроконтроллера автоматически завершает транзакцию с ошибкой типа «AXI SLAVE ERROR». Получение такого рода ошибок ядром приводит к исключению (exception) и в статусных регистрах ядра сохраняются следующие данные:

- регистр MCAUSE принимает значение 0x7 (store/load access fault);
- регистр MTVAL содержит адрес, при обращении к которому произошло исключение;
- регистр MEPC содержит адрес инструкции, при исполнении которой произошло исключение.

Дополнительно, в регистре INTEN блока SIU (поле BUSTIMEOUT) можно разрешить прерывание от блока SIU по возникновению зависания (имеет меньший приоритет чем исключение).

8 Организация памяти

Память микроконтроллера имеет предопределенное 32-разрядное адресное пространство с областями: программ, данных, периферии и внутренних ресурсов, жестко соединенных с процессором. Микроконтроллер начинает выполнение программы с адреса 0000_0000h. Адресное пространство разбито на несколько областей, см. таблицу 8.1.

Таблица 8.1 – Общая организация памяти микроконтроллера

Адресное пространство	Описание
FE00_0000h – FFFF_FFFFh	Системная область PLIC
8000_0000h – 81FF_FFFFh	Внешняя память SDRAM
6000_0000h – 6FFF_FFFFh	Внешняя память (восемь последовательных окон по 16 Мбайт)
4000_0000h – 52FF_FFFFh	Регистры управления периферийными блоками
2000_0000h – 200F_FFFFh	Внутреннее ОЗУ (SRAM) 1 Мбайт с ECC
0040_0000h – 0047_FFFFh	FlashD-память 512 Кбайт с ECC
0000_0000h – 003F_FFFFh	FlashM-память 4 Мбайт с ECC

Регистры периферийных блоков микроконтроллера доступны в адресном пространстве 4000_0000h – 52FF_FFFFh. Карта размещения блоков микроконтроллера в памяти представлена в таблице 8.2.

Таблица 8.2 – Карта регистров периферийных блоков

Адресное пространство	Обозначение блока	Описание
5201_C000h – 5201_CFFFh	MSC	Контроллер MSC
5201_B000h – 5201_BFFFh	I2C3	Последовательный интерфейс I2C3
5201_A000h – 5201_AFFFh	I2C2	Последовательный интерфейс I2C2
5201_9000h – 5201_9FFFh	I2C1	Последовательный интерфейс I2C1
5201_8000h – 5201_8FFFh	I2C0	Последовательный интерфейс I2C0
5201_7000h – 5201_7FFFh	UART7	Последовательный приемопередатчик UART7
5201_6000h – 5201_6FFFh	UART6	Последовательный приемопередатчик UART6
5201_5000h – 5201_5FFFh	UART5	Последовательный приемопередатчик UART5
5201_4000h – 5201_4FFFh	UART4	Последовательный приемопередатчик UART4
5201_3000h – 5201_3FFFh	UART3	Последовательный приемопередатчик UART3
5201_2000h – 5201_2FFFh	UART2	Последовательный приемопередатчик UART2
5201_1000h – 5201_1FFFh	UART1	Последовательный приемопередатчик UART1
5201_0000h – 5201_0FFFh	UART0	Последовательный приемопередатчик UART0
5200_F000h – 5200_FFFFh	SPI7	Последовательный интерфейс SPI7
5200_E000h – 5200_EFFFh	SPI6	Последовательный интерфейс SPI6
5200_D000h – 5200_DFFFh	SPI5	Последовательный интерфейс SPI5
5200_C000h – 5200_CFFFh	SPI4	Последовательный интерфейс SPI4
5200_B000h – 5200_BFFFh	SPI3	Последовательный интерфейс SPI3

Продолжение таблицы 8.2

Адресное пространство	Обозначение блока	Описание
5200_A000h – 5200_AFFFh	SPI2	Последовательный интерфейс SPI2
5200_9000h – 5200_9FFFh	SPI1	Последовательный интерфейс SPI1
5200_8000h – 5200_8FFFh	SPI0	Последовательный интерфейс SPI0
5200_7000h – 5200_7FFFh	LIN7	Последовательный интерфейс LIN7
5200_6000h – 5200_6FFFh	LIN6	Последовательный интерфейс LIN6
5200_5000h – 5200_5FFFh	LIN5	Последовательный интерфейс LIN5
5200_4000h – 5200_4FFFh	LIN4	Последовательный интерфейс LIN4
5200_3000h – 5200_3FFFh	LIN3	Последовательный интерфейс LIN3
5200_2000h – 5200_2FFFh	LIN2	Последовательный интерфейс LIN2
5200_1000h – 5200_1FFFh	LIN1	Последовательный интерфейс LIN1
5200_0000h – 5200_0FFFh	LIN0	Последовательный интерфейс LIN0
5101_7000h – 5101_7FFFh	TMR15	Блок таймера 15
5101_6000h – 5101_6FFFh	TMR14	Блок таймера 14
5101_5000h – 5101_5FFFh	TMR13	Блок таймера 13
5101_4000h – 5101_4FFFh	TMR12	Блок таймера 12
5101_3000h – 5101_3FFFh	TMR11	Блок таймера 11
5101_2000h – 5101_2FFFh	TMR10	Блок таймера 10
5101_1000h – 5101_1FFFh	TMR9	Блок таймера 9
5101_0000h – 5101_0FFFh	TMR8	Блок таймера 8
5100_F000h – 5100_FFFFh	TMR7	Блок таймера 7
5100_E000h – 5100_EFFFh	TMR6	Блок таймера 6
5100_D000h – 5100_DFFFh	TMR5	Блок таймера 5
5100_C000h – 5100_CFFFh	TMR4	Блок таймера 4
5100_B000h – 5100_BFFFh	TMR3	Блок таймера 3
5100_A000h – 5100_AFFFh	TMR2	Блок таймера 2
5100_9000h – 5100_9FFFh	TMR1	Блок таймера 1
5100_8000h – 5100_8FFFh	TMR0	Блок таймера 0
5100_7000h – 5100_7FFFh	I2S1	Последовательный интерфейс I2S1
5100_6000h – 5100_6FFFh	I2S0	Последовательный интерфейс I2S0
5100_5000h – 5100_5FFFh	DAC1	Блок ЦАП1
5100_4000h – 5100_4FFFh	DAC0	Блок ЦАП0
5100_3000h – 5100_3FFFh	ACMP3	Блок аналоговых компараторов 3
5100_2000h – 5100_2FFFh	ACMP2	Блок аналоговых компараторов 2
5100_1000h – 5100_1FFFh	ACMP1	Блок аналоговых компараторов 1
5100_0000h – 5100_0FFFh	ACMP0	Блок аналоговых компараторов 0
5001_C000h – 5001_CFFFh	PWM15	Блок ШИМ15
5001_B000h – 5001_BFFFh	PWM14	Блок ШИМ14
5001_A000h – 5001_AFFFh	PWM13	Блок ШИМ13
5001_9000h – 5001_9FFFh	PWM12	Блок ШИМ12
5001_8000h – 5001_8FFFh	PWM11	Блок ШИМ11
5001_7000h – 5001_7FFFh	PWM10	Блок ШИМ10
5001_6000h – 5001_6FFFh	PWM9	Блок ШИМ9
5001_5000h – 5000_5FFFh	PWM8	Блок ШИМ8
5001_4000h – 5000_4FFFh	PWM7	Блок ШИМ7
5001_3000h – 5000_3FFFh	PWM6	Блок ШИМ6
5001_2000h – 5000_2FFFh	PWM5	Блок ШИМ5
5001_1000h – 5000_1FFFh	PWM4	Блок ШИМ4

Окончание таблицы 8.2

Адресное пространство	Обозначение блока	Описание
5001_0000h – 5000_0FFFh	PWM3	Блок ШИМ3
5000_F000h – 5000_FFFFh	PWM2	Блок ШИМ2
5000_E000h – 5000_EFFFh	PWM1	Блок ШИМ1
5000_D000h – 5000_DFFFh	PWM0	Блок ШИМ0
5000_C000h – 5000_CFFFh	FLASHD	Блок управления FlashD-памятью с OTP
5000_B000h – 5000_BFFFh	ETH	Контроллер Ethernet
5000_A000h – 5000_AFFFh	SDRAM	Интерфейс внешней памяти SDRAM
5000_9000h – 5000_9FFFh	EMC	Интерфейс внешней памяти EMC
5000_8000h – 5000_8FFFh	TRNG	Блок генератора случайных чисел
5000_7000h – 5000_7FFFh	SRAM	Регистры управления SRAM ECC
5000_6000h – 5000_6FFFh	WDT	Сторожевой таймер
5000_5000h – 5000_5FFFh	RTC	Блок часов реального времени
5000_4000h – 5000_4FFFh	RCU	Блок управления сбросом и синхронизацией
5000_3000h – 5000_3FFFh	SIU	Блок управления системой
5000_2000h – 5000_2FFFh	FLASHM	Блок управления FlashM-памятью
5000_0000h – 5000_1FFFh	DMA	Контроллер прямого доступа к памяти
4109_1000h – 4109_1FFFh	USB1H	Контроллер хоста USB1
4109_0000h – 4109_0FFFh	USB1D	Контроллер устройства USB1
4108_1000h – 4108_1FFFh	USB0H	Контроллер хоста USB0
4108_0000h – 4108_0FFFh	USB0D	Контроллер устройства USB0
4107_0000h – 4107_0FFFh	CRYPTO	Блок криптографии
4106_0000h – 4106_0FFFh	HASH	Хеш процессор
4105_0000h – 4105_0FFFh	CRC	Блок подсчета циклического избыточного кода CRC
4104_0000h – 4104_0FFFh	QSPI1	Последовательный интерфейс QSPI1
4103_0000h – 4103_0FFFh	QSPI0	Последовательный интерфейс QSPI0
4102_0000h – 4102_FFFFh	ADC	Блок АЦП
4101_0000h – 4101_FFFFh	CAN	Контроллер CAN 2.0B
4100_3000h – 4100_3FFFh	CANFD3	Контроллер CANFD3
4100_2000h – 4100_2FFFh	CANFD2	Контроллер CANFD2
4100_1000h – 4100_1FFFh	CANFD1	Контроллер CANFD1
4100_0000h – 4100_0FFFh	CANFD0	Контроллер CANFD0
4000_6000h – 4000_6FFFh	GPIOG	Порт G
4000_5000h – 4000_5FFFh	GPIOF	Порт F
4000_4000h – 4000_4FFFh	GPIOE	Порт E
4000_3000h – 4000_3FFFh	GPIOD	Порт D
4000_2000h – 4000_2FFFh	GPIOC	Порт C
4000_1000h – 4000_1FFFh	GPIOB	Порт B
4000_0000h – 4000_0FFFh	GPIOA	Порт A

8.1 Модуль коррекции ошибок ECC SRAM

Внутренняя SRAM память поддерживает функционал коррекции ошибок Хемминга, позволяющий исправить одну ошибку и определить две ошибки. Каждый байт защищен четырьмя битами кода коррекции, соответственно 32-разрядное слово защищено 20 битами коррекции.

Для включения функционала коррекции ошибок необходимо установить бит EN регистра CFG модуля SRAM.

Определить количество обнаруженных ошибок можно с помощью регистров счетчиков SEU_CNTR и DEU_CNTR для одиночных и двойных ошибок соответственно.

Адрес слова, в котором обнаружена ошибка, можно определить с помощью регистров SEU_ADDR и DEU_ADDR для одиночных и двойных ошибок соответственно.

Также можно настроить прерывания при обнаружении одиночных и двойных ошибок с помощью бит SEUEN и DEUEN регистра IRQEN. Текущий статус прерываний можно узнать с помощью регистра IRQSTAT.

8.2 Модуль внедрения ошибок EIM

Модуль внедрения ошибок позволяет вставлять одиночную или многобитовую ошибку при записи, что позволяет проверить работоспособность системы ECC. Он позволяет внедрять ошибки за счёт пересчёта битов чётности ECC не изменяя исходных данных и имеет защиту от случайного включения или реконфигурации, используя двухступенчатый механизм включения.

Описание работы

Модуль внедрения ошибок не является отдельным периферийным модулем, он встроен в контроллеры FLASH и SRAM. Управляющий регистр (FLASHx->FECCCR или SRAM->CFG) позволяет включить ECC, инжекцию ошибок, разрешить прерывания по возникновению одиночной или двойной ошибки. Регистры адресов и счётчиков одиночной и двойной ошибки хранят адрес ячейки памяти в которой была определена одиночная или двойная ошибка и их количество (FLASHx->FECC(S/D)EC, FLASHx->FECC(S/D)EA, SRAM->(S/D)EU_CNTR и SRAM->(S/D)EU_ADDR).

Структурная схема блока внедрения ошибок приведена на рисунке 8.1. Инжекция ошибки осуществляется за счёт коррекции битов чётности во время операций записи или программирования. Данные для записи инвертируются в позициях (если установлен бит ERRINJEN в управляющем регистре), указанных в регистре SRAM->ERRMASK или массиве регистров FLASHx->ERRMASK[] и затем используются для расчёта новых битов чётности. Далее, исходные, неинвертированные данные вместе с обновлёнными битами чётности, рассчитанными в модуле внедрения ошибок, используются для записи в память. При чтении возникает желаемая ошибка ECC (одиночная или двойная).

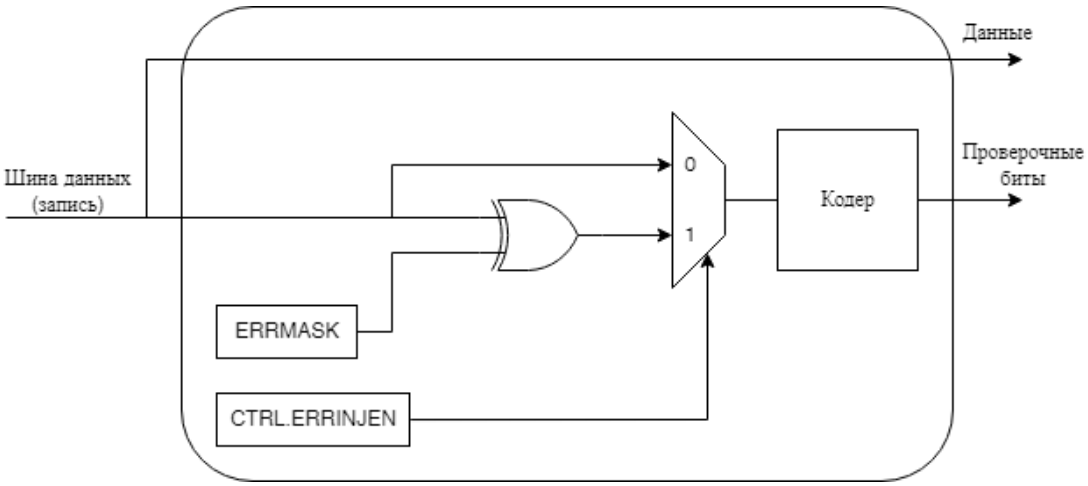


Рисунок 8.1 – Структурная схема блока внедрения ошибок (EIM)

Пример работы модуля внедрения ошибки

Пример работы модуля внедрения ошибки рассмотрим на основе контроллера FLASH.

1. Загрузка тестовых данных в массив регистров DATA:

```
for (int i = 0; i < 4; i++) {
FLASHx->DATA[i].DATA = test_data[i];
}
```

2. Загрузка маски в массив регистров ERRMASK (указывает в какой бит будет внесена ошибка):

```
FLASHx->ERRMASK[i].ERRMASK = 0x00000001; // inject single error, bit 0 in word i
```

или

```
FLASHx->ERRMASK[i].ERRMASK = 0x00010001; // inject double error, bit 0 & bit 16 in word i
```

3. Настройка регистра управления ECC и последующий запуск на программирование из командного регистра FLASHx:

```
FLASHx->FECCCR = (1 << FLASHx_FECCCR_ECCEN_Pos) | // ecc enable
                 (1 << FLASHx_FECCCR_ISEUEN_Pos) | // enable interrupt when single error
                 (1 << FLASHx_FECCCR_IDEUEN_Pos) | // enable interrupt when double error
                 (1 << FLASHx_FECCCR_ERRINJEN_Pos); // enable selftest mechanism
FLASHx->CMD = (0xc0de << FLASHx_CMD_KEY_Pos) |
                  (1 << FLASHx_CMD_WR_Pos);
while (FLASHx->STAT_bit.BUSY)
{};
```

4. После программирования линии FLASHx производится чтение:

```
FLASHx->CMD = (0xc0de << FLASHx_CMD_KEY_Pos) |
                 (1 << FLASHx_CMD_RD_Pos);
```

При возникновении ошибки, адрес возникновения заносится в регистр FLASHx→FECC(S/D)EA, инкрементируется счётчик ошибок FLASHx→FECC(S/D)EC, устанавливаются флаги в регистре FLASHx→FECCISTAT и, если сконфигурировано, возникает прерывание с флагом установленном в регистре FLASHx->STAT.

5. Для проверки что механизм определения и коррекции ошибок работает успешно следует для одиночной ошибки выполнить операцию “исключающее ИЛИ” (XOR) прочитанных из FLASHx данных со значением маски и сверить с изначально заданными данными:

```
for (int i = 0; i < 4; i++) {
   result_data = FLASHx->DATA[i].DATA ^ FLASHx->ERRMASK[i].ERRMASK;
   if (result_data != test_data[i]) {
      err_counter = err_counter + 1;
   }
}
```

Для двойной ошибки, которая только детектируется и при которой коррекция данных не происходит, следует проверить идентичность прочитанных из FLASHx данных с изначально заданными:

```
for (int i = 0; i < 4; i++) {
   result_data = FLASHx->DATA[i].DATA;
   if (result_data != test_data[i]) {
      err_counter = err_counter + 1;
   }
}
```

9 Контроллер Flash-памяти

9.1 FlashM-память

FlashM-память может использоваться для хранения программ и данных пользователя. Размер основной FlashM-памяти составляет 4 Мбайт (4 банка по 1 Мбайт, 2048 страниц по 2 Кбайта), и в адресном пространстве она занимает диапазон с 0000_0000h по 003F_FFFFh.

Управляющие регистры контроллера FlashM-памяти приведены в приложении А.3.

Чтение FlashM-памяти осуществляется через шину AXI. При попытке записи в область FlashM-памяти происходит выдача ответа SLAVE ERROR. При чтении из несуществующей области транзакция проходит успешно с неопределенными данными на выходе. Формирование ответа на запрос чтения, который пришел в процессе стирания или программирования Flash, будет отложено до завершения текущей операции.

Память доступна для чтения, записи, полного и постраничного стирания через регистры данных DATAn (n от 0 до 3), адреса ADDR, команд CMD, статуса STAT блока FLASHM. Запись необходимо производить в предварительно очищенную ячейку памяти. Стирание памяти осуществляется по банкам или постранично. Для полного стирания всей FlashM-памяти необходимо поочередно подать четыре команды полного стирания на каждый из четырех банков памяти.

Пример кода для полного стирания FlashM-памяти:

```
for (int i = 0; i < 4; i++) {
   FLASHM->EXPAND_ADDR_bit.BANK = i;
   FLASHM->CMD = (0xc0de << FLASHM_CMD_KEY_Pos) |
                 (1 << FLASHM_CMD_ALLSEC_Pos) |
                 (1 << FLASHM_CMD_ERSEC_Pos);

   delay(5);

   while (FLASHM->STAT_bit.BUSY) //wait done command
   {};
}
```

Минимальное время чтения данных из FlashM-памяти составляет до 25 нс (типовое значение задержки). Поэтому, исходя из выбранной рабочей частоты, следует задать определенное количество дополнительных тактов ожидания, необходимое для стабильного чтения из FlashM-памяти. В таблице 9.1 представлены значения дополнительных тактов ожидания FlashM и FlashD-памяти.

Стирание страницы/одного банка памяти выполняется за время до 20 мс. Типичное значение времени программирования 16 байт (четыре 32-разрядных регистра данных DATA[0], DATA[1], DATA[2] и DATA[3]) – 20 мкс.

Таблица 9.1 – Значения параметра (количество дополнительных тактов ожидания) в зависимости от частоты сигнала SYSCLK

f_{SYSCLK}, не более, МГц	Количество дополнительных тактов ожидания при чтении Flash-памяти
204	5
175	4
140	3
105	2
70	1
35	0
Примечание – Значение параметра после сброса равно 1	

Помимо основной области памяти FlashM (4 Мбайт), доступной как через глобальное адресное пространство, так и через регистры контроллера, существует дополнительная NVR область (64 страницы по 2 Кбайта, располагающихся в диапазоне адресов 0000_0000h – 0000_3FFFh, 000F_C000 – 003F_FFFFh). В последней ячейке 64 страницы NVR области (по адресу 003F_FFF0h) располагается конфигурационное слово микроконтроллера CFGWORD. NVR область доступна для чтения, записи и постраничного стирания только через регистры блока FLASHM.

Распределение адресов в NVR области Flash-памяти представлено на рисунке 9.1.

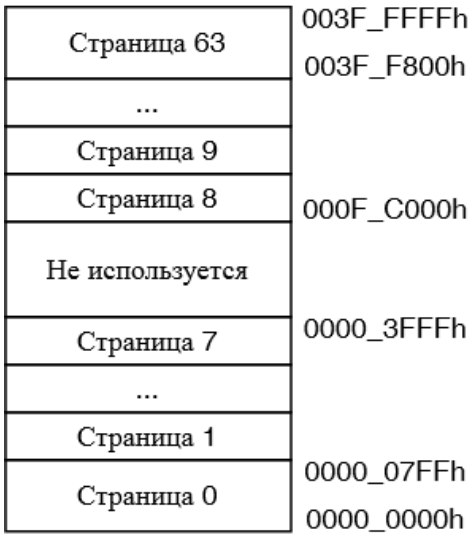


Рисунок 9.1 – Адресация NVR области Flash-памяти

Буфер предвыборки

Блок Flash-памяти позволяет активировать дополнительный буфер предвыборки на четыре 32-битных слова. Включение буфера предвыборки выполняется путем установки бита FBEN регистра CTRL. Перед включением буфера предвыборки необходимо сбросить его, установив бит FBFLUSH регистра CTRL.

При запросе данных на шине по адресу, по которому не осуществлялась предвыборка, выполняются следующие действия:

1) Сигнал готовности на шине устанавливается в ноль и задерживает транзакцию.

2) По запрашиваемому адресу считываются 4 32-битных слова (128 бит) данных из Flash-памяти. Далее эти данные записываются во внутренний первый буфер.

3) Требуемое слово передается на внутреннюю шину, и сигнал готовности устанавливается в единицу.

4) Сразу после установки сигнала готовности из Flash-памяти считываются 128 бит данных по следующему адресу. Данные сохраняются во втором буфере. Если во время считывания этих данных появляются запросы по адресам, сохраненным в первом буфере, ответ возникает мгновенно, если по другим адресам, то готовность на шине устанавливается в ноль и происходит ожидание завершения считывания во второй буфер и далее возврат к действию 2.

5) Если приходят запросы по адресам, сохраненным в первом буфере, ответ возникает мгновенно, если по адресам, находящимся во втором буфере, ответ также возникает мгновенно. Далее переписывается первый буфер значением второго и считывается следующий адрес из Flash-памяти. Если приходят запросы по адресам не из первого и второго буферов, тогда возврат к действию 1.

Конфигурационное слово CFGWORD

В последнем секторе NVR области FlashM-памяти расположено конфигурационное слово CFGWORD, содержащее параметры микроконтроллера. Описание битов представлено в таблице 9.2. Чтение конфигурационного слова контроллером происходит каждый раз после сброса по включению (POR).

Таблица 9.2 – Конфигурационное слово CFGWORD

CFGWORD +3FFFF0h (локальный адрес NVR области FLASHM)			
Поле	Биты	Описание	
SWMODEEN	4	Бит разрешения программного изменения режима работы ядер (LOCKSTEP, DUALCORE)	
		0	Изменение режима запрещено
		1	Изменение режима разрешено (по умолчанию)
LOCKSTEPEN	3	Бит выбора режима работы LOCKSTEP при запуске от POR	
		0	Режим LOCKSTEP отключен
		1	Режим LOCKSTEP включен (по умолчанию)
JTAGEN	2	Бит разрешения работы интерфейса JTAG	
		0	Отладка отключена
		1	Отладка включена (по умолчанию)
FLASHWE	1-0	Бит включения защиты всей Flash-памяти (основной и NVR областей FlashM и основной области FlashD)	
		00	Запрет записи и стирания Flash-памяти
		01	
		10	
		11	Защита выключена (по умолчанию)
–	31-5	Зарезервировано	

При установленном бите FLASHWE запрещается запись во Flash-память через регистровый интерфейс. Операция записи интерпретируется как операция чтения, при этом стандартным образом продолжают работать механизмы флагов из регистра STAT.

9.2 FlashD-память

FlashD-память может использоваться для хранения программ и данных пользователя. Размер основной FlashD-памяти составляет 512 Кбайт (512 страниц по 1 Кбайту), и в адресном пространстве она занимает диапазон с 0040_0000h по 0047_FFFFh.

Управляющие регистры контроллера FlashD-памяти приведены в приложении А.4.

Память доступна для чтения, записи, полного и постраничного стирания через регистры данных DATAn (n от 0 до 1), адреса ADDR, команд CMD, статуса STAT блока FLASHD. Запись необходимо производить в предварительно очищенную ячейку памяти. Стирание памяти осуществляется полностью или постранично.

Минимальное время чтения данных из FlashD-памяти составляет до 25 нс (типовое значение задержки). Поэтому, исходя из выбранной рабочей частоты, следует задать определенное количество дополнительных тактов ожидания, необходимое для стабильного чтения из FlashD-памяти. В таблице 7.1 представлены значения дополнительных тактов ожидания Flash-памяти.

Стирание страницы / всей памяти выполняется за время до 20 мс. Время программирования 8 байт (два 32-разрядных регистра данных DATA[0] и DATA[1]) – 20 мкс.

9.3 OTP-память

OTP-память может использоваться только для хранения данных пользователя. Размер OTP области FlashD-памяти составляет 16 Кбайт (16 страниц по 1 Кбайт).

Память доступна для чтения и однократной записи через регистры данных DATAn (n от 0 до 1), адреса ADDR, команд CMD, статуса STAT блока FLASHD. При обращении к OTP-памяти необходимо устанавливать бит NVRON регистра команд CMD блока FLASHD. Запись производится однократно в ячейки памяти с исходным значением 0xFFFFFFFF.

Распределение адресов в OTP области FlashD-памяти представлено на рисунке 9.2.

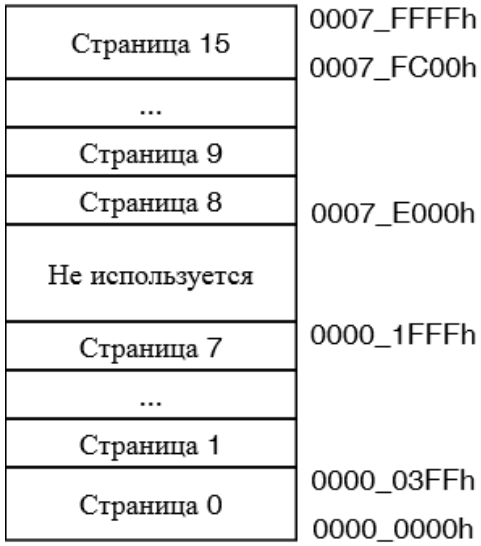


Рисунок 9.2 – Адресация OTP области FlashD-памяти

9.4 Сервисный сброс всей Flash-памяти

Во время сброса микроконтроллера анализируется состояние вывода SERVEN. Если вывод находится в состоянии логической единицы (подтянут к 3,3 В), то FlashM-память, FlashD-память и OTP-память переводится в режим, в котором чтение запрещено (при чтении возвращаются нули). При этом игнорируется состояние бита JTAGEN.

Далее по отладочному интерфейсу JTAG должна быть подана команда записи бита SERVCMD в регистр SERVCTL блока SIU, после чего будет активировано полное стирание всех областей основной и загрузочной памяти (кроме OTP).

Примечание – Если полное стирание не требуется, во время сброса на выводе SERVEN должен удерживаться логический ноль.

9.5 Модуль коррекции ошибок ECC FLASH

Контроллер Flash-памяти поддерживает функционал коррекции ошибок Хемминга, позволяющий исправить одну ошибку и определить две ошибки. Каждые 64 бита данных защищены 8 битами кода коррекции.

Для включения функционала коррекции ошибок необходимо установить бит ECCEN регистра FECCCR.

Определить количество обнаруженных ошибок можно с помощью регистров счетчиков FECCSEC и FECCDEC для одиночных и двойных ошибок соответственно.

Адрес слова, в котором обнаружена ошибка, можно определить с помощью регистров FECCSEA и FECCDEA для одиночных и двойных ошибок соответственно.

Также можно настроить прерывания при обнаружении одиночных и двойных ошибок с помощью бит ISEUEN и IDEUEN регистра FECCCR. Текущий статус прерываний можно узнать с помощью регистра FECCISTAT.

Для проверки работоспособности системы ECC контроллеры FlashM и FlashD поддерживают модули внедрения ошибок EIM (см. подраздел 8.2).

10 Контроллер внешней памяти EMC

Контроллер внешней памяти позволяет осуществлять коммутацию процессора микроконтроллера с внешней памятью. Контроллер может работать в 8-/16-разрядном режиме и согласовывает сигналы AHB-шины с сигналами внешней памяти.

Управляющие регистры контроллера внешней памяти EMC приведены в приложении А.6.

На рисунке 10.1 показаны временные диаграммы циклов чтения и записи при работе с внешней памятью, описание обозначений приведено в таблице 10.1.

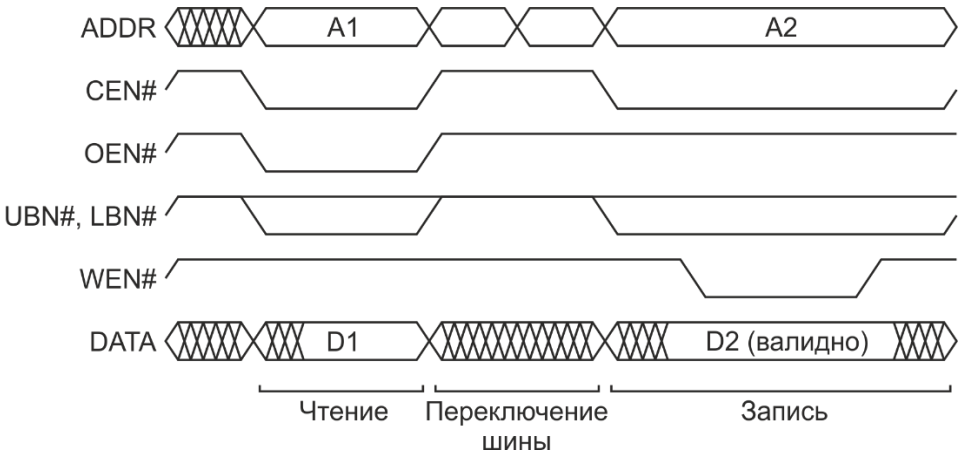


Рисунок 10.1 – Временные диаграммы работы контроллера внешней памяти

Таблица 10.1 – Сигналы, используемые при работе с внешней памятью

Обозначение сигнала	Обозначение альтернативной функции	Вывод на портах GPIO	Назначение
ADDR0 - ADDR23	MEM_ADDR0-MEM_ADDR7, MEM_ADDR8-MEM_ADDR23	D8-D15, E0-E15	Шина адреса для чтения/записи. Сигнал, передаваемый во внешнюю память
CEN0# - CEN7#	EMC_CEN0-EMC_CEN7	F4-F11	Сигнал выбора внешнего устройства
OEN#	EMC_OEN	F15	Сигнал (строб) разрешения чтения для внешнего устройства
UBN#, LBN#	EMC_UBN, EMC_LBN	F14, F13	Сигналы разрешения для старшего и младшего байт
WEN#	EMC_WEN	F12	Сигнал (строб) записи для внешнего устройства
DATA0-DATA15	MEM_DATA0-MEM_DATA15	G0-G15	Данные внешней памяти
Примечание – Сигналы с символом «#» имеют низкий активный уровень.			

Для работы EMC контроллера все выводы задействованных портов GPIO должны быть переведены в режим второй альтернативной функции (AF2). Подробнее в приложении В.

Максимально доступный диапазон адресного пространства в 128 Мбайт (ширина адресной шины 24 бита) разбит на 8 окон по 16 Мбайт, см. таблицу 10.2.

Таблица 10.2 – Адресное пространство внешней памяти

Номер окна	Сигнал выбора	Адрес начала	Адрес конца
7	EMC_CEN[7]	0x6700_0000	0x67FF_FFFF
6	EMC_CEN[6]	0x6600_0000	0x66FF_FFFF
5	EMC_CEN[5]	0x6500_0000	0x65FF_FFFF
4	EMC_CEN[4]	0x6400_0000	0x64FF_FFFF
3	EMC_CEN[3]	0x6300_0000	0x63FF_FFFF
2	EMC_CEN[2]	0x6200_0000	0x62FF_FFFF
1	EMC_CEN[1]	0x6100_0000	0x61FF_FFFF
0	EMC_CEN[0]	0x6000_0000	0x60FF_FFFF

Каждое окно может использовать индивидуальные настройки разрядности и циклов доступа. Также каждому окну соответствует свой сигнал выбора EMC_CEN, который вырабатывается автоматически при обращении к адресу, попадающему в соответствующее окно.

Управление контроллером внешней памяти осуществляется посредством группы из восьми регистров WINCFG для каждого окна в отдельности. Разрядность шины задается битом MODE. В контроллере внешней памяти предусмотрен цикл переключения шины между циклами чтения и записи. Это реализовано в целях предотвращения кратковременных скачков тока в то время, когда процессор и внешнее устройство пытаются одновременно управлять шиной данных. Настройка длительности циклов чтения, записи и переключения осуществляется посредством полей RDCYC, WRCYC и TACYC. На рисунке 10.1 показан пример для случая, когда RDCYC = WRCYC = TACYC = 001h, т. е. длительность каждого цикла составляет два такта сигнала системной частоты.

11 Контроллер внешней памяти SDRAM

Контроллер внешней памяти SDRAM обеспечивает взаимосвязь между системным интерфейсом и командным интерфейсом памяти SDRAM.

Управляющие регистры контроллера внешней памяти SDRAM приведены в приложении А.7.

На рисунке 11.1 показана блок-схема контроллера внешней памяти SDRAM.

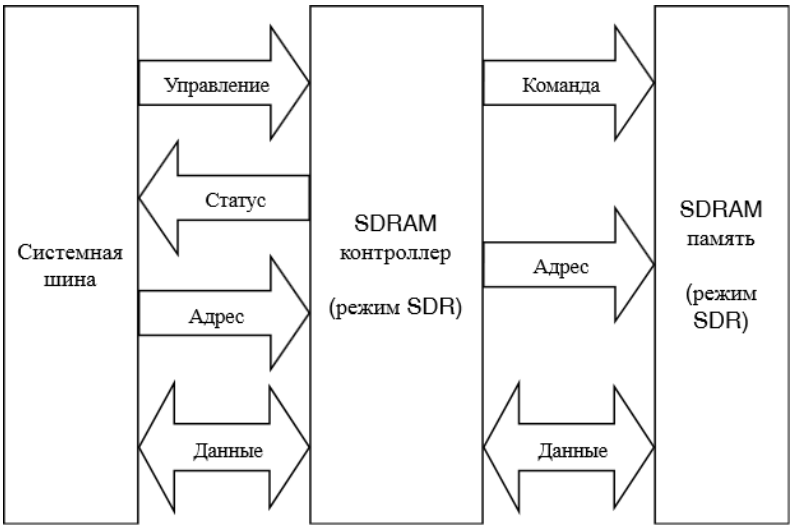


Рисунок 11.1 – Блок-схема контроллера SDRAM

11.1 Особенности контроллера SDRAM

– Настраиваемая ширина данных памяти SDRAM (8/16/32 бита);
– Поддержка стандартных устройств и модулей SDRAM;
– Поддержка всех стандартных функций SDRAM;
– Полностью синхронный интерфейс. Все сигналы регистрируются положительным фронтом системного тактового сигнала;
– Один сигнал выбора микросхемы (CS#);
– Поддержка памяти SDRAM с четырьмя банками памяти;
– Программируемая задержка CAS#;
– Архитектура управления банками памяти, которая сводит к минимуму задержки;
– Автоматическое контролируемое обновление.

11.2 Описание внешних выводов

Описание сигналов внешних выводов SDRAM контроллера приведено в таблице 11.1.

Таблица 11.1 – Внешние сигналы SDRAM контроллера

Сигнал	Разрядность	Направление	Вывод порта GPIO	Описание
CKE	1	Выход	E7	Разрешение тактирования
CS#	1	Выход	E11	Вывод управляющего сигнала CS#
RAS#	1	Выход	E9	Вывод управляющего сигнала RAS#
CAS#	1	Выход	E10	Вывод управляющего сигнала CAS#
WE#	1	Выход	E8	Вывод управляющего сигнала WE#
DQ0-DQ3	1/2/4	Выход	E12-E15	Маска шины данных
BA0-BA1	2	Выход	E5-E6	Адрес банка памяти
ADDR0-ADDR7, ADDR8-ADDR12	13	Выход	D8-D15, E0-E4	Шина адреса
DATA0-DATA15, DATA16-DATA31	8/16/32	Вход/Выход	G0-G15, F0-F15	Входные/выходные данные
CLK	1	Выход	D7	Вывод тактового сигнала

Для работы SDRAM контроллера все выводы задействованных портов GPIO должны быть переведены в режим второй альтернативной функции (AF2). Подробнее в приложении В.

11.3 Описание работы SDRAM

SDRAM – это высокоскоростная динамическая оперативная память (DRAM) с синхронным интерфейсом. Синхронный интерфейс и полностью конвейерная внутренняя архитектура SDRAM обеспечивают чрезвычайно высокую скорость передачи данных при эффективном использовании. SDRAM организована в виде банков памяти, адресуемых по строкам и столбцам. Количество бит адреса строки и столбца зависит от размера и конфигурации памяти. SDRAM управляется командами шины, которые формируются с использованием комбинаций сигналов RAS#, CAS# и WE#. Например, в тактовом цикле, когда все три сигнала имеют высокий уровень, соответствующей командой является отсутствие операции (NOP). Об отсутствии операции также говорит не установленный сигнал CS# (высокий уровень).

Стандартные команды шины SDRAM показаны в таблице 11.2.

Таблица 11.2 – Команды SDRAM

Команда	Состояние RAS#	Состояние CAS#	Состояние WE#
Отсутствие операции (NOP)	Высокий	Высокий	Высокий
Активация	Низкий	Высокий	Высокий
Чтение	Высокий	Низкий	Высокий
Запись	Высокий	Низкий	Низкий
Остановка операции пакетного чтения/записи	Высокий	Высокий	Низкий
Деактивация/закрытие строки/подзарядка	Низкий	Высокий	Низкий
Автообновление/самообновление	Низкий	Низкий	Высокий
Загрузка регистра MODE	Низкий	Низкий	Низкий

Устройства SDRAM обычно делятся на четыре банка. Эти банки должны быть открыты до того, как на них можно будет записать или считать данные с диапазона адресов. Строка и банк, которые будут открыты, записываются одновременно с активной командой. При обращении к новой строке в банке данных для чтения или записи может потребоваться сначала закрыть банк, а затем повторно открыть банк для новой строки. Закрытие банка осуществляется с помощью команды перезагрузки. Открытие и закрытие банков требует увеличения пропускной способности памяти, поэтому SDRAM контроллер был разработан с учетом одновременного мониторинга состояния четырех банков и управления ими. Это позволяет контроллеру рационально открывать и закрывать банки только в случае необходимости.

Когда выдается команда чтения или записи, устройствам SDRAM передается начальный адрес столбца. Исходные данные выводятся одновременно с командой записи. При команде чтения исходные данные появляются на шине данных через 1-4 такта. Это называется задержкой CAS и связано со временем, необходимым для физического считывания внутренней памяти DRAM и регистрации данных на шине. Задержка CAS зависит от скорости SDRAM и тактовой частоты памяти. В целом, чем быстрее тактовый сигнал, тем больше требуется циклов задержки CAS. После начальной команды чтения или записи последовательные операции чтения и записи будут продолжаться до тех пор, пока не будет достигнута требуемая длина пакета или пока не будет выдана команда завершения пакета. Устройства SDRAM поддерживают длительность пакета до 8 циклов передачи данных. Контроллер SDRAM способен каскадировать пакеты данных, чтобы максимально увеличить пропускную способность SDRAM.

Устройства SDRAM требуют периодического обновления для поддержания целостности сохраненных данных. Контроллер SDRAM периодически автоматически выдает команду автообновления. Никакого вмешательства пользователя не требуется.

Команда загрузки регистра MODE используется для настройки работы SDRAM. В этом регистре хранятся задержка CAS, длина пакета, тип пакета и режим пакетной записи. Контроллер SDR выполняет запись только в регистр базового режима.

Чтобы уменьшить количество выводов, адреса строк и столбцов SDRAM мультиплексируются на одних и тех же выводах. В таблице 11.3 указано количество строк, столбцов, банков данных и чипов, необходимых для различных стандартных дискретных устройств SDR SDRAM. Контроллер SDRAM будет поддерживать любое из этих устройств.

Таблица 11.3 – Возможная конфигурация поддерживаемых чипов памяти

Общий объем памяти чипа	Конфигурация	Строки	Столбцы	Банки
256 Мбит	8M*8*4	8K A[12:0]	1K A[9:0]	4
256 Мбит	4M*16*4	8K A[12:0]	512 A[8:0]	4

11.4 Функциональное описание

Функциональная схема SDRAM контроллера приведена на рисунке 11.2.

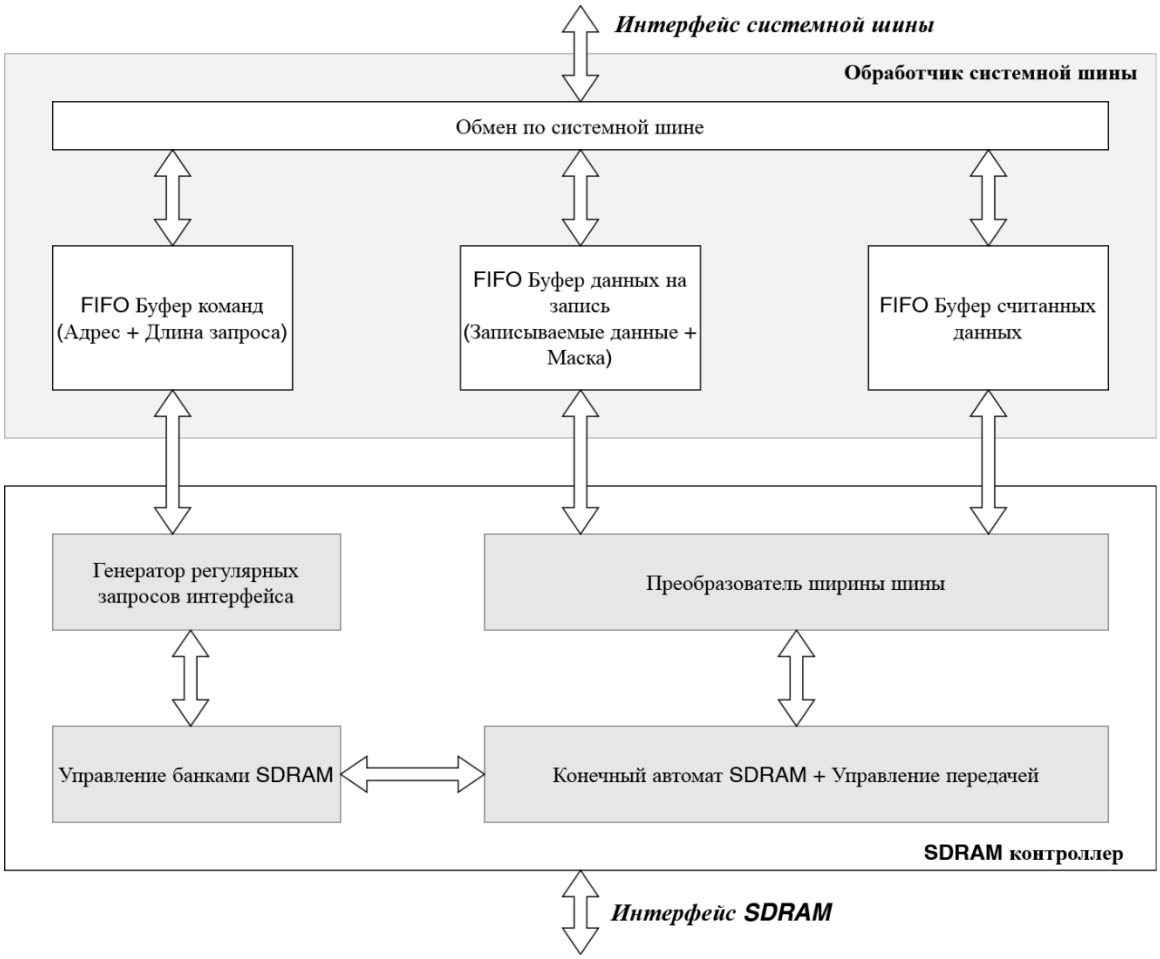


Рисунок 11.2 – Функциональная схема SDRAM контроллера

Обработчик системной шины

Этот блок обрабатывает обмен данными между системной шиной и контроллером SDRAM. Он также отвечает за необходимую смену тактового домена.

Данный блок включает в себя:

– Асинхронный буфер FIFO команд;

– Асинхронный буфер FIFO записываемых данных;

– Асинхронный буфер FIFO считанных данных.

Контроллер SDRAM

Этот блок включает в себя четыре подблока:

1 Преобразователь шины SDRAM. Этот блок преобразует и повторно выравнивает 32-битный системный формат в эквивалентный формат SDR 8/16/32. Во время транзакции записи он разбивает 32-разрядные данные на формат ширины шины SDRAM 8/16/32. Во время транзакции чтения он упаковывает 8/16/32-битные данные SDRAM в 32-битные данные.

2 Генератор запросов SDRAM. Данный блок изменяет внутренний адрес и длину пакета в зависимости от ширины шины SDRAM. Если текущая длина пакета пересекает границу страницы, то запрос будет разделен на две части с соответствующим изменением адреса запроса и длины запроса. Если текущая длина пакета не пересекает границы

страницы, то запрос будет прозрачно передан в блок управления банком. На основе бита конфигурации столбца этот блок генерирует адрес столбца, адрес строки и адрес банка.

3 Контроллер банков SDRAM. Этот модуль принимает запросы от генератора запросов SDRAM, проверяет попадание/пропуск страницы и выдает команды предварительной зарядки/активации, а затем передает запрос контроллеру передачи SDRAM.

4 Контроллер передачи SDRAM. Этот модуль принимает запросы от контроллера банков SDRAM, выполняет передачу и управляет потоком данных. В конце передачи он выдает команду на завершение пакета, если пакет не завершен и недоступна другая команда для этого банка.

11.5 Интерфейс SDRAM

Перед началом работы SDRAM необходимо инициализировать. В следующих разделах представлена подробная информация об инициализации устройства, определении регистров, описаниях команд и работе устройства.

Инициализация

SDRAM должна быть включена и инициализирована заранее определенным образом. Операции, отличные от указанных, могут привести к неопределенной работе. После подачи питания на VCC и стабилизации тактовой частоты, SDRAM требует задержки в 100 мкс перед выдачей любой команды, отличной от COMMAND INHIBIT или NOP. В течение и до конца этого периода следует применять команды COMMAND INHIBIT или NOP.

Как только задержка в 100 мкс будет выдержана и будет подана хотя бы одна команда COMMAND INHIBIT или NOP, следует подать команду на подзарядку (PRECHARGE). Затем все банки устройства должны быть предварительно заряжены, что переводит устройство и все банки в состояние ожидания. После перехода в состояние ожидания, необходимо выполнить два цикла автообновления (AUTO REFRESH). После завершения двух циклов автообновления, SDRAM будет готова к программированию регистра MODE. Поскольку регистр MODE после включения будет в неизвестном состоянии, их следует загрузить перед применением какой-либо рабочей команды.

Регистр управления MODE

Регистр MODE используется для определения конкретного режима работы SDRAM. Это определение включает в себя выбор длины пакета, типа пакета, задержки CAS, режима работы и режима пакетной записи, как показано на схеме определения регистра MODE на рисунке 11.3. Регистр режимов программируется с помощью команды LOAD MODE REGISTER и хранит записанную информацию до тех пор, пока он не будет перезаписан или не будет снято питание устройства.

Биты регистра MODE M0-M2 определяют длину пакета, M3 определяет тип пакета (последовательный или с чередованием), M4-M6 определяют задержку CAS, M7-M8 задают режим работы, M9 определяет режим пакетной записи, M10 и M11 зарезервированы для использования в будущем и должны быть установлены в значение 00b для обеспечения совместимости с будущими устройствами. Значение бита M12 не определено, но при загрузке регистра MODE оно должно быть нулевым. Регистр MODE должен быть загружен, когда все банки устройства находятся в режиме ожидания, и контроллер должен подождать указанное время, прежде чем начать последующую операцию. Нарушение любого из этих требований может привести к неопределенному режиму работы.

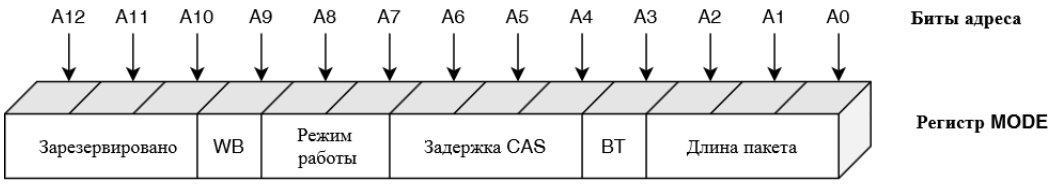


Рисунок 11.3 – Схема регистра MODE

Режим работы (Operating Mode)

Нормальный режим работы выбирается путем установки бит M7 и M8 в нулевое значение. Остальные комбинации значений для M7 и M8 зарезервированы для дальнейшего использования и/или тестовых режимов. Программируемая длина пакета применяется как к пакетам чтения, так и к пакетам записи.

Режим пакетной записи (Writing Burst Mode)

Если бит M9 сброшен, длина пакета, запрограммированная с помощью M0-M2, применяется как к пакетам чтения, так и к пакетам записи. Когда бит M9 установлен, запрограммированная длина пакета применяется только к пакетам чтения, а доступ на запись осуществляется в отдельные ячейки (без пакета).

Длина пакета (Burst Length)

Длина пакета определяет максимальное количество ячеек, к которым можно получить доступ для данной команды чтения или записи. Пакеты длиной в 1, 2, 4 или 8 слов доступны как для последовательного, так и для чередующегося типов пакетов, а полностраничный пакет доступен только для последовательного типа. Полностраничный пакет данных используется в сочетании с командой завершения пакета данных (BURST TERMINATE), для формирования пакетов произвольной длины. При подаче команды чтения или записи фактически выбирается блок столбцов, равный длине пакета. Все обращения одного пакета данных выполняются внутри одного блока, что означает, что при достижении границы блока, пакет данных будет перенесен внутри данного блока, как показано в таблице определения пакета данных (см. таблицу 11.5). Блок однозначно выбирается битами A1-A9, когда длина пакета установлена равной двум; по A2-A9, когда длина пакета установлена равной четырем; и по A3-A9, когда длина пакета установлена равной восьми. Оставшиеся биты адреса используются для выбора начального местоположения внутри блока. Значение длина пакета в зависимости от значений бит M2-M0 приведено в таблице 11.4.

Таблица 11.4 – Значение длины пакета

M2	M1	M0	BT = 1	BT = 0
0	0	0	1	1
0	0	1	2	2
0	1	0	4	4
0	1	1	8	8
1	0	0	–	–
1	0	1	–	–
1	1	0	–	–
1	1	1	Вся страница	–

Тип пакета (Burst Type)

Последовательный или чередующийся пакет данных выбирается с помощью бита М3 (0 – последовательный, 1– с чередованием).

Порядок доступа в пакете данных определяется длиной пакета данных, его типом и адресом начального столбца, как показано в таблице определения пакета данных (см. таблицу 11.5).

Таблица 11.5 – Определение порядка доступа при пакетной передаче данных

Длина пакета	Стартовый адрес столбца			Порядок доступа в пакете данных	
				Тип пакета – последовательный	Тип пакета – с чередованием
2	A0				
	0			0-1	0-1
	1			1-0	1-0
4	A1		A0		
	0		0	0-1-2-3	0-1-2-3
	0		1	1-2-3-0	1-0-3-2
	1		0	2-3-0-1	2-3-0-1
	1		1	3-0-1-2	3-2-1-0
8	A2	A1	A0		
	0	0	0	0-1-2-3-4-5-6-7	0-1-2-3-4-5-6-7
	0	0	1	1-2-3-4-5-6-7-0	1-0-3-2-5-4-7-6
	0	1	0	2-3-4-5-6-7-0-1	2-3-0-1-6-7-4-5
	0	1	1	3-4-5-6-7-0-1-2	3-2-1-0-7-6-5-4
	1	0	0	4-5-6-7-0-1-2-3	4-5-6-7-0-1-2-3
	1	0	1	5-6-7-0-1-2-3-4	5-4-7-6-1-0-3-2
	1	1	0	6-7-0-1-2-3-4-5	6-7-4-5-2-3-0-1
	1	1	1	7-0-1-2-3-4-5-6	7-6-5-4-3-2-1-0
Вся страница (y)	n=A0-A11/9/8 (от 0 до y)			Cn, Cn+1, Cn+2, Cn+3, Cn+4, Cn-1, Cn	Не поддерживается

Примечания:

1 Для полностраничного доступа y = 1,024(x4), y=512(x8), y=256(x16);

2 Для пакета длиной в два слова, A1-A9 выбирают блок из двух слов, A0 выбирает начальный столбец внутри блока;

3 Для пакета длиной в четыре слова, A2-A9 выбирают блок из четырех слов, A0-A1 выбирают начальный столбец внутри блока;

4 Для пакета длиной в восемь слов, A3-A9 выбирают блок из восьми слов; A0-A2 выбирают начальный столбец внутри блока;

5 Для полностраничного пакета выбирается полный ряд, а A0-A9 выбирает начальный столбец;

6 Когда достигается граница блока, следующий доступ производится к началу блока;

7 Для одного слова A0-A9 выбирают единственный столбец и бит М3 игнорируется.

Задержка CAS (CAS Latency)

Задержка CAS – это задержка в тактовых циклах между регистрацией команды чтения и доступностью первого фрагмента выходных данных. Задержка может быть установлена в два или три такта. Значение задержки CAS в зависимости от значений бит M6-M4 приведено в таблице 11.6.

Таблица 11.6 – Значение задержки CAS

M6	M5	M4	Задержка CAS
0	0	0	–
0	0	1	–
0	1	0	2
0	1	1	3
1	0	0	–
1	0	1	–
1	1	0	–
1	1	1	–

12 Микропроцессорное ядро SCR5

Основой микроконтроллера К1921ВГ1Т является процессорное ядро SCR5 с архитектурой RISC-V.

Особенности

Краткое описание ключевых характеристик процессорного ядра SCR5:

1) Эффективное двухпроцессорное ядро RISC-V для прикладных задач среднего класса с поддержкой Linux;
2) Гарвардская архитектура, отдельная память для инструкций и данных;
3) Набор команд RV32IMAFDCP:
   – Integer (32-bit) инструкции (Zicsr и Zifencei);
   – Muliply/Divide инструкции;
   – Atomic инструкции;
   – Floating-point инструкции (одинарной точности);
   – D Floating-point инструкции (двойной точности);
   – Compact (16-bit) инструкции;
   – P расширение для DSP/SIMD инструкции;
4) От 7 до 9 стадий наполнения конвейера;
5) Режимы привилегированности machine, user и supervisor;
6) Поддержка межъядерной связи SCRX;
   – Генерация машинного программного прерывания отдельно для каждого ядра;
   – Позволяет осуществлять короткий обмен сообщениями между ядрами;
7) Высокопроизводительный модуль умножения/деления;
8) Четыре 64-разрядных высокопроизводительных монитора;
9) Высокопроизводительный модуль с плавающей запятой, соответствующий стандарту IEEE 754-2008:
   – Модуль FP двойной точности;
   – Регистровый файл из 32 регистров данных с плавающей запятой;
10) Блок управления памятью (MMU)
   – Виртуальная память на основе страниц;
   – 32 записи в ITLB;
   – 32 записи в DTLB;
11) Блок защиты памяти (MPU) на 16 записей;
12) Согласованный кэш L1 и L2;
   – Кэш команд L1 32 КБ;
   – Кэш данных L1 32 КБ;
   – Кэш L2 128 КБ;
13) Встроенный 64-разрядный RTC-таймер:
   – Генерирует машинное прерывание по таймеру отдельно для каждого ядра;
   – Генерирует программное машинное прерывание;
14) Программируемый контроллер прерываний:
   – Настраиваемый контроллер прерываний на уровне платформы (PLIC);
   – 254 IRQ – запроса;
   – 16 уровней приоритета;
   – 4 целевых прерывания;
15) Внешний интерфейс, совместимый с AXI4:
   – Два главных порта: кэшируемый, некэшируемый;
   – Размер адреса 32 бита;
   – Размер данных 32 бита;
16) Интегрированная подсистема отладки с обширным функционалом:
   – Совместимый с JTAG интерфейс;

– Поддержка точек останова HW/SW;
– Поддержка точек останова ROM.

Более подробная информация в SCR5 External Architecture Specification.

12.1 Структура ядра

Ядро реализовано как архитектура с выделенным доступом к памяти (Load/Store architecture). Другими словами, доступ к памяти осуществляется только для операций чтения (load) и записи (store), а все арифметические команды взаимодействуют только с целочисленными регистрами или регистрами с плавающей точкой. Ядро поддерживает 32-битовое адресное пространство с возможностью побайтового доступа и порядком байтов от младшего к старшему (little endian). Среда исполнения определяет, к каким частям адресного пространства разрешён доступ.

На рисунке 12.1 показана блок-схема 2-ядерного процессорного кластера, включающая кэш второго уровня (L2).

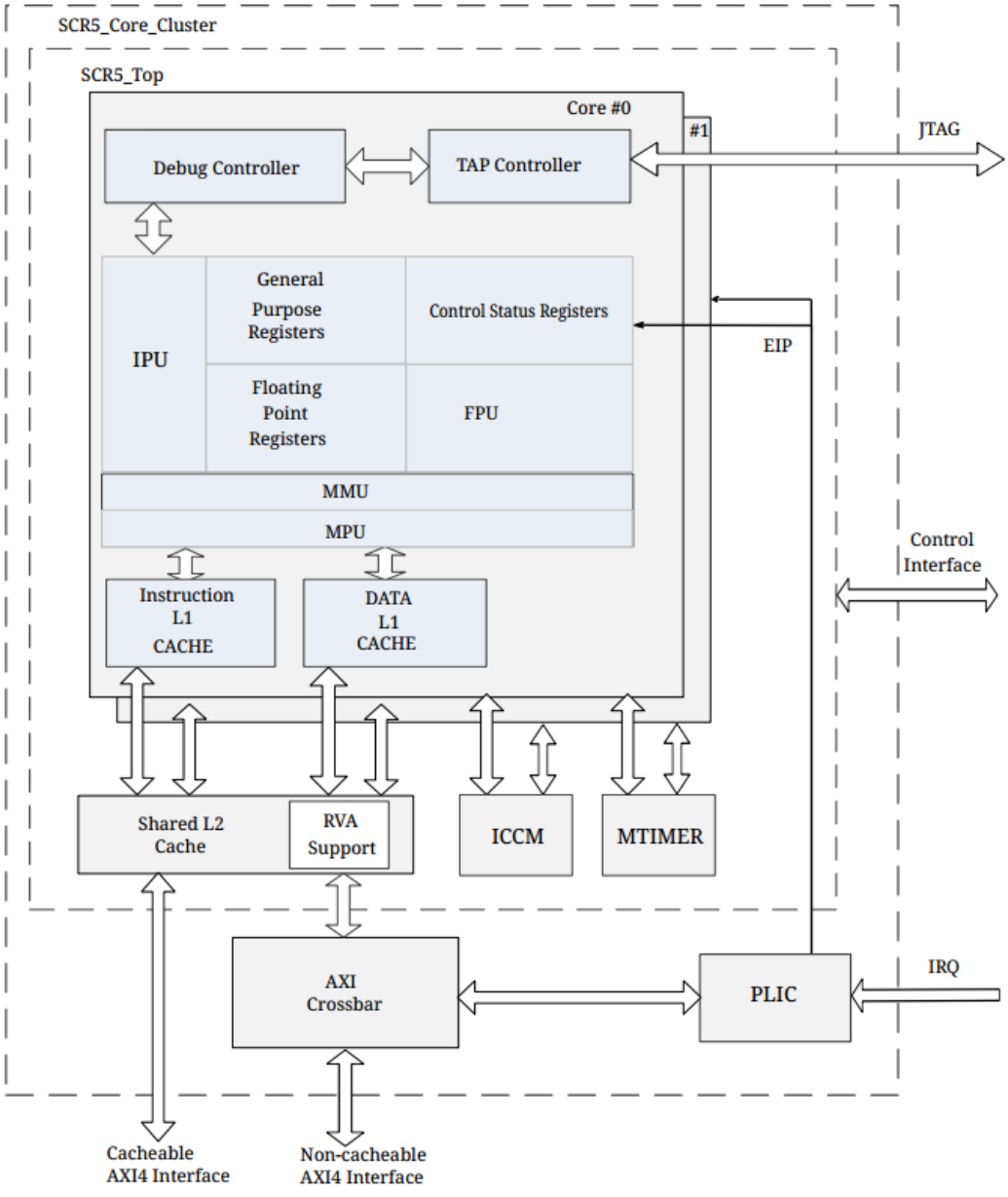


Рисунок 12.1 – Блок-схема 2-ядерного процессорного кластера

Основные элементы

• Двойное процессорное ядро SCR5, каждый процессор имеет свои собственные:
  ◦ Блок обработки целых чисел (IPU), который реализует основные функции ядра:
  ▪ Выборка инструкций;
    ▪ Расшифровка инструкций;
    ▪ Целочисленная арифметика (включая целочисленное умножение/деление);
    ▪ Управление загрузкой/хранением;
    ▪ Управление потоком инструкций;
    ▪ Обработка исключений;
    ▪ Запись в регистрационный файл (упорядочивание результатов);
  ◦ Регистры общего назначения (GPR);
  ◦ Регистры управления/статуса (CSR);
  ◦ Регистры данных с плавающей запятой (FPDR);
  ◦ Модуль операций с плавающей запятой (FPU);
  ◦ Модуль DSP для расширения RISC-V «P»;
  ◦ Блок управления памятью (MMU). Включающий в себя:
  ▪ Блок защиты памяти (MPU);
  ▪ Отдельный TLB для выборки инструкций и доступа к данным;
  ◦ Кэш инструкций L1;
  ◦ Кэш данных L1;
  ◦ Подсистема отладки, которая подразделяется на:
    ▪ Контроллер точек тестового доступа (TAPC);
    ▪ Контроллер отладки (модуль отладки RISC-V, DM);
• Кэш-память L2;
  ◦ Включена поддержка RVA;
• Настраиваемый контроллер прерываний на уровне платформы (PLIC);
• Счетчик реального времени (MTIMER);
• Межъядерный коммуникационный модуль (ICCM);
• Основные кэшируемые и некэшируемые интерфейсы внешней памяти на основе AXI4.

12.2 Конвейер инструкций

SCR5 имеет многоступенчатый конвейер обработки.

Функциональные модули базового конвейера показаны на упрощенной структурной схеме на рисунке 12.2.

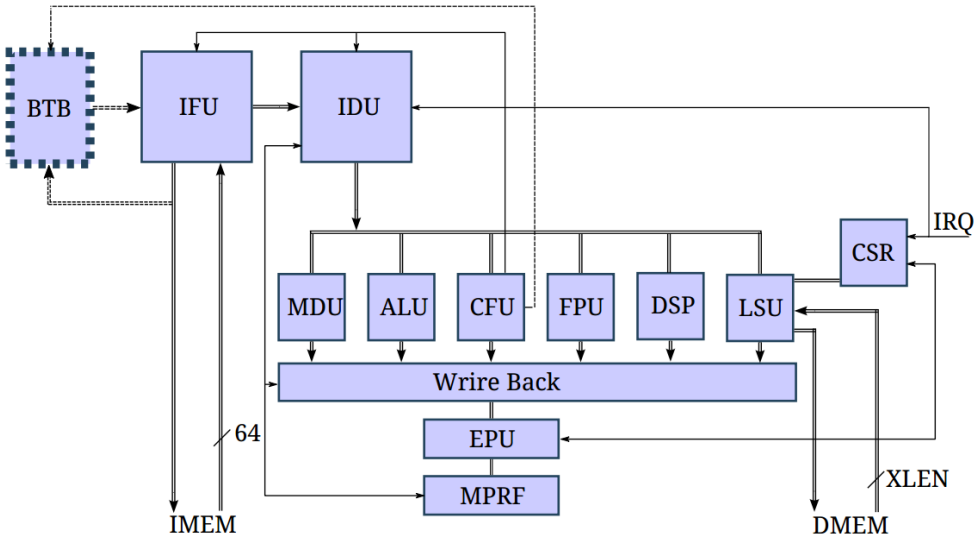


Рисунок 12.2 – Блок-схема конвейера

Конвейер SCR5 содержит:

1) IFU – Модуль выборки инструкций.

Модуль выборки инструкций извлекает инструкции из кэша инструкций L1 и передает инструкции в модуль декодирования инструкций.

2) BTB – Буфер целевой ветки (Branch Target Buffer).

Буфер целевой ветки (Branch target buffer, BTB) – это механизм для уменьшения штрафа за неправильное предсказание перехода. Он хранит целевой адрес для условных или безусловных переходов.

3) IDU – Модуль декодирования команд.

Модуль декодирования команд выполняет декодирование кода операции, проверяет наличие недопустимых кодов операций и комбинаций кодов операций, обеспечивает доступность операндов для исполнительных блоков.

4) Исполнительные модули. Декодированная информация в виде ряда сигналов управления и данных передается в соответствующие исполнительные устройства для выполнения действий, требуемых инструкцией:

– ALU – арифметико-логический модуль. Он реализует арифметические и логические операции с целыми значениями.

– MDU – модуль умножения/деления. MDU используется для выполнения операций умножения и деления с целыми значениями.

– FPU – модуль операций с плавающей запятой. FPU обеспечивает полную аппаратную поддержку стандарта IEEE 754-2008 с плавающей запятой для 32-разрядной арифметики с однозначной точностью и 64-разрядной арифметики с двойной точностью.

– DSP – модуль DSP-инструкций. Модуль DSP реализует расширение RISC-V “P” для DSP/SIMD ISA. С добавлением расширения набора команд RISCV P ядро процессора может запускать различные приложения DSP с меньшим энергопотреблением и более высокой производительностью.

– CFU – Блок управления потоком. Он содержит логику разрешения кодов переходов и условий команд, вычисляет адреса переходов и ветвлений и проверяет предсказанные ветвления.

– LSU – Модуль загрузки и хранения. Модуль загрузки и хранения отвечает за доступ к памяти данных и регистрам CSR.

5) WB – Обратная запись. На этапе обратной записи результаты выполнения модулей и операций чтения из памяти записываются в многопортовый регистровый файл (MPRF).

6) EPU – Модуль обработки событий. Модуль обработки событий контролирует наличие исключений или прерываний для перехода на новый адрес.

12.3 Модуль операций с плавающей запятой

Ядро включает в себя высокопроизводительный модуль операций с плавающей запятой (Floating Point Unit, FPU), совместимый со стандартом операций с плавающей запятой IEEE 754–2008. Модуль FPU может быть сконфигурирован для реализации арифметики с плавающей точкой одинарной или двойной точности. Размер регистров для операций с плавающей запятой (FPR) может быть выбран 32 или 64 бита в зависимости от выбранной точности (одинарной или двойной, соответственно) для данной операции.

12.4 Модуль DSP инструкций

Модуль DSP реализует ограниченный набор инструкций из расширения RISC-V "P".

С добавлением расширения набора команд RISC-V P процессорное ядро может запускать различные приложения DSP с меньшим энергопотреблением и более высокой производительностью.

Текущая реализация RV32 совместима с Zpn и поддерживает подмножества инструкций, сгруппированных в p-расширениях, которые включают:

– Сложение и вычитание;

– Битовый сдвиг;

– Перемещение данных;

– Умножение-накопление;

– Арифметика с фиксированной запятой.

Поддерживаемая конфигурация указана в регистре конфигурации MISA (Machine ISA).

Для получения информации о кодировках расширения набора команд "P" см. подраздел "Система команд".

Для получения подробных инструкций см. руководство по набору команд SCR5 RISC-V.

12.5 Уровни привилегий

Уровни привилегий обеспечивают защиту между различными компонентами программного стека. Попытки выполнения операций, не разрешенных текущему уровню привилегий, приведут к возникновению исключения типа «запрещенная инструкция». В обычных условиях это приводит к переходу в обработчик исключений в базовой среде исполнения или HAL (англ. Hardware Abstraction Level, уровень аппаратной абстракции).

Ядро SCR5 поддерживает три уровня привилегированности: машинный (M-mode), пользовательский (U-mode) и режим супервизора (S-mode).

В любой момент времени ядро работает на одном из реализованных уровней привилегий, закодированных как 2-битовое значение в одном или нескольких регистрах управления и состояния (CSR регистрах).

Машинный уровень имеет самые высокие привилегии. Код, работающий в режиме M-mode, по своей сути, является доверенным, так как он имеет низкоуровневый доступ ко всем реализованным функциям ядра.

Режим U-mode (пользовательский уровень) предназначен для исполнения пользовательских приложений.

Режим S-mode (уровень супервизора) предназначен для работы операционной системы.

Ядро обычно исполняет код приложения в режиме U-mode до тех пор, пока какое-либо исключение или внешнее прерывание, не заставит переключиться на вызов соответствующего обработчика исключения/прерывания, который обычно работает в более привилегированном режиме. После этого ядро исполняет команды обработчика исключения/прерывания, который по окончании своей работы возвращается к исполнению прерванной команды, вызвавшей исключение/прерывание, в режиме U-mode. К исключениям/прерываниям, увеличивающим уровень привилегий, применяется термин вертикальные исключения/прерывания, в то время как к исключениям/прерываниям, оставляющим тот же уровень привилегий, применяется термин горизонтальные исключения/прерывания.

12.6 Регистры управления/состояния (CSR)

Доступ к регистрам управления/состояния (CSR) ядра осуществляется атомарно с использованием инструкций, специально разработанных для доступа к CSR. Инструкции по доступу к CSR перечислены в подразделе "Система команд".

Согласно спецификации RISC-V, ядро использует 12-разрядное пространство кодирования для адресации до 4096 регистров управления/состояния (CSR) в инструкциях, которые атомарно считывают и модифицируют CSR.

Ядро следует соглашению RISC-V, где верхние 4 бита адреса CSR [11:8] используются для кодирования доступности CSR для чтения и записи в соответствии с уровнем привилегий. Два верхних бита [11:10] указывают, доступен ли регистр для чтения/записи (00, 01 или 10) или только для чтения (11). Следующие два бита [9:8] указывают самый низкий уровень привилегий, который может обеспечить доступ к CSR (например, 00 для U-уровня).

Следующие сокращения используются для обозначения разрядности или свойств битовых полей в отдельных описаниях CSR:

– RO - Только для чтения, запись игнорируется;
– RZ - Считывается как ноль, запись игнорируется;
– R1 - Считывается как единица, запись игнорируется;
– RW - Чтение/запись;
– RW1C - Чтение/запись 1 для сброса (только для доступа к IPIC);
– RW0C - Чтение/запись 0 для сброса;
– RW1P - Чтение/запись 1 для импульса;
– RSW1S - Чтение статуса/запись 1 для установки (управление сбросом L1).

Ядро не генерирует исключение «недопустимая инструкция» (illegal instruction), если команда пытается записать неподдерживаемое значение в поле CSR.

Ядро реализует следующие правила для доступа CSR:

1) Попытки доступа к несуществующему CSR приводят к возникновению исключения «недопустимая инструкция»;

2) Попытки доступа к CSR без соответствующего уровня привилегий или записи в регистр, доступный только для чтения, также приводят к возникновению исключений «недопустимая инструкция»;

3) Если регистр чтения/записи содержит некоторые биты, доступные только для чтения, то запись в биты, доступные только для чтения, игнорируется.

Карта регистров CSR

Карта регистров управления/состояния CSR в режиме пользователя (U-mode) приведена в таблице 12.1.

Таблица 12.1 – CSR в U-mode

Адрес	Обозначение
FPU (стандарт, чтение/запись 0x000..0x0FF)	
0x001	FFLAGS
0x002	FRM
0x003	FCSR
Пользовательский счетчик/таймеры (стандарт, только чтение 0xC00..0xCBF)	
0xC00	CYCLE
0xC01	TIME
0xC02	INSTRET
0xC03	HPMCNT3
0xC04	HPMCNT4
0xC05	HPMCNT5
0xC06	HPMCNT6
0xC80	CYCLEH
0xC81	TIMEH
0xC82	INSTRETH
0xC83	HPMCNT3H
0xC84	HPMCNT4H
0xC85	HPMCNT5H
0xC86	HPMCNT6H

Карта регистров управления/состояния CSR в режиме супервизора (S-mode) приведена в таблице 12.2.

Таблица 12.2 – CSR в S-mode

Адрес	Обозначение
Настройка ловушки (trap) супервизора (стандарт, чтение/запись 0x100..0x1FF)	
0x100	SSTATUS
0x104	SIE
0x105	STVEC
0x106	SCOUNTEREN
Обработка ловушки супервизора (стандарт, чтение/запись 0x100..0x1FF)	
0x140	SSCRATCH
0x141	SEPC
0x142	SCAUSE
0x143	STVAL
0x144	SIP
Защита и трансляция супервизора (стандарт, чтение/запись 0x100..0x1FF)	
0x180	SATP
Регистры ICCM (нестандарт, чтение/запись 0x9C0-0x9FF)	
0x9D8	ICCM_SADDRESS
0x9D9	ICCM_SSTATUS
0x9DA	ICCM_SRDDATA
0x9DB	ICCM_SWRDATA

Карта регистров управления/состояния CSR в машинном режиме (M-mode) приведена в таблице 12.3.

Таблица 12.3 – CSR в M-mode

Адрес	Обозначение
Настройка машинной ловушки (стандарт, чтение/запись 0x300..0x3FF)	
0x300	MSTATUS
0x301	MISA
0x302	MEDELEG
0x303	MIDELEG
0x304	MIE
0x305	MTVEC
0x306	MCOUNTEREN
Машинная настройка счетчика (стандарт, чтение/запись 0x300..0x3FF)	
0x320	MCOUNTINHIBIT
0x323	MHPMEVENT3
0x324	MHPMEVENT4
0x325	MHPMEVENT5
0x326	MHPMEVENT6
Машинная обработка ловушки (стандарт, чтение/запись 0x300..0x3FF)	
0x340	MSCRATCH
0x341	MEPC
0x342	MCAUSE
0x343	MTVAL
0x344	MIP
Стандартные регистры чтения/записи и отладки (0x7A0..0x7AF)	
0x7A0	TSELECT
0x7A1	TDATA1
0x7A2	TDATA2
0x7A3	TDATA3
0x7A4	TINFO
Регистры режима отладки (0x7B0..0x7BF)	
0x7B0	DSCR
0x7B1	DPC
0x7B2	DSCRATCH0
Машинные счетчики/таймеры (стандартное чтение/запись 0xB00..0xBBF)	
0xB00	MCYCLE
0xB01	MTIME
0xB02	MINSTRET
0xB03	MHPMCNT3
0xB04	MHPMCNT4
0xB05	MHPMCNT5
0xB06	MHPMCNT6
0xB80	MCYCLEH
0xB81	MTIMEH
0xB82	MINSTRETH
0xB83	MHPMCNT3H
0xB84	MHPMCNT4H
0xB85	MHPMCNT5H
0xB86	MHPMCNT6H

Продолжение таблицы 12.3

Адрес	Обозначение
Регистры MMU/MPU (нестандарт, чтение/запись 0xBC0..0xBFF)	
0xBC0	MMU_TLB_ATTR
0xBC1	MMU_TLB_VA
0xBC2	MMU_TLB_UPDATE
0xBC3	MMU_TLB_SCAN
0xBC4	MPU_SELECT
0xBC5	MPU_CONTROL
0xBC6	MPU_ADDRESS
0xBC7	MPU_MASK
Регистры PCU (нестандарт, чтение/запись 0xBC0..0xBFF)	
0xBD4	MEM_CTRL_GLOBAL
0xBD8	ICCM_MADDRESS
0xBD9	ICCM_MSTATUS
0xBDA	ICCM_MRDDATA
0xBDB	ICCM_MWRDATA
0xBFC	PWR_CTRL
0xBFD	PWR_STATE
0xBFE	CG_CLKC
0xBFF	FG_CLKC
Стандарт, только чтение (0xF00..0xFBF)	
0xF11	MVENDORID
0xF12	MARCHID
0xF13	MIMPID
0xF14	MHARTID
Нестандарт, только чтение (0xFC0..0xFFF)	
0xFC1	MEFADDR
0xFC2	MEFCMD
0xFC3	CACHE_L1_DESCR
0xFD0	SWPW_R0_ADDR
0xFD1	SWPW_R1_ADDR

Регистры U-mode

FFLAGS

FFLAGS – это регистр, содержащий флаги исключений, возникшие при вычислении чисел с плавающей запятой.

Данный регистр является отображаемой частью регистра FCSR.

Поле FFLAGS в этом регистре совпадает с полем FFLAGS в реестре FCSR.

Структура регистра FFLAGS приведена в таблице 12.4.

Таблица 12.4 – Структура регистра FFLAGS

Биты	Название	Доступ	Описание
4-0	FFLAGS	RW	Накопленные флаги исключений с плавающей запятой
31-8	RSV	RZ	Зарезервировано

FRM

FRM – это регистр для настройки режима округления при вычислении чисел с плавающей запятой.

Данный регистр является отображаемой частью регистра FCSR.

Поле FRM в этом регистре совпадает с полем FRM в реестре FCSR.

Структура регистра FRM приведена в таблице 12.5.

Таблица 12.5 – Структура регистра FRM

Биты	Название	Доступ	Описание
2-0	FRM	RW	Режим округления с плавающей запятой
31-3	RSV	RZ	Зарезервировано

FCSR

Регистр управления/состояния с плавающей запятой в режиме U-mode FCSR представляет собой 32-разрядный регистр чтения/записи. Структура регистра FCSR, состоящего из поля FFLAGS, в котором хранятся накопленные флаги исключений, и поля FRM, в котором выбирается режим округления, показана в таблице 12.6.

Таблица 12.6 – Структура регистра FCSR

Биты	Название	Доступ	Описание
4-0	FFLAGS	RW	Накопленные флаги исключений с плавающей запятой
7-5	FRM	RW	Режим округления с плавающей запятой
31-8	RSV	RZ	Зарезервировано

Поле FRM выбирает режим округления для арифметических операций с плавающей запятой, как показано в таблице 12.7.

Поле FFLAGS содержит собранные флаги исключений, как показано в таблице 12.8.

Таблица 12.7 – Коды режимов округления FRM

FRM[2..0]	Мнемоника	Обозначение
000	RNE	Округление до ближайшего, привязка к четному
001	RTZ	Округление до нуля
010	RDN	Округлить в меньшую сторону
011	RUP	Округлить в большую сторону
100	RMM	Округление до ближайшего, привязка к максимальной величине
101	–	Зарезервировано
110	–	Зарезервировано
111	DYN	Динамический выбор режима округления в выделенном поле инструкции; Недопустимо в поле FRM

Операции с плавающей запятой используют либо статический режим округления, закодированный в инструкции, либо динамический режим округления, сохраняемый в FRM. Значение 111 в поле rm инструкции выбирает режим динамического округления, поддерживаемый в FRM. Если для в поле FRM задано недопустимое значение (101..111), любая последующая попытка выполнить операцию с плавающей запятой в режиме динамического округления приведет к сбою «недопустимая команда». На некоторые команды, имеющие поле rm, режим округления, тем не менее, не влияет; для их поля rm должно быть установлено значение RNE (000).

Таблица 12.8 – Кодировка полученного флага исключения FFLAGS

Бит	Мнемоника флага	Обозначение флага
4	NV	Недопустимая операция
3	DZ	Деление на ноль
2	OF	Переполнение
1	UF	Недостаточное заполнение
0	NX	Неточный результат

Регистр FCSR может быть прочитан и записан с помощью соответствующих псевдоинструкций ассемблера, FRCSR и FSCSR, которые построены на основе базовых инструкций доступа к CSR.

К полям в FCSR также можно обращаться индивидуально через разные адреса CSR, и для этих обращений определены отдельные псевдоинструкции ассемблера, такие как FRRM и FRFLAGS. Команда FRRM считывает 3-разрядное поле режима округления и копирует его в три младших бита данного целочисленного регистра с нулем во всех остальных битах.

FSRM меняет местами значение 3-разрядного поля режима округления, копируя исходное значение в заданный целочисленный регистр, а затем записывая новое значение, полученное из трех младших разрядов данного целочисленного регистра.

FRFLAGS и FSFLAGS определены аналогично для флагов полей полученных флагов исключений. Дополнительные псевдоинструкции ассемблера FSRMI и FSFLAGSI меняют значения местами, используя непосредственное значение вместо целочисленного регистра.

TIME/TIMEH

TIME и TIMEH в U-режиме являются теневыми регистрами, доступными только для чтения, регистров MTIME и MTIMEH соответственно. Для подробно информации см. «Карта регистров Mtimer».

INSTRET/INSTRETH

INSTRET и INSTRETH в U-режиме являются регистрами, доступными только для чтения, регистров MINSTRET и MINSTRETH MINSTRETH соответственно.

HPMCNTn/HPMCNTHn

HPMCNT3...HPMCNT6 и HPMCNT3H...HPMCNT6H в U-режиме являются теневыми регистрами, доступными только для чтения, регистров MHPMCNT3...MHPMCNT6 и MHPMCNT3H...MHPMCNT6H соответственно.

Регистры S-mode

SSTATUS

Регистр SSTATUS отслеживает текущее рабочее состояние ядра. Структура регистра SSTATUS приведена в таблице 12.9.

Таблица 12.9 – Структура регистра SSTATUS

Биты	Название	Доступ	Описание
0	RSV	RZ	Зарезервировано
1	SIE	RW	Включение прерывания в режиме супервизора
3-2	RSV	RZ	Зарезервировано
4	RSV	RZ	Зарезервировано
5	SPIE	RW	Включение прерывания в предыдущем режиме супервизора
7-6	RSV	RZ	Зарезервировано
8	SPP	RW	Предыдущий привилегированный режим супервизора
12-9	RSV	RZ	Зарезервировано
14-13	FS	RW	Код статуса режима с плавающей запятой
16-15	XS	RZ	Аппаратно установлен в 0
17	RSV	RZ	Зарезервировано
18	SUM	RW	Разрешение доступа к памяти супервизора пользователя
19	MXR	RW	Сделать исполняемый доступным для чтения
30-20	RSV	RZ	Зарезервировано
31	SD	RO	Итоговый «грязный» («dirty state») бит. Означает, что данные были изменены и отличаются от исходных

Значение по умолчанию после сброса 0x20.

SSTATUS – это ограниченное представление MSTATUS. Поля SSTATUS SIE, SPIE, SPP, FS, SD, SUM, MXR, UXL являются омонимами одноименных полей MSTATUS. Запись этих полей в SSTATUS имеет эффект в MSTATUS и наоборот.

SSTATUS доступен для чтения/записи из режимов S-mode и M-mode. Попытки доступа к SSTATUS из U-mode приводят к исключениям «недопустимая команда».

Бит SPP указывает уровень привилегий, в котором работало ядро перед переходом в режим супервизора. При срабатывании ловушки (trap), SPP устанавливается равным 0, если ловушка была вызвана из пользовательского режима, в противном случае – 1.

При выполнении инструкции SRET, для возврата из обработчика ловушки, уровень привилегий устанавливается в режим пользователя, если бит SPP равен 0, или в режим супервизора, если бит SPP равен 1, затем SPP устанавливается равным 0.

Бит SIE включает или отключает все прерывания в режиме супервизора. Когда SIE сброшен, прерывания в режиме супервизора не обрабатываются. Когда ядро работает в U-режиме, значение в SIE игнорируется, а прерывания на уровне супервизора включены. Супервизор может отключить отдельные источники прерываний, используя регистр SIE.

Бит SPIE указывает, были ли включены прерывания супервизора до перехода в режим супервизора. Когда получена ловушка в режиме супервизора, SPIE устанавливается в значение SIE, а SIE - в 0.

Когда выполняется инструкция SRET, для SIE устанавливается значение SPIE, затем для SPIE устанавливается значение 1.

Привилегии доступа к памяти в регистре SSTATUS

Бит MXR (Make eXecutable Readable) изменяет привилегию, с помощью которой загружаемые файлы получают доступ к виртуальной памяти. Если MXR=0, загрузка будет успешной только со страниц, помеченных как доступные для чтения (R=1).

Если MXR=1, загрузка со страниц, помеченных как доступные для чтения или исполняемые (R=1 или X=1), будет успешной.

MXR не имеет никакого эффекта, если не задействована виртуальная память на основе страниц.

Бит SUM (Supervisor User Memory access) изменяет привилегию, с помощью которой режим S-mode получает доступ к виртуальной памяти для загрузки и сохранения. При значении SUM=0 произойдет сбой при обращении к памяти в режиме S-mode к страницам, доступным в режиме U-mode (U=1). Когда SUM =1, эти обращения разрешены. SUM не имеет никакого эффекта, если не задействована постраничная виртуальная память, а также при выполнении в U-режиме. Обратите внимание, что S-mode никогда не сможет выполнять инструкции со страниц пользователя, независимо от состояния SUM.

Базовое управление ISA в регистре SSTATUS

UXL – это поле, которое кодирует текущее значение UXLEN (пользовательская ширина целочисленного регистра). В текущей реализации UXLEN = MXLEN (машинная ширина целочисленного регистра). Всякий раз, когда для XLEN в любом режиме задается значение, меньшее, чем самый широкий поддерживаемый XLEN, все операции должны игнорировать разряды регистра исходного операнда, превышающие настроенный XLEN, и должны расширять результаты для заполнения всего самого широкого поддерживаемого XLEN в целевом регистре. Аналогично, биты pc выше XLEN игнорируются, и когда pc записывается, он расширяется по знаку, чтобы заполнить максимально широкий поддерживаемый XLEN.

Статус контекста расширения в регистре SSTATUS

Поле FS[1:0] RW используется для снижения затрат на сохранение и восстановление контекста путем настройки и отслеживания текущего состояния модуля с плавающей запятой. Поле FS кодирует статус единицы измерения с плавающей запятой, включая CSR FCSR и регистры данных с плавающей запятой f0..f64. Это поле можно проверить с помощью процедуры переключения контекста, чтобы быстро определить, требуется ли сохранение или восстановление состояния. Если требуется сохранение или восстановление, обычно требуются дополнительные инструкции и CSR для улучшения и оптимизации процесса. В поле FS используется кодировка состояния, показанная в таблице 12.10, с четырьмя возможными значениями состояния.

Таблица 12.10 – Кодировка поля состояния FS[1:0]

Состояние	Значение поля FS
00	Выключено
01	Инициализация
10	Очищено
11	«Грязный»

Бит SD – это бит, доступный только для чтения, который обобщает, сигнализирует ли поле FS о наличии некоторого «грязного» состояния («dirty state»), требующего сохранения расширенного пользовательского контекста в памяти.

SIE

SIE – это 32-разрядный регистр, доступный для чтения и записи, содержащий биты разрешения прерывания, в то время как регистр SIP - это соответствующий 32-разрядный RW регистр, содержащий информацию об ожидающих прерываниях.

Структура регистра SIE приведена в таблице 12.11.

Таблица 12.11 – Структура регистра SIE

Биты	Наименование	Доступ	Описание
0	RSV	RZ	Зарезервировано
1	SSIE	RW	Разрешение программных прерываний в режиме супервизора
4-2	RSV	RZ	Зарезервировано
5	STIE	RW	Разрешение прерывания по таймеру в режиме супервизора
8-6	RSV	RZ	Зарезервировано
9	SEIE	RW	Разрешение внешних прерываний в режиме супервизора
31-10	RSV	RZ	Зарезервировано

Определены три типа прерываний: программные прерывания, прерывания по таймеру и внешние прерывания. Программное прерывание в режиме супервизора запускается на текущем ядре путем записи 1 в бит ожидающих программных прерываний супервизора (SSIP) в регистре SIP. Ожидающее программное прерывание в режиме супервизора может быть очищено путем записи 0 в бит SSIP в регистре SIP. Программные прерывания в режиме супервизора отключаются, когда бит SSIE в регистре SIE очищен.

Межпроцессорные прерывания передаются другому ядру посредством вызовов SBI, что в конечном итоге приводит к установке бита SSIP в SIP-регистре ядра получателя.

Прерывание по таймеру уровня супервизора находится в ожидании, если установлен бит STIP в регистре SIP. Прерывания по таймеру в режиме супервизора отключаются, когда бит STIE в регистре SIE очищен. Вызов SBI в SEE может быть использован для сброса ожидающего прерывания по таймеру.

Внешнее прерывание в режиме супервизора находится в ожидании, если установлен бит SEIP в регистре SIP. Внешние прерывания в режиме супервизора отключаются, когда бит SEIE в регистре SIE сброшен. SBI предоставляет средства для маскирования, размаскирования и запроса источника внешних прерываний.

Регистры SIP и SIE являются подгруппами регистров MIP и MIE. Чтение любого поля или запись любого доступного для записи поля регистров SIP/SIE влияет на чтение или запись одноименного поля MIP/MIE.

SIE доступен для чтения/записи из S-режима и M-режима. Попытки доступа к SIE из режима U-mode приводят к возникновению исключения «недопустимая инструкция».

STVEC

Регистр STVEC представляет собой 32-разрядный регистр чтения/записи, который содержит векторную конфигурацию ловушки (trap), состоящую из вектора базового адреса (BASE) и вектора режима (MODE). Структура регистра STVEC приведена в таблице 12.12.

Таблица 12.12 – Структура регистра STVEC

Биты	Наименование	Доступ	Описание
1-0	MODE	RW	Вектор режима
31-2	BASE	RW	Вектор базового адреса

Значение по умолчанию после сброса 0x0.

Поле BASE в STVEC – это поле, доступное для чтения и записи, которое может содержать любой допустимый виртуальный или физический адрес при соблюдении следующих ограничений по выравниванию: адрес всегда должен быть выровнен по границе 4 байт, а настройка MODE может накладывать дополнительные ограничения по выравниванию на значение в поле BASE.

Кодировка поля MODE приведена в таблице 12.13.

Когда MODE=Direct, все ловушки в режиме супервизора приводят к тому, что программный счетчик (pc) устанавливается по адресу, указанному в поле BASE. Когда MODE=Vectored все синхронные исключения в режиме супервизора приводят к тому, что для pc устанавливается адрес, указанный в поле BASE, в то время как прерывания приводят к тому, что для pc устанавливается адрес, указанный в поле BASE, плюс четырехкратный номер прерывания CAUSE.

Таблица 12.13 – Кодировка поля MODE

Значение	Наименование	Описание
0	Direct	Все исключения устанавливают pc в BASE
1	Vectored	Асинхронные прерывания устанавливают pc в BASE +4*CAUSE
2	RSV	Зарезервировано, будет записано значение 0
3	RSV	Зарезервировано, будет записано значение 1

STVEC доступен для чтения/записи из режимов S-mode и M-mode. Попытки доступа к STVEC из U-mode приводят к исключению «недопустимая инструкция».

SCOUNTEREN

Дополнительные сведения см. в разделе «Аппаратные счетчики мониторинга производительности».

SSCRATCH

Регистр SSCRATCH – это 32-разрядный регистр, доступный для чтения и записи, предназначенный для использования супервизором. Как правило, SSCRATCH используется для сохранения указателя на локальный для ядра контекст супервизора, пока ядро выполняет пользовательский код. В начале обработчика перехвата SSCRATCH заменяется регистром пользователя, чтобы обеспечить начальный рабочий регистр.

Структура регистра SSCRATCH приведена в таблице 12.14.

Таблица 12.14 – Структура регистра SSCRATCH

Биты	Наименование	Доступ	Описание
31-0	SSCRATCH	RW	Как правило, он используется для хранения указателя на локальный для ядра контекст супервизора, пока ядро выполняет пользовательский код

SSCRATCH доступен для чтения/записи из режимов S-mode и M-mode. Попытки доступа к SSCRATCH из режима U-mode приводят к исключению «недопустимая инструкция».

SEPC

SEPC – это 32-разрядный регистр, доступный для чтения и записи, со структурой, показанной в таблице 12.15. Младший бит SEPC (SEPC[0]) всегда равен нулю.

Когда в режиме S-mode срабатывает ловушка (trap), SEPC записывается виртуальным адресом инструкции, которая столкнулась с исключением. В противном случае SEPC никогда не записывается в данной реализации, хотя он может быть явно записан программным обеспечением.

Таблица 12.15 – Структура регистра SEPC

Биты	Наименование	Доступ	Описание
0	RSV	RZ	Аппаратно установлен в 0
31-1	sepc	RW	Адрес инструкции, в которой возникло исключение

SEPC доступен для чтения/записи из режимов S-mode и M-mode. Попытки доступа к SEPC из режима U-mode приводят к исключению «недопустимая инструкция».

SCAUSE

Регистр SCAUSE представляет собой 32-разрядный RW-регистр со структурой, показанной в таблице 12.16. Когда в режиме S-mode срабатывает ловушка (trap), SCAUSE записывается кодом, указывающим на событие, вызвавшее ловушку. В противном случае SCAUSE никогда не записывается в данной реализации, хотя может быть явно записан программным обеспечением.

Таблица 12.16 – Структура регистра SCAUSE

Биты	Наименование	Доступ	Описание
4-0	EC	RW	Код исключения
30-5	RSV	RZ	Зарезервировано
31	INT	RO	Прерывание

Бит прерывания в регистре SCAUSE устанавливается, если сбой был вызван прерыванием. Код исключения содержит код, идентифицирующий последнее исключение.

Список возможных кодов исключений SCAUSE приведен в таблице 12.17 в порядке убывания приоритета.

Таблица 12.17 – Описание кодов исключений SCAUSE

Прерывание	Исключение	Описание
0	0	Адрес инструкции не выровнен
0	1	Ошибка доступа к инструкции
0	2	Недопустимая инструкция
0	3	Точка останова
0	4	Загружаемый адрес не выровнен
0	5	Ошибка доступа к загрузке
0	6	Адрес сохранения/AMO не выровнен
0	7	Ошибка доступа сохранения/AMO
0	8	Вызов среды из режима U-mode
0	9-11	Зарезервировано
0	12	Ошибка страницы инструкции
0	13	Ошибка загрузки страницы
0	14	Зарезервировано
0	15	Ошибка страницы сохранения/AMO
0	31-16	Зарезервировано
1	0	Зарезервировано
1	1	Программное прерывание супервизора
1	4-2	Зарезервировано
1	5	Прерывание таймера супервизора
1	8-6	Зарезервировано
1	9	Внешнее прерывание супервизора
1	31-10	Зарезервировано

STVAL

Регистр STVAL представляет собой 32-разрядный регистр, доступный для чтения и записи. При срабатывании ловушки (trap) в режиме S-mode, STVAL записывается информацией об исключениях, чтобы помочь программной обработке ловушки. В противном случае STVAL никогда не записывается в данной реализации, хотя может быть явно записан программным обеспечением. Структура регистра STVAL представлена в таблице 12.18.

Таблица 12.18 – Структура регистра STVAL

Биты	Наименование	Доступ	Описание
31-0	STVAL	RW	STVAL записывается информацией об исключениях, чтобы помочь программной обработке ловушки

Когда срабатывает аппаратная точка останова или возникает ошибка при выборке команды, загрузке или сохранении адреса, доступе или сбое страницы, STVAL записывается виртуальным адресом, вызывающим сбой. В случае срабатывания ловушки «недопустимая инструкция» STVAL может быть записан с первыми битами XLEN команды, вызывающей ошибку, как описано ниже. Для других исключений значение STVAL равно нулю. STVAL – это регистр WARL, способный хранить все допустимые физические и виртуальные адреса и значение 0. STVAL доступен для чтения/записи из режимов S-mode и M-mode. Попытки доступа к STVAL из режима U-mode приводят к возникновению исключения «недопустимая инструкция».

SIP

Регистр SIP – это 32-разрядный регистр, доступный для чтения и записи, содержащий информацию об ожидающих прерываниях, в то время как SIE – это соответствующий 32-разрядный регистр, доступный для чтения и записи, содержащий биты разрешения прерывания. Структура регистра SIP представлена в таблице 12.19.

Таблица 12.19 – Структура регистра SIP

Биты	Наименование	Доступ	Описание
0	RSV	RZ	Зарезервировано
1	SSIP	RW	Ожидающее программное прерывание супервизора
4-2	RSV	RZ	Зарезервировано
5	STIP	RO	Ожидающее прерывание таймера супервизора
8-6	RSV	RZ	Зарезервировано
9	SEIP	RO	Ожидающее внешнее прерывание супервизора
31-10	RSV	RZ	Зарезервировано

Только бит SSIP регистра SIP является доступным для чтения и записи, все остальные доступны для чтения. SIP доступен для чтения/записи из режимов S-mode и M-mode. Попытки доступа к SIP из режима U-mode приводят к возникновению исключения «недопустимая инструкция».

SATP

Регистр SATP представляет собой 32-разрядный регистр RW со структурой, показанной в таблице 12.20. Регистр SATP управляет трансляцией адресов и защитой в режиме супервизора. Этот регистр содержит номер физической страницы (PPN) таблицы корневых страниц, т.е. физический адрес супервизора, разделенный на 4 Кбайт; идентификатор адресного пространства (ASID), который облегчает преобразование адресов для каждого адресного пространства; и поле MODE, в котором выбирается текущая схема преобразования адресов.

Таблица 12.20 – Структура регистра SATP

Биты	Наименование	Доступ	Описание
21-0	PPN	WARL	Физический номер страницы в таблице корневых страниц
30-22	ASID	WARL	Идентификатор адресного пространства
31	MODE	WARL	Селектор текущей схемы преобразования адресов: 0 – Без трансляции или защиты (Bare); 1 – 32-разрядная виртуальная адресация на основе страниц (Sv32).

Количество реализованных битов ASID может быть определено с помощью:

– записи единицы в каждую позицию бита в поле ASID;

– дальнейшего считывания обратно значения в SATP, чтобы увидеть, какие позиции битов в поле ASID содержат единицу.

Младшие биты ASID вводятся в первую очередь.

Запись SATP не подразумевает каких-либо ограничений по порядку между обновлениями таблицы страниц и последующими преобразованиями адресов. Если таблицы страниц нового адресного пространства были изменены, может потребоваться выполнить инструкцию SFENCE.VMA перед записью SATP.

Регистры ICCM супервизора

Регистры ICCM – это пользовательские регистры, которые не определены спецификацией RISC-V. Для получения подробной информации обратитесь к подразделу «Модуль межъядерной связи (ICCM)».

Регистры M-mode

MSTATUS

Регистр MSTATUS – это 32-разрядный регистр, доступный для чтения/записи, который отслеживает текущее рабочее состояние ядра и управляет им. Ограниченные представления регистра MSTATUS отображаются как регистры SSTATUS и USTATUS в ISA уровней S-mode и U-mode соответственно. Структура регистра MSTATUS представлена в таблице 12.21.

Таблица 12.21 – Структура регистра MSTATUS

Поле	Биты	Доступ	Описание
RSV	0	RZ	Зарезервировано
SIE	1	RW	Включение прерывания в режиме супервизора
RSV	2	RZ	Зарезервировано
MIE	3	RW	Включение прерывания в машинном режиме
RSV	4	RZ	Зарезервировано
SPIE	5	RW	Включение прерывания в предыдущем режиме супервизора
RSV	6	RZ	Зарезервировано
MPIE	7	RW	Включение прерывания в предыдущем машинном режиме
SPP	8	RW	Предыдущий приоритетный режим супервизора
RSV	9-10	RZ	Зарезервировано
MPP	11-12	RW	Предыдущий приоритетный машинный режим
FS	13-14	RW	Кодирование состояния с плавающей запятой 00 Отключено 01 Инициализация 10 Стирание 11 «Грязный»
XS	15-16	RZ	Подключение к «0»
MPRV	17	RW	Изменение режима приоритета
SUM	18	RW	Разрешение доступа пользователя к памяти супервизора
MXR	19	RW	Исполняемый файл доступен для чтения
TVM	20	RW	Ловушка виртуальной памяти
TW	21	RW	Тайм-аут ожидания
TSR	22	RW	Ловушка SRET
RSV	23-30	RZ	Зарезервировано
SD	31	RO	Итоговый «грязный» бит

Значение MSTATUS по умолчанию после сброса равно 0xA0.

Стек включения привилегий и глобальных прерываний в регистре MSTATUS

Биты разрешения прерывания MIE, SIE предусмотрены для каждого привилегированного режима. Эти биты в основном используются для обеспечения атомарности по отношению к обработчикам прерываний на текущем уровне привилегий. Как правило, когда ядро выполняется в привилегированном режиме x, прерывания

включаются, когда xIE=1. Прерывания для режимов с более низкими привилегиями всегда отключены, в то время как прерывания для режимов с более высокими привилегиями всегда включены. Код с более высоким уровнем привилегий может использовать отдельные биты включения для каждого прерывания, чтобы отключить выбранные прерывания перед передачей управления более низкому уровню привилегий.

Для поддержки вложенных ловушек каждый режим привилегий x имеет двухуровневый стек битов, разрешающих прерывание, и режимов привилегий. xPIE сохраняет значение бита разрешения прерывания активным до срабатывания ловушки, а xPP сохраняет предыдущий режим привилегий. Поля xPP могут содержать только режимы привилегий до x, поэтому ширина MPP составляет два бита. Когда ловушка переводится из режима привилегий y в режим привилегий x, соответствующему биту xPIE устанавливается значение xIE; xIE устанавливается равным 0; а xPP устанавливается равным y.

Команда MRET используется для возврата из обработчиков ловушек в режиме M-mode. В общем случае, при выполнении инструкции xRET, предполагающей, что xPP содержит значение y, для xIE устанавливается значение xPIE; режим привилегий изменяется на y; для xPIE устанавливается значение 1; а для xPP устанавливается значение U. Поля xPP должны содержать только поддерживаемые режимы привилегий, включая x и любой реализованный режим привилегий ниже x.

Привилегия памяти в регистре MSTATUS

Бит MPRV (Modify Privilege) изменяет уровень привилегий, при котором выполняется загрузка и сохранение данных во всех режимах привилегий. Когда MPRV=0, трансляция и защита работают в обычном режиме. Когда MPRV=1, адреса памяти для загрузки и сохранения преобразуются и защищаются так, как если бы для текущего режима привилегий было установлено значение MPP. Инструкция адрес-трансляция и защита остаются неизменными.

Бит SUM (permit Supervisor User Memory access) изменяет привилегию, с помощью которой S-mode загружает, сохраняет и извлекает команды с доступом к виртуальной памяти.

Когда SUM=0, в режиме S-mode память будет получать доступ к страницам, которые доступны в режиме U-mode (U=1, см. подраздел «Регистр атрибутов страницы памяти»), произойдет сбой.

Если SUM=1, эти обращения разрешены. SUM не имеет влияния, если виртуальная память на основе страниц не используется. Обратите внимание, что, хотя SUM обычно игнорируется, когда он не работает в режиме S-mode, он действует, когда MPRV=1 и MPP=S. SUM аппаратно установлен в 0, если режим S-mode не поддерживается.

Бит MXR (Make Executable Readable) изменяет привилегию, с помощью которой загрузки получают доступ к виртуальной памяти. Если MXR=0, загрузка будет успешной только со страниц, помеченных как доступные для чтения. Если MXR=1, загрузка со страниц, помеченных как доступные для чтения или исполняемые (R=1 или X=1), будет успешной. MXR не имеет влияния, когда виртуальная память на основе страниц отключена.

Поддержка виртуализации в регистре MSTATUS

Бит TVM (Trap Virtual Memory) поддерживает перехват операций управления виртуальной памятью супервизора. Когда TVM=1, выполняется попытка считывания или записи в регистр SATP или выполнения SFENCE. Инструкция VMA при выполнении в режиме S-mode вызовет исключение «недопустимая инструкция». Когда TVM=0, эти операции разрешены в режиме S-mode.

Бит TW (Timeout Wait) поддерживает перехват команды WFI. Когда TW=0, инструкция WFI разрешена в S-режиме. Когда TW=1, если WFI выполняется в режиме S-mode и не завершается в течение определенного для конкретной реализации ограниченного периода времени, инструкция WFI вызывает ловушку с исключением «недопустимая

инструкция». Ограничение по времени всегда может быть равно 0, и в этом случае WFI всегда приводит к срабатыванию ловушки «недопустимая инструкция» в режиме S-mode при TW=1.

Бит TSR (Trap SRET) поддерживает перехват команды возврата исключения супервизора, SRET. Когда TSR=1, попытки выполнить SRET при выполнении в режиме S-mode вызовут исключение «недопустимая инструкция». Когда TSR=0, эта операция разрешена в режиме S-mode.

Расширенный статус контекста в регистре MSTATUS

Поле FS[1:0] RW используется для снижения затрат на сохранение и восстановление контекста путем настройки и отслеживания текущего состояния модуля вычислений с плавающей запятой. Поле FS кодирует статус модуля вычислений с плавающей запятой, включая CSR FCSR и регистры данных с плавающей запятой f0..f31. Это поле можно проверить с помощью процедуры переключения контекста, чтобы быстро определить, требуется ли сохранение или восстановление состояния. Если требуется сохранение или восстановление, обычно требуются дополнительные инструкции и CSR для улучшения и оптимизации процесса.

Поле FS содержит код состояния с четырьмя возможными значениями (см. Таблицу 12.22).

Таблица 12.22 – Кодировка поля FS

Код статуса	Значение статуса
00	Выключен
01	Инициализация
10	Очистка
11	«Грязный»

Бит SD – это бит, доступный для чтения, который обобщает, сигнализирует ли поле FS о наличии некоторого «грязного» состояния, требующего сохранения расширенного пользовательского контекста в памяти.

MISA

Регистр MISA – это 32-разрядный регистр, доступный только для чтения, который сообщает об ISA, поддерживаемых ядром.

MISA в ядре аппаратно установлен в значение 0x4014912D.

Структура регистра MISA представлена в таблице 12.23.

Таблица 12.23 – Структура регистра MISA

Поле	Биты	Доступ	Описание
RVA	0	R1	Реализовано расширение атомарных инструкций
RSV	1	RZ	Зарезервировано
RVC	2	R1	Реализовано расширение компактных инструкций
RVD	3	R1	Реализовано расширение с плавающей запятой двойной точности
RSV	4	RZ	Зарезервировано
RVF	5	R1	Реализовано расширение с плавающей запятой одинарной точности
RSV	6-7	RZ	Зарезервировано
RVI	8	R1	Реализован базовый ISA RV32I
RSV	9-11	RZ	Зарезервировано

Продолжение таблицы 12.23

Поле	Биты	Доступ	Описание
RVM	12	R1	Реализовано расширение для умножения/деления целых чисел
RSV	13-14	RZ	Зарезервировано
RVP	15	R1	Реализовано расширение инструкций RV32P
RSV	16-17	RZ	Зарезервировано
RVS	18	R1	Реализован режим супервизора
RSV	19	RZ	Зарезервировано
RVU	20	R1	Реализован пользовательский режим
RSV	21-29	RZ	Зарезервировано
MXL	30-31	RO	Исходным базовым целым ISA является константа 2b01, которая обозначает RV32I

Регистры делегирования машинных ловушек

По умолчанию все ловушки на любом уровне привилегий обрабатываются в машинном режиме, хотя обработчик машинного режима может перенаправить ловушки обратно на соответствующий уровень с помощью команды MRET. Ядро предоставляет отдельные биты чтения/записи в MEDELEG и MIDELEG, указывающие на то, что определенные исключения и прерывания должны обрабатываться непосредственно с более низким уровнем привилегий. Регистр делегирования машинных исключений (MEDELEG) и регистры делегирования машинных прерываний (MIDELEG) являются 32-разрядными регистрами чтения/записи.

Поскольку ядро реализует три режима привилегий (M/S/U), установка бита в MEDELEG или MIDELEG делегирует соответствующую ловушку в режиме S-mode или U-mode обработчику ловушки в режиме S-mode.

MEDELEG

Структура регистра MEDELEG представлена в таблице 12.24.

Таблица 12.24 – Структура регистра MEDELEG

Поле	Биты	Доступ	Описание
IAM	0	RW	Адрес инструкции не выровнен
IAF	1	RZ	Ошибка доступа к инструкции. Аппаратно установлен в 0
ILI	2	RW	Недопустимая инструкция
BRK	3	RW	Точка останова
LAM	4	RW	Адрес загрузки не выровнен
LAF	5	RZ	Ошибка доступа к загрузке. Аппаратно установлен в 0
SAM	6	RW	Адрес сохранения/AMO не выровнен
SAF	7	RZ	Ошибка доступа к сохранению/AMO. Аппаратно установлен в 0
UCALL	8	RW	Вызов среды из режима U-mode
SCALL	9	RZ	Вызов среды из режима S-mode. Аппаратно установлен в 0
RSV	10	RZ	Зарезервировано
MCALL	11	RZ	Вызов среды из M-режима. Аппаратно установлен в 0
IAPF	12	RW	Ошибка страницы инструкции
LAPF	13	RW	Ошибка страницы загрузки
TMPF	14	RZ	Ошибка TLB с пропущенной страницей. Аппаратно установлен в 0
SAPF	15	RW	Ошибка страницы сохранения/AMO
RSV	16-31	RZ	Зарезервировано

MIDELEG

Структура регистра MIDELEG представлена в таблице 12.25.

Таблица 12.25 – Структура регистра MIDELEG

Поле	Биты	Доступ	Описание
SW прерывание			
RSV	0	RZ	Зарезервировано
SSI	1	RW	Программное прерывание в режиме супервизора
RSV	2	RZ	Зарезервировано
MSI	3	RZ	Программное прерывание в режиме M-mode. Аппаратно установлен в 0
Прерывание таймера			
RSV	4	RZ	Зарезервировано
STI	5	RW	Прерывание по таймеру в режиме супервизора
RSV	6	RZ	Зарезервировано
MTI	7	RZ	Прерывание по таймеру в режиме M-mode. Аппаратно установлен в 0
Внешнее прерывание			
RSV	8	RZ	Зарезервировано
SEI	9	RW	Внешнее прерывание в режиме супервизора
RSV	10	RZ	Зарезервировано
MEI	11	RZ	Внешнее прерывание в режиме M-mode. Аппаратно установлен в 0
RSV	12-30	RZ	Зарезервировано
Ошибка памяти, прерывание из-за сбоя			
MMEFI	31	RZ	Прерывание по сбою ошибки памяти в режиме M-mode. Аппаратно установлен в 0

MIE

MIE – это 32-разрядный регистр, доступный для чтения и записи, содержащий биты разрешения прерывания, в то время как регистр MIP – это соответствующий 32-разрядный регистр, доступный для чтения и записи, содержащий информацию об ожидающих прерываниях.

Структура регистра MIE представлена в таблице 12.26.

Таблица 12.26 – Структура регистра MIE

Поле	Биты	Доступ	Описание
RSV	0	RZ	Зарезервировано
SSIE	1	RW	Включение программных прерываний супервизора
RSV	2	RZ	Зарезервировано
MSIE	3	RW	Включение программных прерываний машинного режима
RSV	4	RZ	Зарезервировано
STIE	5	RW	Включение прерывания по таймеру супервизора
RSV	6	RZ	Зарезервировано
MTIE	7	RW	Включение прерывания по таймеру машинного режима
RSV	8	RZ	Зарезервировано
SEIE	9	RW	Включение внешнего прерывания супервизора
RSV	10	RZ	Зарезервировано
MEIE	11	RW	Включение внешнего прерывания машинного режима
RSV	12-30	RZ	Зарезервировано
MEFIE	31	RW	Включение прерывания из-за ошибки памяти

Ограниченные виды регистров MIE и MIP отображаются соответственно как SIE и SIP в режиме S-mode. Если прерывание делегируется привилегированному режиму x путем установки бита в регистре MIDELEG, оно становится видимым в регистре SIP и может быть замаскировано с помощью регистра SIE. В противном случае соответствующие биты в SIP и SIE будут аппаратно установлены в ноль.

Биты MSIE и SSIE в регистре MIE, если они установлены, включают биты разрешения программного прерывания для режимов M-mode и S-mode соответственно.

Биты MTIE и STIE в регистре MIE, если они установлены, включают биты разрешения прерывания по таймеру для режимов M-mode и S-mode соответственно.

Биты MEIE и SEIE в регистре MIE, если они установлены, включают биты, разрешающие внешние прерывания, для режимов M-mode и S-mode соответственно.

Бит MEFIE в MIE включает прерывания из-за ошибок памяти, если он установлен.

Прерывание i будет принято, если бит i установлен как в MIP, так и в MIE, и если прерывания включены глобально. По умолчанию прерывания в режиме M-mode включены глобально, если текущий режим привилегий ядра меньше M или если текущий режим привилегий равен M и установлен бит MIE в регистре MSTATUS.

Однако, если установлен бит i в MIDELEG, прерывания считаются глобально разрешенными, если текущий режим привилегий ядра равен режиму делегированных привилегий (например, режиму S-mode) и установлен бит разрешения прерываний в этом режиме (SIE в MSTATUS), или если текущий режим привилегий меньше, чем установленный в MSTATUS режим делегированных привилегий.

Множественные одновременные прерывания и ловушки на одном и том же уровне привилегий обрабатываются в следующем порядке убывания приоритета:

– прерывание из-за ошибки памяти;
– внешние прерывания;
– программные прерывания;
– прерывания по таймеру;
– любые синхронные ловушки.

MTVEC

Регистр MTVEC представляет собой 32-разрядный регистр, доступный для чтения и записи, который содержит конфигурацию вектора ловушки, состоящую из вектора базового адреса (BASE) и вектора режима (MODE). Структура регистра MTVEC приведена в таблице 12.27.

Таблица 12.27 – Структура регистра MTVEC

Поле	Биты	Доступ	Описание
MODE	1-0	RW	Вектор режима
BASE	31-2	RW	Вектор базового адреса

Значение по умолчанию после сброса – 0x000001C0.

Поле BASE в MTVEC – это поле, доступное для чтения и записи, которое может содержать любой допустимый виртуальный или физический адрес при соблюдении следующих ограничений по выравниванию: адрес всегда должен быть выровнен по границе 4 байта, а настройка MODE может накладывать дополнительные ограничения по выравниванию на значение в поле BASE.

Кодировка поля MODE приведена в таблице 12.28. При значении MODE=Direct все ловушки в машинном режиме приводят к тому, что pc переходит в адрес, указанный в поле BASE.

При выборе MODE=Vectored все синхронные исключения в машинном режиме приводят к тому, что pc переходит в адрес, указанный в поле BASE, в то время как прерывания приводят к тому, что pc переходит в адрес, указанный в поле BASE, плюс четырехкратный номер источника прерывания. Например, прерывание таймера в машинном режиме приводит к тому, что pc устанавливается в BASE + 0x1C. Установка MODE=Vectored может наложить дополнительное ограничение на выравнивание по BASE, требуя выравнивания до 4x32 байт.

Таблица 12.28 – Кодировка поля MODE

Биты	Наименование	Описание
0	Прямой (Direct)	Все исключения устанавливают pc в BASE
1	Векторизованный (Vectored)	Асинхронные прерывания переводят pc в BASE +4*CAUSE
2	Зарезервировано	Зарезервировано, будет записано значение 0
3	Зарезервировано	Зарезервировано, будет записано значение 1

MCOUNTEREN

Информацию см. в подразделе «Счетчики мониторинга аппаратной производительности».

MCOUNTINHIBIT

Информацию см. в подразделе «Счетчики мониторинга аппаратной производительности».

MHPMEVENTn

Информацию см. в подразделе «Счетчики мониторинга аппаратной производительности».

MSCRATCH

Регистр MSCRATCH – это 32-разрядный регистр, доступный для чтения и записи, предназначенный для использования в машинном режиме. Как правило, он используется для хранения указателя на локальное для ядра контекстное пространство в машинном режиме и заменяется пользовательским регистром при входе в обработчик ловушки в режиме M-mode.

Структура регистра MSCRATCH приведена в таблице 12.29.

Таблица 12.29 – Структура регистра MSCRATCH

Поле	Биты	Доступ	Описание
MSCRATCH	31-0	RW	Используется для хранения указателя на локальное для ядра контекстное пространство

MEPC

MEPC представляет собой 32-разрядный регистр, доступный для чтения и записи, со структурой, приведенной в таблице 12.30. Младший бит MEPC (MEPC[0]) всегда равен нулю.

При срабатывании ловушки в режиме M-mode, MEPC записывается виртуальным адресом инструкции, которая столкнулась с исключением. В иных случаях MEPC не записывается, однако он может быть модифицирован программно с использованием инструкции для доступа к CSR.

Таблица 12.30 – Структура регистра MEPC

Поле	Биты	Доступ	Описание
RSV	0	RZ	Зарезервировано
MEPC	31-1	RW	Адрес инструкции, в которой возникло исключение

MCAUSE

Регистр MCAUSE представляет собой 32-разрядный регистр, доступный для чтения и записи, со структурой, показанной в таблице 12.31. При срабатывании ловушки в режиме M-mode, MCAUSE записывается кодом, указывающим на событие, вызвавшее ловушку.

В иных случаях MCAUSE не записывается, однако он может быть модифицирован программно с использованием инструкции для доступа к CSR.

Таблица 12.31 – Структура регистра MCAUSE

Поле	Биты	Доступ	Описание
EC	4-0	RW	Код исключения
RSV	30-5	RZ	Зарезервировано
INT	31	RW	Прерывание

Бит прерывания в регистре MCAUSE устанавливается, если сбой был вызван прерыванием. Поле Код исключения содержит код, идентифицирующий последнее исключение.

Список возможных кодов исключений MCAUSE приведен в таблице 12.32.

Таблица 12.32 – Список кодов исключений MCAUSE

Прерывание	Исключение	Описание
0	0	Адрес инструкции не выровнен
0	1	Ошибка доступа к инструкции
0	2	Недопустимая инструкция
0	3	Точка останова
0	4	Загружаемый адрес не выровнен
0	5	Ошибка доступа к загрузке
0	6	Адрес сохранения/AMO не выровнен
0	7	Ошибка доступа сохранения/AMO
0	8	Вызов среды из режима U-mode
0	9	Вызов среды из режима S-mode

Продолжение таблицы 12.32

Прерывание	Исключение	Описание
0	10	Зарезервировано
0	11	Вызов среды из режима M-mode
0	12	Ошибка страницы инструкции
0	13	Ошибка загрузки страницы
0	14	Ошибка пропуска страницы (misspage) TLB
0	15	Ошибка страницы сохранения/AMO
0	31-16	Зарезервировано
1	0	Зарезервировано
1	1	Программное прерывание супервизора
1	2	Зарезервировано
1	3	Программное прерывание машинное
1	4	Зарезервировано
1	5	Прерывание таймера супервизора
1	6	Зарезервировано
1	7	Прерывание таймера машинное
1	8	Зарезервировано
1	9	Внешнее прерывание супервизора
1	10	Зарезервировано
1	11	Внешнее прерывание машинное
1	31	Ошибка памяти

MTVAL

Регистр MTVAL представляет собой 32-разрядный регистр, доступный для чтения и записи. При срабатывании ловушки в режиме M-mode, в MTVAL записывается информация, относящаяся к конкретным исключениям, чтобы помочь программной обработке ловушки. В иных случаях MTVAL не записывается, однако он может быть модифицирован программно с использованием инструкции для доступа к CSR.

При срабатывании аппаратной точки останова или при возникновении ошибки, связанной с неправильным выравниванием адреса выборки инструкции, загрузки или сохранения, ошибке доступа или ошибке страницы, MTVAL записывается адресом, вызвавшим сбой. В случае срабатывания ловушки «недопустимая инструкция» MTVAL записывается первыми 32 битами инструкции, вызывающей ошибку. Для других исключений MTVAL записывается нулем.

MIP

Регистр MIP – это 32-разрядный регистр, доступный для чтения и записи, содержащий информацию об ожидающих прерываниях, в то время как MIE – это соответствующий 32-разрядный регистр, доступный для чтения и записи, содержащий биты разрешения прерывания.

Структура регистра MIP приведена в таблице 12.33.

Таблица 12.33 – Структура регистра MIP

Поле	Биты	Доступ	Описание
RSV	0	RZ	Зарезервировано
SSIP	1	RW	Ожидающее программное прерывание супервизора
RSV	2	RZ	Зарезервировано
MSIP	3	RO	Ожидающее программное прерывание машинное
RSV	4	RZ	Зарезервировано
STIP	5	RW	Ожидающее прерывание таймера супервизора
RSV	6	RZ	Зарезервировано
MTIP	7	RO	Ожидающее прерывание таймера машинное
RSV	8	RZ	Зарезервировано
SEIP	9	RWRS	Ожидающее внешнее прерывание супервизора
RSV	10	RZ	Зарезервировано
MEIP	11	RO	Ожидающее внешнее прерывание машинное
RSV	30-12	RZ	Зарезервировано
MEFIP	31	RW0C	Ожидающее прерывание из-за ошибки памяти

Только биты, соответствующие программному прерыванию с более низкими привилегиями (SSIP), прерыванию по таймеру (STIP) и внешним прерываниям (SEIP) в регистре MIP, доступны для записи через этот адрес CSR. Остальные биты доступны только для чтения.

Биты MTIP и STIP соответствуют битам ожидающих прерываний по таймеру для прерываний по таймеру машинного уровня и супервизора соответственно. Бит MTIP доступен только для чтения и очищается путем записи в регистр сравнения таймера режима M-mode, отображаемый в памяти. Биты STIP могут быть записаны в режиме M-mode программно, для передачи прерываний по таймеру на более низкие уровни привилегий. Программа супервизора может очистить бит STIP с помощью вызовов SEE.

Уровень привилегий супервизора имеет отдельный бит ожидающего прерывания SSIP, который может считываться и записываться с помощью CSR-доступа из кода, выполняющегося локально ядром с соответствующим или более высоким уровнем привилегий.

Поле SEIP в MIP содержит один бит доступный для чтения и записи. SEIP может быть записан программно из M-mode для указания S-mode на ожидаемое внешнее прерывание. Логическое ИЛИ программно записываемого бита и сигнал от внешнего контроллера прерываний используются для генерации внешних прерываний супервизора. Когда бит SEIP считывается с помощью инструкций CSRRW, CSRRS или CSRRC, значение, возвращаемое в регистр предназначенный для чтения, содержит логическое ИЛИ программно записываемого бита и сигнал прерывания от контроллера прерываний. Однако значение, используемое в последовательности чтения-изменения-записи команд CSRRS или CSRRC, является только программно записываемым битом SEIP, игнорируя значение прерывания от внешнего контроллера прерываний.

Бит MSIP машинного уровня записывается при обращении к регистрам ICCM, которые используются другим ядром для обеспечения межпроцессорных программных прерываний в машинном режиме. Каждое ядро имеет линию прерывания, подключенную к линии MSI (Machine Software Interrupt) от ICCM, что в конечном итоге может привести к записи бита MSIP принимающего ядра в машинном режиме. Ядро может записать свой собственный бит MSIP, используя тот же механизм для доступа к регистрам модуля ICCM. Для получения подробной информации обратитесь к подразделу «Модуль межъядерной связи (ICCM)».

Стандартные CSR-регистры триггеров для чтения/записи и отладки

Эти регистры являются CSR, доступными с помощью кодов операций RISC-V csr и, при необходимости, также с помощью абстрактных команд отладки.

Некоторые комбинации активированных функций могут быть неподдерживаемыми. Все регистры TDATA соответствуют семантике «writeany-read-legal». Если отладчик записывает неподдерживаемую конфигурацию, регистр считывает обратно поддерживаемое значение (которое может быть просто отключенным триггером). Это означает, что отладчик должен всегда считывать значения, которые он записывает в регистры TDATA, если только он уже не знает, что поддерживается. Запись в один регистр TDATA не может изменить содержимое других регистров TDATA, а также конфигурацию любого триггера, кроме того, который выбран в данный момент.

Триггерные регистры доступны только в машинном и отладочном режимах, чтобы предотвратить переход ненадежного пользовательского кода в режим отладки без разрешения операционной системы.

Карта CSR-регистров триггеров представлена в таблице 12.34.

Таблица 12.34 – Карта CSR-регистров триггеров

Адрес	Обозначение	Полное наименование
0x7A0	TSELECT	Trigger Select
0x7A1	TDATA1 / MCONTROL / ICOUNT	Trigger Data 1 / Match Control / Instruction Count
0x7A2	TDATA2	Trigger Data 2
0x7A3	–	Зарезервировано
0x7A4	TINFO	Trigger Info
0x7A5..0x7AF	–	Зарезервировано

TSELECT

Этот регистр определяет, какой триггер доступен через другие триггерные регистры. Набор доступных триггеров начинается с 0 и является непрерывным.

Запись значений, превышающих или равных количеству поддерживаемых триггеров, может привести к тому, что значение в этом регистре будет отличаться от того, которое было записано. Чтобы убедиться, что записанное значение является допустимым индексом, отладчики могут считывать значение и проверять, что TSELECT сохраняет то, что они записали.

Поскольку триггеры могут использоваться как в режиме отладки, так и в режиме M-mode, отладчик должен восстановить этот регистр, если он его изменяет.

Структура регистра TSELECT приведена в таблице 12.35.

Таблица 12.35 – Структура регистра TSELECT

Поле	Биты	Доступ	Описание
INDEX	2-0	RW	Это поле определяет, какой триггер доступен через другие триггерные регистры. Максимальный индекс, поддерживаемый в данном ядре, равен 4.
RSV	31-3	RZ	Зарезервировано

TDATA1

Структура регистра TDATA1 приведена в таблице 12.36.

Таблица 12.36 – Структура регистра TDATA1

Поле	Биты	Доступ	Описание
DATA	26-0	RW	Данные, относящиеся к конкретному триггеру
DMODE	27	RW	Кодировка: 0 – Как в режиме отладки, так и в режиме M-mode возможна запись в регистры TDATA при выбранном параметре. 1 – Только в режиме отладки возможна запись в регистры TDATA при выбранном параметре. Записи из других режимов игнорируются. Этот бит доступен для записи только из режима отладки.
TYPE	31-28	RW	Кодировка: 0 – Нет триггера. 2 – Триггер является триггером сопоставления адресов. Остальные биты в этом регистре действуют так, как описано в MCONTROL. 3 – Триггер является триггером подсчета команд. Остальные биты в этом регистре действуют так, как описано в ICOUNT. 15 – Этот триггер существует (нумерация не должна прерываться), но в данный момент недоступен. Остальные значения зарезервированы для использования в будущем.

MCONTROL

Данный регистр доступен как TDATA1, если в поле TYPE установлено значение 2. Структура регистра MCONTROL приведена в таблице 12.37.

Таблица 12.37 – Структура регистра MCONTROL

Поле	Биты	Доступ	Описание
LOAD	0	RW	Когда данный бит установлен, триггер срабатывает по виртуальному адресу загрузки
STORE	1	RW	Когда данный бит установлен, триггер срабатывает по виртуальному адресу сохранения
EXECUTE	2	RW	Когда данный бит установлен, триггер срабатывает по виртуальному адресу выполняемой команды
U	3	RW	Установка бита включает этот триггер в режиме U-mode
S	4	RW	Установка бита включает этот триггер в режиме S-mode
RSV	5	RZ	Зарезервировано
M	6	RW	Установка бита включает этот триггер в режиме M-mode

Продолжение таблицы 12.37

Поле	Биты	Доступ	Описание
MATCH	10-7	RW	Кодировка: 0 - Совпадение, когда значение равно TDATA2. 2 - Совпадение, когда значение больше (без знака) или равно TDATA2. Этот тип совпадения не поддерживается, если EXECUTE = 1. 3 - Совпадение, если значение меньше (беззнакового) TDATA2. Этот тип совпадения не поддерживается, если EXECUTE = 1. Другие значения зарезервированы
CHAIN	11	RW	Кодировка: 0 - Когда этот триггер совпадает, выполняется настроенное действие. 1 - Если этот триггер не совпадает, он предотвращает совпадение триггера со следующим индексом.
ACTION	15-12	RW	Действие, которое необходимо предпринять при срабатывании триггера. Кодировка: 0 - Вызывает исключение точки останова. (Используется, когда программное обеспечение хочет использовать модуль триггера без подключения внешнего отладчика.) 1 - Переход в режим отладки. (Поддерживается только в том случае, если DMODE триггера равен 1.) 2..5 - Зарезервированы для использования в соответствии со спецификацией трассировки. Другие - зарезервированы для использования в будущем.
SIZELO	17-16	RO	Данное поле аппаратно установлено в нулевое значение. Таким образом, триггер попытается выполнить сопоставление с доступом любого размера.
TIMING	18	RO	Данный бит аппаратно установлен в нулевое значение. Это означает, что действие для этого триггера будет выполнено непосредственно перед выполнением команды, которая его запустила, но после выполнения всех предыдущих инструкций.
SELECT	19	RO	Данный бит аппаратно установлен в нулевое значение. Это означает, что триггер выполняет сопоставление только с виртуальным адресом.
HIT	20	RW	Данный бит аппаратно устанавливается, когда данный триггер совпадает. Пользователь триггера может установить или отменить его в любое время. Он используется для определения того, какие триггеры совпадают.
RSV	26-21	RZ	Зарезервировано
DMODE	27	RW	Данный бит описывается как часть регистра TDATA1
TYPE	31-28	RW	Данное поле описывается как часть регистра TDATA1

ICOUNT

Данный регистр доступен как TDATA1, если в поле TYPE установлено значение 3.

Этот тип триггера предназначен для использования в качестве отдельного шага, который полезен как для внешних отладчиков, так и для программ мониторинга программного обеспечения. В этом случае поле COUNT должно быть равно 1.

Структура регистра ICOUNT приведена в таблице 12.38.

Таблица 12.38 – Структура регистра ICOUNT

Поле	Биты	Доступ	Описание
ACTION	5-0	RW	Действие, которое необходимо предпринять при срабатывании триггера. Кодировка: 0 - Вызывает исключение точки останова. (Используется, когда программное обеспечение хочет использовать модуль триггера без подключения внешнего отладчика.) 1 - Вход в режим отладки. (Поддерживается только в том случае, если бит DMODE триггера равен 1.) 2..5 - Зарезервированы для использования в спецификации трассировки. Другие - зарезервированы для использования в будущем
U	6	RW	Если данный бит установлен, каждая выполненная команда или исключение, возникающее в режиме U-mode, уменьшает COUNT на 1
S	7	RW	Если данный бит установлен, каждая выполненная команда или исключение, возникающее в режиме S-mode, уменьшает COUNT на 1
RSV	8	RZ	Зарезервировано
M	9	RW	Если данный бит установлен, каждая выполненная команда или исключение, возникающее в режиме M-mode, уменьшает COUNT на 1
COUNT	23-10	RW	Когда значение COUNT уменьшается до 0, срабатывает триггер
HIT	24	RW	Данный бит устанавливается аппаратно, при совпадении триггера. Пользователь триггера может установить или сбросить его в любое время. Он используется для определения того, какие триггеры совпадают.
RSV	26-25	RZ	Зарезервировано
DMODE	27	RW	Данный бит описывается как часть регистра TDATA1
TYPE	31-28	RW	Данное поле описывается как часть регистра TDATA1

TDATA2

Структура регистра TDATA2 приведена в таблице 12.39.

Таблица 12.39 – Структура регистра TDATA2

Поле	Биты	Доступ	Описание
DATA	31-0	RW	Это поле содержит данные, относящиеся к конкретному триггеру

TINFO

Структура регистра TINFO приведена в таблице 12.40.

Таблица 12.40 – Структура регистра TINFO

Поле	Биты	Доступ	Описание
INFO	15-0	RO	В данном битовом поле отображаются поддерживаемые типы для выбранного триггера: по одному биту для каждого возможного типа, перечисленного в TDATA1. Бит N соответствует типу N. Если бит установлен, то этот тип поддерживается выбранным в данный момент триггером. Если выбранный в данный момент триггер не существует, это поле содержит значение 1. Возможности триггеров в данном ядре распределены следующим образом: 0..3 - поддерживает только триггер типа = 2 (MCONTROL), INFO = 0x04; 4 - поддерживает триггер только типа = 3 (ICOUNT), INFO = 0x08; другие индексы указывают на INFO = 0x01.
RSV	31-16	RZ	Зарезервировано

CSR-регистры отладки

Данные CSR-регистры, доступные с помощью операционных кодов «csr» RISC-V и, при необходимости, с использованием абстрактных команд отладки.

Карта CSR-регистров отладки представлена в таблице 12.41.

Таблица 12.41 – Карта CSR-регистров отладки

Адрес	Обозначение	Полное наименование
0x7B0	DCSR	Debug Control and Status
0x7B1	DPC	Debug PC
0x7B2	DSCRATCH0	Debug Scratch Register 0
0x7B3..0x7BF	–	Зарезервировано

DCSR

Структура регистра DCSR приведена в таблице 12.42.

Таблица 12.42 – Структура регистра DCSR

Поле	Биты	Доступ	Описание
PRV	1-0	RW	Содержит уровень привилегий, на котором работало ядро при переходе в режим отладки. Кодировка: 0 – User/Application; 3 – Machine. Отладчик может изменить это значение, чтобы изменить уровень привилегий ядра при выходе из режима отладки. Если записанная кодировка не поддерживается или отладчику запрещено изменять ее, ядро может изменить уровень привилегий на любой поддерживаемый
STEP	2	RW	Если данный бит установлен и ядро не находится в режиме отладки, ядро выполнит только одну команду, а затем перейдет в режим отладки. Если команда не будет выполнена в случае исключения, ядро немедленно перейдет в режим отладки до выполнения обработчика ловушки с установленными соответствующими регистрами исключений. Отладчик не должен изменять значение этого бита во время работы ядра
RSV	3	RO	Зарезервировано
MPRVEN	4	RW	Кодировка: 0 – Бит MPRV регистра MSTATUS игнорируется в режиме отладки; 1 – Бит MPRV регистра MSTATUS активен в режиме отладки.
RSV	5	RO	Зарезервировано
CAUSE	8-6	RO	Поле содержит информацию, почему был введен режим отладки. Если существует несколько причин для перехода в режим отладки за один цикл, аппаратное обеспечение должно установить причину с наивысшим приоритетом. Кодировка: 1 – Была выполнена команда EBREAK (приоритет 3); 2 – Модуль триггера вызвал исключение точки останова (приоритет 4, наивысший); 3 – Отладчик запросил переход в режим отладки с использованием «haltreq» (приоритет 1); 4 - Однократный шаг ядра, поскольку был установлен шаг (приоритет 0, самый низкий); Остальные значения зарезервированы для использования в будущем
RSV	10-9	RO	Зарезервировано

Продолжение таблицы 12.42

Поле	Биты	Доступ	Описание
STEPIE	11	RW	Кодировка: 0 – Прерывания отключены при выполнении одиночного шага. 1 – Прерывания включены при выполнении одиночного шага. Отладчик не должен изменять значение этого бита во время работы ядра
EBREAKU	12	RW	Кодировка: 0 – Инструкции EBREAK в режиме U-mode работают так, как описано в спецификации привилегий. 1 – Инструкции EBREAK в режиме U-mode переходят в режим отладки
EBREAKS	13	RW	Кодировка: 0 – Инструкции EBREAK в режиме S-mode работают так, как описано в спецификации привилегий. 1 – Инструкции EBREAK в режиме S-mode переходят в режим отладки
RSV	14	RO	Зарезервировано
EBREAKM	15	RW	Кодировка: 0 – Инструкции EBREAK в режиме M-mode работают так, как описано в спецификации привилегий. 1 – Инструкции EBREAK в режиме M-mode переходят в режим отладки
RSV	27-16	RO	Зарезервировано
XDEBUGVER	31-28	RO	Значение поля (4) указывает на то, что существует поддержка отладки, как описано в спецификации отладки RISC-V

DPC

При переходе в режим отладки в DPC обновляется виртуальный адрес следующей команды, которая должна быть выполнена. Поведение более подробно описано в таблице 12.43.

Таблица 12.43 – Виртуальный адрес в DPC при входе в режим отладки

Причина	Виртуальный адрес в DPC
EBREAK	Адрес инструкции EBREAK
Single Step	Адрес инструкции, которая была бы выполнена следующей, если бы не запуск отладки. Т.е. pc + 4 для 32-разрядных инструкций, которые не изменяют ход выполнения программы, программный счетчик назначения при выполнении переходов / ответвлений и т.д.
Trigger Module	Адрес команды, которая вызвала срабатывание триггера (поскольку MCONTROL.TIMING всегда равен 0 в данном ядре).
Halt request	Адрес следующей команды, которая должна быть выполнена в момент перехода в режим отладки

При возобновлении работы программный счетчик ядра обновляется до виртуального адреса, сохраненного в DPC. Отладчик может записать DPC, чтобы изменить место возобновления работы ядра.

Структура регистра DPC приведена в таблице 12.44.

Таблица 12.44 – Структура регистра DPC

Поле	Биты	Доступ	Описание
DPC	31-0	RW	Это поле содержит значение программного счетчика отладки

DSCRATCH0

Структура регистра DSCRATCH0 приведена в таблице 12.45.

Таблица 12.45 – Структура регистра DSCRATCH0

Поле	Биты	Доступ	Описание
DATA	31-0	RW	Поле содержит данные, которые могут использоваться инструкциями, выполняемыми из программного буфера

MCYCLE/MCYCLEH

Подробная информация в подразделе «Аппаратные счетчики мониторинга производительности».

MTIME/MTIMEH

Регистры MTIME и MTIMEH являются теневыми регистрами регистров MTIMER_MTIMECMPLO и MTIMER_MTIMECMPHI.

Для дополнительной информации см. структуру регистров Mtimer в подразделе «Машинный таймер с отображением в памяти».

MINSTRET/MINSTRETH

Подробная информация в подразделе «Счетчики мониторинга аппаратной производительности».

Регистры MMU

Регистры MMU – это нестандартные регистры, которые не определены спецификацией RISC-V. Подробная информация в подразделе «Блок управления памятью (MMU)».

Регистры MPU

Регистры MPU – это нестандартные регистры, которые не определены спецификацией RISC-V. Подробная информация в подразделе «Блок защиты памяти (MPU)».

Регистр MEM_CTRL_GLOBAL

Глобальный регистр управления областями памяти (MEM_CTRL_GLOBAL) – это пользовательский регистр, который не определен спецификацией RISC-V.

Структура регистра MEM_CTRL_GLOBAL приведена в таблице 12.46.

Таблица 12.46 – Структура регистра MEM_CTRL_GLOBAL

Поле	Биты	Доступ	Описание
L1ICE	0	RW	Включение кэша инструкций L1
L1DCE	1	RW	Включение кэша данных L1
L1ICF	2	RW1S	Управление очисткой кэша инструкций L1. Запись единицы запускает процедуру очистки/аннулирования кэша. Чтение определяет статус процедуры: 0 - очистка/аннулирование кэша завершено; 1 - выполняется очистка/аннулирование кэша.
L1DCF	3	RW1S	Управление очисткой кэша данных L1. Чтение определяет статус процедуры: 0 - очистка/аннулирование кэша завершено; 1 - выполняется очистка/аннулирование кэша.
RSV	31-4	RZ	Зарезервировано

Значение регистра по сбросу – 0x0.

Регистры ICCM в режиме M-mode

Регистры модуля межъядерной связи (ICCM) являются нестандартными регистрами, которые не определены спецификацией RISC-V.

Подробная информация в подразделе «Модуль межъядерной связи (ICCM)»

MVENDORID

MVENDORID – это 32-разрядный регистр, доступный для чтения, содержащий идентификатор производителя JEDEC поставщика ядра.

Идентификаторы производителя JEDEC обычно кодируются как последовательность однобайтовых кодов расширения 0x7f, заканчивающихся однобайтовым идентификатором, не равным 0x7f, с нечетным битом четности в старшем бите каждого байта. MVENDORID кодирует количество однобайтовых кодов расширения в поле BANK и кодирует последний байт в поле OFFSET, отбрасывая бит четности.

Структура регистра MVENDORID приведена в таблице 12.47.

Таблица 12.47 – Структура регистра MVENDORID

Поле	Биты	Доступ	Описание
OFFSET	6-0	RO	Последний байт
BANK	31-7	RO	Количество однобайтовых кодов расширения в идентификаторе производителя JEDEC

Данный регистр в ядре аппаратно установлен в ноль.

MARCHID

MARCHID – это 32-разрядный регистр, доступный для чтения, кодирующий базовую микроархитектуру ядра. Архитектурные идентификаторы проектов с открытым исходным кодом распределяются по всему миру фондом RISC-V и имеют ненулевые архитектурные идентификаторы с нулевым старшим разрядом (MSB). Коммерческие идентификаторы архитектуры назначаются каждым коммерческим поставщиком независимо, но должны иметь установленный MSB и не могут содержать ноль в оставшихся 31 бите.

Текущее значение регистра MARCHID равно 0x93000104.

MIMPID

MIMPID – это 32-разрядный регистр, доступный только для чтения, который обеспечивает уникальную кодировку версии реализации ядра. Реализация отражает дизайн самого ядра, а не какой-либо окружающей системы.

Структура регистра MIMPID приведена в таблице 12.48.

Таблица 12.48 – Структура регистра MIMPID

Поле	Биты	Доступ	Описание
REL	7-0	RO	Шестнадцатеричное значение версии
Day	15-8	RO	Шестнадцатеричное значение дня
Mon	23-16	RO	Шестнадцатеричное значение месяца
Year	31-24	RO	Шестнадцатеричное значение года

MIMPID в ядре аппаратно установлен в 0x24082055.

MHARTID

MHARTID – это 32-разрядный регистр, доступный для чтения, который содержит целочисленный идентификатор аппаратного потока, выполняющего код.

MHARTID определяется внешним портом.

MEFADDR

MEFADDR – это нестандартный регистр, который не определен спецификацией RISC-V.

MEFADDR содержит адрес ошибки для асинхронного исключения "Ошибка памяти".

Структура регистра MEFADDR приведена в таблице 12.49.

Таблица 12.49 – Структура регистра MEFADDR

Поле	Биты	Доступ	Описание
Addr	31-0	RO	Адрес ошибки памяти

MEFCMD

MEFCMD – это нестандартный регистр, который не определен спецификацией RISC-V.

Содержит команду доступа к памяти, которая вызывает асинхронное исключение "Ошибка памяти".

Структура регистра MEFCMD приведена в таблице 12.50.

Таблица 12.50 – Структура регистра MEFCMD

Поле	Биты	Доступ	Описание
CMD	1-0	R0	Команда: 01 - запись, остальные значения зарезервированы
RSV	31-2	RZ	Зарезервировано

CACHE_L1_DESCR

Структура регистра CACHE_L1_DESCR приведена в таблице 12.51.

Таблица 12.51 – Структура регистра CACHE_L1_DESCR

Поле	Биты	Доступ	Значение по сбросу	Описание
L1I WAYS	2-0	RO	2	Поле содержит количество путей кэша инструкций L1. Рассчитывается как $2^{\text{L1I WAYS}}$
RSV	3	RO	0	Зарезервировано
L1I LINE	7-4	RO	4	Поле содержит размер строки кэша инструкций L1 (в байтах). Рассчитывается как $2^{\text{L1I LINE}}$
L1I WIDTH	12-8	RO	9	Размер индекса кэша инструкций L1в битах. Рассчитывается как $2^{\text{L1I WIDTH}}$
L1I TYPE	15-13	RO	0	Тип кэша инструкций L1: 0 – только чтение; 1 – сквозная запись, без выделения; 7 – обратная запись, распределение, включение.
L1D WAYS	18-16	RO	2	Поле содержит количество путей кэша данных L1. Рассчитывается как $2^{\text{L1D WAYS}}$
RSV	19	RO	0	Зарезервировано
L1D LINE	23-20	RO	4	Поле содержит размер строки кэша данных L1 (в байтах). Рассчитывается как $2^{\text{L1D LINE}}$
L1D WIDTH	28-24	RO	9	Размер индекса кэша данных L1в битах. Рассчитывается как $2^{\text{L1D WIDTH}}$
L1D TYPE	31-29	RO	1	Тип кэша данных L1: 0 – только чтение; 1 – сквозная запись, без выделения; 7 – обратная запись, распределение, включение.

Регистр описания кэша L1 – это нестандартный регистр, который не определен спецификацией RISC-V.

SWPW_R0_ADDR/SWPW_R1_ADDR

Существует 2 специальных регистра CSR, которые загружаются после исключения "Ошибка пропущенной страницы TLB". Эти регистры используются для сокращения обработки пропусков TLB.

Вычисление адреса SWPW_R0_ADDR: Содержит предварительно вычисленный адрес записи корневой таблицы страниц (PTE2_ADDR).

SWPW_R0_ADDR = (SATP.PPN << 12) + VADDR.VPN1 * 4

Смещение адреса SWPW_R1_ADDR: Содержит смещение PTE для базового адреса таблицы страниц второго уровня (PTE0_OFFS).

SWPW_R1_ADDR = VADDR.VPN0 * 4

12.7 Система мониторинга производительности ядра

Счетчики мониторинга аппаратной производительности

В ядре реализована система мониторинга производительности аппаратного обеспечения компьютера. В частности:

– 64-разрядный регистр MCYCLE содержит подсчет количества циклов, выполненных ядром с некоторого произвольного времени;

– 64-разрядный регистр MINSTRET содержит подсчет количества инструкций, которые были отменены ядром с некоторого произвольного времени;

– Монитор аппаратной производительности в машинном режиме (MHPM) состоит из четырех универсальных счетчиков событий MHPMCNT3...MHPMCNT6;

– Счетчики MHPM поддерживаются регистрами выбора событий MHPMEVENT3...MHPMEVENT6, обеспечивающими возможность выбора соответствующего события, которое приведет к инкрементированию соответствующего счетчика.

Событие 0 зарезервировано для обозначения "отсутствие события", в то время как значение других событий определяется платформой.

Чтение MCYCLEH, MINSTRET и MHPMCNT3...MHPMCNT6 возвращает 32 младших бита (31..0), в то время как чтение MCYCLEH, MINSTRETH и MHPMCNT3H...MHPMCNT6H возвращает 32 старших бита (63..32) соответствующих счетчиков.

Регистры включения счетчиков мониторинга аппаратной производительности

Регистры включения счетчика MCOUNTEREN [0x306] и SCOUNTEREN [0x106] управляют доступностью реализованных счетчиков мониторинга аппаратной производительности для следующего пониженного уровня привилегий.

Единая структура регистров MCOUNTEREN и SCOUNTEREN представлена в таблице 12.52.

Таблица 12.52 – Структура регистров MCOUNTEREN и SCOUNTEREN

Биты	Наименование	Описание
0	CY	1 – Включение CYCLE; 0 – Выключение CYCLE.
1	TM	1 – Включение TIME; 0 – Выключение TIME.
2	IR	1 – Включение INSTRET; 0 – Выключение INSTRET.
3	HPM3	1 – Включение HPMCNT3; 0 – Выключение HPMCNT3.
…	…	…
6	HPM6	1 – Включение HPMCNT6; 0 – Выключение HPMCNT6.
31-7	RSV	Зарезервировано

Когда биты CY, TM, IR или HPM3...HPM6 в регистре MCOUNTEREN очищены, попытки считывания регистров CYCLE, TIME, INSTRET или HPMCNT3...HPMCNT6 при работе в режимах S-mode или U-mode приведут к исключению «недопустимая инструкция».

Когда один из этих битов установлен в MCOUNTEREN, доступ к соответствующему регистру разрешен при выполнении в режиме S-mode.

Когда один из этих битов установлен в SCOUNTEREN, доступ к соответствующему регистру разрешен при выполнении в режиме U-mode.

Регистры CYCLE, INSTRET и HPMCNT3...HPMCNT6 являются доступными только для чтения теневыми регистрами MCYCLE, MINSTRET и MHPMCNT3...MHPMCNT6, соответственно.

Регистр TIME является доступным только для чтения теневым регистром отображенного в памяти регистра MTIME.

Регистр ограничения машинного таймера

Регистр ограничения MCOUNTINHIBIT – это регистр WARL, который определяет, какой из счетчиков мониторинга аппаратной производительности инкрементируется. Настройки в этом регистре определяют только то, инкрементируются ли значения счетчиков; настройки этого регистра не влияют на их доступность. Когда биты CY, IR или HPMn в регистре MCOUNTINHIBIT очищены, регистры CYCLE, INSTRET или HPMCNTn инкрементируются, как обычно. Когда установлены биты CY, IR или HPMn, соответствующий счетчик не инкрементируется.

Структура регистра MCOUNTINHIBIT приведена в таблице 12.53.

Таблица 12.53 – Структура регистра MCOUNTINHIBIT

Поле	Биты	Значение по сбросу	Описание
CY	0	0	1 – Включение инкрементирования CYCLE; 0 – Выключение инкрементирования CYCLE.
RSV	1	0	Зарезервировано
IR	2	0	1 – Включение инкрементирования INSTRET; 0 – Выключение инкрементирования INSTRET.
HPM3	3	0	1 – Включение инкрементирования HPMCNT3; 0 – Выключение инкрементирования HPMCNT3.
…	…	…	…
HPM6	6	0	1 – Включение инкрементирования HPMCNT6; 0 – Выключение инкрементирования HPMCNT6.
RSV	31-7	0	Зарезервировано

Регистры событий счетчиков мониторинга аппаратной производительности

Реализованные счетчики HPMCNT3...HPMCNT6 могут использоваться для подсчета одного из предопределенных событий, при этом событие выбирается через двухступенчатый мультиплексор с возможностью дополнительной фильтрации, такой как фильтр привилегированного режима M/S/U и/или фильтр ASID супервизора для конвейерных событий (подробнее об ASID смотрите в регистре SATP).

Единая структура регистров MHPMEVENT3...MHPMEVENT6 представлена в таблице 12.54.

Таблица 12.54 – Структура регистров MHPMEVENT3...MHPMEVENT6

Поле	Биты	Описание
GRP	3-0	Селектор группы для событий конвейера
SEL	8-4	Селектор событий для конвейерных событий
RSV	18-9	Зарезервировано
ASFILT	27-19	Значение ASID для событий конвейера
ASF_EN	28	1 – Включить ASID-фильтр супервизора; 0 – Отключить ASID-фильтр супервизора.
MFILT	30-29	Значение привилегий M/S/U-mode
MF_EN	31	1 – Включить фильтр привилегированного режима; 0 – Отключить фильтр привилегированного режима.

Двухступенчатый мультиплексор событий обеспечивает:
– Выбор группы событий по полю GRP;
– Выбор события внутри группы по полю SEL.

В таблице 12.55 приведен список групп событий, одна из которых может быть выбрана путем соответствующей настройки поля GRP в регистре событий MHPMEVENT.

Таблица 12.55 – Список групп событий

Значение в поле GRP	Наименование группы	Описание
0	GRP_DIS	Счетчик HPM отключен
1	GRP_GEN	Общая группа
2	GRP_EXC	Группа исключений
3	GRP_IRQ	Группа прерываний
4	GRP_PRD	Прогнозирующая группа
5	GRP_STAL	Группа остановов конвейера
6	GRP_L1	Группа кэша L1
15-7	RSV	Зарезервировано

События, доступные для выбора в каждой группе, показаны в таблицах 12.56 – 12.61.

Группа событий GRP_GEN

Эта группа (за исключением GEN_CYC и GEN_IDLE) подсчитывает количество выбранных событий. GEN_CYC подсчитывает циклы. GEN_IDLE подсчитывает циклы, когда ядро находится в состоянии IDLE.

Таблица 12.56 – События GRP_GEN

Поле селектора	Наименование события	Описание
0	GEN_CYC	Циклы
1	RSV	Зарезервировано
2	GEN_INST	Удаленные инструкции
3	GEN_IDLE	Ядро в состоянии IDLE
8-4	RSV	Зарезервировано
9	GEN_TRAP_S	Переключение в режим S-mode с помощью ловушки (исключение или IRQ)
10	RSV	Зарезервировано
11	GEN_TRAP_M	Переключение в режим M-mode с помощью ловушки (исключение или IRQ)
15-12	RSV	Зарезервировано
16	GEN_xRET_U	Переключение в режим U-mode с помощью MRET/SRET
17	GEN_xRET_S	Переключение в режим S-mode с помощью MRET/SRET
18	RSV	Зарезервировано
19	GEN_xRET_M	Переключение в режим M-mode с помощью MRET
31-20	RSV	Зарезервировано

Группа событий GRP_EXC

Эта группа подсчитывает количество выбранных исключений.

Таблица 12.57 – События GRP_EXC

Поле селектора	Наименование события	Описание
0	RSV	Зарезервировано
1	EXC_IAF	Ошибка доступа к инструкции
4-2	RSV	Зарезервировано
5	EXC_LAF	Ошибка доступа к загрузке
6	RSV	Зарезервировано
7	EXC_SAF	Ошибка доступа к хранилищу
11-8	RSV	Зарезервировано
12	EXC_IPF	Ошибка страницы инструкции
13	EXC_LPF	Ошибка загрузки страницы
14	RSV	Зарезервировано
15	EXC_SPF	Ошибка страницы хранилища (store/AMO)
31-16	RSV	Зарезервировано

Группа событий GRP_IRQ

Эта группа подсчитывает количество выбранных прерываний.

Таблица 12.58 – События GRP_IRQ

Поле селектора	Наименование события	Описание
0	RSV	Зарезервировано
1	IRQ_SSI	Программное прерывание супервизора
2	RSV	Зарезервировано
3	IRQ_MSI	Программное прерывание машинное
4	RSV	Зарезервировано
5	IRQ_STI	Прерывание таймера супервизора
6	RSV	Зарезервировано
7	IRQ_MTI	Прерывание таймера машинное
8	RSV	Зарезервировано
9	IRQ_SEI	Внешнее прерывание супервизора
10	RSV	Зарезервировано
11	IRQ_MEI	Внешнее прерывание машинное
30-12	RSV	Зарезервировано
31	IRQ_MMEFI	Прерывание по ошибке памяти в машинном режиме

Группа событий GRP_PRD

Эта группа подсчитывает количество выбранных событий прогнозирования.

Таблица 12.59 – События GRP_PRD

Поле селектора	Наименование события	Описание
0	CFU	Подсчет инструкций управления потоком, выполняемых в CFU (блоке управления потоком). Инструкции включают CBR (условную ветвь) и безусловные переходы JL/JLR.
15-1	RSV	Зарезервировано
16	NPT	CFU, предсказание не получено. Счетчик всех инструкций CFU, которые не осуществили переход (только CBR без ветвления, так как JL/JLR всегда выполняют ветвление)
17	PT	CFU, предсказание получено. Подсчет всех инструкций CFU, которые выполнили ветвление (все инструкции CBR с выполненным ветвлением и плюс все JL/JLR)
18	NPT_WP	CFU, неверное предсказание, предсказание не получено; Подсчет ошибочного предсказания, отсутствия ветвления, в CFU-модуле (неправильный предсказанный адрес или сам факт отсутствия ветвления)

Продолжение таблицы 12.59

Поле селектора	Наименование события	Описание
19	PT_WP	CFU, неверное предсказание, предсказание получено; Подсчет ошибочно предсказанных ответвлений, выполненных в модуле CFU (неверно предсказанный адрес или сам факт ответвления)
20	CBR_NPT	CBR (условная ветвь), предсказание не получено; Подсчет количества инструкций условного перехода, которые не выполнили передачу управления
21	CBR_PT	CBR, предсказание получено; Подсчет количества инструкций условного перехода, которые выполнили передачу управления
22	CBR_NPT_WP	CBR, неверное предсказание, предсказание не получено; Подсчет количества ошибочно предсказанных инструкций условного перехода, которые не выполнили передачу управления
23	CBR_PT_WP	CBR, неверное предсказание, предсказание получено; Подсчет количества ошибочно предсказанных инструкций условного перехода, которые выполнили передачу управления
24	BTB_NPT	BTB попадание, CFU, предсказание не получено
25	BTB_PT	BTB попадание, CFU, предсказание получено
26	BTB_NPT_WP	BTB попадание, CFU, неверное предсказание, предсказание не получено
27	BTB_PT_WP	BTB попадание, CFU, неверное предсказание, предсказание получено
28	BTB_CBR_NPT	BTB попадание, CBR, предсказание не получено
29	BTB_CBR_PT	BTB попадание, CBR, предсказание получено
30	BTB_CBR_NPT_WP	BTB попадание, CBR, неверное предсказание, предсказание не получено
31	BTB_CBR_PT_WP	BTB попадание, CBR, неверное предсказание, предсказание получено

Группа событий GRP_STAL

Эта группа подсчитывает циклы выбранных событий.

Таблица 12.60 – События GRP_STAL

Поле селектора	Наименование события	Описание
0	STAL_IDU_VD	Подсчитывает, сколько тактов на выходе IDU содержалась декодированная инструкция (независимо от наличия опасностей).
1	STAL_IDU_ISSU	Количество инструкций, переданных в конвейер
2	STAL_IDU_NISSU	Инструкция действительна, но не выдана; количество тактов, в течение которых декодированная команда находилась на выходе IDU и не была передана на выполнение (эквивалентно STAL_IDU_VD - STAL_IDU_ISSU)

Продолжение таблицы 12.60

Поле селектора	Наименование события	Описание
3	STAL_EXENR	Инструкция на выходе IDU корректна, исполнительные модули не готовы к получению инструкции
4	STAL_WBNR	Инструкция на выходе IDU корректна, но буфер WB заполнен
5	STAL_SYNC	Инструкция на выходе IDU корректна, ожидание синхронизации конвейера (выполняется синхронизация конвейера из-за выборки инструкций, операции с некоторыми CSR)
6	STAL_HZRD	Инструкция на выходе IDU корректна, но существует опасность потери данных
16-7	RSV	Зарезервировано
17	STAL_MDU_NR	Инструкция на выходе IDU корректна, инструкция требовала доступа к MDU, но MDU занят
18	STAL_LSU_NR	Инструкция на выходе IDU корректна, инструкция требовала доступа к LSU, но LSU занят
19	STAL_FPU_NR	Инструкция на выходе IDU корректна, инструкция требовала доступа к FPU, но FPU занят
31-20	RSV	Зарезервировано

Группа событий GRP_L1

Эта группа подсчитывает количество выбранных событий.

Таблица 12.61 – События GRP_L1

Поле селектора	Наименование события	Описание
0	L1I_HIT	Попадание в кэш инструкций L1
1	L1I_MISS	Промах в кэш инструкций L1
2	L1I_REF	Перезаполнение кэша инструкций L1
15-3	RSV	Зарезервировано
16	L1D_HIT	Попадание в кэш данных L1
17	L1D_MISS	Промах в кэш данных L1
18	L1D_REF	Перезаполнение кэша данных L1
31-19	RSV	Зарезервировано

12.8 Управление доступом к памяти

Модуль защиты памяти (MPU – Memory Protection Unit)

Модуль защиты памяти (MPU) определяет права доступа к физической памяти. Эти привилегии проверяются при доступе к памяти как для инструкций, так и для данных.

MPU состоит из 16 записей. Каждая запись определяет область в физическом адресном пространстве с определенными атрибутами доступа. Каждая запись реализует выделенные привилегии физической памяти (чтение, запись, выполнение) для каждого режима (M-mode/S-mode/U-mode) и один атрибут типа памяти (WO/SO, Кэшируемый/Некэшируемый).

Каждая запись управляется тремя выделенными регистрами конфигурации. Доступ к этим регистрам осуществляется через четыре CSR с одним CSR "выбор записи" и тремя общими CSR управления/маски/шаблона.

Регистр MPU_SELECT

Структура регистра MPU_SELECT показана в таблице 12.62.

Таблица 12.62 – Структура регистра MPU_SELECT

Поле	Биты	Доступ	Значение по сбросу	Описание
ENTRY	3-0	RW	0	Выбранная запись: 0..15
RSV	31-4	RZ	0	Зарезервировано

Регистр MPU_CONTROL

Структура регистра MPU_CONTROL показана в таблице 12.63.

Значение RSTVAL для битов [3..0] регистра MPU_CONTROL после сброса зависит от поля ENTRY регистра MPU_SELECT.

Если ENTRY = 0, тогда RSTVAL = 1;

Если ENTRY ≠ 0, тогда RSTVAL = 0.

Формальное определение RSTVAL таково: RSTVAL = (ENTRY == 0)?(1):(0).

Таблица 12.63 – Структура регистра MPU_CONTROL

Поле	Биты	Доступ	Значение по сбросу	Описание
VALID	0	RW	RSTVAL	Текущая запись включена
MMR	1	RW	RSTVAL	Чтения в режиме M-mode разрешены
MMW	2	RW	RSTVAL	Записи в режиме M-mode разрешены
MMX	3	RW	RSTVAL	Выполнение в режиме M-mode разрешено
UMR	4	RW	0	Чтения в режиме U-mode разрешены
UMW	5	RW	0	Записи в режиме U-mode разрешены
UMX	6	RW	0	Выполнение в режиме U-mode разрешено
SMR	7	RW	0	Чтения в режиме S-mode разрешены
SMW	8	RW	0	Записи в режиме S-mode разрешены
SMX	9	RW	0	Выполнение в режиме S-mode разрешено
RSV	15-10	RZ	0	Зарезервировано

Продолжение таблицы 12.63

Поле	Биты	Доступ	Значение по сбросу	Описание
MTYPE	17-16	RW	RSTVAL	Тип памяти: 0 - кэшируемая, слабо упорядоченная; 1 - не кэшируемая, строго упорядоченная; 2 - не кэшируемая, слабо упорядоченная; 3 - Конфигурация CPU для ввода-вывода с отображением в память, не кэшируемая, строго упорядоченная
RSV	30-18	RZ	0	Зарезервировано
LOCK	31	RW	0	Текущая запись заблокирована. Если поле "LOCK" активно, любые записи в управляющие или связанные с ними регистры адресов/масок игнорируются. Запись может быть разблокирована только при сбросе.

Регистр MPU_ADDRESS

Структура регистра MPU_ADDRESS показана в таблице 12.64.

Поле ADDR регистра MPU_ADDRESS после сброса зависит от поля ENTRY регистра MPU_SELECT.

Если ENTRY = 0, тогда ADDRVAL= 1;

Если ENTRY ≠ 0, тогда ADDRVAL не определен.

Таблица 12.64 – Структура регистра MPU_ADDRESS

Поле	Биты	Доступ	Значение по сбросу	Описание
RSV	9-0	RZ	0	Зарезервировано
ADDR	29-10	RW	ADDRVAL	Адрес [31:12]
RSV	31-30	RZ	0	Зарезервировано

Регистр MPU_MASK

Структура регистра MPU_MASK показана в таблице 12.65.

Поле MASK регистра MPU_MASK после сброса зависит от поля ENTRY регистра MPU_SELECT.

Если ENTRY = 0, тогда MASKVAL = 1;

Если ENTRY ≠ 0, тогда MASKVAL не определен.

Таблица 12.65 – Структура регистра MPU_MASK

Поле	Биты	Доступ	Значение по сбросу	Описание
RSV	9-0	RZ	0	Зарезервировано
MASK	29-10	RW	MASKVAL	Маска [31:12]
RSV	31-30	RZ	0	Зарезервировано

Функциональное описание MPU

Регистр MPU_MASK определяет размер соответствующей записи в диапазоне 4 КБ..16 ГБ.

Считается, что транзакция с памятью попала в запись MPU, если маскированный адрес транзакции становится равным маскированному адресу MPU.

Поскольку минимальный размер записей MPU составляет 4 КБ, страницы памяти MMU объемом 4 МБ могут содержать несколько записей MPU.

Допустимо (но не рекомендуется) настраивать несколько записей MPU с разными атрибутами памяти и привилегиями на одной странице MMU.

Настройки MPU для результирующего физического адреса могут быть проверены в любой момент между преобразованием адреса и явным доступом к виртуальной памяти. Следовательно, когда настройки MPU изменяются таким образом, что это влияет либо на физическую память, в которой хранятся таблицы страниц, либо на физическую память, на которую указывают таблицы страниц, программа в режиме M-mode должна синхронизировать настройки MPU с системой виртуальной памяти. Это достигается путем выполнения инструкции SFENCE.VMA с rs1=x0 и rs2=x0 после записи CSR MPU.

Ядро не поддерживает несогласованный доступ к инструкциям/данным, т.е. незавершенная транзакция с памятью не может быть частично разложена на несколько записей MPU.

Для корректной работы MPU при настройке регионов необходимо соблюдать следующие требования:

– Размер области должен быть кратен степени двойки (2^n).

– Минимальный размер области составляет 4 КБ.

– Базовый адрес области должен быть выровнен по размеру области.

Области ввода-вывода, отображенной в памяти конфигурации ядра

Кластер ядра дополнительно включает область, отображаемую во внутренней памяти (например, Mtimer). Чтобы получить доступ к этой области, настройте запись с MTYPE=3 (ввод-вывод, отображаемый в памяти, с конфигурацией ядра, не кэшируемый, строго упорядоченный).

С помощью MTYPE=3 можно настроить только одну запись MPU.

Создание области ввода-вывода, отображенной в памяти конфигурации ядра (MTYPE=3), с правами на выполнение (execute permission set) запрещено для любого режима привилегий, попытка записать такое значение в MPU_CONTROL CSR будет проигнорирована.

Программа проверки невыполненных транзакций

Доступ инструкции передается в подсистему памяти, если выполняются следующие условия:

– Существует по крайней мере одна доступная запись MPU, указывающая адрес транзакции, с активным полем MX для соответствующего режима;

– Нет доступных записей MPU, охватывающих адрес транзакции, с неактивным полем MX для соответствующего режима.

Неинвазивный (загрузочный) доступ к данным передается в подсистему памяти, если выполняются следующие условия:

– Существует по крайней мере одна доступная запись MPU, указывающая адрес транзакции, с активным полем MR для соответствующего режима;

– Нет доступных записей MPU, содержащих адрес транзакции, с неактивным полем MR для соответствующего режима.

Инвазивный (store/AMO) доступ к данным передается в подсистему памяти, если выполняются следующие условия:

– Существует по крайней мере одна доступная запись MPU, указывающая адрес транзакции, с активным полем MW для соответствующего режима;

– Нет доступных записей MPU, содержащих адрес транзакции, с неактивным полем MW для соответствующего режима.

Атрибуту памяти для любого типа доступа присваивается значение «слабо упорядоченный», если выполняются следующие условия:

– Существует по крайней мере одна доступная запись MPU, указывающая адрес транзакции, с полем MTYPE, установленным в значение «слабо упорядоченный»;

– Нет доступных записей MPU, содержащих адрес транзакции, с полем MTYPE установленным в значение «строго упорядоченный».

12.9 Кэш первого уровня (L1)

Ядро SCR5 содержит отдельные 4-канальные множественно-ассоциативные кэши команд и данных первого уровня (L1) размером 32 Кбайт, который обеспечивает исполнительным модулям и регистрам быстрый доступ к памяти команд и данных.

Данная реализация кэша уровня L1 имеет следующие характеристики:

- кэш разделен на кэш команд (32 Кб) и кэш данных (32 Кб) (Гарвардская архитектура);
- 4-канальная множественно-ассоциативная организация;
- кэш использует для формирования индексов и тегов физический адрес (PIPT);
- кэш инструкций и кэш данных имеют размер кэш-линии по 16 байт;
- каждый кэш может быть независимо друг от друга очищен с помощью управляющих битов L1ICE/L1DCE, расположенных в регистре MEM_CTRL_GLOBAL;
- в качестве алгоритма вытеснения кэш-линии используется алгоритм LRU (Least-recently-used, вытеснение из кэша значений, которые дольше всего не запрашивались).

Взаимодействие кэша команд и кэша данных с ядром показано на рисунке 12.3.

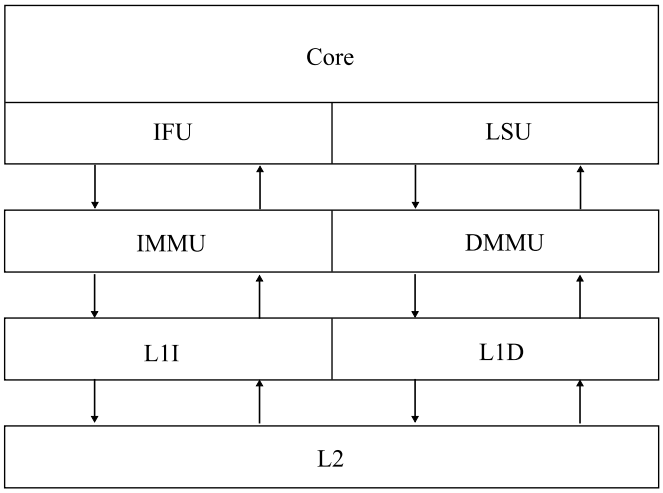


Рисунок 12.3 – Блок-схема кэша уровня L1

Модуль операций чтения/записи (Load Store Unit, LSU)

Модуль операций чтения/записи (Load Store Unit, LSU) выполняет:

- расчёт виртуального адреса для операций чтения/записи (load/store);

- проверку выравнивания адреса. Адрес, по которому выполняется операция, требующая доступа к памяти, должен быть выровнен по ширине доступа к памяти;
- проверку запроса в память в модуле управления памятью (MMU) для операций чтения/записи (load/store);
- извлечение запрошенных данных из прочитанного слова памяти.

Модуль выборки команд (Instruction Fetch Unit, IFU)

Модуль выборки команд (Instruction Fetch Unit, IFU) выполняет:
- генерацию виртуального адреса обращения к памяти команд. Адрес всегда выравнивается на границу 64-битового слова;
- проверку запроса в память в модуле MMU для операций выборки команды;
- буферизацию прочитанных слов команд;
- извлечение команды из прочитанного слова команд.

Модуль управления памятью (Memory Management Unit, MMU)

Модуль управления памятью (Memory Management Unit, MMU) выполняет:
- трансляцию виртуального адреса в физический;
- проверку прав доступа к памяти и, при нарушении прав доступа, генерацию соответствующего исключения «ошибка доступа к памяти».

Структура кэша уровня L1

Кэш данных уровня L1 организован в виде 512 наборов из четырех блоков по 16 байт в каждой строке.

Кэш команд уровня L1 организован в виде 512 наборов из четырех блоков по 16 байт в каждой строке.

Организация кэшей данных и команд L1 показана на рисунке 12.4.

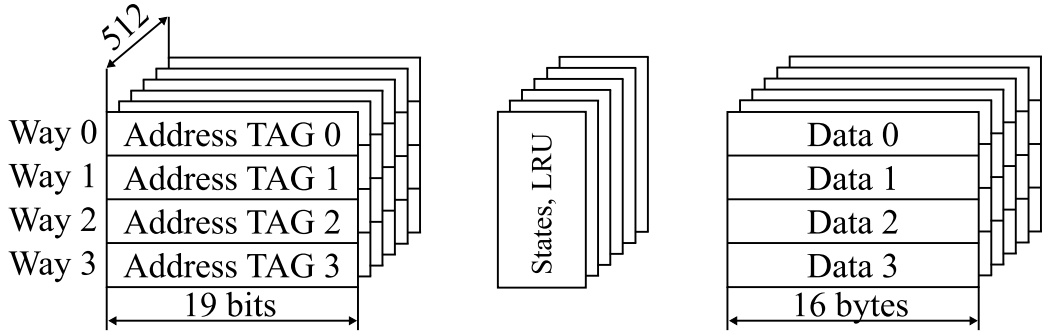


Рисунок 12.4 – Структура кэша данных уровня L1

Организация кэша данных уровня L1

Каждая кэш-линия содержит:
- 16 байт данных;
- 1 бит состояния (V);
- 19-битный адрес тега.

Кроме того, хотя это не показано на рисунке 12.4, кэш-линия данных содержит ECC биты для полей данных, тэга адреса и статуса. Каждая кэш-линия содержит 16 байтов, последовательно расположенных в памяти, и загружается из подсистемы памяти по адресу, выровненному по границе 16 байтов (другими словами, биты [3:0] физического адреса всегда равны нулю). Кэш-линии также выровнены по границе страниц. Биты [12:4] физического адреса (PA[12:4]) представляют собой индекс для выбора набора кэш-линий

(набор кэш-линий – это совокупность кэш-линий из разных каналов, соответствующих данному индексу). Тег адреса состоит из битов [31:13] физического адреса (PA [31:13]).

Организация кэша команд уровня L1

Каждая кэш-линия содержит:
- 16 байт данных команды;
- 1 бит состояния (V);
- 19-битный адрес тега.

Кроме того, хотя это не показано на рисунке 12.4, кэш-линия инструкций содержит ECC биты для полей данных, тэга адреса и статуса. Как и в случае с кэшем данных, каждая кэш-линия кэша команд содержит 16 байтов, последовательно расположенных в памяти, и загружается из подсистемы памяти по адресу, выровненному на границу 16 байтов (другими словами, биты [3:0] в физическом адресе всегда равны нулю). Кэш-линии также выровнены на границу 2-Килобайтных страниц. Также биты [12:4] в физическом адресе (PA[12:4]) представляют собой индекс для выбора набора кэш-линий. Тег адреса состоит из битов [31:13] физического адреса (PA [31:13]).

Управление кэшем уровня L1

Кэши команд и данных могут быть включены или выключены с помощью битов включения кэша MEM_CTRL_GLOBAL[L1ICE] и MEM_CTRL_GLOBAL[L1DCE] соответственно. Если кэш команд или кэш данных выключен, операции доступа к соответствующему кэшу не выполняются. В этом случае все операции передаются непосредственно в кэш уровня L2.

После выполнения процедуры сброса ядра биты MEM_CTRL_GLOBAL[L1ICE] и MEM_CTRL_GLOBAL[L1DCE] установлены в 0 и кэши выключены.

Установка в битах MEM_CTRL_GLOBAL[L1ICE] или MEM_CTRL_GLOBAL[L1DCE] значения 1 включает соответствующий кэш уровня L1. Далее кэш выполняет очистку, во время которой кэшируемые операции не смогут получить доступ к кэшу. Операции с атрибутом «некэшируемая операция» будут переданы непосредственно в кэш уровня L2.

Кэш команд и кэш данных могут быть очищены независимо друг от друга путем установки необходимых значений в биты MEM_CTRL_GLOBAL[L1ICF] и MEM_CTRL_GLOBAL[L1DCF] соответственно.

Поскольку кэш команд не содержит модифицированных данных, то его очистка не требует обновления данных в подсистеме памяти. Поскольку кэш данных обеспечивает политику сквозной записи (write-through), то его очистка не требует обновления данных в подсистеме памяти.

Биты активных кэш-линий автоматически сбрасываются при выполнении процедуры сброса. Если во время работы ядра программному обеспечению необходимо очистить все активные кэш-линии в кэшах, то оно может использовать соответствующие биты MEM_CTRL_GLOBAL[L1ICF] и MEM_CTRL_GLOBAL[L1DCF]. Установка этих битов приводит к выполнению операции очистки соответствующего кэша. После окончания процедуры очистки соответствующие биты MEM_CTRL_GLOBAL[L1ICF] и MEM_CTRL_GLOBAL[L1DCF] сбрасываются в 0 автоматически. Операции очистки кэшей являются локальными только для процессора, который их выполняет. Кэши уровня L1 других процессоров при этом не затрагиваются. Для определения окончания процедуры очистки кэшей программное обеспечение должно проверять соответствующие значения битов MEM_CTRL_GLOBAL[L1ICF] и MEM_CTRL_GLOBAL[L1DCF].

12.10 Операции кэша уровня L1

Заполнение кэша данных

При детектировании промахов в кэше, во время операций чтения, кэш осуществляет загрузку кэш-линии из кэша уровня L2. Если кэш данных выключен (MEM_CTRL_GLOBAL[L1DCE] = 0), все операции обращения с атрибутом «кэшируемая операция» передаются непосредственно в кэш уровня L2, минуя обращение к кэшу данных. В этом случае возвращаемые данные передаются в ядро и не загружаются непосредственно в кэш. Обратите внимание, что операции с атрибутом «некэшируемая операция» не осуществляет доступ к содержимому кэшей.

Заполнение кэша команд

Кэш команд обеспечивает 64-битовый интерфейс для модуля выборки команд (IFU). Таким образом модулю IFU могут быть доступны до четырех команд за один тактовый цикл при попадании в кэш команд. Команды, при попадании в кэш, передаются непосредственно из кэша команд в модуль выборки команд (IFU).

В случае детектирования промаха в кэше команд в нем резервируется кэш-линия и запрос выборки команды направляется в кэш уровня L2. При попадании в кэш уровня L2 данные из него записываются в зарезервированную кэш-линию кэша команд. При получении всех данных запрошенной кэш-линии запрошенное слово команд передается в модуль выборки команд (IFU), а сама кэш-линия записывается в кэш команд уровня L1. В случае детектирования промаха в кэше уровня L2 кэшируемая операция передаётся в подсистему памяти. При получении запрошенной кэш-линии запрошенное слово команд передается в модуль выборки команд (IFU), а сама кэш-линия записывается в кэш команд. В этом случае в кэш уровня L2 передаются данные, полученные из интерфейса памяти, и записываются кэш-линии. Если кэш команд выключен (MEM_CTRL_GLOBAL[L1ICE] = 0), все операции обращения с атрибутом «кэшируемая операция» передаются в кэш уровня L2, минуя обращение к кэшу команд уровня L1. В этом случае возвращаемые данные передаются в ядро и не загружаются непосредственно в кэш команд уровня L1.

Алгоритм выделения кэш-линий при промахах в кэшах

Промахи в кэше команд приводят к резервированию в нем новой линии с использованием алгоритма вытеснения LRU при условии, что этот кэш включен. Если был продетектирован промах в кэше данных при выполнении операции чтения (load) и если при этом для этой кэш-линии не было выполнено резервирование в кэше данных в результате предыдущей операции обращения в эту же кэш-линию, которая так же завершилась кэш промахом, то происходит резервирование новой кэш-линии с использованием алгоритма вытеснения LRU при условии, что этот кэш включён. При детектировании промаха в кэше данных при выполнении операции записи (store) новая кэш-линия не резервируется.

Алгоритм выбора канала для резервирования кэш-линии в кэшах уровня L1

Когда новая кэш-линия должна быть размещена в кэше команд или данных, осуществляется выбор линии в наборе, в которой будет размещаться эта кэш-линия. Если в наборе кэш-линий присутствуют неактивные кэш-линии, то в этом случае для резервирования выбирается неактивная линия с наименьшим номером. Если в наборе кэш-линий нет неактивных линий, тогда в этом случае выбирается одна из активных линий с использованием алгоритма вытеснения LRU (least-recently-used).

Резервирование кэш-линии осуществляется при детектировании промаха в кэше, а загрузка новой кэш-линии осуществляется после окончания выполнения операции чтения кэш-линии. При обращении к кэш-линии она помечается как самая недавно используемая кэш-линия в наборе. При детектировании промаха в кэше в случае, когда все линии в наборе

кэш-линий доступны, для резервирования выделяется кэш-линия, к которой дольше всего не было обращений. Биты, используемые алгоритмом вытеснения LRU, обновляются каждый раз при детектировании попадания в кэш.

Когерентность кэша

Кэш уровня L1 в многопроцессорной системе поддерживает когерентность. Каждый кэш уровня L1 поддерживает интерфейс отслеживания когерентности (snoop interface), каждый запрос отслеживания (snoop request) деактивирует указанную в запросе кэш-линию. Кэш уровня L2 использует интерфейс отслеживания когерентности (snoop interface) для обеспечения синхронности с кэшами более низкого уровня. При вытеснении кэш-линии из кэша уровня L2 или при выполнении операции, изменяющей содержимое в кэше уровня L2, кэш уровня L2 отправляет запрос отслеживания (snoop request) другим ядрам.

Все локальные операции доступа к памяти данных, выполняющие изменение содержимого (операции записи (store) и атомарные операции (atomic)), выполняющиеся в кэше уровня L1, всегда передаются в кэш уровня L2 после обработки (политика сквозной записи (writethrough)).

Локальные операции доступа к памяти данных, выполняющие изменение содержимого, имеющие атрибут «кэшируемая операция», становятся видимыми другими ядрами после завершения обработки в кэше L2.

12.11 Кэш второго уровня (L2)

Встроенный кэш L2 обрабатывает как запросы команд, так и запросы данных. Кэш L2 и подключенные к нему ядра процессора вместе образуют процессорный кластер и создают иерархию кэша с индивидуальными кэшами L1 и кэшем L2 (см. рисунок 12.1).

Реализация кэша L2 имеет следующие особенности:
– Общий размер: 128 КБ;
– Ассоциативность (количество путей): 8;
– Размер строки кэша: 64 КБ;
– Количество банков: 4;
– Политика записи: обратная запись, выделение;
– Политика замены: LRU;
– Кэш L2 включает в себя L1 (содержит все строки L1);
– Кэш L2 содержит включающий каталог;
– Протокол с ядром: проприетарный;
– Протокол с памятью: AMBA 4 AXI;
– Некэшируемые запросы направляются в память в обход L2.

Карта регистров управления/состояния (CSR) кэша L2

Карта регистров управления/состояния кэша L2 приведена в таблице 12.66.

Определяемый пользователем адрес L2_CSR_BASE для модуля L2_CSR должен быть определен MPU как выделенная область памяти с MPU_CONTROL.MTYPE=3. Для более подробной информации см. структуру регистра MPU_CONTROL. Фактический адрес L2_CSR_BASE является базовым адресом в этом регионе плюс 0x1000.

В случае, если адрес L2_CSR_BASE не определен должным образом, тогда доступ к регистрам модуля L2_CSR невозможен.

Таблица 12.66 – Карта CSR L2

Смещение адреса	Тип регистра запись/чтение	Обозначение	Описание
0x000	RO	L2_VID	Версия ID
0x004	RO	L2_DESCR0	Регистр 0 описания L2
0x008	RO	L2_DESCR1	Регистр 1 описания L2
0x010	RW	L2_CTRL	Регистр включения кэша
0x014	RW	L2_FLUSH	Регистр очистки кэша
0x018	RW	L2_INV	Регистр аннулирования кэша
0x020	RO	L2_CTRL_DENY	Регистр статуса отмены включения кэша
0x024	RO	L2_FLUSH_DENY	Регистр статуса отмены очистки кэша
0x028	RO	L2_INV_DENY	Регистр статуса отмены аннулирования кэша
0x02C	RO	L2_CTRL_BUSY	Регистр статуса кэша
0x040	RW	L2_MCP	Регистр многоциклового доступа
0x080	RSW1C	L2_ERR_STATUS0	Регистр статуса ошибок транзакций на шине AXI
0x088	RSW1C	L2_ERR_STATUS1	Регистр статуса ошибок в памяти тегов
0x08C	RSW1C	L2_ERR_STATUS2	Регистр статуса ошибок в памяти данных
0x090	RW	L2_ERR_IRQ	Регистр разрешения прерываний по ошибкам
0x100	RO	L2_MAU_RERR_ADDR	Регистр адреса ошибки на запрос чтения MAU
0x110	RO	L2_MAU_BERR_ADDR	Регистр адреса ошибки на запрос записи MAU
0x120	RO	L2_NCAU_RERR_ADDR	Регистр адреса ошибки на запрос чтения NCAU
0x130	RO	L2_NCAU_BERR_ADDR	Регистр адреса ошибки на запрос записи NCAU
0x140	RO	L2_MAU_RERR_CODE	Регистр кода ошибки на запрос чтения MAU
0x150	RO	L2_MAU_BERR_CODE	Регистр кода ошибки на запрос записи MAU
0x160	RO	L2_NCAU_RERR_CODE	Регистр кода ошибки на запрос чтения NCAU
0x170	RO	L2_NCAU_BERR_CODE	Регистр кода ошибки на запрос записи NCAU
0x180	RO	L2_SNOOP_LOCK	Регистр слежения за когерентностью L2 к L1

Продолжение таблицы 12.66

Смещение адреса	Тип регистра запись/чтение	Обозначение	Описание
0x200	RO	L2_ERR_TEB0	Регистр информации об ошибке ECC при обращении к памяти тегов для банка 0
0x204	RO	L2_ERR_TEB1	Регистр информации об ошибке ECC при обращении к памяти тегов для банка 1
0x208	RO	L2_ERR_TEB2	Регистр информации об ошибке ECC при обращении к памяти тегов для банка 2
0x20C	RO	L2_ERR_TEB3	Регистр информации об ошибке ECC при обращении к памяти тегов для банка 3
0x300	RO	L2_ERR_DEB0	Регистр информации об ошибке ECC при обращении к памяти данных для банка 0
0x304	RO	L2_ERR_DEB1	Регистр информации об ошибке ECC при обращении к памяти данных для банка 1
0x308	RO	L2_ERR_DEB2	Регистр информации об ошибке ECC при обращении к памяти данных для банка 2
0x30C	RO	L2_ERR_DEB3	Регистр информации об ошибке ECC при обращении к памяти данных для банка 3
0x400	RW	L2_PC_0_CTRL	Регистр управления нулевым счетчиком производительности
0x408	RW	L2_PC_0_LO	Регистр содержит младшую часть нулевого счетчика производительности
0x40C	RW	L2_PC_0_HI	Регистр содержит старшую часть нулевого счетчика производительности
0x410	RW	L2_PC_1_CTRL	Регистр управления первым счетчиком производительности
0x418	RW	L2_PC_1_LO	Регистр содержит младшую часть первого счетчика производительности
0x41C	RW	L2_PC_1_HI	Регистр содержит старшую часть первого счетчика производительности

Продолжение таблицы 12.66

Смещение адреса	Тип регистра запись/чтение	Обозначение	Описание
0x420	RW	L2_PC_2_CTRL	Регистр управления вторым счетчиком производительности
0x428	RW	L2_PC_2_LO	Регистр содержит младшую часть второго счетчика производительности
0x42C	RW	L2_PC_2_HI	Регистр содержит старшую часть второго счетчика производительности
0x430	RW	L2_PC_3_CTRL	Регистр управления третьим счетчиком производительности
0x438	RW	L2_PC_3_LO	Регистр содержит младшую часть третьего счетчика производительности
0x43C	RW	L2_PC_3_HI	Регистр содержит старшую часть третьего счетчика производительности
0x500	RW	L2_SWERR_CNTR_EN	Регистр включения счетчика программных ошибок
0x504	RO	L2_SWERR_CNTR_OVFL	Регистр флагов переполнения счетчика программных ошибок
0x508	RW	L2_SWERR_CNTR_THLD	Регистр порога счетчика программных ошибок
0x50C	RW	L2_SWERR_CNTR_SEL	Регистр селектора счетчика программных ошибок
0x510	RO	L2_SWERR_CNTR_CNTR	Регистр счетчика программных ошибок

Описание регистров CSR

L2_VID

32-разрядный, доступный только для чтения, регистр, содержащий уникальный код версии реализации L2. Структура регистра L2_VID приведена в таблице 12.67.

Таблица 12.67 – Структура регистра L2_VID

Поле	Биты	Доступ	Описание
REL	0-7	RO	Релиз (шестнадцатиразрядное значение)
Day	8-15	RO	День (шестнадцатиразрядное значение)
Mon	16-23	RO	Месяц (шестнадцатиразрядное значение)
Year	24-31	RO	Год (шестнадцатиразрядное значение)

L2_DESCR0

Данный регистр описывает микроархитектуру L2. Структура регистра L2_DESCR0 приведена в таблице 12.68.

Таблица 12.68 – Структура регистра L2_DESCR0

Поле	Биты	Доступ	Сброс	Описание
L2 WAYS	0-2	RO	3	2^n количества путей
L2 ED port	3	RO	1	Наличие встроенного порта: 1 – имеется; 0 – недоступен.
L2 LINE	4-7	RO	5	2^n размера строки кэша (в байтах)
L2 WIDTH	8-12	RO	7	Размер индекса кэша
L2 TYPE	13-15	RO	7	Тип кэша L2: с обратной записью, распределением, инклюзивный
L2 BANKS	16-19	RO	3	Количество банков минус 1
RSV	20-27	RO	0	Зарезервировано
CPUNUM	28-31	RO	1	Количество CPU минус 1

L2_DESCR1

В этом регистре отображаются биты адреса для чередования банков. Текущая реализация использует биты [12] и [5] для чередования банков.

L2_CTRL

Этот регистр управляет включением банков. Если запрошенный банк отключен, доступ к памяти перенаправляется на кэшируемый интерфейс AXI непосредственно через модуль MAU. Структура регистра L2_CTRL приведена в таблице 12.69.

Таблица 12.69 – Структура регистра L2_CTRL

Поле	Биты	Доступ	Сброс	Описание
BANK_EN	0-3	RW	0	Управление включением банков
RSV	4-31	RZ	0	Зарезервировано

L2_FLUSH

Этот регистр управляет очисткой банков (все «грязные» строки кэша в банке будут перезаписаны и аннулированы). Структура регистра L2_FLUSH приведена в таблице 12.70.

Таблица 12.70 – Структура регистра L2_FLUSH

Поле	Биты	Доступ	Сброс	Описание
BANK_EN	0-3	RW	0	Управление очисткой банков: 0 – банк не в процессе очистки; 1 – банк в процессе очистки.
RSV	4-31	RZ	0	Зарезервировано

L2_INV

Этот регистр контролирует аннулирование банков (все строки кэша в банке будут аннулированы). Структура регистра L2_INV приведена в таблице 12.71.

Таблица 12.71 – Структура регистра L2_INV

Поле	Биты	Доступ	Сброс	Описание
BANK_EN	0-3	RW	0	Управление аннулированием банков: 0 – банк не в процессе аннулирования; 1 – банк в процессе аннулирования.
RSV	4-31	RZ	0	Зарезервировано

L2_CTRL_DENY

Этот регистр содержит статусы отмены включения банков кэша. Структура регистра L2_CTRL_DENY приведена в таблице 12.72.

Таблица 12.72 – Структура регистра L2_CTRL_DENY

Поле	Биты	Доступ	Сброс	Описание
BANK_EN	0-3	RO	0	Статус отмены включения банков кэша
RSV	4-31	RZ	0	Зарезервировано

L2_FLUSH_DENY

Этот регистр содержит статусы отмены очистки банков кэша. Структура регистра L2_FLUSH_DENY приведена в таблице 12.73.

Таблица 12.73 – Структура регистра L2_FLUSH_DENY

Поле	Биты	Доступ	Сброс	Описание
BANK_EN	0-3	RO	0	Статус отмены очистки банков кэша
RSV	4-31	RZ	0	Зарезервировано

L2_INV_DENY

Этот регистр содержит статусы отмены аннулирования банков кэша. Структура регистра L2_INV_DENY приведена в таблице 12.74.

Таблица 12.74 – Структура регистра L2_INV_DENY

Поле	Биты	Доступ	Сброс	Описание
BANK_EN	0-3	RO	0	Статус отмены аннулирования банков кэша
RSV	4-31	RZ	0	Зарезервировано

L2_CTRL_BUSY

Этот регистр показывает, какие банки находятся в состоянии занятости. Это состояние устанавливается, когда банк находится в одном из следующих активных состояний:

– включен;
– в процессе очистки;
– в процессе аннулирования;
– в процессе MCP.

Структура регистра L2_CTRL_BUSY приведена в таблице 12.75.

Таблица 12.75 – Структура регистра L2_CTRL_BUSY

Поле	Биты	Доступ	Сброс	Описание
BANK_EN	0-3	RO	0	Банки в активном состоянии
RSV	4-31	RZ	0	Зарезервировано

L2_MCP

Этот регистр используется для изменения многоцикловой логики в пути данных для доступа к памяти. Структура регистра L2_MCP приведена в таблице 12.76.

Таблица 12.76 – Структура регистра L2_MCP

Поле	Биты	Доступ	Сброс	Описание
UPD	0	RW	0	Выполняется обновление MCP
DENY	1	RO	0	Статус отмены
CFG	2-3	RW	0	Конфигурация MCP. Настройка количества тактов для операции чтения из памяти (3 – зарезервировано)
RSV	4-31	RZ	0	Зарезервировано

Примечание – MCP не может быть обновлен, т.к. банк включен, либо находится в процессе очистки или аннулирования в этом случае в регистре будет установлен бит DENY.

L2_ERR_STATUS0

Этот регистр собирает статусы ошибок транзакций по шине AXI. Структура регистра L2_ERR_STATUS0 приведена в таблице 12.77.

Таблица 12.77 – Структура регистра L2_ERR_STATUS0

Поле	Биты	Доступ	Сброс	Описание
MAURE	0	RW1C	0	Обнаружена ошибка чтения в кэшируемом интерфейсе AXI
MAUWE	1	RW1C	0	Обнаружена ошибка записи в кэшируемом интерфейсе AXI
NCAURE	2	RW1C	0	Обнаружена ошибка чтения в некэшируемом интерфейсе AXI
NCAUWE	3	RW1C	0	Обнаружена ошибка записи в некэшируемом интерфейсе AXI
RSV	4-31	RZ	0	Зарезервировано

L2_ERR_STATUS1

Этот регистр собирает статус обнаруженной ошибки ECC в памяти тегов (TAG) для каждого банка. Структура регистра L2_ERR_STATUS1 приведена в таблице 12.78.

Таблица 12.78 – Структура регистра L2_ERR_STATUS1

Поле	Биты	Доступ	Сброс	Описание
ETM	0-3	RW1C	0	Возникает ошибка ECC в памяти тегов (по одному биту для каждого банка)
RSV	4-31	RZ	0	Зарезервировано

L2_ERR_STATUS2

Этот регистр собирает статус обнаруженной ошибки ECC в памяти данных (DATA) для каждого банка. Структура регистра L2_ERR_STATUS2 приведена в таблице 12.79.

Таблица 12.79 – Структура регистра L2_ERR_STATUS2

Поле	Биты	Доступ	Сброс	Описание
EDM	0-3	RW1C	0	Возникает ошибка ECC в памяти данных (по одному биту для каждого банка)
RSV	4-31	RZ	0	Зарезервировано

L2_ERR_IRQ

Этот регистр определяет, какие события приведут к установлению сигнала l2_irq. Структура регистра L2_ERR_IRQ приведена в таблице 12.80.

Таблица 12.80 – Структура регистра L2_ERR_IRQ

Поле	Биты	Доступ	Сброс	Описание
ECC2	0	RW	0	Двойная ошибка ECC
ECC1	1	RW	0	Единичная ошибка ECC
AXI	2	RW	0	Событие ошибки AXI
SW	3	RW	0	Программная генерация прерывания
SWERR	4	RZ	0	Программная ошибка
SNOOP	5	RW	0	Прерывание блокировки когерентрости
RSV	6-31	RZ	0	Зарезервировано

L2_MAU_RERR_ADDR

Этот регистр содержит адрес ошибки запроса чтения из кэшируемого интерфейса AXI. Данный регистр доступен только для чтения. Запись единицы в бит MAURE регистра L2_ERR_STATUS0 приводит к сбросу данного регистра.

L2_MAU_BERR_ADDR

Этот регистр содержит адрес ошибки запроса записи из кэшируемого интерфейса AXI. Данный регистр доступен только для чтения. Запись единицы в бит MAUWE регистра L2_ERR_STATUS0 приводит к сбросу данного регистра.

L2_NCAU_RERR_ADDR

Этот регистр содержит адрес ошибки запроса чтения из кэшируемого интерфейса AXI. Данный регистр доступен только для чтения. Запись единицы в бит NCAURE регистра L2_ERR_STATUS0 приводит к сбросу данного регистра.

L2_NCAU_BERR_ADDR

Этот регистр содержит адрес ошибки запроса записи из кэшируемого интерфейса AXI. Данный регистр доступен только для чтения. Запись единицы в бит NCAUWE регистра L2_ERR_STATUS0 приводит к сбросу данного регистра.

L2_MAU_RERR_CODE

Этот регистр содержит дополнительную информацию об ошибке запроса чтения MAU. Этот регистр доступен только для чтения. Запись единицы в бит MAURE регистра L2_ERR_STATUS0 приводит к сбросу данного регистра. Структура регистра L2_MAU_RERR_CODE приведена в таблице 12.81.

Таблица 12.81 – Структура регистра L2_MAU_RERR_CODE

Поле	Биты	Доступ	Сброс	Описание
RESP	0-1	RO	0	Код ошибки чтения из кэшируемого AXI (AXI_RRESP)
RSV	2-3	RZ	0	Зарезервировано
ID	4-7	RO	0	ID ошибки чтения из кэшируемого AXI (AXI_RID)
RSV	8-31	RZ	0	Зарезервировано

L2_MAU_BERR_CODE

Этот регистр содержит дополнительную информацию об ошибке запроса записи MAU. Этот регистр доступен только для чтения. Запись единицы в бит MAUWE регистра L2_ERR_STATUS0 приводит к сбросу данного регистра. Структура регистра L2_MAU_BERR_CODE приведена в таблице 12.82.

Таблица 12.82 – Структура регистра L2_MAU_BERR_CODE

Поле	Биты	Доступ	Сброс	Описание
RESP	0-1	RO	0	Код ошибки записи из кэшируемого AXI (AXI_BRESP)
RSV	2-3	RZ	0	Зарезервировано
ID	4-7	RO	0	ID ошибки записи из кэшируемого AXI (AXI_BID)
RSV	8-31	RZ	0	Зарезервировано

L2_NCAU_RERR_CODE

Этот регистр содержит дополнительную информацию об ошибке запроса чтения NCAU. Этот регистр доступен только для чтения. Запись единицы в бит NCAURE регистра L2_ERR_STATUS0 приводит к сбросу данного регистра. Структура регистра L2_NCAU_RERR_CODE приведена в таблице 12.83.

Таблица 12.83 – Структура регистра L2_NCAU_RERR_CODE

Поле	Биты	Доступ	Сброс	Описание
RESP	0-1	RO	0	Код ошибки чтения из некэшируемого AXI (AXI_NC_RRESP)
RSV	2-3	RZ	0	Зарезервировано
ID	4-7	RO	0	ID ошибки чтения из некэшируемого AXI (AXI_NC_RID)
RSV	8-31	RZ	0	Зарезервировано

L2_NCAU_BERR_CODE

Этот регистр содержит дополнительную информацию об ошибке запроса записи NCAU. Этот регистр доступен только для чтения. Запись единицы в бит NCAUWE регистра L2_ERR_STATUS0 приводит к сбросу данного регистра. Структура регистра L2_NCAU_BERR_CODE приведена в таблице 12.84.

Таблица 12.84 – Структура регистра L2_NCAU_BERR_CODE

Поле	Биты	Доступ	Сброс	Описание
RESP	0-1	RO	0	Код ошибки записи из некэшируемого AXI (AXI_NC_BRESP)
RSV	2-3	RZ	0	Зарезервировано
ID	4-7	RO	0	ID ошибки записи из некэшируемого AXI (AXI_NC_BID)
RSV	8-31	RZ	0	Зарезервировано

L2_SNOOP_LOCK

Этот регистр собирает данные о нарушении когерентрости, когда L2 сигнализирует L1 об изменении данных для каждого интерфейса кэша L1.

Когда какой-либо кэш не отвечает на запрос когерентрости в течение 255 тактов, L2 устанавливает соответствующий бит в этом регистре. Данные биты маскируют запросы когерентности для соответствующего кэша L1.

Эти биты являются привязанными и могут быть очищены только путем сброса кэша L2. Структура регистра L2_SNOOP_LOCK приведена в таблице 12.85.

Таблица 12.85 – Структура регистра L2_SNOOP_LOCK

Поле	Биты	Доступ	Сброс	Описание
SLD0L1I	0	RO	0	Обнаружено нарушение когерентрости кэша L2 к L1I для ядра 0
SLD0L1D	1	RO	0	Обнаружено нарушение когерентрости кэша L2 к L1D для ядра 0
SLD1L1I	2	RO	0	Обнаружено нарушение когерентрости кэша L2 к L1I для ядра 1
SLD1L1D	3	RO	0	Обнаружено нарушение когерентрости кэша L2 к L1D для ядра 1
RSV	4-31	RZ	0	Зарезервировано

L2_ERR_TEB0..3

Эти регистры содержат информацию об ошибке ECC при обращении к памяти тегов для каждого банка. Данный регистр доступен только для чтения. Запись единицы в биты ETM регистра L2_ERR_STATUS1 сбрасывает ошибку для соответствующего банка. Структура регистров L2_ERR_TEB0 - L2_ERR_TEB3 приведена в таблице 12.86.

Таблица 12.86 – Структура регистров L2_ERR_TEB0 - L2_ERR_TEB3

Поле	Биты	Доступ	Сброс	Описание
TYPE	0	RO	0	Тип ошибки: 0 – одиночная ошибка с возможностью исправления; 1 – двойнаяошибка, которую невозможно исправить.
IDX	1-9	RO	–	Индекс строки кэша
RSV	10-31	RZ	0	Зарезервировано

L2_ERR_DEB0..3

Эти регистры содержат информацию об ошибке ECC при обращении к памяти данных для каждого банка.

Этот регистр доступен только для чтения. Запись единицы в биты EDM регистра L2_ERR_STATUS2 сбрасывает ошибку для соответствующего банка. Структура регистров L2_ERR_DEB0 - L2_ERR_DEB3 приведена в таблице 12.87.

Таблица 12.87 – Структура регистров L2_ERR_DEB0 - L2_ERR_DEB3

Поле	Биты	Доступ	Сброс	Описание
ERRT	0	RO	0	Тип ошибки: 0 – ошибка с возможностью исправления; 1 – ошибка, которую невозможно исправить.
IDX	1-9	RO	–	Индекс строки кэша
RSV	10-31	RZ	0	Зарезервировано

L2_PC_0_CTRL .. L2_PC_3_CTRL

Управление счетчиком производительности.

Модуль CSR L2 содержит 4 счетчика производительности. Каждый счетчик может быть сконфигурирован для сбора данных об одном типе событий из требуемых банков.

Банк выбирается в поле BANK в этом регистре (по одному разряду для каждого банка). Если не выбран ни один банк, счетчик производительности отключается. Событие кодируется в поле PCE этого регистра.

Структура регистров L2_PC_0_CTRL - L2_PC_3_CTRL приведена в таблице 12.88. Кодировка событий приведена в таблице 12.89.

Таблица 12.88 – Структура регистров L2_PC_0_CTRL - L2_PC_3_CTRL

Поле	Биты	Доступ	Сброс	Описание
PCE	0-3	RW	0	Селектор событий счетчика производительности
BANK	4-7	RW	0	Селектор банка (по одному разряду для каждого банка)
RSV	8-31	RZ	0	Зарезервировано

Таблица 12.89 – Кодировка событий кэша для счетчиков производительности

ID	Описание
0	HIT
1	MISS
2	REFILL
3	EVICTC
4	EVICTD
5	ROLLBACK
6	COLLISION
7	REQ
8	SNOOP
9-15	Зарезервировано

L2_PC_0_LO .. L2_PC_3_LO
Содержит младшую часть счетчика производительности N(0..3).

L2_PC_0_HI .. L2_PC_3_HI
Содержит старшую часть счетчика производительности N(0..3)

L2_SWERR_CNTR_EN
Регистр включения счетчика программных ошибок. Структура регистра L2_SWERR_CNTR_EN приведена в таблице 12.90.

Таблица 12.90 – Структура регистра L2_SWERR_CNTR_EN

Поле	Биты	Доступ	Сброс	Описание
SC	0	RW	0	Включение счетчика ошибок LR/SC
SWAP	1	RW	0	Включение счетчика ошибок AMO SWAP
RSV	2-31	RZ	0	Зарезервировано

L2_SWERR_CNTR_OVFL
Регистр переполнения счетчика программных ошибок. Структура регистра L2_SWERR_CNTR_OVFL приведена в таблице 12.91.

Таблица 12.91 – Структура регистра L2_SWERR_CNTR_OVFL

Поле	Биты	Доступ	Сброс	Описание
OVFL	0-3	RO	0	Флаг переполнения счетчика программных ошибок для каждого ядра (сравнивается с L2_SWERR_CNTR_THLD)
RSV	4-31	RZ	0	Зарезервировано

Примечание – Запись в L2_SWERR_CNTR_EN сбрасывает регистр переполнения (для каждого ядра).

L2_SWERR_CNTR_THLD
Регистр порога счетчика программных ошибок. Структура регистра L2_SWERR_CNTR_THLD приведена в таблице 12.92.

Таблица 12.92 – Структура регистра L2_SWERR_CNTR_THLD

Поле	Биты	Доступ	Сброс	Описание
THLD	0-15	RW	ffff	Порог счетчика программных ошибок (общий)
RSV	16-31	RZ	0	Зарезервировано

L2_SWERR_CNTR_SEL
Регистр селектора счетчика программных ошибок. Структура регистра L2_SWERR_CNTR_SEL приведена в таблице 12.93.

Таблица 12.93 – Структура регистра L2_SWERR_CNTR_SEL

Поле	Биты	Доступ	Сброс	Описание
SEL	0-1	RW	0	Выбор счетчика программных ошибок (выбор ядра)
RSV	2-31	RZ	0	Зарезервировано

L2_SWERR_CNTR

Регистр счетчика программных ошибок.

Счетчик является общим для всех типов ошибок. Если какое-либо из отслеживаемых действий (LR/SC или AMO SWAP) прошло без сбоев – счетчик будет обнулен. Структура регистра L2_SWERR_CNTR приведена в таблице 12.94.

Таблица 12.94 – Структура регистра L2_SWERR_CNTR

Поле	Биты	Доступ	Сброс	Описание
SNTR	0-15	RO	0	Счетчик
RSV	16-31	RZ	0	Зарезервировано

Примечание – Запись в L2_SWERR_CNTR_EN сбрасывает регистр счетчика (для каждого ядра).

Управление кэшем L2

Включение управления

Процесс кэширования в кэше L2 может быть включен или отключен. Эта функция контролируется полем L2_CTRL.BANK_EN в процессе работы.

Если включен кэш L2 (L2_CTRL.BANK_EN = 0xf), все кэшируемые запросы обрабатываются в банках L2 (включая поиск в кэше и все необходимые действия по обеспечению согласованности).

Если кэш L2 отключен (L2_CTRL.BANK_EN = 0x0), все кэшируемые запросы выполняются в L2 MAU без поиска в кэше, но с согласованными атрибутами в AXI/ACE.

L2_CTRL.BANK_EN не влияет на некэшируемые запросы.

Переключение между включенным и отключенным состояниями не является мгновенным процессом, чтобы определить завершение процесса программно опрашивается L2_CTRL_BUSY.BANK_EN.

Последовательность включения кэша L2:

1) Запишите 0xF в L2_CTRL.BANK_EN.

2) Проверьте L2_CTRL_DENY.BANK_EN на наличие нулевого значения (в противном случае во включении было отказано, см. L2_CTRL_DENY).

3) Считывайте бит L2_CTRL_BUSY.BANK_EN до тех пор, пока он не станет равен нулю.

Последовательность отключения кэша L2:

1) Запишите 0x0 в L2_CTRL.BANK_EN.

2) Считывайте бит L2_CTRL_BUSY.BANK_EN до тех пор, пока он не станет равен нулю.

Примечание – Чтобы узнать текущее состояние кэша L2, можно прочитать L2_CTRL.BANK_EN.

Работа с кэшем

Самоочистка

L2 может запустить процесс собственной полной очистки. В этом случае все содержимое L2 будет аннулировано с записью «грязных» данных в память.

Последовательность сброса L2:

1) Установите бит L2_FLUSH.FLUSH_EN;

2) Проверьте L2_FLUSH_DENY.BANK_FL_DEN на наличие нулевого значения (в противном случае сброс был отменен, см. L2_FLUSH_DENY);

3) Считывайте бит L2_CTRL_BUSY.BANK_BUSY до тех пор, пока он не станет равен нулю.

Примечание – Чтобы узнать текущее состояние очистки L2, можно также прочитать бит L2_FLUSH.FLUSH_EN.

Самостоятельное аннулирование

L2 может запустить процесс собственной полной аннуляции. В этом случае все содержимое L2 будет аннулировано без записи в память «грязных» данных.

Последовательность аннулирования L2:

1) Установите бит L2_INV.INV_EN;

2) Проверьте L2_INV_DENY.BANK_INV_DEN на наличие нулевого значения (в противном случае аннулирование было отклонено, см. L2_INV_DENY);

3) Считывайте бит L2_CTRL_BUSY.BANK_BUSY до тех пор, пока он не станет равен нулю.

Примечание – Чтобы узнать текущее состояние аннулирования L2, можно также прочитать бит L2_INV.INV_EN.

Обработка ошибок

Кэш L2 обрабатывает следующие типы ошибок:

– Ошибка данных в RAM;

– Ошибка шины, которая является ответом от AXI/ACE со статусом ошибки (SLVERR, DECERR);

– Программная ошибка, которая означает неправильный доступ/использование CSR кэша L2.

Кэш L2 поддерживает исправление одиночных ошибок и обнаружение двойных ошибок (SECDED) в оперативной памяти тегов и оперативной памяти данных с использованием кода коррекции ошибок (ECC), хранящегося в оперативной памяти RAM вместе с данными.

Ошибки ECC в оперативной памяти тегов

В случае возникновения единичной ошибки в RAM тегов L2 выполняет аппаратную коррекцию ошибок, но при этом появляется следующая индикация:

1) L2_ERR_STATUS1.ETM устанавливается в 1 (для каждого банка);

2) L2_ERR_TEB (для каждого банка) заполняется информацией о возникшей ошибке ECC, при следующих ошибках ECC этот регистр не будет обновляться до тех пор, пока в L2_ERR_STATUS.ETM не будет записано значение 1.

– L2_ERR_TEB.TYPE сбрасывается в 0 (флаг одиночной ошибки);

– L2_ERR_TEB.IDX заполняется индексом поврежденной строки кэша;

3) Если L2_ERR_IRQ.ECC1 содержит 1 (единичные ошибки не маскированы), тогда L2 генерирует запрос на внешнее прерывание (IRQ).

В случае возникновения двойной ошибки в RAM тегов L2 выдает следующую индикацию:

1) L2_ERR_STATUS1.ETM устанавливается в 1 (для каждого банка);

2) L2_ERR_TEB (для каждого банка) заполняется информацией о возникшей ошибке ECC, при следующих ошибках ECC этот регистр не будет обновляться до тех пор, пока в L2_ERR_STATUS.ETM не будет записано значение 1;

– L2_ERR_TEB.TYPE будет содержать значение 1 (флаг двойной ошибки);

– L2_ERR_TEB.IDX заполняется индексом поврежденной строки кэша;

– Если L2_ERR_IRQ.ECC2 содержит 1 (двойные ошибки не маскированы), тогда кэш L2 генерирует запрос на внешнее прерывание (IRQ).

В случае возникновения двойной ошибки в RAM тегов предполагается, что когерентность кэша потеряна (потерян трек) и дальнейшее поведение не определено. В качестве наилучшего доступного решения предлагается аннулировать весь битый L2 (см. L2_INV) с потерей данных.

Ошибки ECC в оперативной памяти данных

В случае возникновения единичной ошибки в данных RAM L2 выполняет аппаратную коррекцию ошибок, но при этом появляется следующая индикация:

1) L2_ERR_STATUS2 устанавливается в 1 (для каждого банка);

2) L2_ERR_DEB (для каждого банка) заполняется информацией о возникшей ошибке ECC, при следующих ошибках ECC этот регистр не будет обновляться до тех пор, пока в L2_DERR_STS.EDM не будет записано значение 1;

– L2_ERR_DEB.TYPE сбрасывается в 0 (флаг одиночной ошибки);

– L2_ERR_DEB.IDX заполняется индексом поврежденной строки кэша;

– L2_ERR_DEB.WAY заполняется идентификатором поврежденной строки кэша;

3) Если L2_ERR_IRQ.ECC1 содержит 1 (единичные ошибки не маскированы), то кэш L2 генерирует запрос на внешнее прерывание (IRQ).

В случае возникновения двойной ошибки в RAM данных L2 удаляет строку кэша (аннулирует) со следующей индикацией:

1) Значение L2_ERR_STATUS2.EDM устанавливается в 1 (для каждого банка);

2) L2_ERR_DEB (для каждого банка) заполняется информацией о возникшей ошибке ECC, при следующих ошибках ECC этот регистр не будет обновляться до тех пор, пока в L2_DERR_STS.EDM не будет записано значение 1;

– L2_ERR_DEB.TYPE будет содержать значение 1 (флаг двойной ошибки);

– L2_ERR_DEB.IDX заполняется индексом поврежденной строки кэша;

– L2_ERR_DEB.WAY заполняется идентификатором пути поврежденной строки кэша;

3) Если L2_ERR_IRQ.ECC2 содержит 1 (двойные ошибки не маскированы), то кэш L2 генерирует запрос на внешнее прерывание (IRQ).

В случае двойной ошибки в RAM данных предполагается, что когерентность кэша потеряна (данные потеряны) и дальнейшее поведение не определено. В качестве наилучшего доступного решения предлагается аннулировать весь нарушенный L2 (см. L2_INV) с потерей данных.

Ошибки шины AXI

Ответы на шине AXI/ACE могут возвращать ошибку (BRESP/RRESP не "В порядке"). В кэше L2 есть указание для следующих случаев:

– AXI RRESP не OKAY в случае неисключительных транзакций (не LR);

– AXI RRESP не OKAY или EXOKAY для эксклюзивных транзакций (LR);

– AXI BRESP не OKAY в случае неисключительных транзакций (не SC);

– AXI BRESP не OKAY или EXOKAY для эксклюзивных транзакций (SC).

В случае ошибок чтения/записи MAU:

– L2_ERR_STATUS0.MAURE/L2_ERR_STATUS0.MAUBE устанавливаются в 1;

– L2_MAU_RERR_ADDR/L2_MAU_BERR_ADDR заполняются адресом, на который был получен ответ на запрос;

– L2_MAU_RERR_CODE/L2_MAU_BERR_CODE заполняются AXI ID и AXI RESP;

– Если L2_ERR_IRQ.AXI содержит 1 (ошибки AXI не маскированы), тогда L2 генерирует запрос на внешнее прерывание (IRQ).

В случае ошибок чтения/записи NCAU:

– L2_ERR_STATUS0.NCAURE/L2_ERR_STATUS0.NCAUBE устанавливаются в 1;

– L2_NCAU_RERR_ADDR/L2_NCAU_BERR_ADDR заполняются адресом, на который был получен ответ на запрос;

– L2_NCAU_RERR_CODE/L2_NCAU_BERR_CODE заполняются AXI ID и AXI RESP;

– Если L2_ERR_IRQ.AXI содержит 1 (ошибки AXI не маскированы), тогда L2 генерирует запрос на внешнее прерывание (IRQ).

В случае ошибки на шине AXI во время обработки кэшируемой операции кэш L2 будет:

– Не выделять эту поврежденную строку кэша (или сделать ее недействительной);

– Ответ отправителю запроса с заявленным битом ошибки в интерфейсе.

Программные ошибки

Некоторые циклические цепочки операций L2 (инициируемые соответствующими последовательностями циклических команд), которые включают в себя условные операции, подразумевающие получение конечного результата, ведущего к выходу из цикла. Для целей отладки кэш L2 обнаруживает некоторые из этих последовательностей и подсчитывает количество итераций цикла. Превышение лимита (установленного в регистре threshold) приводит к ошибке, указанной в регистрах кэша L2. Кэш L2 имеет один счетчик для всех типов на ядро.

Сбой при повторении LR/SC: SC не может продолжить, потому что другой запрос отменил резервирование LR.

– Кэш L2 подсчитывает повторяющиеся сбои SC для каждого ядра.

– Если хотя бы один SC для ядра завершен успешно, счетчик сбрасывается в 0.

– Ошибки подсчитываются, если установлен бит L2_SWERR_CNTR_EN.SC.

– Если счетчик ошибок превышает пороговое значение (L2_SWERR_CNTR_THLD), тогда устанавливается флаг переполнения (L2_SWERR_CNTR_OVFL).

– Если установлен бит L2_ERR_IRQ.SWERR (программные ошибки не маскированы), тогда L2 отправляет запрос на внешнее прерывание (IRQ).

– Чтобы прочитать значение счетчика, необходимо выбрать нужное ядро с помощью L2_SWERR_CNTR_SEL и прочитать L2_SWERR_CNTR (косвенная адресация).

Ошибка AMO SWAP: AMO SWAP меняет местами значения 1 и 1, а не 1 и 0 (mutex не разблокирован):

– Кэш L2 подсчитывает количество повторяющихся сбоев AMO SWAP для каждого ядра.

– Если хотя бы один AMO SWAP для ядра завершен успешно (возвращается значение 0), то счетчик сбрасывается в 0.

– Сбои считаются, если значение L2_SWERR_CNTR_EN.SWAP равно 1.

– Если счетчик сбоев превышает пороговое значение (L2_SWERR_CNTR_THLD), тогда устанавливается флаг переполнения (L2_SWERR_CNTR_OVFL).

– Если установлен бит L2_ERR_IRQ.SWERR (программные ошибки не маскированы), тогда L2 отправляет запрос на внешнее прерывание (IRQ).

– Чтобы прочитать значение счетчика, необходимо выбрать нужное ядро с помощью L2_SWERR_CNTR_SEL и прочитать L2_SWERR_CNTR (косвенная адресация).

12.12 Блок управления памятью (MMU)

Программная модель

Карта регистров блока MMU приведена в таблице 12.95.

Таблица 12.95 – Карта регистров блока MMU

Смещение	Название	Доступ	Описание
0xBC0	TLB_ATTR	RW	Атрибуты страницы TLB
0xBC1	TLB_VA	RW	Виртуальный адрес TLB. Адрес выравнивается по границе 4 килобайта.
0xBC2	TLB_UPDATE	WO	Триггер обновления TLB. Создает запись с данными TLB_ATTR и TLB_VA в TLB, указанные первыми двумя битами LSB. [0] бит для ITLB, [1] бит для DTLB. Есть возможность записи в оба TLB одновременно. [2] бит задает атрибут Megapage, который не может быть указан в регистре TLB_ATTR
0xBC3	TLB_SCAN	WO	Номер сканируемой записи. [31] бит определяет какой тип записи TLB считывается: 0 для ITLB, 1 для DTLB. Указанные ATTRS и VA считываются из соответствующих CSR.

Регистр атрибутов страницы памяти

В таблицах 12.96 и 12.97 приведены поля атрибутов для страницы размером 4 Кбайт и страницы размером 4 Мбайт соответственно.

Таблица 12.96 – Атрибуты страницы 4 Кбайт

Поле	Биты	Доступ	Описание
V	0	RW	Доступность (Valid)
R	1	RW	Атрибут доступа для чтения (Read access)
W	2	RW	Атрибут доступа на запись (Write access)
X	3	RW	Атрибут доступа на выполнение (Execute access)
U	4	RW	Атрибут доступа пользователя (User access)
G	5	RW	Атрибут глобальной страницы (Global page)
A	6	RW	Атрибут доступа (Accessed)
D	7	RW	Атрибут «грязный» (Dirty)
M	8	RZ	Атрибут мегастраницы, он равен 0 для страницы размером 4 Кбайт
RSV	9	RZ	Зарезервировано
PPN	29-10	RW	Физический номер страницы
RSV	31-30	R0	Зарезервировано

Таблица 12.97 – Атрибуты страницы 4 Мбайт

Поле	Биты	Доступ	Описание
V	0	RW	Доступность (Valid)
R	1	RW	Атрибут доступа для чтения (Read access)
W	2	RW	Атрибут доступа на запись (Write access)
X	3	RW	Атрибут доступа на выполнение (Execute access)
U	4	RW	Атрибут доступа пользователя (User access)
G	5	RW	Атрибут глобальной страницы (Global page)
A	6	RW	Атрибут доступа (Accessed)
D	7	RW	Атрибут «грязный» (Dirty)
M	8	RZ	Атрибут мегастраницы, он равен 1 для страницы размером 4 Мбайт
RSV	9-19	RZ	Зарезервировано
PPN	20-29	RW	Физический номер страницы
RSV	30-31	R0	Зарезервировано

Бит V указывает, является ли PTE доступным. Если V-бит равен 0, тогда биты 31..1 PTE не имеют значения и могут свободно использоваться программным обеспечением.

Биты разрешений, R, W и X указывают, доступна ли страница для чтения, записи и выполнения, соответственно. Если все три параметра равны нулю, PTE является указателем на следующий уровень таблицы страниц, в противном случае это конечный PTE. Страницы, доступные для записи, также должны быть помечены как доступные для чтения. Другие комбинации зарезервированы для использования в будущем. В таблице 12.98 приведена краткая информация о кодировке битов разрешений.

Таблица 12.98 – Кодирование полей R/W/X PTE

X	W	R	Описание
0	0	0	Указатель на следующий уровень таблицы страниц
0	0	1	Страница, доступная только для чтения
0	1	0	Зарезервировано для использования в будущем
0	1	1	Страница, доступная для чтения и записи
1	0	0	Страница, доступная только для выполнения
1	0	1	Страница, доступная для чтения и выполнения
1	1	0	Зарезервировано для использования в будущем
1	1	1	Страница, доступная для чтения, записи и выполнения

Бит U указывает, доступна ли страница в режиме пользователя (U-mode). Программа в режиме пользователя может получить доступ к странице только при U=1.

Если установлен бит SUM в регистре SSTATUS, программное обеспечение в режиме супервизора также может получать доступ к страницам с U=1. Однако код супервизора обычно работает со сброшенным битом SUM, и в этом случае код супервизора будет выдавать ошибку при доступе к страницам в пользовательском режиме.

Бит G обозначает глобальное отображение. Глобальные отображения – это те, которые существуют во всех адресных пространствах. Для неконечных PTE глобальная настройка подразумевает, что все отображения на последующих уровнях таблицы страниц являются глобальными. Обратите внимание, что невозможность пометить глобальное отображение как глобальное лишь снижает производительность, в то время как пометка неглобального отображения как глобальное является ошибкой.

Поле RSV зарезервировано для использования ПО супервизор; программа должна игнорировать это поле.

Каждый конечный PTE содержит доступный (A) и неактивный (D) биты. Бит A указывает, что виртуальная страница была прочитана, записана или извлечена с момента последнего удаления бита A. Бит D указывает, что виртуальная страница была записана с момента последнего удаления бита D.

Любой уровень PTE может быть конечным, поэтому в дополнение к страницам размером 4 КБ ядро поддерживает страницы размером 4 МБ. Мегастраница должна быть виртуально и физически выровнена по границе в 4 МБ; если физический адрес недостаточно выровнен, возникает исключение из-за ошибки страницы. Для неконечных PTE биты D, A и U зарезервированы для использования в будущем и должны быть очищены программным обеспечением для обеспечения прямой совместимости.

Регистр виртуального адреса

В таблицах 12.99 и 12.100 показана структура полей регистра VA для страницы размером 4 Кбайт и страницы размером 4 Мбайт соответственно.

Таблица 12.99 – Структура регистра VA для страницы 4 Кбайт

Поле	Биты	Доступ	Описание
ASID	0-8	RO	ASID. Это поле в записи TLB обновляется на текущий ASID во время загрузки TLB
SO	9	RO	Атрибут строгого упорядочения, обновляемый из MPU во время загрузки TLB
NC	10	RO	Атрибут «не кэшируемый», обновляется из MPU во время загрузки TLB
VPN	11-31	RW	Номер виртуальной страницы

Таблица 12.100 – Структура регистра VA для страницы 4 Мбайт

Поле	Биты	Доступ	Описание
ASID	0-8	RO	ASID. Это поле в записи TLB обновляется на текущий ASID во время загрузки TLB
SO	9	RO	Атрибут строгого упорядочения, обновляемый из MPU во время загрузки TLB
NC	10	RO	Атрибут «не кэшируемый», обновляется из MPU во время загрузки TLB
RSV	11-21	–	Зарезервировано
VPN	22-31	RW	Номер виртуальной страницы

Чтобы повысить производительность SWPW, поле "VPN" в регистре автоматически загружается:

– в DTLB при возникновении ошибки загрузки/сохранения страницы;

– в ITLB при возникновении ошибки страницы выборки инструкций.

Регистр TLB_UPDATE

В таблице 12.101 приведена структура полей регистра TLB_UPDATE.

Таблица 12.101 – Структура регистра TLB_UPDATE

Поле	Биты	Доступ	Описание
ITLB	0	WO	Обновить TLB для памяти инструкций
DTLB	1	WO	Обновить TLB для памяти данных
MP	2	WO	Установить атрибут мегастраницы
RSV	3-31	RZ	Зарезервировано

После любого события «ошибка страницы» аппаратно устанавливается соответствующий бит xTLB (I или/и D) в регистре TLB_UPDATE для ускорения обработки ошибок страницы.

Для получения дополнительной информации обратитесь к разделам "Обработка ошибок страницы" и "Исключения ошибок страницы".

Регистр TLB_SCAN

В таблице 12.102 приведена структура полей регистра TLB_SCAN.

Таблица 12.102 – Структура регистра TLB_SCAN

Поле	Биты	Доступ	Описание
PTE_IDX	0-4	WO	Индекс записи в выбранном массиве TLB
RSV	5-30	RZ	Зарезервировано
TLB_SEL	31	WO	Выбор TLB 0 – TLB для памяти команд; 1 – TLB для памяти данных.

Первичное считывание записей TLB проиллюстрировано на рисунке 12.5.

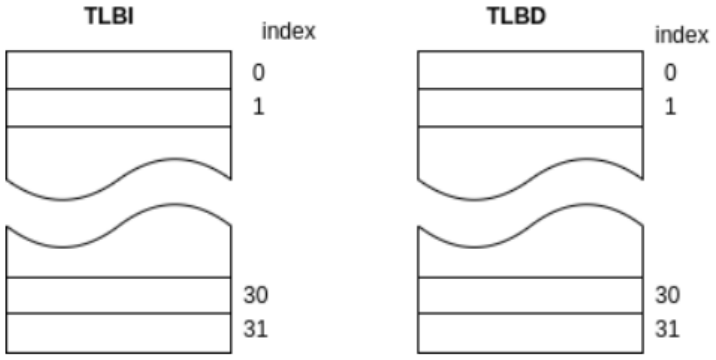


Рисунок 12.5 – Первичное считывание записей TLB

Сканирование:

1) Запишите адрес записи в регистр TLB_SCAN. Значение регистра интерпретируется как entry_number. 31 бит позволяет выбрать, какой именно TLB мы считываем (1 – из DTLB, иначе – ITLB);

2) Считываем атрибут TLB_ATTR;

3) Считываем виртуальный адрес TLB_VA.

Очистка TLB инструкций/данных

Команда SFENCE.VMA для управления памятью супервизора используется для очистки записей TLB, относящихся к текущему или всем адресным пространствам.

Просмотр страниц

В текущей реализации средство просмотра страниц SW используется для загрузки PTE в TLB.

Обработка ошибок страницы

В текущей реализации сгенерировано четыре исключения ошибки страницы:

– Ошибка страницы TLB Miss;

– Ошибка страницы инструкций;

– Ошибка страницы загрузки;

– Ошибка страницы Store/AMO.

Чтобы ускорить обработку ошибок страницы, аппаратное обеспечение выполняет следующие действия после любой из этих ошибок страницы:

– Загружает регистры SWPW_R0_ADDR, SWPW_R1_ADDR с предварительно вычисленными значениями для просмотра страниц;

– Загружает регистр TLB_VA с ошибочным виртуальным адресом в соответствующий TLBx (ITLB для доступа к инструкциям или DTLB для доступа к данным);

– Устанавливает соответствующий бит xTLB в регистре TLB_UPDATE.

Исключения ошибок страницы из MMU имеют приоритет над соответствующими исключениями ошибок доступа из MPU.

Ошибка страницы TLB Miss

Исключение генерируется, когда в соответствующем TLB нет доступного PTE для доступа к памяти (при выборке команд происходит ошибка ITLB, а при

загрузке/сохранении/AMO-операции – сбой DTLB). Это исключение не удалось передать в режим S-mode.

Средство просмотра страниц SW должно загрузить требуемый PTE в соответствующий xTLB. Пострадавший PTE в TLB выбирается с помощью псевдоалгоритма LRU.

Ошибки страницы TLB Instruction/Load/Store/AMO

Эти исключения генерируются, когда соответствующий xTLB содержит допустимый PTE, но атрибуты не разрешают доступ. Любое из этих исключений может быть передано в режим S-mode с помощью установленного соответствующего бита в регистре MEDELEG.

12.13 Исключения и прерывания

Термин «исключение» используется для обозначения необычного состояния, возникающего в ядре во время выполнения.

Термин trap используется для обозначения синхронной передачи управления контролирующей среде, когда это вызвано исключительным состоянием, возникающим в ядре. Контролирующая среда может быть представлена кодом, работающим либо в режиме S-mode, либо в режиме M-mode.

Термин «прерывание» используется для обозначения асинхронной передачи управления контролирующей среде, вызванной событием за пределами ядра.

Коды исключений режима M-mode

Коды исключений, поддерживаемые ядром в режиме M-mode, перечислены в таблице 12.103.

Таблица 12.103 – Список кодов исключения в режиме M-mode

Код исключения	Причина/описание исключения
0	Смещен адрес инструкции
1	Ошибка доступа к инструкции
2	Недопустимая инструкция
3	Точка останова
4	Адрес загрузки не выровнен
5	Ошибка доступа при загрузке
6	Store/AMO адрес не выровнен
7	Store/AMO ошибка доступа
8	Вызов среды из режима U-mode
9	Вызов среды из режима S-mode
10	Зарезервировано
11	Вызов среды из режима M-mode
12	Ошибка на странице инструкции
13	Ошибка загрузки страницы
14	Ошибка пропуска страницы TLB
15	Store/AMO ошибка страницы
16-31	Зарезервировано

Коды исключений режима S-mode

Коды исключений, поддерживаемые ядром в режиме S-mode, перечислены в таблице 12.104.

Таблица 12.104 – Список кодов исключения в режиме S-mode

Код исключения	Причина/описание исключения
0	Смещен адрес инструкции
1	Ошибка доступа к инструкции
2	Недопустимая инструкция
3	Точка останова
4	Зарезервировано
5	Ошибка доступа при загрузке
6	Store/AMO адрес не выровнен
7	Store/AMO ошибка доступа
8	Вызов среды из режима U-mode
9-11	Зарезервировано
12	Ошибка на странице инструкции
13	Ошибка загрузки страницы
14	Зарезервировано
15	Store/AMO ошибка страницы
16-31	Зарезервировано

Исключения ошибок страницы

Исключение «Ошибка пропущенной страницы TLB» возникает при возникновении ошибки TLB (не найдена допустимая запись TLB). Программный обработчик должен пройти через таблицы страниц, чтобы найти PTE и заполнить TLB, записав pte CSR в TLB_ATTR и нулевое значение в TLB_UPDATE. Целевые биты TLB (I или/и D) для TLB_UPDATE зависят от неудачной операции. Исключения "Ошибка страницы инструкций", "Ошибка страницы загрузки" и "Ошибка страницы Store/AMO" должны быть вызваны только в случае сбоя доступа (при действительной записи в TLB).

Существует 2 специальных регистра CSR, которые загружаются после исключения "Ошибка пропущенной страницы TLB".

Вычисление адреса SWPW_R0_ADDR: Содержит предварительно вычисленный адрес записи в таблице корневой страницы (PTE1_ADDR)

SWPW_R0_ADDR = (SATP.PPN << 12) + VADDR.VPN1 * 4

Смещение адреса SWPW_R1_ADDR: Содержит смещение PTE для базовой таблицы страниц второго уровня (PTE0_OFFS)

SWPW_R1_ADDR = VADDR.VPN0 * 4

Прерывания

Ядро поддерживает:

1) Включение/выключение прерываний;

2) Работу с ожидающими прерываниями;

3) Информирование об источнике прерываний с помощью соответствующих сообщений CSR:

– Регистры MIE, MIP и MCAUSE используются в режиме M-mode;

– Регистры SIE, SIP и SCAUSE используются в режиме S-mode.

Коды источников прерываний, поддерживаемые ядром, перечислены в таблице 12.105.

Таблица 12.105 – Список кодов прерываний

Код прерывания	Причина/описание прерывания
0	Зарезервировано
1	Программное прерывание супервизора
2	Зарезервировано
3	Программное прерывание машинное
4	Зарезервировано
5	Прерывание таймера супервизора
6	Зарезервировано
7	Прерывание таймера машинное
8	Зарезервировано
9	Внешнее прерывание супервизора
10	Зарезервировано
11	Внешнее прерывание машинное
12-30	Зарезервировано
31	Сбой по ошибке памяти

Ядро реализует приоритеты для прерываний (IRQ), как показано в таблице 12.106.

Таблица 12.106 – Таблица приоритетов прерываний

Приоритет	Прерывание
0 (высший)	Сбой по ошибке памяти
1	Внешнее прерывание машинное
2	Программное прерывание машинное
3	Прерывание таймера машинное
4	Внешнее прерывание супервизора
5	Программное прерывание супервизора
6 (низший)	Прерывание таймера супервизора

12.14 Ввод-вывод с отображением в памяти

Машинный таймер с отображением в памяти

Модуль Mtimer генерирует 64-разрядное значение в режиме реального времени, которое может быть разделено на несколько ядер RISC-V. В дополнение к обычному машинному таймеру, Mtimer поддерживает 64-разрядный регистр сравнения по таймеру для каждого ядра, который вызывает прерывание по таймеру, если значение таймера равно значению в регистре сравнения по таймеру.

В Mtimer также реализован делитель для управления скоростью счетчика, и он может синхронизироваться либо с помощью тактового сигнала ядра, либо с помощью часов реального времени.

Mtimer предлагает программный интерфейс, отображаемый в памяти, со следующими ограничениями:

– Интерфейс поддерживает только операции загрузки/сохранения;

– Атрибуты доступа к памяти должны быть "некэшированными, строго упорядоченными";

– Доступ к 32-разрядному регистру должен быть выровнен по 4-байтовой границе;

– Каждому ядру доступен единый комбинированный прием.

Если какое-либо из этих условий не выполняется, Mtimer игнорирует транзакцию и генерирует "Ответ об ошибке" на шине памяти.

Карта регистров Mtimer

Mtimer реализует два набора регистров:

1) Общие – к этим регистрам можно получить доступ из любого ядра, подключенного к Mtimer;

2) Индивидуальные – для каждого ядра существуют специальные регистры, к которым нельзя получить доступ из другого ядра.

Каждый набор индивидуальных регистров имеет один и тот же базовый адрес как в Mtimer, так и в карте памяти ввода-вывода.

Система внутренних связей предоставляет каждой транзакции идентификатор ядра источника для доступа к соответствующему индивидуальному регистру.

Отображение регистров Mtimer относительно определенного пользователем адреса MTIMER_BASE показано в таблице 12.107.

Определяемый пользователем адрес MTIMER_BASE для модуля Mtimer должен быть определен MPU как выделенная область памяти с MPU_CONTROL.MTYPE=3. Подробности см. в структуре регистра MPU_CONTROL.

Если адрес MTIMER_BASE не определен должным образом, тогда доступ к регистрам модуля Mtimer невозможен.

Таблица 12.107 – Карта регистров Mtimer

Смещение	Наименование
Общие регистры	
0x0	MTIMER_CONTROL
0x4	MTIMER_DIVIDER
0x8	MTIMER_MTIMELO
0xC	MTIMER_MTIMEHI
Индивидуальные регистры (Hart-Private)	
0x10	MTIMER_MTIMECMPLO
0x14	MTIMER_MTIMECMPHI

Структура регистров Mtimer

Структура регистров Mtimer приведена в таблице 12.108.

Таблица 12.108 – Структура регистров Mtimer

Поле	Бит	Доступ	Сброс	Описание
Общие регистры				
Регистр MTIMER_CONTROL [MTIMER_BASE+0x0]				
ENABLE	0	RW	1	Mtimer включен и инкрементируется
CLKSRC	1	RW	0	Источник тактового сигнала: 0 – Тактовый сигнал ядра; 1 – Тактовый сигнал реального времени.
RSV	2-31	RZ	0	Зарезервировано
Регистр MTIMER_DIVIDER [MTIMER_BASE+0x4]				
DIVIDER	0-9	RW	0	Делитель инкремента таймера: 0 – Таймер инкрементируется каждый цикл выбранного тактового сигнала; 1..2023 – Таймер инкрементируется каждый DIVIDER+1 цикл выбранного тактового сигнала.
RSV	10-31	RZ	0	Зарезервировано
Регистр MTIMER_MTIMELO [MTIMER_BASE+0x8]				
MTIMELO	0-31	RW	0	Младшие 32 бита счетчика MTIME
Регистр MTIMER_MTIMEHI [MTIMER_BASE+0xC]				
MTIMEHI	0-31	RW	0	Старшие 32 бита счетчика MTIME
Регистры Hart-Private				
Регистр MTIMER_MTIMECMPLO [MTIMER_BASE+0x10]				
MTIMECMPLO	0-31	RW	0	Младшие 32 бита компаратора MTIMECMP
Регистр MTIMER_MTIMECMPHI [MTIMER_BASE+0x14]				
MTIMECMPHI	0-31	RW	0	Старшие 32 бита компаратора MTIMECMP

Регистр делителя управляет внутренним счетчиком деления. Этот счетчик обнуляется при любом вмешательстве в регистры Mtimer или Divider.

Если значение регистра (MTIMEHI, MTIMELO) становится равным (MTIMECMPHI, MTIMECMPLO) для любого ядра, то для этого ядра запрашивается прерывание по таймеру. Это прерывание будет отменено после выполнения операции сохранения в регистре MTIMER_MTIMECMPLO или MTIMER_MTIMECMPHI.

12.15 Модуль межъядерной связи (ICCM)

Модуль межъядерной связи (ICCM) используется для поддержания синхронизации и обмена данными между ядрами с помощью механизма передачи сообщений.

В простом примере с двумя ядрами, работающими параллельно, если запись в таблице страниц ядра изменяется, оба ядра должны либо очистить свои TLB, либо аннулировать эту конкретную запись в таблице страниц. Какое бы ядро ни изменило отображение, у него достаточно информации, чтобы своевременно и автоматически аннулировать TLB, но у другого ядра такой информации нет. Следовательно, ядро, изменившее отображение, должно отправить сообщение другому ядру, чтобы оно очистило его TLB или аннулировало запись в таблице страниц. Именно здесь может быть использован механизм ICCM.

Функциональное описание

Каждое ядро имеет набор «почтовых ящиков» (равный количеству подключенных ядер), уникальный адрес (проводной) и линию прерывания, подключенную к линии MSI (Machine Software Interrupt).

Линия прерывания ядра чувствительна к уровню.

Линия прерывания ядра остается активной до тех пор, пока почтовый ящик не опустеет.

Отправка сообщения

ICCM может использоваться в ядре как для отправки сообщений определенному ядру, так и для самостоятельного обмена сообщениями.

Чтобы отправить сообщение, необходимо ввести адрес получателя в регистр адресов ADDRESS, затем ввести значение WRDATA для отправки сообщения.

Чтение входящего сообщения

Считайте значение регистра RDDATA, чтобы проверить статус входящего почтового ящика. Инструкция CSRRW должна использоваться для атомарного считывания и очистки регистра RDDATA.

Старший бит – это статус входящего сообщения:

– 1 – почтовый ящик пуст,

– 0 – в младшей части сообщения присутствуют биты данных и адреса источника.

Структура регистра RDDATA показана в таблице 12.109.

Таблица 12.109 – Структура регистра RDDATA

Поле	Биты	Описание
SRC	0-1	Адрес источника или максимально доступный адрес, если установлен бит STATUS
DATA	2-30	Биты данных (когда STATUS =0, содержит биты передаваемого сообщения)
STATUS	31	Статус почтового ящика: 1 – пуст, 0 – заполнен.

Карта регистров

Отображение машинных (M-mode) регистров ICCM показано в таблице 12.110.

Таблица 12.110 – Карта регистров ICCM в M-mode

Поле	Адрес	Доступ	Описание
ADDRESS	0xBD8	RW	Адрес назначения
STATUS	0xBD9	RO	Статус почтового ящика назначения (0 - пуст, 1 - занят)
RDDATA	0xBDA	RW	Статус/значение входящего почтового ящика
WRDATA	0xBDB	WO	Исходящее сообщение (триггер «Отправить сообщение»)

Отображение регистров ICCM супервизора (S-mode) показано в таблице 12.111.

Таблица 12.111 – Карта регистров ICCM в S-mode

Поле	Адрес	Доступ	Описание
ADDRESS	0x9D8	RW	Адрес назначения
STATUS	0x9D9	RO	Статус почтового ящика назначения (0 - пуст, 1 - занят)
RDDATA	0x9DA	RW	Статус/значение входящего почтового ящика
WRDATA	0x9DB	WO	Исходящее сообщение (триггер «Отправить сообщение»)

12.16 Система команд

В процессоре реализована RISC-V система команд RV32IMAFDCP включающая в себя следующие расширения:

– RV32: 32-битная архитектура (ширина регистров общего назначения x0-x31 составляет 32 бита).

– I: Базовый набор целочисленных команд (Integer).

– M: Умножение и деление (Multiply).

– A: Атомарные операции (Atomic).

– F: Числа с плавающей запятой одинарной точности (Single-precision Float, 32-bit).

– D: Числа с плавающей запятой двойной точности (Double-precision Float, 64-bit).

– C: Сжатые инструкции (Compressed, 16-bit).

– P: Упакованные SIMD-инструкции (Packed/DSP extension) .

Поддерживаемые команды представлены в таблицах 12.112 – 12.118.

Таблица 12.112 – RV32I: базовый целочисленный набор

Мнемокод команды	Формат	Операнды	Краткое описание
1	2	3	4
Арифметические:			
LUI	U	rd, imm	Load Upper Immediate. Загружает 20-битное число imm в старшие 20 бит регистра rd. Младшие 12 бит обнуляются
AUIPC	U	rd, imm	Add Upper Immediate to PC. Складывает 20-битное число imm (сдвинутое на 12 бит влево) с текущим PC и пишет в rd
ADDI	I, Cx	rd, rs1, imm	Add Immediate. Складывает регистр rs1 и 12-битное знаковое число imm. Результат в rd
SLTI	I	rd, rs1, imm	Set Less Than Immediate. Если rs1 меньше imm (знаковое сравнение), то rd = 1, иначе rd = 0
SLTIU	I	rd, rs1, imm	Set Less Than Immediate Unsigned. То же, что SLTI, но сравнение беззнаковое
XORI	I	rd, rs1, imm	XOR Immediate. Побитовое исключающее «ИЛИ» с непосредственным значением
ORI	I	rd, rs1, imm	OR Immediate. Побитовое «ИЛИ» с непосредственным значением
ANDI	I	rd, rs1, imm	AND Immediate. Побитовое «И» с непосредственным значением
SLLI	I, Cx	rd, rs1, shamt	Shift Left Logical Immediate. Логический сдвиг влево на непосредственное значение (shamt). В младшие биты попадают нули.
SRLI	I	rd, rs1, shamt	Shift Right Logical Immediate. Логический сдвиг вправо на непосредственное значение. В старшие биты попадают нули.

Продолжение таблицы 12.112

1	2	3	4
SRAI	I	rd, rs1, shamt	Shift Right Arithmetic Immediate. Арифметический сдвиг вправо на непосредственное значение
ADD	R, Cx	rd, rs1, rs2	Add. Арифметическое сложение: rd = rs1 + rs2 Переполнения игнорируются
SUB	R, Cx	rd, rs1, rs2	Subtract. Арифметическое вычитание: rd = rs1 - rs2
SLL	R	rd, rs1, rs2	Shift Left Logical. Логический сдвиг rs1 влево на количество бит, указанное в младших 5 битах rs2
SLT	R	rd, rs1, rs2	Set Less Than. Если rs1 < rs2 (знаковое), rd = 1, иначе 0
SLTU	R	rd, rs1, rs2	Set Less Than Unsigned. Если rs1 < rs2 (беззнаковое), rd = 1, иначе 0
XOR	R	rd, rs1, rs2	Exclusive OR. Побитовое исключающее «ИЛИ»
SRL	R	rd, rs1, rs2	Shift Right Logical. Логический сдвиг rs1 вправо (заполнение нулями) на число бит из rs2
SRA	R	rd, rs1, rs2	Shift Right Arithmetic. Арифметический сдвиг rs1 вправо (дублирование знака) на число бит из rs2
OR	R, Cx	rd, rs1, rs2	OR. Побитовое «ИЛИ»
AND	R, Cx	rd, rs1, rs2	AND. Побитовое «И»
Загрузка/Сохранение:			
LB	I	rd, rs1, imm	Load Byte. Загружает 8 бит из памяти по адресу rs1 + offset. Выполняет знаковое расширение в 32 бита (sign-extend)
LH	I	rd, rs1, imm	Load Halfword. Загружает 16 бит
LW	I, Cx	rd, rs1, imm	Load Word. Загружает 32 бита
LBU	I	rd, rs1, imm	Load Byte Unsigned. Загружает 8 бит. Старшие 24 бита заполняются нулями (zero-extend)
LHU	I	rd, rs1, imm	Load Halfword Unsigned. Загружает 16 бит. Расширяет старшие 16 бит нулями
SB	S	rs1, rs2, imm	Store Byte. Сохраняет младшие 8 бит регистра rs2 в память по адресу rs1 + offset
SH	S	rs1, rs2, imm	Store Halfword. Сохраняет младшие 16 бит rs2 в память
SW	S, Cx	rs1, rs2, imm	Store Word. Сохраняет все 32 бита rs2 в память

Продолжение таблицы 12.112

1	2	3	4
Управление переходами:			
BEQ	SB, Cx	rs1, rs2, imm	Branch if Equal. Переход на offset, если rs1 = rs2
BNE	SB, Cx	rs1, rs2, imm	Branch if Not Equal. Переход, если rs1 ≠ rs2
BLT	SB	rs1, rs2, imm	Branch if Less Than. Переход, если rs1 < rs2 (знаковое)
BGE	SB	rs1, rs2, imm	Branch if Greater or Equal. Переход, если rs1 ≥ rs2 (знаковое).
BLTU	SB	rs1, rs2, imm	Branch Less Than Unsigned. Переход, если rs1 < rs2 (беззнаковое).
BGEU	SB	rs1, rs2, imm	Branch Greater/Equal Unsigned. Переход, если rs1 ≥ rs2 (беззнаковое)
JAL	UJ, Cx	rd, imm	Jump And Link. Безусловный переход. Сохраняет адрес следующей инструкции (PC+4) в rd и прыгает на PC + offset. Используется для вызова функций.
JALR	UJ, Cx	rd, rs1, imm	Jump And Link Register. Косвенный переход. Прыгает по адресу rs1 + offset. Сохраняет адрес возврата в rd. Используется для возврата из функций или вызова по указателю.
Системные:			
FENCE		-	Memory Fence. Синхронизация порядка доступа к памяти
FENCE.I	I	-	Instruction-Fetch Fence. Обеспечивает согласованность между памятью данных и кэшем инструкций: гарантирует, что изменения кода в памяти (через store) будут видны процессору при последующей выборке инструкций
ECALL	I	-	Environment Call. Вызов системной функции (syscall). Передает управление ядру ОС (генерирует исключение)
EBREAK	I	-	Environment Break. Вызывает исключение для отладчика (breakpoint)
CLINV	I	rs1	Удаляет строку кэша из L1 и L2. Аннулирует, если присутствует в L1. Удаляет из L2 и выполняет проверку когерентности для ядер.
CLFLUSH	I	rs1	Записывает строку кэша в память (в случае «грязных данных») и удаляет ее из L1 и L2. Делает недействительной, если она присутствует в L1. Записывает в память, если она присутствует в L2 и «загрязнена», удаляет из L2 и выполняет проверку когерентности для ядер

Примечание – Попытки выполнить инструкции CLINV и CLFLUSH из режима U-mode приводят к исключению «недопустимая инструкция»

Таблица 12.113 – Расширение M: умножение и деление

Мнемокод команды	Формат	Операнды	Краткое описание
1	2	3	4
MUL	R	rd, rs1, rs2	Умножение с возвратом младших 32 бит результата
MULH	R	rd, rs1, rs2	Умножение с возвратом старших 32 бит результата 64-битного произведения
MULHSU	R	rd, rs1, rs2	Умножение знакового числа на беззнаковое с возвратом старших 32 бит результата
MULHU	R	rd, rs1, rs2	Умножение беззнакового числа на беззнаковое с возвратом старших 32 бит результата
DIV	R	rd, rs1, rs2	Деление (знаковое). Округление происходит в сторону нуля
DIVU	R	rd, rs1, rs2	Беззнаковое деление
REM	R	rd, rs1, rs2	Остаток от деления (знаковая операция)
REMU	R	rd, rs1, rs2	Беззнаковый остаток от деления

Таблица 12.114 – Расширение A: атомарные операции с памятью

Мнемокод команды	Операнды	Краткое описание
1	2	3
LR.W	rd, rs1, rs2	Load Reserved Word. Загружает 32-битное слово из памяти по адресу rs1 в rd. Одновременно резервирует соответствующую область памяти, чтобы последующая инструкция SC.W могла проверить, не изменялась ли ещё эта ячейка. Используется только совместно с SC.W для реализации атомарных примитивов (lock‑free).
SC.W	rd, rs1, rs2	Store Conditional Word. Запись значения из rs2 по адресу в rs1. Запись происходит только если резервирование, установленное командой LR, всё еще актуально (т.е. никто другой не писал в эту память между LR и SC). В rd записывается код результата: 0 – запись прошла успешно, не 0 – запись не удалась.
AMOSWAP.W	rd, rs1, rs2	Atomic Swap. Атомарно меняет местами значение в памяти и регистре. Старое значение из памяти записывается в rd, а значение из rs2 записывается в память.
AMOADD.W	rd, rs1, rs2	Atomic Add. Атомарное сложение
AMOXOR.W	rd, rs1, rs2	Atomic XOR. Атомарное исключающее ИЛИ
AMOAND.W	rd, rs1, rs2	Atomic AND. Атомарное побитовое И
AMOOR.W	rd, rs1, rs2	Atomic OR. Атомарное побитовое ИЛИ
AMOMIN.W	rd, rs1, rs2	Atomic Minimum (Signed). Ищет минимальное число (знаковое). В память записывается меньшее из двух значений.
AMOMAX.W	rd, rs1, rs2	Atomic Maximum (Signed). Ищет максимальное число (знаковое).
AMOMINU.W	rd, rs1, rs2	Atomic Minimum (Unsigned). Аналогично AMOMIN, но числа интерпретируются как беззнаковые.
AMOMAXU.W	rd, rs1, rs2	Atomic Maximum (Unsigned). Аналогично AMOMAX, но числа интерпретируются как беззнаковые.

Таблица 12.115 – Расширение F: числа с плавающей запятой одинарной точности

Мнемокод команды	Операнды	Краткое описание
1	2	3
Загрузка/Сохранение:		
FLW	rd, rs1, imm	Загрузка слова с плавающей запятой
FSW	rs1, rs2, imm	Сохранение слова с плавающей запятой
Арифметические операции:		
FADD.S	rd, rs1, rs2	Сложение
FSUB.S	rd, rs1, rs2	Вычитание
FMUL.S	rd, rs1, rs2	Умножение
FDIV.S	rd, rs1, rs2	Деление
FSQRT.S	rd, rs1	Квадратный корень
FMIN.S/FMAX.S	rd, rs1, rs2	Минимум/Максимум
Комбинированные операции умножения-сложения:		
FMADD.S	rd, rs1, rs2, rs3	Вычисление rd = (rs1 * rs2) + rs3
FMSUB.S	rd, rs1, rs2, rs3	Вычисление rd = (rs1 * rs2) - rs3
FNMADD.S	rd, rs1, rs2, rs3	Вычисление rd = -(rs1 * rs2) - rs3
FNMSUB.S	rd, rs1, rs2, rs3	Вычисление rd = -(rs1 * rs2) + rs3
Сравнения:		
FLE.S	rd, rs1, rs2	Меньше или равно (устанавливает целочисленный флаг)
FLT.S	rd, rs1, rs2	Меньше (устанавливает целочисленный флаг)
FEQ.S	rd, rs1, rs2	Равно (устанавливает целочисленный флаг)
Преобразования:		
FCVT.W.S	rd, rs1	Преобразование float -> int (знаковое)
FCVT.WU.S	rd, rs1	Преобразование float -> int (беззнаковое)
FCVT.S.W	rd, rs1	Преобразование int -> float (знаковое)
FCVT.S.WU	rd, rs1	Преобразование int -> float (беззнаковое)
Перемещения и классификация:		
FMV.X.W	rd, rs1	Переместить биты float в целочисленный регистр
FMV.W.X	rd, rs1	Переместить биты из целочисленного регистра в float
FCLASS.S	rd, rs1	Классификация числа (получает класс числа, например, NaN, ±беск., ±норм., ±денорм., ±0)
Управление состоянием (CSR):		
FRCSR	rd	Чтение регистра F_{CSR} (регистр статуса и управления).
FSCSR	rd, rs1	Запись регистра F_{CSR} (регистр статуса и управления).
FRFLAGS	rd	Чтение накопленных флагов исключений (fflags) регистра F_{CSR}
FSFLAGS	rd, rs1	Запись fflags регистра F_{CSR}
FRRM	rd	Считывание поля режима округления (frm) регистра F_{CSR}
FSRM	rd, rs1	Запись поля режима округления (frm) регистра F_{CSR}

Таблица 12.116 – Расширение D: числа с плавающей запятой двойной точности

Мнемокод команды	Операнды	Краткое описание
1	2	3
Загрузка/Сохранение:		
FLD	rd, rs1, imm	Загрузка 64-битного значения с плавающей запятой
FSD	rs1, rs2, imm	Сохранение 64-битного значения с плавающей запятой
Арифметические операции:		
FADD.D	rd, rs1, rs2	Сложение
FSUB.D	rd, rs1, rs2	Вычитание
FMUL.D	rd, rs1, rs2	Умножение
FDIV.D	rd, rs1, rs2	Деление
FSQRT.D	rd, rs1	Квадратный корень
FMIN.D/FMAX.D	rd, rs1, rs2	Минимум/Максимум
Комбинированные операции умножения-сложения:		
FMADD.D	rd, rs1, rs2, rs3	Вычисление rd = (rs1 * rs2) + rs3
FMSUB.D	rd, rs1, rs2, rs3	Вычисление rd = (rs1 * rs2) - rs3
FNMADD.D	rd, rs1, rs2, rs3	Вычисление rd = -(rs1 * rs2) - rs3
FNMSUB.D	rd, rs1, rs2, rs3	Вычисление rd = -(rs1 * rs2) + rs3
Сравнения:		
FLE.D	rd, rs1, rs2	Меньше или равно (устанавливает целочисленный флаг)
FLT.D	rd, rs1, rs2	Меньше (устанавливает целочисленный флаг)
FEQ.D	rd, rs1, rs2	Равно (устанавливает целочисленный флаг)
Преобразования:		
FCVT.W.D	rd, rs1	Преобразование double -> int (знаковое)
FCVT.WU.D	rd, rs1	Преобразование double -> int (беззнаковое)
FCVT.D.W	rd, rs1	Преобразование int -> double (знаковое)
FCVT.D.WU	rd, rs1	Преобразование int -> double (беззнаковое)
Перемещения и классификация:		
FMV.2X.D	rd1(l), rd2(h), rs	Переместить биты double в два целочисленных регистра (нестандартная инструкция)
FMV.D.2X	rd, rs1(l), rs2(h)	Переместить биты из двух целочисленных регистров в double (нестандартная инструкция)
FCLASS.D	rd, rs1	Классификация 64-разрядного числа (получает класс числа, например, NaN, ±беск., ±норм., ±денорм., ±0)

Таблица 12.117 – Расширение C: сжатые инструкции (16-битные)

Мнемокод команды	Операнды	Краткое описание
1	2	3
Целочисленные:		
C.LW	rd	Загрузка 32-разрядного слова с плавающей запятой из памяти
C.SW	rs2	Сохранение 32-битного слова с плавающей запятой в память
C.LWSP	rd, offset(sp)	Загружает 32-битное значение из стека
C.SWSP	rs2, offset(sp)	Сохраняет 32-битное значение из регистра в стек
C.ADD	rd, rs2	Арифметическое сложение
C.ADDI	rd, imm	Складывает значение в rd с константой и записывает обратно в rd
C.ADDI16SP	imm	Добавление к sp (указателю стека) константы, кратной 16
C.ADDI4SPN	rd, imm	Вычисляет адрес sp + imm и сохраняет его в регистр rd
C.NOP	–	Нет операции. Не меняет состояние регистров и ячеек памяти
C.SUB	rd, rs2	Арифметическое вычитание
C.XOR	rd, rs2	Логическое исключающее «ИЛИ»
C.AND	rd, rs2	Логическое «И»
C.OR	rd, rs2	Логическое «ИЛИ»
C.MV	rd, rs2	Перемещение
C.JR	rs1	Безусловный переход по адресу в регистре rs1
C.JALR	rd, rs1	Позволяет переходить на требуемый адрес путём модификации программного счётчика PC, при этом сохраняя адрес возврата
C.EBREAK	–	Вызывает исключение для передачи управления отладчику
C.J	pc, offset	Безусловный переход относительно текущего PC, адрес возврата не сохраняется
C.JAL	pc, offset	Безусловный переход относительно текущего PC с сохранением адреса возврата
C.BEQZ	rs1, offset	Команда условного перехода, если rs1=0
C.BNEZ	rs1, offset	Команда условного перехода, если rs1≠0
C.SLLI	rd, imm	Логический сдвиг влево
C.SRLI	rd, imm	Логический сдвиг вправо
C.SRAI	rd, imm	Арифметический сдвиг вправо
C.LI	rd, imm	Загрузка малой знаковой константы в регистр
C.LUI	rd, imm	Загрузка константы в старшие биты регистра

Продолжение таблицы 12.117

1	2	3
С плавающей запятой:		
C.FLW	frd, offset(rs1)	Загрузка 32‑битного float из памяти в регистр
C.FSW	frs2, offset(rs1)	Запись 32‑битного float в память
C.FLWSP	frd, offset(sp)	Загрузка 32‑битного float из стека в регистр
C.FSWSP	frs2, offset(sp)	Запись 32‑битного float в стек
C.FLD	frd, offset(rs1)	Загрузка 64‑битного double из памяти
C.FSD	frs2, offset(rs1)	Запись 64‑битного double в память
C.FLDSP	frd, offset(sp)	Загрузка 64‑битного double из стека
C.FSDSP	frs2, offset(sp)	Запись 64‑битного double в стек

Таблица 12.118 – Расширение P (Packed SIMD/DSP): для цифровой обработки сигналов (DSP) и медиа-данных

Мнемокод команды	Операнды	Краткое описание
1	2	3
Инструкции по сложению и вычитанию:		
RADD16	rd, rs1, rs2	Округлённое усреднение. Используется для вычисления среднего по двум отсчётам.
KADD16	rd, rs1, rs2	Сложение 16-битных чисел со знаковой арифметикой насыщения
RSUB16	rd, rs1, rs2	16-разрядное делящее пополам вычитание со знаком
KSUB16	rd, rs1, rs2	16-битное знаковое вычитание с насыщением
RCRAS16	rd, rs1, rs2	16-разрядная знаковая кросс-операция (Cross-add-sub) с делением пополам
KCRAS16	rd, rs1, rs2	16-разрядная знаковая кросс-операция (Cross-add-sub) с насыщением
RCRSA16	rd, rs1, rs2	16-разрядная знаковая кросс-операция (Cross-sub-add) с делением пополам
KCRSA16	rd, rs1, rs2	16-разрядная знаковая кросс-операция (Cross-sub-add) с насыщением
RSTAS16	rd, rs1, rs2	16-разрядная знаковая операция (Straight-Add-Sub) с делением пополам
KSTAS16	rd, rs1, rs2	16-разрядная знаковая операция (Straight-Add-Sub) с насыщением
RSTSA16	rd, rs1, rs2	16-разрядная знаковая операция (Straight-Sub-Add) с делением пополам
KSTSA16	rd, rs1, rs2	16-разрядная знаковая операция (Straight-Sub-Add) с насыщением
RADDW	rd, rs1, rs2	32-разрядное знаковое сложение с делением пополам
RSUBW	rd, rs1, rs2	32-разрядное знаковое вычитание с делением пополам

Продолжение таблицы 12.118

1	2	3
Битовый сдвиг:		
SRAI16.u	rd, rs1	16-разрядный моментальный арифметический сдвиг вправо с округлением
Инструкции по перемещению данных:		
PACK	rd, rs1, rs2	Упаковать два 16‑битных значения из младших частей rs1 и rs2 в один 32-битный регистр
PKBT16	rd, rs1, rs2	Упаковать два 16-битных значения из разных частей регистров rs1 и rs2 в один 32-битный регистр (Bottom & Top)
PKTB16	rd, rs1, rs2	Упаковать два 16-битных значения из разных частей регистров rs1 и rs2 в один 32-битный регистр (Top & Bottom)
PACKU	rd, rs1, rs2	Упаковать два 16‑битных значения из старших частей rs1 и rs2 в один 32-битный регистр
REV	rd, rs1	Зеркально переворачивает биты в слове (XLEN)
Инструкции по умножению и накоплению:		
SMTT16	rd, rs1, rs2	Знаковое умножение старшей половины 16‑битного слова со старшей половиной другого 16‑битного слова
SMBT16	rd, rs1, rs2	Знаковое умножение младшей половины 16‑битного слова со старшей половиной другого 16‑битного слова
SMBB16	rd, rs1, rs2	Знаковое умножение младшей половины 16‑битного слова с младшей половиной другого 16‑битного слова
KMDA	rd, rs1, rs2	Знаковое умножение двух слов и сложение
KMXDA	rd, rs1, rs2	Знаковое перекрестное умножение двух слов и сложение
SMDS	rd, rs1, rs2	Знаковое умножение двух половин слов и вычитание
SMDRS	rd, rs1, rs2	Знаковое умножение двух половин слов
SMXDS	rd, rs1, rs2	Знаковое перекрестное умножение двух половин слов и вычитание
KMABB	rd, rs1, rs2	Знаковое умножение младших слов и сложение с насыщением
KMABT	rd, rs1, rs2	Знаковое умножение младшего и старшего слов и сложение с насыщением
KMATT	rd, rs1, rs2	Знаковое умножение старших слов и сложение с насыщением

Продолжение таблицы 12.118

1	2	3
KMADA	rd, rs1, rs2	Знаковое умножение двух слов и прибавление 2 с насыщением
KMAXDA	rd, rs1, rs2	Знаковое перекрестное умножение двух слов и прибавление 2 с насыщением
KMADS	rd, rs1, rs2	Знаковое умножение двух слов, вычитание и сложение с насыщением
KMADRS	rd, rs1, rs2	Знаковое умножение двух слов, реверсивное вычитание и сложение с насыщением
KMAXDS	rd, rs1, rs2	Знаковое перекрестное умножение двух слов, вычитание и сложение с насыщением
KMSDA	rd, rs1, rs2	Знаковое умножение двух слов, сложение и вычитание с насыщением
KMSXDA	rd, rs1, rs2	Знаковое перекрестное умножение двух слов, сложение и вычитание с насыщением
Арифметические инструкции с фиксированной запятой:		
KHM16	rd, rs1, rs2	Знаковое умножение с насыщением с фиксированной запятой Q15
KHMX16	rd, rs1, rs2	Знаковое перекрестное умножение с насыщением с фиксированной запятой Q15
Прочие инструкции:		
SCLIP32	rd, rs1, imm	Знаковая обрезка 32‑битного слова до диапазона ±(2^(imm−1))
SCLIP16	rd, rs1, imm	Знаковая обрезка каждого 16‑битного подслова до диапазона ±(2^(imm−1))
SMIN16	rd, rs1, rs2	Сравнение пары 16-битных знаковых чисел и запись в rd наименьшего из них
SMAX16	rd, rs1, rs2	Запись в rd наибольшего из пары 16-битных знаковых чисел
KABS16	rd, rs1	Модуль 16-битного числа

Подробное описание каждой команды и форматы команд приведены в документе «The RISC-V Instruction Set Manual Volume I: Unprivileged Architecture».

12.17 Режим работы ядер Lockstep

Общая информация

Критически важные для безопасности приложения во время исполнения требуют определение возникновения неисправности в основных компонентах микроконтроллера. Одним из способов для достижения необходимой защиты является режим работы «lockstep». В этом режиме одно ядро является рабочим, а второе выступает в качестве проверочного. Модуль сравнения определяет различие в выходных шинах ядер и посылает сигнал в модуль SIU в случае ошибки. Для диагностических целей, в модуле сравнения присутствует механизм самотестирования. Режим работы ядер определяются конфигурационным словом (см. п. 9.1 FlashM-память).

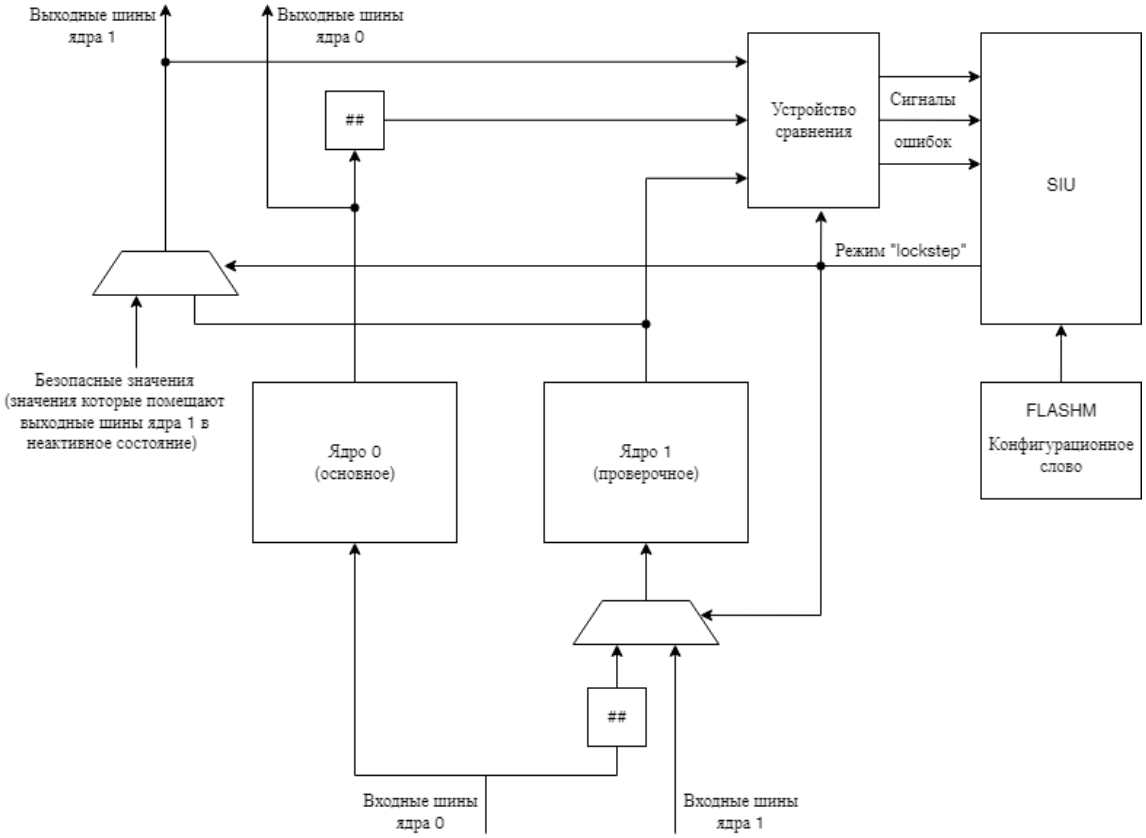


Рисунок 12.6 – Структурная схема

Выходные сигналы основного ядра задерживаются на два такта, в то время как входные сигналы для проверочного ядра так же задерживаются на два такта.

Устройство сравнения

Группы сигналов объединяются в узлы по общим признакам для удобного определения неисправностей:

– Узел IC (шины инструкций ядра и сигналы управления);
– Узел DC (шины данных ядра и сигналы управления);
– Узел L1I (шины и все сигналы управления к кэшу инструкций L1);
– Узел L1D (шины и все сигналы управления к кэшу инструкций LD);
– Узел SMP.

Выходные сигналы, которые не контролируются:

– Сигналы и шины PLIC;
– Сигналы и шины интерфейса к кэшу L2.

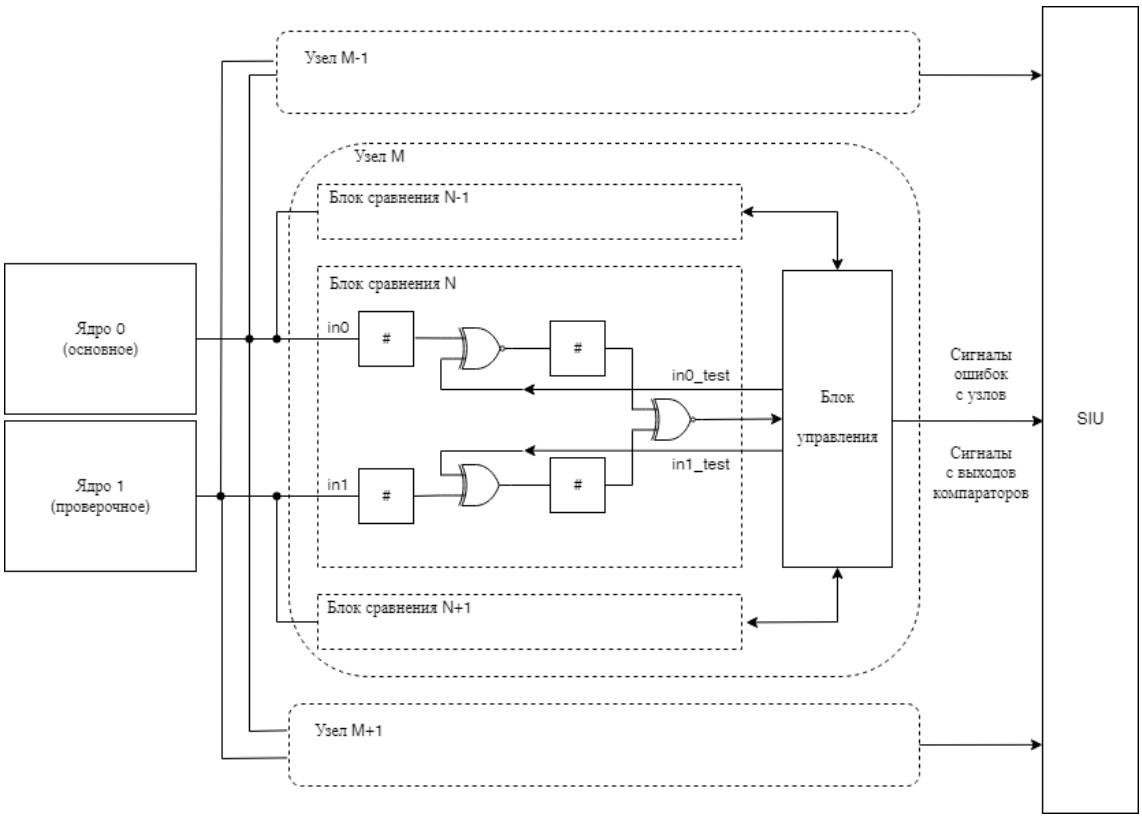


Рисунок 12.7 – Структурная схема узла

На рисунке 12.7 показано устройство узла. Он содержит определённое количество компараторов (число которых равно разрядности всех шин и сигналов управления подключённых к узлу) и блок управления. Компаратор наблюдает за двумя сигналами in0 и in1 которые являются выходными сигналами основного и проверочных ядер, и при наличии ошибки вырабатывает сигнал. Сигналы с выходов всех блоков сравнений поступают в блок управления. Блок управления собирает данные с компараторов, формирует общий сигнал ошибки и отправляет его на блок SIU, а также выполняет операцию самотестирования. Самотестирование заключается в принудительной инжекции ошибки по образцу «бегущая единица» на входы in0_test и in1_test для формирования ложного срабатывания компаратора. Если при самотестировании обнаружена ошибка, блок управления также формирует сигнал и отправляет его на блок SIU. Самотестирование может быть неудачным, если в момент подачи тестовой единицы на компаратор произойдёт реальный сбой.

Смена режима

В зависимости от битов SWMODEEN и LOCKSTEPEN указанных в конфигурационном слове разрешается смена режима работы ядер в режиме реального времени. Для этого, требуется записать в регистр CORE1_CONF «1» в поле DUALCORE, при этом произойдёт сброс микроконтроллера и установится бит LSTEP в блоке RCU регистре RSTSTAT.

13 Конфигурируемый контроллер прерываний уровня платформы PLIC

13.1 Структура контроллера PLIC

Контроллер PLIC содержит четыре приемника прерываний (T), 254 источника прерываний от периферийных блоков микроконтроллера и поддерживает 16 уровней приоритета для каждого из источников прерываний.

На рисунке 13.1 представлена блок схема контроллера PLIC.
Условные обозначения:
- G – модуль интерфейсный (interrupt gateway);
- S – источник прерывания (interrupt source);
- T – приёмник прерывания (interrupt target);
- H – аппаратный поток (hart).

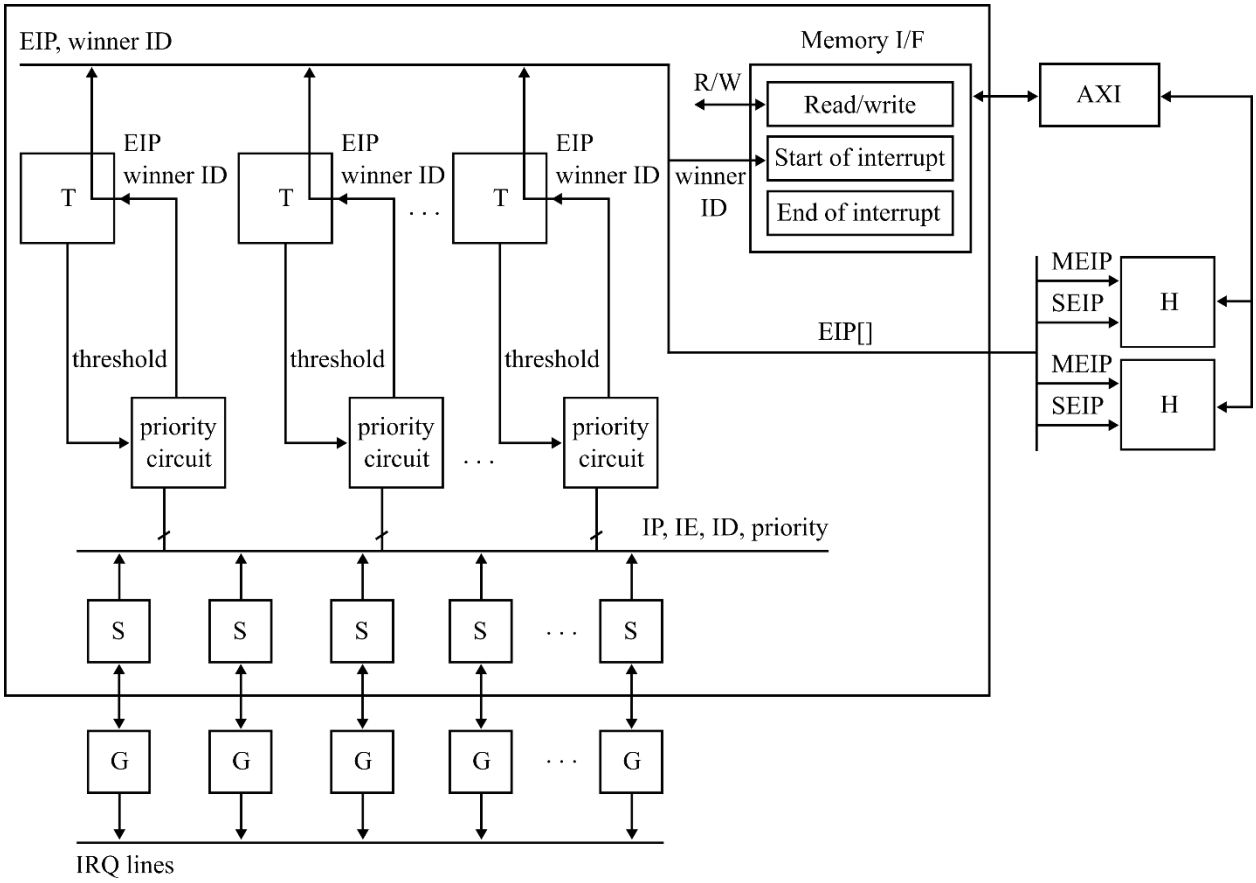


Рисунок 13.1 – Блок схема контроллера PLIC

Каждый приемник прерываний (T) отвечает за определенный режим привилегий:
– T0 – обрабатывает прерывания ядра 0 в режиме M-mode;
– T1 – обрабатывает прерывания ядра 0 в режиме S-mode;
– T2 – обрабатывает прерывания ядра 1 в режиме M-mode;
– T3 – обрабатывает прерывания ядра 1 в режиме S-mode.

13.2 Определения

Модуль интерфейсный – модуль, который подключает внешнюю линию прерывания и управляет битом ожидания прерывания внутри контроллера прерываний PLIC. Интерфейсная функция определяется установкой режима MODE источника прерывания. Модуль имеет в своем составе дополнительный блок синхронизации. Для каждого источника прерывания существует отдельный интерфейсный модуль interrupt gateway.

Источник прерывания представляет собой набор внутренних данных контроллера PLIC, которые определяют состояние прерывания. Каждому источнику прерывания назначается уникальный фиксированный идентификатор ID (номер вектора прерывания) в соответствии с таблицей 13.1.

Приёмником прерывания обычно выступает контекст аппаратного потока hart context (аппаратный поток, выполняющийся в определенном привилегированном режиме). С каждым приёмником прерывания связан отдельный набор внутренних данных контроллера PLIC.

13.3 Обзор функционирования

Цикл прерывания:

1 Состояние линии прерывания соответствует условию, определяемому настройкой режима MODE прерывания.

2 Устанавливается бит ожидания прерывания. Бит ожидания прерывания может быть очищен путем записи в соответствующий исходный регистр MODE или по событию «start of interrupt».

3 Устанавливается сигнал EIP для связанного приемника.

4 Запрос «Start of interrupt» обработчика прерываний вызывает переход в состояние «в обработке» («in service») ждущего прерывания с наивысшим приоритетом.

5 Когда обработчик завершает обработку прерывания, он отправляет сообщение «конец прерывания» («end of interrupt»).

13.4 Векторы прерываний

Каждый приёмник прерываний имеет EIP сигнал, который уведомляет о существовании разрешённого ждущего обработки прерывания. Сигнал EIP активируется только в случае, если значение приоритета ждущего прерывания строго выше установленного порога.

Таблица прерываний представляет собой перечень векторов, соответствующих определенным обработчикам прерываний, см. таблицу 13.1.

Таблица 13.1 – Таблица прерываний

Номер вектора	Обозначение	Описание
1	WDT	Прерывание блока сторожевого таймера
2	RCU	Прерывание блока RCU
3	RTC	Прерывание блока часов реального времени
4	FLASH	Прерывание контроллера Flash-памяти
5	SRAM	Прерывание контроллера внутренней памяти SRAM
6	L2_ERROR	Прерывание по ошибке кэш L2
7	SIU	Прерывание контроллера SIU
8	GPIOA	Прерывание порта GPIOA
9	GPIOB	Прерывание порта GPIOB
10	GPIOC	Прерывание порта GPIOC
11	GPIOD	Прерывание порта GPIOD
12	GPIOE	Прерывание порта GPIOE
13	GPIOF	Прерывание порта GPIOF
14	GPIOG	Прерывание порта GPIOG
15	ETH	Прерывание контроллера Ethernet
16	TMR0	Прерывание таймера 0 общее
17	TMR1	Прерывание таймера 1 общее
18	TMR2	Прерывание таймера 2 общее
19	TMR3	Прерывание таймера 3 общее
20	TMR4	Прерывание таймера 4 общее
21	TMR5	Прерывание таймера 5 общее
22	TMR6	Прерывание таймера 6 общее
23	TMR7	Прерывание таймера 7 общее
24	TMR8	Прерывание таймера 8 общее
25	TMR9	Прерывание таймера 9 общее
26	TMR10	Прерывание таймера 10 общее
27	TMR11	Прерывание таймера 11 общее
28	TMR12	Прерывание таймера 12 общее
29	TMR13	Прерывание таймера 13 общее
30	TMR14	Прерывание таймера 14 общее
31	TMR15	Прерывание таймера 15 общее

Продолжение таблицы 13.1

Номер вектора	Обозначение	Описание
32	DMA_CH0	Прерывание контроллера DMA по каналу 0
33	DMA_CH1	Прерывание контроллера DMA по каналу 1
34	DMA_CH2	Прерывание контроллера DMA по каналу 2
35	DMA_CH3	Прерывание контроллера DMA по каналу 3
36	DMA_CH4	Прерывание контроллера DMA по каналу 4
37	DMA_CH5	Прерывание контроллера DMA по каналу 5
38	DMA_CH6	Прерывание контроллера DMA по каналу 6
39	DMA_CH7	Прерывание контроллера DMA по каналу 7
40	DMA_CH8	Прерывание контроллера DMA по каналу 8
41	DMA_CH9	Прерывание контроллера DMA по каналу 9
42	DMA_CH10	Прерывание контроллера DMA по каналу 10
43	DMA_CH11	Прерывание контроллера DMA по каналу 11
44	DMA_CH12	Прерывание контроллера DMA по каналу 12
45	DMA_CH13	Прерывание контроллера DMA по каналу 13
46	DMA_CH14	Прерывание контроллера DMA по каналу 14
47	DMA_CH15	Прерывание контроллера DMA по каналу 15
48	DMA_CH16	Прерывание контроллера DMA по каналу 16
49	DMA_CH17	Прерывание контроллера DMA по каналу 17
50	DMA_CH18	Прерывание контроллера DMA по каналу 18
51	DMA_CH19	Прерывание контроллера DMA по каналу 19
52	DMA_CH20	Прерывание контроллера DMA по каналу 20
53	DMA_CH21	Прерывание контроллера DMA по каналу 21
54	DMA_CH22	Прерывание контроллера DMA по каналу 22
55	DMA_CH23	Прерывание контроллера DMA по каналу 23
56	DMA_CH24	Прерывание контроллера DMA по каналу 24
57	DMA_CH25	Прерывание контроллера DMA по каналу 25
58	DMA_CH26	Прерывание контроллера DMA по каналу 26
59	DMA_CH27	Прерывание контроллера DMA по каналу 27
60	DMA_CH28	Прерывание контроллера DMA по каналу 28
61	DMA_CH29	Прерывание контроллера DMA по каналу 29
62	DMA_CH30	Прерывание контроллера DMA по каналу 30
63	DMA_CH31	Прерывание контроллера DMA по каналу 31

Продолжение таблицы 13.1

Номер вектора	Обозначение	Описание
64	UART0	Общее прерывание блока UART0
65	UART0_TD	Прерывание UART0 по окончанию передачи
66	UART0_RX	Прерывание UART0 по заполнению FIFO
67	UART1	Общее прерывание блока UART1
68	UART1_TD	Прерывание UART1 по окончанию передачи
69	UART1_RX	Прерывание UART1 по заполнению FIFO
70	UART2	Общее прерывание блока UART2
71	UART2_TD	Прерывание UART2 по окончанию передачи
72	UART2_RX	Прерывание UART2 по заполнению FIFO
73	UART3	Общее прерывание блока UART3
74	UART3_TD	Прерывание UART3 по окончанию передачи
75	UART3_RX	Прерывание UART3 по заполнению FIFO
76	UART4	Общее прерывание блока UART4
77	UART4_TD	Прерывание UART4 по окончанию передачи
78	UART4_RX	Прерывание UART4 по заполнению FIFO
79	UART5	Общее прерывание блока UART5
80	UART5_TD	Прерывание UART5 по окончанию передачи
81	UART5_RX	Прерывание UART5 по заполнению FIFO
82	UART6	Общее прерывание блока UART6
83	UART6_TD	Прерывание UART6 по окончанию передачи
84	UART6_RX	Прерывание UART6 по заполнению FIFO
85	UART7	Общее прерывание блока UART7
86	UART7_TD	Прерывание UART7 по окончанию передачи
87	UART7_RX	Прерывание UART7 по заполнению FIFO
88	TMR0_CC0	Прерывание блока захвата/сравнения 0 TMR0
89	TMR0_CC1	Прерывание блока захвата/сравнения 1 TMR0
90	TMR0_CC2	Прерывание блока захвата/сравнения 2 TMR0
91	TMR0_CC3	Прерывание блока захвата/сравнения 3 TMR0
92	TMR0_CC4	Прерывание блока захвата/сравнения 4 TMR0
93	TMR0_CC5	Прерывание блока захвата/сравнения 5 TMR0
94	TMR0_CC6	Прерывание блока захвата/сравнения 6 TMR0
95	TMR0_CC7	Прерывание блока захвата/сравнения 7 TMR0
96	TMR1_CC0	Прерывание блока захвата/сравнения 0 TMR1
97	TMR1_CC1	Прерывание блока захвата/сравнения 1 TMR1
98	TMR1_CC2	Прерывание блока захвата/сравнения 2 TMR1
99	TMR1_CC3	Прерывание блока захвата/сравнения 3 TMR1
100	TMR1_CC4	Прерывание блока захвата/сравнения 4 TMR1
101	TMR1_CC5	Прерывание блока захвата/сравнения 5 TMR1
102	TMR1_CC6	Прерывание блока захвата/сравнения 6 TMR1
103	TMR1_CC7	Прерывание блока захвата/сравнения 7 TMR1

Продолжение таблицы 13.1

Номер вектора	Обозначение	Описание
104	TMR2_CC0	Прерывание блока захвата/сравнения 0 TMR2
105	TMR2_CC1	Прерывание блока захвата/сравнения 1 TMR2
106	TMR2_CC2	Прерывание блока захвата/сравнения 2 TMR2
107	TMR2_CC3	Прерывание блока захвата/сравнения 3 TMR2
108	TMR2_CC4	Прерывание блока захвата/сравнения 4 TMR2
109	TMR2_CC5	Прерывание блока захвата/сравнения 5 TMR2
110	TMR2_CC6	Прерывание блока захвата/сравнения 6 TMR2
111	TMR2_CC7	Прерывание блока захвата/сравнения 7 TMR3
112	TMR3_CC0	Прерывание блока захвата/сравнения 0 TMR3
113	TMR3_CC1	Прерывание блока захвата/сравнения 1 TMR3
114	TMR3_CC2	Прерывание блока захвата/сравнения 2 TMR3
115	TMR3_CC3	Прерывание блока захвата/сравнения 3 TMR3
116	TMR3_CC4	Прерывание блока захвата/сравнения 4 TMR3
117	TMR3_CC5	Прерывание блока захвата/сравнения 5 TMR3
118	TMR3_CC6	Прерывание блока захвата/сравнения 6 TMR3
119	TMR3_CC7	Прерывание блока захвата/сравнения 7 TMR3
120	SPI0	Общее прерывание контроллера SPI0
121	SPI0_RX	Прерывание SPI0 по заполнению FIFO
122	SPI0_TX	Прерывание SPI0 по опустошению FIFO
123	SPI1	Общее прерывание контроллера SPI1
124	SPI1_RX	Прерывание SPI1 по заполнению FIFO
125	SPI1_TX	Прерывание SPI1 по опустошению FIFO
126	SPI2	Общее прерывание контроллера SPI2
127	SPI2_RX	Прерывание SPI2 по заполнению FIFO
128	SPI2_TX	Прерывание SPI2 по опустошению FIFO
129	SPI3	Общее прерывание контроллера SPI3
130	SPI3_RX	Прерывание SPI3 по заполнению FIFO
131	SPI3_TX	Прерывание SPI3 по опустошению FIFO
132	SPI4	Общее прерывание контроллера SPI4
133	SPI4_RX	Прерывание SPI4 по заполнению FIFO
134	SPI4_TX	Прерывание SPI4 по опустошению FIFO
135	SPI5	Общее прерывание контроллера SPI5
136	SPI5_RX	Прерывание SPI5 по заполнению FIFO
137	SPI5_TX	Прерывание SPI5 по опустошению FIFO
138	SPI6	Общее прерывание контроллера SPI6
139	SPI6_RX	Прерывание SPI6 по заполнению FIFO
140	SPI6_TX	Прерывание SPI6 по опустошению FIFO
141	SPI7	Общее прерывание контроллера SPI7
142	SPI7_RX	Прерывание SPI7 по заполнению FIFO
143	SPI7_TX	Прерывание SPI7 по опустошению FIFO

Продолжение таблицы 13.1

Номер вектора	Обозначение	Описание
144	LIN0	Прерывание LIN0
145	LIN1	Прерывание LIN1
146	LIN2	Прерывание LIN2
147	LIN3	Прерывание LIN3
148	LIN4	Прерывание LIN4
149	LIN5	Прерывание LIN5
150	LIN6	Прерывание LIN6
151	LIN7	Прерывание LIN7
152	TMR0_PERIOD	Прерывание по достижении периода TMR0
153	TMR1_PERIOD	Прерывание по достижении периода TMR1
154	TMR2_PERIOD	Прерывание по достижении периода TMR2
155	TMR3_PERIOD	Прерывание по достижении периода TMR3
156	I2C0	Прерывание контроллера I2C0
157	I2C1	Прерывание контроллера I2C1
158	I2C2	Прерывание контроллера I2C2
159	I2C3	Прерывание контроллера I2C3
160	I2S0_TX	Прерывание I2S0 по опустошению FIFO
161	I2S0_RX	Прерывание I2S0 по заполнению FIFO
162	I2S1_TX	Прерывание I2S1 по опустошению FIFO
163	I2S1_RX	Прерывание I2S1 по заполнению FIFO
164	USB0	Прерывание контроллера USB0
165	USB1	Прерывание контроллера USB1
166	QSPI0	Общее прерывание контроллера QSPI0
167	QSPI1	Общее прерывание контроллера QSPI1
168	Reserved0	Зарезервировано
169	Reserved1	Зарезервировано
170	CANFD0	Прерывание контроллера CANFD0
171	CANFD0_TIMER	Прерывание таймера CANFD0
172	CANFD0_ECC	Прерывание CANFD0 при обнаружении одинарной или двойной ошибки
173	CANFD1	Прерывание контроллера CANFD1
174	CANFD1_TIMER	Прерывание таймера CANFD1
175	CANFD1_ECC	Прерывание CANFD1 при обнаружении одинарной или двойной ошибки
176	CANFD2	Прерывание контроллера CANFD2
177	CANFD2_TIMER	Прерывание таймера CANFD2
178	CANFD2_ECC	Прерывание CANFD2 при обнаружении одинарной или двойной ошибки
179	CANFD3	Прерывание контроллера CANFD3
180	CANFD3_TIMER	Прерывание таймера CANFD3
181	CANFD3_ECC	Прерывание CANFD3 при обнаружении одинарной или двойной ошибки

Продолжение таблицы 13.1

Номер вектора	Обозначение	Описание
182	CAN_0	Прерывание блока CAN, линия 0
183	CAN_1	Прерывание блока CAN, линия 1
184	CAN_2	Прерывание блока CAN, линия 2
185	CAN_3	Прерывание блока CAN, линия 3
186	CAN_4	Прерывание блока CAN, линия 4
187	CAN_5	Прерывание блока CAN, линия 5
188	CAN_6	Прерывание блока CAN, линия 6
189	CAN_7	Прерывание блока CAN, линия 7
190	CAN_8	Прерывание блока CAN, линия 8
191	CAN_9	Прерывание блока CAN, линия 9
192	CAN_10	Прерывание блока CAN, линия 10
193	CAN_11	Прерывание блока CAN, линия 11
194	CAN_12	Прерывание блока CAN, линия 12
195	CAN_13	Прерывание блока CAN, линия 13
196	CAN_14	Прерывание блока CAN, линия 14
197	CAN_15	Прерывание блока CAN, линия 15
198	ADC_SEQ0	Прерывание секвенсора 0 блока АЦП
199	ADC_SEQ1	Прерывание секвенсора 1 блока АЦП
200	ADC_SEQ2	Прерывание секвенсора 2 блока АЦП
201	ADC_SEQ3	Прерывание секвенсора 3 блока АЦП
202	ADC_SEQ4	Прерывание секвенсора 4 блока АЦП
203	ADC_SEQ5	Прерывание секвенсора 5 блока АЦП
204	ADC_SEQ6	Прерывание секвенсора 6 блока АЦП
205	ADC_SEQ7	Прерывание секвенсора 7 блока АЦП
206	ADC_DC	Прерывание компараторов блока АЦП
207	ACMP0	Прерывание аналогового компаратора 0
208	ACMP1	Прерывание аналогового компаратора 1
209	ACMP2	Прерывание аналогового компаратора 2
210	ACMP3	Прерывание аналогового компаратора 3
211	TRNG	Прерывание блока TRNG
212	HASH	Прерывание блока HASH
213	CRYPTO	Прерывание блока CRYPTO
214	RTC_ALARM1	Прерывание по флагу срабатывания будильника 1 RTC
215	RTC_ALARM2	Прерывание по флагу срабатывания будильника 2 RTC
216	RTC_BATTERY	Прерывание по флагу переключения на батарейное питание
217	RTC_N_POWER	Прерывание по флагу переключения на штатное питание
218	RTC_OVERFLOW	Прерывание по флагу переполнения регистра времени

Продолжение таблицы 13.1

Номер вектора	Обозначение	Описание
219	PWMx_TZ	Общее прерывание детекторов аварий блоков ШИМx
220	PWM0	Общее прерывание блока ШИМ0
221	PWM0_HD	Прерывание схемы удержания блока ШИМ0
222	PWM1	Общее прерывание блока ШИМ1
223	PWM1_HD	Прерывание схемы удержания блока ШИМ1
224	PWM2	Общее прерывание блока ШИМ2
225	PWM2_HD	Прерывание схемы удержания блока ШИМ2
226	PWM3	Общее прерывание блока ШИМ3
227	PWM3_HD	Прерывание схемы удержания блока ШИМ3
228	PWM4	Общее прерывание блока ШИМ4
229	PWM4_HD	Прерывание схемы удержания блока ШИМ4
230	PWM5	Общее прерывание блока ШИМ5
231	PWM5_HD	Прерывание схемы удержания блока ШИМ5
232	PWM6	Общее прерывание блока ШИМ6
233	PWM6_HD	Прерывание схемы удержания блока ШИМ6
234	PWM7	Общее прерывание блока ШИМ7
235	PWM7_HD	Прерывание схемы удержания блока ШИМ7
236	PWM8	Общее прерывание блока ШИМ8
237	PWM8_HD	Прерывание схемы удержания блока ШИМ8
238	PWM9	Общее прерывание блока ШИМ9
239	PWM9_HD	Прерывание схемы удержания блока ШИМ9
240	PWM10	Общее прерывание блока ШИМ10
241	PWM10_HD	Прерывание схемы удержания блока ШИМ10
242	PWM11	Общее прерывание блока ШИМ11
243	PWM11_HD	Прерывание схемы удержания блока ШИМ11
244	PWM12	Общее прерывание блока ШИМ12
245	PWM12_HD	Прерывание схемы удержания блока ШИМ12
246	PWM13	Общее прерывание блока ШИМ13
247	PWM13_HD	Прерывание схемы удержания блока ШИМ13
248	PWM14	Общее прерывание блока ШИМ14
249	PWM14_HD	Прерывание схемы удержания блока ШИМ14
250	PWM15	Общее прерывание блока ШИМ15
251	PWM15_HD	Прерывание схемы удержания блока ШИМ15
252	MSC	Общее прерывание контроллера MSC
253	MSC_upstream	Общее прерывание ведущего (upstream) MSC
254	MSC_downstream	Общее прерывание ведомого (downstream) MSC

13.5 Конфигурация прерываний

Карта регистров контроллера PLIC показана в таблице 13.2.

Таблица 13.2 – Карта регистров контроллера прерываний PLIC

Адрес	Обозначение	Описание
0xFE000004	SRC_PRI[1]	Регистр уровня приоритета для источника прерываний 1 (WDT)
0xFE000008	SRC_PRI[2]	Регистр уровня приоритета для источника прерываний 2 (RCU)
…		
0xFE0003F8	SRC_PRI[254]	Регистр уровня приоритета для источника прерываний 254 (MSC_downstream)
…		
0xFE001000	INT_PEND[0]	Регистр ждущих прерываний для источников 1 – 31. Бит 0 всегда считывается как 0
0xFE001004	INT_PEND[1]	Регистр ждущих прерываний для источников 32 – 63
…		
0xFE00101C	INT_PEND[7]	Регистр ждущих прерываний для источников 224 – 254
…		
0xFE002000	INTEN[0]	Регистр разрешения прерываний для источников 1 – 31. Бит 0 всегда считывается как 0
0xFE002004	INTEN[1]	Регистр разрешения прерываний для источников 32 – 63
…		
0xFE00201C	INTEN[7]	Регистр разрешения прерываний для источников 224 – 254
…		
0xFE1F0004	SRC_MODE[1]	Регистр MODE для источника прерываний 1 (WDT)
0xFE1F0008	SRC_MODE[2]	Регистр MODE для источника прерываний 2 (RCU)
…		
0xFE1F03F8	SRC_MODE[254]	Регистр MODE для источника прерываний 254 (MSC_downstream)
…		
0xFE200000	PRI_TRSHLD0	Регистр порога приоритета для приемника прерываний T0
0xFE200004	START_END0	Регистр начала/конца прерывания для приемника прерываний T0
0xFE201000	PRI_TRSHLD1	Регистр порога приоритета для приемника прерываний T1
0xFE201004	START_END1	Регистр начала/конца прерывания для приемника прерываний T1
0xFE202000	PRI_TRSHLD2	Регистр порога приоритета для приемника прерываний T2
0xFE202004	START_END2	Регистр начала/конца прерывания для приемника прерываний T2
0xFE203000	PRI_TRSHLD3	Регистр порога приоритета для приемника прерываний T3
0xFE203004	START_END3	Регистр начала/конца прерывания для приемника прерываний T3

Для каждого источника прерываний существует отдельный регистр SRC_PRI, позволяющий задать его приоритет. Контроллер прерываний поддерживает семь уровней

приоритета прерываний. Приоритеты прерываний нумеруются с первого (низкий) по седьмой (высокий). После сброса приоритет прерывания имеет нулевое значение, что означает прерывание запрещено. Поэтому перед разрешением прерывания, необходимо обязательно сконфигурировать его приоритет.

Текущее состояние битов ожидания источника прерывания в ядре PLIC может быть считано из массива, организованного в виде 32 слов по 32 бита. Бит ожидания для идентификатора прерывания ID N хранится в бите (N mod 32) слова (N / 32). Бит 0 слова 0, который представляет собой несуществующий источник прерывания 0, всегда аппаратно обнулён. Биты ожидания, расположенные в регистрах INT_PEND, доступны только для чтения. Бит ожидания в ядре PLIC очищается после запуска обработки прерывания (по событию «Start of interrupt») для соответствующего источника или после записи в регистр mode. Модуль gateway направляет новый запрос прерывания в ядро контроллера PLIC только после получения уведомления о том, что обработка предыдущего запроса прерывания из того же источника, завершена. Если источник прерывания, чувствительный к уровню, отключает прерывание после того, как ядро PLIC установит бит IP и прежде, чем прерывание будет обслужено, IP бит остается установленным и источник прерывания будет обслуживаться обработчиком, который затем должен будет определить, что устройство вызвавшее прерывание больше не требует обслуживания.

Регистр INTEN представляет собой битовую маску. Единица в i-ом бите указывает на разрешение отправки прерывания от (i − 1) источника в аппаратный поток исполнения; ноль – на запрет. После сброса PLIC значения всех битов в регистре INTEN равны 0. По готовности принимать прерывания, ПО должно разрешить обработку желаемых прерываний путем записи 1 в соответствующие биты регистра INTEN. Рекомендуется устанавливать в 1 только те биты INTEN, которые соответствуют действительно существующим источникам, подключенным к ядру, чтобы избежать появления прерываний от несуществующих источников.

Регистры SRC_MODE предназначены для настройки режима прерывания. Настройка режима для каждого источника прерывания определяет состояние линии прерывания, по которому устанавливается бит ожидания обработки прерывания. Запись в этот регистр очищает соответствующий бит ожидания обработки прерывания. Значение регистра после сброса равно 0. В таблице 13.3 приведены значения регистра SRC_MODE. При конфигурации прерываний в соответствующие регистры SRC_MODE должно быть записано значение 0x1, соответствующее прерыванию по высокому уровню. Для выключения прерывания в регистр SRC_MODE, соответствующий источнику прерывания, необходимо записать значение 0.

Таблица 13.3 – Значения регистра SRC_MODE контроллера прерываний PLIC

Значение SRC_MODE	Обозначение	Описание
0x0	off	Прерывание выключено
0x1	high level	Прерывание по высокому уровню
0x2	low level	Прерывание по низкому уровню
0x3	rising edge	Прерывание по положительному фронту сигнала
0x4	falling edge	Прерывание по отрицательному фронту сигнала
0x5	dual edge	Прерывание по положительному или отрицательному фронту сигнала

Регистр порога приоритета PRI_TRSHLD определяет уровень приоритета прерывания, который может привести к активизации сигнала EIP, если источник прерывания находится в ожидании и включен. Значение по умолчанию в регистре PRI_TRSHLD равно нулю.

Чтение регистра начала/конца прерывания START_END возвращает ID ожидающего прерывания с наивысшим приоритетом, либо 0, если нет ожидающих прерываний. Успешный запуск прерывания также приводит к атомарной очистке соответствующего бита ожидающих прерываний источника прерываний.

Запись в регистр начала/конца прерывания приводит к завершению прерывания при выполнении двух условий:

– Записанное значение соответствует идентификатору источника прерывания, который в данный момент находится в работе;

– Этот источник прерывания включен для данного приемника прерываний.

13.6 Программная модель обработки внешних прерываний

Источник прерываний (interrupt source) через PLIC сигнализирует в ядро о наличии прерывания. После выполнения необходимых аппаратных проверок о допустимости выполнения прерываний и выбора наиболее приоритетного прерывания (при готовности нескольких прерываний), PLIC формирует сигнал int_meip к ядру, указывающий на наличие прерывания, готового к обработке. Сигнал int_meip подключается в CSR регистр mip в бит MEIP (machine external interrupt pending, бит 11).

Если прерывания не запрещены (в регистре mie установлен бит MEIE и в регистре mstatus установлен бит MIE), управление передается в обработчик прерывания (выполняется переход на адрес обработчика прерываний), а в регистр mcause записывается «причина», т.е. тип источника, вызвавшего прерывание. Для внешних прерываний тип устанавливается в значение Machine external interrupt.

Для «взятия» внешнего прерывания на исполнение необходимо выполнить команду чтения по адресу MMR-регистра ICC. Эта транзакция сигнализирует в PLIC о том, что аппаратный поток принял прерывание к исполнению. После завершения обработки прерывания необходимо выполнить команду записи по адресу регистра ICC, что сигнализирует в PLIC о том, что обработка прерывания завершена.

Помимо использования модели обработки прерываний, описанной выше, для отслеживания наличия прерываний может быть использована модель опроса (poll model), в которой вместо «ожидания» появления прерывания и «запроса» исполнения программы в обработчик прерывания, ПО может самостоятельно проверять наличие запросов на прерывания путем чтения регистра IPM. ПО также может выполнять «взятие» прерывания на обработку путем чтения регистра ICC.

При использовании данной модели программирования надо учитывать, что чтение IPM не гарантирует, что последующее чтение ICC вернет прерывание с тем же номером. Последующее чтение ICC может вернуть прерывание с тем же номером или с другим номером, а также вернуть номер 0, указывающий на отсутствие прерывания.

14 Контроллер прямого доступа к памяти DMA

DMA передает данные между различными точками в адресном пространстве без вмешательства центрального процессора. DMA обычно используется для замены двух функций процессора: копирования в память и управления периферийными устройствами (обычно имеющих невысокую скорость приемопередачи, таких как SPI, UART и т.д.).

Управляющие регистры контроллера DMA приведены в приложении А.8.

Основные свойства и отличительные особенности:
- 32 канала контроллера прямого доступа к памяти (DMA);
- арбитраж с приоритетом;
- настраиваемый контроллер прерываний;
- поддержка любого выравнивания адресов;
- поддержка выравнивания любого размера буфера передачи;
- поддержка связанных списков дескрипторов;
- поточное управление периферией;
- передача между периферией;
- изменение порядка следования байт;
- программное управление периферийными запросами;
- сторожевой таймер;
- приостановка и возобновление работы канала;
- полный набор регистров состояния для отладки.

Аппаратные источники запросов каналов контроллера DMA указаны в таблице 14.1.

Таблица 14.1 – Аппаратные источники запросов каналов контроллера DMA

Номер	Аппаратный источник запросов	Описание
0	MEMORY	Источник – память (по умолчанию)
1	SPI0	Запрос модуля последовательного интерфейса SPI0
2	SPI1	Запрос модуля последовательного интерфейса SPI1
3	SPI2	Запрос модуля последовательного интерфейса SPI2
4	SPI3	Запрос модуля последовательного интерфейса SPI3
5	SPI4	Запрос модуля последовательного интерфейса SPI4
6	SPI5	Запрос модуля последовательного интерфейса SPI5
7	SPI6	Запрос модуля последовательного интерфейса SPI6
8	SPI7	Запрос модуля последовательного интерфейса SPI7
9	UART0	Запрос модуля приемопередатчика UART0
10	UART1	Запрос модуля приемопередатчика UART1
11	UART2	Запрос модуля приемопередатчика UART2
12	UART3	Запрос модуля приемопередатчика UART3
13	UART4	Запрос модуля приемопередатчика UART4
14	UART5	Запрос модуля приемопередатчика UART5
15	UART6	Запрос модуля приемопередатчика UART6
16	UART7	Запрос модуля приемопередатчика UART7
17	TMR0	Запрос таймера TMR0
18	TMR1	Запрос таймера TMR1
19	TMR2	Запрос таймера TMR2
20	TMR3	Запрос таймера TMR3
21	TMR4	Запрос таймера TMR4
22	TMR5	Запрос таймера TMR5

Продолжение таблицы 14.1

Номер	Аппаратный источник запросов	Описание
23	TMR6	Запрос таймера TMR6
24	TMR7	Запрос таймера TMR7
25	TMR8	Запрос таймера TMR8
26	TMR9	Запрос таймера TMR9
27	TMR10	Запрос таймера TMR10
28	TMR11	Запрос таймера TMR11
29	TMR12	Запрос таймера TMR12
30	TMR13	Запрос таймера TMR13
31	TMR14	Запрос таймера TMR14
32	TMR15	Запрос таймера TMR15
33	ADCSEQ0	Запрос секвенсора 0 блока АЦП
34	ADCSEQ1	Запрос секвенсора 1 блока АЦП
35	ADCSEQ2	Запрос секвенсора 2 блока АЦП
36	ADCSEQ3	Запрос секвенсора 3 блока АЦП
37	ADCSEQ4	Запрос секвенсора 4 блока АЦП
38	ADCSEQ5	Запрос секвенсора 5 блока АЦП
39	ADCSEQ6	Запрос секвенсора 6 блока АЦП
40	ADCSEQ7	Запрос секвенсора 7 блока АЦП
41	RESERVED0	Зарезервировано
42	RESERVED1	Зарезервировано
43	RESERVED2	Зарезервировано
44	RESERVED3	Зарезервировано
45	GPIOA	Запрос порта GPIOA
46	GPIOB	Запрос порта GPIOB
47	GPIOC	Запрос порта GPIOC
48	GPIOD	Запрос порта GPIOD
49	GPIOE	Запрос порта GPIOE
50	GPIOF	Запрос порта GPIOF
51	GPIOG	Запрос порта GPIOG
52	HASH	Запрос блока HASH
53	PWM0	Запрос блока PWM0
54	PWM1	Запрос блока PWM1
55	PWM2	Запрос блока PWM2
56	PWM3	Запрос блока PWM3
57	PWM4	Запрос блока PWM4
58	PWM5	Запрос блока PWM5
59	PWM6	Запрос блока PWM6
60	PWM7	Запрос блока PWM7
61	PWM8	Запрос блока PWM8
62	PWM9	Запрос блока PWM9
63	PWM10	Запрос блока PWM10
64	PWM11	Запрос блока PWM11
65	PWM12	Запрос блока PWM12
66	PWM13	Запрос блока PWM13
67	PWM14	Запрос блока PWM14
68	PWM15	Запрос блока PWM15

Продолжение таблицы 14.1

Номер	Аппаратный источник запросов	Описание
69	DAC0	Запрос ЦАП DAC0
70	DAC1	Запрос ЦАП DAC1
Примечание – Каждый из периферийных блоков имеет линии запросов как RX, так и TX. Исключения составляют: – ADCSEQn – только RX запросы; – HASH – только TX запросы; – DACn – только TX запросы.		

Схема формирования запросов к каналам контроллера DMA приведена на рисунке 14.1. Для передачи данных должны быть сформированы как запрос на чтение (RX), так и запрос на запись (TX). Источником запроса может быть периферийный блок, память (MEMORY) и программное управление с помощью регистров PER_TXn_CTRL / PER_RXn_CTRL. Запросы RX и TX от памяти (MEMORY) формируются автоматически.

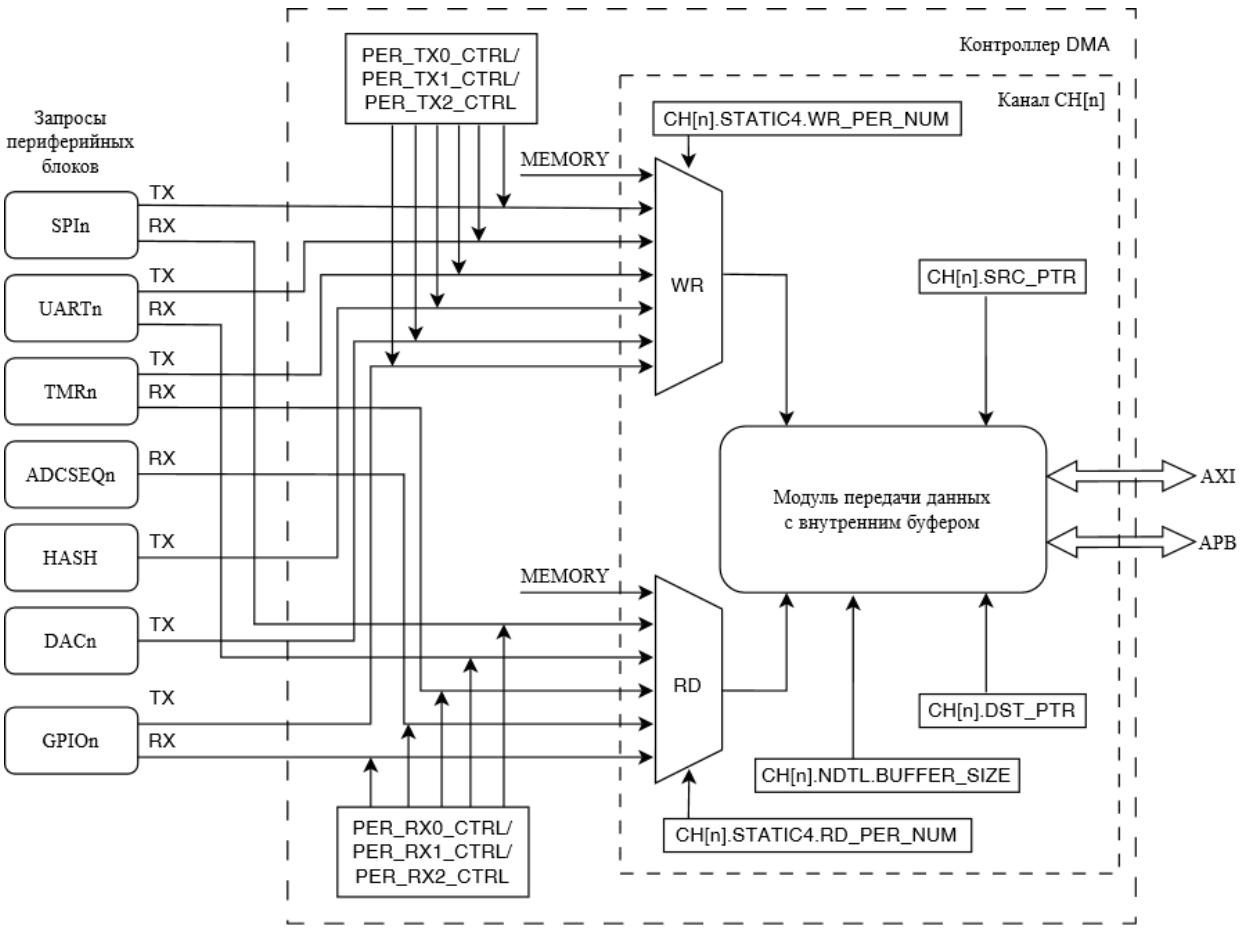


Рисунок 14.1 – Формирование запросов к каналам DMA

14.1 Режимы работы

Режим работы определяет, как каналы используют шины чтения и записи AXI. Если для контроллера DMA выбран совместный режим работы (установлен бит JOINT_MODE в регистре CONTROL), то каждый канал может настроен на работу в обычном (независимом) или в совместном режиме (при установленном бите JOINT в регистре CH[n].STATIC2).

Обычный режим

Реализован независимый арбитраж операций чтения и записи. Операции чтения на шине AXI не будут напрямую блокироваться ожиданием завершения операций записи.

Каждый канал будет работать следующим образом:

1) Канал вычисляет размеры следующих пакетов чтения и записи в соответствии с AXI, адресом и программными ограничениями;

2) Канал будет выполнять пакетное чтение до тех пор, пока внутренний буфер данных не будет заполнен или до получения полного запрашиваемого объема данных;

3) Канал будет выполнять пакетную запись до тех пор, пока данные для нее присутствуют во внутреннем буфере.

Этот режим наиболее эффективен, когда:

- пакеты чтения и записи каналов имеют разный размер (например, при передаче данных с периферийных устройств на DRAM);
- ведомые устройства на шине AXI имеют длительное или непредсказуемое время отклика;
- несколько каналов работают одновременно, и главной целью является общая производительность.

Преимущества:

- максимальная общая производительность для вышеуказанных случаев.

Недостатки:

- канал не начнет пакетную запись до тех пор, пока не поступят считанные данные, это может приводить к большой задержке, ограниченность объема внутреннего буфера данных может негативно сказываться на пропускной способности;
- при использовании адресов чтения или записи, которые не выровнены по размеру внутреннего буфера данных (64 байт), максимальная длина пакетных доступов по шине AXI будет составлять половину размера буфера данных.

Чтобы достичь максимальной пропускной способности независимо от выравниваний, контроллер начнет с одиночных доступов по шине AXI до достижения выровненного адреса (в соответствии с разрядностью передаваемых данных), откуда возможны пакетные доступы с размером, соответствующим ширине шины AXI (32 бита).

При расчете длины пакета учитывается следующее:

- не пересекает границу адреса в 4 Кбайта (ограничение AXI);
- не превышает значения, установленного в регистре BURST_MAX_SIZE;
- не превышает размер FIFO;
- ширина пакета должна быть выровнена по адресу пакета.

Совместный режим

При выборе этого режима работы будет использован один арбитр как для операций чтения, так и для операций записи. Этот режим используется для каналов, которые выполняют операции чтения и записи в одинаковом темпе. Текущий канал занимает шину AXI как на чтение, так и на запись и выполняет прямую передачу считанных данных для записи. При работе в этом режиме настройка канала выполняется только в регистрах чтения, и они влияют как на операции чтения, так и на операции записи.

Каждый канал будет работать следующим образом:

1) Канал будет считывать отдельные пакеты до тех пор, пока не достигнет согласованного адреса, с которого он сможет выполнять запрошенные пакеты;

2) Канал будет записывать одиночные пакеты до тех пор, пока не достигнет согласованного адреса, с которого он сможет выполнять запрошенные пакеты;

3) Канал переходит в стадию совместного использования, на этой стадии канал будет считывать и записывать одновременно по обеим шинам AXI;

4) Канал будет выполнять одиночные пакеты, чтобы завершить работу с оставшимися последними байтами.

Этот режим наиболее эффективен, когда:

- ведомые устройства на шине AXI чтения и записи работают пакетами одинакового размера (обязательно).
- у ведомого устройства на шине AXI нет большой задержки в выполнении доступов (циклы ожидания на шине чтения останавливают шину записи и наоборот).
- необходимо передать большой объем данных.

Совместный режим не подходит для обслуживания периферийных устройств.

Преимущества:

- уменьшенная задержка, повышенная производительность.
- максимальный размер пакета не ограничен независимо от выравниваний и размера внутреннего буфера данных. Максимальная поддерживаемая длина пакета составляет 16 доступов (64 байта для 32 разрядной шины данных).

Недостатки:

- хотя использование одного и того же канала в совместном и обычном режимах возможно, они не является эффективным, поскольку переходе с совместного режима работы канала на обычный выполняется через стадию очистки;
- каналы, которые работают в обычном режиме, по-прежнему могут использовать только одни арбитр, если контроллер DMA переведен в совместный режим.

В совместном режиме каждый канал будет выдавать одиночные пакеты чтения и записи до тех пор, пока не станет возможным выдавать пакеты, равные по длине значению поля BURST_MAX. Затем канал перейдет в совместную фазу, выполняя одновременные пакетные чтения и записи. В конце, если во внутреннем буфере всё еще остаются данные, канал снова переходит на передачу пакетов меньшего размера.

Примечание – При использовании совместного режима размер внутреннего буфера канала (глубина FIFO составляет 8 32 битных записей) не накладывает ограничений на длину пакетных доступов по шине AXI. Но может ограничивать длину пакетов на начальной и конечной стадиях канала (если совместные пакетные передачи невозможны).

Примечание – Пакетные доступы, которые пересекают границу в 4 Кбайт, запрещены спецификацией шины AXI. Совместный режим будет отключен, когда передача дойдет до этого адреса, будь то чтение или запись. Такая пакетная запись будет разделена надвое. Совместный режим будет возобновлен после пересечения границы.

14.1 Дескрипторы DMA

Дескриптор DMA состоит из четырех 32-разрядных полей. Каждый канал хранит текущий дескриптор в своих регистрах SRC_PTR, DST_PTR, NDTL, CONFIG. Дескриптор канала может быть настроен непосредственно с помощью записи упомянутых выше регистров по шине APB3 или канал может загрузить дескриптор, считав его через запрос по шине AXI. Поля дескриптора DMA описаны в приложении А.8 «Регистры контроллера прямого доступа к памяти DMA».

14.2 Цепочки дескрипторов DMA

Цепочки дескрипторов DMA – это связанные списки дескрипторов, которые могут быть размещены в любом месте памяти. Когда канал завершит выполнение своего текущего дескриптора, если поле CMD_LAST регистра CH[n].CONFIG равно 0, блок управления каналом считает следующий дескриптор по шине AXI с адреса, указанного в поле NEXT_ADDR регистра CH[n].CONFIG. Если значение CMD_LAST равно 1, канал остановится.

Первый дескриптор может быть записан непосредственно в регистры дескриптора канала, но предпочтительней записать всю цепочку дескрипторов в память и установить в текущем дескрипторе NEXT_ADDR, который указывает на начало списка, выставив нулевой размер буфера передачи данных.

Последовательность настройки:

1) Запишите цепочку дескрипторов в память;

2) В регистре CH[n].NDTL установите BUFFER_SIZE = 0;

3) В регистре CH[n].CONFIG установите CMD_SET_INT = 0, CMD_LAST = 0 и NEXT_ADDR = адресу первого дескриптора в памяти.

Примечание – NEXT_ADDR должен быть выровнен по размеру дескриптора (16 байт).

Списки команд могут использоваться для двух целей:

1) Разборка – Сборка. Когда система выделяет большие блоки памяти, память является непрерывной в виртуальном адресном пространстве, но ограниченной в физическом адресном пространстве. Список пар адрес-размер в процессе распределения может быть записан в память в виде списка команд DMA, позволяющих DMA передавать все эти фрагменты данных без вмешательства центрального процессора.

Пример – Распределенный список, представляющий собой блок памяти объемом 20 КБ, фрагментированный на пять страниц объемом по 4 КБ. После записи последней страницы в списке указано выдать процессору прерывание по завершению.

2) Циклические буферы для управления периферией. При обслуживании периферийных устройств рекомендуется использовать, по крайней мере, двойные буферы для хранения периферийных RX или TX-данных. Циклические буферы можно легко настроить, создав зацикленный список команд.

14.1 Запросы от периферийных модулей

DMA является очень простым и эффективным способом обслуживания периферийных устройств с минимальным вмешательством центрального процессора. Это особенно полезно при работе с медленными устройствами. Обычно периферийные устройства являются либо приемными, либо передающими и содержат FIFO фиксированной ширины.

Конфигурация канала для периферийного управления (данный пример для принимающего периферийного устройства, передающее периферийное устройство конфигурируется точно так же, просто замените RX на TX, а RD на WR):

1) Настройте канал на выдачу пакетов, соответствующих функциональности его периферийного устройства. Запишите количество байт в каждом пакете в поле RD_BURST_MAX регистра CH[n].STATIC0 для RX (STATIC1 для TX). Если требуется, измените значение поля RD_TOKENS, определяющее максимальное количество последовательных пакетов, которые будут выдаваться при каждом обслуживании периферийного устройства.

2) Установите номер периферийного устройства в поле RD_PER_NUM регистра для RX. Если периферийному блоку на сброс отработанного запроса требуется больше 1 такта,

то занесите соответствующее значение в поле RD_PER_DELAY (оказывает эффект только при значении RD_TOKENS больше 1).

14.2 Передача между периферийными устройствами

Возможна передача с периферийного устройства на периферийное устройство. Настраивается как обычная передача с периферийного устройства, устанавливая периферийные регистры, как для чтения, так и для записи. Нет никаких ограничений в отношении различных размеров пакетов для периферийных устройств RX и TX-передачи.

14.1 Арбитраж

Арбитраж каждого канала может происходить в двух режимах: независимом для чтения и записи и совместном. Режим арбитража задаётся в регистре CONTROL. В независимом режиме DMA использует два отдельных арбитра для операций чтения и записи. В совместном режиме, арбитраж происходит с использованием одного арбитра.

Контроллер DMA имеет настройки, которые определяют количество передач по шине AXI до повторения арбитража (переарбитража). Это значение задается полями RD_TOKENS для чтения в регистре STATIC0 и WR_TOKENS для записи в регистре STATIC1. Количество пакетных транзакций по шине AXI одного канала до переарбитража при этом равно значениям в этих полях.

Приоритет

При проведении арбитража определяется канал для обслуживания в следующем цикле DMA. На выбор следующего канала влияют:

- номер канала;
- уровень приоритета, присвоенного каналу.

Каждому каналу может быть присвоен уровень приоритета по умолчанию или высокий. Изменение уровня приоритета осуществляется установкой соответствующего бита CHi (i – номер канала) в регистрах RD_PRIORITY и WR_PRIORITY.

После окончания транзакции чтения/записи DMA контроллер выбирает следующий для обслуживания канал из всех включенных каналов. Рисунок 14.2 иллюстрирует процесс выбора следующего канала для обслуживания.

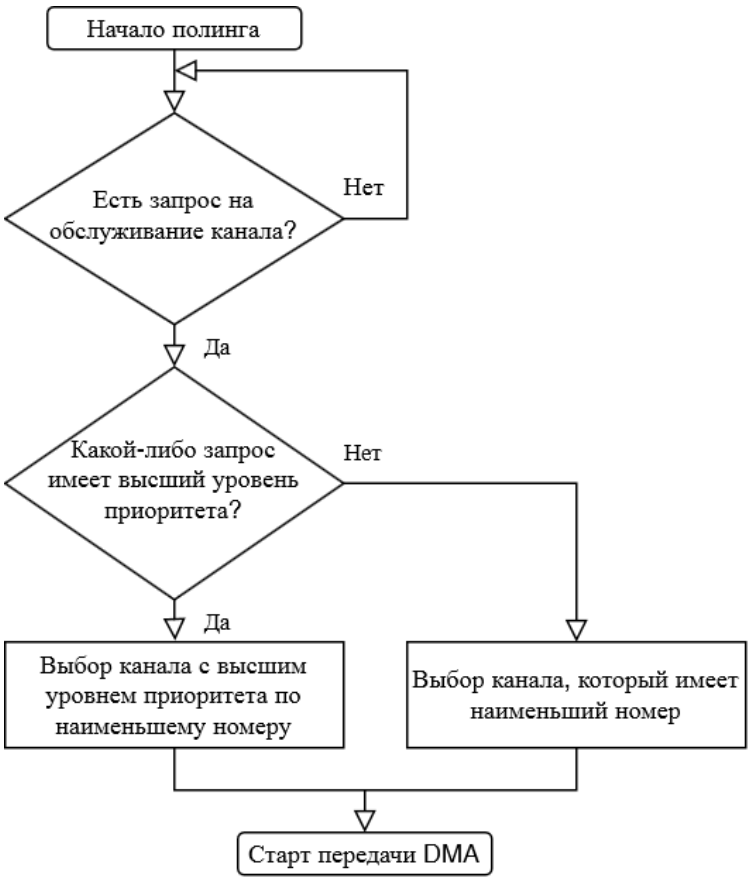


Рисунок 14.2 – Алгоритм выбора (полинга) следующего канала для обслуживания

14.2 Глубина прерывания

Не редка ситуация, когда ядру может потребоваться много времени, прежде чем очередь выполнения дойдет до кода обработки прерывания. В таких случаях канал может успеть выполнить несколько дескрипторов до того, как процессор получит возможность обработать прерывание завершения работы канала. Для корректной обработки подобных ситуаций в поле INT_COUNT регистра COUNT сохраняется количество необслуженных прерываний. После очистки прерывания завершения работы канала, если имеются ожидающие прерывания (INT_COUNT > 1), прерывание завершения работы канала будет установлено повторно.

14.3 Программное управление периферийным запросом

Каналом можно управлять непосредственно программным способом, задействовав механизм управления периферийными запросами (см. рисунок 14.1).

Формирование запроса записи (TX):

1) Для работы канала в поле WR_PER_NUM регистра STATIC4 записывается номер периферийного блока, при этом поле RD_PER_NUM содержит 0.

2) Программное обеспечение записывает данные в область памяти, откуда они будут считаны (например, область ОЗУ).

3) DMA вычитывает данные в буфер данных канала, но не записывает их.

4) Фактическая запись в регистры периферийного блока будет осуществлена только после установки соответствующего бита в регистре PER_TXn_CTRL (формирование TX-запроса).

5) DMA записывает данные, очищая соответствующий бит в регистре PER_TXn_CTRL.

Формирование запроса чтения (RX):

1) Для работы канала в поле RD_PER_NUM регистра STATIC4 записывается номер периферийного блока, при этом поле WR_PER_NUM содержит 0.

2) Когда регистры периферийного модуля готовы к считыванию, программное обеспечение устанавливает соответствующий бит в регистре PER_RXn_CTRL (формирование RX-запроса).

3) DMA копирует данные из регистра периферийного блока по указанному адресу, очищая соответствующий бит в регистре PER_RXn_CTRL.

14.4 Тайм-ауты AXI

Тайм-ауты могут выдаваться на всех пяти шинах AXI (AW, W, B, AR, R). В случае, если какая-либо из этих шин не отвечает в течение 1024 тактов, соответствующий канал выдаст прерывание по тайм-ауту. Таким образом, прерывание не только указывает на то, что ведомое устройство не отвечает, но и указывает, какой канал отправил запрос.

14.5 Сторожевой таймер

Сторожевой таймер проверяет каждый активный канал (включенный и не завершенный) в циклическом порядке. Если проверяемый канал не начнет работать в течение 2048 циклов, по нему будет выдано прерывание по тайм-ауту.

14.6 Ограничение тактирования

Для снижения динамической потребляемой мощности реализовано отключение тактирования каналов, для которых не установлен соответствующий номеру канала бит в регистре CH_ENABLE (после сброса ни один из каналов не тактируется). Запись в регистры отключенного канала игнорируется, чтение возвращает 0.

14.1 Ограничение ожидающих выполнения команд AXI

Количество одновременных команд AXI, которое разрешается выставить каналу до получения ответов на предыдущие запросы, может быть ограничено с помощью полей RD_OUTS_MAX и WR_OUTS_MAX регистров STATIC0 и STATIC1. Рекомендуется ограничить это значение для каналов с низким приоритетом.

14.2 Изменение порядка следования байт

Каждый канал может манипулировать записываемыми данными для поддержки little endian / big endian портов. Перестановка байтов настраивается в поле END_SWAP регистра CH[n].STATIC2. Поддерживается перестановка байтов в пределах 16 или 32-битных данных.

Ограничения

Следующие параметры должны быть выровнены в соответствии с размером перестанавливаемых байтов: RD_ADDR, WR_ADDR, BUFFER_SIZE, FRAME_WIDTH.

Пример – Если значение END_SWAP равно 1 (перестановка в пределах 16 бит), приведенные выше параметры должны быть выровнены по границе в 16 бит.

14.1 Логгер ошибочных передач

Каждый канал содержит логгер ошибок по AXI шине. При их возникновении или по тайм-ауту начальный адрес пакетных транзакции на чтение или запись будет занесен в регистры CH[n].ERR_RD_ADDR и CH[n].ERR_WR_ADDR соответственно. Выбор события для логирования осуществляется в регистре CH[n].ERR_STOP. В случае настройки цепочки дескрипторов и возникновении ошибки, исполнение текущего дескриптора прекращается и автоматически происходит переход к следующему.

Таблица 14.2 – Регистры логгера ошибок

Имя регистра	Описание			
ERR_STOP	Позиция	Имя поля	Тип	Описание
	0	Reserved	RO	Зарезервировано
	1	RD_SLVERR	RW	Если установлен, останавливает выполнение транзакций DMA на чтение и запись при возникновении этого события
	2	WR_SLVERR	RW	
	3	RD_DECERR	RW	
	4	WR_DECERR	RW	
	5	OVERFLOW	RW	Не реализовано
	6	UNDERFLOW	RW	Не реализовано
	7	TIMEOUT_R	RW	Если установлен, останавливает выполнение транзакций DMA на чтение и запись при возникновении этого события
	8	TIMEOUT_AR	RW	
	9	TIMEOUT_B	RW	
	10	TIMEOUT_W	RW	
	11	TIMEOUT_AW	RW	
	12	WDT_TIMEOUT	RW	
	31:13	Reserved	RO	Зарезервировано
ERR_RD_ADDR	Захватывается адрес на чтение в AXI burst транзакции во время которой произошло одно из событий, флаг которого установился в CH[n].INT_RAWSTAT: RD_SLVERR, RD_DECERR, TIMEOUT_R, TIMEOUT_AR, TIMEOUT_WDT. Регистр только для чтения (RO), сбрасывается в «0» при очистке флага, вызвавшего событие записью в регистр CH[n].INT_CLEAR			
ERR_WR_ADDR	Захватывается адрес на запись в AXI burst транзакции во время которой произошло одно из событий, флаг которого установился в CH[n].INT_RAWSTAT: WR_SLVERR, WR_DECERR, TIMEOUT_B, TIMEOUT_W, TIMEOUT_AW, TIMEOUT_WDT. Регистр только для чтения (RO), сбрасывается в «0» при очистке флага, вызвавшего событие записью в регистр CH[n].INT_CLEAR			

14.2 Сценарии настройки

Общая конфигурация

Первоначально, исходя из сценариев использования DMA, нужно определить режим работы контроллера (поле JOINT_MODE регистра CONTROL). Если в основном требуется пересылка больших блоков памяти, то предпочтение следует отдать совместному режиму работы. Для обслуживания периферийных модулей стоит оставить обычный режим с двумя арбитрами, один из которых отвечает за поток чтения, а второй за поток записи.

Настройка и запуск канала

Все каналы полностью независимы и не используют общих регистров конфигурации. Настройка канала в основном состоит из двух частей: статической конфигурации и

дескриптора. Статическая конфигурация содержит информацию, которая не меняется в течении работы, дескриптор содержит текущую активность канала.

Как правило, настройка канала состоит из:

1) включения тактирования канала;

2) настройки статических регистров;

3) настройки контроллера прерываний;

4) настройки дескриптора или цепочки дескрипторов;

5) включения канала, если он неактивен (сброшен регистр CH[n].CH_ACTIVE);

6) запуска канала (записью подходящего значения в CH_START или в регистр канала CH[n].CH_START).

Остановка канала

Канал будет работать до тех пор, пока не выполнит свой последний дескриптор, затем самостоятельно остановится. После этого значения полей CH_RD_ACTIVE и CH_WR_ACTIVE регистра CH[n].CH_STATUS будут равны 0. Канал можно остановить, сбросив бит CH_ACTIVE регистра CH[n].CH_ACTIVE. Позже работу канала можно возобновить, снова установив бит CH_ACTIVE.

Перезапуск канала

Чтобы перезапустить канал, необходимо выполнить последовательность выполнения операций:

1) Остановите канал, сбросив бит CH_ACTIVE в регистре CH[n].CH_ACTIVE;

2) Заново включите канал, записав 1 в регистр CH[n].CH_START;

3) Запустите канал, записав 1 в регистр CH[n].CH_ACTIVE.

Обработка прерываний

Когда процессор получает прерывание, он уже должен знать, к какому биту шины прерываний он подключен.

Необходимо выполнить следующие действия:

1) Прочитайте регистр CH_INTSTAT, чтобы выяснить, какой канал вызвал прерывание (не исключена ситуация, когда несколько каналов могут ожидать обработки);

2) Прочитайте регистр CH[n].INT_STATUS соответствующего канала, чтобы определить, какое прерывание обрабатывать;

3) Выполните необходимые действия;

4) Сбросьте прерывание, записью подходящего значения в соответствующий регистр CH[n].INT_CLEAR.

Последовательность отключения питания

Для отключения питания DMA:

1) Сбросьте регистр CH[n].CH_ACTIVE для всех активных каналов (это остановит канал при завершении текущих ожидающих транзакций) или установите бит CMD_LAST регистра CH[n].CONFIG во всех активных каналах (это остановит канал при завершении обработки текущего буфера передачи).

2) Дождитесь установки состояния ожидания (регистр IDLE).

3) DMA готов к отключению питания.

15 Порты ввода-вывода GPIO

В состав микроконтроллера входят семь 16-разрядных портов ввода-вывода: A, B, C, D, E, F, G. Структуры портов и функционирование – идентичны.

Управляющие регистры портов GPIO приведены в приложении А.9.

Каждый цифровой вывод порта микроконтроллера может использоваться как двунаправленный вывод общего назначения (режим GPIO). Помимо этого, все выводы имеют альтернативную функцию (или функции).

Максимальное входное напряжение для всех выводов GPIO не должно превышать (U_{CC1} + 0.5) В, где U_{CC1} – напряжение питания VCC.

Выводы портов GPIOA, GPIOB, GPIOC также являются входами блока АЦП. Вход АЦП не является альтернативной функцией, он активен постоянно. Для минимизации влияния на измерения блока АЦП, при его работе не следует переключать вывод или использовать его в режиме альтернативной функции.

По умолчанию порты находятся в сбросе и не тактируются. Активировать порты можно с помощью соответствующих бит регистров CGCFGAHB, RSTDISAHB блока RCU.

15.1 Функционирование порта

Полученные данные сохраняются в регистре DATA порта. Данные для передачи записываются в регистр DATAOUT порта. Существует возможность модификации состояния регистра DATAOUT путем записи единиц в регистр DATAOUTSET для установки соответствующих бит, в регистр DATAOUTCLR – для сброса бит, в регистр DATAOUTTGL – для инверсии бит.

Примечание – Регистр DATAOUTTGL, а также все регистры xxxCLR предназначены для изменения конфигурации порта одной записью. Использование операции чтение-модификация-запись с любым из этих регистров может привести к нежелательным результатам, влияющим на другие биты регистра.

На рисунке 15.1 приведена структурная схема вывода цифрового порта микроконтроллера. Схемы всех выводов идентичны.

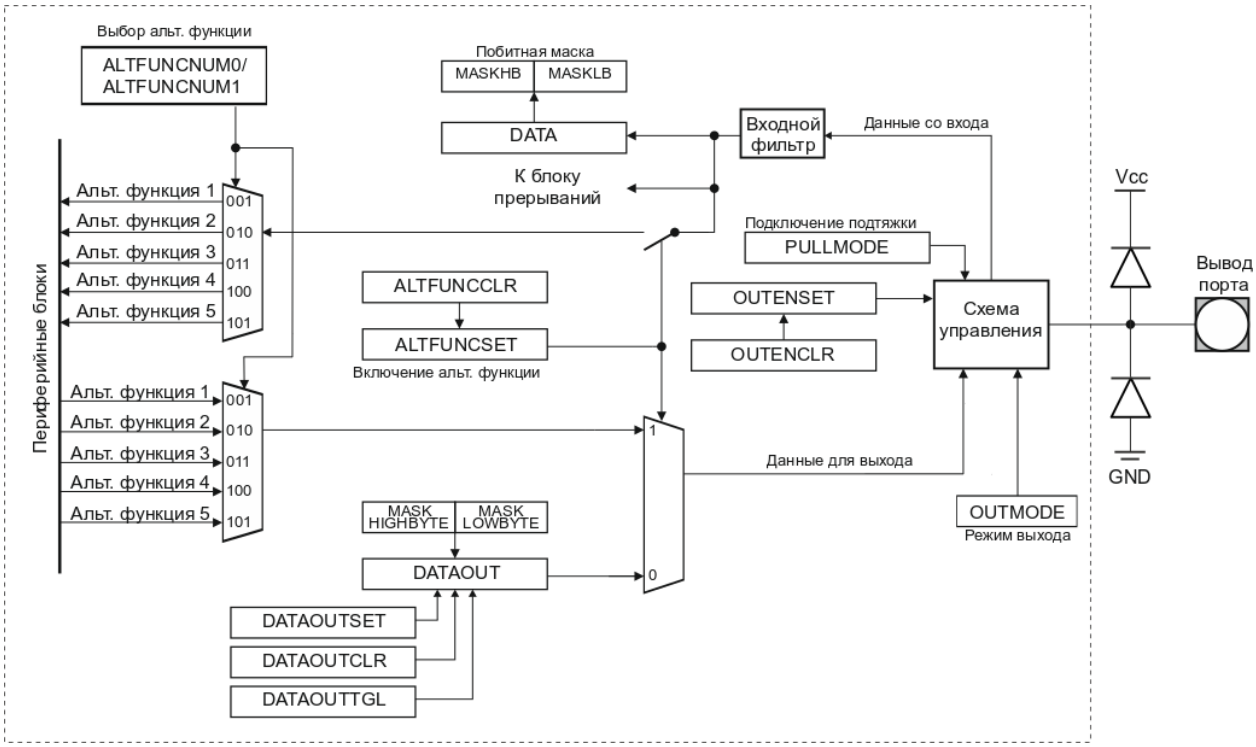


Рисунок 15.1 – Структурная схема вывода цифрового порта микроконтроллера

Схема состоит из двунаправленной площадки вывода, фильтра входных сигналов, мультиплексора выбора номера альтернативной функции, мультиплексора выбора режима работы (режим GPIO либо режим альтернативной функции).

Для каждого вывода задается режим работы, номер альтернативной функции, режим подтяжки, а также производится настройка порта на работу в режиме с открытым стоком/истоком. Для работы с периферийными блоками включается режим альтернативной функции. Входной сигнал может подаваться для дальнейшей обработки как напрямую (асинхронный вход), так и проходить обработку через фильтр.

После сброса все выводы конфигурируются как выводы общего назначения (режим GPIO) и находятся в третьем состоянии.

Выбор режима подтяжки конфигурируется регистром PULLMODE. Установленный бит регистра PULLMODE включает подтяжку для соответствующего вывода к уровню логической единицы (Pull-up). Сопротивление внутренних резисторов подтяжки Pull-up в диапазоне от 27 кОм до 65 кОм, типовое – 40 кОм.

Разрешение работы выходных каскадов определяется состоянием бит регистра OUTENSET (для сброса установленных бит следует записать единицы в регистр OUTENCLR), а их режимы («push-pull», открытый сток/исток) состоянием полей в регистре OUTMODE. В режимах открытый сток/исток управление состоянием вывода осуществляется регистрами DATAOUT, DATAOUTCLR, DATAOUTSET, DATAOUTTGL, при этом режим выхода должен быть отключен записью в регистр OUTENCLR.

В таблице 15.1 представлены настройки регистров ALTFUNCSET, OUTENSET, OUTMODE для каждого из режимов работы вывода порта.

Таблица 15.1 – Режимы работы вывода порта

Режим	ALTFUNCSET	OUTENSET	OUTMODE
Digital Input / Analog Input	0	0	00 (Push-Pull)
Output Push-Pull	0	1	00 (Push-Pull)
Output OpenDrain	0	0	01 (Open Drain)
Output OpenSource	0	0	10 (Open Source)
AltFunc Push-Pull	1	1	00 (Push-Pull)
AltFunc OpenDrain	1	0	01 (Open Drain)
AltFunc OpenSource	1	0	10 (Open Source)
Примечание – Для сброса установленных бит следует записать единицы в регистры ALTFUNCCLR и OUTENCLR.			

15.2 Режим альтернативных функций

Для перевода желаемого вывода порта в режим альтернативной функции необходимо установить соответствующий бит в регистре ALTFUNCSET порта. Для отключения альтернативной функции нужно записать единицу в соответствующий бит регистра ALTFUNCCLR. Выбор номера альтернативной функции для младшей части порта (выводы 0-7) осуществляется с помощью регистра ALTFUNCNUM0, а для старшей (выводы 8-15) – ALTFUNCNUM1. Каждому выводу соответствуют три бита регистра.

Входы и выходы периферийных блоков в процессе работы коммутируются с выводами микроконтроллера при условии, что для этих выводов включен режим альтернативной функции. В связи с этим периферийный блок может передавать информацию на несколько выводов одновременно. В то же время прием информации может осуществляться только с одного вывода, во избежание конфликтов уровней сигналов (для этого дополнительно предусмотрена система приоритета альтернативных функций). Количество выводов, сигналы с которых могут быть переданы на периферийный блок, для каждого блока различно.

Распределение приоритетов для входных функций происходит следующим образом: наивысший приоритет имеют функции с наименьшим номером, низший приоритет – функции с наибольшим номером. Но при этом, если на разных выводах встречаются идентичные функции с одинаковыми номерами, то их приоритеты распределяются по букве порта и номеру пина: чем номер пина меньше, а буква порта ближе к началу алфавита, тем приоритет больше.

Указанные правила программирования выводов для приема данных распространяются на все периферийные блоки.

15.3 Входные фильтры

Ко всем площадкам выводов подключены входные фильтры. На рисунке 15.2 показана структурная схема фильтра вывода порта.

Входной сигнал с вывода порта может приниматься как напрямую (асинхронный режим), так и пересинхронизироваться (синхронизироваться с тактовой частотой работы микроконтроллера). Управление осуществляется регистрами SYNCSET/SYNCLCR.

Дополнительно есть возможность включения накопления трех или шести отсчетов входного сигнала для помехоустойчивости вывода. Если результаты всех отсчетов совпадают, сигнал передается дальше по схеме, в противном случае состояние сигнала не меняется. Временные интервалы между отсчетами задаются в количестве тактов системной частоты посредством регистра QUALSAMPLE. Временной интервал задается один для всех выводов порта.

Включение фильтра и задание режима его работы осуществляется посредством регистров QUALSET/QUALCLR и QUALMODESET/QUALMODECLR соответственно.

Одновременно оба режима активными быть не могут – при установленных единицах в одних и тех же разрядах SYNCSET и QUALSET сигнал будет проходить напрямую с входа фильтра на его выход.

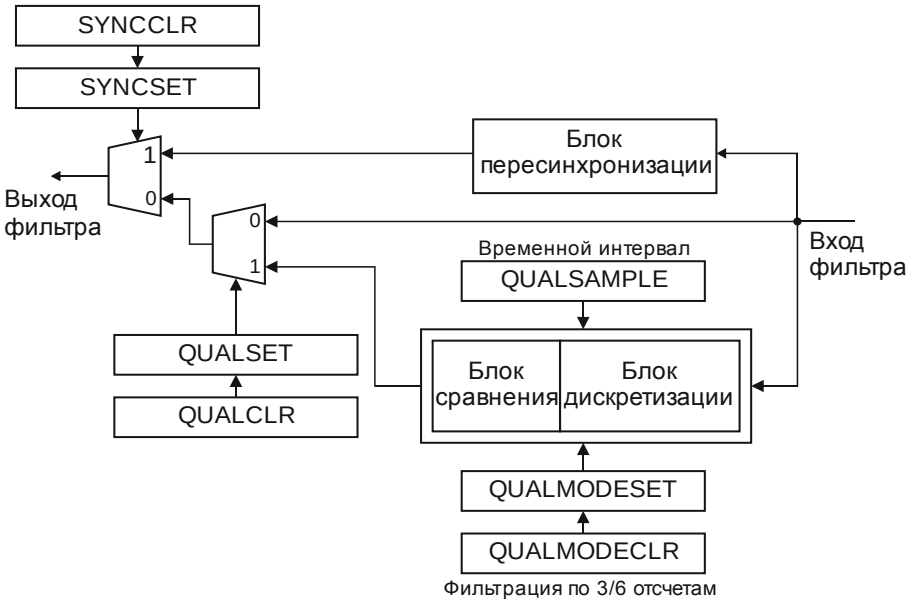


Рисунок 15.2 – Структурная схема фильтра вывода порта

15.4 Прерывания

Каждый из выводов способен генерировать прерывание. На рисунке 15.3 показана структурная схема блока генерации прерываний.

Схема вывода позволяет также осуществлять гибкое управление прерываниями и задавать, по какому аппаратному событию генерировать прерывание (по какому фронту

или уровню). При возникновении прерывания в регистре INTSTATUS устанавливается соответствующий флаг, и выставляется прерывание в контроллере прерываний PLIC.

Прерывание может быть сброшено программно записью единицы в соответствующий бит регистра INTSTATUS. Для разрешения прерывания вывода порта следует записать единицу в соответствующий выводу бит регистра INTENSET, а для запрета прерывания – единицу в бит регистра INTENCLR.

Для задания типа события (уровень или фронт), по которому генерируется прерывание, используется регистр INTTYPESET, для задания полярности (низкий/высокий уровень или положительный/отрицательный фронт) используется INTPOLSET, а для сброса настроек – INTTYPECLR и INTPOLCLR соответственно.

Существует возможность организации прерывания по обоим фронтам – сначала необходимо задать тип прерывания – фронт (запись в INTTYPESET), а затем записать соответствующие единицы в INTEDGESET. В этом режиме состояние регистра полярности INTPOLSET игнорируется. Отключить режим генерации прерывания по обоим фронтам можно записью в INTEDGECLR, в таком случае для генерации в дальнейшем будет использована текущая настройка полярности (регистр INTPOLSET).

В режиме прерывания по уровню состояние регистра INTEDGESET не влияет на их генерацию.

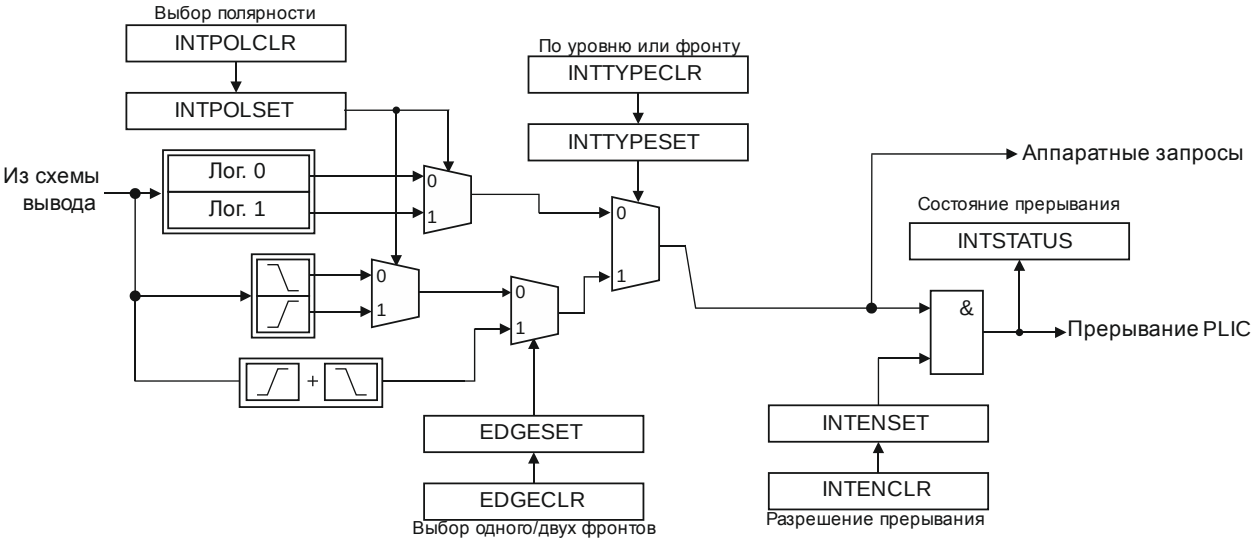


Рисунок 15.3 – Структурная схема блока генерации прерываний

15.5 Генерация аппаратных запросов

Кроме генерации прерываний, порты также имеют возможность генерации запросов к другой периферии. Генерация запроса происходит по условиям возникновения прерывания на выбранном выводе, при этом не важно, маскировано само прерывание или нет (состояние INTENSET), влияние оказывают только настройки генерации прерывания (INTTYPESET, INTPOLSET, INTEDGESET).

Для того чтобы в соответствии с настройками прерывания генерировался запрос к контроллеру DMA, необходимо осуществить запись единиц в DMATXREQSET/DMARXREQSET. Отключение генерации запросов к контроллеру DMA осуществляется через DMATXREQCLR/DMARXREQCLR.

Для того чтобы в соответствии с настройками прерывания сгенерировался запрос начала преобразования АЦП/ЦАП, необходимо осуществить запись единиц в ADCSOCSET. Отключение генерации запросов начала преобразования АЦП/ЦАП осуществляется через регистр ADCSOCLR.

Для того чтобы в соответствии с настройками прерывания сгенерировался запрос RXEV к ядру, необходимо осуществить запись единиц в RXEVSET. Отключение генерации запросов RXEV к ядру осуществляется через RXEVCLR.

15.6 Механизм блокировки конфигурации

Для защиты конфигурации выводов от несанкционированного изменения реализован механизм блокировки.

Блокировка конфигурации осуществляется записью соответствующих единиц в регистр LOCKSET, сброс блокировки – в регистр LOCKCLR. При включенной блокировке игнорируется запись во все биты и поля настройки маскированных выводов – доступны для записи лишь регистры INTSTATUS и QUALSAMPLE.

По умолчанию все выводы являются разблокированными, а запись в регистры LOCKSET/LOCKCLR игнорируется. Для того чтобы сделать доступными для изменения регистры LOCKSET/LOCKCLR, необходимо в регистр LOCKKEY занести значение ключа ADEADBEEh. Если необходимо снова защитить от изменений регистры LOCKSET/LOCKCLR, то достаточно записать в LOCKKEY любое значение, отличное от ключа.

Текущий статус защиты регистров LOCKSET/LOCKCLR можно узнать, прочитав регистр LOCKKEY – значение 0000_0001h соответствует снятой защите, 0000_0000h – установленной.

15.7 Механизм маскирования

Для управления состоянием выводов порта дополнительно используется механизм маскирования. Он позволяет устанавливать желаемый уровень сигнала на нужном выводе, не затрагивая состояние других выводов. 16-разрядный порт условно разбивается на старший байт и младший байт. Для доступа по маске к младшему байту используется массив регистров MASKLB, а к старшему – MASKHB.

Каждый массив состоит из 256 регистров, каждый регистр имеет порядковый номер (от 00h до FFh), который является маской. Так, например, для порта A выделены две области памяти с адресами: 4001_0400h – 4001_07FCh для младшего байта и 4001_0800h – 4001_0BFCh для старшего байта. Биты с 9 по 2 адреса являются маской. Таким образом, адресу 4001_0400h соответствует маска 00h (MASKLB[0x00]), адресу 4001_0404h – 01h (MASKLB[0x01]) и т. д.

Для того чтобы изменить состояние выводов порта с использованием маски, нужно записать новое значение в соответствующий элемент массива (MASKLB или MASKHB) с порядковым номером, совпадающим со значением маски.

Разряды порта, закрытые «нулями» маски, останутся неизменными, а остальные примут новые значения. На рисунке 15.4 показан механизм маскирования младшего байта порта A.

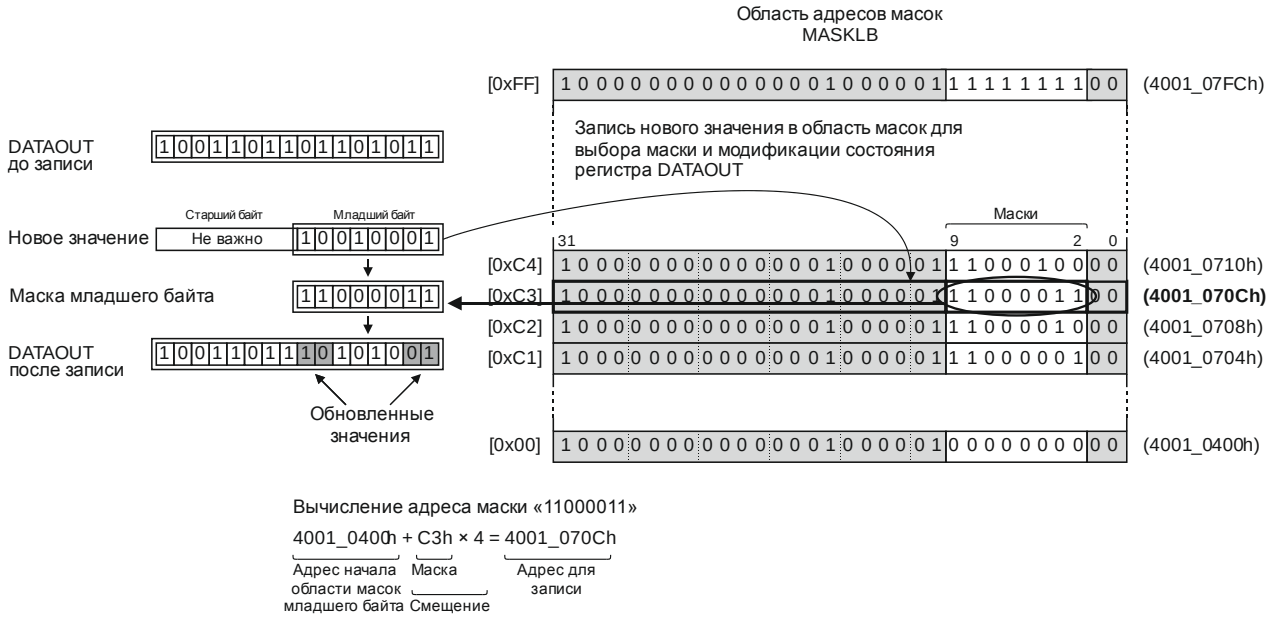


Рисунок 15.4 – Механизм изменения состояния младшего байта порта A с маскированием

Для изменения 0, 1, 6 и 7 битов регистра порта нужно использовать маску 1100_0011b. Эта маска является частью (биты с 9 по 2) адреса 4001_070Ch. Новое значение XX90h данных, которое требуется передать в порт (при этом старший байт числа не важен), нужно записать в ячейку с адресом 4001_070Ch. Далее это значение будет аппаратно маскировано и размещено в регистре порта DATAOUT.

Аналогично выполняется маскирование старшего байта, см. рисунок 15.5. Разница лишь в том, что в данном случае берется старший байт нового значения, а младший не важен.

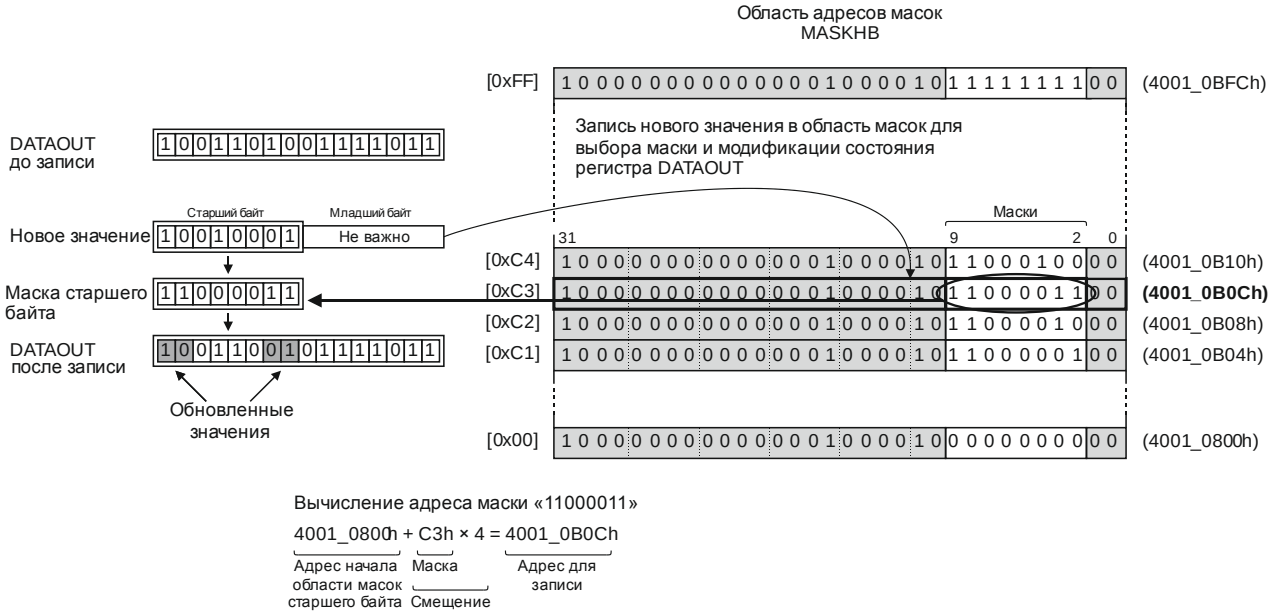


Рисунок 15.5 – Механизм изменения состояния старшего байта порта A с маскированием

16 Таймеры TMR

Микроконтроллер содержит шестнадцать блоков 32-разрядного таймера TMR, первые четыре из которых (TMR0 - TMR3), имеют расширенный функционал в части блоков захвата/сравнения.

TMR – 32-разрядный таймер/счетчик с устройствами захвата/сравнения. Таймер поддерживает несколько захватов/сравнений, выход ШИМ и выдержку временных интервалов. Обладает обширными возможностями прерываний, которые могут быть сгенерированы как при переполнении счетчика, так и от регистров захвата/сравнения. Таймер TMR имеет внешние выводы, подключаемые через альтернативные функции портов GPIO. Так, сигнал TMRx_IO – вход/выход блоков CAPCOM, входящих в состав таймера.

Управляющие регистры таймеров приведены в приложении А.10.

Характеристики таймера:

- 32-разрядный таймер/счетчик с 4-мя режимами работы;
- конфигурируемый источник тактового сигнала;
- 4 конфигурируемых блока захвата/сравнения для TMR0 - TMR3 и один блок захвата/сравнения для TMR4 – TMR15.

Структурная схема таймеров TMR0 - TMR3 представлена на рисунке 16.1.

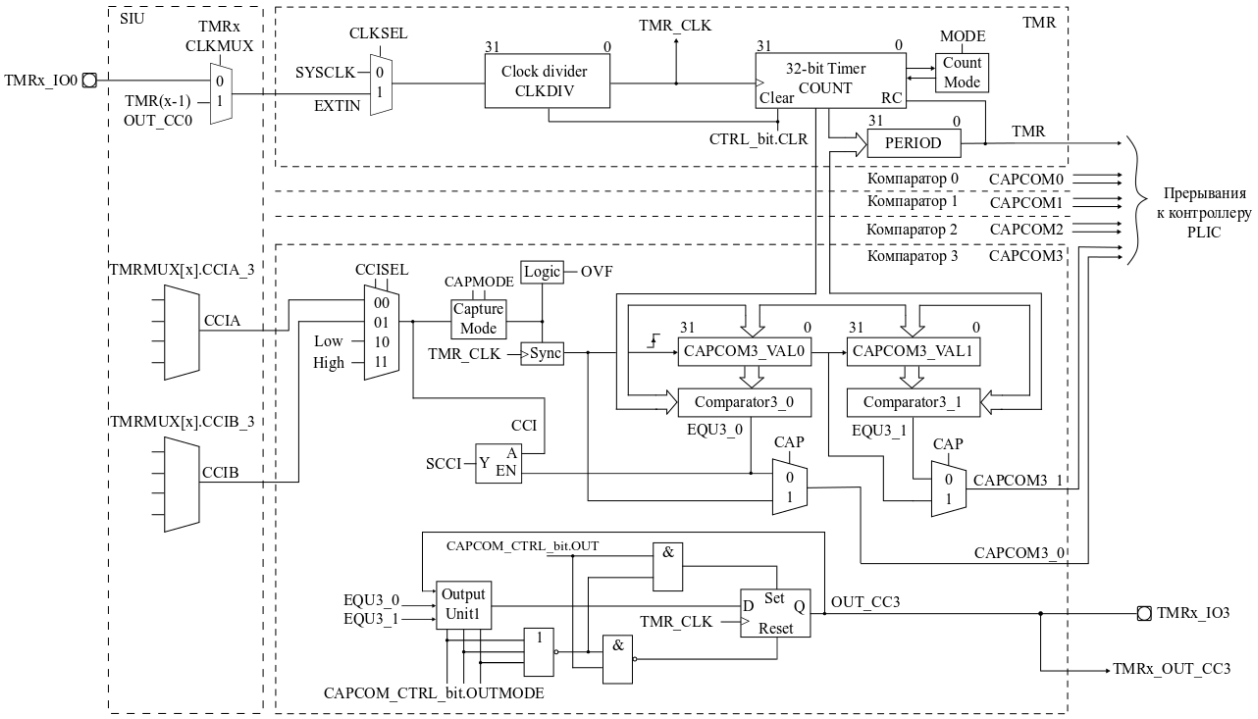


Рисунок 16.1 – Структурная схема таймеров TMR0 - TMR3

Структурная схема таймеров TMR4 – TMR15 представлена на рисунке 16.2.

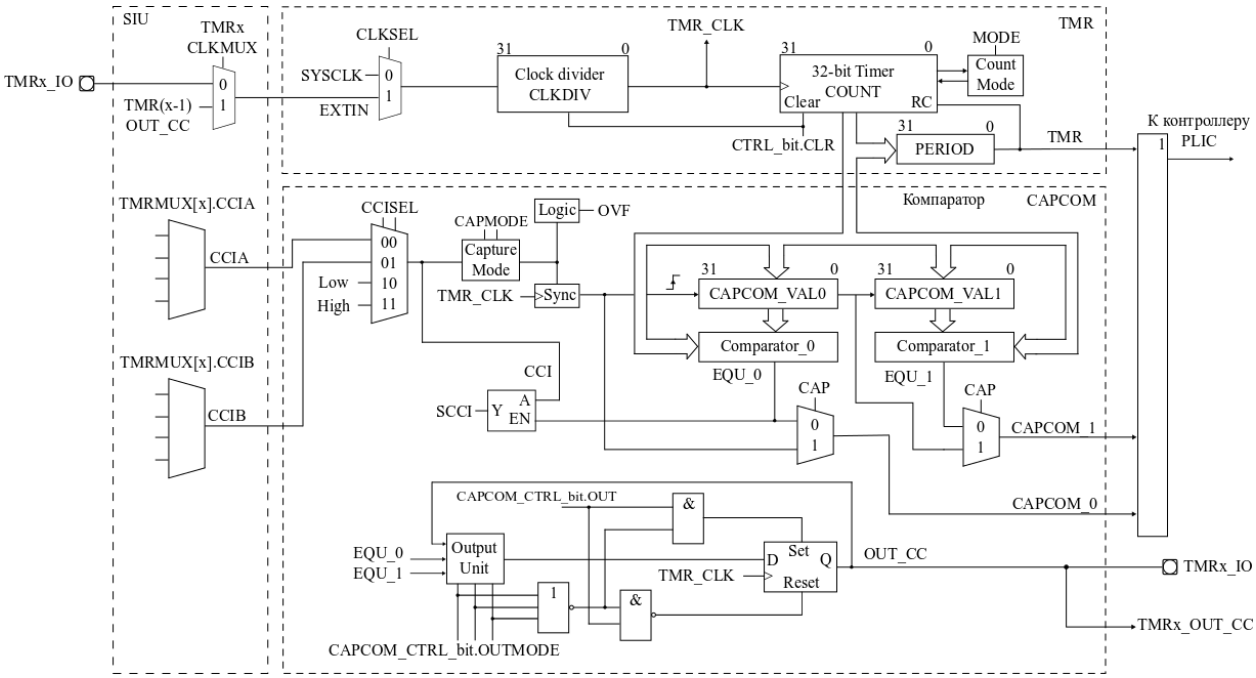


Рисунок 16.2 – Структурная схема таймеров TMR4 – TMR15

16.1 Функционирование таймера

32-разрядный таймер-счетчик COUNT

Регистр таймера-счетчика COUNT инкрементируется или декрементируется (в зависимости от режима работы) по переднему фронту тактового сигнала. Счетчик может быть считан или записан программно. Кроме того, таймер может генерировать прерывания при переполнении. Регистр COUNT очищается установкой бита CLR регистра CTRL. Бит CLR также очищает делитель тактовой частоты и направление счета для режима вверх/вниз.

Примечание: рекомендуется останавливать таймер перед изменением режима его работы (за исключением разрешения прерываний, операций с флагом прерываний и битом CLR) во избежание некорректной работы. В случае, если тактовый сигнал таймера асинхронен тактовому сигналу SYSCLK, любое чтение регистра COUNT должно происходить при остановленном таймере, иначе результат может быть непредсказуем. В качестве альтернативы возможно считывание регистра в рабочем состоянии несколько раз и выбор корректного значения программно (наиболее частое значение). Любая запись в регистр COUNT приводит к изменению счетчика таймера.

Выбор источника тактирования и делитель

Тактирование таймера осуществляется тактовым сигналом SYSCLK, либо от внешнего источника через вывод TMR_IO. Источник тактирования определяется битом CLKSEL. Выбранный сигнал синхронизации передается либо напрямую, либо через делитель, значение которого определяется регистром CLKDIV. При выставлении бита CLR делитель сбрасывается.

Предусмотрена возможность тактирования таймера с помощью выходного ШИМ-сигнала предыдущего таймера. Переключение между сигналом с внешней ножки TMR_IO и выходом предыдущего таймера осуществляется с помощью мультиплексора TMRxCLKMUX, управляемого регистром TMRCFG блока SIU. Для таймера TMR0 предыдущим является таймер TMR15. В случае, если предыдущий таймер имеет в составе 4 блока CAPCOM (таймеры TMR0-TMR3) тактовый сигнал берется с выхода CAPCOM0 (TMR(x-1) OUT_CC0).

Запуск и остановка таймера

Таймер считает, если в регистре CTRL значение поля MODE > 0 (режим работы), источник тактового сигнала активен и работа счетчика таймера разрешена в регистре CNTEN блока SIU.

Таймер может быть остановлен путем записи значения 0 в поле MODE регистра CTRL. При этом регистры COUNT и PERIOD сохранят свои значения и при повторном запуске таймер продолжит счет. Чтобы при перезапуске таймер начал счет с нуля, необходимо перед запуском сбросить регистр COUNT путем записи нулевого значения.

16.2 Режимы счёта

Управление режимом таймера.

Таймер имеет четыре режима работы, описанных в таблице 16.1: «стоп», «вверх», «непрерывный» и «вверх/вниз». Режимы выбираются с помощью поля MODE регистра CTRL.

Таблица 16.1 - Режимы работы таймера

MODE	Режим	Описание
00	Стоп	Остановка таймера
01	Вверх	Многократный счет от 0 до значения в регистре PERIOD.
10	Непрерывный	Многократный счет от 0 до значения FFFF_FFFFh.
11	Вверх/вниз	Многократный счет от 0 вверх до значения в регистре PERIOD и обратно до 0.

Режим «вверх»

Режим «вверх» используется при значении периода таймера, отличном от значения FFFF_FFFFh. Таймер циклично считает до значения регистра сравнения PERIOD, задающего период, как показано на рисунке 16.3. Количество отсчетов в периоде равно (PERIOD + 1). Когда значение таймера становится равным значению регистра сравнения (COUNT = PERIOD), таймер перезапускается, начиная счет с нуля. Если режим «вверх» выбран, когда COUNT > PERIOD, таймер перезапускается немедленно.

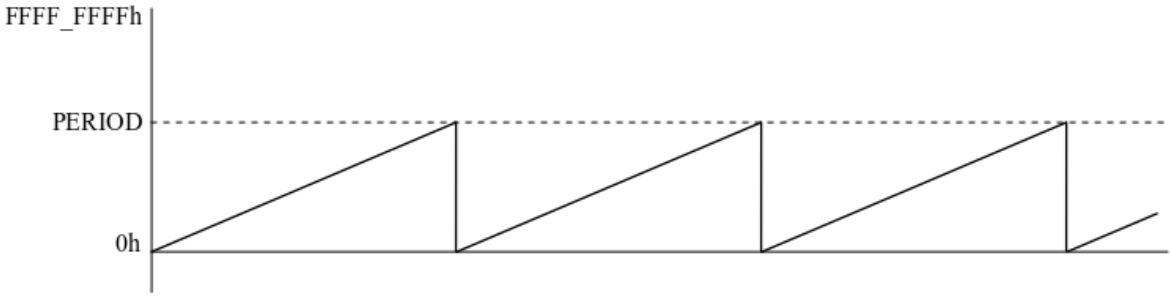


Рисунок 16.3 — Режим «вверх»

Флаг прерывания TMR регистра RIS устанавливается, когда значение счетчика COUNT становится равным значению регистра сравнения PERIOD.



Рисунок 16.4 — Установка флагов прерываний в режиме «вверх»

При изменении регистра PERIOD во время работы таймера, таймер продолжает отсчет вверх до новой границы периода, если новый период больше или равен старому или больше текущего значения счетчика. Если новый период меньше текущего значения счетчика, таймер обнуляется. Однако до обнуления счетчика может произойти один дополнительный отсчет.

Непрерывный режим

В непрерывном режиме таймер циклично считает вверх до значения FFFF_FFFFh и перезапускается с 0, как показано на рисунке 16.5.

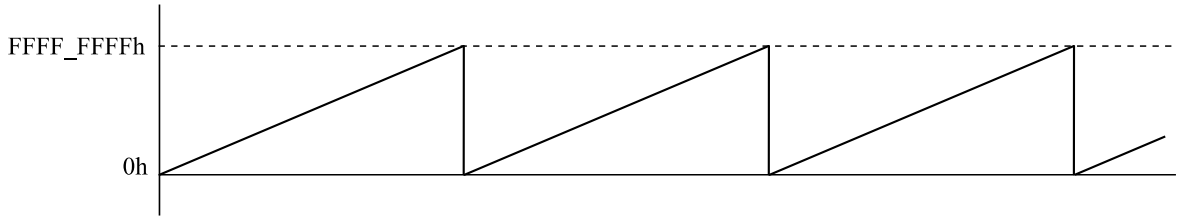


Рисунок 16.5 — Непрерывный режим

Флаг прерывания TMR устанавливается при переключении счетчика из значения FFFF_FFFFh в значение 0. Цикл выставления флага представлен на рисунке 16.6.

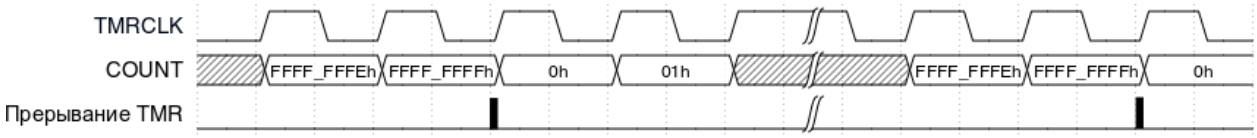


Рисунок 16.6 — Установка флага прерываний в непрерывном режиме

Использование непрерывного режима

Непрерывный режим может использоваться для генерации независимых временных интервалов и выходных частот. Каждый раз по завершении интервала генерируется прерывание. Затем временной интервал добавляется к регистру CAPCOM[n].VALx (где n - номер блока CAPCOM, x – номер регистра CAPCOM[n].VAL0 или CAPCOM[n].VAL1) в подпрограмме обработки прерывания. На рисунке 16.7 показаны два разных временных интервала t0 и t1, записываемые в регистры захвата / сравнения. В этом случае временной интервал контролируется аппаратно, а не программно, без влияния задержки обработки прерывания.

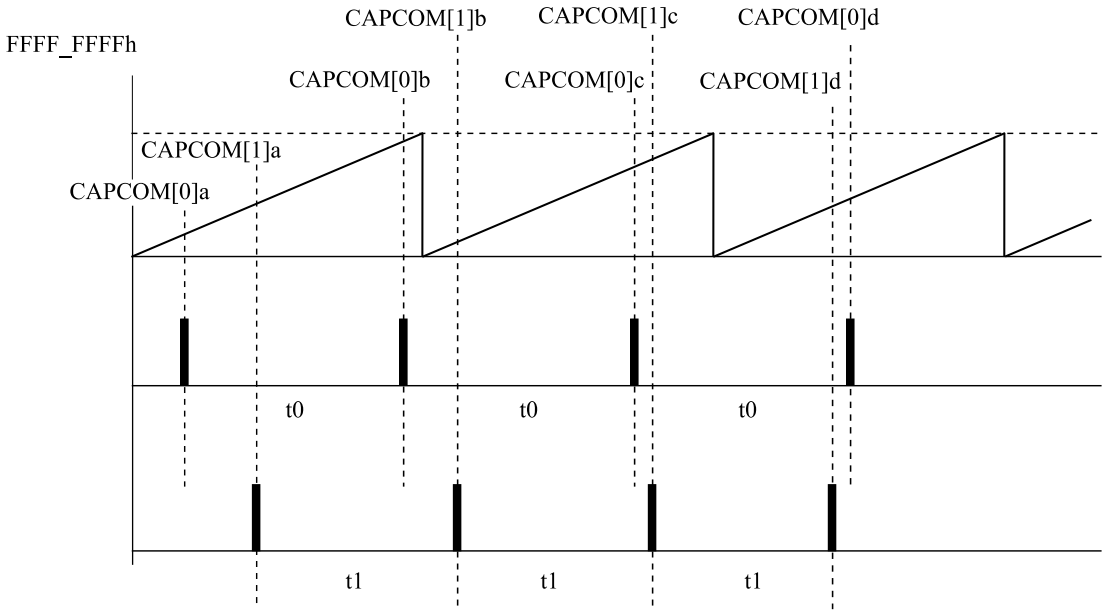


Рисунок 16.7 — Временные интервалы в непрерывном режиме

Временные интервалы могут быть сформированы и в других режимах, где в качестве регистра периода используется PERIOD. Их обработка более сложная, поскольку сумма старых данных CAPCOM[n].VALx и нового периода может быть выше значения PERIOD. Если предыдущее значение CAPCOM[n].VALx плюс tx больше, чем данные PERIOD, значение PERIOD необходимо вычесть, чтобы получить правильный временной интервал.

Режим «вверх/вниз»

Режим «вверх/вниз» используется при необходимости генерации симметричных импульсов в периоде таймера, отличном от FFFF_FFFFh. Таймер непрерывно считает до значения регистра PERIOD и назад к 0, как показано на рисунке 16.8. Период равен удвоенному значению регистра PERIOD.

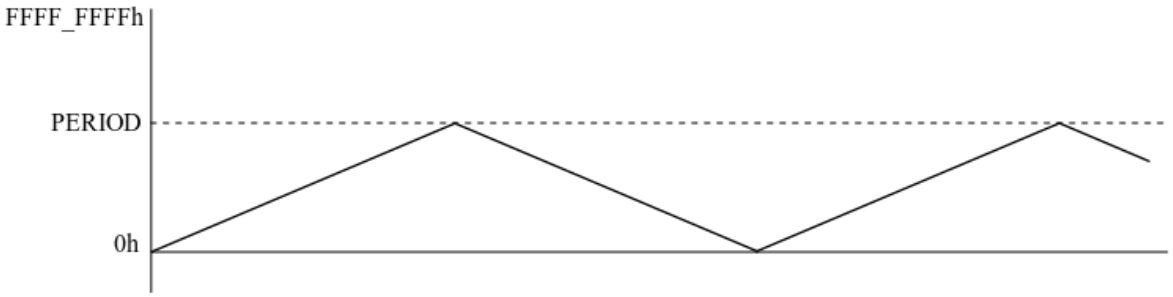


Рисунок 16.8 — Режим «вверх/вниз»

Направление счета запоминается. Это позволяет таймеру останавливаться и запускаться в том же направлении отсчета, которое было до останова. Очистить направление можно выставлением бита CLR, который очищает также значения регистра COUNT и регистра делителя тактовой частоты CLKDIV.

В режиме «вверх/вниз» флаг прерывания TMR устанавливается при переключении значения COUNT из 1 в 0. Цикл выставления флага представлен на рисунке 16.9.

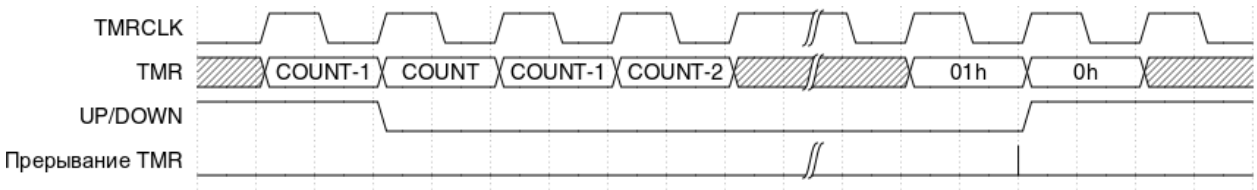


Рисунок 16.9 – Установка флага прерывания в режиме «вверх/вниз»

Изменение регистра периода PERIOD

При изменении PERIOD во время работы таймера и отсчете в направлении вниз таймер продолжает свой спуск до тех пор, пока не достигнет нуля. Новый период вступает в силу после того, как счетчик начнет обратный отсчет до нуля.

Когда таймер ведет отсчет в направлении вверх, и новый период больше или равен старому периоду, или больше текущего значения отсчета, таймер ведет отсчет до нового периода, прежде чем начать обратный отсчет. Когда таймер ведет отсчет в направлении вверх, и новый период меньше текущего значения отсчета, таймер начинает обратный отсчет. Однако может произойти еще один подсчет, прежде чем счетчик начнет обратный отсчет.

Использование режима «Вверх/вниз»

Режим вверх/вниз необходим, когда требуется время простоя между выходными сигналами. Например, чтобы избежать перегрузки, два выхода, управляющие Н-мостом, никогда не должны находиться в высоком состоянии одновременно. В примере, показанном на рисунке 16.10, t_{dead} является:

$$t_{dead} = t_{timer} (CAPCOM[0].VAL0 – CAPCOM[1].VAL0)$$

где: t_{dead} - Время, в течение которого оба выхода должны быть неактивны.
t_{timer} - Время цикла работы таймера.
CAPCOM[n].VALx - Содержимое регистра захвата/сравнения.

Регистры CAPCOM[n].VALx не буферизуются. Они обновляются сразу же после записи. Таким образом, любое требуемое время простоя не будет поддерживаться автоматически.

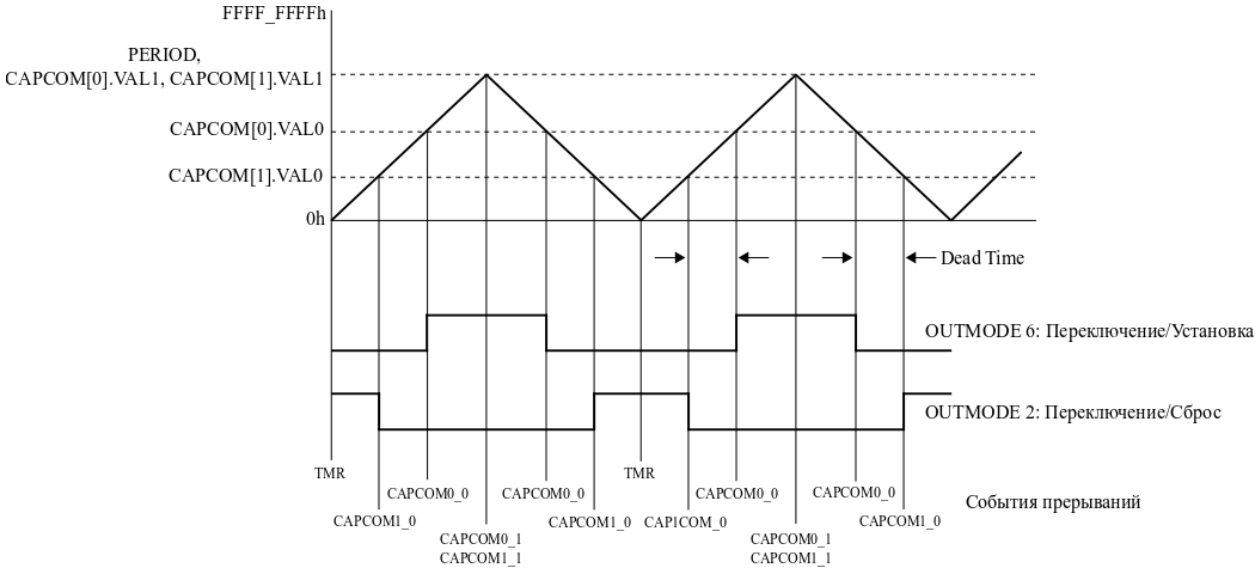


Рисунок 16.10 – Блок вывода в режиме «вверх/вниз»

Примечание – На рисунке 16.10 значения регистров PERIOD, CAPCOM[0].VAL1 и CAPCOM[1].VAL1 совпадают.

16.3 Блоки захвата/сравнения

В TMR0-TMR3 присутствуют четыре идентичных блока захвата/сравнения CAPCOM. В TMR4-TMR15 имеется по одному блоку захвата/сравнения CAPCOM. Любой из блоков может быть использован для сбора данных таймера или для генерации выходного сигнала с заданными временными интервалами.

Режим захвата

Режим захвата выбирается, при установке бита CAP = 1 в регистре CAPCOM[n].CTRL. Режим захвата используется для записи временных событий. Он может быть использован для вычисления скорости или измерения времени. Входы захвата TMR_CCIA и TMR_CCIB могут быть подключены к внешним выводам TMRx_IO или внутренним сигналам (источник сигнала выбирается с помощью селекторов TMRMUX[x].CCIA и TMRMUX[x].CCIB блока SIU). Выбор входа захвата осуществляется с помощью битового поля CCISEL. Биты CAPMODE выбирают фронт захвата входного сигнала как нарастающий, спадающий или и то, и другое вместе. Захват происходит на выбранном фронте входного сигнала.

В случае захвата:

- Значение таймера копируется в регистр CAPCOM[n].VALx;
- Устанавливается флаг прерывания CAPCOMn_x в регистре RIS.

Уровень входного сигнала может быть считан в любое время с помощью бита CCI регистра CAPCOM[n].CTRL. Сигнал захвата TMR32_CCIA синхронизируется с частотой таймера. Синхронизация внешнего сигнала захвата изображена на рисунке 16.11.

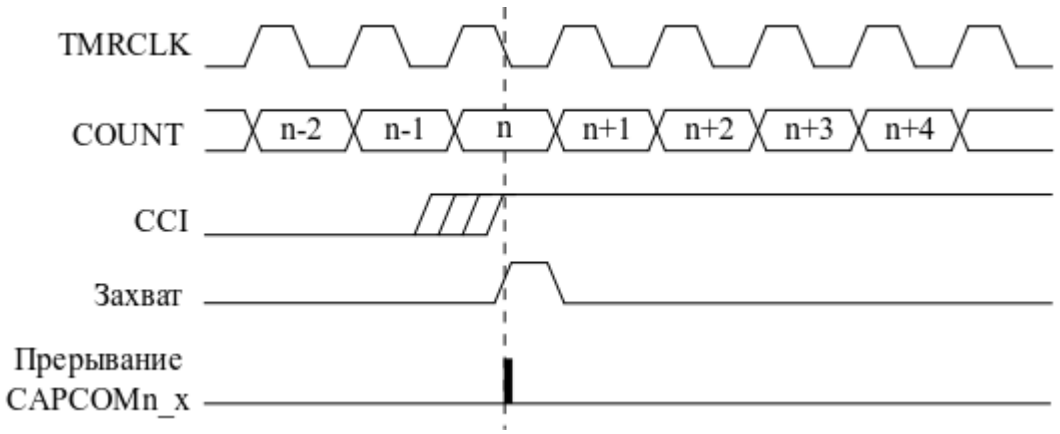


Рисунок 16.11 – Синхронизация внешнего сигнала захвата

В случае захвата нового события без считывания предыдущего значения из регистра CAPCOM[n].VAL0 происходит перенос значения из регистра CAPCOM[n].VAL0 в регистр CAPCOM[n].VAL1, одновременно с этим устанавливается бит OVF регистра CAPCOM[n].CTRL как проиллюстрировано на рисунке 16.12. Сброс бита OVF производится программно записью «0» или при чтении регистра CAPCOM[n].VAL1.

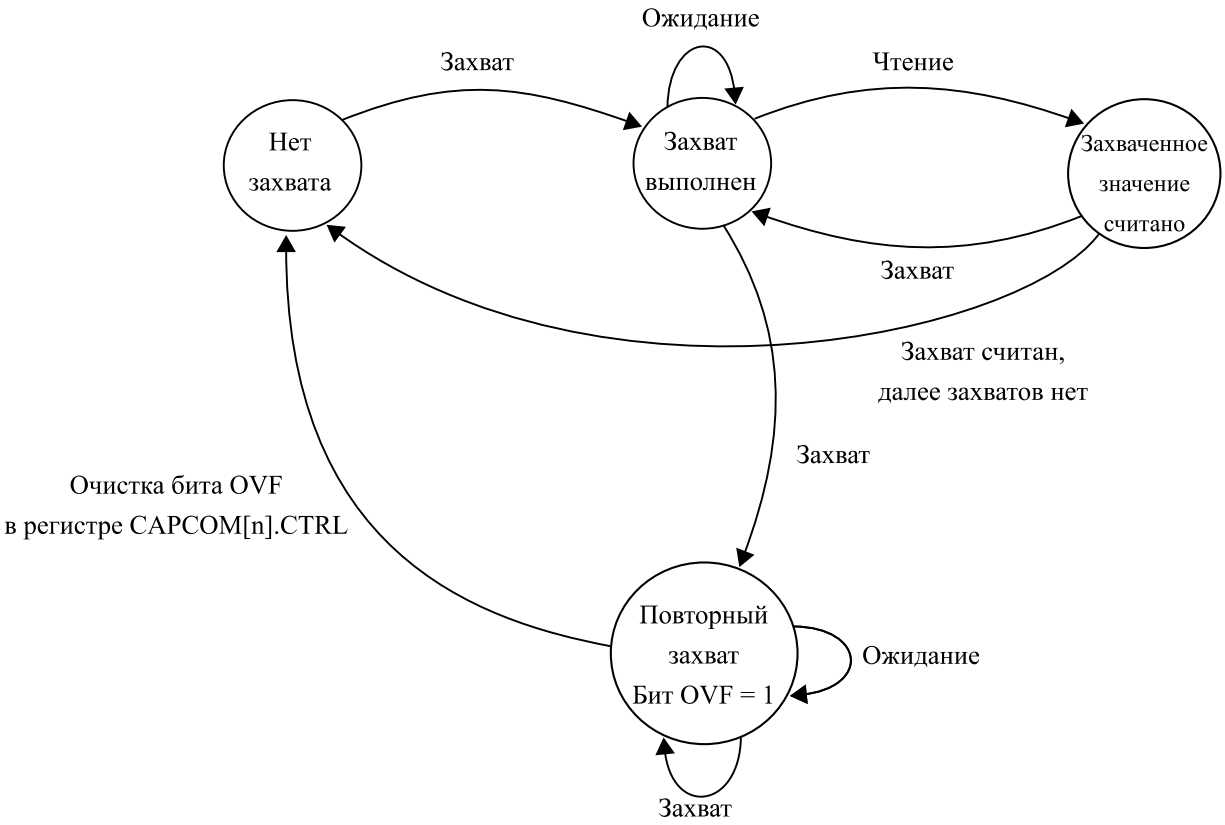


Рисунок 16.12 – Цикл захвата

Режим сравнения

Режим сравнения выбирается при CAP = 0 в регистре CAPCOM[n].CTRL. Режим сравнения используется для генерации выходных сигналов ШИМ или прерываний через определенные промежутки времени. Когда COUNT досчитывает до значения в CAPCOM[n].VALx:

- Устанавливается флаг прерывания CAPCOMn_x;
- Внутренний сигнал EQUn_x = 1;
- EQUn_x влияет на выход в соответствии с режимом вывода;
- Входной сигнал CCI фиксируется в SCCI.

16.4 Выходной модуль

Каждый блок захвата/сравнения содержит блок вывода. Блок вывода используется для генерации выходных сигналов, таких как ШИМ-сигналы. Каждый выходной блок имеет восемь режимов работы, которые генерируют сигналы на основе сигналов EQUn_0 и EQUn_1.

Режимы вывода

Режимы вывода определяются битами OUTMODE регистра CAPCOM[n].CTRL, они описаны в таблице 16.2. Выходной сигнал изменяется по переднему фронту тактов таймера для всех режимов, кроме режима 0.

Таблица 16.2 – Режимы вывода

№	OUTMODE	Режим	Описание
0	000	Вывод	Выходной сигнал TMR_OUTn определяется битом OUT регистра CAPCOM[n].CTRL. Сигнал TMR_OUTn обновляется немедленно при обновлении бита OUT.
1	001	Установка	Выходной сигнал устанавливается, когда таймер отсчитывает значение CAPCOM[n].VAL0. Он остается установленным до тех пор, пока не произойдет сброс таймера или пока не будет выбран другой режим вывода, который повлияет на выходной сигнал.
2	010	Переключение/сброс	Выход переключается, когда таймер отсчитывает значение CAPCOM[n].VAL0. Он сбрасывается, когда таймер отсчитывает значение CAPCOM[n].VAL1.
3	011	Установка/сброс	Выходной сигнал устанавливается, когда таймер отсчитывает значение CAPCOM[n].VAL0. Он сбрасывается, когда таймер отсчитывает значение CAPCOM[n].VAL1.
4	100	Переключение	Выход переключается, когда таймер отсчитывает значение CAPCOM[n].VAL0. Период вывода в два раза превышает период таймера.
5	101	Сброс	Выходной сигнал сбрасывается, когда таймер отсчитывает значение CAPCOM[n].VAL0. Он остается сброшенным до тех пор, пока не будет выбран другой режим вывода, который повлияет на выходной сигнал.
6	110	Переключение/ установка	Выход переключается, когда таймер отсчитывает значение CAPCOM[n].VAL0. Он устанавливается, когда таймер отсчитывает значение CAPCOM[n].VAL1.
7	111	Сброс/установка	Выходной сигнал сбрасывается, когда таймер отсчитывает значение CAPCOM[n].VAL0. Он устанавливается, когда таймер отсчитывает значение CAPCOM[n].VAL1.
Примечания: 1) n – номер блока CAPCOM; 2) x – номер регистра CAPCOM[n].VAL.			

Пример вывода — таймер в режиме счета вверх

Сигнал TMR_OUTn изменяется, когда таймер достигает значений CAPCOM[0].VAL0 и CAPCOM[0].VAL1, в зависимости от режима вывода. Пример показан на рисунке 16.13 с использованием CAPCOM[0].VAL0 и CAPCOM[0].VAL1.

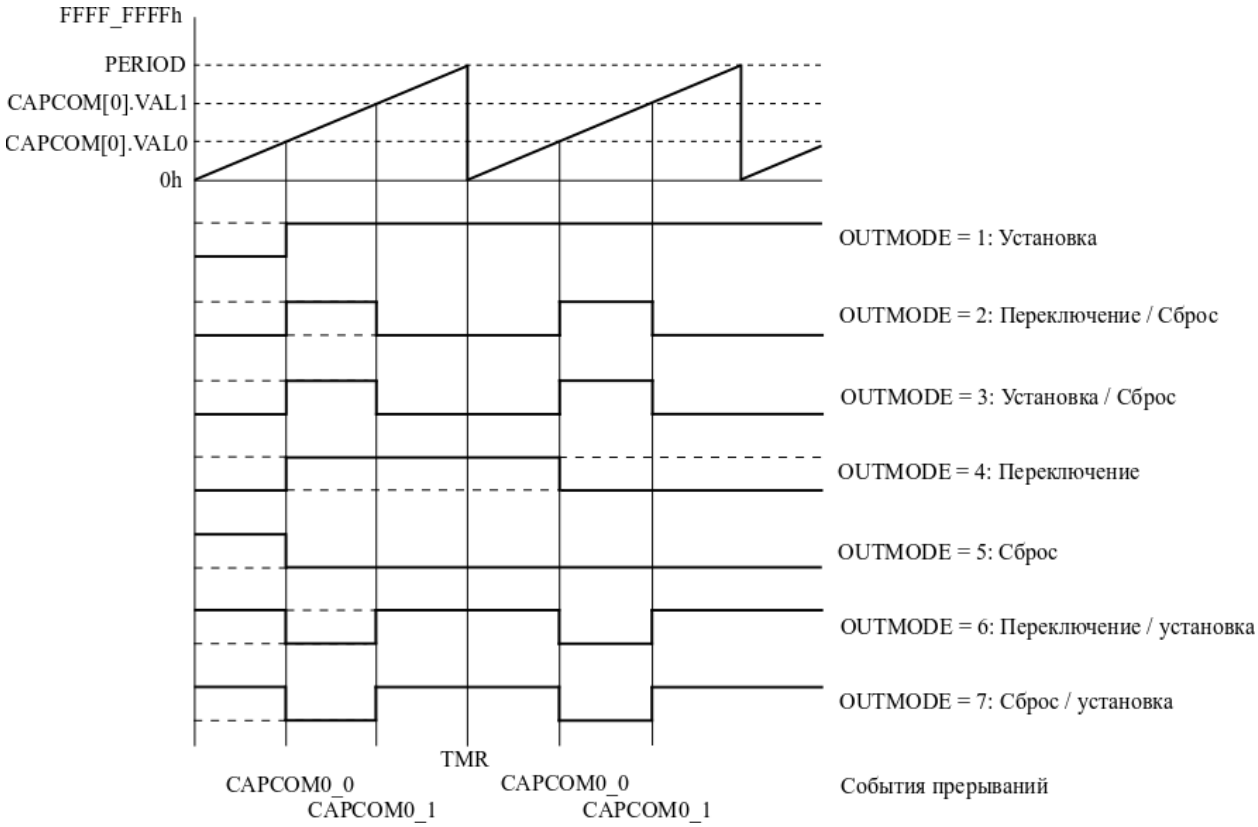


Рисунок 16.13 – Пример вывода. Таймер в режиме счета вверх

Пример вывода — Таймер в непрерывном режиме

Сигнал TMR_OUTn изменяется, когда таймер достигает значений CAPCOM[0].VAL0 и CAPCOM[0].VAL1, в зависимости от режима вывода. Пример показан на рисунке 16.14 с использованием CAPCOM[0].VAL0 и CAPCOM[0].VAL1.

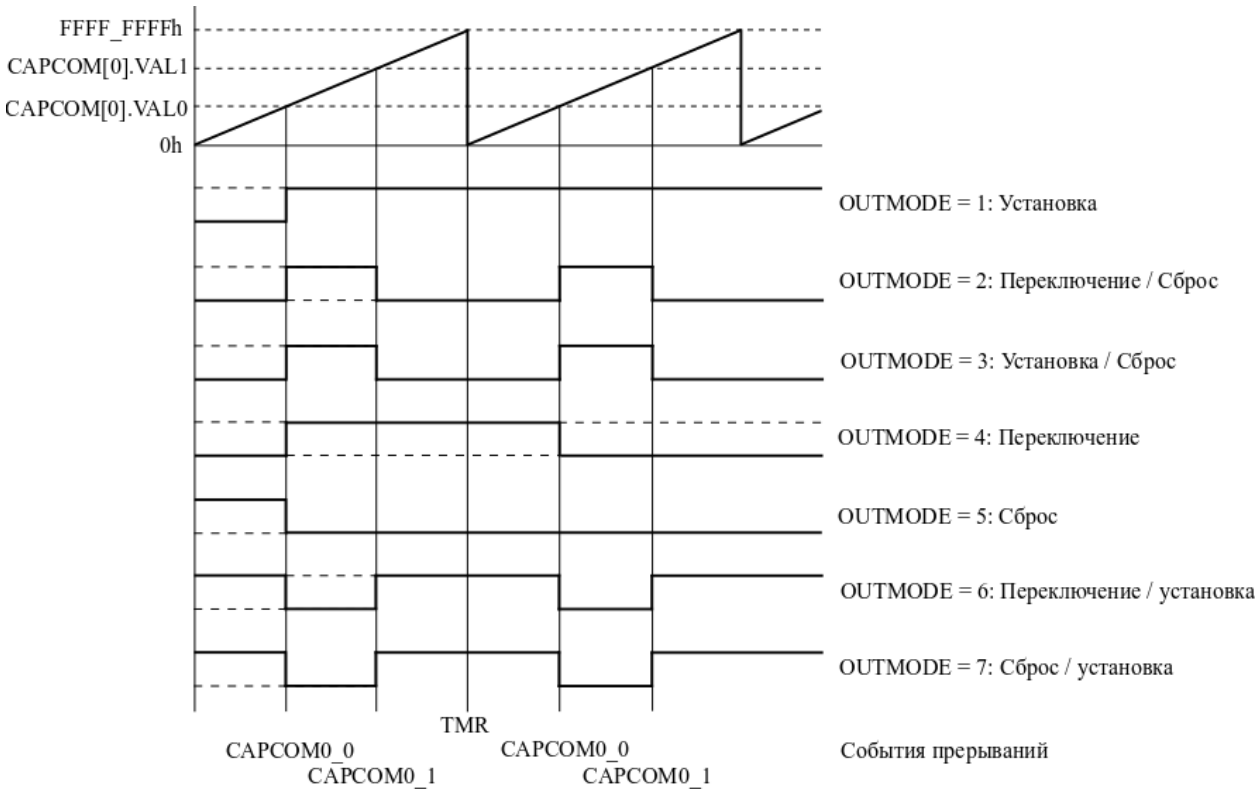


Рисунок 16.14 – Пример вывода. Таймер в непрерывном режиме

Пример вывода — Таймер в режиме счета «вверх/вниз»

Сигнал TMR_OUTn изменяется, когда таймер равен CAPCOM[n].VALx в любом направлении счета и когда таймер равен PERIOD, в зависимости от режима вывода. Пример показан на рисунке 16.15 с использованием PERIOD и CAPCOM[0].VAL0.

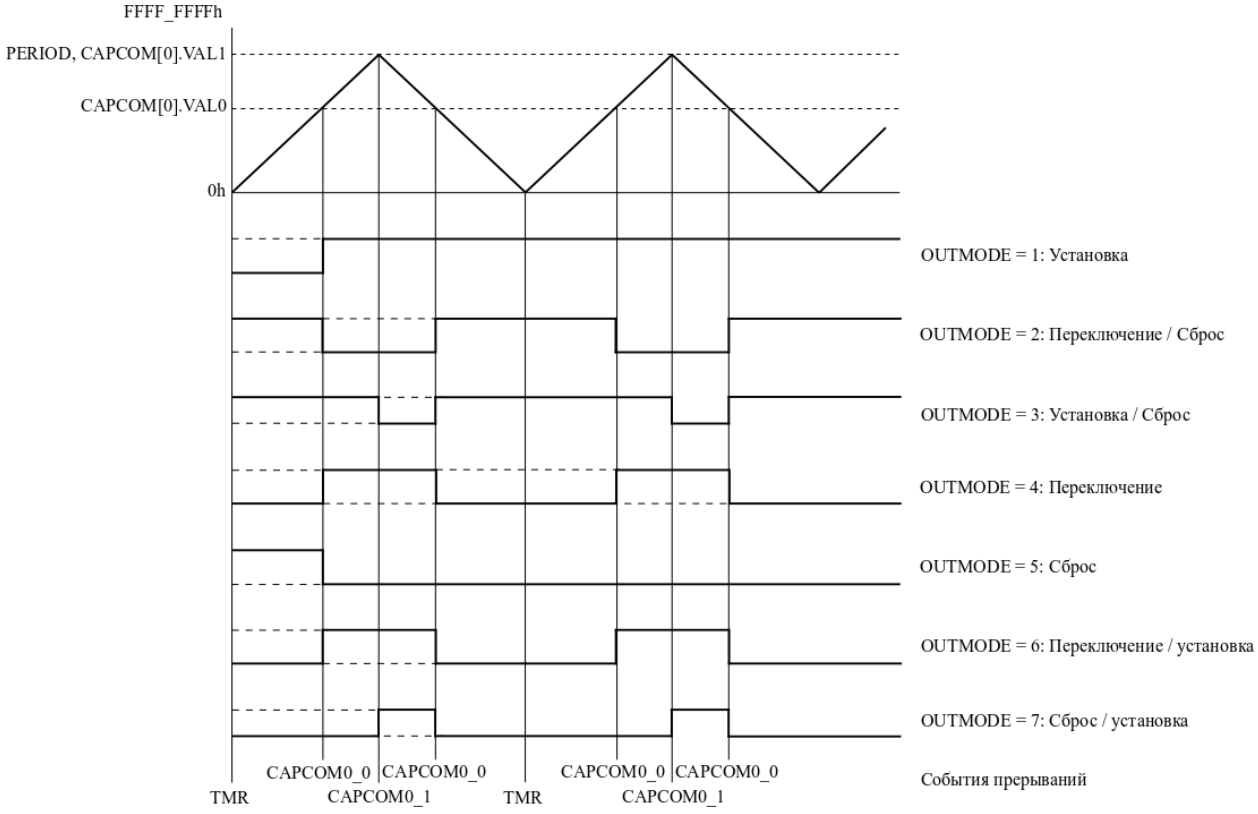


Рисунок 16.15 – Пример вывода. Таймер в режиме счета «вверх/вниз»

Примечание – На рисунке 16.15 значения регистров PERIOD и CAPCOM[0].VAL1 совпадают.

Переключение между режимами вывода

При переключении между режимами вывода во избежание сбоя один из битов OUTMODE регистра CAPCOM[n].CTRL должен оставаться установленным, кроме переключения в режим 0. Безопасным методом переключения между режимами вывода является использование режима вывода 7 в качестве переходного состояния.

16.5 Прерывания таймера

В модуле таймера предусмотрено девять маскируемых источников прерываний для TMR0 – TMR3 и три для TMR4 – TMR15.

Сигналы запросов на прерывания для TMR0 – TMR3:
- TMR – сигнал прерывания от таймера;
- CAPCOM0_0 – сигнал 0 от 0 модуля CAPCOM;
- CAPCOM0_1 – сигнал 1 от 0 модуля CAPCOM;
- CAPCOM1_0 – сигнал 0 от 1 модуля CAPCOM;
- CAPCOM1_1 – сигнал 1 от 1 модуля CAPCOM;
- CAPCOM2_0 – сигнал 0 от 2 модуля CAPCOM;
- CAPCOM2_1 – сигнал 1 от 2 модуля CAPCOM;
- CAPCOM3_0 – сигнал 0 от 3 модуля CAPCOM;
- CAPCOM3_1 – сигнал 1 от 3 модуля CAPCOM.

Сигналы запросов на прерывания для TMR4 – TMR15:
- TMR – сигнал прерывания от таймера;
- CAPCOM_0 – сигнал 0 от модуля CAPCOM;
- CAPCOM_1 – сигнал 1 от модуля CAPCOM.

Каждый из сигналов может быть маскирован путем установки соответствующего бита в регистре маски IM.

Источник прерывания также можно определить, считав состояние регистра RIS или регистра MIS (маскированные прерывания). Сброс прерывания осуществляется программно путём установки соответствующего бита в регистре IC.

Запрос TMR возникает в зависимости от выбранного режима счета:

– при режиме счета ВВЕРХ – при достижении счетчиком значения периода PERIOD;

– при НЕПРЕРЫВНОМ режиме счета – при достижении счетчиком значения 0xFFFF_FFFF;

– при режиме счета ВВЕРХ/ВНИЗ – при достижении счетчиком значения 0x0000_0000.

В режиме захвата запросы CAPCOMn_0 возникают при наступлении события захвата внешнего сигнала в соответствии с конфигурацией модуля CAPCOM[n]. Запросы CAPCOMn_1 в режиме захвата сигнализируют о том, что очередь из регистров CAPCOM[n].VAL0 и CAPCOM[n].VAL1 заполнена, оба регистра записаны и необходимо прочитать регистр CAPCOM[n].VAL1, чтобы не потерять данные.

В режиме сравнения запросы CAPCOMn_x возникают, когда значение счетчика таймера COUNT становится равным значению регистра сравнения (COUNT = CAPCOM[n].VALx).

В контроллер PLIC поступает по 9 линий прерываний от таймеров TMR0-TMR3 и по одной линии прерывания от TMR4-TMR15, являющейся логическим ИЛИ сигналов TMR, CAPCOM_0 и CAPCOM_1.

16.6 Взаимодействие с DMA

Модуль таймера может формировать запросы к контроллеру прямого доступа к памяти DMA. Выбор события для формирования запросов к DMA производится с помощью регистра DMA_TXIM и/или DMA_RXIM. Биты CAPCOM0_0 – CAPCOM3_1 отвечают за разрешение запросов от регистров захвата/сравнения модулей CAPCOM, бит TMR отвечает за разрешение запросов от таймера.

Счетчик таймера может быть автоматически остановлен после завершения всех транзакций TX и/или RX каналов DMA. Для этого, установите биты DMATXSTOP и/или DMARXSTOP в регистре управления таймером CTRL. При возникновении этого события, поле CTRL.MODE обнуляется и счёт останавливается.

16.7 Генерация запросов внешних событий

Модуль таймера может формировать запросы на внешние события (такие как запуск АЦП или ЦАП). Выбор внутреннего события таймера для формирования запроса на внешнее событие производится с помощью регистра EXTEVT_IM.

17 Контроллер интерфейса MSC (Micro Second Channel)

17.1 Общие положения

MSC – это последовательный интерфейс, который специально разработан для подключения внешних устройств питания. Возможность последовательной передачи данных сводит к минимуму количество контактов, необходимых для подключения таких внешних устройств питания. Данные или команда, которые были записаны в модуль через системную шину, отправляется на устройство питания через высокоскоростной синхронный последовательный канал данных (downstream). MSC получает данные и статус обратно от устройства питания через низкоскоростной асинхронный последовательный канал данных (upstream).

Управляющие регистры контроллера MSC приведены в приложении А.11.

На рисунке 17.1 приведено типичное подключение контроллера интерфейса MSC к двум устройствам питания.

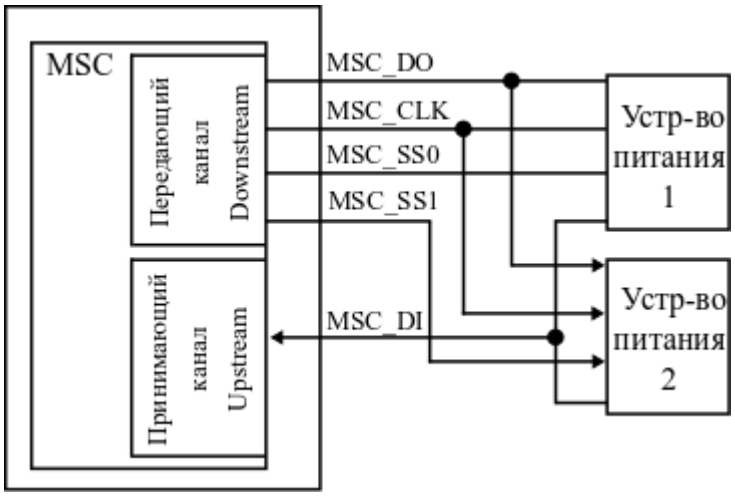


Рисунок 17.1 – Подключение интерфейса MSC к внешним устройствам питания

Возможности высокоскоростного синхронного последовательного передающего канала:

– Частота последовательного тактового выхода: $f_{CLK} = f_{MSC} / 2$ (максимальное значение f_{MSC} соответствует частоте системной шины);

– Дробный делитель тактовой частоты для точного управления частотой тактового сигнала f_{MSC};

– Передача кадров данных, команд и пассивных кадров;

– Запуск передачи кадров может быть инициирован как программным путем, так и в автоматическом режиме;

– Передача кадров данных с битами выбора типа кадра (SEL) или без них;

– Гибкая система генерации сигналов выбора ведомого устройства.

Возможности низкоскоростного асинхронного последовательного канала приема:

– Скорость приема: f_{MSC}, деленная на 4, 8, 16, 32, 64, 128 или 256;

– Стандартные асинхронные последовательные кадры;

– Программируемая длина принимаемого кадра данных (16 или 12 бит);

– Проверка ошибок четности, корректности значений стартового и завершающих битов;

– Встроенная фильтрация высокочастотных выбросов.

17.2 Передающий канал (downstream)

Общие положения

Передающий канал осуществляет высокоскоростную синхронную последовательную передачу данных на внешние устройства. Его 32-битный сдвиговый регистр разделен на две 16-битные части: SRL и SRH. Значение, которое будет загружено в SRL и SRH, поступает из регистра передаваемых данных (D_DATA) или регистра передаваемой команды (D_CMD).

Сигналы разрешения ENL, ENH и ENC (активный уровень – логическая 1) соответствуют определенным фазам последовательной передачи кадра данных или команд относительно тактового сигнала CLK. С помощью функционала регистра IO_CTRL сигналы можно перенаправить на два выхода выбора ведомого MSC_SSn (где n – номер ведомого) и инвертировать (см. рисунок 17.16). Для поддержки драйверов дифференциального вывода последовательный выходной сигнал CLK и последовательный вывод данных DO доступны в обеих полярностях.

Формат кадра

На рисунке 17.2 приведен общий вид кадра передающего канала. Кадр состоит из активной и пассивной фаз. Во время активной фазы происходит передача данных, а во время пассивной фазы данные на DO не передаются. Активная фаза разделена на две части: активная фаза SRL, в которой передается содержимое младшей части сдвигового регистра (SRL), и активная фаза SRH, в которой передается содержимое старшей части сдвигового регистра (SRH). В начале активной фазы SRL и SRH в поток последовательных данных может быть дополнительно вставлен бит выбора (SELL). В кадре, показанном на рисунке 17.2, SELL генерируется в начале активной фазы SRL (не для активной фазы SRH). Передача ведется, начиная с младших бит.

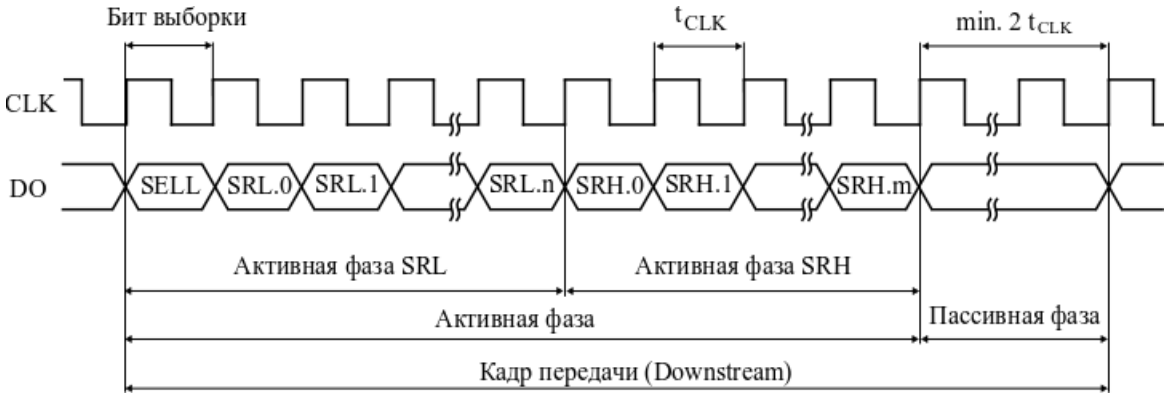


Рисунок 17.2 – Кадр передающего канала

В работе MSC downstream используются три типа форматов кадров:

– Кадры команд, в которых SELL = 1;

– Кадры данных, обозначаемые SELL = 0 или отсутствием бита SELL;

– Кадры пассивного времени, в течении которых ENL, ENH остаются в не активном состоянии.

Кадры команд

Активная фаза кадра команд разделена на две части – активные фазы SRL и SRH. Кадр команды всегда начинается с установленного бита выбора, независимо от того, разрешена или нет вставка бита выбора с помощью поля D_CTRL.FSB. Младшая и старшая части сдвигового регистра (SRL и SRH) объединяются в один 32 битный сдвиговый регистр. Другими словами, всякий раз, когда передаются биты SRH, им всегда предшествует передача полного содержимого SRL. Количество битов, передаваемых во время активных

фаз SRL и SRH (кроме бита выбора), определяется значением в поле D_CTRL.CBN. Диапазон его разрешенных значений лежит от 0 до 32 включительно.

Во время активной фазы кадра команды выходной сигнал разрешения ENC становится активным. Выходные сигналы разрешения ENL и ENH остаются неактивными.

Пассивная фаза кадра команды всегда имеет фиксированную длину (2 * t_{CLK}). Временная диаграмма, приведенная на рисунке 17.3, предполагает, что тактовый сигнал CLK генерируется только во время активной фазы кадра команды (IO_CTRL.CAT = 0).

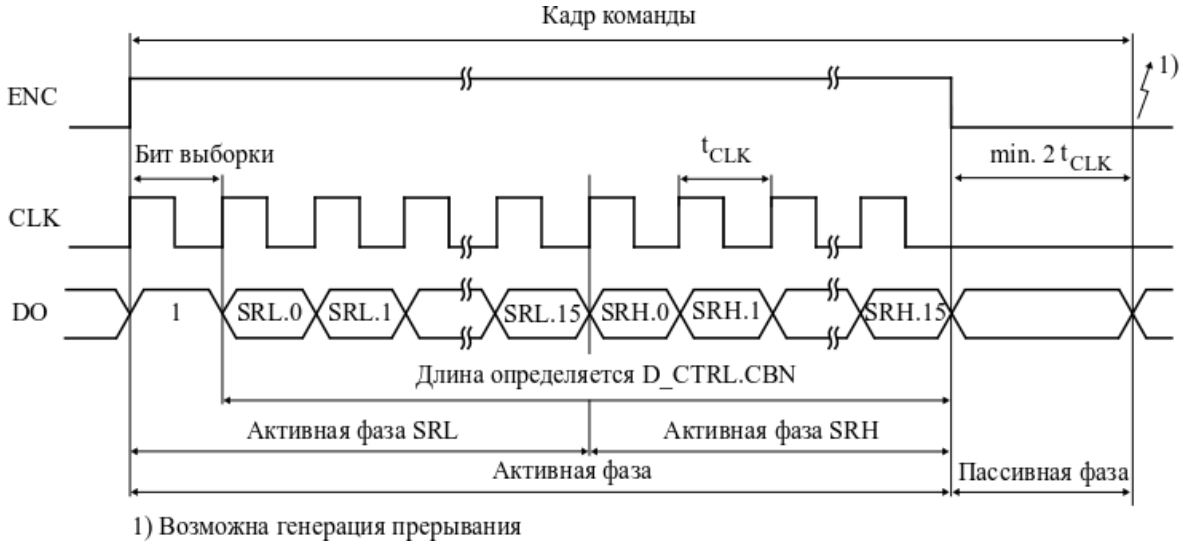


Рисунок 17.3 – Формат кадра команды

В таблице 17.1 продемонстрирована связь запрограммированного количества передаваемых бит и результирующей длительности завершенного кадра команды.

Таблица 17.1 – Длина кадра команды

D_CTRL.CBN	Биты SRL/SRH, передаваемые в активную фазу	Длина кадра команд в t_{MSC_CLK}
000000b	Значащие биты не передаются	1 + 0 + 2 = 3
000001b	SRL[0]	1 + 1 + 2 = 4
000010b	SRL[0:1]	1 + 2 + 2 = 5
000011b	SRL[0:2]	1 + 3 + 2 = 6
…	…	…
001111b	SRL[0:14]	1 + 15 + 2 = 18
010000b	SRL[0:15]	1 + 16 + 2 = 19
010001b	SRL[0:15], SRH[0]	1 + 17 + 2 = 20
010010b	SRL[0:15], SRH[0:1]	1 + 18 + 2 = 21
010011b	SRL[0:15], SRH[0:2]	1 + 19 + 2 = 22
…	…	…
011111b	SRL[0:15], SRH[0:14]	1 + 31 + 2 = 34
100000b	SRL[0:15], SRH[0:15]	1 + 32 + 2 = 35
> 100000b	Значения зарезервированы, не должны использоваться	

Кадры данных

Активная фаза кадра данных разделена на две части – активные фазы SRL и SRH. Количество передаваемых битов можно запрограммировать отдельно для каждой из этих двух частей. Битовое поле D_CTRL.DLBN определяет количество битов SRL, которые передаются во время активной фазы SRL, а D_CTRL.DHBN определяет количество битов SRH, которые передаются во время активной фазы SRH.

Активные фазы SRL и SRH могут начинаться с бита выбора низкого уровня, если это разрешено битами D_CTRL.FSB или D_CTRL.SSB.

Во время активной фазы SRL кадра данных разрешающий выходной сигнал ENL становится активным, а во время активной фазы SRH кадра данных разрешающий выходной сигнал ENH становится активным. Выходной сигнал разрешения ENC остается неактивным.

Если бит D_CTRL.FSB установлен, а значение поля D_CTRL.DLBN не равно 0, то перед передачей содержимого SRL будет вставлен дополнительный бит выбора (SELL) с уровнем логического 0.

Если бит D_CTRL.SSB установлен, а значение поля D_CTRL.DHBN не равно 0, то перед передачей содержимого SRH будет вставлен дополнительный бит выбора (SELH) с уровнем логического 0.

Длина пассивной фазы кадра данных является переменной и определяется битовым полем D_CTRL.PPL. Оно может находиться в диапазоне от 2 * t_{CLK} до 31 * t_{CLK}. Временная диаграмма, приведенная на рисунке 17.4, предполагает, что тактовый сигнал CLK генерируется только во время активной фазы кадра данных (IO_CTRL.CAT = 0), добавлены оба бита выбора типа кадра (D_CTRL.FSB = 1 = D_CTRL.SSB).

В таблицах 17.2 – 17.4 отображено влияние полей DLBN, DHBN, PPL регистра D_CTRL, которые дают наибольший вклад в длительность кадра данных.

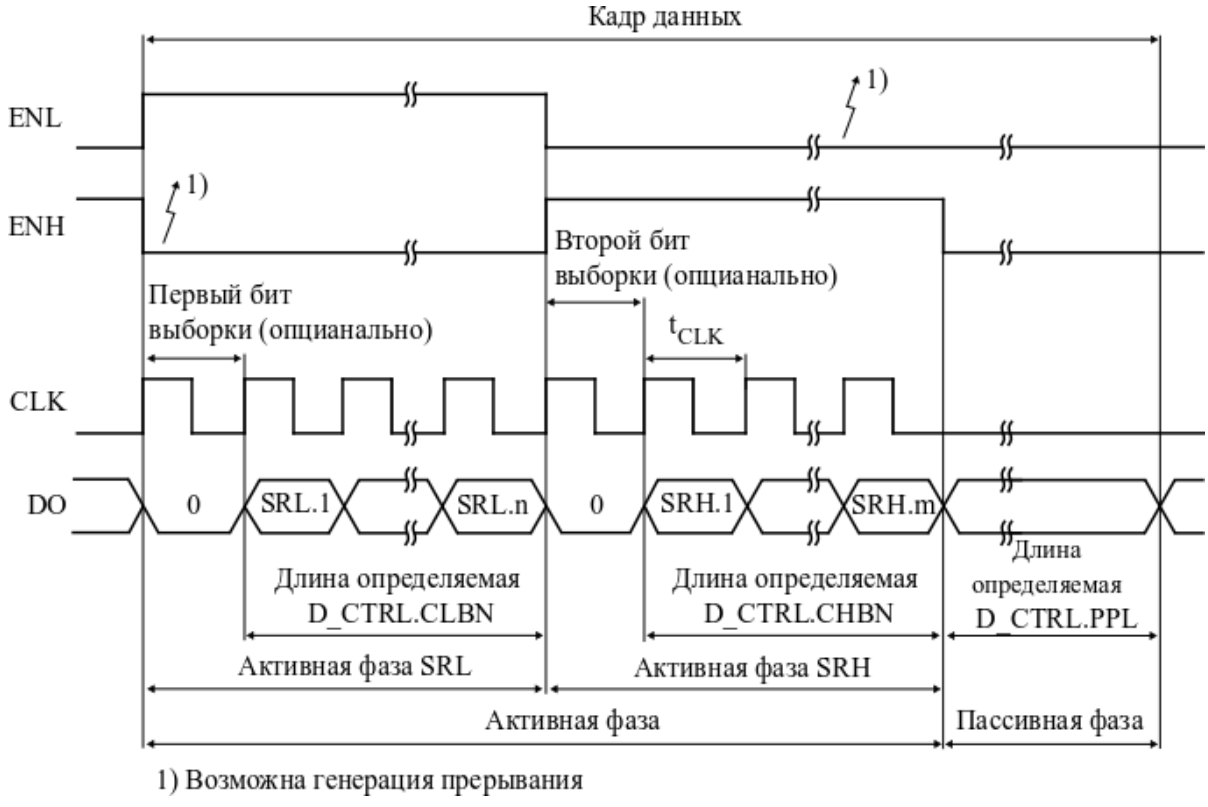


Рисунок 17.4 – Формат кадра данных

Таблица 17.2 – Количество передаваемых из SRL бит данных

D_CTRL.DLBN	Количество бит SRL, передаваемых в активную фазу SRL
00000b	Значащие биты не передаются, значение D_CTRL.FSB игнорируется
00001b	SRL[0]
00010b	SRL[0:1]
00011b	SRL[0:2]
…	…
01111b	SRL[0:14]
10000b	SRL[0:15]
> 10000b	Значения зарезервированы и не должны использоваться

Таблица 17.3 – Количество передаваемых из SRH бит данных

D_CTRL.DHBN	Количество бит SRL, передаваемых в активную фазу SRH
00000b	Значащие биты не передаются, значение D_CTRL.SSB игнорируется
00001b	SRH[0]
00010b	SRH[0:1]
00011b	SRH[0:2]
…	…
01111b	SRH[0:14]
10000b	SRH[0:15]
> 10000b	Значения зарезервированы и не должны использоваться

Таблица 17.4 – Длительность пассивной фазы кадра данных

D_CTRL.PPL	Длительность пассивной фазы
00000b	2 * t_{CLK}
00001b	2 * t_{CLK}
00010b	2 * t_{CLK}
00011b	3 * t_{CLK}
…	…
11110b	30 * t_{CLK}
11111b	31 * t_{CLK}

Следующая формула определяет длительность кадра данных:

$$N_{cycles} = FSB + DLBN + SSB + DHBN + PPL, \quad (17.1)$$

где вместо составляющих слагаемых нужно подставить значение одноименных полей регистра D_CTRL.

Примечание – Вместо FSB следует подставлять 0, если D_CTRL.DLBN = 0; вместо SSB следует подставлять 0, если D_CTRL.DHBN = 0; вместо PPL следует подставлять 2, если D_CTRL.PPL < 2.

Кадр пассивного времени

Кадр пассивного времени имеет такую же длительность, что и кадр данных. Они генерируются только в режиме повторяющейся передачи данных (Data Repetition mode). При особых условиях (в случае вставки кадра команды) кадр пассивного времени может быть укорочен (см. рисунок 17.6).

Выход тактового сигнала CLK может как переключаться, так и находится в неактивном состоянии (в зависимости от выбора бита IO_CTRL.CAT). Сигналы включения ENL и ENH остаются на низком уровне в течение всего кадра пассивного времени, ведомое устройство должно игнорировать значение вывода данных DO.

Режим передачи по программному запросу (Triggered mode)

В режиме передачи по программному запросу кадры команд или кадры данных отправляются в результате программного события. Когда передача кадра завершена и не было выставлено новых запросов, передающий канал возвращается в состояние ожидания.

Установка бита D_CTRL.MODE (режим повторяющейся передачи данных) немедленно запускает передачу кадра с использованием значений, существующих на данный момент.

Запись в регистр D_CMD приводит к аппаратной установке бита ожидания передачи кадра команды (D_STAT.CP). Если передающий канал находится в состоянии ожидания, то немедленно начнется отправка кадра команды. Если на момент записи ведется передача кадра данных или команды, то передача нового кадра команды будет выполнена после завершения текущего активного кадра. Бит D_STAT.CP очищается аппаратно.

Если передающий канал свободен и бит ожидания передачи кадра данных (D_STAT.DP) был установлен путем записи логической 1 в бит FLAG_CTRL.SET_DP, то немедленно начнется отправка кадра данных. Если на момент записи ведется передача кадра данных или команды, то передача нового кадра данных будет выполнена после завершения текущего активного кадра. Бит D_STAT.DP очищается аппаратно.

Кадр команды всегда имеет приоритет над кадром данных. На рисунке 17.5 приведена блок-схема режима передачи по программному запросу. На этой диаграмме особый акцент сделан на поведение битов ожидания передачи данных и команды (D_STAT.DP и D_STAT.CP). Если установлены оба бита ожидания передачи кадра (D_STAT.DP = D_STAT.CP = 1), сначала всегда будет отправляться кадр команды, а затем кадр данных (при условии, что в процессе не будет выставлен новый запрос на передачу кадра команды).

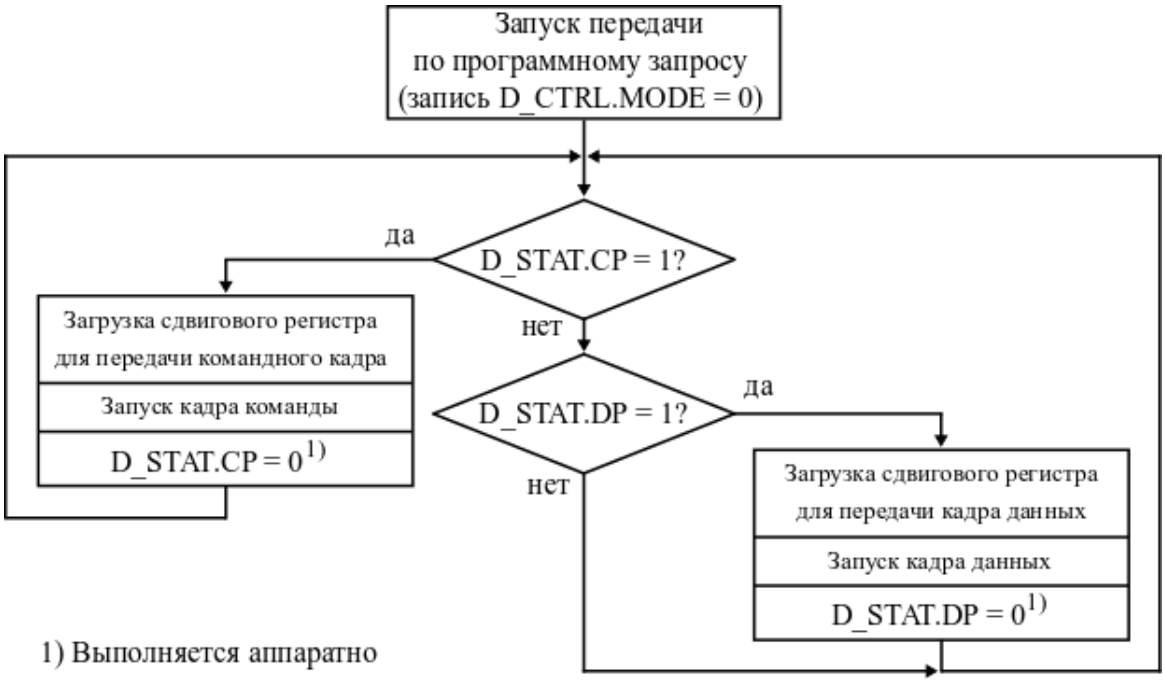


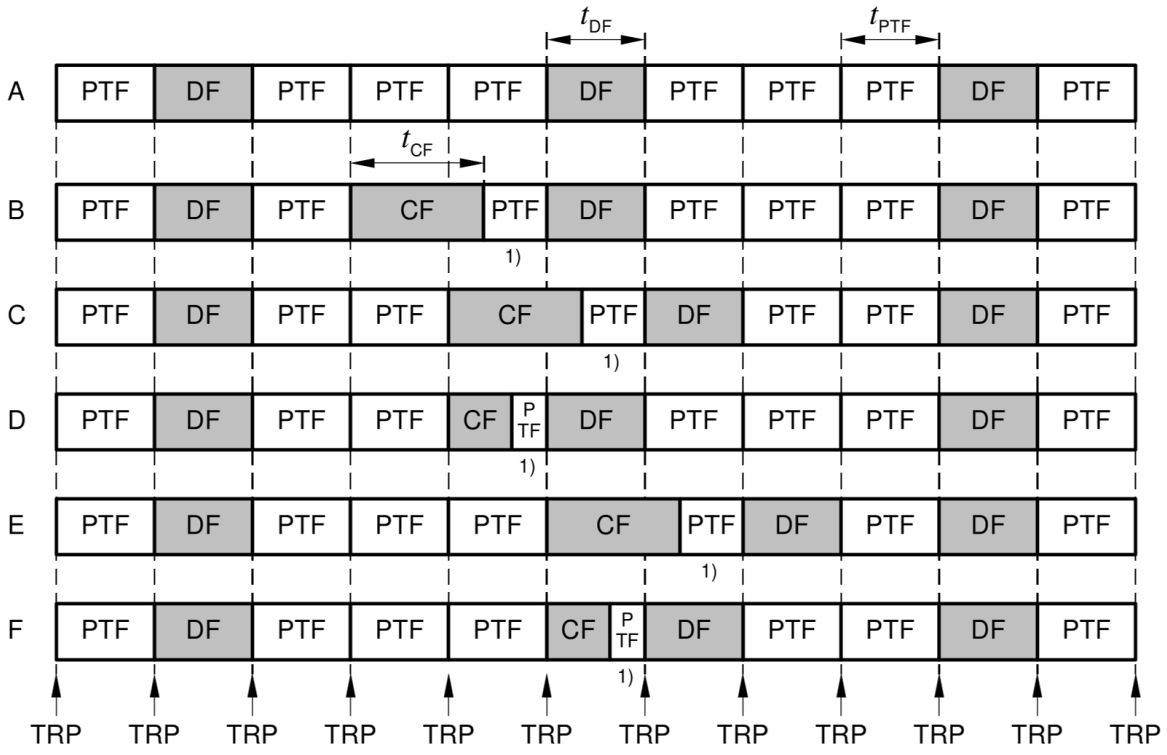
Рисунок 17.5 – Алгоритм работы в режиме передачи по программному запросу

Тип активного кадра, который в данный момент передается, характеризуется двумя флагами состояния D_STAT.DFA и D_STAT.CFA, которые устанавливаются во время передачи кадра данных или команды соответственно. Кроме того, счетчик D_STAT.BC передающего канала указывает количество периодов тактового сигнала сдвига, прошедших с момента начала текущего кадра.

В момент старта кадра команды выполняется загрузка сдвигового регистра значением из регистра D_CMD, буферизируются значение поля D_CTRL.CBN. При запуске кадра данных значение из регистра D_DATA заносится в сдвиговый регистр, буферизируются значения полей PPL, SSB, FSB, DHBN, DLBN регистра D_CTRL. Изменение значения D_CTRL не оказывает влияния на структуру и формат текущего передаваемого кадра.

Режим повторяющейся передачи данных (Data Repetition mode)

В режиме повторяющейся передачи данных кадры данных отправляются непрерывно без какого-либо взаимодействия с программным обеспечением. Между двумя последовательными кадрами данных могут быть вставлены временные промежутки бездействия – кадры пассивного времени. Количество добавляемых кадров пассивного времени (от 0 до 15) определяется значением битового поля D_CTRL.TFN. Продолжительность кадра данных (t_{DF}) и кадров пассивного времени (t_{PTF}) определяется параметрами кадра данных, согласно формуле 17.1. Эти параметры определяют временные точки отсчета (TRP), в которых запускаются данные, команды или пассивный временной интервал (см. диаграмму A на рисунке 17.6).



PTF - Кадр пассивного времени; DF - Кадр данных; CF - Кадр команды; TRP - Точка отсчета времени

1) Данные кадры пассивного времени укорочены

Рисунок 17.6 – Пример следования кадров в режиме повторяющейся передачи данных

Автоматическая генерация кадра данных управляется битом ожидания передачи кадра данных D_STAT.DP. Этот бит устанавливается в конце последнего кадра пассивного времени, если кадр команды передается или начинается с этого TRP, отменяя запланированную отправку кадра данных. D_STAT.DP очищается аппаратно, когда на следующем TRP стартует ожидающий передачи кадр данных. Кадры данных всегда привязаны к контрольным точкам времени. Это означает, что они всегда начинаются с TRP. Кадры пассивного времени могут быть укороченными. Это главным образом происходит при вставке кадров команд.

Непрерывная передача кадров данных может быть прервана вставкой кадра команды, которая инициируется программным обеспечением. Когда записывается регистр D_CMD, аппаратно устанавливается бит ожидания передачи кадра команды D_STAT.CP, который указывает на то, что MSC запустит кадр команды на следующем TRP, независимо от того, должен ли со следующего TRP начинаться кадр данных (D_STAT.DP = 1) или кадр пассивного времени. Это также означает, что кадры команд всегда привязаны к контрольным точкам времени.

Диаграммы B – F на рисунке 17.6 показывают вставку кадра команды в режиме повторяющейся передачи данных.

На диаграмме B кадр команды был запрошен в течение первого пассивного периода времени после кадра данных и вставлен в следующий TRP. На диаграммах C и D кадр команды был запрошен во время второго пассивного временного интервала и вставлен в момент номинального начала последнего кадра пассивного времени.

Когда длина кадров команды и данных отличаются друг от друга (это случай рассмотрен на диаграммах от B до F), вставляется сокращенный кадр пассивного времени, который длится до следующего TRP. Это гарантирует, что следующие данные или обычный кадр пассивного времени снова будут соответствовать TRP. Важно отметить, что при нулевом значении поля D_CTRL.TFN вставка кадров пассивного времени не осуществляется, кадр данных будет отравлен сразу после передачи кадра команды.

При передаче последнего кадра пассивного времени и в случае конфликта с кадром команды, будет аппаратно установлен бит D_STAT.DP. Это приведет к запуску кадра данных при достижении следующего TRP.

Тип активного кадра, который в данный момент передается, отображается двумя флагами состояния – D_STAT.DFA, D_STAT.CFA. Они устанавливаются во время передачи кадра данных и кадра команды соответственно. Кроме того, счетчик D_STAT.BC передающего канала указывает количество периодов тактового сигнала сдвига, прошедших с момента начала текущих данных, команды или пассивного временного кадра.

Загрузка сдвигового регистра и буферизация используемых полей регистра D_CTRL осуществляется также, как и в режиме передачи по командному запросу.

В режиме повторения данных счетчик кадров пассивного времени D_STAT.TFC показывает, сколько таких кадров уже было передано после появления последнего регулярного кадра данных. Счетчик кадров пассивного времени ведет отсчет от 0 до значения, записанного в битовое поле D_CTRL.TFN.

Значение D_STAT.TFC равное 0 указывает, что кадр данных запрошен для передачи.

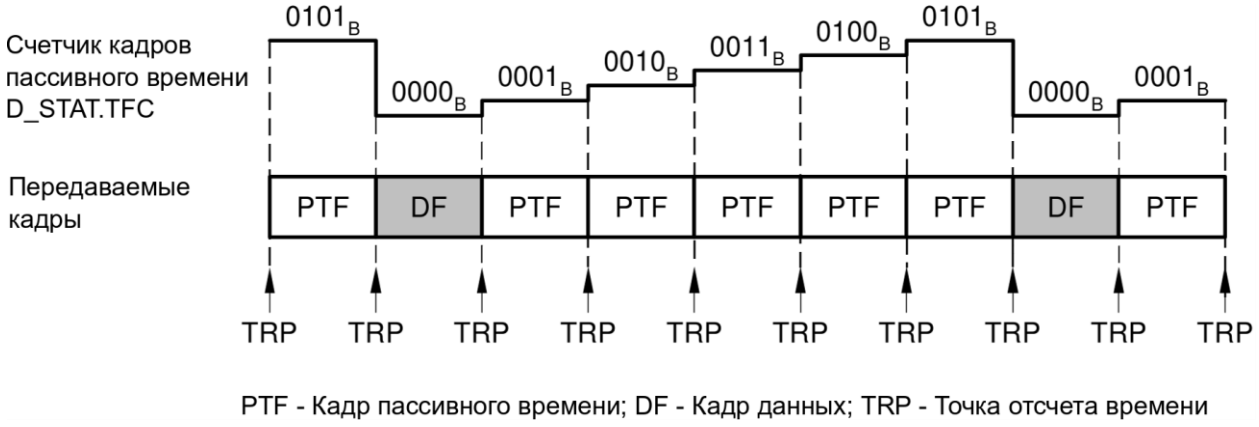


Рисунок 17.7 – Пример работы счетчика кадров пассивного времени

Счетчик бит и сигналы разрешения передающего канала

Во время работы нисходящего канала 7-битный нисходящий счетчик D_STAT.BC подсчитывает периоды тактового сигнала сдвига CLK. При загрузке сдвигового регистра счетчик тактов передающего канала сбрасывается в 0, счет ведется до конца пассивной фазы кадра.

В режиме передачи по программному запросу счетчик тактов передающего канала прекращает отсчет в конце пассивной фазы, сбрасывается и ждет запуска нового кадра.

В режиме повторяющейся передачи данных счетчик тактов передающего канала не останавливается в конце пассивной фазы, а сбрасывается и начинает отсчет снова, независимо от того, следует ли далее кадр данных, команды или пассивного времени.

На рисунке 17.8 показан пример передачи кадра данных, в котором бит выбора типа кадра добавляется только в активную фазу SRL, но не в SRH (FSB = 1, SSB = 0 в регистре D_CTRL).

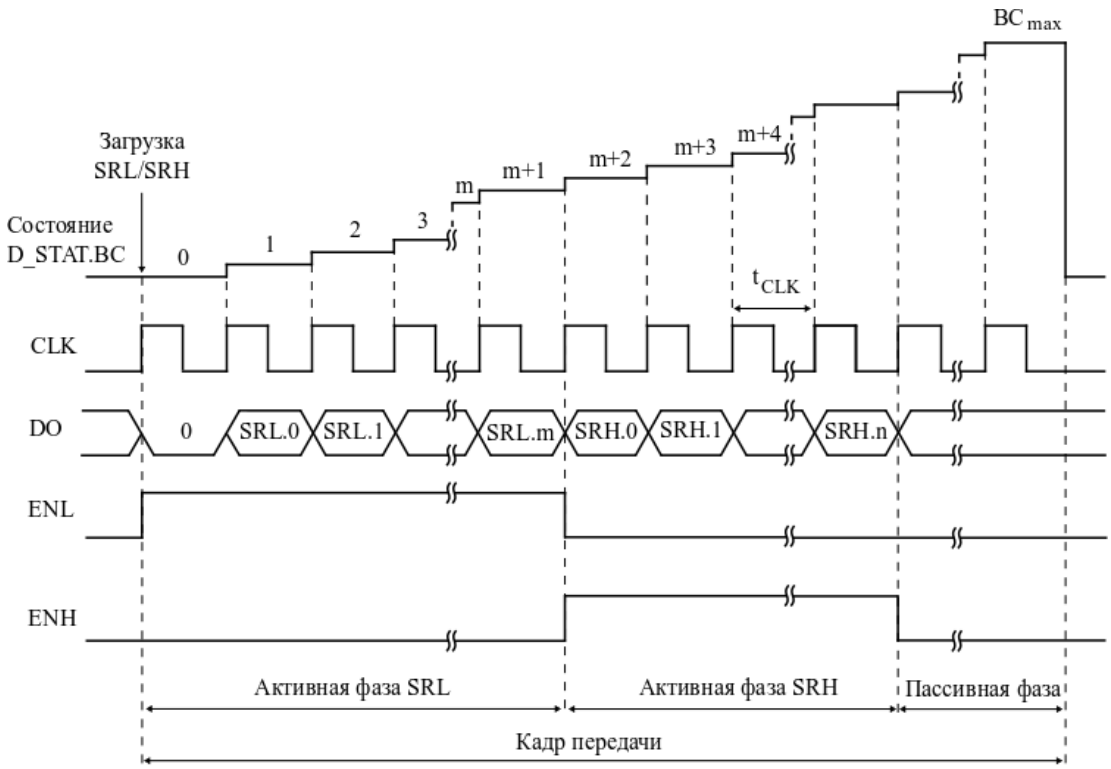


Рисунок 17.8 – Пример работы счетчика бит передающего канала при FSB = 1 и SSB = 0

При загрузке сдвигового регистра SRL/SRH счетчик тактов передающего канала сбрасывается и затем начинает отсчет с каждым тактом CLK до конца пассивной фазы. Сигнал выбора ведомого младшей части кадра данных (ENL) устанавливается одновременно с началом бита выбора типа кадра в активную фазу SRL.

Скорость передачи

Скорость в бодах канала передачи определяется частотой последовательного тактового сигнала f_{CLK} и всегда равняется $f_{MSC} / 2$. Частота генерируемого тактового сигнала f_{MSC} определяется значениями частоты системной шины и регистра дробного делителя входного сигнала (FDR).

Преждевременное прекращение передачи кадра

Только сигнал сброса может прервать передачу текущего кадра. Отключение передающего канала (отображается в поле D_STAT.DD), выполняемое с помощью записи логической единицы в бит FLAG_CTRL.SET_DD, будет препятствовать только старту новых кадров. Текущая передача будет полностью завершена, не смотря на установленный флаг D_STAT.DD.

17.3 Канал приема (upstream)

Общие положения

Канал приема интерфейса MSC представляет собой асинхронный последовательный приемник, основанный на стандартизированном протоколе асинхронной передачи данных. Он предназначен для приема потока последовательных данных от периферийного устройства через вход последовательного данных MSC_DI с использованием двух конкретных форматов кадров данных.

На рисунке 17.9 представлена структурная схема канала приема MSC.

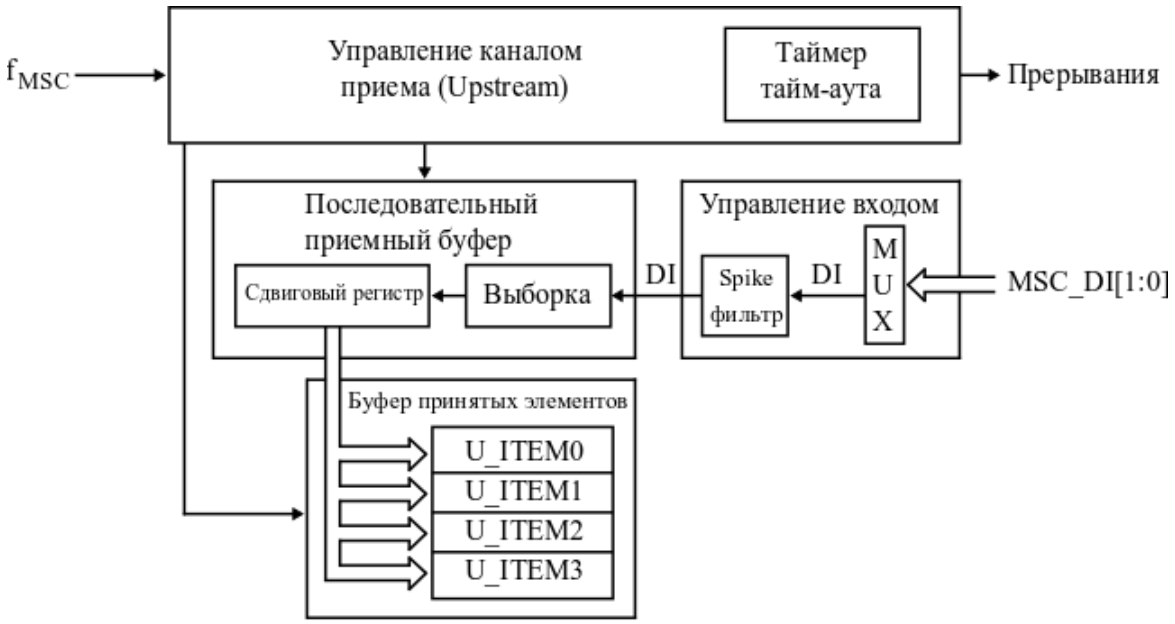


Рисунок 17.9 – Структурная схема канала приема

Выборка данных со входа DI осуществляется после фильтрации от высокочастотных выбросов. Обнаруженные значения с последовательного входа попадают в сдвиговый регистр. После полного приема кадра последовательных данных содержимое сдвигового регистра передается в одну из четырех записей канала приема. Если это разрешено в регистре IRQ_EN, то генерируется соответствующее прерывание.

Скорость передачи данных напрямую связана с тактовой частотой модуля f_{MSC} и может находиться в диапазоне от fMSC / 4 до f_{MSC} / 256.

Кадры данных

Кадры асинхронных данных, используемые восходящим каналом, состоят из четырех основных частей:

1. Один стартовый бит, всегда имеющий уровень логического нуля.

2. 8-битное поле данных D[7:0], принимаемое начиная с младшего бита.

3. Необязательное 4-битное адресное поле A[3:0], принимаемое начиная с младшего бита.

4. Один бит четности и два завершающих бита, которые всегда имеют уровень логической единицы.

Прерывание приема кадра данных генерируется после второго завершающего бита.

Как показано на рисунке 17.10, 16-битный кадр данных канала приема включает в себя дополнительное 4-битное поле адреса. Тип кадра приема выбирается битом U_CTRL.FT. Если он сброшен, то ожидается прием 12-битных кадров данных. Если бит U_CTRL.FT установлен, то используется 16-битный формат принимаемых кадров данных, который содержит в себе 4-битное поле адреса.

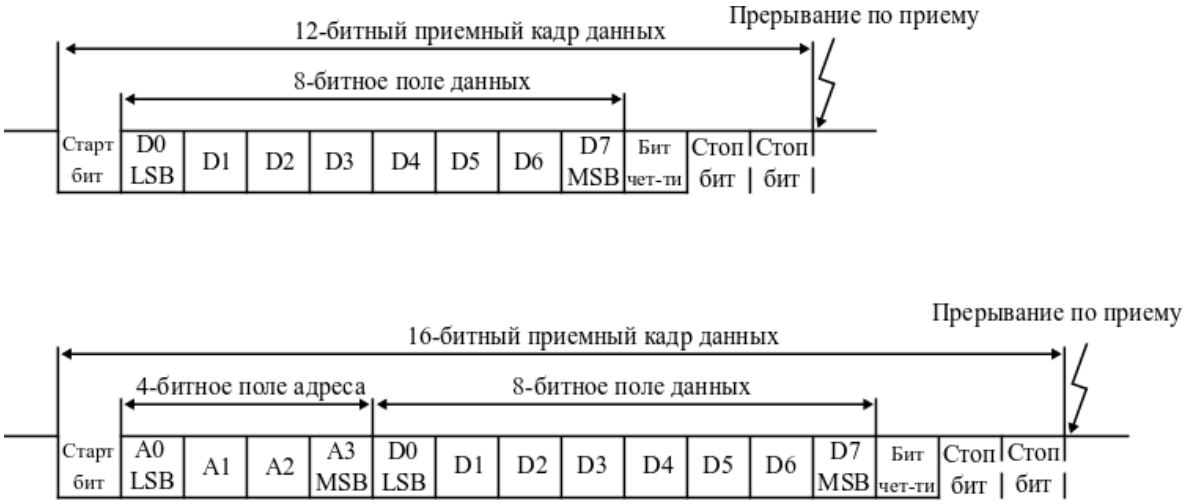


Рисунок 17.10 – Типы кадров канала приема

Проверка корректности принятого кадра

Основным средством проверки корректности принятого кадра данных является контроль значения бита четности. Когда обнаруживается ошибка четности, устанавливается флаг ошибки четности ERR_P в соответствующей записи канала приема U_ITEM[n]. Регистры U_ITEM[n] также хранят бит четности принятых кадров данных (U_ITEM[n].RP).

Бит U_CTRL.PT определяет выбранный режим вычисления ожидаемого значения бита четности принимаемого кадра. При U_CTRL.PT = 0 выбирается режим дополнения до четного. Бит четности будет установлен, если суммарное количество установленных бит в полях адреса и данных нечетное. При U_CTRL.PT = 1 выбирается режим дополнения до нечетного. Бит четности будет установлен, если суммарное количество установленных бит в полях адреса и данных четное. Для 12-битных кадров поле адреса следует считать равным 0, оно не влияет на конечный результат.

Помимо проверки четности контролируется, что на момент выборки значение стартового бита осталось логическим 0. При невыполнении данного условия будет установлен бит флага ERR_HS. Также осуществляется контроль корректности завершающих битов. Если хотя бы один из них на момент выборки не будет иметь уровень логической 1, то будет установлен бит ERR_LS.

Стоит обратить внимание, что установка одного или нескольких флагов ошибки не предполагает генерации отдельного прерывания. Все биты ошибок в каждой действующей записи должны проверяться программным обеспечением.

Прием данных

Прием кадра данных начинается с обнаружения ниспадающего фронта на линии DI (переход из 1 в 0). Производится буферизация значений полей FT, RBR, PT регистра U_CTRL. Новые значения типа принимаемого кадра, скорости приема и типа четности вступят в силу только после завершения приема текущего кадра. Далее следует выборка входящих последовательных данных в буфер приема. После обнаружения второго завершающего бита содержимое приемного буфера заносится в одну из записей канала приема (U_ITEM[n], где n определяется значением старших разрядов поля адреса). Затем схема приема ожидает следующего стартового бита (переход из 1 в 0) на линии DI. Когда содержимое буфера приема передано в U_ITEM[n], выполняется аппаратная установка бита VALID в соответствующей записи канала приема. Если это разрешено с помощью IRQ_CTRL.IRI*, то происходит установка бита IRQ_STAT.RD. Если был установлен бит IRQ_CTRL.RD, генерируется запрос на прерывание, вне зависимости от наличия или отсутствия ошибок в ходе приема (U_ITEM[n].ERR_*).

Чтение регистров U_ITEM[n] не является деструктивным, то есть не приводит к автоматическому изменению каких-либо битов.

Примечание – Входная линия DI представляет собой отфильтрованный не инвертированный (IO_CTRL.IU = 0) или инвертированный (IO_CTRL.IU = 1) входной сигнал с выбранной линии SDI, индекс которой задается значением поля IO_CTRL.UI.

Прием 16-битного кадра данных (с 4 битным полем адреса) активируется установкой бита U_CTRL.FT (см. рисунок 17.11). Когда содержимое приемного буфера полностью получено, оно передается в один из четырех регистров U_ITEM[n]. Два старших бита адреса ADDR[3:2] полученного 4 битного поля адреса задают номер n регистра U_ITEM[n], в который будет сохранено содержимое принятого кадра.

Регистр записи канала приема содержит поля адреса (ADDR), данных (DATA), полученного бита четности (RP), бит доступности (VALID), указывающий на то, что регистр U_ITEM[n] содержит полученный кадр данных, биты ошибок приема логического уровня стартового бита (ERR_HS), бита четности (ERR_P), логического уровня завершающих бит (ERR_LS).

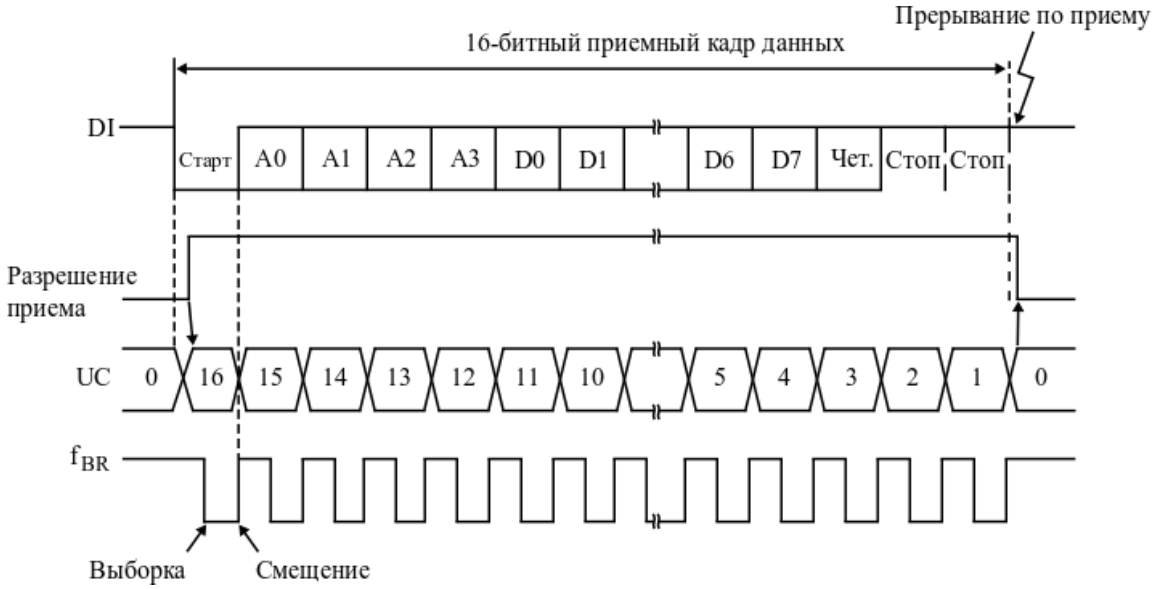


Рисунок 17.11 – Прием 16-битного кадра

Текущее состояние принимаемого кадра характеризуется содержимым счетчика бит, который отображается в 5 битное поле U_CRTL.UC. Как показано на рисунке 17.11, счетчик бит канала приема загружается значением 10000b при обнаружении стартового бита. Он ведет обратный отсчет и снова достигает 00000b после получения второго завершающего бита, когда прием кадра завершен.

Выборка значения с линии последовательных входных данных DI производится близко к середине предполагаемой длительности такта принимаемого кадра. В конце такта оно сдвигается в приемный буфер. Частота тактового сигнала сдвига f_{BR} зависит от выбранной скорости передачи данных (см. формулу 17.2).

Прием кадров данных без поля адреса активируется сбросом U_CTRL.FT. Схема приема аналогична таковой для 16-битного кадра данных за исключением нескольких отличий:

– В счетчик бит канала приема изначально загружается значение 01100b;
– Содержимое полученного кадра всегда сохраняется в регистр U_ITEM[0];
– В битовое поле U_ITEM[0].ADDR при сохранении кадра всегда заносится 0h.

Скорость приема

Скорость в бодах канала приема является производной от тактового сигнала модуля MSC – f_{MSC}. Схема тактирования канала приема приведена на рисунке 17.12.

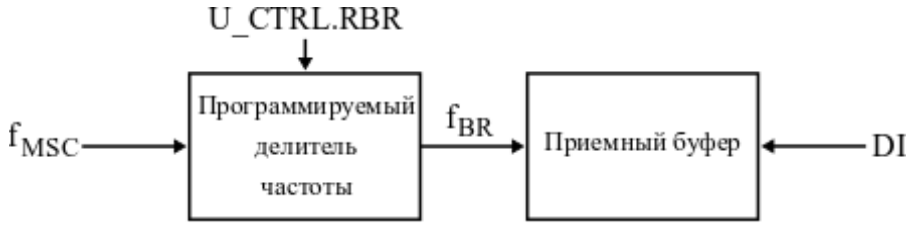


Рисунок 17.12 – Схема тактирования канала приема

Измерение значения на последовательном входе данных DI выполняется с частотой f_{BR} в середине битового интервала. Тактовый сигнал скорости передачи данных f_{BR} получается из f_{MSC} с помощью целочисленного программируемого делителя тактового сигнала, управляемого с помощью значения поля U_CTRL.RBR (см. таблицу 17.5). Частота f_{BR} определяет ширину битового интервала и, следовательно, скорость приема данных.

Итоговая формула, которая используется при расчете скорости передачи данных при U_CTRL.RBR не равном 0:

$$Receive_baud_rate = f_{MSC} / 2^{(U_CTRL.RBR + 1)} \quad (17.2)$$

Таблица 17.5 – Значения делителя DF и скорости приема в бодах

U_CTRL.RBR	Значение делителя DF	Скорость приема
000b	Прием запрещен	–
001b	4	f_{MSC} / 4
010b	8	f_{MSC} / 8
011b	16	f_{MSC} / 16
100b	32	f_{MSC} / 32
101b	64	f_{MSC} / 64
110b	128	f_{MSC} / 128
111b	256	f_{MSC} / 256

Примечание – При U_CTRL.RBR = 0 канал приема отключается, мониторинг линии входных данных прекращается.

Содержимое битового поля U_CTRL.RBR определяет значение, при котором выполняется сброс внутреннего счетчика выборки, который тактируется с помощью f_{MSC}. На рисунке 17.13 показана работа счетчика выборки в начале принимаемого кадра при коэффициенте деления DF равном 8 (U_CTRL.RBR = 2), что означает восемь тактов выборки на каждый битовый интервал кадра.

Когда канал приема находится в состоянии бездействия, он ожидает спадающего фронта на входе DI (переход из 1 в 0). При обнаружении данного события счетчик выборок начинает инкрементироваться. Он сбрасывается при достижении выбранного коэффициента деления DF, возможные значения которого приведены в таблице 17.5. В середине диапазона счетчика выборки оценивается логическое состояние DI и, в случае бита данных, фиксируется в сдвиговом регистре приемного буфера. При перезагрузке счетчика выборок в сдвиговый регистр заносится очередной полученный бит.

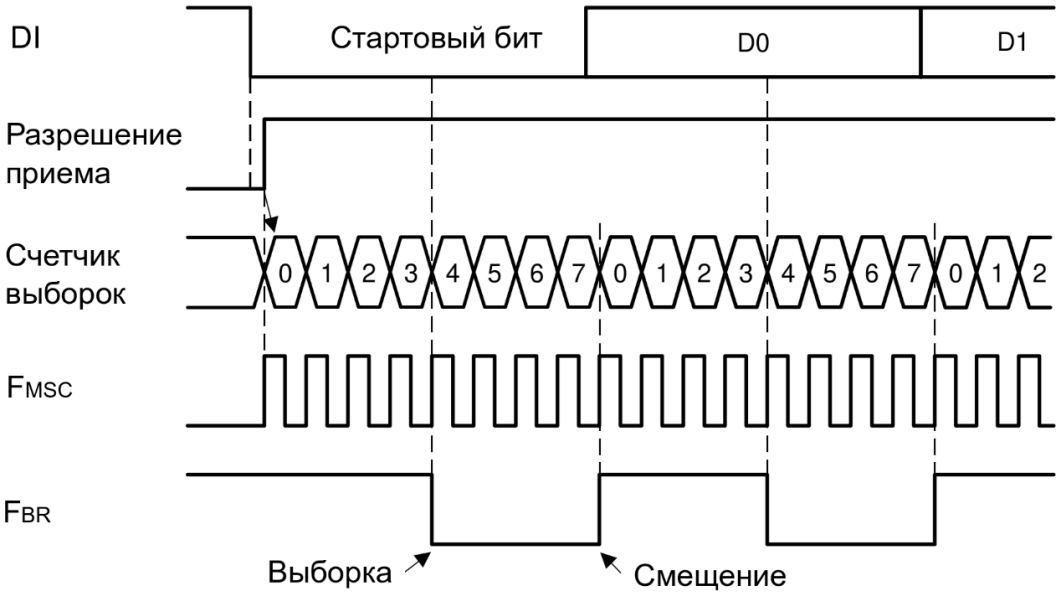


Рисунок 17.13 – Пример выборки значений для RBR = 010b

Фильтрация выбросов по входу данных

Выбранная с помощью поля IO_CTRL.UI, входная линия MSC_DI (значение может быть инвертировано, если установлен бит IO_CTRL.IU) канала приема семплируется в фильтре импульсных помех (spike filter), который тактируются с помощью f_{MSC}. Данный фильтр высокочастотных выбросов представляет собой цепочку триггеров с логикой принятия решения по мажоритарному принципу (2 из 3). Значение, которое встречается как минимум дважды в трех выборках, принимается в качестве входного значения данных DI.

Таймер тайм-аута

Таймер тайм-аута контролирует временной интервал между двумя последовательными кадрами канала приема. Таймер перезагружает свои начальные значения из регистра U_CTRL и начинает отсчет в нисходящем направлении. Если временной интервал между двумя кадрами превышает допустимый, счетчик достигает нулевого значения и прекращает счет. Следующий отсчет, сформированный предделителем, вызывает установку флага прерывания по тайм-ауту канала приема (IRQ_STAT.UTO). Программно флаг может быть сброшен или установлен записью логической 1 в биты FLAG_CTRL.CLR_UTO или FLAG_CTRL.SET_UTO соответственно. Текущее значение счетчика отображается в поле U_CATRL.TOC.

Счетчик перезагружается и начинает считать по трем событиям:
– Сброс модуля (по умолчанию предделитель таймера выключен);
– Запись битовых полей U_CTRL.TOP или U_CTRL.TOV;
– Обнаружение заднего фронта начального бита кадра.
На рисунке 17.14 демонстрируется работа таймера тайм-аута канала приема.

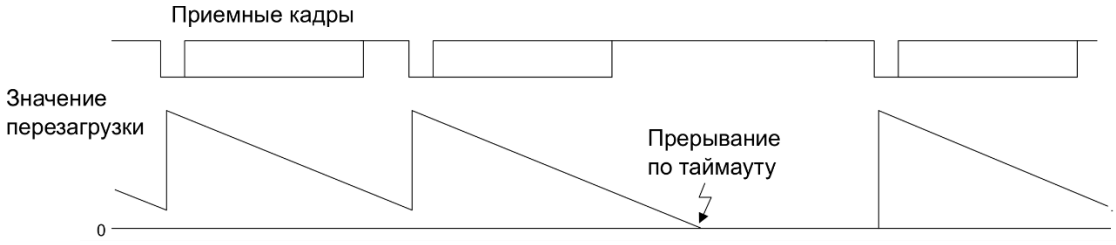


Рисунок 17.14 – Работа таймера тайм-аута

Таймер тайм-аута ведет подсчет в интервалах времени, которые требуются на прием одного бита (BitTime). Для U_CTRL.TOP не равного 0 временное окно, в котором должен начаться прием следующего кадра, можно рассчитать по формуле:

$$Ttime\text{-}out = BitTime * 2(U_CTRL.TOP - 1) * (U_CTRL.TOV + 1). \quad (17.3)$$

17.4 Прерывания

Прерывание при передаче кадра данных (Data Frame interrupt)

Флаг передачи кадра данных (IRQ_STAT.DF) аппаратно устанавливается при старте передачи первого или последнего бита данных (если бит IRQ_EN.DFEI сброшен). Моменты, в которые может быть установлен запрос на прерывание, приведены на рисунке 17.4. IRQ_STAT.DF может быть программно установлен или сброшен записью логической 1 в бит SET_DF или CLR_DF регистра FLAG_CTRL.

Сброс бита IRQ_EN.DF приводит к запрету на формирование прерывания при установленном флаге IRQ_STAT.DF.

Прерывание при передаче кадра команды (Command Frame interrupt)

Флаг передачи кадра команды (IRQ_STAT.CF) аппаратно устанавливается непосредственно после завершения активной фазы передачи кадра команды. Он может быть программно установлен или сброшен записью логической 1 в бит SET_CF или CLR_CF регистра FLAG_CTRL

Сброс бита IRQ_EN.CF приводит к запрету на формирование прерывания при установленном флаге IRQ_STAT.CF.

Прерывание при завершении кадра пассивного времени (Passive Time Frame interrupt)

Флаг завершения передачи кадра пассивного времени (IRQ_STAT.TF) аппаратно устанавливается в конце пассивной фазы кадра пассивного времени. Момент, в который может быть установлен запрос на прерывание, приведен на рисунке 17.3. IRQ_STAT.TF может быть программно установлен или сброшен путем записи логической 1 в биты FLAG_CTRL.SET_TF или FLAG_CTRL.CLR_TF соответственно. Программный сброс флага имеет больший приоритет по отношению к программной или аппаратной установке.

Сброс бита IRQ_EN.TF приводит к запрету на формирование прерывания при установленном флаге тайм-аута канала приема (IRQ_STAT.TF).

Прерывание после получения кадра данных

Всякий раз, когда канал приема обновляет данные в регистрах U_ITEM[n] (n = 0-3), модуль MSC может сгенерировать прерывание, если это разрешено значениями полей IRQ_EN.RD и IRQ_EN.IRI*.

Некоторые ведомые устройства с интерфейсом MSC могут пакетно отправлять по 4 кадра данных с инкрементом поля адреса на 0100b, которые будут подать в различные записи канала приема. С помощью полей IRQ_EN.IRI* можно замаскировать часть событий обновления записей канала приема, например, чтобы генерировать одно прерывание после получения последнего в пакете кадра.

IRQ_STAT.RD – это флаг приема кадра данных, который может быть поднят аппаратно, установлен или сброшен программным путем записи логической 1 в биты FLAG_CTRL.SET_RD и FLAG_CTRL.CLR_RD соответственно. Программный сброс флага имеет больший приоритет по отношению к программной или аппаратной установке.

Сброс бита IRQ_EN.RD приводит к запрещению формирования прерывания при установленном флаге прерывания после получения кадра данных (IRQ_STAT.RD).

Прерывание по тайм-ауту канала приема (Upstream Time-Out interrupt)

IRQ_STAT.UTO – это флаг тайм-аута канала приема, который может быть поднят аппаратно, установлен или сброшен программным путем записи логической 1 в биты FLAG_CTRL.SET_UTO и FLAG_CTRL.CLR_UTO соответственно. Программный сброс флага имеет больший приоритет по отношению к программной или аппаратной установке.

Сброс бита IRQ_EN.UTO приводит к запрету на формирование прерывания при установленном флаге тайм-аута канала приема (IRQ_STAT.UTO).

17.5 Конфигурация входов/выходов

Структурная схема управления входами и выходами контроллера приведена на рисунке 17.15.

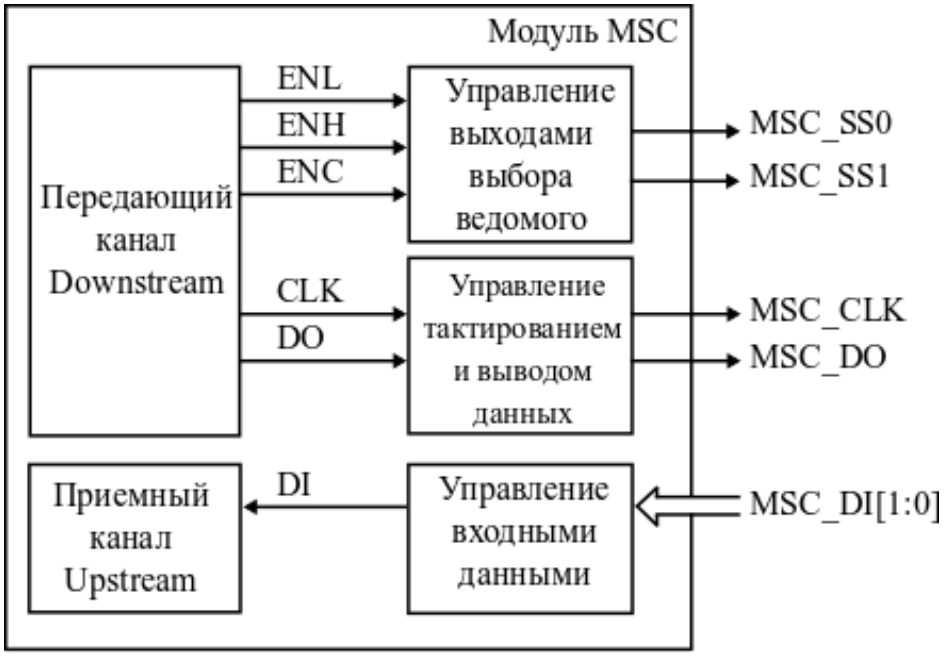


Рисунок 17.15 – Схема управления входами и выходами контроллера

Как показано на рисунках 17.3 и 17.4, активные фазы во время работы передающего канала обозначаются тремя разрешающими сигналами:

– ENL указывает активную фазу SRL кадра данных;

– ENH указывает на активную фазу SRH кадра данных;

– ENC указывает активную фазу кадра команды.

Реализована гибкая схема коммутации выходов выбора ведомого MSC, которая позволяет направлять каждый из трех сигналов разрешения через 2 битный селектор на один из двух выходов MSC_SS* . Это также позволяет подключить более одного внутреннего сигнала разрешения (ENL, ENH, ENC) к одному выходу MSC_SSn. Три битовых поля в регистре IO_CTRL (LSI, HSI и CSI) определяют, какой из выходов выбора ведомого становится активным при наличии установленного внутреннего сигнала разрешения.

В MSC активным уровнем сигнала разрешения является уровень логической 1. Если это требуется в конкретном приложении, можно инвертировать активный уровень всех выходов выбора ведомого MSC_SSn, установив бит ISS в регистре IO_CTRL.

На рисунке 17.16 приведена описанная выше схема коммутация выходов выбора ведомого.

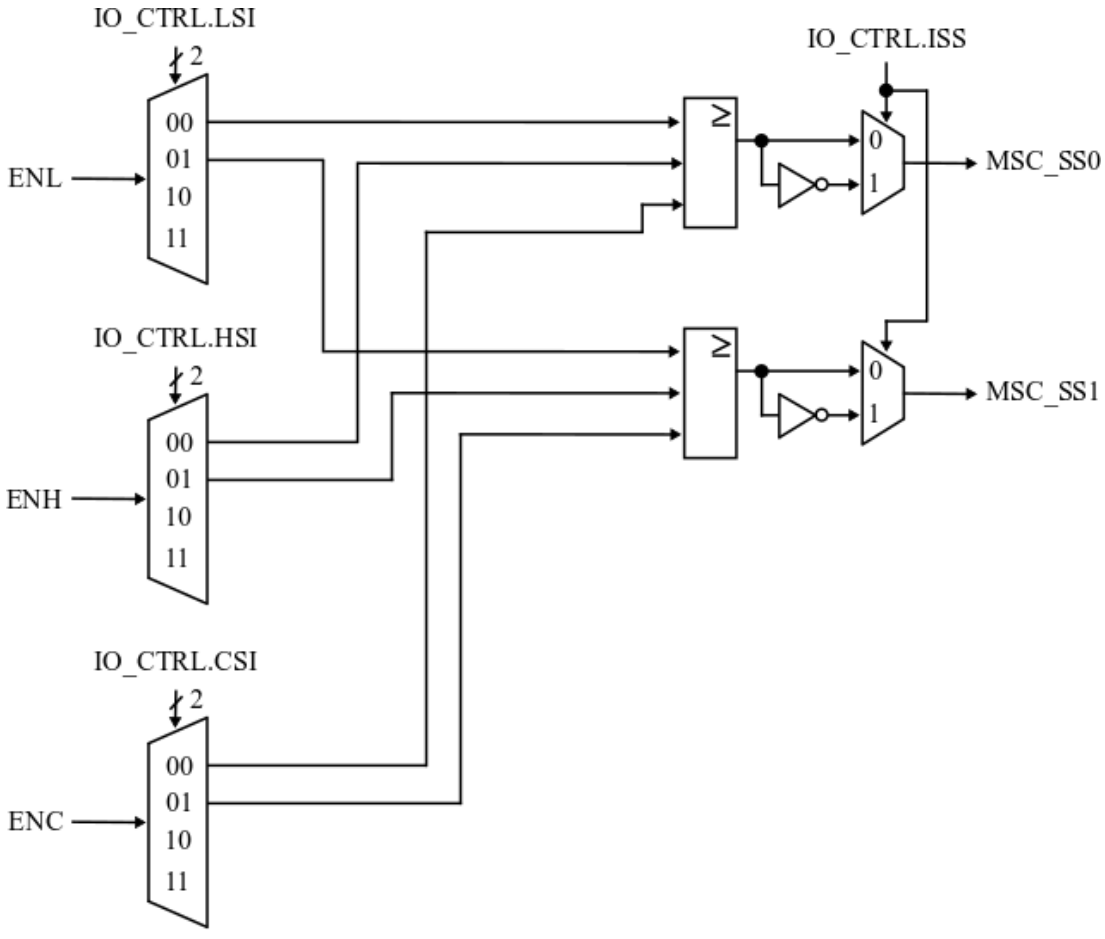


Рисунок 17.16 – Схема коммутации выходов выбора ведомого

По умолчанию активным уровнем выходов MSC_CLK и MSC_DO является уровень логической 1. Установленные биты IO_CTRL.IC, IO_CTRL.IDTS позволяют инвертировать активный уровень данных выходов.

При установленном бите IO_CTRL.CAT тактовый сигнал MSC_CLK будет генерироваться всегда, вне зависимости от того, ведется ли передача в данный момент или

нет. В противном случае MSC_CLK становится активным только во время активных фаз кадров данных или команд (не во время пассивной фазы или кадров пассивного времени). Схема управления выходом тактового сигнала приведена на рисунке 17.17.

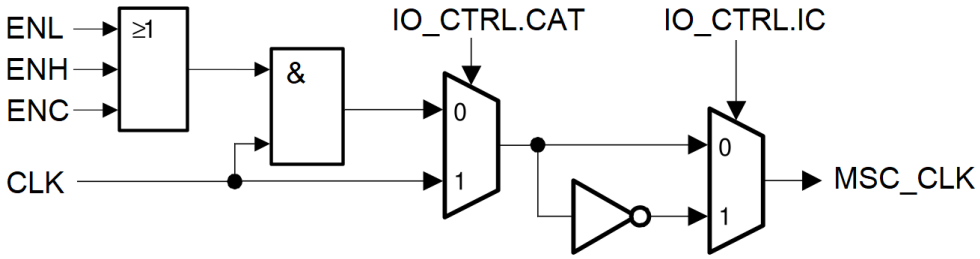


Рисунок 17.17 – Схема управления выходом тактового сигнала

Вход канала приема MSC может быть подключен к одному из двух последовательных входов MSC_DI[1:0] . Битовое поле IO_CTRL.UI выбирает одну из этих входных линий (входной сигнал MSC_DI[n]). Если бит IO_CTRL.IU сброшен, то MSC_DI[n] напрямую подключен к последовательному входу буфера приема DI. Если IO_CTRL.IU установлен, то MSC_DI[n] подключается к входу DI через инвертор.

18 Блоки ШИМ (PWM)

Архитектура блоков ШИМ разработана по принципу минимальной нагрузки на процессор, что достигается автоматизацией формирования выходных импульсов с настраиваемыми пользователем параметрами. Так, после минимальных настроек, эти блоки способны работать самостоятельно, формируя выходные сигналы на выводах PWMx_A и PWMx_B (где x от 0 до 15) микроконтроллера.

Микроконтроллер содержит 16 блоков ШИМ, объединенных схемой синхронизации. Все блоки ШИМ0 – ШИМ15 являются блоками ШИМ высокого разрешения.

Управляющие регистры блоков ШИМ приведены в приложении А.12.

Блоки ШИМ тактируются частотой PCLK, которая соответствует системной частоте SYSCLK.

Каждый блок ШИМ поддерживает следующую функциональность:

- 16-разрядный таймер;
- выходы PWMx_A и PWMx_B могут работать в режиме фронтовой и центрированной модуляции как полностью независимо, так и комплементарно с разделением генератором «мертвого» времени;
- выходы PWMx_A и PWMx_B могут управляться в зависимости от событий цифровых компараторов блока АЦП, обеспечивая автоматический релейный режим поддержания заданной величины;
- программное управление выходами ШИМ;
- программное задание фазы счетчиков таймера для координации работы нескольких блоков ШИМ;
- аппаратный контроль фазы при координации работы нескольких блоков ШИМ;
- предотвращение наложения фронтов за счет генератора «мертвого» времени с независимой схемой задержки переднего и заднего фронтов выходного сигнала;
- сигнал аварии может переводить выходы PWMx_A и PWMx_B в высокое, низкое или Z-состояние;
- однократная и циклическая обработка сигналов аварии;
- все события могут инициировать прерывания;
- при обработке прерываний программируемый предделитель событий позволяет снизить нагрузку на процессор;
- сигнал ШИМ может модулироваться высокочастотным сигналом при использовании драйверов ключей с импульсным трансформатором.

Для начала работы с блоками ШИМ необходимо снять с них сброс и разрешить тактирование – необходимо установить биты PWMxEN регистров PCLKCFG и PRSTCFG.

Описание сигналов и выводов блока ШИМ:

- PWMx_A и PWMx_B (где x от 0 до 15) – выходы ШИМ;
- PWM_TZ0 – PWM_TZ5 – входы, с которых принимаются сигналы аварии (общие для всех блоков ШИМ, и каждый блок может использовать или не использовать эти сигналы);
- PWM_SYNC – вход микроконтроллера, служащий для приема синхросигнала;
- SYNCI – вход блока ШИМ, служащий для приема синхросигнала;
- SYNCO – выход блока ШИМ, использующийся для синхронизации блоков;
- PWM_INT – прерывание обработчика событий;
- PWM_TZINT – прерывание по сигналу аварии;
- PWM_HDINT – прерывание от порогового выключателя.

Функциональная схема блока ШИМ показана на рисунке 18.1, где заштрихованные области соответствуют функциям высокого разрешения.

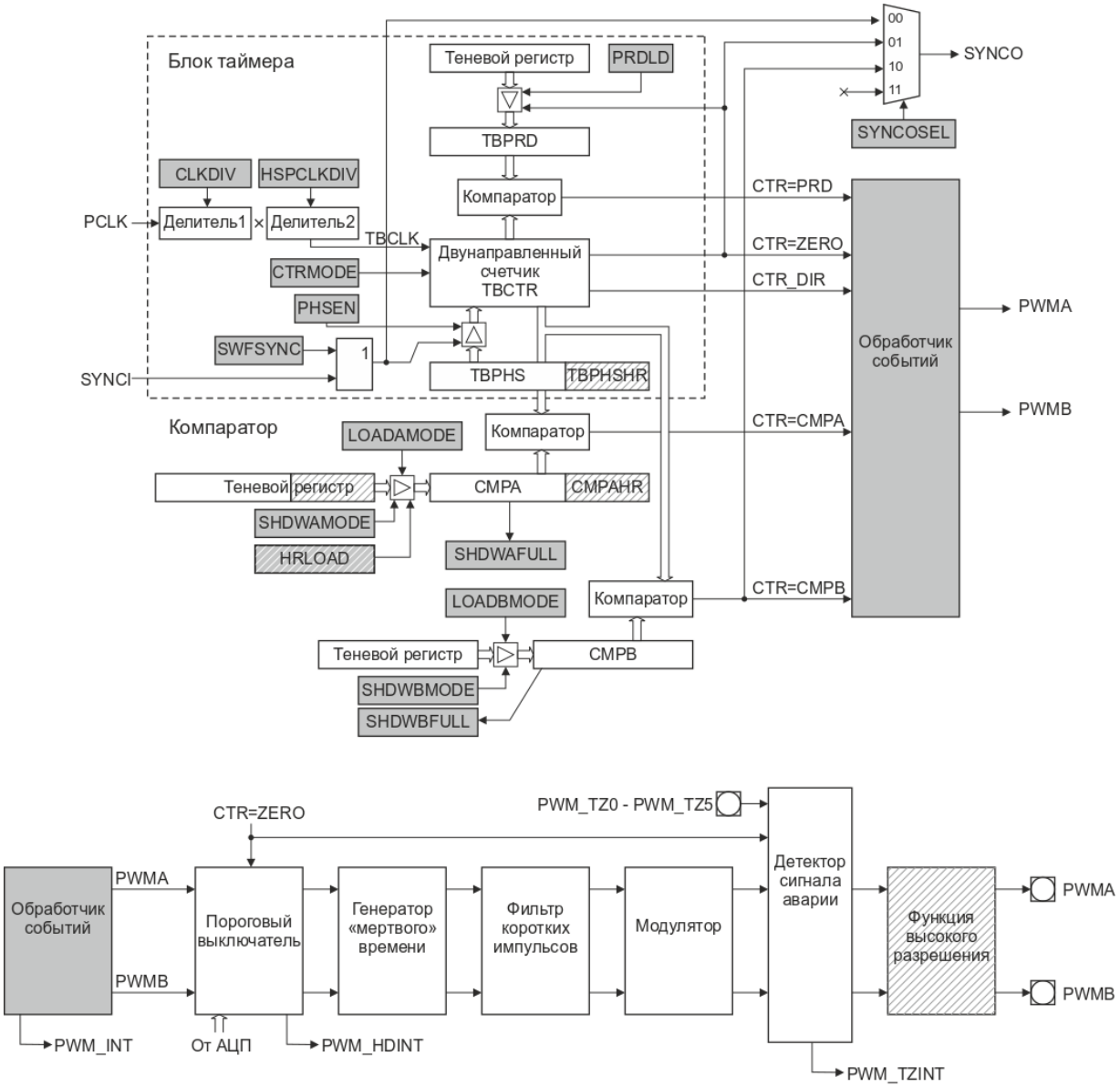


Рисунок 18.1 – Функциональная схема блока ШИМ

18.1 Таймер

Таймер представляет собой двунаправленный счетчик TBCTR, тактируемый сигналом TBCLK, который формируется на основе синхросигнала PCLK. Частота сигнала TBCLK задается произведением коэффициентов двух делителей. Коэффициенты задаются полями CLKDIV и HPCLKDIV регистра TBCTL. По умолчанию, работа делителей остановлена - TBCLK не генерируется. Чтобы таймер начал счет по TBCLK, необходимо записью единиц в биты PWMxEN регистра PWMEN блока SIU разрешить счет блока.

Для работы других блоков ШИМ счетчик позволяет формировать события такие, как совпадение по периоду CTR = PRD (TBCTR = TBPRD), совпадение с нулем CTR = Zero (TBCTR = 0000h), совпадение с регистрами CTR = CMPA и CTR = CMPB (TBCTR = CMPA и TBCTR = CMPB соответственно). Событие TBCTR = FFFFh влияет только на флаг CTRMAX регистра TBSTS.

Всеми настройками работы счетчика таймера управляет регистр TBCTL.

Состояние счетчика отражают флаги регистра TBSTS.

На выходе первого блока ШИМ формируется сигнал SYNCO, который является, синхросигналом для остальных блоков ШИМ, см. рисунок 18.3а. Сигнал SYNCO имеет три источника – программно сгенерированный синхроимпульс, посредством записи единицы в

бит SWFSYNC, события CTR = Zero и CTR = CMPB. Выбор источника осуществляется посредством поля SYNCOSEL.

В блоке таймера находятся регистры начальной фазы счета TBPHS и периода (максимального значения счетчика) TBPRD. Регистр периода имеет теневой регистр для синхронной загрузки значения, до которого счетчик осуществляет счет. Управление загрузкой осуществляется битом PRDLD.

Счетчик может работать в трех режимах счета, см. рисунок 18.2:

- вверх (от 0000h до значения TBPRD, затем сброс в 0000h и т. д.);
- вниз (от значения TBPRD до 0000h, затем загрузка значения TBPRD и т. д.);
- вверх-вниз (от 0000h до значения TBPRD, затем от значения TBPRD до 0000h и т. д.).

Параметры счета задаются полем CTRMODE.

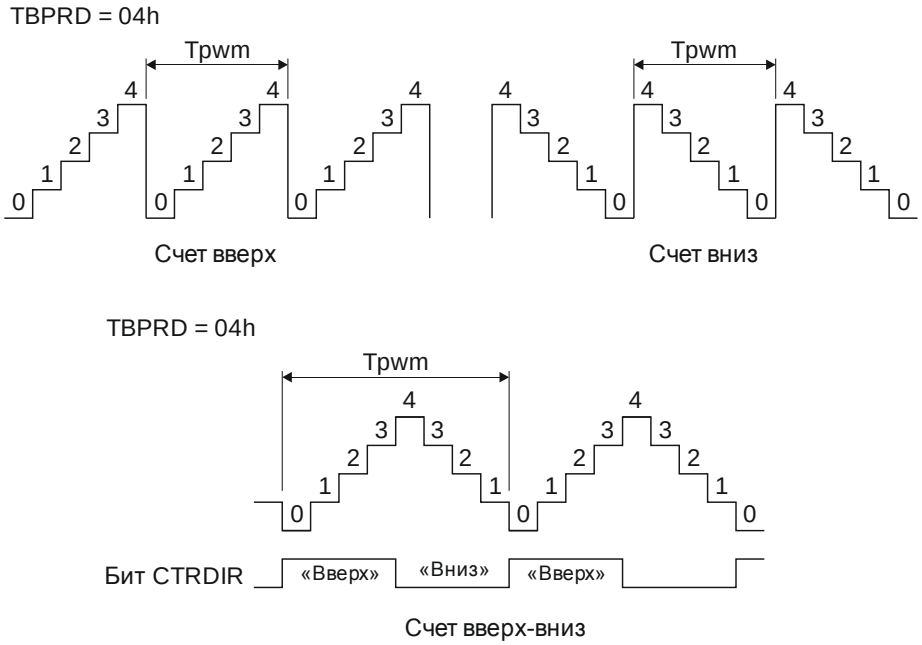


Рисунок 18.2 – Режимы работы счетчика при значении периода 0004h (Tpwm); для режима счета «вверх-вниз» дополнительно указано поведение флага CTRDIR

Теневая загрузка

Регистры TBPRD, CMPA, CMPAHR, CMPB имеют соответствующие теневые регистры и управляющие биты, регулирующие режим и событие загрузки. Помимо индивидуальной настройки теневого режима есть бит SHDWGLOB регистра TBCTL. По умолчанию он установлен, и теневая загрузка работает согласно настройкам. Но возможны ситуации, когда необходимо обеспечить одновременную загрузку всех теневых регистров. Тогда перед загрузкой необходимо сбросить SHDWGLOB, в результате значения будут попадать в теневые регистры, но перезапись в активные будет блокирована. Когда все необходимые регистры будут записаны, следует установить этот бит, и тогда все активные регистры будут перезаписаны в соответствии с заданными настройками по соответствующим событиям. Данный бит не оказывает влияния, если для регистров выбрана прямая загрузка, без участия теневых.

Синхронизация таймеров блоков ШИМ

Реализована схема синхронизации блоков ШИМ, см. рисунок 18.3а.

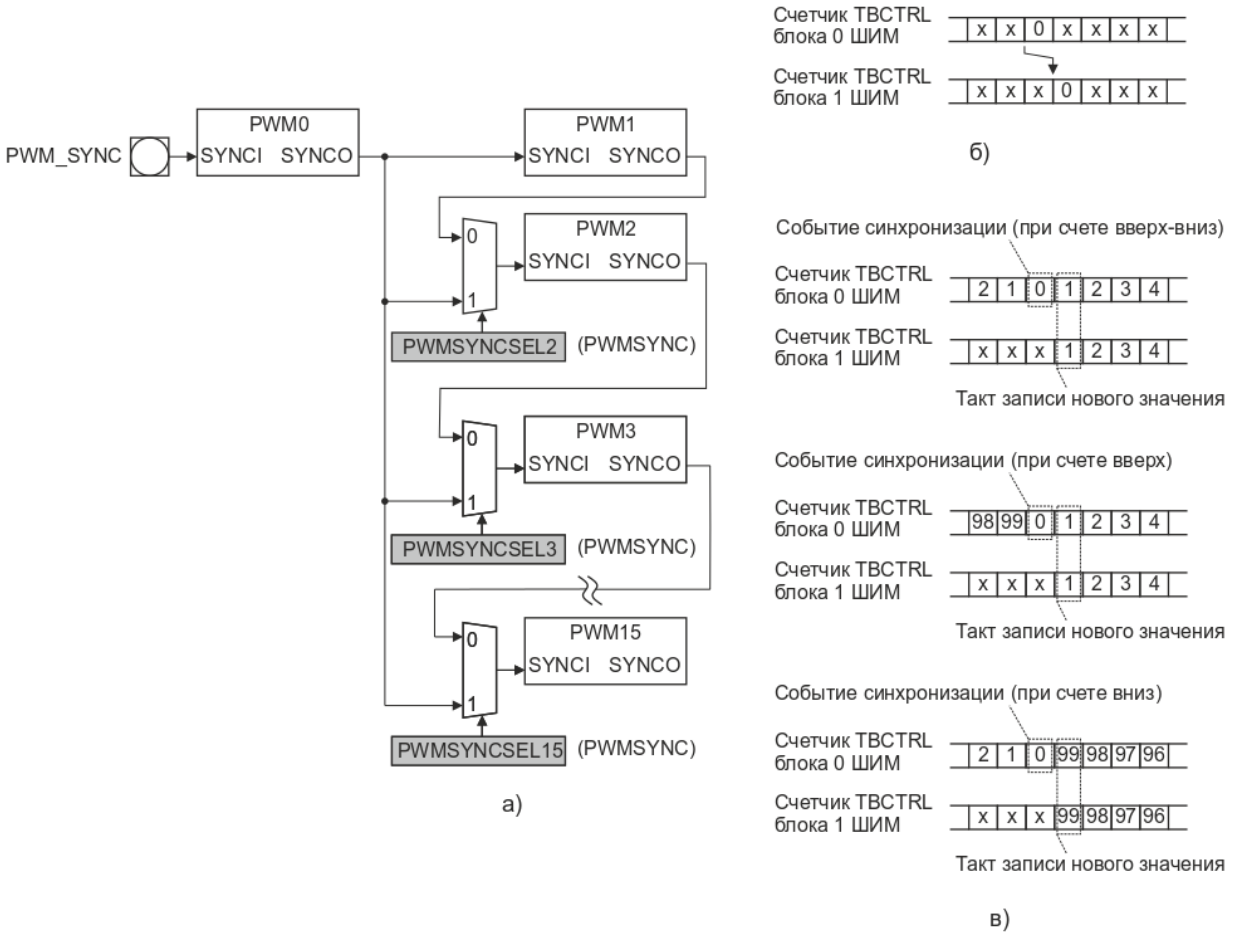


Рисунок 18.3 – Схема синхронизации блоков ШИМ

Система синхронизации таймеров включает в себя таймеры всех блоков ШИМ. Каждый блок ШИМ имеет вход синхронизации SYNCI и выход синхронизации SYNCO.

Если бит PHSEN установлен, то в счетчик таймера будет автоматически загружаться значение регистра TBPHS, при выполнении каждого из условий:

- изменение входного сигнала SYNCI (в этом случае загрузка значения TBPHS в регистр TBCTR происходит на следующий такт TBCLK после поступления импульса на вход SYNCI с задержкой в два системных такта, если TBCLK = PCLK, или один такт, если TBCLK ≠ PCLK);

- запись единицы в бит SWFSYNC (программная синхронизация), которая генерирует импульс синхронизации, аналогичный импульсу с входа SYNCI.

В режиме счета вверх-вниз необходимо запрограммировать бит PHSDIR, чтобы задать направление счета таймера после синхронизации.

Если бит PHSEN сброшен, блок ШИМ не будет реагировать на входной сигнал синхронизации, а только передавать напрямую этот сигнал на выход SYNCO, чтобы тактировать другие блоки ШИМ. Следующая особенность схемы: генерация и распространение сигнала синхронизации от блока ШИМ – занимает один такт TBCLK.

К примеру, если по событию синхронизации блока 0 в счетчик блока 1 должен быть записан ноль, то этот ноль запишется только на следующий такт после события, см. рисунок 18.3б. Таким образом, при синхронизации от другого блока ШИМ нужно всегда учитывать этот такт и записывать значение фазы, следующее по порядку, в соответствии с режимом счета, см. рисунок 18.3в.

18.2 Компаратор

Компаратор – это блок, сравнивающий значение счетчика таймера с заданными значениями порогов срабатывания. Значения хранятся в регистрах CMPA и CMPB. Значения, записываемые по адресам регистров CMPA и CMPB, предварительно размещаются в теневых регистрах. Это нужно для синхронной загрузки новых значений. Управление загрузкой регистров CMPA и CMPB осуществляется битами SHDWAMODE и SHDWBMODE, а также полями LOADAMODE и LOADBMODE регистра CMPCTL.

Блок компаратора формирует на выходах два события CTR = CMPA и CTR = CMPB, возникающие в случае совпадения значения счетчика с регистром CMPA и/или регистром CMPB соответственно.

Для каждого компаратора событие может возникать:

- один раз за период, если счетчик считает вверх или вниз;
- один раз за период, если счетчик считает вверх-вниз, но при этом значение в регистре CMPA/CMPB равно 0000h или значению TBPRD;
- два раза за период, если счетчик считает вверх-вниз и при этом значение в регистре CMPA/CMPB лежит в диапазоне 0001h – (TBPRD – 1).

На рисунках 18.4 – 18.7 приведены примеры формирования сигналов событий.

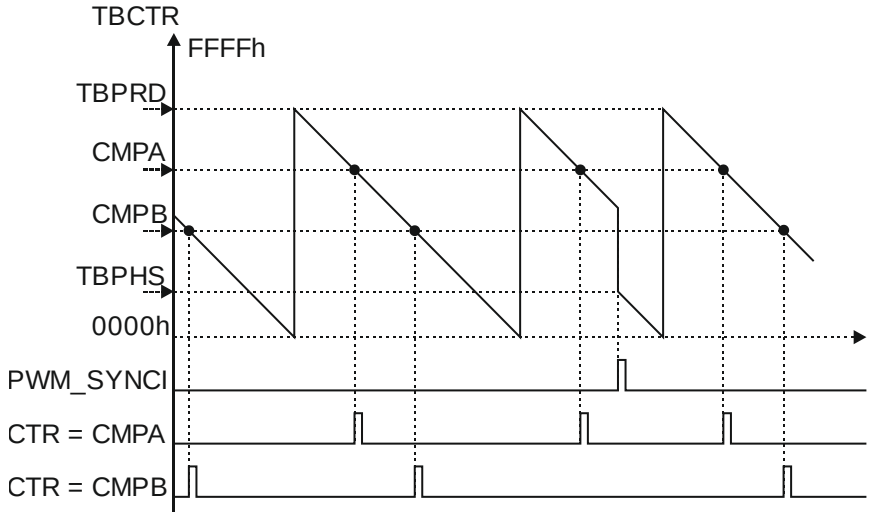


Рисунок 18.4 – Пример формирования сигналов событий при счете вниз

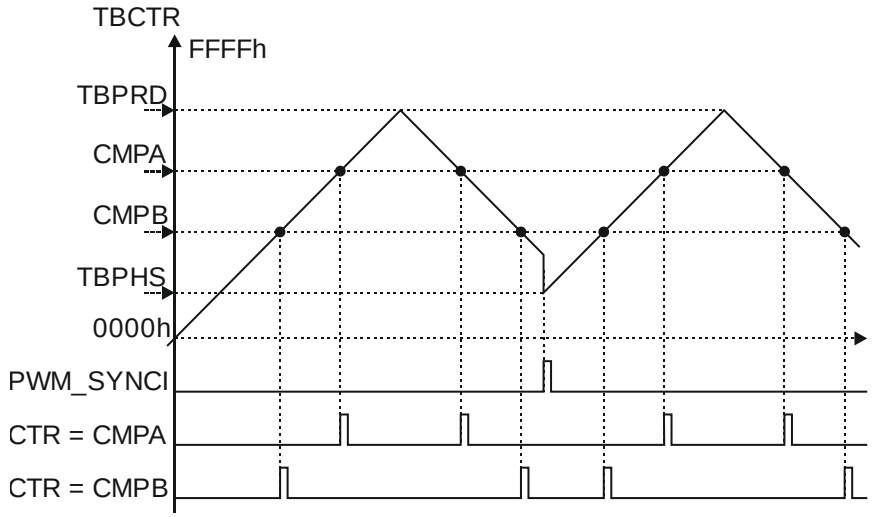


Рисунок 18.5 – Пример формирования сигналов событий при счете вверх-вниз.
Синхронизация при счете вверх

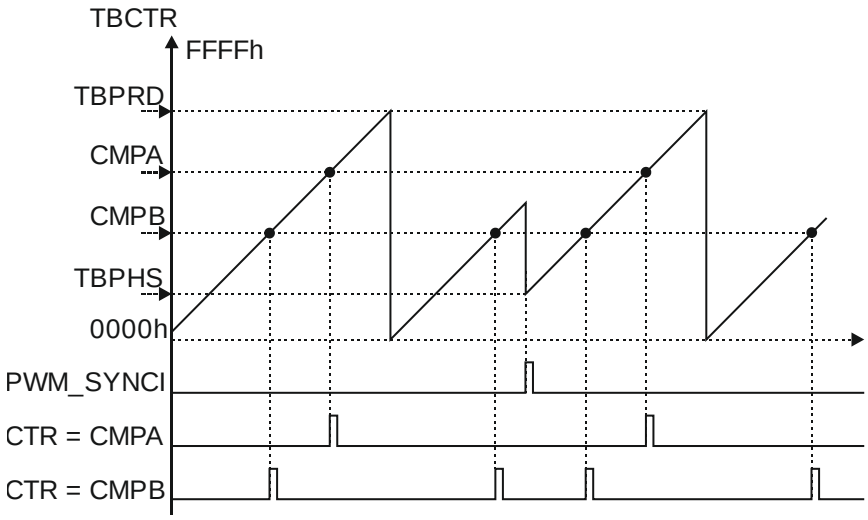


Рисунок 18.6 – Пример формирования сигналов событий при счете вверх

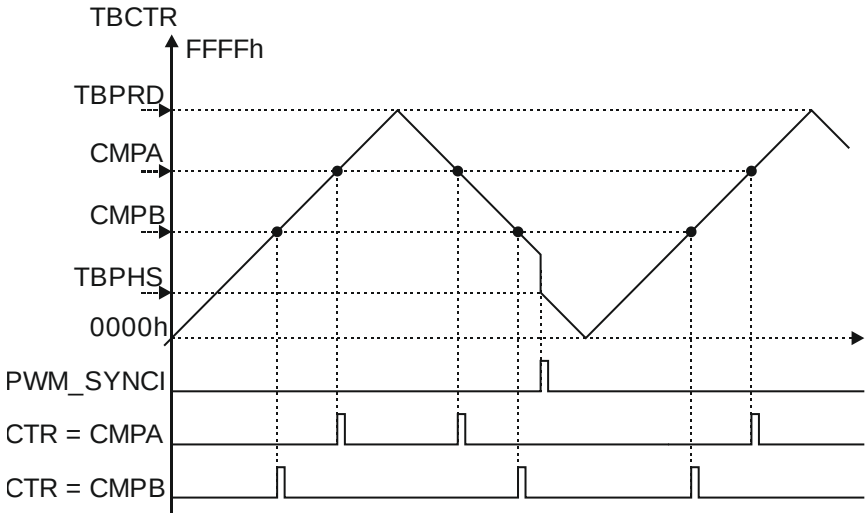


Рисунок 18.7 – Пример формирования сигналов событий при счете вверх-вниз. Синхронизация при счете вниз

18.3 Обработчик событий

Обработчик событий – блок, управляющий поведением сигналов на линиях PWMA и PWMB, см. рисунок 18.1, в зависимости от возникающих событий на входе блока и направления счета счетчика таймера. На поведение выходных сигналов влияют импульсы входных сигналов при возникновении событий: CTR = PRD, CTR = Zero, CTR = CMPA, CTR = CMPB.

Основные действия с сигналами PWMA и PWMB:

- переключение в единицу или ноль;
- инверсия (переключение в противоположное состояние);
- сохранение без изменений.

Поведение сигналов задается независимо друг от друга. Кроме этого, обработчик событий позволяет программно задавать состояние сигналов PWMA и PWMB и величину «мертвого» времени ШИМ. Управление работой блока производится посредством регистров AQCTLA, AQCTLB, AQSFRC, AQCSFRC.

Существует вероятность того, что несколько событий могут произойти одновременно. Для таких ситуаций обработчик событий использует систему приоритетов событий.

Таблица 18.1 – Распределение приоритетов событий при счете вверх

Событие	Приоритет
Программное	1 (самый высокий)
CTR = TBPRD	2
CTR = CMPB (счет вверх) при счете вверх	3
CTR = CMPA (счет вверх) при счете вверх	4 (самый низкий)

Таблица 18.2 – Распределение приоритетов событий при счете вниз

Событие	Приоритет
Программное	1 (самый высокий)
CTR = Zero	2
CTR = CMPB (счет вниз) при счете вниз	3
CTR = CMPA (счет вниз) при счете вниз	4 (самый низкий)

Таблица 18.3 – Распределение приоритетов событий при счете вверх-вниз

Событие	Приоритет
Программное	1 (самый высокий)
CTR = CMPB (счет вверх) при счете вверх или CTR = CMPB (счет вниз) при счете вниз	2
CTR = CMPA (счет вверх) при счете вверх или CTR = CMPA (счет вниз) при счете вниз	3
CTR = Zero или CTL = PRD	4
CTR = CMPB (счет вверх) при счете вниз или CTR = CMPB (счет вниз) при счете вверх	5
CTR = CMPA (счет вверх) при счете вниз или CTR = CMPA (счет вниз) при счете вверх	6 (самый низкий)

В режиме счета вверх:

- если компаратор запрограммирован так, что CMPA/CMPB ≤ TBPRD (счет вверх), то событие произойдет при CTR = CMPA/CMPB;
- если компаратор запрограммирован так, что CMPA/CMPB > TBPRD (счет вверх), то событие не произойдет;
- если компаратор запрограммирован на срабатывание при счете вниз, то событие не произойдет.

В режиме счета вниз:

- если компаратор запрограммирован так, что CMPA/CMPB ≤ TBPRD (счет вниз), то событие произойдет при CTR = CMPA/CMPB;
- если компаратор запрограммирован так, что CMPA/CMPB ≥ TBPRD (счет вниз), то событие произойдет при CTR = TBPRD;
- если компаратор запрограммирован на срабатывание при счете вверх, то событие не произойдет.

В режиме счета вверх-вниз, см. рисунок 18.8:

- если счетчик считает вверх, а компаратор запрограммирован так, что CMPA/CMPB < TBPRD (счет вверх), то событие произойдет при CTR = CMPA/CMPB;
- если CMPA/CMPB ≥ TBPRD (счет вверх), то событие произойдет при CTR = TBPRD;
- если счетчик считает вниз, а компаратор запрограммирован так, что CMPA/CMPB < TBPRD (счет вниз), то событие произойдет при CTR = CMPA/CMPB;
- если CMPA/CMPB ≥ TBPRD (счет вверх), то событие произойдет при CTR = TBPRD.

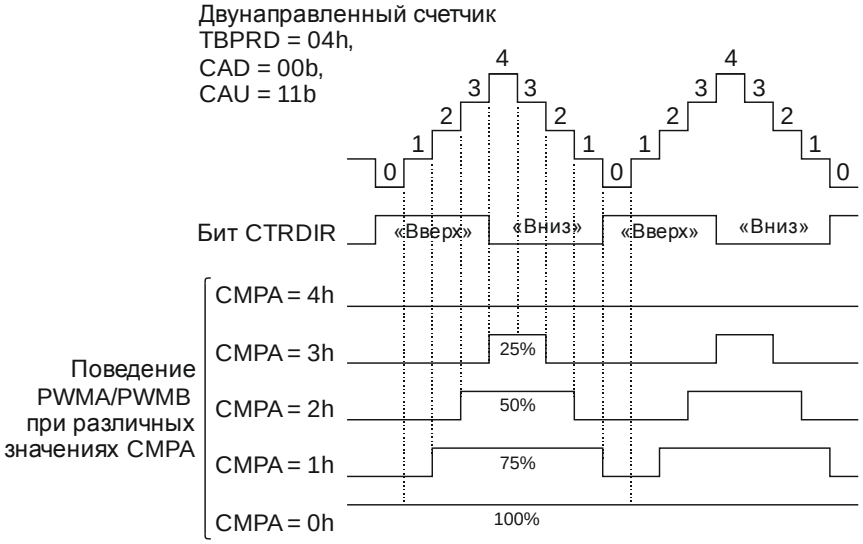


Рисунок 18.8 – Работа таймера при счете вверх-вниз с симметричным выходом (центрированная модуляция)

На рисунках 18.9 – 18.11 показано поведение линий PWMA и PWMB при различных видах модуляции, принятые обозначения и пояснения приведены в таблице 18.4.

Таблица 18.4 – Пояснения к обозначениям на рисунках 18.9 – 18.11

<table>
<tr><td colspan="3">Обозначение</td><td colspan="2">Пояснение</td></tr>
<tr><td>P
×</td><td>CA
×</td><td>CB
×</td><td colspan="2">События CTR = PRD, CTR = CMPA, CTR = CTRB, соответственно. Символ «×» указывает на то, что при возникновении этого события сигнал на линии PWMA/PWMB остается без изменений. Пунктирными линиями отмечены моменты возникновения события. Так, например, см. рисунок 18.9, при возникновении события CTR = CTRB, сигнал на линии PWMA остается без изменения, а сигнал на линии PWMB переключается в ноль</td></tr>
<tr><td colspan="2">Z
↑</td><td>Z
↓</td><td>Событие CTR = Zero</td><td rowspan="3">Символы «↑»/«↓» указывают на то, что при возникновении этого события сигнал на линии PWMA/PWMB переключается в единицу/ноль</td></tr>
<tr><td colspan="2">CA
↑</td><td>CA
↓</td><td>Событие CTR = CMPA</td></tr>
<tr><td colspan="2">CB
↑</td><td>CB
↓</td><td>Событие CTR = CMPB</td></tr>
</table>

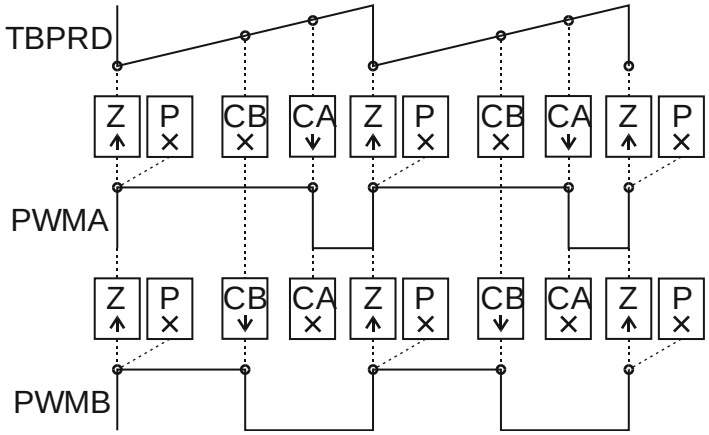


Рисунок 18.9 – Независимый режим работы выходов (фронтовая модуляция)

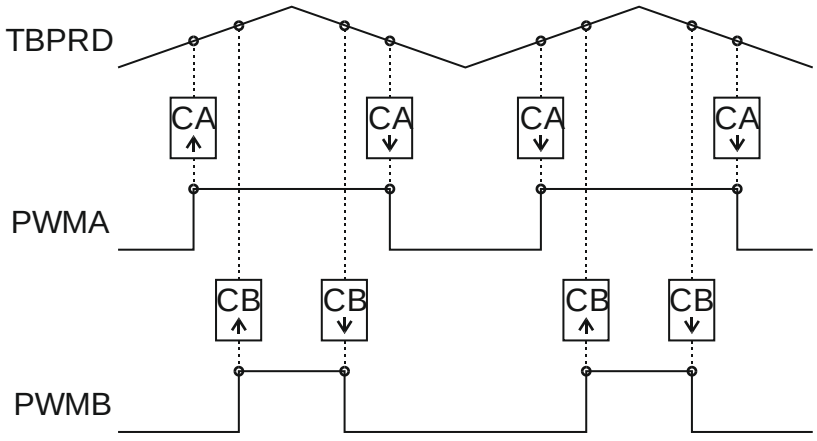


Рисунок 18.10 – Симметричный режим работы при счете вверх-вниз (центрированная модуляция)

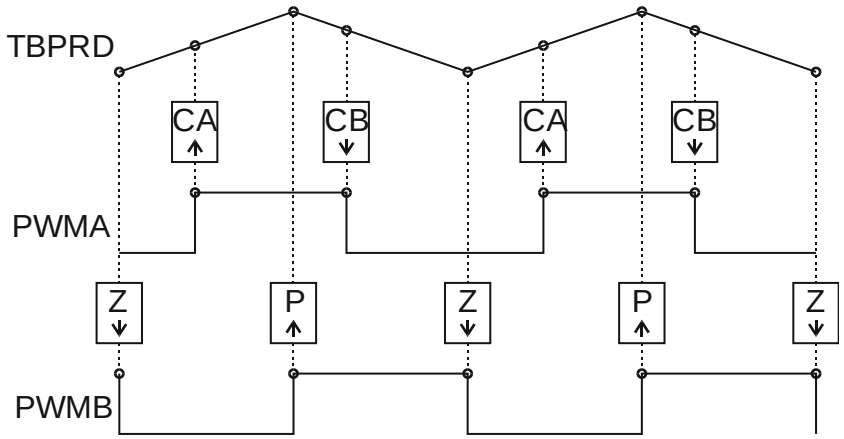


Рисунок 18.11 – Ассиметричный режим работы при счете вверх-вниз

18.4 Пороговый выключатель

Пороговый выключатель контролирует выходные сигналы PWMA и PWMB обработчика событий и позволяет удерживать их в определенном заданном пользователем состоянии в случае прихода сигнала триггера от цифровых компараторов блока АЦП. Этот блок удобен для организации релейного поддержания заданного уровня какой-либо физической величины, например для организации контура тока. В этом случае цифровой компаратор, к каналу которого подключен сигнал датчика тока контура, формирует сигнал о превышении током задания, а соответствующий пороговый выключатель реагирует на это превышение и включает/отключает соответствующий силовой транзистор посредством влияния на выход ШИМ.

Функциональные возможности:

- входные события от компараторов блока АЦП могут использоваться всеми блоками ШИМ;
- при регистрации события от компаратора выходные сигналы обработчика событий могут быть переведены в состояние логической единицы, нуля или оставлены без изменений;
- поддерживаются однократное и циклическое срабатывания для удержания выхода;
- входное событие от компаратора может анализироваться в однократном и циклическом режимах;
- событие срабатывания компаратора может быть сгенерировано программно;
- пороговый выключатель может быть отключен, если он не требуется;
- может генерироваться прерывание по событиям порогового выключателя.

Управление пороговым выключателем осуществляется посредством регистров HDSEL, HDCTL и HDFRC.

Компараторы блока АЦП

Когда выходные сигналы компараторов блока АЦП переходят в состояние высокого уровня, формируется событие. Каждый пороговый выключатель блока ШИМ может использовать, а может не использовать эти события в своей работе; выбор, по сигналу какого компаратора блока АЦП формировать событие удержания, задается с помощью регистра HDSEL. Длительность импульса на входном сигнале от компаратора блока АЦП не должна быть меньше периода системного синхросигнала. Каждый входной сигнал компаратора блока АЦП должен быть настроен на однократное или циклическое формирование события, выбор режима задается битами CBC и OST регистра HDCTL, а источник события – полем ADCDC регистра HDSEL.

При получении события от компаратора блока АЦП в режиме циклической обработки немедленно формируется реакция на основе содержимого регистра HDCTL, в результате

чего меняется состояние сигналов на выходе порогового выключателя взамен полученных от обработчика событий PWMA и/или PWMB на заданное пользователем в регистре HDCTL. Дополнительно устанавливается флаг CBC в регистре HDFLG, и генерируется прерывание PWM_HDINT. Удержание выходных сигналов PWMA и PWMB заканчивается по событию TBCTR = 0000h, при условии, что событие компаратора блока АЦП уже не активно. Таким образом, в режиме циклической обработки состояние удержания сбрасывается в каждом периоде ШИМ. При этом флаг CBC остается активным до его программного сброса. Если после сброса флага CBC вновь будет получено событие компаратора блока АЦП, то флаг установится вновь.

При получении события компаратора блока АЦП в режиме однократной обработки, также немедленно формируется реакция на основе содержимого регистра HDCTL, которая меняет состояние выходных сигналов PWMA и/или PWMB. В дополнение, устанавливается флаг OST, и генерируется прерывание PWM_HDINT. Удержание выходных сигналов будет производиться до программного сброса записью единицы в бит OST регистра HDCLR.

Способ удержания выходных сигналов при получении события компаратора блока АЦП программируется индивидуально для выходных сигналов PWMA и PWMB в регистр HDCTL.

18.5 Генератор задержки ШИМ

Блок имеет на входе сигналы PWMA и PWMB с выходов обработчика событий, а на выходах повторяет эти сигналы, но со вставкой задержки («мертвое» время) в момент переключения сигналов (если это необходимо).

Задержку можно учесть при программировании обработчика событий, но, чтобы с высокой вероятностью избежать ошибок, желательно использовать генератор задержки ШИМ.

Основные функции генератора задержки ШИМ:

- генерация пары выходных сигналов PWMA и PWMB с выдержкой интервалов (задержек) времени относительно входных сигналов PWMA, PWMB;
- программирование задержки для активного высокого и активного низкого уровня сигналов каналов PWMA и PWMB;
- добавление программируемой задержки передних фронтов сигналов;
- добавление программируемой задержки для задних фронтов сигналов;
- возможность передачи сигналов с входов на выходы без изменений.

Генератор задержки ШИМ программируется посредством регистров DBCTL, DBRED и DBFED. Величина задержки в регистрах DBRED и DBFED задается в тактах частоты PCLK. Структурная схема генератора «мертвого» времени ШИМ представлена на рисунке 18.12.

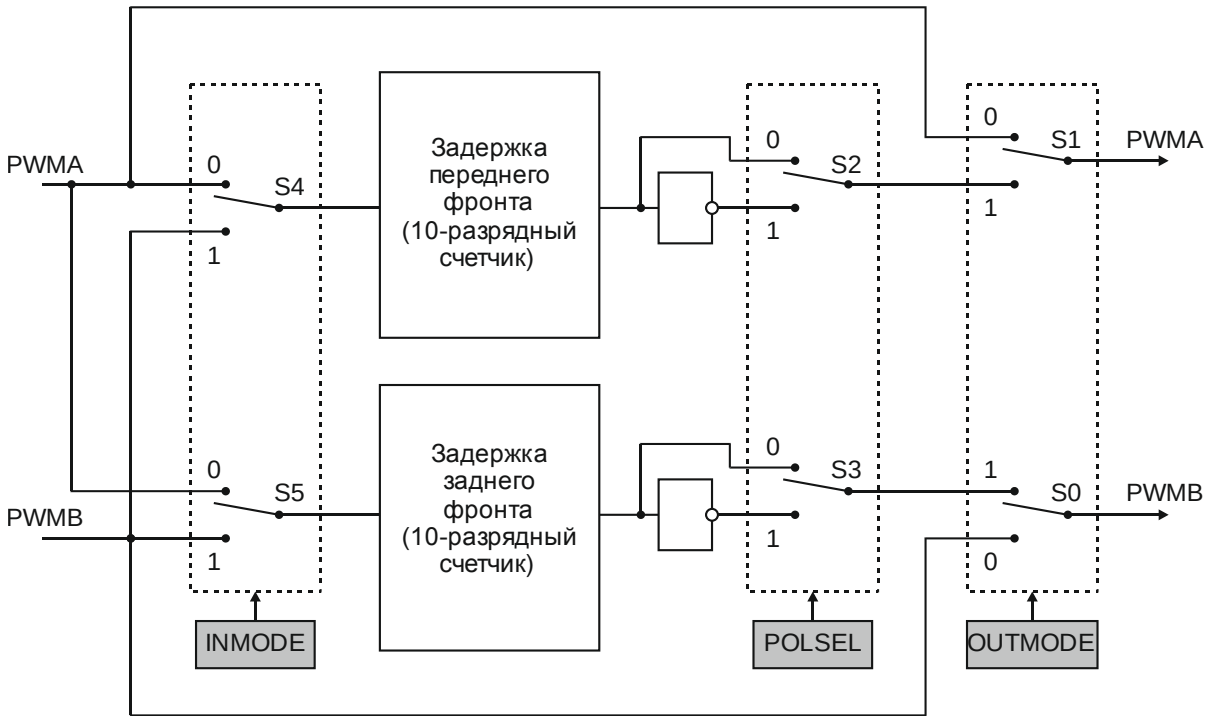


Рисунок 18.12 – Структурная схема генератора «мертвого» времени ШИМ

Функционирование

Генератор задержки ШИМ может работать с четырьмя источниками (фронты сигналов PWMA и PWMB). Выбор источника задается полем MODE регистра DBCTL.

Поле POLSEL позволяет задать инверсию (переключение значения на противоположное) сигнала после внесения задержки, см. рисунок 18.13.

Величины задержек по переднему и заднему фронту программируются раздельно посредством регистров DBRED и DBFED соответственно.

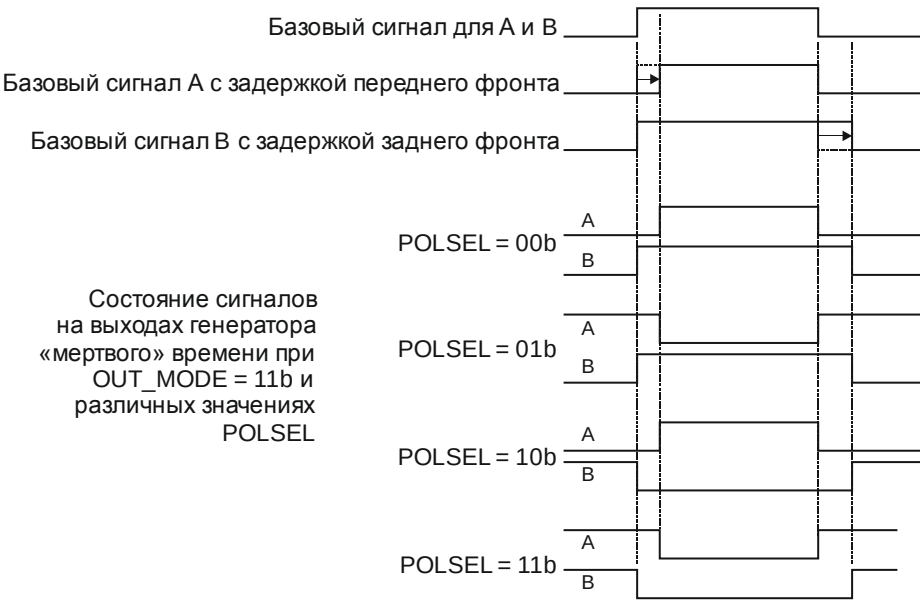


Рисунок 18.13 – Временные диаграммы работы генератора «мертвого» времени в типовой конфигурации

18.6 Фильтр коротких импульсов

Фильтр коротких импульсов предназначен для блокирования прохождения на выход импульсов с длительностью меньше заданной. Этот блок может применяться, если драйвер силового ключа инвертора не имеет такой функции, а для обеспечения правильного режима работы транзистора необходимо запретить открытие/закрытие транзистора на очень короткое время.

Основные функции фильтра:

- программируемая ширина минимального пропускаемого импульса;
- фильтр может быть отключен, если он не требуется.

Ширина минимального импульса, допускаемого к прохождению на выход, задается в регистре FWDTH в тактах PCLK и может принимать значение от 00h (фильтр выключен) до FFh. Импульсы, длительностью меньше заданной, пропускаться не будут.

18.7 Модулятор

Блок позволяет модулировать выходной сигнал ШИМ с помощью высокочастотных импульсов программируемой скважности. Модулирование требуется для управления силовыми ключами через импульсный трансформатор.

Основные функции модулятора:

- программируемая частота;
- программируемая ширина первого импульса;
- программируемый коэффициент заполнения второго и последующего импульсов;
- модулятор может быть отключен (бит CHPEN регистра PCCTL).

Модулятор программируется посредством регистра PCCTL. Структурная схема модулятора приведена на рисунке 18.14.

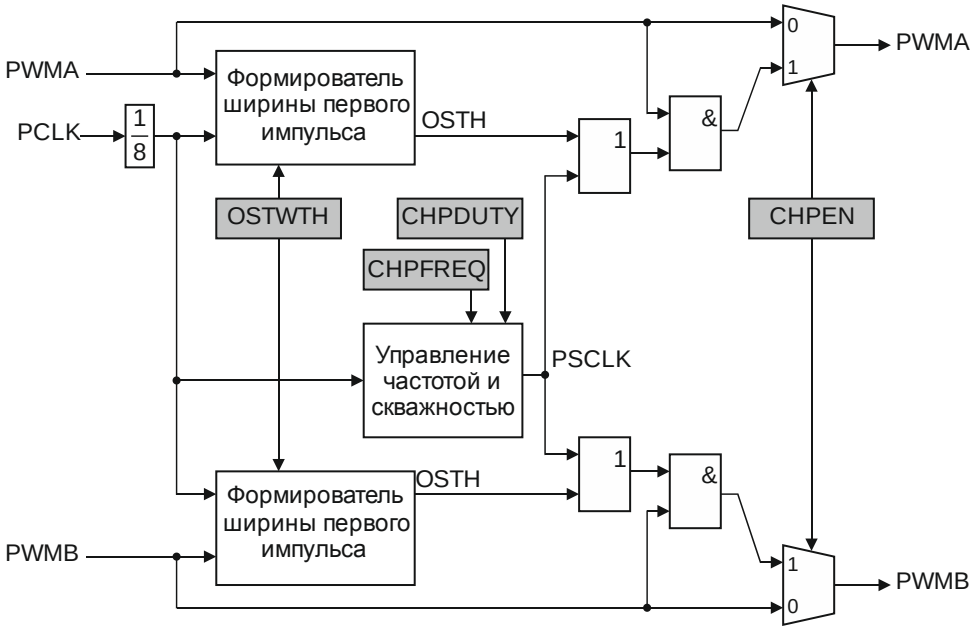


Рисунок 18.14 – Структурная схема модулятора

Функционирование

Частота модуляции формируется на основе системной частоты при помощи делителя, программируемого полем CHPFREQ.

На рисунке 18.15 приведен пример временных диаграмм работы модулятора.

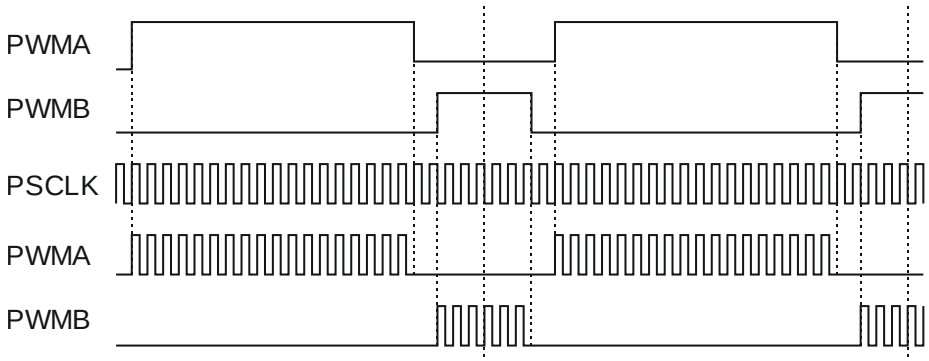


Рисунок 18.15 – Временные диаграммы работы модулятора

Ширина первого импульса может программироваться независимо с помощью поля OSTWTH, это требуется для открывания ключа. Значения поля OSTWTH лежат в диапазоне 0h – Fh.

Ширина L первого импульса определяется по формуле:

$$L = T \times 8 \times (OSTWTH + 1), \quad (18.1)$$

где T – период синхросигнала PCLK.

На рисунке 18.16 приведен пример формирования расширенного первого импульса.

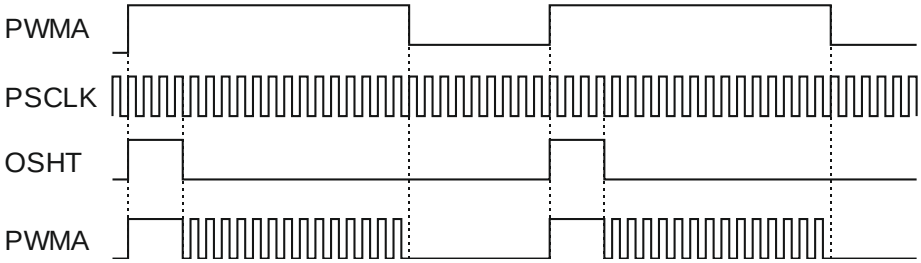


Рисунок 18.16 – Временные диаграммы работы модулятора с раширенным первым импульсом

Также существует возможность регулировать коэффициент заполнения импульсов посредством поля CHPDUTY. Значения поля CHPDUTY лежат в диапазоне 0h – 7h.

Коэффициент заполнения D (с шагом 12,5 %) последующих импульсов определяется по формуле:

$$D = 12{,}5 \times (CHPDUTY + 1). \quad (18.2)$$

На рисунке 18.17 приведена иллюстрация регулировки коэффициента заполнения.

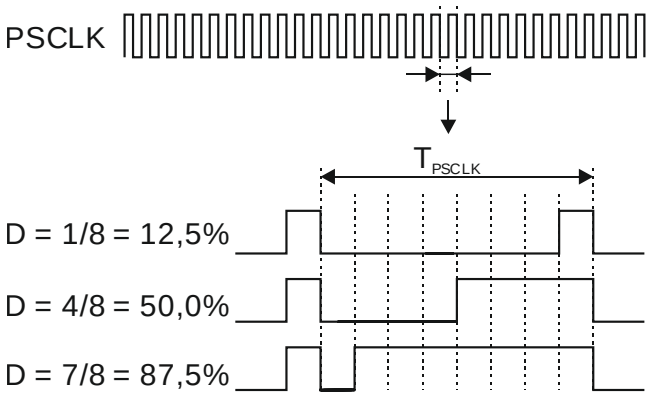


Рисунок 18.17 – Регулировка коэффициента заполнения D

18.8 Детектор сигнала аварии

Детектор сигнала аварии контролирует выходы PWMx_A и PWMx_B и может переводить их в определенное (запрограммированное) состояние в случае, если поступит сигнал аварии.

Основные функции детектора сигнала аварии:

- входные сигналы аварии с выводов микроконтроллера PWM_TZ0 – PWM_TZ5 могут использоваться любым блоком ШИМ;
- в случае если поступит сигнал аварии, выводы ШИМ могут быть переведены в одно из состояний: логического нуля, логической единицы, высокоимпедансное или оставлены без изменения;
- поддерживается однократная блокировка выводов для ситуации короткого замыкания или перегрузки по току;
- поддерживается циклическая блокировка для режима ограничения тока;
- каждый входной источник сигнала аварии может быть обработан в однократном и циклическом режимах;
- поддерживается программная генерация сигнала аварии;
- детектор сигнала аварии может быть отключен, если он не требуется.

Детектор сигнала аварии программируется посредством регистров TZSEL, TZCTL, TZEINT, TZFLG, TZCLR и TZFRC.

Функционирование

Структурная схема детектора сигналов аварии показана на рисунке 18.18.

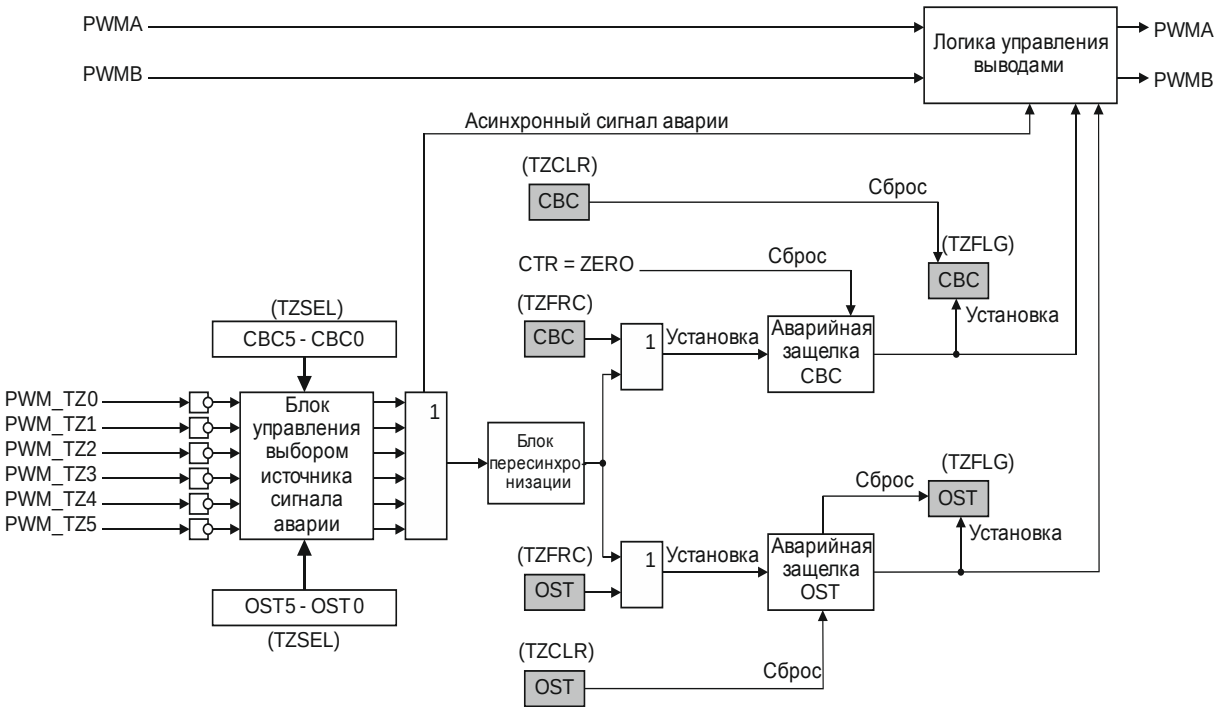


Рисунок 18.18 – Структурная схема детектора сигналов аварии

Переход входного сигнала аварии из состояния логической единицы в состояние логического нуля формирует событие аварии. Каждый блок ШИМ может использовать или не использовать это событие в своей работе (программируется посредством регистра TZSEL). События могут формироваться синхронно (с цифровым фильтром помех) или асинхронно (программируется через регистры GPIO микроконтроллера). При синхронной обработке длительность импульса на входном сигнале сбоя должна быть не меньше периода синхросигнала TBCLK. Если же обработка производится в асинхронном режиме, то событие формируется и обрабатывается даже в том случае, если по какой-либо причине отключилось тактирование. Каждый входной сигнал аварии должен быть настроен на

однократное или циклическое формирование события аварии (программируется посредством регистра TZSEL).

При получении события аварии в режиме циклической обработки немедленно выполняется действие, заданное регистром TZCTL, и устанавливается флаг CBC в регистре TZFLG, а также генерируется прерывание PWM_TZINT, если разрешено в регистре TZEINT и контроллером прерываний. Аварийное удержание выводов заканчивается по событию TBCTR = 0000h при условии, что событие аварии уже неактивно. Таким образом, в режиме циклической обработки событие аварии сбрасывается в каждом периоде ШИМ, хотя флаг аварии CBC остается установленным до принудительного программного сброса. Если после сброса регистра флага CBC вновь будет получено событие аварии, то флаг установится вновь. На рисунке 18.19 показана схема формирования прерывания.

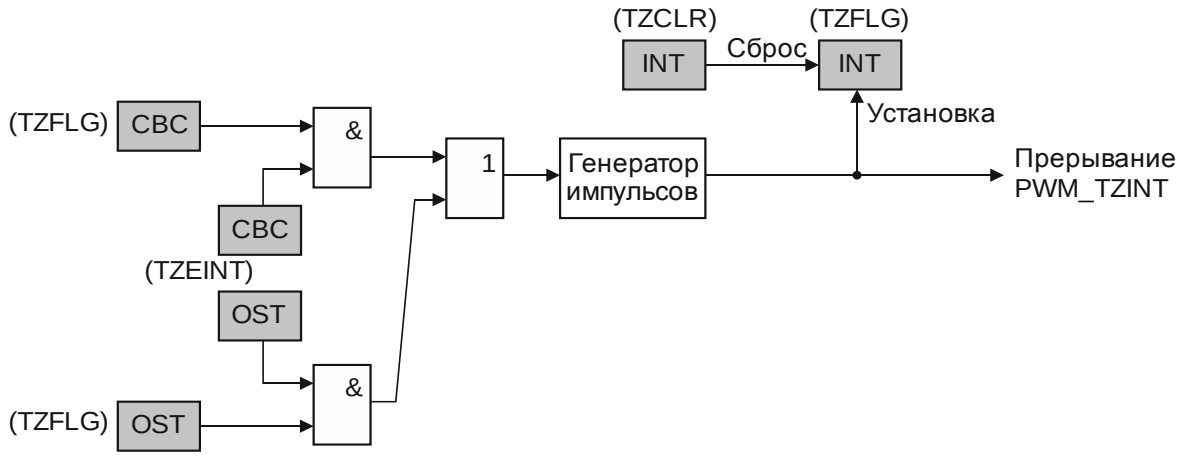


Рисунок 18.19 – Схема формирования прерывания

При получении события аварии в режиме однократной обработки немедленно выполняется действие, заданное регистром TZCTL, и устанавливается флаг OST в регистре TZFLG, а также генерируется прерывание PWM_TZINT, если разрешено в регистре TZEINT и контроллером прерываний. Аварийное удержание выводов заканчивается после принудительного программного сброса записью в бит OST регистра TZCLR.

Аварийное состояние выводов при получении события сбоя программируется индивидуально для выходов PWM_A и PWM_B полями TZA и TZB регистра TZCTL.

18.9 Триггер событий

Основные функции триггера событий:

- получение событий, сформированных таймером и компаратором;
- использование информации о направлении счета (вверх-вниз);
- использование делителя событий для формирования сигнала прерывания, запросов к блокам контроллера DMA, АЦП;
- предоставление доступа процессора к содержимому регистра флагов событий и счетчикам событий.

Триггер событий программируется посредством регистров ETSEL, ETPS, ETFLG, ETCLR и ETFRC.

Прерывания

Функциональная схема триггера событий для генерации прерываний показана на рисунке 18.20. Триггер может генерировать прерывания (если разрешено битом INTEN регистра ETSEL) по каждому событию, а также в два, три и четыре раза реже. Коэффициент деления событий задается полем INTPRD. Источник события выбирается полем INTSEL. Количество возникших событий отражается в поле INTCNT. Счетчик INTCNT считает от 00b до INTPRD и сбрасывается только вместе с отправкой активного прерывания.

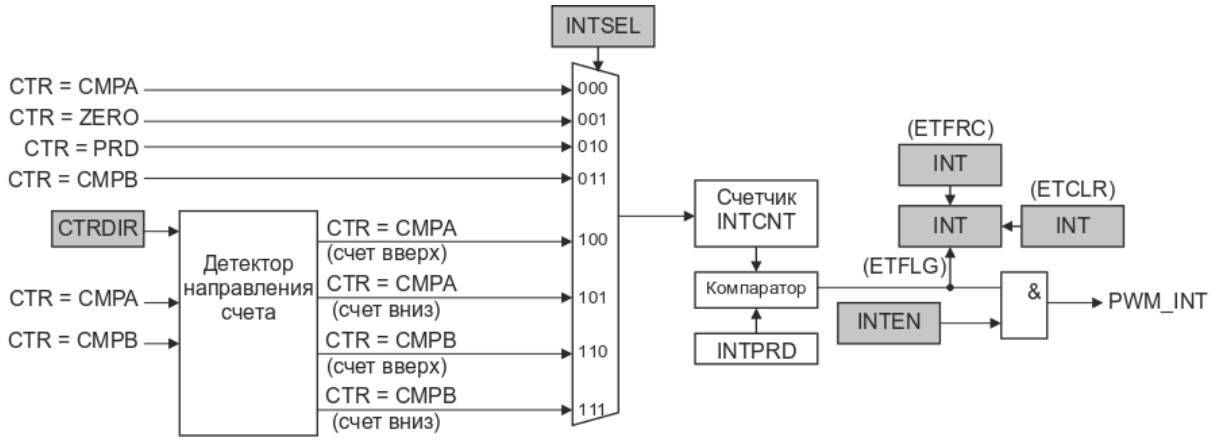


Рисунок 18.20 – Структурная схема триггера событий для генерации прерываний

Когда возникает совпадение значений счетчиков INTCNT и INTPRD, то возможны варианты:

- если прерывание разрешено и сброшен флаг INT (регистр ETFLG), то генерируется прерывание и устанавливается флаг INT, а счетчик INTCNT сбрасывается в 00b и начинает считать заново;
- если прерывание запрещено или флаг INT установлен, то счетчик перестает считать события;
- если прерывание разрешено, но флаг от предыдущего прерывания еще не сброшен, то счетчик хранит свое максимально достигнутое значение INTCNT = INTPRD до сброса флага INT. Это позволяет обработать еще прерывание, пришедшее за то время, пока обрабатывалось предыдущее.

Каждая запись в INTPRD сбрасывает счетчик INTCNT. Запись единицы в бит INT регистра ETFRC увеличит значение счетчика на единицу.

Сопряжение с блоком DMA

Механизм сопряжения модуля ШИМ с блоком DMA реализован аналогично механизму сопряжения модуля ШИМ с триггером событий.

Управление запросами блока DMA: сопряжение с сигналами PWM_DRQA (DMA_TX), PWM_DRQB (DMA_RX) осуществляется через регистры ETSEL, ETPS, ETFRC аналогично механизму управления сигналом прерывания PWM_INT. Функциональная схема триггера событий для генерации запросов блока DMA показана на рисунке 18.21.

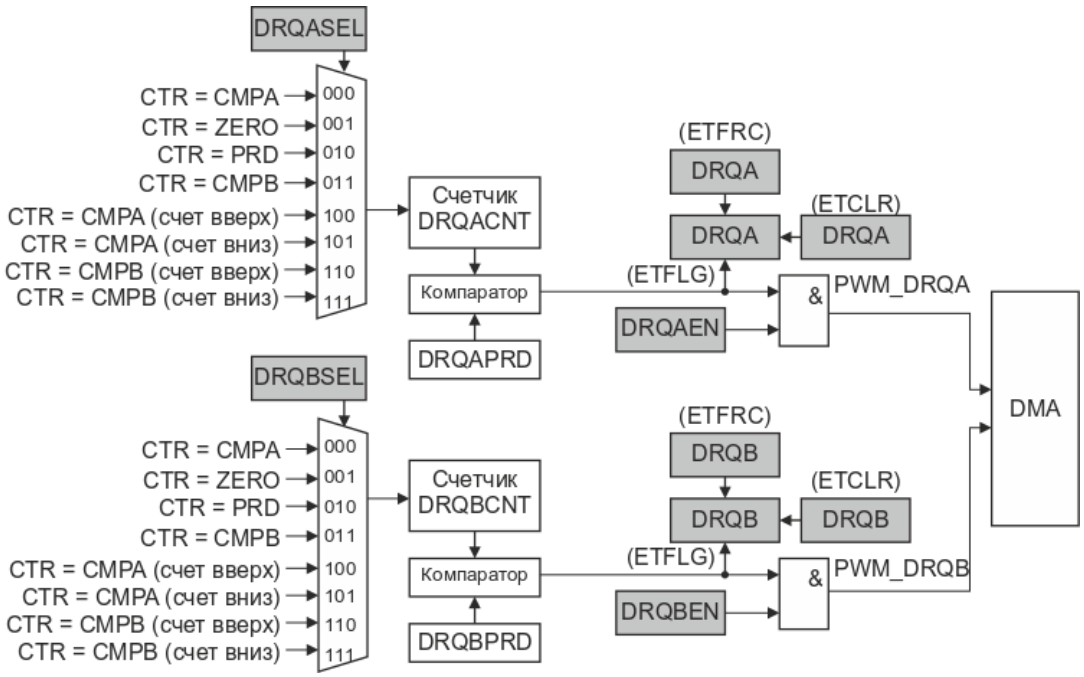


Рисунок 18.21 – Структурная схема триггера событий для генерации запросов контроллера DMA

Сопряжение с блоком АЦП

Управление сигналами запуска блока АЦП PWM_SOCA, PWM_SOCB осуществляется аналогично сигналу прерывания PWM_INT и запросам PWM_DRQA, PWM_DRQB через регистры ETSEL, ETPS, ETFRC.

Функциональная схема триггера событий для генерации сигналов запуска АЦП показана на рисунке 18.22.

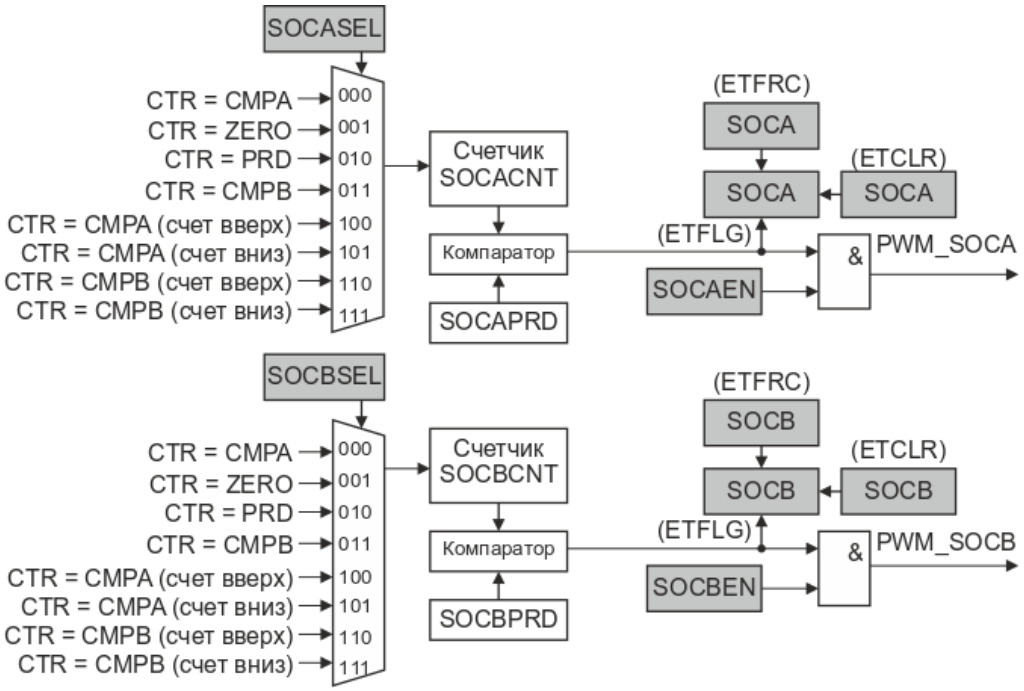


Рисунок 18.22 – Структурная схема триггера событий для генерации сигналов запуска АЦП

Выходы всех блоков ШИМ0 – ШИМ15 (PWMx_SOCA и PWMx_SOCB каждого блока, где x от 0 до 15) объединяются по ИЛИ, как показано на рисунке 18.23. Сигналы с выходов элементов ИЛИ защелкиваются в триггерах и формируют импульсы запуска.

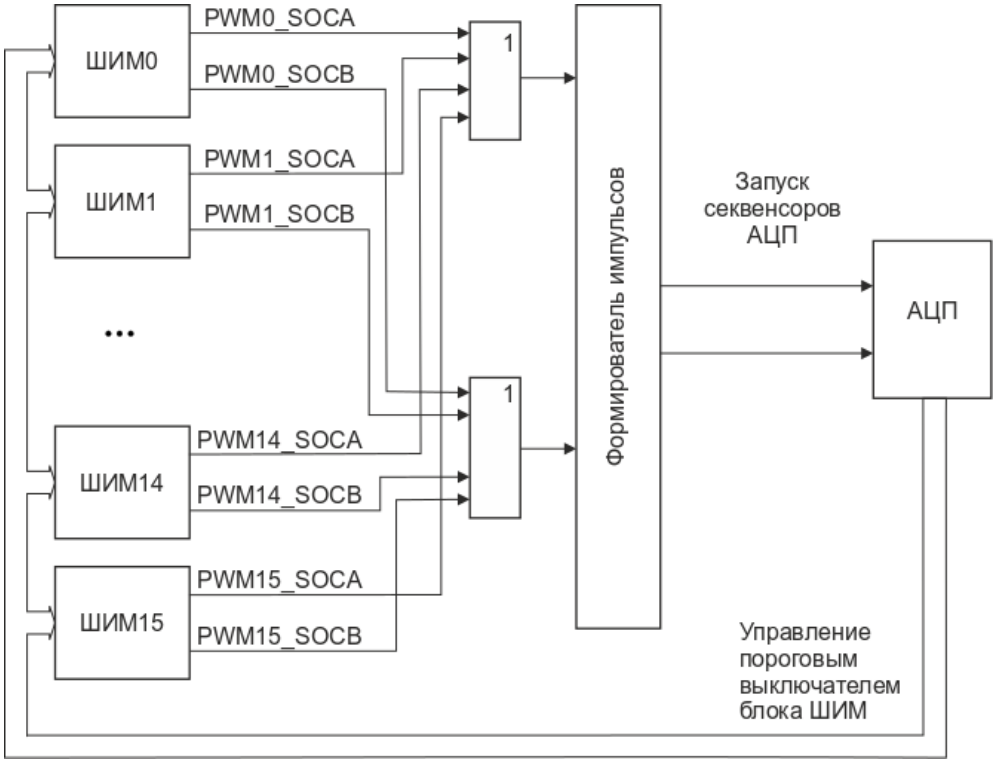


Рисунок 18.23 – Схема соединений между блоками ШИМ и АЦП

18.10 ШИМ высокого разрешения

Все шестнадцать блоков ШИМ0 – ШИМ15 микроконтроллера поддерживают функцию высокого разрешения.

Функция высокого разрешения блока ШИМ является дополнительной и имеет особенности:

- улучшенный контроль скважности выходного сигнала ШИМ канала А (сигнал PWMA);
- улучшенный контроль скважности выходного сигнала ШИМ канала В (сигнал PWMB);
- улучшенная точность переключения фронтов, с использованием увеличенной разрядности регистров CMPA и TBPHS (подключаются поля CMPAHR и TBPHSHR);
- логика калибровки для более точного определения значения реальной задержки.

Дополнительный регистр управления расширенными возможностями – HRCTL.

Функциональная схема функции высокого разрешения показана на рисунке 18.24.

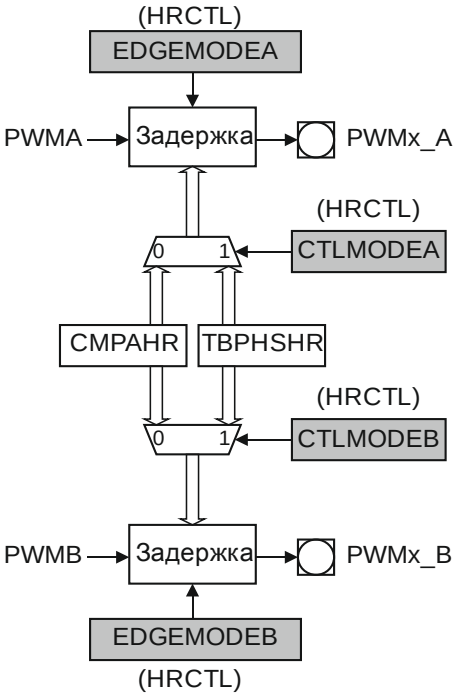


Рисунок 18.24 – Схема функции высокого разрешения

Улучшение разрешающей способности

В блоке ШИМ высокого разрешения используется специальная линия задержки с шириной отсчета 130 пс, что увеличивает разрешающую способность. Общее время, добавляемое линией задержки, складывается из времени задержки выходных мультиплексоров (t_{MUX_TYP}) в 1 нс и непосредственно времени задержки элементов (t_{ELEM_TYP}) по 130 пс за каждый элемент. Таким образом, при выбранной ненулевой задержке общее время в наносекундах рассчитывается по формуле:

$$t_{DEL} = 1 + 0{,}13 \times n, \qquad (18.3)$$

где n – количество подключаемых элементов задержки (через поле CMPAHR или TBPHSHR).

Конфигурация

Для каждого из выводов PWMA и PWMB возможен независимый выбор фронта, на который оказывает влияние линия задержки (поле EDGMODE). Можно установить влияние линии задержки только на передний фронт, на задний фронт или на любой из них. Также для каналов можно осуществить независимый выбор регистра, устанавливающего время задержки (поле CTLMODE). Можно использовать регистр CMPAHR или TBPHSHR. Поле CMPAHR может использовать отложенную загрузку (теневой регистр), полностью аналогично тому, как это реализовано для регистра CMPA. Дополнительно полем HRLOAD регулируется событие, по которому произойдет теневая загрузка – при сброшенном HRLOAD загрузка CMPAHR из теневого регистра будет произведена по достижению счетчиком нуля, а при установленном HRLOAD загрузка будет произведена по достижению счетчиком периода.

Ограничения диапазона скважности

Линия задержки не может работать в течение всего периода формируемого сигнала ШИМ, поэтому вводится ограничение на минимальную длительность импульса – три периода PCLK.

В случае если в соответствии с требуемой скважностью длительность импульса выходного сигнала ШИМ оказалась короче минимальной, то фронты будут выставлены с точностью, как и в обычном блоке ШИМ.

При работе на низких частотах ограничение по управлению скважностью сигнала ШИМ практически незаметно.

Суммарная задержка, генерируемая блоком ШИМ высокого разрешения, не может превышать двух периодов PCLK.

Калибровка

Линия задержки состоит из 255 одинаковых, последовательно включенных элементов. Приблизительное время задержки одного элемента составляет 130 пс и определяется технологическими нормами, в соответствии с которыми выполнен микроконтроллер. Длительность отсчета также может незначительно меняться, в зависимости от текущего значения питания микроконтроллера и текущей температуры.

Для коррекции значения задержки в регистре HRCTL предусмотрены поля флагов DELAYCALA и DELAYCALB. Процесс калибровки заключается в настройке блока ШИМ и постепенном увеличении величины задержки с постоянным контролем флагов. Как только величина суммарной задержки превысит 1 период PCLK – DELAYCALA или DELAYCALB примут значение, отличное от нуля. Далее необходимо обнулить эти поля и сделать инкремент задержки на величину в половину текущего значения. Затем продолжить инкрементировать величину задержки. Когда суммарная задержка достигнет двух периодов, DELAYCALA или DELAYCALB примут значение 3h.

Пример: при частоте сигнала PCLK в 100 МГц (T_{PCLK} = 10 нс) соответствующие флаги появятся при значении задержки в 68 и 148 тактов соответственно. Следовательно,

$$1 \times T_{PCLK} = t_{MUX} + 68 \times t_{ELEM}, \quad (18.4)$$

$$2 \times T_{PCLK} = t_{MUX} + 148 \times t_{ELEM}. \quad (18.5)$$

Используя полученные равенства, можно вычислить реальное время задержки мультиплексора t_{MUX} = 1,5 нс и реальное время задержки одного элемента t_{ELEM} = 0,125 нс.

19 Блок АЦП (ADC)

19.1 Общее описание

Блок АЦП объединяет четыре модуля АЦП последовательного приближения, схему управления, буферы результатов измерений и схему управления прерываниями.

Управляющие регистры блока АЦП приведены в приложении А.13.

Структурная схема блока АЦП показана на рисунке 19.1.

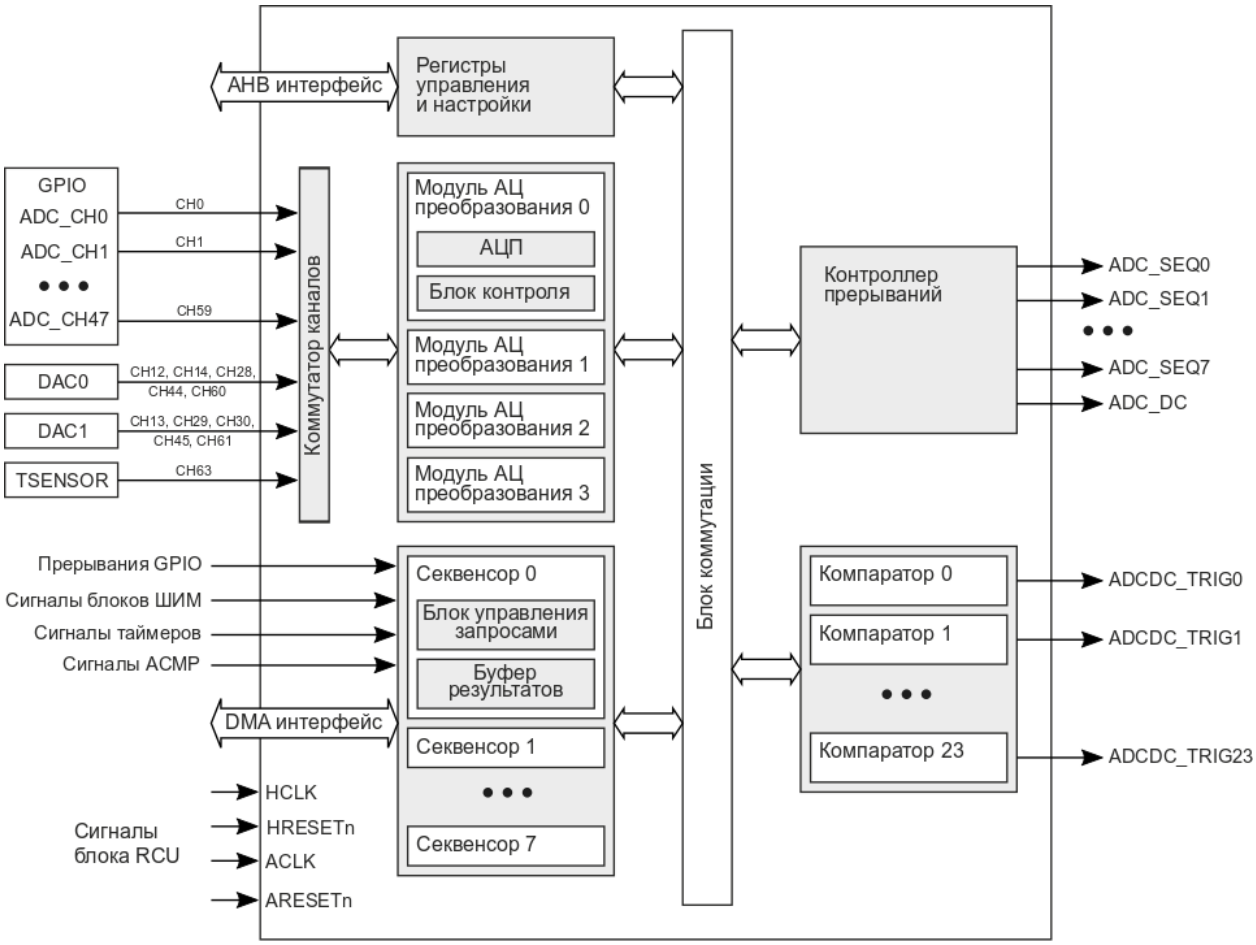


Рисунок 19.1 – Структурная схема блока АЦП

В блок АЦП входят:

- четыре модуля АЦП с настраиваемой разрядностью 6, 8, 10, 12 бит и скоростью измерения по одному каналу в 12-битном режиме до 2,5 миллионов измерений в секунду;
- восемь секвенсоров, каждый из которых позволяет независимо произвести запуск измерений по необходимым каналам АЦП и сгенерировать прерывание;
- 24 независимых цифровых компаратора, отслеживающих и сравнивающих измерения с пороговыми значениями для формирования прерываний и сигналов управления другими блоками микроконтроллера;
- восемь буферов результатов измерений (каждый организован по типу FIFO);
- блок управления прерываниями.

Блок АЦП суммарно имеет суммарно 64 аналоговых входных канала, из которых:

- 48 выводов подключены к портам ввода-вывода GPIO для измерения внешних сигналов;
- 10 тестовых выводов подключены к блокам ЦАП для проверки функционирования АЦП;
- 1 вывод подключен к внутреннему датчику температуры TSENSOR;
- 5 выводов принудительно притянуты к уровню GND.

В таблице 19.1 приведено соответствие каналов блока АЦП и их подключение.

Таблица 19.1 – Подключение каналов блока АЦП

№ модуля АЦП	№ канала блока АЦП	Подключение канала АЦП (Аналоговая функция)	№ модуля АЦП	№ канала блока АЦП	Подключение канала АЦП (Аналоговая функция)
0	CH0	GPIOA[0] (ADC_CH0)	2	CH32	GPIOB[8] (ADC_CH24)
	CH1	GPIOA[1] (ADC_CH1)		CH33	GPIOB[9] (ADC_CH25)
	CH2	GPIOA[2] (ADC_CH2)		CH34	GPIOB[10] (ADC_CH26)
	CH3	GPIOA[3] (ADC_CH3)		CH35	GPIOB[11] (ADC_CH27)
	CH4	GPIOA[4] (ADC_CH4)		CH36	GPIOB[12] (ADC_CH28)
	CH5	GPIOA[5] (ADC_CH5)		CH37	GPIOB[13] (ADC_CH29)
	CH6	GPIOA[6] (ADC_CH6)		CH38	GPIOB[14] (ADC_CH30)
	CH7	GPIOA[7] (ADC_CH7)		CH39	GPIOB[15] (ADC_CH31)
	CH8	GPIOA[8] (ADC_CH8)		CH40	GPIOC[0] (ADC_CH32)
	CH9	GPIOA[9] (ADC_CH9)		CH41	GPIOC[1] (ADC_CH33)
	CH10	GPIOA[10] (ADC_CH10)		CH42	GPIOC[2] (ADC_CH34)
	CH11	GPIOA[11] (ADC_CH11)		CH43	GPIOC[3] (ADC_CH35)
	CH12	DAC0 (ADC0_TEST)		CH44	DAC0 (ADC2_TEST)
	CH13	DAC1 (ADC0_TEST)		CH45	DAC1 (ADC2_TEST)
	CH14	DAC0 (REF_TEST)		CH46	GND
	CH15	GND		CH47	GND
1	CH16	GPIOA[12](ADC_CH12)	3	CH48	GPIOC[4] (ADC_CH36)
	CH17	GPIOA[13] (ADC_CH13)		CH49	GPIOC[5] (ADC_CH37)
	CH18	GPIOA[14] (ADC_CH14)		CH50	GPIOC[6] (ADC_CH38)
	CH19	GPIOA[15] (ADC_CH15)		CH51	GPIOC[7] (ADC_CH39)
	CH20	GPIOB[0] (ADC_CH16)		CH52	GPIOC[8] (ADC_CH40)
	CH21	GPIOB[1] (ADC_CH17)		CH53	GPIOC[9] (ADC_CH41)
	CH22	GPIOB[2] (ADC_CH18)		CH54	GPIOC[10] (ADC_CH42)
	CH23	GPIOB[3] (ADC_CH19)		CH55	GPIOC[11] (ADC_CH43)
	CH24	GPIOB[4] (ADC_CH20)		CH56	GPIOC[12] (ADC_CH44)
	CH25	GPIOB[5] (ADC_CH21)		CH57	GPIOC[13] (ADC_CH45)
	CH26	GPIOB[6] (ADC_CH22)		CH58	GPIOC[14] (ADC_CH46)
	CH27	GPIOB[7] (ADC_CH23)		CH59	GPIOC[15] (ADC_CH47)
	CH28	DAC0 (ADC1_TEST)		CH60	DAC0 (ADC3_TEST)
	CH29	DAC1 (ADC1_TEST)		CH61	DAC1 (ADC3_TEST)
	CH30	DAC1 (REF_TEST)		CH62	GND
	CH31	GND		CH63	TSENSOR

Примечание – № канала блока АЦП используется при управлении каналами АЦП с помощью регистров. Аналоговая функция указана в соответствии с таблицей 3.1.

Диапазон измеряемых напряжений ограничен напряжением на выводе AVCC_ADC и землей GND.

Настройка тактирования и сброса блока АЦП и его модулей осуществляется посредством регистра ADCCFG блока управления тактовыми сигналами RCU. Вся внутренняя логика блока АЦП тактируется частотой ACLK, но запись/чтение контрольно-статусных регистров осуществляется на частоте HCLK.

Для правильной работы блока, необходимо обеспечить тактирование модулей АЦП частотой внутреннего сигнала ACLK от 140 кГц до 35 МГц, которую можно получить, выбрав источник тактового сигнала полем CLKSEL, а также, при необходимости, включив и настроив делитель полями DIVEN и DIVN (поля регистра ADCCFG блока RCU). Частота ACLK не должна быть больше частоты тактового сигнала HCLK.

Время, на которое канал измерения подключается к внутренней зарядной емкости АЦП составляет период одного такта частоты ACLK. Предусмотрена возможность увеличения этого времени с помощью дополнительных тактов ожидания, которые задаются отдельно для каждого канала в массиве регистров ADC->CHDELAY[]. В первую очередь это необходимо при подключении на вход канала АЦП высокоомной нагрузки.

Для проверки функционирования АЦП предлагается программно-аппаратный метод с использованием блоков ЦАП. У каждого модуля АЦП канал 12 соединен с тестовым выходом ЦАП0, а канал 13 с тестовым выходом ЦАП1. Тестовый выход ADCx_TEST или REF_TEST активируется записью в регистр TEST блока ЦАП в соответствующее поле.

19.2 Секвенсор

Секвенсор представляет собой управляющий блок, позволяющий разгрузить процессор от управления модулями АЦП. Секвенсор управляет запуском модулей АЦП, обработкой полученных результатов измерений и генерацией прерываний. В состав блока АЦП входят s секвенсоров (где s от 0 до 7). Структурная схема секвенсора представлена на рисунке 19.2.

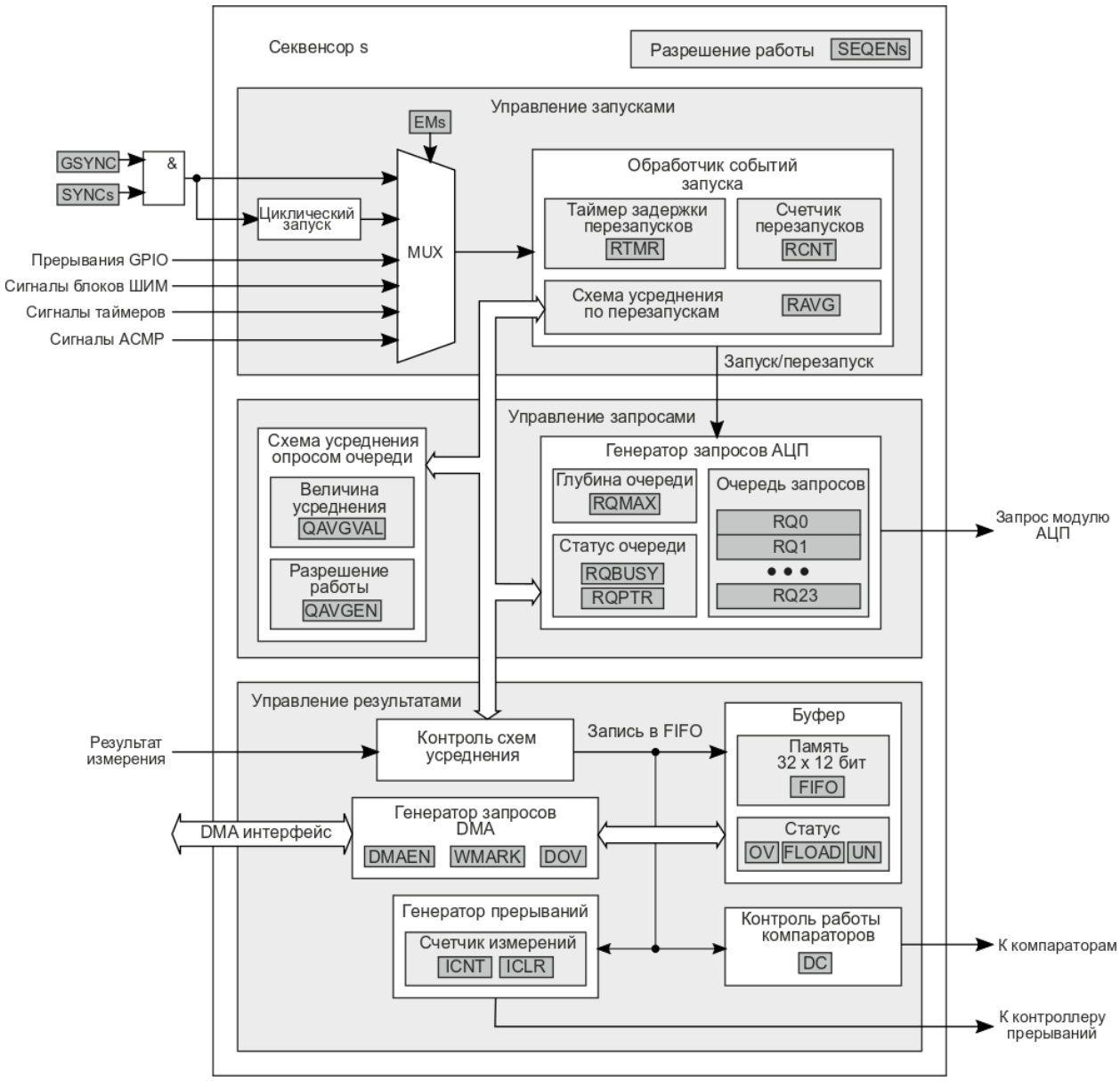


Рисунок 19.2 – Структурная схема секвенсора

Одиночные запуски по событиям

Разрешение работы секвенсора осуществляется установкой соответствующего бита SEQENs (где s от 0 до 7) в регистре SEQEN.

Каждый секвенсор может совершать независимые однократные запуски по одному из событий, которое выбирается полем EMs регистра EMUX:

- установка бита GSYNC регистра SEQSYNC (запустятся только секвенсоры, для которых установлены биты SYNCs того же регистра);
- сигналы от таймеров TMR;
- сигналы от блоков ШИМ;
- сигналы от блоков ACMP;
- сигналы от GPIO.

Когда секвенсор s запускается по одному из сигналов событий, выставляется соответствующий флаг SEQBUSYs в регистре BSTAT. Также в это же время все настройки секвенсора сохраняются в теневых регистрах, и секвенсор начинает работу согласно полученным настройкам. Изменять настройки секвенсора во время его работы можно, но вступят в силу они лишь при следующем запуске. Флаг занятости держится установленным до тех пор, пока задача, инициированная событием, не будет полностью выполнена

секвенсором (будет осуществлена запись последнего результата в FIFO).

События запуска не кэшируются, если секвенсор был занят выполнением текущей задачи (установлен SEQBUSYs), когда пришло очередное событие запуска, то оно будет проигнорировано. Необходимо учитывать это при настройке запуска по событиям от сторонних периферийных модулей так, чтобы время между возникновением событий было не меньше времени измерений. Диаграммы работы секвенсора при одиночных запусках по событиям показаны на рисунке 19.3.

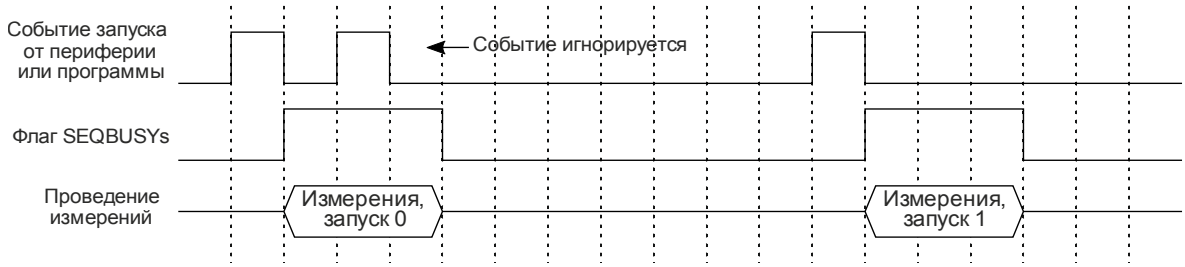


Рисунок 19.3 – Диаграммы работы секвенсора при одиночных запусках по событиям

Одиночные запуски по событиям с немедленными перезапусками

Здесь и далее, под перезапусками понимается внутренний механизм работы секвенсора, а под запусками – приход внешнего события, которое переводит секвенсор из ожидающего состояния в активное.

Секвенсор имеет возможность осуществлять как отложенные, так и немедленные автоматические перезапуски серий измерений, после прихода первого «инициирующего» события запуска от периферии. Перезапуск может выполняться до 255 раз (поле RCNT регистра SCCTL). Текущее состояние счетчика перезапусков можно узнать, прочитав поле RCNT регистра SCVAL. Диаграммы работы секвенсора при разрешенных, немедленных перезапусках показаны на рисунке 19.4.

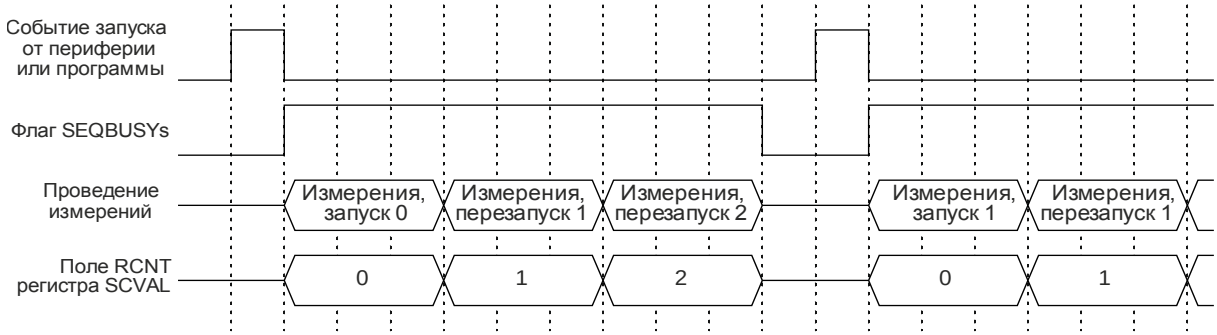


Рисунок 19.4 – Диаграммы работы секвенсора при одиночных запусках по событиям с немедленными перезапусками при RCNT = 2 (регистр SCCTL)

Одиночные запуски по событиям с отложенными перезапусками

Отложенные перезапуски осуществляются спустя некоторое время от запуска, задаваемое регистром SRTMR. Задержка задается в тактах ACLK и ведет счет независимо от текущего состояния секвенсора, поэтому, при её выборе необходимо учитывать, что текущие измерения должны завершиться до прихода сигнала перезапуска, иначе он будет пропущен.

В режиме одиночных запусков по событиям счетчик задержки перезапуска не будет считать, если поле RCNT регистра SCCTL равно нулю.

Как говорилось ранее, все настройки секвенсора сохраняются в теневых регистрах при его переходе в режим занятости и их изменение никак не повлияет на текущую работу. Однако существует возможность обновить задержку перезапуска еще во время работы секвенсора. Для этого надо записать новое значение задержки в регистр SRTMR с одновременно установленным последним битом NOWAIT. В этом случае, новая величина задержки вступит в силу после ближайшего события отложенного перезапуска.

Диаграммы работы секвенсора при активных отложенных перезапусках показаны на рисунке 19.5.

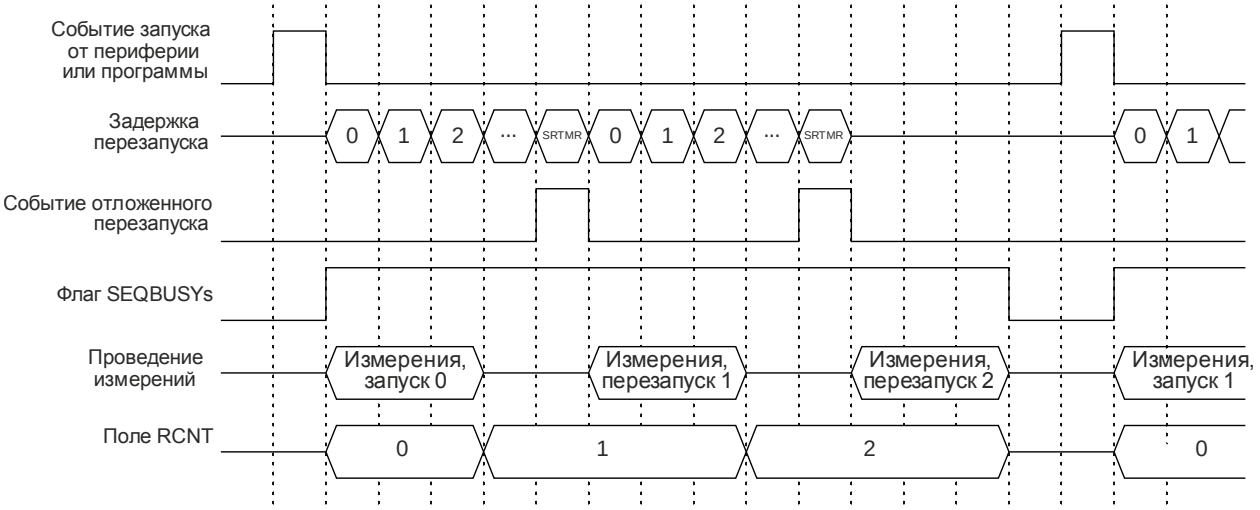


Рисунок 19.5 – Диаграммы работы секвенсора при одиночных запусках по событиям с отложенными перезапусками через определенное время (регистр SRTMR) при RCNT = 2 (регистр SCCTL)

Одиночные запуски с усреднением по перезапускам

Существует режим усреднения результатов по перезапускам, который включается установкой бита RAVGEN в регистре SCCTL. Главным условием работы этого режима является то, что поле RCNT регистра SCCTL должно содержать любое значение, соответствующее $2^p - 1$, где p равно от 1 до 8. Значение 2^p и является количеством серий измерений, которые будут усреднены (запуск и все перезапуски).

Работа этого режима заключается в том, что пока идут перезапуски, результаты попадут в буфер не сразу, а будут накапливаться во внутренних регистрах (каждому запросу на измерение соответствует такой регистр). Лишь во время последнего перезапуска, будут получены усреднённые значения по каждому из измерений, которые и будут помещаться в FIFO в порядке очереди.

Работа режима усреднения при немедленных перезапусках продемонстрирована на рисунке 19.6. При отложенных перезапусках данный вид усреднения работает аналогично, позволяя распределить равномерно измерения на длительном промежутке времени и получить среднее значение сигнала в конце него.

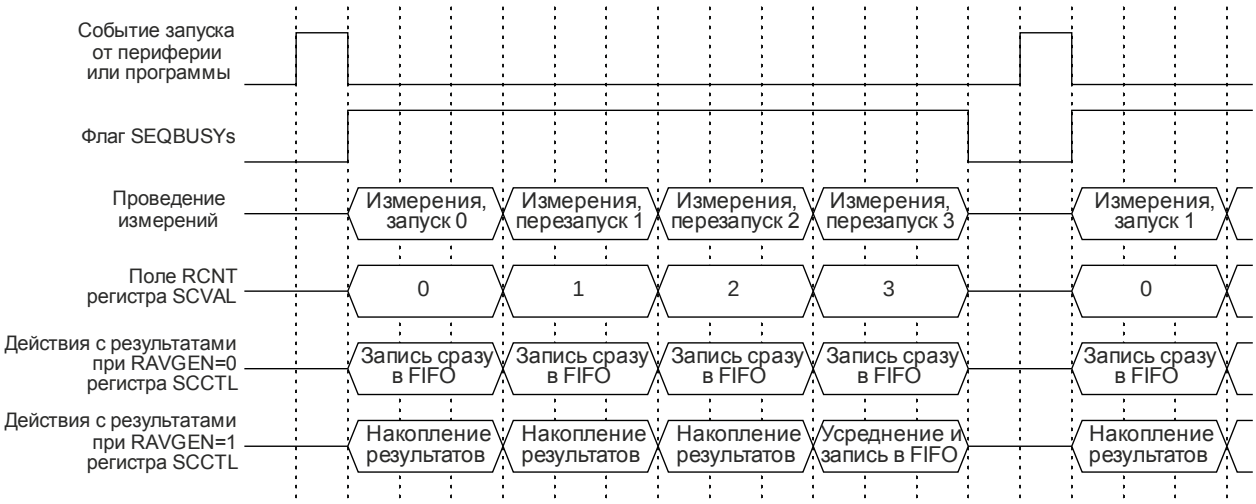


Рисунок 19.6 – Диаграммы работы секвенсора при одиночных запусках по событиям с усреднением по перезапускам при RCNT=3, RAVGEN=1 (регистр SCCTL)

Например, если измерения проводились по четырём каналам при RAVGEN = 0, то на момент снятия флага SEQBUSYs в FIFO было бы 16 результатов, но если усреднение по перезапускам было бы активно, то в FIFO находилось бы четыре усредненных результата по каждому из каналов.

Циклический запуск

Секвенсор может быть запрограммирован на циклический запуск – он будет запускаться снова каждый раз при завершении предыдущего запуска (имитация постоянно активного внешнего события). Чтобы начать работу в циклическом режиме, необходимо, после соответствующей конфигурации регистра EMUX, установить бит GSYNC регистра SEQSYNC. Чтобы завершить работу в циклическом режиме, необходимо выбрать в поле EMs регистра EMUX любое событие однократного запуска. Флаг занятости секвенсора SEQBUSYs в регистре BSTAT будет установлен сразу же по входу в циклический режим и будет сброшен только лишь по выходу из него. Диаграммы работы секвенсора при циклическом запуске показаны на рисунке 19.7.



Рисунок 19.7 – Диаграммы работы секвенсора при циклическом запуске

Циклический отложенный запуск

В отличие от режима одиночных запусков, циклический режим позволяет активировать счетчик задержки SRTMR, даже если поле RCNT регистра SCCTL равно нулю. В этом случае измерения будут запускаться не непрерывно, а с некоторой паузой (определяемой регистром SRTMR). Диаграммы работы секвенсора при циклическом запуске показаны на рисунке 19.8.

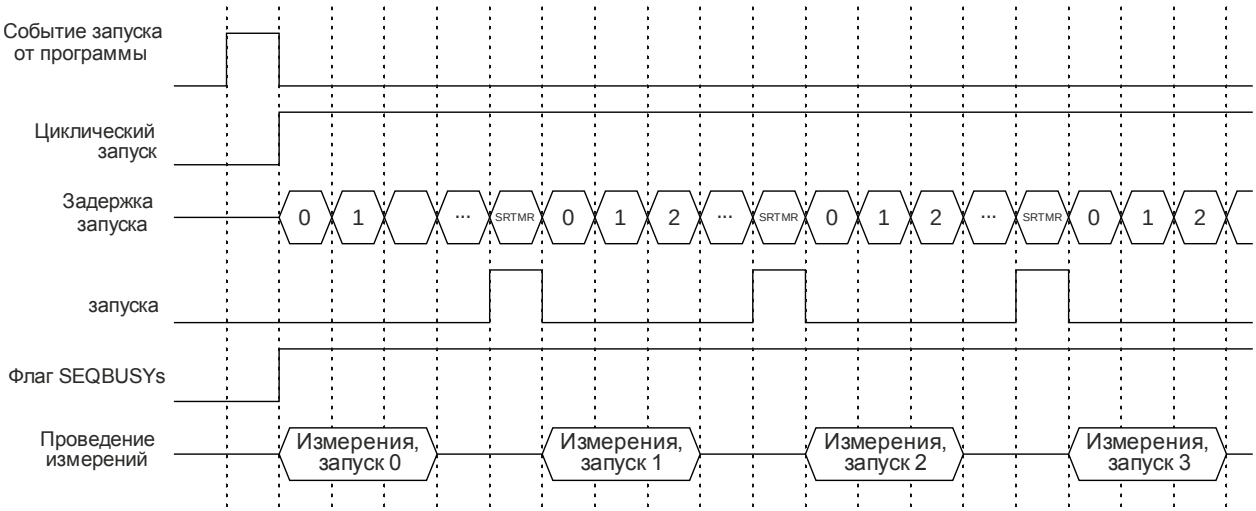


Рисунок 19.8 – Диаграммы работы секвенсора при циклическом отложенном запуске через время SRTMR

Циклический запуск с усреднением по перезапускам

Если в циклическом режиме установить поле RCNT регистра SCCTL значением, отличным от нуля, то секвенсор между запусками начнет делать нужное количество перезапусков. Но механика такого режима внешне никак не будет отличаться от обычной циклической работы, поэтому данный режим имеет смысл лишь в том случае, когда необходимо скомбинировать циклический режим с усреднением по перезапускам (как немедленным, так и отложенным).

Диаграммы работы секвенсора в циклическом режиме запуска с усреднением по отложенным перезапускам показаны на рисунке 19.9.

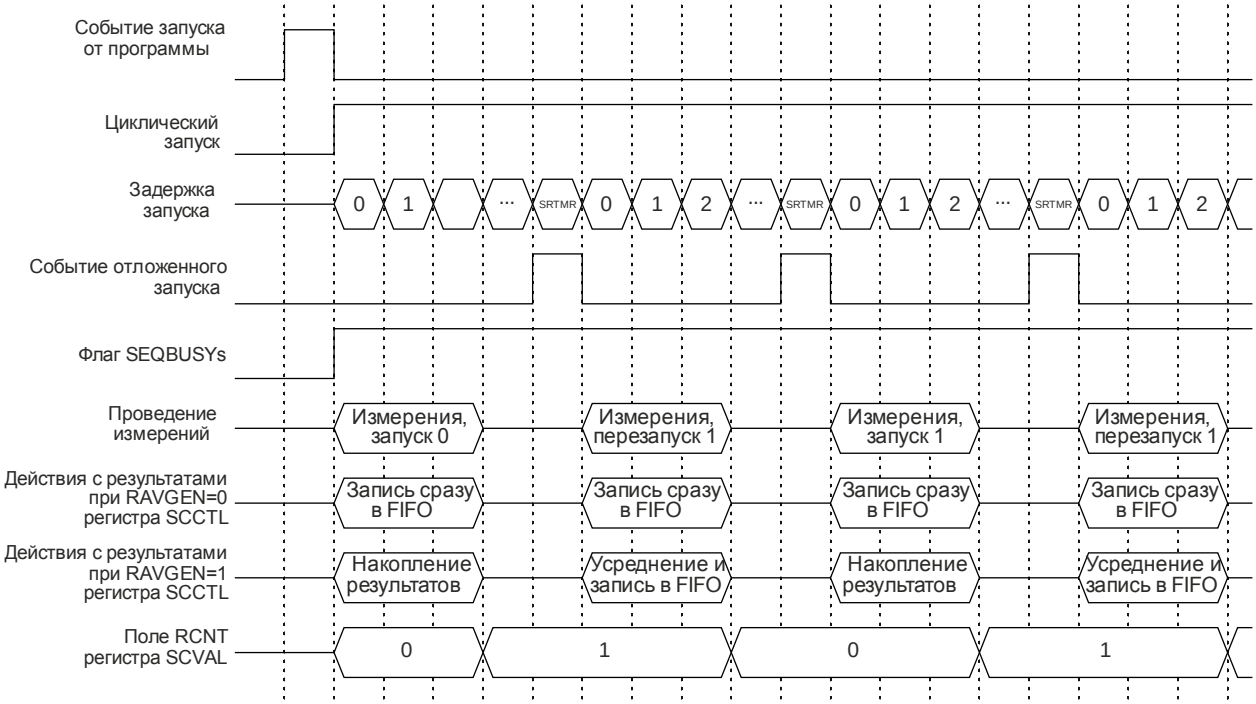


Рисунок 19.9 – Диаграммы работы секвенсора в циклическом запуске с отложенными перезапусками через определенное время (регистр SRTMR) при RCNT=1 (регистр SCCTL)

Генерация запросов на измерение

После запуска по событию или очередного перезапуска секвенсор начинает формировать запросы на измерения по каналам в порядке очереди, заданной регистром SRQSEL. Конец очереди или её «глубина» задается полем RQMAX регистра SRQCTL.

Определить состояние очереди можно с помощью регистра SRQSTAT – в поле RQPTR находится номер текущего запроса по порядку, а установленный флаг занятости RQBUSY говорит о том, что запрос выставлен и в состоянии обработки.

Иллюстрация к механизму генерации запросов на измерение представлена на рисунке 19.10.

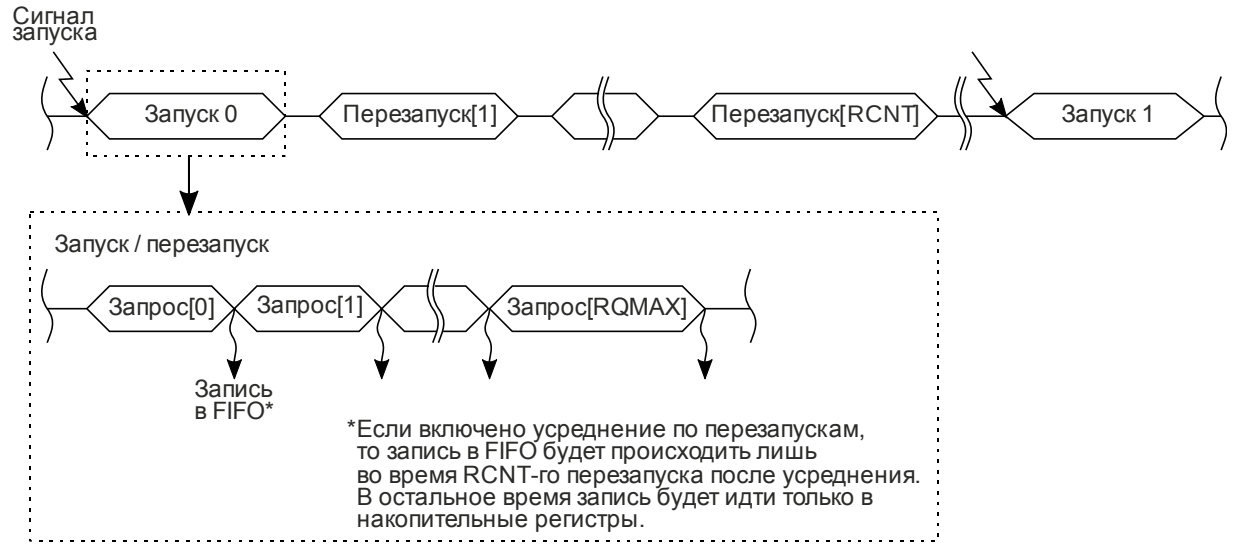


Рисунок 19.10 – Генерация секвенсором запросов на измерение

Пояснения к рисунку 19.10:

1 Как только секвенсор получает сигнал запуска, он выставляет первый запрос из очереди и ожидает его принятия модулем АЦП.

2 Когда запрос будет принят, АЦП проведет измерение. По окончании измерения результат выполнения запроса будет передан секвенсору.

3 Секвенсор сохраняет результат в FIFO и продолжает выставлять запросы до окончания их очереди.

4 Если секвенсор настроен на проведение перезапусков (отложенных или немедленных), то по окончании очереди секвенсор перезапускается и снова выставляет первый в очереди запрос.

5 Лишь когда завершен последний перезапуск, то работа секвенсора, начатая по первичному событию запуска (пояснение 1), считается завершенной.

6 Секвенсор переходит в состояние ожидания следующего сигнала запуска.

Результат каждого запроса сохраняется в кольцевом буфере результатов секвенсора SFIFO. Буферы всех секвенсоров имеют емкость в 32 12-битных слова. Текущее количество слов в буфере можно узнать, прочитав регистр SFLOAD.

Контроль состояния буфера осуществляется посредством флагов регистра FSTAT. Установленный флаг OVs, указывает на то, что в FIFO не осталось свободных ячеек. Любая запись в буфер в таком случае будет игнорироваться до появления хотя бы одной свободной ячейки. Сброшенный флаг UNs свидетельствует о наличии в FIFO как минимум одного результата измерения. Соответственно, флаг UNs установится, когда FIFO будет полностью пуст, при этом результатом чтения пустого FIFO будут нули. Сброс флагов осуществляется путем записи в них единицы.

Усреднение сканированием очереди

Наряду с усреднением по перезапускам секвенсор имеет дополнительный механизм усреднения – усреднение сканированием (опросом) очереди измерений. Оба механизма могут работать как вместе, так и по отдельности.

Включается усреднение сканированием путем установки бита QAVGEN в регистре SRQCTL. Перед включением режима необходимо задать количество усредняемых опросов

в поле QAVGVAL того же регистра – оно считается как 2^{QAVGVAL}.

Работа режима заключается в том, что очередь запросов выполняется не один раз (рисунок 19.10) с сохранением результатов в буфер после каждого запроса, а 2^{QAVGVAL} раз с сохранением результатов лишь на последней итерации сканирования после усреднения всех полученных измерений. Результаты измерений накапливаются и усредняются индивидуально для каждого запроса (аналогично усреднению по перезапускам).

Диаграммы работы усреднения сканированием показаны на рисунке 19.11.

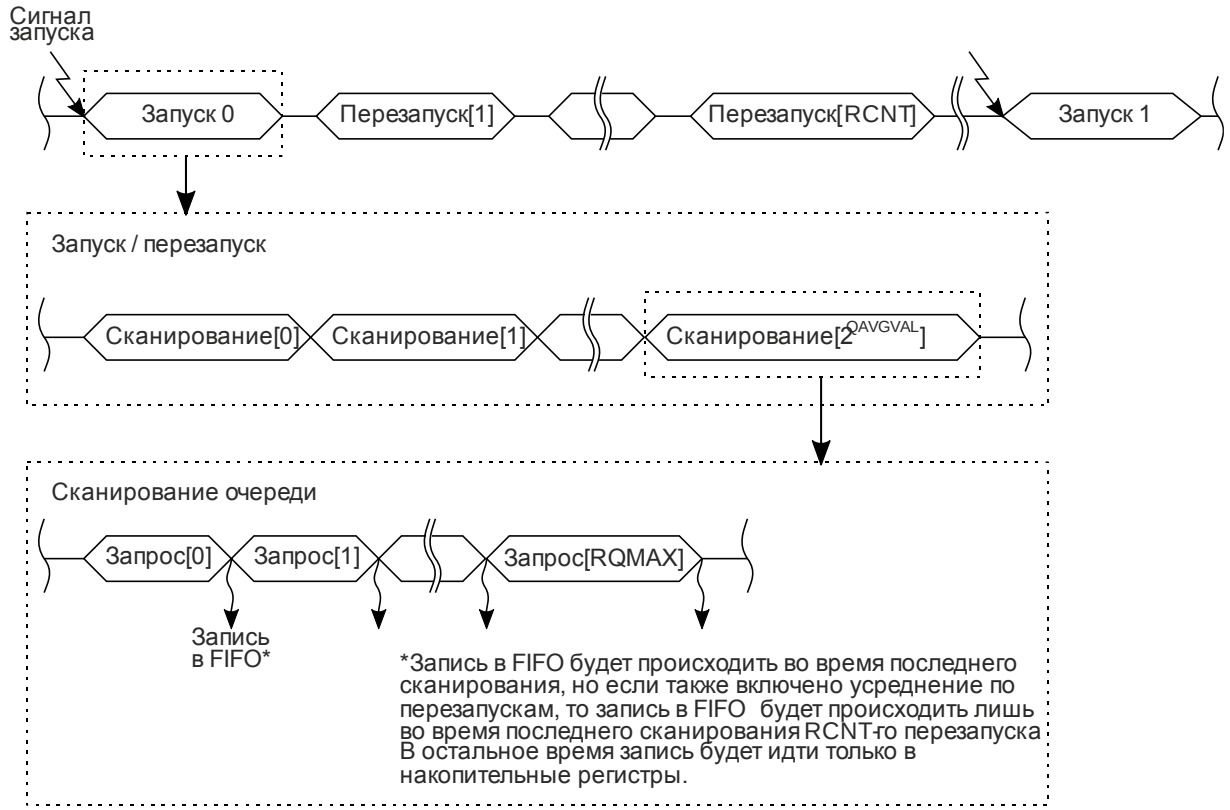


Рисунок 19.11 – Режим усреднения сканированием

Усреднение сканированием позволяет с относительно малой фазовой задержкой измерить уровень сигнала по нескольким каналам в пределах одной временной точки, а усреднение по перезапускам дает возможность усреднить значения, полученных таким образом точек, на достаточно большом интервале времени. В совокупности, оба механизма предлагают довольно гибкий инструментарий для автоматизации и усреднения измерений.

Генерация прерываний

Секвенсор может генерировать прерывания с заданной периодичностью. По завершении каждой записи в FIFO инкрементируется счетчик измерений. Как только было зафиксировано ICNT+1 записей (поле регистра SCCTL), генерируется прерывание. Стоит отметить, что даже если FIFO заполнено полностью (установлен OVs в регистре FSTAT), а измерения продолжают проводиться – счетчик измерений все равно будет считать последующие попытки записи секвенсора в FIFO (хотя они и будут игнорироваться самим FIFO). Текущее состояние счетчика можно узнать, прочитав поле ICNT регистра SCVAL. Сброс счетчика запросов происходит в следующих случаях:

- зафиксировано ICNT+1 записей;
- при запуске секвенсора по событию, если сброшен бит ICNTs в регистре CICNT;
- бит разрешения работы секвенсора ENs регистра SEQEN сброшен;
- программно – при каждой записи единицы в поле ICLR регистра SCVAL.

Для каждого секвенсора выделена линия прерываний – ADC_SEQs.

Использование прямого доступа к памяти

Для разрешения использования DMA секвенсором, необходимо установить бит DMAEN в регистре SDMACTL. Поле WMARK того же регистра задает уровень заполнения буфера секвенсора, по достижении которого будет запущен DMA. Перенос данных будет выполняться, пока не будет передано число результатов измерений, соответствующее состоянию поля WMARK.

Если очередной запрос на запуск DMA пришел раньше, чем закончился предыдущий цикл DMA от того же секвенсора, то будет выставлен флаг ошибки DOVs в регистре FSTAT.

19.3 Модуль АЦП

Структурная схема одного 16-канального модуля АЦП показана на рисунке 19.12. Суммарное количество модулей АЦП равняется m (m равно 0 – 3). Подключение каналов к модулям АЦП приведено в таблице 19.1.

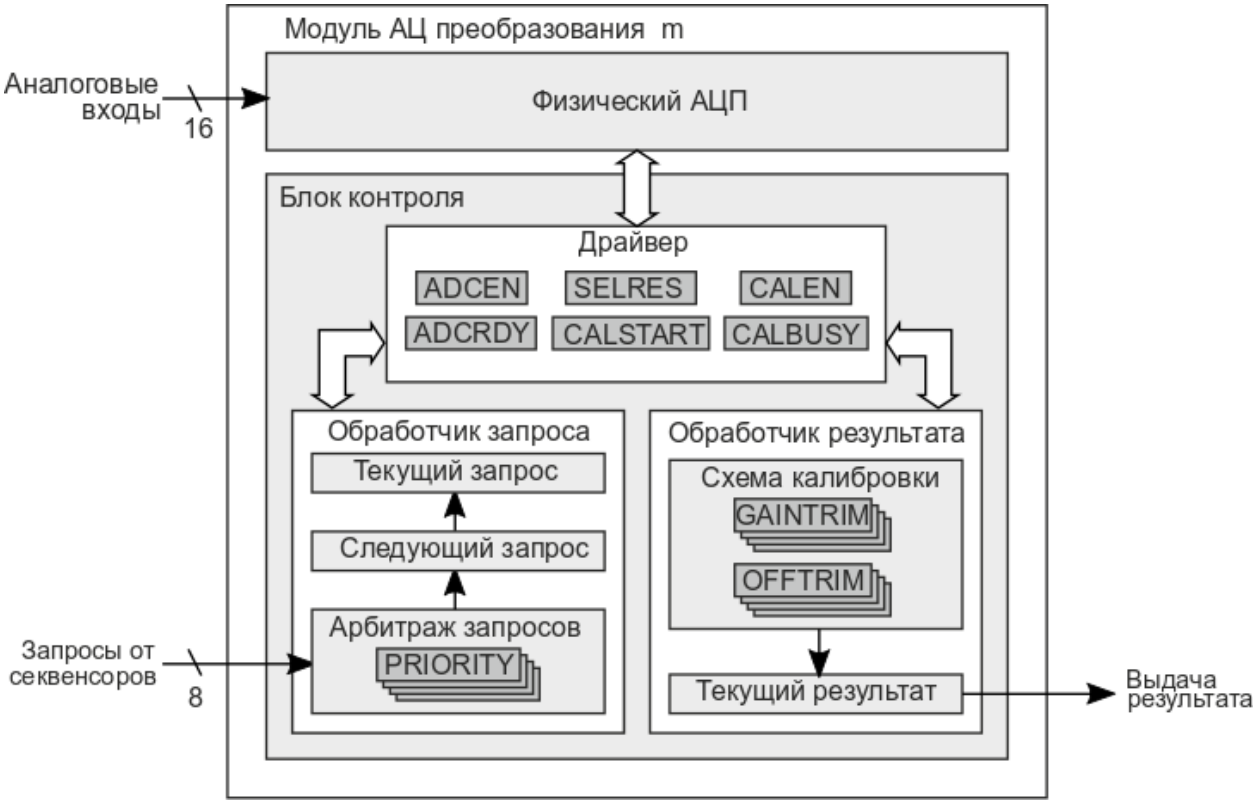


Рисунок 19.12 – Структурная схема 16-канального модуля АЦП

Перед началом работы с АЦП необходимо подготовить к работе каждый из используемых модулей. Для начала - вывести его из режима электрической изоляции, сбросив бит DISLVL регистра ACTLm. Каждый физический АЦП содержит внутренний линейный регулятор напряжения (LDO), который должен быть включен (бит ADCENLDO регистра ACTLm) перед разрешением работы модуля. Далее следует разрешить работу самого АЦП, установив бит ADCEN.

Старт работы модуля АЦПm (где m от 0 до 3) осуществляется установкой бита ADCSTART в регистре ACTLm. При каждом переключении ADCSTART в единицу запускается процедура инициализации АЦП, которая завершается установкой бита ADCRDY регистра ACTLm. АЦП готов к работе, когда ADCRDY = 1. Для правильной работы блока, необходимо обеспечить тактирование модулей АЦП (сигнал ACLK) частотой от 140 кГц до 35 МГц.

Физический блок АЦП имеет внутреннюю схему калибровки смещения нуля. Схема включается при установке бита CALEN регистра ACTLm. Ручной запуск калибровки запускается при установленном бите CALEN по событию записи единицы в бит CALSTART того же регистра. В начале процесса калибровки устанавливается статусный бит CALBUSY, который сбросится по её окончанию. Также по завершению процесса обновится поле CALOUT, которое будет содержать полученное поправочное значение. Длительность процедуры – 82 такта ACLK.

Поправочное значение можно загрузить вручную – величина записывается в поле CALIN регистра ACTLm, затем устанавливается бит CALLOAD.

Процесс внутренней калибровки также может быть запущен автоматически в процессе инициализации АЦП, если на момент переключения ADCEN в единицу бит CALEN был установлен.

Разрядность результата настраивается с помощью поля SELRES регистра ACTLm, можно осуществить выбор между 6, 8, 10, 12 бит. Выравнивается результат по младшему биту. В таблице 19.2 показаны соответствия режима разрядности и длительности преобразования.

Таблица 19.2 – Зависимость скорости преобразования одного канала от разрядности данных

Поле SELRES регистра ACTL	Разрядность данных, бит	Количество тактов сигнала ACLK на преобразование	Максимальная скорость, выборок/с
00b	6	8	$4{,}38\times10^6$
01b	8	10	$3{,}50\times10^6$
10b	10	12	$2{,}92\times10^6$
11b	12	14	$2{,}50\times10^6$

Общая процедура включения модуля АЦПm:

1) Настроить и включить тактирование АЦП частотой ACLK от 140 кГц до 35 МГц, снять сброс (регистр RCU – ADCCFG).

2) Если необходимо, активировать автоматическую внутреннюю калибровку по включению – установить бит CALEN регистра ACTLm.

3) Вывести из режима электрической изоляции, сбросив бит DISLVL.

4) Включить встроенный LDO, установив бит ADCENLDO.

5) Разрешить работу модуля АЦП с помощью установки бита ADCEN в регистре ACTLm.

6) Запустить АПП, установив бит ADCSTART.

7) Если на момент разрешения работы был установлен CALEN, то запустится процедура калибровки. Если этот бит был сброшен, то выполнится одиночное пустое преобразование.

8) По окончании инициирующих процедур установится бит ADCRDY регистра ACTLm. Модуль АЦП готов к работе.

Процедура деинициализации модуля АЦПm для уменьшения потребления:

1) Модуль АЦП должен находиться в состоянии ожидания – измерения не проводятся.

2) Сбрасывается бит ADCEN и ADCSTART регистра ACTLm.

3) Сбрасывается бит ADCENLDO регистра ACTLm.

4) Устанавливается бит DISLVL регистра ACTLm.

Правила арбитража

Когда АЦП начинает выполнять измерения по запросам, он устанавливает флаг ADCBUSY в регистре BSTAT. Флаг занятости будет сброшен лишь при полном отсутствии запросов, при последовательном выполнении запросов флаг не сбрасывается. Во время установки флага все настройки каналов АЦП сохраняются в теневых регистрах. Изменять их настройки во время работы можно, но вступят в силу они лишь при следующей установке флага ADCBUSY после сброса.

Секвенсоры могут выставлять запросы как одновременно, так и независимо друг от друга по разным событиям запуска. Для определения порядка выполнения запросов модулем АЦП реализована схема арбитража, со следующими правилами работы:

1 Если модуль АЦП был в режиме ожидания (включен и измерения не проводятся), то при поступлении запроса незамедлительно начинается его обслуживание.

2 Секвенсоры выставляют «ждущие» запросы – запрос будет активен до тех пор, пока модуль АЦП не начнет его обработку. Как только его обслуживание началось, запрос сбрасывается.

3 Как только текущий запрос секвенсора был сброшен, он сразу выставляет следующий запрос (если таковой имеется в наличии), чтобы успеть принять участие в ближайшей процедуре арбитража, и чтобы АЦП не тратил такты на переход из активного режима в режим ожидания и обратно.

4 Секвенсор, во время работы, может не выставить следующий запрос после сброса текущего, но лишь в случае ожидания отложенного события перезапуска. Во всех остальных режимах запросы выставляются неразрывно друг за другом.

5 Если несколько секвенсоров выставили одинаковые запросы, то обслуживаться они будут параллельно. Результат запроса попадет в буферы всех соответствующих секвенсоров.

6 Арбитраж происходит перед началом обработки первого запроса (если АЦП был в ожидании) и в конце каждого обрабатываемого в текущий момент.

7 Если несколько секвенсоров одновременно выставят запросы, то запросы по каналам с меньшим номером имеют приоритет выше, чем каналы с большим.

8 Каналы с установленным битом PRIORITY в регистрах CHCTLn (где n – номер канала от 0 до 47) имеют более высокий приоритет, чем те, у которых бит сброшен.

9 Если одновременно будут выставлены запросы по каналам с установленным PRIORITY, то запросы по каналам с меньшим номером имеют приоритет выше, чем каналы с большим.

10 Необходимо с осторожностью производить частый опрос более высокоприоритетных каналов – это может значительно затруднить опрос остальных каналов.

Иллюстрация работы схемы арбитража запросов приведена на рисунке 19.13.

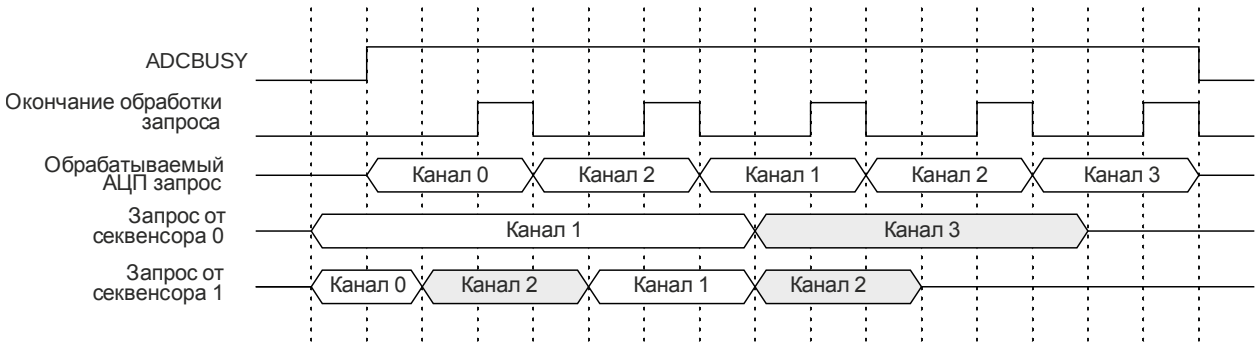


Рисунок 19.13 – Пример арбитража запросов, серым выделены запросы по каналам с установленным битом PRIORITY

Схема коррекции результатов измерений

Результат преобразования передается на схему коррекции, которая нивелирует ошибку усиления и смещения нуля, и работа которой описывается формулой:

$$D_C = \frac{D_R \times (D_{MAX} + GAINTRIM)}{D_{MAX}} + OFFTRIM, \quad (19.1)$$

где D_R – данные, полученные непосредственно с АЦП;

D_C – скорректированные данные, выдаваемые секвенсору;

D_{MAX} – максимальная величина данных в зависимости от разрядности результата;

GAINTRIM – коэффициент корректировки усиления;

OFFTRIM – коэффициент корректировки смещения нуля.

Значения GAINTRIM и OFFTRIM заносятся в одноименные поля регистров CHCTLn (где n – номер канала от 0 до 47) в дополнительном коде. По умолчанию результат измерения проходит через схему коррекции, не изменяясь, т. к. коэффициенты равны нулю.

Реализована математика «насыщения» – когда значение OFFTRIM отрицательное и больше дроби, то результат будет равен нулю, если сумма OFFTRIM и дроби больше $D_{MAX} - 1$, то результат равен $D_{MAX} - 1$.

В зависимости от разрядности результата диапазоны коэффициентов коррекции отличаются. Все значения вносятся в дополнительном коде и выравниваются по младшему биту.

Таблица 19.3 – Зависимость диапазонов коэффициентов коррекции от разрядности данных

Поле SELRES регистра ACTL	Разрядность данных, бит	Диапазон значений коэффициентов коррекции OFFTRIM, GAINTRIM
00b	6	от –4 до 3
01b	8	от –16 до 15
10b	10	от –64 до 63
11b	12	от –256 до 255

19.4 Цифровой компаратор

В состав блока АЦП входят d компараторов (где d от 0 до 23). Структурная схема компаратора показана на рисунке 19.14.

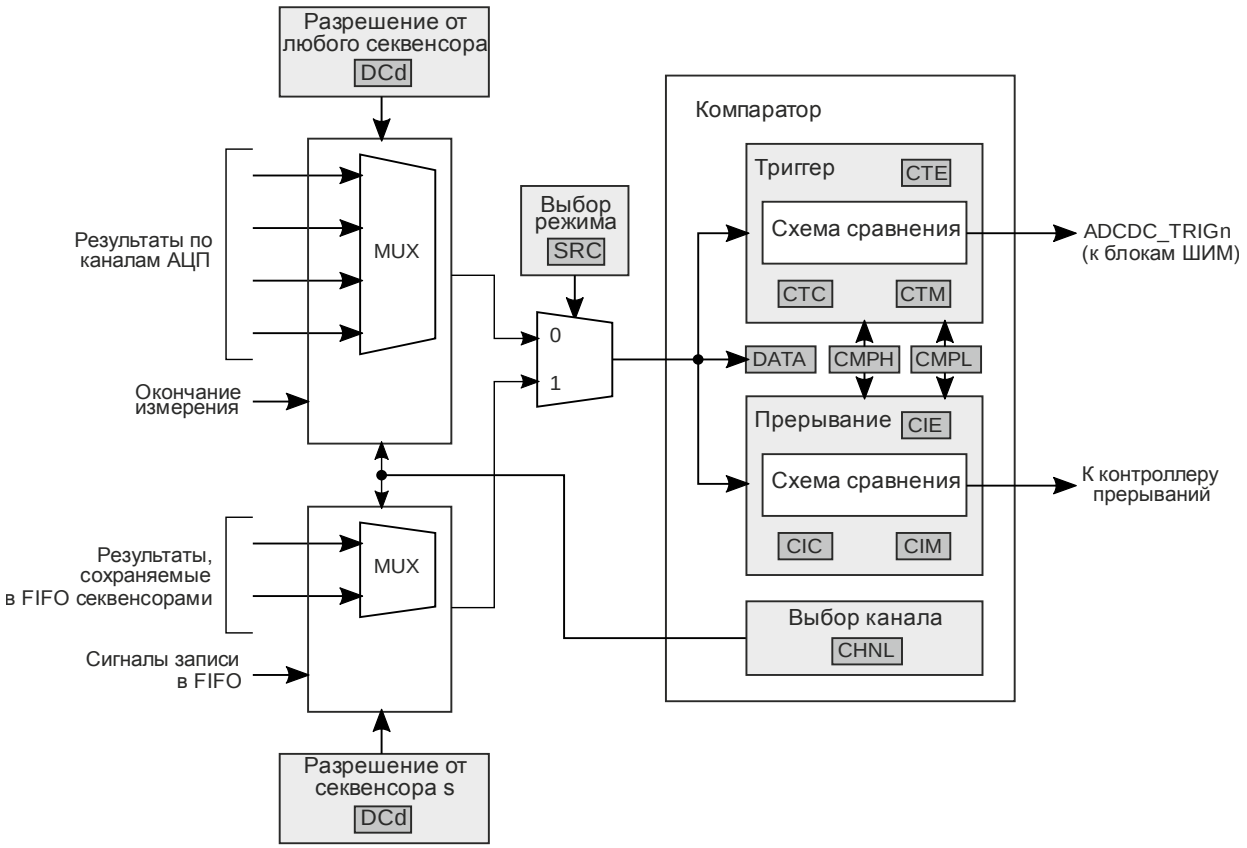


Рисунок 19.14 – Структурная схема компаратора

Все компараторы блока АЦП независимы. Каждый компаратор может обрабатывать результат измерения любого канала.

По умолчанию, когда модуль АЦП завершает обработку запроса по каналу, он выставляет результат, который захватывает как секвенсор, так и разрешенный (любым из секвенсоров) и настроенный на этот канал компаратор.

Возможен и другой режим работы, когда на компаратор будет подаваться результат, который секвенсор записывает в FIFO, но только в том случае, если канал, соответствующий сохраняемому результату, также совпадает с каналом, на который настроен компаратор. Включение этого режима осуществляется установкой бита SRC в регистре DCTL.

Правила настройки цифровых компараторов:

1) Посредством регистра SRQSEL секвенсора s выбираются каналы для измерений.

2) Посредством регистра SDC выбираются (разрешаются) компараторы для обработки полученных результатов запросов (установкой битов DCd). Запрещенные компараторы не обрабатывают полученные результаты.

3) Для каждого компаратора d в его регистре DCTL в поле CHNL указывается номер канала, результат измерения которого будет передан на него. По умолчанию значение CHNL = 0h, т. е. все компараторы настроены на работу с нулевым каналом.

4) Битом SRC в регистре DCTL выбирается источник данных для компаратора – результаты непосредственно с АЦП или результаты, записываемые секвенсором в FIFO (которые могут быть уже усреднены).

Результат измерения, полученный компаратором d, передается во внутреннюю схему сравнения и одновременно с этим сохраняется в регистре DDATA. Схема сравнения выполняет проверку соответствия результата измерения заданному условию (поля CTC, CTM регистра DCTL и поля CMPL, CMPH регистра DCMP) и в зависимости от результата проверки переключает выходной триггер и устанавливает флаг события сравнения DCEVd в регистре DCTRIG. Работа триггера разрешается установкой бита CTE регистра DCTL.

Сравнение по условию «Измерение ≤ CMPL» (CTC = 00b):

- В однократном режиме (CTM = 01b) выходной триггер переключится в единицу только в случае, если результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме (CTM = 00b) выходной триггер будет переключаться в единицу каждый раз, когда результат сравнения будет положительным.

- В однократном режиме с гистерезисом (CTM = 11b) выходной триггер переключится в единицу только в случае, если после прекращения выполнения условия «CMPH ≤ Измерение», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме с гистерезисом (CTM = 10b) выходной триггер переключится в единицу в случае, если после прекращения выполнения условия «CMPH ≤ Измерение», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным, и далее триггер будет оставаться в состоянии единицы до тех пор, пока снова не выполнится условие «CMPH ≤ Измерение».

Пример функционирования триггера показан на рисунке 19.15.

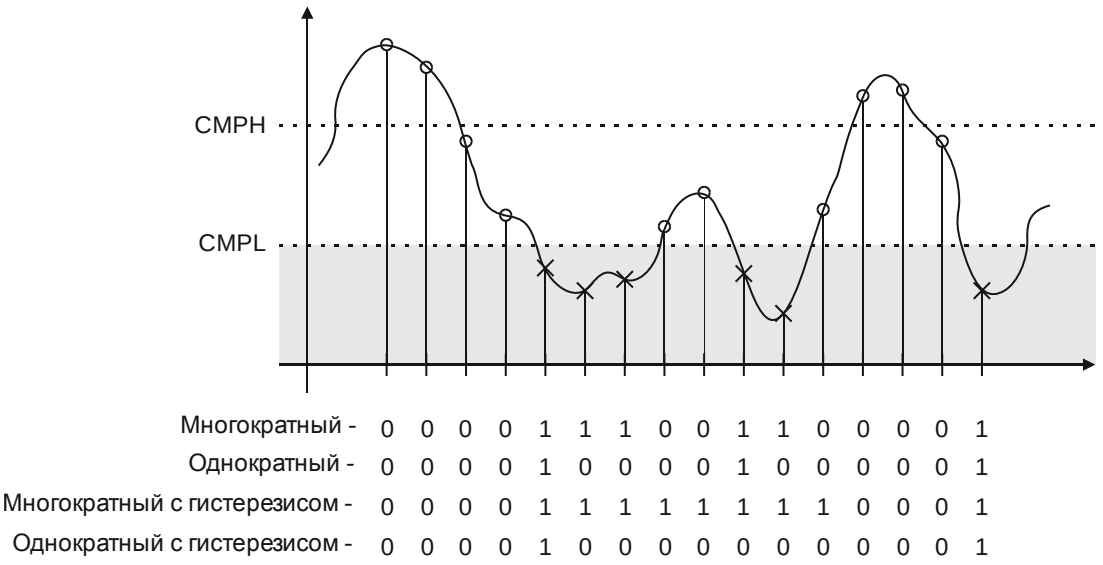


Рисунок 19.15 – Функционирование триггера при CTC = 00b

Сравнение по условию «CMPL ≤ Измерение ≤ CMPH» (CTC = 01b):

- В однократном режиме выходной триггер переключится в единицу только в случае, если результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме выходной триггер будет переключаться в единицу каждый раз, когда результат сравнения будет положительным.

- Однократный и многократный режимы с гистерезисом не поддерживаются.

Пример функционирования триггера показан на рисунке 19.16.

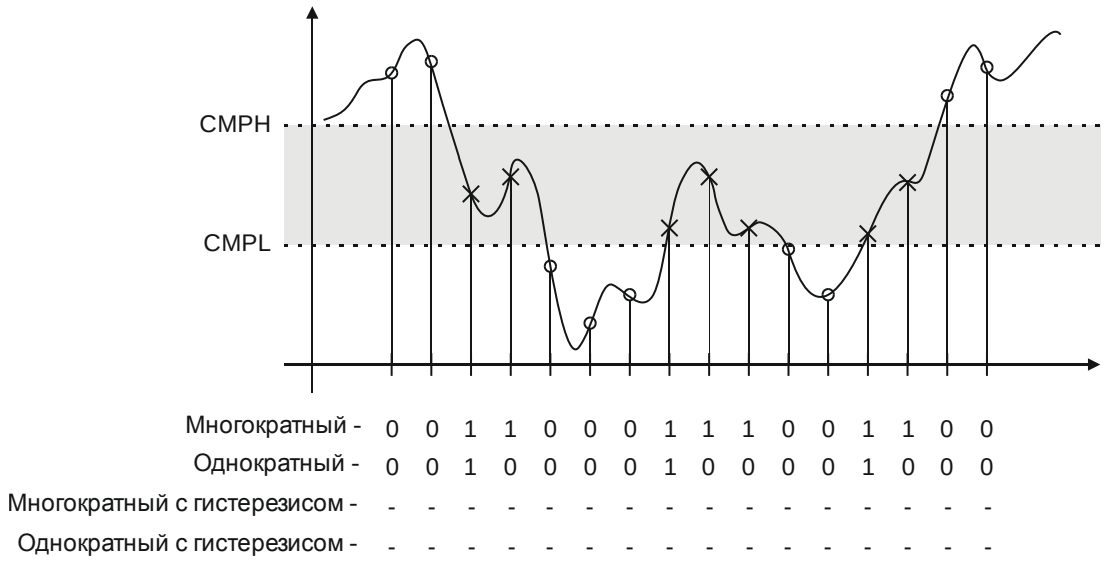


Рисунок 19.16 – Функционирование триггера при СТС = 01b

Сравнение по условию «CMPH ≤ Измерение» (СТС = 10b):

- В однократном режиме выходной триггер переключится в единицу только в случае, если результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме выходной триггер будет переключаться в единицу каждый раз, когда результат сравнения будет положительным.

- В однократном режиме с гистерезисом выходной триггер переключится в единицу только в случае, если после прекращения выполнения условия «Измерение ≤ CMPL», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным. В остальных случаях состояние триггера – ноль.

- В многократном режиме с гистерезисом (СТМ = 10b) выходной триггер переключится в единицу в случае, если после прекращения выполнения условия «Измерение ≤ CMPL», результат сравнения окажется положительным при том, что результат предыдущего сравнения был отрицательным, и далее триггер будет оставаться в состоянии единицы до тех пор, пока снова не выполнится условие «Измерение ≤ CMPL».

Пример функционирования триггера показан на рисунке 19.17.

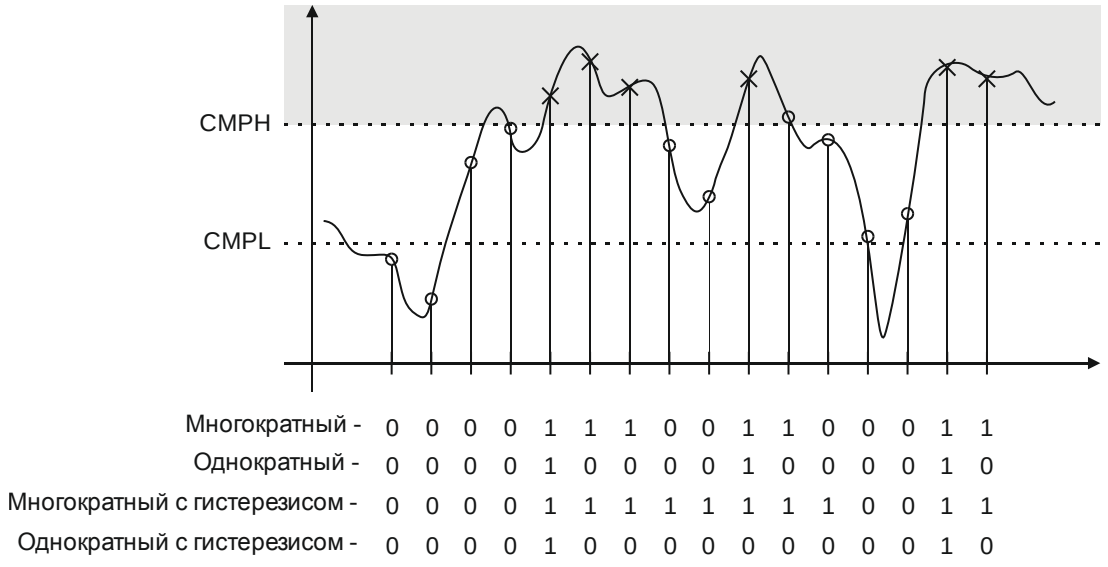


Рисунок 19.17 – Функционирование триггера при СТС = 10b

Переключение выходного триггера в единицу устанавливает соответствующий флаг TOSd в регистре DCTRIG и генерирует управляющий сигнал для пороговых выключателей блоков ШИМ. Вне зависимости от состояния флагов TOSd по каждому событию сравнения устанавливается соответствующий флаг DCEVd того же регистра. Флаги события сравнения сбрасываются записью единицы. Сброс самого триггера и его статусного флага выполняется также записью единицы в соответствующий бит TOSd регистра DCTRIG.

Независимо от состояния триггера (разрешен или запрещен) в случае положительного результата сравнения компаратор может генерировать прерывание. Для этого следует установить бит CIE регистра DCTL и задать условия CIC и CIM (аналогичны по функционалу CTC и CTM).

Примечание – Условия срабатывания выходного триггера компаратора и условия генерирования прерываний могут не совпадать.

19.5 Прерывания

При генерировании прерываний устанавливаются флаги SEQRISs и DCRISd в регистре RIS. Если были установлены маски прерываний в регистре IM (поля SEQIMs и DCIMd), то также устанавливаются соответствующие маскированные флаги прерываний SEQMISs и DCMISd. Сброс флагов (маскированных и немаскированных) осуществляется записью единицы в соответствующие поля регистра IC (поля SEQICs и DCICd).

Установка маскированных флагов SEQMISs вызывает формирование соответствующих прерываний ADC_SEQs блока АЦП.

Флаги DCMISd компараторов объединены по ИЛИ, и установка любого из них вызывает формирование прерывания ADC_DC блока АЦП.

Структурная схема контроллера прерываний показана на рисунке 19.18.

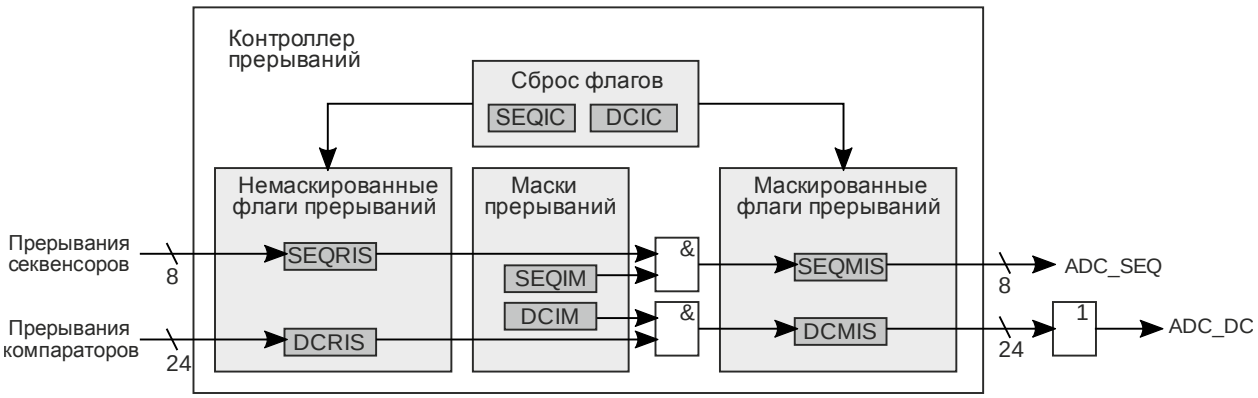


Рисунок 19.18 – Контроллер прерываний

19.6 Примеры работы блока АЦП

В этом подразделе представлены разнообразные примеры настройки блока АЦП для осуществления измерений в различных режимах. Для всех примеров подразумевается, что АЦП тактируется частотой в 33,3 МГц.

Настройка тактирования

Один из примеров настройки тактирования АЦП совместно с настройкой системного тактового сигнала представлен ниже.

1 С помощью регистра PLLCFG блока RCU настраиваем выходную частоту PLL 200 МГц. Осуществляем процедуру перевода системной частоты на источник PLL. Таким образом, частота системного тактового сигнала SYSCLK будет равна 200 МГц.

2 Настройка рабочего тактового сигнала АЦП ACLK производится с помощью регистра ADCCFG блока RCU. Выбираем в качестве источника выходную частоту PLL (поле CLKSEL = 2), включаем делитель на 6 (поле DIVN=2, бит DIVEN=1). Таким образом, f_{ACLK} = 33,3 МГц.

```
RCU->ADCCFG_bit.CLKSEL = 2;
RCU->ADCCFG_bit.DIVN = 2;
RCU->ADCCFG_bit.DIVEN = 1;
```

3 Включаем тактирование блока (бит CLKEN=1) и снимаем сброс (бит RSTDIS=1). Блок АЦП готов к дальнейшим конфигурациям.

```
RCU->ADCCFG_bit.CLKEN = 1;
RCU->ADCCFG_bit.RSTDIS = 1;
```

Пример 1 – Программный запуск одного секвенсора

Требуемый режим работы:
- программный запуск;
- секвенсор 0
- однократное измерение каналов 0-3;
- без прерываний (опрос флагов).

Код, соответствующий настройке и запуску необходимого режима, представлен ниже.

```
// Настройка модуля АЦП – 12 бит и калибровка при включении
ADC -> ACTL[0].ACTL_bit.SELRES = 3;
ADC -> ACTL[0].ACTL_bit.CALEN = 1;
ADC -> ACTL[0].ACTL_bit.DISLVL = 0;
ADC -> ACTL[0].ACTL_bit.ADCENLDO = 1;
ADC -> ACTL[0].ACTL_bit.ADCEN = 1;
delay(2000); //Задержка не менее 20 мкс
ADC -> ACTL[0].ACTL_bit.ADCSTART = 1;
//Настройка секвенсора 0
ADC ->EMUX_bit.EM0 = 0;
ADC ->SEQ[0].SRQCTL_bit.RQMAX = 0x3;
ADC ->SEQ[0].SRQSEL0_bit.RQ0 = 0x0;
ADC ->SEQ[0].SRQSEL0_bit.RQ1 = 0x1;
ADC ->SEQ[0].SRQSEL0_bit.RQ2 = 0x2;
ADC ->SEQ[0].SRQSEL0_bit.RQ3 = 0x3;
ADC ->SEQEN_bit.SEQEN0 = 1;
// Запуск
while(!ADC->ACTL[0].ACTL_bit.ADCRDY);
ADC ->SEQSYNC_bit.SYNC0 = 1;
ADC ->SEQSYNC_bit.GSYNC = 1;
```

1 Перед разрешением работы АЦП устанавливаем 12-битный режим измерений (SELRES=3) и включаем механизм калибровки (CALEN). Затем разрешаем работу модуля АЦП – необходимо установить бит ADCEN в регистре ACTL.

2 Для работы выбран секвенсор 0. Настроим его источник запуска – поле EM0 регистра EMUX должно быть равно 0, т. к. запуск программный.

3 В поле RQMAX регистра SRQCTL необходимо внести значение 3h, т. к. измерения будут проводиться по четырем каналам.

4 Настраиваем каналы для запросов. Допустим, необходимо опросить каналы последовательно от нулевого к третьему, значит, в регистр SRQSEL0 необходимо внести значения в поля: RQ0=0h, RQ1=1h, RQ2=2h, RQ3=3h.

5 Разрешаем работу секвенсора. Для этого необходимо установить бит SEQEN0 в регистре SEQEN.

6 Запускаем измерения. Перед запуском проверяем флаг ADCRDY, чтобы быть уверенными в том, что модуль АЦП провел необходимые инициализации. Для того чтобы начать измерения, необходимо установить бит SYNC0 в регистре SEQSYNC и записать в бит GSYNC единицу.

7 Проводим опрос флагов, чтобы установить окончание измерений. Например, можно опрашивать регистр SFLOAD, ожидая, пока он не станет равен 4h.

8 Считываем четыре результата измерения из буфера SFIFO.

Работа режима проиллюстрирована на рисунке 19.19.

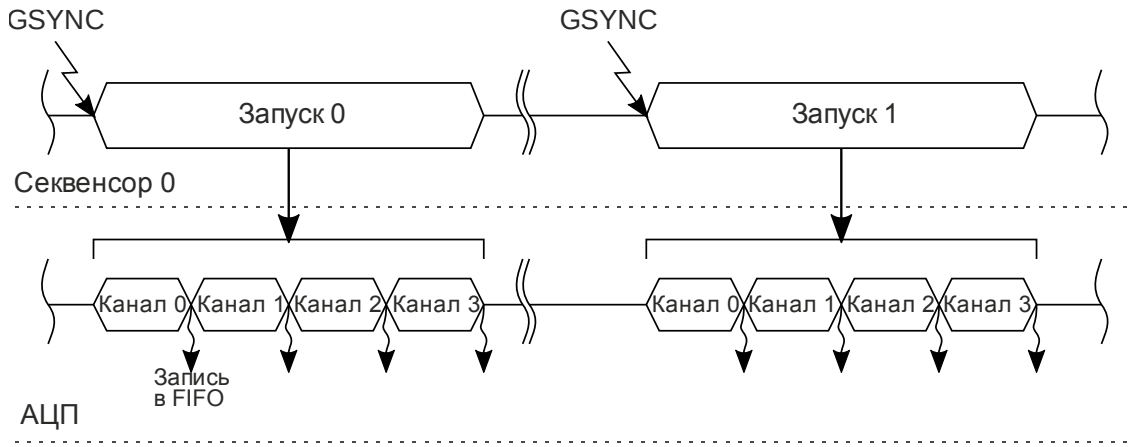


Рисунок 19.19 – Диаграммы программного запуска одного секвенсора

Пример 2 – Циклический опрос канала с задержками

Требуемый режим работы:

- циклическая работа;
- секвенсор 0
- опрос канала номер 2 каждую 1мс;
- коррекция канала 2;
- каждый опрос из 16 последовательных измерений и усреднения;
- прерывание по каждой записи в FIFO.

Код, соответствующий настройке и запуску необходимого режима, представлен ниже.

```
//Настройка модуля АЦП – 12 бит и калибровка при включении
ADC->ACTL[0].ACTL_bit.SELRES = 3;
ADC->ACTL[0].ACTL_bit.CALEN = 1;
ADC->ACTL[0].ACTL_bit.DISLVL = 0;
ADC->ACTL[0].ACTL_bit.ADCENLDO = 1;
ADC->ACTL[0].ACTL_bit.ADCEN = 1;
delay(2000); //Задержка не менее 20 мкс
ADC->ACTL[0].ACTL_bit.ADCSTART = 1;
```

```
//Коррекция канала 2
ADC->CHCTL[2].CHCTL_bit.GAINTRIM = 5; // значения для примера
ADC->CHCTL[2].CHCTL_bit.OFFTRIM = (uint32_t)(-5);
//Настройка секвенсора 0
ADC->EMUX_bit.EM0 = 0xF;
ADC->SEQ[0].SCCTL_bit.ICNT = 0;
ADC->SEQ[0].SRTMR = 33332;
ADC->SEQ[0].SRQCTL_bit.RQMAX = 0;
ADC->SEQ[0].SRQCTL_bit.QAVGVAL = 4;
ADC->SEQ[0].SRQCTL_bit.QAVGEN = 1;
ADC->SEQ[0].SRQSEL0_bit.RQ0 = 2;
ADC->SEQEN_bit.SEQEN0 = 1;
// PLIC прерывание
ADC->IM_bit.SEQIM0 = 1;
PLIC_SetIrqHandler (Plic_Mach_Target, IsrVect_IRQ_ADC,
                    ADC_IRQHandler);
PLIC_SetPriority(IsrVect_IRQ_ADC, 0x1);
PLIC_IntEnable(Plic_Mach_Target, IsrVect_IRQ_ADC);
// Запуск
while(!ADC->ACTL[0].ACTL_bit.ADCRDY);
ADC->SEQSYNC_bit.SYNC0 = 1;
ADC->SEQSYNC_bit.GSYNC = 1;
```

1 Перед разрешением работы АЦП устанавливаем 12-битный режим измерений (SELRES=3) и включаем механизм калибровки (CALEN). Затем разрешаем работу модуля АЦП – необходимо установить бит ADCEN в регистре ACTL.

2 Если предварительно была произведена процедура коррекции канала 2 и были вычислены поправочные коэффициенты, их необходимо внести в поля GAINTRIM и OFFTRIM регистра CHCTL[2].

3 Для работы выберем секвенсор 0. Настроим его источник запуска – поле EM0 регистра EMUX должно быть равно Fh, т. к. запуск циклический.

4 В поле RQMAX регистра SRQCTL необходимо внести значение 0h, т. к. измерения будут проводиться по одному каналу.

5 Настраиваем канал для запроса. В регистр SRQSEL0 необходимо внести значение RQ0=2h.

6 Включаем усреднение сканированием по 16 опросам очереди. В регистре SEQ[0].SRQCTL нужно установить поля QAVGVAL=4, QAVGEN=1.

7 Для того чтобы опрос проводился каждую 1 мс (при f_{ACLK} = 33,3 МГц), необходимо внести в регистр SEQ[0].SRTMR значение (1000000 нс/30 нс) – 1 = 33332.

8 Разрешаем генерацию прерываний по каждой записи в FIFO – нужно установить бит SEQIM0 в регистре IM и проследить, чтобы поле ICNT регистра SCCTL оставалось равным нулю.

9 Разрешаем работу секвенсора. Для этого необходимо установит бит SEQEN0 в регистре SEQEN.

10 Запускаем измерения. Перед запуском проверяем флаг ADCRDY, чтобы быть уверенными в том, что модуль АЦП провел необходимые инициализации. Для того чтобы начать измерения, необходимо установить бит SYNC0 в регистре SEQSYNC и записать в бит GSYNC единицу.

11 Измерения будут сразу же запущены и будут запускаться каждую 1 мс далее. После каждого запуска будет проведено 16 измерений, результат усреднения которых будет записан в FIFO, и будет вызвано прерывание.

Работа режима проиллюстрирована на рисунке 19.20.

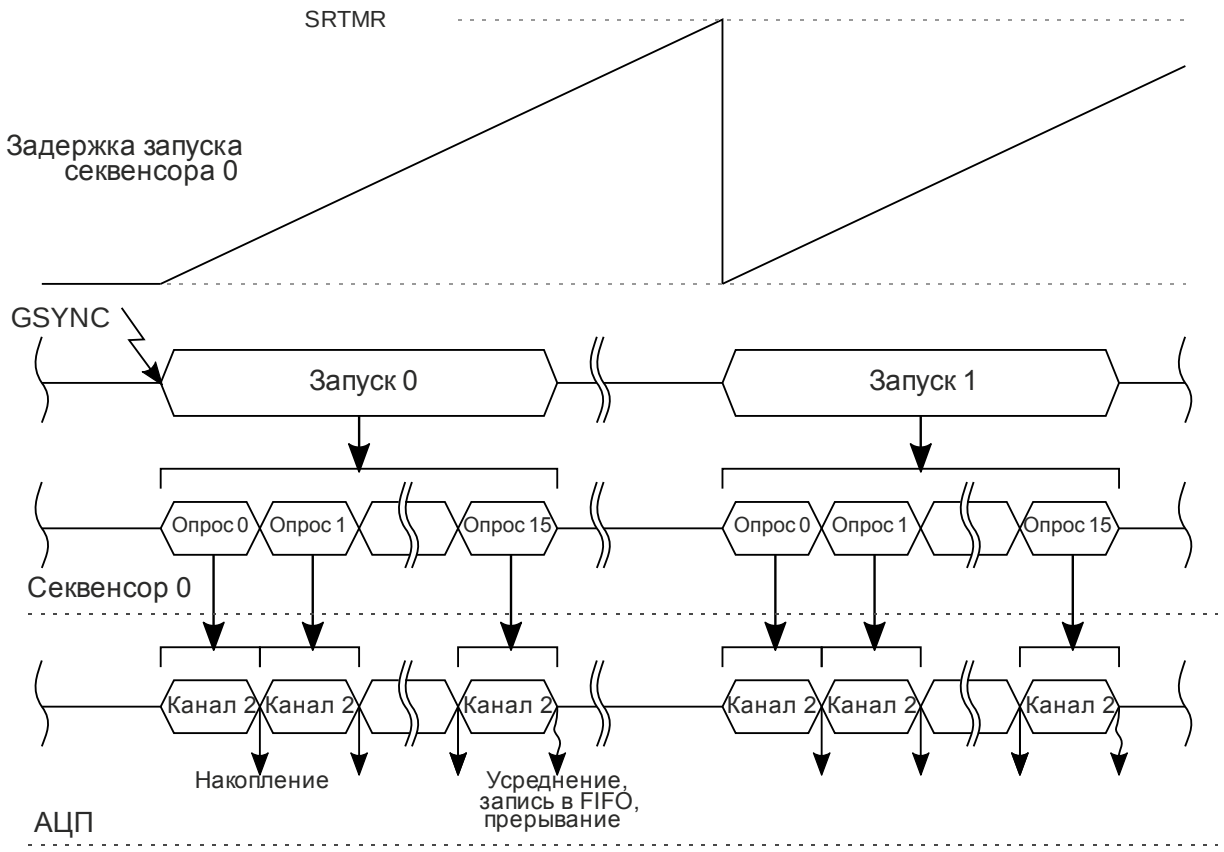


Рисунок 19.20 – Диаграммы циклического опроса канала

Пример 3 – Равномерно распределенные измерения по периоду таймера

Требуемый режим работы:

- запуск по таймеру 0 с частотой 10 кГц;
- четыре точки измерений, равномерно распределенные по периоду;
- секвенсор 0;
- опросы по каналам 3 и 0;
- каждый опрос из четырех последовательных измерений и усреднения;
- прерывание каждые восемь записей в FIFO (по окончанию измерений в текущем периоде таймера).

Код, соответствующий настройке и запуску необходимого режима, представлен ниже.

```
// Настройка таймера
RCU->CGCFGAPB1_bit.TMR0EN = 1;
RCU->RSTDISAPB1_bit.TMR0EN = 1;
TMR0->PERIOD = 19999;
TMR0->EXTEVT_IM_bit. TMR = 1;
//Настройка модуля АЦП – 12 бит и калибровка при включении
ADC->ACTL[0].ACTL_bit.SELRES = 3;
ADC->ACTL[0].ACTL_bit.CALEN = 1;
ADC->ACTL[0].ACTL_bit.DISLVL = 0;
ADC->ACTL[0].ACTL_bit.ADCENLDO = 1;
ADC->ACTL[0].ACTL_bit.ADCEN = 1;
delay(2000); //Задержка не менее 20 мкс
ADC->ACTL[0].ACTL_bit.ADCSTART = 1;
//Настройка секвенсора
ADC->EMUX_bit.EM0 = 5;
ADC->SEQ[0].SCCTL_bit.ICNT = 7;
```

```
ADC->SEQ[0].SCCTL_bit.RCNT = 3;
ADC->SEQ[0].SRTMR = 832;
ADC->SEQ[0].SRQCTL_bit.RQMAX = 1;
ADC->SEQ[0].SRQCTL_bit.QAVGVAL = 2;
ADC->SEQ[0].SRQCTL_bit.QAVGEN = 1;
ADC->SEQ[0].SRQSEL0_bit.RQ0 = 3;
ADC->SEQ[0].SRQSEL0_bit.RQ1 = 0;
ADC->SEQEN_bit.SEQEN0 = 1;
// PLIC прерывание
ADC->IM_bit.SEQIM0 = 1;
PLIC_SetIrqHandler (Plic_Mach_Target, IsrVect_IRQ_ADC,
                    ADC_IRQHandler);
PLIC_SetPriority(IsrVect_IRQ_ADC, 0x1);
PLIC_IntEnable(Plic_Mach_Target, IsrVect_IRQ_ADC);
// Запуск
while(!ADC->ACTL[0].ACTL_bit.ADCRDY);
SIU->TMREN_bit.TMR0EN = 1;
TMR0->CTRL_bit. MODE = 1;
```

1 Включаем тактирование таймера 0 и снимаем сброс – установим бит TMR0EN в регистрах CGCFGAPB1 и RSTDISAPB1 соответственно.

2 Для того, чтобы таймер достигал периода с частотой 10 кГц (при f_{PCLK}=200 МГц), необходимо внести в регистр таймера PERIOD значение (200000 кГц/10 кГц) – 1 = 19999.

3 Разрешаем генерацию таймером запросов на старт преобразования, установив бит TMR в регистре EXTEVT_IM таймера.

4 Перед разрешением работы АЦП устанавливаем 12-битный режим измерений (SELRES=3) и включаем механизм калибровки (CALEN). Затем разрешаем работу модуля АЦП – необходимо установить бит ADCEN в регистре ACTL.

5 Для работы выберем секвенсор 0. Настроим его источник запуска – поле EM0 регистра EMUX должно быть равно 5h, т. к. запуск по достижению периода таймером 0.

6 Необходимое количество записей в FIFO для генерации прерывания – 8h, поэтому запишем в поле ICNT регистра SEQ0_CCTL значение 8 – 1=7h.

7 По каждому запуску должно совершиться три перезапуска с паузой в четверть периода таймера 0. В поле количества перезапусков RCNT регистра SCCTL внесем 3h.

В регистр задержки перезапуска SRTMR внесем значение 33333 кГц/(10 кГц×4) – 1 = 832.

8 В поле RQMAX регистра SRQCTL необходимо внести значение 1h, т. к. измерения будут проводиться по двум каналам.

9 Настраиваем каналы для запроса. В регистр SRQSEL необходимо внести значения RQ0=3h, RQ1=0h.

10 Включаем усреднение сканированием по четырем опросам очереди. В регистре SRQCTL нужно установить поля QAVGVAL=2, QAVGEN=1.

11 Разрешаем генерацию прерываний после каждой восьмой записи в FIFO – нужно установить бит SEQIM0 в регистре IM.

12 Разрешаем работу секвенсора. Для этого необходимо установить бит SEQEN0 в регистре SEQEN.

13 Запускаем измерения. Перед запуском проверяем флаг ADCRDY, чтобы быть уверенными в том, что модуль АЦП провел необходимые инициализации. Для того чтобы их начать, необходимо включить таймер 0, установив бит TMR0EN в регистре TMREN блока SIU, а также переключить режим счета записью 1 в поле MODE регистра CTRL TMR0.

14 Измерения будут запускаться по каждому достижению периода таймера 0. После каждого запуска будет проведено по три отложенных перезапуска. Т. к. после каждого перезапуска в FIFO будет попадать по 2 усредненных результата, то прерывание будет

сгенерировано после записи последнего результата в последнем третьем перезапуске.

Работа режима проиллюстрирована на рисунке 19.21.

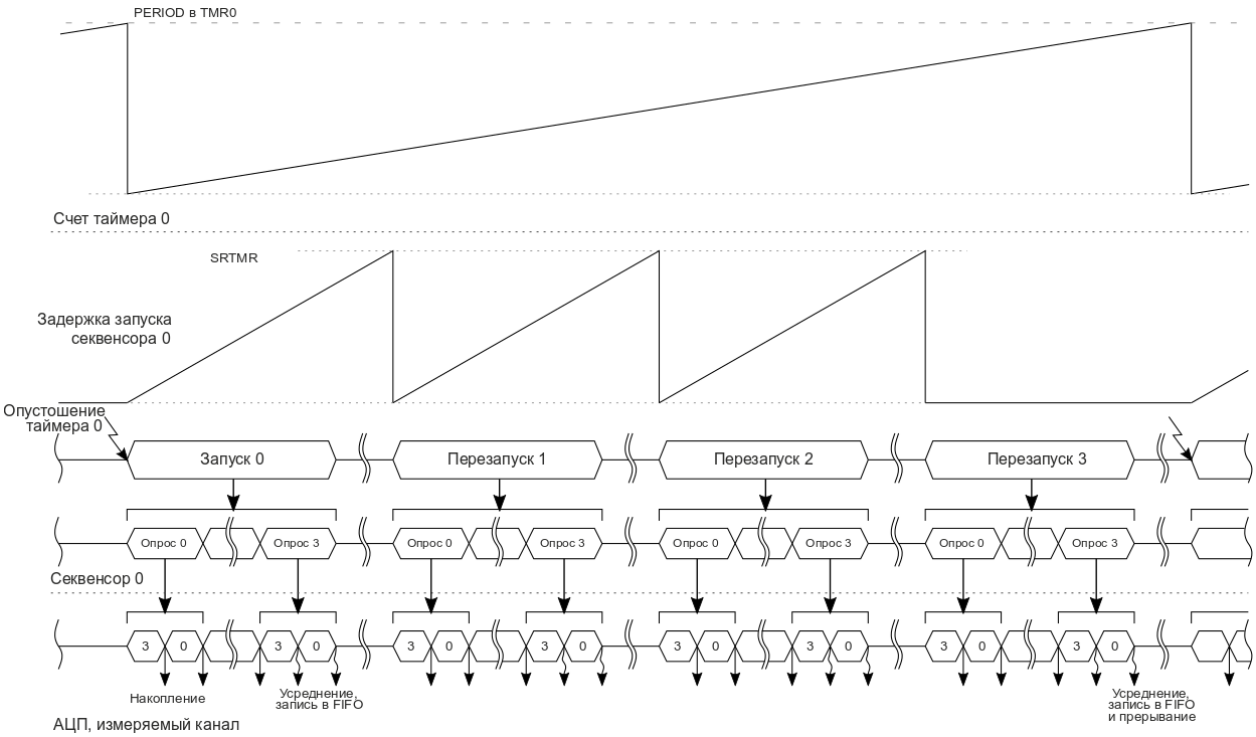


Рисунок 19.21 – Диаграммы равномерно распределенных измерений по периоду таймера

20 Блок ЦАП (DAC)

Микроконтроллер содержит два идентичных блока ЦАП.

Физический блок ЦАП (DAC) представляет собой 12-битный цифро-аналоговый преобразователь структурная схема которого изображена на рисунке 20.1.

Управляющие регистры блока ЦАП приведены в приложении А.14.

Основные электрические параметры блока ЦАП указаны в таблице 20.1.

Таблица 20.1 – Электрические параметры ЦАП

Наименование параметра, единица измерения (режим измерения)		Норма		
		не менее	номинал	не более
Количество каналов, шт			1	
Разрядность, бит			12	
Источники опорного напряжения			внутр./ AVCC	
Диапазон опорных напряжений, В		1,8		AVCC
Диапазон выходных напряжений, В (BUFFER ON)		0,2		U_{AVCC}-0,2
Диапазон выходных напряжений, В (BUFFER OFF)		0		U_{REF+}
Максимальная частота обновления выходных данных, MSPS CL=50пФ RL=5кОм (BUFFER ON) full-scale		0,5		
Параметры нагрузки, кОм режим BUFFER ON	Нагрузка подключена к AGND	5		
	Нагрузка подключена к AVCC	25		
Параметры нагрузки, пФ режим BUFFER OFF				50
Интегральная нелинейность преобразования (INL), LSB		-4		4
Дифференциальная нелинейность преобразования (DNL), LSB		-2		2
Погрешность смещения нуля, LSB VREF+=1,8 режим BUFFER ON		-30		30
Погрешность смещения нуля, LSB VREF+=1,8 режим BUFFER OFF		-8		8
Погрешность коэффициента усиления, %		-1		1
Ток потребления I_{DDA}, мкА Normal (BUFFER ON) AVCC, Normal (BUFFER OFF) AVCC			490 20	
Ток потребления I_{DDV}, мкА Normal (BUFFER ON) VREF+, Normal (BUFFER OFF)VREF+			170 160	

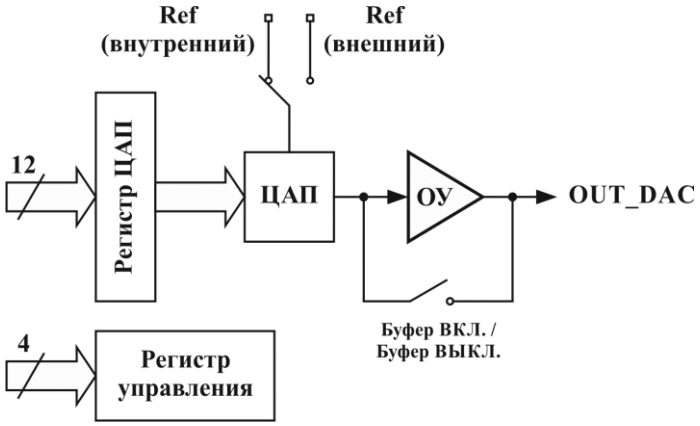


Рисунок 20.1 – Структурная схема физического блока ЦАП

Блок ЦАП имеет выбор из двух источников опорного напряжения:
– внешний (разделяется с другими периферийными устройствами, AVCC)
– внутренний от встроенного источника опорного напряжения (свой для каждого блока ЦАП).

Каждый физический блок ЦАП имеет собственный блок управления, выходной канал и контроллер. В зависимости от режима работы, преобразования могут быть выполнены независимо или одновременно в случае, когда оба канала объединяются для синхронной операции обновления входных данных.

Настройка тактирования и сброса блока управления ЦАП осуществляется с помощью регистров CGCFGAPB1/RSTDISAPB1 блока управления тактовыми сигналами RCU (поля DAC0EN/DAC1EN). Запись/чтение управляющих и статусных регистров осуществляется на частоте PCLK.

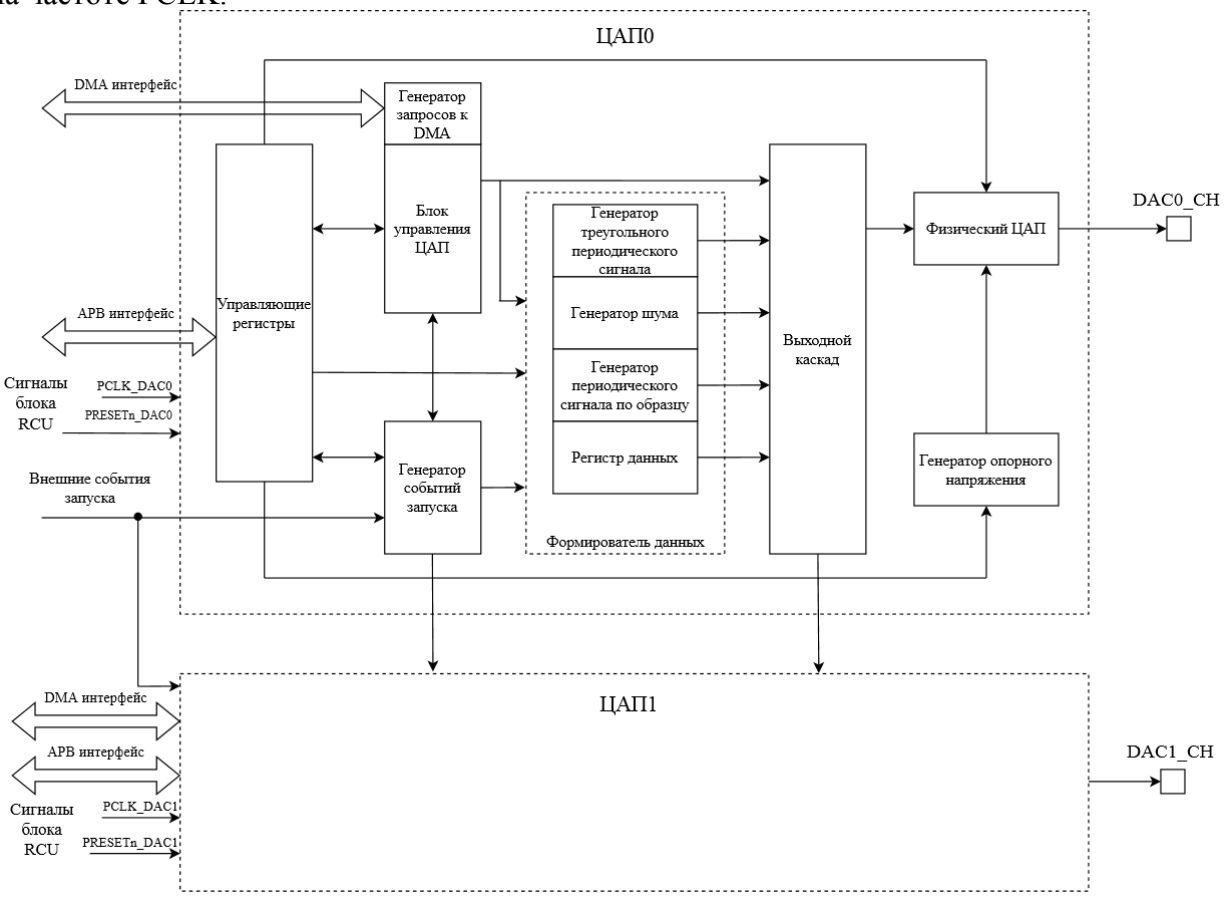


Рисунок 20.2 – Структурная схема блоков управления ЦАП

20.1 Генератор событий запуска

Генератор событий запуска используется для формирования запросов на старт преобразования ЦАП. Он состоит из одного мультиплексора и счетчика. Мультиплексор в зависимости от значения поля TSEL регистра CONTROL выбирает одно из возможных событий: внутреннее программное, внутреннее аппаратное или внешнее. Структурная схема генератора событий запуска приведена на рисунке 20.3.

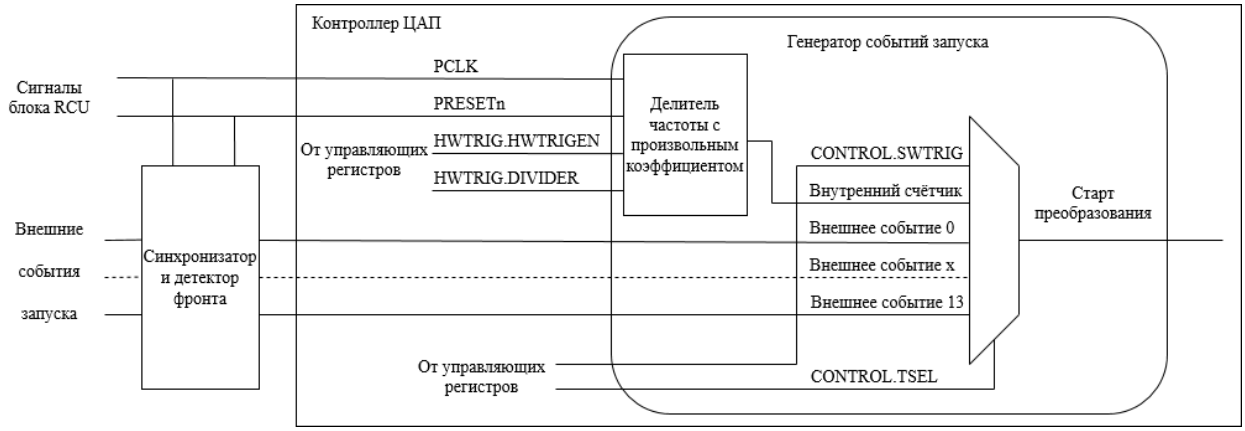


Рисунок 20.3 – Структурная схема генератора событий запуска

Внутреннее программное событие запуска формируется установкой бита SWTRIG в одноимённом регистре. Бит сбрасывается автоматически сразу после установки.

Внутреннее аппаратное событие запуска формируется счётчиком, который управляется с помощью регистра HWTRIG. На вход счетчика поступает частота PCLK, которая затем проходит через делитель. Период счёта (частота преобразования) задаётся с помощью поля DIVIDER регистра HWTRIG. Запуск счетчика осуществляется установкой бита ENABLE в том же регистре.

Внешние события синхронизированы с частотой PCLK вне блока ЦАП и срабатывают по переднему фронту сигнала PCLK.

20.2 Запросы от периферийных блоков к ЦАП

Блок ЦАП позволяет выбрать источник запроса для запуска преобразования. В поле TSEL регистра CONTROL устанавливается номер запроса для выбора события запуска (см. таблицу 20.2).

Таблица 20.2 – Список событий запуска ЦАП

Номер запроса	Модули/режимы	Регистр разрешения старта преобразования ЦАП
0	Программный запрос	DACx->SWTRIG
1	Внутренний аппаратный запрос	DACx->HWTRIG
2	TMR0 | TMR1 | TMR2 | TMR3	TMRx->EXTEVT_IM
3	TMR4 | TMR5 | TMR6 | TMR7	TMRx->EXTEVT_IM
4	TMR8 | TMR9 | TMR10 | TMR11	TMRx->EXTEVT_IM
5	TMR12 | TMR13 | TMR14 | TMR15	TMRx->EXTEVT_IM
6	GPIOA | GPIOB	GPIOx->ADCSOCSET/CLR
7	GPIOC | GPIOD	GPIOx->ADCSOCSET/CLR
8	GPIOE | GPIOF	GPIOx->ADCSOCSET/CLR
9	GPIOG	GPIOx->ADCSOCSET/CLR
10	PWM_A_0 | PWM_A_1 | … | PWM_A_15	PWMx-> ETSEL_bit.SOCAEN
11	PWM_B_0 | PWM_B_1 | … | PWM_B_15	PWMx-> ETSEL_bit.SOCBEN
12 - 15	Зарезервировано	–

20.3 Запросы DMA

Каждый блок ЦАП может генерировать запросы к контроллеру прямого доступа к памяти DMA. Когда контроллер ЦАП настроен на работу по любым событиям запуска, кроме программного (поле TSEL регистра CONTROL не равно нулю) и в регистре DMACTRL установлен бит ENABLE, при возникновении этого события происходит генерация запроса к модулю DMA на пересылку данных. Если в регистре DMACTRL дополнительно установлен бит TXSTOP, тогда при окончании загрузки всех данных, генерация запросов прекратится и бит ENABLE сбросится автоматически.

20.4 Формат данных ЦАП

Поддерживается беззнаковый и знаковый формат входных данных. Выбор формата осуществляется установкой бита SIGN в регистре CONTROL.

В знаковом режиме двоичные целые числа представляются с использованием смещённого кода. Таким образом с помощью двенадцати разрядов можно представить числа в диапазоне от -2048 до +2047. При вводе чисел в ЦАП этот диапазон аппаратно сдвигается до 0...4095 путем прибавления 2048. Числа, которые больше 2048, при этом считаются положительными, а числа, которые меньше 2048 – отрицательными. Среднее число 2048 – соответствует нулю. Соответствие рассмотренных кодов иллюстрируется в таблице 20.3.

Таблица 20.3 – Примеры соответствия десятичного числа смещённому коду

Десятичное число	Смещённый код
2047	1111_1111_1111
1	1000_0000_0001
0	1000_0000_0000
-1	0111_1111_1111
-2047	0000_0000_0001
-2048	0000_0000_0000

Для использования шкалы ЦАП в знаковом формате (в поле SIGN регистра CONTROL установлено значение «1»), данные перед загрузкой следует объявить как int16_t или int8_t и загрузить значение в один из регистров (см. далее). Корректировка с использованием смещённого кода (сложение с 2048) произойдёт автоматически.

В беззнаковом режиме (в поле SIGN регистра CONTROL установлено значение «0»), никаких предварительных преобразований делать не нужно, числа представлены от 0 до 4095 и идентичны входному типу uint16_t.

Данные загружаются для дальнейшего преобразования в один из двух возможных регистров доступных пользователю:

1) Регистр DATA8 – биты [7:0], данные из которого также попадают во внутренний регистр преобразователя DATAH, биты [11:4] (биты [3:0] обнуляются);

2) Регистр DATA12 – биты [11:0], данные из которого также попадают во внутренний регистр преобразователя DATAH, биты [11:0].

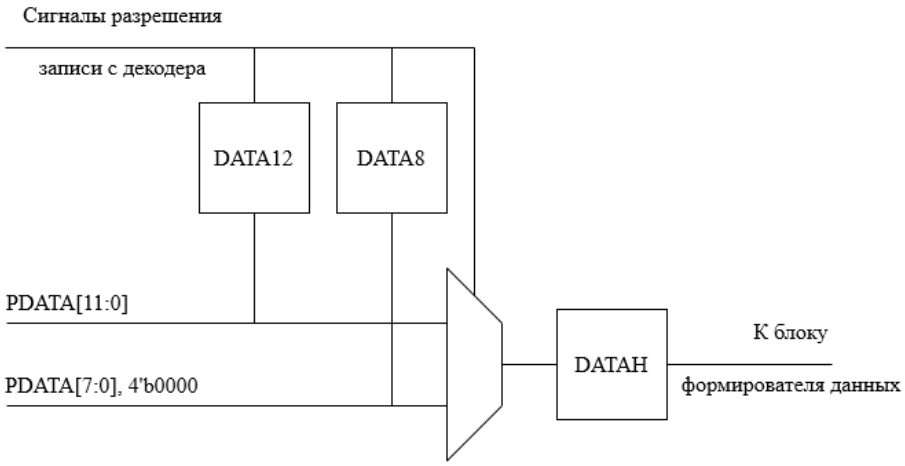


Рисунок 20.4 – Внутреннее устройство набора регистров данных контроллера ЦАП

Для работы в сдвоенном режиме (см.20.6 «Режимы работы») в наборе регистров данных контроллера ЦАП0 присутствует третий регистр DATA_DUAL – биты [11:0] которого попадают во внутренний регистр преобразователя DATAH контроллера ЦАП0, и, биты [27:16] которого попадают во внутренний регистр преобразователя DATAH контроллера ЦАП1. В сдвоенном режиме работы запись в регистры DATA12 и DATA8 обоих контроллеров запрещена и не влияет на результат преобразования.

20.5 Устройство выходного каскада к физическому блоку ЦАП

Выходной каскад к физическому блоку ЦАП состоит из мультиплексора, сумматора, устройства сдвига и выходного регистра. Его схематическое представление приведено на рисунке 20.5.

Мультиплексор выбирает один из нескольких вариантов данных, подаваемых на вход физического ЦАП в зависимости от настройки полей WAVE и DATAMODE регистра CONTROL.

Сумматор складывает данные с содержимым поля OFFSET регистра CONTROL. В основном, это требуется для режима генератора сигнала по образцу (поле WAVE регистра CONTROL имеет значение «3»).

Устройство сдвига позволяет выполнять умножение или деление, в зависимости от установки бита MULDIV регистра CONTROL (1 – умножение, 0 – деление), значения после сумматора на степень 2 (выбирается полем SCALE регистра CONTROL). В дополнение к этому, выходное значение может быть сконвертировано в дополнительный код (поле NEGATE регистра CONTROL), а также возможно превратить беззнаковое число в знаковое (поле SIGN регистра CONTROL). Все эти операции возможны во всех режимах работы.

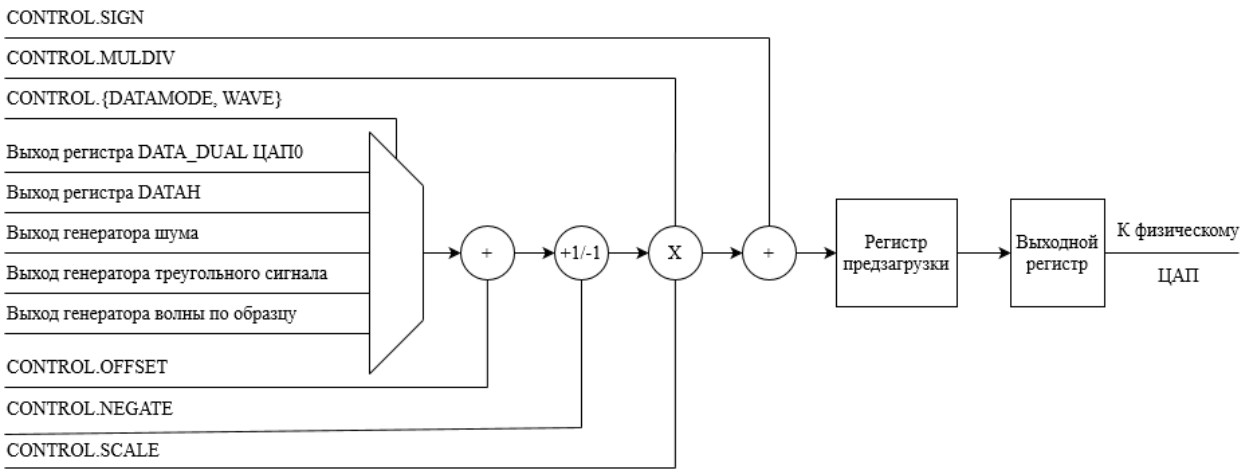


Рисунок 20.5 – Схематичное представление выходного каскада контроллера ЦАП

Примечание – При сдвоенном режиме работы и наличии в системе двух блоков ЦАП значение с выхода каскада ЦАП0 после выхода с внутреннего регистра предзагрузки ЦАП0 может поступать на вход мультиплексора в ЦАП1.

20.6 Режимы работы

Загрузка в регистр DATA

Загрузка данных для преобразования посредством записи в один из регистров DATA8 или DATA12 (single value mode). Не требует предварительной настройки регистров PATDORL/PATDORH и поля WAVE регистра CONTROL (значение 0). По умолчанию, данные считаются беззнаковыми.

Режим генерации шума (noise wave mode)

При установленном значении «1» в поле WAVE регистра CONTROL (DAC_CONTROL_WAVE_NOISE) включается генератор псевдослучайного шума, основанный на регистре сдвига с обратной связью (LFSR). Алгоритм формирования псевдослучайного шума представлен на рисунке 20.6.

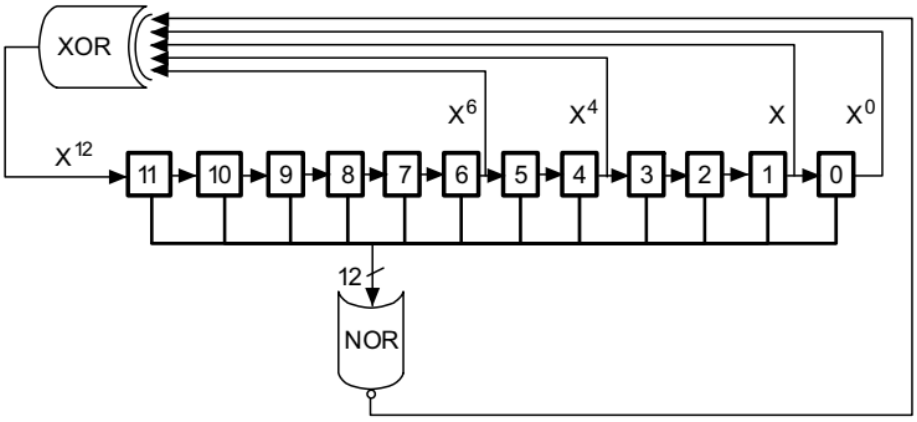


Рисунок 20.6 – Алгоритм формирования псевдослучайного шума

Предустановленное значение в этом регистре равно 0xAAA. Он не доступен пользователю для чтения/записи, а обновляется каждый раз по событию запуска с применением специального алгоритма. Значение в регистре LFSR может быть замаскировано частично или полностью записью в поле MASK регистра CONTROL, после этого происходит сложение с содержимым регистра DATAH (см. подраздел 20.4 «Формат

данных ЦАП») без переполнения и отправляется на блок ЦАП для преобразования. Если LFSR при обновлении данных получился равным нулю, «1» встраивается в него автоматически (защита от залипания). По умолчанию, данные считаются беззнаковыми. Влияние поля MASK регистра CONTROL на регистр сдвига с обратной связью LFSR показано в таблице 20.4.

Таблица 20.4 – Значения поля MASK регистра CONTROL и его влияние на LFSR

Значение в поле MASK регистра CONTROL	Поведение
0000	Бит 0 LFSR размаскирован
0001	Биты [1:0] LFSR размаскированы
0010	Биты [2:0] LFSR размаскированы
0011	Биты [3:0] LFSR размаскированы
0100	Биты [4:0] LFSR размаскированы
0101	Биты [5:0] LFSR размаскированы
0110	Биты [6:0] LFSR размаскированы
0111	Биты [7:0] LFSR размаскированы
1000	Биты [8:0] LFSR размаскированы
1001	Биты [9:0] LFSR размаскированы
1010	Биты [10:0] LFSR размаскированы
>1011	Биты [11:0] LFSR размаскированы

Режим генерации треугольной волны (triangle wave mode)

При установленном значении «2» в поле WAVE регистра CONTROL (DAC_CONTROL_WAVE_TRIANGULAR) включается генератор треугольной волны. Амплитуда волны определяется полем MASK регистра CONTROL. Пример формирования треугольного сигнала приведен на рисунке 20.7.

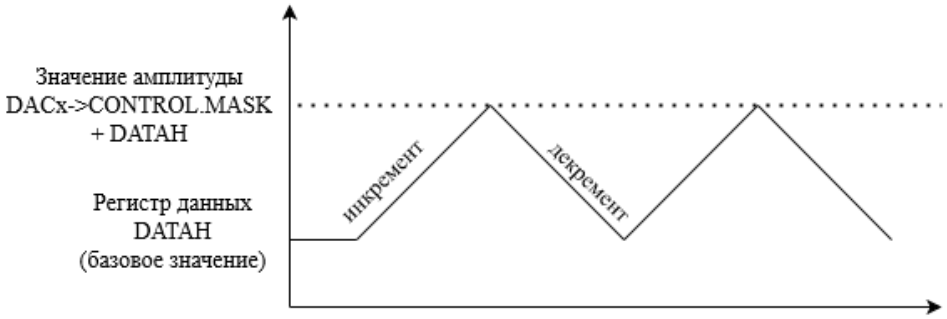


Рисунок 20.7 – Пример формирования треугольного сигнала

Внутренний счетчик инкрементируется каждый раз, когда происходит событие запуска. Значение этого счётчика затем складывается со значением в регистре DATAH (см. подраздел 20.4 «Формат данных ЦАП») без переполнения и результат отправляется на блок ЦАП для преобразования. Счётчик инкрементируется до тех пор, пока не достигнет максимального значения амплитуды, указанной в поле MASK регистра CONTROL. При достижении этого значения счётчик начинает считать вниз, до нуля, а затем вверх и так далее. По умолчанию, данные считаются беззнаковыми. Влияние поля MASK регистра CONTROL на амплитуду треугольной волны показано в таблице 20.5.

Таблица 20.5 – Значения поля MASK регистра CONTROL и его влияние на амплитуду

Значение в поле MASK регистра CONTROL	Поведение
0000	Амплитуда равна 1
0001	Амплитуда равна 3
0010	Амплитуда равна 7
0011	Амплитуда равна 15
0100	Амплитуда равна 31
0101	Амплитуда равна 63
0110	Амплитуда равна 127
0111	Амплитуда равна 255
1000	Амплитуда равна 511
1001	Амплитуда равна 1023
1010	Амплитуда равна 2047
>1011	Амплитуда равна 4095

Режим генерации волны по образцу (pattern mode)

При установленном значении «3» в поле WAVE регистра CONTROL (DAC_CONTROL_WAVE_PATTERN) включается генератор волны по образцу. Поля PATx регистров PATDORL/PATDORH напрямую подсоединены ко входам генератора. Схема внутреннего устройства генератора волны по образцу приведена на рисунке 20.8.

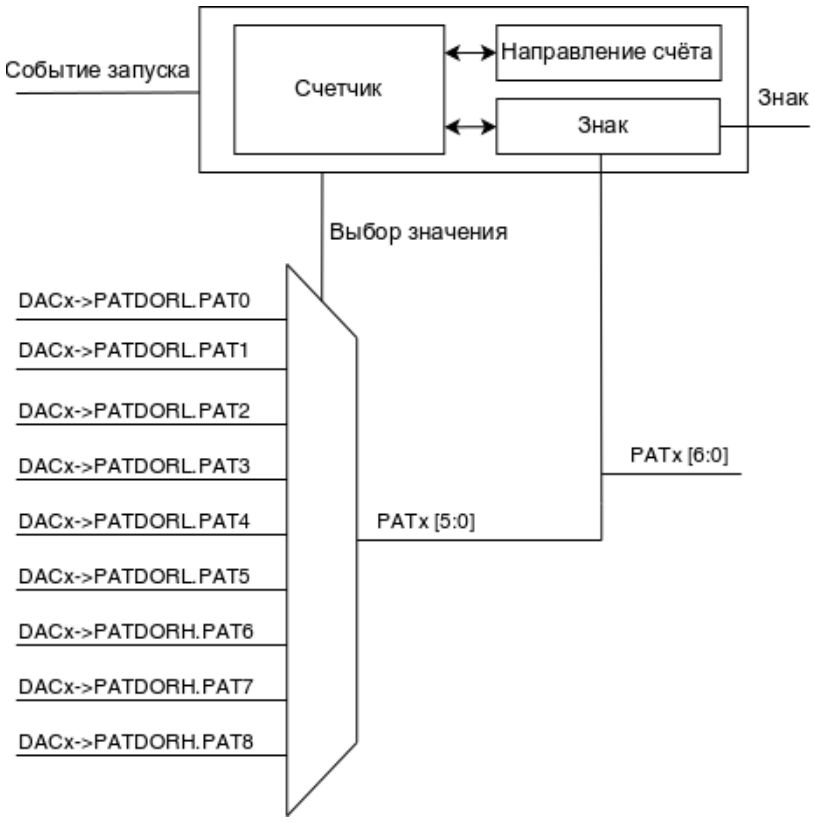


Рисунок 20.8 – Схема устройства генератора периодического сигнала по образцу

Регистры PATDORL/PATDORH содержат только одну четверть запрограммированного периодичного сигнала. Выход внутреннего счётчика используется для выбора одного из девяти входных значений PATx. Этот счётчик считает вверх и вниз и имеет дополнительный выход для формирования знака, инвертируется каждый раз, когда

значение счётчика достигло 0. Знак объединяется с текущим выбранным значением PATx и позволяет полностью формировать полный период волны периодической формы. По умолчанию, данные считаются знаковыми.

Форма волны задается с помощью пятиразрядных полей PAT0 – PAT8 регистров PATDORL/PATDORH. Для использования всей шкалы блока ЦАП требуется умножить выход генератора на 32, для чего следует установить значение «5» в поле SCALE регистра CONTROL (DAC_CONTROL_SCALE_SHIFT5).

Так как по умолчанию, данные являются знаковыми и если требуется генерация волны в беззнаковом режиме, дополнительно следует установить значение «31» в поле OFFSET в регистре CONTROL. Поле SIGN регистра CONTROL устанавливать при этом не нужно. В таблице 20.6 представлены стандартные значения для генерации периодических сигналов различных типов.

Таблица 20.6 – Стандартные значения для генерации периодических сигналов различных типов

Значение поля	LUT треугольник	LUT синус	LUT меандр
PAT0	0	0	31
PAT1	4	6	31
PAT2	8	12	31
PAT3	12	17	31
PAT4	16	22	31
PAT5	19	26	31
PAT6	23	29	31
PAT7	27	30	31
PAT8	31	31	31

Сдвоенный режим работы (Dual Mode)

Для того чтобы сконфигурировать этот режим необходимо установить в регистре CONTROL в ЦАП0 и ЦАП1 бит DATAMODE в «1» (DAC_CONTROL_DATAMODE_SIMULTANEOUS). В этом режиме ведущим становится контроллер ЦАП0:

- режим работы ЦАП1 задаётся в поле CONTROL.WAVE ЦАП0;
- события, которые вызывают старт преобразования ЦАП1 определяются в регистре CONTROL ЦАП0;
- контроллер ЦАП1 не формирует запросы к механизму DMA;
- при CONTROL.WAVE равным 0 (DAC_CONTROL_WAVE_NONE) данные для преобразования используются из регистра DATA_DUAL (для ЦАП0 используются поле DAC0, для ЦАП1 - DAC1);
- при CONTROL.WAVE равным 1 или 2 (DAC_CONTROL_WAVE_NOISE или DAC_CONTROL_WAVE_TRIANGULAR) базовые данные для генерации этих типов волн также используются из регистра DATA_DUAL (для ЦАП0 используются поле DAC0, для ЦАП1 - DAC1);
- при CONTROL.WAVE равным 3 (DAC_CONTROL_WAVE_PATTERN) значения образцов для формирования периодических волн используются каждым контроллером ЦАП из своего набора регистров PATDORL/PATDORH;
- значения в полях MASK, OFFSET, MULDIV, SCALE, SIGN регистра CONTROL влияют на результат у каждого из контроллеров.

Особенности:

- ЦАП1 должен быть включен (DAC1->CONTROL.DISABLE = 0);

- настройки WAVE и TSEL в регистре CONTROL ЦАП1 не влияют на результат преобразования;

- Настройки для источника опорного напряжения в регистре BGCTRL и полей BUF_EN, EXTREF регистра CONTROL в ЦАП1 следует выставить идентичными ЦАП0.

20.7 Преобразование

Для начала преобразования требуется:

1) Установить в регистре CONTROL все необходимые настройки (см. подразделы 20.4 «Формат данных ЦАП», 20.5 «Устройство выходного каскада к физическому блоку ЦАП» и 20.6 «Режимы работы»).

2) При необходимости использования внутреннего источника опорного напряжения сконфигурировать регистр BGCTRL.

3) В режиме генерации периодического сигнала по образцу, нужно заполнить данными поля PATx регистров PATDORL и PATDORH.

4) Сбросить бит DISABLE регистра CONTROL, тем самым разрешая работу ЦАП. После этого, поля регистра CONTROL и регистры PATDORL и PATDORH не следует изменять до тех пор, пока не будет запрещена работа ЦАП (установкой бита DISABLE в регистре CONTROL).

5) В зависимости от режима работы, загрузить регистр DATA12 или DATA8 данными на преобразование.

Старт преобразования может быть выполнен двумя способами:

1) Программно:

- Установить в поле TSEL регистра управления CONTROL значение «0» (DAC_CONTROL_TSEL_SWTRIG);
- Установить бит DISABLE регистра CONTROL равным «0»;
- Установить «1» в поле SWTRIG регистра SWTRIG;
- После окончания времени установления ЦАП данные появятся в выходном канале ЦАП.

2) Аппаратно:

- Установить в поле TSEL регистра управления CONTROL значение, соответствующее номеру запроса события запуска ЦАП от периферийного устройства (см. таблицу 20.2);
- Установить бит DISABLE регистра CONTROL равным «0»;
- Дождаться внешнего события, которое вызовет старт преобразования.

21 Датчик температуры TSENSOR

21.1 Функциональное описание

Микроконтроллер содержит датчик температуры, позволяющий измерять температуру кристалла во время работы. Выход датчика подключается на вход канала № 63 АЦП.

Структурная схема блока датчика температуры представлена на рисунке 21.1.

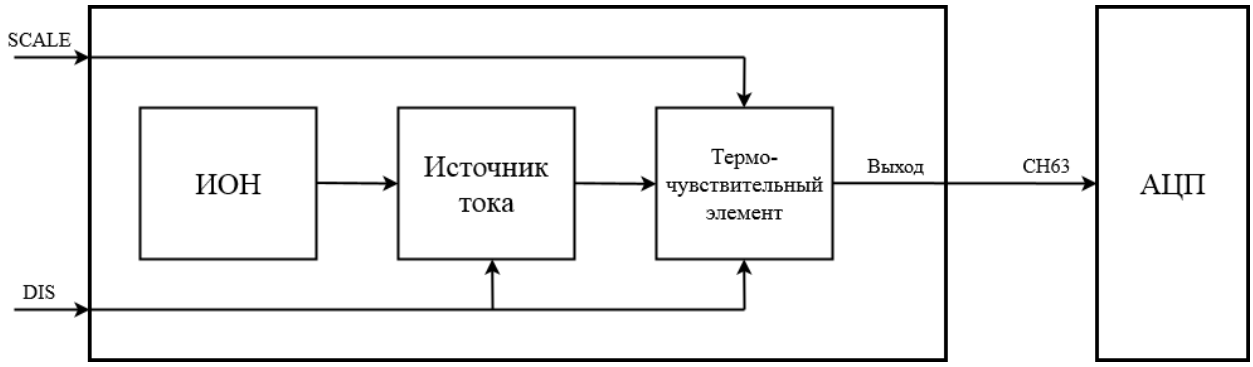


Рисунок 21.1 – Структурная схема датчика температуры

В состав датчика температуры входят:
- источник опорного напряжения (ИОН);
- источник тока;
- термочувствительный элемент.

Термочувствительным элементом схемы являются PN-переходы биполярных NPN транзисторов. Схема предусматривает возможность снятия температурной зависимости как с одного PN-перехода (СТАТ1), так и с двух последовательно соединенных PN-переходов (СТАТ2) (см. рисунок 21.2).

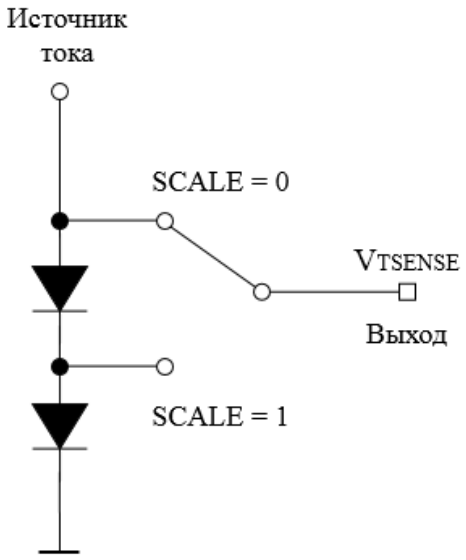


Рисунок 21.2 – Режимы подключения к термочувствительному элементу

Настройка выходного напряжения осуществляется с помощью управляющего бита SCALE регистра TSENSCTRL блока SIU согласно таблице 21.1.

Таблица 21.1 – Уровни выходного напряжения

SCALE	Уровень выходного напряжения OUT, В	Формула линейной регрессии
0	1,14 – 1,75	1,594 – 0,00356×T
1	0,575 – 0,88	0,793 – 0,00174×T

21.2 Электрические характеристики

Основные электрические характеристики датчика температуры представлены в таблице 21.2.

Таблица 21.2 – Электрические характеристики датчика температуры

Наименование параметра	Норма			Примечание
	не менее	номинал	не более	
Точность, °С	-3,8 -7	–	4 1,3	SCALE = 1 SCALE = 0
Наклон, мВ/°С	–	1,8 3,5	–	SCALE = 1 SCALE = 0
Время запуска, мкс	–	0,2	1	–
Ток потребления, мкА (статический)	7	16	24	–

Графики зависимости выходного напряжения от температуры представлены на рисунке 21.3.

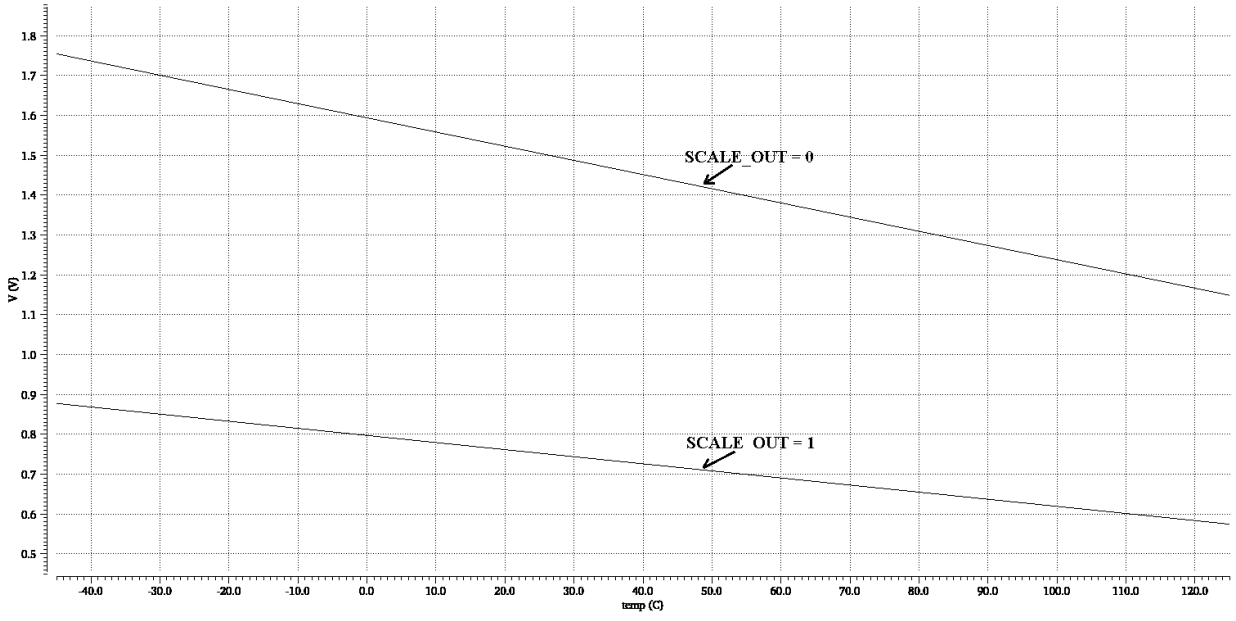


Рисунок 21.3 – Зависимость выходного напряжения от температуры

21.3 Управление датчиком температуры

Управление датчиком температуры осуществляется с помощью регистра TSENSCTRL блока SIU. Для включения датчика, необходимо сбросить бит DIS регистра TSENSCTRL.

В связи с тем, что датчик температуры имеет высокое выходное сопротивление, для получения достоверных значений напряжения необходимо увеличить время подключения к цепи зарядной емкости АЦП (тактировать АЦП частотой, которая не превышает 180 кГц). Либо использовать задержку запуска канала на основе регистра CHDELAY.

22 Модули аналоговых компараторов ACMP

22.1 Описание модуля

Микроконтроллер содержит четыре модуля аналоговых компараторов.

Аналоговый компаратор сравнивает два аналоговых сигнала и формирует логический выходной сигнал с результатом сравнения. Выходной сигнал может быть использован как внутри контроллера, так и подан на внешний вывод. Также, компаратор может формировать прерывания и/или сигналы запуска секвенсоров модуля АЦП. Модуль компаратора содержит один аналоговый компаратор и один ЦАП с двумя настраиваемыми уровнями выходного напряжения.

Управляющие регистры аналоговых компараторов приведены в приложении А.15.

Компаратор может сравнивать напряжение на внешнем выводе микросхемы с напряжением на другом выводе, а также с напряжением на встроенном ЦАП.

Структурная схема компаратора показана на рисунке 22.1.

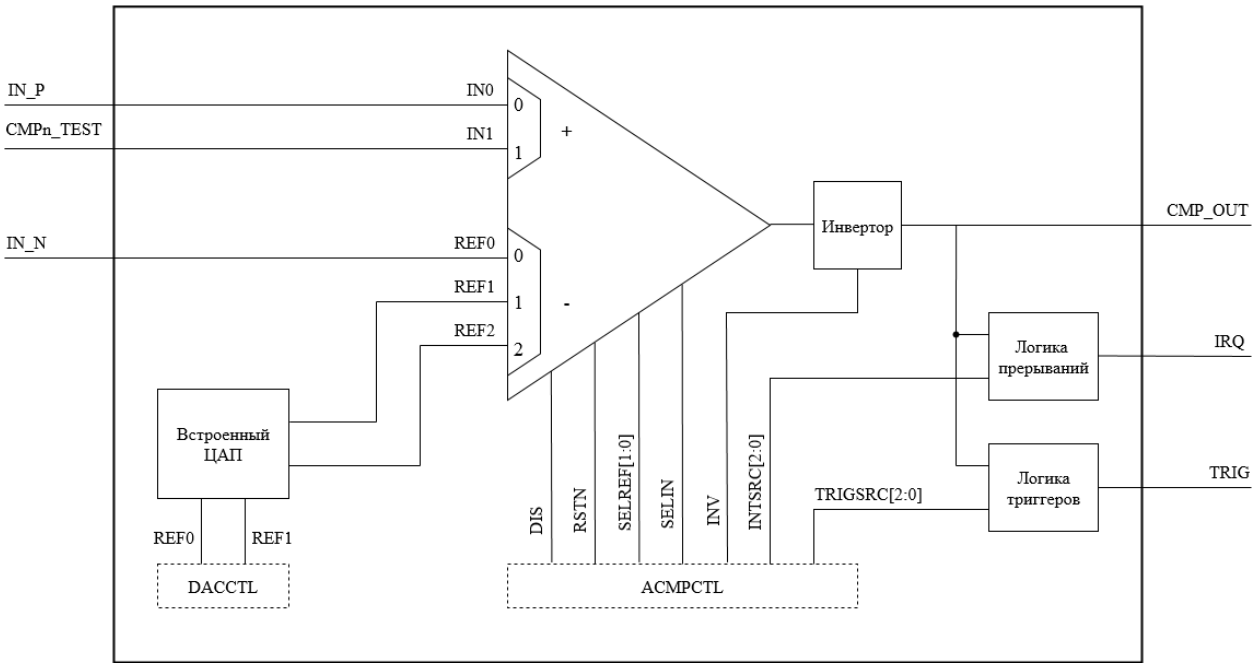


Рисунок 22.1 – Структурная схема компаратора

Логика сравнения уровней аналоговых сигналов осуществляется формулами 22.1 и 22.2.

$$VIN- < VIN+, CMP_OUT = 1. \quad (22.1)$$

$$VIN- > VIN+, CMP_OUT = 0. \quad (22.2)$$

В этой формуле VIN+ напряжение на выводе IN_P, а VIN- может выбираться из нескольких источников, заданных в регистре ACMPCTL поле SELREF.

- вывод IN_N;
- опорное напряжение с 0 канала встроенного ЦАП;
- опорное напряжение с 1 канала встроенного ЦАП.

При необходимости можно инвертировать выход компаратора установив бит INV регистра ACMPCTL.

Опорное напряжение встроенных ЦАП программируется через регистр DACCTL. Можно настроить два независимых выходных канала.

Опорное напряжение вычисляется по формуле 22.3.

$$Vref = \frac{AVDD \times REF_n}{16} \quad (22.3)$$

Блок может генерировать прерывание по каждому из каналов. Событие для генерации выбирается полем INTSRC регистра ACMPCTL. Аналогично сигналу прерывания генерируется сигнал для запуска секвенсоров АЦП. Событие для генерации выбирается полем TRIGSRC регистра ACMPCTL.

В качестве событий могут выступать следующие состояния выхода CMP_OUT:
- положительный фронт;
- отрицательный фронт;
- любой фронт;
- логический уровень нуля;
- логический уровень единицы.

В регистре INTEN разрешается генерация соответствующего прерывания на системный уровень. Регистр RIS содержит статус прерываний без учета маски разрешения INTEN. Регистр MIS содержит статус прерываний с учетом маски разрешения INTEN. Для сброса прерывания необходимо записать «1» в соответствующий бит регистра ICLR.

22.2 Функциональная безопасность

Для проверки функционирования компаратора предлагается программно-аппаратный метод с использованием отдельного блока ЦАП (DAC). У каждого модуля аналогового компаратора имеется тестовый вход IN1, который выбирается полем SELIN регистра ACMPCTL. Вход IN1 соединён с тестовым выходом ЦАП. Тестовый выход CMPx_TEST активируется записью в регистр TEST блока ЦАП (DAC0/DAC1) в соответствующее поле.

Таблица 22.1 – Подключение тестового выхода ЦАП к блоку аналогового компаратора

Блок ЦАП (DAC)	Тестовый выход ЦАП	Номер аналогового компаратора
DAC0	CMP0_TEST	ACMP0
DAC0	CMP1_TEST	ACMP1
DAC1	CMP0_TEST	ACMP2
DAC1	CMP1_TEST	ACMP3

23 Блок генератора случайных чисел TRNG

Аппаратный генератор случайных чисел – это устройство, которое генерирует последовательность случайных 32-разрядных чисел на основе измеряемых, хаотически изменяющихся, параметров протекающего физического процесса.

Криптографические модули могут использовать Генераторы Случайных Чисел (ГСЧ). В криптографии случайные последовательности играют определяющую роль. Они используются, в частности, для получения ключевой последовательности используемого алгоритма шифрования, для генерации гаммы поточных шифров, а также для выработки векторов инициализации (блочных шифров).

Управляющие регистры генератора случайных чисел приведены в приложении А.16.

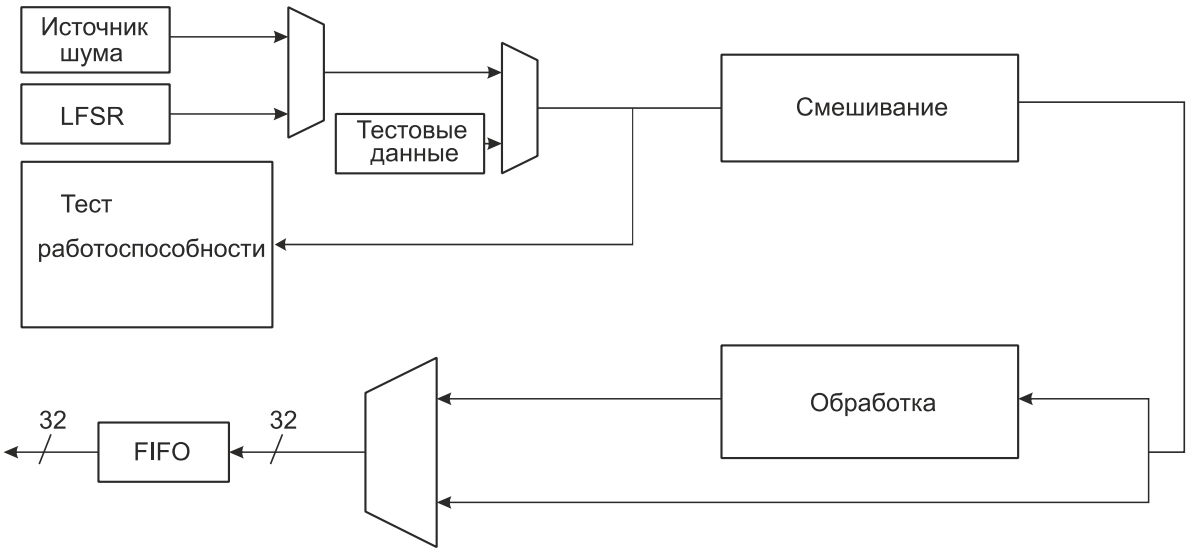


Рисунок 23.1 – Блок-диаграмма ИГСЧ

23.1 Источник энтропии

Фазовое дрожание цифрового сигнала данных (джиттер от англ. jitter) — нежелательные фазовые и/или частотные случайные отклонения передаваемого сигнала. Возникают вследствие нестабильности задающего генератора, изменений параметров линии передачи во времени и различной скорости распространения частотных составляющих одного и того же сигнала. Поскольку фазовое дрожание зависит от различных факторов, некоторые из которых полностью случайны, оно может быть использовано как источник случайных чисел.

В ГСЧ на базе такого явления сравниваются случайные задержки прохождения сигнала через кольцевые генераторы. Простейший кольцевой генератор состоит из нечетного числа инверторов, соединенных последовательно, при этом выход последнего соединен с входом первого инвертора, образуя линию обратной связи. Частота колебания такого генератора определяется суммой задержек всех его инверторов, это время зависит от множества параметров, включающих в себя тепловой шум в проводниках и полупроводниках и помехи в источниках питания.

Данная реализация Истинного Генератора Случайных Чисел (ИГСЧ) основана на использовании 32 отдельных колец генератора, каждое кольцо состоит из 5 инверторов. Выход аналогового сигнала преобразуется в цифровой сигнал путем выборки значений на постоянной тактовой частоте.

Структуру, реализованную в данном блоке, очень трудно имитировать. С помощью задания конфигурации битов возможно заставить данную структуру работать в качестве Псевдо-Случайного Генератора (ПСГ) для определенных задач моделирования.

Выборку из источника шума можно делать напрямую, в обход функции обработки (управляющий бит CONDBYPASS = 1).

23.2 Процесс дискретизации

Если выходной сигнал высокочастотного кольцевого генератора (RO) дискретизируется низкочастотным тактовым сигналом, то отобранный бит является случайным. Должно выполняться следующее утверждение $T_{OSC} \gg T_{RO}$, где T_{OSC} – период, в течение которого сигнал с кольцевого генератора (RO) находится в свободном режиме (независимо генерируется). Как показано на рисунках 23.2 и 23.3, поддерживаются два режима генерации:

- В горячем режиме (регистр COOLDPERIOD = 0), сигнал с кольцевых генераторов (RO) свободно генерируется, пока FIFO заполняется. Результат берётся каждые D_{OSC} тактового сигнала, где D_{OSC} можно настроить с помощью регистра SAMPERIOD. При этом период колебаний $T_{OSC}=D_{OSC}*T_{sysclk}$.

- В холодном режиме сигналы с кольцевых генераторов (RO) – запускаются установкой в значение «0» на время Drst тактового сигнала после каждой выборки результата, где Drst можно настроить через интерфейс APB (регистр COOLDPERIOD).

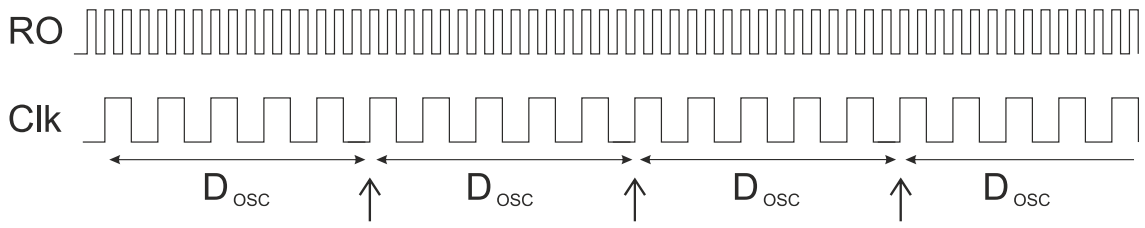


Рисунок 23.2 – Процесс выборки (горячий режим)

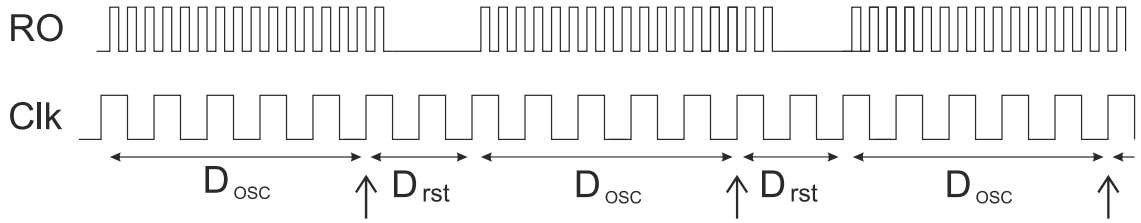


Рисунок 23.3 – Процесс выборки (холодный режим)

В действительности, идеальное значение D_{OSC} обычно гораздо больше, чем в примере, указанном на рисунке выше. Также типичное время удержания T_{OSC} >> Trst.

Горячий режим, теоретически, может иметь некоторые временные корреляции, которые в холодном режиме не проявятся, поскольку он «сбрасывается» перед генерацией каждого случайного значения. В горячем режиме дополнительная эффективность может быть получена с помощью регулировки параметра T_{OSC} (регистр SAMPERIOD); тесты на работоспособность сработают в любом случае при обнаружении ошибок.

Холодный режим может быть более подверженным сторонним атакам, поскольку остановки генерации предоставляют точки синхронизации, то есть злоумышленнику легче узнать, когда генерируются данные. Горячий режим не полностью защищен от такой проблемы, поскольку AES генерирует сигнатуру мощности, которая может использоваться в качестве точек синхронизации. Горячий режим имеет лучшую пропускную способность.

23.3 Непрерывное тестирование

Выборки из источника энтропии постоянно контролируются с помощью тестов на повторения и пропорциональных тестов, проходящих параллельно. Опционально также могут использоваться тесты на корреляцию и автокорреляцию.

23.4 Тест на количество повторений

В ИГСЧ реализован тест подсчета количества повторений. Допустимый уровень ложноположительных результатов установлен на значении $\alpha = 2^{-20}$. Источник случайного шума выдаёт недвоичную выборку. Для минимальной энтропии H = 8*0,5 = 4.0, допустимое пороговое значение повторений по умолчанию равно C = 1+(-log2 (α)/H) = 6.

Пороговое значение программируется в регистре REPTSTCUTOFF. Тест может быть отключен с помощью бита REPTSTDIS в регистре управления (CR). О провале теста повторений сообщается с помощью битов REPTSTFAILSH в регистре статуса (STAT).

23.5 Тест на подсчет пропорций (окно 512-недвоичных выборок)

Тест на количество пропорций, был реализован с размером окна W=512 недвоичных выборок. Допустимый уровень ложноположительных результатов – $\alpha = 2^{-20}$. Источник случайного шума выдаёт однобитную выборку, и после выбранного метода смешивания создаются недвоичные выборки. Для минимальной энтропии H = 8 * 0.5 = 4.0, допустимое пороговое значение пропорций по умолчанию равно C = 1+CRITBINOM(W, 2^{-H},1- α) = 63.

Пороговое значение программируется в регистре PROPTESTCUTOFF. Тест может быть отключен с помощью бита PROPTSTDIS в регистре управления (CR). О провале теста на подсчет пропорций сообщается с помощью битов PROPTESTFAIL в регистре статуса (STAT).

23.6 Тест корреляции

Когда два кольцевых генератора связаны между собой с помощью внешней синхронизации, их битовые потоки становятся коррелированными. Тест на подсчет корреляции может обнаружить это явление. Он использует мультиплексирование с временным разделением, и одновременно обрабатывает одну пару кольцевых генераторов. Окно размером W = 128 выборок используются с пороговыми значениями, определяемыми в регистрах CORRTESTCUTOFF. Отдельные проверки со смещением (или все проверки) могут быть отключены с помощью битов CORRTSTDIS в регистре управления (CR).

В каждом окне проверяется одна пара кольцевых генераторов (RO). Все возможные комбинации различных индексов генераторов (RO) будут проверены в порядке возрастания (1-0,...,7-0,0-1,2-1,...,7-1,...). Результаты проверок отключенных кольцевых генераторов будут замаскированы. Пара кольцевых генераторов, не прошедших тест, то есть их индексы будут установлены как битовые индексы в регистре CORRTESTFAILED.

23.7 Тест на подсчёт автокорреляции

Когда кольцевой генератор непрерывно дискретизируется без накопления достаточного дрожания (jitter), создаваемый битовый поток демонстрирует периодическое поведение. Тест на подсчет автокорреляции способен обнаружить это поведение. Он использует мультиплексирование с временным разделением и поддерживает один кольцевой генератор. Окно размером W = 128 выборок используется с пороговыми значениями, определяемыми в регистрах AUTOCORRTESTCUTOFF. Отсрочки для отдельных проверок (или все проверки) могут быть отключены с помощью битов ACORRTSTDIS в регистре управления (CR).

В каждом окне проверяется один кольцевой генератор (RO). Все кольцевые генераторы поиндексно будут проверены в порядке возрастания. Результаты отключенных

генераторов будут замаскированы. Генератор не прошедший тест, то есть его индекс, будет установлен как битовый индекс в регистре AUTOCORRTESTFAILED.

23.8 Тест при запуске

Тест на количество повторений и тест на подсчет пропорций (окно из 512 выборок) должны быть пройдены, прежде чем какие-либо данные будут считаны из FIFO.

23.9 Смешивание XOR

Наиболее эффективным способом получения случайных данных, с самой низкой задержкой, является объединение выходных данных кольцевых генераторов (RO). Выходные данные кольцевых генераторов могут объединяться по методу XOR (Исключающего Или), чтобы увеличить уровень энтропии. Операции XOR могут быть выполнены до или после выборки, но в данном ИГСЧ сначала проводится выборка, а затем XOR.

Имеются два уровня смешивания XOR. При выборе XOR1 в качестве метода смешивания будет применено сокращение с 32 до 4 бит. Дальнейшее сокращение возможно с помощью метода смешивания XOR2 как с 4 до 1 бита.

Последний возможный метод смешивания – это использование алгоритма исключения систематической ошибки Фон-Неймана. Он опирается на предположение – если входные биты являются независимыми и идентично распределенными, выходные, вероятно, распределены равномерно.

23.10 Обработка

Функция обработки берет значение поля COUNTBLOCK регистра CR и умножает его на 128 бит от источника энтропии в качестве входных данных и генерирует n_{out} = 128 бит в качестве выходных. Предполагая, что минимальная энтропия H=0.5 и значение поля COUNTBLOCK регистра CR 4, входная энтропия составляет h_{in} = 512 * 0.5 = 256 бит. Можно считать, что на выходе функции обработки имеем 128 бит полной энтропии. Функция обработки может быть протестирована путем отправки известных данных (используя TESTDATA с установкой контрольного бита TESTEN в значение 1).

23.11 FIFO

Выходные данные функции обработки разбиваются на 32-разрядные слова, перед сохранением в FIFO (программируемый размер 256 слов *32 бита).

Независимо от того – пропущена ли функция обработки, и необработанные шумовые выборки, и обработанные выборки будут записываться в FIFO, если бит старта/включения установлен, а КА находится не в состоянии «Error». Тесты работоспособности не препятствуют записи в FIFO, даже на этапе запуска, если установлен FIFOFILLST. Пользователь самостоятельно должен определить, являются ли данные в FIFO действительными. 32-разрядные случайные числа доступны через интерфейс APB, в диапазоне адресов [0x80 - 0xFC].

23.12 LFSR

Для проверки корректности данных, произведенных ИГСЧ, выборки могут быть получены из регистра LFSR (сдвиговый регистр линейной обратной связи) вместо источника шума. Эта функция доступна через регистры.

23.13 Прерывания

ИГСЧ имеет два источника прерываний:

Таблица 23.1 – Источники прерывания

Имя	Описание
FIFOFULLINT	Этот бит прерывания устанавливается, когда FIFO заполняется. Он сбрасывается при записи регистра FIFOLEV или при записи ‘0’ в 7 бит регистра STAT (FIFOFULL) или при программном сбросе установкой 8-го бита SOFTRST регистра CR.
ANYTESTFAILINT	Этот бит прерывания устанавливается при обнаружении ошибки в любом из тестов работоспособности. Он сбрасывается при записи ‘0’ в 6 бит регистра STAT (ANYTESTFAIL) или при программном сбросе установкой 8-го бита SOFTRST регистра CR.

Для каждого источника прерывания имеется один бит в регистре состояния (STAT) и один бит разрешения в регистре управления (CR). Состояния в регистре состояния не зависят от значений битов разрешения прерывания (TFAILINTEN, FFULLINTEN) в регистре управления. Биты разрешения прерывания используются только для генерации внешнего сигнала прерывания IRQ. Если какое-то прерывание активно и включено, то формируется сигнал IRQ высокого уровня.

В контроллер PLIC поступает одна линия прерывания TRNGINT от модуля ИГСЧ объединяющая в себе по принципу «логического ИЛИ» оба сигнала прерывания FIFOFULLINT и ANYTESTFAILINT.

23.14 Программирование ИГСЧ

Управление Конечным Автоматом

ИГСЧ управляется Конечным Автоматом (КА), который обрабатывает все низкоуровневые взаимодействия между различными внутренними компонентами, указанными на рисунке 23.1. Во время нормальной работы КА обеспечивает управление источником энтропии. Диаграмма состояния КА показана на рисунке 23.4.

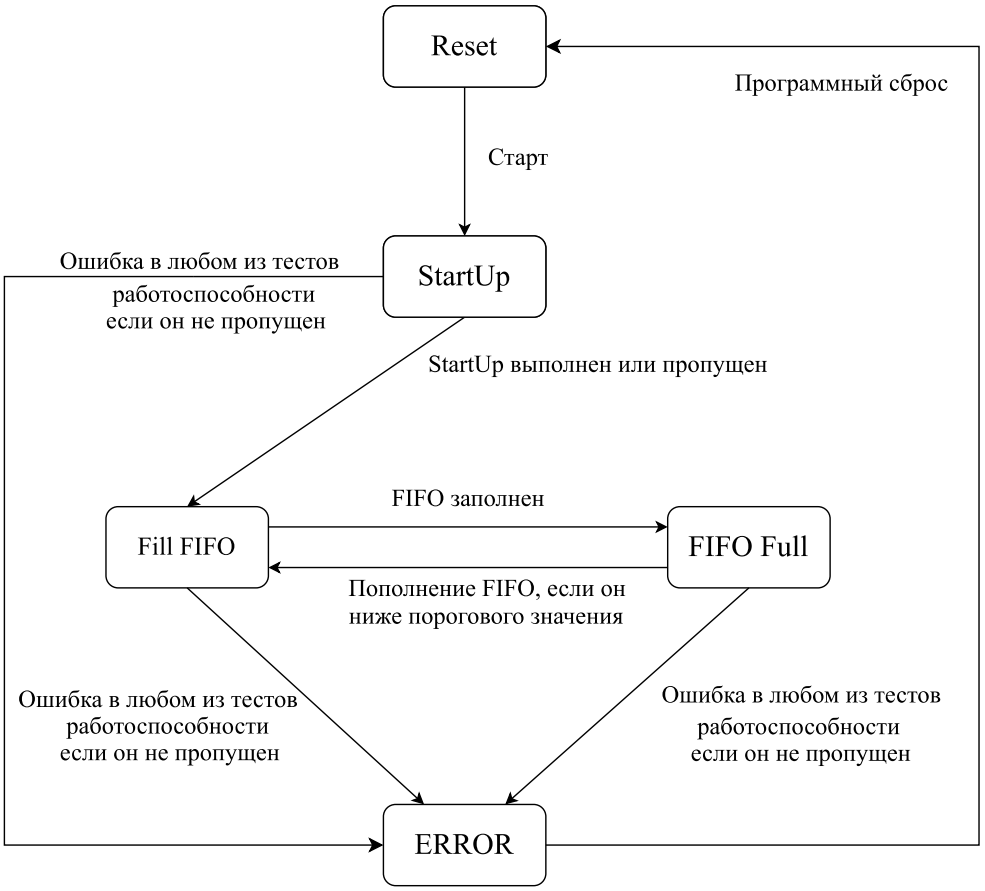


Рисунок 23.4 – Диаграмма состояний КА ИГСЧ

Диаграмма состояний не показывает, что аппаратный или программный сброс приводит к тому, что КА возвращает ИГСЧ обратно в состояние «Reset».

Порядок действий при проверке функции обработки:

1) Запустить программный сброс;
2) Записать управляющий регистр;
Убедиться, что COUNTBLOCK = 4 и TESTEN = 1;
3) Записать регистр ключей;
4) Записать данные в регистр TESTDATA (после каждого ввода данных подождать, пока флаг занятости DATABUSYTEST не сбросится);
5) Прочитать результат из FIFO.

В таблице ниже приведен пример данных.

Таблица 23.2 – Тест с известным ответом для функции обработки

	Формат 128 бит	Формат 32 бита APB
Ключ	0x2B7E151628AED2A6ABF7158809CF4F3C	0x16157E2B 0xA6D2AE28 0x8815F7AB 0x3C4FCF09
Входной сигнал	0x6BC0BCE12A459991E134741A7F9E1925	0xE1BCC06B 0x9199452A 0x1A7434E1 0x25199E7F
	0xAE2D8A571E03AC9C9EB76FAC45AF8E51	0x578A2DAE 0x9CAC031E 0xAC6FB79E 0x518EAF45
	0x30C81C46A35CE411E5FBC1191A0A52EF	0x461CC830 0x11E45CA3 0x19C1FBE5 0xEF520A1A
	0xF69F2445DF4F9B17AD2B417BE66C3710	0x45249FF6 0x179B4FDF 0x7B412BAD 0x10376CE6
Ожидаемый результат	0x3FF1CAA1681FAC09120ECA307586E1A7	0xA1CAF13F 0x09AC1F68 0x30CA0E12 0xA7E18675

Порядок действий при проверке источника энтропии:

1) Запустить программный сброс;

2) Записать управляющий регистр, убедившись, что ни один из тестов работоспособности не будет пропущен, и установить бит запуска/включения;

3) Дождаться пропадания состояний «Сброс», «Запуск» или «Ошибка». В этом случае все тесты пройдены, и источник энтропии работает правильно;

4) Если состояние «Ошибка» не пропадает, выбрать другие значения для SAMPERIOD и WARMPERIOD и вернуться к шагу 1.

Порядок действий при программировании ключа:

1) Проверить регистр FIFOLEV, чтобы узнать количество сгенерированных случайных чисел, или дождаться FIFOFULL IRQ;

2) Когда FIFOLEV достигнет ожидаемого значения или когда будет получен FIFOFULL IRQ, считать случайные числа через интерфейс APB;

3) Используйте 4x32-битных случайных значения для программирования случайного ключа;

4) Применить программный сброс, для очистки FIFO.

Порядок действий для получения выборок:

1) Проверить регистр FIFOLEV, чтобы узнать количество сгенерированных случайных чисел, или дождаться FIFOFULL IRQ;

2) Когда FIFOLEV достигнет ожидаемого значения или когда будет получен FIFOFULL IRQ, считать случайные числа через интерфейс APB.

Порядок действий для изменения конфигурации:

1) Отключить ИГСЧ (сбросить бит [0] – START – управляющего регистра CR);

2) Настройте регистр управления (CR);

3) Выполнить сброс пути к данным, чтобы получить чистую начальную точку (установить бит [8] – SOFTRST – управляющего регистра)

4) Включить ИГСЧ обратно (установите бит [0] – START – управляющего регистра).

Формат данных – порядок байтов

Все криптографические данные обрабатывают в соответствии с форматом big-endian (стандарт AES). Это означает, что первый байт (младший адрес) данных является самым значимым байтом (MSB).

Шина APB использует формат little endian. Это означает, что наименее значимый байт (LSB) хранится по самому младшему адресу.

Например, 128-битный блок D[A] = 0x00112233445566778899AABBCCDDEEFF, хранящийся по адресу A (кратному 4), отображается на шине APB в виде следующих 32-разрядных слов:

- D[A+0] = 0x33221100;
- D[A+4] = 0x77665544;
- D[A+8] = 0xBBAA9988;
- D[A+12] = 0xFFDDEECC.

23.15 Производительность

Количество циклов (тактового сигнала), необходимых ИГСЧ для генерации данных, можно оценить по следующей формуле:

$$Nbcycles \approx 1024 * SAMPERIOD + WARMPERIOD, \quad (23.1)$$

где 1024 – начальное количество выборок,

SAMPERIOD и WARMPERIOD – два значения конфигурации (задаются с помощью одноименных регистров).

При типичном использовании, если установить SAMPERIOD = 30 и WARMPERIOD = 512, с момента установки управляющего регистра до появления первого случайного слова в выходном буфере FIFO потребуется приблизительно 1024*30+512=31232 цикла.

24 Хеш-процессор (HASH)

24.1 Функционирование хеш-процессора

Хеш-процессор представляет собой полный набор реализаций алгоритмов безопасного хеширования (SHA-1, SHA-224, SHA-256), хеш-алгоритм MD5 (алгоритм хеш-суммы 5) и HMAC (хеш-код аутентификации сообщений) алгоритм, подходящий для различных приложений. Он вычисляет хеш-сумму (160 бит для алгоритма SHA-1, 224 бита для алгоритма SHA-224, 256 бит для алгоритма SHA-256 и 128 бит для алгоритма MD5) для сообщений длинной до 264 – 1 бит, когда как алгоритм HMAC обеспечивает способ аутентификации сообщений с использованием хеш-функций. Алгоритмы HMAC заключаются в вызове SHA-1, SHA-224, SHA-256 или MD5 хеш-функций дважды.

Управляющие регистры хеш-процессора приведены в приложении А.17.

24.2 Основные особенности

Основные особенности хеш-процессора:

- может применяться в системах аутентификации данных;
- быстрое вычисление алгоритмов SHA-1, SHA-224 и SHA-256, и MD5;
- 32-битные слова данных для входных данных, поддержка словного, полусловного, байтового и битовой строки с порядком от младшего к старшему;
- поддержка автоматического изменения порядка следования бит;
- автоматическая процедура заполнения: вставки ведущей единицы, длины сообщения до размера 512 (16 * 32) бит при формировании блока;
- автоматическая работа с потоком данных для прямого доступа к памяти.

Примечание – Автоматическое заполнение данных согласно определению в спецификациях к алгоритмам SHA-1, SHA-224 и SHA-256, состоит из добавления дополнительного бита «1» за которым следуют N-бит «0» для получения общей длины, равной 448 по модулю 512. После этого, сообщение завершается 64-битным целочисленным значением, которое является двоичным представлением длинны оригинального сообщения. Для хеш-процессора, передаваемые данные являются 32-битными словами, таким образом нужно предоставить дополнительную информацию в конце записи сообщения, а именно количество допустимых битов в последнем записанном 32-битном слове.

24.3 Описание хеш-процессора

Блок-схема хеш-процессора представлена на рисунке 24.1.

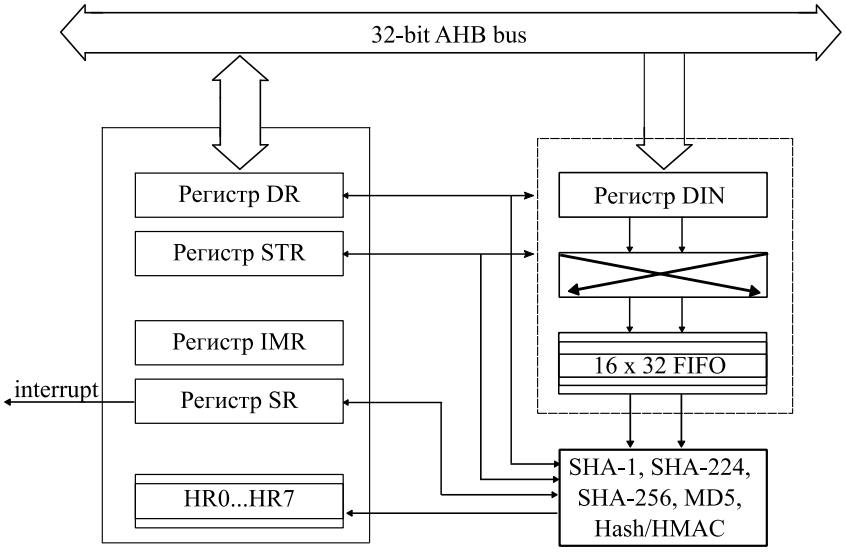


Рисунок 24.1 – Блок схема хеш-процессора

При передаче сообщения длинной до 264 бита на вход хеш-процессора, алгоритмы SHA-1, SHA-224, SHA-256 и MD5 вычисляют 160-битную, 224-битную, 256-битную и 128-битную выходную строку, соответственно, называемую хеш-суммой сообщения. После этого хеш-сумму сообщения можно обработать с помощью алгоритма цифровой подписи, чтобы сгенерировать или проверить подпись для сообщения. Подписание хеш-суммы сообщения, а не самого сообщения, часто повышает эффективность процесса, поскольку хеш-сумма сообщения обычно намного меньше по размеру, чем само сообщение. При проверке цифровой подписи необходимо использовать тот же хеш-алгоритм, что и при формировании цифровой подписи.

Алгоритмы SHA-1, SHA-224, SHA-256 и MD5 квалифицируются как «безопасные», потому что с точки зрения вычислений невозможно найти сообщение, соответствующее данной хеш-сумме сообщения, или найти два разных сообщения, которые создают одну и ту же хеш-сумму сообщения. Любое изменение сообщения при передаче с очень высокой вероятностью приведет к вычислению другой хеш-суммы сообщения, и подпись не сможет быть проверена.

Сообщение или массив данных для обработки хеш-процессором следует воспринимать как битовую строку. Длинна этого сообщения определяется количеством битов в этом сообщении (пустое сообщение имеет длину 0). Так, 32-бита этой битовой строки формируют 32-битное слово.

Примечание – Битовая строка растёт слева направо, и биты могут группироваться как байты (8-бит) или слова (32-бит).

24.4 Длительность обработки

Время вычисления промежуточного блока сообщения занимает:

- 66 циклов тактового сигнала HCLK для алгоритма SHA-1;
- 50 циклов тактового сигнала HCLK для алгоритма SHA-224;
- 50 циклов тактового сигнала HCLK для алгоритма SHA-256;
- 50 циклов тактового сигнала HCLK для алгоритма MD5.

Помимо времени вычисления необходимо добавить время, необходимое для загрузки 16 слов блока в процессор (по крайней мере 16 циклов тактового сигнала для блока в 512 бит).

Время, необходимое для обработки последнего блока сообщения (или ключа в HMAC) может быть длиннее. Время зависит от длины последнего блока и размера ключа (в режиме HMAC). В сравнении с обработкой промежуточного блока, оно может быть увеличено в:

- от 1 до 2.5 раз для хеш-функции;
- около 2.5 раз для входного ключа HMAC;
- от 1 до 2.5 раз для сообщения HMAC;
- около 2.5 раз для внутреннего ключа HMAC в случае короткого ключа, либо если используются одинаковые внутренний и внешний длинные ключи;
- от 3.5 до 5 для внешнего ключа HMAC в случае разных внутренних и внешних длинных ключей.

24.5 Типы данных

Данные передаются в хеш-процессор по 32 бита (слова) с помощью записи в регистр DIN. При этом, исходная битовая строка может быть организована в байты, полуслова или слова, или даже представлена в виде битов. Так как организация системной памяти имеет обратный порядок байтов, а вычисления SHA1, SHA-224 и SHA-256 выполняются с прямым порядком байтов, в зависимости от того, как сгруппирована исходная битовая строка,

операция замены битов, байтов или полуслов выполняется автоматически хеш-процессором.

Тип обрабатываемых данных настраивается битовым полем DATATYPE в управляющем регистре CR:

- при представлении входных данных как битовой строки, все биты следует поменять местами (записать значение 0x3 в битовое поле DATATYPE), как показано на рисунке 24.2.
- при представлении входных данных как байтов, байты следует поменять местами (записать значение 0x2 в битовое поле DATATYPE), как показано на рисунке 24.3.
- при представлении входных данных как полу слов, полу слова следует поменять местами (записать значение 0x1 в битовое поле DATATYPE), как показано на рисунке 24.4.

Младший значащий бит сообщения должен быть в позиции 0 (справа) в первом слове, введенном в хеш-процессор, 32-й бит битовой строки должен быть в позиции 0 во втором слове, введенном в хеш-процессор, и т.д.

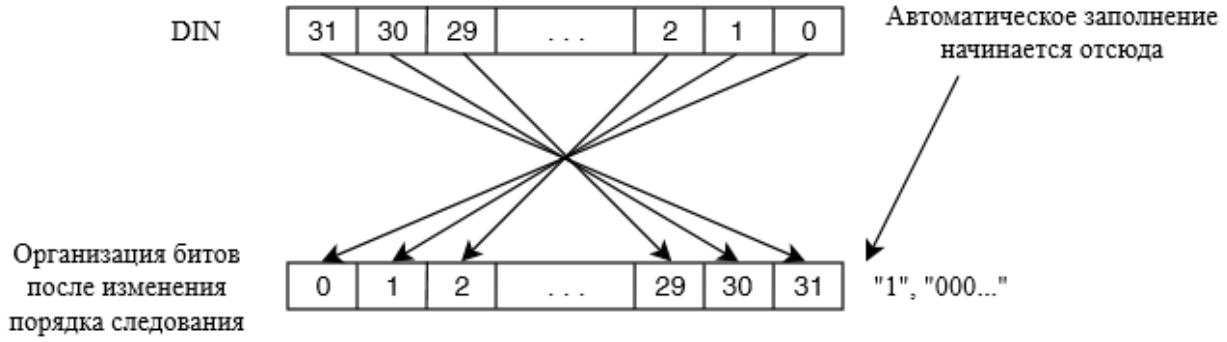


Рисунок 24.2 – Представление входных данных как битовой строки

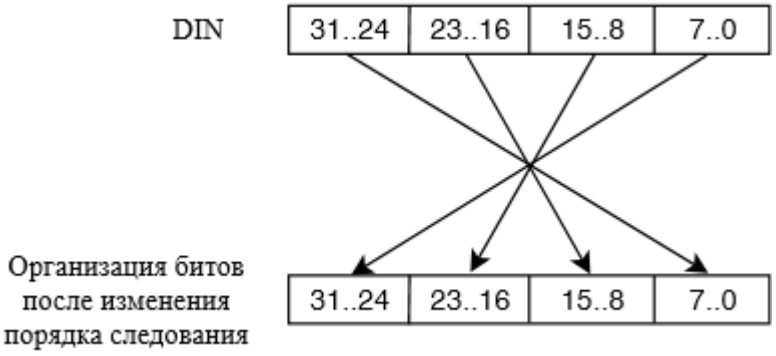


Рисунок 24.3 – Представление входных данных как байтов

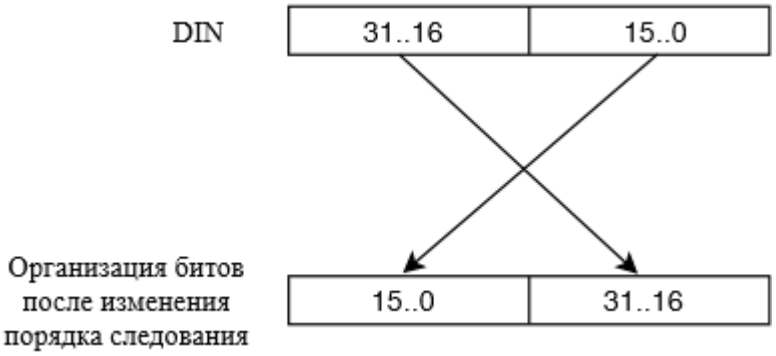


Рисунок 24.4 – Представление входных данных как полуслов

24.6 Вычисление хеш-суммы сообщения

Хеш-процессор последовательно обрабатывает блоки по 512 бит при вычислении хеш-суммы сообщения. Таким образом, каждый раз, когда DMA или ЦП записывают 16 × 32-битных слов (512 бит) в хеш-процессор, хеш-функция автоматически начинает вычислять хеш-сумму сообщения. Данная операция – частичное вычисление хеш-суммы.

Сообщение для обработки вводится в периферийное устройство 32-битными словами, записанными в регистр DIN. Текущее содержимое регистра DIN передается в FIFO каждый раз, когда в регистр записываются новые данные. Регистр DIN и входной буфер FIFO образуют входной буфер длиной 17 слов.

Обработка блока может начаться только после того, как последнее значение блока войдет во входной буфер. Периферийное устройство должно получить информацию о том, содержит ли регистр DIN последние биты сообщения или нет. Возможны два случая:

1) Прямой доступ к памяти не используется:

- вычисление частичной хеш-суммы сообщения выполняется путем записи слов блока сообщения в регистр DIN. Программное обеспечение должно дождаться, пока процессор снова будет готов (когда DINIS=1), прежде чем записывать новые данные следующего блока в DIN.

- вычисление окончательной хеш-суммы сообщения (последний введенный блок) делается путем записи бита DCAL в 1.

2) При использовании прямого доступа к памяти: содержимое регистра DIN автоматически определяется информацией, отправленной контроллером DMA.

- в случае одиночной передачи DMA: когда последний блок был передан в регистр DIN по каналу DMA, бит DCAL регистра STR будет автоматически установлен в 1 в регистре STR, чтобы запустить окончательный расчет хеш-суммы сообщения.

- в случае множественной передачи DMA: Бит множественной передачи DMA (MDMAT) должен быть установлен в 1 программным обеспечением, чтобы бит DCAL не устанавливался автоматически аппаратно, в этом случае окончательный расчет хеш-суммы сообщения для хеша и для каждой фазы алгоритма HMAC (для получения более подробной информации об этапах HMAC см. раздел операций HMAC) не будет запущен в конце запроса на передачу DMA, что позволит процессору переконфигурировать DMA. Во время последней передачи DMA бит множественной передачи DMA (MDMAT) может быть очищен программным обеспечением, чтобы автоматически установить бит DCAL в конце последнего блока и запустить окончательное вычисление хеш-суммы сообщения.

Процесс ввода данных и вычисление частичной хеш-суммы продолжается до тех пор, пока последние биты исходного сообщения не будут записаны в регистр DIN. Поскольку длина (количество битов) сообщения может быть любым целым числом, последнее слово, записанное в хеш-процессор, может иметь допустимое количество битов от 1 до 32. Это количество допустимых битов в последнем слове (NBLW) должно быть записано в регистр STR, чтобы правильно выполнить заполнение сообщения перед окончательным вычислением хеш-суммы сообщения.

Как только это будет сделано, запись в STR с битом DCAL = 1 запускает обработку последнего введенного блока сообщения хеш-процессором. Эта обработка заключается в:

- автоматическое выполнение операции заполнения сообщения: цель этой операции – сделать общую длину дополненного сообщения кратной 512. Хеш-процессор последовательно обрабатывает блоки по 512 бит при вычислении хеш-суммы сообщения;

- вычисление окончательной хеш-суммы сообщения.

Когда DMA включен, он предоставляет информацию хеш-процессору при передаче последнего слова данных. Затем заполнение и вычисление хеш-суммы выполняются автоматически, как и при установке бита DCAL.

24.7 Автоматическое заполнение сообщения

Заполнение сообщения заключается в добавлении «1», за которой следуют m «0», за которым следует 64-битное целое число до конца исходного сообщения для создания дополненного блока сообщения длиной 512. «1» добавляется к последнему слову, записанному в регистр DIN в битовую позицию, определяемой битовым полем NBLW регистра STR, а оставшиеся старшие биты очищаются («0»).

Пример – Предположим, что исходное сообщение представляет собой двоично-кодированную форму ASCII «abc» длиной L = 24:

```
  байт 0   байт1    байт 2   байт 3
 01100001 01100010 01100011 UUUUUUUU
<-- первое слово записанное в DIN -->
```

Поле NBLW должно быть загружено со значением 24: «1» добавляется к битовой позиции 24 в битовой строке (начиная отсчета слева направо в приведенной выше битовой строке), что соответствует биту 31 в регистре DIN (порядок следования байтов от младшего к старшему):

```
01100001 01100010 01100011 1UUUUUUU
```

Поскольку L = 24, количество битов в приведенной выше битовой строке равно 25, и к ним добавляются 423 «0», что дает теперь 448. Это дает (в шестнадцатеричном формате с порядком следования битов от старшего к младшему):

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

Значение L в двухсловном представлении (то есть 00000000 00000018) добавляется. Отсюда окончательное заполненное сообщение в шестнадцатеричном формате:

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000028
```

Если хеш-процессор запрограммирован на использование порядка следования байтов от младшего к старшему, указанное выше сообщение необходимо ввести, выполнив следующие шаги:

1 В регистр DIN записывается 0xUU636261 (где «U» означает «все равно»);

2 В регистр STR записывается 0x18 (количество допустимых битов в последнем слове, записанном в регистр DIN, равно 24, так как исходная длина сообщения составляет 24 бита);

3 0x10 записывается в регистр STR, чтобы начать заполнение сообщения и вычисляется хеш-сумма. Когда NBLW ≠ 0x00, заполнение сообщения помещает «1» в регистр DIN в битовой позиции, определяемой значением NBLW, и вставляет «0» в битовые позиции [31:(NBLW+1)]. Когда NBLW == 0x00, заполнение сообщения вставляет одно новое слово со значением 0x0000 0001. Затем добавляется полное нулевое слово (0x0000 0000) и длина сообщения в представлении из двух слов, чтобы получить блок размером 16 x 32 бит слова;

4 Выполняется вычисление хеш-суммы, после чего хеш-сумма сообщения становится доступной в регистрах HRx. Например, для алгоритма SHA-1:

```
HR0 = 0xA9993E36
HR1 = 0x4706816A
HR2 = 0xBA3E2571
HR3 = 0x7850C26C
HR4 = 0x9CD0D89D
```

24.8 Вычисление хеш-функции

Хеш-функция (SHA-1, SHA-224, SHA-256 и MD5) выбирается, когда бит INIT записывается в «1» в регистре CR, а бит MODE в «0» в CR. Алгоритм (SHA-1, SHA-224, SHA-256 или MD5) выбирается одновременно (то есть, когда бит INIT установлен) с использованием битов ALGO.

Затем сообщение может начинать обрабатываться, записав его пословно в регистр DIN. Когда блок из 512 бит, то есть 16 слов, был записан, вычисление частичной хеш-суммы сообщения начинается с записи первых данных следующего блока. Хеш-процессор остается занятым в течение 66 тактов для алгоритма SHA-1 или 50 тактов для алгоритма MD5, алгоритма SHA-224 и алгоритма SHA-256.

Затем процесс можно повторять до последнего слова сообщения. Если используются передачи с помощью DMA, обратитесь к подразделу 24.10 «DMA интерфейс». В противном случае, если длина сообщения не кратна 512 битам, то для запуска вычисления окончательной хеш-суммы необходимо записать в регистр STR.

После вычисления хеш-сумма может быть прочитана из регистров HR0...HR7, где:

- HR4..HR7 не имеют значения, когда выбран алгоритм MD5;
- HR5..HR7 не имеют значения, когда выбран алгоритм SHA-1;
- HR7 не имеет значения, если выбран алгоритм SHA-224.

24.9 Вычисление HMAC

Алгоритм HMAC используется для аутентификации сообщений путем необратимой привязки обрабатываемого сообщения к ключу, выбранному пользователем.

По сути, алгоритм состоит из двух вложенных хеш-операций:

```
HMAC(text) = H((key | pad) XOR opad | H((key | pad) XOR ipad | text));
```

Где:

- opad – число вида (0x5c 0x5c 0x5c … 0x5c), где байт 0x5c повторяется b раз, a ipad число вида (0x36 0x36 0x36 … 0x36), где байт 0x36 повторяется b раз (b – размер блока в байтах, b = 64 для 512-битного блока);
- key – секретный ключ;
- text – сообщение (данные);
- pad – представляет собой последовательность нулей, необходимую для расширения ключа до длины блока данных базовой хеш-функции (то есть 512 бит для алгоритмов хеширования SHA-1, SHA-224, SHA-256 и MD5);
- | представляет собой оператор конкатенации.

Для расчета HMAC требуются четыре различных этапа:

1) Блок инициализируется путем записи бита INIT в «1» с битом MODE в «1» и битами ALGO, установленными в значение, соответствующее желаемому алгоритму. Бит LKEY также должен быть установлен на этом этапе, если используемый ключ длиннее 64 байт (в этом случае спецификации HMAC указывают, что вместо реального ключа следует использовать хеш-ключ). Также, при использовании длинного ключа и, если внутренний и внешний ключ одинаковый, следует установить бит SAMK для ускорения вычисления HMAC.

2) Затем ключ (который будет использоваться для внутренней хеш-функции) передается ядру. Эта операция использует тот же механизм, что и для отправки сообщения в хеш-операции (то есть путем записи в DIN и, наконец, в STR).

3) После ввода последнего слова и начала вычисления хеш-процессор обрабатывает ключ. Затем он готов принять текст сообщения, используя тот же механизм, который использовался для отправки сообщения в операции хеширования.

4) После первого раунда хеширования хеш-процессор возвращает «готово», чтобы указать, что он готов получить ключ, который будет использоваться для внешней хеш-функции (обычно этот ключ совпадает с тем, который используется для внутренней хеш-функции). Когда вводится последнее слово ключа и начинается вычисление, результат HMAC становится доступным в регистрах HR0... HR7.

Примечания:

1) Задержка вычисления HMAC зависит от длины ключей и сообщения. Вы можете использовать HMAC как две вложенные базовые хеш-функции с одинаковой длиной ключа (длинным или коротким).

2) При использовании длинного ключа и, если внутренний и внешний ключ одинаковый и бит SAMK в регистре CR установлен, не требуется подавать внешний ключ (см. п.4). Результат HMAC будет доступен после обработки текста сообщения. Хеш-процессор возвратит сигнал «готово» и вы можете прочитать регистры HRx.

Пример – Ниже приведен пример алгоритма HMAC SHA-1 (ALGO=00 и MODE=1 в CR), из спецификации NIST. Предположим, что исходное сообщение представляет собой двоично-кодированную форму ASCII «Sample message for keylen=blocklen» длиной L = 34 байта.

1) В регистр DIN вводится внутренний ключ (длина = 64, т. е. без заполнения), указанный в примере NIST. Поскольку длина ключа = 64, бит LKEY установлен в 0 в регистре CR.

00010203	***04050607***	***08090A0B***	***0C0D0E0F***	***10111213***	***14151617***
18191A1B	***1C1D1E1F***	***20212223***	***24252627***	***28292A2B***	***2C2D2E2F***
30313233	***34353637***	***38393A3B***	***3C3D3E3F***		

2) В регистр DIN вводится сообщение (длина = 34, т.е. требуется заполнение). STR должен быть установлен на 0x20, чтобы начать заполнение сообщения и вычисление внутреннего хеша («U» – не важно).

53616D70	***6C65206D***	***65737361***	***67652066***	***6F72206B***	***65796C65***
6E3D626C	***6F636B6C***	***656EUUUU***			

3) В регистр DIN вводится внешний ключ (длина = 64, т.е. без заполнения), идентичный внутреннему ключу.

4) Затем выполняется окончательное вычисление хеш-функции. Результат HMAC-SHA1 доступен в регистрах HRx (x = от 0 до 4), как показано ниже:

```
HR0 = 0x5FD596EE
HR1 = 0x78D5553C
HR2 = 0x8FF4E72D
HR3 = 0x266DFD19
HR4 = 0x2366DA29
```

24.10 DMA интерфейс

Хеш-процессор поддерживает передачи с использованием протокола прямого доступа к памяти (DMA). Его можно использовать для записи данных, предварительно сконфигурировав канал контроллера DMA и установив бит DMAE в регистре CR. Когда этот бит установлен, хеш-процессор инициирует запрос к контроллеру DMA каждый раз, когда блок должен быть записан в регистр данных DIN. Запросы генерируются автоматически до тех пор, пока число оставшихся передач не станет равным 0 в регистре управления каналом. Для получения дополнительной информации обратитесь к разделу 14 «Контроллер прямого доступа к памяти DMA» и подразделу 24.6 «Вычисление хеш-суммы сообщения».

Перед началом передачи DMA программное обеспечение должно запрограммировать количество допустимых битов в последнем слове, которое будет скопировано в регистр DIN. Это делается путем записи в регистр STR следующего значения:

```
NBLW = Len(Message)% 32, где «x%32» дает остаток от деления x на 32.
```

Бит DMAS регистра SR предоставляет информацию об активности интерфейса DMA. Этот бит устанавливается с помощью DMAE и сбрасывается, когда DMAE очищается и передача по DMA не выполняется.

Примечание – С битом DMAS не связано прерывание.

24.11 Прерывания

Хеш-процессор имеет два отдельных маскируемых источника прерываний, подключённых к одному и тому же вектору прерывания.

Вы можете включить или отключить источники прерывания по отдельности, изменив биты маски в регистре IMR. Установка соответствующего бита маски в 1 разрешает прерывание.

Статус отдельных источников прерываний можно прочитать из регистра SR.

На рисунке 24.5 представлена схема формирования прерываний.

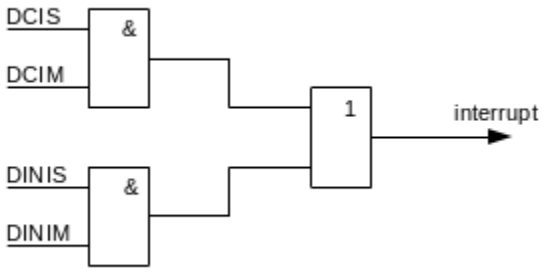


Рисунок 24.5 – Схема прерываний хеш-процессора.

25 Блок подсчёта циклического избыточного кода CRC

Блок проверки подсчётом циклически избыточным кодом (CRC) используется для получения CRC из 8-, 16-, 32-битных слов данных и генератора полиномов.

Среди других применений, методы, основанные на подсчёте CRC, используются для определения корректности переданных данных или целостности хранимых данных. Блок вычисления CRC помогает посчитать контрольную сумму данных во время выполнения, для сравнения с эталонной суммой, созданной ранее и сохранённой в памяти.

Управляющие регистры блока CRC приведены в приложении А.18.

25.1 Особенности блока CRC

- Программируемый генератор полиномов (7-, 8-, 16-, 32-бит);
- Использует полином CRC32-IEEE-802.3 (0x04C1_1DB7) по умолчанию;
- Оперирует с 8-, 16-, 32-битными входными данными;
- Программируемое начальное значение CRC;
- Один 32-битный регистр данных;
- Вычисление CRC осуществляется за 4 такта HCLK для входных 32-битных данных;
- Возможность обратимости входных/выходных данных для поддержки различных порядков байт данных, опциональный XOR для выходных данных

25.2 Функции блока CRC

Структура блока CRC

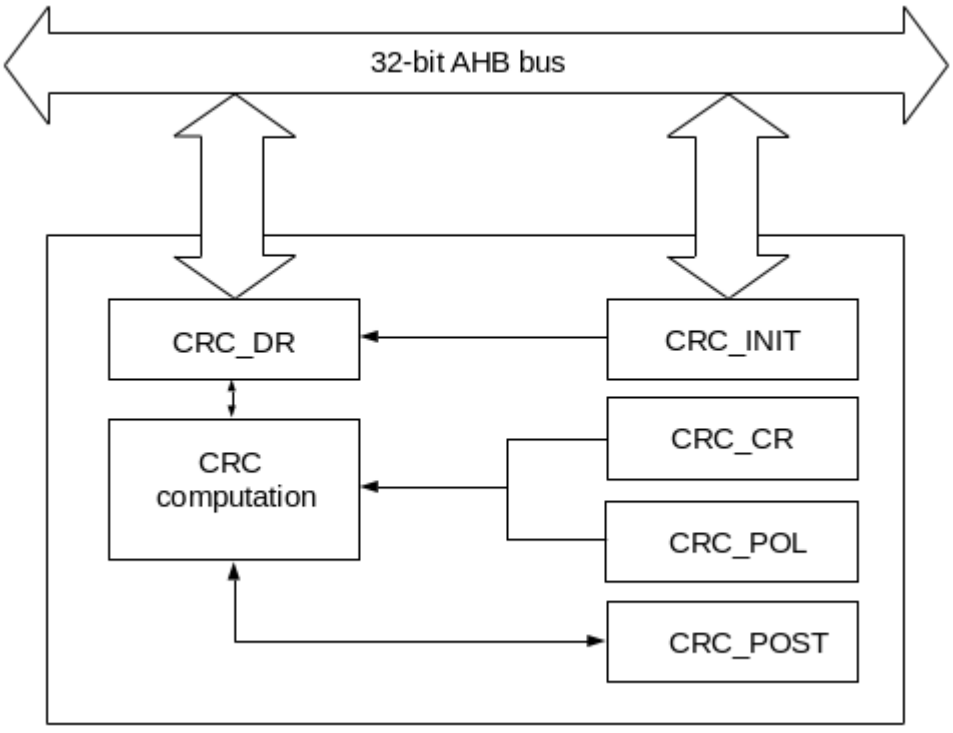


Рисунок 25.1 – Структура блока проверки подсчётом циклически избыточным кодом

Описание блока CRC

Аппаратный блок подсчёта CRC имеет один 32-разрядный регистр данных (DR) доступный для чтения и записи. Он используется для ввода новых входных данных при записи в регистр, и содержит результат предыдущего подсчёта CRC при чтении.

Каждая операция записи в регистр данных создаёт комбинацию предыдущего значения CRC (содержащегося в регистре DR) и нового. Подсчёт CRC выполняется над всеми 32-битными словами данных, полусловами или байтами, в зависимости от формата записываемых данных. Аппаратный алгоритм реализован методом побитового сдвига влево.

Доступ к регистру DR может быть по словам (32 бита), выровненным справа полусловам (16 бит) и выровненным справа байтам. Другие регистры блока CRC поддерживают любой доступ. Длительность вычисления зависит от ширины входных данных:

- 4 такта AHB для 32-битных входных данных;
- 2 такта AHB для 16-битных входных данных;
- 1 такт AHB для 8-битных входных данных.

Размер данных может быть динамически скорректирован для уменьшения количества операций записи в регистр данных DR в соответствии с необходимым количеством байт. Например, CRC для 5 байт исходных данных может быть подсчитана записью слова, а затем записью байта.

После завершения подсчёта CRC, результат помещается в регистр DR. Если используется пост-обработка (установлены биты XOR_OUT и/или REV_OUT в регистре CR), то результат вычисления CRC помещается в регистр DR, а результат после пост-обработки помещается в регистр POST.

Порядок следования бит внутри каждого байта, полуслова или слова целиком входных данных может быть изменён в соответствии со значением, записанным в битовое поле REV_IN[1:0] регистра CR.

Например, входные данные 0x1A2B3C4D для подсчёта CRC преобразуются в:

- 0x58D43CB2 с побайтовым обращением;
- 0xD458B23C с полусловным обращением;
- 0xB23CD458 с обращением по словам (четыре байта).

Выходные данные могут быть также обращены установкой бита REV_OUT в регистре CR. В этом случае, операция обращения исполняется на битовом уровне: к примеру, выходные данные 0x11223344 преобразуются в 0x22CC4488.

Бит XOR_OUT в регистре CR указывает, что результат будет подвергнут операции «XOR» (исключающее ИЛИ) со всеми единицами. Операция «XOR» производится над результатом после обращения выходных данных, если был установлен соответствующий бит в регистре CR, к примеру: результат из предыдущего примера 0x22CC4488 преобразуется в 0xDD33BB77.

Бит MODE в регистре CR устанавливает режим работы блока подсчёта CRC. При установке его в «1», входные данные после операции изменения порядка следования байтов (если была выбрана соответствующая опция) создают комбинацию (операция XOR – «Исключающее ИЛИ») с предыдущим содержимым регистра данных.

Таблица 25.1 – Влияние бита MODE на режим работы блока подсчёта CRC

Входные данные	Регистр данных после подсчёта	
	бит MODE сброшен	Бит MODE установлен
Первые: 0x0000 0000	0xc704 dd7b	0xc704 dd7b
Вторые: 0x1111 1111	0x7ab7 9290	0xbdb3 4feb

Параметры для расчета CRC указанные в таблице 25.1: REV_IN = 0x0, REV_OUT = 0x0, XOR_OUT = 0x0, стандартные значения регистров порождающего полинома и инициализации. Запись данных производится словами (32-бит).

В случае MODE, установленном в «1», вторые данные для подсчета используются 0xc704 dd7b (предыдущее содержимое регистра данных) ^ 0x1111 1111(новые входные данные) и равны 0xd615 cc6a.

Блок подсчёта CRC может быть инициализирован запрограммированным значением используя управляющий бит RESET в регистре CR (стандартное значение 0xFFFFFFFF).

Начальное значение для алгоритма CRC может быть запрограммировано в регистре INIT. Регистр DR автоматически инициализируется при записи в регистр INIT.

Программирование порождающего полинома

Коэффициенты порождающего полинома задаются в регистре POL, а их размер может быть 7, 8, 16 или 32-битным в зависимости от поля POLY_SIZE[1:0] в регистре CR. Чётные порождающие полиномы не поддерживаются.

Примечание – Чётный порождающий полином имеет коэффициенты X+X2+..+Xn, тогда как нечётный - 1+X+X2+..+Xn.

Если разрядность данных CRC меньше, чем 32-бит, значение может быть прочитано из младших разрядов регистра DR.

Для получения корректного значения CRC, во время вычисления не допускается изменение коэффициентов порождающего полинома или его размера. В результате, если вычисление CRC продолжается, необходимо выполнить сброс или прочитать регистр DR до изменения порождающего полинома. Начальное значение порождающего полинома CRC-32 (Ethernet) 0x4C11DB7.

26 Модуль криптографии CRYPTO

26.1 Общие положения

Модуль реализует следующие алгоритмы блочного шифрования:

- AES-128 (из стандарта FIPS PUB 197). Размер блока 128 бит, ключа – 128 бит;
- AES-256 (из стандарта FIPS PUB 197). Размер блока 128 бит, ключа – 256 бит;
- Кузнечик (из ГОСТ 34.12 2018). Размер блока 128 бит, ключа – 256 бит;
- Магма (из ГОСТ 34.12 2018). Размер блока 64 бит, ключа – 256 бит. В определенный момент времени может выполняться либо шифрование, либо дешифрование блока текста одним из реализованных алгоритмов.

Модуль может формировать прерывание при возникновении одного или нескольких отслеживаемых событий, о чем информирует регистр IRQ. Прерывания маскируются с помощью регистра IRQ_ENABLE и бита INTERRUPT слова управления DMA текущего дескриптора.

AES-128 и AES-256 используют общий участок памяти внутри модуля для хранения раундовых ключей. Выполнение криптографической операции означает инвалидацию ранее рассчитанных раундовых ключей для остальных алгоритмов, которые совместно используют упомянутое хранилище раундовых ключей.

Важно отметить, что блоки открытого текста, шифртекста, ключа и инициализационного вектора сохраняются в регистрах с обратным порядком байт (Big-Endian).

Описание регистров модуля криптографии приведено в приложении А.19.

26.2 Структура блока криптографии

Упрощенная функциональная схема блока криптографии показана на рисунке 26.1.

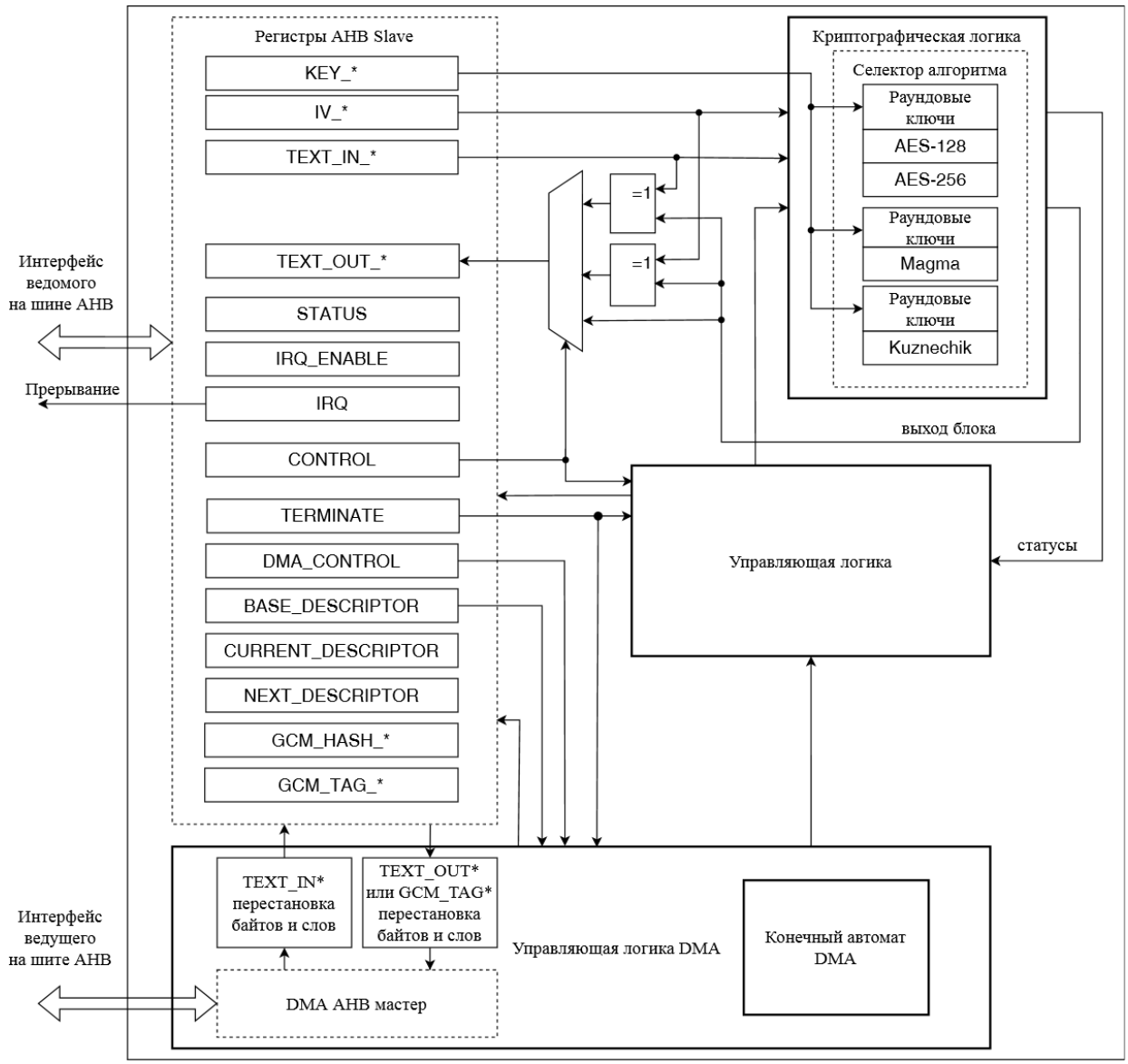


Рисунок 26.1 – Структурная схема блока криптографии

26.3 Режимы выполнения криптографических операций

Доступны следующие режимы:

- ECB – режим электронной кодовой книги (режим простой замены из ГОСТ 34.13 2018);
- CBC – режим простой замены с зацеплением» с параметром z = 1 ГОСТ 34.13 2018);
- CTR – режим гаммирования с параметром z = 1 из (ГОСТ 34.13-2018).
- GCM – режим счётчика с аутентификацией Галуа и его вырожденная версия, которая подразумевает только аутентификацию сообщения.

За выбор режима выполнения криптографической операции отвечает поле MODE регистра CONTROL.

Во всех случаях регистры KEY_* используются для формирования раундовых ключей выбранного алгоритма шифрования (если выбран алгоритм AES-128, то в формировании раундовых ключей участвуют только KEY_0, …, KEY_3).

При выполнении шифрования в регистры TEXT_IN_* записывается блок открытого текста, при дешифровании – блок зашифрованного текста. Если выбран алгоритм Magma, то значение TEXT_IN_2 и TEXT_IN_3 не влияют на результат выполнения криптографической операции.

При описании режимов выполнения криптографических операций используются следующие сокращения:

- P – открытый текст (plain text);
- C – зашифрованный текст (cipher text);
- IV – вектор инициализации текущего блока текста (формируется из значений регистров IV_*);
- Next_IV – вектор инициализации для следующего блока текста (данное значение автоматически будет записано в регистры IV_*);
- ^ – побитовая операция исключающего ИЛИ;
- E_in – входные данные криптографического блока в режиме шифрования (если используется алгоритм Magma, то в качестве входных данных будут использоваться старшие 64 бита, регистры с индексами 0 и 1);
- E_out – выходные данные криптографического блока в режиме шифрования (если используется алгоритм Magma, то результат будет дополнен справа 64 нулевыми битами);
- D_in – входные данные криптографического блока в режиме дешифрования;
- D_out – выходные данные криптографического блока в режиме дешифрования.

Режим электронной кодовой книги (ECB)

Режим электронной кодовой книги (ELECTRONIC_CODEBOOK, ECB). В ГОСТ 34.12-2018 он называется режимом простой замены.

Регистры IV_* не используются.

Шифрование (значение поля DIRECTION равно 0 – Encryption):

E_in = TEXT_IN = P;

C =TEXT_OUT = E_out.

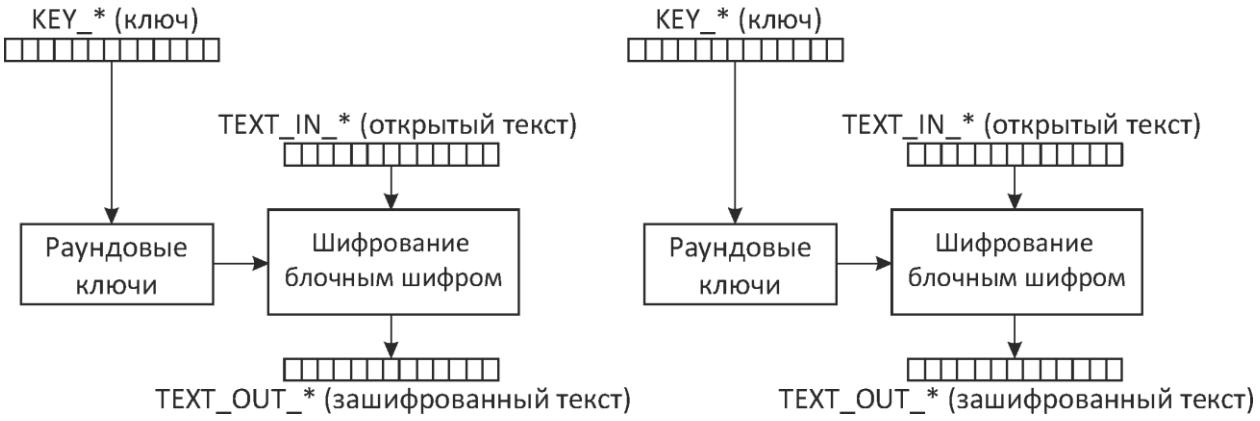


Рисунок 26.2 – Шифрование в режиме ECB

Дешифрование (значение поля DIRECTION равно 1 – Decryption):

D_in = TEXT_IN = C;

P = TEXT_OUT = D_out.

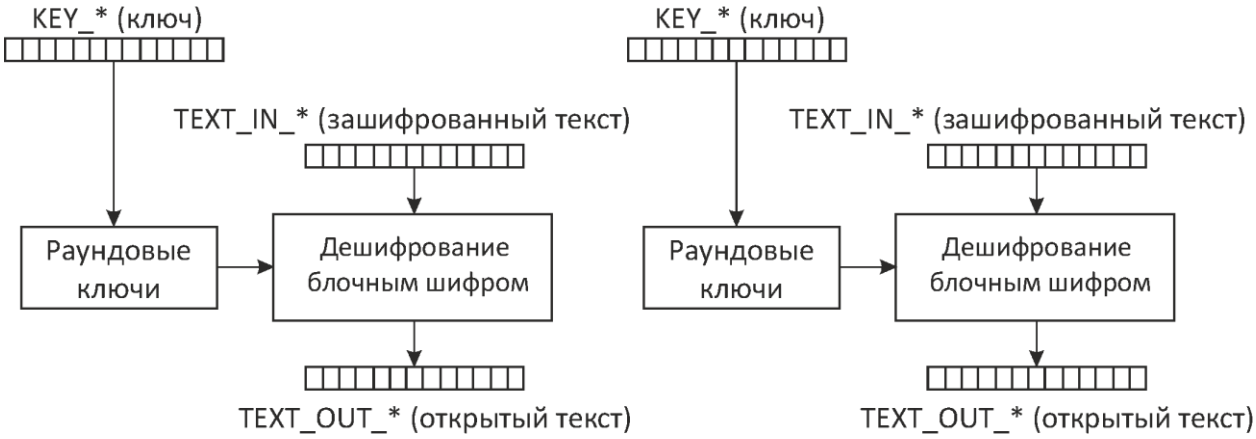


Рисунок 26.3 – Дешифрование в режиме ECB

Режим сцепления блоков шифротекста (CBC)

Режим сцепления блоков шифротекста (CIPHER_BLOCK_CHAINING, CBC). В ГОСТ 34.13-2018 он называется режимом простой замены с зацеплением (для случая, когда параметр z = 1).

Данный криптографический режим может быть использован для организации аутентификации сообщения (защиты его целостности), именуемой CBC-MAC (message authentication code). В качестве MAC, также называемого аутентификационным тегом или имитовставкой, берётся последний блок сообщения, зашифрованного блочным алгоритмом в режиме CBC.

Шифрование (значение поля DIRECTION равно 0 – Encryption):

E_in = TEXT_IN ^ IV = P ^ IV;

C =TEXT_OUT = E_out;

Next_IV = C = TEXT_OUT.

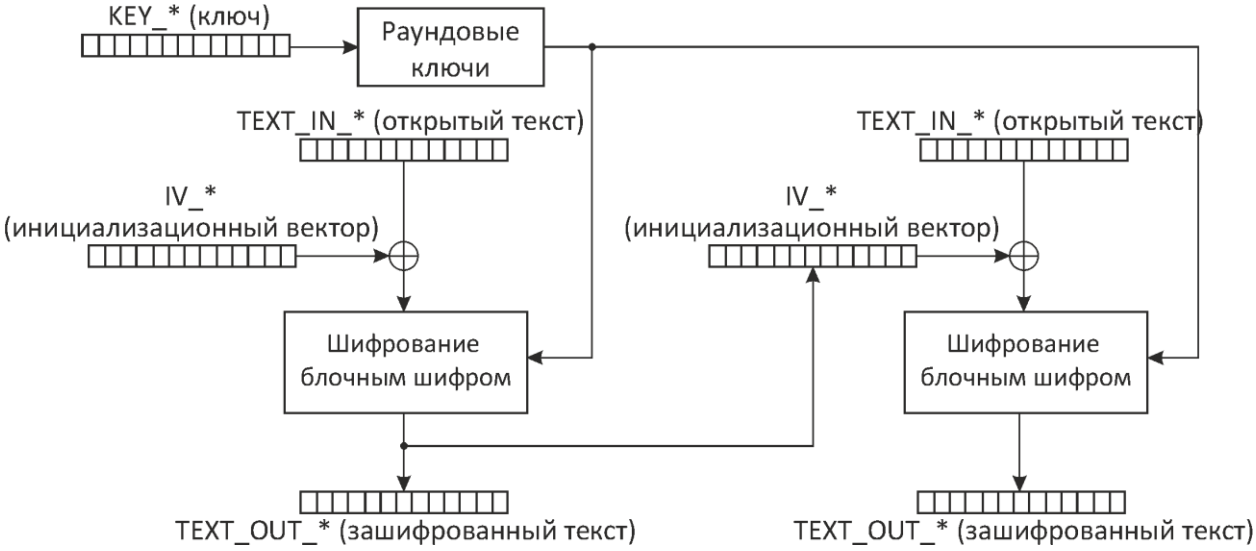


Рисунок 26.4 – Шифрование в режиме CBC

Дешифрование (значение поля DIRECTION равно 1 – Decryption):
D_in = TEXT_IN = C;
P = TEXT_OUT = D_out ^ IV;
Next_IV = C = TEXT_IN.

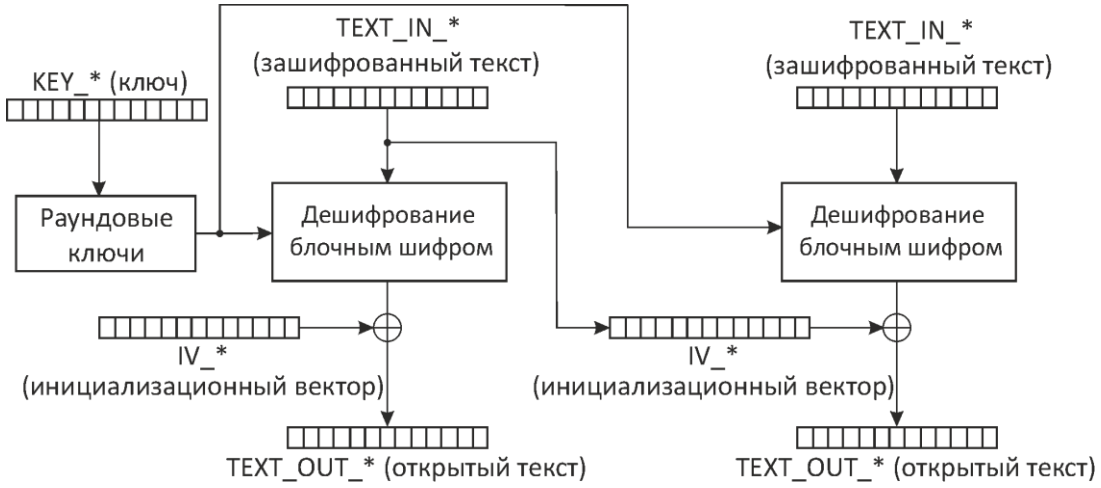


Рисунок 26.5 – Дешифрование в режиме CBC

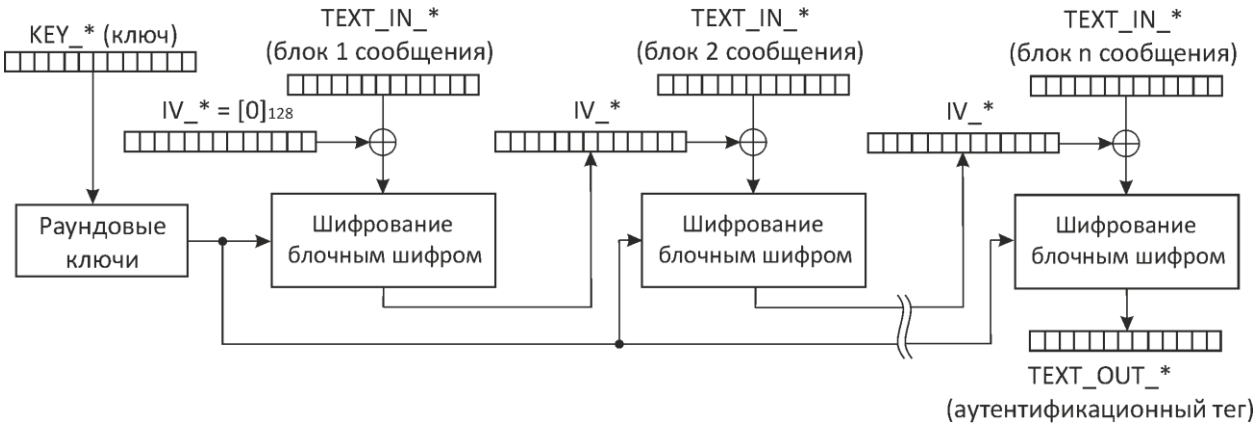


Рисунок 26.6 – Формирование аутентификационного тега сообщения (CBC-MAC)

Режим счётчика (CTR)

Режим счётчика (COUNTER_MODE, CTR). В ГОСТ 34.13-2018 он называется режимом гаммировани (для случая, когда параметр z = 1).

Шифрование (значение поля DIRECTION равно 0 – Encryption):

E_in = IV;

C = TEXT_OUT = E_out ^ TEXT_IN = E_out ^ P;

Next_IV = IV + 1 (если алгоритм – Magma, то инкрементируется IV_1, в остальных случаях IV_3).

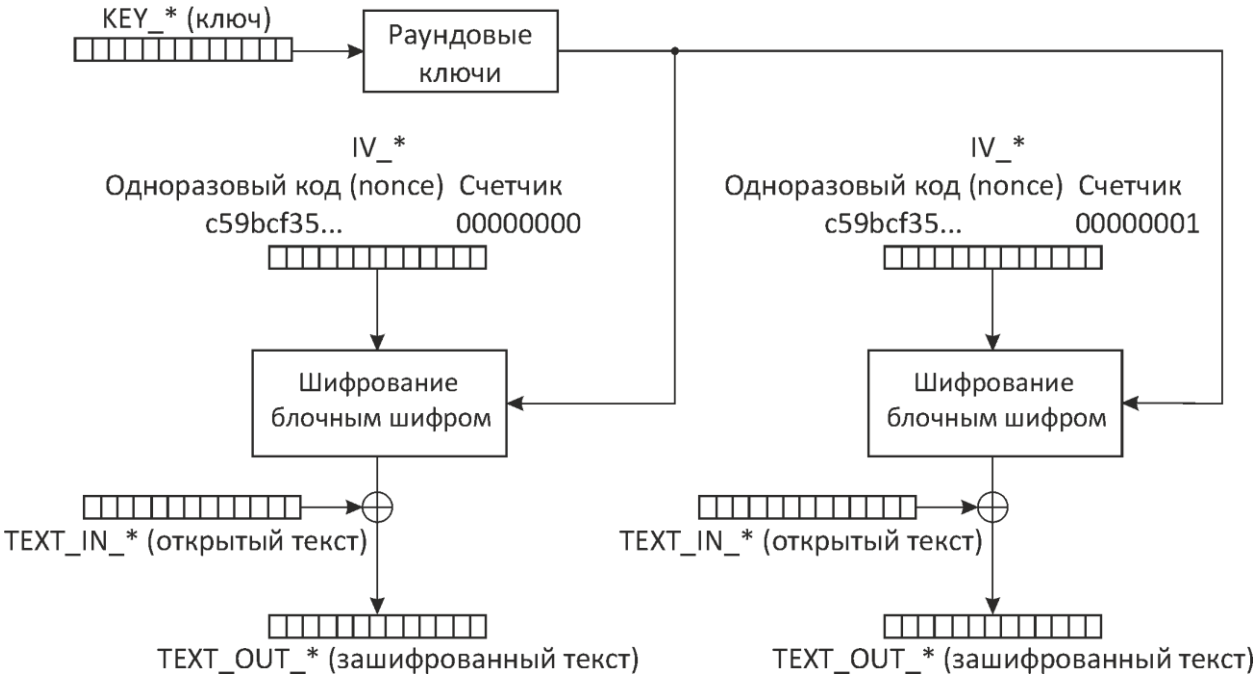


Рисунок 26.7 – Шифрование в режиме CNT

Дешифрование (значение поля DIRECTION равно 1 – Decryption):

E_in = IV;

P = TEXT_OUT = E_out ^ TEXT_IN = E_out ^ C;

Next_IV = IV + 1 (если алгоритм – Magma, то инкрементируется IV_1, в остальных случаях IV_3).

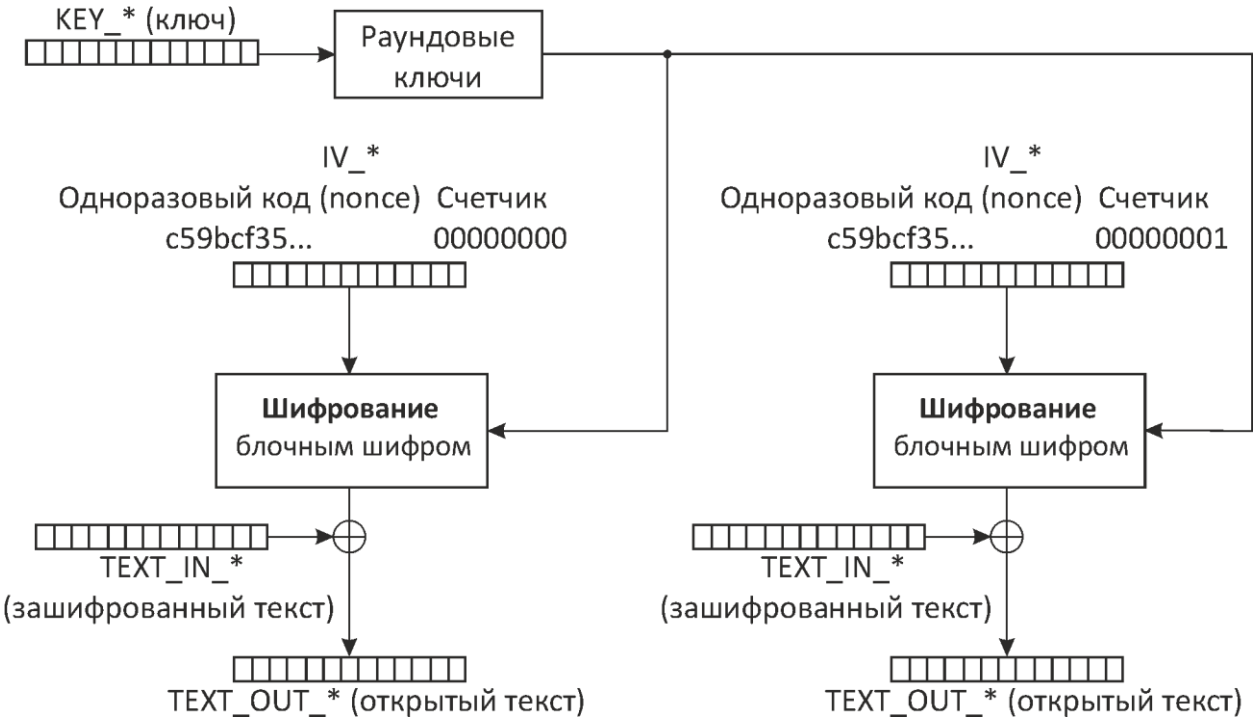


Рисунок 26.8 – Дешифрование в режиме CNT

Режим счётчика с аутентификацией Галуа (GCM)

Шифрование и дешифрования блоков текста в режиме GCM аналогично таковому в режиме счетчика (CTR). Данный режим является режимом аутентифицированного шифрования (Authenticated Encryption with Associated Data, AEAD). AEAD – класс блочных режимов шифрования, при котором часть сообщения шифруется, часть остается открытой, и всё сообщение целиком аутентифицировано (при совпадении значения аутентификационного тега сообщений гарантируется неизменность количества блоков текста, их содержимого и порядка следования в сообщении).

Выполнение расчета аутентификационного тега разбито на четыре фазы, которые должны выполняться последовательно:

– GCM_INIT. Обнуляется значения регистров GCM_TAG_*. На основе текущего состояния регистров KEY_* производится расчет хеш-ключа, который будет использован в качестве множителя при выполнении умножения в поле Галуа GF(2^{128}). Значение хеш-ключа отображается в GCM_HASH_*;

– GCM_HEADER. В регистры TEXT_IN_* помещается блок дополнительных аутентифицируемых данных (Additional authenticated data – AAD). В GCM_TAG_* попадает результат выполнения ИСКЛЮЧАЮЩЕГО ИЛИ над текущим значением тэга и произведением AAD на хеш-ключ;

– GCM_PAYLOAD. В регистры TEXT_IN_* помещаются открытый текст/шифрованный текст, который будет зашифрован/расшифрован соответственно. В GCM_TAG_* попадает результат выполнения ИСКЛЮЧАЮЩЕГО ИЛИ над текущим значением тэга и произведением блока шифрованного текста на хеш-ключ;

– GCM_LAST_BLOCK. В регистры TEXT_IN_0, TEXT_IN_2 заносятся нули. В TEXT_IN_1 – количество бит AAD, которые обрабатываются в фазе GCM_HEADER. В TEXT_IN_3 – количество шифруемых/дешифруемых бит полезной нагрузки, которые обрабатываются в фазе GCM_PAYLOAD. В GCM_TAG_* попадает результат выполнения ИСКЛЮЧАЮЩЕГО ИЛИ над текущим значением тэга, произведением TEXT_IN_* на хеш-ключ, результатом шифрования значения IV_*, в котором IV_3 подменяется на 0x00000001.

Ниже приведены рисунки, которые иллюстрируют состав сообщения, операции над которым производятся в режиме GCM согласно стандарту NIST SP 800-38D.

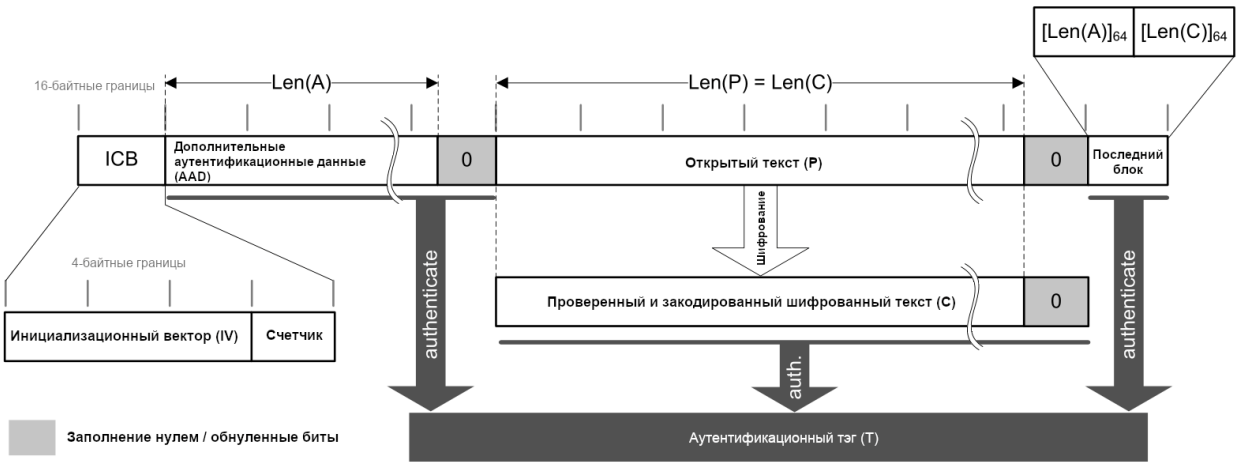


Рисунок 26.9 – Структура шифруемого сообщения в режиме GCM

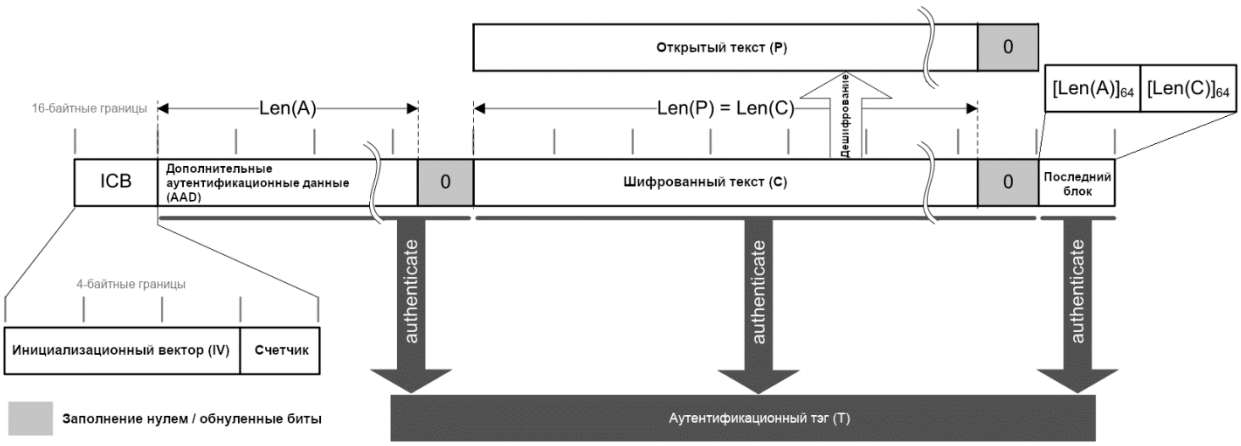


Рисунок 26.10 – Структура дешифруемого сообщения в режиме GCM

Регистры IV_* формируют начальное значение счетчика (initial counter block, ICB), состоящее из двух частей:

– Регистры IV_0, IV_1, IV_2 составляют 96-битное значение инициализационного вектора, которое должно вместе с ключом составлять уникальную для каждого шифруемого сообщения пару. Стоит отметить, что стандарт GCM поддерживает использование вектора инициализации меньшей длины, но в данном случае применяется точное значение в 96 бит;

– Регистр IV_3, который содержит счетчик, инкрементируемый каждый раз после завершения шифрования/дешифрования блока текста. Согласно стандарту NIST SP 800-38D значение счетчика при обработке первого блока текста – 0x00000002. Данное значение будет занесено в IV_3 после выполнения фазы GCM_INIT, если был установлен бит SELF_UPDATE регистра CONTROL или выполнение фазы является результатом работы в поточном режиме.

Неправильное использование инициализационного вектора (IV) может негативно сказаться на безопасности GCM. Например, повторное использование пары ключа и IV приведет к потере конфиденциальности сообщений, в которых использовалось одно и то же значение IV. Повторное использование IV отправителем может позволить злоумышленнику идентифицировать ключ. Это может привести к подделке сообщений. Рекомендации по генерации инициализационного вектора приведены в NIST SP 800-38D (в разделе «8 Uniqueness Requirement on IVs and Keys»).

Если длина AAD не кратна 128 бит, пользователь должен дополнить нулями последний блок дополнительных аутентифицируемые данных.

Если длина дешифруемого текста не кратна 128 бит, пользователь должен дополнить нулями последний блок. Обнуление неактуальных бит последнего блока открытого текста не предусмотрено, их значение следует игнорировать.

Важно отметить, что модуль не накладывает маску на последний зашифрованный блок текста. Поэтому шифрование сообщений с длиной полезной нагрузки, которая не кратна 128 бит, не поддерживается (будет получен некорректный аутентификационный тег).

Ниже приведены рисунки, которые иллюстрируют процесс шифрования и дешифрования сообщения в режиме GCM.

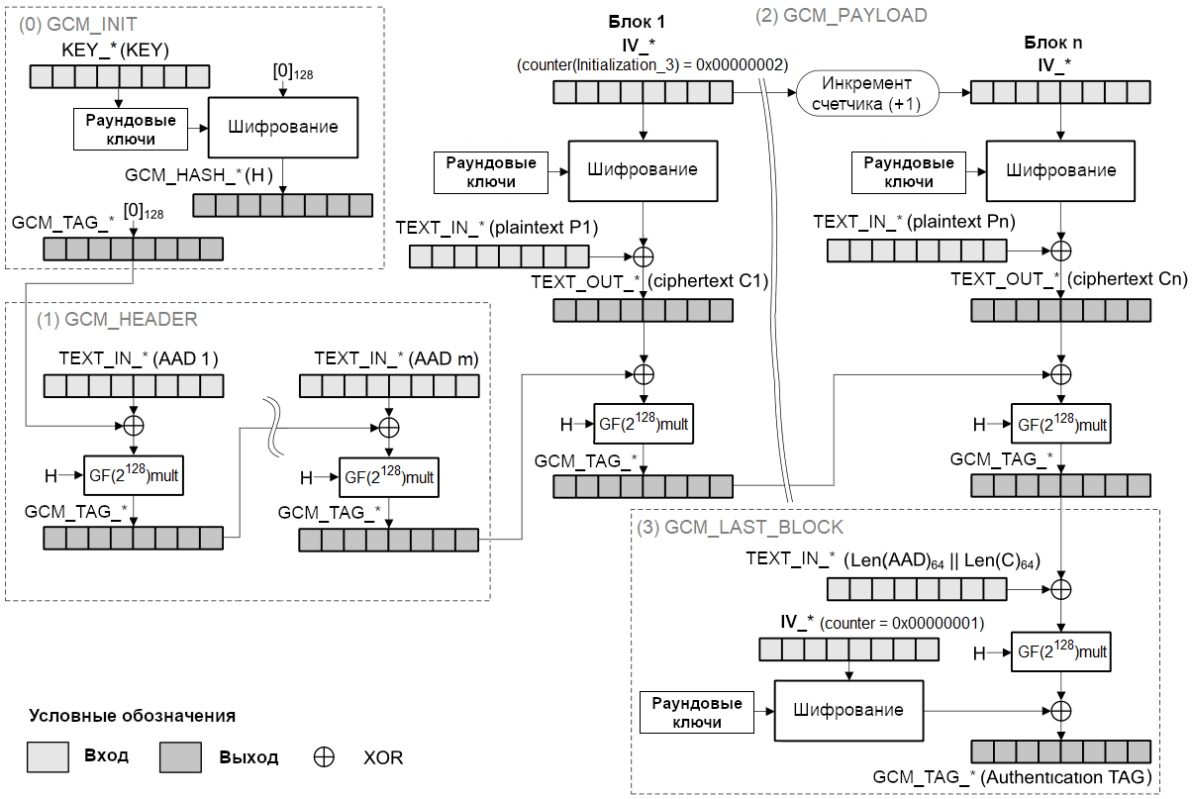


Рисунок 26.11 – Шифрование сообщения в режиме GCM

Дешифрование в фазе GCM_PAYLOAD выполняется параллельно с получением блока открытого текста. Поэтому суммарно на обработку сообщения будет потрачено меньше времени, чем при его шифровании.

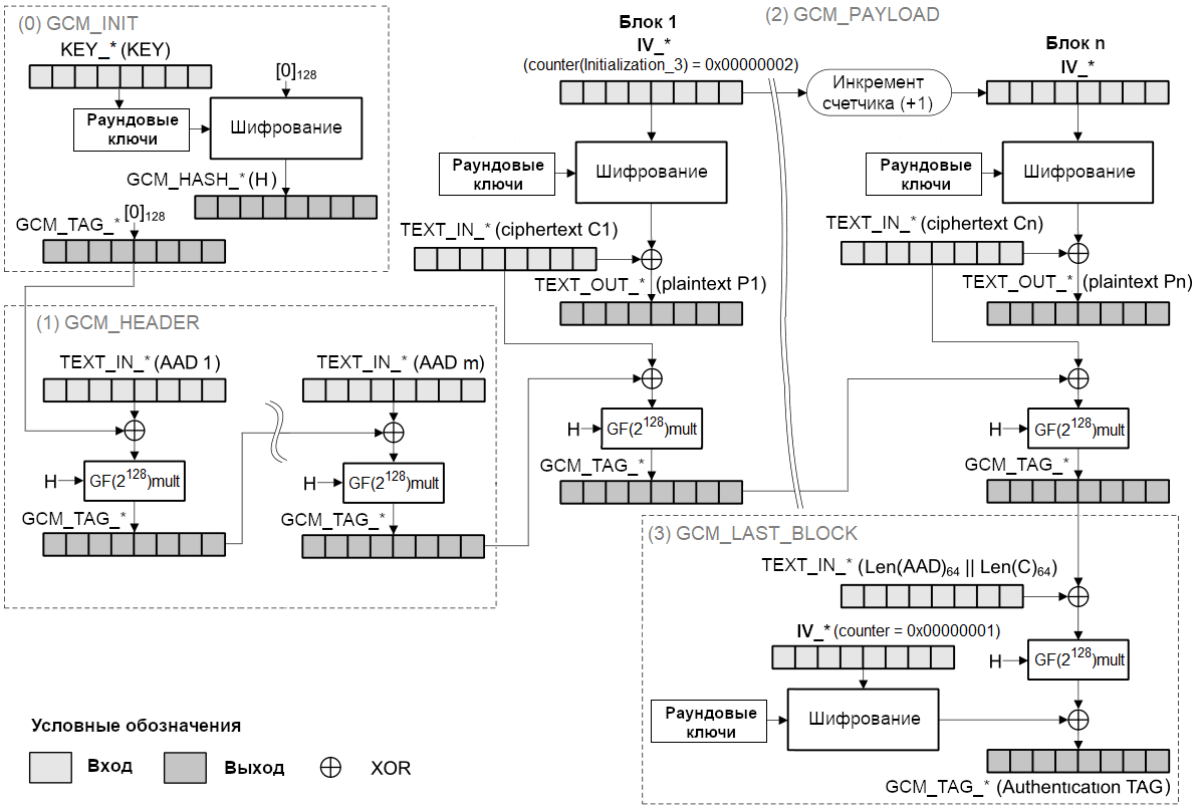


Рисунок 26.12 – Дешифрование сообщения в режиме GCM

Вырожденным случаем GCM является код аутентификации сообщения на основе полей Галуа (Galois message authentication code – GMAC). Процесс получения тега аналогичен таковому для обычного GCM, в котором пропущена фаза GCM_PAYLOAD.

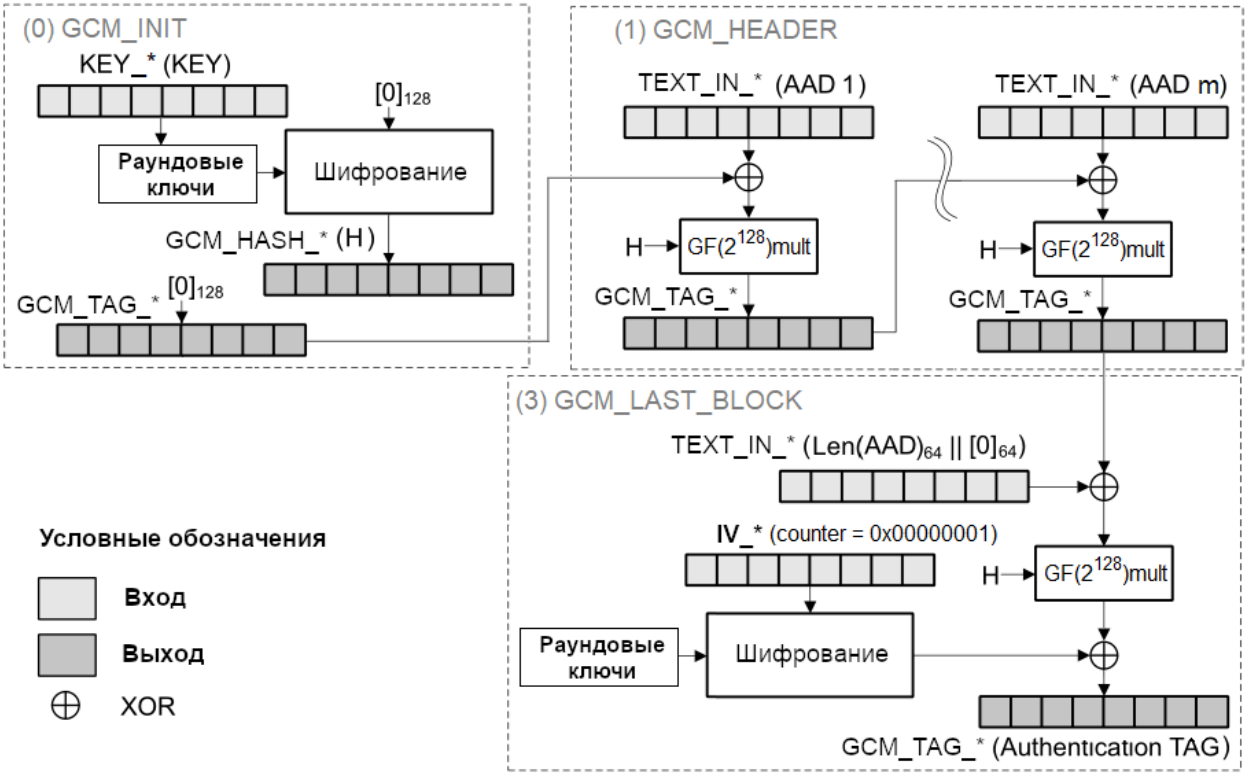


Рисунок 26.13 – Формирование аутентификационного тега в режиме GMAC

Неподдерживаемые конфигурации режима GCM

Запись в регистр CONTROL значения MODE равного GALOIS_COUNTER_MODE в сочетании с ALGORITHM соответствующим MAGMA блокирует запуск криптографической операции (установленный бит START регистра CONTROL). В случае использования режима с прямым доступом к памяти (DMA), если в слове управления DMA выставлен GALOIS_COUNTER_MODE в качестве MODE и значение ALGORITHM соответствует MAGMA, выполнение дескриптора завершится с ошибкой. О данном факте будет свидетельствовать значение поля BAD_DESCR равный 0001_2 (CONTROL) регистра STATUS и, если был установлен бит DMA_FAIL в регистре IRQ_ENABLE, установка бита прерывание DMA_FAIL в регистре IRQ.

Как оговорено ранее, в режиме GALOIS_COUNTER_MODE не поддерживается шифрование сообщений с длиной полезной нагрузки (блоков шифрованного текста), которая не кратна 128 бит.

26.4 Режимы работы

Общие положения

В криптографическом модуле реализованы режим работы с одним блоком текста и поточный режим. Последний из них также можно назвать режимом работы с использованием прямого доступа к памяти (DMA).

Регистры IV _*, KEY_*, STATUS, IRQ используются одинаково в обоих режимах. Регистры BASE_DESCRIPTOR, CURRENT_DESCRIPTOR, NEXT_DESCRIPTOR используются только в режиме работы с DMA.

Важно понимать, что в TEXT_OUT_* не хранится результат криптографической операции (зашифрованный текст/открытый текст). Отображаемое в данную область

значение зависит от состояния полей MODE, DIRECTION и ALGORITHM регистра CONTROL (описание полей приведено в «Описании режима работы с одним блоком»). Для режимов криптографической операции отличных от ELECTRONIC_CODEBOOK корректность TEXT_OUT_* после завершения операции (зашифрованный текст для шифрования/открытый текст для дешифрования) гарантируется только при условии неизменности IV_*/TEXT_IN_* и вышеупомянутых полей регистра CONTROL. В таблице ниже приведена зависимость отображаемого значения от MODE и DIRECTION (см. «Режимы выполнения криптографических операций»).

Таблица 26.1 – Значение, отображаемое в регистры TEXT_OUT_*

Поля регистра CONTROL	Значение TEXT_OUT_*
MODE = 1 (CIPHER_BLOCK_CHAINING), DIRECION = 1 (Decrypt)	algorithm_result ^ IV_*
MODE = 2 (COUNTER_MODE) или 3 (GALOIS_COUNTER_MODE)	algorithm_result ^ TEXT_IN_*
В остальных случаях	algorithm_result

В регистре STATUS не хранится значение поля KEYS_READY, в него отображается соответствующий бит поля KEYS_STORED текущего выбранного алгоритма, который определяется значением поля CONTROL.ALGORITHM. Поле TEXT_OUT_VALID имеет схожую зависимость от CONTROL.ALGORITHM. Установленное значение TEXT_OUT_VALID означает, что выходные данные активного криптографического подмодуля не изменились с момента завершения последней операции шифрования/дешифрования.

Модуль в состоянии хранить до 3 результатов шифрования/дешифрования входных данных (TEXT_IN_*, IV_* или TEXT_IN_* ^ IV _* в зависимости от режима выполнения криптографической операции, задаваемого CONTROL.MODE). По одному на криптографический подмодуль. Их 3, поскольку AES-128 и AES-256, объединены в один подмодуль.

Регистр STATUS (структура регистров и битовых полей блока криптографии приведена в приложении А.19).

DMA_LAST – Поле отображает значение бита LAST слова управления DMA последнего прочитанного дескриптора. Бит установлен, если прочитанный дескриптор является последним в цепочке дескрипторов. Значение обновляется при считывании нового дескриптора. Бит сбрасывается при экстренном прекращении выполнения операции.

DMA_COUNTER – Поле отображает количество обработанных DMA блоков текст. Значение обнуляется при считывании нового дескриптора. Значение сбрасывается при экстренном прекращении выполнения операции. Предупреждение: если в слове управления DMA в поле COUNT записано максимально возможное значение, то после обработки последнего блока текста произойдет переполнение счетчика, который отображается в DMA_COUNTER.

BAD_DESCR – Поле хранит сведения об обнаруженных некорректных словах в прочитанном дескрипторе DMA. Список возможных значений:

0000_2 – NO_ERROR;

0001_2 – CONTROL:

0010_2 – TEXT_IN_ADDRESS;

0100_2 – TEXT_OUT_ADDRESS;

1000_2 – NEXT_DESCRIPTOR.

После записи в регистр DMA_CONTROL любого значения поле будет переведено в состояние NO_ERROR.

AHB_ERROR – Поле хранит сведения об области памяти, на которую пришелся ответ ERROR на транзакцию по шине AHB. Список возможных значений:

000_2 – NO_ERROR;

001_2 – READ_DESCRIPTOR;

010_2 – READ_TEXT_IN;

100_2 – WRITE_TEXT_OUT.

После записи в регистр DMA_CONTROL любого значения поле будет переведено в состояние NO_ERROR.

DMA_ACTIVE – Если бит установлен, то конечный автомат DMA занят выполнением дескриптора. Бит READY будет находиться в сброшенном состоянии, пока DMA_ACTIVE установлен. Бит сбрасывается после выполнения цепочки дескрипторов, обнаружения некорректного дескриптора, получения ответа ERROR по шине AHB или при экстренном прекращении выполнения операции.

KEYS_STORED – Каждый бит данного поля соответствует заданному криптографическому алгоритму. Биты нумеруются справа налево. 0 – AES-128, 1 – AES-256, 2 – MAGMA, 3 – KUZNECHIK. Если бит, который относится к интересующему криптографическому алгоритму, установлен, то в модуле хранятся его раундовые ключи. Бит сбрасывается при старте обновления раундовых ключей заданного алгоритма, в случае, когда одновременное хранение раундовых ключей не предусмотрено (это актуально для пары алгоритмов AES-128 и AES-256), при экстренном прекращении выполнения операции.

IRQ_PENDING – Если бит установлен, то регистр IRQ содержит хотя бы 1 не сброшенный бит.

TEXT_OUT_VALID – Если бит установлен, то на выходе криптографического подмодуля находится результат шифрования/дешифрования входных данных (TEXT_IN_*, IV_* или TEXT_IN_* ^ IV_* в зависимости от алгоритма и режима выполнения криптографической операции). Для операций кроме дешифрования в режиме CIPHER_BLOCK_CHAINING, режимах COUNTER_MODE и GALOIS_COUNTER_MODE установленный бит TEXT_OUT_VALID эквивалентен тому, что регистры TEXT_OUT содержат результат выполнения предыдущей криптографической операции. Для обоих режимов счетчика для корректности результата требуется еще и неизменность значения TEXT_IN_*, для дешифрования в режиме CIPHER_BLOCK_CHAINING – IV_*. Выполнение обновления раундовых ключей не оказывает на данный бит влияния (например, запись в регистр CONTROL установленных бит UPDATE_KEY и START сбросит данный бит только после перерасчета раундовых ключей). Бит сбрасывается при старте выполнения криптографической операции (запись в регистр CONTROL установленного бита START) или при экстренном прекращении выполнения операции.

KEYS_READY – Если бит установлен, то раундовые ключи текущего активного криптографического алгоритма сгенерированы. Бит сбрасывается при старте обновления раундовых ключей или при экстренном прекращении выполнения операции.

READY – Если бит установлен, то модуль готов к выполнению очередной криптографической операции и/или обновлению раундовых ключей шифрования. Бит сбрасывается при старте обновления раундовых ключей, выполнения криптографической операции и считывания первого дескриптора цепочки. Бит устанавливается после завершения всех запланированных действий или при экстренном прекращении выполнения операции.

Регистр IRQ_ENABLE

DMA_FAIL – Если бит установлен, то функционал бита IRQ.DMA_FAIL разрешен.

DMA_DONE – Если бит установлен, то функционал бита IRQ.DMA_DONE разрешен. Бит доступен только для чтения и отображает значение бита INTERRUPT управляющего слова дескриптора DMA.

WR_IGNORED – Если бит установлен, то функционал бита IRQ.WR_IGNORED разрешен.

DONE – Если бит установлен, то функционал бита IRQ.DONE разрешен.

Регистр IRQ

DMA_FAIL – Бит устанавливается, если установлен соответствующий бит регистра IRQ_ENABLE, при преждевременном завершении работы DMA из-за обнаружения ошибки (см. поля AHB_ERROR и BAD_DESCR регистра STATUS). Бит может быть сброшен записью 1.

DMA_DONE – Бит устанавливается, если бит INTERRUPT управляющего слова текущего дескриптора DMA был установлен (его состояние отображается в соответствующий бит регистра IRQ_ENABLE), при успешном завершении выполнения дескриптора DMA (поля STATUS.BAD_DESCR и STATUS.AHB_ERROR содержат нулевое значение). Бит может быть сброшен записью 1.

WR_IGNORED – Бит устанавливается, если установлен соответствующий бит регистра IRQ_ENABLE, при выполнении записи в регистр CONTROL, TEXT_IN_* или IV_* пока STATUS.READY сброшен. Бит может быть сброшен записью 1.

DONE – Бит устанавливается, если установлен соответствующий бит регистра IRQ_ENABLE, при переходе в режим готовности (переключение STATUS.READY в 1). Если запись в регистр CONTROL была выполнена с одновременно установленными битами UPDATE_KEY и START, то прерывание будет сгенерировано однократно и только после готовности выходных данных. Если сброс бита STATUS.READY вызван записью установленного бита START регистра DMA_CONTROL, то прерывание будет сгенерировано однократно после успешного завершения всей цепочки дескрипторов, либо при обнаружении некорректного дескриптора (ненулевое значение STATUS.BAD_DESCR), либо в результате возникновения ошибки на шине AHB (ненулевое значение STATUS.AHB_ERROR). Бит может быть сброшен записью 1.

Описание режима работы с одним блоком

Запуск режима работы с одним блоком текста осуществляется с помощью регистра CONTROL.

Запись в регистр CONTROL, DMA_CONTROL, TEXT_IN_* или IV_* будет проигнорирована, если бит STATUS.READY сброшен. Если предварительно был установлен бит IRQ_ENABLE.WR_IGNORED, то будет сгенерировано соответствующее прерывание (INTERRUPT.WR_IGNORED).

Для получения корректного результата криптографической операции необходимо иметь предварительно рассчитанные для выбранного алгоритма (поле ALGORITHM регистра CONTROL) раундовые ключи. Их расчёт может быть выполнен как отдельное действие (CONTROL.START = 0, CONTROL.UPDATE_KEY = 1), либо как предварительный этап криптографической операции (CONTROL.START = 1, CONTROL.UPDATE_KEY = 1). Если для текущего алгоритма нет рассчитанных раундовых ключей (см. соответствующий бит поля KEYS_STORED регистра STATUS), то автоматически будет выполнен перерасчет раундовых ключей на основе текущего значения регистров KEY_*.

Регистр CONTROL

GCM_PHASE – Фаза обработки пакета в режиме GALOIS_COUNTER_MODE:

0 – GCM_INIT;

1 – GCM_HEADER;

2 – GCM_PAYLOAD;

3 – GCM_LAST_BLOCK.

Если режим выполнения криптографической операции не GALOIS_COUNTER_MODE, то значение поля игнорируется.

SELF_UPDATE – Разрешение автоматического обновления содержимого регистров IV_* при завершении выполнения криптографической операции. После шифрования в режиме CIPHER_BLOCK_CHAINING в IV_* будет записано содержимое TEXT_OUT_*, после дешифрования – TEXT_IN_* (что повлияет на считываемое из регистров TEXT_OUT_* значение, как это описано в замечании ниже). Для режима COUNTER_MODE на 1 будет увеличено значение или регистра IV_1 для алгоритма MAGMA, или IV_3 для остальных случаев. Для режима GALOIS_COUNTER_MODE в фазе GCM_INIT в регистр IV_3 будет занесено значение 0x00000002. Для режима GALOIS_COUNTER_MODE в фазах GCM_HEADER или GCM_PAYLOAD на 1 будет увеличено значение регистра IV_3.

Замечание: если CONTROL.SELF_UPDATE был установлен при запуске дешифрования в режиме CIPHER_BLOCK_CHAINING, то для получения расшифрованного блока текста потребуется вычислить (TEXT_OUT_* ^ IV_* ^ предыдущее значение IV_*), где ^ – операция побитового ИЛИ (см. «Режим сцепления блоков шифротекста CBC»). Альтернативным способом будет являться переключение DIRECTION в 0 или MODE в 0 с последующим вычислением (TEXT_OUT_* ^ предыдущее значение IV_*).

MODE – Выбор режима выполнения криптографической операции:

0 – ELECTRONIC_CODEBOOK (ECB);

1 – CIPHER_BLOCK_CHAINING (CBC);

2 – COUNTER_MODE (CTR);

3 – GALOIS_COUNTER_MODE (GCM).

ALGORITHM – выбора алгоритма криптографической операции (00_2 – AES_128, 01_2 – AES_256, 10_2 – MAGMA, 11_2 – KUZNECHIK).

DIRECTION – направление криптографической операции (0 – Encryption, 1 – Decryption). При выполнении операции шифрования (Encryption) значение регистров {TEXT_IN_0, ..., TEXT_IN_3} или {TEXT_IN_0, TEXT_IN_1} для алгоритма MAGMA будут интерпретированы модулем как открытый текст (plain text), а при завершении выполнения в соответствующие регистры TEXT_OUT попадет шифрованный текст (cipher text). В случае дешифрования (Decryption) TEXT_IN будет считаться шифрованным текстом, а в TEXT_OUT будет помещен дешифрованный открытый текст.

START – Запрос на выполнение криптографической операции. Записываемое в данный бит значение будет проигнорировано, если в поле MODE записывается значение, которое соответствует GALOIS_COUNTER_MODE, а в поле ALGORITHM – MAGMA. Непосредственный старт может быть отложен до завершения обновления раундовых ключей (спровоцированное установленным битом UPDATE_KEY или выполненное из-за отсутствия раундовых ключей для выбранного криптографического алгоритма). При чтении всегда возвращает 0.

UPDATE_KEY – Принудительно выполнить обновление раундовых ключей шифрования на основе текущего значения регистров {KEY_0, ..., KEY_7} или {KEY_0, ..., KEY_3} для алгоритма AES-128. Записываемое в данный бит значение будет проигнорировано, если в поле MODE записывается значение, которое соответствует

GALOIS_COUNTER_MODE, а в поле ALGORITHM – MAGMA. При чтении всегда возвращает 0.

Описание поточного режима работы

Размещение в памяти любого из дескрипторов должно иметь выравнивание по границе в 4 слова. В регистр BASE_DESCRIPTOR можно записать только корректный адрес дескриптора, поскольку младшие 4 бита доступны только для чтения.

Запуск потокового режима работы осуществляется с помощью регистра DMA_CONTROL.

Запись в регистр DMA_CONTROL, CONTROL, TEXT_IN_* или IV_* будет проигнорирована, если бит STATUS.READY сброшен. Если предварительно был установлен бит IRQ_ENABLE.WR_IGNORED, то будет сгенерировано соответствующее прерывание (IRQ.WR_IGNORED).

Для получения корректного результата криптографической операции необходимо иметь предварительно рассчитанные для выбранного алгоритма раундовые ключи. Если таковых нет для заданного в слове управления DMA поля ALGORITHM, будет автоматически инициирован их расчет, даже при сброшенном бите UPDATE_KEY. Если набор операций, заданный дескриптором, должен быть осуществлен с использованием нового ключа, то необходимо установить бит UPDATE_KEY.

Регистр управления DMA (**DMA_CONTROL**)

WORDS_SWAP – Если бит сброшен, то заполнение регистров TEXT_IN_* считанными из памяти словами входного текста идет в порядке увеличения индекса регистра. То есть данные попадают в регистр TEXT_IN_0, следующее слово (адрес которого больше на 4) в TEXT_IN_1 и так далее для всех алгоритмов кроме «Магма», в котором регистры TEXT_IN_2 и TEXT_IN_3 не используются. Если бит установлен, то последовательность заполнения регистров TEXT_IN_* меняется на обратную (от TEXT_IN_1 к TEXT_IN_0 для алгоритма MAGMA, от TEXT_IN_3 к TEXT_IN_0 в остальных случаях). Аналогичное влияние оказывается на пересылку выходных слов текста из регистров TEXT_OUT_* в память. Данный бит не меняет порядок считывания слов дескриптора.

BYTES_SWAP – Если бит установлен, то перед записью в регистры TEXT_IN_* в прочитанном слове будет изменен порядок следования байт, пересылаемые из TEXT_OUT_* слова будут выставлены на шину AHB с инвертированным порядком байт. Данный бит не оказывает влияние на считываемые слова дескриптора.

START – Установленный бит запускает последовательность операций в режиме прямого доступа к памяти. При чтении всегда возвращает 0.

Ниже приведены варианты размещения байт данных в регистрах TEXT_IN_* для оперативной памяти с порядком байт от младшего к старшему (в которой байт 0xD4 со смещением адреса 0x0, байт 0xC3 со смещением 0x1, байт 0xB2 со смещением 0x2 и байт 0xA1 со смещением 0x3 будут образовывать число 0xA1B2C3D4) и размера блока текста в 128 бит.

Размещение байт в TEXT_IN [127 : 0] / TEXT_OUT [127 : 0] при WORDS_SWAP = 0, BYTES_SWAP = 0:

TEXT_IN_0 [31:0]				TEXT_IN_1 [31:0]				TEXT_IN_2 [31:0]				TEXT_IN_3 [31:0]			
0x3	0x2	0x1	0x0	0x7	0x6	0x5	0x4	0xB	0xA	0x9	0x8	0xF	0xE	0xD	0xC

Размещение байт в TEXT_IN [127 : 0] / TEXT_OUT [127 : 0] при WORDS_SWAP = 0, BYTES_SWAP = 1:

TEXT_IN_0 [31:0]				TEXT_IN_1 [31:0]				TEXT_IN_2 [31:0]				TEXT_IN_3 [31:0]			
0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF

Размещение байт в TEXT_IN [127 : 0] / TEXT_OUT [127 : 0] при WORDS_SWAP = 1, BYTES_SWAP = 0:

TEXT_IN_0 [31:0]				TEXT_IN_1 [31:0]				TEXT_IN_2 [31:0]				TEXT_IN_3 [31:0]			
0xF	0xE	0xD	0xC	0xB	0xA	0x9	0x8	0x7	0x6	0x5	0x4	0x3	0x2	0x1	0x0

Размещение байт в TEXT_IN [127 : 0] / TEXT_OUT [127 : 0] при WORDS_SWAP = 1, BYTES_SWAP = 1:

TEXT_IN_0 [31:0]				TEXT_IN_1 [31:0]				TEXT_IN_2 [31:0]				TEXT_IN_3 [31:0]			
0xC	0xD	0xE	0xF	0x8	0x9	0xA	0xB	0x4	0x5	0x6	0x7	0x0	0x1	0x2	0x3

Регистр базового адреса дескриптора DMA (**BASE_DESCRIPTOR**)

ADDRESS – Адрес, который указывает на первый дескриптор DMA в цепочке, который будет прочитан из памяти после записи 1 в бит START регистра DMA_CONTROL.

Регистр активного дескриптора DMA (**CURRENT_DESCRIPTOR**)

ADDRESS – Адрес, который указывает на текущий выполняемый дескриптор. При записи 1 в бит START регистра DMA_CONTROL в данный регистр попадает значение из регистра BASE_DESCRIPTOR. Если значение бита LAST слова управления DMA текущего дескриптора сброшен, то при успешном завершении текущего дескриптора в CURRENT_DESCRIPTOR заносится значение NEXT_DESCRIPTOR.

Регистр следующего дескриптора DMA (**NEXT_DESCRIPTOR**)

ADDRESS – Адрес, который указывает на следующий дескриптор цепочки. Значение регистра будет сброшено, когда в слове управления текущего дескриптора DMA будет установлен бит LAST.

Из дескрипторов может быть составлен однонаправленный список (цепочка дескрипторов), который будет определять параметры выполняемых модулем действий. Начало цепочки задается значением регистра BASE_DESCRIPTOR. Ниже в виде таблиц приведена структура дескриптора данных DMA.

Таблица 26.2 – Структура дескриптора данных DMA

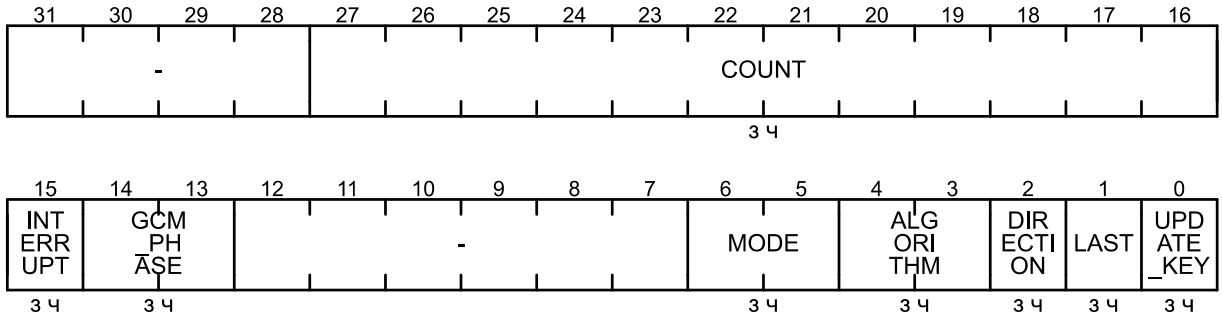
Индекс	Смещение адреса	Слова дескриптора
0	0x0	Слово управления DMA.
1	0x4	32-битовый стартовый адрес источника.
2	0x8	32-битовый стартовый адрес назначения.
3	0xC	Следующий дескриптор данных DMA (32-битовый адрес, указывающий на следующий дескриптор данных).

Адреса источника данных должны быть выровнены по границе 32-разрядного слова. Бит LAST в управляющем слове дескриптора маркирует последний дескриптор цепочки. Если он сброшен, то адрес следующего дескриптора данных должен быть выровнен по границе в 4 слова. Если бит LAST установлен, последнее слово дескриптора не будет считано, а значение регистра NEXT_DESCRIPTOR будет сброшено.

В фазах GCM_INIT и GCM_HEADER режима GALOIS_COUNTER_MODE значение слова стартового адреса назначения игнорируется, поскольку результат их выполнения не подразумевает записи какого-либо результата в память. В фазе GCM_INIT режима GALOIS_COUNTER_MODE также будет проигнорировано значение слова стартового адреса источника, поскольку единственными изменяемыми входными данными является секретный ключ, хранимый в регистрах KEY_*.

Стоит отметить, что при считывании валидного значения слова управления DMA будут перезаписаны поля DIRECTION, ALGORITHM, MODE, GCM_PHASE регистра CONTROL.

Таблица 26.3 –Формат слова управления DMA



Поле	Биты	Описание
COUNT	27-16	Количество обрабатываемых блоков входного текста COUNT + 1 (размер блока зависит от значения поля TYPE: 64 бит для шифра Магма, 128 бит для остальных). В режиме GALOIS_COUNTER_MODE в фазах GCM_INIT и GCM_LAST_BLOCK данное поле должно иметь нулевое значение, в противном случае дескриптор будет считаться некорректным.
INTERRUPT	15	Разрешение генерации прерывания при завершении выполнения дескриптора.

Поле	Биты	Описание	
GCM_PHASE	14-13	Фаза обработки пакета в режиме GALOIS_COUNTER_MODE. Если режим выполнения криптографической операции не GALOIS_COUNTER_MODE, то поле должно иметь нулевое значение (GCM_INIT), в противном случае дескриптор будет считаться некорректным.	
		0	GCM_INIT
		1	GCM_HEADER
		2	GCM_PAYLOAD
		3	GCM_LAST_BLOCK
MODE	6-5	Выбор режима выполнения криптографической операции. Использование GALOIS_COUNTER_MODE в сочетании с криптографическим алгоритмом MAGMA не поддерживается, дескриптор будет считаться некорректным.	
		0	ELECTRONIC_CODEBOOK (ECB)
		1	CIPHER_BLOCK_CHAINING (CBC)
		2	COUNTER_MODE (CTR)
		3	GALOIS_COUNTER_MODE (GCM)
ALGORITHM	4-3	Тип криптографического алгоритма	
		00b	AES-128
		01b	AES-256
		10b	Магма
		11b	Кузнечик
DIRECTION	2	Режим работы	
		0	Шифрование (encryption)
		1	Дешифрование (decryption)
LAST	1	Установленный бит указывает на то, что данный дескриптор является последним в цепочке дескрипторов	
UPDATE_KEY	0	После считывания корректного дескриптора данных DMA принудительно выполнить обновление раундовых ключей шифрования на основе текущего значения регистров {KEY_0, ..., KEY_7} или {KEY_0, ..., KEY_3} для алгоритма AES-128.	
-	31-28, 12-7	Зарезервированные биты. Должны быть равны 0, в противном случае дескриптор будет считаться некорректным	

Таблица 26.4 – Формат 32-битового стартового адреса источника (Source_Address)

Поле	Биты	Описание
ADDRESS	31-2	Старшие биты адреса, который указывает на начало массива слов входных данных
-	1-0	Младшие биты адреса. Должны быть равны 0, в противном случае дескриптор будет считаться некорректным

Таблица 26.5 – Формат 32-битового стартового адреса назначения (Destination_Address)

Поле	Биты	Описание
ADDRESS	31-2	Старшие биты адреса, который указывает на начало участка памяти, в который будет сохраняться результат выполнения криптографической операции
-	1-0	Младшие биты адреса. Должны быть равны 0, в противном случае дескриптор будет считаться некорректным

Таблица 26.6 – Формат адреса следующего дескриптора данных DMA (NEXT_DESCRIPTOR)

Поле	Биты	Описание
ADDRESS	31-4	Старшие биты адреса, который указывает на следующий дескриптор в цепочке. Если в слове управления DMA установлен бит LAST, то слово NEXT_DESCRIPTOR не будет прочитано
-	3-0	Младшие биты адреса. Если в слове управления DMA не установлен бит LAST, данные биты должны быть равны 0, в противном случае дескриптор будет считаться некорректным

В процессе чтения дескриптора производится анализ значений полей очередного слова дескриптора. Если оно признано некорректным, то работа поточного режима прекращается. Устанавливается бит DONE регистра IRQ, если был установлен соответствующий бит в регистре IRQ_ENABLE. В поле BAD_DESCR регистра STATUS устанавливается значение, которое указывает на слово с недопустимым значением. Устанавливается бит DMA_FAIL регистра IRQ, если это разрешено через регистр IRQ_ENABLE.

Причины получения ненулевого значения STATUS.BAD_DESCR сведены в таблицу, приведенную ниже.

Таблица 26.7 – Причины получения ненулевого значения STATUS.BAD_DESCR

Индекс слова	Причины некорректности слова дескриптора
0	Обнаружен один или несколько ненулевых битов на позициях с 7 по 12 или с 28 по 31.
	Обнаружено неподдерживаемое сочетание значений полей MODE и ALGORITHM (GALOIS_COUNTER_MODE в паре с криптографическим алгоритмом MAGMA).
	Обнаружено ненулевое значение в поле GCM_PHASE, когда режим криптографической операции отличается от GALOIS_COUNTER_MODE.
	Обнаружено ненулевое значение в поле COUNT, когда MODE соответствует GALOIS_COUNTER_MODE, а значение GCM_PHASE – GCM_INIT или GCM_LAST_BLOCK.
1	Обнаружен один или несколько ненулевых битов на позициях с 0 по 1.
2	Обнаружен один или несколько ненулевых битов на позициях с 0 по 1.
3	Бит LAST слова управления DMA сброшен и обнаружен один или несколько ненулевых битов на позициях с 0 по 3.

К нештатному завершению выполнения цепочки дескрипторов может привести получение ответа ERROR на доступ чтения по шине AHB слова дескриптора, чтения

входных блоков текста или записи выходных блоков текста. О данном факте будет свидетельствовать соответствующее значение поля AHB_ERROR регистра STATUS. Генерация прерываний аналогична таковой для ненулевого значения BAD_DESCR.

Алгоритм функционирования поточного режима работы

Ниже приведен алгоритм работы машины конечных состояний, которая отвечает за реализацию поточного режима работы:

1) Ожидание записи DMA_CONTROL.START = 1;

2) В качестве текущего адреса дескриптора (отображается в CURRENT_DESCRIPTOR.ADDRESS) устанавливается значение из регистра BASE_DESCRIPTOR.ADDRESS;

3) Чтение слова управления DMA. Если по шине AHB получен ответ ERROR или слово признано некорректным, то происходит установка соответствующего статуса, если соответствующие прерывания разрешены в регистре IRQ_ENABLE, генерируются прерывания DMA_FAIL и DONE, происходит переход к пункту 1;

4) Значения полей DIRECTION, ALGORITHM, MODE, GCM_PHASE заносятся в регистр CONTROL. В STATUS.DMA_LAST отображается состояние бита LAST. В IRQ_ENABLE.DMA_DONE – значение бита INTERRUPT;

5) Чтение слова стартового адреса источника. Получение ответа ERROR или обнаружение ошибки в слове обрабатывается аналогично пункту 3;

6) Чтение слова стартового адреса назначения. Получение ответа ERROR или обнаружение ошибки в слове обрабатывается аналогично пункту 3;

7) Данный пункт выполняется, только если был сброшен бит LAST слова управления DMA. Чтение слова адреса следующего дескриптора (отображается в NEXT_DESCRIPTOR.ADDRESS). Получение ответа ERROR или обнаружение ошибки в слове обрабатывается аналогично пункту 3;

8) Если для текущего криптографического алгоритма нет рассчитанных раундовых ключей, активируется их расчет на основе текущего значения из регистров KEY_*;

9) Данный пункт не выполняется, если CONTROL.MODE равен GALOIS_COUNTER_MODE, а CONTROL.GCM_PHASE соответствует GCM_INIT. Чтение блока входного текста. Четыре или два слова, если выбран алгоритм «Магма», заносятся в регистры TEXT_IN_* (с учетом WORDS_SWAP и BYTES_SWAP регистра DMA_CONTROL). Получение ответа ERROR обрабатывается аналогично пункту 3;

10) Ожидание готовности раундовых ключей;

11) Запуск выполнения шифрования/дешифрования;

12) Ожидание готовности выходных данных;

13) Данный пункт не выполняется, если CONTROL.MODE равен GALOIS_COUNTER_MODE, а CONTROL.GCM_PHASE соответствует GCM_INIT или GCM_HEADER. В остальных случаях выполняется запись в память блока выходного текста (содержимое TEXT_OUT_*) или аутентификационного тега (содержимое GCM_Tag_*). Тег является результирующими данными, если CONTROL.MODE равен GALOIS_COUNTER_MODE, а CONTROL.GCM_PHASE соответствует GCM_LAST_BLOCK. В обоих случаях данные сохраняются с учетом WORDS_SWAP и BYTES_SWAP регистра DMA_CONTROL. Получение ответа ERROR обрабатывается аналогично пункту 3;

14) Если CONTROL.MODE не равен ELECTRONIC_CODEBOOK, выполняем обновление IV_*;

15) Если значение счетчика обработанных блоков STATUS.DMA_COUNTER меньше значения поля COUNT слова управления DMA, то переходим к пункту 9;

16) Если в слове управления DMA бит LAST был сброшен, генерируется прерывание DMA_DONE, если был установлен бит INTERRUPT в слове управления DMA, заносим в CURRENT_DESCRIPTOR значение из NEXT_DESCRIPTOR, возвращаемся к пункту 3;

17) Если в слове управления DMA бит LAST был установлен, генерируется прерывание DMA_DONE, если был установлен бит INTERRUPT в слове управления DMA, генерируется прерывание DONE, если оно разрешено в регистре IRQ_ENABLE, происходит переход к пункту 1.

26.5 Экстренное прекращение выполнения операции

Для организации экстренного прекращения нужно записать в поле CODE регистра TERMINATE значение 0xD0. На следующий такт после такой записи будет сформирован сигнал сброса состояния модулей криптографических операций и хранилища раундовых ключей. Данное событие вернет биты READY, KEYS_READY, TEXT_OUT_VALID, KEYS_STORED, DMA_ACTIVE, DMA_COUNTER и DMA_LAST регистра STATUS к значениям по умолчанию. Стоит отметить, что бит IRQ_PENDING, поля BAD_DESCR и AHB_ERROR не изменят своего значения.

Регистр TERMINATE

CODE – Запись в данное поле значения 0xD0 приведет к сбросу внутреннего состояния модулей криптографических операций и хранилища раундовых ключей. Запись других значений будет проигнорирована. При чтении всегда возвращает 0.

26.6 Производительность

Таблица 26.8 – Количество тактов на операцию в режиме работы с одним блоком

Операция	Длительность в тактах			
	AES-128	AES-256	Magma	Kuznechik
Расчет раундовых ключей	10 + 1	14 + 1	1 + 1	32 + 1
Шифрование/дешифрование	10 + 1	14 + 1	8 + 1	10 + 1
Шифрование/дешифрование с новым ключом	20 + 1	28 + 1	9 + 1	42 + 1

В поточном режиме работы к длительности операции прибавляется задержка на считывание дескриптора, считывание входных блоков текста и запись выходных блоков текста.

В данной реализации на выполнение умножение в поле Галуа (GF(2^{128})mult), используемого в криптографическом режиме GALOIS_COUNTER_MODE, тратится 8 тактов. Длительность операций отличается для разных фаз:

1 В фазе GCM_INIT определяется длительностью шифрования выбранного алгоритма;

2 В фазе GCM_HEADER равняется длительности GF(2^{128})mult плюс 1 такт;

3 В фазе GCM_PAYLOAD:

3.1. при выполнении шифрования определяется суммой длительности шифрования выбранного алгоритма, длительности GF(2^{128})mult плюс 1 такт;

3.2. при выполнении дешифрования определяется длительностью шифрования выбранного алгоритма. Это обусловлено тем, что умножение на блок шифрованного текста, находящегося в TEXT_IN_*, на хэш-ключ и шифрование IV_* выполняются параллельно;

4 В фазе GCM_LAST_BLOCK определяется длительностью шифрования выбранного алгоритма. Это происходит, поскольку умножение блока длины на хэш-ключ и шифрование значения на основе IV_* (вместо IV_3 используется 0x00000001) выполняются параллельно.

27 Сторожевой таймер WDT

Сторожевой таймер позволяет сбросить систему в случае отказа программного обеспечения. Пользователь может включать или выключать сторожевой таймер по собственному усмотрению.

Управляющие регистры сторожевого таймера приведены в приложении А.20.

Сторожевой таймер представляет собой 32-битный обратный счетчик, который загружается значением из регистра LOAD. Счетчик уменьшается на единицу по каждому нарастающему фронту тактового сигнала WDTCLK.

По умолчанию сторожевой таймер не тактируется и находится в сбросе. Активировать таймер можно с помощью регистра WDTCFG блока RCU.

Сброс сторожевого таймера WDT осуществляется записью любого значения в регистр INTCLR.

Бит запрета записи (REGWRDIS) во все регистры сторожевого таймера, кроме регистра LOCK, необходим для предотвращения отключения сторожевого таймера сбойными программами. Чтобы узнать состояние бита блокировки необходимо прочитать бит REGWRDIS регистра LOCK. Для сброса бита следует записать в регистр LOCK значение 1ACCE551h. Для установки бита следует записать в регистр LOCK любое значение, кроме 1ACCE551h.

Включение счета сторожевого таймера и его прерывания осуществляются установкой бита INTEN в регистре CTRL. Когда счетчик таймера достигает нуля, устанавливается флаг WDTINT в регистре MIS, а в счетчик загружается значение из регистра LOAD.

Далее, если установлен бит RESEN, счетчик продолжает декрементироваться. Если на момент повторного достижения нуля флаг WDTINT установлен, производится сброс микроконтроллера.

Для предотвращения сброса микроконтроллера необходимо сбросить флаг WDTINT путем записи любого значения в регистр сброса INTCLR, что приводит к сбросу прерывания сторожевого таймера и загрузке счетчика значением из регистра LOAD.

Алгоритм работы сторожевого таймера показан на рисунке 27.1.

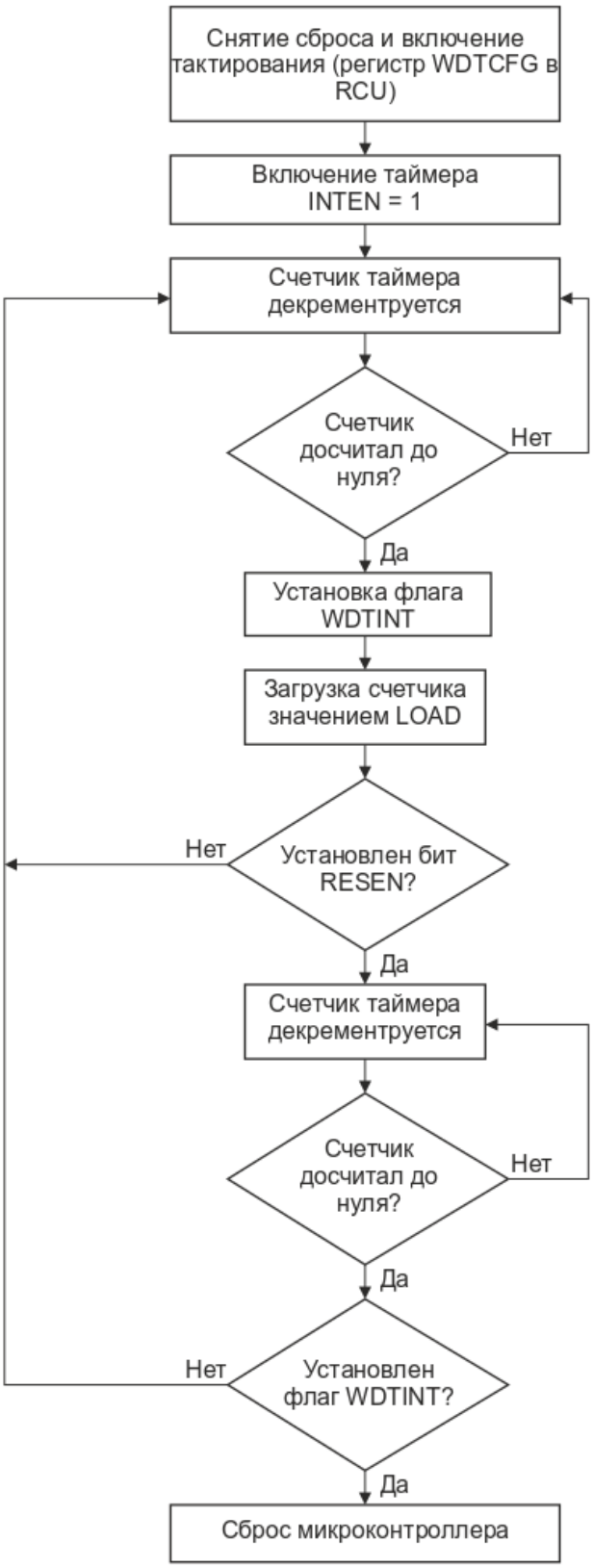


Рисунок 27.1 – Алгоритм работы сторожевого таймера

28 Блок часов реального времени RTC

Блок RTC с батарейным питанием представляет собой модуль, реализующий функцию часов реального времени.

Блок RTC с батарейным питанием обеспечивает:

– возможность установки начального времени;

– контроль напряжения питания основного домена;

– переключение на батарейный домен при опускании напряжения питания основного домена ниже порога с формированием сигнала прерывания;

– тактирование от внешнего кварцевого генератора, или от внутреннего RC осциллятора;

– два программируемых будильника с возможностью генерации прерывания.

Следует заметить, что сброс, формируемый схемами POR/BOR, не приводит к автоматической подаче сигнала сброса на модуль RTC. Прикладному ПО необходимо самостоятельно сформировать сброс данного блока с помощью установки и последующего сброса бита RCU.RTCCFG.RST. Например, в ходе инициализации контроллера можно проверять наличие некоторого кодового значения в одном из регистров GPR модуля RTC. Если значение не совпало с ожидаемым, то производить сброс модуля RTC и записывать кодовое значение в выбранный регистр GPR. Схема сброса организована подобным образом, чтобы модуль RTC мог продолжать работу от батарейного питания при временном исчезновении основного питания.

Управляющие регистры блока RTC приведены в приложении А.21.

Структурная схема блока часов реального времени представлена на рисунке 28.1.

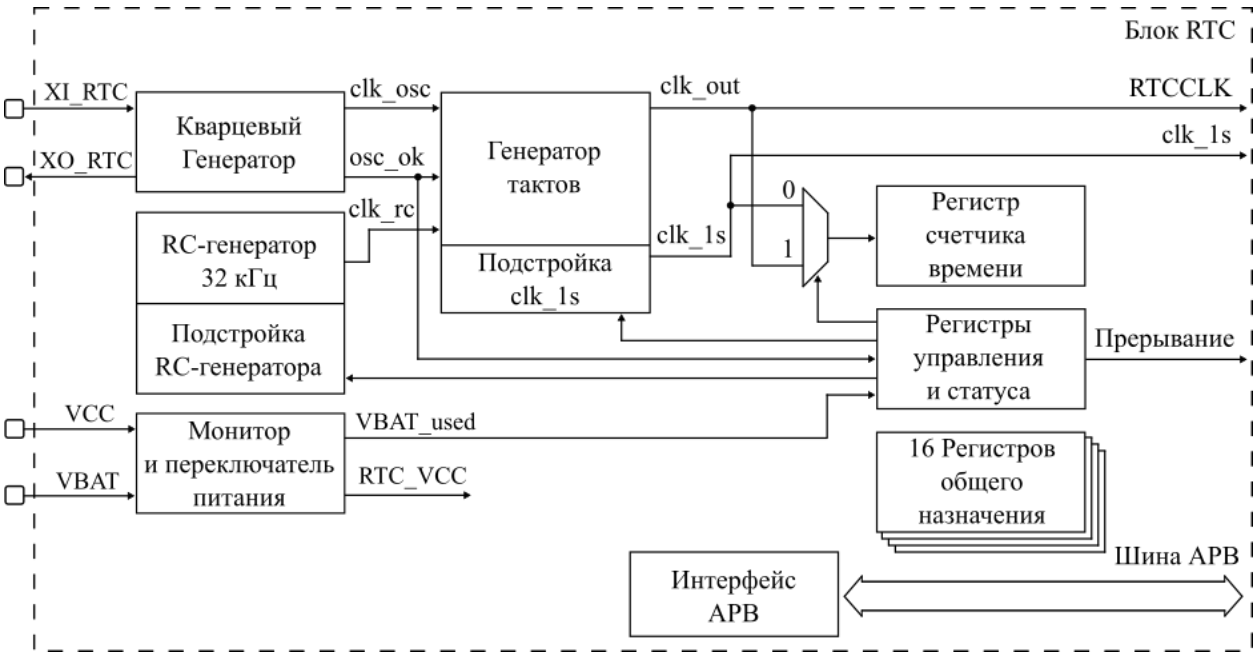


Рисунок 28.1 – Структурная схема блока RTC

В основе модуля RTC лежат два счетчика: счетчик тактов частоты 32кГц (Tick generator) и счетчик секунд (регистр TIME).

Счетчик тактов может тактироваться частотой 32кГц, получаемой из двух источников: внешней осцилляторной площадки (OSC) или внутреннего RC-осциллятора. Переключение между источниками тактового сигнала происходит записью бита DRC регистра CONTROL.

С помощью регистра TRIM_CLK1S возможна подстройка счетчика, формирующего 1с метки. Для подстройки в регистр вписывается количество тактов синхросигнала в дополнительном коде. Причем положительное число приводит каждые 64 секунды к увеличению на заданное число тактов одного интервала 1с, а отрицательное число, аналогично каждые 64 секунды, приводит к уменьшению одного интервала 1с.

На выходе счетчика тактов 32кГц формируется сигнал 1с метки, поступающий на выход блока. Данный сигнал имеет скважность 50%.

Дополнительно 1с метка поступает в счетчик секунд. Значение счетчика секунд может быть вычитано в 32 битном формате (регистр TIME). Имеется возможность предустановки начального значения времени в 32 битном формате через программирование регистра TIME. Важно отметить, что фактическое изменение значения регистра TIME произойдет только в момент прихода очередной 1с метки.

В блоке RTC реализованы 2 будильника ALARM. Для установки будильника нужно задать значение времени срабатывания будильника с помощью регистров ALARM1 и ALARM2 для первого и второго будильника соответственно.

Примечание – Для срабатывания будильника необходимо, чтобы заданное значение было больше значения текущего времени.

При достижении регистра времени значения, записанного в регистры будильника формируются флаги A1F и A2F.

A2F (флаг срабатывания будильника 2) – значение флага доступно на чтение через соответствующий бит в регистре STATUS. Бит устанавливается, когда текущее значение времени становится равным значению регистра будильника 2 (ALARM2). Бит можно сбросить записью логической 1, запись 0 игнорируется. При значении бита A2IE регистра управления равном логической 1, совместно с флагом формируется прерывание.

A1F (флаг срабатывания будильника 1) – значение флага доступно на чтение через соответствующий бит в регистре STATUS. Бит устанавливается, когда текущее значение времени становится равным значению регистра будильника 1 (ALARM1). Бит можно сбросить записью логической 1, запись 0 игнорируется. При значении бита A1IE регистра управления равном логической 1, совместно с флагом формируется прерывание.

С помощью регистра TRIM_RC происходит корректировка выходной частоты RC генератора. Корректировка происходит через вписывание в регистр числа корректировки в диапазоне от -128 до +127, соответствующего требуемой частоте. Значение корректировки вписывается в дополнительном коде. Шаг подстройки частоты меняется не линейно.

В ходе работы блока RTC дополнительно вырабатываются флаги:

– OSC_OK – логическая единица в этом бите указывает на то, что осциллятор XTAL перешел в штатный режим работы со стабильной генерацией тактового сигнала.

– BF (флаг переключения на батарейное питание) – логическая 1 в бите BF указывает на то, что произошло переключение питания от VCC1 на VBAT. Бит можно сбросить записью логической 1, запись 0 игнорируется.

– NPF (флаг переключения на штатное питание) – логическая 1 в бите NPF указывает на то, что произошло переключение с питания от VBAT на VCC1. Бит можно сбросить записью логической 1, запись 0 игнорируется.

– OF (флаг переполнения регистра времени) – логическая 1 в бите OF указывает на то, что произошло переполнение регистра времени. Бит можно сбросить записью логической 1, запись 0 игнорируется.

– BU (используется батарейное питание) – логическая 1 в бите BF указывает на то, что в качестве источника питания используется VBAT.

Значения всех флагов доступны при чтении регистра STATUS.

В блоке RTC реализован режим быстрого тактового сигнала. В данном режиме регистр времени вместо сигнала 1с тактируется сигналом 32кГц. Режим быстрого тактового сигнала задается записью 1 в бит FCM регистра управления CONTROL.

Блок RTC имеет шестнадцать 32 битных регистров общего назначения GRPn (где n – номер регистра). Регистры доступны на чтение и запись по соответствующим адресам.

На рисунке 28.2 показана схема подключения кварцевого резонатора в колебательный контур кварцевого генератора блока RTC.

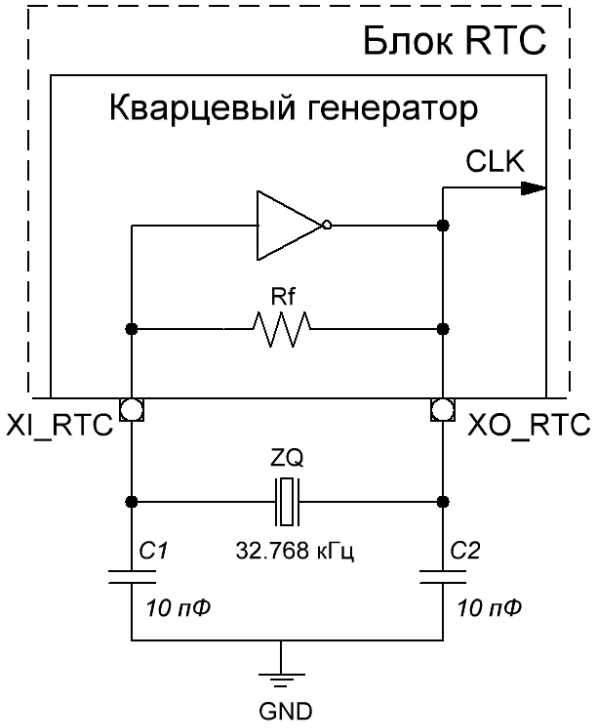


Рисунок 28.2 – Подключение кварцевого резонатора в колебательный контур RTC

Внутренний резистор обратной связи Rf, а также конденсаторы C1, C2 используются для регулировки времени включения, поддержания стабильности и точности работы генератора.

Рекомендуемая ёмкость конденсаторов C1 и C2 составляет 10 пФ, рекомендуемое эквивалентное сопротивление (ESR) – не более 100 кОм. Время запуска кварцевого генератора – не более 500 мс.

При трассировке печатной платы необходимо устанавливать конденсаторы и резонатор как можно ближе к соответствующим выводам микроконтроллера. Рекомендуется использовать многослойные керамические конденсаторы с диэлектриком типа C0G (NP0), но также допускается использовать X7R с рабочим напряжением не ниже 6.3 В для типоразмеров 0402 или 0603.

При тактировании блока RTC от внешнего тактового генератора необходимо учитывать следующие условия:

– Верхняя граница напряжения для тактового сигнала должна превышать значение 600 мВ;

– Нижняя граница напряжения для тактового сигнала не должна превышать значение 400 мВ.

29 Приемопередатчики UART

В состав микроконтроллера входят восемь идентичных универсальных асинхронных приемопередатчиков UART0 – UART7.

В состав приемопередатчика входят два буфера типа FIFO. Буфер приемника имеет разрядность 12, буфер передатчика – разрядность восемь. Каждый буфер может хранить до 32 байт данных, и каждый буфер может быть сконфигурирован (программно) как 32-байтный или как однобайтный.

Приемопередатчик обеспечивает:

- независимое маскирование прерываний от буфера передатчика, буфера приемника, по таймауту приемника, а также в случае обнаружения ошибки;
- возможность деления тактовой частоты в диапазоне от 1 до 65535 (допускается использование нецелых коэффициентов деления, что позволяет использовать любой опорный генератор с частотой более 3,6864 МГц);
- поддержку прямого доступа к памяти;

Приемопередатчик реализует:

- передачу данных длиной от 5 до 8 бит со скоростью до 3000 Кбит/с;
- контроль четности (проверочный бит выставляется по четности, нечетности, имеет фиксированное значение либо не передается);
- формирование одного или двух стоповых бит;
- обнаружение ложных стартовых битов;
- формирование и обнаружение сигнала разрыва линии.

Управляющие регистры приемопередатчиков UART приведены в приложении А.22.

Функциональные возможности

Режим работы приемопередатчика и скорость обмена данными контролируются регистром LCRH и регистрами делителя IBRD и FBRD.

Устройство может формировать следующие сигналы:

- независимые маскируемые прерывания от приемника (в том числе по таймауту), передатчика и в случае обнаружения ошибки;
- общее прерывание, возникающее в случае, если возникло одно из независимых немаскированных прерываний;
- сигналы запроса на прямой доступ к памяти для совместной работы с контроллером DMA.

В случае возникновения ошибки в структуре сигнала, четности данных, а также разрыва линии, соответствующий бит ошибки устанавливается и сохраняется в буфере приемника. В случае переполнения буфера приемника также устанавливается соответствующий бит, а буфер становится недоступным для записи.

29.1 Функционирование блока UART

На рисунке 29.1 показана упрощенная функциональная схема приемопередатчика.

Генератор тактового сигнала приемопередатчика формирует синхросигнал последовательного обмена данными, который представляет собой последовательность импульсов с шириной, равной одному периоду сигнала UARTCLK, и частотой в 16 раз превышающей частоту передачи данных.

Буфер передатчика предназначен для хранения данных, полученных от ЦП, до тех пор, пока они не будут переданы внешнему устройству.

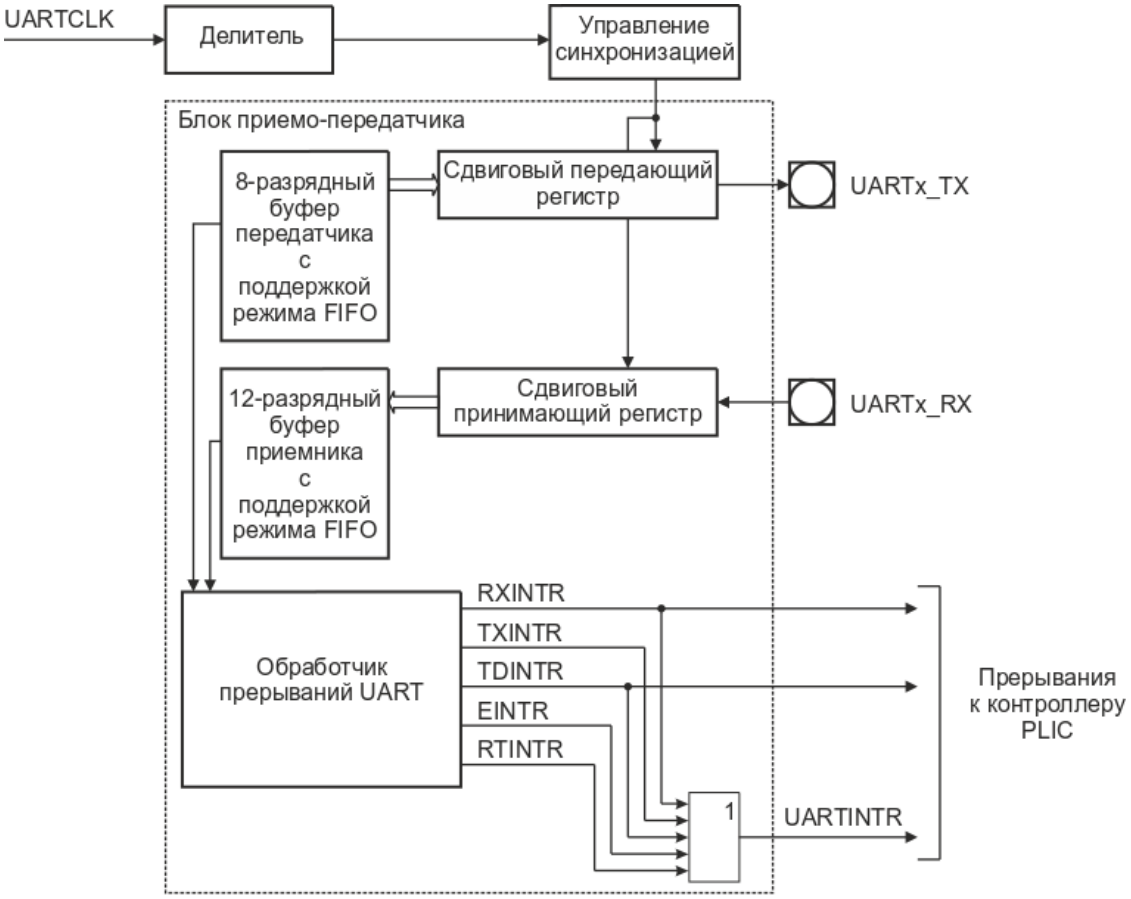


Рисунок 29.1 – Функциональная схема приемопередатчика

Буфер приемника предназначен для хранения данных и кодов ошибки (принятых от внешнего устройства) до тех пор, пока они не будут прочитаны ЦП.

Обработчик прерываний генерирует независимые маскируемые прерывания с активным высоким уровнем. Кроме того, формируется комбинированное прерывание путем объединения независимых прерываний по схеме ИЛИ. Сигнал прерывания передается на контроллер PLIC.

Сброс модуля

Приемопередающая логика модуля, управляющие регистры и FIFO по умолчанию находятся в сбросе. Снять сброс можно путём установки бита RSTDIS в соответствующем регистре UARTCFGi (i – номер модуля от 0 до 7) блока RCU.

Синхронизация

Существует ограничение на соотношение между частотами тактовых сигналов PCLK и UARTCLK, представленное формулой 29.1.

$$f_{UARTCLK} \leq f_{PCLK} \times 5/3 \quad (29.1)$$

Например, для достижения максимальной скорости передачи данных 921600 бод (при $f_{UARTCLK}$ = 921600 × 16 = 14,7456 МГц) частота f_{PCLK} должна быть не менее 8,84736 МГц.

Для точной настройки частоты передачи данных используются два делителя – один управляется регистром UARTCFGx блока RCU, второй находится внутри модуля UART. Коэффициент деления второго делителя имеет целую и дробную части, которые задаются регистрами IBRD и FBRD. Возможность задания нецелых коэффициентов деления позволяет осуществлять обмен данными на стандартных скоростях, используя в качестве источника тактовый сигнал с произвольной частотой более 3,6864 МГц.

Коэффициент деления K частоты сигнала UARTCLK рассчитывается по формуле 29.2.

$$K = f_{UARTCLK} / (16 \times baudrate), \quad (29.2)$$

где $f_{UARTCLK}$ – частота сигнала синхронизации блока UART – UARTCLK, Гц;
baudrate – скорость передачи, бод.

Получившееся дробное десятичное число следует разделить на две части – целую и дробную.

Целая часть после преобразования в двоичный формат записывается в регистр IBRD.

Дробная часть умножается на 64 и округляется до ближайшего целого числа. Полученное число преобразовывается в двоичный формат и записывается в регистр FBRD.

Для примера, пусть требуемая скорость передачи данных 230 400 бит/с и частота тактового сигнала UARTCLK равна 4 МГц. Тогда:

$$K = (4 \times 10^6) / (16 \times 230400) = 1{,}085.$$

Получившееся число разбивается на две части – 1 и 0,085.

В регистр IBRD записывается значение 0001h.

Значение (0,085 × 64) округляется и преобразовывается в 05h для записи в регистр FBRD.

Таким образом, реальные значения коэффициента деления частоты и скорости передачи будут следующими:

$$K = 1 + 5/64 = 1{,}078,$$

$$baudrate = (4 \times 10^6)/(16 \times 1{,}078) = 231911 \text{ бит/с}.$$

Ошибка установки скорости:

$$\Delta = ((231911 - 230400)/230400) \times 100\ \% = 0{,}656\ \%.$$

Максимальная ошибка установки скорости передачи данных:

$$\Delta = (1/64) \times 100\ \% = 1{,}5625\ \%.$$

Такая ошибка возникает в случае K = 1, при этом разница накапливается в течение 64 тактовых интервалов.

Содержимое регистров LCRH, IBRD и FBRD обновляется при записи в регистр LCRH. Таким образом, для того, чтобы новые параметры коэффициента деления вступили в силу, после их записи в регистры IBRD и FBRD, необходимо осуществить запись в регистр LCRH и только в такой последовательности.

Примечание – Изменение содержимого регистров IBRD, FBRD и LCRH допускается только в то время, когда приемопередатчик запрещен и не осуществляется передача/прием байта.

Передача и прием данных

Данные для передачи заносятся в буфер передатчика посредством записи в регистр DR. После записи хотя бы одного байта в буфер передатчика устанавливается флаг BUSY в регистре FR. Это состояние флага сохраняется, пока буфер передатчика не пуст (даже если работа приемопередатчика запрещена). Далее, если работа приемопередатчика разрешена (установлены биты UARTEN и TXE регистра CR), начинается передача информационного кадра с параметрами, указанными в регистре управления линией LCRH. Передача данных продолжается до опустошения буфера передатчика (до окончания передачи всех байт). По окончании передачи сбрасывается флаг BUSY.

При приеме байта данных (установлены биты UARTEN и RXE регистра CR) для каждого бита производятся три выборки уровня, и решение о значении бита принимается по мажоритарному принципу.

В случае если приемник находился в неактивном состоянии (постоянный высокий уровень сигнала на линии UARTx_RX), и произошла смена уровня входного сигнала с высокого на низкий (стартовый бит), включается счетчик, тактируемый внутренним сигналом, после чего отсчеты сигнала на входе приемника регистрируются каждые восемь тактов.

Стартовый бит считается достоверным в случае, если сигнал на линии UARTx_RX сохраняет низкий логический уровень в течение восьми периодов внутреннего синхросигнала с момента включения счетчика. В противном случае переход в ноль рассматривается как ложный старт и игнорируется.

После обнаружения достоверного стартового бита очередной бит данных фиксируется каждые 16 отсчетов тактового сигнала. Производится регистрация всех бит данных (согласно запрограммированным параметрам) и бита четности (если включен режим контроля четности).

По окончании приема байта производится проверка присутствия корректного стопового бита (высокий логический уровень сигнала UARTx_RX). После этого байт заносится в буфер приемника вместе с тремя битами признаков ошибки, см. рисунок 29.2, и битом переполнения буфера.

В 12-разрядной ячейке буфера байт данных располагается в области младших восьми бит, три бита признаков ошибки – в битах с 8 по 10.

Флаг переполнения буфера приемника выставляется в том случае, если к моменту, когда очередной кадр данных полностью принят, буфер уже заполнен. В этом случае принятый кадр остается в сдвиговом принимающем регистре и, в случае приема следующего кадра данных, будет потерян.

Как только в буфере приемника освобождается место для записи, кадр данных, находящийся в сдвиговом регистре, переписывается в буфер, а флаг переполнения сбрасывается.

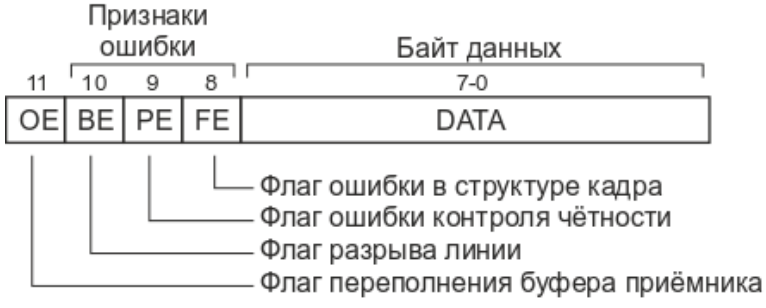


Рисунок 29.2 – 12-разрядная ячейка буфера приема

Данные из буфера приемника можно прочитать посредством регистра DR. Состояние признаков ошибки и флага переполнения определяется чтением регистра RSR и относится к последнему байту, считанному из регистра DR, в связи с этим регистр DR всегда должен считываться первым.

Все флаги сбрасываются одновременно записью любого значения в регистр RSR или после сброса устройства.

Примечания:

1 Необходимо запрещать работу приемопередатчика перед перепрограммированием его регистров управления. Если приемопередатчик переводится в отключенное состояние во время передачи или приема, то перед остановкой он завершает выполняемую операцию.

2 Целостность данных в буферах передатчика и приемника не гарантируется, если установился флаг BRK (разрыв линии), или если программное обеспечение произвело остановку приемопередатчика после его повторного перевода в разрешенное состояние.

29.2 Интерфейс прямого доступа к памяти

Приемопередатчик оснащен интерфейсом подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром управления DMACR. Интерфейс DMA включает в себя шесть сигналов.

Сигнал RXDMAREQ/TXDMAREQ (для приема/передачи) – запрос, инициируемый приемопередатчиком. Размер символа в режиме приема данных – до 12 бит. Сигнал переходит в активное состояние в случае, если заполнение буфера приемника превысило заданный порог. Порог программируется индивидуально для каждого буфера посредством полей регистра IFLS.

Примечание – Сигналы запросов RXDMAREQ/TXDMAREQ формируются только в случае использования буферов FIFO (установлен бит FEN регистра LCRH).

Сигнал RXDMACLR/TXDMACLR (для приема/передачи) – сигнал обновления запроса, инициируемый DMA контроллером, служит флагом для внутреннего генератора запросов DMA в приёмопередатчике что DMA обслужил запрос и готов получить новый.

Сигнал RXDMADONE/TXDMADONE (для приема/передачи) – сигнал окончания работы канала DMA.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом DMACLR.

После снятия сигнала сброса модуль приемопередатчика вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае установки в ноль бит управления TXDMAE или RXDMAE в регистре управления DMACR.

Также, есть возможность автоматического окончания генерации запросов к DMA при установке управляющих битов DMATXSTOP или DMARXSTOP. Если данные биты установлены, пришедший сигнал окончания работы канала TXDMADONE/RXDMADONE автоматически очистит бит TXDMAE или RXDMAE соответственно.

Срабатывание запросов обмена RXDMAREQ и TXDMAREQ зависит от указанных порогов заполнения буферов FIFO, выбираемых с помощью полей RXIFLSEL и TXIFLSEL регистра IFLS (диапазон значений: 1 – 32).

В регистре управления DMACR предусмотрен бит DMAONERR, который позволяет запретить DMA от приемника в случае активного состояния линии прерывания по обнаружению ошибки EINTR. При этом соответствующие линии запроса DMA – RXDMAREQ и RXDMAREQ – переводятся в неактивное состояние (маскируются) до сброса EINTR. На линии запроса DMA, обслуживающие передатчик, состояние EINTR не влияет.

29.3 Прерывания

В модуле предусмотрено семь маскируемых источников прерываний.
Сигналы запросов на прерывания:
- RXINTR – от приемного FIFO;
- TXINTR – от передающего FIFO;
- RTINTR – по таймауту приемника;
- TDINTR – по окончанию передачи в линии;
- EINTR – по ошибке;
- UARTINTR – логическое ИЛИ сигналов запросов на прерывания.

Каждый из сигналов может быть маскирован путем установки соответствующего бита в регистре маски IMSC. Для разрешения соответствующего прерывания необходимо установить бит, для запрещения прерывания – сбросить.

Источник прерывания также можно определить, считав состояние регистра RIS или регистра MIS (маскированные прерывания). Сброс прерывания осуществляется программно путём установки соответствующего бита в регистре ICR.

Пороги прерываний программируются индивидуально для буферов приема и передачи посредством полей RXIFLSEL и TXIFLSEL регистра IFLS соответственно.

Сигнал RXINTR

Запрос возникает в случае обнаружения одного из событий:

- буфер приемника в режиме FIFO и его заполнение достигло заданного порогового значения (поле RXIFLSEL регистра IFLS);

- буфер приемника имеет одну ячейку (режим FIFO запрещен) и принят один кадр данных.

Линия прерывания переходит в высокое состояние и удерживается в нем до тех пор, пока из буфера не будет прочитан, как минимум, один байт или выполнен программный сброс прерывания (регистр ICR).

Сигнал TXINTR

Запрос возникает в случае обнаружения одного из событий:

- буфер передатчика в режиме FIFO и его опустошение достигло заданного порогового значения (поле TXIFLSEL регистра IFLS);

- буфер передатчика имеет одну ячейку (режим FIFO запрещен) и пуст.

Линия прерывания переходит в высокое состояние и удерживается в нем до тех пор, пока в буфер не будет записан, как минимум, один байт или выполнен программный сброс прерывания.

Запись данных в буфер передатчика допускается как перед разрешением работы приемопередатчика и прерываний, так и после разрешения.

Примечание – Прерывание передатчика работает по фронту, а не по уровню сигнала. В случае если работа приемопередатчика и прерывания от него разрешена до осуществления записи данных в буфер передатчика, прерывание не формируется. Прерывание возникает только при опустошении буфера.

Сигнал TDINTR

Запрос возникает в случае окончания передачи в линии.

Сигнал RTINTR

Запрос возникает в случае, если буфер приемника не пуст и на вход приемника не поступало новых данных в течение периода времени, необходимого для передачи 32 бит. Прерывание сбрасывается после считывания данных из буфера приемника до его опустошения или программно.

Сигнал EINTR

Запрос возникает в случае ошибки при приеме данных. Оно может быть вызвано:
- ошибкой в структуре кадра;
- ошибкой контроля четности;
- разрывом линии;
- переполнением буфера приемника.

Сигнал UARTINTR

Логическое ИЛИ сигналов RXINTR, TXINTR, TDINTR, MSINTR, RTINTR, EINTR.

В контроллер PLIC поступает 3 линии прерываний (RXINTR, TDINTR, UARTINTR) от каждого блока UART.

29.4 Программирование

Для программирования рекомендуется следующая последовательность действий:
- запретить работу приемопередатчика;
- дождаться окончания приема и/или передачи текущего байта данных;
- сбросить буфер передатчика посредством сброса бита FEN регистра LCRH;
- изменить настройки регистра CR;
- разрешить работу приемопередатчика.

30 Контроллеры интерфейса SPI

В состав микроконтроллера входят восемь идентичных контроллера интерфейса SPI, реализующих интерфейс последовательной синхронной связи в режиме ведущего (мастера) и ведомого устройства и обеспечивает обмен данными с подключенным ведомым или ведущим периферийным устройством в соответствии с одним из трех протоколов фирм Motorola, National Semiconductor, Texas Instruments.

Последовательный синхронный интерфейс SPI фирмы Motorola обеспечивает полнодуплексный обмен данными по четырехпроводной линии и программное задание фазы и полярности тактового сигнала.

Интерфейс Microwire фирмы National Semiconductor обеспечивает полудуплексный обмен данными с использованием 8-битных управляющих последовательностей.

Интерфейс SSI фирмы Texas Instruments обеспечивает полнодуплексный обмен данными по четырехпроводной линии и возможность перевода линии передачи данных в третье (высокоимпедансное) состояние.

Выбор интерфейса осуществляется посредством поля FRF регистра CR.

В режиме мастера и в режиме ведомого устройства контроллер SPI обеспечивает:

- передачу данных, размещенных в буфере передатчика (восемь 32-разрядных ячеек);
- прием данных и размещение их в буфере приемника (восемь 32-разрядных ячеек).

Контроллер SPI формирует сигналы прерываний по следующим событиям:

- необходимость обслуживания буферов приемника и/или передатчика;
- переполнение буфера приемника;
- наличие данных в буфере приемника по истечении времени таймаута.

Управляющие регистры контроллеров SPI приведены в приложении А.23.

Основные характеристики контроллера SPI:

- программное управление скоростью обмена;
- программируемая длительность информационного кадра от 4 до 32 бит;
- независимое маскирование прерываний от буферов передатчика и приемника;
- поддержка прямого доступа к памяти (блок DMA).

30.1 Структура контроллера SPI

Упрощенная функциональная схема контроллера SPI с блоком синхронизации показана на рисунке 30.1.

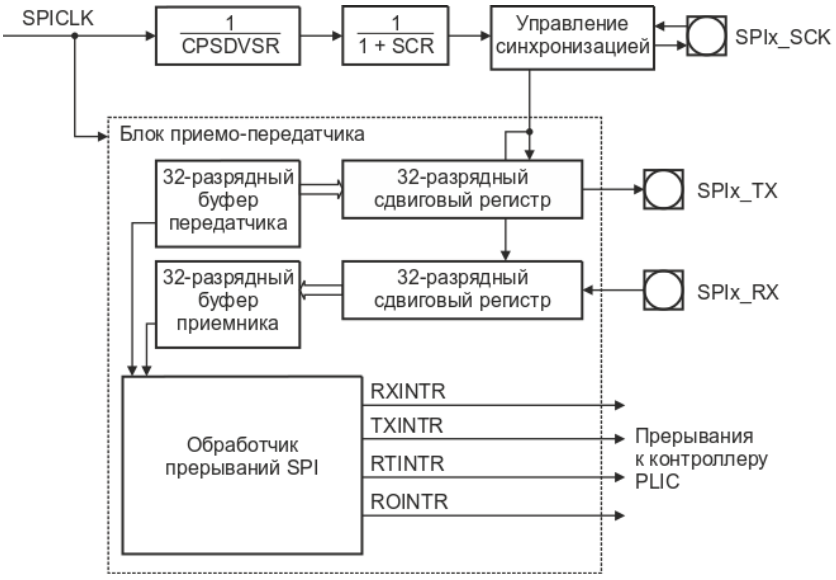


Рисунок 30.1 – Упрощенная функциональная схема контроллера SPI

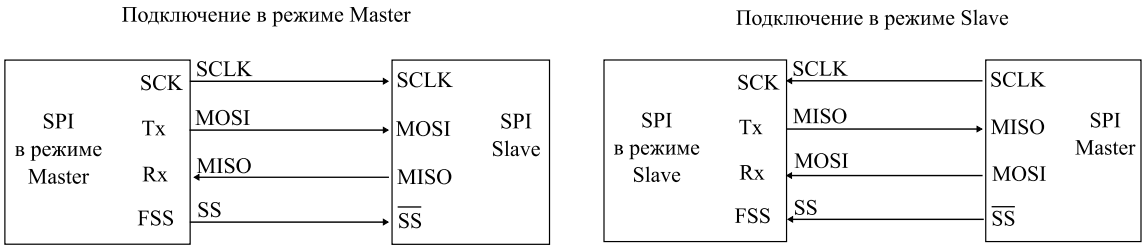


Рисунок 30.2 – Примеры схем включения контроллера SPI в режимах Master и Slave

Синхронизация

Тактирование контроллера SPI осуществляется тактовым сигналом SPICLK, который формируется на основе одного из нескольких базовых синхросигналов (см. раздел 5).

Существует ограничение на соотношение между частотами тактовых сигналов PCLK и SPICLK:

$$f_{SPICLK} \le f_{PCLK}. \quad (30.1)$$

В режиме мастера на основе сигнала SPICLK посредством двух последовательно стоящих делителей формируется сигнал тактирования передачи и приема данных с частотой f_{SPIx_SCK}, которую можно вычислить по формуле

$$f_{SPIx_SCK} = f_{SPICLK} / (CPSDVSR \times (1+SCR)), \quad (30.2)$$

где f_{SPICLK} – частота входного синхросигнала SPICLK;

CPSDVSR – коэффициент первого делителя частоты (задается в регистре CPSR);

SCR – коэффициент второго делителя частоты (задается в регистре CR).

Сформированный синхросигнал подается на вывод SPIx_SCK (где x – номер контролера 0, 1, 2, 3) микроконтроллера и далее к подключенным внешним ведомым устройствам.

В режиме ведомого значения коэффициентов делителей не важны. Внешний синхросигнал подается на вывод SPIx_SCK и тактирует прием и передачу данных.

Для корректной работы всегда должны соблюдаться условия:

- в режиме мастера для формируемого синхросигнала

$$f_{SPIx_SCK} \le f_{PCLK} / 2; \quad (30.3)$$

- в режиме ведомого для входящего внешнего синхросигнала

$$f_{SPIx_SCK} \le f_{PCLK} / 12. \quad (30.4)$$

Буферы приема и передачи

Для хранения передаваемых и принятых данных в контроллере SPI имеются два 32-разрядных буфера, организованных по типу FIFO. Каждый буфер может хранить до восьми слов данных. Буфер для передаваемых данных доступен только для записи, а буфер принятых данных – только для чтения.

Данные для передачи записываются в буфер через регистр DR. Допускается заранее заполнить буфер или записывать в него данные в течение работы контроллера. Состояние буфера можно контролировать с помощью битов TNF и TFE регистра SR. Если контроллер выключен (сброшен бит SSE регистра CR), то запись в регистр DR приведет к тому, что данные будут размещены в буфере и будут переданы после включения контроллера. Если

контроллер включен и выбран режим мастера, то в случае отсутствия данных в буфере запись в регистр DR приведет к немедленному началу передачи. Если запись данных в регистр DR происходит во время текущей передачи, то данные размещаются в буфере.

Полученные данные автоматически сохраняются в буфере принятых данных. Извлечь данные из буфера возможно чтением регистра DR. Состояние буфера можно контролировать с помощью битов RFF и RNE регистра SR.

Размер передаваемого кадра данных может быть от 4 до 32 бит, что задается полем DSS регистра CR. Если выбран размер кадра менее 32 бит, данные выравниваются по правой границе; неиспользуемые биты игнорируются.

30.2 Интерфейс прямого доступа к памяти

Контроллер SPI имеет интерфейс подключения к контроллеру прямого доступа к памяти. Работа в данном режиме контролируется регистром DMACR.

- RXDMAREQ/TXDMAREQ (для приема/для передачи) – запрос, инициируемый приемопередатчиком. Сигнал переходит в активное состояние в случае, если заполнение буфера приемника превысило заданный порог. Порог программируется индивидуально посредством поля RXIFLSEL регистра CR (для приёмника) и поля TXIFLSEL регистра CR (для передатчика);

- RXDMACLR/TXDMACLR (для приема/для передачи) – сигнал обновления запроса, инициируемый DMA контроллером, служит флагом для внутреннего генератора запросов DMA в приёмопередатчике о том, что DMA обслужил запрос и готов получить новый.

- RXDMADONE/TXDMADONE (для приема/для передачи) – сигнал окончания работы канала DMA.

Каждый инициированный приемопередатчиком сигнал запроса DMA остается активным до момента его сброса соответствующим сигналом от контроллера DMA.

После снятия сигнала сброса приемопередатчик вновь получает возможность сформировать запрос на DMA в случае выполнения описанных выше условий. Все запросы DMA снимаются после запрета работы приемопередатчика, а также в случае снятия сигнала разрешения DMA.

Также есть возможность автоматического окончания генерации запросов к DMA при установке управляющих битов DMATXSTOP или DMARXSTOP. Если данные биты установлены, пришедший сигнал окончания работы канала TXDMADONE/RXDMADONE автоматически очистит бит TXDMAE или RXDMAE соответственно.

30.3 Функционирование

Приемопередающая логика модуля, управляющие регистры и FIFO по умолчанию находятся в сбросе. Снять сброс можно путём установки бита RSTDIS в соответствующем регистре SPICFG блока RCU.

Прежде чем разрешить работу битом SSE регистра CR, следует сконфигурировать контроллер SPI посредством регистра CR, а также, если это необходимо, запрограммировать маски прерываний.

Динамическое изменение конфигурации устройства не допускается.

Для протокола SPI дополнительно задаются полярность и фаза сигнала (биты SPH и SPO регистра CR).

После разрешения работы приемопередатчик готов к обмену данными с внешними устройствами по линиям SPIx_TX (передача данных к внешнему устройству) и SPIx_RX (прием данных от внешнего устройства).

В зависимости от режима работы сигнал на линии SPIx_FSS используется либо для кадровой синхронизации (интерфейс SSI, активное состояние – высокий уровень), либо для выбора устройства в режиме ведомого (интерфейсы SPI и Microwire, активное состояние – низкий уровень).

Во всех трех режимах SPI, Microwire и SSI синхросигнал SPIx_SCK формируется только тогда, когда приемопередатчик готов к обмену данными. Перевод сигнала SPIx_SCK в неактивное состояние используется как признак таймаута приемника, то есть наличия в буфере приемника необработанных данных по истечении заданного интервала времени.

Установка бита MS регистра CR включает режим ведомого устройства. В этом режиме разрешение или запрещение передачи данных через выход SPIx_TX контролируется битом SOD. На прием синхросигнала и данных состояние этого бита влияния не оказывает.

Интерфейс SPI

Интерфейс SPI реализует полнодуплексный режим передачи данных.

Включает одну линию синхронизации SPIx_SCK, две линии приема и передачи данных SPIx_RX и SPIx_TX, а также линию выбора устройства (для режима ведомого) SPIx_FSS.

Если устройство функционирует в режиме ведомого, то на его вход SPIx_FSS должен подаваться низкий уровень сигнала в течение всей передачи кадра (последовательность передаваемых бит данных длиною от 4 до 32 бит).

Передача данных может быть одиночной (один кадр) или непрерывной (более одного кадра подряд). Данные передаются старшим битом вперед.

Особенностью интерфейса SPI является то, что в нем реализована возможность задания полярности и фазы тактового сигнала. Бит SPO регистра CR задает полярность тактового сигнала, т. е. определяет, какой уровень сигнала будет удерживаться на линии SPIx_SCK в то время, когда линия не активна.

Бит SPH задает фазу тактового сигнала. Фактически, он задает порядок считывания и выставления данных. По умолчанию бит SPH сброшен и выборка данных на линиях SPIx_TX и SPIx_RX происходит по переднему фронту сигнала синхронизации, а установка – по заднему.

Передним всегда считается тот фронт сигнала, который является началом передачи первого бита, см. рисунок 30.3.

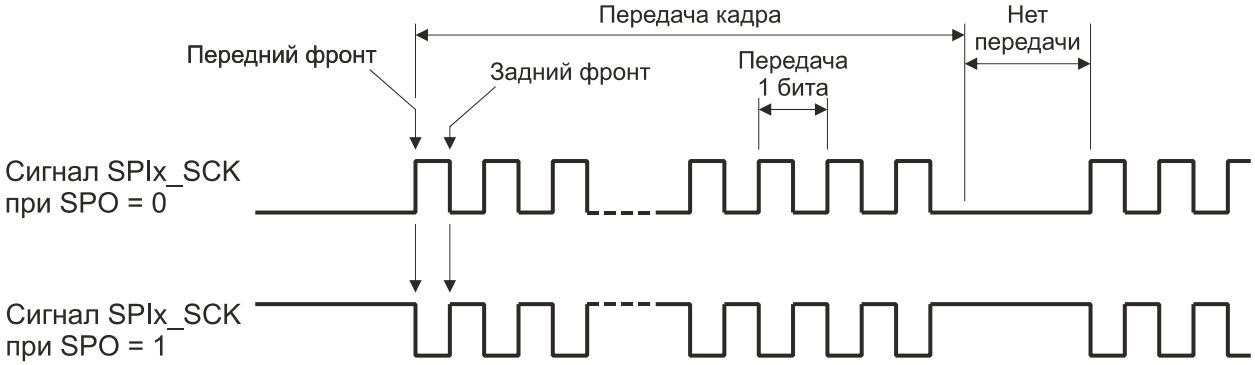


Рисунок 30.3 – Сигнал синхронизации SPIx_SCK при разных состояниях бита SPO

Комбинации битов SPO и SPH задают четыре режима обмена данными, см. рисунок 30.4.

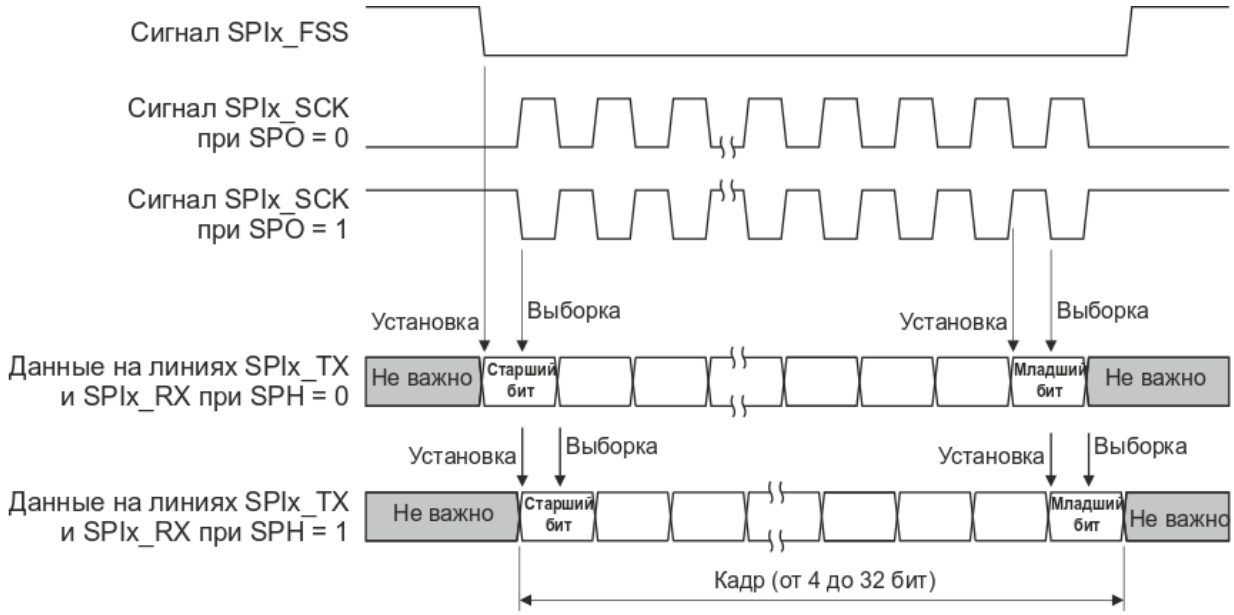


Рисунок 30.4 – Передача кадров данных в интерфейсе SPI

На рисунке 30.5 показано поведение сигналов при непрерывной передаче кадров данных.

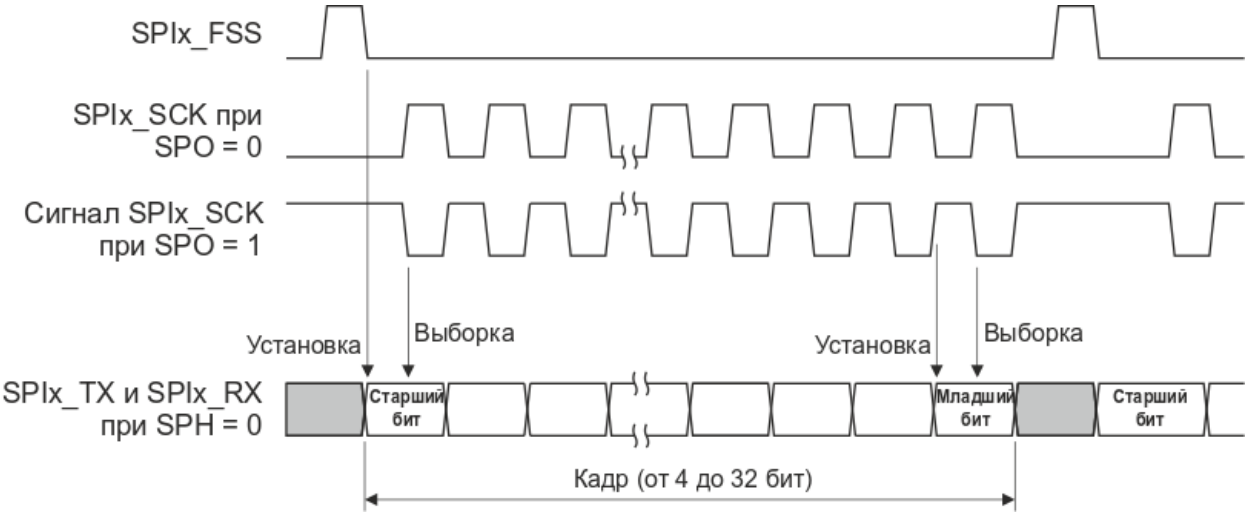


Рисунок 30.5 – Поведение сигналов при непрерывной передаче кадров данных

В режиме непрерывной передачи данных при условии SPH = 0 на линии SPIx_FSS должны формироваться импульсы между передачами кадров данных. Это связано с тем, что в этом режиме низкий уровень сигнала на линии SPIx_FSS ведомого устройства блокирует запись в сдвиговый регистр. Поэтому мастер должен переводить линию SPIx_FSS в высокий уровень по окончании передачи каждого кадра, разрешая, таким образом, запись новых данных. По окончании приема последнего бита кадра линия SPIx_FSS переводится в состояние логической единицы по истечении одного такта сигнала SPIx_SCK.

В режиме непрерывной передачи данных при условии SPH = 1 низкий уровень сигнала на линии SPIx_FSS не блокирует запись в сдвиговый регистр. Линия SPIx_FSS может оставаться в состоянии нуля в течение передачи всех кадров. Только по окончании передачи линия может быть переведена в состояние логической единицы.

Интерфейс Microwire

Интерфейс Microwire реализует полудуплексный режим передачи данных.

Включает линию синхронизации SPIx_SCK, две линии приема и передачи данных SPIx_RX и SPIx_TX, а также линию выбора устройства (для режима ведомого) SPIx_FSS.

Если устройство функционирует в режиме ведомого, то на его вход SPIx_FSS должен подаваться низкий уровень сигнала в течение всей передачи кадра (последовательность передаваемых бит данных длиною от 4 до 32 бит).

Передача данных может быть одиночной (один кадр) или непрерывной (более одного кадра подряд). Данные передаются старшим битом вперед.

Перед началом передачи линия SPIx_FSS переводится в низкое состояние.

Каждая передача начинается с передачи от мастера к ведомой 8-битной управляющей последовательности. В течение передачи этой последовательности приемник мастера не обрабатывает входящие данные. После того как управляющая последовательность передана и декодирована одним из ведомых устройств, этот ведомый выдерживает паузу в один такт синхросигнала и начинает передавать мастеру кадр данных, см. рисунок 30.6.

Выставление данных происходит по заднему фронту сигнала SPIx_SCK, а считывание – по переднему.

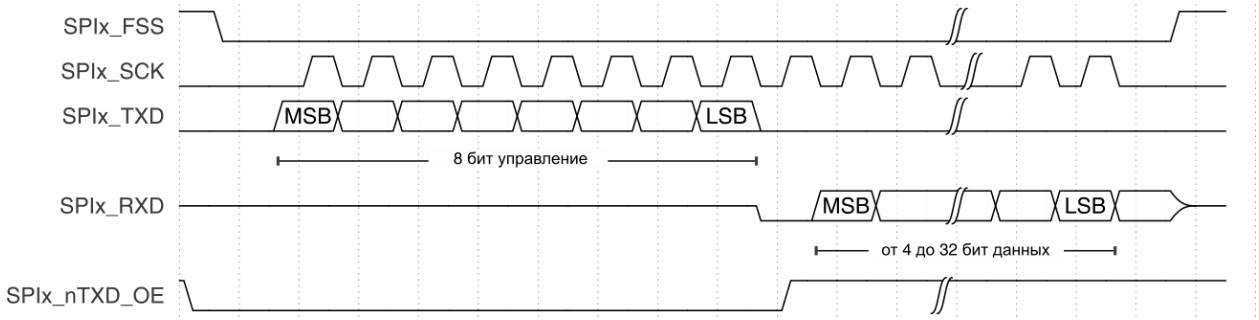


Рисунок 30.6– Передача кадра данных в интерфейсе Microwire

По окончании приема данных линия SPIx_FSS переводится в высокое состояние.

Примечание – В течение времени, когда передается управляющая последовательность и между передачами, линия SPIx_RX может находиться в третьем состоянии.

В режиме непрерывной передачи начало и завершение передачи нескольких кадров данных аналогично передаче одного кадра. Линия SPIx_FSS удерживается в нуле в течение всего сеанса передачи. По окончании передачи одного кадра данных начинается передача управляющей последовательности без паузы, см. рисунок 30.7.

Примечание – Буферы FIFO приема и передачи данных не очищаются автоматически, даже в случае запрещения работы сбросом бита SSE.

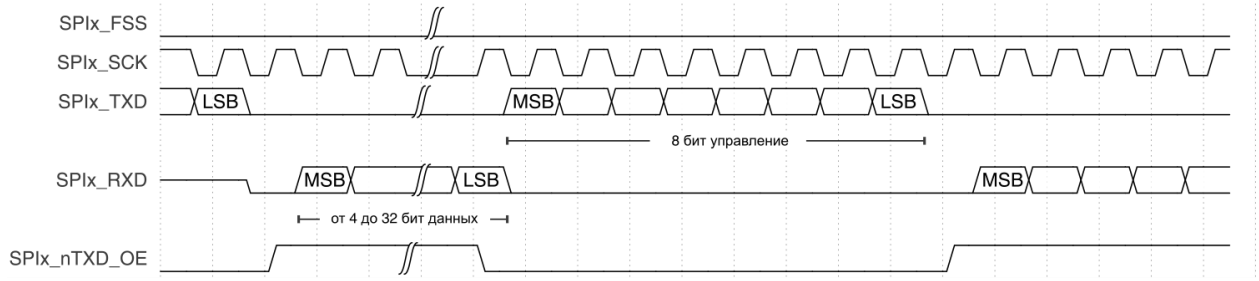


Рисунок 30.7 – Передача кадров данных в интерфейсе Microwire

Для реализации возможности объединения двух линий TxD и RxD при работе в режиме MICROWIRE модуль интерфейса имеет в составе сигнал nSSPOE. Это внутренний сигнал управления выходом линии TxD интерфейса SPI. В фазе чтения по линии RxD на

нем устанавливается высокий уровень, и он переводит вывод порта TxD в высокоимпедансное состояние.

Интерфейс SSI

Интерфейс SSI реализует полнодуплексный режим передачи данных.

Включает одну линию синхронизации SPIx_SCK, две линии приема и передачи данных SPIx_RX и SPIx_TX, а также линию выбора устройства SPIx_FSS.

Перед началом передачи каждого кадра на линии SPIx_FSS формируется импульс длительностью в один период сигнала SPIx_SCK. Далее мастер и ведомый передают данные. Установка данных производится по переднему фронту синхросигнала, а выборка – по заднему, см. рисунок 30.8. Весь цикл передачи начинается сразу же после появления хотя бы одного элемента в буфере FIFO передатчика.

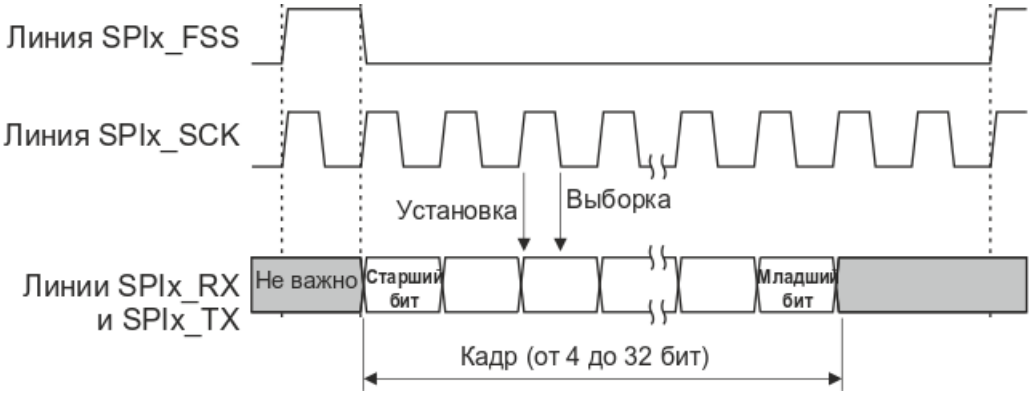


Рисунок 30.8 – Передача кадра данных в интерфейсе SSI

Режим непрерывной передачи кадров данных показан на рисунке 30.9.

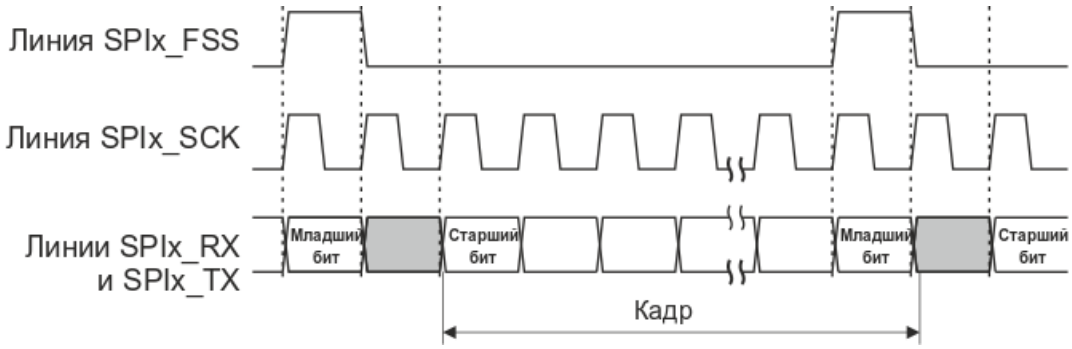


Рисунок 30.9 – Непрерывная передача кадров данных в интерфейсе SSI

30.4 Прерывания

Буфер приема и буфер передачи могут генерировать 4 независимых маскируемых запроса на прерывания:

- TXINTR – запрос на обслуживание буфера передатчика (опустошение буфера равно или ниже порога);
- RXINTR – запрос на обслуживание буфера приемника (заполнение буфера равно или выше порога);
- RTINTR – таймаут ожидания чтения данных из буфера приемника (буфер приемника не пуст и не было попыток обращения к нему в течение времени, равного передаче 32 бит);
- RORINTR – переполнение буфера приемника.

Пороги программируются индивидуально для каждого буфера посредством полей TXIFLSEL и RXIFLSEL регистра CR.

Каждый из сигналов может быть маскирован путем установки соответствующего бита в регистре маски IMSC. Для разрешения соответствующего прерывания необходимо установить бит, для запрещения прерывания – сбросить.

Источник прерывания также можно определить, считав состояние регистра RIS или регистра MIS (маскированные прерывания). Сброс прерывания осуществляется программно путём установки соответствующего бита в регистре ICR.

31 Контроллеры интерфейса QSPI

В состав микроконтроллера входят два идентичных контроллера интерфейса QSPI.

Контроллер интерфейса QSPI реализует интерфейс последовательной синхронной связи в режиме ведущего (мастера) устройства и обеспечивает обмен данными с подключенным ведомым периферийным устройством в соответствии с одним из двух протоколов: стандартный протокол передачи SPI и интерфейс памяти QuadSPI.

Управляющие регистры контроллеров QSPI приведены в приложении А.24.

Основные характеристики контроллера QSPI:

- поддержка SDR и DDR;
- программируемый код операции для непрямого режима;
- программируемый формат кадра для непрямого режима;
- встроенный FIFO для приема и передачи;
- доступ к 8-, 16- и 32-битным данным;
- генерация прерывания по событиям:
    - порогового заполнения/опустошение буфера передатчика/приемника;
    - заполнение/опустошение буфера передатчика/приемника;
    - останов передачи;
    - завершение передачи.

31.1 Структура контроллера QSPI

Упрощенная функциональная схема контроллера QSPI с блоком синхронизации показана на рисунке 31.1.

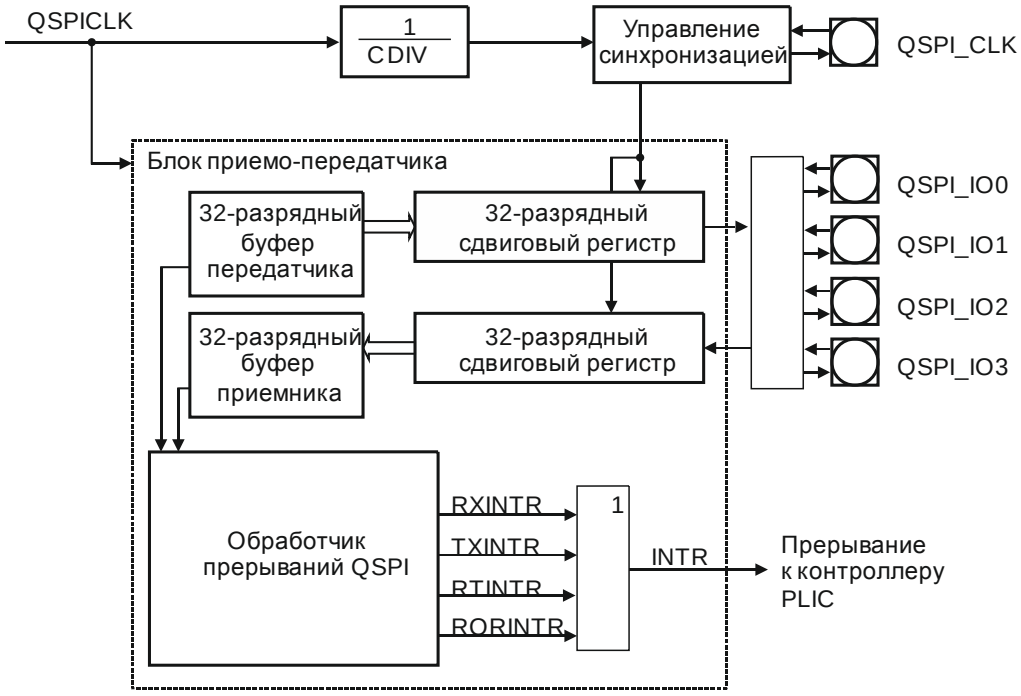


Рисунок 31.1 – Структурная схема контроллера QSPI

31.2 Режим работы SPI

QSPI может работать как стандартный SPI в режиме мастера. Выбор данного режима осуществляется с помощью битового поля FMODE регистра DCR. Данный режим может использоваться при начальном обмене с микросхемой памяти до перехода в режим QSPI.

Схема подключения внешнего устройства приведена на рисунке 31.2.

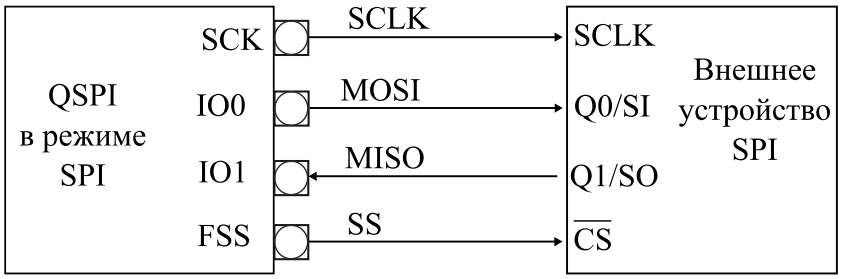


Рисунок 31.2 – Подключение внешнего устройства в режиме SPI

Выбор режима «SPI» и базовые настройки режима осуществляются установкой значений битов регистра DCR. Регистр TCR используется для конфигурации передачи, в регистр TDS записывается количество слов данных для передачи/приема, а регистр TDR используется для передачи и приема данных. В регистре DSR можно прочитать текущий статус устройства.

Порядок инициализации блока в режиме «SPI»

1 Перед конфигурацией регистра DCR сбросить бит DEN регистра HCR.

2 Записать значения в регистр DCR:

– В битовое поле FMOD записать значение 0 для выбора режима SPI;

– В битовое поле CDIV записать значение делителя;

– Настроить полярность и фазу сигнала SCLK в битовых полях CPOL и CPHA.

3 Настроить дополнительные параметры приема/передачи данных в регистре TCR:

– В битовое поле IOM записать значение 0 для выбора режима передачи по одной линии (классический режим SPI);

– Установить бит RXE для разрешения приема данных;

– Установить бит TXE для разрешения передачи данных;

– Сбросить бит DDR для отключения режима Double Data Rate;

– Сбросить бит LSBF для последовательности передачи MSB First, установить для LSB First;

– В битовое поле LEN записать значение длины слова данных (1 - 32 бит).

4 Разрешаем работу блока в режиме «SPI» установкой бита DEN регистра HCR.

Примечания

1) Изменение регистра DCR возможно только при сброшенном бите DEN регистра HCR.

2) Изменение параметров в регистре TCR возможно и при установленном бите DEN регистра HCR, но только между посылками.

Инициализация передачи/приема данных

1 Запись нового отличного от нуля значения в регистр TDS инициирует передачу/прием указанного количества слов. При этом длина каждого слова в посылке задается битовым полем LEN регистра TCR.

2 Обмен данными продолжается до завершения передачи/приема указанного количества слов, может быть прерван установкой бита DRS регистра HCR или сбросом бита DEN регистра HCR.

Примечания

1 Передача данных при установленном бите TXE регистра TCR может быть приостановлена при индикации статусного бита TFE (Tx FIFO Empty). Возобновление передачи данных происходит при записи новых данных в Tx FIFO путем записи в регистр TDR.

2 Прием данных при установленном бите RXE регистра TCR может быть приостановлен при индикации статусного бита RFF (Rx FIFO Full). Возобновление приема данных происходит при чтении данных из Rx FIFO путем чтения из регистра TDR.

3 Приостановление передачи/приема данных осуществляется переводом сигнала синхронизации SCLK в неактивное состояние в соответствии с настройкой полярности в битовом поле CPOL регистра DCR, при этом сигнал SS (Slave Select) остается активным.

31.3 Последовательность команд QSPI

QSPI взаимодействует с флэш-памятью с помощью команд. Каждая команда может включать 5 фаз:

- инструкция;
- адрес;
- дополнительные данные;
- цикл ожидания;
- данные.

Любую из этих фаз можно опционально пропускать, но по крайней мере одна из фаз инструкций, адресов, альтернативный байт или фаза данных должна присутствовать.

Сигнал FSS переключается в ноль перед началом каждой команды и снова повышается после завершения каждой команды.

Фаза инструкции

На этом этапе 8-битная инструкция, записанная в поле INST регистра QCC[7:0], отправляется во флэш-память с указанием типа операции. Несмотря на то, что большинство ИС флэш-памяти могут получать инструкции только по одному биту за раз от сигнала IO0 (режим одиночного SPI), фаза инструкции может опционально отправлять 2 бита за раз (через IO0/IO1 в режиме двойного SPI) или 4 бита за раз (через IO0/IO1/IO2/IO3 в режиме Quad SPI). Режим инструкции выбирается с помощью битового поля IMOD[1:0] регистра QCC[9:8]. Когда битовое поле IMOD = 00, фаза инструкции пропускается, и начинается последовательность команд с фазой адреса, если она присутствует.

Фаза адреса

В адресной фазе 1-4 байта отправляются во флэш-память, чтобы указать адрес операции. Количество отправляемых байтов адреса настраивается в поле ADSIZ[1:0] регистра QCC[13:12]. Байты адреса для отправки записываются в регистр QAD[31:0].

Фаза адреса может отправлять 1 бит за раз (через SO в режиме одиночного SPI), 2 бита за раз (через IO0/IO1 в режиме двойного SPI) или 4 бита за раз (через IO0/IO1/IO2/IO3 в режиме Quad SPI). Режим передачи адреса выбирается с помощью битового поля ADMOD[1:0] регистра QCC[11:10].

Когда ADMOD = 00, фаза адреса пропускается, и последовательность команд продолжается следующим этапом, если таковой имеется.

Фаза дополнительных данных

В фазе дополнительных данных 1-4 байта отправляются во флэш-память, как правило, для управления режимом работы. Количество дополнительных данных для отправки настраивается в битовом поле ABSIZ регистра QCC. Отправляемые дополнительные данные записываются в регистр QAB.

Фаза дополнительных данных может отправлять 1 бит за раз (через SO в режиме одиночного SPI), 2 бита за раз (через IO0/IO1 в режиме двойного SPI) или 4 бита за раз (через IO0/IO1/IO2/IO3 в режиме Quad SPI). Режим передачи адреса выбирается с помощью битового поля ABMOD регистра QCC.

Когда ABMOD = 00, фаза дополнительных данных пропускается, и последовательность команд переходит непосредственно к следующей фазе, если таковая имеется.

Фаза циклов ожидания

В фазе циклов ожидания предоставляется 0-31 цикла (периодов сигнала QSPI_CLK) без отправки или получения каких-либо данных, чтобы дать флэш-памяти время подготовиться к фазе данных в случае обмена на высоких частотах. Количество циклов ожидания задается в битовом поле DCYCS регистра QCC. В обоих режимах SDR и DDR продолжительность указывается как количество полных циклов QSPI_CLK.

Когда в битовое поле DCYCS записано нулевое значение, фаза циклов ожидания пропускается, и последовательность команд переходит непосредственно к фазе данных, если она присутствует.

Режим работы фазы циклов ожидания определяется битовым полем DMOD регистра QCC. В случае использования двойного или Quad SPI режима приема данных из флэш-памяти для обеспечения достаточного времени ожидания при переключении режима линий IO (из режима ввода в режим вывода) ведомого необходимо использовать хотя бы один цикл ожидания.

Фаза данных

Во время фазы данных любое количество байт может быть отправлено или получено из флэш-памяти.

Количество отправляемых/принимаемых слов данных указывается в регистре TDS.

Данные, которые должны быть отправлены во флэш-память, должны быть записаны в регистр TDR, данные, полученные из Flash-памяти можно получить, прочитав регистр TDR.

31.4 Режимы протокола интерфейса сигнала QSPI

Режим передачи по одной линии данных

Устаревший режим SPI позволяет последовательно отправлять/принимать только один бит. В этом режиме данные отправляются во флэш-память через сигнал SO, подключенный к выводу IO0. Данные, полученные из флэш-памяти, поступают через сигнал SI, подключенный к выводу IO1.

Каждая из фаз команды может быть сконфигурирована отдельно для использования однобитового режима записью значения 0x01 в битовые поля IMOD/ADMOD/ ABMOD/DMOD регистра QCC.

В каждой фазе, настроенной на передачу по одной линии, выводы контроллера QSPI функционируют в следующих режимах:

- IO0 (SO) находится в режиме вывода;
- IO1 (SI) находится в режиме ввода (высокий импеданс);
- IO2 находится в режиме вывода и установлен на «0»;

- IO3 находится в режиме вывода и принудительно установлен на «1» (для деактивации функции «удержания»).

Данная конфигурация выводов QSPI активна и в фазе циклов ожидания, если DMOD = 01.

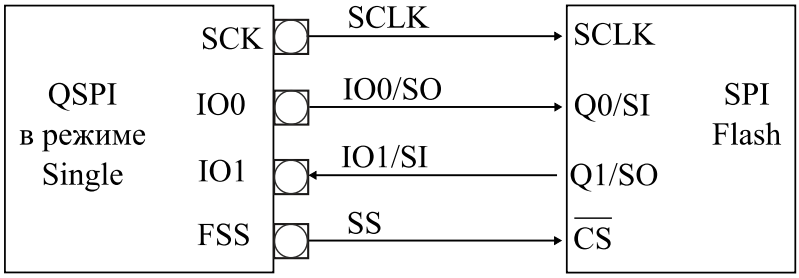


Рисунок 31.3 – Подключение внешней Flash-памяти в режиме single QSPI

Режим передачи по двум линиям данных

В режиме передачи по двум линиям (DIO), по выводам IO0/IO1 одновременно отправляются/принимаются два бита.

Каждая из фаз команды настраивается отдельно для использования режима передачи по двум линиям записью значения 0x02 в битовые поля IMOD/ADMOD/ABMOD/DMOD регистра QCC.

В режиме передачи по двум линиям во всех фазах команды, выводы контроллера QSPI функционируют в следующих режимах:

- IO0/IO1 имеют высокий импеданс (вход) во время фазы данных для операций чтения, и выходы во всех остальных случаях;
- IO2 находится в режиме вывода и установлен на «0»;
- IO3 находится в режиме вывода и принудительно установлен на «1».

В фазе циклов ожидания, при записи в битовое поле DMOD значения 1, выводы IO0/IO1 всегда имеют высокий импеданс.

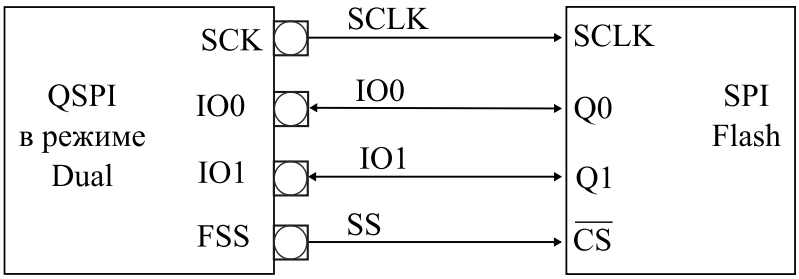


Рисунок 31.4 – Подключение внешней Flash-памяти в режиме dual QSPI

Режим передачи по четырем линиям данных

В режиме передачи по четырем линиям (QIO) по выводам IO0/IO1/IO2/IO3 одновременно отправляются/принимаются четыре бита сигнала.

Каждая из фаз команды настраивается отдельно для использования режима передачи по четырем линиям записью значения 0x03 в битовые поля IMOD/ADMOD/ABMOD/DMOD регистра QSPI_QCC.

В режиме передачи по четырем линиям во всех фазах команды, выводы контроллера QSPI IO0/IO1/IO2/IO3 находятся в состоянии высокого импеданса во время фазы данных для операций чтения и в режиме выхода во всех остальных фазах команды.

В фазе циклов ожидания, при записи в битовое поле DMOD значения 3, все выводы IO0/IO1/IO2/IO3 имеют высокий импеданс.

Выводы IO2 и IO3 используются только в режиме передачи по четырем линиям. Если ни одна из фаз команды не настроена на использование режима передачи по четырем линиям, тогда выводы микроконтроллера, соответствующие выводам интерфейса IO2 и IO3, можно использовать для других функций, даже когда QSPI активен.

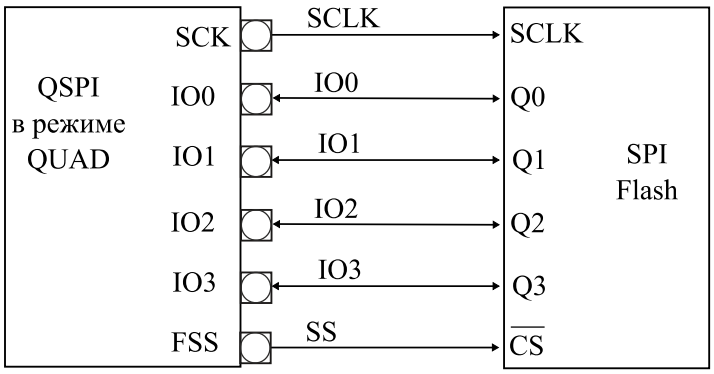


Рисунок 31.5 – Подключение внешней Flash-памяти в режиме quad QSPI

Режим SDR

По умолчанию бит DDR регистра TCR равен 0, и QSPI работает в обычном режиме передачи данных (изменение данных происходит по одному фронту сигнала QSPI_CLK).

В режиме SDR, когда QSPI управляет сигналами IO0/SO, IO1, IO2, IO3, эти сигналы переключаются только по одному фронту сигнала QSPI_CLK.

При приеме данных в режиме SDR QSPI предполагает, что флэш-память также отправит данные по одному фронту QSPI_CLK.

Режим DDR

Когда бит DDR регистра TCR установлен в 1, QSPI работает в режиме двойной скорости передачи данных (DDR).

В режиме DDR, когда QSPI управляет сигналами IO0/SO, IO1, IO2, IO3 в фазах адреса/дополнительных данных/данных, биты отправляются по обоим фронтам сигнала QSPI_CLK.

Фаза инструкции не зависит от состояния бита DDR. Инструкция всегда передается по одному фронту сигнала QSPI_CLK. При приеме данных в режиме DDR контроллер QSPI предполагает, что флэш-память также отправляет данные по обоим фронтам сигнала QSPI_CLK.

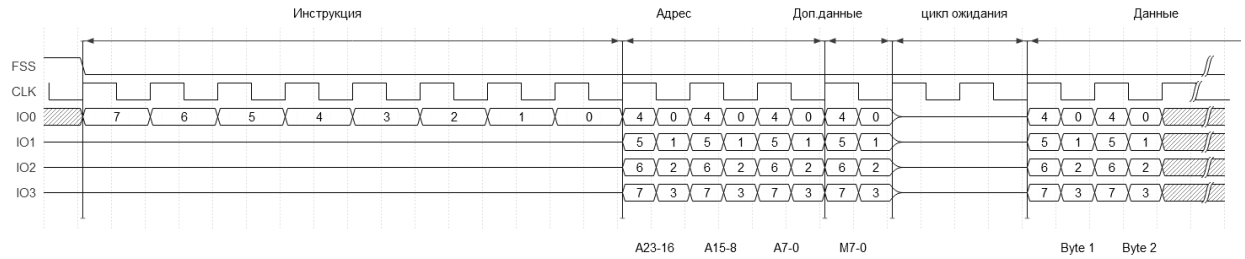


Рисунок 31.6 – Пример передачи команды DDR в режиме передачи данных по четырем линиям

31.5 Программирование QSPI

Режим работы выбирается с помощью FMOD регистра DCR.

Для выбора стандартного режима работы в битовое поле FMOD необходимо записать значение 0, а для выбора режима работы QSPI – значение 2.

Порядок действий при настройке работы в стандартном режиме SPI:

1 Указать значение делителя, полярность и фазу сигнала в регистре DCR.

2 Разрешить прием/передачу данных, указать размер слова данных и порядок передачи битов в слове данных (MSB/LSB) регистре TCR.

3 Чтение/запись данных из/в FIFO осуществляется с помощью регистра TDR.

Порядок действий при настройке работы в режиме QSPI:

1 Указать количество слов данных для чтения или записи в регистре TDS (размер слова данных задается битовым полем LEN регистра TCR).

2 Указать значение делителя, полярность и фазу сигнала в регистре DCR, а также режим и код инструкции в регистре QCC.

3 Указать необязательные дополнительные данные, которые будут отправлены сразу после фазы адреса в регистре QAB.

4 Указать целевой адрес в регистре QAD.

5 Чтение/запись данных из/в FIFO осуществляется с помощью регистра TDR.

При записи управляющих регистров TCR, FWM, IMR необходимо указать следующие настройки:

- Установить бит разрешения работы DEN в регистре HCR и биты RXE, TXE в регистре TCR;
- Указать пороговый уровень FIFO записав значения в битовые поля RHEC и TLEC регистра FWM.
- Разрешить прерывания в регистре IMR

При записи регистра конфигурации обмена QCC необходимо указать следующие параметры:

- байт инструкции, записав в битовое поле INST;
- режим инструкции, записав в битовое поле IMOD;
- режим передачи адреса, записав в битовое поле ADMOD;
- размер адреса, записав в битовое поле ADSIZ;
- режим передачи дополнительных данных, записав в битовое поле ABMOD;
- количество байт дополнительных данных, записав в битовое поле ABSIZ;
- количество циклов ожидания, записав в битовое поле DCYCS.
- режим отправки/получения данных, записав в битовое поле DMOD.

Если ни регистр адреса (QAD), ни регистр данных (TDR) не требуется обновлять для конкретной команды, то последовательность команд запускается, как только записывается регистр QCC. Это происходит только в том случае, когда в битовые поля ADMOD и DMOD записаны нулевые значения, т.е. пропускаются фазы адреса и данных.

Когда при передаче необходимо передать только фазу адреса (в битовое поле ADMOD записано ненулевое значение) и при этом пропустить фазу данных (в битовое поле DMOD записано значение 0), то передача последовательности команд начинается сразу после записи значения адреса в регистр QAD.

31.6 Бит занятости QSPI и функция прерывания

Как только QSPI начинает операцию с флэш-памятью, бит BUSY автоматически устанавливается в DSR.

Бит BUSY сбрасывается после того, как QSPI завершит запрошенную последовательность команд и FIFO опустеет.

Любую операцию можно приостановить, установив бит DRS в регистре HCR. Когда передача приостановлена, биты BUSY и TST автоматически сбрасываются. Для очистки буферов FIFO по приему/передаче необходимо установить биты RXCLR и TXCLR регистра HCR.

31.7 Поведение сигнала FSS

По умолчанию сигнал FSS имеет высокий уровень, что означает неактивное состояние выбора внешней флэш-памяти. Сигнал FSS сбрасывается перед началом операции и устанавливается после ее завершения.

31.8 Прерывания QSPI

Прерывание может быть вызвано по следующим событиям:
- пороговое переполнение буфера приемника;
- пороговое опустошение буфера передатчика;
- полное заполнение буфера приемника;
- полное опустошение буфера приемника;
- полное заполнение буфера передатчика;
- полное опустошение буфера передатчика;
- останов передачи;
- завершение передачи.

32 Контроллеры интерфейса I2S

Микроконтроллер содержит два идентичных блока интерфейса последовательной шины I2S (Inter-IC Sound), предназначенного для передачи цифровых аудиоданных.

Блок I2S поддерживает стандартный двухканальный формат I2S, форматы, выровненные по левому или правому краю (MSB/LSB-justified), а также форматы PCM Short frame и PCM Long frame.

Управляющие регистры контроллеров I2S приведены в приложении А.25.

32.1 Режимы работы

Спецификация протокола I2S поддерживает два режима работы: ведущий и ведомый.

Режим I2S определяет, какая из двух сторон (передатчик или приемник) должна подавать тактовые сигналы LRCLK и BCLK, и эти сигналы всегда передаются от ведущего к ведомому см. рисунок 32.1.

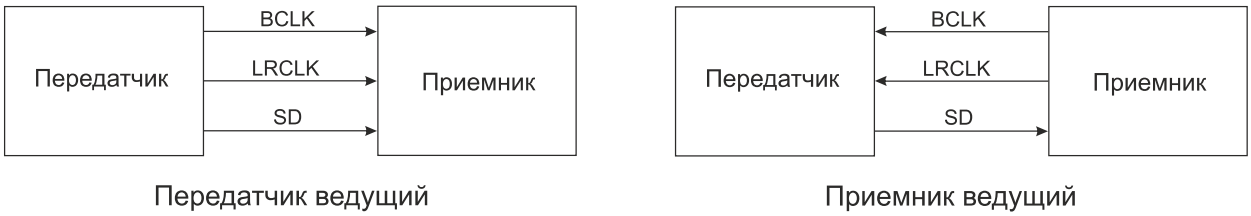


Рисунок 32.1 – Режимы работы

32.2 Передача и прием данных

Модуль I2S поддерживает передачу (TX) и прием (RX) последовательных данных. В обоих случаях последовательные данные передаются синхронно с тактовыми сигналами BCLK и LRCLK.

Данные на передачу (TX) выставляются на линию SDOUT по заднему фронту тактового сигнала BCLK, а данные RX считываются с вывода SDIN по переднему фронту сигнала BCLK. Старший бит (MSB) всегда передается первым. Пример обмена данными приведен на рисунке 32.2.

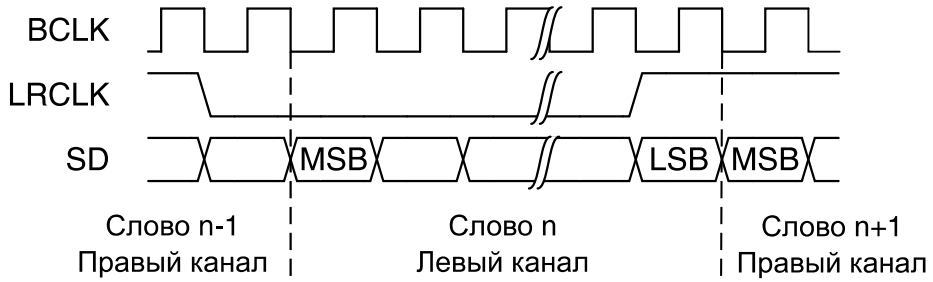


Рисунок 32.2 – Пример передачи данных

Тактовый сигнал LRCLK

Сигнал выбора левого и правого канала (LRCLK), называемый также сигналом «выбора слова» (WS) является тактовым сигналом, разделяющим кадры в последовательном потоке бит, отправляемых и получаемых по линиям SDOUT и SDIN, соответственно.

В режиме I2S каждый период сигнала LRCLK содержит пару выборок, одну для левого и одну для правого канала соответственно, причем левая выборка передается во время низкого полупериода LRCLK, за которым следует передача правой выборки во время высокого полупериода LRCLK.

В режимах выравнивания (MSB/LSB-justified) левая выборка передается во время высокого полупериода LRCK, за которым следует передача правой выборки во время низкого полупериода LRCK.

Следовательно, частота LRCLK эквивалентна частоте дискретизации звука. LRCLK всегда переключается по заднему фронту последовательного тактового сигнала BCLK.

Для тактирования внешних микросхем блок I2S может дополнительно генерировать частоту MCLK, причем после включения приемника или передатчика в режиме Мастер, MCLK будет пересинхронизирована под BCLK того блока, который включили в режиме Мастера последним.

Тактовый сигнал BCLK

Тактовый сигнал BCLK используется для синхронизации данных, передаваемых по линиям SDIN и SDOUT. Для каждого передаваемого бита данных используется один такт сигнала BCLK. При работе в режиме Master сигнал BCLK генерируется блоком I2S. При работе в ведомом режиме сигнал BCLK генерируется внешним ведущим устройством I2S.

32.3 Особенности блока I2S

Блок I2S представляет собой совокупность независимых приемника и передатчика интерфейса I2S. Основные особенности:

1. Независимость приемника и передатчика. Приемник и передатчик могут работать как одновременно, так и по отдельности по одному и тому же или разным протоколам.
2. Блоки могут работать с данными 16 – 256 бит на канал.
3. Несколько блоков I2S в одном микроконтроллере могут работать синхронно от одного из блоков, сконфигурированного в режиме «ведущего» (через биты 31..16 регистра CTRL).
4. В случае опустошения буфера на передачу, передатчик будет повторять последний полный пакет до тех пор, пока не появятся новые данные. При этом выдача информации не прекратится (что может быть критично для принимающей данные микросхемы); нежелательные шумы в момент окончания/возобновления передачи не возникнут.
5. В случае отсутствия считывания данных из входного буфера приемника, последний будет осуществлять сдвиг данных на величину пакета так, чтобы в буфере всегда находились самые последние (актуальные) отсчеты.
6. При конфигурации приемника в ведомом режиме от внешних сигналов для правильной синхронизации первый пакет (левый и правый канал) может быть потерян.
7. При включении передатчика и приемника в режиме Master разных блоков (ядер), данные на приеме будут сдвинуты относительно передачи, т.к. синхроимпульсы передатчика и приемника полностью независимы друг от друга и приемник будет стробировать входные данные по своим тактам.

32.4 Логика работы блока

До включения блока должны быть заданы регистр CLKDIV1 для передатчика и (или) CLKDIV2 для приемника (в режиме «ведущего»), FRMCTRL1 (если включается) и FRMCTRL2 (если включается). После этого можно включить блок записью «1» в нулевой бит регистра CTRL. Изменение регистров CLKDIV1 и CLKDIV2, FRMCTRL1 и FRMCTRL2 на включенном блоке игнорируется. Если блок включить с нулями в CLKDIV1 (CLKDIV2), FRMCTRL1 и FRMCTRL2, он останется в режиме простоя (полностью или соответствующий субблок). В таком случае его надо выключить и включить заново в последовательности, указанной выше.

В случае передачи данные записываются в регистр TRANSDATA. В режиме I2S каждый кадр содержит одну пару левой и правой выборок. Передатчик настроен таким образом, что первое записанное 32-битное слово после включения всегда будет относиться

к левому каналу и левая выборка передается в течение нижнего полупериода LRCLK, а затем правая выборка передается в течение высокого полупериода LRCLK. Принадлежность слов, относящихся к левому/правому каналам будет определяться величиной FRMCTRL1, и, например, при значении FRMCTRL1=0x40 (64 бит) второе записанное слово тоже будет относиться к левому каналу. В режимах выравниваний (MSB/LSB-justified) каждый кадр содержит одну пару левой и правой выборок, причем левая выборка передается в течение высокого полупериода LRCLK, а затем правая выборка передается в течение низкого периода LRCLK. Затем данные из буфера поступают в линию. После считывания последнего отсчета из пакета (нет больше данных или следующий пакет не полный) выставляется прерывание (если разрешено в CTRL). Прерывание автоматически снимается по записи новых данных в передающий буфер. Если за время передачи последнего отсчета новые данные не поступили, передача не прекращается, в бит TXFIFO регистра STATE записывается «1», а в линию повторяется последний пакет. При поступлении новых данных выдача актуальной информации возобновляется.

Непрерывный последовательный тактовый сигнал BCLKT (bit clk) и сигнал выбора канала LRCLKT (left-rigth clk) для передатчика, BCLKR, LRCLKR для приемника, сигналы SDOUT и SDIN однонаправленные, т.е. приемник и передатчик могут работать независимо друг от друга. На рисунке 32.3 представлен пример передачи/приёма по стандарту I2S Philips.

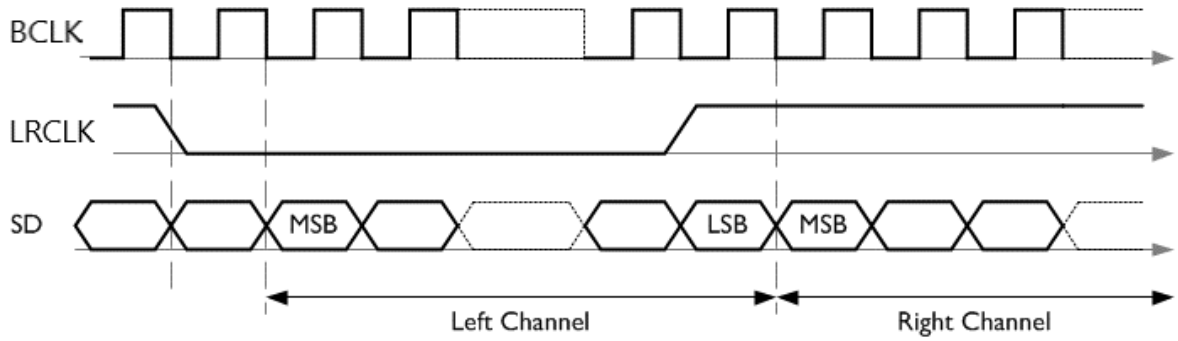


Рисунок 32.3 – пример передачи/приёма по стандарту I2S Philips

MSB/LSB-justified протокол от I2S Philips отличается тем, что LRCLK генерируется одновременно с MSB данных, то есть первый бит полукадра выборка производится по первому нарастающему фронту BCLK, следующему за фронтом LRCK (см. рисунки 32.4 – 32.5).

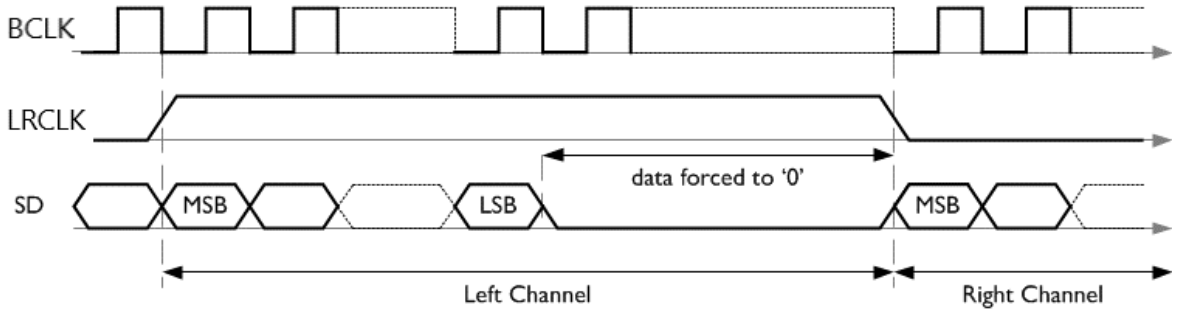


Рисунок 32.4 – пример передачи/приёма по MSB-justified протоколу со сдвигом влево.

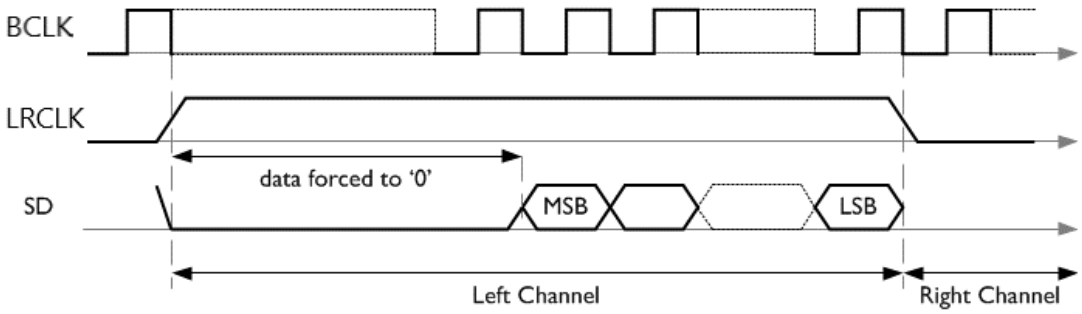


Рисунок 32.5 – пример передачи/приёма по LSB-justified протоколу со сдвигом вправо.

При использовании протокола MSB-justified (выравнивания по левому краю) каждый полукадр начинается со старшего бита значения выборки (как для отправляемых данных по SDOUT, так и для полученных по SDIN).

При использовании протокола LSB-justified (выравнивания по правому краю) каждый полукадр данных, принимаемых на SDIN, заканчивается младшим битом значения выборки, в то время как каждый полукадр данных, отправляемых по линии SDOUT, начинается со старшего бита значения выборки (то же самое, что и для MSB-justified).

В режиме работы протокола PCM Short frame sync задний фронт LRCLK показывает начало передачи на линии SD. LRCLK длится 1 такт BCLK. В режиме работы протокола PCM Long frame sync передний фронт LRCLK показывает начало передачи на линии SD. LRCLK длится 13 тактов BCLK (рисунок 32.6).

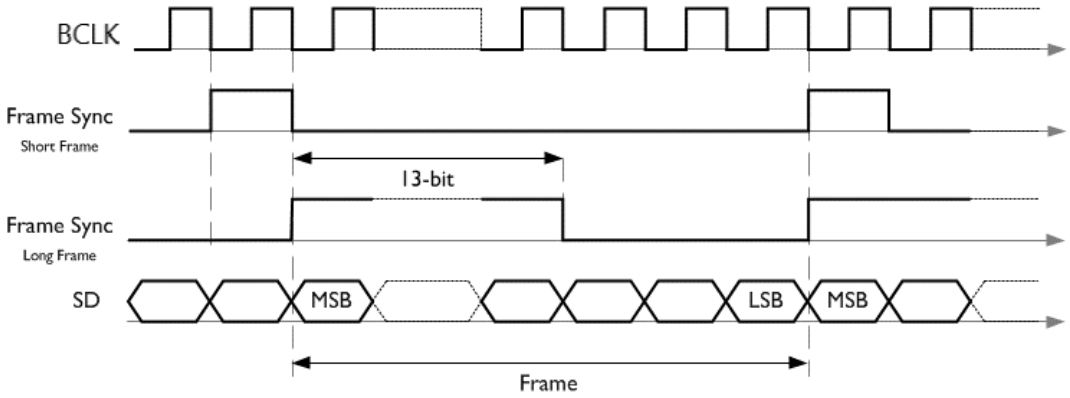


Рисунок 32.6 – пример передачи/ приёма по протоколу PCM (Frame Sync = LRCLK)

В случае приема данные сохраняются в буфер поотсчетно, причем первым всегда будет отсчет для левого канала. В режиме MSB-justified передаваемые данные перед попаданием в буфер сдвигаются влево на величину, заданную в битах [4..0] регистра SHIFTDATA. В режиме LSB-justified принимаемые данные перед попаданием в буфер сдвигаются вправо на величину, заданную в битах [12..8] регистра SHIFTDATA. В случае наполнения буфера выставляется прерывание (если разрешено в CTRL). Прерывание автоматически снимается по чтению хотя бы одного отсчета из приемного буфера. Если после заполнения данные не считываются (при работе отладчика, зависании программы или ошибок в коде), в бит RXFIFO регистра STATE записывается «1», а буфер осуществляет сдвиг данных на величину пакета так, чтобы в нем всегда находились самые последние (актуальные) отсчеты.

Для работы с некоторыми микросхемами может понадобиться выдача сигнала MCLK. Если приемник и передатчик не включены, ядро может работать как генератор MCLK. Для тактирования процесса оцифровки данных, реализована пересинхронизация по BCLK для

случая, когда вначале включили ядро (нулевым битом; MCLK тогда стартует в режиме "free running"), а затем включили передатчик/приемник в мастере. Тогда MCLK пересинхронизируется под BCLK того блока, который включили в режиме мастера последним. Блок I2S может выдавать частоту MCLK при записи в регистр MCLKDIV значения делителя, отличного от нуля.

32.5 Примеры конфигурации блока

В таблице ниже приведены примеры возможной конфигурации блока I2S.

Таблица 32.1

№ п/п	Описание	Значение управляющих регистров, hex
1	Включен только передатчик в режиме «ведущего». Выбран формат данных I2S, 24 бит полезной нагрузки, 32 бита на канал размер фрейма. Битовая скорость равна 1/16 частоты блока I2S.	CTRL = 0x03 CLKDIV1 = 0x08 FRMCTRL1 = 0x20 SHIFTDATA = 0x08
2	Включен только передатчик в режиме «ведущего». Выбран формат данных I2S, TDM, 256 бит размер фрейма (8 каналов, например, AD1938). На каждые 32 бита в линии приходится 24 бита полезной информации. Битовая скорость равна 1/16 частоты блока I2S.	CTRL = 0x03 CLKDIV1 = 0x08 FRMCTRL1 = 0x80 SHIFTDATA = 0x08
3	Включен передатчик в режиме «ведущего» и приемник в режиме «ведомого» (от внешних сигналов). И там и там выбран формат данных I2S, 32 бита на канал размер фрейма. Битовая скорость равна 1/16 частоты блока I2S (для передатчика).	CTRL = 0x01000007 CLKDIV1 = 0x08 FRMCTRL1 = 0x20 FRMCTRL2 = 0x20 SHIFTDATA = 0x88
4	Передатчики первого и второго блока I2S должны работать синхронно. Формат данных I2S, 32 бита на канал. Битовая скорость равна 1/16 частоты блока I2S. На первом блоке выставляется режим «ведущий», на втором – «ведомый» от другого блока. На внешние ножки подаются сигналы BCLK, LRCLK с первого блока.	На первом блоке: CTRL = 0x03 CLKDIV1 = 0x08 FRMCTRL1 = 0x20 На втором блоке: CTRL = 0x00020003 FRMCTRL1 = 0x20
5	Включен передатчик в режиме «ведущего» и приемник в режиме «ведущего». И там и там выбран формат данных I2S, 32 бита на канал размер фрейма при полезной нагрузке 24 бита. Битовая скорость равна 1/16 частоты блока I2S (для передатчика). Примечание – В режиме «оба ведущих» значения FRMCTRL1 и FRMCTRL2 должны совпадать.	CTRL = 0x00000007 CLKDIV1 = 0x08 CLKDIV2 = 0x08 FRMCTRL1 = 0x20 FRMCTRL2 = 0x20 SHIFTDATA = 0x88

33 Контроллеры интерфейса LIN

В состав микроконтроллера входят восемь контроллеров интерфейса LIN.

LIN (Local Interconnect Network) представляет собой последовательный коммуникационный протокол, предназначенный для создания локальных сетей обмена данными на коротких расстояниях. Сеть строится из единственного ведущего устройства (мастера) и множества ведомых устройств. Мастер контролирует все процессы, связанные с передачей данных (сообщений) по шине. Ведомые могут реагировать на сообщения мастера или других узлов сети, но отвечать они могут, только будучи адресованными мастером. Шина LIN состоит из одного канала, по которому передаются синхросигналы и данные. Физическая среда канала представляет собой однопроводную линию, подключенную через подтягивающий резистор к шине питания. Высокий уровень сигнала на шине является рецессивным и показывает, что шина свободна. Низкий уровень сигнала на шине является доминантным и показывает, что шина занята.

Управляющие регистры контроллеров LIN приведены в приложении А.26.

Кадр сообщения

Вся информация, передаваемая по шине LIN, разделяется на кадры. Кадр сообщения состоит из двух частей – заголовка, посылаемого мастером, и ответа, который может формироваться как мастером, так и ведомым (см. рисунок 33.1).

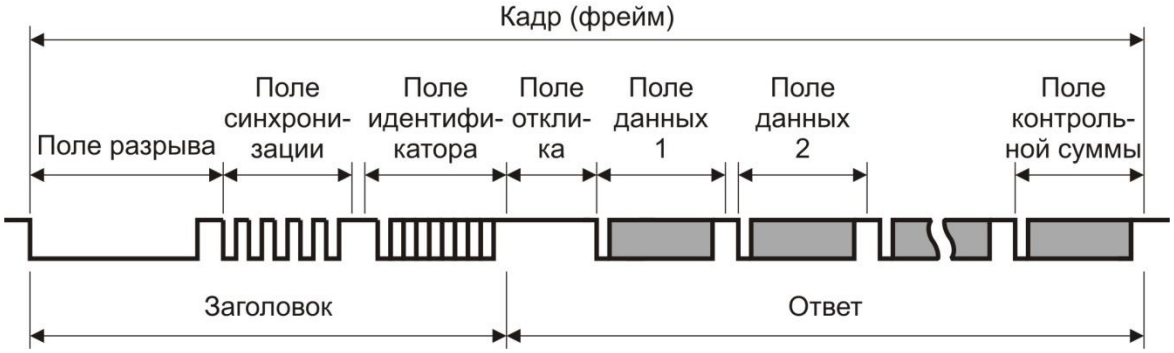


Рисунок 33.1 – Кадр сообщения

Заголовок состоит из:
- поля разрыва;
- поля синхронизации;
- поля идентификатора.

Ответ состоит из:
- поля отклика;
- полей данных;
- поля контрольной суммы.

Заголовок начинается с первого доминантного бита поля разрыва синхронизации и заканчивается со стоповым битом поля PID. Часть сообщения, начинающаяся со стопового бита PID и до стопового бита контрольной суммы, является ответом.

33.1 Байтовые поля

Каждое байтовое поле содержит 10 бит – старт бит, 8 бит данных, стоп (см. рисунок 33.2). Старт-бит указывает на начало байтового поля и является доминантным, в то время как стоп-бит является рецессивным. Восемь информационных битов могут быть как доминантными, так и рецессивными.



Рисунок 33.2 – Байтовое поле

Поле разрыва синхронизации указывает на начало передачи кадра сообщения. Это поле всегда передается мастером и указывает на необходимость ведомым подготовиться к приему поля синхронизации. Поле разрыва синхронизации состоит из двух частей –низкого уровня, длительность которого должна быть не менее 13 битов, и высокого уровня, длительностью не менее одного бита.

Длительность первой части выбрана таким образом, чтобы обеспечить различие между полем разрыва синхронизации и максимально возможной допустимой последовательностью нулевых битов в пределах одного кадра данных. Для ведомых устройств длительность поля разрыва может быть снижена до 9,5 бит.

Вторая часть поля необходима для обеспечения возможности обнаружения стартового бита следующего поля синхронизации (см. рисунок 33.3).

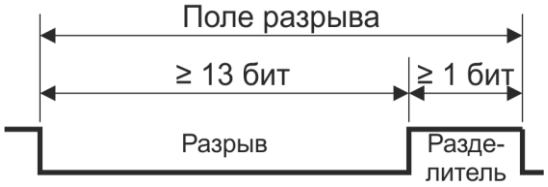


Рисунок 33.3 – Поле разрыва синхронизации

Поле синхронизации содержит сигналы, необходимые для синхронизации задающих генераторов ведомых устройств сети. Поле синхронизации является байтовым полем, имеющем значение 55h (см. рисунок 33.4).

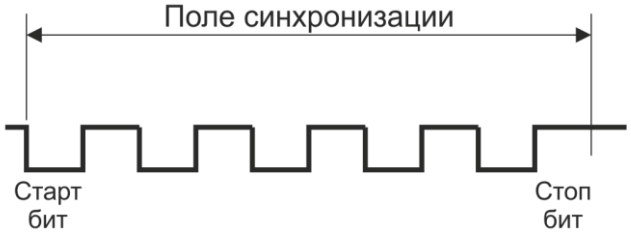


Рисунок 33.4 – Поле синхронизации

Поле синхронизации содержит пять спадающих фронтов (пять переходов от рецессивного состояния к доминантному). Эти спадающие фронты должны использоваться ведомыми устройствами для настройки скорости обмена данными.

Контроллер интерфейса LIN может работать с идентификаторами (PID) как новой версии протокола 2.X, так и версией 1.X. Основное отличие в том, что в ранних версиях протокола PID использовался в том числе и для контроля количества полей данных в кадре сообщения. В версии 2.X количество полей данных определяется пользователем и не включено в PID.

В версии протокола 1.X в идентификационном поле находится информация о содержимом и длине сообщения. Это поле разделено на три секции: четыре идентификационных бита, два служебных бита, указывающих на длину сообщения и два бита проверки на четность (см. рисунок 33.5). Таким образом, происходит деление 64 идентификационных номеров на четыре комплекта, каждый из которых содержит 16 идентификаторов.

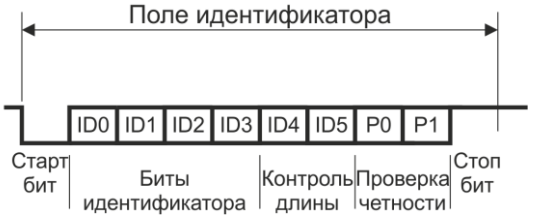


Рисунок 33.5 – Идентификационное поле в версии 1.X

В соответствии с протоколом 1.X количество полей данных в кадре данных определяется по таблице 33.1.

Таблица 33.1 – Количество полей данных в кадре данных

ID4	ID5	Количество полей данных
0	0	2
1	0	
0	1	4
1	1	8

Следует заметить, что в идентификационном поле не указывается длина сообщения, а только содержание. Это позволяет ведомым приемникам определить, все ли данные были приняты. Последние два бита идентификационного поля являются битами проверки на четность. Контроллер LIN использует алгоритм смешанного контроля четности, который гарантирует, что идентификационное поле никогда не будет состоять из одних нулей или единиц. Этот алгоритм позволяет только обнаруживать ошибки, но не исправляет их.

Биты проверки на четность формируются в соответствии с приведенными ниже формулами:

P0 = ID0 XOR ID1 XOR ID2 XOR ID4

P1 = NOT (ID1 XOR ID3 XOR ID4 XOR ID5)

В версии протокола 2.X идентификационное поле содержит только две секции – шесть идентификационных бит и два бита четности (см. рисунок 33.6). Четность вычисляется так же, как и в версии 1.X. Количество полей данных в кадре сообщения устанавливается пользователем. Сделать это можно посредством поля DATALEN регистра LINCON.

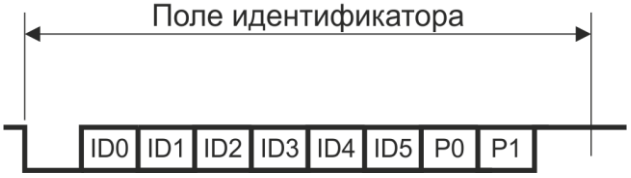


Рисунок 33.6 – Идентификационное поле в версии 2.X

В кадре сообщения может быть от одного до восьми полей данных. Их количество должно быть согласовано для всей сети. Байты данных передаются как часть поля данных. Передача данных осуществляется младшим значащим битом вперед (см. рисунок 33.7).



Рисунок 33.7 – Поле данных

Последним полем в кадре сообщения является поле контрольной суммы (см. рисунок 33.8).

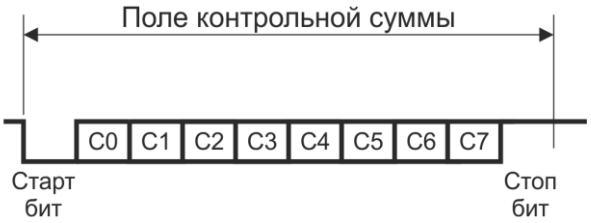


Рисунок 33.8 – Поле контрольной суммы

Контрольная сумма представляет собой инвертированную сумму по модулю 256 всех байт. Контроллер LIN может вычислять как обычную контрольную сумму (только для байт данных), так и расширенную – для байт данных и идентификатора.

Вычисление контрольной суммы представляет собой последовательное сложение элементов, а при превышении на очередном шаге значения 256 от промежуточной суммы отнимается 255. Конечный результат инвертируется. Свойства инвертированной суммы по модулю 256 таково, что если ее сложить с суммой всех байт, то результат будет равен FFh.

33.2 Режим сна

Мастер сети может быть инициатором вхождения в режим сна. Кадр сна представляет собой кадр сообщения, в котором первый байт данных равен 00h, а все другие – FFh. Ведомые устройства входят в режим сна автоматически или по сигналу от мастера. Запрос выхода из режима сна может послать любое устройство сети. Данный запрос представляет собой 5 доминантных бит (см. рисунок 33.9).

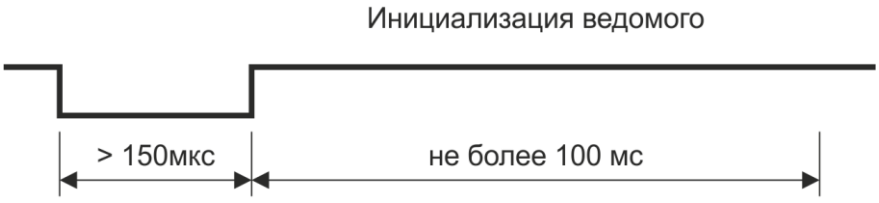


Рисунок 33.9 – Запрос выхода из сна

33.3 Регистры управления и контроля

Контроллер LIN содержит 8 регистров данных на передачу и 8 регистров данных на прием.

Регистры DTXn отведены под восемь байт данных, которые будут передаваться в кадре сообщения (каждый регистр содержит один байт).

Количество отправляемых данных задается полем UDL регистра CR. Неиспользуемые регистры можно оставлять незаполненными.

Регистры DRXn отведены под восемь байт данных, которые будут получены. Количество принимаемых байт определяется полем PID кадра сообщения или полем UDL.

Регистр CR хранит информацию о количестве байт данных, передаваемых в одном кадре сообщения, настройки протокола обмена и тип текущего кадра сообщения. Заполнять поле UDL необходимо только при использовании протокола версии 2.X.

Поскольку в версии протокола 1.X число байт данных содержится в PID, а в версии протокола 2.X – задается полем UDL, следует правильно задавать состояние бита PROTVR.

С учетом типа устройства – мастер или ведомый – поле MS определяет режим работы – передача или прием (см. таблицу 33.2).

Таблица 33.2 – Тип кадра сообщения и соответствующие действия устройства

Тип кадра сообщения, заданный полем MESTP	Действия, в зависимости от типа устройства	
	Мастер	Ведомый
IDLE	Неактивен	
HEADER	Передача	Прием
HEADER+RESPONSE		
RESPONSE RECEIVE	Прием	Прием
RESPONSE TRANSMIT	Не влияет	Передача ответа
WAKE	Передача запроса на пробуждение	
Примечание – Примененные в таблице типы означают: HEADER – в кадре передаются поля разрыва, синхронизации и идентификатора; RESPONSE – в кадре передаются поля данных и контрольной суммы.		

В регистре состояния SR находятся биты, несущие информацию о действиях контроллера LIN, его текущем состоянии и ошибках передачи. Флаги регистра SR маскируются битами регистра разрешений прерываний IM. По умолчанию биты разрешения прерываний сброшены и поэтому флаги не выставляются. Если установка того или иного флага разрешена, то одновременно с установкой генерируется прерывание. Установленные флаги не сбрасываются аппаратно. Для того, чтобы сбросить бит (флаг) в регистре SR, следует в этот бит записать единицу.

Биты TR, RV и FRS не маскируются – устанавливаются и сбрасываются аппаратно. Биты TREN и RVEN управляют только генерированием прерываний. Поле FSEN задает код режима, при входе в который будет генерироваться прерывание и выставляться соответствующий код режима в поле FRS регистра SR.

Флаг ENDTXFR устанавливается, когда устройство в режиме передачи заканчивает передавать текущий кадр сообщения. Если передается HEADER, то флаг будет установлен после стоп-бита поля PID. Если передается RESPONSE или полный кадр сообщения, то сигнал будет выработан после стопового бита поля контрольной суммы.

Биты PIDRV, DATARV и CSUMRV устанавливаются, когда устройство принимает одно из соответствующих полей. После приема поля PID необходимо решить – адресован ли принятый кадр сообщения данному устройству или следует ожидать следующий кадр. После приема поля контрольной суммы можно начинать обработку принятых данных.

Можно выполнять обработку данных и после каждого принятого поля данных, не дожидаясь контрольной суммы, но в этом случае есть риск столкнуться с ситуацией, когда контрольная сумма не совпадет с принятой, а данные уже будут в обработке.

Регистр настройки длины поля разрыва синхронизации BL используется, если устройство в сети является ведомым. В регистре содержится 16-разрядное значение минимальной длины поля разрыва синхронизации, выраженное в периодах тактового сигнала. Значение вычисляется по формуле:

$$N = \frac{Fxtal}{2 \cdot Flin} \times \frac{Nbit}{16} \quad (33.1)$$

где Fxtal – тактовая частота микроконтроллера,
Flin – скорость передачи данных блоком LIN,
Nbit – минимальное количество бит в поле разрыва синхронизации.

Пример. Требуется рассчитать длину поля для микроконтроллера с частотой 20 МГц, скоростью LIN 20 Кб/с и стандартной длиной поля разрыва в 13 бит.

$$N = \frac{20 \times 10^6}{2 \times 20 \times 10^3} \times \frac{13}{16} \quad (33.2)$$

Результат равен 406. После перевода в шестнадцатеричный формат получается значение 0196h, после чего в регистр SRR записывается вычисленное значение.

Если отклонение частоты велико, можно занижать длительность поля разрыва синхронизации, обеспечивая более уверенное его детектирование.

Регистр настройки скорости передачи SRR используется, если устройство в сети является мастером. 16-разрядное значение вычисляется по формуле:

$$N = \frac{Fxtal}{4 \cdot Flin} \quad (33.3)$$

Так, для микроконтроллера с частотой 24 МГц и скоростью LIN 20 Кб/с рассчитанное значение равно 012Ch.

Регистр PTX передаваемого PID используется для записи значения идентификатора, который будет отправлен в следующем кадре сообщения. Загружаемое значение не зависит от версии протокола и типа контрольной суммы. При необходимости два неиспользованных разряда будут автоматически заменены на биты четности при отправке.

Регистр PRX принятого PID используется для считывания полученного значения идентификатора. Обновление регистра происходит одновременно с установкой флага PIDRV. Биты четности в данный регистр не записываются, но в случае нарушения четности будет установлен флаг INVPID.

Регистр контрольной суммы CSRX используется для считывания полученного значения контрольной суммы. Обновление регистра происходит одновременно с установкой флага CSUMRV. Если значение контрольной суммы, вычисленное на основе полученных данных, не совпадает с принятой контрольной суммой, то устанавливается флаг INVCSUM.

34 Контроллеры интерфейса I2C

В состав микроконтроллера входят четыре идентичных контроллера интерфейса I2C, обеспечивающих полную поддержку двухпроводного последовательного синхронного интерфейса I2C/SMBus. Результат такой совместимости – легкое соединение со многими запоминающими устройствами и устройствами ввода-вывода, включая EEPROM, SRAM, счетчики, АЦП, ЦАП, периферийные устройства.

Управляющие регистры контроллеров I2C приведены в приложении А.27.

Функциональные возможности модуля контроллера I2C:

- совместимость с протоколами SMBus 1.1 и SMBus 2.0, ACCESS.Bus, I2C 2.1;
- поддержка скоростного/стандартного (FS) и высокоскоростного (HS) режимов;
- программирование действий мастера/ведомого;
- возможность подключения к шине нескольких ведущих устройств, т. е. поддержка режима мультимастер (MM);
- один программно задаваемый адрес;
- поддержка режима множественного адреса и безадресного режима;
- 7- или 10-битная адресация ведомого;
- поддержка адреса общего вызова.

Особые возможности протокола SMBus:

- отслеживание времени простоя линии SCL;
- наличие функции отслеживания ошибок в пакетах данных PEC с использованием метода расчета контрольной суммы CRC;
- поддержка адреса отклика мастера;
- поддержка полинга и контроля прерываний.

34.1 Протокол шины

Протокол I2C использует двухпроводной интерфейс для двухсторонней связи между устройствами, подключенными к шине. Двунаправленная шина состоит из двух линий: данных SDA и тактового сигнала SCL. Эти линии подключены к источнику питания через подтягивающие резисторы. Шинные формирователи любых устройств, подключаемых к шине, выполняются по схеме с открытым коллектором или открытым стоком. Устройства могут выставить только низкий уровень на соответствующей линии. Следовательно, обе линии SDA и SCL реализуют функцию «монтажное И». Протокол поддерживает режим мультимастер, в котором шина может контролироваться одним или несколькими устройствами из подключенных к шине. Каждое устройство, подключенное к шине, имеет свой адрес и может быть как приемником, так и передатчиком (некоторые только приемниками).

Операции с данными

Устройство, которое начинает передачу данных, становится мастером. Мастер генерирует тактовый сигнал SCL, а также инициирует и завершает передачу данных по шине. За один такт сигнала SCL передается один бит данных по линии SDA, см. рисунок 34.1.

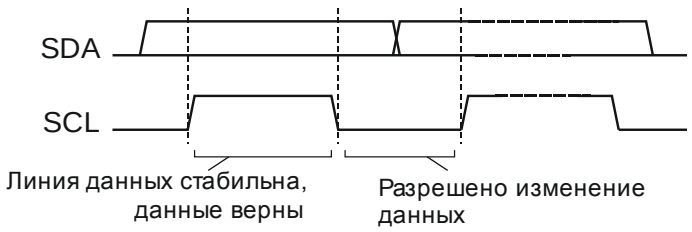


Рисунок 34.1 – Передача бита данных

Данные валидны (верны), пока уровень сигнала на линии SCL высокий. Когда на линии SCL низкий уровень сигнала, данные могут меняться.

Старт и стоп

Состояние старта формируется тогда, когда на линии SCL держится высокий уровень сигнала, а на линии SDA возникает перепад уровня сигнала из высокого уровня в низкий, см. рисунок 34.2.

Состояние стопа (останова) формируется тогда, когда на линии SCL держится высокий уровень сигнала, а на линии SDA возникает перепад уровня сигнала из низкого уровня в высокий, см. рисунок 34.2.

Состояния старта и стопа формирует только мастер.

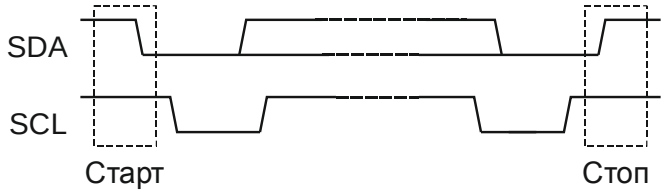


Рисунок 34.2 – Состояния старта и стопа

После того, как сформировано состояние старта, шина считается занятой и другие устройства не должны пытаться управлять ею. Шина считается занятой до тех пор, пока не будет сформировано состояние стопа. В середине передачи может быть сформировано состояние повторного старта, если мастеру нужно обратиться к другому ведомому или если требуется изменение направления передачи данных без потери контроля над шиной, см. рисунок 34.3.

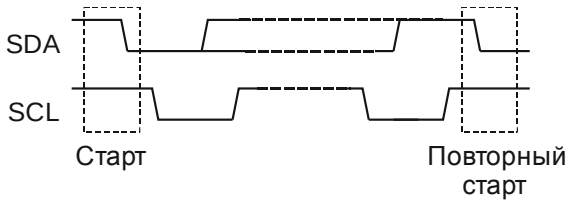


Рисунок 34.3 – Состояние повторного старта

Арбитраж

Арбитраж выполняется в момент времени, когда на линии SCL находится «1». Два устройства могут сгенерировать стартовое состояние в одно и то же время. Далее арбитраж будет продолжаться до тех пор, пока одно из устройств сформирует «0», а другое – «1» на линии SDA. Устройство, которое установило «1» на линии SDA, проигрывает арбитраж. На рисунке 34.4 приведен пример арбитража.

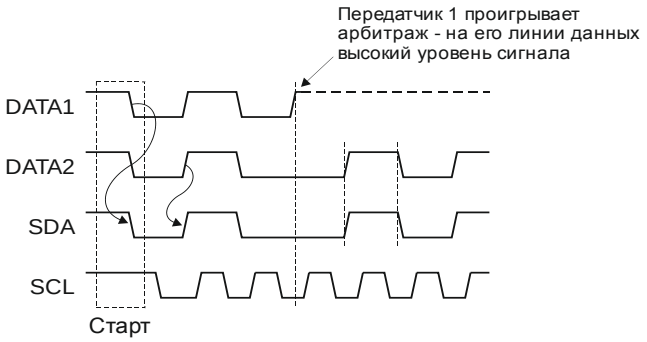


Рисунок 34.4 – Арбитраж на линии SDA

Два устройства передают свои данные DATA1 и DATA2 на линию SDA. В момент времени, когда очередной бит данных DATA1 равен «1», а бит данных DATA2 равен «0», второе устройство выигрывает арбитраж и продолжает передачу своих данных, а первое устройство прекращает передачу.

Если устройство проигрывает арбитраж во время передачи первого байта после старта (во время передачи адреса ведомого), оно становится ведомым приемником и мониторит передаваемый адрес на случай совпадения. Арбитраж также может быть проигран в режиме мастера приемника во время квитирования или в режиме ведомого передатчика во время ответа на адрес отклика на сигнал предупреждения.

В случае проигрывания арбитража в битовом поле MODE регистра ST устанавливается соответствующий код и генерируется прерывание.

Синхронизация

Синхронизация тактовых сигналов разных устройств, подключенных к шине I2C, реализуется в случаях, когда несколько устройств являются мастерами, и выполняется с использованием той особенности, что линия SCL реализована как монтажное «И» линий тактовых сигналов этих устройств. Для примера рассмотрим синхронизацию двух мастеров с линиями тактовых сигналов CLK1 и CLK2, см. рисунок 34.5.

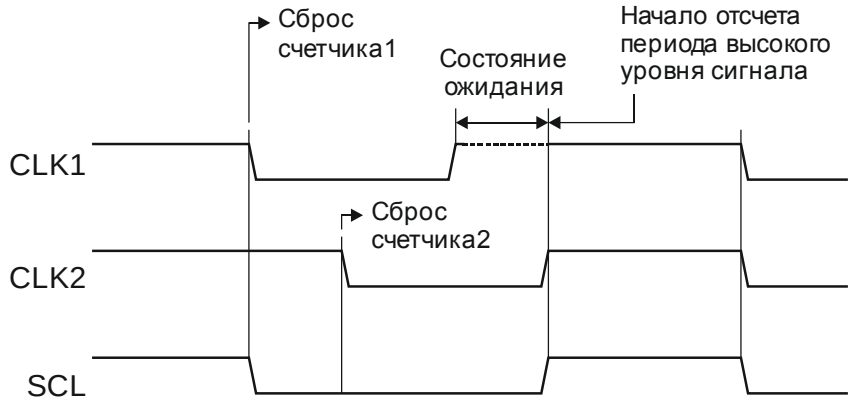


Рисунок 34.5 – Синхронизация

Линия SCL переводится в состояние «0» сразу, как только один из мастеров выставляет на своей линии тактового сигнала низкий уровень сигнала (сигнал CLK1 на рисунке 34.5). При этом его внутренний счетчик длительности низкого уровня сигнала сбрасывается и начинает отсчет. Второй мастер выставляет низкий уровень позже, и его счетчик также сбрасывается (сигнал CLK2 на рисунке 34.5).

Как только внутренний счетчик первого мастера переполнится, мастер выставит на линии CLK1 высокий уровень сигнала. Тем не менее, линия SCL будет по-прежнему оставаться в состоянии «0», удерживаемая вторым мастером. В связи с этим, первый мастер перейдет в состояние ожидания, см. рисунок 34.5. Когда переполнится счетчик второго мастера, он выставит на линии CLK2 высокий уровень сигнала, и в этот момент линия SCL перейдет в состояние «1». С этого момента внутренние счетчики длительности высокого уровня сигнала обоих мастеров начнут синхронный отсчет.

Каждая передача данных состоит из начального состояния «старт», состояний передач битов и состояния «стоп». Данные передаются старшим битом MSB вперед. Передача каждого байта завершается квитированием, т. е. приемник подтверждает окончание приема сигналом подтверждения ACK. Ведомое устройство может увеличивать паузу между тактовыми импульсами, удерживая на линии SCL сигнал низкого уровня, пока происходит

обработка принятых данных или подготовка данных для следующей передачи. Этот процесс может происходить после передачи любого бита/байта, см. рисунок 34.6.

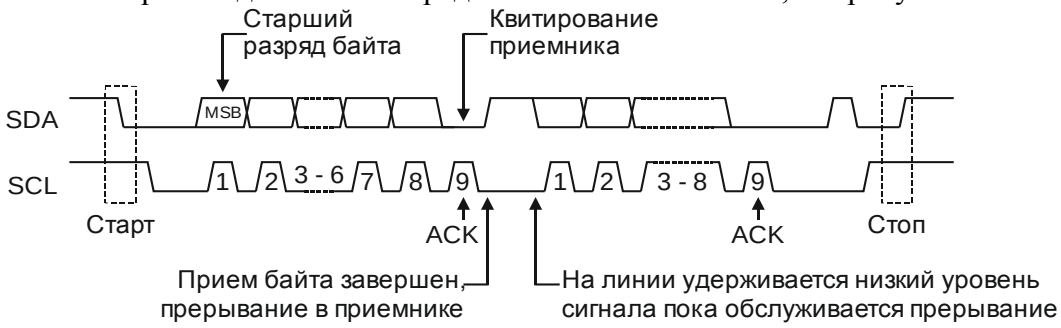


Рисунок 34.6 – Передача данных

Квитирование

Каждый байт посылки должен быть завершен квитированием, т. е. ответом на прием сигнала запроса подтверждения приема (бит ACK). На рисунках 34.6 и 34.7 показано положение момента квитирования в пределах посылки.

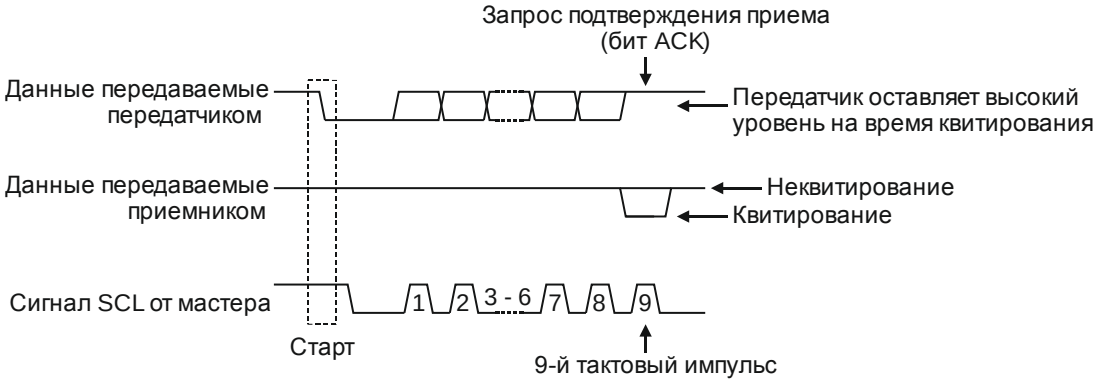


Рисунок 34.7 – Квитирование и неквитирование бита подтверждения ACK

Бит запроса подтверждения приема генерируется мастером. Передатчик (мастер или ведомый) в момент девятого такта синхросигнала оставляет линию SDA в состоянии «1» (бит ACK). В свою очередь, приемник должен сбросить линию SDA в «0» в течение времени, пока на линии SCL удерживается высокий уровень девятого импульса тактового сигнала, т. е. квитировать прием, см. рисунок 34.7. Если приемник не отвечает на запрос подтверждения и не подтверждает прием байта, то он оставляет линию SDA без изменений в состоянии «1», т. е. не квитирует прием.

Примечание – Все устройства, подсоединенные к шине I2C, в обязательном порядке должны квитировать бит ACK при получении байта с их собственным адресом. Этот механизм используется для отслеживания наличия отключившихся (самостоятельно или по каким-то причинам) от шины устройств.

Ведомое устройство имеет право не квитировать бит ACK в следующих случаях:

- если ведомый не может принять данные или он занят. Мастер, обнаружив неквитирование байта, должен сгенерировать состояние стопа и прервать передачу. Как альтернатива, ведомый может затянуть период низкого уровня сигнала тактирования на линии SCL для завершения своих операций и продолжить передачу;
- если ведомый обнаружил некорректную команду или некорректные данные. В этом случае ведомый должен неквитировать принятый байт. Мастер, обнаружив неквитирование байта, должен сгенерировать состояние стопа и повторить передачу;
- если мастер функционирует как приемник, то, приняв байт, он должен сообщить ведомому об окончании данных неквитированием бита ACK, посланного ведомым. После

этого ведомый передатчик должен освободить линию SDA для того, чтобы мастер смог сгенерировать состояние завершения передачи (состояние стопа).

Формат передачи данных с 7-битной адресацией

На рисунке 34.8 показана передача адреса и двух байт данных. Каждому устройству, подключенному к шине, присваивается уникальный 7-битный адрес. Первые семь бит, передаваемые после старта, представляют собой адрес ведомого, восьмой бит R/W# определяет направление передачи – от ведомого (чтение, если R/W# = «1») или к ведомому (запись, если R/W# = «0»).

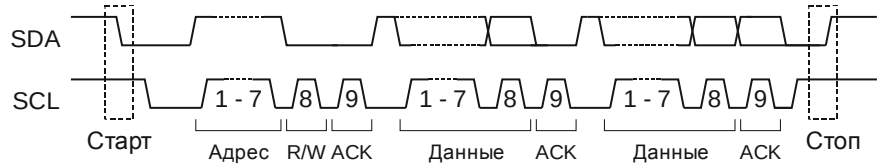


Рисунок 34.8 – Передача данных с 7-битной адресацией

Каждый ведомый, получивший байт адреса, сравнивает его со своим собственным адресом. Если адрес распознается как «свой», ведомый квитирует прием и далее, в зависимости от состояния бита R/W#, становится передатчиком или приемником.

Протокол SMBus/I2C позволяет генерировать адрес общего вызова для одновременного обращения ко всем устройствам, подключенным к шине. Первым передается адрес общего вызова (00h), затем следует байт назначения общего вызова. Ведомые, которые ожидают данные, квитируют этот байт и становятся приемниками, остальные игнорируют общий вызов.

Протокол SMBus/I2C поддерживает уникальную функцию – распознавание адреса отклика на сигнал предупреждения (Alert Response Address – ARA). В системах с несколькими ведомыми каждое устройство может послать мастеру сигнал предупреждения. Для этого используется дополнительная третья линия ALERT#, физически идентичная линиям SDA и SCL, реализованная по принципу монтажное «И». К этой линии также подключаются все устройства. Когда какому-то ведомому (или нескольким ведомым) необходимо обратиться к мастеру, он (или они) выставляет (выставляют) на линии ALERT# низкий уровень сигнала – это сигнал предупреждения, см. рисунок 34.9.

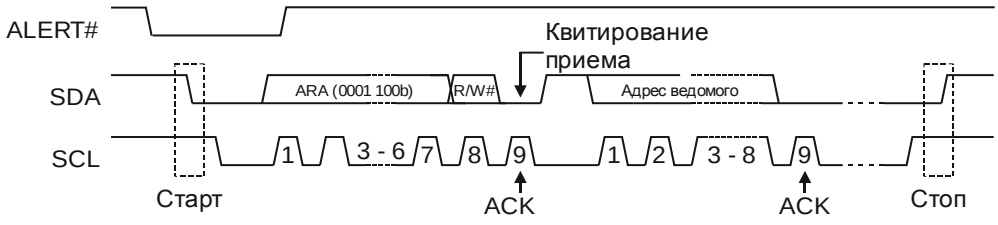


Рисунок 34.9 – Передача адреса отклика на сигнал предупреждения

Мастер, обнаружив «0» на линии ALERT#, обращается ко всем ведомым, посылая адрес отклика на сигнал предупреждения ARA. Адрес состоит из семи битов (000_1100b) и бита R/W# = «1» (чтение). Ведомый, который отправил сигнал предупреждения, получив сигнал ARA, квитирует его и затем отправляет свой 7-битный адрес (восьмой бит может быть как «0», так и «1»), сообщая, таким образом, ведомому, какое именно устройство посылало сигнал предупреждения. Кроме этого, ведомый, который выставлял «0» на линии ALERT#, должен перестать удерживать линию, чтобы на ней установился высокий уровень сигнала. В том случае, если несколько устройств посылали сигнал предупреждения, то после получения ARA, свой адрес передает то устройство, которое захватывает шину по

стандартным правилам арбитража. Если после обслуживания ведомого мастер все еще обнаруживает на линии ALERT# низкий уровень сигнала, он понимает это как то, что сигнал предупреждения посылался несколькими ведомыми. Мастер снова отправляет сигнал ARA и затем общается со следующим ведомым. Появление на линии ALERT# высокого уровня сигнала означает, что все ведомые, которые требовали обращения, обслужены.

Примечание – Описываемый в настоящем техническом описании модуль I2C не имеет выделенной линии ALERT#. При необходимости, пользователь может задействовать свободный вывод микроконтроллера и программно реализовать возможность передачи сигнала предупреждения от ведомого к мастеру. В свою очередь, функция распознавания адреса отклика ARA и последующей отправки собственного адреса реализована полностью. Включить функцию можно установкой бита SMBARE в регистре CTL0.

Формат передачи данных с 10-битной адресацией

10-битная адресация позволяет адресовать до 1024 ведомых устройств с использованием резервной комбинации 1111_0xxb, которая передается по линии SDA сразу после старта. 10-битный формат полностью совместим с 7-битным форматом и может использоваться одновременно с ним, что позволяет соединять по шине I2C устройства с разной адресацией.

Основной идеей формата является передача 10-битного адреса в двух первых байтах, следующих сразу после старта. В первом байте передается значение 1111_0xxb, где «xx» – это два старших бита адреса и бит R/W# (на рисунке 34.10 обозначен символом «W» – запись), который должен быть равен «0», чтобы ведомый понял, что в следующем байте будут переданы остальные 8 бит адреса. Во втором байте передаются 8 бит адреса, см. рисунок 34.10.

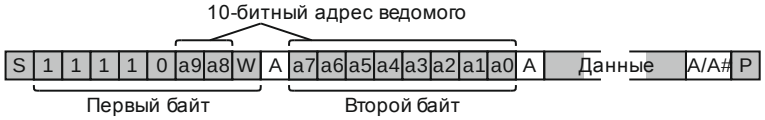


Рисунок 34.10 – Передача данных ведомому с 10-битным адресом (для расшифровки применяемых обозначений следует обратиться к таблице 34.1)

Чтобы осуществить чтение ведомого, которого адресует мастер после второго байта адреса, следует отправить бит повторного старта и затем комбинацию 1111_0xxb и бит R/W# (обозначен символом «R» – чтение), который на этот раз равен «1», см. рисунок 34.11.

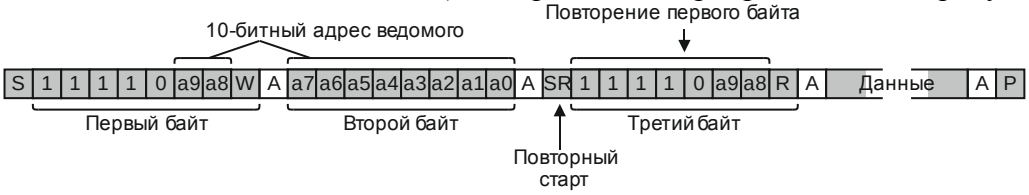
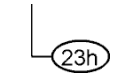


Рисунок 34.11 – Получение данных от ведомого с 10-битным адресом

На рисунках 34.10 и 34.11 биты посылки условно обозначены буквами S, W и др. или состояния битов указаны непосредственно «0» или «1». В дальнейшем на подобных рисунках, поясняющих содержимое посылки при передаче или приеме данных, будут применяться такие же и другие обозначения. Все обозначения, которые будут использоваться, указаны в таблице 34.1 с подробными пояснениями.

Таблица 34.1 – Условные обозначения, принятые на рисунках, показывающих содержимое посылок данных или адресов на линии SDA

Обозначение	Расшифровка обозначения
S	Состояние старта. Символом «S» обозначается стартовый бит посылки
SR	Состояние повторного старта
R/W	Бит указания направления передачи. В тексте настоящего описания он упоминается как R/W#. Наличие этого обозначения в бите посылки указывает на то, что этот бит может быть равен как «0», так и «1»
R	Частный случай обозначения бита направления передачи R/W#. Если в обозначении бита стоит символ «R», то это указывает на то, что в данной посылке бит R/W# должен быть равен «1», т. е. направление передачи данных происходит от ведомого к мастеру (чтение)
W	Частный случай обозначения бита направления передачи R/W#. Если в обозначении бита стоит символ «W», то это указывает на то, что в данной посылке бит R/W# должен быть равен «0», т. е. направление передачи данных происходит от мастера к ведомому (запись)
A/A#	Бит квитированого/неквитированого приема, посылаемый приемником в ответ на запрос передатчика подтвердить прием. Наличие этого обозначения в бите посылки указывает на то, что этот бит может быть равен как «0», так и «1»
A	Частный случай обозначения бита A/A#. Если в обозначении бита стоит символ «A», то это указывает на то, что в данной посылке в ответ на запрос подтверждения приема байта произошло квитирование, т. е. приемник установил линию SDA в «0». В тексте настоящего описания квитированный бит запроса подтверждения приема обозначается как ACK
A#	Частный случай обозначения бита A/A#. Если в обозначении бита стоит символ «A#», то это указывает на то, что в данной посылке в ответ на запрос подтверждения приема байта произошло неквитирование, т. е. приемник не изменил линию SDA и оставил ее в состоянии «1». В тексте настоящего описания, неквитированный бит запроса подтверждения приема обозначается как NACK
P	Состояние окончания передачи. Символом «P» обозначается стоповый бит посылки
Код мастера	8-битный код мастера. Значение 0000_1xxxb, где «xxx» – уникальный код каждого мастера в системе нескольких устройств
Адрес	7-битный адрес ведомого, передаваемый мастером
Адрес ведомого	Адрес ведомого, передаваемый во втором байте посылки. В режиме HS это 7-битный адрес, на что указывает идущий следом бит R/W#, в остальных случаях это восемь младших бит 10-битного адреса
Данные	Байт или несколько байт данных
GC	Байт адреса общего вызова (0000_0000b)
AR	Адрес отклика (0001_100b)
	Изображение числа в овале с линией, прикрепляющей его к изображению передачи битов, обозначает код операции и указывает момент, в который этот код записывается в поле MODE регистра ST
Цветное поле	Серым цветом обозначены биты, передаваемые от мастера к ведомому
Белое поле	Белым цветом обозначены биты, передаваемые от ведомого к мастеру

34.2 Функциональное описание

Структурная схема модуля I2C представлена на рисунке 34.12. Далее приводится краткое описание назначения блоков модуля.

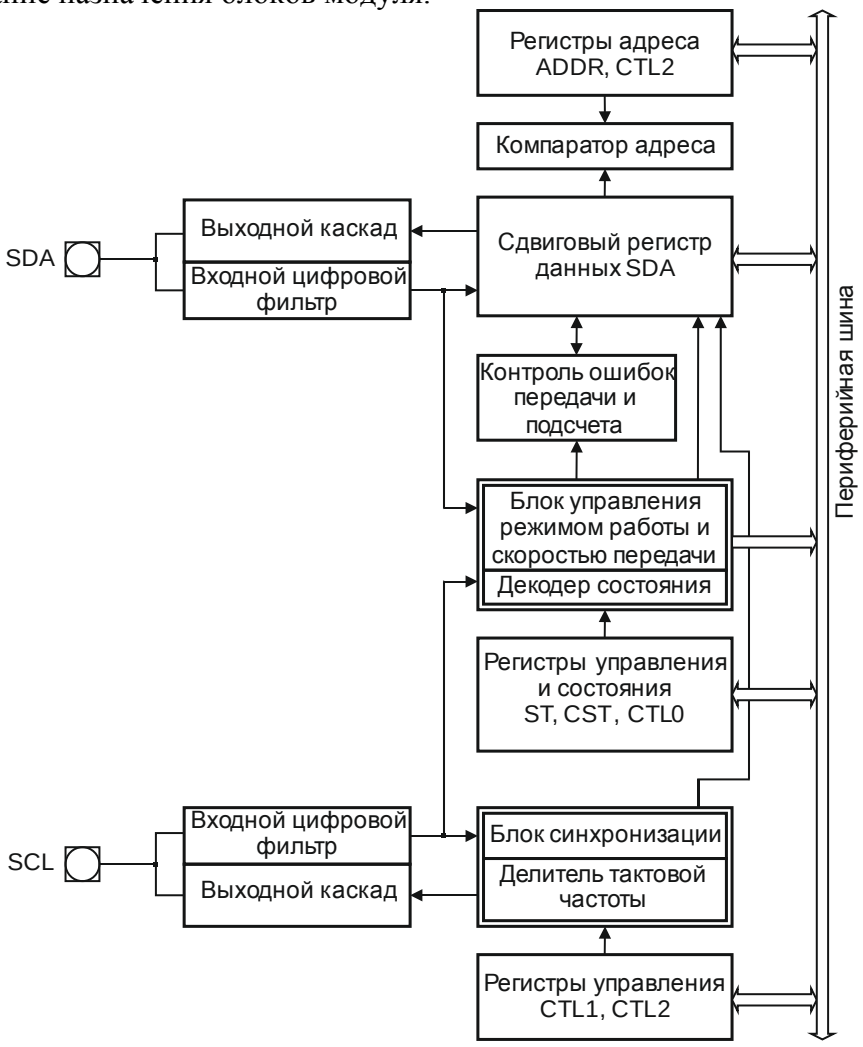


Рисунок 34.12 – Структурная схема модуля I2C

Входные и выходные каскады линий SDA и SCL

Для обеих линий используются входные шумовые фильтры. В режиме FS эти фильтры подавляют любые импульсы входного сигнала, длительность которых не превышает один такт системного синхросигнала. Выходные каскады включают в себя понижающие (до уровня «0») устройства с открытым стоком. Функционирование входных и выходных каскадов зависит от состояния модуля I2C, т. е. модуль включен или выключен.

Управление режимом работы и опрос состояния

Управление модулем I2C осуществляют блоки управления режимом работы и скоростью передачи регистров управления и состояния. В состав этих блоков входят следующие регистры:

- ST – содержит биты, отражающие текущую конфигурацию модуля I2C (мастер или ведомый, передатчик или приемник) и бит флага прерывания;
- CST – является одновременно регистром управления шиной и регистром состояния шины;
- CTL0 – управляет генерированием состояний старта, повторного старта и останова, а также квитированием;
- CTL1 и CTL2 – устанавливают параметры тактового сигнала в режиме мастера и контролируют режим 10-битной адресации.

Регистры адреса и компаратор адреса

В регистр адреса ADDR может быть записан 7-битный адрес, который является адресом устройства при работе его в режиме ведомого. Распознавание адреса включается установкой бита SAEN.

Также возможна настройка, при которой устройство будет иметь несколько адресов и отвечать на любой адрес из списка. Дополнительные адреса заносятся в массив EADDR. Поддерживается как 7-битный, так и 10-битный адрес. Для активации каждого дополнительного адреса, должен быть установлен бит SAEN в соответствующем регистре EADDR. Таким образом, имеется возможность присвоить устройству до 9 различных адресов.

Дополнительно имеется возможность разрешить устройству отвечать на любой полученный адрес. Для этого необходимо установить бит UNADDR регистра CTL0.

Компаратор адреса сравнивает принятый 7-битный адрес со значением, хранящимся в поле ADDR. Если разрешено распознавание адреса общего вызова (установлен бит GCMEN регистра CTL0), то компаратор сравнивает принятый адрес со значением 0000_000b. Если разрешено распознавание адреса отклика на сигнал предупреждения (установлен бит SMBARE регистра CTL0), то компаратор сравнивает принятый адрес со значением 0001_100b.

Если включен режим 10-битной адресации (одновременно установлены биты SAEN и S10EN регистров ADDR и CTL2 соответственно), компаратор сравнивает старшие пять битов первого полученного байта со значением 11110b, а следующие два бита – со значением второго и первого битов поля S10ADR регистра CTL2. Старший бит второго полученного байта сравнивается со значением нулевого бита поля S10ADR, а оставшиеся семь битов – со значением битового поля ADDR регистра ADDR, см. рисунок 34.13.

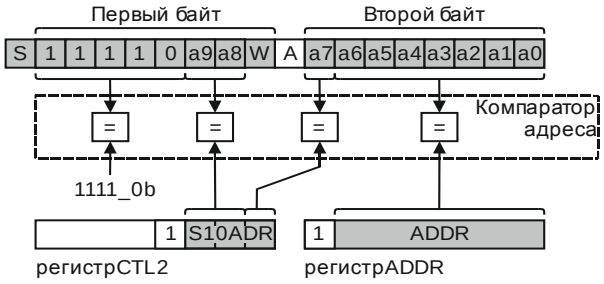


Рисунок 34.13 – Компаратор адреса в режиме 10-битной адресации

Сдвиговый регистр данных

Регистр SDA представляет собой сдвиговый регистр, используемый для приема и передачи данных. Старший бит регистра передается/принимается первым, младший бит – последним. Запись в регистр SDA возможна только, если установлен бит INT регистра ST. Регистр может быть прочитан в любой момент времени, но прочитанные данные будут гарантированно достоверными только при установленном бите INT. Регистр SDA не очищается при сбросе и хранит случайные данные до тех пор, пока не будет перезаписан программно или аппаратно после приема байта.

Генерация тактового сигнала и синхронизация

Последовательный тактовый сигнал (выходной сигнал модуля I2C в режиме мастера) формируется генератором на базе системного тактового сигнала (с частотой f_{PCLK}).

Модуль I2C может функционировать в двух глобальных режимах – стандартном/скоростном FS и высокоскоростном HS.

В режиме FS используется 15-битный предделитель. Значение младших 6 бит определяется значением битового поля SCLFRQ регистра CTL1, а нулевой бит всегда равен

нулю (деление производится только на четное число). Значение старших 8 бит задается полем SCLFRQ регистра CTL3. Блок синхронизации тактового сигнала производит синхронизацию генератора тактового сигнала и выходного сигнала SCL с тактовым сигналом других устройств, подключенных к шине.

В режиме HS используется 12-битный предделитель. Значение младших 4 бит определяется значением битового поля HSDIV регистра CTL2, нулевой бит при этом всегда равен нулю. Значение старших 8 бит задается полем HSDIV регистра CTL4.

Определяемое спецификацией протокола SMBus наименьшее время ожидания на линии SCL составляет 25 мс. Если пауза между двумя тактовыми импульсами превысила 25 мс, то устройство должно прервать текущую передачу. Мастер должен сформировать состояние старта в процессе передачи или после ее окончания. Ведомый должен освободить шину. Устройства, обнаружившие данное состояние, должны восстановить свои соединения и ожидать формирования состояния старта в пределах 10 мс.

Для отслеживания периодов ожиданий на шине в модуле I2C имеется счетчик времени ожидания. Функциональная схема счетчика показана на рисунке 34.14.

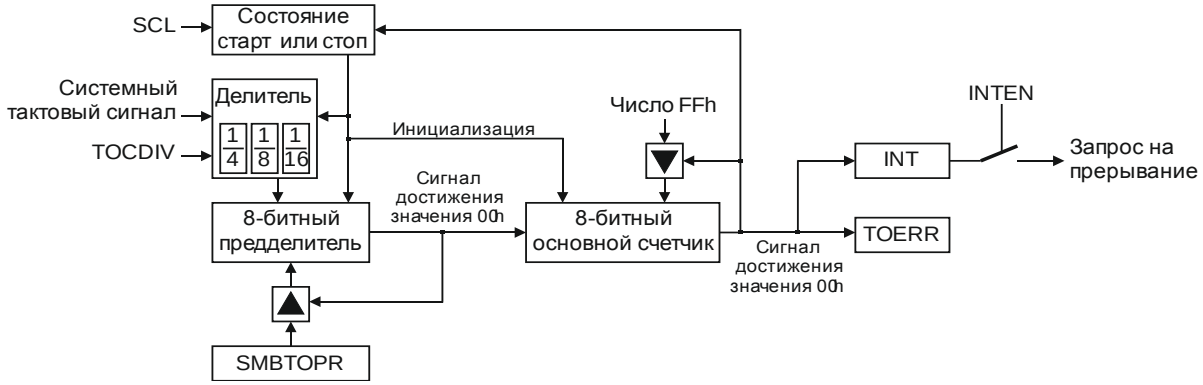


Рисунок 34.14 – Функциональная схема счетчика времени ожидания

Счетчик времени ожидания состоит из делителя, 8-битного программируемого предделителя и 8-битного основного счетчика. Все элементы счетчика времени ожидания начинают работу по отрицательному фронту сигнала на линии SCL (если работа счетчика разрешена). Положительный фронт сигнала на линии SCL сбрасывает значения делителя, предделителя и основного счетчика. Предделитель считает вниз, начиная со значения, записанного в регистр TOPR. После достижения нуля счетчиком предделителя, он загружается значением из регистра TOPR. Основной счетчик считает вниз от значения FFh. Каждое достижение нуля предделителем декрементирует значение основного счетчика. Обнуление основного счетчика и загрузка его значением FFh вызывает остановку основного счетчика предделителя и делителя и установку флага TOERR в регистре CST. Дополнительно устанавливается флаг INT, и если разрешено, генерируется прерывание.

Период времени ожидания определяется следующим выражением

$$T_{ожид} = T_{PCLK} \times TOCDIV \times (SMBTOPR + 1) \times 256, \qquad (34.1)$$

где T_{PCLK} – период системного тактового сигнала с частотой f_{PCLK}.

Арбитраж и обнаружение ошибок на шине I2C

Арбитраж в режиме мастера передатчика может быть потерян в случае, когда два мастера одновременно формируют состояние старта и начинают передачу данных. Потеря арбитража может происходить как во время передачи адреса, так и во время передачи данных.

В случае потери приоритета при передаче байта адреса, мастер переходит в режим ведомого приемника и начинает принимать адрес. Если принятый адрес оказался «своим»,

модуль I2C далее функционирует в режиме ведомого. Если принятый адрес не оказался «своим», то модуль I2C переходит в режим безадресного ведомого.

В случае потери приоритета при передаче байта данных модуль I2C сразу переходит в режим безадресного ведомого.

Обнаружение и исправление ошибок на шине I2C

Состояние ошибки на шине I2C возникает в том случае, если во время передачи адреса/данных или во время квитирования на шине I2C обнаруживаются состояния старта или стопа. При обнаружении ошибки на шине I2C выполняются действия:

- в поле MODE регистра ST записывается код ошибки 1Fh;
- генерируется прерывание (если разрешено);
- модуль I2C переходит в режим безадресного ведомого;
- линии SDA и SCL освобождаются.

Обнаружение ошибки на шине I2C может вызвать у простой шины некорректное формирование состояния старта и отключение модуля I2C. Поэтому для возврата к нормальной работе следует выполнить действия:

- выключить и снова включить модуль I2C (бит ENABLE в регистре CTL1);
- в течение времени простоя проверить, не подключен ли другой активный мастер к шине I2C (бит BB регистра CST должен быть обнулен);
- в режиме мастера сформировать состояние старта, передать адрес и затем сформировать состояние останова, таким образом, проведя синхронизацию всех ведомых устройств (в том числе и тех, которые не обнаружили ошибку на шине I2C).

Режим IDLE

Переход в режим IDLE происходит при отключении внешнего сигнала тактирования модуля I2C записью нуля в бит I2CEN регистра APB_CLK. Переход в режим IDLE подобен программному выключению модуля I2C (очистка бита ENABLE в регистре CTL1). Регистры CTL0, ST и CST очищаются, чтобы гарантировать нормальный старт после возобновления функционирования модуля.

Выход из режима IDLE осуществляется записью единицы в бит I2CEN и включением модуля I2C битом ENABLE.

34.3 Инициализация и функционирование

В целом модуль I2C поддерживает два базовых режима – режим FS и режим HS.

Стандартный/скоростной режим или режим FS – стандартный режим работы, в котором модуль функционирует по умолчанию. Диапазон частот сигнала на линии SCL – от 763 Гц до 6,25 МГц (при f_{PCLK} = 100 МГц).

Высокоскоростной режим или режим HS – режим работы, который включается программно. Режим HS значительно превосходит режим FS по скорости – диапазон частот сигнала на линии SCL от 8,14 кГц до 16,67 МГц (при f_{PCLK} = 100 МГц).

Все операции режима HS начинаются в режиме FS в следующем порядке:

- стартовое состояние;
- 8-битный код мастера (значение 0000_1xxxb, где «xxx» – уникальный код каждого мастера в системе нескольких устройств);
- неквитирование.

Арбитраж на шине I2C происходит в момент передачи несколькими мастерами своих уникальных кодов. Выигравший арбитраж мастер захватывает шину. В связи с такой организацией режима HS дальнейший арбитраж и синхронизация на шине не реализуются.

После выполнения вышеуказанных шагов устройства, поддерживающие режим HS, переключаются в этот режим. Мастер генерирует состояние повторного старта SR, а затем передает адрес ведомого и бит направления передачи R/W#, см. рисунок 34.15. Расшифровка обозначений, принятых на рисунке 34.15, приведена в таблице 34.1.

Все передачи в режиме HS по формату идентичны передачам режима FS, что делает эти два режима полностью совместимыми.



Рисунок 34.15 – Переход в режим HS и обратно в режим FS

Выход из режима HS происходит генерированием состояния окончания передачи P, после которого все устройства переключаются обратно в режим FS. В каждом из двух базовых режимов – FS и HS – модуль I2C может функционировать как мастер или ведомый, получать или передавать информацию.

Далее все режимы работы модуля I2C будут рассмотрены подробно.

Инициализация

Для начала работы следует произвести инициализацию:

1 Включить модуль I2C установкой бита ENABLE в регистре CTL1.

2 Если активен режим мастера, записать нужный коэффициент деления в битовое поле SCLFRQ (регистры CTL1, CTL3) для выбора периода тактового сигнала SCL (для режима HS записать коэффициент деления в поле HSDIV регистров CTL2, CTL4).

3 Если активен режим ведомого, необходимо:

- записать «собственный» адрес ведомого в битовое поле ADDR и установить бит SAEN регистра ADDR;
- для реализации 10-битной адресации записать старшие биты адреса в битовое поле S10AD и установить бит S10EN регистра CTL2;
- для включения функции распознавания адреса общего вызова установить бит GCMEN в регистре CTL0;
- для включения функции распознавания адреса отклика установить бит SMBARE в регистре CTL0.

4 При необходимости отслеживания периодов ожидания на шине I2C записать желаемые значения в регистр TOPR и в битовое поле TOCDIV (регистр CST) для отсчета времени ожидания на линии SCL. Для автоматического отслеживания времени ожидания записать ненулевое значение в битовое поле TOCDIV регистра CST.

5 Для разрешения формирования запроса на прерывание установить бит INTEN в регистре CTL0.

Функционирование

Модуль I2C может работать в режиме мастера или ведомого. Также он может функционировать как передатчик или приемник, т. е. модуль I2C поддерживает девять режимов:

- безадресный ведомый;
- мастер передатчик в режиме FS;
- мастер передатчик в режиме HS;
- мастер приемник в режиме FS;
- мастер приемник в режиме HS;
- ведомый передатчик в режиме FS;
- ведомый передатчик в режиме HS;
- ведомый приемник в режиме FS;
- ведомый приемник в режиме HS.

Передача информации по шине I2C состоит из последовательности различных действий (начало передачи, прием данных и др.). Каждое действие называется состоянием (состояние старта, состояние останова и др.). После того, как то или иное состояние сформировано, его код аппаратно записывается в регистр ST в битовое поле MODE и может

быть прочитано программно. В таблицах 34.2 и 34.3 приводятся все возможные состояния, их мнемонические обозначения и коды. На квитирование или неквитирование приема указывает запись «ACK» или «NACK» соответственно. Так, например, если мастер отправил байт адреса ведомому, который после получения квитировал прием, то на это будет указывать «ACK», а в поле MODE регистра ST будет записан код 04h, соответствующий состоянию с мнемоническим обозначением «MTADPA». Более подробно каждый режим работы модуля I2C будет рассмотрен далее. На рисунках 34.16 – 34.23, поясняющих работу модуля I2C в том или ином режиме, приняты обозначения, расшифровка которых приводится в таблице 34.1. Для получения дополнительной информации и понимания работы модуля I2C можно воспользоваться приложением Б.

Таблица 34.2 – Коды функционирования модуля I2C в режиме FS

Режим		Код	Мнемоника	Описание состояния на момент записи кода в поле MODE регистра ST	ACK/ NACK
Общий		00h	IDLE	IDLE, нет доступной валидной информации о статусе	–
Мастер в режиме FS	–	01h	STDONE	Сформировано состояние старта	–
		02h	RSDONE	Сформировано состояние повторного старта	–
		03h	IDLARL	Потеря арбитража, переход в режим безадресного ведомого	–
	Передача	04h	MTADPA	Отправлен адрес ведомого	ACK
		05h	MTADNA	Отправлен адрес ведомого	NACK
		06h	MTDAPA	Отправлен байт данных	ACK
		07h	MTDANA	Отправлен байт данных	NACK
	Прием	08h	MRADPA	Отправлен адрес ведомого	ACK
		09h	MRADNA	Отправлен адрес ведомого	NACK
		0Ah	MRDAPA	Принят байт данных	ACK
		0Bh	MRDANA	Принят байт данных	NACK
	–	0Ch	MTMCER	Отправлен код мастера, обнаружена ошибка	ACK
–		0Dh – 0Fh		Зарезервировано. Не использовать!	–
Ведомый в режиме FS	Прием	10h	SRADPA	Принят адрес	ACK
		11h	SRAAPA	Принят адрес после потери арбитража	ACK
		12h	SRDAPA	Принят байт данных	ACK
		13h	SRDANA	Принят байт данных	NACK
	Передача	14h	STADPA	Принят адрес	ACK
		15h	STAAPA	Принят адрес после потери арбитража	ACK
		16h	STDAPA	Отправлен байт данных	ACK
		17h	STDANA	Отправлен байт данных	NACK
	Передача адреса отклика	18h	SATADP	Принят адрес отклика на предупреждение	ACK
		19h	SATAAP	Принят адрес отклика на предупреждение после потери арбитража	ACK
		1Ah	SATDAP	Отправлены данные в ответ на получение адреса отклика	ACK
		1Bh	SATDAN	Отправлены данные в ответ на получение адреса отклика	NACK

Окончание таблицы 34.2

Режим		Код	Мнемоника	Описание состояния на момент записи кода в поле MODE регистра ST	ACK/ NACK
Ведомый в режиме FS	–	1Ch	SSTOP	Обнаружено состояние останова ведомого	–
		1Dh	SGADPA	Принят адрес общего вызова	ACK
		1Eh	SDAAPA	Принят адрес общего вызова после потери арбитража	ACK
Общий		1Fh	BERROR	Обнаружена ошибка на шине (некорректное состояние старта или останова)	–
Примечание – Диапазон значений кодов 0Dh–0Fh зарезервирован и не доступен для использования. Дополнительная информация находится в приложении Б данного ТО.					

Таблица 34.3 – Коды функционирования модуля I2C в режиме HS

Режим		Код	Мнемоника	Описание состояния на момент записи кода в поле MODE регистра ST	ACK/ NACK
Мастер в режиме HS	–	21h	HMTMCOK	Код мастера передан успешно, переход в режим HS	–
		22h	HRSDONE	Сформировано состояние повторного старта	–
		23h	HIDLARL	Потеря арбитража, переход в режим HS безадресного ведомого	–
	Передача	24h	HMTADPA	Отправлен адрес ведомого	ACK
		25h	HMTADNA	Отправлен адрес ведомого	NACK
		26h	HMTDAPA	Отправлен байт данных	ACK
		27h	HMTDANA	Отправлен байт данных	NACK
	Прием	28h	HMRADPA	Отправлен адрес ведомого	ACK
		29h	HMRADNA	Отправлен адрес ведомого	NACK
		2Ah	HMRDAPA	Принят байт данных	ACK
		2Bh	HMRDANA	Принят байт данных	NACK
Ведомый в режиме HS	Прием	30h	HSRADPA	Принят адрес	ACK
		32h	HSRDAPA	Принят байт данных	ACK
		33h	HSRDANA	Принят байт данных	NACK
	Передача	34h	HSTADPA	Принят адрес	ACK
		36h	HSTDAPA	Отправлен байт данных	ACK
		37h	HSTDANA	Отправлен байт данных	NACK
Примечание – Диапазоны значений кодов 2Ch–2Fh и 38h–3Fh, а также коды 20h, 31h, 35h зарезервированы и недоступны для использования. Дополнительная информация находится в приложении Б данного ТО.					

Режим безадресного ведомого

Режим работы по умолчанию (MODE = 00h). После включения модуль I2C начинает функционировать в режиме безадресного ведомого и непрерывно мониторит шину. При обнаружении состояния старта или повторного старта переходит в режим ведомого приемника. Для перехода в режим мастера передатчика нужно сформировать корректное состояние старта.

Переключение в режим безадресного ведомого происходит в случаях:

- стартовое состояние не было успешно сформировано, так как другое устройство удерживало на линии SCL низкий уровень сигнала;

- произошла потеря арбитража во время передачи байта данных в режиме мастера передатчика или во время передачи бита R/W# в режиме мастера приемника;
- произошла потеря арбитража во время ответа на полученный адрес отклика;
- неквитирование принятого адреса в режиме ведомого приемника (адрес не совпал со «своим» или запрещен);
- неквитирование в конце переданного байта в режиме ведомого передатчика;
- обнаружено состояние останова;
- обнаружена ошибка на шине;
- модуль I2C был сброшен;
- модуль I2C был выключен.

Режим FS мастера передатчика

Включение режима FS мастера передатчика:

1 Переход в режим мастера передатчика происходит после успешного формирования состояния старта. Первый байт, передаваемый мастером сразу после старта, состоит из адреса ведомого и бита направления.

2 В зависимости от состояния бита направления (R/W#), модуль I2C далее функционирует как мастер передатчика (если R/W# = «0») или как мастер приемника (если R/W# = «1»). Для перехода в режим HS мастер может передать код мастера (0000_1xxxb) вместо первого байта адреса.

3 Переход в режим мастера произойдет после установки бита START в регистре CTL0. Если бит BB в регистре CST сброшен, т. е. шина свободна, будет сгенерировано состояние старта. Если бит BB = 1b, то бит START останется установленным, а состояние старта будет сгенерировано по истечении времени, равного одному такту сигнала тактирования на линии SCL, после освобождения шины.

4 Как только стартовое состояние будет сгенерировано успешно, бит START сбросится, модуль I2C перейдет в состояние STDONE (в поле MODE запишется значение 01h), установится флаг INT, и линия SCL будет удерживаться в «0» до тех пор, пока флаг INT не будет сброшен. Если разрешено битом INTEN (регистр CTL0), сгенерируется прерывание.

Передача адреса и данных:

1 Пока удерживается флаг INT, программа записывает адрес ведомого и бит направления передачи в регистр данных SDA (адрес записывается в биты с седьмого по первый).

2 После записи в регистр SDA флаг INT сбрасывается программно установкой бита CLRST в регистре CTL0.

3 После сброса флага INT и по истечении времени, требуемого для установки данных, на линии SCL появляется тактовый сигнал и данные, хранящиеся в регистре SDA, начинают передаваться по линии SDA.

4 После завершения передачи байта и получения ответа на запрос подтверждения передачи (ACK), т. е. после девятого такта сигнала тактирования на линии SCL, аппаратная часть анализирует квитирование/неквитирование передачи и устанавливает соответствующий код в поле MODE.

5 Во время передачи линии SCL и SDA постоянно мониторятся с целью выявления возможных конфликтов с другими устройствами, подключенными к шине. В случае обнаружения конфликта передача прерывается и в поле MODE записывается код 11b (состояние SRAAPA – переход в режим ведомого приемника после потери арбитража) или код 03h (состояние IDLARL – переход в режим безадресного ведомого после потери арбитража).

6 Если бит направления равен единице и не обнаружено ошибок на шине, модуль I2C переходит в режим мастера приемника.

7 Если бит направления равен нулю и передача адреса ведомого завершена успешно (значение кода в поле MODE не равно 05h/1Fh), устанавливается флаг INT, указывая на то, что ожидается запись первого байта данных в регистр SDA для дальнейшей передачи, и, если разрешено, генерируется прерывание. Пока флаг INT будет оставаться установленным, линия SCL будет удерживаться в «0».

8 Байт данных записывается программно в регистр SDA и передача продолжается.

9 Если ведомый приемник не квитирует отправленный ему байт данных, в поле MODE записывается код 0Bh (состояние MRDANA). На линии SCL будет установлен низкий уровень сигнала и, если разрешено, сгенерировано прерывание.

Для отслеживания ошибок в пакетах данных применяется механизм вычисления контрольной суммы (CRC) для нескольких байт данных. В режиме мастера передатчика установка бита PECNEXT в регистре CST вызовет перенос содержимого регистра ошибок (не доступен программно) в регистр SDA и инициирует передачу байта CRC (байт контрольной суммы) ведомому. Передача байта CRC должна выполняться после передачи последнего байта данных и перед формированием состояния останова или повторного старта.

Мастер передатчика контролирует шину и может адресовать любое ведомое устройство и изменять направление передачи без потери контроля над шиной, используя возможность формирования состояния повторного старта. Для формирования состояния следует:

1 Установить бит START.

2 В режиме мастера приемника прочитать последний полученный байт из регистра SDA.

3 Сбросить флаг прерывания INT.

После этих действий будет освобождена линия SCL, сгенерировано состояние повторного старта и сгенерировано прерывание. В поле MODE будет записан код 02h (состояние RSDONE).

Модуль I2C может быть выведен из режима мастера передатчика генерированием состояния останова. Для этого необходимо:

1 Установить бит STOP в регистре CTL0.

2 В режиме мастера приемника прочитать последний полученный байт из регистра SDA.

3 Сбросить флаг INT.

Вышеуказанные действия приведут к незамедлительному формированию состояния останова и очистке бита STOP.

Состояние останова может быть сформировано только если модуль I2C функционирует как мастер и контролирует шину (в поле MODE находится любое значение кода из диапазона 01h – 0Bh).

Дополнительно можно обратиться к приложению Б данного технического описания.

На рисунке 34.16 представлено графическое пояснение к описанию режима.

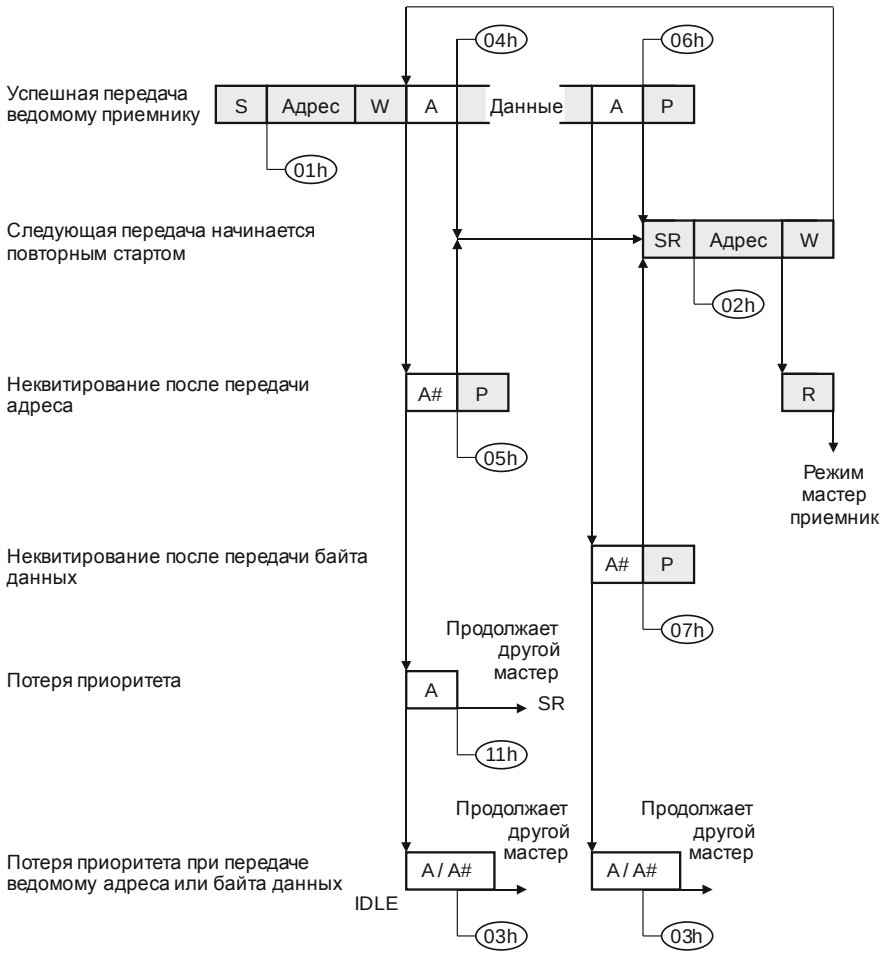


Рисунок 34.16 – Режим FS мастера передатчика

Режим HS мастера передатчика

Переход в режим HS мастера передатчика происходит в том случае, если после состояния старта мастер передает код мастера (0000_1xxxb) вместо адреса ведомого. По окончании передачи кода мастера устанавливается флаг INT и, если разрешено, генерируется прерывание. Вслед за успешной передачей кода мастера в поле MODE записывается код 21h (состояние HMTMCOK), и мастер переходит в режим HS.

Далее необходимо сформировать состояние повторного старта, записав единицу в бит START, и сбросить флаг INT, записью единицы в бит CLRST.

После сгенерированного состояния повторного старта устанавливается флаг INT, и в поле MODE записывается код 22h (состояние HRSDONE). Дальнейший порядок действий по передаче адреса и данных аналогичен описанному режиму FS мастера передатчика.

Дополнительно можно обратиться к приложению Б.

На рисунке 34.17 представлено графическое пояснение к описанию режима.

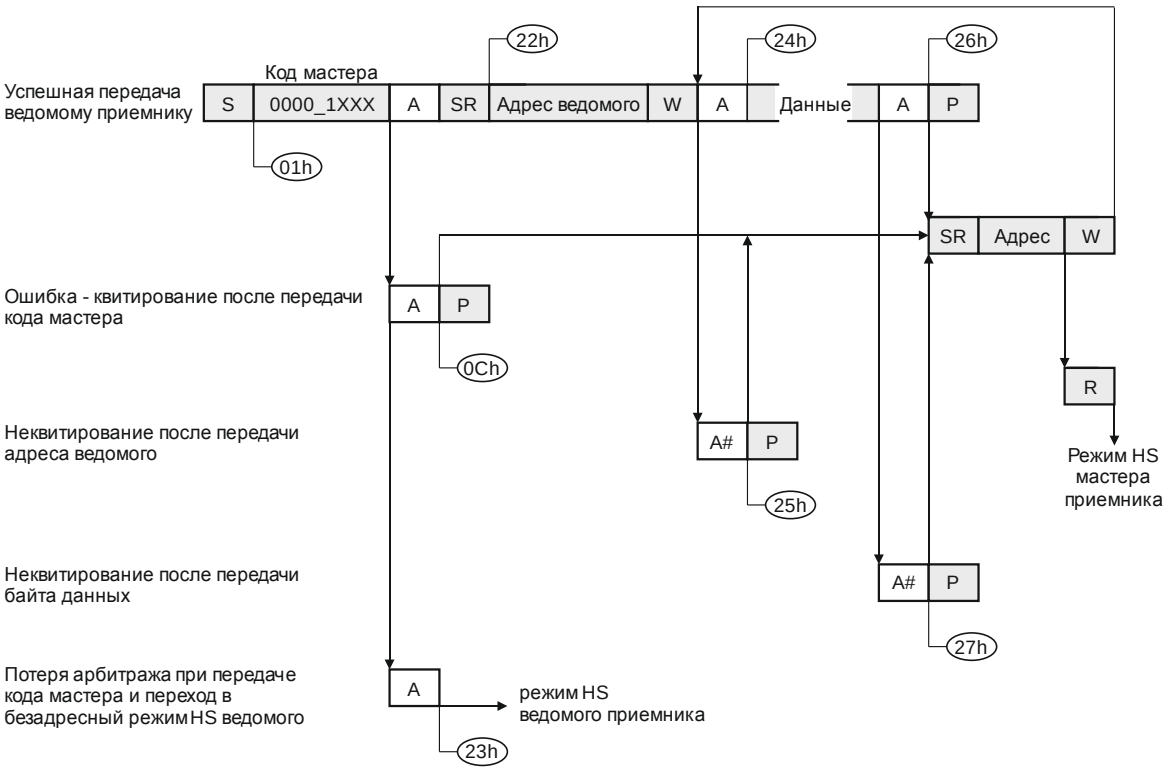


Рисунок 34.17 – Режим HS мастера передатчика

Режим FS мастера приемника

Переход в режим мастера приемника происходит после успешной передачи адреса ведомого с единичным битом направления (R/W# = «1»). В режиме мастера приемника модуль I2C получает данные от ведомого устройства, поэтому теряет контроль над шиной SDA. В тоже время мастер продолжает тактировать передачу и должен отвечать на бит ACK каждого принятого байта.

После каждого принятого байта устанавливается флаг INT, и пользовательская программа читает полученные данные из регистра SDA. Линия SCL удерживается в «0», пока установлен флаг INT. После сброса флага INT может стартовать прием следующего байта. После этого (согласно протоколу SMBus) состояния повторного старта или стопа не должны генерироваться мастером, поскольку мастер теперь не является единственным контролером линии SDA. В конце приема каждого байта мастер не квитирует прием, сообщая, таким образом, ведомому об успешном приеме.

После приема предпоследнего байта перед сбросом флага INT следует записать ноль в бит ACK регистра CTL0. В тоже время, если требуется отправка байта CRC, следует установить бит PECNEXT в регистре CST. После сброса флага INT будет принят последний байт данных и не квитирован. По окончании приема мастер возвращается в режим передатчика и теперь может сгенерировать состояние повторного старта или останова.

Если механизм отслеживания ошибок включен, то последний переданный от ведомого байт будет байтом CRC. В случае если результат вычисления контрольной суммы не нулевой, то установится флаг ошибки PECFAULT в регистре CST.

Дополнительно можно обратиться к приложению Б.

На рисунке 34.18 представлено графическое пояснение к описанию режима.

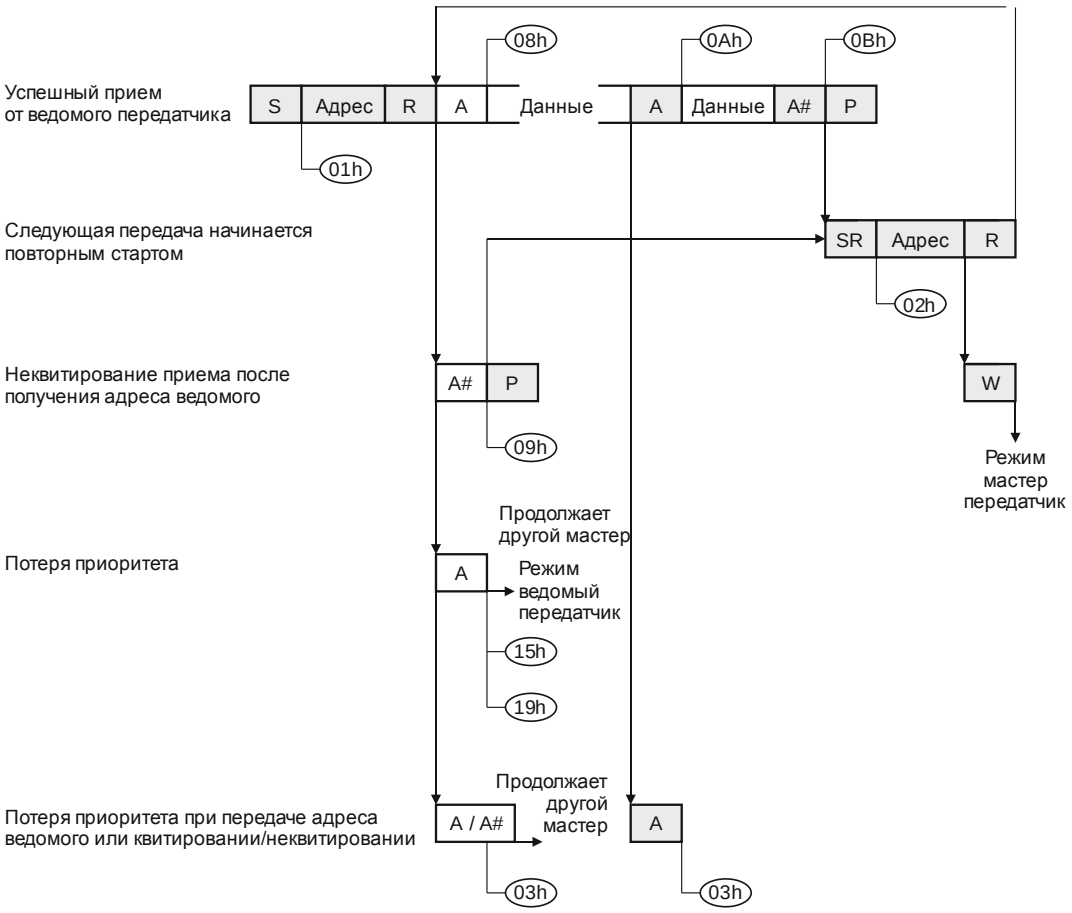


Рисунок 34.18 – Режим FS мастера приемника

Режим HS мастера приемника

Переход в режим HS мастера приемника происходит, если после переданного кода мастера и последовавшего за ним состояния повторного старта, производится передача адреса ведомого с битом направления R/W# = «1». Модуль I2C переходит в режим HS мастера приемника, устанавливается флаг INT, а в поле MODE записывается соответствующий код из диапазона 28h – 2Bh.

Дополнительно можно обратиться к приложению Б.

На рисунке 34.19 представлено графическое пояснение к описанию режима.

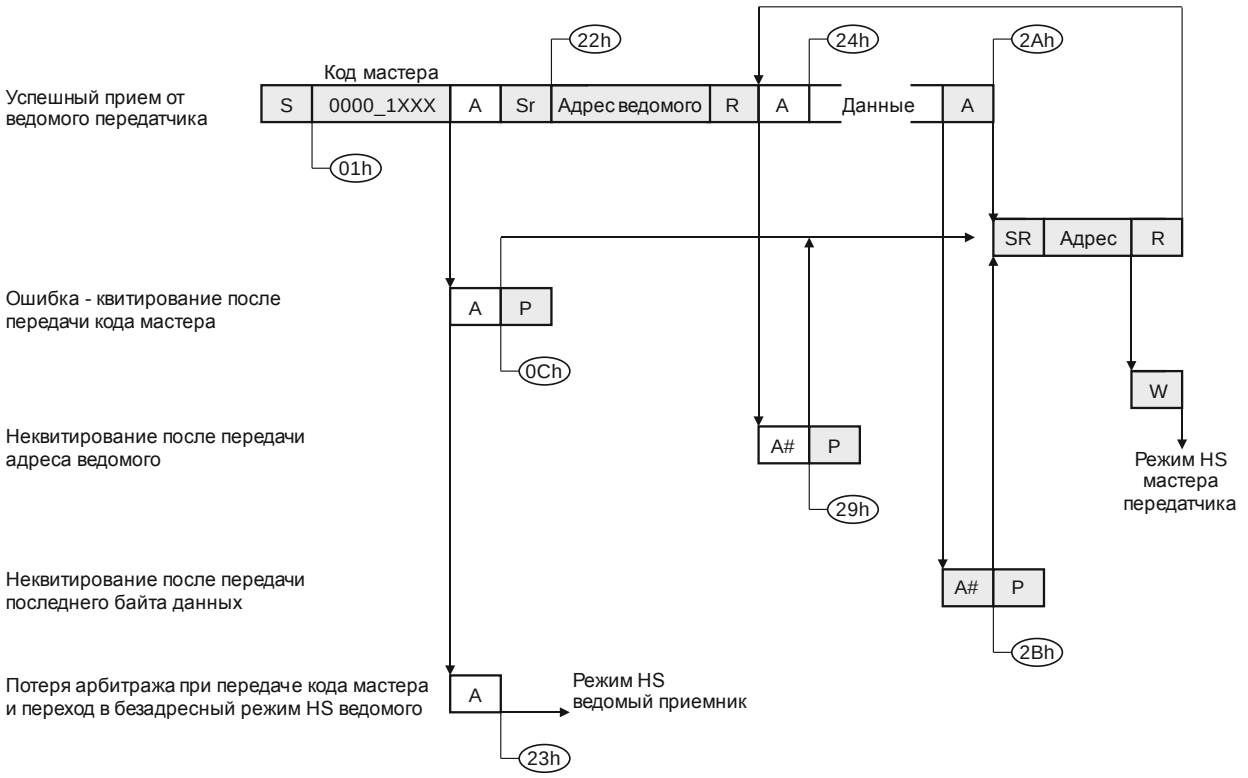


Рисунок 34.19 – Режим HS мастера приемника

Режим FS ведомого приемника

В этом режиме данные принимаются от мастера передатчика. Ведомый квитирует или не квитирует прием каждого байта.

После включения модуль I2C мониторит шину. При обнаружении состояния старта, модуль I2C переключается в режим ведомого приемника и начинает принимать семь бит адреса и бит направления передачи от мастера. Мастер передатчика может переключиться в режим ведомого приемника вследствие потери арбитража при передаче адреса.

После получения байта адреса ведомый сравнивает полученный адрес:

- полем ADDR регистра ADDR, если установлен бит SAEN;
- со значением 0000_000b (адрес общего вызова), если установлен бит GCMEN;
- со значением 0001_100b (адрес отклика), если установлен бит SMBARE.

Квитирование приема производится, если принятый адрес совпал с «собственным» (запрограммированным пользователем) адресом общего вызова или адресом отклика. После обнаружения совпадения адреса и квитирования в поле MODE записывается соответствующий код и устанавливается флаг INT. Также, если разрешено битом INTEN, генерируется прерывание. Принятый байт (адрес и бит направления) переписывается в регистр SDA.

В зависимости от состояния бита направления, модуль I2C переходит в режим ведомого передатчика (если R/W# = «1») или остается в режиме ведомого приемника (R/W# = «0»).

После каждого принятого байта устанавливается флаг INT, указывающий на то, что необходимо прочитать данные из регистра SDA, а линия SCL удерживается в «0». После программного чтения регистра SDA флаг INT сбрасывается (записью единицы в бит CLRST), и линия SCL освобождается.

Установка битов SAEN и S10EN включает режим 10-битной адресации ведомого приемника. После обнаружения состояния старта ведомый последовательно принимает два байта, в которых содержится адрес.

Последовательность передачи бит в посылке при 10-битной адресации была рассмотрена ранее в подразделе 34.1 настоящего ТО.

Механизм распознавания адреса изложен в подразделе 0 настоящего ТО и показан на рисунке 34.13.

После корректного приема ведомым двух байтов и совпадения принятого адреса с собственным, байты сохраняются в регистре SDA и сдвиговом регистре, прием квитируется, устанавливается флаг INT, а в поле MODE записывается соответствующий код состояния – 10h или 17h.

Если включен механизм обнаружения ошибок, последний байт, принятый от мастера передатчика, будет байтом CRC. Если результат вычисления контрольной суммы не нулевой, устанавливается флаг ошибки PECFAULT и передача не квитируется. Программа пользователя должна «знать» о количестве передаваемых мастером байт и устанавливать бит PECNEXT перед чтением предпоследнего байта из регистра SDA и потом сбрасывать флаг INT. В результате будет аппаратно рассчитана контрольная сумма, и результат отправлен мастеру в момент передачи бита ACK. Если ошибок нет, будет выполнено квитирование (отправлен «0» в ответ на запрос ACK), если ошибки есть – неквитирование (отправлена «1» в ответ на запрос ACK).

Если ведомому приемнику нужно сообщить мастеру, что он не может более принимать данные, следует сначала установить бит ACK, а затем – бит CLRST (для сброса флага INT). Далее будет принят последний байт данных, который не будет квитирован (бит ACK = 1b), и установится флаг INT. После этого программа может прочитать последний полученный байт из регистра SDA и сбросить флаг INT, после чего модуль I2C освободит шину.

Дополнительно можно обратиться к приложению Б.

На рисунке 34.20 представлено графическое пояснение к описанию режима.

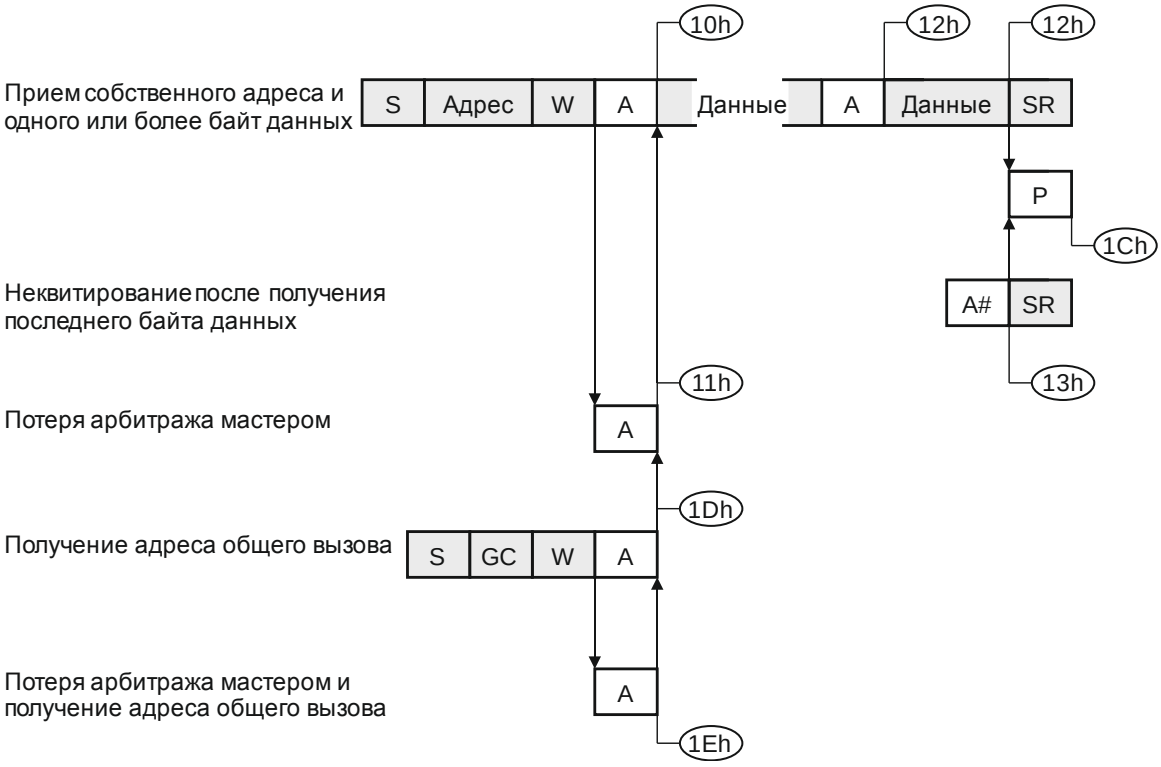


Рисунок 34.20 – Режим FS ведомого приемника

Режим HS ведомого приемника

Включение режима происходит после получения валидного кода мастера (0000_1xxxb). После передачи кода мастера формируется состояние повторного старта, а затем передается адрес ведомого с нулевым битом направления (R/W# = «0»). После получения байта адреса ведомый проверяет его на совпадение (см. ранее «Режим FS ведомого приемника»).

Дополнительно можно обратиться к приложению Б.

На рисунке 34.21 представлено графическое пояснение к описанию режима.



Рисунок 34.21 – Режим HS ведомого приемника

Режим FS ведомого передатчика

В этом режиме данные передаются от ведомого передатчика к мастеру приемника. Ведомый проверяет ответ мастера на бит ACK.

Переход в режим передатчика происходит из режима ведомого приемника. После получения собственного адреса и бита направления, равного единице (R/W# = «1»), ведомый становится передатчиком. Флаг INT устанавливается, указывая на то, что в регистр SDA следует записать данные.

Пока установлен флаг INT, линия SCL удерживается в «0». После записи данных в регистр SDA следует сбросить флаг INT. После этого, по истечении времени, необходимого для установки данных на линии SDA, линия SCL освобождается, и данные начинают передаваться.

Передача данных аналогична передаче в режиме мастера передатчика. После каждого успешного приема байта устанавливается флаг INT, а в поле MODE записывается соответствующий код. Линия SCL удерживается в состоянии «0» до тех пор, пока флаг INT остается установленным. Флаг INT должен сбрасываться только после записи данных в регистр SDA. Каждый последующий байт должен записываться в регистр SDA до тех пор, пока в поле MODE не появится код 17h (состояние STDANA), указывающий на то, что мастер «не желает» далее принимать данные.

Вывод ведомого из режима передатчика осуществляется только мастером приемника. Мастер приемника должен не квитировать последний (согласно запланированному количеству) полученный байт данных. При обнаружении неквитирования переданных данных, модуль I2C переходит в режим безадресного ведомого, и в поле MODE записывается код 00h (состояние IDLE). Далее ведомый мониторит шину в ожидании состояния старта или повторного старта.

Для работы в режиме с 10-битной адресацией следует осуществить действия аналогичные описанным для режима FS ведомого приемника.

Сначала модуль I2C переходит в режим ведомого приемника и получает 10-битный адрес. Если программно не требуется никаких действий, то флаг INT не устанавливается, линия SCL не удерживается в «0», и поле MODE содержит соответствующую информацию о состоянии. Далее (см. ранее «Формат передачи данных с 10-битной адресацией»), вслед за вторым байтом адреса, может последовать состояние повторного старта, и затем повторная передача первого байта адреса, с той лишь разницей, что бит направления содержит единицу (R/W# = «1»). Таким образом, после приема трех байт, если принятый 10-битный адрес окажется «своим», установится флаг INT, и ведомый переключится в режим передатчика. В поле MODE запишется один из двух кодов – 14h или 15h.

Если включен механизм распознавания ошибок, то последний отправленный ведомым передатчиком байт будет байтом CRC. Программа должна «знать» количество байт, посылаемых в пакете данных, и после отправки всех байт устанавливать бит PECNEXT (вместо записи очередных данных в регистр SDA) для того, чтобы в регистр SDA записался байт контрольной суммы.

В модуле I2C поддерживается функция распознавания адреса отклика, который передается мастером шины ко всем ведомым. Ведомое устройство, получившее адрес отклика (0001_100b), переключается в режим передатчика и начинает передавать свой собственный адрес (подробнее – см. «Формат передачи данных с 7-битной адресацией»).

Для включения функции распознавания адреса отклика следует установить бит SMBARE в регистре CTL0.

Модуль I2C реагирует на адрес отклика только при работе в режиме ведомого. В ответ на получение адреса отклика начать передачу адресов могут несколько ведомых. Ведомый, выигравший арбитраж, продолжает передачу, остальные – освобождают шину.

Дополнительно можно обратиться к приложению Б.

На рисунке 34.22 представлено графическое пояснение к описанию режима.

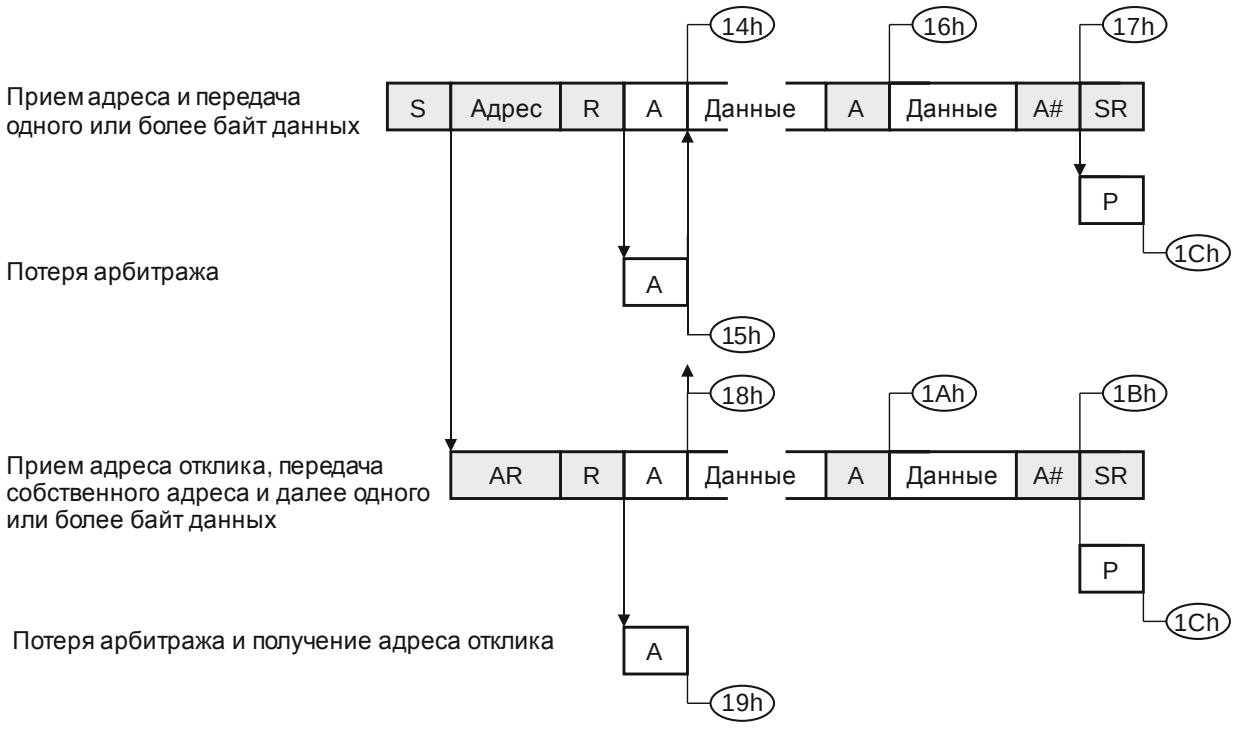


Рисунок 34.22 – Режим FS ведомого передатчика

Режим HS ведомого передатчика

Модуль I2C переходит в режим HS ведомого после получения валидного кода мастера (0000_1xxxb). Далее следует состояние повторного старта и передача адреса ведомого с единичным битом направления (R/W# = «1»). После этого ведомый переключается в режим HS ведомого передатчика. Функционирование в этом режиме в целом идентично режиму FS ведомого передатчика, с теми отличиями, что поддерживается более высокая скорость передачи, а значения кодов состояний (поле MODE) находятся в диапазоне 34h – 37h.

Дополнительно можно обратиться к приложению Б.

На рисунке 34.23 представлено графическое пояснение к описанию режима.

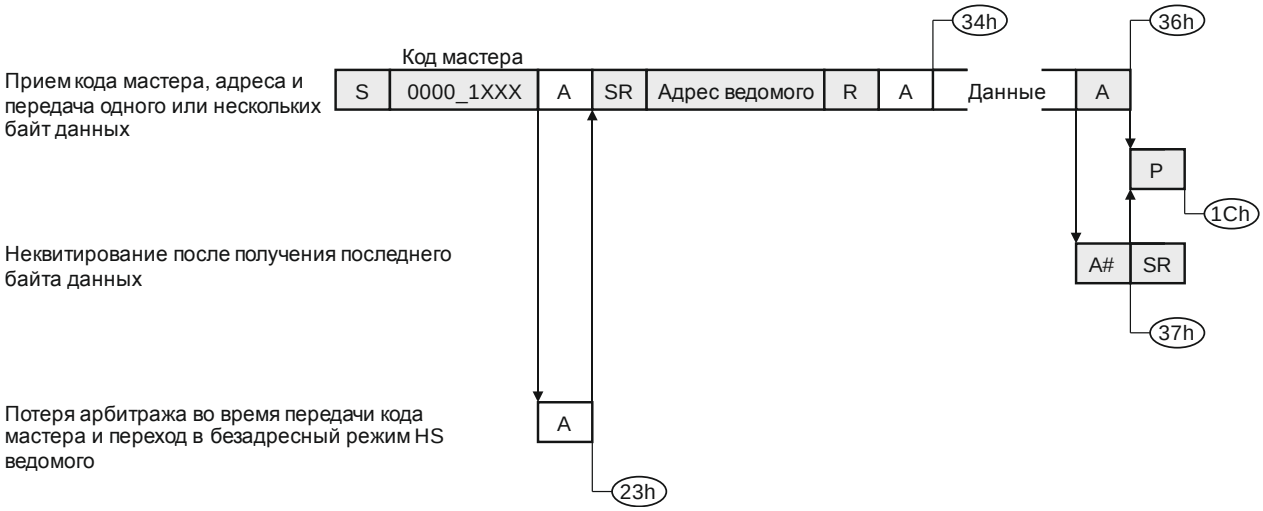


Рисунок 34.23 – Режим HS ведомого передатчика

Режим множественного адреса

В данном контроллере предусмотрена возможность использовать более одного адреса, на которые будет отвечать устройство. Возможно занести до 8 дополнительных адресов в регистры EADDR[i]. Для активации каждого из них необходимо установить бит SAEN в этом же регистре. Также есть возможность отвечать вообще на все приходящие адреса, для этого необходимо установить бит UNADDR в регистре CTL0.

Дополнительная информация о работе модуля

1 Когда модуль I2C выключен, бит BB регистра CST очищен. Включение модуля в системе, с более чем одним мастером, может произойти в момент времени, когда по шине идет передача. Бит BB не сможет это показать. Во избежание создания ошибок на шине модуль I2C должен синхронизироваться с сигналами на шине прежде, чем сделать попытку стать мастером. Для этого следует дождаться момента, когда на шине не будет выявлена активность, т. е. периодически проверять бит BB через периоды времени, равные периоду ожидания на шине.

2 Бит BB позволяет мониторить шину и не допускать формирования ошибочных состояний старта в процессе передачи между другими устройствами на шине.

3 В некоторых случаях шина может «зависать» при активных (с нулевым уровнем) сигналах на линиях SDA и/или SCL. Источниками таких состояний могут быть необнаруженные ошибочные стартовые или стоповые состояния, сформировавшиеся в течение приема ведомых данных. Если считать, что причиной зависания явился модуль I2C, то возможны следующие два варианта развития событий:

а) если зависла линия SCL, ничего не будет происходить, а мастер, захвативший шину, должен освободить ее;

б) если зависла линия SDA, мастер должен освободить шину. Следует помнить, что в нормальном состоянии удерживать линию SCL может только текущий мастер шины.

Последовательность действий для выхода из зависания следующая (при условии, что на шине только один мастер):

- выключить и включить модуль I2C для перевода его в режим безадресного ведомого;
- установить бит START для создания состояния старта;
- проверить, удерживается ли линия SDA в «0» (активное состояние) чтением бита TSDA регистра CST. Если линия активна, отправить одиночный импульс по линии SCL, установив бит TGSCL в регистре CST;
- проверить, что в поле MODE записан код 01b (состояние STDONE), который укажет на то, что состояние старта сформировано. Если нет, то повторять предыдущий и этот шаги до тех пор, пока линия SDA не освободится.

35 Контроллер интерфейса USB 2.0 DC

35.1 Общее описание

Контроллер Universal Serial Bus Device USB 2.0 (далее – DC контроллер или DC) обеспечивает интерфейс между микроконтроллером и USB шиной. DC контроллер должен соединяться с USB шиной через PHY, обеспечивающий физический уровень доступа к шине, см. рисунок 35.1, который совместим с интерфейсом UTMI+. DC контроллер поддерживает два режима работы: FULL SPEED (FS) и HIGH SPEED (HS). Существует возможность использования как внутреннего PHY (без поддержки режима HS), так и внешней микросхемы PHY, подключаемой к выводам GPIO с использованием альтернативных функций. Выбор внутреннего или внешнего PHY определяется битом EXTPHY регистра PHYCFG0.

Управляющие регистры контроллера USB DC приведены в приложении А.28.

Конечные точки (EPx, где x=0, 1, 2, 3) программируются через локальную шину процессора. Каждая из трех входных (IN EPx) и трех выходных (OUT EPx) конечных точек может отдельно конфигурироваться для любого из четырех типов USB обменов. USB данные записываются/читаются во внутренние буферы (размеры буферов IN и OUT по 8 КБайт). Далее, используя управляющую информацию из дескрипторов, механизм прямого доступа к памяти (далее – DMA) переносит данные в ОЗУ.

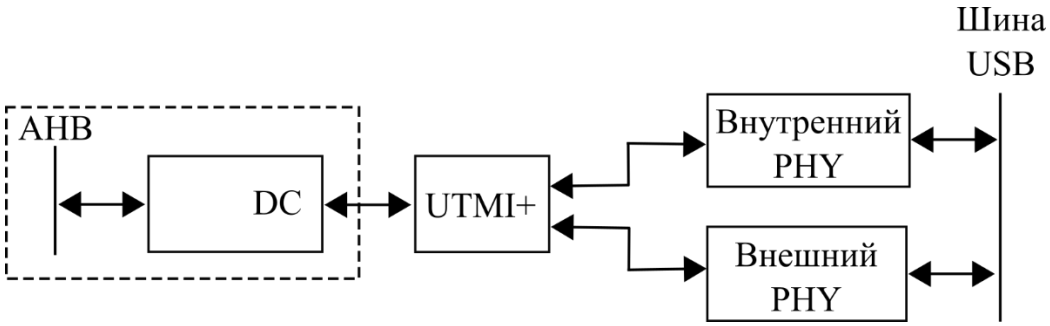


Рисунок 35.1 – Подключение DC контроллера к PHY

Примечания
1 AHB – локальная шина;
2 DC – Device контроллер;
3 UTMI+ – интерфейс связи с PHY;
4 USB – универсальная последовательная шина.

35.1 Операции DC контроллера

Описание системы

На рисунке 35.2 приведена внутренняя структура DC контроллера. Далее коротко описаны функции различных блоков DC контроллера.

DC контроллер состоит из следующих блоков:

- SNE – блок детектирования подключения к USB шине;
- SIE – блок последовательного интерфейса;
- AIE – блок связи с процессорной шиной;
- блок DMA.

Блок детектирования подключения к USB шине (далее – Speed Negotiation Engine или SNE) обнаруживает подключение путем опроса сигнала VBUS на разъеме USB.

Когда напряжение достигнет 3,3 В, значение сигнала VBUS будет равно «1». Блок SNE ждет RESET от HOST (устройство, управляющее USB шиной) и начинается обмен. Когда процедура сброса и первоначального обмена заканчивается, то выбирается скорость обмена FS (12 Мбит/с) или HS (480 Мбит/с). Об этом сообщается блоку последовательного

интерфейса (далее – Serial Interface Engine или SIE), который может начать операции приема/передачи. Блок SNE также детектирует и управляет состояниями приостановки (далее – SUSPEND) и восстановления (RESUME) работы DC контроллера.

Блок SIE включается, когда SNE оповещает о завершении процедуры сброса. SIE ждет прихода пакетов и обрабатывает их в соответствии со спецификацией USB 2.0. Данные записываются во внутренний буфер, принадлежащий конкретной конечной точке EPx.

Блок AIE (AHB Interface Engine) отвечает за передачу данных USB из внутренних буферов конечной точки на шину AHB с использованием DMA в соответствии с дескрипторами.

Блок DMA, используя дескрипторы, передает данные из внутреннего буфера конечной точки в общую память, которая также доступна ядру микроконтроллера.

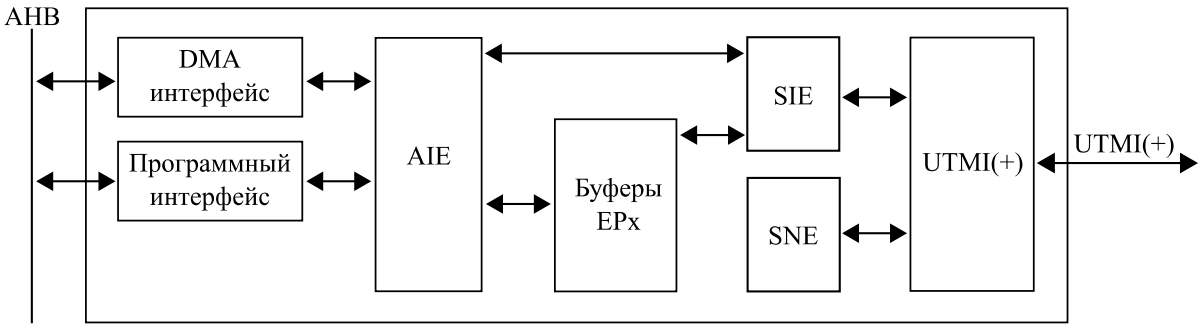


Рисунок 35.2 – Блок-схема внутренней структуры DC контроллера

Примечание – Принятые условные обозначения:
- SNE – блок, реализующий протокол USB 2.0;
- SIE – блок последовательного канала USB 2.0;
- UTMI+ – интерфейс с микросхемой физического уровня;
- EPx – конечная точка с номером x = 0, 1, 2, 3;
- AIE – блок, реализующий протокол процессорной шины AHB.

Интерфейс связи с микросхемой физического уровня

DC контроллер соединен с микросхемой физического уровня (далее – PHY) через интерфейс UTMI+. В таблице 35.1 приведено функциональное описание сигналов UTMI+ интерфейса.

Таблица 35.1 – Функциональное описание сигналов UTMI+ интерфейса

Обозначение сигнала	Описание	Тип
1	2	3
SUSPENDM	Установка трансивера в режим ожидания	O
XCVRSELECT	1: Выбор режима трансивера Full speed 0: Выбор режима трансивера High speed	O
TERMSELECT	1: Разрешение окончания режима Full speed 0: Разрешение окончания режима High speed	O
LINESTATE[1:0]	Сигнал состояния линии	I
OPMODE[1:0]	Сигнал выбора режима работы	O
DATAOUT[7:0]	Выходные данные	O
TXVALID	Передаваемые данные правильные	O
TXREADY	Готовность передачи	I
DATAIN[7:0]	Входные данные	I
RXVALID	Принимаемые данные правильные	I
RXACTIVE	Готовность приема данных	I

Окончание таблицы 35.1

1	2	3
RXERROR	Ошибка приема	I
CLK	Тактирующий сигнал от трансивера	I
RESET	Сигнал сброса трансивера	O
VBUSVALID	Сигнал готовности USB шины	I
CHRGVBUS	Сигнал управления включением заряда VBUS	O
DISCHRGVBUS	Сигнал управления включением разряда VBUS	O
DPPULLDOWN	Сигнал включения подтяжки pull-down для линии DP	O
DMPULLDOWN	Сигнал включения подтяжки pull-down для линии DM	O
IDPULLUP	Сигнал включения подтяжки pull-up линии ID	O
HOSTDISCONNECT	Сигнал отключения хоста (окончания сессии)	I
Примечание – Принятые условные обозначения в графе «Тип»: I – вход, O – выход.		

Описание блока SNE

Блок SNE детектирует подключение, управляет сбросом, управляет формированием пакета подтверждения HS HANDSHAKE и управляет операциями SUSPEND/RESUME. Он также поддерживает различные тестовые режимы, поддерживаемые всеми USB устройствами.

Состояние подключения обнаруживается, когда сигнал VBUS становится активным (высокий уровень). После этого DC контроллер ждет сброса от хоста и запускает HS HANDSHAKE, определяющий какой из режимов работы HS или FS будет установлен. DC контроллер не будет поддерживать трафик по сети и не будет подтверждать пакеты, направленные к нему, пока не выполнится процедура сброса.

DC контроллер поддерживает мягкое подключение/отключение. Это означает, что подтяжка PULL UP резистором на линии D+ может управляться из доступного пользователю регистра DC контроллера. Подтяжка PULL UP отключена после сброса, и поэтому данная функция полностью контролирует, в какой момент DC контроллер станет виден хосту.

Блок SNE также непрерывно отслеживает состояние приостановки SUSPEND (3 мс бездействия на USB шине), когда будет установлено состояние SUSPEND. Выход из состояния SUSPEND происходит либо с помощью сброса USB, либо по сигналу возобновления RESUME.

Сигнал возобновления RESUME может поступать либо от нижестоящего порта (HUB узел или HOST контроллер) или от самого устройства (REMOTE WAKEUP). Возможность DC контроллером генерировать сигнал REMOTE WAKEUP задается с помощью регистра GCTRL (бит RW), доступного пользователю. Если данная возможность используется, управляющая функция должна отобразить это в возвращаемом дескрипторе по запросу GETSTATUS к DC контроллеру.

Транзакции управляются SIE только в том случае, если SNE находится в режиме FS или HS. Все транзакции будут сброшены, когда DC контроллер находится в состояниях: SUSPEND, не подключено, подключено или во время процесса сброса (RESET).

Текущие состояния SNE, такие как: разрешение VBUS, активный SUSPEND, принятый USB reset и текущий режим скорости, могут быть определены через регистр состояния GSTAT DC контроллера. Каждый из этих статусных битов имеет соответствующий бит разрешения прерывания в регистре GCTRL, который может использоваться для генерации прерывания, когда происходит изменение конкретного статусного бита.

Если используется только режим FS, то можно настроить DC контроллер не выполнять HS HANDSHAKE.

Различные тестовые режимы, требуемые стандартом USB, также доступны через регистр GCTRL. Существуют следующие тестовые режимы: Test_SE0_NAK, Test_J, Test_K и Test_Packet.

В режиме Test_SE0_NAK HS-приемник включен и отвечает NAK только на верные IN-транзакции (правильный CRC, адрес DC и номер ENDPOINT совпадают, PID не поврежден).

Test_J постоянно управляет состоянием HS J.

Test_K постоянно управляет состоянием HS K.

Test_Packet многократно посылает тестовые пакеты. Содержание пакета приведено в стандарте USB 2.0. Минимальная задержка между пакетами, в случае, когда DC контроллер посылает два или более последовательных пакетов, не оговорена стандартом. DC контроллер использует 192-битное время, как минимальную задержку, которая равна максимальному значению различных минимальных задержек в стандарте для любой последовательности пакетов и должна быть совместима с требованиями USB 2.0 стандарта.

Описание блока SIE

Блок SIE управляет приемом и передачей USB пакетов. DC не будет отвечать ни на какие транзакции, пока не будет получен RESET и установлен режим FS или HS.

Блок SIE всегда начинает работу с ожидания маркер-пакета (далее – TOKEN). В зависимости от типа TOKEN-пакета данные либо передаются из DC контроллера к хосту, либо, наоборот, принимаются DC контроллером от хоста. Специальные TOKEN-пакеты обрабатываются без какой-либо передачи данных. Специальные TOKEN-пакеты PING и SOF вызывают только отправку HANDSHAKE-пакета или сохраняют номер кадра. IN TOKEN-пакеты инициируют передачу данных от DC контроллера к хосту, в то время как SETUP TOKEN и OUT TOKEN передают данные от хоста к DC контроллеру. Пакеты, полученные на этапе приема TOKEN-пакета с PID, отличающимся от перечисленных в этом пункте, отбрасываются.

Пакеты данных передаются на следующем этапе, если TOKEN определил, что данные должны передаваться от DC контроллера к HOST. Когда передача данных завершена, DC контроллер ждет HANDSHAKE-пакета, прежде чем вернуться к этапу TOKEN.

Если TOKEN определил, что данные должны передаваться от HOST к DC контроллеру, тогда DC ожидает и принимает пакет с данными, а затем отправляет в ответ HANDSHAKE-пакет, перед тем как снова вернуться к этапу TOKEN.

Буферы конечных точек

Каждая из конечных точек (EPx) имеет два буфера, в которые сохраняются пакеты. DC контроллер автоматически переключается между ними после получения (либо передачи) пакета, так что данные из одного из буферов могут быть переданы по процессорной шине, в то время как новый пакет принимается по USB во второй буфер (либо передается из него). Состояние токена DATA1/DATA2 зависит о того, из какого буфера был отправлен или в какой буфер был принят пакет.

Состояние буферов влияет на отправку HANDSHAKE-пакета к HOST в конце транзакции. В HS режиме конечные точки, сконфигурированные как BULK OUT и CONTROL OUT, не находящиеся в фазе SETUP поддерживают PING протокол. Это значит, что в конце OUT транзакции к одной из этих конечных точек DC контроллер должен вернуть ACK, если принял текущие данные и имеется пространство для следующего пакета. Для DC контроллера это выполняется, если второй буфер конечной точки EPx пуст, к моменту готовности транзакции.

Если второй буфер не пуст, то отправляется NYET. Если текущие данные не принимаются (оба буфера не пусты, когда приходит пакет), то в таком случае отправляется NAK.

Для других типов конечных точек в режиме HS и для всех типов конечных точек в режиме FS всегда отправляется ACK, если данные приняты и NAK, если данные не приняты.

Буфер конечной точки можно настроить так, чтобы он превышал максимальный размер полезной нагрузки (maximum payload) для данной конечной точки. Для конечных

точек типа IN при записи данных большего размера, чем размер полезной нагрузки, буфером будут приняты некоторое количество пакетов максимального размера, заканчивающихся пакетом меньшего или эквивалентного максимальному размеру.

В направлении OUT большие буферы используются только для широкополосных (с высокой пропускной способностью) конечных точек, где может выполняться более одной транзакции на микрофрейм. В этом случае данные из всех пакетов одного микрофрейма запоминаются в одном буфере в том порядке, в котором они поступают, и после этого передаются на процессорную шину. Все не широкополосные (с невысокой пропускной способностью) конечные точки всегда хранят один пакет данных в буфере.

Буферы конечных точек в DC контроллере представляют собой два раздельных блока RAM для IN и для OUT конечных точек. Буферы для каждой отдельно взятой конечной точки выделяются общем пространстве внутри блока с целью избежать нерационального использования памяти

Блок интерфейса с процессорной шиной AIE

DC контроллер управляет процессорной шиной, и все данные передает и принимает в буферы конечных точек используя DMA операции. Существуют два отдельных блока DMA для IN и OUT направлений. Они мультиплексируются в один мастер интерфейс на AHB шине. Данная схема используется для уменьшения нагрузки на шину AHB.

Если оба DMA блока одновременно запрашивают шину, то ее предоставление будет всегда переключаться между ними. Например, если блоку DMA OUT в настоящий момент предоставлен доступ для работы с процессорной шиной, и после завершения обмена он снова выставит запрос, но в это же время DMA IN блок будет запрашивать процессорную шину, то доступ к процессорной шине сначала будет предоставлен DMA IN блоку. Если ситуация повторится, то право доступа снова перейдет к DMA OUT блоку.

DMA IN блок только читает данные из шины (это применимо только к данным, обозначенным в статусном дескрипторе) и всегда выполняет прием слова. Можно использовать любое выравнивание и длину байтов, это приведет к тому, что DC контроллер пропустит необходимое количество начальных и конечных байтов из первого и последнего прочитанных слов.

DMA OUT блок записывает данные на шину и выполняет передачи как слов, так и байтов. Если началась передача без выравнивания на слово, то запись байтов будет выполняться до тех пор, пока не будет достигнута граница слова. С этого момента и далее выполняется запись слов в пакетном режиме до тех пор, пока для передачи не останется меньше 4 байт. Если оставшееся число байт для передачи не равно нулю, то выполняется последний доступ. Запись байта всегда выполняется одиночным обращением к шине.

35.2 Операции DMA

Каждая IN и каждая OUT конечные точки имеют выделенный канал DMA, который передает данные во внутренние буферы конечной точки и из них, используя автономный DMA, опирающийся на дескрипторы. Каждая конечная точка в заданном направлении конкурирует за использование механизма DMA с другими конечными точками в том же направлении. Арбитраж выполняется циклически для всех конечных точек, которые активны и имеют данные для отправки или получения.

Операции DMA приблизительно одинаковые для обоих направлений обмена. Возможные различия будут описаны ниже.

Операции DMA основываются на связанном списке дескрипторов, расположенных в памяти. Каждая конечная точка EPx имеет свой собственный связанный список дескрипторов. Первое слово в дескрипторе является управляющим словом, которое содержит бит разрешения (бит EN), определяющий, является ли дескриптор активным или нет, а также другие управляющие биты. Следующее слово является указателем на буфер

памяти, в который записываются или читаются данные для этого дескриптора. Последнее слово является указателем на ячейку первого слова следующего дескриптора. Один бит в управляющем слове (бит NX) определяет действительный или нет следующий дескриптор. Если следующий дескриптор недействительный, тогда выборка дескрипторов прекращается и после обработки текущего дескриптора канал DMA останавливается.

Работа DMA начинается с настройки списка дескрипторов в памяти, после этого записывается указатель на первый дескриптор в регистр указателя дескриптора (DMADESC) для конечной точки EPx и установкой бита разрешения (бит DA в регистре DMACTRL). Регистр указателя обновляется по мере движения по списку и может читаться процессором. Если список заканчивается дескриптором, имеющим сброшенный бит разрешения (бит EN), то список нельзя затрагивать до тех пор, пока DC контроллер не закончит обработку списка и DMA канал не будет отключен. Иначе может произойти непредвиденная ситуация и поведение будет неопределенным.

Другой способ использования связанного списка – это всегда устанавливать бит разрешения в следующем дескрипторе и при этом быть уверенным, что последний дескриптор отключен. Таким способом новые дескрипторы могут добавляться в конец списка и разрешаться «на лету» при условии, что бит разрешения дескриптора всегда устанавливается после записи нового дескриптора в память. Такой способ даст уверенность, что не произойдет неопределенной ситуации и что новые дескрипторы не будут пропущены.

На рисунке 35.3 указана структура связанного списка дескрипторов.

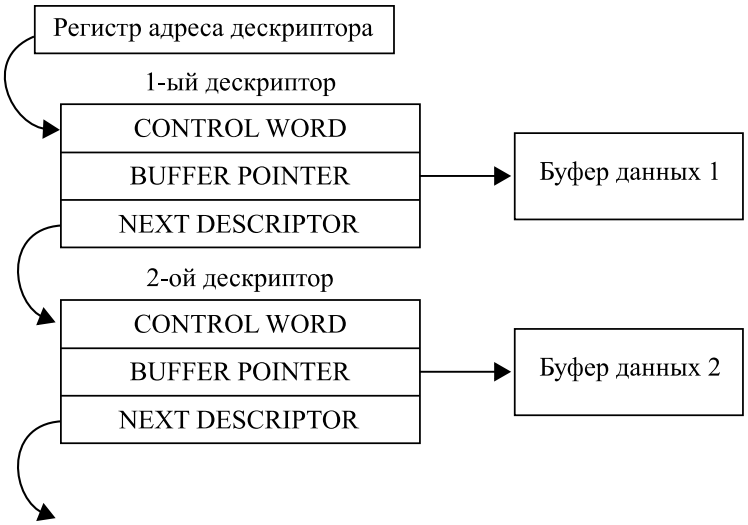


Рисунок 35.3 – Пример структуры связанного списка DMA дескрипторов

Операции с конечными точками OUT

Операции DMA для конечных точек типа OUT EP соответствуют основному описанию в подразделе «Операции DMA». Имеются небольшие отличия в отдельных битах и значении поля длины. Содержимое слов дескрипторов можно найти в таблицах 35.2 – 35.4.

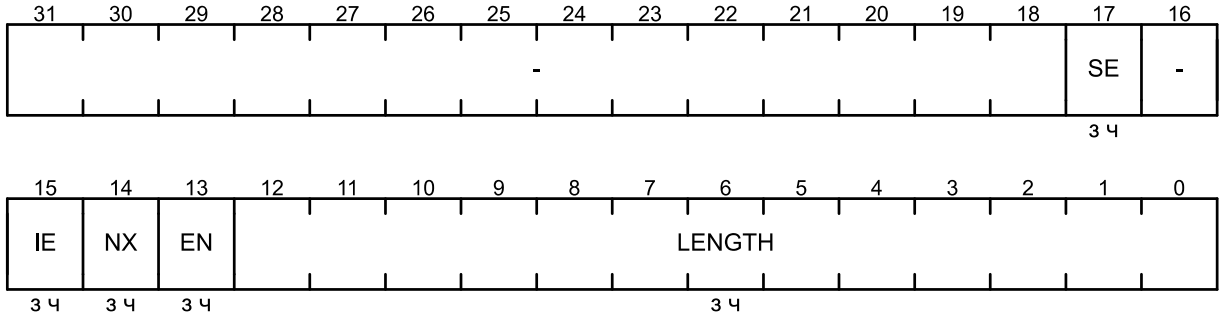
Когда дескриптор включен (установлен бит EN в слове 0), он будет считан DMA DC контроллера после установки бита доступности дескриптора (бит DA регистра EPOx.DMACTRL), и как только буфер для соответствующей конечной точки будет содержать данные, принятые с шины USB, полученные данные будут записаны в память, начиная с адреса, указанного в слове указателя буфера дескриптора. Содержимое одного буфера внутренней памяти DC контроллера всегда записывается в один буфер дескриптора. Это всегда соответствует одному USB пакету, за исключением широкополосных изохронных (ISOCHRONOUS) конечных точек и конечных точек прерывания

(INTERRUPT). Число записанных байт сохраняется в поле длины LENGTH по окончании записи в буфер, на что указывает сброшенный бит включения EN. Когда бит включения сбрасывается, пространство памяти может быть использовано снова.

Прерывания генерируются по запросу, как только завершится запись в память. Конечная точка может также конфигурироваться для немедленной генерации прерывания, как только пакет будет принят во внутренний буфер. Данная возможность не может быть включена отдельно для каждого пакета, поскольку DC контроллер не может заранее сопоставить полученный пакет с определенным дескриптором. Это прерывание включается в регистре управления (EPOx.CTRL) конечной точки.

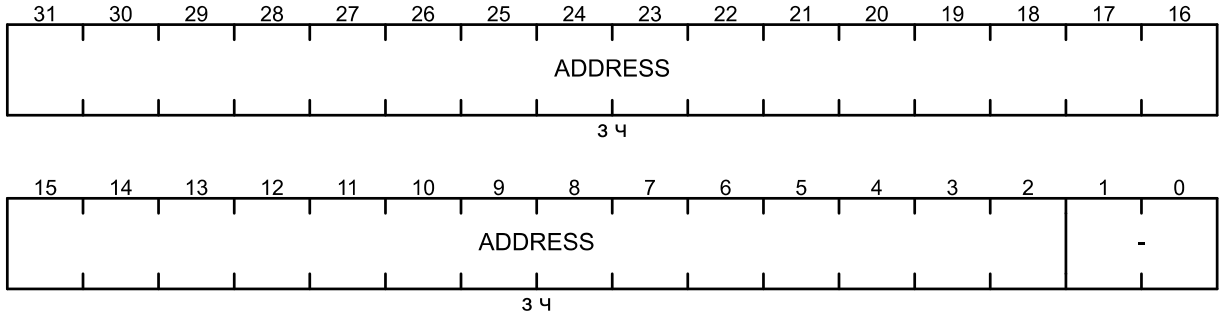
Когда данные были извлечены из внутреннего буфера, он очищается и может быть использован снова блоком SIE для приема нового пакета.

Таблица 35.2 – Описание битов слова «0» управляющего слова OUT дескриптора



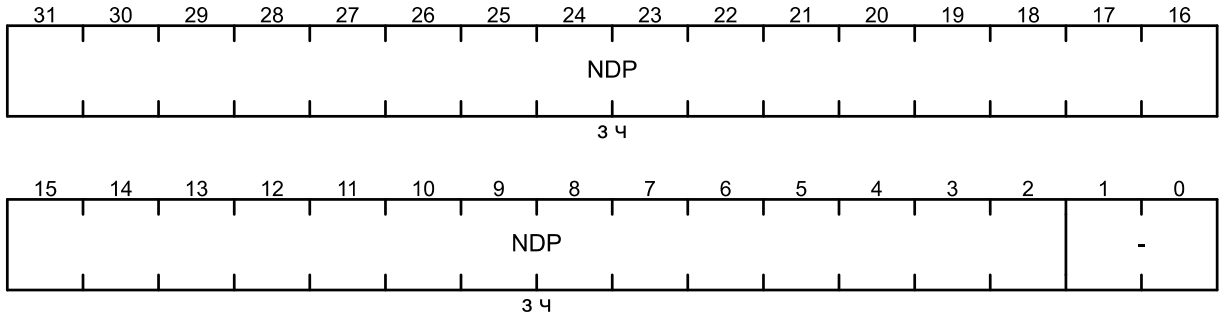
Поле	Биты	Описание
SE	17	Пакет SETUP. Данные были получены из пакета SETUP вместо пакета OUT.
IE	15	Разрешение прерывания. Прерывание будет генерироваться, когда пакет от этого дескриптора был принят во внутренний буфер и передан к SIE. Это не значит, что пакет также был передан в память.
NX	14	Следующий дескриптор готов. Поле следующего дескриптора действительно и указывает на следующий дескриптор.
EN	13	Бит разрешения дескриптора Установка этого бита в «1» разрешает использовать дескриптор. Он должен быть всегда установлен последним из всех полей дескриптора.
LENGTH	12-0	Число полученных байт. Действителен, когда бит EN сброшен DC контроллером.
–	31-18, 16	Зарезервировано

Таблица 35.3 – Описание полей слова «1» OUT дескриптора



Поле	Биты	Описание
ADDRESS	31-2	Указатель на область буфера, куда будут загружены данные пакета.
–	1-0	В целях выравнивания дескрипторов по 4 байта данные биты должны иметь нулевое значение

Таблица 35.4 – Описание полей слова «2» OUT дескриптора



Поле	Биты	Описание
NDP	31-2	Указатель на следующий дескриптор.
–	1-0	В целях выравнивания дескрипторов по 4 байта данные биты должны иметь нулевое значение

Примечание – Принятые условные обозначения: Ч – чтение; З – запись.

Операции с конечными точками IN

Операции DMA для конечных точек типа IN EP соответствуют описанным выше основным операциям DMA. Имеются небольшие различия в отдельных битах и значениях длины полей. Содержимое различных слов дескриптора представлено в таблицах 35.5 – 35.7.

Когда дескриптор включен (установлен бит EN в слове 0) и установлен бит доступности дескриптора (бит DA регистра EPIx.DMACTRL), DC контроллер начнет обработку дескриптора и извлечет количество байт, указанное в поле длины пакета (поле LENGTH) во внутренний буфер, принадлежащий конечной точке, как только он будет готов. Прерывание будет сгенерировано по запросу, когда данные будут записаны во внутренний буфер и статус будет записан обратно в дескриптор, но пакет может быть еще не передан на USB шину.

Отдельное прерывание может генерироваться, когда пакет фактически был передан. Для этого нужно установить в регистре управления (EPIx.CTRL) конечной точки бит PI, а также в дескрипторе (используя бит PI слова 0) для каждого пакета, по которому требуется генерация прерывания.

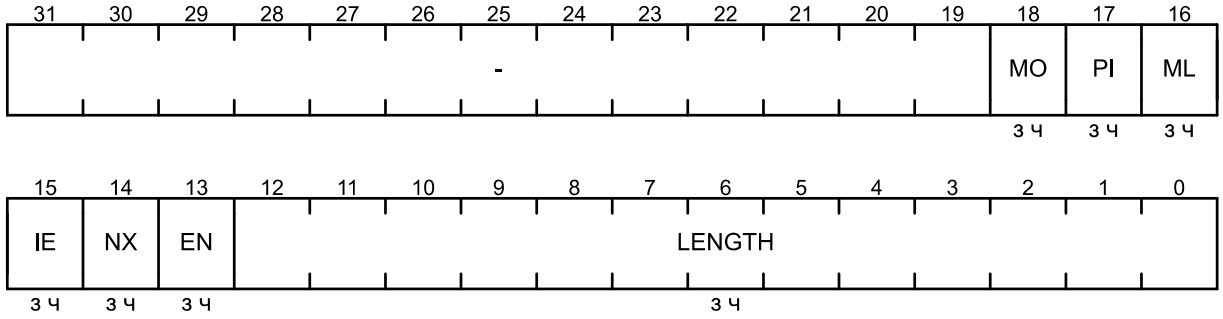
Дескриптор с нулевой длиной приведет к передаче пакета нулевой длины. Использование длины большей, чем максимально возможное значение полезной нагрузки для конечной точки приведет к передаче двух или более пакетов, все из которых будут иметь максимальную длину полезной нагрузки, кроме последнего. Последняя транзакция может быть равна или быть меньше максимально возможного значения полезной нагрузки для конечной точки. Если длина поля будет превышать размер внутреннего буфера, то данные не будут записаны в буфер, и в статусе дескриптора будет немедленно записана «1» в бит ошибки (ML).

Когда бит MO установлен, данные текущего дескриптора будут записаны во внутренний буфер, и после этого DC контроллер переходит к обработке следующего дескриптора без включения отправки данных из буфера. Данные следующего дескриптора читаются в тот же буфер, и это будет продолжаться до тех пор, пока не будет обнаружен дескриптор со сброшенным битом MO.

Если общее количество байт становится большим, чем размер внутреннего буфера, то пакет не отсылается (данные из внутреннего буфера теряются), и бит ML устанавливается в «1» на последнем дескрипторе. После этого выборка дескриптора начинается снова.

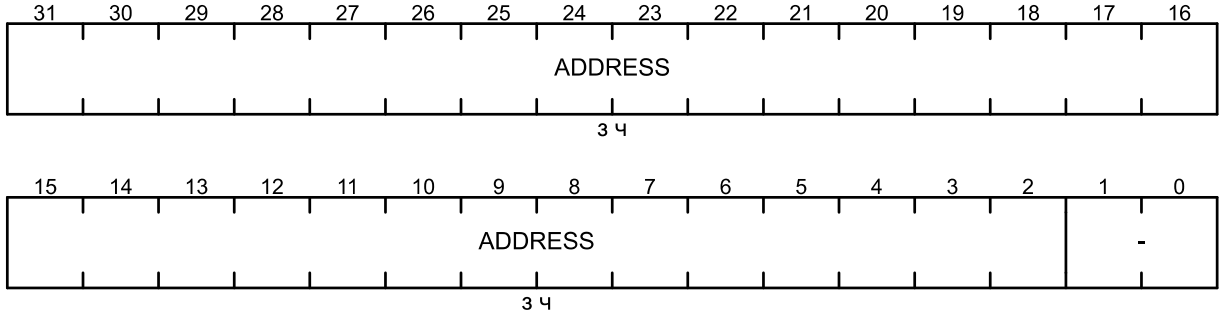
Если бит NX не установлен, при установленном бите MO, тогда DC контроллер будет ждать дескриптор, не позволяя другим конечным точкам получить доступ к процессорной шине.

Таблица 35.5 – Описание битов слова «0» IN дескриптора



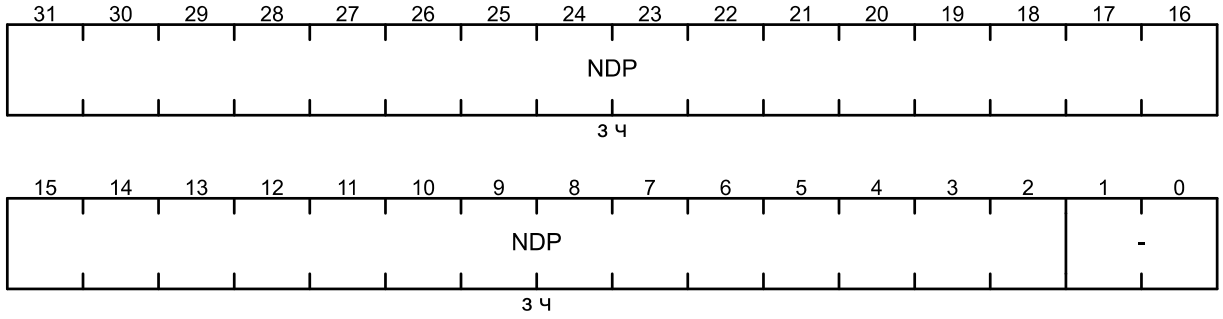
Поле	Биты	Описание
MO	18	MORE. Данные следующего дескриптора должны читаться в тот же буфер.
PI	17	Генерирует прерывание, когда пакет был передан на USB шину.
ML	16	Бит ML устанавливается в «1», если была попытка передать данные, объем которых больше, чем размер буфера.
IE	15	Разрешение прерывания. Прерывание будет генерироваться, когда пакет для этого дескриптора был прочитан во внутренний буфер и передан SIE. Но это не означает, что пакет был передан.
NX	14	Поле следующего дескриптора. Когда бит NX установлен в «1», нужно обрабатывать следующий дескриптор.
EN	13	Бит разрешения дескриптора. Устанавливается в «1» для разрешения дескриптора. Этот бит всегда устанавливается последним из всех полей дескриптора.
LENGTH	12-0	Длина. Число байт, которые должны быть переданы.
–	31-19	Зарезервировано

Таблица 35.6 – Описание битов слова «1» IN дескриптора



Поле	Биты	Описание
ADDRESS	31-2	Указатель на область буфера, откуда данные пакета были загружены.
–	1-0	В целях выравнивания дескрипторов по 4 байта данные биты должны иметь нулевое значение

Таблица 35.7 – Описание битов слова «2» IN дескриптора



Поле	Биты	Описание
NDP	31-2	Указатель на следующий дескриптор.
–	1-0	В целях выравнивания дескрипторов по 4 байта данные биты должны иметь нулевое значение

Примечание – Принятые условные обозначения: Ч – чтение; З – запись.

35.3 Конечные точки EP

Для каждой конечной точки перед использованием необходимо настроить функции AHB и USB. Конфигурация процессорной шины включает операции DMA, описанные в предыдущих разделах. Конфигурация USB включает: включение конечной точки для USB транзакций, установку типа передачи (CONTROL, BULK, ISOCHRONOUS, INTERRUPT), размер полезной нагрузки, высокую пропускную способность и др. Выбор конфигурации выполняется через регистры EPOx.CTRL/EPIx.CTRL, доступные для программной конфигурации через шину AHB.

Когда выбор конфигурации выполнен, должен быть установлен бит готовности конечной точки (бит EV регистра EPOx.CTRL/EPIx.CTRL). Это позволит разрешить обмен данными с этой конечной точкой, как только будет получен сброс по USB. При переконфигурации конечной точки первым делом должен быть сброшен бит готовности EV. В противном случае конечная точка будет проинициализирована некорректно. При включении конечной точки схема переключения сбрасывается, буферы данных очищаются и селектор буфера устанавливается на нулевой буфер. Поля максимальной полезной нагрузки MAXPL, количество дополнительных транзакций NT (для изохронных точек) и тип обмена TT могут быть изменены только тогда, когда бит готовности EV конечной точки сброшен, либо перед повторным установлением бита EV после выключения конечной точки. Остальные биты регистра управления конечной точки могут быть изменены в любое время.

При включенной конечной точке настройки конфигурации не должны меняться, за исключением битов остановки EH и выключения ED.

Конечную точку можно остановить установкой бита EH (Endpoint halted) регистра EPOx.CTRL/EPIx.CTRL. Это вызовет отправку пакета подтверждения STALL в ответ на все принятые транзакции. При снятии бита EH схема переключения также сбрасывается, как это требует USB стандарт.

Когда конечная точка настроена, передача данных в конечную точку и из нее могут быть возобновлены. Нет никаких различий между передачами данных по процессорной шине, зависящих от выбранного типа передачи. Это влияет только на передачу данных по шине USB. Для конечных точек управления (CONTROL EP) может потребоваться дополнительная обработка при возникновении ошибок, что будет объяснено далее в подразделе «Управляющие конечные точки».

Пакеты, полученные конечной точкой (независимо от типа конечной точки) с большим числом байт, чем сконфигурированное максимальное значение для конечной точки, в ответ получат пакет подтверждения STALL и заставят конечную точку перейти в режим HALT.

При обнаружении сброса по шине USB, биты CS, ED и EH регистра управления (EPOx.CTRL/EPIx.CTRL) конечной точки будут сброшены для всех конечных точек. Бит EV будет также очищаться за исключением управляющей конечной точки (CONTROL EP).

Бит CB в управляющем регистре (EPOx.CTRL/EPIx.CTRL) конечной точки предназначен для очистки внутренних буферов конечной точки. При его установке данные в буферах будут очищены, биты готовности данных для соответствующих буферов будут сброшены, и в конце бит CB также очищается. Однако это не сработает, если в данный момент активна транзакция с данной конечной точкой, поэтому следует использовать эту функцию с осторожностью.

Управляющие конечные точки CONTROL EP

В соответствии с USB стандартом конечная точка с номером «0» всегда является управляющей конечной точкой (CONTROL EP) и должна быть доступной по сбросу USB. DC контроллер не принимает никаких транзакций до тех пор, пока не будет получен сброс по USB. Таким образом, управляющая конечная точка может быть включена непосредственно после включения питания. При необходимости можно задействовать больше управляющих конечных точек с теми же ограничениями, что и конечная точка по умолчанию, за исключением того, что они должны быть доступны только после завершения конфигурирования. Если функция управления DC контроллером медленная, во время запуска DC контроллера управляющая конечная точка может не сконфигурироваться должным образом перед получением сброса по USB. Этой проблемы можно избежать, поскольку после получения сброса подтягивающий резистор на линии D+ отключается, что дает полный функциональный контроль над тем, когда DC контроллер будет виден на USB шине.

Управляющая конечная точка EP0 является каналом обмена сообщениями, и, следовательно, передача данных может осуществляться как в направлении IN, так и в направлении OUT. Поэтому управляющие конечные точки должны использовать конечные точки обоих направлений с одним и тем же номером в DC контроллере. Это требует конфигурирования обоих направлений в одном и том же режиме (один тип передачи, объем принимаемых данных и т. п.). Иначе поведение DC контроллера будет неопределенным.

Управляющая передача всегда начинается с SETUP транзакции, которая всегда будет получена конечной точкой OUT EP. Если управляющая передача с записью данных, тогда в фазе DATA передача данных будет идти в направлении OUT, и эти данные будут получены конечной точкой OUT EP. Функция обработчик должна считывать как SETUP данные, так и другие приходящие данные и реагировать соответствующим образом. Если обнаружится ошибка, то необходимо остановить конечную точку (перевести в состояние HALT). Существуют две альтернативы для этого: остановка без сброса состояния HALT, которое сохранится даже после следующей SETUP транзакции, или остановка со сбросом состояния HALT, которое будет очищаться к моменту, когда будет принят следующий SETUP. Последнее является рекомендуемым поведением в USB стандарте, поскольку в противном случае потребуется полный сброс DC контроллера для продолжения работы, если на управляющей конечной точке по умолчанию EP0 установится постоянный HALT.

DC контроллер может обнаруживать ошибки в одиночных транзакциях, тогда конечная точка переводится в режим остановки (HALT) автоматически. В таком случае для управляющих конечных точек будет использоваться функция очистки состояния остановки (HALT).

Если SETUP транзакция указывает на то, что это управляющее чтение, то в фазе DATA передача данных будет в направлении IN. В этом случае пользователь DC

контроллера должен разрешить передачу данных для конечной точки в IN направлении, если запрос был принят, в противном случае необходимо установить функцию HALT. Передача заканчивается, когда хост отправляет пакет нулевой длины в конечную точку OUT EP.

Стоит обратить внимание на то, что при вводе состояния остановки (HALT) для управляющей конечной точки, также должны быть установлены биты HALT для конечных точек IN и OUT.

Каждый раз, когда управляющая конечная точка получает маркер-пакет SETUP, буферы в направлении IN очищаются. Это сделано для предотвращения несоответствий, если фазы DATA и STATUS отсутствовали или были повреждены и, таким образом, данные никогда не извлекались. Старые данные по-прежнему останутся в буфере, и следующая SETUP транзакция получит ошибочные данные. Стандарт USB утверждает, что это может произойти при возникновении ошибок и новый SETUP пакет может быть передан прежде, чем предыдущая передача будет окончена. Пользователь DC контроллера может очистить буфер самостоятельно через регистр управления конечной точки IN (регистр EPIx.CTRL, бит CB), рекомендуется это сделать, как только обнаруживается новый SETUP-пакет до окончания предыдущей передачи.

Это необходимо сделать, так как пользователь мог включить буферы после того, как DC контроллер очистил их при получении нового SETUP-пакета.

В статусном бите дескриптора (бит SE слова 0) указывается, были ли данные получены во время SETUP транзакции или же во время OUT транзакции.

Конечные точки массового обмена BULK EP

Конечные точки в режиме BULK являются потоковыми магистралями и, следовательно, могут использоваться только в одном направлении: либо IN, либо OUT. Конечная точка с тем же номером в другом направлении может использоваться независимо. Доступ к данным осуществляется через интерфейс AHB, и не требуется никаких специальных мер, помимо общих рекомендаций по работе с конечными точками.

Конечные точки прерывания INTERRUPT EP

Данные от конечных точек в режиме INTERRUPT, обрабатываются таким же образом, как и в режиме BULK. Отличия появляются только на USB шине. Эти конечные точки также являются потоковыми.

Конечные точки INTERRUPT поддерживают режим высокой пропускной способности, что означает выполнение более одной транзакция на микрофрейм. Для этого требуются буферы большего размера, чем максимальный размер полезной нагрузки. Конечная точка должна конфигурироваться с размером буфера большим или эквивалентным полезной нагрузке, умноженной на количество транзакций. Все транзакции будут приниматься в один и тот же буфер и передаваться из него.

Изохронные конечные точки ISOCHRONOUS EP

Конечные точки в режиме ISOCHRONOUS также являются потоковыми и идентичны другим конечным точкам в части обработки через AHB шину.

Главное отличие изохронных конечных точек от других состоит в том, что в этом режиме отсутствуют пакеты подтверждения HANDSHAKE. Если во время поступления IN маркера нет доступных данных для передачи, тогда изохронная конечная точка передает пакет нулевой длины. Это укажет хосту, что ошибки не произошло, просто данные не готовы. Если пакет не будет отправлен, хост не будет знать, что произошло и был ли пакет поврежден или нет.

Когда EP не находится в режиме высокой пропускной способности, только одна транзакция в направлении OUT будет сохранена в одном буфере. В режиме высокой пропускной способности все транзакции микрофрейма сохраняются в одном буфере.

В IN направлении данные всегда передаются из одного и того же буфера, до тех пор, пока в нем не закончатся данные. Для высокоскоростных конечных точек размер буфера должен конфигурироваться как размер максимальной полезной нагрузки, умноженный на количество транзакций.

Высокоскоростные изохронные конечные точки используют PID последовательность. При обнаружении ошибки в PID последовательности в OUT направлении, по процессорной шине не передаются никакие данные для всего микрофрейма.

35.4 Пример реализации DC контроллера

Потребуется функция управления DC контроллером, реализующая конкретное прикладное устройство. Единственное требование, чтобы функция управляла DC контроллером через процессорную шину.

Прежде всего, для работы необходим правильно сконфигурированный PHY. Он конфигурируется автоматически DC контроллером.

После этого DC контроллер ждет подключения к шине USB анализом сигнала VBUS. Состояние сигнала VBUS может определяться опросом флага или прерыванием. Функция может контролировать время подключения с помощью бита EP (включение/выключение подтяжки) управляющего регистра GCTRL. Если бит EP сброшен, USB хост может не обнаружить DC контроллер, даже если он подключен.

После подключения необходимо получить USB сброс от хоста, прежде чем транзакции смогут приниматься DC контроллером. Это также определяется опросом флага или прерыванием.

После сброса должна быть доступна только управляющая конечная точка EP0. Функция управления DC контроллером отвечает за включение и настройку в нужное время. Функция может подождать, пока DC контроллер получит USB сброс, но проще запустить ее сразу после включения питания. Это возможно сделать, так как DC контроллер не будет принимать никаких транзакций до тех пор, пока не будет получен USB сброс. После включения DC контроллера также должны быть включены дескрипторы конечных точек как в направлении IN, так и в направлении OUT, а также должны быть установлены биты разрешения дескрипторов (бит DA регистра DMACTRL).

После этого конечная точка готова принимать пакеты, и управляющая функция должна ожидать прибытие пакетов SETUP. Функция может получить информацию о прибывающих пакетах либо через опрос флага, либо с помощью прерывания. DC контроллер должен обработать запросы и выдавать дескрипторы в соответствии с запросом. Когда приходит запрос на установку адреса (SET ADDRESS), управляющая программа DC контроллера должна записать новый адрес в регистр управления (поле UA регистра GCTRL). Данный адрес вступит в силу после следующей успешной IN транзакции для управляющей конечной точки. Эта процедура должна совпадать со стадией статуса передачи адреса (SET ADDRESS).

При получении запроса на установку конфигурации (SET CONFIGURATION), управляющая программа DC контроллера должна включить подходящий интерфейс и конечные точки в соответствии с выбранной конфигурацией. Это выполняется путем записи управляющих регистров конечных точек (EPOx.CTRL и EPIx.CTRL). Управляющая программа отвечает за информирование о конфигурациях и интерфейсах с помощью дескрипторов, запрашиваемых SETUP транзакциями.

Когда конечные точки для выбранной конфигурации включены, программа также должна настроить работу DMA. После этого DC контроллер готов для передачи и приема данных через конечные точки, определенные программой. Прерывания могут использоваться для уведомления о передаче новых пакетов, а затем, через опрос флагов, можно будет определить, какая конечная точка изменила состояние.

36 Контроллер интерфейса USB 2.0 HC

36.1 Общие сведения

Контроллер хост-системы USB 2.0 обеспечивает связь между шиной AHB и универсальной последовательной шиной (USB). Хост-контроллер поддерживает поток обмена USB в высокоскоростном (High-Speed), полноскоростном (Full-Speed) и низкоскоростном (Low-Speed) режимах. Для поддержки высокоскоростного режима USB 2.0 (High-Speed) используется усовершенствованный контроллер (EHCI 1.0). Полноскоростной (Full-Speed) и низкоскоростной (Low-Speed) режимы обеспечиваются несколькими контроллерами-спутниками, реализующим интерфейс версии 1.1 (UHCI 1.1). На рисунке 36.1 показана блок-схема хост-системы USB 2.0.

Управляющие регистры контроллера USB HC приведены в приложении А.29.

Контроллер поддерживает приемо-передатчики UTMI+.

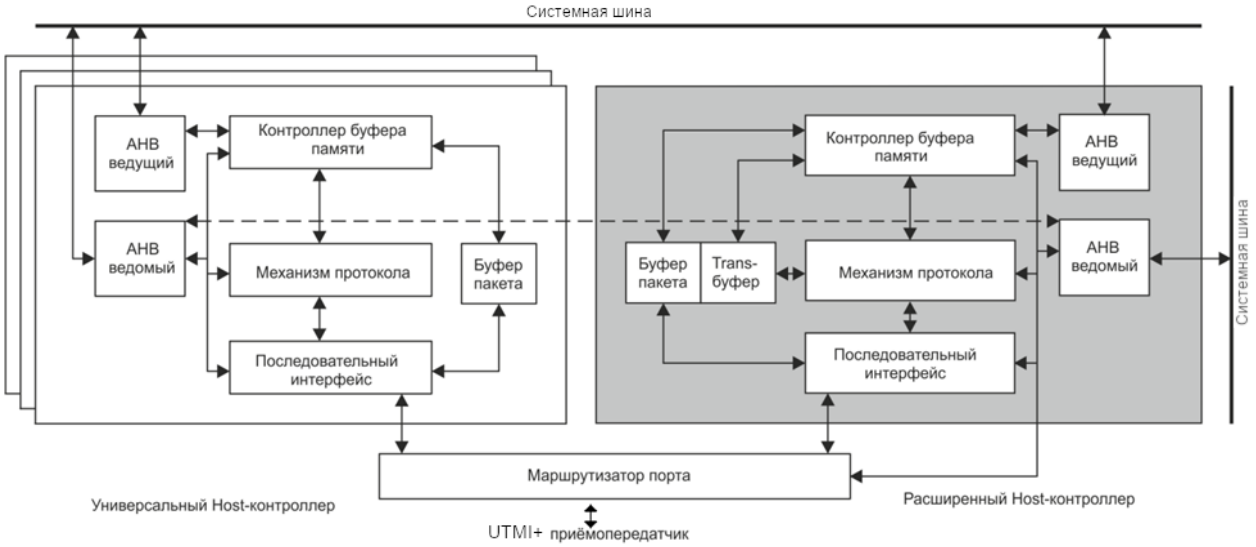


Рисунок 36.1 – Блок-схема хост-системы

Контроллер хост-системы имеет возможность подключения как к внутреннему PHY (без поддержки режима HS), так и ко внешней микросхеме PHY, подключаемой к выводам GPIO с использованием альтернативных функций (см. рисунок 36.2). Выбор внутреннего или внешнего PHY определяется битом EXTPHY регистра PHYCFG0.

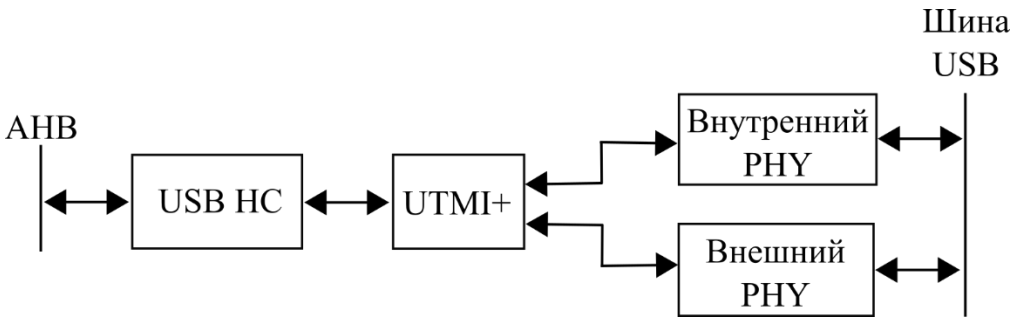


Рисунок 36.2 – Подключение HC контроллера к PHY

36.1 Принцип работы

Общие сведения о системе

В системе содержатся два типа контроллеров: усовершенствованный контроллер и универсальный контроллер. Универсальный контроллер работает, как сопровождающий контроллер усовершенствованного контроллера.

Усовершенствованный контроллер соответствует интерфейсу EHCI за исключением опции Light Host Controller Reset (см. спецификацию Enhanced Host Controller Interface), которая не реализована.

Универсальный хост-контроллер соответствует интерфейсу UHCI с некоторыми исключениями. Бит HCH в регистре статуса USB (UHCCMDSTS) доступен только для чтения, вместо чтения/записи. Регистр статуса/управления портом (UHCPORTSC) расширен битами «превышение тока» (Over Current Active, бит OA) и «изменение превышения тока» (Over Current Change, бит OC). Изменения в обоих регистрах были внесены в соответствии с актуальной реализацией интерфейса. Оба изменения соответствуют описанию соответствующих битов в спецификации EHCI.

Поддержка протокола

Усовершенствованный контроллер полностью поддерживает поток обмена в высокоскоростном режиме (High-Speed) в соответствии со спецификацией USB 2.0. Кроме того, поддерживается режим Asynchronous Park Mode, а контроллер оснащен счетчиком NAK.

Универсальный контроллер хост системы поддерживает поток обмена в полноскоростном (Full-Speed) и низкоскоростном (Low-Speed) режимах.

Дескриптор и буферизация данных

Усовершенствованный контроллер хост-системы выполняет предвыборку одного кадра изохронных дескрипторов. Перед выполнением передачи производится выборка всех данных полезной нагрузки. Усовершенствованный контроллер имеет 2048-байтовый буфер для дескрипторов и 2048-байтовый буфер для данных полезной нагрузки, которые могут содержать данные для двух передач.

Универсальный контроллер хост-системы не выполняет предвыборку дескрипторов. Передача на шине может начинаться до извлечения из памяти всех данных полезной нагрузки. Универсальный контроллер имеет 1024-байтовый буфер для данных полезной нагрузки. Дескриптор передачи в UHCI может описывать передачу, полезная нагрузка которой составляет 1280 байт. Согласно спецификации USB, максимально допустимая полезная нагрузка данных ограничена 1023 байтами, поэтому контроллер не будет передавать полезную нагрузку, превышающую 1023 байта. Если дескриптор имеет разрешенную полезную нагрузку, превышающую 1023 байта, контроллер попытается передать только первые 1023 байта, прежде чем передача будет помечена как завершенная.

Поскольку хост-контроллер имеет только один порт, универсальный контроллер и усовершенствованный контроллер используют общий буфер данных полезной нагрузки. Таким образом, требуется только два 2048-байтных буфера.

Тактирование и сброс

Ядро модуля USB содержит два тактовых домена: системный тактовый домен и тактовый домен USB. Тактовый домен USB всегда работает с частотой либо 60 МГц, либо 30 МГц, в зависимости от интерфейса приемопередатчика. Все сигналы, пересекающие границу тактового домена, синхронизированы во избежание нестабильности.

Переключение входа сброса может быть асинхронным или синхронным. Все регистры в системном тактовом домене, требующие сброса, сбрасываются синхронно. Предполагается, что вход сброса удерживается установленным до стабилизации сигнала тактирования. Во избежание нежелательных действий USB, несколько регистров в

тактовом домене USB сбрасываются асинхронно. При старте тактирования USB все остальные регистры в домене USB, требующие значения по сбросу, также сбрасываются.

От входного сигнала сброса устройство также генерирует сигнал сброса приемо-передатчиков USB, который асинхронно устанавливается применительно к тактовому сигналу USB, и асинхронно снимается.

36.2 Операции DMA

В обоих типах контроллеров размер пакетов DMA составляет 16 слов.

Это значение показывает, сколько слов в пакете контроллер может получить из памяти, а не количество операций с памятью при получении доступа к шине.

При записи данных полезной нагрузки обратно в память, требующей адресации по байтам или в полусловах, количество операций может достигнуть 17, прежде чем шина будет освобождена.

Если хост-контроллером дается побайтно выровненный буфер данных, длина пакета может достичь 17 слов при выборке данных полезной нагрузки из памяти.

Универсальный контроллер хост системы использует пакет размером в четыре слова при выборке дескрипторов. Конфигурация универсального контроллера позволяет запускать передачи по USB до полного извлечения данных из памяти. Количество слов, которые должны быть извлечены из памяти до запуска передачи по USB, равно двум.

36.3 Поддержка приемо-передатчиков

Контроллер поддерживает приемо-передатчики UTMI+. При использовании приемопередатчиков UTMI+, для хост-контроллера не имеет значения, как обрабатываются генерация VBUS и обнаружение неисправностей VBUS, пока сигнал включения VBUS и индикатор неисправности VBUS подключены к сигналам drvvbus и vbusvalid. Спецификация UTMI+ определяет эти два сигнала как сигналы с высоким активным уровнем.

Описание интерфейса UTMI+ приводится в *USB 2.0 Transceiver Macrocell Interface (UTMI) Specification* и *UTMI+ Specification Revision 1.0*.

37 Блок физического уровня интерфейса USB 1.1 (PHY)

37.1 Назначение блока

Микроконтроллер содержит два идентичных блока USB PHY.

Блок USB FS PHY предназначен для реализации аналогового (физического) интерфейса USB в режиме Full Speed.

Управляющие регистры блоков USB PHY приведены в приложении А.30.

37.2 Общая структура блока

Общая структура разработанного блока USB PHY представлена на рисунке 37.1.

Аналоговая часть блока реализует следующие функции:

– осуществляет преобразование поступающих на вход последовательных данных в аналоговый сигнал, пригодный для передачи по шине USB;

– осуществляет преобразование аналогового сигнала на шине USB в цифровую последовательность, пригодную для дальнейшей обработки;

– детектирует состояния SE0 и SE1 на шине USB;

– обеспечивает подключение и отключение подтягивающих и согласующих сопротивлений на выводах DP и DM;

– вырабатывает набор опорных токов и напряжений, необходимых для своей работы;

– обеспечивает подстройку выходного сопротивления передатчика.

Цифровая часть блока реализует следующие функции:

– осуществляет кодирование битового потока для передачи;

– осуществляет декодирование входного битового потока;

– обеспечивает добавление и удаление битов из потока;

– обеспечивает сериализацию передаваемых данных;

– обеспечивает десериализацию принимаемых данных;

– реализует восстановление данных и тактового сигнала при приеме данных;

– обеспечивает выработку сигналов для управления аналоговой частью блока.

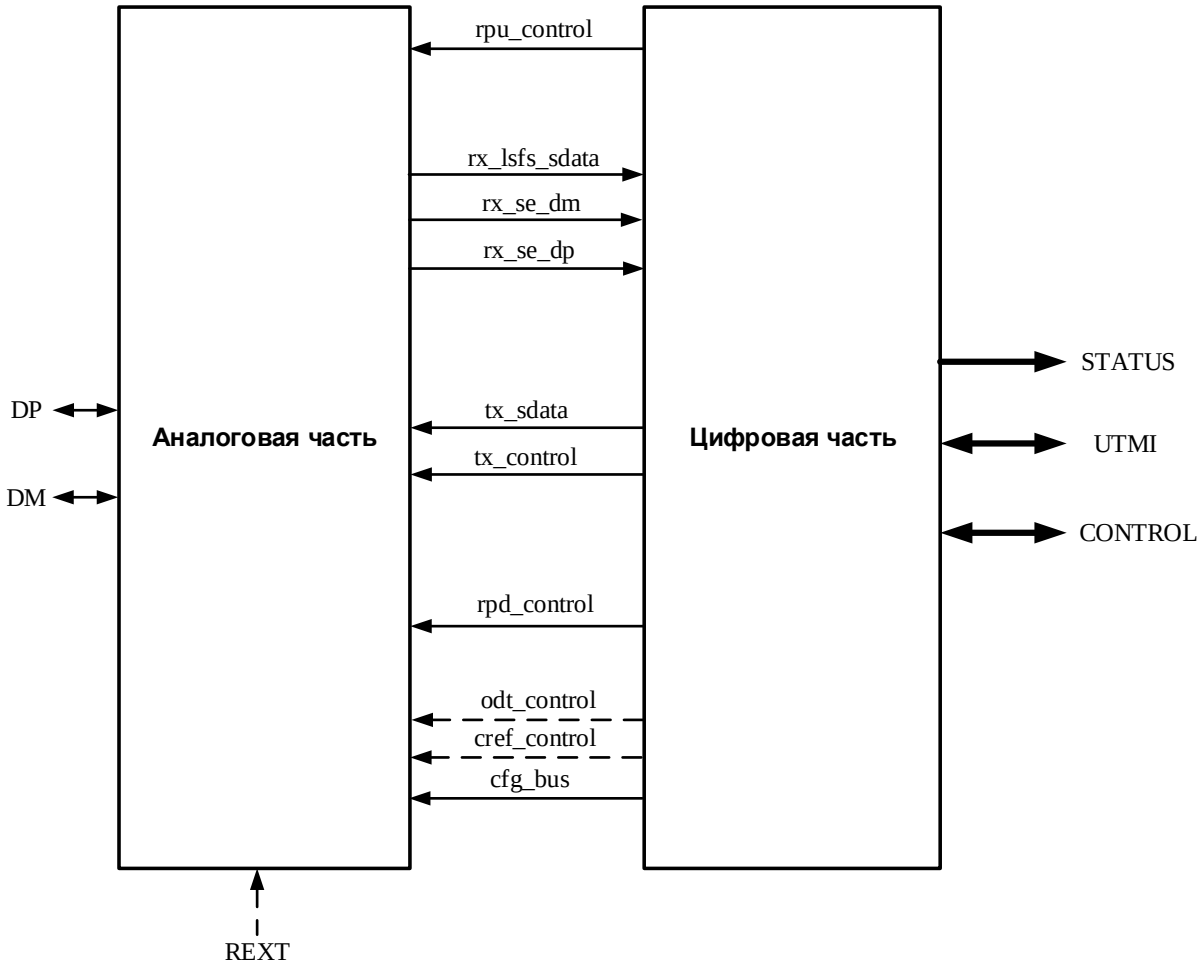


Рисунок 37.1 – Общая структура блока USB PHY

Для сигналов DP, DM, REXT, ATP_0, ATP_1 домен питания 3,3 В (VCCH), для остальных сигналов – 1,1 В (vcc).

37.3 Описание цифровой части USB PHY

Общая структурная схема цифровой части блока представлена на рисунке 37.2.
В представленной структуре выделяются следующие основные блоки:
– блок сериализации данных;
– блок добавления битов в поток (bit-stuffer);
– кодер NRZI;
– блок управления передающей частью;
– блок восстановления данных и тактового сигнала;
– детекторы начала и конца пакета (детектор SYNC и детектор EOP);
– декодер NRZI;
– блок удаления битов из потока (bit-unstuffer);
– блок десериализации данных;
– блок управления приемной частью;
– детектор состояния линии;
– блок фильтрации состояния SE0;
– блок управления приемопередатчиком;
– детектор разрыва соединения;
– блок конфигурации и статуса.

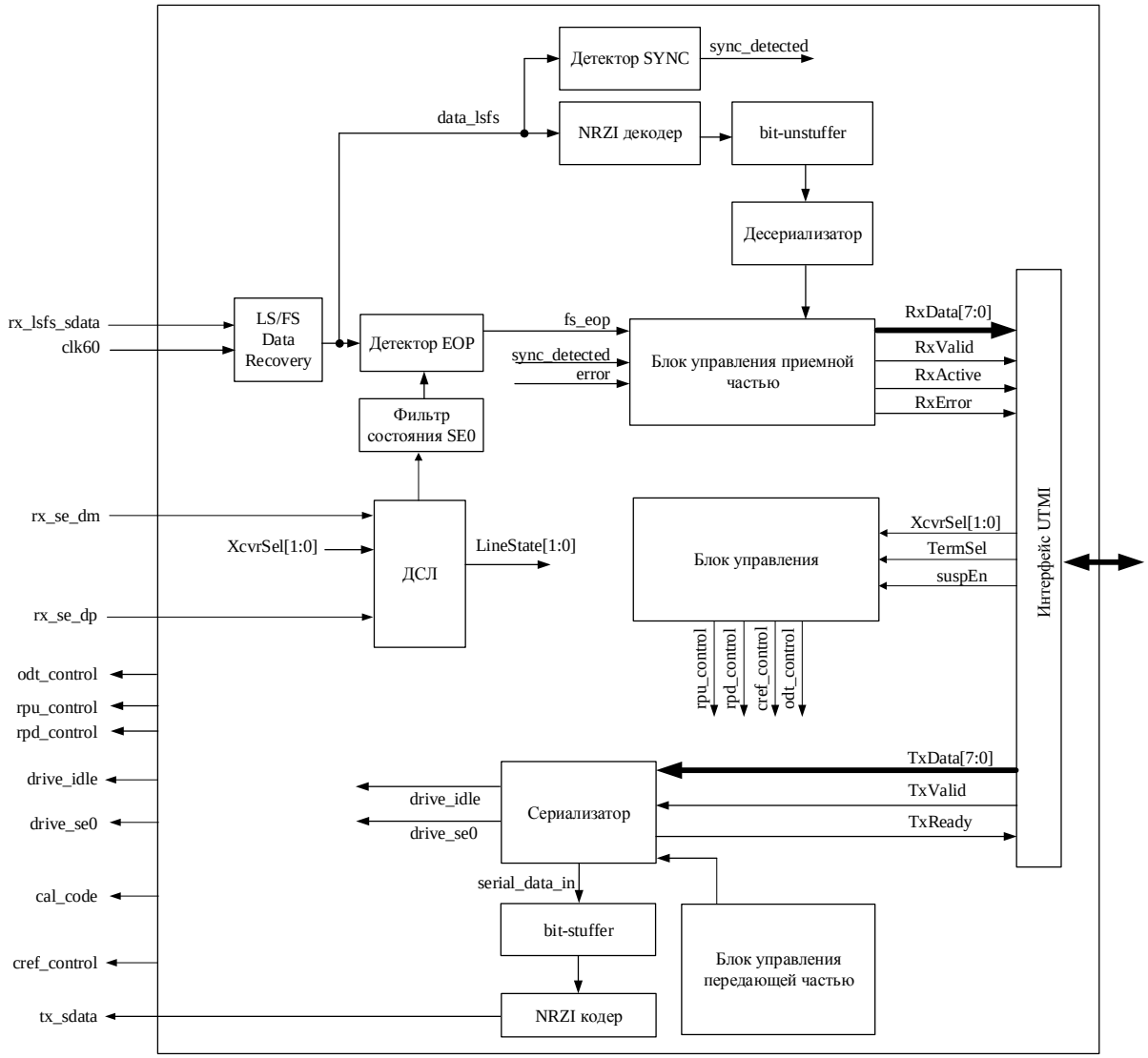


Рисунок 37.2 – Общая структура цифровой части блока USB PHY

Детектор состояния линии (ДСЛ)

Детектор состояния линии анализирует текущее состояние сигналов на шине USB и формирует соответствующий код на шине LineState[1:0].

Шина LineState[1:0] является частью интерфейса UTMI и имеет следующее назначение битов:

– бит [0] соответствует состоянию сигнала DP;

– бит [1] соответствует состоянию сигнала DM.

Для режима работы Full Speed код на шине LineState всегда соответствует текущему состоянию шины USB.

В таблице 37.1 приведены возможные значения для шины LineState[1:0] для режима работы Full Speed.

Таблица 37.1 – Кодирование значений шины LineState[1:0]

Значение LineState[1:0]	Режим Full Speed
00b	SE0
01b	FS-J
10b	FS-K
11b	SE1

Поскольку для различных скоростных режимов должны быть использованы разные пороговые напряжения для определения состояния шины USB, блок детектора состояний линии использует сигнал с выхода приемника с несимметричным входом. Таким образом, используя сигналы PortType (сигнал определяет направленность порта), XcvrSelect[1:0], TermSelect и OpMode[1:0] определяется текущий режим работы и используемый тип приемника. На рисунке 37.3 изображена схема блока ДСЛ с обозначением основных входов.

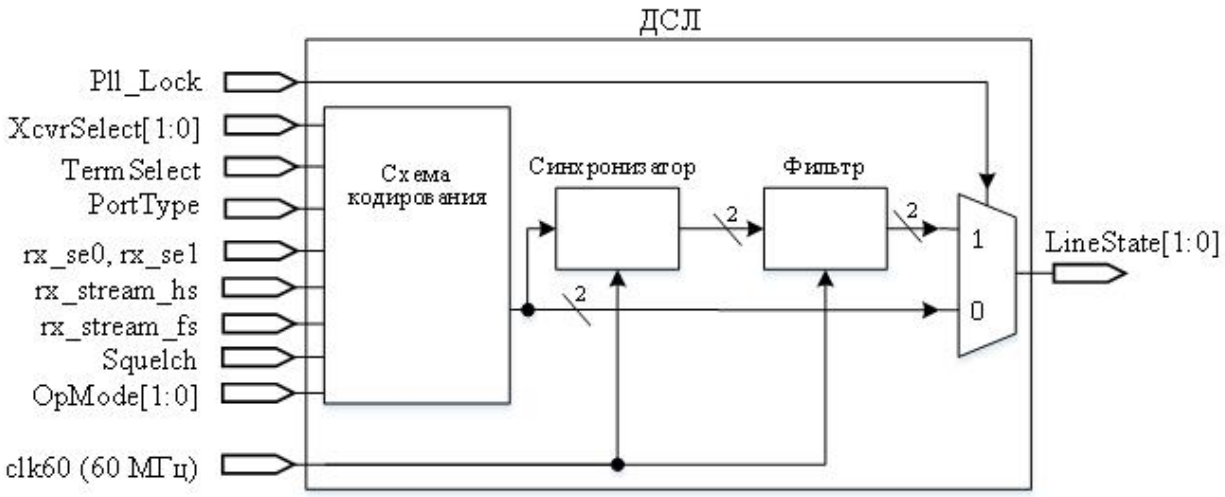


Рисунок 37.3 – Структурная схема детектора состояния линии (ДСЛ)

Спецификация UTMI определяет параметры фильтрации состояния SE0 на шине USB при формировании кода LineState[1:0]. Это требование исходит из спецификации USB, которая задает максимально допустимую длительность состояния SE0 на шине USB при переключении сигналов DP и DM. В связи с этим, в блоке ДСЛ применен цифровой фильтр, ограничивающий прохождение данного состояния на шину LineState[1:0] при длительности меньше пороговой. Минимальная длительность состояния SE0 составляет 2 периода тактового сигнала CLK для режима FS.

Спецификация UTMI также определяет поведение шины LineState[1:0] при отсутствии стабильного тактового сигнала. В таком случае, в соответствии со спецификацией UTMI (пункт 4.1.4.1), на вывод LineState[1:0] выдается код без какой-либо синхронизации и фильтрации. Как только частота тактового сигнала интерфейса UTMI стабилизирована, происходит переключение на выход цифрового фильтра.

В режиме работы Full Speed вывод LineState[0] всегда отражает сигнал DP, а LineState[1] отражает сигнал DM. На рисунке 37.4 приведена диаграмма работы ДСЛ.

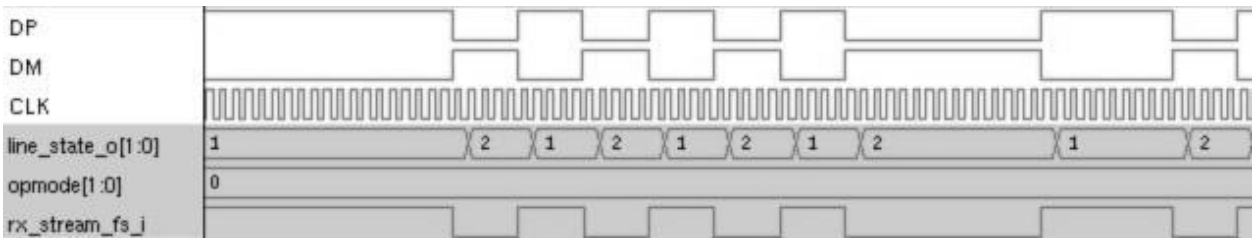


Рисунок 37.4 – Детектор состояния линии в режиме FS

Блок восстановления данных в режиме FS

Блок восстановления данных в режиме FS выполняет функцию синхронизации потока данных дифференциального приемника FS с локальным тактовым сигналом, а также выделения признака достоверности для каждого бита данных. Блок функционирует в режиме работы на полной (FS) скорости, что соответствует номинальной скорости битового потока 12 Мбит/с. Для синхронизации данных используется локальный тактовый сигнал с частотой 60 МГц. Поскольку период тактового сигнала в пять раз меньше битового интервала данных на скорости Full Speed, то это позволяет использовать метод избыточной дискретизации (англ.: oversampling) для подстройки и формирования признака достоверности.

Структурно блок состоит из фильтра-синхронизатора, детектора фронта, таймера и формирователя флага достоверности данных. Структурная схема блока приведена на рисунке 37.5.

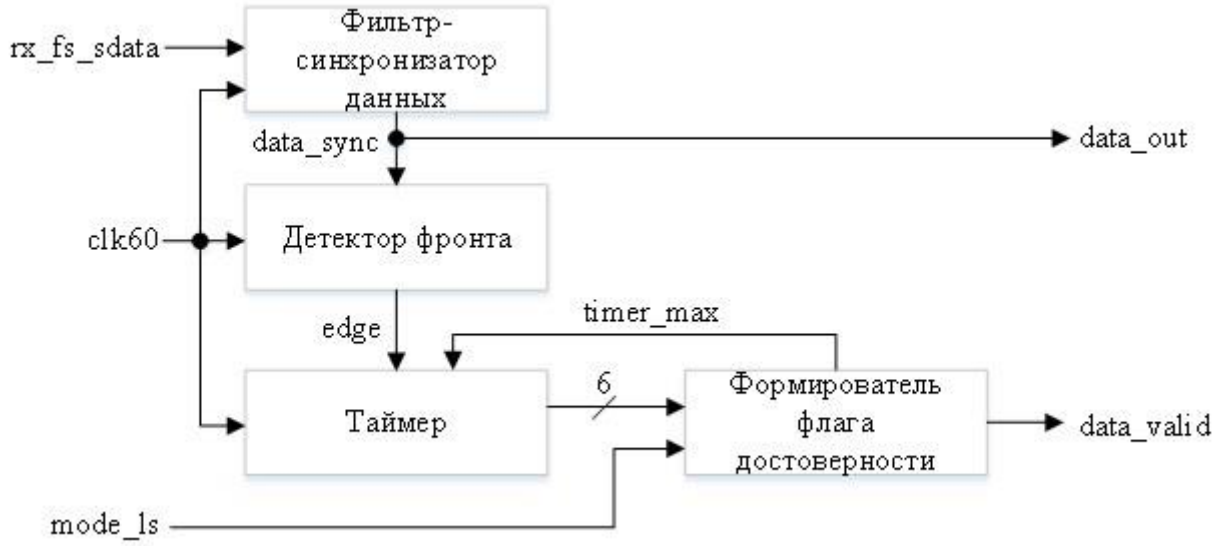


Рисунок 37.5 – Структурная схема блока восстановления данных режима FS

Блок функционирует следующим образом. После сброса и подачи тактового сигнала происходит ожидание смены значения (фронта) сигнала данных, прошедших фильтрацию. После этого в регистр таймера записывается значение «0» и по тактовому сигналу clk60 происходит инкремент значения таймера. Значение таймера обнуляется либо при достижении максимального значения, либо при обнаружении фронта на выходе фильтра-синхронизатора.

Максимальное значение таймера определяется его текущим режимом работы и должно соответствовать количеству периодов тактового сигнала, эквивалентное одному номинальному битовому интервалу для текущей скорости работы устройства. То есть, для режима FS это значение равно пяти. Таким образом, каждый раз при обнаружении перехода выполняется подстройка схемы синхронизации. Если какое-то время в потоке данных отсутствуют переходы, то таймер будет отсчитывать номинальные битовые интервалы, и блок будет продолжать формировать признаки достоверности битовых интервалов

(data_valid). Флаг достоверности выходных данных устанавливается на 1 период тактового сигнала clk60 таким образом, чтобы он совпадал с серединой битового интервала. На рисунке 37.6 приведена диаграмма работы блока в режиме Full Speed.

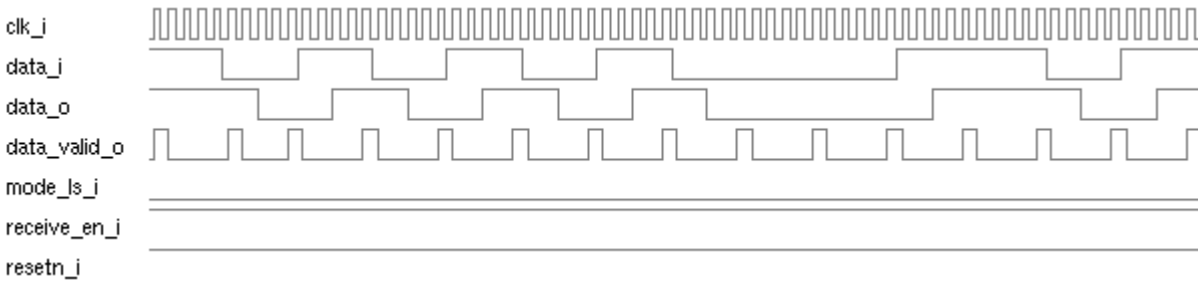


Рисунок 37.6 – Восстановление данных в режиме Full Speed

Спецификация USB определяет максимальное отклонение несущей частоты (соответственно и битрейта) передатчика от номинального на 2500ppm (0,25 %) для устройств Full Speed. Для расчетов примем максимальное расхождение локальной частоты с несущей частотой принимаемых данных как 2x2500ppm=5000ppm (0,5 %).

Максимальное количество одинаковых бит, которые могут быть переданы подряд на шине USB, составляет семь (в каждые 6 подряд идущих единиц добавляется ноль по правилам bit stuffing, после чего происходит кодирование NRZI). В течение приема семи битов будет происходить накопление ошибки, то есть, в сумме разница будет рассчитываться, как приведено в формуле (37.1):

$$\Delta = N_bits \times \Delta_ui , \quad (37.1)$$

где N_bits – это количество подряд идущих одинаковых бит и равно семи;

Δ_ui – ошибка в пределах одного битового интервала, и составляет 5000ppm.

Таким образом, общая ошибка будет составлять 35000ppm или 3,5 %, что составляет 2,91 нс.

Джиттер, определяемый спецификацией USB для приемников на скорости Full Speed равен 18,5 нс. Максимальное отклонение фронта данных от номинального рассчитаем как сумму ошибки по разности частот и максимального джиттера приемника. Таким образом, максимальное смещение фронта данных будет составлять 2,91 нс + 18,5 нс = 21,41нс.

Рассчитанное значение не превышает двух периодов локального тактового сигнала с частотой 60 МГц. Это означает, что установленный на середине битового интервала признак достоверности данных гарантированно будет совпадать по времени с достоверными данными.

Декодер NRZI

Блок декодера NRZI используется во всех скоростных режимах работы блока USB PHY и предназначен для декодирования входного потока данных по методу NRZI (Non Return to Zero Invert).

Декодирование происходит следующим методом. Если текущее и предшествующее значение бита данных в кодированном сигнале имеют одинаковое значение, это означает прием логической единицы. Если эти значения отличаются, то означает, что принят логический ноль.

На вход блока поступает кодированный сигнал данных и признак достоверности данных. Каждый такт (при условии достоверности входных данных) происходит запись текущего значения входного сигнала в регистр для последующего сравнения.

В зависимости от скоростного режима, признак достоверности данных может быть установлен раз в 5 тактов (Full Speed). На рисунке 37.7 приведена диаграмма декодирования потока в режиме Full Speed.

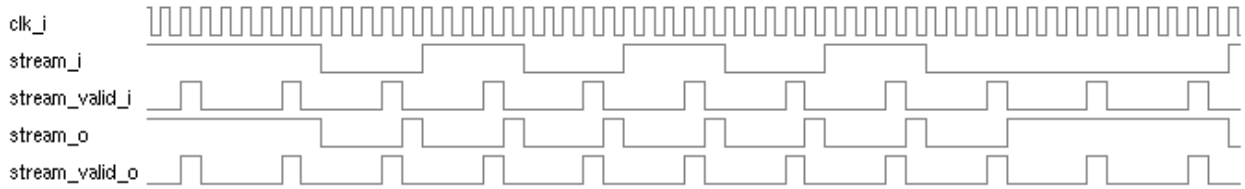


Рисунок 37.7 – Декодирование потока данных в режиме Full Speed

Опорным тактовым сигналом (clk_i) для блока является локальный тактовый сигнал с частотой 60 МГц.

Блок удаления битов из потока (bit unstuffer)

В соответствии со спецификацией USB, в поток данных могут добавляться дополнительные биты таким образом, чтобы максимальное количество подряд идущих единиц не превышало шести. Последовательность единиц всегда прерывается нулем, вне зависимости от того, какое значение следует далее. Блок удаления битов из потока выполняет обратную функцию, удаляя биты из потока декодированных данных.

На вход блока поступает поток данных, синхронизированных с локальным тактовым сигналом, частота которого составляет 60 МГц, а также признак достоверности данных. Для режима Full Speed признак достоверности устанавливается раз в пять тактов.

Реализация блока основана на счетчике битов, который инкрементируется при получении единицы и сбрасывается при получении нуля. Также стоит отметить, что в режиме Full Speed входные данные могут быть некоторое время недостоверными в момент, когда на шине USB установлено состояние SE0. Для разрешения этой ситуации в блок поступает признак получения состояния SE0, при котором счетчик битов сбрасывается, что предотвращает ложное определение ошибки добавления битов в поток.

Выходные данные имеют два сигнала достоверности. Первый повторяет сигнал достоверности на входе блока и означает достоверность данных внутри битового интервала (1/5 от времени битового интервала для FS). Второй сигнал помечает текущий бит данных как недостоверный, если произошло его удаление из потока данных. Используя уже упомянутый счетчик битов, блок способен определить ошибку добавления битов во входном потоке. При этом будет установлен флаг ошибки. На рисунке 37.8 приведена диаграмма работы блока в режиме Full Speed.

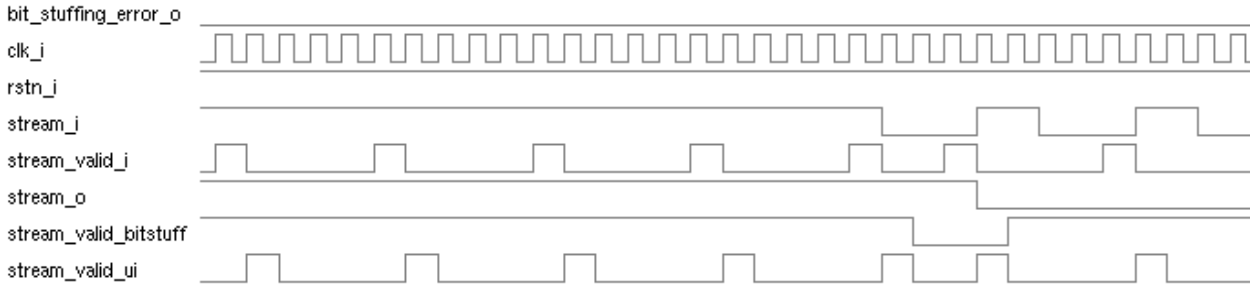


Рисунок 37.8 – Удаление битов из потока в режиме Full Speed

Как видно из рисунка, происходит удаление бита «0» из потока. Признак достоверности данных в битовом интервале (stream_valid_ui) установлен, однако, сигнал stream_valid_bitstuff указывает на то, что данный бит был удален из потока.

Блок десериализации

Блок десериализации выполняет функцию преобразования данных из последовательного вида в параллельный. Для соответствия спецификации UTMI используется преобразование принимаемых последовательных данных в 8-битный параллельный код.

Входными данными для блока десериализатора также являются признаки достоверности данных: один для обозначения битового интервала, другой для обозначения удаленных битов данных (в случае удаления битов из потока). Признак битового интервала используется при работе в режиме Full Speed, при этом признак устанавливается раз в пять тактов.

Основным элементом десериализатора является сдвиговый регистр. На его вход подаются последовательные данные на каждом такте синхросигнала (каждые пять тактов для режима FS). В качестве тактового сигнала входных данных используется сигнал с частотой 60 МГц. Также блок содержит регистр хранения промежуточных данных (hold), счетчик бит для определения границы символа и логику формирования признаков ошибок и достоверности данных. На рисунке 37.9 показана структурная схема блока десериализации.

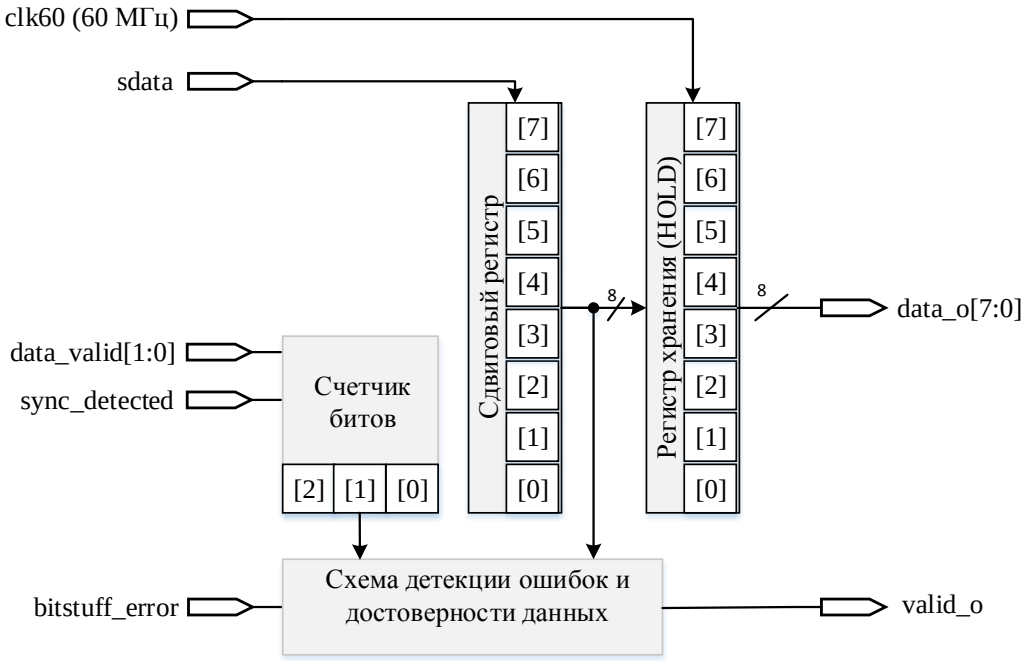


Рисунок 37.9 – Структура блока десериализации

Запись значения последовательных данных в сдвиговый регистр происходит, если установлены оба признака достоверности данных (битовый интервал и достоверность битового потока). Также одновременно инкрементируется счетчик битов. При накоплении восьми бит данных в сдвиговом регистре его содержимое копируется в регистр хранения (HOLD), а заполнение сдвигового регистра начинается заново. На рисунке 37.10 приведена диаграмма работы блока десериализации данных в режиме Full Speed.

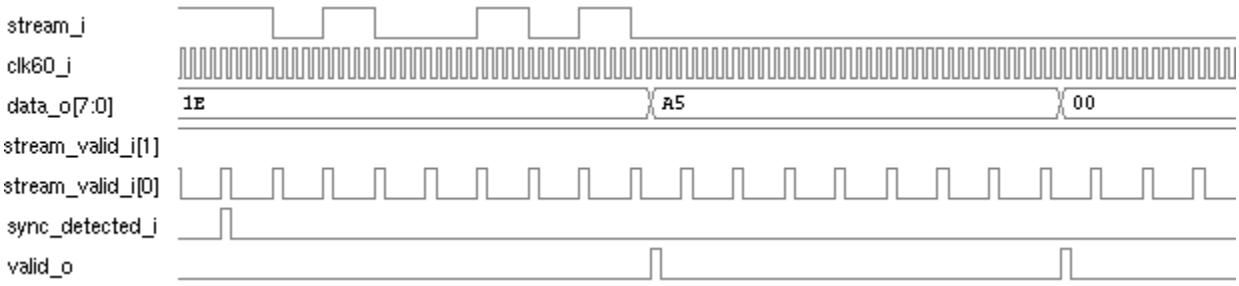


Рисунок 37.10 – Диаграмма работы блока десериализации (режим Full Speed)

Детектор синхроимпульса (SYNC)

Данный блок выполняет поиск паттерна SYNC (поле синхронизации пакета USB) в потоке принимаемых данных, что необходимо для его отделения от пользовательских данных. Полученный признак также используется другими блоками цифровой части приемника для выравнивания потока данных по байтовой границе

Блок функционирует на частоте 60 МГц. Основной элемент блока – сдвиговый регистр, в который помещаются входные последовательные данные, если они являются достоверными. В процессе сдвига содержимое регистра сравнивается с ожидаемым значением (паттерном) и, в случае совпадения, устанавливается признак обнаружения поля SYNC.

Согласно спецификации USB, паттерн SYNC представляет собой определенную последовательность состояний J и K на шине USB. Для режима FS эта последовательность представляет собой 3 пары состояние KJ с последующими двумя состояниями K (KJKJKJKK). Таким образом, в идеальной ситуации приемник получает паттерн SYNC длиной 8.

На рисунке 37.11 приведена диаграмма работы блока детектора SYNC.

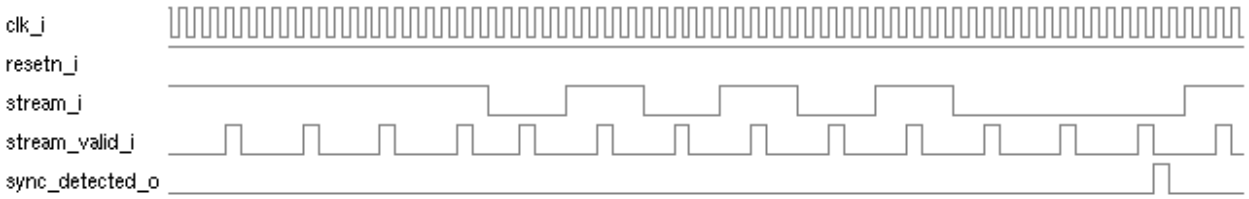


Рисунок 37.11 – Диаграмма работы блока детектора SYNC (режим Full Speed)

Следует обратить внимание, что в принимаемом пакете часть пользовательских данных может совпадать с паттерном синхронизации. Если выполнять поиск паттерна SYNC по шаблону, используя лишь сдвиговый регистр, то это может привести к множественному обнаружению паттерна и повреждению данных (из-за сброса блока десериализации при каждом новом SYNC).

Чтобы избежать этого, необходимо блокировать детектор SYNC после обнаружения первого паттерна, а разблокировку делать только после окончания пакета. В качестве признака разблокировки используется либо переход шины USB в состояние IDLE, либо событие включения приемника после его блокировки передатчиком. Структурная схема блока детектора SYNC приведена на рисунке 37.12.

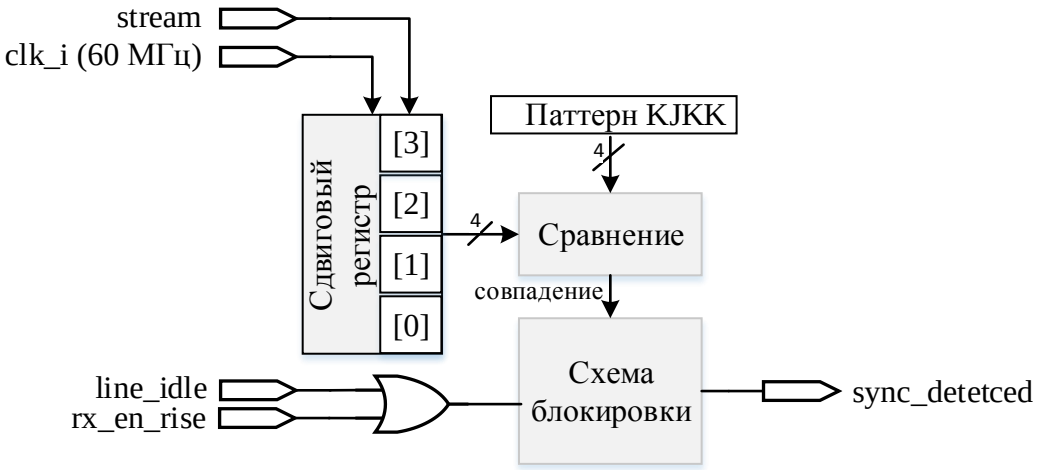


Рисунок 37.12 – Структурная схема блока детектора SYNC

Детектор конца пакета (EOP)

Блок детектора EOP (End Of Packet, признак окончания пакета) выполняет функцию детектирования конца пакета USB. Эта функция используется приемником для формирования соответствующих управляющих сигналов и данных в интерфейсе UTMI.

Согласно спецификации USB, достаточным условием для определения конца пакета в режиме FS является наличие состояния SE0 минимальной длительностью 1 битовый интервал с последующим состоянием J на шине USB. Соответственно, блок детектора EOP использует выходы дифференциального и несимметричного (single-ended) приемников. Диаграмма работы блока обнаружения поля EOP приведена на рисунке 37.13.

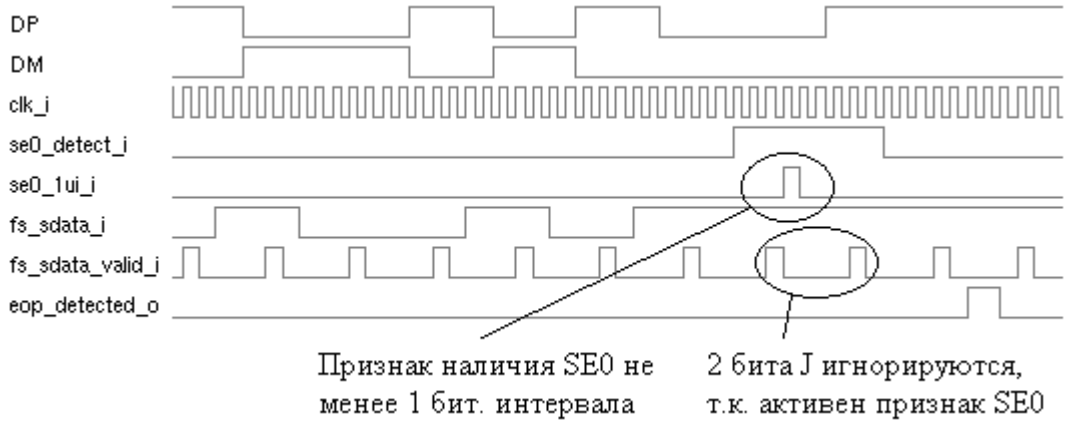


Рисунок 37.13 – Диаграмма работы блока детектора EOP (режим Full Speed)

Для определения признака EOP используется счетчики количества битовых интервалов с состоянием SE0 и состоянием J. Достаточное условие определения EOP непрерывно проверяется, и при его выполнении устанавливается признак обнаружения конца пакета. Также необходимо учесть, что выход дифференциального приемника FS во время приема состояния SE0 использовать нельзя, то есть это состояние не должно интерпретироваться как J.

Работа блока в режиме Full Speed отличается лишь длительностью битового интервала, которая учитывается определении конца пакета. Так, для режима FS требуется минимум 5 тактов синхросигнала с частотой 60 МГц для достижения длительности в 1 битовый интервал.

Выход eop_detected_o будет использован конечным автоматом приемника для установки признака конца пакета на интерфейсе UTMI.

Фильтр состояния SE0

Фильтр состояния SE0 (Single-Ended 0) предназначен для отделения кратковременных переключений признака SE0 при приеме данных. Признак SE0 формируется на основе выхода несимметричных (single-ended) приемников комбинационной логикой. Данный блок реализован в соответствии параметрами, приведенными в спецификации USB, которые определяет максимальную длительность состояния SE0 между переключениями J и K. Результатом работы блока является синхронизированной сигнал, представляющий наличие состояние SE0 на шине USB, исключающий при этом переключения длительностью меньше, чем указано в спецификации USB для режима Full Speed. Минимальная длительность составляет 14 нс для режима Full Speed.

Принцип работы блока схож с блоком детектора состояния линии (ДСЛ) и имеет те же временные параметры. Основные элементы блока – синхронизатор и фильтр, основанный на таймере. Изначально входной признак SE0, полученный комбинационно, проходит синхронизацию. Далее, если состояние SE0 активно, таймер фильтра запускается,

при неактивном состоянии SE0 – сбрасывается. При достижении таймером минимально допустимого значения происходит установка признака SE0 на выходе блока. Диаграмма работы блока фильтрации состояние SE0 приведена на рисунке 37.14.

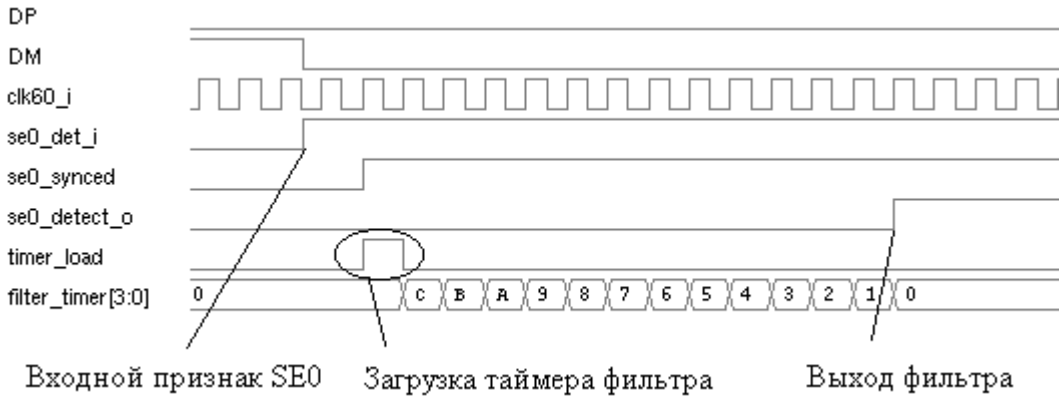


Рисунок 37.14 – Диаграмма работы фильтра состояния SE0

Таким образом, если длительность состояния SE0 меньше, чем минимально допустимая, на выходе блока признак SE0 останется без изменений. При длительности состояния SE0 больше пороговой фильтр пропускает его, однако это добавляет некоторую несущественную задержку в его распространение по схеме от непосредственно приемника до использующих признак блоков.

Блок управления приемной частью

Блок управления приемной частью выполняет функции управления интерфейсом UTMI в части приема данных. Блок использует управляющие сигналы и шины данных от других модулей блока USB PHY. Управляющие сигналы проходят промежуточную обработку конечным автоматом (далее – автомат). Реализация блока соответствует спецификации UTMI (пункт 5.8 «Receive State Machine»). Диаграмма состояний конечного автомата приведена на рисунке 37.15.

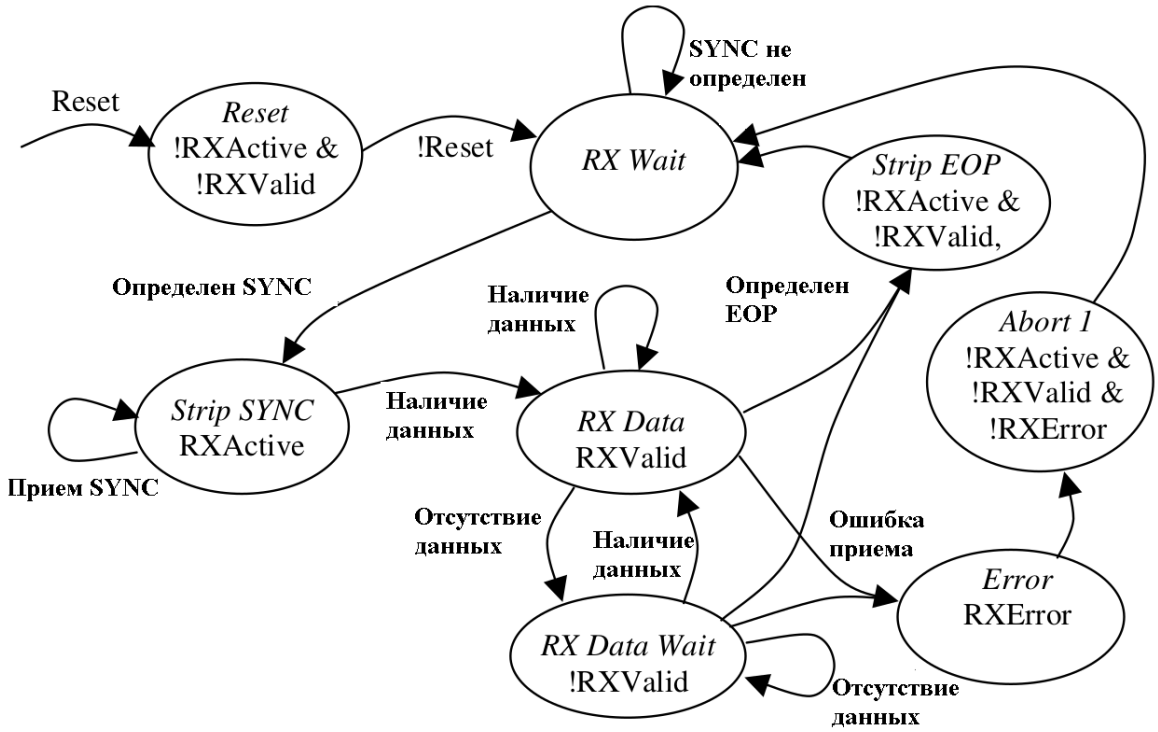


Рисунок 37.15 – Диаграмма состояний блока управления приемной частью

При возникновении аппаратного сброса схема переходит в состояние «Reset». После окончания сигнала сброса автомат переходит в состояние ожидания получения синхросигнала приемником (SYNC), который является признаком начала пакета. Данное состояние называется «RX Wait» и оно является состоянием, в которое автомат переходит при возникновении ошибок приема и при завершении пакета.

Признак получения SYNC-паттерна переводит автомат в состояние «Strip SYNC», в котором устанавливается сигнал RXActive, что означает начало приема пакета. Данные, поступающие от блока десериализации, в этом состоянии отклоняются, а на шину данных интерфейса UTMI выводится значение «0». После окончания приема SYNC-паттерна происходит смена состояния на «RX Data».

В состоянии «RX Data» устанавливается логическая единица на выходе RxValid, что помечает данные, выводимые на интерфейс UTMI, как достоверные. Шина данных блока десериализации копируется в выходную шину данных интерфейса UTMI (Data_Out[7:0]) на каждом положительном фронте тактового сигнала CLK (60 МГц). Автомат переключается между состояниями «RX Data» и «RX Data Wait», в зависимости от наличия актуальных данных в блоке десериализации.

В состоянии «RX Data Wait» данные не выводятся на интерфейс UTMI, а признак RX_Valid становится неактивным. Вход в данное состояние происходит при накоплении восьми удаленных бит из потока (bit-unstuff), при этом блок десериализаци снимает признак достоверности данных (data_valid) на 1 период тактового сигнала с частотой 60 МГц.

При получении признака конца пакета (EOP, end of packet) происходит переход в состояние «Strip EOP». Данный признак проверяется только в состояниях приема данных («RX Data» и «RX Data Wait»). В этом состоянии данные не выводятся на интерфейс UTMI (заполняются нулями), а сигналы RX_Active и RX_Valid переходят в неактивное состояние. В режиме Full Speed функцию определения наличия EOP берет на себя блок детектора EOP. После деактивации признака окончания пакета автомат вновь переходит в состояние ожидания паттерна SYNC («RX Wait»).

Находясь в состоянии «RX Data» или «RX Data Wait», автомат состояния отслеживает сигнал об ошибке, поступающий на вход блока. При получении этого сигнала происходит установка сигнала RxError в состояние логической единицы и выполняется переход в состояние «Abort 1». В состоянии «Abort 1» на выходах RxActive, RxValid и RxError устанавливается логический ноль, затем автомат состояния переходит в состояние ожидания SYNC («RX Wait»).

Блок управления передающей частью

Блок управления передающей частью предназначен для выработки управляющих сигналов для блока сериализации. Блок реализован в соответствии с спецификации UTMI (пункт 5.13 «Transmit State Machine») в виде конечного автомата, диаграмма состояний которого приведена на рисунке 37.16.

При аппаратном сбросе конечный автомат переходит в состояние «Reset», при этом выходной сигнал готовности TXReady переходит в неактивное состояние. После окончания сброса выполняется переход в состояние «TX Wait», в котором ожидается подача данных и сигнала TXValid от интерфейса UTMI. Выход из данного состояния осуществляется только после получения сигнала TXValid.

При получении TXValid передатчик должен сгенерировать синхроимпульс (SYNC) перед отправкой пользовательских данных, если данная функция разрешена установленным кодом на шине OpMode[1:0]. Для этого выполняется переход в состояние «Send SYNC», в котором формируются необходимые управляющие сигналы для блока сериализации.

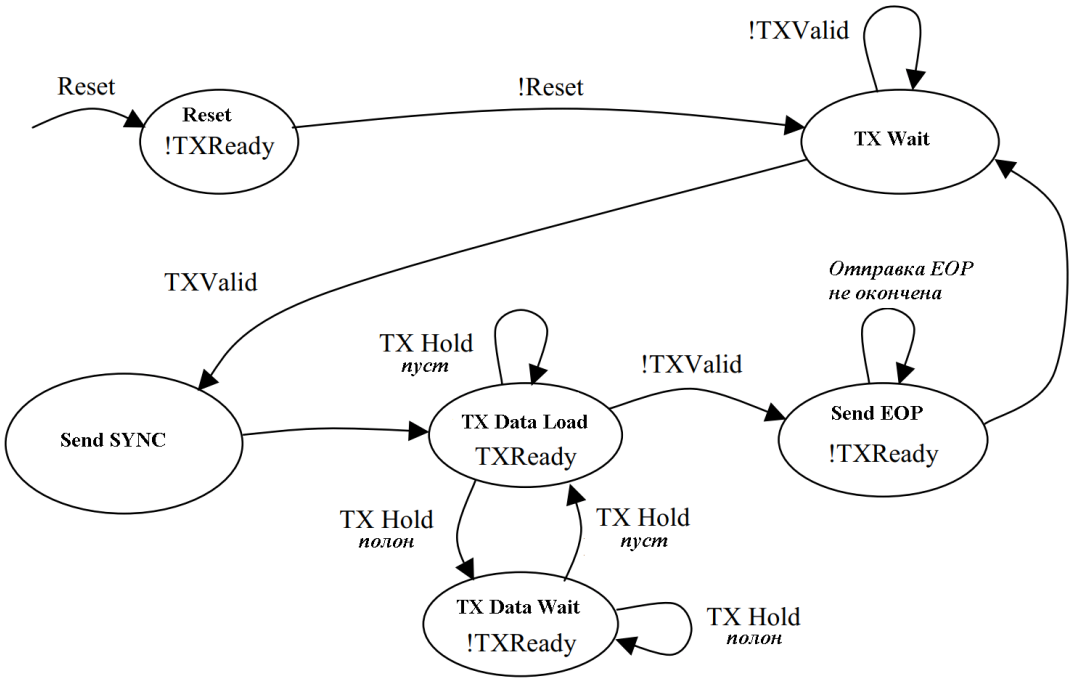


Рисунок 37.16 – Диаграмма состояний конечного автомата блока управления передающей частью

После отправки SYNC-паттерна автомат переходит в состояние «TX Data Load», в котором разрешается загрузка данных в блок сериализации. При временной неготовности блока сериализации к поступлению новых данных конечный автомат переходит в состояние «TX Data Wait». Выход из данного состояния происходит при опустошении буфера блока сериализации (регистра HOLD).

При сбросе сигнала TXValid конечный автомат переходит в состояние «Send EOP», в котором формируется управляющий сигнал для добавления паттерна конца пакета к передаваемым данным. После завершения передачи пакета выполняется переход в состояние по умолчанию («TX Wait»).

Блок добавления битов в поток (bit stuffer)

Блок добавления битов в поток предназначен для предотвращения передачи большого количества подряд идущих единиц. Нулевой бит вставляется после последовательности шести единичных битов в потоке до осуществления NRZI кодирования, что соответствует спецификации USB. В соответствии с правилами кодирования NRZI, добавление нуля в последовательность данных приведет к смене полярности выходного сигнала. Достаточное количество переключений (смен полярности) в потоке данных обеспечивает надежное восстановление данных и тактового сигнала при приеме данных. Блок реализован в соответствии со спецификацией UTMI (пункт 5.11).

Добавление битов начинается с передачи паттерна SYNC. Единичный бит, завершающий SYNC-последовательность, рассматривается как первый бит последовательности единичных битов. Добавление единичных битов происходит при передаче любых данных. Блок реализован с использованием счетчика, который сбрасывается при получении нулевого бита и инкрементируется при получении каждого единичного бита.

Блок вырабатывает управляющий сигнал для схемы сериализации, который необходим для учета количества добавленных в поток битов. При добавлении каждого восьмого бита в поток сериализатор должен приостановить загрузку новых параллельных данных на один период тактового сигнала, чтобы избежать перезаписи данных. Временная диаграмма сигналов блока приведена на рисунке 37.17.

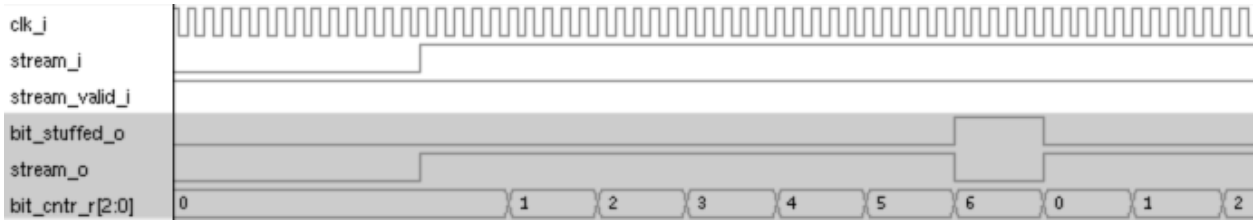


Рисунок 37.17 – Диаграмма сигналов блока добавления битов в поток (режим FS)

Кодер NRZI

Кодер NRZI представляет собой стандартную схему, выполненную в соответствии с требованиями спецификации USB. В кодированном потоке единичный бит представляется отсутствием переключения, а нулевой бит представляется переключением полярности сигнала на противоположную.

Входные данные имеют признак достоверности (valid), который в режиме FS устанавливается раз в пять периодов опорного тактового сигнала 60 МГц.

Блок кодера NRZI является последней стадией обработки данных передатчика, его выходной поток данных передается на драйверы шины USB. Временная диаграмма сигналов блока приведена на рисунке 37.18.

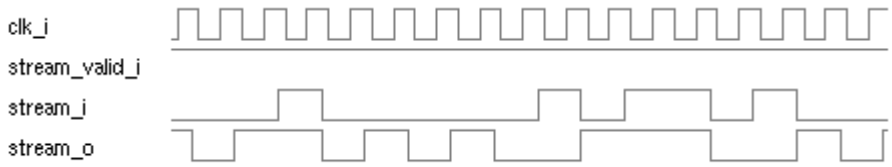


Рисунок 37.18 – Временная диаграмма сигналов блока NRZI-кодера

Блок управления приемопередатчиком

Блок управления приемопередатчиком контролирует основные функции, связанные управлением аналоговой частью блока USB PHY:

– управление подтягивающими резисторами с сопротивлением 1,5 кОм (Rpu), включая режим уменьшенного сопротивления;

– управление подтягивающими резисторами с номинальным сопротивлением 15 кОм (Rpd);

– включение и отключение аналоговых драйверов и приемников;

– установка калибровочных кодов для схем ODT и BANDGAP;

– установка запроса на запуск и отключение внешнего блока ФАПЧ;

– вход в состояние пониженного энергопотребления SUSPEND и выход из него;

– установка отладочных и конфигурационных параметров для цифровой и аналоговой части.

Основными элементами блока управления являются: конечный автомат (далее – FSM), схема управления подтягивающими резисторами, а также схема управления аналоговыми блоками (такими как приемники, драйверы, детектор уровня сигнала и т.д.). На рисунке 37.19 показана структурная схема блока управления.

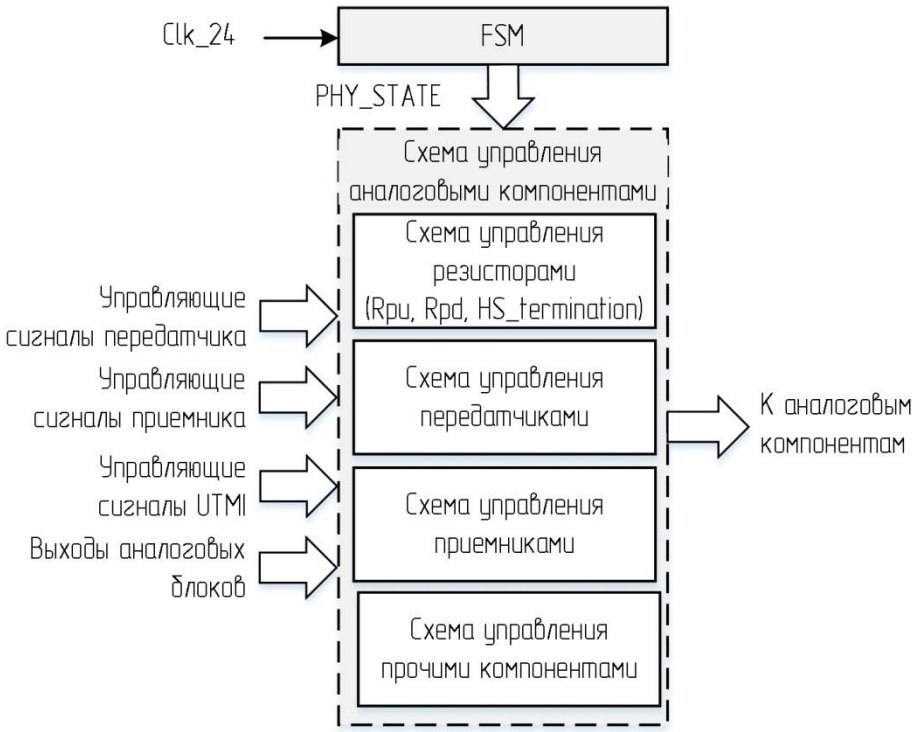


Рисунок 37.19 – Структурная схема блока управления приемопередатчиком

Поведение всего блока зависит от текущего состояния FSM. В таблице 37.2 приведено описание состояний конечного автомата.

Таблица 37.2 – Возможные состояния конечного автомата блока управления

Наименование состояния	Описание
1	2
HW_RESET	Состояние, в которое FSM переходит при аппаратном сбросе. В данном состоянии отключен блок ФАПЧ, на интерфейс UTMI не подается тактовый сигнал. Включен только блок детектора состояния VBUS.
CALIBRATION_BG	Состояние калибровки BANDGAP. Блоку управления схемой BANDGAP передается сигнал разрешения калибровки, в ответ ожидается признак завершения калибровки. Состояние не используется, если включено использование внешней схемы BANDGAP
CALIBRATION_ODT	Состояние калибровки ODT. Блоку управления схемой ODT передается сигнал разрешения калибровки, в ответ ожидается признак завершения калибровки. Состояние не используется, если включено использование внешней схемы ODT
WAIT_PLL_LOCK	Состояние ожидания перехода блока ФАПЧ в стабильный режим. Анализируется состояние сигнала PLL_LOCK. Внутренние тактовые сигналы остановлены, за исключением опорного тактового сигнала частотой 24 МГц.
UTMI_CLK_STOP	Состояние ожидания завершения переходных процессов аналоговых компонентов блока USB PHY. Внутренние тактовые сигналы функционируют, однако, на интерфейс UTMI подача тактового сигнала отключена. Длительность нахождения в данном состоянии составляет около 30 мкс.
UTMI_NORMAL	Состояние нормальной работы блока. Включена подача всех тактовых сигналов. Управление блоком полностью передается интерфейсу UTMI.
UTMI_SUSPENDED	Состояние пониженного энергопотребления. Часть аналоговых блоков отключается либо переводятся в режим пониженного энергопотребления. Подача тактового сигнала интерфейса UTMI приостановлена. Установлен запрос на отключение внешнего блока ФАПЧ.

На рисунке 37.20 приведена схема возможных переходов между состояниями FSM.

Конечный автомат использует тактовый сигнал с частотой 24 МГц, поступающий от внутреннего блока ФАПЧ непрерывно. Данный тактовый сигнал не зависит от того, включен или выключен блок ФАПЧ, что позволяет постоянно поддерживать работоспособное состояние блока.

Состояния CALIBRATION_BG и CALIBRATION_ODT являются опциональными, переход с них осуществляется только при наличии встроенных схем калибровки BANDGAP и ODT.

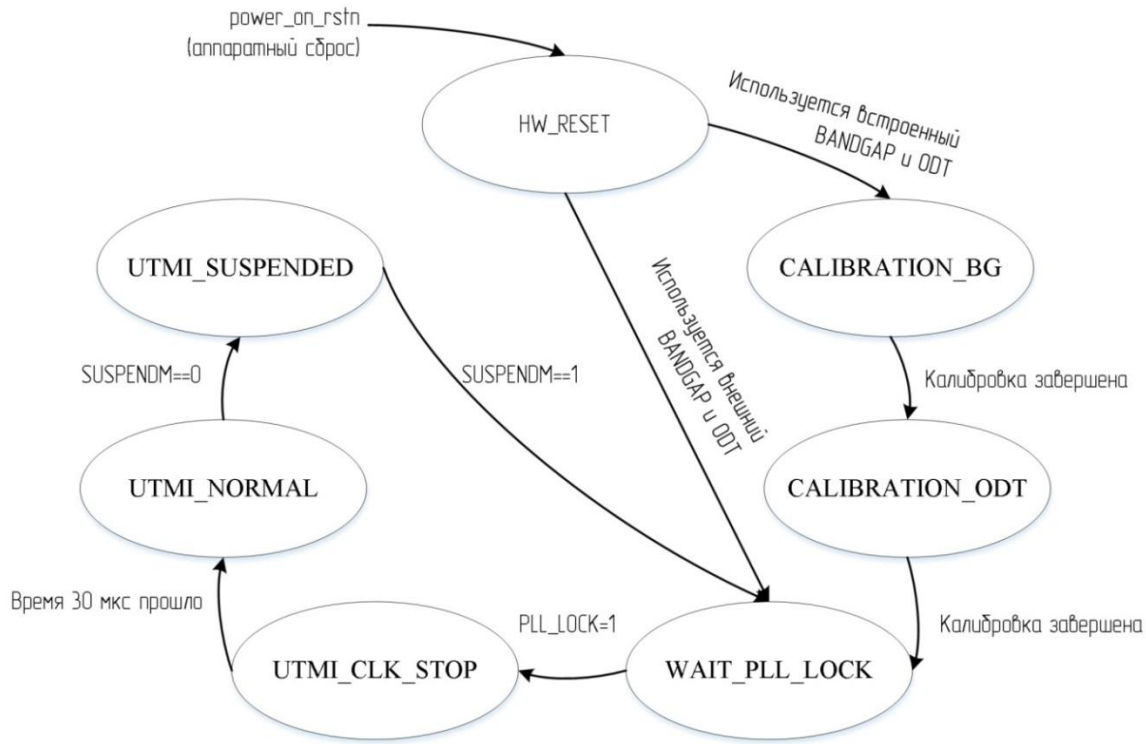


Рисунок 37.20 – Схема возможных переходов между состояниями FSM

Детектор разрыва соединения

Данный блок выполняет функцию определения разрыва соединения с устройством, подключенным к шине USB. Блок используется, если блок USB PHY работает в режиме нисходящего порта.

Определение разрыва соединения выполняется при совместной работе двух компонентов – аналогового блока, реализованного в соответствии с требованиями спецификации USB и цифрового блока детектора разрыва соединения. В данном разделе описана работа цифрового блока.

Реализация блока является также требованием спецификации UTMI. Целью определения разрыва соединения является формирование сигнала интерфейса UTMI HostDisconnect. Данный сигнал имеет высокий активный уровень и устанавливается каждый раз при обнаружении разрыва соединения с устройством.

В режиме FS критерием разрыва соединения является наличие состояния SE0 на шине USB в течение более 2,5 мкс. При любом другом состоянии длительностью более 2,5 мкс соединение считается установленным. Работа блока в данном режиме реализована с помощью анализа шины LineState[1:0], которая отражает текущее состояние шины USB и входит в интерфейс UTMI. Время 2,5 мкс отсчитывается с помощью таймера, работающего на частоте 24 МГц.

37.4 Тактирование блока

Источник тактирования блока USB PHY выбирается битовым полем CLKSEL регистра USBCFGn блока RCU. Частота тактового сигнала USBCLK должна соответствовать 60 МГц. Схема выбора синхросигнала блока USB показана на рисунке 37.48.

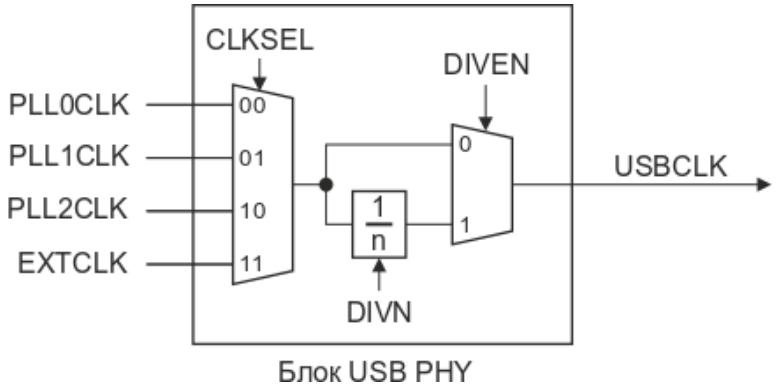


Рисунок 37.21 – Схема селектора синхросигнала блока USB PHY

38 Контроллер интерфейса CAN 2.0B

Микроконтроллер содержит контроллер интерфейса CAN с четырьмя узлами, 256 объектами сообщений и 16 линиями прерываний.

Управляющие регистры контроллера CAN 2.0B приведены в приложении А.31.

38.1 Протокол CAN

Последовательный интерфейс CAN (Controller Area Network) – интерфейс связи, эффективно поддерживающий распределенное управление в масштабе реального времени с высокой помехозащищенностью. Протокол связи определен в спецификации CAN 2.0B.

Протокол CAN оптимизирован для систем, в которых должно передаваться относительно небольшое количество информации (по сравнению с Ethernet или USB) к любому или всем узлам сети. Множественный доступ с опросом состояния шины позволяет каждому узлу получить доступ к шине с учетом приоритетов. Неадресная структура сообщений позволяет организовать многоабонентскую доставку данных с сокращением трафика шины. Быстрая устойчивая передача информации с системой контроля ошибок позволяет отключать неисправные узлы от шины, что гарантирует доставку критических по времени сообщений.

Область применения протокола CAN: от высокоскоростных сетей связи до применения в автомобиле. Высокая скорость передачи данных (до 1 Мбит/с), хорошая помехозащищенность протокола, защита от неисправности узлов – делают шину CAN подходящей для индустриальных приложений управления на основе протокола связи DeviceNet или подобных ему.

CAN имеет асинхронную последовательную структуру шины с одним логическим сегментом сети. CAN сеть может состоять из двух или более узлов с возможностью подключения/отключения узлов от шины без перенастройки других устройств, см. рисунок 38.1.

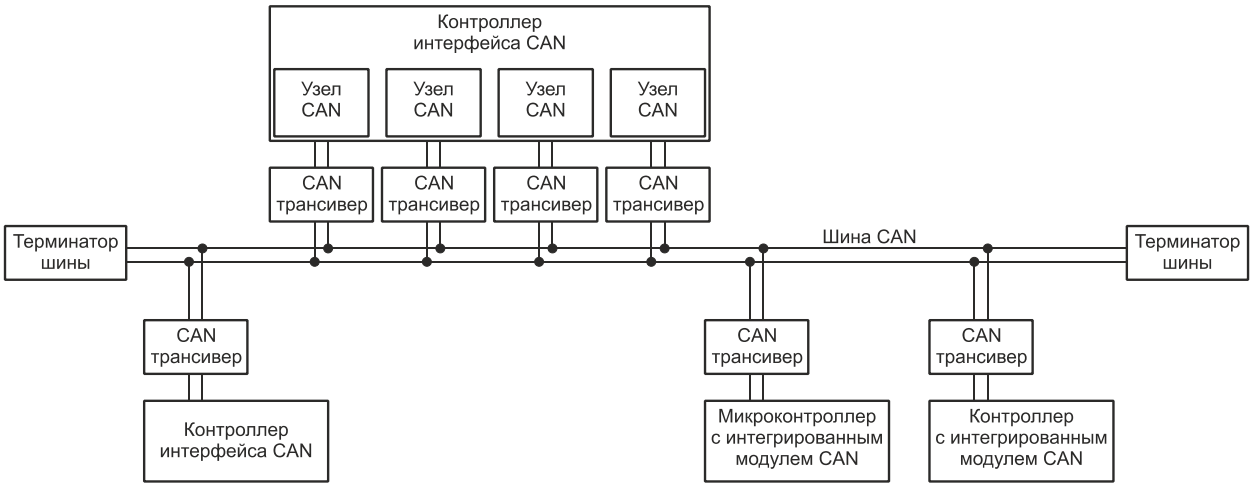


Рисунок 38.1 – Общая структура CAN сети

Логика шины работает по механизму «монтажного И», в котором рецессивный бит соответствует логической единице, а доминантный – логическому нулю. Пока ни один узел не формирует доминантный бит, шина находится в рецессивном состоянии. Появление на шине доминантного бита (выставленного одним или несколькими узлами) создает доминантное состояние шины. Отсюда следует, что при выборе среды передачи данных необходимо точно определить, какое состояние будет доминантным, а какое – рецессивным. Одним из наиболее распространенных и дешевых вариантов линии связи является витая пара. Линии шины тогда называются CANH и CANL и могут быть

подключены непосредственно к устройствам. Не существует никакого дополнительного стандарта на среду передачи данных.

При использовании в качестве линии связи витой пары с нагрузочными резисторами на концах можно получить максимальную скорость передачи данных 1 Мбит/с при длине линии до 40 м. Для линий связи протяженностью более 40 м необходимо снизить скорость передачи данных (для линии 1000 м скорость шины должна быть не более 40 Кбит/с). Из-за дифференциального характера линии связи, шина CAN малочувствительна к электромагнитным помехам. Экранирование шины значительно снизит воздействие внешнего электромагнитного поля, что особенно важно для высокоскоростных режимов работы.

Двоичная информация кодируется. Доминантным является низкий уровень, рецессивным – высокий уровень. Для гарантированной синхронизации потока данных всеми узлами шины используется принцип «битстаффинга». Это означает, что при последовательной передаче пяти бит одинаковой полярности передатчик вставляет один дополнительный бит противоположной полярности перед передачей остальных битов. Приемник также проверяет полярность и удаляет дополнительные биты.

В CAN протоколе при передаче данных приемные узлы не адресуются, а указывается идентификатор передатчика. С помощью идентификатора указывается содержание сообщения (например, применительно автомобиля – обороты, температура двигателя и т. д.) и степень приоритета сообщения. Более высокий приоритет у идентификатора, имеющего меньшее бинарное значение.

При коллективном доступе к шине используется неразрушающий арбитраж с опросом состояния шины. Перед началом передачи данных узел проверяет состояние шины (отсутствие активности на шине). При начале передачи сообщения узел становится управляющим шины, все остальные узлы переходят в режим приема. После приема сообщения (подтвержденного каждым узлом) каждый узел проверяет идентификатор в сообщении и сохраняет сообщение, если это требуется. В противном случае, сообщение сбрасывается. Если два или более узлов начинают передачу данных одновременно, поразрядный арбитраж позволяет избежать конфликта на шине. Каждый узел выдает на шину свой идентификатор (старший бит формируется первым) и контролирует ее состояние. Если узел посылает «1», а считывает «0», значит арбитраж потерян, и узел переключается в режим приема. Это происходит тогда, когда идентификатор конкурирующего узла имеет меньшее бинарное значение. Таким образом, узел с высоким приоритетом выигрывает арбитраж без необходимости повторять сообщение. Все остальные узлы будут пытаться передать сообщение после освобождения шины. Данный механизм не позволяет передавать сообщения одновременно разными узлами. Для этого программно должно быть обеспечено, чтобы узлы, передающие данные, не имели одинаковых идентификаторов. Оригинальная спецификация в версии CAN 2.0B (так называемая расширенная версия CAN 2.0) определяет возможность идентификатора иметь длину 11 или 29 бит.

Протокол CAN предусматривает следующие типы сообщений:

- сообщение данных (стандартное и расширенное);
- удаленный запрос данных;
- сообщение об ошибке;
- сообщение о перезагрузке.

Стандартное сообщение данных

Стандартное сообщение данных формируется, когда узел желает передать данные. Формат сообщения показан на рисунке 38.2.

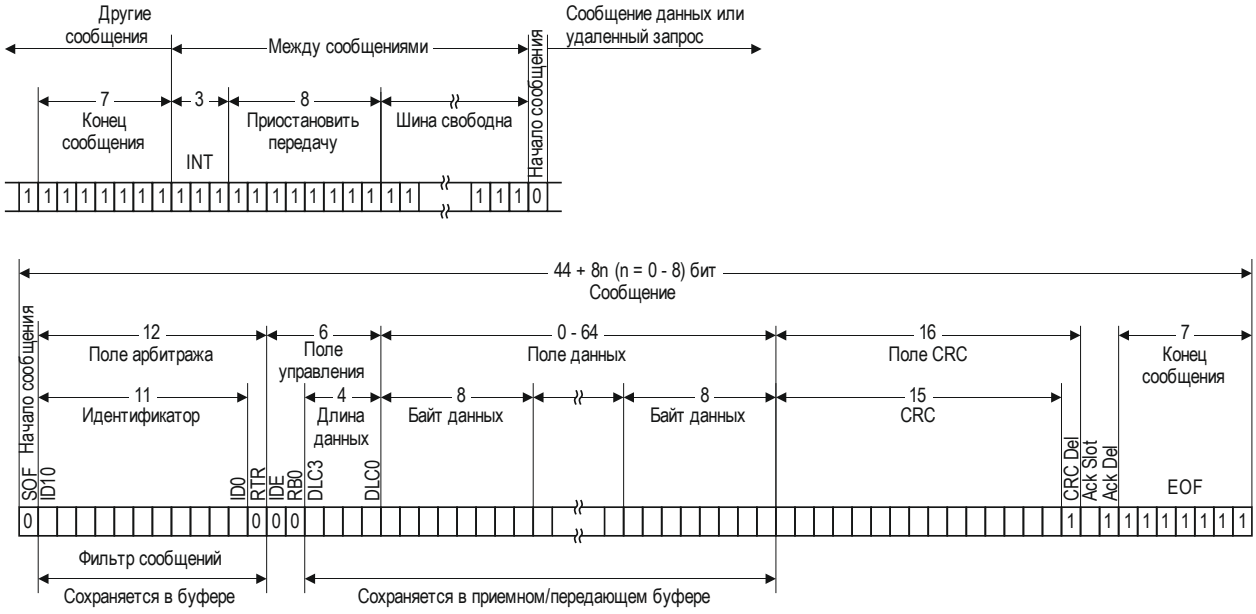


Рисунок 38.2 – Стандартное сообщение данных

Стандартное сообщение имеет в своем составе:

- бит SOF – доминантный («0») бит начала сообщения для жесткой синхронизации всех узлов;

- поле арбитража (12 бит), включающее поле ID идентификатора (11 бит), и бит RTR передачи по удаленному запросу (RTR = «0» соответствует сообщению данных, RTR = «1» соответствует удаленному запросу);

- поле управления (6 бит), включающее бит IDE, – указатель расширенного идентификатора (IDE = «0» соответствует стандартному идентификатору, IDE = «1» соответствует расширенному идентификатору), бит RB0 – резервный доминантный бит и поле DLC – числа байт данных (4 бита), которое указывает, сколько байт данных содержится в сообщении (значения от 0 до 8 используются непосредственно, значения превышающие 8 соответствуют 8 байтам данных);

- поле данных (от 0 до 64 бит), содержащее целое число байт данных;

- поле контрольной суммы CRC (16 бит), включающее поле CRC (15 бит), используемое для обнаружения возможных ошибок передачи данных, и рецессивный («1») бит разделителя CRC (CRCDel);

- поле подтверждения (2 бита), включающее бит подтверждения передачи – ACK Slot (передающий узел выдает рецессивный бит, а любой узел, который принял сообщение без ошибок, заменяет его сформированным доминантным битом), и рецессивный бит разделителя подтверждения – ACKDel;

- поле EOF конца сообщения (7 бит).

Между передачами двух любых сообщений шина должна оставаться в рецессивном состоянии как минимум в течение времени появления 3 бит (поле INT простоя). Если после появления трех рецессивных бит (поле INT) ни один узел не начал передачу, шина переходит в состояние бездействия IDLE и находится в рецессивном состоянии до появления доминантного бита сообщения.

Расширенное сообщение данных

Расширенное сообщение данных формируется, когда узел желает передать данные. Формат сообщения показан на рисунке 38.3.

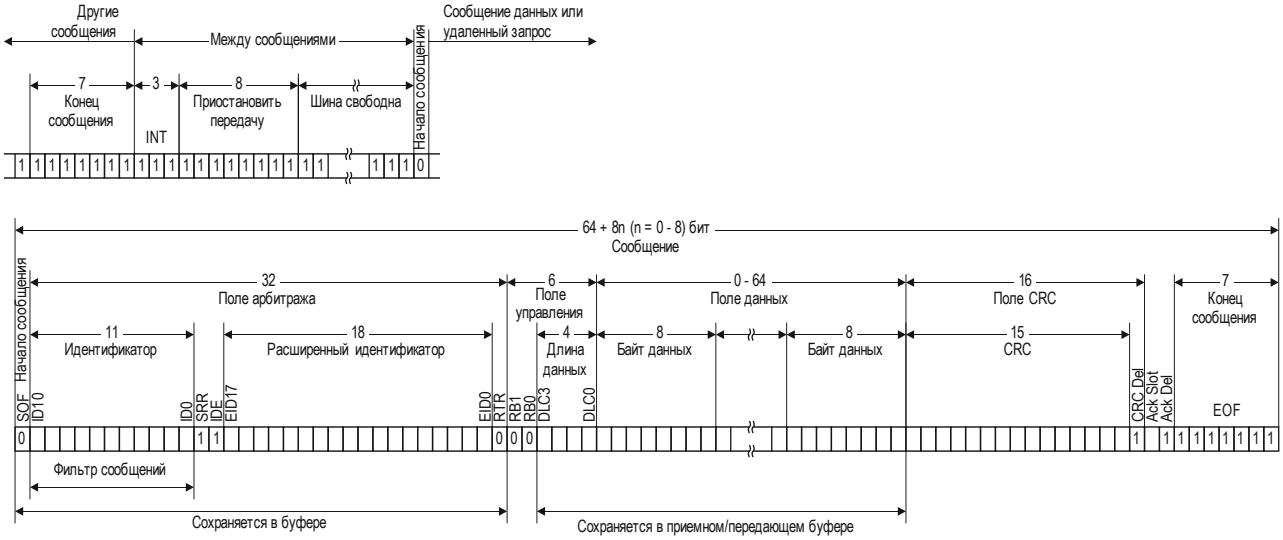


Рисунок 38.3 – Расширенное сообщение данных

Расширенное сообщение имеет в своем составе:

- бит SOF – доминантный («0») бит начала сообщения для жесткой синхронизации всех узлов;

- поле арбитража (38 бит), включающее поле стандартного идентификатора (11 бит), бит SRR заменитель удаленного запроса, бит IDE указатель расширенного идентификатора (рецессивный, что соответствует расширенному идентификатору) и поле расширенного идентификатора (18 бит);

- бит RTR передачи по удаленному запросу (RTR = «0» соответствует сообщению данных, RTR = «1» соответствует удаленному запросу);

- поле управления (6 бит), включающее бит RB0 – резервный доминантный бит, бит RB1 – резервный доминантный бит и поле DLC числа байт данных (4 бита), которое указывает, сколько байт данных содержится в сообщении (значения от 0 до 8 используются непосредственно, значения, превышающие 8, соответствуют 8 байтам данных);

- поле данных (от 0 до 64 бит), содержащее целое число байт данных;

- поле контрольной суммы CRC (16 бит), включающее поле CRC (15 бит), используемое для обнаружения возможных ошибок передачи данных и рецессивный («1») бит разделителя CRC (CRCDel);

- поле подтверждения (2 бита), включающее бит подтверждения передачи – ACK Slot (передающий узел выдает рецессивный бит, а любой узел, который принял сообщение без ошибок, заменяет его сформированным доминантным битом) и рецессивный бит разделитель подтверждения – ACKDel;

- поле EOF конца сообщения (7 бит).

Удаленный запрос данных

Формируется, когда узлу требуются данные другого узла. Узел назначения посылает удаленный запрос с идентификатором источника. Соответствующий узел источника (распознавший свой идентификатор) посылает стандартное или расширенное сообщение в ответ на запрос.

Удаленный запрос данных существует в стандартном и расширенном вариантах. На рисунке 38.4 представлен вариант удаленного запроса со стандартным идентификатором.

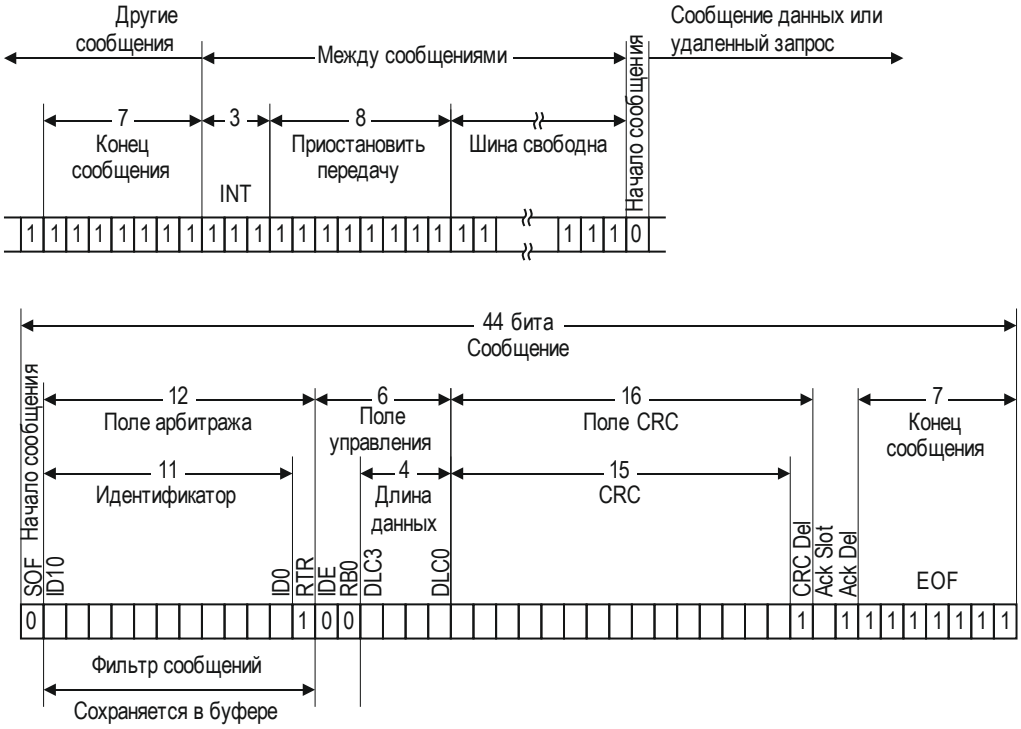


Рисунок 38.4 – Удаленный запрос данных (стандартный формат)

Имеются только два отличия содержимого удаленного запроса от сообщения данных:

- бит RTR в удаленном запросе передается в рецессивном состоянии;
- поле данных отсутствует (в сообщении не передается никаких данных, значение в поле DLC любое).

В самом маловероятном случае, когда одновременно формируется удаленный запрос, и устройство пытается передать данные с одинаковыми идентификаторами, арбитраж будет выигран устройством, передающим данные, из-за доминантного состояния бита RTR. Узел, который посылал запрос, получит данные без задержки.

Сообщение об ошибке

Сообщение об ошибке формируется любым узлом, который обнаруживает ошибку на шине. Формат сообщения показан на рисунке 38.5.

Сообщение об ошибке состоит из поля флага ошибки и следующего за ним поля разделителя ошибки. Возможны два типа поля флага ошибки, в зависимости от состояния узла (активен к ошибкам / пассивен к ошибкам), обнаружившего ее.

Если ошибку обнаружил активный узел (как в примере на рисунке 38.5), тогда он прерывает передачу текущего сообщения, формируя активный флаг ошибки. Активный флаг ошибки состоит из шести последовательных доминантных бит, которые нарушают правила «битстаффинга» (правила наполнения и передачи бит на шине). Остальные узлы также обнаруживают ошибку и начинают формировать сообщение об ошибке. Таким образом, поле флага ошибки может содержать от 6 до 12 доминантных бит (сформированных одним или более узлами). Поле флага ошибки дополняется разделителем ошибки, состоящим из восьми рецессивных бит и позволяющим возобновить обмен данными по шине после обнаружения ошибки. После перехода шины в нормальное состояние узел, чье сообщение было прервано, вновь попробует передать данное сообщение.

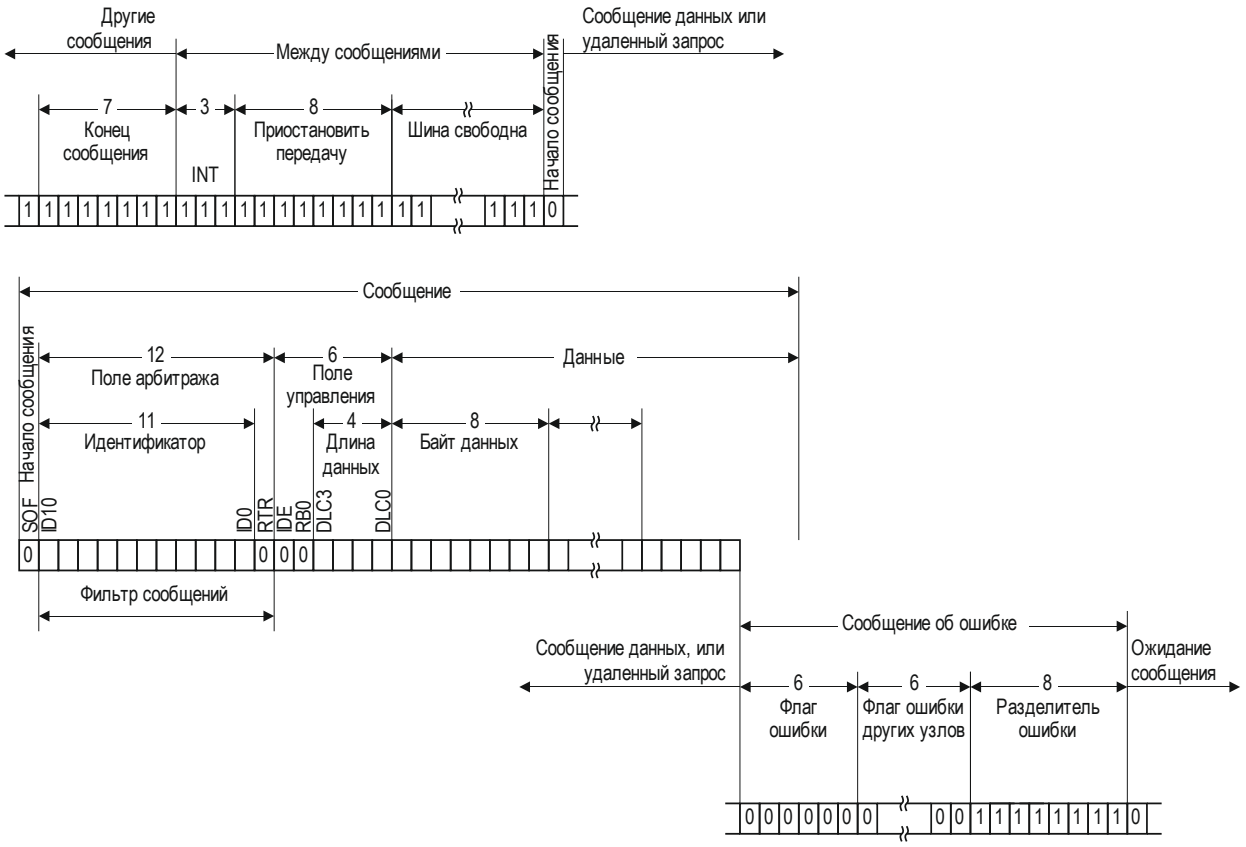


Рисунок 38.5 – Сообщение об ошибке

Если ошибку обнаружил пассивный узел, тогда он формирует пассивный флаг ошибки, состоящий из шести последовательных рецессивных бит, и затем разделитель ошибки. Таким образом, сообщение об ошибке от пассивного узла состоит из 14 рецессивных бит. Это не нарушает правила «битстаффинга» на шине и не оказывает влияния на другие узлы, если узел, который обнаружил ошибку, не ведет передачу сообщения (является текущим владельцем шины). В этом случае правила «битстаффинга» нарушаются, и передача данных прекращается. После передачи пассивной ошибки узел должен ожидать шесть последовательных рецессивных бит, прежде чем попытаться возобновить связь по шине.

Сообщение о перегрузке

Формат сообщения о перегрузке аналогичен формату сообщения об ошибке, но может быть сформирован, только когда шина простаивает.

Сообщение о перегрузке показано на рисунке 38.6.

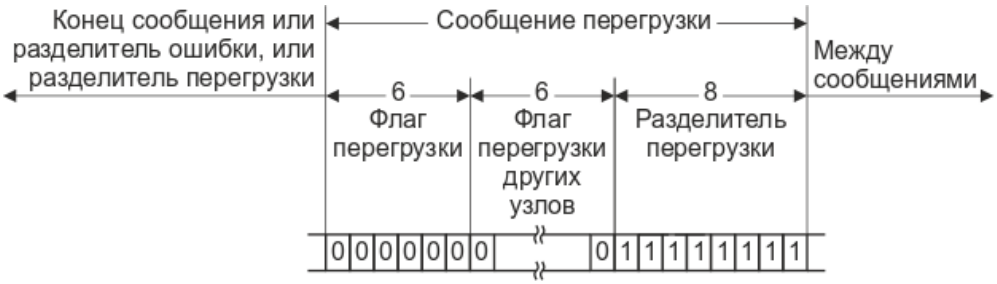


Рисунок 38.6 – Сообщение о перегрузке

Разделитель перегрузки состоит из восьми последовательных рецессивных бит.

Узел может сформировать сообщение о перегрузке в двух случаях:

- для задержки передачи нового сообщения;
- в качестве ответного кадра перегрузки при обнаружении доминантного бита во время первых двух битов паузы, обнаружении доминантного бита на месте последнего бита EOF приемником или обнаружении доминантного бита любым узлом на месте последнего бита разделителя ошибки или разделителя перегрузки.

Узел может последовательно сформировать не более двух сообщений перегрузки.

Флаг перегрузки состоит из шести последовательных доминантных бит. Другие узлы обнаруживают перегрузку и начинают формировать ее самостоятельно. Поэтому на шине во время выполнения перегрузки может быть до 12 доминантных бит.

38.2 Структура и функционирование контроллера CAN

В состав контроллера CAN входят четыре идентичных независимых узла CAN0, CAN1, CAN2 и CAN3. ОЗУ для хранения сообщений, которое является общим для узлов, и система управления. Контроллер CAN имеет следующие функциональные особенности:

- соответствие ISO 11898;
- функционирование согласно спецификации CAN 2.0B;
- отдельные управляющие регистры для каждого из двух узлов;
- программируемая скорость передачи информации до 1 Мбит/с;
- гибкий и полный контроль передачи сообщений и обработки ошибок.

Контроллер CAN реализует 16 линий прерываний и 256 объектов сообщений для хранения сообщений и их параметров в ОЗУ. Каждый объект сообщения может быть привязан к любому из узлов, сконфигурирован для передачи или приема как стандартных, так и расширенных сообщений и удаленных запросов. Каждый объект имеет индивидуальную маску для фильтрации принимаемых сообщений. Объекты сообщений могут объединяться в классы, с разными уровнями приоритета, могут объединяться для построения структур FIFO произвольных размеров (до 256 объектов в одной структуре). Кроме того, реализована возможность попарного соединения объектов для формирования шлюзов для автоматической передачи сообщений между узлами. Параллельно с вышеуказанными свойствами объекты сообщений могут организовываться в списки с постоянно доступной реорганизацией (совместимость с TwinCan-устройствами, которые не имеют списков).

Структура контроллера CAN приведена на рисунке 38.7.

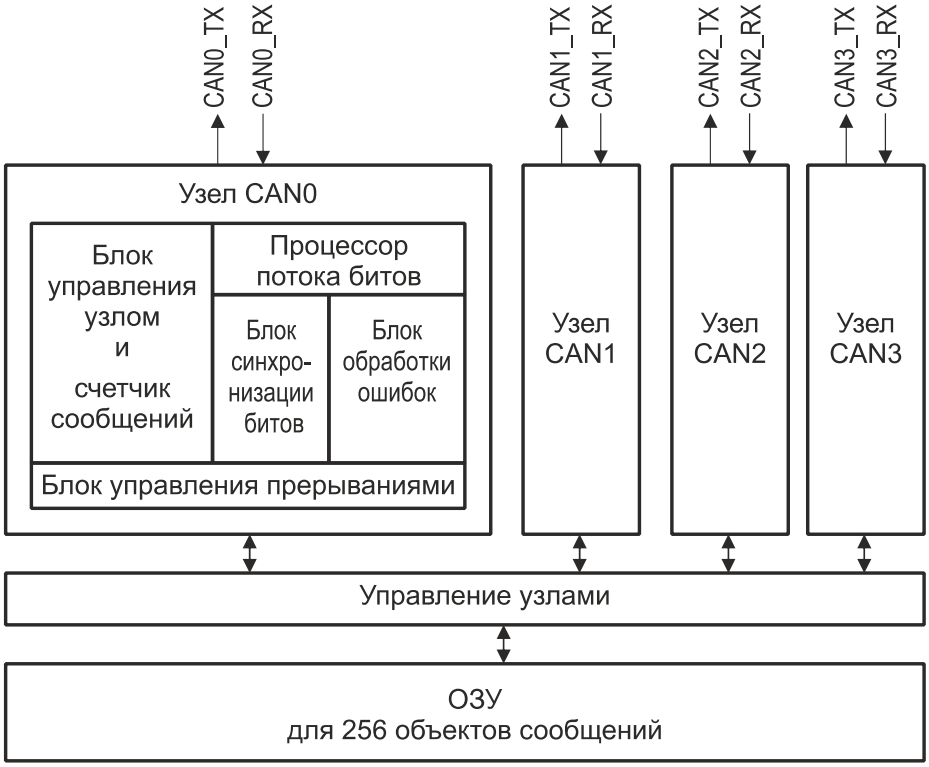


Рисунок 38.7 – Общая структура контроллера CAN

Синхронизация

Тактирующим сигналом контроллера CAN является сигнал FIN (HLCK), приходящий с генератора тактовых сигналов. На основе этого сигнала посредством программируемого дробного делителя частоты формируется внутренний синхросигнал FCAN (FOUT), синхронизирующий работу контроллера и являющийся базовым синхросигналом для передачи/приема сообщений по внешней шине CAN.

Включение контроллера CAN

По умолчанию после сброса микроконтроллера контроллер CAN выключен. На это также указывает состояние флага DISS регистра CLC. Когда контроллер выключен, этот флаг установлен.

Для включения контроллера CAN следует записать ноль в бит DISR регистра CLC. После этого флаг DISS сбросится. Рекомендуется проверять состояние флага DISS, перед началом программирования регистров контроллера, которые не доступны в выключенном состоянии.

Выключение контроллера CAN

Программно можно перевести контроллер CAN в режим выключения установкой бита DISR. Контроллер завершает все текущие операции, после чего устанавливает флаг DISS и отключает внутреннее тактирование, в связи с чем, все регистры становятся недоступными для обращения.

Простой шины

Между передачами сообщений шина CAN находится в рецессивном состоянии. Для выполнения условий простоя шины необходимо, чтобы было получено, как минимум, три рецессивных бита после завершения передачи/приема очередного сообщения.

Анализ работы контроллера CAN

Для анализа работы контроллера CAN доступны два режима – общего анализа и внутренней петли «loop-back».

Режим общего анализа включается установкой бита CALM регистра NCR узла и позволяет осуществлять независимый мониторинг работы узла, не затрагивая шину CAN. В этом режиме сообщения данных и удаленные запросы отслеживаются без участия узла в операциях на шине. Выходы узла находятся в рецессивном состоянии. Узел может получать сообщения данных, сообщения удаленных запросов и сообщения об ошибках, но работа узла на передачу запрещена. Полученные сообщения данных/удаленных запросов остаются без подтверждения (бит подтверждения остается в рецессивном состоянии), но принимаются и сохраняются (при совпадении идентификаторов) в соответствующих объектах сообщений. В ответ на входящие сообщения не выдается подтверждение, и не генерируются сообщения об ошибках. На удаленные запросы не выдаются сообщения данных, а сами сообщения данных не могут быть переданы установкой бита запроса передачи TXRQ регистра состояния объекта сообщения MOSTAT. Прерывания после приема генерируются (если это разрешено) для всех принятых сообщений, не содержащих ошибок.

Режим внутренней петли включается установкой бита LBM регистра NPCR и позволяет проводить внутреннее тестирование контроллера CAN, а также отладку управляющей программы без доступа к внешней шине CAN. Внутренняя петля состоит из внутренней шины CAN (внутри контроллера CAN) и переключателя выбора шины для каждого узла, см. рисунок 38.8. С помощью переключателя каждый узел CAN может быть подключен либо к внутренней шине (режим внутренней петли), либо к внешней шине (нормальный режим работы). Если выбран режим внутренней петли, то на внешнем передающем выводе узла CAN поддерживается рецессивный уровень сигнала, а состояние принимающего вывода игнорируется.

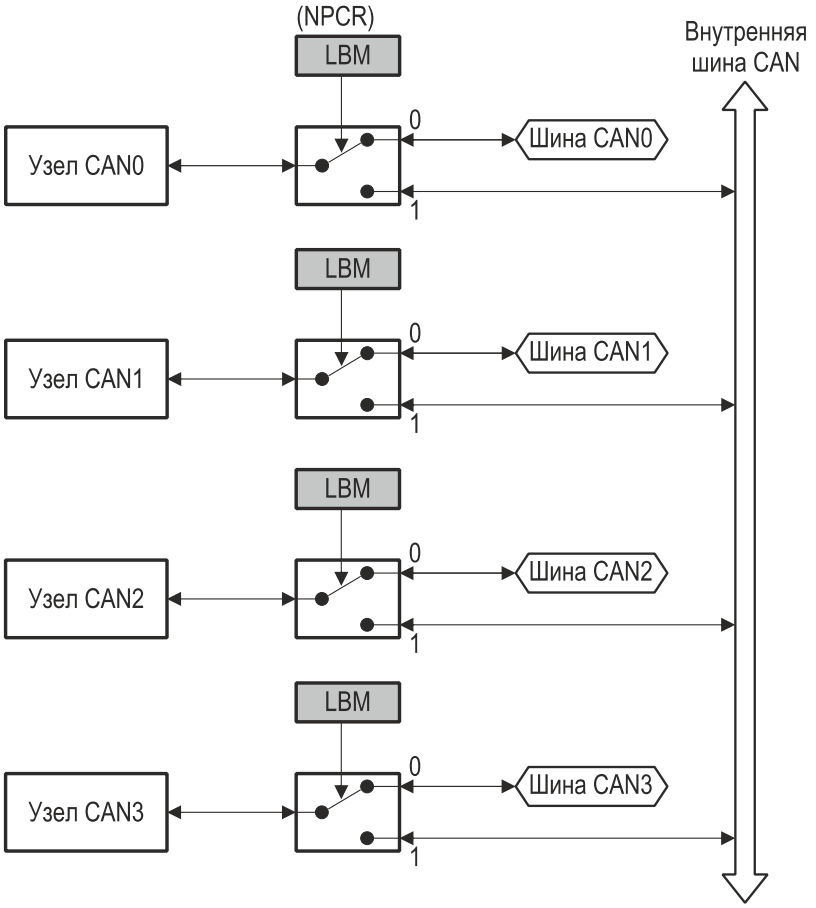


Рисунок 38.8 – Режим внутренней петли «loop-back»

Если оба узла CAN функционируют в режиме внутренней петли, они взаимодействуют друг с другом посредством внутренней шины CAN, не оказывая влияние на работу других модулей, функционирующих в нормальном режиме.

Дробный делитель

Дробный делитель позволяет генерировать тактовый сигнал FOUT из входного FIN (HCLK) путем программирования делителя посредством регистра FDR.

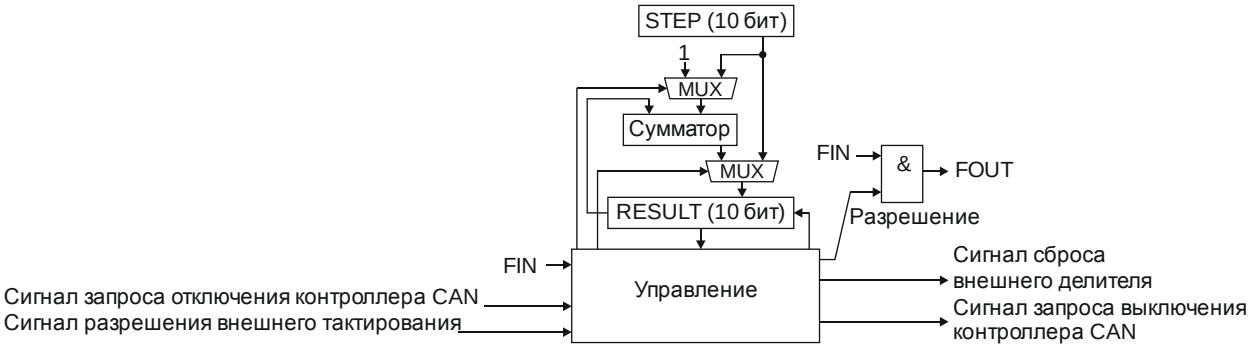


Рисунок 38.9 – Схема дробного делителя

Задаваемое значение частоты сигнала FIN зависит от длительности передачи одного бита информации и должно быть n-кратно ей. Поскольку длительность передачи бита определяется количеством квантов времени Ntq, (см. далее), то для расчета частоты сигнала FIN в МГц следует пользоваться формулой:

$$fFIN = n \times Ntq / tq, \qquad (38.1)$$

где Ntq – количество квантов времени tq;

n – целое число, начиная с 1 (для задания кратности).

tq – длительность одного кванта времени;

Дробный делитель делит f_{FIN} путем умножения на величину 1 / val или величину 1024 / val для любого val от 0 до 1023, получая на выходе тактовый сигнал FOUT (FCAN).

На рисунке 38.9 показана блок-схема дробного делителя. Логика дробного делителя работает по-разному, в зависимости от режима, задаваемого полем DM.

В режиме нормального деления (DM = 01b) делитель работает как перегружаемый счетчик с шагом инкрементирования, равным единице. Состояние счетчика доступно посредством поля RESULT. Каждый раз, при переполнении (т. е. когда RESULT = 3FFh), формируется импульс сигнала FOUT, после чего в счетчик загружается значение из поля STEP.

Частота сигнала FOUT определяется по формуле:

$$f_{FOUT} = f_{FIN} / (1024 - STEPd), \qquad (38.2)$$

где STEPd – значение поля STEP в десятичном формате.

Отсюда следует, что для получения сигнала FOUT с частотой, равной частоте сигнала FIN, значение STEP должно быть равно 3FFh. На рисунке 38.10 показано формирование сигнала FOUT при значении STEP = 3FDh (1021d).

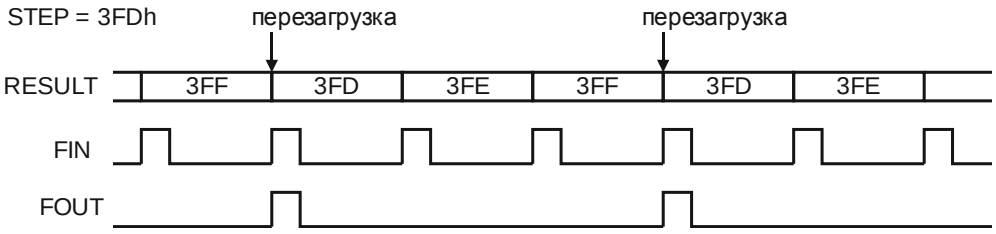


Рисунок 38.10 – Формирование сигнала FOUT в нормальном режиме

В режиме дробного деления (DM = 10b) делитель работает как перезагружаемый счетчик, но шаг инкрементирования в этом случае равен значению поля STEP. Если результат инкрементирования значения RESULT на величину STEP превышает 3FFh, возникает переполнение счетчика, формируется импульс сигнала FOUT, после чего в счетчик загружается значение, на которое результат инкрементирования превысил 3FFh.

Частота выходного сигнала FOUT определяется по формуле:

$$f_{FOUT} = f_{FIN} \times STEPd / 1024d \qquad (38.3)$$

В целом, режим дробного деления позволяет программировать частоту сигнала FOUT с более высокой точностью, чем нормальный режим, но сигнал может иметь джиттер периода, не превышающий одного периода FIN, в связи с чем, не рекомендуется использовать режим дробного деления при высоких скоростях передач.

На рисунке 38.11 показано формирование сигнала FOUT при значении STEP = 234h (564d). $f_{FOUT} = f_{FIN} \times 564 / 1024 = 0{,}55 \times f_{FIN}$.

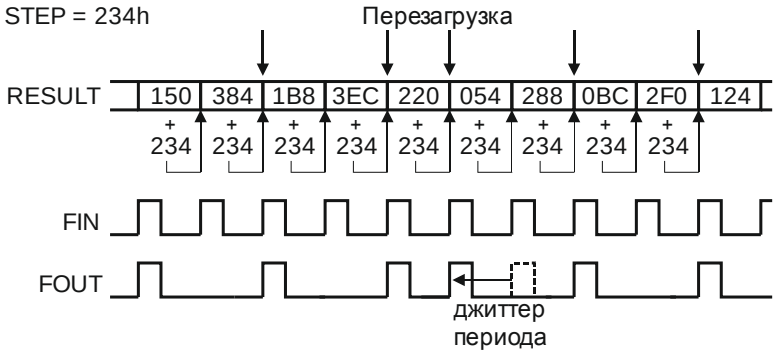


Рисунок 38.11 – Формирование сигнала FOUT в режиме дробного деления

Процесс выключения делителя начинается одновременно с возникновением запроса выключения контроллера CAN.

Контроллер сообщений

Контроллер сообщений управляет обменом сообщениями между CAN узлами и памятью сообщений и выполняет следующие функции:

- фильтрация входящих сообщений для определения корректного объекта сообщения для сохранения полученных данных;
- определение объекта сообщения, содержимое которого будет передано в первую очередь (для каждого узла индивидуально);
- передача содержимого объекта сообщения к CAN узлу с параллельной вставкой в сообщение битов управления и состояния;
- осуществление буферизации FIFO и работу в режиме шлюза;
- объединение битов уведомления ждущих обработки сообщений.

Управление прерываниями блока CAN

На рисунке 38.12 показана структура формирования запроса на прерывание.

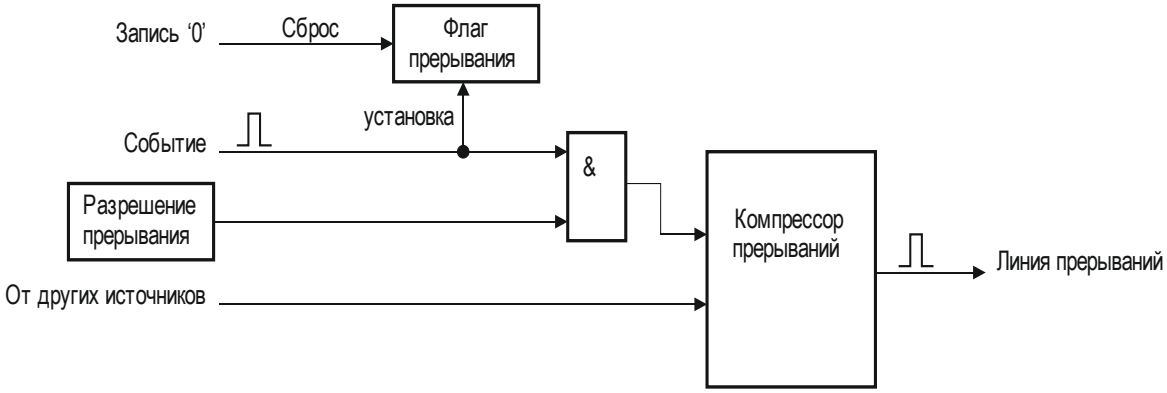


Рисунок 38.12 – Структура формирования запроса на прерывание

Событие, по которому должен быть сгенерирован запрос на прерывание, устанавливает флаг прерывания и (если разрешено) формирует запрос на прерывание на одной из 16 линий прерываний. Импульс запроса на прерывание генерируется независимо от состояния флага прерывания. Флаг прерывания может быть сброшен программно, записью нуля. Если к одной линии прерываний подключены несколько источников прерываний, то появление импульса от любого источника сформирует запрос на прерывание. Логика управления прерываниями использует схему компрессии прерываний.

Источниками прерываний являются:

- CAN узлы (шестнадцать источников – по четыре для каждого узла);
- объекты сообщений (512 источников – по два для каждого объекта);
- программное прерывание (источник – регистр MITR).

Каждый аппаратный источник прерывания управляется четырьмя битами указателя прерываний, который определяет для него одну из 16 линий прерываний, что позволяет коммутировать на одну линию несколько источников прерываний. На рисунке 38.13 представлена схема коммутации линий прерываний.

Когда объект сообщения Msg_x генерирует запрос на прерывание по окончании приема или передачи сообщения, запрос передается на линию прерываний, выбранную в битовом поле RXINP или TXINP регистра MOIPR объекта сообщения Msg_x. Если количество объектов сообщений больше, чем количество линий прерываний, то на одну линию могут приходить несколько запросов прерываний. Для разрешения конфликтов на линиях прерываний в контроллере CAN предусмотрен механизм распределения приоритетов для объектов сообщений.

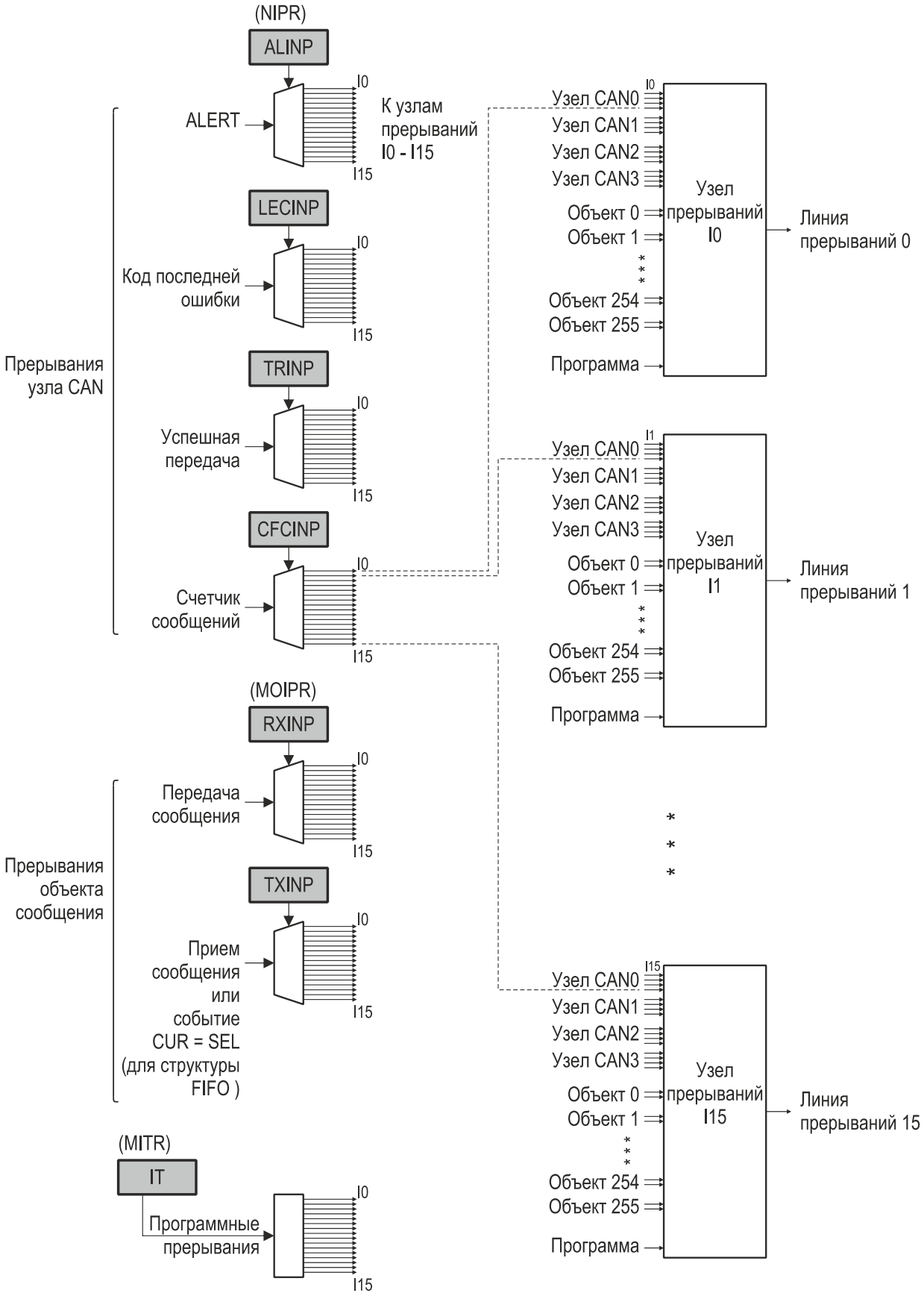


Рисунок 38.13 – Схема коммутации линий прерываний

38.3 Узел CAN

Каждый узел CAN имеет свою собственную логику управления и выдачи информации о состоянии и может быть сконфигурирован и работать независимо от другого узла.

Режим конфигурации включается установкой бита CCE регистра NCR. Режим конфигурации позволяет изменять параметры синхронизации битов и состояния счетчиков ошибок.

Конфигурация прерываний задается битами TRIE, ALIE и LECIE:

- бит TRIE управляет разрешением прерывания после передачи сообщения;
- бит ALIE управляет разрешением прерываний по ошибке;
- бит LECIE управляет разрешением прерывания по коду последней ошибки.

Регистр NSR отражает текущее состояние, содержит информацию о передачах и ошибках узла CAN.

Блок управления узлом CAN

Блок управления узлом координирует работу:

- разрешает/запрещает действия узла на шине;
- разрешает/запрещает и генерирует различные события, касающиеся работы узла (ошибка на шине, успешное завершение передачи сообщения), которые приводят к формированию запросов на прерывания;
- управляет счетчиком сообщений.

Блок синхронизации битов CAN

Согласно стандарту ISO 11898 время передачи одного бита разделено на сегменты, которые, в свою очередь, составлены из целочисленных отрезков времени, называемых квантами времени tq, см. рисунок 38.14. Квант времени – фиксированная единица времени, получаемая из частоты синхронизации и делителя контроллера CAN.

Сегмент синхронизации Tsync позволяет синхронизировать начало обмена данными между передатчиком и приемником. Длительность сегмента всегда равна одному кванту времени.

Сегмент распространения Tprop используется для компенсации физического времени запаздывания сигнала в пределах сети. Длительность сегмента рассчитывается с учетом времени прохождения сигнала от передатчика к приемнику и обратно, входной задержки компаратора и задержки выхода драйвера и может составлять от 1 до 8 квантов времени.

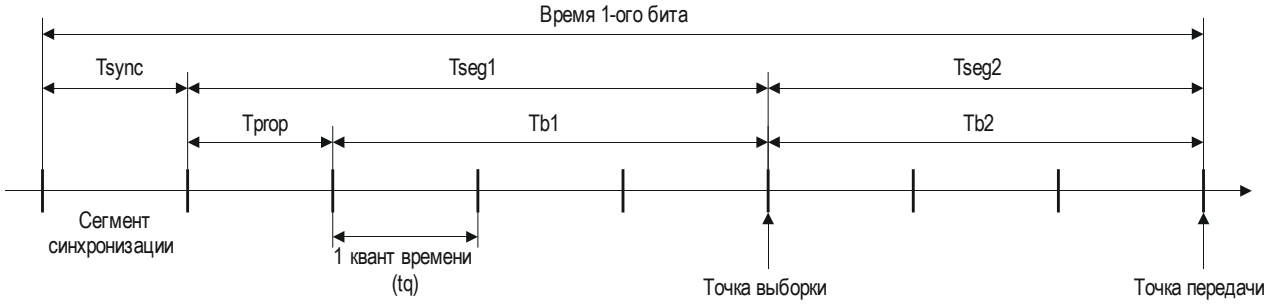


Рисунок 38.14 – Структура одного бита

Сегменты буфера фазы 1 и буфера фазы 2 – Tb1 и Tb2, расположенные до и после точки выборки, используются для компенсации смещения фазы тактовых частот источника и приемника, обнаруживаемой после появления сегмента синхронизации, а также для оптимального расположения точки выборки полученного бита.

Точка выборки – момент, когда читается состояние шины CAN для определения принятого бита. Как правило, длительность временного интервала от начала бита до точки выборки составляет (60 – 70) % времени бита, в зависимости от параметров сети CAN.

Сегмент распространения и сегмент буфера фазы 1 вместе составляют сегмент параметра 1 (Tseg1), который определяется битовым полем TSEG1 регистра синхронизации битов NBTR (может быть записан, только если установлен бит CCE регистра NCR). Согласно стандарту ISO 11898, минимальная длительность сегмента параметра 1 должна составлять два кванта времени (длительность каждого из сегментов Tprop и Tb1 не может быть меньше одного кванта времени).

Сегмент параметра 2 (Tseg2) определяется битовым полем TSEG2 регистра NBTR и охватывает сегмент буфера фазы 2. Минимальная длительность сегмента параметра 2 составляет один квант времени, но не может быть меньше, чем Tb1.

Не рекомендуется выбирать минимальную длительность одного бита, которая формируется сложением длительностей сегментов Tsync, Tseg1 и Tseg2, менее 8 квантов времени (стандарт ISO 11898 не обязывает поддерживать возможность запрограммировать значение меньше 8).

Максимальная длительность бита – 25 квантов времени.

Примечание – Минимальное номинальное время передачи одного бита составляет 1 мкс, что соответствует скорости передачи 1 Мбит/с.

Формулы вычисления значений сегментов и времени одного бита Tbit:

- при DIV8 = 0 значение кванта времени

$$tq = (BRP + 1) / f_{FOUT}; \quad (38.4)$$

- при DIV8 = 1 значение кванта времени

$$tq = 8 \times (BRP + 1) / f_{FOUT}; \quad (38.5)$$

- Tsync = 1 × tq;
- Tseg1 = Tprop + Tb1 = (TSEG1 + 1) × tq ≥ 2 × tq;
- Tseg2 ≥ Tb1;
- Tseg2 = (TSEG2 + 1) × tq ≥ 1 × tq;
- Tbit = Tsync + Tseg1 + Tseg2.

Чтобы компенсировать смещение фазы между частотами генераторов различных узлов шины CAN, каждое устройство должно синхронизироваться по фронту смены уровня сигнала на шине CAN от рецессивного к доминантному. Если такой фронт обнаруживается за пределами сегмента синхронизации, то фактическая длительность сегментов буфера фазы будет отличаться от запрограммированных значений.

Стандартом CAN предусматривается две формы синхронизации – жесткая синхронизация и повторная синхронизация.

В течении времени передачи одного бита (между двумя точками выборки) возможно выполнение только одной синхронизации. Синхронизация выполняется по каждому фронту смены уровня сигнала на шине CAN от рецессивного к доминантному, если на момент предыдущей выборки состояние шины было рецессивным. Если такое событие обнаружено в промежутке между сообщениями, то выполняется жесткая синхронизация. После неё внутреннее время передачи бита перезапускается с сегмента синхронизации. Таким образом, жесткая синхронизация должна заставить фронт, который вызвал жесткую синхронизацию, переместиться в пределы сегмента синхронизации перезапущенного битового интервала. Все остальные переключения из рецессивного в доминантное состояние шины CAN, которые удовлетворяют описанным в начале абзаца условиям, используются для повторной синхронизации.

Повторная синхронизация приводит к удлинению сегмента Tseg1 или укорачиванию сегмента Tseg2, если фронт сигнала на шине CAN не попал в сегмент синхронизации. Максимальное значение изменения сегментов колеблется в пределах от 1 до 4 квантов

времени, оно не должно превышать длительность Tb1. Фиксированное значение максимального числа последовательных бит одинаковой полярности гарантирует регулярное выполнение повторной синхронизации в пределах сообщения. Смещение фазы фронта смены уровня сигнала на шине отслеживается относительно сегмента синхронизации и измеряется в квантах времени.

Если величина фазового смещения меньше или равна запрограммированному значению ширины прыжка повторной синхронизации TSJW, именно на такое количество квантов времени изменится сегмент Tseg1 или Tseg2.

Если величина смещения фазы больше, чем TSJW, а фазовое смещение положительно, то сегмент Tseg1 удлиняется на величину TSJW, в случае отрицательного фазового смещения сегмент Tseg2 укорачивается на величину TSJW.

Значение T_{SJW} определяется полем SJW регистра NBTRx по формуле:

$$T_{SJW} = (SJW + 1) \times tq. \quad (38.6)$$

Помимо прочего, должны соблюдаться следующие правила:

$$Tseg1 \geq T_{SJW} + Tprop; \quad (38.7)$$

$$\text{и } Tseg2 \geq T_{SJW}. \quad (38.8)$$

Соотношения между максимальным отклонением частоты сигнала FOUT и сегментами буферов фаз, и шириной перехода ресинхронизации следующие:

$$\Delta f_{FOUT} \leq T / (2 \times (13 \times Tbit - Tb2)); \quad (38.9)$$

$$\Delta f_{FOUT} \leq T_{SJW} / (20 \times Tbit), \quad (38.10)$$

где T – меньшее из Tb1 и Tb2.

В итоге:

- Tsync составляет 1 квант времени;
- Tprop – от 1 до 8 квантов времени;
- Tb1 – от 1 до 8 квантов времени;
- Tb2 – выбирается равным двум квантам времени или равным сегменту Tb1, если его значение более двух квантов времени;
- T_{SJW} может составлять максимально 4 кванта времени, однако, в типовых приложениях достаточно 1.

Корректные значения параметров синхронизации битов должны быть записаны в регистр NBTR (доступен, если установлен бит CCE) до окончания инициализации (до сброса бита INIT регистра NCR), т. е. до начала работы CAN узла.

Процессор потока битов

Процессор потока битов формирует (на основе содержимого объектов сообщений) сообщения данных и удаленные запросы, которые планируется отправить по шине CAN. Процессор потока бит управляет генератором CRC (генератор контрольной суммы) и добавляет контрольную сумму к сообщению. После вставки битов начала SOF и конца EOF сообщения, процессор потока бит запускает процедуру арбитража шины CAN и, когда шина будет находиться в состоянии ожидания, инициирует передачу сообщения. В течение всего времени передачи сообщения процессор потока битов ведет мониторинг шины. Если обнаруживается несовпадение текущего (определяемого мониторингом) и ожидаемого (выдаваемого CAN узлом) уровня напряжения на шине CAN, генерируется ошибка и соответствующий ей запрос на прерывание. Код возникшей ошибки отражается в битовом поле LEC регистра NSR.

Корректность получаемого сообщения проверяется и подтверждается или не подтверждается кодом CRC. Если значение не соответствует ожидаемому, генерируется запрос на прерывание и код ошибки выставляется в регистре NSR, кроме этого, на шину выдается сообщение об ошибке.

После получения сообщения, не содержащего ошибок, и разбиения его на идентификатор и пакет данных, полученная информация записывается в буфер блока обработки сообщений, формируется соответствующее прерывание, и обновляются регистры состояния.

Блок обработки ошибок

Блок обработки ошибок узла CAN предназначен для выявления ошибок в работе устройств сети CAN. В составе блока есть два счетчика: счетчик ошибок приема (поле REC в регистре NECNT) и счетчик ошибок передачи (поле TEC). Инкрементированием и декрементированием счетчиков управляет процессор потока битов.

Если процессор потока битов сам выявляет ошибку в процессе передачи, то счетчик TEC инкрементируется на 8. Инкрементирование на 1 происходит, если об ошибке сообщено внешним CAN-устройством путем генерирования сообщения об ошибке. Направление передачи с ошибочным сообщением и узел, сообщивший об ошибке передачи, указывают на соответствующие узлы CAN в регистрах NECNT, что используется для анализа ошибки.

В зависимости от значений счетчиков ошибок узел CAN может находиться в одном из трех состояний:

- активен к ошибкам;
- пассивен к ошибкам;
- отключен от шины.

Узел находится в состоянии активности к ошибкам, если значение каждого из счетчиков ошибок меньше 128. Узел в состоянии активности к ошибкам присоединен к шине CAN и посылает активный флаг ошибки при обнаружении ошибок.

Узел находится в состоянии пассивности к ошибкам, если значение хотя бы одного из счетчиков ошибок больше или равно 128. Узел подключен к шине, но при обнаружении ошибок посылает пассивный флаг ошибки. После передачи узел в состоянии пассивности к ошибкам будет ждать инициализации дальнейшей передачи.

Узел находится в состоянии отключения от шины, если значение счетчика ошибок TEC больше или равно 256. О том, что CAN узел находится в состоянии отключения от шины CAN, сигнализирует флаг BOFF регистра NSR. Узел в состоянии отключения от шины CAN не может работать с шиной CAN (выходные передатчики отключены) Он будет оставаться в данном состоянии до завершения последовательности восстановления.

Узел в состоянии отключения от шины может стать активным к ошибкам и больше не являться отключенным от шины, если оба его счетчика ошибок устанавливаются в ноль после контроля ста двадцати восьми появлений на шине одиннадцати последовательных рецессивных битов (см. рисунок 38.15).

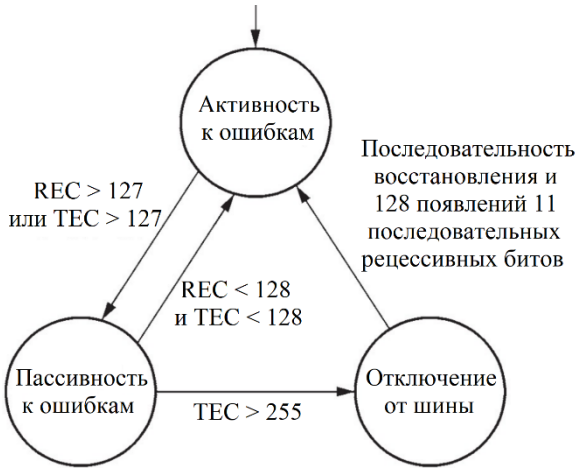


Рисунок 38.15 – Схема смены состояния узла

Флаг EWRN регистра NSR устанавливается, когда хотя бы один из счетчиков достиг или превысил лимит ошибок, определенный в битовом поле EWRNLVL регистра NECNT. Как только значения обоих счетчиков перестанут превышать лимит ошибок, флаг EWRN сбросится.

Счетчик сообщений

Счетчик сообщений может использоваться для получения информации о завершении передачи/приема сообщения соответствующего узла CAN. Подсчет сообщений осуществляется 16-разрядным счетчиком, который управляется регистром NFCR. Битовые поля CFMOD и CFSEL определяют режим работы и событие для инкрементирования счетчика.

Каждый узел CAN имеет в своем составе 16-разрядный счетчик сообщений/синхросчетчик, который подсчитывает количество принятых и переданных сообщений. Битовое поле CFSEL определяет один из трех режимов работы счетчика.

В режиме подсчета сообщений после успешной передачи и/или приема сообщения, содержимое счетчика копируется в битовое поле CFCVAL регистра MOIPR объекта сообщения Msg_x, участвующего в пересылке данных. После чего счетчик сообщений инкрементируется.

Прерывания узла CAN

Коммутация линий запросов прерываний показана на рисунке 38.16.

Узел может генерировать запросы на прерывания в случае:

- успешной передачи/приема сообщения;
- обнаружения кода последней ошибки;
- переполнения счетчика сообщений;
- состояния ALERT (состояние, возникающее, когда хотя бы один из счетчиков ошибок узла достиг значения своего лимита, изменяется состояние «отключен от шины», возникает ошибка длины списка или ошибка списка объектов).

После каждой успешной передачи или успешного приема сообщения генерируется (если разрешено соответствующими битами TXOK и RXOK) прерывание. Битовое поле TRINP регистра NIPR задает одну (из шестнадцати) линию прерывания.

Прерывание узла при возникновении кода последней ошибки формируется (если разрешено битом LECIE), если после модификации поля LEC его значение больше нуля. Битовое поле LECINP задает линию прерывания.

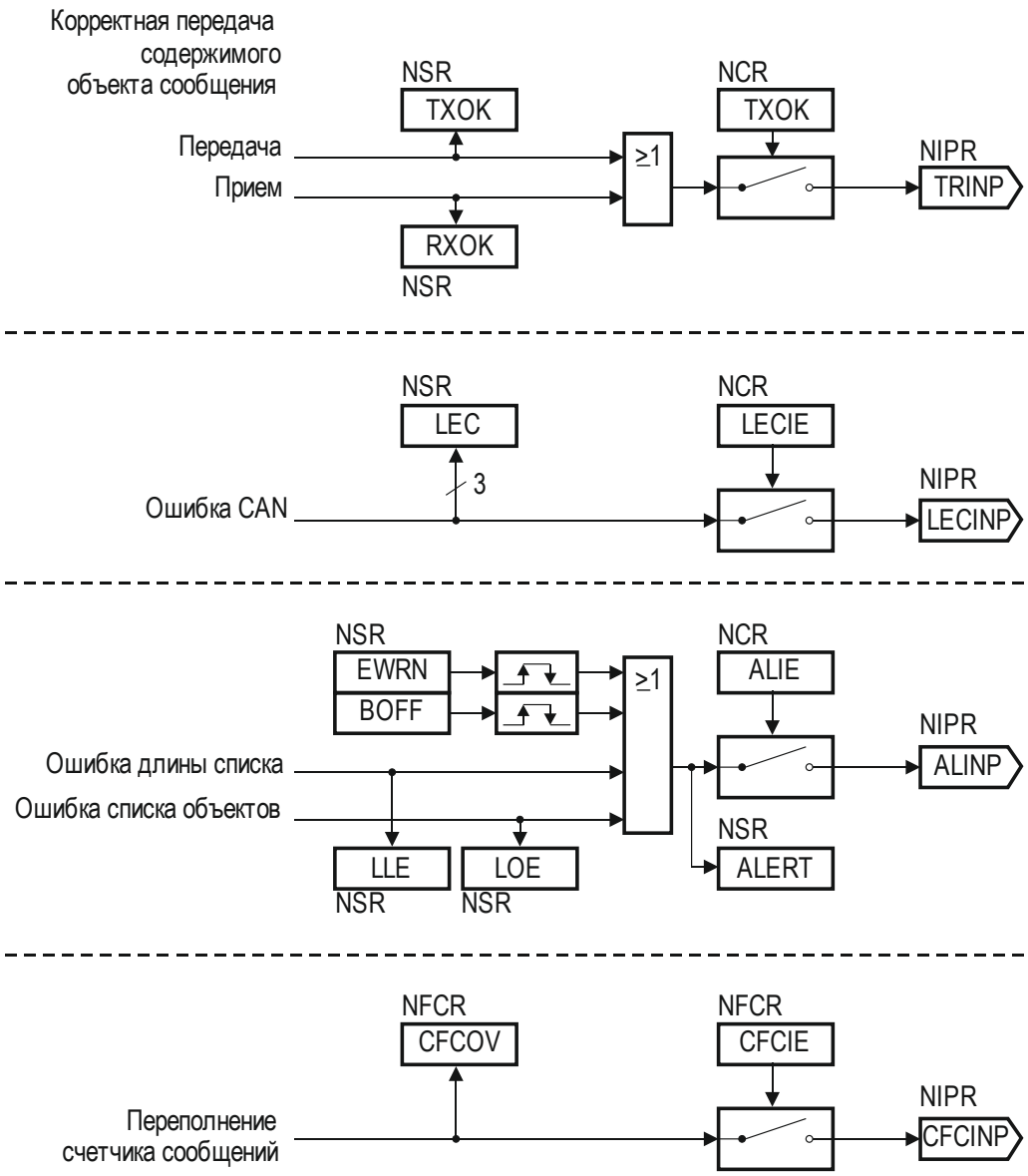


Рисунок 38.16 – Прерывания CAN узла

Прерывание узла при переполнении счетчика сообщений генерируется, если оно разрешено битом CFCIE регистра NFCR. Битовое поле CFCINP задает линию прерывания.

Прерывание ALERT может быть сформировано (если разрешено битом ALERT) любым из следующих событий:

- изменение состояния бита BOFF;
- изменение состояния бита EWRN;
- ошибка длины списка, которая также выставляет бит LLE;
- ошибка элемента списка, которая также выставляет бит LOE;
- бит INIT выставлен аппаратно.

Битовое поле ALINP задает линию прерывания.

В дополнение к аппаратным прерываниям есть возможность программного генерирования прерываний с использованием регистра прерываний MITR. Запись единицы в n-й разряд битового поля IT генерирует сигнал запроса прерывания на соответствующей ему n-ой линии прерываний (одной из шестнадцати). Установка нескольких битов приводит к параллельному генерированию запросов прерываний на соответствующих установленным битам линиях прерываний.

38.4 Объекты сообщений

Регистры управления и состояния объектов сообщений

В состав каждого объекта сообщения входят девять 32-разрядных регистров:

- управления и состояния – MOCTR (только запись) и MOSTAT (только чтение), доступные по одному адресу;
- арбитража – MOAR;
- данных – MODATAH и MODATAL;
- маски – MOAMR;
- указателя прерываний – MOIPR;
- указателя FIFO или шлюза – MOFGPR;
- управления функционированием – MOFCR.

Расположение регистров представлено на рисунке 38.17, где для примера взят пятый объект сообщения.

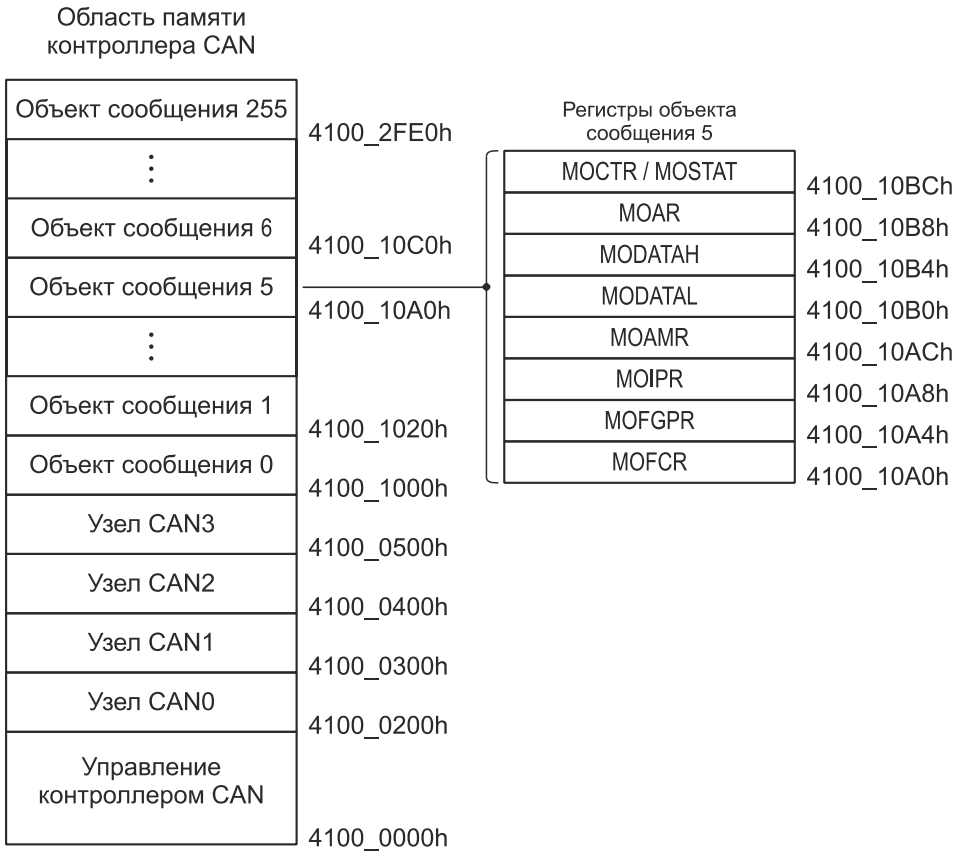


Рисунок 38.17 – Структура памяти регистров

Объекты сообщений контроллера CAN могут быть организованы в восемь списков, см. рисунок 38.18.

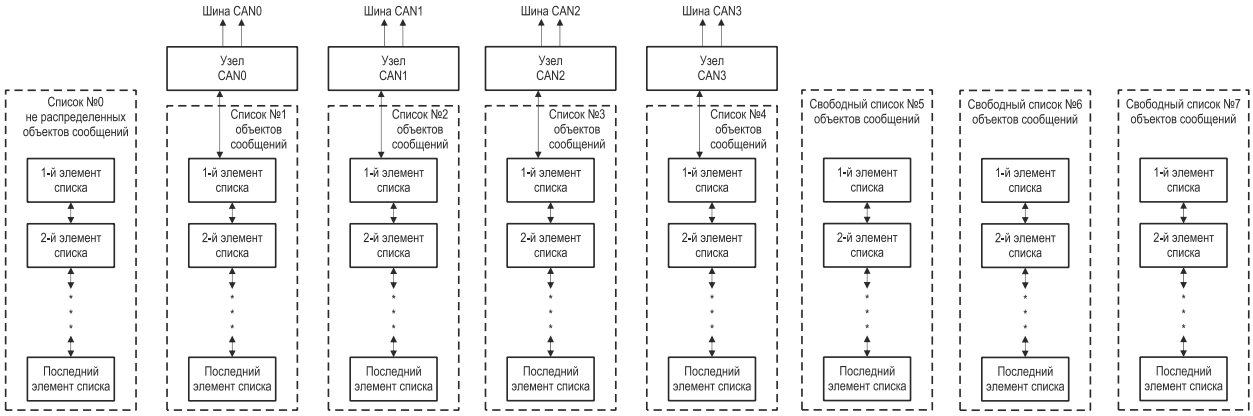


Рисунок 38.18 – Списки контроллера CAN

Каждый объект сообщения может быть добавлен в один из списков. Каждый узел CAN имеет свой список и соответствующий регистр списка. Регистр LIST1 отражает состояние списка №1 узла CAN0, регистр LIST2 – списка №2 узла CAN1, регистр LIST3 – списка №3 узла CAN2, регистр LIST4 – списка №4 узла CAN3.

Примечание – Узел может оперировать только с теми объектами сообщений, которые занесены в принадлежащий ему список.

Положение объекта сообщения Msg_x в списке определяется посредством регистра MOSTAT, который содержит указатели на предшествующий ему и следующий за ним элементы списка (объекты). Нераспределенные между узлами CAN объекты сообщений по умолчанию организуются в отдельный список №0, состояние которого отражается в регистре LIST0. Остальные три списка с номерами от 5 до 7 являются свободными (не принадлежат ни одному узлу) и имеют соответствующие регистры LIST5 – LIST7.

Примечание – Объекты сообщений, распределенные в списки с 5 по 7, не могут быть использованы узлами CAN.

Механизмы FIFO и шлюза оперируют с объектами сообщений независимо от их распределения по спискам, что дает возможность работы со всеми восемью списками. Следовательно, при использовании механизмов FIFO и шлюза следует внимательно следить за содержимым списков.

На рисунке 38.19 представлен вариант, когда объекты сообщений с номерами 3, 5 и 16 занесены в список №2, принадлежащий узлу CAN1. Состояние списка отражено в регистре LIST2.

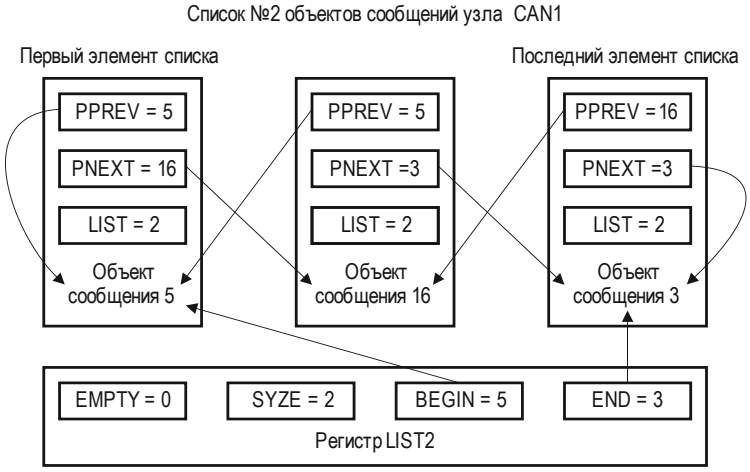


Рисунок 38.19 – Пример списка объектов сообщений

Значение поля BEGIN регистра LIST2 указывает на первый элемент списка (объект сообщения 5). Значение поля END указывает на последний элемент списка (объект сообщения 3). Количество элементов списка (количество объектов сообщений в списке) отражается в поле SIZE (значение SIZE всегда на единицу меньше количества элементов списка). Бит EMPTY является индикатором заполнения списка. Если список пуст, бит EMPTY установлен, в противном случае бит сброшен.

Каждый объект сообщения содержит номер списка (поле LIST), к которому он относится, а также указатели PNEXT и PPREV на следующий по списку объект сообщения и предшествующий соответственно. Поле PPREV первого по списку объекта сообщения должно указывать на этот же объект. Поле PNEXT последнего по списку объекта сообщения должно указывать на этот же объект.

На рисунке 38.19 указатель PPREV пятого объекта сообщения (первого в списке) имеет значение 5h, а указатель PNEXT третьего объекта сообщения (последнего в списке) имеет значение 3h. Значение поля LIST всех трех объектов сообщений равно 2h.

Объект сообщения, у которого LIST = 0h относится к нулевому списку нераспределенных объектов. После сброса все объекты сообщений считаются нераспределенными. По умолчанию, порядок элементов списка №0 следующий: объект сообщения (n – 1) является предыдущим объекта сообщения Msg_x, а объект сообщения (Msg_x + 1) – следующим.

Для просмотра структуры списка объектов сообщений узла достаточно обратиться к соответствующим регистрам LIST1/LIST2 и MOSTAT.

Структура списков управляется и изменяется посредством контроллера списков, который, в свою очередь, управляется панелью команд. Контроллер списков гарантирует, что при внесении изменений не нарушается согласованность структуры списков, и обеспечивает простоту и гибкость использования.

Панель команд запускается записью соответствующей команды в битовое поле PANCMD регистра PANCTR. Соответствующие аргументы команды должны быть записаны в битовые поля PANAR1 и PANAR2 до записи кода команды или, самое позднее, вместе с кодом команды одним 32 битным доступом в регистр панели команд.

Примечание – Запись новых значений в поля PANAR1 и PANAR2 не изменяет сразу их содержимого. Новые значения сначала попадают в специальный теневой регистр. Далее, одновременно с записью кода команды в поле PANCMD, новые значения из теневого регистра переносятся в поля PANAR1 и PANAR2.

С записью корректного кода команды выставляется флаг BUSY, и в дальнейшем все попытки записи в регистр PANCTR игнорируются. Флаг BUSY остается активным, а панель команд заблокированной до тех пор, пока не завершится выполнение записанной команды.

После сброса микроконтроллера контроллер списка формирует список №0 нераспределенных объектов сообщений. Во время этой операции флаг BUSY установлен, и все обращения к объектам сообщений запрещены. По окончании этой операции флаг BUSY сбрасывается, и объекты становятся доступными.

В случае появления команды динамического распределения, по которой какой-либо элемент забирается из списка №0 и переносится в другой указанный список, наряду с битом BUSY, устанавливается бит RBUSY. Это указывает на то, что значения битовых полей PANAR1 и PANAR2 будут обновлены контроллером списка, следующим образом:

- номер объекта сообщения, переносимого из списка №0 нераспределенных объектов сообщений, записывается в PANAR1;

- если установлен бит ERR (седьмой бит поля PANAR2), значит список №0 пуст и выполнение команды завершается; если бит ERR сброшен – список №0 не пуст и команда выполняется.

Результаты выполнения команды динамического распределения записываются до того, как контроллер списка начнет процесс распределения. Как только результаты станут доступны, бит RBUSY сбрасывается. Это позволяет пользователю запрограммировать настройки желаемого объекта сообщения, в то время как контроллер списка распределяет

объекты. Во время операций со списками доступ к объектам сообщений не запрещен, но следует помнить, что любой доступ к регистрам объектов сообщений в течение процесса распределения объектов вносит задержку (в процесс), равную длительности доступа.

При завершении выполнения команды код «нет операции» автоматически записывается в битовое поле PANCMD.

Новая команда может быть записана в любое время, когда бит BUSY сброшен.

Все битовые поля регистра PANCTR, исключая биты BUSY и RBUSY, могут быть записаны программно, что делает возможным сохранять и восстанавливать значения регистра PANCTR, если панель команд используется независимой подпрограммой обработки прерываний. Если возникает такая ситуация, то любые задачи, которые используют панель команд и которые могут прерывать выполнение других задач, тоже использующих панель команд, будут опрашивать состояние флага BUSY. До тех пор, пока флаг BUSY будет оставаться установленным, содержимое регистра PANCTR будет сохранено в соответствующей области памяти до операции восстановления. Как только подпрограмма обработки прерываний закончится, содержимое регистра PANCTR будет восстановлено.

До того, как объект сообщения, занесенный в список активного узла CAN, будет перенесен на другую позицию этого же списка или перенесен в другой список, бит MSGVAL регистра MOSTAT объекта сообщения Msg_x должен быть очищен.

Примечание – Если требуется перераспределить объекты сообщений в списки повторно, необходимо приостановить работу узлов CAN (установить бит INIT регистра NCR), а после занесения объектов в списки возобновить ее (сбросить бит INIT).

38.5 Прием и передача сообщений

Прием сообщения

После завершения приема сообщение сохраняется в объекте сообщения в соответствии с установленным алгоритмом, см. рисунок 38.20.

Помимо сохранения данных в объекте сообщения, контроллер CAN осуществляет обмен данными с ЦП.

При приеме сообщения информация сохраняется в объекте сообщения только в том случае, если установлен бит MSGVAL регистра MOSTAT. Если ЦП очищает бит MSGVAL, контроллер CAN останавливает запись в объект сообщения, и далее объект может быть реконфигурирован центральным процессором с последующей записью в него информации без участия контроллера CAN.

Полученное с шины сообщение может быть сохранено в объекте сообщения только в случае, если установлен бит RXEN. Контроллер CAN проверяет состояние бита RXEN только во время фильтрации принимаемого сообщения. После того, как сообщение принято, состояние бита не имеет значения и не оказывает влияния на дальнейшее сохранение данных в объекте сообщения.

Бит RXEN позволяет управлять блокированием объекта сообщения – после сброса бита RXEN полученное сообщение сохраняется в объекте сообщения, который получил приоритет, но в сохранении последующих сообщений этот объект не принимает участия.

Реконфигурация объекта сообщения центральным процессором во время работы контроллера CAN (например, сброс бита MSGVAL, изменение объекта сообщения и повторная установка бита MSGVAL) происходят следующим образом:

- объект сообщения получает приоритет;
- ЦП очищает бит MSGVAL для реконфигурации объекта сообщения;
- после реконфигурации ЦП снова устанавливает бит MSGVAL;
- завершается получение сообщения;
- если установлен бит MSGVAL, полученные данные сохраняются в объекте сообщения, генерируется запрос на прерывание, устанавливается соответствующий флаг;
- если сконфигурировано, производятся шлюзовые и FIFO операции.

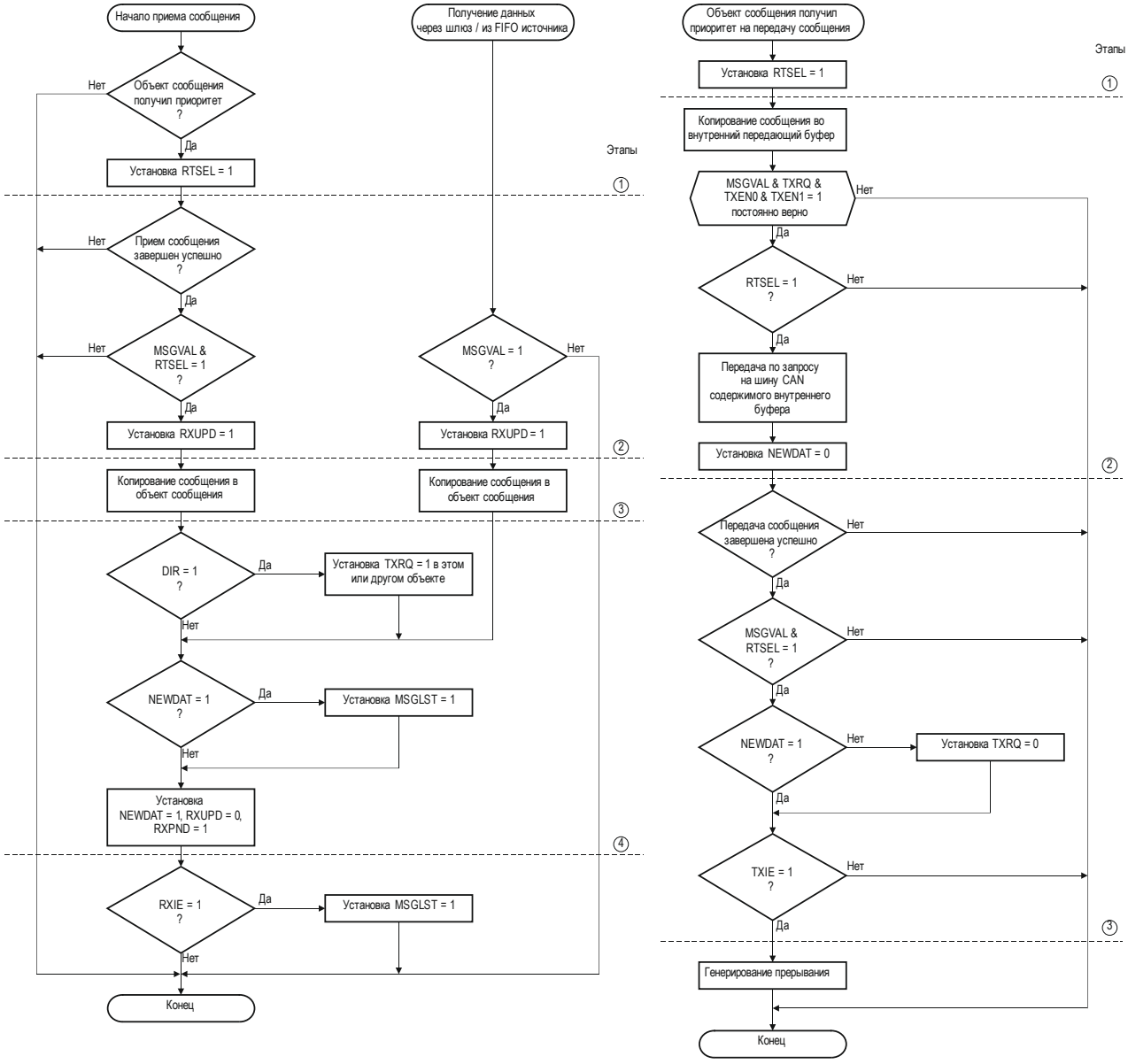


Рисунок 38.20 – Алгоритмы приема и передачи сообщения

Примечание – После реконфигурации объекта сохранение данных по завершении получения сообщения может быть нежелательным. Запретить запись данных в объект сообщения можно посредством бита RTSEL.

После получения объектом сообщения приоритета, его бит RTSEL устанавливается контроллером CAN, открывая, таким образом, объект сообщения для записи. После приема сообщения контроллер CAN дополнительно проверяет возможность записи в объект сообщения, а именно – установлен ли все еще бит RTSEL. И только в том случае, если бит RTSEL установлен, полученные данные сохраняются в объекте сообщения (вместе со всеми последующими действиями, которые указаны выше).

Если во время операций контроллера CAN объект сообщения становится некорректным (сброс бита MSGVAL), бит RTSEL должен быть сброшен до того, как бит MSGVAL будет установлен снова, или, по крайней мере, одновременно с ним. Это необходимо для предотвращения сохранения старой информации в объекте сообщения.

Реконфигурация объекта сообщения должна происходить следующим образом:

- сброс бита MSGVAL;
- реконфигурация объекта сообщения, пока бит MSGVAL сброшен;
- сброс бита RTSEL и далее установка бита MSGVAL.

Индикатором процесса сохранения (изменения) данных в объекте сообщения является флаг RXUPD, который выставляется с началом процесса сохранения (изменения) и сбрасывается с его окончанием.

После сохранения полученного сообщения (идентификатора, бита IDE, кода длины данных, поля данных, в случае сообщения данных) выставляется флаг NEWDAT. Если к моменту выставления (завершение сохранения/изменения данных) флаг NEWDAT был уже установлен, выставляется флаг MSGLST, который говорит о том, что произошла потеря данных.

Флаги RXUPD и NEWDAT позволяют произвести чтение корректных данных из объекта сообщения во время текущих операций контроллера CAN. Рекомендуемая последовательность действий следующая:

- сброс флага NEWDAT;
- чтение данных (идентификатор, данные и т. д.) из объекта сообщения;
- проверка флагов NEWDAT и RXUPD – оба флага должны быть сброшены. В случае невыполнения этого условия – возвращение к первому действию;
- если флаги NEWDAT и RXUPD сброшены, то содержимое объекта сообщения корректно и не используется контроллером CAN в течение операции чтения.

Поведение флагов RXUPD, NEWDAT и MSGLST идентично как для сообщений данных, так и для сообщений удаленных запросов.

Передача сообщения

Алгоритм передачи сообщений показан на рисунке 38.20. Одновременно с копированием данных (идентификатора, бита IDE, бита RTR, равного биту DIR, кода длины данных и собственно данных) из объекта сообщения, содержимое которого должно быть передано во внутренний передающий буфер соответствующего узла CAN, для контроля соблюдения четкой последовательности выполнения всех операций устанавливаются биты состояния.

Сообщение может быть передано только в случае, когда все четыре бита MSGVAL, TXEN0, TXEN1 и TXRQ установлены.

Бит RTSEL выставляется после того, как объект сообщения получает приоритет для передачи своего содержимого. Когда данные объекта сообщения копируются в передающий буфер, бит RTSEL проверяется и, если он установлен, сообщение передается. После успешной передачи сообщения бит RTSEL проверяется снова и, если он установлен, осуществляются дальнейшие операции.

Для полной и завершенной реконфигурации корректного объекта сообщения должны быть выполнены следующие шаги:

- очистка бита MSGVAL;
- реконфигурация объекта сообщения, пока бит MSGVAL сброшен;
- сброс бита RTSEL и установка бита MSGVAL.

Сброс бита RTSEL гарантирует как полное отключение объекта сообщения от текущей передачи, так и то, что никакие операции (копирование данных в передающий буфер, включая сброс бита NEWDAT, очистка бита TXRQ, прерывание сообщения и т. д.), относящиеся к старой конфигурации этого объекта сообщения, не повлияют на новую конфигурацию после установки бита MSGVAL.

После завершения передачи содержимого объекта сообщения в передающий буфер узла CAN, флаг NEWDAT аппаратно сбрасывается, тем самым обозначая, что объект сообщения открыт для записи новых данных.

Если после успешной передачи сообщения (на шину CAN) флаг NEWDAT все еще остается сброшенным (в объект сообщения не были записаны новые данные), флаг TXRQ аппаратно сбрасывается. Если же флаг NEWDAT был установлен программно (в связи с необходимостью передачи новых данных), флаг TXRQ не сбрасывается, тем самым разрешая передачу новых данных.

38.6 Фильтрация сообщений

Фильтрация при получении сообщений

При получении узлом CAN сообщения определяется объект сообщения, в котором будут сохранены получаемые данные в случае успешного приема.

Объект сообщения считается корректным для приема, если одновременно соблюдаются условия:

- объект сообщения распределен в список объектов сообщений узла, который принимает сообщение;
- бит MSGVAL установлен;
- бит RXEN установлен;
- бит DIR равен биту RTR принимаемого сообщения. Если бит DIR установлен, объект сообщения (объект передачи) может принять только сообщение удаленного запроса. Если бит DIR сброшен (объект приема), объект сообщения может принять только сообщение данных;
- если бит MIDE установлен, то бит IDE получаемого сообщения оказывает следующее влияние:
  - если бит IDE (регистр MOAR) установлен, то бит IDE принимаемого сообщения должен быть равен единице (расширенный идентификатор);
  - если бит IDE сброшен, бит IDE принимаемого сообщения должен быть равен нулю (стандартный идентификатор);
  - если бит MIDE сброшен, значение бита IDE принимаемого сообщения не важно, т. е. допускаются сообщения как со стандартным, так и с расширенным идентификатором;
- идентификатор полученного сообщения полностью (побитно) совпадает с идентификатором, хранящимся в регистре MOARn объекта сообщения, за исключением битов, закрытых маской регистра MOAMRn, значение которых не важно.

На рисунке 38.21 показан пример проверки идентификатора.

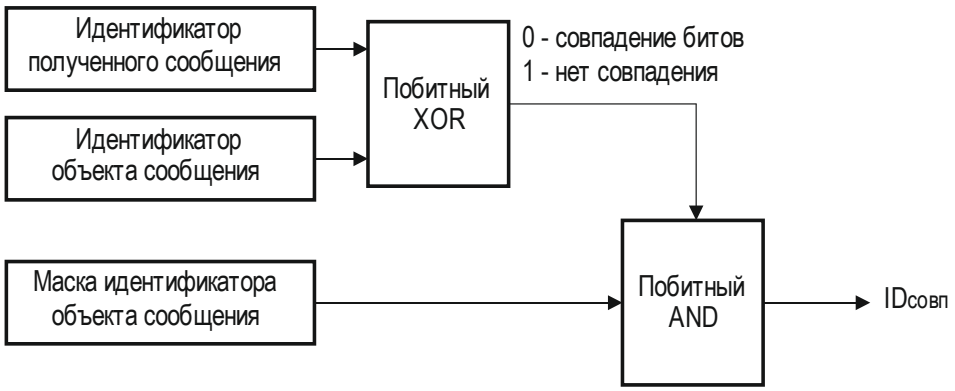


Рисунок 38.21 – Проверка идентификатора полученного сообщения

Среди всех объектов сообщений, которые отвечают указанным выше критериям, для сохранения полученного сообщения выбирается объект с наивысшим приоритетом. Для задания приоритета используется поле PRI в регистре MOAR. Объект сообщения, у которого значение поля PRI меньше, имеет больший приоритет. При равенстве значений поля PRI приоритетным считается объект сообщения, который находится ближе к началу списка.

Фильтрация при передаче сообщений

Когда требуется передача содержимого какого-либо объекта сообщения, в соответствующих управляющих регистрах выставляются флаги, указывающие на необходимость передачи. Объект сообщения считается корректным для передачи, если одновременно соблюдаются условия:

- объект сообщения распределен в список объектов сообщений узла CAN;
- флаг MSGVAL установлен;
- флаг TXRQ установлен;
- флаги TXEN0 и TXEN1 установлены.

Может возникнуть ситуация, когда передачи требуют одновременно несколько объектов сообщений. Среди всех объектов, которые отвечают указанным выше критериям, для передачи выбирается объект с наивысшим приоритетом.

Объект сообщения, у которого значение поля PRI меньше, имеет больший приоритет. При равенстве значений поля PRI разных объектов приоритет определяется следующим образом:

- при PRI = 10b – согласно правилам арбитража передачи сообщения (см. таблицу А.31.2);
- при PRI = 01b/11b приоритет имеет объект сообщения, который находится ближе к началу списка.

Объект сообщения, являющийся корректным для передачи и имеющий приоритет, будет осуществлять передачу первым. Остальные объекты сообщений будут переданы по очереди, согласно их приоритетам.

Объект сообщения определяется как стандартный объект сообщения, если в регистре MOFCR значение битового поля MMC равно нулю. Стандартный объект сообщения может принимать и передавать сообщения, согласно правилам, описанным выше.

На рисунке 38.22 показано формирование запроса на передачу объекта сообщения.

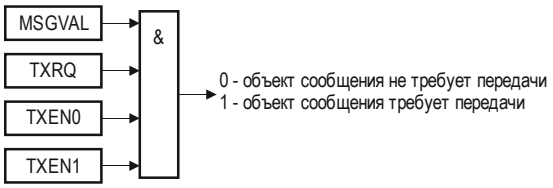


Рисунок 38.22 – Формирование запроса на передачу объекта сообщения

38.7 Удаленные запросы

После получения узлом CAN сообщения удаленного запроса и сохранения его в объекте сообщения, выставляется бит запроса передачи для ответа на удаленный запрос (отправка сообщения данных) или для автоматического повторения запроса.

В зависимости от состояния бита FRREN объекта сообщения, который принял сообщение удаленного запроса, возможны два варианта действий:

- если бит FRREN сброшен, то устанавливается флаг TXRQ этого объекта;
- если бит FRREN установлен, то устанавливается флаг TXRQ того объекта, на который указывает поле CUR объекта, принявшего удаленный запрос. При этом поле CUR не меняет своего значения.

Состояние регистров объекта сообщения, передающего сообщение удаленного запроса

У объекта сообщения, передающего сообщение удаленного запроса, в регистре MOSTAT должен быть сброшен бит DIR (объект передает сообщение данных) и установлены биты TXEN0, TXEN1, MSGVAL и TXRQ. Значение идентификатора в регистре MOAR передающего объекта сообщения должно быть равно значению

идентификатора принимающего объекта сообщения (или совместно с регистром MOAMR обеспечивать успешное прохождение фильтрации), чтобы сообщение удаленного запроса было принято принимающим объектом другого узла. Само сообщение удаленного запроса должно содержать идентификатор принимающего объекта сообщения, поэтому значение регистра MODATAL передающего объекта сообщения должно быть равно значению регистра MOAR принимающего объекта.

Состояние регистров объекта сообщения, принимающего сообщение удаленного запроса при FRREN = 0

У объекта сообщения, принимающего сообщение удаленного запроса, должны быть установлены биты DIR (объект принимает сообщение удаленного запроса), TXEN0 и TXEN1 (если ответ на запрос будет отправлен объектом), RXEN и MSGVAL. Регистры MODATAL и MODATAH должны содержать данные, которые будут переданы в ответ на запрос.

Состояние регистров объекта сообщения, принимающего сообщение удаленного запроса (при FRREN = 1) и содержащего данные для ответа на запрос

У объекта сообщения, принимающего сообщение удаленного запроса, должны быть установлены биты DIR, RXEN и MSGVAL. Битовое поле CUR должно указывать на номер объекта сообщения (должен находиться в том же узле, что и объект принявший сообщение удаленного запроса), содержащего данные, предназначенные для передачи в ответ на поступивший удаленный запрос.

В свою очередь у объекта сообщения, хранящего данные для отправки в ответ на запрос, должны быть установлены биты DIR, (объект передает сообщение данных), TXEN0, TXEN1 и MSGVAL. Бит TXRQ устанавливается автоматически при приеме сообщения удаленного запроса принимающим объектом сообщения.

Прием ответа на запрос (переданного сообщения данных) осуществляется стандартным объектом сообщения запрашивающего узла CAN (обмен данными происходит между объектом сообщения, хранящим данные для отправки в ответ на запрос, и объектом сообщения запрашивающего узла).

38.8 Дополнительные режимы передачи

Дополнительно имеются два режима, каждый из которых может быть выбран индивидуально:

- режим передачи данных с защитой от повторений;
- режим однократной пересылки данных.

Режим передачи данных с защитой от повторения

Режим передачи данных с защитой от повторения выбирается установкой бита SDT регистра MOFCR.

После приема сообщения данных и сохранения его в объекте с установленным битом SDT, бит MSGVAL этого объекта аппаратно сбрасывается, чтобы исключить возможность повторного приема и записи в этот объект. Этот режим нельзя использовать для базового объекта FIFO структуры.

В ответ на сообщение удаленного запроса, принятое объектом с установленным битом SDT, будут отправлены данные из объекта сообщения, на который указывает поле CUR объекта, принявшего удаленный запрос. После этого бит MSGVAL объекта, принявшего сообщение удаленного запроса, сбросится.

Примечание – Объект, принявший сообщение удаленного запроса, не может быть источником данных, передаваемых в ответ на запрос. Это означает, что в данном режиме бит FRREN объекта, принявшего удаленный запрос, обязательно должен быть установлен.

Режим однократной пересылки данных

Режим однократной пересылки данных выбирается установкой бита STT.

Бит TXRQ сбрасывается, когда содержимое объекта сообщения копируется в передающий буфер узла CAN. Таким образом, в дальнейшем, при неудачной (вследствие ошибок) пересылке сообщения по CAN-шине, повторной передачи не будет.

38.9 FIFO структура объектов сообщений

Регистр MOFGPR объекта сообщения Msg_x содержит установки указателей на объекты сообщений, которые используются при операциях FIFO и шлюзовых операциях.

В случае сильной загрузки ЦП обработка серии сообщений может быть затруднена – например, вследствие получения и/или передачи большого числа сообщений за малые промежутки времени. Для таких случаев предусмотрена система буферов быстрого ввода-вывода, так называемая FIFO структура, которая может функционировать автоматически, и позволяет избежать потери принимаемых сообщений, минимизировать время подготовки сообщений к отправке, а также генерировать прерывания по окончании операций.

Допускается организация нескольких параллельных FIFO структур. Число структур и их составляющих зависит только от количества доступных объектов сообщений. FIFO структура может быть создана, изменена и удалена в любой момент времени, даже во время операций контроллера CAN.

На рисунке 38.23 представлена основная FIFO структура. Она состоит из одного базового объекта и n-ого числа вспомогательных объектов.

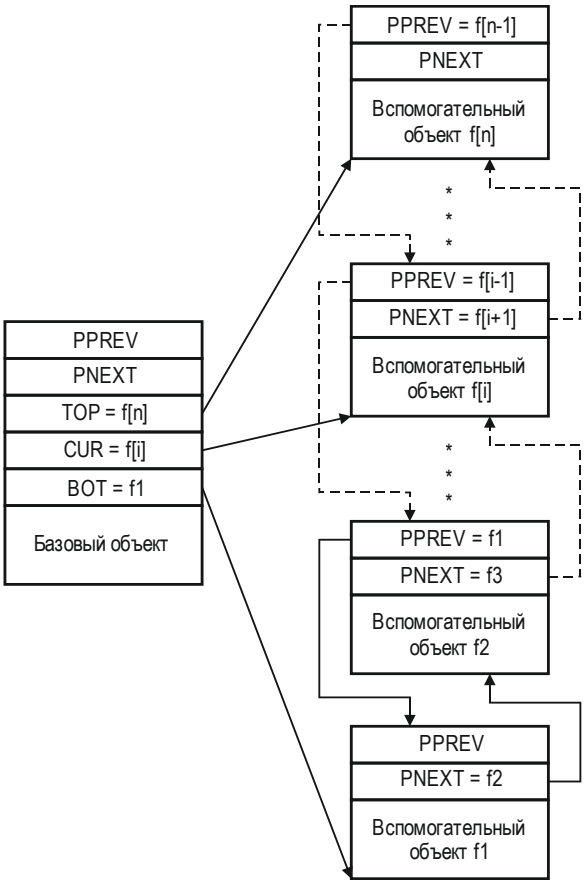


Рисунок 38.23 – FIFO структура с базовым объектом и n вспомогательными объектами

Вспомогательные объекты передающей FIFO структуры объединяются последовательно в списки (подобно спискам объектов сообщений). Базовый объект может быть занесен в любой список. Хотя на рисунке 38.23 базовый объект не относится ни к одному из списков, он может быть вставлен в любую последовательность вспомогательных объектов. Это означает, что базовый объект одновременно является и вспомогательным объектом (шлюзовые операции не возможны). Порядковые номера объектов сообщений (0, 1, 2 и т. д.) не имеют никакого значения при FIFO операциях с объектами.

Вспомогательные объекты должны быть определены в общий список (так как они последовательно связаны). С помощью указателей (битовые поля BOT, CUR и TOP) можно присоединять базовый объект к вспомогательному объекту, независимо от того, принадлежат базовый и вспомогательный объекты одному списку или разным спискам, но базовый должен быть первым в списке в таком случае.

FIFO структура минимального размера может состоять из одного объекта сообщения, который будет одновременно являться и базовым, и вспомогательным (фактически не используется). FIFO структура максимального размера может включать в себя все 256 объектов сообщений.

В базовом объекте FIFO границы установлены: поле BOT указывает на самый младший элемент FIFO структуры, поле TOP – на самый старший элемент, поле CUR – на вспомогательный объект, который в настоящий момент выбран контроллером CAN для передачи сообщения. Как только начинается передача, в CUR записывается номер следующего по списку вспомогательного объекта сообщения (CUR = PNEXT используемого объекта). Если значение битового поля CUR достигло номера старшего элемента списка (CUR = TOP), то следующим значением будет BOT (реализация автоматического перехода в начало списка). Таким образом, реализуется замкнутая FIFO структура, в которой битовые поля TOP и BOT устанавливают связь между началом и концом списка.

Битовое поле SEL позволяет определить вспомогательный объект, в пределах списка, для которого генерируется прерывание всякий раз, когда указатель CUR достигает значения указателя SEL. Также битовое поле SEL позволяет отследить окончание запланированной передачи серии сообщений или выдать прерывание, предупреждающее о том, что FIFO структура становится заполненной.

Вспомогательные объекты приемной FIFO структуры могут принадлежать списку любого узла.

FIFO структура для приема

FIFO структура для приема используется для буферизации входящих сообщений данных и удаленных запросов.

FIFO структура для приема активируется записью значения 0001b в битовое поле MMC регистра MOFCR базового объекта. Эта запись автоматически определяет объект как базовый объект приема FIFO. Типы вспомогательных объектов FIFO не имеют значения при операциях.

Когда базовый объект FIFO получает сообщение, оно сохраняется не в этом базовом объекте, а во вспомогательном объекте сообщения, на который указывает битовое поле CUR. При этом по умолчанию предполагается, что для вспомогательного объекта MMC = 0000b (действительное значение MMC игнорируется) и никаких операций фильтрации принимаемого сообщения не производится.

Одновременно с приемом сообщения текущее значение указателя CUR базового объекта меняется на номер следующего по списку вспомогательного объекта FIFO структуры. Этот вспомогательный объект будет использован для приема следующего сообщения.

Если установлен флаг OVIE регистра MOFCR базового объекта и значение указателя CUR становится равным значению указателя SEL, генерируется прерывание переполнения.

Это прерывание генерируется на узле прерываний с указателем TXINP базового объекта сразу после сохранения полученного сообщения во вспомогательном объекте. Прерывания генерируются, если это разрешено битом TXIE.

Следует помнить, что сообщение сохраняется в базовом и вспомогательном объектах FIFO, только если установлен бит MSGVAL.

Во избежание непосредственного приема сообщения вспомогательным объектом, как если бы он был независимым объектом и не принадлежал FIFO структуре, флаги RXEN всех вспомогательных объектов должны быть сброшены. Состояние флага RXEN неважно в случае, когда вспомогательный объект занесен в список, не связанный с узлом CAN.

FIFO структура для передачи

FIFO структура для передачи используется с целью буферизации серий сообщений данных или удаленных запросов, которые должны быть отправлены. FIFO структура для передачи состоит из базового объекта и одного или более вспомогательных объектов.

FIFO структура для передачи активируется записью значения 0010b в поле MMC регистра MOFCR базового объекта. В отличие от FIFO структуры для приема, в битовые поля MMC вспомогательных объектов (FIFO структуры для передачи) должно быть записано значение 0011b. Указатели CUR всех вспомогательных объектов должны указывать на базовый объект FIFO передачи (чтобы инициализироваться программно).

Флаги TXEN1 всех вспомогательных объектов сообщений, за исключением одного, на который указывает указатель CUR базового объекта, должны быть программно сброшены. Флаг TXEN1 указанного объекта должен быть установлен. Указатель CUR базового объекта может быть инициализирован для любого вспомогательного объекта.

При определении корректности объектов сообщений FIFO структуры для начала FIFO операций базовый объект должен быть определен первым как корректный, т. е. MSGVAL должен быть установлен.

В случае необходимости удаления FIFO структуры, прежде чем начнется операция удаления, все вспомогательные объекты, принадлежащие этой FIFO структуре, должны быть определены как некорректные (биты MSGVAL должны быть сброшены).

FIFO структура для передачи использует флаги TXEN1 всех своих объектов для выбора сообщения для передачи. В результате фильтрации право передавать сообщение получает тот объект, у которого выставлен флаг TXEN1. После передачи сообщения флаг TXEN1 аппаратно сбрасывается, а в указатель CUR записывается номер следующего объекта, требующего отправки сообщения, для которого уже выставлен (аппаратно) свой флаг TXEN1, и так далее – для всей FIFO структуры.

Если установлен флаг OVIE регистра MOFCRn базового объекта и значение указателя CUR становится равным значению указателя SEL, генерируется прерывание переполнения. Это прерывание генерируется на узле прерываний с указателем RXINP базового объекта после завершения операций получения сообщения. Прерывания приема базового объекта генерируются, если это разрешено битом RXIE.

Программирование регистров для FIFO структуры

1 Для передающего базового объекта:

- сбросить бит MSGVAL;
- задать поля CUR, BOT, TOP, SEL;
- записать значение 0010b в поле MMC, задать DLC, установить биты OVIE и RXIE (если необходимо).

Примечание – Состояние регистров MOAR и MOAMR передающего базового объекта не важно, поскольку в передаче участвуют передающие вспомогательные объекты и принимающий базовый объект. Поле RXINP указывает линию, на которую будет выдаваться прерывание переполнения (CUR = SEL).

2 Для передающих вспомогательных объектов:

- сбросить бит MSGVAL;
- установить биты DIR, TXEN1 (только для того вспомогательного объекта, на который указывает поле CUR передающего базового объекта, у остальных вспомогательных объектов бит TXEN1 должен быть сброшен), TXEN0;
- записать в поле CUR номер передающего базового объекта;
- записать значение 0011b в поле MMC, задать DLC.

Примечание – Значение регистров MOAR передающих вспомогательных объектов должно совпадать (или совместно с регистрами MOAMR обеспечивать успешное прохождение фильтрации) со значением регистра MOAR принимающего базового объекта, так как процесс передачи фактически происходит между ними (или иного принимающего объекта, если на приеме используется не FIFO структура).

3 Для принимающего базового объекта:

- установить бит RXEN;
- задать поля CUR, BOT, TOP, SEL;
- записать значение 0001b в поле MMC, задать DLC, установить биты OVIE и TXIE (если необходимо).

Примечание – Значение регистра MOAR принимающего базового объекта должно быть равно значению регистров MOAR передающих вспомогательных объектов передачи (или совместно с регистром MOAMR обеспечивать успешное прохождение фильтрации). Поле TXINP указывает, на какую линию будет выдаваться прерывание переполнения (прерывание после операции сохранения полученного сообщения во вспомогательных объектах при CUR = SEL).

4 Для принимающих вспомогательных объектов:

- сбросить бит RXEN (не требуется, если вспомогательные объекты занесены в список, не связанный с узлом CAN);
- задать поле DLC (состояние поля MMC не важно).

Примечание – Состояние регистров MOAR, принимающих вспомогательные объекты, не важно.

5 Установить бит MSGVAL в первую очередь у передающего базового объекта, а затем у всех остальных объектов.

6 Установить бит TXRQ для всех передающих вспомогательных объектов, начиная с того, на который указывает поле CUR передающего базового объекта.

38.10 Режим шлюза

Режим позволяет реализовывать автоматическую передачу информации через шлюз между двумя независимыми шинами CAN без участия ЦП.

Шлюз можно сформировать на уровне объектов сообщений и осуществлять передачу информации между узлами CAN. Шлюз может быть сформирован между двумя любыми объектами сообщений, принадлежащими разным узлам CAN. Количество шлюзов зависит только от количества объектов сообщений, допускающих формирование шлюзов.

Режим шлюза активируется записью значения 0100b в битовое поле MMC регистра MOFCR объекта сообщения Msg_x, инициализирует его как шлюзовый объект-источник. Объект сообщения, который будет являться шлюзовым объектом-приемником, выбирается указателем CUR объекта-источника. Для формирования шлюза достаточно, чтобы объект-приемник был корректным (установлен бит MSGVAL). Остальные параметры не влияют на возможность осуществления передачи между объектами от источника к приемнику.

Шлюзовый объект-источник, см. рисунок 38.24, функционирует как обычный объект сообщения, с тем отличием, что возможны дополнительные действия контроллера CAN при приеме и сохранении сообщения в объекте-приемнике:

1 Если установлен флаг DLCC регистра MOFCRn объекта-источника, код длины данных DLC копируется из шлюзового объекта-источника в шлюзовый объект-приемник.

2 Если установлен флаг IDC объекта-источника, идентификатор ID и расширение IDE копируются из шлюзового объекта-источника в шлюзовый объект-приемник.

3 Если установлен флаг DATC объекта-источника, байты данных, хранящиеся в двух регистрах MODATAL и MODATAH объекта-источника, копируются из шлюзового объекта-источника в шлюзовый объект-приемник. Копируются все восемь байт данных, вне зависимости от значения поля DLC.

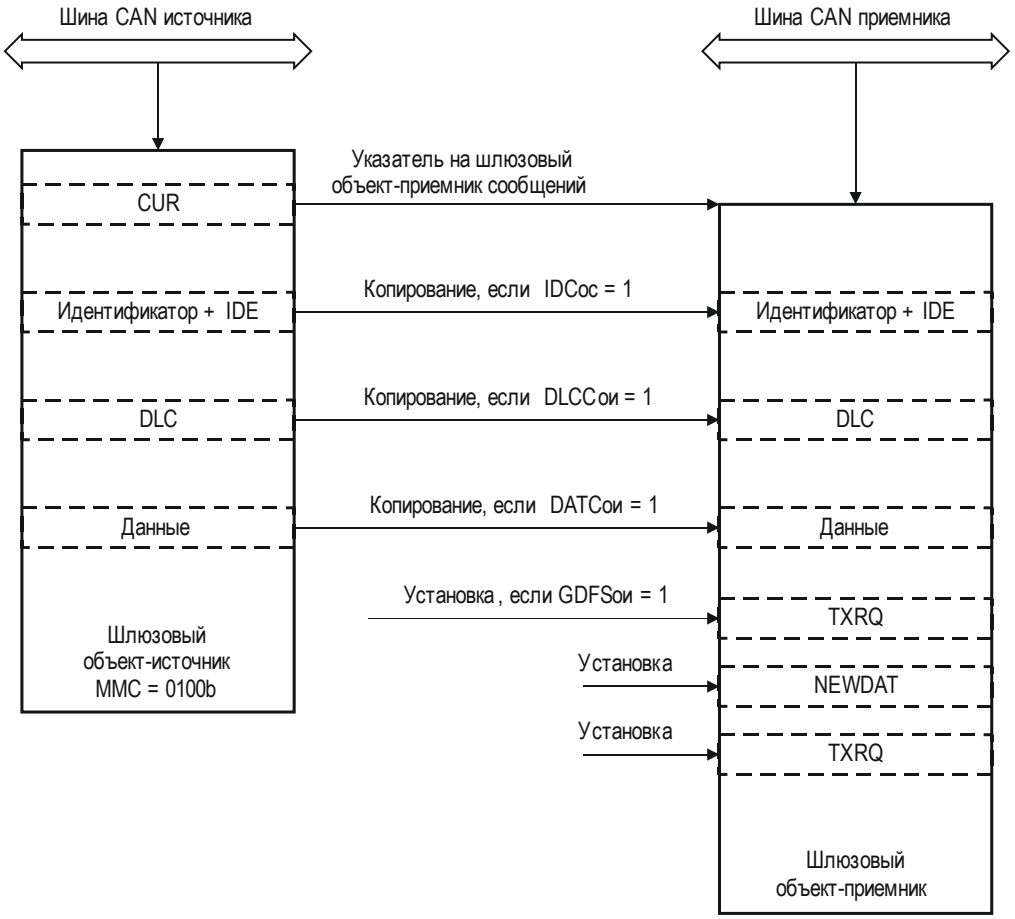


Рисунок 38.24 – Передача через шлюз от источника к приемнику

4 Если установлен флаг GDFS объекта-источника, то устанавливается бит запроса передачи TXRQ объекта-приемника.

5 Устанавливаются флаги RXPND и NEWDAT регистра MOSTAT объекта-приемника.

6 Если установлен флаг RXIE регистра MOSTAT объекта-приемника, то генерируется запрос на прерывание.

7 Указатель CUR объекта-источника переводится на следующий объект-приемник по правилам FIFO структуры. Сформировать шлюз между объектом-источником и одним объектом-приемником (значение указателя CUR будет оставаться неизменным) возможно программированием:

TOP = BOT = CUR = номер объекта-приемника.

Организация шлюза «объект-источник – объект-приемник» аналогична организации FIFO структуры «базовый объект – вспомогательный объект», что указывает на возможность формирования шлюза с интегрированным FIFO приемником. При получении сообщения данных (объект-источник является объектом приема, т. е. его бит DIR сброшен) и при получении удаленного запроса (объект-источник является объектом передачи) через шлюз используется один и тот же механизм.

Несмотря на то, что механизм удаленных запросов работает независимо от типа объекта сообщения, он наиболее полезен при использовании шлюзов, для формирования удаленных запросов на шине шлюзового объекта-источника после получения удаленного запроса на шине шлюзового объекта-приемника. В зависимости от значения бита FRREN шлюзового объекта-приемника, есть два варианта обработки удаленного запроса, возникшего с той стороны шлюза, где расположен объект-приемник (при условии, что происходит передача из объекта-источника в объект-приемник, т. е. DIR (источника) = 0 и DIR (приемника) = 1):

1 Обработка запроса шлюзового объекта-приемника с FRREN = 0b:

- сообщение удаленного запроса принимается шлюзовым объектом-приемником;
- бит TXRQ шлюзового объекта-приемника устанавливается автоматически;
- сообщение данных с текущей информацией, хранящейся в объекте-приемнике, передается на шину приемника.

2 Обработка запроса шлюзового объекта-приемника с FRREN = 1b:

- сообщение удаленного запроса принимается шлюзовым объектом-приемником;
- бит TXRQ шлюзового объекта-источника (объект должен быть указан в поле CUR объекта-приемника), устанавливается автоматически;
- сообщение данных передается объектом-источником на шину CAN источника;
- получатель удаленного запроса в ответ выдает сообщение данных на шину источника;
- сообщение данных сохраняется в объекте-источнике;
- сообщение данных копируется в объект-приемник (через шлюз);
- выставляется бит TXRQ объекта-приемника (при условии, что GDFS источника = 1);
- новые данные, сохраненные в объекте-приемнике, передаются на шину приемника, в ответ на удаленный запрос на шине приемника.

Рекомендации по записи в регистры в режиме шлюза при передаче удаленного запроса с FRREN = 1

Обмен запрос – данные происходит в данном случае между стандартным объектом сообщения одного узла и объектом-приемником шлюза другого узла. Но при этом данные для ответа на запрос в шлюзовый объект-приемник поступают по шлюзу от объекта-источника. При получении удаленного запроса от объекта сообщения объектом-приемником флаг TXRQ устанавливается не у самого объекта-приемника, а у объекта-источника, благодаря установленному биту FRREN и битовому полю CUR (указывает на объект-источник) объекта-приемника. Данные из MODATAL и MODATAH объекта-источника копируются в MODATAL и MODATAH объекта-приемника (установлен бит DATC регистра MOFCR объекта-источника), вследствие чего автоматически устанавливается бит TXRQ регистра MOCTR объекта-приемника (установлен бит GDFS объекта-источника шлюза), и осуществляется передача сообщения данных (ответ на запрос) запрашивающему объекту сообщения.

После успешного приема/передачи сообщения ЦП получает уведомление о завершении операции для задания дальнейших действий, связанных с объектом сообщения.

38.11 Прерывания объектов сообщений

После сохранения принятого сообщения в объект сообщения или успешной передачи формируется соответствующее прерывание. Каждый объект сообщения может формировать прерывания. Каждое прерывание направляется на одну из 16 выходных линий прерываний. Прерывания приема (после сохранения сообщения) также формируются после операций FIFO и шлюзовых операций. Флаги TXPND и RXPND всегда устанавливаются после успешной операции передачи/приема, независимо от состояния соответствующих флагов разрешения прерываний.

Объект сообщения может формировать FIFO прерывания. Если флаг OVIE регистра MOFCR установлен, то формирование FIFO прерывания будет зависеть от типа объекта сообщений, см. рисунок 38.25:

- если объект сообщения является принимающим базовым объектом, то выходная линия прерываний для этого объекта определяется битовым полем TXINP регистра MOIPR;
- если объект сообщения является передающим базовым объектом, то выходная линия прерываний определяется битовым полем RXINP.

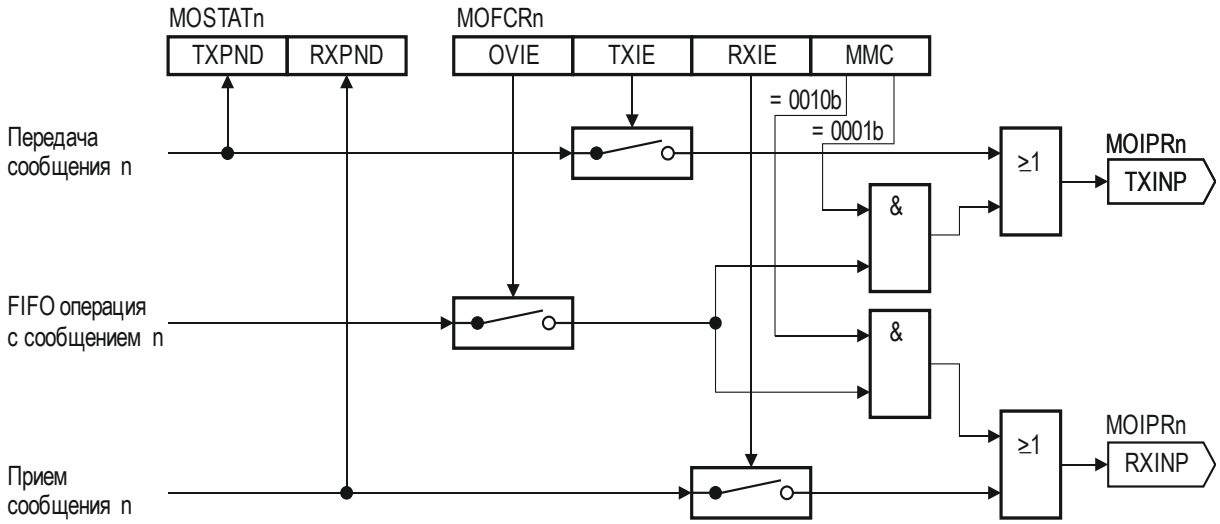


Рисунок 38.25 – Распределение прерываний

Ожидающие обработки сообщения

Когда генерируется запрос на прерывание (после приема/передачи сообщения), в одном из восьми регистров ожидающих обработки прерываний MSPNDx (x от 0 до 7) выставляется флаг ожидающего обработки сообщения. Восемь регистров образуют область из 32 × 8 бит – по два бита (один бит для операций приема и один бит для операций передачи) для каждого из объектов сообщений. Позиция флага ожидающего обработки сообщения определяется демультиплексорами DMUX, см. рисунки 38.26 и 38.27.

В зависимости от значения поля MPSEL регистра MCR, реализуется один из двух режимов выбора и установки флагов, ожидающих обработки сообщения:

- режим 1 в случае MPSEL = 0h;
- режим 2 в случае MPSEL = Fh.

Если нет необходимости в определении источника прерывания (прием или передача сообщения), то можно использовать любой из двух режимов, в противном случае, следует использовать второй режим.

В первом режиме установка флага ожидающего обработки сообщения происходит следующим образом:

- 7, 6 и 5 биты поля MPN выбирают регистр MSPNDx, в котором будет установлен флаг ожидающего обработки сообщения;
- пять младших бит поля MPN (на рисунке 38.26 выделены серым цветом) выбирают позицию флага (от 0 до 31), который будет установлен в выбранном регистре MSPNDx.

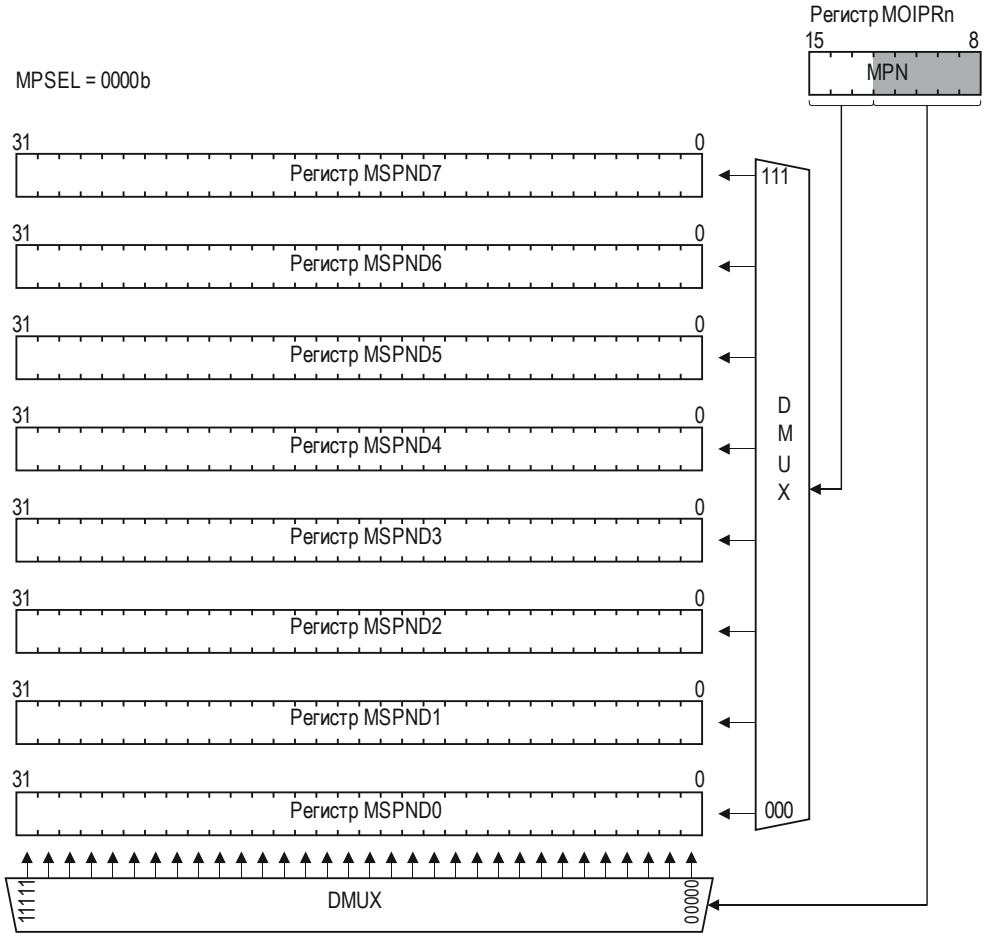


Рисунок 38.26 – Режим выбора и установки флагов при MPSEL = 0h

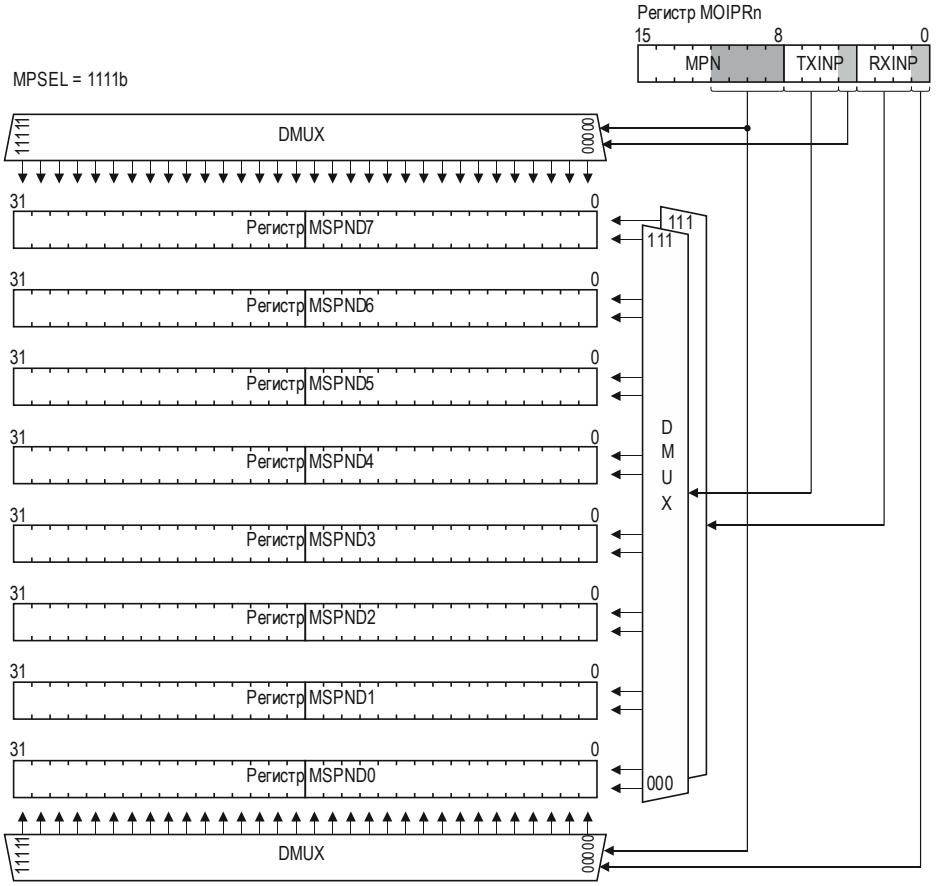


Рисунок 38.27 – Режим выбора и установки флагов при MPSEL = Fh

Во втором режиме при определении позиции флага ожидающего обработки сообщения принимаются в расчет значения поля MPN, полей RXINP (для приема) и TXINP (для передачи). При этом для флагов могут использоваться любые биты выбранного регистра MSPNDx. Установка флага ожидающего обработки сообщения происходит следующим образом:

- 3, 2 и 1 биты полей TXINP/RXINP выбирают регистр MSPNDx, в котором будет установлен флаг по окончанию передачи/приема сообщения;

- четыре младших бита поля MPN (на рисунке 38.27 выделены серым цветом) совместно с нулевыми битами полей TXINP и RXINP выбирают позицию флага (от 0 до 31). Фактически нулевой бит полей TXINP/RXINP выбирает старшее или младшее слово выбранного регистра MSPNDx, а четыре бита поля MPN задают позицию в выбранном слове.

Регистры MSPNDx могут быть записаны программно. Биты, в которые записываются единицы, остаются без изменений, а биты, в которые записываются нули, очищаются. Такой механизм записи позволяет избежать конфликта между одновременной аппаратной установкой и программной очисткой битов регистра.

Каждый регистр MSPNDx связан с соответствующим регистром индекса сообщения MSIDx, который отражает позицию самого младшего бита из всех установленных в регистре MSPNDx. Регистры MSIDx доступны только для чтения и обновляются незамедлительно после изменения (как аппаратного, так и программного) содержимого соответствующих регистров MSPNDx.

Регистр маски индекса сообщения MSIMASK содержит маску для регистров MSPNDx. Только незакрытые маской биты могут обслуживаться. Регистр MSIMASK используется одновременно для всех регистров MSPNDx и соответствующих им регистров MSIDx.

38.12 Программирование контроллера CAN

Для корректной работы контроллера CAN следует соблюдать порядок программирования регистров.

Для запуска контроллера CAN необходимо:

- записать регистр CLC;
- проверить, что сброшен бит DISR, регистр PANCTR = 0000_0000h, и после этого записать регистр FDR.

Далее для конфигурирования узла CAN с номером x (x от 0 до 3) выполнить:

- в регистре узла NCRx установить биты INIT и CCE, после чего регистры NBTRx и NPCRx станут доступны для записи и чтения, а регистр NECNTx – только для чтения;
- записать регистр NPCRx;
- записать регистр NIPRx;
- записать регистр NBTRx;
- записать регистр NFCRx (если необходимо);
- в регистре NCRx сбросить биты INIT и CCE, после чего регистры NBTRx и NPCRx будут не доступны для записи;
- распределить объекты сообщений в списки посредством регистра PANCTR.

Для корректной работы объектов сообщений регистры каждого из них должны быть проинициализированы. Для объектов, использование которых не предусматривается, достаточно записать ноль в бит MSGVAL регистра MOCTR.

Рекомендуемый порядок инициализации регистров объекта сообщения:

- установить бит DIR в регистре MOSTAT для передачи сообщения данных/приема удаленного запроса или сбросить бит DIR для приема сообщения данных/передачи удаленного запроса; установить биты TXEN0 и TXEN1 (для передачи) или RXEN (для приема) в регистре MOCTR;
- записать регистр MOFCR;
- записать регистр MOAR;
- записать регистр MOAMR (если необходимо);
- записать регистр MOFGPR (если будут использоваться FIFO структуры);
- записать регистр MOIPR;
- записать регистры MODATAL и MODATAH;
- установить бит MSGVAL корректности объекта сообщения в регистре MOCTR (для неиспользуемых объектов этот бит должен быть сброшен);
- для активирования передачи установить бит TXRQ регистра MOCTR.

38.13 Пример расчета скорости передачи 1Мбит/с

При частоте f_{FOUT} равной 50 или 100 МГц записью подходящих значений полей регистра NBTR можно сконфигурировать скорость передачи в точности равную 1 Мбит/с.

Для примера выберем f_{FOUT} = 50 МГц. Если системная частота f_{FIN} (HCLK) составляет 100 или 50 МГц, то для получения f_{FOUT} = 50 МГц хорошо подойдет режим нормального деления (DM = 1) с значением поля STEP регистра FDR 3FEh и 3FFh соответственно (см. пункт «Дробный делитель» подраздела 38.2).

Скорости передачи 1 Мбит/с соответствует Tbit = 1000 нс. Бит может быть сформирован только из целого количества квантов времени, лежащего в диапазоне от 4 (рекомендуется от 8) до 25. Длительность одного кванта времени кратна 1 / f_{FOUT} и величине предделителя скорости передачи (BRP + 1). Для f_{FOUT} = 50 МГц длительность кванта времени может быть 20 нс, 40 нс (BRP = 1), 60 нс, 80 нс, 100 нс (BRP = 4) и так далее. Для получения целого количества квантов времени, лежащего в заявленных пределах, получаем три варианта значения предделителя скорости передачи (BRP + 1) – 2, 5 и 10.

При предделителе скорости передачи равном 10 (BRP = 9) в бите будет 5 квантов времени (меньше рекомендованного значения), которые можно распределить всего одним способом Tseg1 = 3 × tq, Tseg2 = 1 × tq, что сильно ограничит максимальную ширину прыжка повторной синхронизации.

При предделителе скорости передачи равном 2 (BRP = 1) в бите будет 25 квантов времени (максимальное доступное для модуля значение), которые также можно распределить всего одним способом Tseg1 = 16 × tq, Tseg2 = 8 × tq, что не позволит сместить точку выборки. Слишком большое количество квантов времени в бите также несколько ограничит максимально допустимое отклонение частоты сигнала FOUT (см. формулу 38.9).

При предделителе скорости передачи равном 5 (BRP = 4) в бите будет 10 квантов времени. Этот вариант можно считать наиболее удачным, поскольку он позволяет как настроить положение момента выборки, так и позволит иметь высокую толерантность к отклонению частот задающих генераторов узлов сети CAN. Одним из вариантов распределения является Tseg1 = 6 × tq (TSEG1 = 5), Tseg2 = 3 × tq (TSEG2 = 2). При SJW равном 2 значение регистра NBTR будет равно 00002584h.

39 Контроллер интерфейса CAN FD

39.1 Область применения

CAN (Controller Area Network) является широко используемым последовательным коммуникационным протоколом, который нашел широкое применение в различных отраслях, таких как автомобилестроение, промышленная автоматизация и другие. Для современных систем, требующих более высокую пропускную способность и увеличенный размер полезной нагрузки, был разработан протокол CAN FD (CAN with Flexible Data-Rate).

Управляющие регистры контроллера CAN FD приведены в приложении А.32.

CAN FD сохраняет совместимость с существующими CAN-системами, но при этом предоставляет дополнительные возможности, такие как:

– Увеличение максимальной скорости передачи данных до 10 Мбит/с;

– Увеличение максимального размера полезной нагрузки до 64 байт;

– Улучшенные механизмы обнаружения и обработки ошибок.

39.2 Основные характеристики

Микроконтроллер содержит четыре контроллера CANFD (CANFD0 – CANFD3), по одному узлу в каждом.

Контроллер CAN обладает следующими характеристиками:

1 Полная совместимость с протоколами CAN 2.0 A и CAN 2.0 B, поддержка CAN FD (ISO и без ISO);

2 16 объектов сообщений (блок 256x32b SP SRAM с кодом коррекции Хэмминга ECC32) у каждого из которых:

– Поддержка 11-битного тэга идентификатора (для CAN 2.0 A), и 29-битного тэга идентификатора (для CAN 2.0 B);

– Поддержка 11-битной маски идентификатора (для CAN 2.0 A), и 29-битной маски идентификатора (для CAN 2.0 B);

– Буфер данных на 64 байта (со статическим размещением);

– Конфигурации Tx, Rx, буфер кадров или автоответ;

– Счетчик временной метки;

3 Максимальная скорость передачи данных 1 Мбит/с на частоте 8 МГц для CAN 2.0 A и CAN 2.0 B;

4 Максимальная скорость передачи данных 10 Мбит/с на частоте 80 МГц для CAN FD;

5 Таймер TTC;

6 Режим прослушивания (для слежения или автоподстройки).

39.3 Описание работы CAN

В контроллере реализованы протоколы CAN 2.0 или CAN Base, и CAN FD, который имеет два варианта реализации – более современный ISO и non-ISO. Переключение режима, соответствующего одному из протоколов, осуществляется за счёт настройки регистра FD_CTRL.

CAN протокол поддерживает 4 типа сообщений (frame):

– Data (данные) – это наиболее часто используемый тип сообщения.

– Remote (удаленный запрос) –это Data Frame без поля данных и с выставленным битом RTR (1 – рецессивный бит). Основное предназначение Remote-кадра – это инициация одним из узлов сети передачи в сеть данных другим узлом. Такая схема позволяет уменьшить суммарный трафик сети.

– Error (ошибка) – это сообщение, которое явно нарушает формат сообщения CAN.

Передача такого сообщения приводит к тому, что все узлы сети регистрируют ошибку формата CAN-кадра, и в свою очередь автоматически передают в сеть Error Frame. Результатом этого процесса является автоматическая повторная передача данных в сеть передающим узлом. Error Frame состоит из поля Error Flag, которое состоит из 6 бит одинакового значения (и таким образом Error frame нарушает проверку Bit Stuffing, см. ниже), и поля Error Delimiter, состоящее из 8 рецессивных битов. Error Delimiter дает возможность другим узлам сети обнаружив Error Frame послать в сеть свой Error Flag.

– Overload (Переполнение) – повторяет структуру и логику работы Error-кадра, с той разницей, что он используется перегруженным узлом, который в данный момент не может обработать поступающее сообщение, и поэтому просит при помощи Overload-кадра о повторной передаче данных.

Примечание – В режиме CAN FD тип сообщения Remote (удаленный запрос) не поддерживается.

В реализации контроллера присутствуют протоколы стандартного (ver. A) и расширенного CAN 2.0.(ver. B). Поддерживаемые форматы сообщений протоколов приведены ниже.

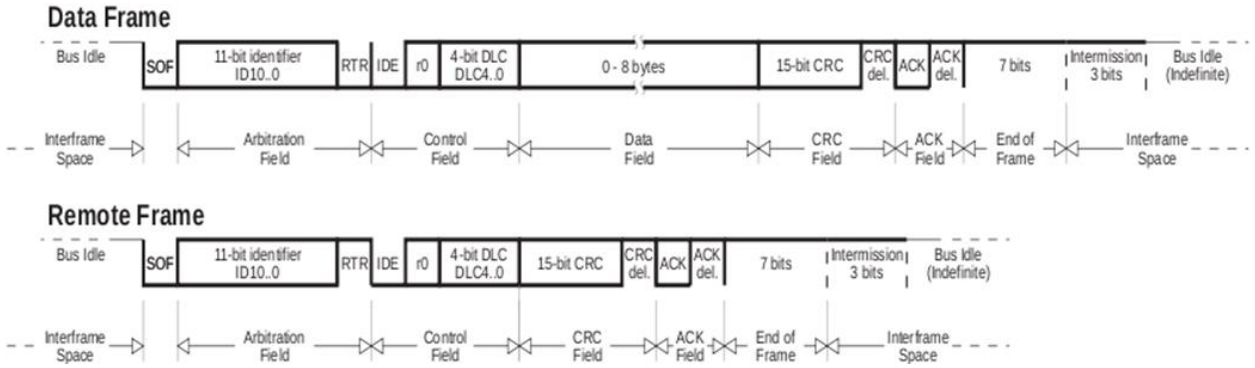


Рисунок 39.1 – Форматы сообщений Data и Remote для стандартного CAN

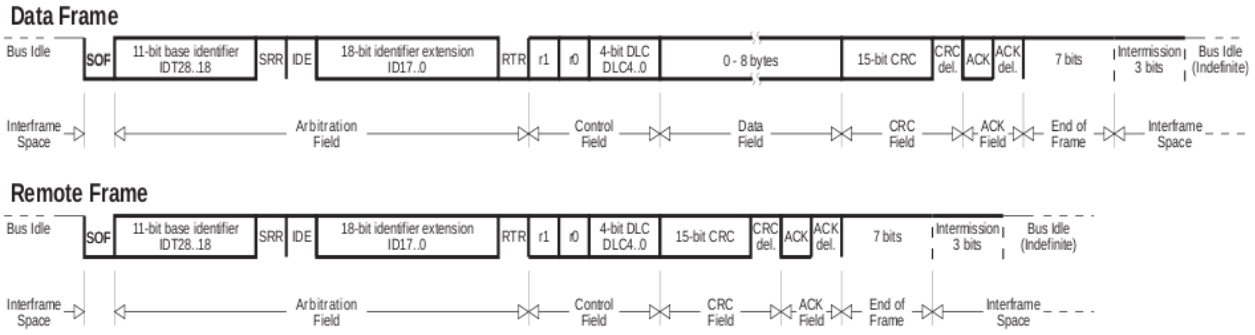


Рисунок 39.2 – Форматы сообщений Data и Remote для расширенного CAN

Стандартный и расширенный протоколы также поддерживаются и в CANFD режиме. Форматы сообщений CANFD в сравнении с CAN 2.0 приведены ниже.

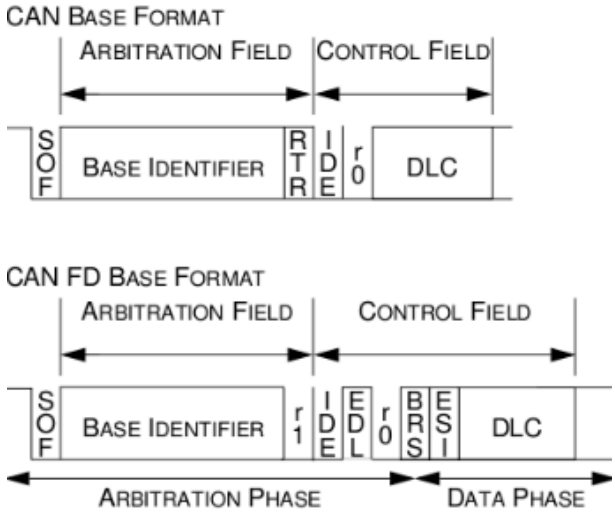


Рисунок 39.3 – Стандартный формат CAN и CANFD (поле арбитража и контроля)

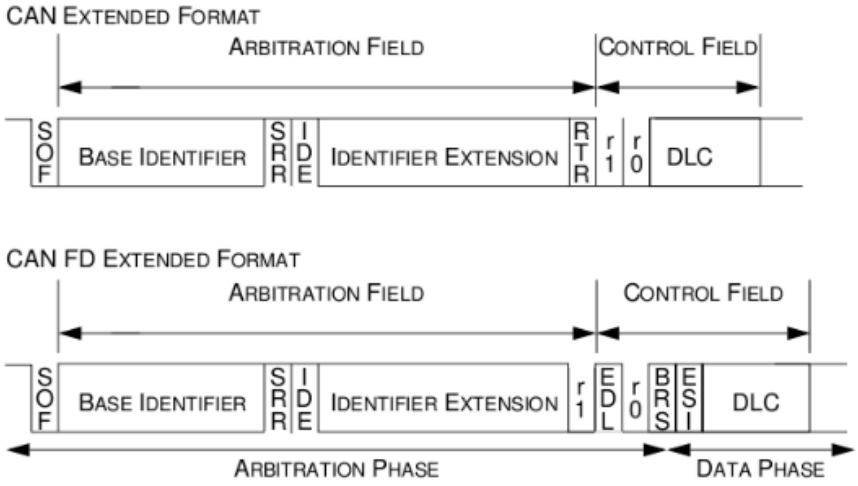


Рисунок 39.4 – Расширенный формат CAN и CANFD (поле арбитража и контроля)

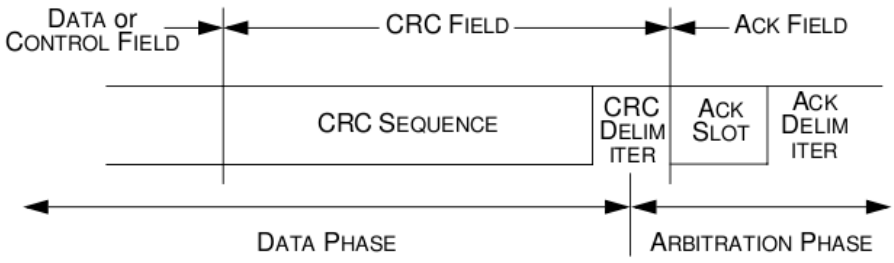


Рисунок 39.5 – Формат окончания кадра в CAN

Формат сообщений в CAN FD также имеет отличия в зависимости от версии протокола. В более современной реализации ISO присутствуют дополнительное поле SBC (счетчик битов) и расширенное поле CRC Sequence.

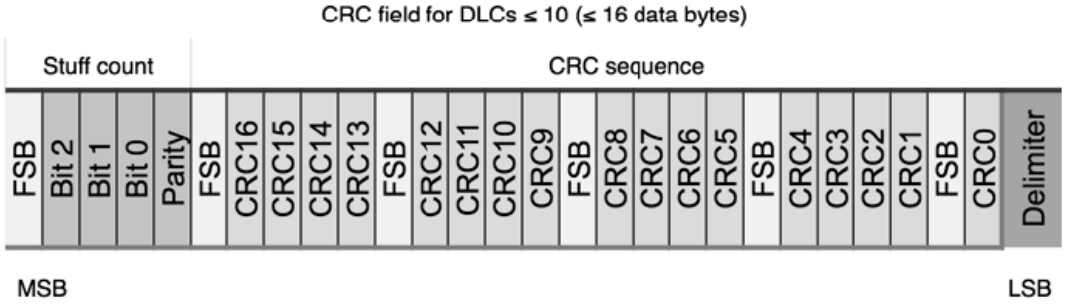


Рисунок 39.6 – Поле CRC CAN FD в стандарте ISO

Кадры данных и кадры удаленного запроса отделены от предыдущих кадров межкадровым промежутком (Interframe space). В этот промежуток входит 3-х битное поле “Intermission”, обязательное для каждого кадра.

Передача каждого поля кадра начинается со старшего бита (MSB). При передаче поля данных больше 1 байта, отправка начинается с младшего байта (биты [7:0]) и заканчивается старшим (в случае 32-х битного слова биты [31:24]).

Поскольку поле DLC (Data length code), содержащее в себе количество бит в сообщении, ограничено 4 битами, для передачи 64 байт в режиме CAN FD существует таблица кодировки DLC (см. таблицу 39.1).

Таблица 39.1 – Кодировка поля длины данных для разных режимов CAN

Количество байт		Код длины данных			
CAN2.0	CAN FD	DLC3	DLC2	DLC1	DLC0
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
8	12	1	0	0	1
8	16	1	0	1	0
8	20	1	0	1	1
8	24	1	1	0	0
8	32	1	1	0	1
8	48	1	1	1	0
8	64	1	1	1	1

Передача 1 бита данных в сети CAN занимает «номинальное время передачи», которое можно разделить на несколько не перекрывающихся участков:

– SYNC_SEG (Synchronization Segment – сегмент синхронизации) – Первый сегмент используется, чтобы синхронизировать различные узлы шины. При передаче, в начале этого сегмента, анализируется текущий уровень бита. Если есть разница между предыдущим битом и текущим битом, то изменение состояния шины произойдет в пределах этого сегмента.

– PROP_SEG (Propagation Time Segment – сегмент времени распространения) – Этот сегмент используется, чтобы компенсировать задержки сигнала через сеть. Это необходимо, чтобы компенсировать задержки распространения сигнала на линии и через приемопередатчики узлов шины.

– PHASE_SEG_1 (Phase Segment 1 – сегмент TSEG1) – Phase Segment 1 используется, чтобы компенсировать ошибки фазы. Этот сегмент может быть удлинен во время пересинхронизации.

– PHASE_SEG_2 (Phase Segment 2 – сегмент TSEG2) – Этот сегмент также используется, чтобы компенсировать ошибки фазы. Этот сегмент может быть сокращен во время пересинхронизации.

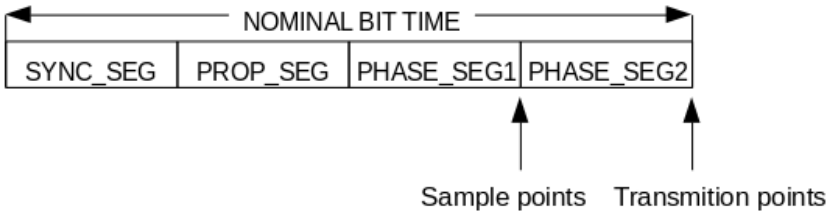


Рисунок 39.7 – Общее представление фаз номинального времени передачи

Переход между TSEG1 и TSEG2 называется точкой считывания (sample point). Точка считывания – точка времени, в которой уровень шины читается и интерпретируется, как величина соответствующего бита.

Каждый участок номинального времени передачи занимает определённое количество единиц квантов времени или же «шаг квантования времени». Шаг квантования времени – фиксированная единица времени, полученная из периода генератора. Существует программируемый делитель со значениями от 1 до 32.

При старте с минимальным шагом квантования времени, шаг квантования времени может иметь длину:

$$c = м * t \quad (39.1)$$

Где c – шаг квантования;

м – значение делителя;

t – минимальный квант времени. t = 1 / f, где f – частота внешнего генератора.

Длительность участков номинального времени передачи:

– SYNC_SEG – 1 шаг квантования времени.

– PROP_SEG – программируется, может быть 1, 2, ..., 8 квантов времени.

– TSEG1 – программируется, может быть 1, 2, ..., 8 квантов времени.

– TSEG2 – максимум из PHASE_SEG1 и времени обработки информации.

– Время обработки информации – меньше или равно 2 квантам времени.

Общее число квантов времени в битовом интервале должно быть программируемым, по крайней мере, от 8 до 25.

Контроллер CAN поддерживает пять способов обнаружения ошибок в сети:
– Bit monitoring;
– Bit stuffing;
– Frame check;
– ACKnowledgement Check;
– CRC Check.

Bit monitoring – каждый узел во время передачи битов в сеть сравнивает значение передаваемого им бита со значением бита, которое появляется на шине. Если эти значения не совпадают, то узел генерирует ошибку Bit Error. Естественно, что во время арбитража на шине (передача поля арбитража в шину) этот механизм проверки ошибок отключается.

Bit stuffing – когда узел передает последовательно в шину 5 бит с одинаковым значением, то он добавляет шестой бит с противоположным значением. Принимающие узлы этот дополнительный бит удаляют. Если узел обнаруживает на шине больше 5 последовательных бит с одинаковым значением, то он генерирует ошибку Stuff Error.

Frame Check – некоторые части CAN-сообщения имеют одинаковое значение во всех типах сообщений. Т.е. протокол CAN точно определяет какие уровни напряжения и когда должны появляться на шине. Если формат сообщений нарушается, то узлы генерируют ошибку Form Error.

ACKnowledgement Check – каждый узел получив правильное сообщение по сети посылает в сеть доминантный (0) бит. Если же этого не происходит, то передающий узел регистрирует ошибку Acknowledgement Error.

CRC Check – каждое сообщение CAN содержит контрольную сумму CRC, и каждый принимающий узел подсчитывает значение CRC для каждого полученного сообщения. Если подсчитанное значение контрольной суммы CRC, не совпадает со значением CRC в сообщении, принимающий узел генерирует ошибку CRC Error.

39.4 Описание сигналов тактирования и сброса

Тактирование CAN осуществляется за счёт двух входных внешних синхросигналов CANFD_CLK и HCLK.

Первый сигнал является тактовым для самого контроллера, его внутренней логики и передачи по каналу CAN (с учетом делителей частоты), второй сигнал является тактовым для шины AHB, через которую может осуществляться обмен данными с CAN и его настройка.

Для контроллера CANFD существуют ограничения по тактированию:

– Для достижения максимальной скорости передачи формата FD частота CANFD_CLK должна быть не менее 80 МГц;

– Для реализации формата CAN 2.0 минимальная частота CANFD_CLK должна быть не менее 8 МГц;

Скорость передачи кадра рассчитывается исходя из значений делителей опорной частоты для полей CAN2.0 (биты BRP регистра BITTIMING) и полей CAN FD (биты BRFD регистра FDBITTIMING) кадра. Помимо делителей частоты в контроллере определяется количество квант фаз номинального времени передачи бита, которые применены к битам передаваемых полей CAN 2.0 (см. регистр BITTIMING) и полей CANFD (см. регистр FDBITTIMING) кадра. 1 квант фазы занимает 1 такт, полученный в результате деления опорной частоты значениями битовых полей BRP или BRFD.

Формула для полей CAN 2.0:

$$T = Tclk * (BRP + 1) * n \quad (39.2)$$

Для полей CANFD:

$$T = Tclk * (BRFD + 1) * n \quad (39.3)$$

Где T – период передачи 1 бита в CAN;
Tclk – период опорной частоты;
n – общее количество квантов фаз номинального времени передачи.

Примечание – Если значения BRP и BRFD равны, бит BRS будет иметь доминантный уровень.

Сброс контроллера CANFD осуществляется путем записи единицы в бит SWRES регистра CONTROL.

39.1 Структура регистров

Регистры контроллера CAN делятся на две категории:

- Регистры общего назначения;
- Регистры объектов сообщений (MObs).

Первая группа регистров отвечает за настройку всего контроллера, вторая отвечает за настройки конкретного MOb и для каждого есть свой набор регистров. Переключение между отображением регистров выбранного объекта сообщений осуществляется с помощью регистра общего назначения MOBPAGE.

Регистровая структура CAN FD представлена на рисунке 39.8.

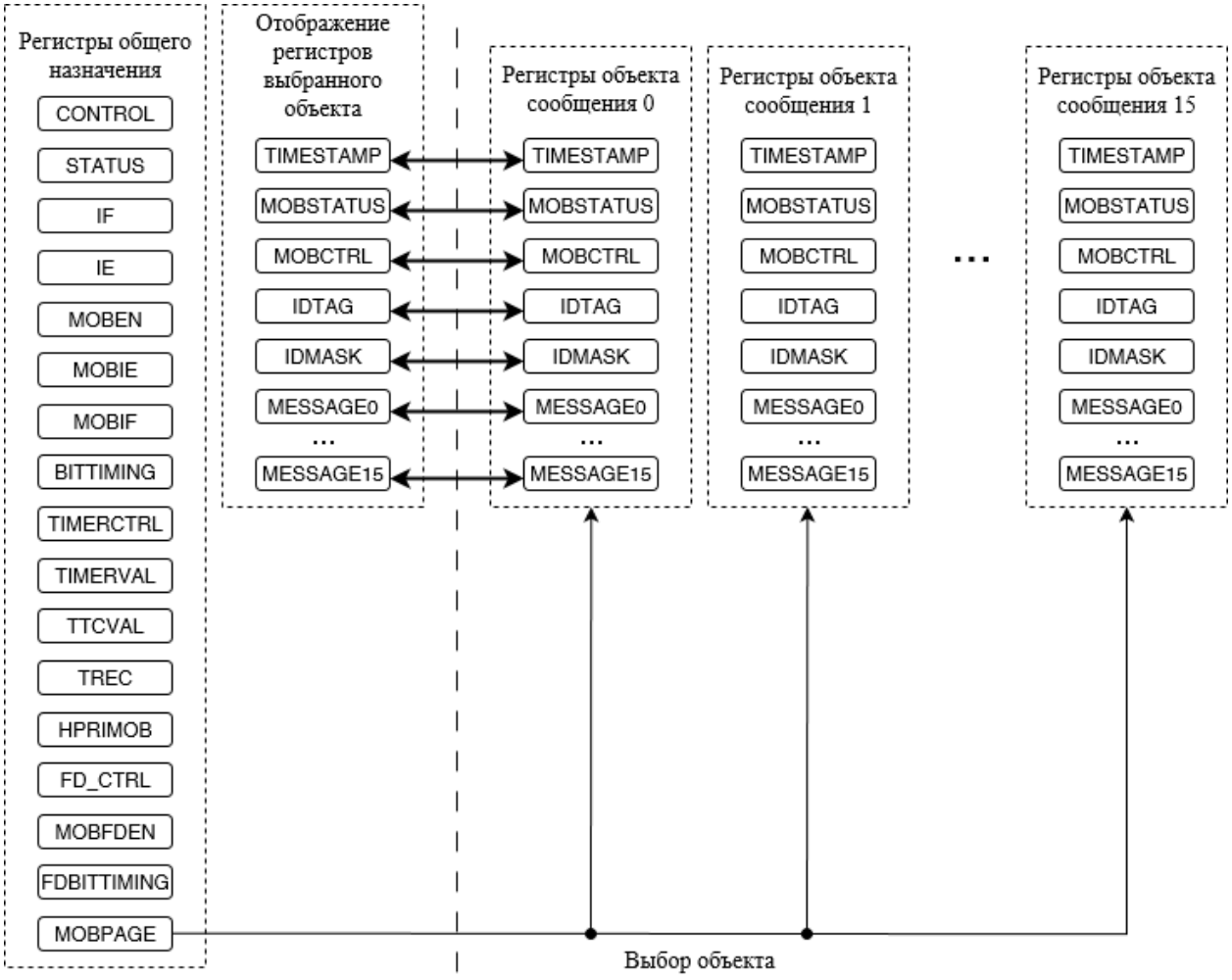


Рисунок 39.8 – Структура регистров CANFD

39.2 Инструкция по программированию

Рекомендации по программированию основного функционала

В данной инструкции будет рассмотрен вариант программирования двух модулей CAN (1 – передатчик, 2 – приемник). Для реализации основного функционала CAN рекомендуется следовать следующей последовательности программирования:

1) Разрешение тактирования и вывод из состояния сброса регистровой части модулей CANFD путем установки бит CANFDEN в регистрах CGCFGAHB и RSTDISAHB блока RCU. Выбор источника тактирования, разрешение тактирования и вывод из сброса модулей CANFD путем установки значения поля CLKSEL, а также установки бит CLKEN, RSTDIS регистра CANFDCFG блока RCU.

2) Подготовка данных на передачу.

– Установка бита FDMOD в регистре FD_CTRL модуля CAN1 для настройки доступа к памяти контроллера в режиме FD;

– Запись в поле MOBNB регистра MOBPAGE модуля CAN1 значения от 0 до 15, указывающее номер текущего объекта сообщения (MOb) (значение по умолчанию – 0, MOb_0). Запись 16-ти 32-х битных слов в память MESSAGE модуля CAN1.

3) Настройка делителя частоты обоих модулей CAN.

– Запись в регистр BITTIMING значения делителя частоты, с которой будут передаваться кадры формата CAN 2.0 (поле BRP), значения количества квантов времени для фазы номинального времени передачи PROP_SEG (поле PTS), значения бит SJW (поле SJW), значения бита опции точки выборки SMP (бит SMP), значений количества квантов времени для фаз номинального времени передачи PHASE_SEG1 (поле PHS1) и PHASE_SEG2 (поле PHS2);

– Запись в регистр FDBITTIMING значения делителя частоты, с которой будут передаваться поля кадра в режиме FD (поле BRFD), значения количества квантов времени для фазы номинального времени передачи PROP_SEG (поле PTS), значения бит SJW (поле SJW), значений количества квантов времени для фаз номинального времени передачи PHASE_SEG1 (поле PHS1) и PHASE_SEG2 (поле PHS2).

4) Включение режима FD.

– Установка в регистрах FD_CTRL обоих модулей CAN битов FDMOD (а также дополнительно EXMOD для ISO).

5) Настройка объектов сообщений MOb модуля CAN2.

– Запись в поле MOBNB регистра MOBPAGE модуля CAN2 значения от 0 до 15, указывающее номер текущего объекта сообщения MOb;

– Запись в поле CONMOB регистра MOBCTRL модуля CAN2 значения 0x2 (приём), в бит IDE 0 или 1 (формат обычный или расширенный соответственно), в поле DLC размер ожидаемого кадра (см. таблицу 39.1);

– Запись единиц в биты регистра MOBEN модуля CAN2, которые соответствуют номеру объекта сообщения MOb, тем самым настраивая отдельно каждый MOb на FD режим.

6) Настройка объектов сообщений MOb модуля CAN1.

– Запись в поле MOBNB регистра MOBPAGE модуля CAN1 значения от 0 до 15, указывающее номер текущего объекта сообщения MOb;

– Запись идентификатора в поле IDT регистра IDTAG модуля CAN1, бит RTRTAG должен быть сброшен;

– Запись в поле CONMOB регистра MOBCTRL модуля CAN1 значения 0x1 (передача), установить или сбросить бит IDE (формат обычный или расширенный соответственно), в поле DLC размер ожидаемого кадра (значения DLC). Если передаётся Remote frame, поле DLC не записывается;

– Запись единиц в биты регистра MOBEN модуля CAN1, которые соответствуют номеру объекта сообщения MOb, тем самым настраивая отдельно каждый MOb на FD режим.

7) Включение модуля CAN2.

– Установка бита ENA в регистре CONTROL модуля CAN2.

8) Разрешение прерывания по приему модулем CAN2 и прерываний объектов сообщений MOb (всех).

– Установка бит ENRX и ENIT в регистре IE, а также бит IEMOB0-IEMOB15 в регистре MOBIE модуля CAN2.

9) Разрешение прерывания по передаче модулем CAN1 и прерываний объектов сообщений MOb (всех).

– Установка бит ENTX и ENIT в регистре IE, а также бит IEMOB0-IEMOB15 в регистре MOBIE модуля CAN1.

10) Включение модуля CAN1.

– Установка бита ENA в регистре CONTROL модуля CAN1.

11) Ожидание прерывания.

12) Снятие флага прерывания CAN1.

– Запись 1 в бит TXOK регистра MOBSTATUS модуля CAN1.

13) Снятие флага прерывания CAN2.

– Запись 1 в бит RXOK регистра MOBSTATUS модуля CAN2.

14) Программный сброс Soft-reset (опционально).

– Установка бита SWRES в регистрах CONTROL обоих модулей CAN или сброс бита ENA в регистрах CONTROL обоих модулей CAN (выключение контроллеров).

Рекомендации по программированию функции автоответа на REMOTE запрос

Для реализации функции автоответа на удаленный REMOTE запрос рекомендуется следовать следующей последовательности программирования:

1) Разрешение тактирования и вывод из состояния сброса регистровой части модулей CANFD путем установки бит CANFDEN в регистрах CGCFGAHB и RSTDISAHB блока RCU. Выбор источника тактирования, разрешение тактирования и вывод из сброса модулей CANFD путем установки значения поля CLKSEL, а также установки бит CLKEN, RSTDIS регистра CANFDCFG блока RCU.

2) Настройка делителя частоты обоих CAN.

– Запись в регистр BITTIMING значения делителя частоты, с которой будут передаваться кадры формата CAN 2.0 (поле BRP);

– Запись значения количества квантов времени для фазы номинального времени передачи PROP_SEG (поле PTS);

– Запись значения бит SJW (поле SJW);

– Запись значения бита опции точки выборки SMP (бит SMP);

– Запись значений количества квантов времени для фаз номинального времени передачи PHASE_SEG1 (поле PHS1) и PHASE_SEG2 (поле PHS2).

3) Отключение режима FD.

– Сброс бита FDMOD в регистрах FD_CTRL обоих модулей CAN.

4) Настройка объектов сообщений MOb модуля CAN2.

– Запись в поле MOBNB регистра MOBPAGE модуля CAN2 значения от 0 до 15, указывающее номер текущего объекта сообщения MOb;

– Запись в поле CONMOB регистра MOBCTRL модуля CAN2 значения 0x2, а также установить бит RPLV (приём в режиме автоответа на REMOTE кадр), в бит IDE 0 или 1 (формат обычный или расширенный соответственно);

– Установить бит RTRTAG регистра IDTAG.

5) Настройка объектов сообщения MOb модуля CAN1.

– Запись в поле MOBNB регистра MOBPAGE модуля CAN1 значения от 0 до 15, указывающее номер текущего объекта сообщения MOb;

– Запись идентификатора в поле IDT регистра IDTAG модуля CAN1, установка бита RTRTAG;

– Запись в поле CONMOB регистра MOBCTRL модуля CAN1 значения 0x1 (передача), установить или сбросить бит IDE (формат обычный или расширенный соответственно).

6) Включение модуля CAN2.

– Установка бита ENA в регистре CONTROL модуля CAN2.

7) Разрешение прерываний по приему и передаче модулем CAN2 и прерываний объектов сообщений MOb (всех).

– Установка бит ENTX, ENRX и ENIT в регистре IE, а также бит IEMOB0-IEMOB15 в регистре MOBIE модуля CAN2.

8) Разрешение прерываний по приему и передаче модулем CAN1 и прерываний от объектов сообщений MOb (всех).

– Установка бит ENTX, ENRX и ENIT в регистре IE, а также бит IEMOB0-IEMOB15 в регистре MOBIE модуля CAN1.

9) Включение модуля CAN1.

– Установка бита ENA в регистре CONTROL модуля CAN1.

10) Ожидание прерывания в CAN1 после отправки REMOTE запроса.

11) Снятие флага прерывания модуля CAN1.

– Запись 1 в бит TXOK регистра MOBSTATUS модуля CAN1.

12) Переключение модуля CAN2 на прием.

– Запись в поле MOBNB регистра MOBPAGE модуля CAN2 значения от 0 до 15, указывающее номер текущего объекта сообщения MOb;

– Запись в поле CONMOB регистра MOBCTRL модуля CAN2 значения 0x2 (приём), в бит IDE 0 или 1 (формат обычный или расширенный соответственно), в поле DLC размер ожидаемого кадра (см. таблицу 39.1);

– Запись единиц в биты регистра MOBEN модуля CAN2, которые соответствуют номеру объекта сообщения MOb, тем самым настраивая отдельно каждый MOb на FD режим.

13) Ожидание в CAN2 прерывания после ответа на REMOTE запрос.

14) Снятие флага прерывания модуля CAN2.

– Запись 1 в бит TXOK регистра MOBSTATUS модуля CAN2.

15) Ожидание в CAN1 прерывания после получения ответа на REMOTE запрос.

16) Снятие флага прерывания CAN1.

– Запись 1 в бит RXOK регистра MOBSTATUS модуля CAN1.

17) Программный сброс Soft-reset (опционально).

– Установка бита SWRES в регистрах CONTROL обоих модулей CAN или сброс бита ENA в регистрах CONTROL обоих модулей CAN (выключение контроллеров).

40 Контроллер Ethernet 10/100/1000

40.1 Общее описание

Контроллер Ethernet 10/100/1000 обеспечивает интерфейс между шиной AHB и сетью Ethernet. Он поддерживает скорость 10/100/1000 Мбит в режимах полного дуплекса и полудуплекса. Интерфейс AMBA состоит из интерфейса APB для управления и настройки и AHB интерфейса в режиме мастера, который обрабатывает поток данных. Поток данных обрабатывается с помощью каналов DMA. Имеются отдельный DMA для передатчика и отдельный для приемника. Оба используют один и тот же интерфейс AHB в режиме мастера.

Интерфейс Ethernet поддерживает интерфейсы MII и GMII, с помощью которых происходит подключение к внешнему PHY устройству. Контроллер Ethernet также обеспечивает доступ к интерфейсу управления MDIO, который используется для настройки PHY.

Управляющие регистры контроллера Ethernet приведены в приложении А.33.

Поддерживаемые функции для каналов DMA: Scatter Gather I/O (ввод/вывод в режиме разборка/сборка) и выгрузка контрольной суммы TCP/UDP по протоколу IPv4 как для приемника, так и для передатчика.

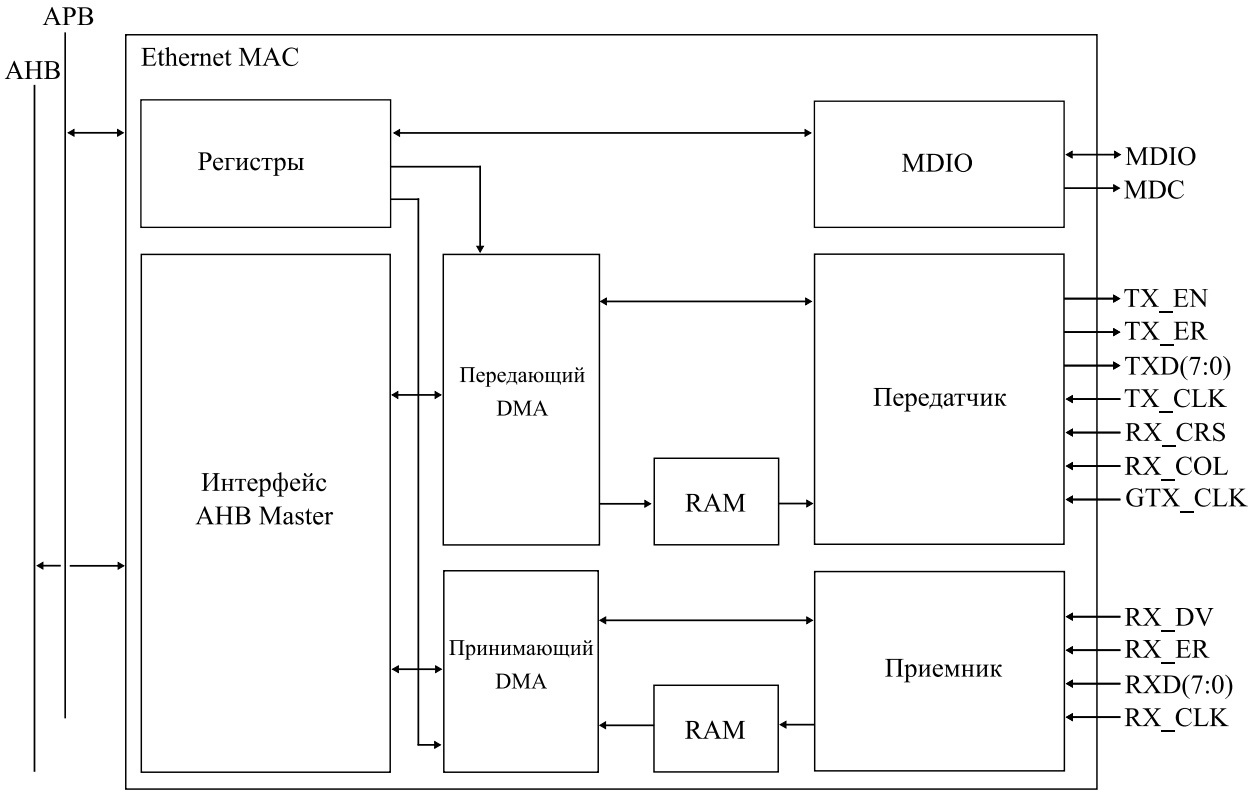


Рисунок 40.1 – Внутренняя структура контроллера Ethernet 10/100/1000

Имеется несколько интерфейсов между подуровнем MAC и физическим уровнем. Контроллер Ethernet поддерживает независимый от среды интерфейс (MII) и гигабитный независимый от среды интерфейс (GMII).

GMII используется в режиме 1000 Мбит, а MII – в режиме 10 и 100 Мбит. Эти интерфейсы определены раздельно в стандарте 802.3-2002, но на практике они совместно используют большинство сигналов. GMII имеет 9 дополнительных сигналов по сравнению с MII (см. таблицу 40.1). К приемнику и передатчику добавляются четыре сигнала передачи данных соответственно, а также вводится новая тактовая частота передачи для гигабитного режима (сигнал gtxclk).

Таблица 40.1 – Сигналы MII и GMII

Сигнал	MII	GMII	Выводы GPIO
txd[7:4]	–	+	E7-E4
txd[3:0]	+	+	E3-E0
rxd[7:4]	–	+	E15-E12
rxd[3:0]	+	+	E11-E8
txclk	+	–	D3
gtxclk	–	+	D4
rxclk	+	+	D5
rxdv	+	+	D8
rxer	+	+	D9
mdio	+	+	D10
txen	+	+	D11
txer	+	+	D12
rxcrs	+	+	D13
rxcol	+	+	D14
mdc	+	+	D15

Подключение к Ethernet 10/100/1000 PHY

Имеется возможность использования как внутреннего блока Ethernet PHY, так и внешней микросхемы Ethernet PHY, подключаемой с помощью выводов GPIO путем настройки альтернативных функций. На рисунке 40.2 изображена схема различных вариантов подключения блоков Ethernet PHY.

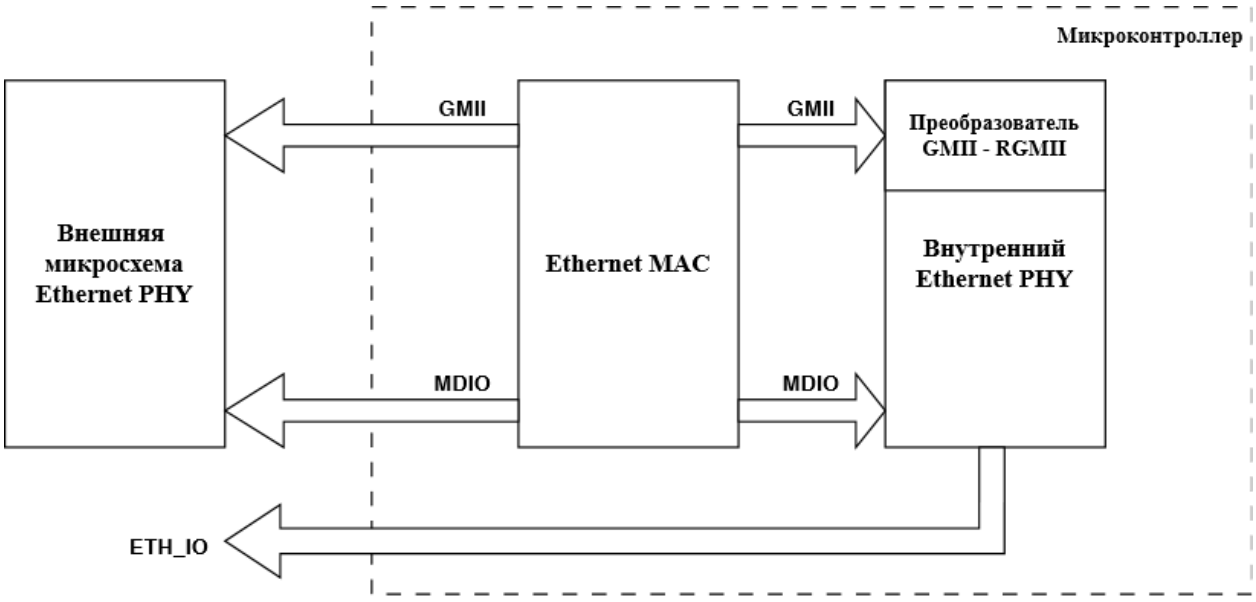


Рисунок 40.2 – Схема подключения Ethernet PHY

Переключение между выбором внутреннего блока Ethernet PHY и внешней микросхемой Ethernet PHY осуществляется с помощью бита EXTPHY регистра PHYCFG. Для работы с внешней микросхемой Ethernet PHY бит EXTPHY должен быть установлен, а также выводы GPIOD5-GPIOD15, GPIOE0-GPIOE15 должны быть настроены на четвертую альтернативную функцию.

Тактирование Ethernet

Настройка тактирования блока Ethernet PHY выполняется с помощью регистров блока RCU. Схема настройки тактирования представлена на рисунке 40.3.

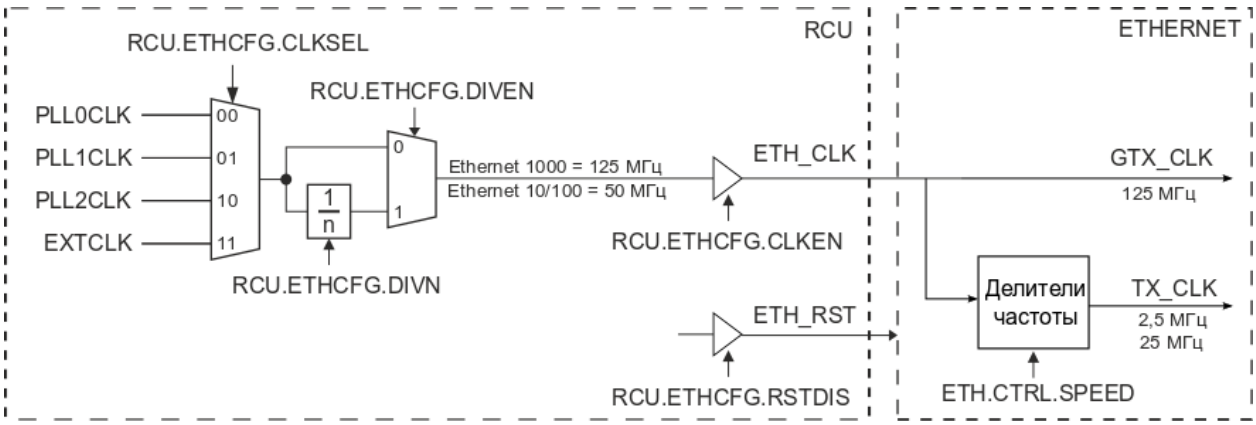


Рисунок 40.3 – Тактирование Ethernet PHY

Для работы контроллера Ethernet в режиме 1000 Мбит частота тактового сигнала должна соответствовать 125 МГц, для работы в режиме 10/100 Мбит частота тактового сигнала должна соответствовать 50 МГц. Переключение между режимами 10 Мбит и 100 Мбит осуществляется с помощью делителей частоты, управляемых битом SPEED регистра CTRL. При работе в режиме 100 Мбит бит SPEED должен быть установлен, в режиме 10 Мбит – сброшен.

Разрешение тактирования, а также вывод из состояния сброса регистровой части контроллера Ethernet выполняется установкой битов ETHEN в регистрах CGCFGAPB0 и RSTDISAPB0 блока RCU.

Сигнал GTX_CLK используется в гигабитном режиме работы и является входным для Ethernet PHY. Независимо от того, используется внешняя микросхема, либо внутренний блок Ethernet PHY, при работе режиме 1000 Мбит необходимо сконфигурировать один из PLL на частоту 125 МГц (либо использовать внешний сигнал EXTCLK с частотой 125 МГц) с помощью регистров блока RCU.

При работе в режиме 10/100 Мбит с использованием внутреннего блока Ethernet PHY также необходимо сконфигурировать один из PLL, но уже на частоту 50 МГц (либо использовать внешний сигнал EXTCLK с частотой 50 МГц) с помощью регистров блока RCU.

При работе в режиме 10/100 Мбит с использованием внешней микросхемы Ethernet PHY, подключенной с помощью альтернативных функций выводов GPIO, настройка тактирования не требуется, поскольку сигналы RXCLK и TXCLK являются входными для микроконтроллера и поступают от внешней микросхемы Ethernet PHY.

40.2 Функционирование

Структура контроллера Ethernet

Контроллер Ethernet состоит из 2 функциональных блоков: каналы DMA и интерфейс MDIO.

Основной функционал ложится на каналы DMA, используемые для передачи данных между шиной AHB и сетью Ethernet. Имеются один канал DMA передатчика и один канал DMA приемника. Управление работой каналов DMA осуществляется с помощью регистров, доступных через интерфейс APB.

Интерфейс MDIO используется для доступа к регистрам конфигурации и состояния на одном или нескольких устройствах физического уровня PHY, подключенных к MAC

(устройство управления доступом к среде). Работа этого интерфейса также контролируется через интерфейс APB.

Медиа-независимый интерфейс (MII) и гигабитный медиа-независимый интерфейс (GMII) используются для связи с PHY.

Поддержка протокола

Контроллер Ethernet реализован в соответствии со стандартом IEEE 802.3-2002. Поддержка необязательного подуровня управления отсутствует. Это означает, что пакеты с типом 0x8808 (единственные в настоящее время определенные ctrl пакеты) отбрасываются.

40.3 Интерфейс DMA на передачу

Передающий DMA интерфейс используется для передачи данных по сети Ethernet. Передача выполняется с использованием дескрипторов, расположенных в памяти.

Настройка дескриптора

Пример одного дескриптора приведен в таблицах 40.2 и 40.3. Количество отправляемых байт должно быть установлено в поле длины (LENGTH) и адресное поле (ADDRESS) должно указывать на данные. Нет ограничений по выравниванию адресного поля. Если установлен бит разрешения прерывания (IE), после отправки пакета будет сформировано прерывание (для этого необходимо, чтобы был установлен бит прерывания передатчика в управляющем регистре). Прерывание будет формироваться независимо от успешной передачи пакета.

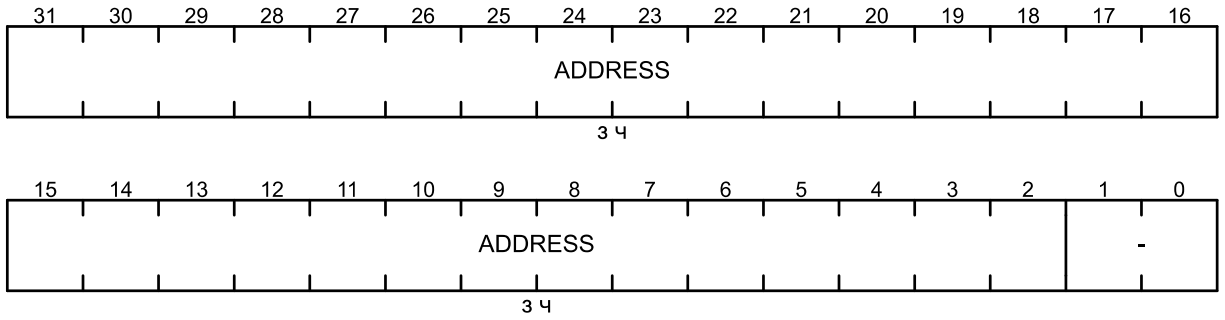
Таблица 40.2 – Слово 0 дескриптора передачи (смещение адреса 0x0)

Поле	Биты	Описание
UC	20	Вычисление и вставка контрольной суммы UDP для данного пакета. Контрольная сумма вставляется только в случае, если обнаружен UDP пакет
TC	19	Вычисление и вставка контрольной суммы TCP для данного пакета. Контрольная сумма вставляется только в случае, если обнаружен TCP пакет
IC	18	Вычисление и вставка контрольной суммы заголовка IP для данного пакета. Контрольная сумма вставляется только в случае, если обнаружен IP пакет
MO	17	Для данного пакета должны быть извлечены дополнительные дескрипторы (Scatter Gather I/O).
LC	16	Во время передачи произошла поздняя коллизия (только в режиме 1000 Мбит)
AL	15	Ошибка ограничения попыток. Пакет не был передан, так как было достигнуто максимальное количество попыток
UE	14	Ошибка недогрузки. Пакет был передан неправильно из-за ошибки недогрузки буфера FIFO
IE	13	Бит разрешения прерываний. Прерывание будет сгенерировано при отправке пакета из этого дескриптора, при условии, что установлен бит разрешения прерывания передатчика (TXINT) в управляющем регистре (CTRL). Прерывание генерируется независимо от того, был ли пакет успешно передан или завершен с ошибкой.

Продолжение таблицы 40.2

Поле	Биты	Описание
WR	12	Устанавливается в 1, чтобы указатель дескриптора обнулялся после использования этого дескриптора. Если этот бит не установлен, указатель будет увеличиваться на 8. Указатель автоматически обнуляется при достижении границы таблицы дескрипторов в 1 КБ
EN	11	Включение дескриптора. Всегда должен быть установлен последним из всех полей дескриптора
LENGTH	10-0	Количество байт для передачи
–	31-21	Зарезервировано

Таблица 40.3 – Слово 1 дескриптора передачи (смещение адреса 0x4)



Поле	Биты	Описание
ADDRESS	31-2	Указатель на область буфера, откуда будут загружаться пакетные данные
–	1-0	В целях выравнивания дескрипторов по границе слова (4 байта) данные биты должны иметь нулевое значение

Чтобы включить дескриптор, нужно установить бит разрешения (EN), после чего поля дескриптора нельзя изменять до тех пор, пока бит включения не будет очищен аппаратно. Остальные поля дескриптора описаны далее в этом разделе.

Запуск передач

Активации дескриптора недостаточно для начала передачи. Первоначально должен быть установлен указатель на область памяти, содержащей дескрипторы. Это делается с помощью регистра указателя на дескриптор передатчика (TXBASE). Адрес должен быть выровнен по границе 1 КБ. Биты с 31 по 10 (поле BADDR) содержат базовый адрес области дескриптора, а биты с 9 по 3 (поле DPNT) формируют указатель на отдельный дескриптор. Первый дескриптор должен располагаться по базовому адресу, и после того, как он будет считан контроллером Ethernet, поле указателя инкрементируется на 8, чтобы указывать на следующий дескриптор. Указатель автоматически вернется к нулю, когда будет достигнута следующая граница в 1 КБ (был использован дескриптор со смещением адреса 0x3F8). Бит WR в дескрипторах может быть установлен для того, чтобы указатель возвращался к нулю до границы в 1 КБ.

Для большей гибкости поле указателя также доступно для записи, но следует соблюдать осторожность при записи в регистр указателя дескриптора. Не допускаются изменения во время активной передачи.

Последним шагом для активации передачи является установка бита разрешения передачи (TXEN) в управляющем регистре (CTRL). Это сообщает контроллеру Ethernet, что

в таблице дескрипторов есть активные дескрипторы. Бит разрешения передачи должен всегда устанавливается при включении новых дескрипторов, даже если передачи уже активны. Дескрипторы всегда должны быть включены до установки бита разрешения передачи.

Обработка дескриптора после передачи

Когда передача пакета завершена, происходит запись статуса в нулевое слово соответствующего дескриптора. Бит ошибки недогрузки (UE) устанавливается, если ОЗУ передатчика не смогла предоставлять данные с достаточной скоростью. Это указывает на проблему синхронизации, которая, вероятнее всего, вызвана низкой тактовой частотой на шине AHB. Весь пакет перед передачей буферизуется в ОЗУ передатчика, поэтому перегрузка шины не может вызвать ошибку недогрузки. Бит ошибки ограничения попыток (AL) устанавливается, если произошло больше коллизий, чем допускается. При работе в режиме 1000 Мбит бит LC указывает, что коллизия произошла после того, как была пройдена граница временного интервала.

Пакет был успешно передан, только если эти три бита равны нулю. Остальные биты в нулевом слове дескриптора обнуляются после передачи, а первое слово остается нетронутым.

Бит включения следует использовать как индикатор того, что дескриптор можно использовать снова, т. е. когда он был очищен контроллером Ethernet. В регистре состояния контроллера Ethernet есть три бита, которые содержат статус передачи. Бит ошибки передачи (TXERR) устанавливается каждый раз, когда передача завершается ошибкой (когда был установлен хотя бы один из трех битов состояния в дескрипторе передачи). Передача успешная (TXSUC) устанавливается каждый раз после успешного завершения передачи.

Бит ошибки AHB при передаче (TXAHBERR) устанавливается при обнаружении ошибки AHB либо при считывании дескриптора, либо при считывании пакетных данных, либо при записи статуса в дескриптор. Любые активные передачи прерываются и передатчик отключается. Передатчик можно снова активировать, установив бит TXEN регистра управления.

Подготовка данных для передачи

Передаваемые данные должны быть размещены, начиная с адреса, указанного в поле ADDRESS первого слова дескриптора передачи. Контроллер Ethernet не добавляет поля адреса Ethernet и типа, поэтому они также должны хранится в буфере с данными. Ethernet-CRC (4 байта) автоматически добавляется в конце каждого пакета. Каждый дескриптор будет отправлен как отдельный пакет Ethernet. Если в поле размера (LENGTH) дескриптора указано больше, чем 1514 байт, пакет не будет отправлен.

Режим разборка-сборка

Пакет может быть сгенерирован из данных, извлеченных из нескольких дескрипторов. Это называется Scatter Gather I/O. Для указания того, что для генерации текущего пакета следует использовать больше дескрипторов устанавливается бит MO. Когда данные из текущего дескриптора считываются в ОЗУ, извлекается следующий дескриптор, и новые данные добавляются к предыдущим. Это продолжается до тех пор, пока не будет обнаружен дескриптор со сброшенным битом MO. После этого пакет будет передан.

Статус записывается немедленно, при считывании данных в ОЗУ для дескрипторов с установленным битом MO. Биты состояния всегда установлены в 0, так как передачи не было. Биты статуса будут записаны в последний дескриптор пакета (у которого сброшен бит MO) после завершения передачи.

Для дескрипторов с установленным битом MO прерывания не генерируются, так что в этом случае бит IE не имеет значения.

Биты управления выгрузкой контрольной суммы (UC, TC, IC) должны быть установлены в одинаковые значения для всех дескрипторов, используемых в рамках одного пакета.

Вычисление контрольной суммы

Предусмотрена поддержка аппаратных вычислений контрольной суммы для TCP и UDP через IPv4. Вычисление контрольной суммы включается в каждом дескрипторе и применяется только к этому пакету (когда установлен бит MO, все дескрипторы, используемые для одного пакета, должны иметь одинаково установленные управляющие биты контрольной суммы).

Бит контрольной суммы IP (IC) включает вычисление контрольной суммы заголовка IP. Если при передаче пакета, связанного с дескриптором, обнаружен пакет IPv4, вычисляется и вставляется контрольная сумма заголовка. Если выбрана контрольная сумма TCP (TC), тогда контрольная сумма TCP вычисляется и вставляется при обнаружении пакета TCP/IPv4. Наконец, если установлен бит контрольной суммы UDP (UC), тогда контрольная сумма UDP вычисляется и вставляется при обнаружении пакета UDP/IPv4. В случае фрагментированных IP-пакетов контрольные суммы для TCP и UDP вставляются только для первого фрагмента (который содержит заголовок TCP или UDP).

40.4 Интерфейс DMA на прием

Приемный DMA интерфейс используется для приема данных из сети Ethernet. Прием осуществляется с использованием дескрипторов, расположенных в памяти.

Настройка дескриптора

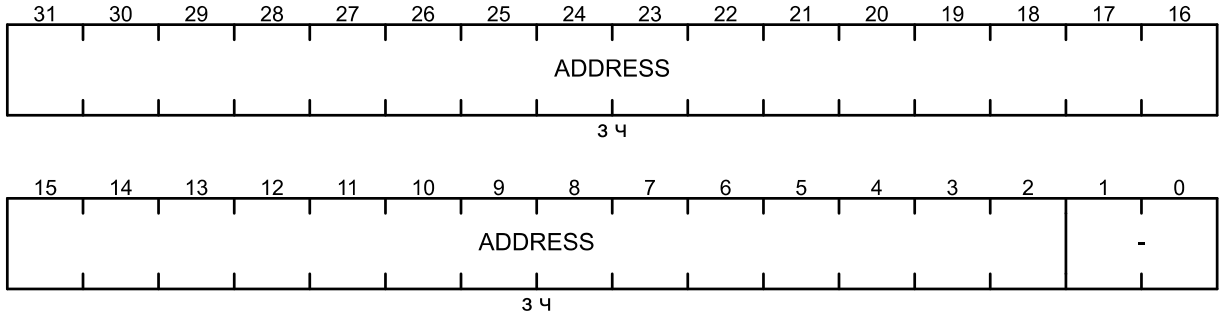
Один из дескрипторов приведен в таблицах 40.4 и 40.5. Поле адреса (ADDRESS) указывает на место, куда должны быть сохранены полученные данные. Ограничений по выравниванию нет. Контроллер Ethernet никогда не сохранит в буфере более 1518 байт (обозначенный максимальный размер кадра без учета CRC). CRC поле (4 байта) никогда не сохраняется в памяти, поэтому оно не включено в этот размер. Если установлен бит разрешения прерывания (IE), то при получении пакета буфером будет сгенерировано прерывание (для этого также требуется, чтобы был установлен бит прерывания приемника (RXINT) в управляющем регистре CTRL). Прерывание будет генерироваться независимо от успешности получения пакета.

Бит включения (EN) устанавливается для указания того, что дескриптор действующий, и может использоваться для сохранения пакета. После его установки дескриптор нельзя изменять до тех пор, пока бит EN не будет очищен контроллером Ethernet.

Таблица 40.4 – Слово 0 дескриптора приема (смещение адреса 0x0)

Поле	Биты	Описание
MC	26	Адрес назначения пакета был адресом многоадресной рассылки (не широковещательной)
OE	17	Кадр был принят неправильно из-за переполнения FIFO
CE	16	В кадре была обнаружена ошибка CRC
FT	15	Полученный кадр больше максимального размера. Избыточная часть была отсечена
AE	14	Ошибка выравнивания. Получение нечетного количества полубайт
IE	13	Разрешение прерываний. Прерывание будет сгенерировано, когда пакет принимается в этот дескриптор при условии, что бит разрешения прерывания приемника (RXINT) в регистре управления (CTRL) установлен. Прерывание генерируется независимо от того, был ли пакет успешно принят или завершен с ошибкой
WR	12	Бит устанавливается в 1, чтобы указатель дескриптора обнулялся после того, как данный дескриптор был использован. Если этот бит не установлен, указатель будет увеличиваться на 8. Указатель автоматически обнуляется, когда достигнута граница таблицы дескрипторов в 1 КБ
EN	11	Бит устанавливается в 1, чтобы включить дескриптор. Всегда должен быть установлен последним из всех полей дескриптора
LENGTH	10-0	Количество байт, полученных этим дескриптором
–	31-27, 25-18	Зарезервировано

Таблица 40.5 – Слово 1 дескриптора приема (смещение адреса 0x4)



Поле	Биты	Описание
ADDRESS	31-2	Указатель на область буфера, куда будут загружаться пакетные данные
–	1-0	В целях выравнивания дескрипторов по границе слова (4 байта) данные биты должны иметь нулевое значение

Начало приема

Включение дескриптора недостаточно для начала приема. Сначала должен быть установлен указатель на область памяти, содержащую дескрипторы. Это делается с помощью регистра указателя дескриптора приемника (RXBASE). Адрес должен быть выровнен по границе 1 Кбайт. Биты с 31 по 10 (поле BADDR) содержат базовый адрес области дескриптора, а биты с 9 по 3 (поле DPNT) формируют указатель на отдельный дескриптор. Первый дескриптор должен быть расположен по базовому адресу, и, после того как он был использован контроллером Ethernet, поле указателя увеличивается на 8, чтобы указать на следующий дескриптор. Указатель автоматически вернется к нулю, когда будет достигнута следующая граница 1 Кбайт (использован дескриптор по смещению адреса 0x3F8). Бит WR в дескрипторах может быть установлен для того, чтобы указатель возвращался к нулю до границы в 1 Кбайт.

Поле указателя также было сделано доступным для записи с целью обеспечения максимальной гибкости, но при записи в регистр указателя дескриптора следует соблюдать осторожность. Нельзя изменять его во время активного приема.

Последним шагом для активации приема является установка бита включения приемника (RXEN) в регистре управления (CTRL). Это заставит контроллер Ethernet считать первый дескриптор и дождаться входящего пакета.

Обработка дескриптора после приема

Контроллер Ethernet указывает на завершение приема путем очистки бита включения дескриптора (EN). Другие управляющие биты (WR, IE) также очищаются. Количество полученных байт отображается в поле длины (LENGTH). Части сохраненного кадра Ethernet представляют собой поля адреса назначения, адреса источника, типа и данных. Биты 24-14 в нулевом слове дескриптора являются статусными битами, указывающими на различные ошибки приема. Биты 18–14 равны нулю после приема без ошибок уровня канала. Статусные биты описаны в таблице 40.4 (кроме битов выгрузки контрольной суммы, которые также описаны в подразделе «Выгрузка контрольной суммы»).

Поступающие пакеты, размер которых меньше минимального размера Ethernet, равного 64 байтам, не считаются принятыми и отбрасываются. Текущий дескриптор приема останется нетронутым и будет использован для первого пакета, поступившего с допустимым размером. Каждый раз, кода происходит данное событие устанавливается бит TOOSMALL в регистре статуса (STATUS).

При получении пакета с адресом, не принятым MAC, будет установлен бит IA в регистре статуса.

Пакеты, размер которых превышает максимальный, вызывают установку бита FT в дескрипторе приема. Не гарантируется, что поле длины будет содержать правильное значение полученных байтов. Подсчет останавливается после того, как слово, содержащее последний байт до максимального размера, будет записано в память.

Адресное слово дескриптора никогда не затрагивается контроллером Ethernet.

Прием с ошибками AHB

Если во время считывания дескриптора или сохранения данных возникает ошибка AHB, в регистре статуса будет установлен бит ошибки AHB приемника (RXAHBERR), и приемник будет отключен. Текущий прием прерывается. Приемник можно снова включить, установив бит включения приема в управляющем регистре.

Принятые MAC-адреса

В конфигурации по умолчанию ядро получает пакеты либо с узконаправленным адресом, заданным в регистре MAC адреса, либо с широковещательным адресом. Также может быть включена поддержка многоадресной рассылки, и в этом случае hash-функция

используется для фильтрации полученных многоадресных пакетов. 64-битный регистр, доступный через интерфейс APB определяет, какие адреса должны быть получены. Каждый адрес сопоставляется с одним из 64 битов с использованием hash-функции, и, если бит установлен в единицу, пакет будет получен. Адрес сопоставляется с таблицей, используя 6 младших битов 32-битной контрольной суммы Ethernet, рассчитанной по адресу назначения кадра MAC. Бит MC в дескрипторе приема устанавливается, если им был получен пакет с групповым адресом.

40.5 Интерфейс MDIO

Интерфейс MDIO обеспечивает доступ к конфигурации PHY и регистрам статуса через двухпроводной канал интерфейса, который включен в интерфейс MII. Контроллер Ethernet обеспечивает полную поддержку интерфейса MDIO.

Интерфейс MDIO может использоваться для доступа к от 1 до 32 PHY, содержащих от 1 до 32 16-битных регистров. Передача чтения настраивается записью адреса PHY (поле PADDR) и адреса регистра (поле RADDR) в управляющий регистр MDIOCSR и установкой бита чтения RD. Это приводит к установке бита занятости (BUSY), операция завершается, когда бит занятости будет очищен. Если операция прошла успешно, бит LINKFAIL будет равен нулю, а поле данных будет содержать прочитанные данные. О неудачной операции свидетельствует установленный бит LINKFAIL. В этом случае поле данных не определено.

Операция записи начинается с записи 16-битных данных (поле DATA), PHY адреса (поле PADDR) и адреса регистра (поле RADDR) в регистр управления MDIOCSR и установкой бита записи WR. Операция завершается, когда бит занятости (BUSY) сбрасывается и успешна, если бит LINKFAIL равен нулю.

Прерывания PHY

Контроллер Ethernet также поддерживает прерывания по изменению состояния PHY. К выводу mdint может быть подключен чувствительный к уровню сигнал прерывания. Бит изменения состояния PHY (PHYSTAT) в регистре статуса (STATUS), устанавливается каждый раз при обнаружении события изменения сигнала. Если установлен бит разрешения прерывания по изменению состояния PHY (бит PI), в момент события, контроллер Ethernet также сгенерирует прерывание на шине AHB.

40.6 Блок физического уровня интерфейса Ethernet 10/100/1000 (PHY)

Блок Ethernet PHY 10/100/1000 представляет собой блок, реализующий физический уровень интерфейса Ethernet 10/100/1000 в соответствии протокольным и электрофизическим требованиям, изложенным в спецификации IEEE 802.3, разделы:

– 10BASE-T (IEEE 802.3i);

– 100BASE-TX (IEEE 802.3u);

– 1000BASE-T (IEEE 802.3ab).

Блок обеспечивает подключение к уровню управления доступом к среде (MAC) по интерфейсу RGMII. Функциональная схема блока приведена на рисунке 40.4.

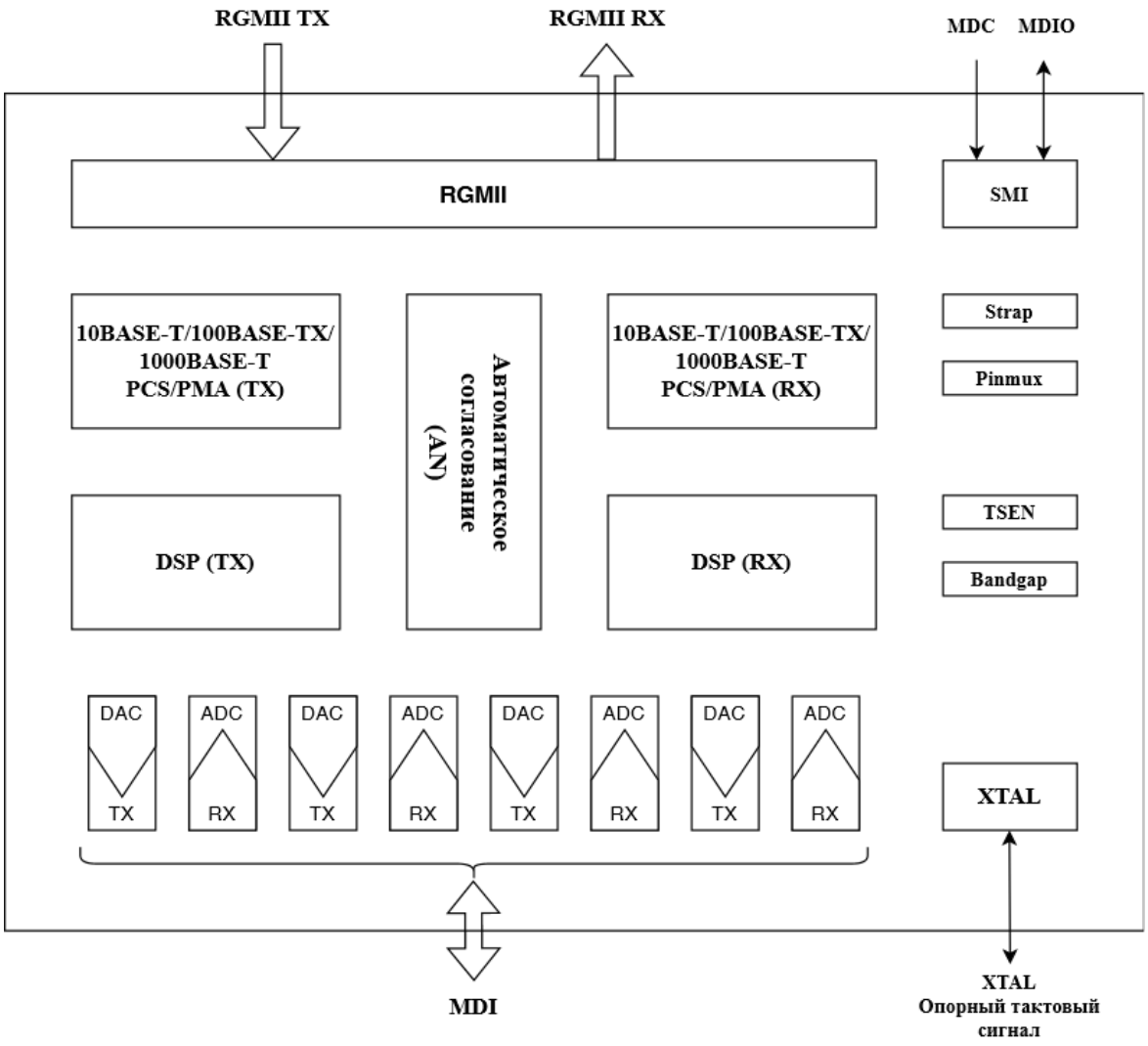


Рисунок 40.4 – Функциональная схема блока Ethernet PHY 10/100/1000

Передатчик

Ethernet PHY использует цифроаналоговый преобразователь для передачи кодированных цифровых сигналов по кабелю категории 5. Внутренний терминальный резистор обеспечивает хорошие показатели обратных потерь и сокращает количество материалов на печатной плате. Технология формирования импульсов и регулирования скорости нарастания используется для устранения проблем с электромагнитными помехами (EMI). Передаваемые данные кодируются в форматах 4D-PAM5, 4B/5B и Manchester для 1000BASE-T, 100BASE-TX и 10BASE-T соответственно.

Приемник

Ethernet PHY использует высокоскоростной аналого-цифровой преобразователь с высоким разрешением для приема канальных сигналов, что гарантирует дальность приема, значительно превышающую стандартную. В 1000BASE-T используется гибридный аналоговый интерфейс для уменьшения эха на ближнем конце, что позволяет передатчику и приемнику совместно использовать один трансформатор. В цифровом домене реализованы такие технологии, как подавление эхо-сигналов, подавление перекрестных помех ближнего диапазона (NEXT), подавление отклонения базовой линии и адаптивное выравнивание.

Интерфейс RGMII

Ethernet PHY поддерживает интерфейс RGMII, совместимый с RGMII 2.0. Интерфейс Обеспечивает 6 выводов для передачи данных (т.е. TXC, TX_CTL, TXD[3:0]) и 6 выводов для приема данных (т.е. RXC, RX_CTL, RXD[3:0]). Интерфейс работает с тактовой частотой 125 МГц в режиме 1000BASE-T, 25 МГц в режиме 100BASE-TX или 2,5 МГц в режиме 10BASE-T.

Статус обмена по интерфейсу RGMII можно узнать с помощью регистра RGMII_STAT. Имеется возможность настройки прерываний по изменению статусных битов с помощью регистра RGMII_IM. Текущий статус прерываний можно узнать с помощью регистра RGMII_IF.

Автоматическое согласование (AN)

Ethernet PHY поддерживает функцию автоматического согласования (AN), которая определена в стандарте 802.3. Функция AN автоматически определяет пересечение линий MDI/MDIX и упрощает процесс подключения. В блоке Ethernet PHY также реализована автоматическая коррекция полярности при неправильном подключении кабеля.

Подготовка к работе с PHY

При подготовке к работе с внутренним PHY необходимо выполнить следующую последовательность действий:

– Разрешение тактирование блока PHY и RGMII-преобразователя записью
RCU->ETHCFG_bit.CLKEN = 1;

– Вывод из сотояния сброса блока PHY и RGMII-преобразователя записью
RCU->ETHCFG_bit.RSTDIS = 1;

– Переключение на использование внутреннего блока PHY записью
ETH->PHYCFG_bit.EXTPHY = 0;

Пример конфигурации PHY в режиме 10 Мбит

```
// RGMII to UTP 10M: reg = 0x4   value = 0x61
ETH->MDIOCSR = (0x0061 << ETH_MDIOCSR_DATA_Pos)  |
                        (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                        (0x4    << ETH_MDIOCSR_RADDR_Pos) |
                        (1      << ETH_MDIOCSR_WR_Pos);
delay(300);

// RGMII to UTP 10M: reg = 0x9   value = 0x0
ETH->MDIOCSR = (0x0000 << ETH_MDIOCSR_DATA_Pos)  |
                        (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                        (0x9    << ETH_MDIOCSR_RADDR_Pos) |
                        (1      << ETH_MDIOCSR_WR_Pos);
delay(300);

// RGMII to UTP 10M: reg = 0x0   value =0x9140
ETH->MDIOCSR = (0x9140 << ETH_MDIOCSR_DATA_Pos)  |
                        (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                        (0x0    << ETH_MDIOCSR_RADDR_Pos) |
                        (1      << ETH_MDIOCSR_WR_Pos);
delay(300);
```

```
  // Read link status
  ETH->MDIOCSR = (0x0000 << ETH_MDIOCSR_DATA_Pos) |
                 (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                 (0x1    << ETH_MDIOCSR_RADDR_Pos) |
                 (1      << ETH_MDIOCSR_RD_Pos);
  delay(300);
```

Пример конфигурации PHY в режиме 100 Мбит

```
// RGMII to UTP 100M: reg = 0x4   value = 0x1E1
  ETH->MDIOCSR = (0x01E1 << ETH_MDIOCSR_DATA_Pos)  |
                 (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                 (0x4    << ETH_MDIOCSR_RADDR_Pos) |
                 (1      << ETH_MDIOCSR_WR_Pos);
  delay(300);

  // RGMII to UTP 100M: reg = 0x9   value = 0x0
  ETH->MDIOCSR = (0x0000 << ETH_MDIOCSR_DATA_Pos)  |
                 (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                 (0x9    << ETH_MDIOCSR_RADDR_Pos) |
                 (1      << ETH_MDIOCSR_WR_Pos);
  delay(300);

  // RGMII to UTP 100M: reg = 0x0   value =0x9140
  ETH->MDIOCSR = (0x9140 << ETH_MDIOCSR_DATA_Pos)  |
                 (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                 (0x0    << ETH_MDIOCSR_RADDR_Pos) |
                 (1      << ETH_MDIOCSR_WR_Pos);
  delay(300);

  // Read link status
  ETH->MDIOCSR = (0x0000 << ETH_MDIOCSR_DATA_Pos) |
                 (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                 (0x1    << ETH_MDIOCSR_RADDR_Pos) |
                 (1      << ETH_MDIOCSR_RD_Pos);
  delay(300);
```

Пример конфигурации PHY в режиме 1000 Мбит

```
// RGMII to UTP 1000M: reg = 0x9   value = 0x200
  ETH->MDIOCSR = (0x0200 << ETH_MDIOCSR_DATA_Pos) |
                 (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                 (0x9    << ETH_MDIOCSR_RADDR_Pos) |
                 (1      << ETH_MDIOCSR_WR_Pos);
  delay(300);

  // RGMII to UTP 1000M: reg = 0x0   value =0x9140
  ETH->MDIOCSR = (0x9140 << ETH_MDIOCSR_DATA_Pos) |
                 (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                 (0x0    << ETH_MDIOCSR_RADDR_Pos) |
                 (1      << ETH_MDIOCSR_WR_Pos);
  delay(300);

  // Read link status
  ETH->MDIOCSR = (0x0000 << ETH_MDIOCSR_DATA_Pos) |
                 (0x1    << ETH_MDIOCSR_PADDR_Pos) |
                 (0x1    << ETH_MDIOCSR_RADDR_Pos) |
                 (1      << ETH_MDIOCSR_RD_Pos);
  delay(300);
```

41 Программно-аппаратные средства отладки

Для освоения и изучения 32-разрядного микроконтроллера К1921ВГ1Т, а также для макетирования и отладки систем пользователя на его основе, следует использовать макетно-отладочную плату, которая позволяет подключать внешние элементы к портам микроконтроллера, работать с внешними интерфейсами, а также программировать встроенную Flash-память и выполнять отладку и оценку работы прикладных программ.

Для создания программного обеспечения можно использовать интегрированную среду разработки Eclipse, компилятор GCC для RISC-V и систему отладки OpenOCD.

Адаптер Olimex ARM-USB-OCD-H (J-Link V9, CMSIS-DAP, FT2232) обеспечивает взаимодействие между интегрированной средой разработки Eclipse (с GCC компилятором для RISC-V и системой отладки OpenOCD), установленной на персональном компьютере, и отладочными ресурсами, встроенными в микроконтроллер К1921ВГ1Т, а также выполнение отладочных функций. Информационный обмен с микроконтроллером осуществляется с помощью интерфейса JTAG.

Поскольку при запуске микроконтроллер всегда тактируется от внутреннего RC-генератора (~4МГц) важно соблюдать следующее соотношение между системной тактовой частотой (SYSCLK) и тактовой частотой JTAG (TCK):

$$F_{\mathrm{SYSCLK}}/F_{\mathrm{TCK}} \geq 12 \qquad (41.1)$$

42 Электрические параметры микроконтроллера

42.1 Предельные нормы параметров

Напряжения и токи, превышающие предельно допустимые значения, указанные в таблицах 42.1 и 42.2, могут привести к необратимому повреждению устройства. Функциональная работа устройства в этих условиях не подразумевается. Воздействие условий с предельно допустимым уровнем в течение длительного времени может повлиять на надежность устройства.

Таблица 42.1 – Предельно допустимые уровни напряжения

Наименование параметра, единица измерения	Буквенное обозначение	Предельная норма	
		не менее	не более
Напряжение питания цифровой части, В	U_{CC1}	–	4,5
Напряжение питания аналоговой части, В	U_{CC2}	–	4,5
Батарейное напряжение питания, В	U_{CC3}	–	4,5
Входное напряжение низкого уровня, В	U_{IL}	-0,6	–
Входное напряжение высокого уровня, В	U_{IH}	–	U_{CC1} + 0,6
Отклонения между различными выводами питания, мВ	ΔU_{CCx}	–	50
Отклонения между различными выводами земли, мВ	ΔU_{GND}	–	50

Таблица 42.2 – Предельно допустимые уровни тока

Наименование параметра, единица измерения	Буквенное обозначение	Предельная норма	
		не менее	не более
Суммарный максимальный ток по выводам портов ввода-вывода, мА	$\sum I_{IO}$	–	150
Выходной ток высокого уровня, мА	I_{OH}	-10	–
Выходной ток низкого уровня, мА	I_{OL}	–	10
Максимальный ток инжекции по каждому выводу портов ввода-вывода, мА	I_{INJ}	–	±5
Максимальный суммарный ток инжекции по всем выводам портов ввода-вывода, мА	$\sum I_{INJ}$	–	±25

Примечание – Значение тока инжекции никогда не должно превышаться. Это неявно гарантируется при соблюдении предельно допустимого входного напряжения. Если максимальное значение входного напряжения не может быть соблюдено, ток инжекции должен быть ограничен внешним значением. Положительный ток инжекции возникает при $U_{IH} > U_{CC1}$, в то время как отрицательный ток инжекции возникает при $U_{IL} < U_{GND}$.

Таблица 42.3 – Предельные нормы по температуре

Наименование параметра, единица измерения	Предельная норма	
	не менее	не более
Рабочий температурный диапазон, ºС	-40	+105
Тепловое сопротивление, ºС/Вт	–	15,1

42.2 Допустимые нормы параметров

Таблица 42.4 – Рабочие уровни напряжения и токи питания

Наименование параметра, единица измерения		Буквенное обозначение	Допустимая норма при эксплуатации	
			не менее	не более
Напряжение питания цифровой части, В		U_{CC1}	2,5	3,6
Напряжение питания аналоговой части, В		U_{CC2}	2,5	3,6
Батарейное напряжение питания, В		U_{CC3}	1,6	3,6
Динамический ток потребления по выводу VCC в активном режиме[1)], мА, U_{CC1}=3,6 В, f_{CI} = 204 МГц	T=+25 ºC	I_{OCC1}	–	150
	T=+105 ºC			250
Динамический ток потребления по выводу VCC в активном режиме [2)], мА, U_{CC1}=3,6 В, f_{CI} = 204 МГц	T=+25 ºC	I_{OCC2}	–	170
	T=+105 ºC			350
Ток потребления по выводу VBAT, мкА U_{CC1} = U_{CC2} = 0 В, U_{CC3} = 3,3 В		I_{VBAT}	–	10
Максимальный ток потребления АЦП по аналоговому питанию AVCC_ADC, мкА U_{CC1} = U_{CC2} = 3,6 В, F_S=2,5 MSPS		I_{AVCC_ADC}	–	450

_____

[1)] Ядра работают в режиме LockStep, используются основная Flash -память и RAM. Задействованы порты ввода-вывода GPIO, остальные периферийные блоки не тактируются и находятся в состоянии сброса.

[2)] Ядра работают в режиме LockStep, используются основная Flash -память и RAM. Задействованы порты ввода-вывода GPIO, АЦП, DMA-контроллер, блок криптографии, остальные периферийные блоки тактируются и выведены из состояния сброса.

42.3 Электрические параметры периферийных блоков

Таблица 42.5 – Электрические параметры супервизоров (POR) и мониторов питания

Наименование параметра, (режим измерения)	Норма, В		
	не менее	номинал	не более
Порог срабатывания блока MVD 1,1 В	–	0,89	–
Порог срабатывания блока LVD 1,1 В	–	0,99	–
Порог срабатывания блока HVD 1,1 В	–	1,21	–
Порог срабатывания блока UVD 1,1 В	–	1,32	–
Порог срабатывания POR 1,1 В (спадающее напряжение питания)	0,5	0,55	0,6
Порог срабатывания POR 1,1 В (нарастающее напряжение питания)	0,70	0,71	0,72
Порог срабатывания блока MVD 3,3 В	–	2,5	–
Порог срабатывания блока LVD 3,3 В	–	2,97	–
Порог срабатывания блока HVD 3,3 В	–	3,63	–
Порог срабатывания блока UVD 3,3 В	–	3,9	–
Порог срабатывания POR 3,3 В (спадающее напряжение питания)	1,6	1,7	1,8
Порог срабатывания POR 3,3 В (нарастающее напряжение питания)	1,90	1,91	1,92

Таблица 42.6 – Пороги срабатывания блока BOR

Значение поля TUNE	Порог срабатывания, В	
	при росте напряжения	при спаде напряжения
00b	3,0	2,9
01b	2,7	2,6
10b	2,2	2,1
11b	Зарезервировано	Зарезервировано

Таблица 42.7 – Характеристики внутреннего RC-генератора (HSI)

Наименование параметра	Норма параметра		
	Не менее	Номинал	Не более
Ток потребления, мкА	–	7	70
Диапазон подстройки частоты, МГц	2,553	3,375	3,894
Примечание – Нестабильность частоты ±4%			

Таблица 42.8 – Характеристики внутреннего осциллятора (HSE)

Наименование параметра	Норма параметра		
	Не менее	Номинал	Не более
Ток потребления, мА	–	–	2,5
Напряжение высокого уровня на выводе XI_OSC, В	2	–	U_{CC1}+0,3
Напряжение низкого уровня на выводе XI_OSC, В	-0,3	–	0,8
Диапазон частот, МГц	2	–	27
Время запуска, мс	–	–	30

Таблица 42.9 – Характеристики внутреннего осциллятора блока RTC (LSI)

Наименование параметра	Норма параметра		
	Не менее	Номинал	Не более
Ток потребления, мА	–	–	2
Напряжение высокого уровня на выводах XI_RTC, В	0,6	–	–
Напряжение низкого уровня на выводах XI_RTC, В	–	–	0,4
Частота, кГц	–	32,768	–
Время запуска, мс	–	–	500

Таблица 42.10 – Характеристики блока RTC

Наименование параметра, единица измерения	Норма параметра	
	Не менее	Не более
Пороговое значение напряжения питания цифровой части U_{CC1} ниже которого выполняется переключение блока RTC на батарейное питание (VCC↔VBAT), В	–	2,85
Гистерезис переключения шин питания блока RTC (VCC↔VBAT), В	0,1	–

Таблица 42.11 – Характеристики аналогового компаратора ACMP

Наименование параметра, единица измерения	Норма параметра	
	Не менее	Не более
Гистерезис аналогового компаратора, мВ	–	120

Таблица 42.12 – Характеристики блока PLL

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра		
		Не менее	Номинал	Не более
Входная опорная частота, МГц	Fref	10	–	100
Частота на выходе ГУН, МГц	Fvco	400	–	1000
Выходная частота PLL, МГц	Fout	0,2	–	204
Выходной коэффициент заполнения, %	–	46	50	54
Время захвата частоты, количество тактовых циклов	–	–	1000	1500
Время захвата фазы, количество тактовых циклов	–	–	–	1500
Выходная задержка, пс	–	–	1000	1400
Выходная задержка в режиме Bypass, пс	–	–	210	330
Ток потребления по линии цифрового питания, мкА/МГц	I_{CC1}	–	1,4	1,68
Ток потребления по линии аналогового питания при Fvco = 1 ГГц, мА	I_{CC2}	–	0,8	1,0
Общий ток потребления в спящем режиме, мкА	I_{CC}	–	5	–

Таблица 42.13 – Характеристики блока Flash-памяти

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра		
		Не менее	Номинал	Не более
Время записи слова данных, мкс	t_{PROG}	15	20	–
Время чтения слова данных, нс	t_{RC}	25	–	–
Время стирания страницы, мс	t_{ERASE}	8	–	20
Время полного стирания всей памяти, мс	t_{SCE}	8	–	20
Активный ток чтения мкА/МГц	$I_{CC\ READ}$	–	–	150
Активный ток записи, мА	$I_{CC\ PROG}$	–	–	5
Активный ток стирания, мА	$I_{CC\ ERASE}$	–	–	4
Ток потребления в режиме ожидания при температуре +25ºС, мкА	$I_{CC\ SB}$	–	–	195
Ток потребления в режиме PowerDown при температуре +25ºС, мкА	$I_{CC\ PD}$	–	–	5
Ресурс сектора, циклы перезаписи	N_{END}	100000	–	–
Срок хранения данных, лет	T_{DR}	10	–	–

Таблица 42.14 – Характеристики портов ввода-вывода

Наименование параметра, единица измерения		Буквенное обозначение	Норма параметра	
			Не менее	Не более
Входное напряжение низкого уровня, В		U_{IL}	–0,5	$0,3U_{CC1}$
Входное напряжение высокого уровня, В		U_{IH}	$0,7U_{CC1}$	U_{CC1}
Выходное напряжение низкого уровня буферов ввода/вывода, В, (I_{OL} = 6 мА, U_{CC1} = 3,3 В)		U_{OL}	–	0,4
Выходное напряжение высокого уровня буферов ввода/вывода, В, (I_{OH} = –6 мА, U_{CC1} = 3,3 В)		U_{OH}	U_{CC1}– 0,4	–
Ток утечки по входам без «pull-up» и «pull-down», мкА, (0 В < U_I < U_{CC1},)		I_{LIL}	–1	–
		I_{LIH}	–	1
Входной ток, мкА, (0 В < U_I < U_{CC1})	вывода «pull-up»	I_{I1}	–200	–
	вывода «pull-down»	I_{I2}	–	200
Сопротивление резисторов подтяжки «pull-up», кОм		R_{PU}	27	65

Таблица 42.15 – Характеристики блока АЦП

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра	
		Не менее	Не более
Дифференциальная нелинейность АЦП, LSB U_{CC1} = U_{CC2} = 3,3 В, f_{CI} = 204 МГц	$E_{DNL\ ADC}$	–2,5	2,5
Интегральная нелинейность АЦП, LSB U_{CC1} = U_{CC2} = 3,3 В, f_{CI} = 204 МГц	$E_{INL\ ADC}$	–5,0	5,0
Максимальный ток потребления АЦП по аналоговому питанию AVCC_ADC, мкА U_{CC1} = U_{CC2} = 3,6 В, F_S=2,5 MSPS	I_{AVCC_ADC}	–	500

Таблица 42.16 – Характеристики датчика температуры TSENSOR

Наименование параметра	Норма			Примечание
	не менее	номинал	не более	
Точность, °С	-3,8 -7	–	4 1,3	SCALE = 1 SCALE = 0
Наклон, мВ/°С	–	1,8 3,5	–	SCALE = 1 SCALE = 0
Время запуска, мкс	–	0,2	1	–
Ток потребления, мкА (статический)	7	16	24	–

Таблица 42.17 – Характеристики блока ЦАП

Наименование параметра, единица измерения (режим измерения)		Норма		
		не менее	номинал	не более
Количество каналов, шт			1	
Разрядность, бит			12	
Аналоговое напряжение питания, В		1,8		3,6
Цифровое напряжения питания, В		0,9		1,2
Температурный диапазон, °С		-45		125
Источники опорного напряжения			внутр./ внеш.	
Диапазон опорных напряжений, В		1,8		AVCC
Диапазон выходных напряжений, В (BUFFER ON)		0,2		U_{VDDA}-0,2
Диапазон выходных напряжений, В (BUFFER OFF)		0		U_{REF+}
Максимальная частота обновления выходных данных, MSPS CL=50пФ RL=5кОм (BUFFER ON) full-scale		0,5		
Параметры нагрузки, кОм режим BUFFER ON	Нагрузка подключена к AGND	5		
	Нагрузка подключена к AVCC	25		
Параметры нагрузки, пФ режим BUFFER OFF				50
Интегральная нелинейность преобразования (INL), LSB		-4		4
Дифференциальная нелинейность преобразования (DNL), LSB		-2		2
Погрешность смещения нуля, LSB VREF+=1,8 режим BUFFER ON		-30		30
Погрешность смещения нуля, LSB VREF+=1,8 режим BUFFER OFF		-8		8
Погрешность коэффициента усиления, %		-1		1
Ток потребления I_{DDA}, мкА Normal (BUFFER ON) AVCC, Normal (BUFFER OFF) AVCC			490 20	
Ток потребления I_{DDV}, мкА Normal (BUFFER ON) VREF+, Normal (BUFFER OFF)VREF+			170 160	

Таблица 42.18 – Характеристики блока USB PHY

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра	
		Не менее	Не более
Выходное напряжение низкого уровня передатчика USB в режимах LS/FS, В, R_L = 1,425кОм на U_{CC1} = 3,6 В	U_{FSOL}	–	0,3
Выходное напряжение высокого уровня передатчика USB в режимах LS/FS, В, R_L = 14,25 кОм на GND	U_{FSOH}	2,8	
Длительность фронта/спада выходного дифференциального сигнала передатчика USB в режиме LS, нс, C_L = (50 – 600) пФ	t_{LR}/t_{LF}	75	300
Длительность фронта/спада выходного дифференциального сигнала передатчика USB в режиме FS, нс, C_L = 50 пФ,	t_{FR}/t_{FF}	4	20

Таблица 42.19 – Характеристики блока Ethernet

Наименование параметра, единица измерения	Буквенное обозначение	Норма параметра		
		Не менее	Номинал	Не более
Опорная тактовая частота блока Ethernet 10/100/1000, МГц, режим 10/100 Mbps	F_{REF1}		25	
Опорная тактовая частота блока Ethernet 10/100/1000, МГц, режим 1000 Mbps	F_{REF2}		125	

43 Заключение

В настоящем техническом описании представлено описание архитектуры, функционального построения и периферии микроконтроллера К1921ВГ1Т. Техническое описание может служить практическим руководством по применению микроконтроллера для разработчиков систем на его основе и программистов.

Приложение А
(обязательное)
Регистры микроконтроллера

Для каждого блока отдельно приведены абсолютные базовые адреса, а в описаниях регистров показаны смещения относительно его базового адреса, если не указано иное. Абсолютный адрес регистра вычисляется путем сложения абсолютного базового адреса блока и смещения этого регистра.

А.1 Регистры блока управления сбросом и синхронизацией RCU

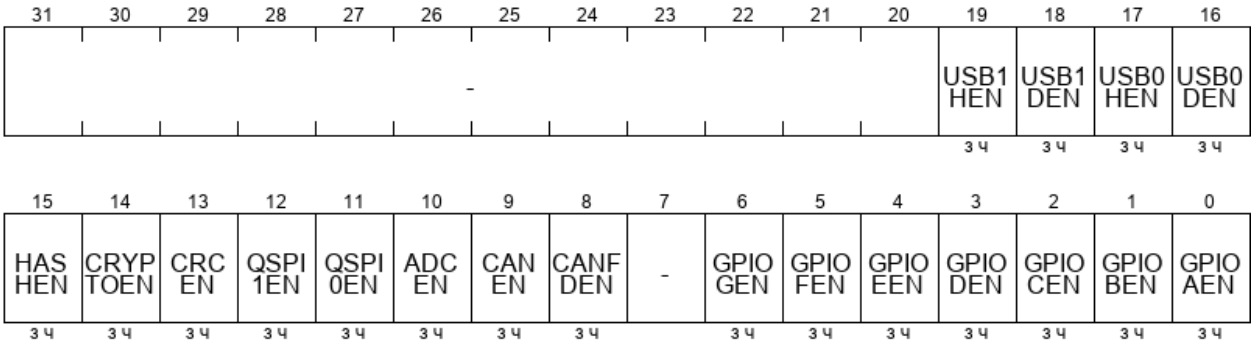
Базовый адрес: 5000_4000h

Смещение:	+ 50h (UARTCFG)	Регистры настройки тактирования UART
	+ 70h (SPICFG)	Регистры настройки тактирования SPI
	+ A0h (I2SCFG)	Регистры настройки тактирования I2S
	+ B0h (USBCFG)	Регистры настройки тактирования USB
	+ 100h (PLL0)	Регистры настройки PLL0
	+ 110h (PLL1)	Регистры настройки PLL1
	+ 120h (PLL2)	Регистры настройки PLL2

Примечание – n – номер блока.

CGCFGAHB – регистр разрешения тактирования периферийных блоков шины AHB

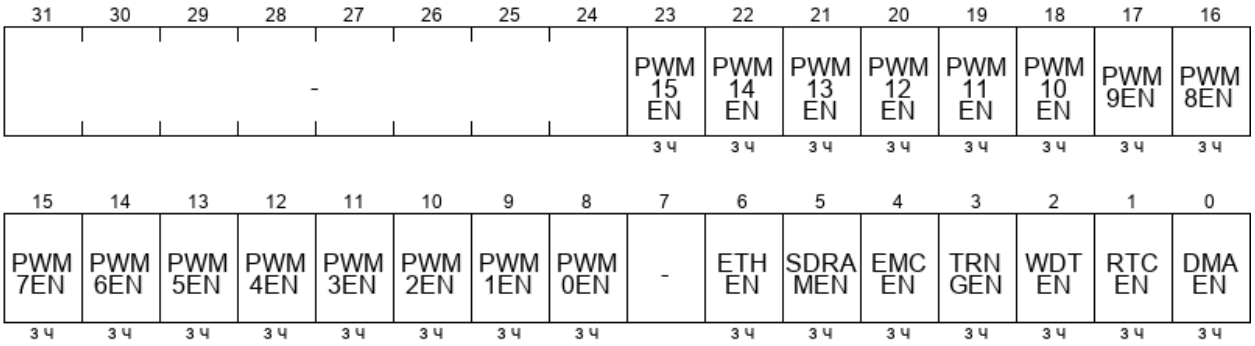
Смещение: + 00h
Сброс: 0000_0000h



Поле	Биты	Описание
USB1HEN	19	Бит разрешения тактирования USB1 HC
USB1DEN	18	Бит разрешения тактирования USB1 DC
USB0HEN	17	Бит разрешения тактирования USB0 HC
USB0DEN	16	Бит разрешения тактирования USB0 DC
HASHEN	15	Бит разрешения тактирования HASH
CRYPTOEN	14	Бит разрешения тактирования CRYPTO
CRCEN	13	Бит разрешения тактирования CRC
QSPI1EN	12	Бит разрешения тактирования QSPI1
QSPI0EN	11	Бит разрешения тактирования QSPI0
ADCEN	10	Бит разрешения тактирования ADC
CANEN	9	Бит разрешения тактирования CAN
CANFDEN	8	Бит разрешения тактирования CANFD (CANFD0 – CANFD3)
GPIOGEN	6	Бит разрешения тактирования порта G
GPIOFEN	5	Бит разрешения тактирования порта F
GPIOEEN	4	Бит разрешения тактирования порта E
GPIODEN	3	Бит разрешения тактирования порта D
GPIOCEN	2	Бит разрешения тактирования порта C
GPIOBEN	1	Бит разрешения тактирования порта B
GPIOAEN	0	Бит разрешения тактирования порта A
–	31-20, 7	Зарезервировано

CGCFGAPB0 – регистр разрешения тактирования периферийных блоков шины APB

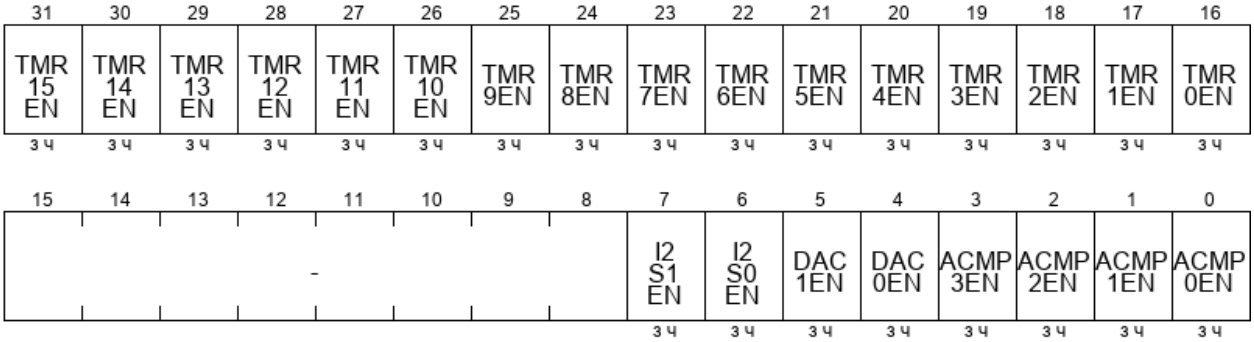
Смещение: + 04h
Сброс: 0000_0000h



Поле	Биты	Описание
PWM15EN	23	Бит разрешения тактирования PWM15
PWM14EN	22	Бит разрешения тактирования PWM14
PWM13EN	21	Бит разрешения тактирования PWM13
PWM12EN	20	Бит разрешения тактирования PWM12
PWM11EN	19	Бит разрешения тактирования PWM11
PWM10EN	18	Бит разрешения тактирования PWM10
PWM9EN	17	Бит разрешения тактирования PWM9
PWM8EN	16	Бит разрешения тактирования PWM8
PWM7EN	15	Бит разрешения тактирования PWM7
PWM6EN	14	Бит разрешения тактирования PWM6
PWM5EN	13	Бит разрешения тактирования PWM5
PWM4EN	12	Бит разрешения тактирования PWM4
PWM3EN	11	Бит разрешения тактирования PWM3
PWM2EN	10	Бит разрешения тактирования PWM2
PWM1EN	9	Бит разрешения тактирования PWM1
PWM0EN	8	Бит разрешения тактирования PWM0
ETHEN	6	Бит разрешения тактирования Ethernet
SDRAMEN	5	Бит разрешения тактирования SDRAM
EMCEN	4	Бит разрешения тактирования EMC
TRNGEN	3	Бит разрешения тактирования TRNG
WDTEN	2	Бит разрешения тактирования WDT
RTCEN	1	Бит разрешения тактирования RTC
DMAEN	0	Бит разрешения тактирования DMA
–	31-24, 7	Зарезервировано

CGCFGAPB1 – регистр разрешения тактирования периферийных блоков шины APB

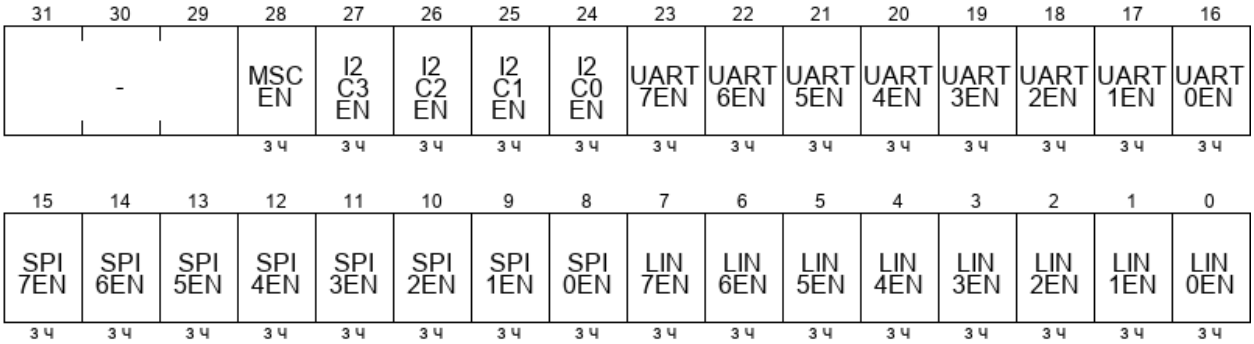
Смещение: + 08h
Сброс: 0000_0000h



Поле	Биты	Описание
TMR15EN	31	Бит разрешения тактирования TMR15
TMR14EN	30	Бит разрешения тактирования TMR14
TMR13EN	29	Бит разрешения тактирования TMR13
TMR12EN	28	Бит разрешения тактирования TMR12
TMR11EN	27	Бит разрешения тактирования TMR11
TMR10EN	26	Бит разрешения тактирования TMR10
TMR9EN	25	Бит разрешения тактирования TMR9
TMR8EN	24	Бит разрешения тактирования TMR8
TMR7EN	23	Бит разрешения тактирования TMR7
TMR6EN	22	Бит разрешения тактирования TMR6
TMR5EN	21	Бит разрешения тактирования TMR5
TMR4EN	20	Бит разрешения тактирования TMR4
TMR3EN	19	Бит разрешения тактирования TMR3
TMR2EN	18	Бит разрешения тактирования TMR2
TMR1EN	17	Бит разрешения тактирования TMR1
TMR0EN	16	Бит разрешения тактирования TMR0
I2S1EN	7	Бит разрешения тактирования I2S1
I2S0EN	6	Бит разрешения тактирования I2S0
DAC1EN	5	Бит разрешения тактирования DAC1
DAC0EN	4	Бит разрешения тактирования DAC0
ACMP3EN	3	Бит разрешения тактирования ACMP3
ACMP2EN	2	Бит разрешения тактирования ACMP2
ACMP1EN	1	Бит разрешения тактирования ACMP1
ACMP0EN	0	Бит разрешения тактирования ACMP0
–	15-8	Зарезервировано

CGCFGAPB2 – регистр разрешения тактирования периферийных блоков шины APB

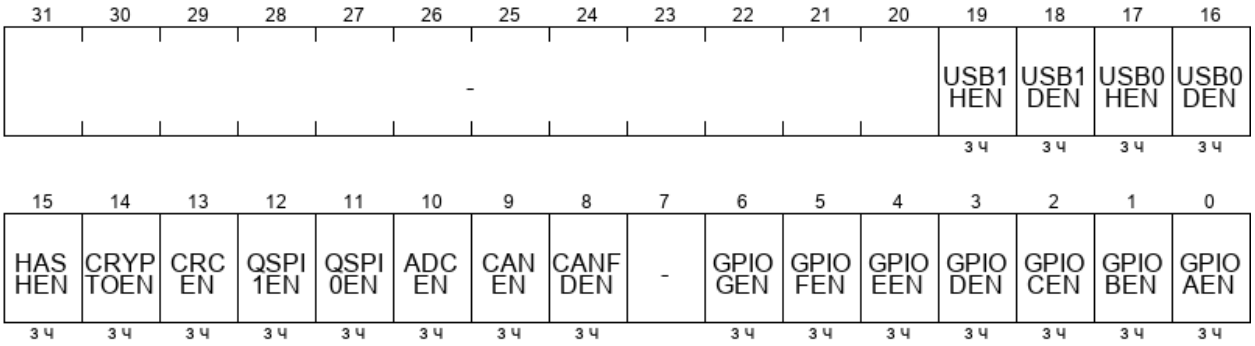
Смещение: + 0Ch
Сброс: 0000_0000h



Поле	Биты	Описание
MSCEN	28	Бит разрешения тактирования MSC
I2C3EN	27	Бит разрешения тактирования I2C3
I2C2EN	26	Бит разрешения тактирования I2C2
I2C1EN	25	Бит разрешения тактирования I2C1
I2C0EN	24	Бит разрешения тактирования I2C0
UART7EN	23	Бит разрешения тактирования UART7
UART6EN	22	Бит разрешения тактирования UART6
UART5EN	21	Бит разрешения тактирования UART5
UART4EN	20	Бит разрешения тактирования UART4
UART3EN	19	Бит разрешения тактирования UART3
UART2EN	18	Бит разрешения тактирования UART2
UART1EN	17	Бит разрешения тактирования UART1
UART0EN	16	Бит разрешения тактирования UART0
SPI7EN	15	Бит разрешения тактирования SPI7
SPI6EN	14	Бит разрешения тактирования SPI6
SPI5EN	13	Бит разрешения тактирования SPI5
SPI4EN	12	Бит разрешения тактирования SPI4
SPI3EN	11	Бит разрешения тактирования SPI3
SPI2EN	10	Бит разрешения тактирования SPI2
SPI1EN	9	Бит разрешения тактирования SPI1
SPI0EN	8	Бит разрешения тактирования SPI0
LIN7EN	7	Бит разрешения тактирования LIN7
LIN6EN	6	Бит разрешения тактирования LIN6
LIN5EN	5	Бит разрешения тактирования LIN5
LIN4EN	4	Бит разрешения тактирования LIN4
LIN3EN	3	Бит разрешения тактирования LIN3
LIN2EN	2	Бит разрешения тактирования LIN2
LIN1EN	1	Бит разрешения тактирования LIN1
LIN0EN	0	Бит разрешения тактирования LIN0
–	31-29	Зарезервировано

RSTDISAHB – регистр включения периферийных блоков шины AHB

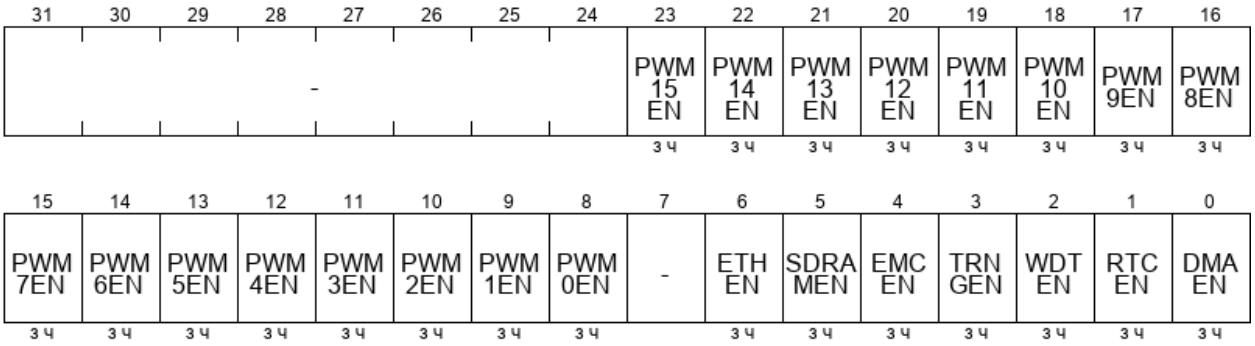
Смещение: + 10h
Сброс: 0000_0000h



Поле	Биты	Описание
USB1HEN	19	Бит включения USB1 HC
USB1DEN	18	Бит включения USB1 DC
USB0HEN	17	Бит включения USB0 HC
USB0DEN	16	Бит включения USB0 DC
HASHEN	15	Бит включения HASH
CRYPTOEN	14	Бит включения CRYPTO
CRCEN	13	Бит включения CRC
QSPI1EN	12	Бит включения QSPI1
QSPI0EN	11	Бит включения QSPI0
ADCEN	10	Бит включения ADC
CANEN	9	Бит включения CAN
CANFDEN	8	Бит включения CANFD (CANFD0 – CANFD3)
GPIOGEN	6	Бит включения порта G
GPIOFEN	5	Бит включения порта F
GPIOEEN	4	Бит включения порта E
GPIODEN	3	Бит включения порта D
GPIOCEN	2	Бит включения порта C
GPIOBEN	1	Бит включения порта B
GPIOAEN	0	Бит включения порта A
–	31-20, 7	Зарезервировано

RSTDISAPB0 – регистр включения периферийных блоков шины APB

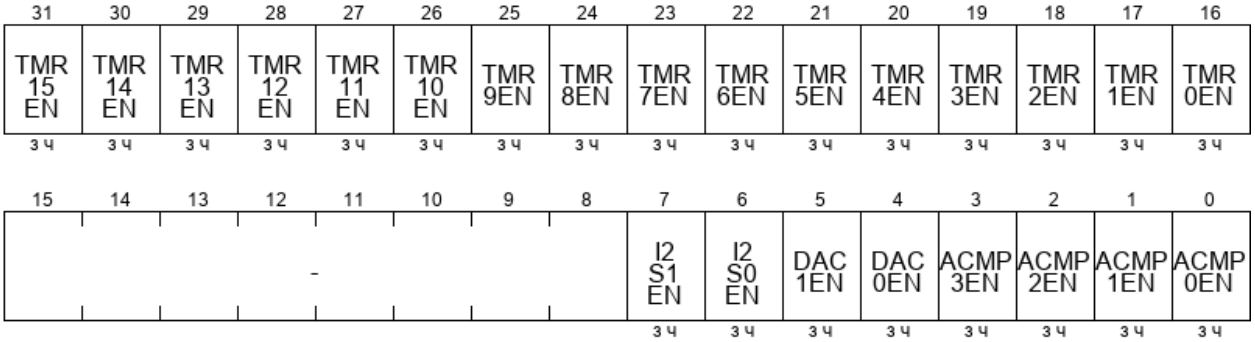
Смещение: + 14h
Сброс: 0000_0000h



Поле	Биты	Описание
PWM15EN	23	Бит включения PWM15
PWM14EN	22	Бит включения PWM14
PWM13EN	21	Бит включения PWM13
PWM12EN	20	Бит включения PWM12
PWM11EN	19	Бит включения PWM11
PWM10EN	18	Бит включения PWM10
PWM9EN	17	Бит включения PWM9
PWM8EN	16	Бит включения PWM8
PWM7EN	15	Бит включения PWM7
PWM6EN	14	Бит включения PWM6
PWM5EN	13	Бит включения PWM5
PWM4EN	12	Бит включения PWM4
PWM3EN	11	Бит включения PWM3
PWM2EN	10	Бит включения PWM2
PWM1EN	9	Бит включения PWM1
PWM0EN	8	Бит включения PWM0
ETHEN	6	Бит включения Ethernet
SDRAMEN	5	Бит включения SDRAM
EMCEN	4	Бит включения EMC
TRNGEN	3	Бит включения TRNG
WDTEN	2	Бит включения WDT
RTCEN	1	Бит включения RTC
DMAEN	0	Бит включения DMA
–	31-24, 7	Зарезервировано

RSTDISAPB1 – регистр включения периферийных блоков шины APB

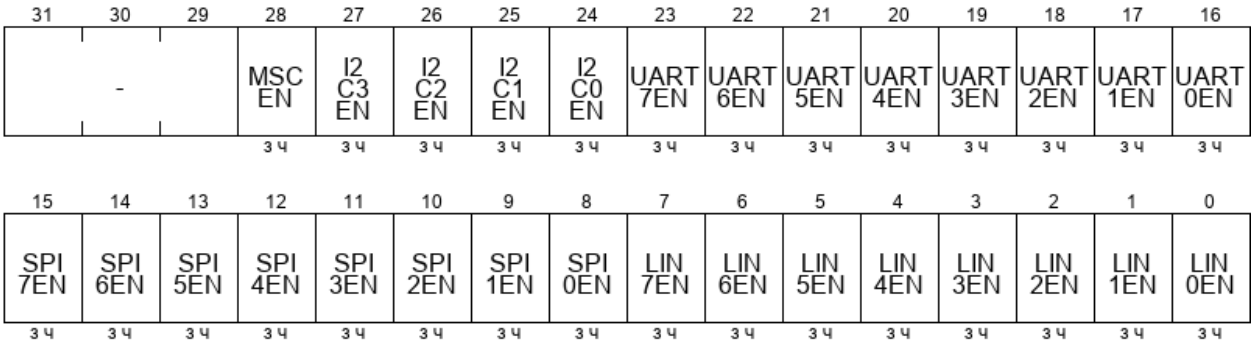
Смещение: + 18h
Сброс: 0000_0000h



Поле	Биты	Описание
TMR15EN	31	Бит включения TMR15
TMR14EN	30	Бит включения TMR14
TMR13EN	29	Бит включения TMR13
TMR12EN	28	Бит включения TMR12
TMR11EN	27	Бит включения TMR11
TMR10EN	26	Бит включения TMR10
TMR9EN	25	Бит включения TMR9
TMR8EN	24	Бит включения TMR8
TMR7EN	23	Бит включения TMR7
TMR6EN	22	Бит включения TMR6
TMR5EN	21	Бит включения TMR5
TMR4EN	20	Бит включения TMR4
TMR3EN	19	Бит включения TMR3
TMR2EN	18	Бит включения TMR2
TMR1EN	17	Бит включения TMR1
TMR0EN	16	Бит включения TMR0
I2S1EN	7	Бит включения I2S1
I2S0EN	6	Бит включения I2S0
DAC1EN	5	Бит включения DAC1
DAC0EN	4	Бит включения DAC0
ACMP3EN	3	Бит включения ACMP3
ACMP2EN	2	Бит включения ACMP2
ACMP1EN	1	Бит включения ACMP1
ACMP0EN	0	Бит включения ACMP0
–	15-8	Зарезервировано

RSTDISAPB2 – регистр включения периферийных блоков шины APB

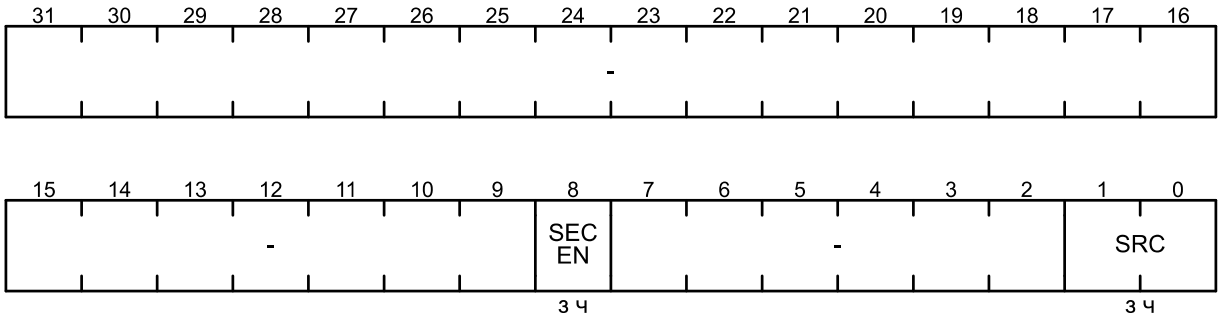
Смещение: + 1Ch
Сброс: 0000_0000h



Поле	Биты	Описание
MSCEN	28	Бит включения MSC
I2C3EN	27	Бит включения I2C3
I2C2EN	26	Бит включения I2C2
I2C1EN	25	Бит включения I2C1
I2C0EN	24	Бит включения I2C0
UART7EN	23	Бит включения UART7
UART6EN	22	Бит включения UART6
UART5EN	21	Бит включения UART5
UART4EN	20	Бит включения UART4
UART3EN	19	Бит включения UART3
UART2EN	18	Бит включения UART2
UART1EN	17	Бит включения UART1
UART0EN	16	Бит включения UART0
SPI7EN	15	Бит включения SPI7
SPI6EN	14	Бит включения SPI6
SPI5EN	13	Бит включения SPI5
SPI4EN	12	Бит включения SPI4
SPI3EN	11	Бит включения SPI3
SPI2EN	10	Бит включения SPI2
SPI1EN	9	Бит включения SPI1
SPI0EN	8	Бит включения SPI0
LIN7EN	7	Бит включения LIN7
LIN6EN	6	Бит включения LIN6
LIN5EN	5	Бит включения LIN5
LIN4EN	4	Бит включения LIN4
LIN3EN	3	Бит включения LIN3
LIN2EN	2	Бит включения LIN2
LIN1EN	1	Бит включения LIN1
LIN0EN	0	Бит включения LIN0
–	31-29	Зарезервировано

SYSCLKCFG – регистр конфигурации системной частоты

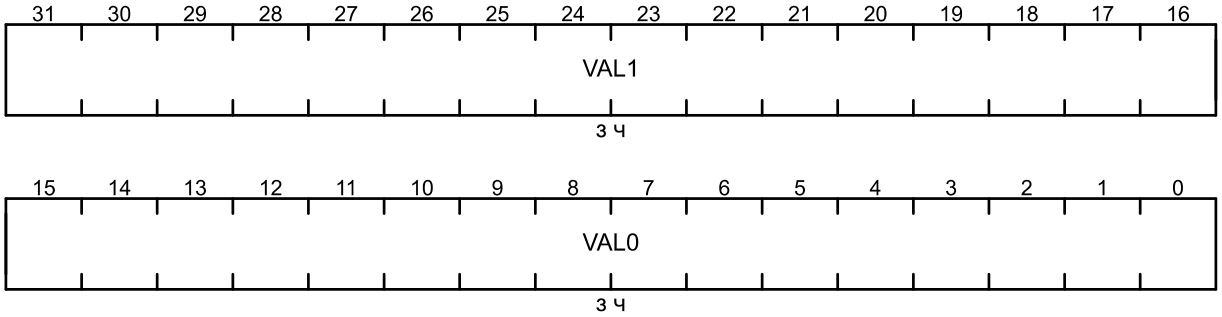
Смещение: + 20h
Сброс: 0h



Поле	Биты	Описание	
SECEN	8	Включение системы слежения за тактовыми сигналами	
SRC	1-0	Выбор источника тактового сигнала	
		00b	Сигнал HSICLK
		01b	Сигнал HSECLK
		10b	Сигнал PLL0CLK
		11b	Сигнал EXTCLK
–	31-9, 7-2	Зарезервировано	

SECCNT0 – регистр 0 счетчика слежения системной частоты

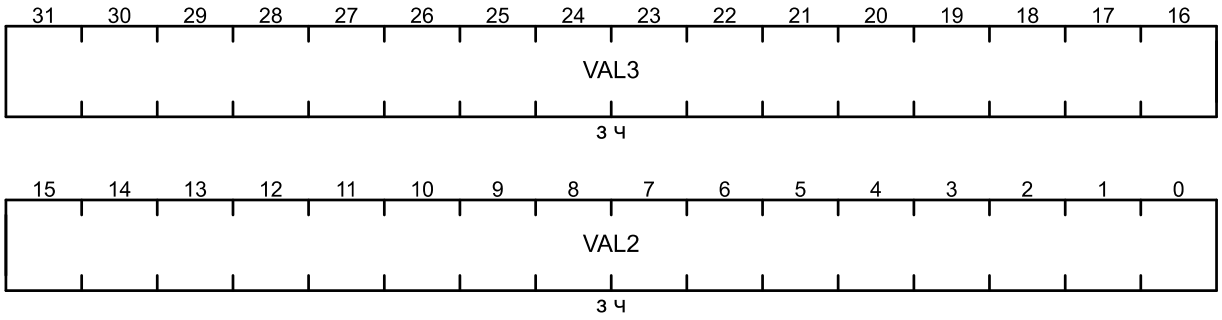
Смещение: + 24h
Сброс: FFFF_0000h



Поле	Биты	Описание
VAL1	31-16	Максимальное значение счетчика периодов для детектирования пропадания сигнала HSECLK
VAL0	15-0	Максимальное значение счетчика периодов для детектирования пропадания сигнала HSICLK

SECCNT1 – регистр 1 счетчика слежения системной частоты

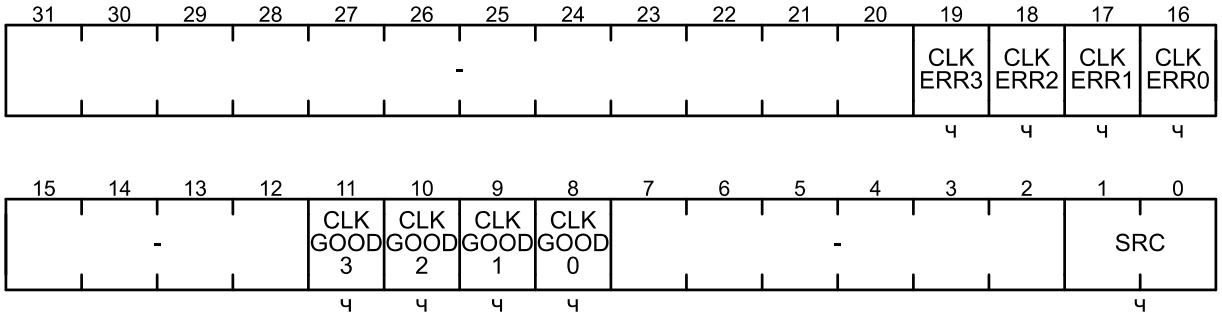
Смещение: + 28h
Сброс: FFFF_FFFFh



Поле	Биты	Описание
VAL3	31-16	Максимальное значение счетчика периодов для детектирования пропадания сигнала EXTCLK
VAL2	15-0	Максимальное значение счетчика периодов для детектирования пропадания сигнала PLL0CLK

CLKSTAT – регистр статуса системной частоты

Смещение: + 2Ch
Сброс: 00000100h

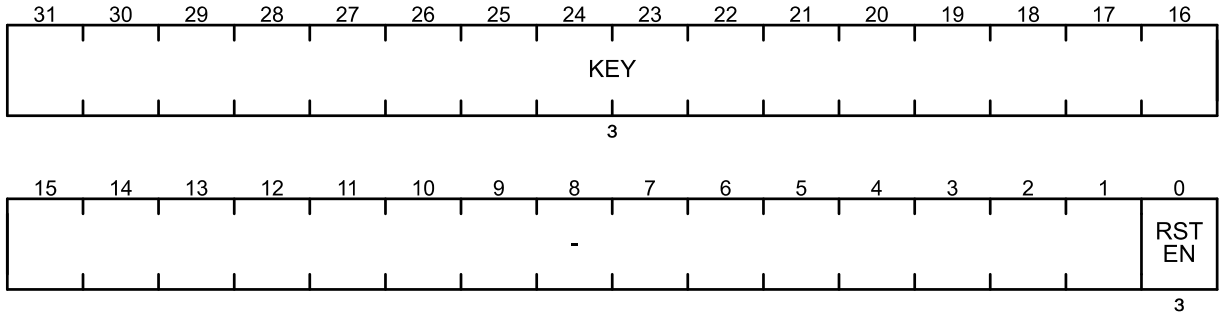


Поле	Биты	Описание
CLKERR3	19	Ошибка при работе источника тактового сигнала EXTCLK
CLKERR2	18	Ошибка при работе источника тактового сигнала PLL0CLK
CLKERR1	17	Ошибка при работе источника тактового сигнала HSECLK
CLKERR0	16	Ошибка при работе источника тактового сигнала HSICLK. Всегда читается как 0
CLKGOOD3	11	Источник тактового сигнала EXTCLK корректный
CLKGOOD2	10	Источник тактового сигнала PLLCLK корректный
CLKGOOD1	9	Источник тактового сигнала HSECLK корректный
CLKGOOD0	8	Источник тактового сигнала HSICLK корректный. Всегда читается как 1, поскольку HSICLK используется как опорный источник тактирования в системе слежения за тактовыми сигналами

Поле	Биты	Описание	
SRC	1-0	Текущий источник тактового сигнала	
		00b	Сигнал HSICLK
		01b	Сигнал HSECLK
		10b	Сигнал PLL0CLK
		11b	Сигнал EXTCLK
–	31-20, 15-12, 7-2	Зарезервировано	

RSTSYS – регистр системного сброса

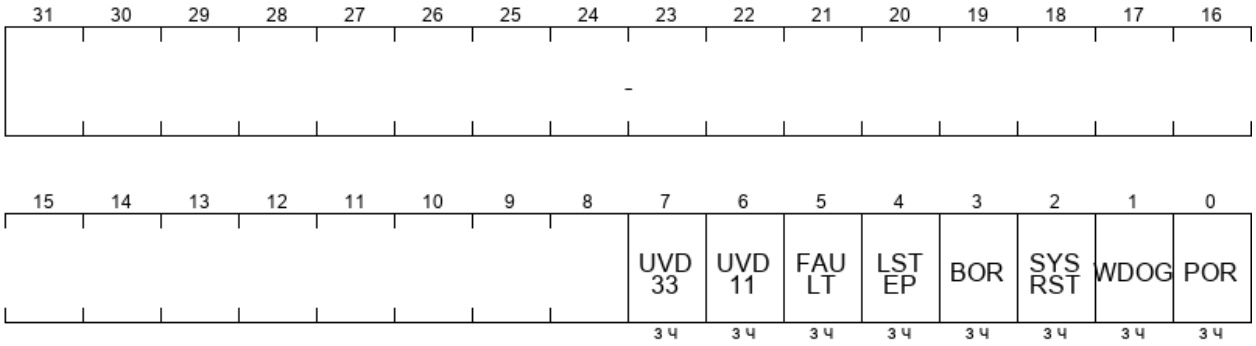
Смещение: + 30h
Сброс: 0h



Поле	Биты	Описание
KEY	31-16	Ключ сброса. Необходимо записывать значение A55Ah
RSTEN	0	Разрешение системного сброса
–	15-1	Зарезервировано

RSTSTAT – регистр статуса системного сброса от блоков

Смещение: + 34h
Сброс: 0h

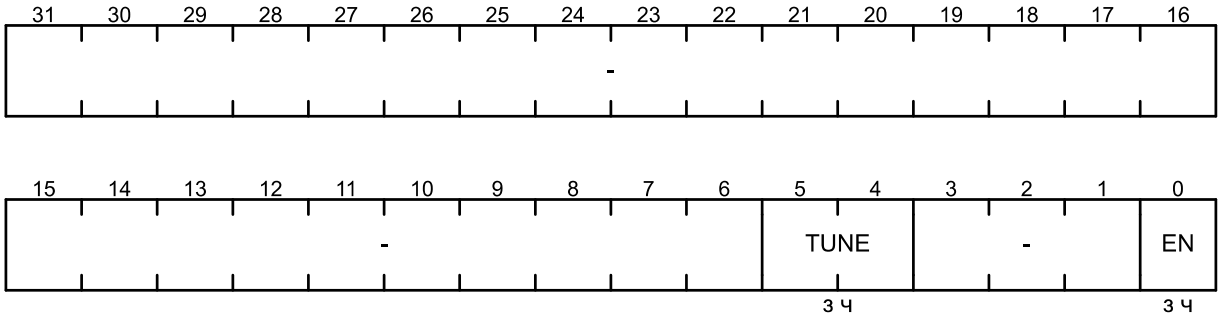


Поле	Биты	Описание
UVD33	7	Бит состояния сброса при обнаружении превышения напряжения AVCC (на 20% по сравнению с номинальным значением)
UVD11	6	Бит состояния сброса при обнаружении превышения напряжения DVCC (на 20% по сравнению с номинальным значением)
FAULT	5	Бит состояния сброса при обнаружении ошибки по таймауту при возникновении неисправности
LSTEP	4	Бит состояния сброса при изменении режима работы ядер
BOR	3	Бит состояния сброса от BOR
SYSRST	2	Бит состояния системного сброса
WDOG	1	Бит состояния сброса от WatchDog
POR	0	Бит состояния сброса от POR
–	31-4	Зарезервировано

Примечание – Биты регистра RSTSTAT сбрасываются записью единицы.

BORCFG – регистр настройки BOR

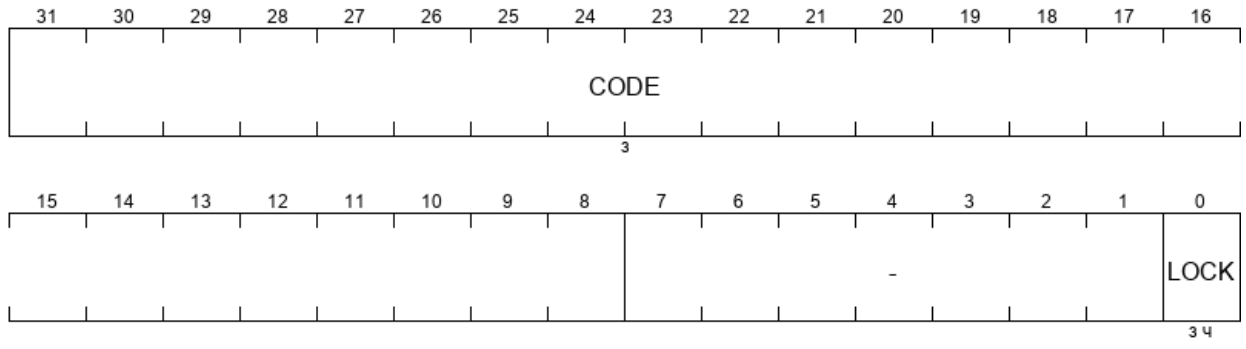
Смещение: + 38h
Сброс: 0h



Поле	Биты	Описание		
TUNE	5-4	Поле регулировки порога срабатывания BOR		
		Значение в поле TUNE	Порог срабатывания при росте напряжения, В	Порог срабатывания при спаде напряжения, В
		00b	3,0	2,9
		01b	2,7	2,6
		10b	2,2	2,1
		11b	Зарезервировано	
EN	0	Бит включения BOR		
–	31-6, 3-1	Зарезервировано		

LOCK – регистр блокировки конфигурации RCU

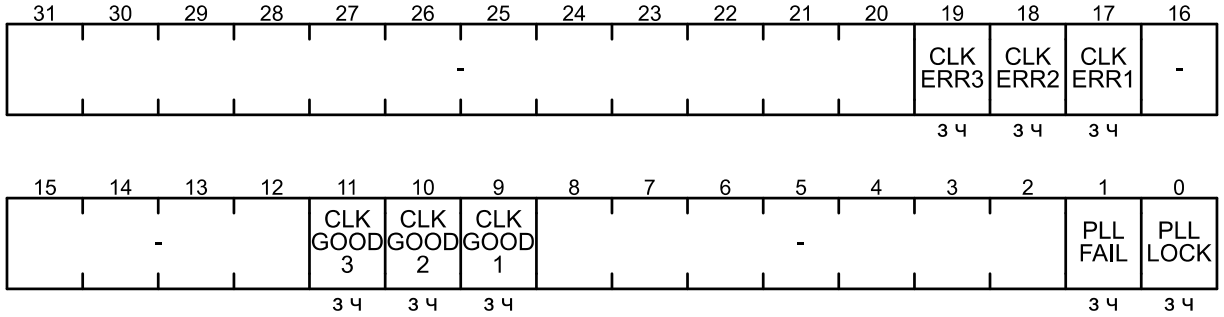
Смещение: + 3Ch
Сброс: 0h



Поле	Биты	Описание
CODE	31-8	Поле для записи ключа. Значение ключа 0xACCE55
LOCK	0	Бит блокировки / разблокировки записи в другие регистры RCU. Установка / сброс данного бита осуществляется совместно с записью ключа в поле CODE
–	7-1	Зарезервировано

INTEN – регистр включения прерываний

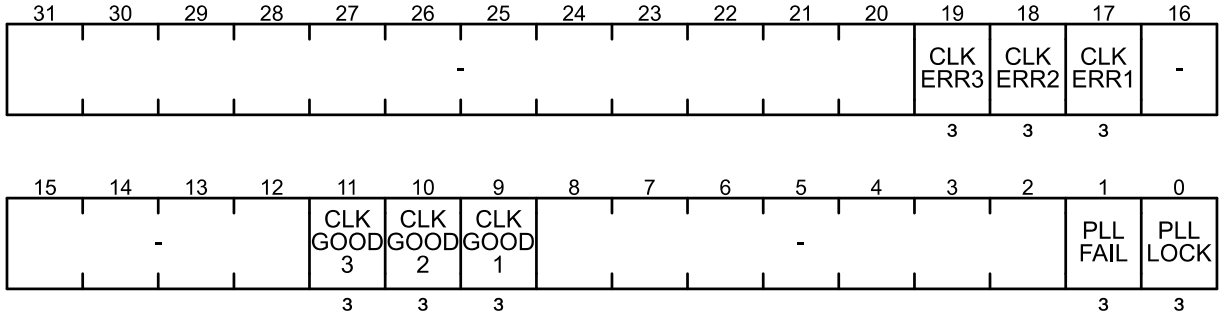
Смещение: + 40h
Сброс: 0h



Поле	Биты	Описание
CLKERR3	19	Бит включения прерывания по ошибке установления корректного сигнала EXTCLK
CLKERR2	18	Бит включения прерывания по ошибке установления корректного сигнала PLL0CLK
CLKERR1	17	Бит включения прерывания по ошибке установления корректного сигнала HSECLK
CLKGOOD3	11	Бит включения прерывания по установлению корректного сигнала EXTCLK
CLKGOOD2	10	Бит включения прерывания по установлению корректного сигнала PLL0CLK
CLKGOOD1	9	Бит включения прерывания по установлению корректного сигнала HSECLK
PLLFAIL	1	Бит включения прерывания по ошибке установления бита LOCK
PLLLOCK	0	Бит включения прерывания по установлению бита LOCK
–	31-20, 16-12, 8-2	Зарезервировано

INTSTAT – регистр статуса прерываний

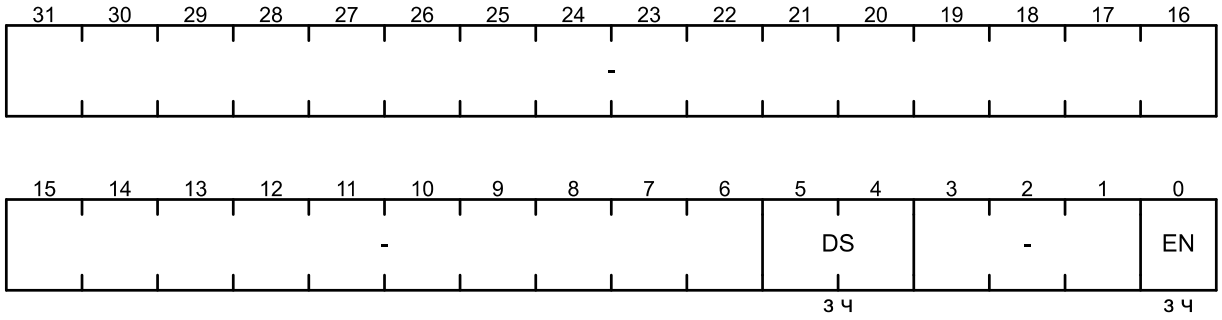
Смещение: + 44h
Сброс: 0h



Поле	Биты	Описание
CLKERR3	19	Бит статуса прерывания по ошибке установления корректного сигнала EXTCLK
CLKERR2	18	Бит статуса прерывания по ошибке установления корректного сигнала PLL0CLK
CLKERR1	17	Бит статуса прерывания по ошибке установления корректного сигнала HSECLK
CLKGOOD3	11	Бит статуса прерывания по установлению корректного сигнала EXTCLK
CLKGOOD2	10	Бит статуса прерывания по установлению корректного сигнала PLL0CLK
CLKGOOD1	9	Бит статуса прерывания по установлению корректного сигнала HSECLK
PLLFAIL	1	Бит статуса прерывания по ошибке установления бита LOCK
PLLLOCK	0	Бит статуса прерывания по установлению бита LOCK
–	31-20, 16-12, 8-2	Зарезервировано

HSECFG – регистр настройки осциллятора HSE

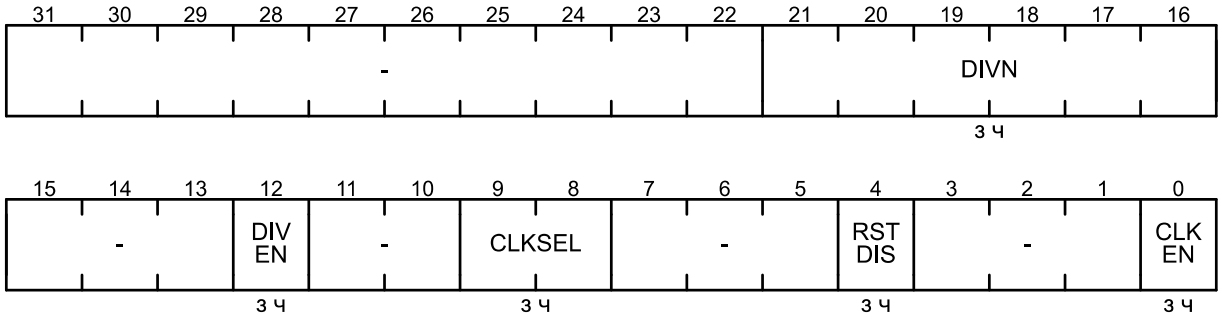
Смещение: + 48h
Сброс: 0000_0001h



Поле	Биты	Описание
DS	5-4	Поле выбора управляющего тока для выводов
EN	0	Бит включения осциллятора HSE
–	31-6, 3-1	Зарезервировано

UARTCFG – массив регистров настройки тактирования и сброса UART

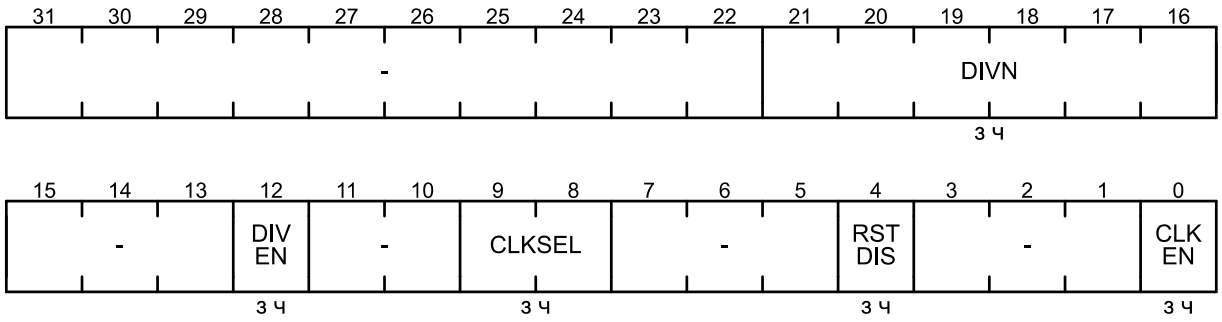
Смещение: UARTCFG + (4*n)h, где n – номер блока 0-7
Сброс: 0h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал HSICLK (4 МГц)
		01b	Сигнал HSECLK (XTAL)
		10b	Сигнал PLL0CLK с выхода PLL0
		11b	Сигнал EXTCLK. Внешний тактовый сигнал
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса	
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-10, 7-5, 3-1	Зарезервировано	

SPICFG – массив регистров настройки SPI

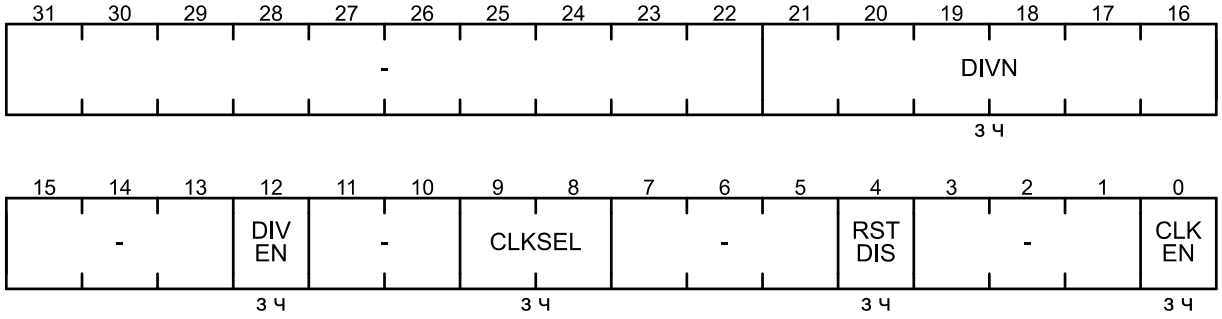
Смещение: SPICFG+ (4*n)h, где n – номер блока 0-7
Сброс: 0h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал HSICLK (4 МГц)
		01b	Сигнал HSECLK (XTAL)
		10b	Сигнал PLL0CLK с выхода PLL0
		11b	Сигнал EXTCLK. Внешний тактовый сигнал
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса	
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-10, 7-5, 3-1	Зарезервировано	

ADCCFG – регистр настройки ADC

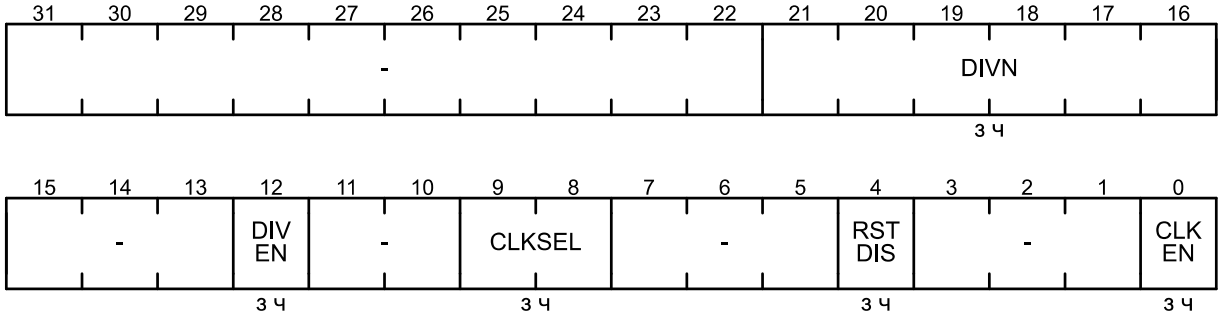
Смещение: + 90h
Сброс: 0h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал HSICLK (4 МГц)
		01b	Сигнал HSECLK (XTAL)
		10b	Сигнал PLL0CLK с выхода PLL0
		11b	Сигнал EXTCLK. Внешний тактовый сигнал
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса	
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-10, 7-5, 3-1	Зарезервировано	

WDTCFG – регистр настройки сторожевого таймера

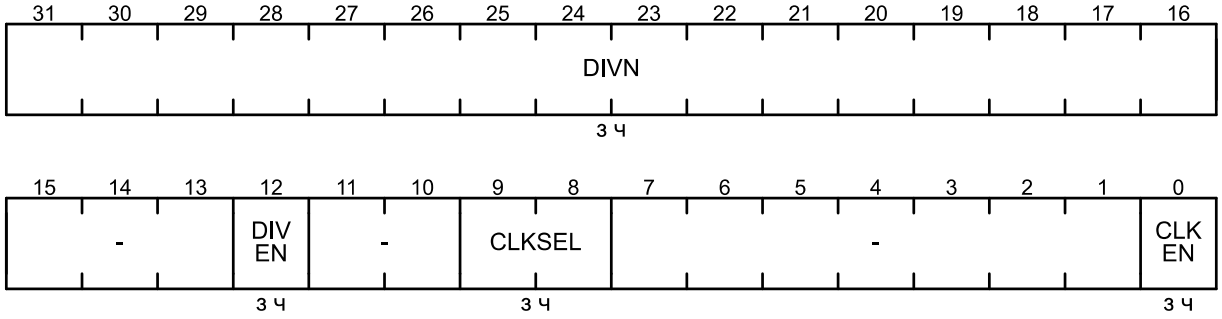
Смещение: + 94h
Сброс: 003F_1000h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал HSICLK (4 МГц)
		01b	Сигнал HSECLK (XTAL)
		10b	Сигнал PLL0CLK с выхода PLL0
		11b	Сигнал EXTCLK. Внешний тактовый сигнал
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса	
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-10, 7-5, 3-1	Зарезервировано	

CLKOUTCFG – регистр настройки вывода тактового сигнала

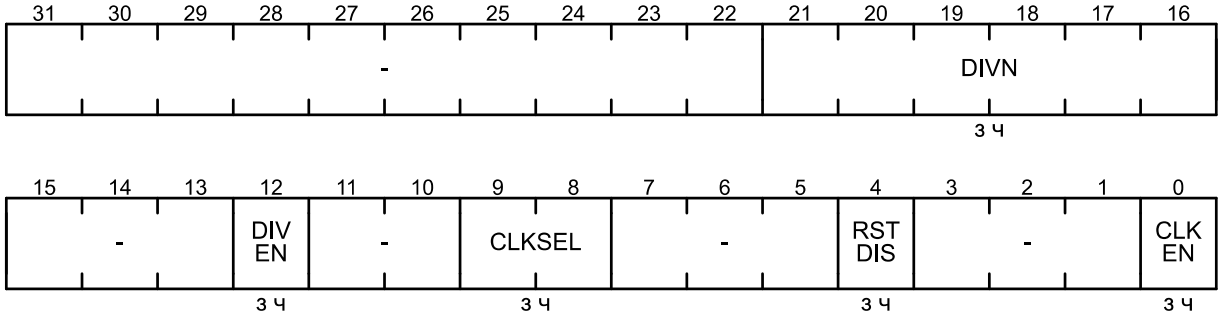
Смещение: + 98h
Сброс: 0h



Поле	Биты	Описание	
DIVN	31-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2 × (DIVN + 1).	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал HSICLK (4 МГц)
		01b	Сигнал HSECLK (XTAL)
		10b	Сигнал PLL0CLK с выхода PLL0
		11b	Сигнал RTCCLK (32кГц) от блока RTC
CLKEN	0	Разрешение тактирования	
–	15-13, 11-10, 7-1	Зарезервировано	

SDRAMCFG – регистр настройки SDRAM

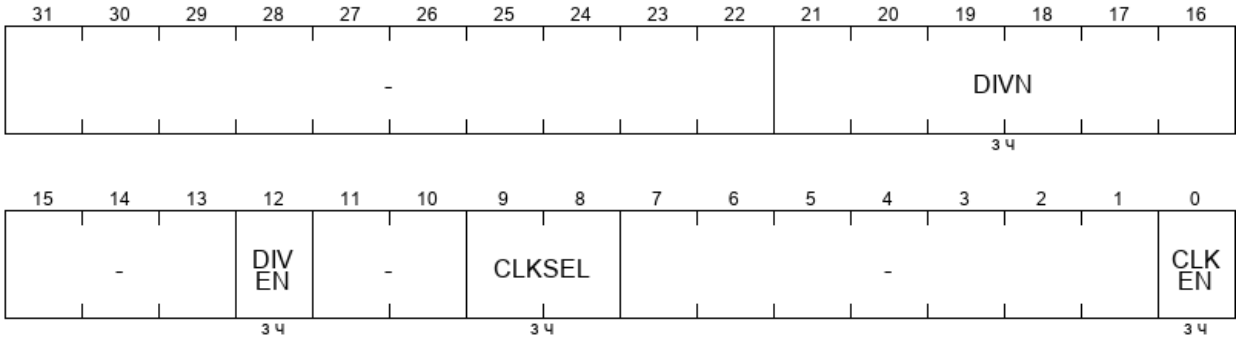
Смещение: + 9Ch
Сброс: 0h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал PLL0CLK с выхода PLL0
		01b	Сигнал PLL1CLK с выхода PLL1
		10b	Сигнал PLL2CLK с выхода PLL2
		11b	Сигнал EXTCLK. Внешний тактовый сигнал
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса	
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-10, 7-5, 3-1	Зарезервировано	

I2SCFG – массив регистров настройки I2S

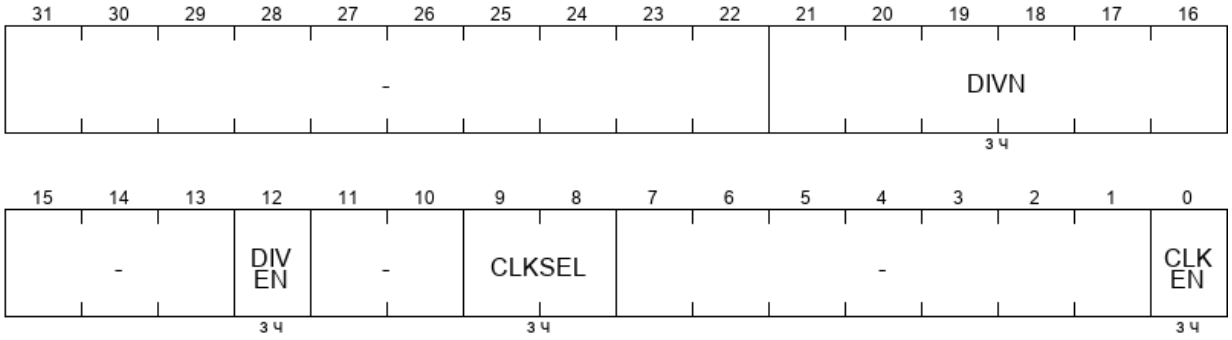
Смещение: I2SCFG+ (4*n)h, где n – номер блока 0-1
Сброс: 0h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал PLL0CLK с выхода PLL0
		01b	Сигнал PLL1CLK с выхода PLL1
		10b	Сигнал PLL2CLK с выхода PLL2
		11b	Сигнал EXTCLK. Внешний тактовый сигнал
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-10, 7-1	Зарезервировано	

USBCFG – массив регистров настройки USB

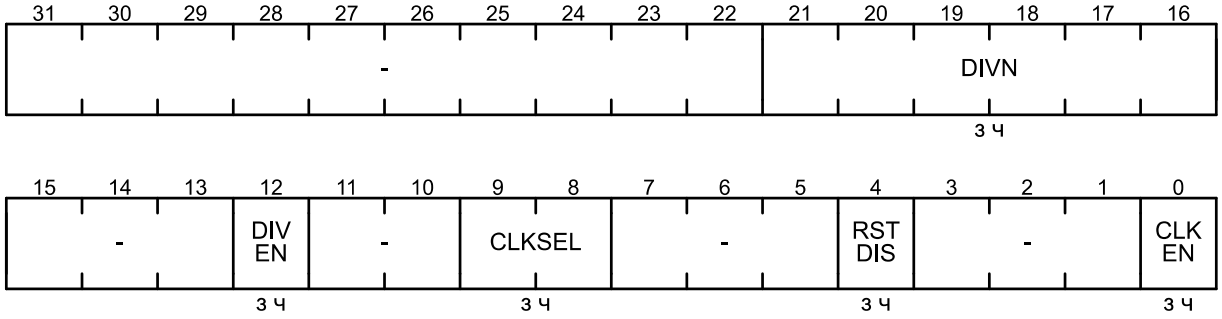
Смещение: USBCFG+ (4*n)h, где n – номер блока 0-1
Сброс: 0h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал PLL0CLK с выхода PLL0
		01b	Сигнал PLL1CLK с выхода PLL1
		10b	Сигнал PLL2CLK с выхода PLL2
		11b	Сигнал EXTCLK. Внешний тактовый сигнал
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-10, 7-1	Зарезервировано	

ETHCFG – регистр настройки Ethernet

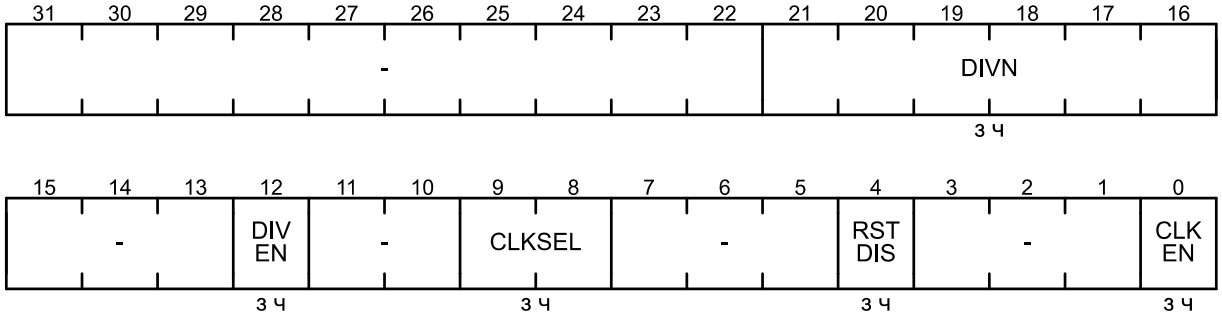
Смещение: + B8h
Сброс: 0h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал PLL0CLK с выхода PLL0
		01b	Сигнал PLL1CLK с выхода PLL1
		10b	Сигнал PLL2CLK с выхода PLL2
		11b	Сигнал EXTCLK. Внешний тактовый сигнал
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса	
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-10, 7-5, 3-1	Зарезервировано	

CANFDCFG – регистр настройки CANFD

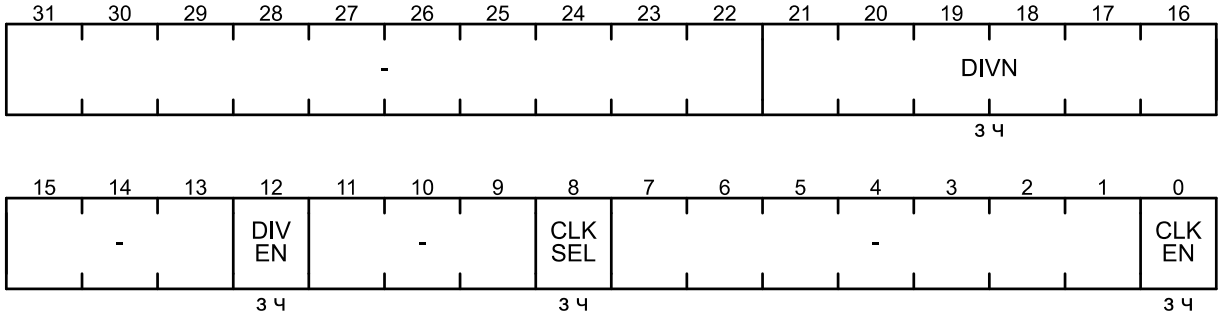
Смещение: + BCh
Сброс: 0h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	9-8	Выбор источника тактового сигнала	
		00b	Сигнал PLL0CLK с выхода PLL0
		01b	Сигнал PLL1CLK с выхода PLL1
		10b	Сигнал PLL2CLK с выхода PLL2
		11b	Сигнал EXTCLK. Внешний тактовый сигнал
RSTDIS	4	Снятие сигнала сброса. Когда сигнал в нуле, блок в состоянии сброса	
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-10, 7-5, 3-1	Зарезервировано	

FLASHCFG – регистр настройки FLASH

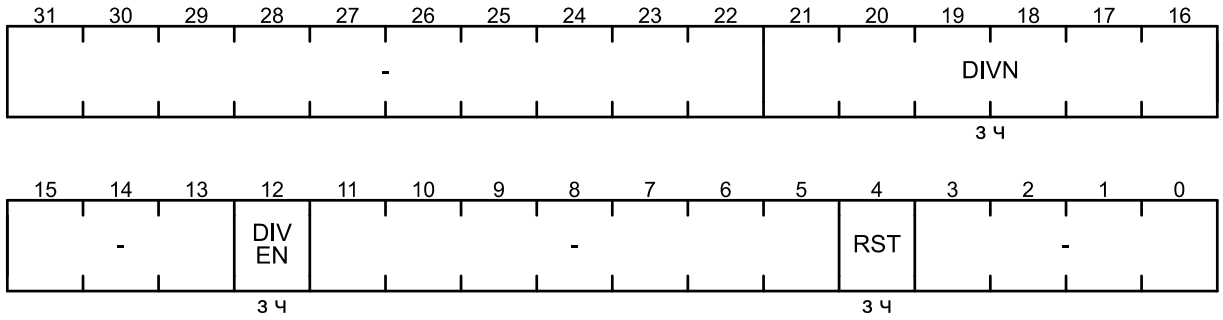
Смещение: + C0h
Сброс: 0001_0001h



Поле	Биты	Описание	
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)	
DIVEN	12	Разрешение делителя входного сигнала	
CLKSEL	8	Выбор источника тактового сигнала	
		0	Сигнал HSICLK (4 МГц)
		1	Сигнал HSECLK (XTAL)
CLKEN	0	Разрешение тактирования	
–	31-22, 15-13, 11-9, 7-1	Зарезервировано	

RTCCFG – регистр настройки делителя частоты для модуля RTC на шине APB

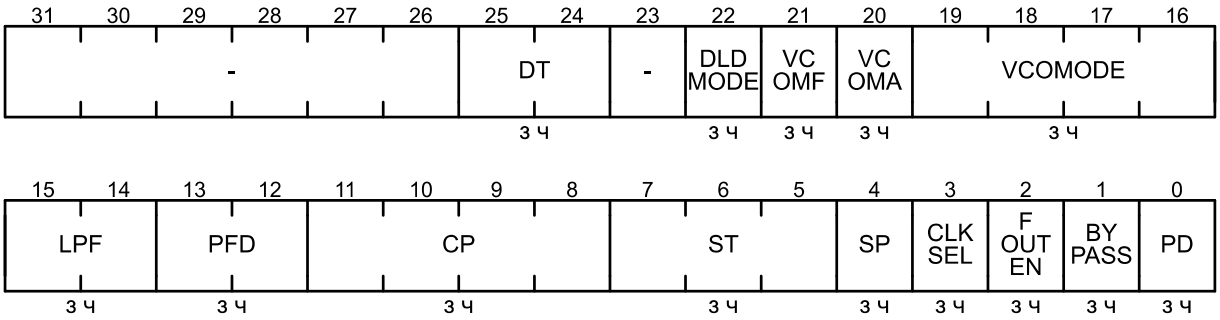
Смещение: + C0h
Сброс: 0003_1000h



Поле	Биты	Описание
DIVN	21-16	Коэффициент деления входного сигнала. Результирующий коэффициент деления N = 2×(DIVN+1)
DIVEN	12	Разрешение делителя входного сигнала
RST	4	Сигнал сброса модуля RTC. При установленном бите блок будет находиться в состоянии сброса. Следует заметить, что сброс, формируемый схемами POR/BOR, не приводит к автоматической подаче сигнала сброса на модуль RTC. Прикладному ПО необходимо самостоятельно сформировать сброс данного блока.
–	31-22, 15-13, 11-5, 3-0	Зарезервировано

PLL[n].CFG – регистр настройки PLL (PLLCFG)

Смещение: PLLn, где n – номер блока 0-2
Сброс: 0010_F001h

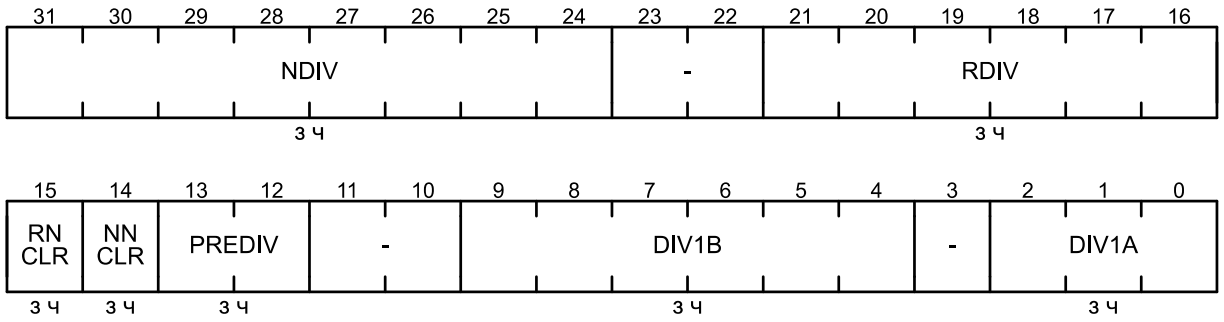


Поле	Биты	Описание	
DT	25-24	Поле установки режима работы блока рандомизации помех дробности	
		00b	Рандомизация отключена
		01b	Рандомизация на основе нефильтрованной псевдослучайной последовательности (ПСП)
		10b	Рандомизация на основе псевдослучайной последовательности, пропущенной через ФВЧ 1-го порядка
		11b	Рандомизация на основе псевдослучайной последовательности, пропущенной через ФВЧ 2-го порядка
DLDMODE	22	Бит выбора задержки срабатывания детектора захвата частоты	
		0	8 тактов
		1	16 тактов
VCOMF	21	Бит подстройки нижней границы выбранного поддиапазона ГУН. При записи нуля устанавливается минимально возможная нижняя граница выбранного поддиапазона	
VCOMA	20	Бит подстройки амплитуды сигнала ГУН. При записи нуля включается режим уменьшенной амплитуды ГУН	
VCOMODE	19-16	Поле VCOMODE[2-0] выбирает поддиапазон ГУН. При записи единицы будет выбран нижний поддиапазон. Значение VCOMODE = 0000b – запрещенное состояние. Бит VCOMODE[3] управляет режимом включения предварительного делителя частоты. VCOMODE = 0001b – Предварительный делитель исключен из цепи выходного сигнала; VCOMODE = 1001b – Предварительный делитель включен в цепь выходного сигнала.	
LPF	15-14	Управление полосой петлевого фильтра	
		00b	Наиболее широкая полоса фильтра
		…	
		11b	Наиболее узкая полоса фильтра

Поле	Биты	Описание	
PFD	13-12	Установка режима работы ЧФД и токового ключа	
		00b	Токовый ключ выключен, высокоимпедансный выход
		01b	Втекающий ток токового ключа
		10b	Вытекающий ток токового ключа
		11b	Рабочий режим
CP	11-8	Задание выходного тока для токового ключа. Итоговое значение рассчитывается как значение в поле CP + 1. Единица деления 40 мкА. Например, при записи в поле значения 0x9, значение выходного тока будет равно 0,4 мА.	
ST	7-5	Поле устанавливает режим работы сигма-дельта модулятора	
		000b	Модулятор выключен (целочисленный режим)
		001b	Активен модулятор 1-го порядка (дробный режим)
		010b	Активен модулятор 2-го порядка (дробный режим)
		011b	Активен модулятор 3-го порядка (дробный режим)
		100b	Активен модулятор 4-го порядка (дробный режим)
		Остальные комбинации являются запрещенными	
SP	4	Бит включения режима работы «Сонга-Парка» (Song-Park) сигма-дельта модулятора	
CLKSEL	3	Выбор источника опорного тактового сигнала PLL	
		0	Сигнал HSE с кварцевого резонатора
		1	Внешний сигнал EXT с ножки порта GPIO
FOUTEN	2	Разрешение выходного сигнала Fout	
BYPASS	1	Бит включения режима «на проход». Включение режима «на проход», в этом режиме на выход подается частота со входа Fref	
PD	0	Бит включения спящего режима PLL. В спящем режиме выключаются блоки токового ключа, ЧФД и ГУН. Блок PLL может в этом случае работать только на проход (режим «на проход» включается битом BYPASS)	
–	31-26, 23	Зарезервировано	

PLL[n].DIV – регистр настройки делителей PLL (PLLDIV)

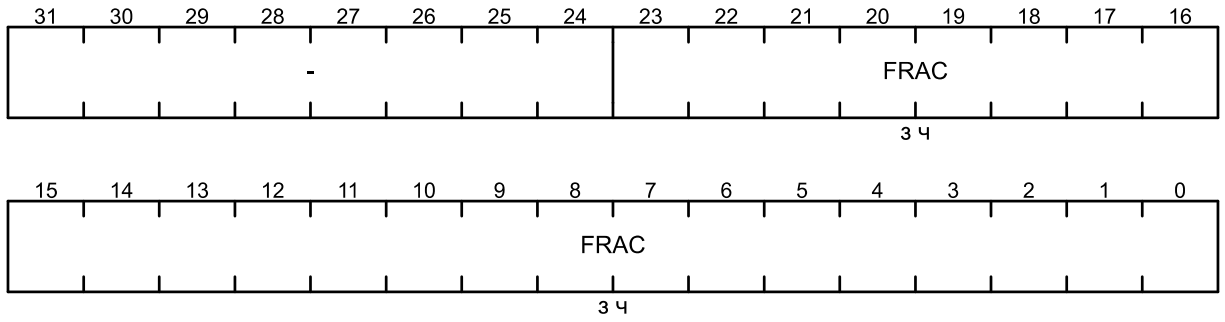
Смещение: PLLn +4h, где n – номер блока 0-2
Сброс: 0000_0000h



Поле	Биты	Описание	
NDIV	31-24	Поле установки целой части коэффициента деления N делителя в цепи обратной связи. Исключение – комбинация 0000_0000b устанавливает коэффициент деления равным 256	
RDIV	21-16	Поле установки коэффициента деления R опорного делителя. Результирующим будет значение, записанное в поле RDIV+1.	
RNCLR	15	Бит сброса R-делителя опорного сигнала. Сброс выполняется записью нуля	
NNCLR	14	Бит сброса N-делителя в цепи обратной связи. Сброс выполняется записью нуля	
PREDIV	13-12	Поле установки коэффициента деления предварительного делителя частоты ГУН	
		00b	Коэффициент деления равен 1
		01b	Коэффициент деления равен 1
		10b	Коэффициент деления равен 2
		11b	Коэффициент деления равен 4
DIV1B	9-4	Установка коэффициента деления делителя 1B. Результирующим будет значение, записанное в поле DIV1B + 1	
DIV1A	2-0	Установка коэффициента деления делителя 1A. Результирующим будет значение, записанное в поле DIV1A + 1	
–	23-22, 11-10, 3	Зарезервировано	

PLL[n].FRAC – регистр конфигурации дробного делителя PLL (PLLFRAC)

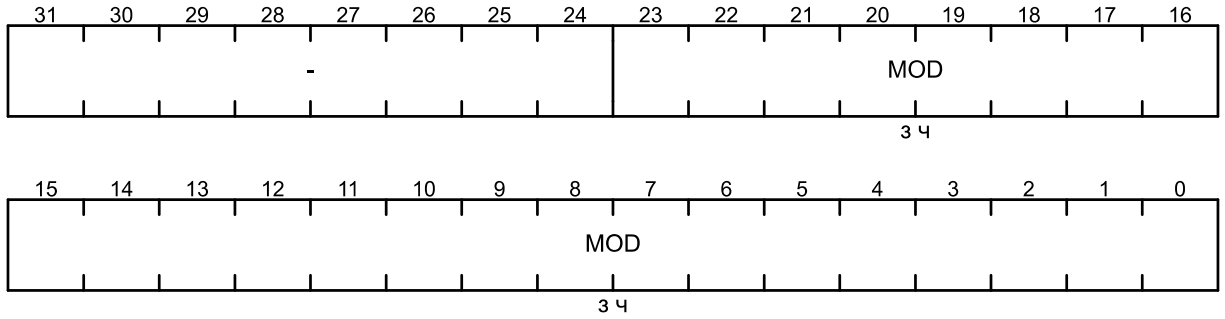
Смещение: PLLn +8h, где n – номер блока 0-2
Сброс: 0000_0000h



Поле	Биты	Описание
FRAC	23-0	Числитель дробной части коэффициента деления N
–	31-24	Зарезервировано

PLL[n].MOD – регистр конфигурации дробного делителя PLL (PLLMOD)

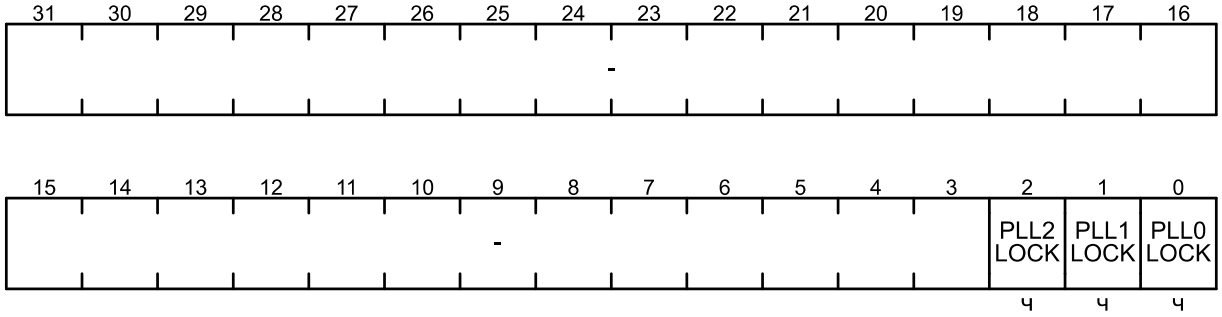
Смещение: PLLn +Ch, где n – номер блока 0-2
Сброс: 0000_0000h



Поле	Биты	Описание
MOD	23-0	Знаменатель дробной части коэффициента деления N
–	31-24	Зарезервировано

PLLSTAT – регистр статуса блоков PLL

Смещение: + 130h
Сброс: 0h



Поле	Биты	Описание
PLL2LOCK	2	Бит установившейся работы блока PLL2
PLL1LOCK	1	Бит установившейся работы блока PLL1
PLL0LOCK	0	Бит установившейся работы блока PLL0
–	31-3	Зарезервировано

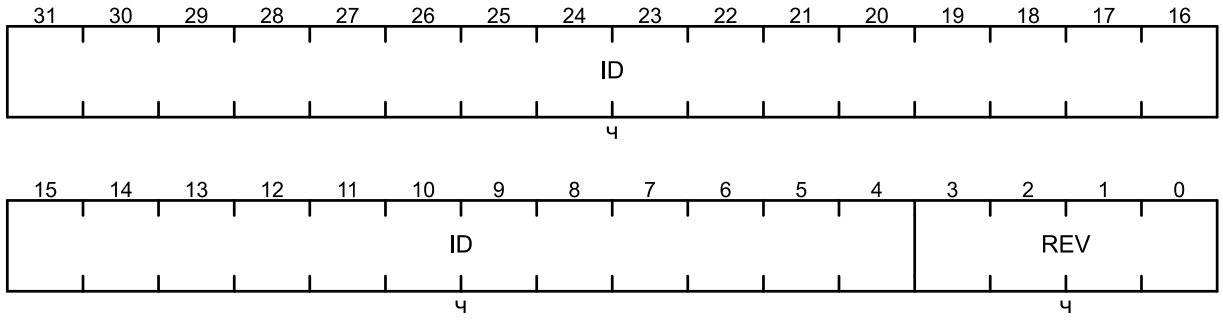
A.2 Регистры блока управления системой SIU

Базовый адрес: 5000_3000h

Смещение: + 70h (TMRMUX) Регистры мультиплексоров таймеров

CHIPID – регистр идентификации системы

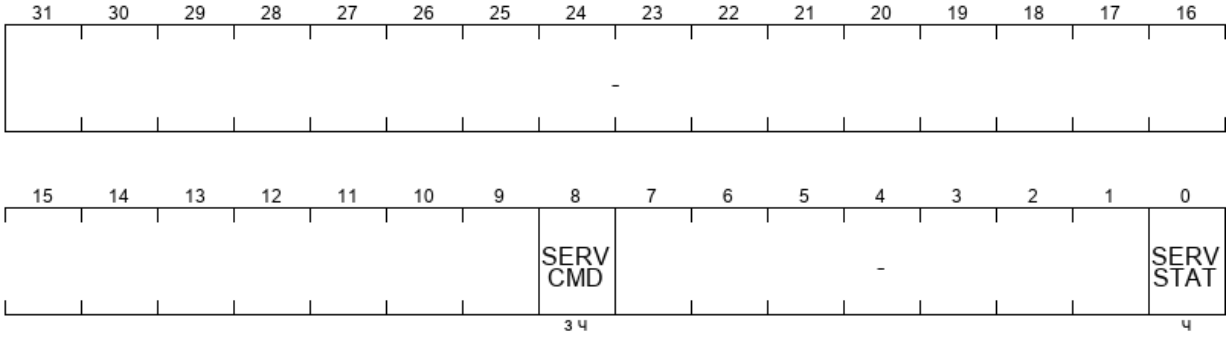
Смещение: + 00h
Сброс: 04E4_C101h



Поле	Биты	Описание
ID	31-4	Номер модели
REV	3-0	Номер ревизии

SERVCTL – регистр управления сервисным режимом

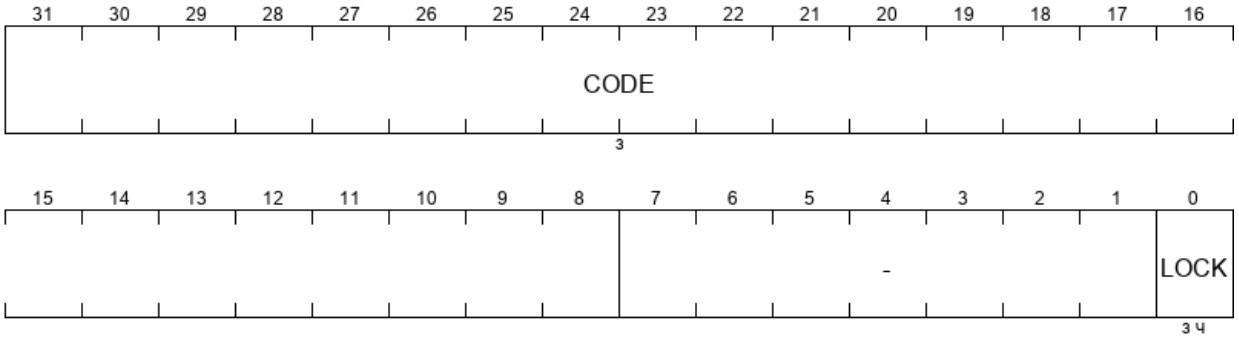
Смещение: + 04h
Сброс: 0h



Поле	Биты	Описание	
SERVCMD	8	Бит команды сервисного стирания	
		0	Команда не завершена
		1	Команда завершена
		Запись единицы активирует полное стирание всей Flash-памяти микроконтроллера	
SERVSTAT	0	Бит статуса сервисного режима	
		0	Обычный режим работы
		1	Сервисный режим. Во время сброса на выводе SERVEN была единица
–	31-9, 7-1	Зарезервировано	

LOCK – регистр блокировки конфигурации SIU

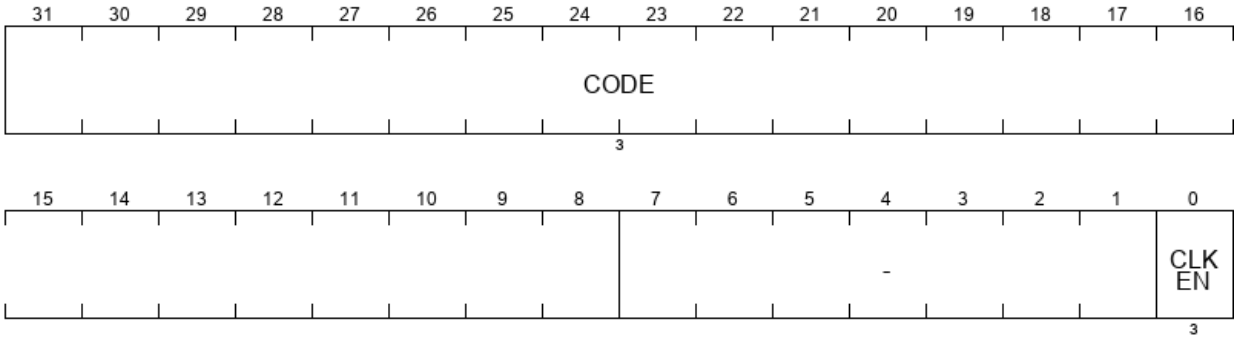
Смещение: + 0Ch
Сброс: 0h



Поле	Биты	Описание
CODE	31-8	Поле для записи ключа. Значение ключа 0xACCE55
LOCK	0	Бит блокировки / разблокировки записи в другие регистры SIU. Установка / сброс данного бита осуществляется совместно с записью ключа в поле CODE
–	7-1	Зарезервировано

CORE1_CLKEN – регистр разрешения тактирования ядра 1

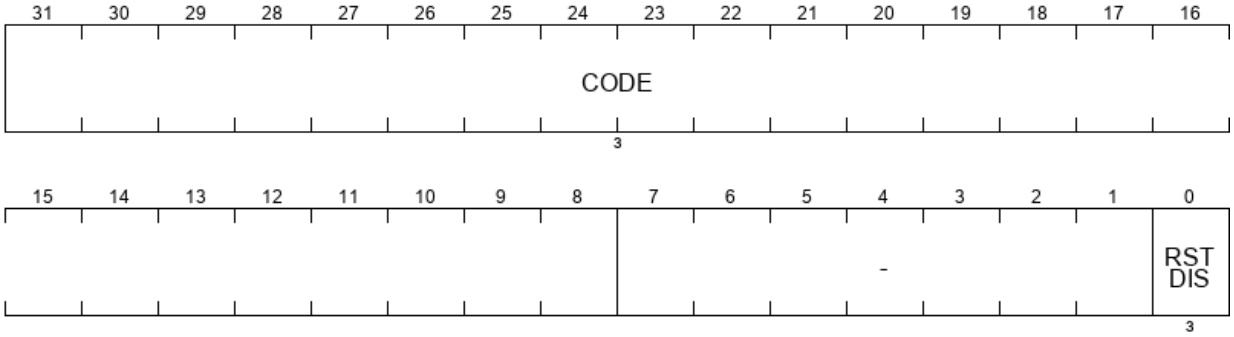
Смещение: + 10h
Сброс: 0000_0001h



Поле	Биты	Описание
CODE	31-8	Поле для записи ключа. Значение ключа 0xACCE55
CLKEN	0	Бит разрешения тактирования ядра 1 в режиме DUALCORE. Установка / сброс данного бита осуществляется совместно с записью ключа в поле CODE
–	7-1	Зарезервировано

CORE1_RSTDIS – регистр включения ядра 1

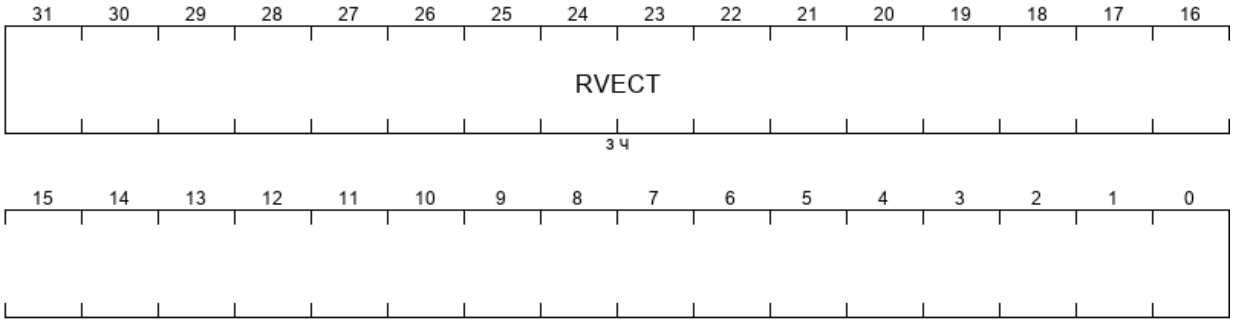
Смещение: + 14h
Сброс: 0000_0001h



Поле	Биты	Описание
CODE	31-8	Поле для записи ключа. Значение ключа 0xACCE55
RSTDIS	0	Бит включения ядра 1 в режиме DUALCORE. Установка / сброс данного бита осуществляется совместно с записью ключа в поле CODE
–	7-1	Зарезервировано

CORE1_RVECT– регистр вектора сброса для ядра 1

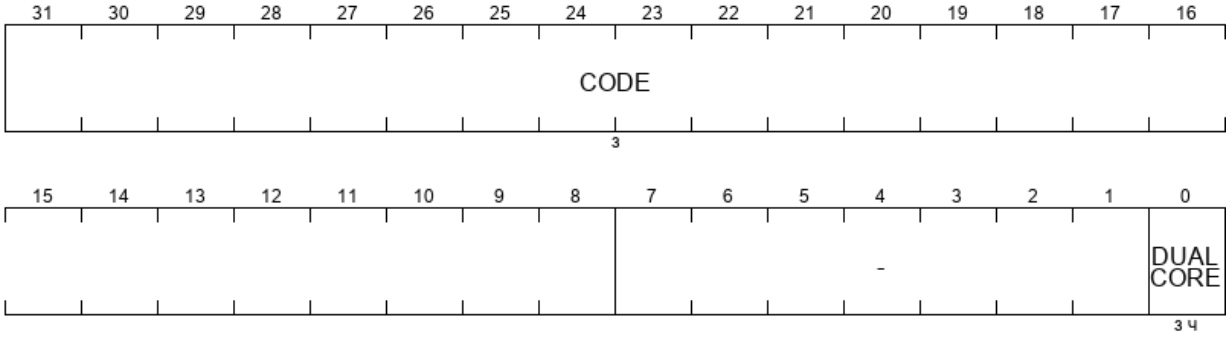
Смещение: + 18h
Сброс: 0h



Поле	Биты	Описание
RVECT	31-0	Вектор сброса для ядра 1 в режиме DUALCORE.

CORE1_CONF – регистр конфигурации ядра 1

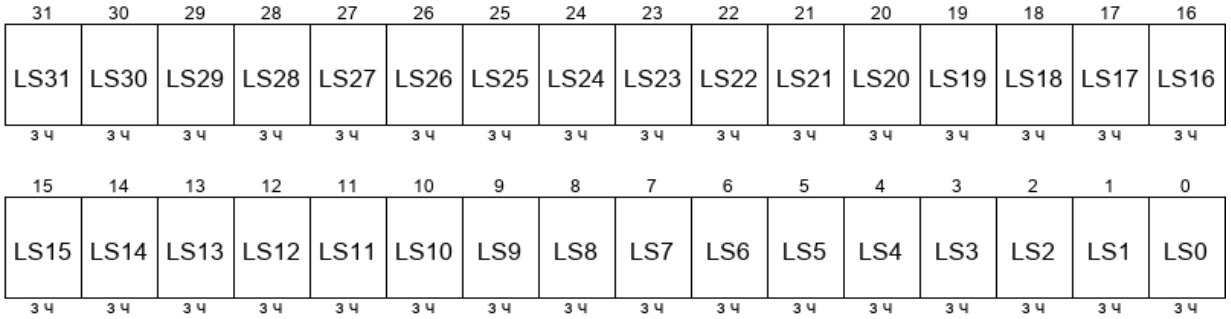
Смещение: + 1Ch
Сброс: 0h



Поле	Биты	Описание	
CODE	31-8	Поле для записи ключа. Значение ключа 0xACCE55	
DUALCORE	0	Бит включения режима DUALCORE для ядра 1. Установка / сброс данного бита осуществляется совместно с записью ключа в поле CODE	
		0	Режим LOCKSTEP
		1	Режим DUALCORE
–	7-1	Зарезервировано	

SSRAMSCR – регистр управления режимом низкого энергопотребления системной памяти SRAM

Смещение: + 20h
Сброс: 0000_0000h



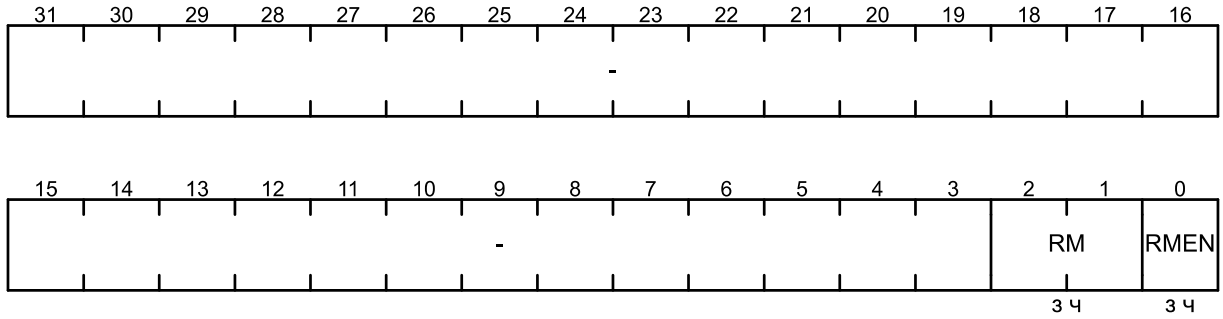
Поле	Биты	Описание
LS31	31	Бит отключения области 31 ОЗУ (992 – 1024 кБайт) для снижения энергопотребления
LS30	30	Бит отключения области 30 ОЗУ (960 – 992 кБайт) для снижения энергопотребления
LS29	29	Бит отключения области 29 ОЗУ (928 – 960 кБайт) для снижения энергопотребления

Поле	Биты	Описание
LS28	28	Бит отключения области 28 ОЗУ (896 – 928 кБайт) для снижения энергопотребления
LS27	27	Бит отключения области 27 ОЗУ (864 – 896 кБайт) для снижения энергопотребления
LS26	26	Бит отключения области 26 ОЗУ (832 – 864 кБайт) для снижения энергопотребления
LS25	25	Бит отключения области 25 ОЗУ (800 – 832 кБайт) для снижения энергопотребления
LS24	24	Бит отключения области 24 ОЗУ (768 – 800 кБайт) для снижения энергопотребления
LS23	23	Бит отключения области 23 ОЗУ (736 – 768 кБайт) для снижения энергопотребления
LS22	22	Бит отключения области 22 ОЗУ (704 – 736 кБайт) для снижения энергопотребления
LS21	21	Бит отключения области 21 ОЗУ (672 – 704 кБайт) для снижения энергопотребления
LS20	20	Бит отключения области 20 ОЗУ (640 – 672 кБайт) для снижения энергопотребления
LS19	19	Бит отключения области19 ОЗУ (608 – 640 кБайт) для снижения энергопотребления
LS18	18	Бит отключения области18 ОЗУ (576 – 608 кБайт) для снижения энергопотребления
LS17	17	Бит отключения области 17 ОЗУ (544 – 576 кБайт) для снижения энергопотребления
LS16	16	Бит отключения области 16 ОЗУ (512 – 544 кБайт) для снижения энергопотребления
LS15	15	Бит отключения области 15 ОЗУ (480 – 512 кБайт) для снижения энергопотребления
LS14	14	Бит отключения области 14 ОЗУ (448 – 480 кБайт) для снижения энергопотребления
LS13	13	Бит отключения области 13 ОЗУ (416 – 448 кБайт) для снижения энергопотребления
LS12	12	Бит отключения области 12 ОЗУ (384 – 416 кБайт) для снижения энергопотребления
LS11	11	Бит отключения области 11 ОЗУ (352 – 384 кБайт) для снижения энергопотребления
LS10	10	Бит отключения области 10 ОЗУ (320 – 352 кБайт) для снижения энергопотребления
LS9	9	Бит отключения области 9 ОЗУ (288 – 320 кБайт) для снижения энергопотребления
LS8	8	Бит отключения области 8 ОЗУ (256 – 288 кБайт) для снижения энергопотребления
LS7	7	Бит отключения области 7 ОЗУ (224 – 256 кБайт) для снижения энергопотребления
LS6	6	Бит отключения области 6 ОЗУ (192 – 224 кБайт) для снижения энергопотребления

Поле	Биты	Описание
LS5	5	Бит отключения области 5 ОЗУ (160 – 192 кБайт) для снижения энергопотребления
LS4	4	Бит отключения области 4 ОЗУ (128 – 160 кБайт) для снижения энергопотребления
LS3	3	Бит отключения области 3 ОЗУ (96 – 128 кБайт) для снижения энергопотребления
LS2	2	Бит отключения области 2 ОЗУ (64 – 96 кБайт) для снижения энергопотребления
LS1	1	Бит отключения области 1 ОЗУ (32 – 64 кБайт) для снижения энергопотребления
LS0	0	Бит отключения области 0 ОЗУ (0 – 32 кБайт) для снижения энергопотребления
–	31-16	Зарезервировано

SSRAMMCR – регистр управления режимами энергопотребления системной памяти SRAM

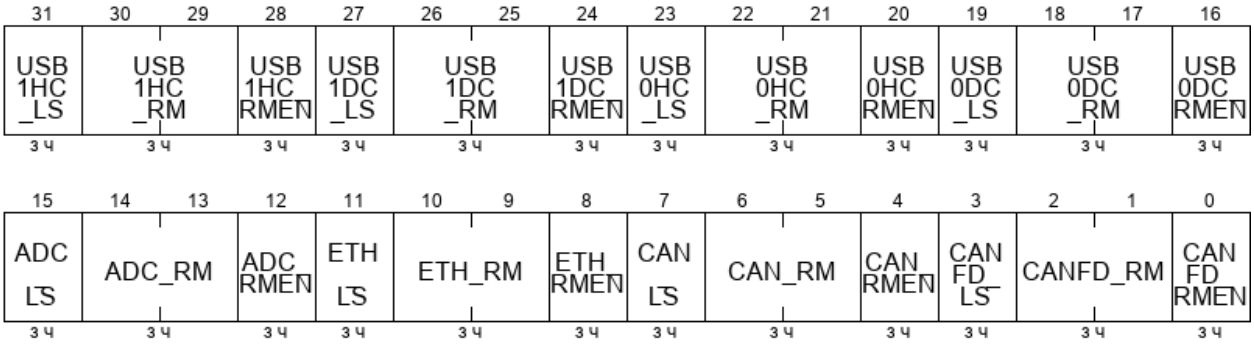
Смещение: + 24h
Сброс: 0000_0000h



Поле	Биты	Описание	
RM	2-1	Поле выбора режима питания	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
RMEN	0	Бит активации управления режимом питания. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
–	31-3	Зарезервировано	

PSRAMSMCR – регистр управления режимами энергопотребления периферийной памяти SRAM

Смещение: + 28h
Сброс: 0000_0000h

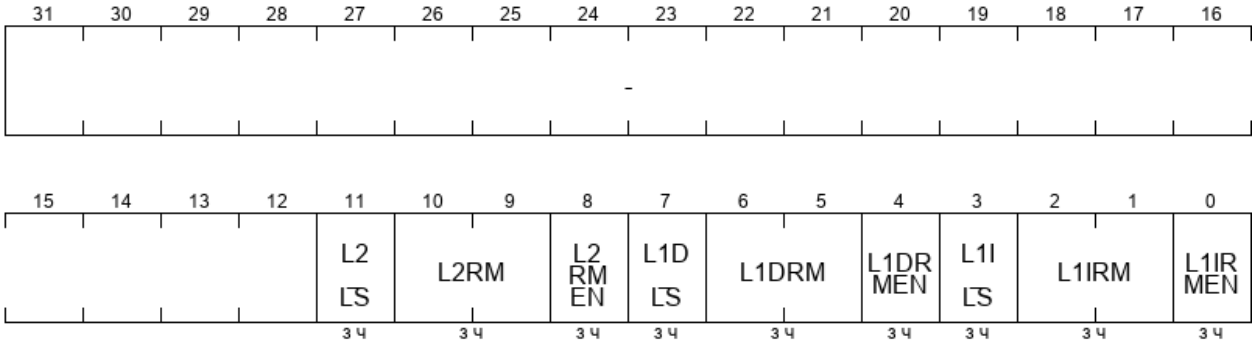


Поле	Биты	Описание	
USB1HC_LS	31	Бит отключения USB1 Host RAM для снижения энергопотребления	
USB1HC_RM	30-29	Поле выбора режима питания для USB1 Host RAM	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
USB1HC_RMEN	28	Бит активации управления режимом питания для USB1 Host RAM. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
USB1DC_LS	27	Бит отключения USB1 Device RAM для снижения энергопотребления	
USB1DC_RM	26-25	Поле выбора режима питания для USB1 Device RAM	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
USB1DC_RMEN	24	Бит активации управления режимом питания для USB1 Device RAM. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
USB0HC_LS	23	Бит отключения USB0 Host RAM для снижения энергопотребления	
USB0HC_RM	22-21	Поле выбора режима питания для USB0 Host RAM	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
USB0HC_RMEN	20	Бит активации управления режимом питания для USB0 Host RAM. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
USB0DC_LS	19	Бит отключения USB0 Device RAM для снижения энергопотребления	

Поле	Биты	Описание	
USB0DC_RM	18-17	Поле выбора режима питания для USB0 Device RAM	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
USB0DC_RMEN	16	Бит активации управления режимом питания для USB0 Device RAM. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
ADC_LS	15	Бит отключения ADC RAM для снижения энергопотребления	
ADC_RM	14-13	Поле выбора режима питания для ADC RAM	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
ADC_RMEN	12	Бит активации управления режимом питания для ADC RAM. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
ETH_LS	11	Бит отключения Ethernet RAM для снижения энергопотребления	
ETH_RM	10-9	Поле выбора режима питания для Ethernet RAM	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
ETH_RMEN	8	Бит активации управления режимом питания для Ethernet RAM. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
CAN_LS	7	Бит отключения CAN RAM для снижения энергопотребления	
CAN_RM	6-5	Поле выбора режима питания для CAN RAM	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
CAN_RMEN	4	Бит активации управления режимом питания для CAN RAM. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
CANFD_LS	3	Бит отключения CANFD RAM для снижения энергопотребления	
CANFD_RM	2-1	Поле выбора режима питания для CANFD RAM	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
CANFD_RMEN	0	Бит активации управления режимом питания для CANFD RAM. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	

CCSMCR – регистр управления режимами энергопотребления кэш памяти ядра

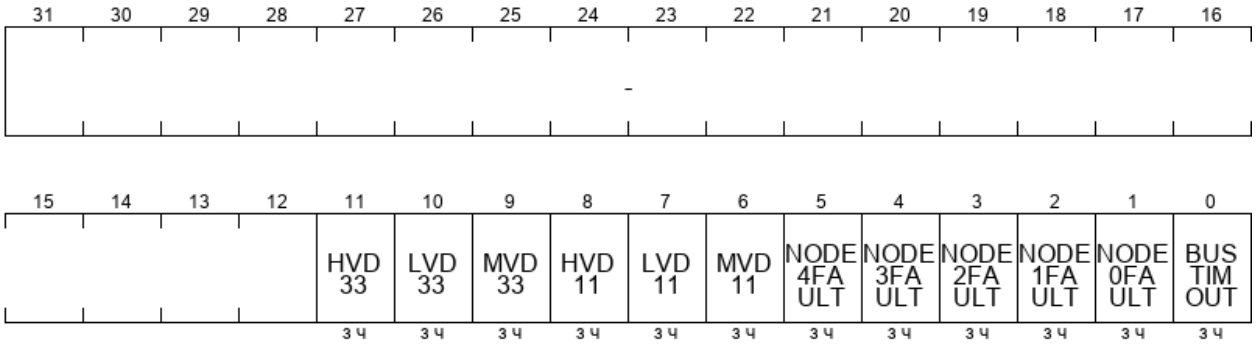
Смещение: + 2Ch
Сброс: 0000_0000h



Поле	Биты	Описание	
L2_LS	11	Бит отключения кэша L2 для снижения энергопотребления	
L2RM	10-9	Поле выбора режима питания кэша L2	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
L2RMEN	8	Бит активации управления режимом питания кэша L2. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
L1D_LS	7	Бит отключения кэша данных L1 для снижения энергопотребления	
L1DRM	6-5	Поле выбора режима питания кэша данных L1	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
L1DRMEN	4	Бит активации управления режимом питания кэша данных L1. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
L1I_LS	3	Бит отключения кэша инструкций L1 для снижения энергопотребления	
L1IRM	2-1	Поле выбора режима питания кэша инструкций L1	
		00b	Пониженное потребление, увеличенные задержки
		…	Промежуточные режимы
		11b	Повышенное потребление, минимальные задержки
L1IRMEN	0	Бит активации управления режимом питания кэша инструкций L1. При сброшенном бите значение поля RM игнорируется, режим питания соответствует RM – 10b	
–	31-12	Зарезервировано	

INTEN – регистр разрешения прерываний

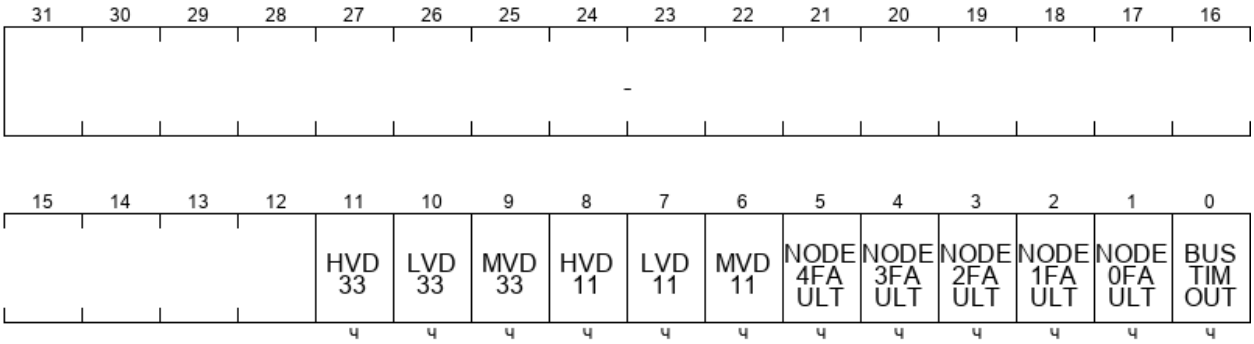
Смещение: + 30h
Сброс: 0000_0000h



Поле	Биты	Описание
HVD33	11	Включение прерывания при превышении напряжения на линии питания 3,3В
LVD33	10	Включение прерывания при понижении напряжения на линии питания 3,3В
MVD33	9	Включение прерывания при определении минимального уровня напряжения на линии питания 3,3В
HVD11	8	Включение прерывания при превышении напряжения на линии внутреннего питания 1,1В
LVD11	7	Включение прерывания при понижении напряжения на линии внутреннего питания 1,1В
MVD11	6	Включение прерывания при определении минимального уровня напряжения на линии внутреннего питания 1,1В
NODE4FAULT	5	Включение прерывания при ошибке 4 узла
NODE3FAULT	4	Включение прерывания при ошибке 3 узла
NODE2FAULT	3	Включение прерывания при ошибке 2 узла
NODE1FAULT	2	Включение прерывания при ошибке 1 узла
NODE0FAULT	1	Включение прерывания при ошибке 0 узла
BUSTIMOUT	0	Включение прерывания по таймауту шины
–	31-12	Зарезервировано

RIS – регистр статуса прерываний

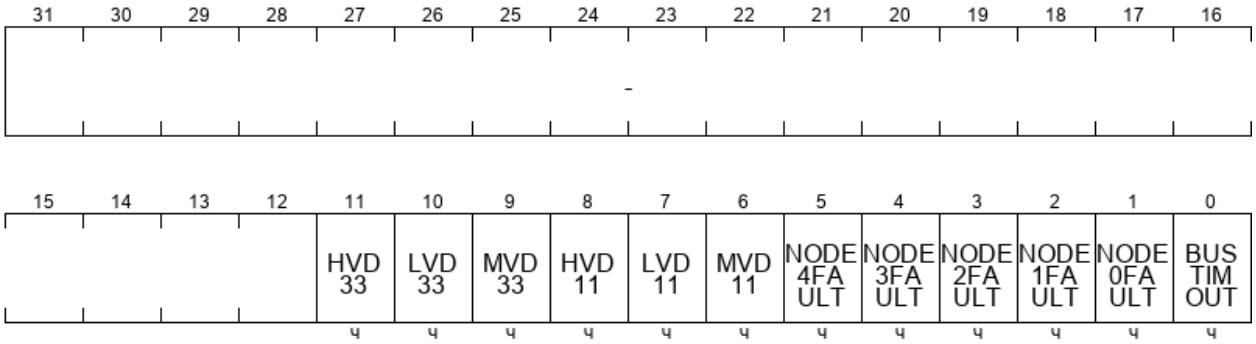
Смещение: + 34h
Сброс: 0000_0000h



Поле	Биты	Описание
HVD33	11	Статус прерывания при превышении напряжения на линии питания 3,3В
LVD33	10	Статус прерывания при понижении напряжения на линии питания 3,3В
MVD33	9	Статус прерывания при определении минимального уровня напряжения на линии питания 3,3В
HVD11	8	Статус прерывания при превышении напряжения на линии внутреннего питания 1,1В
LVD11	7	Статус прерывания при понижении напряжения на линии внутреннего питания 1,1В
MVD11	6	Статус прерывания при определении минимального уровня напряжения на линии внутреннего питания 1,1В
NODE4FAULT	5	Статус прерывания при ошибке 4 узла
NODE3FAULT	4	Статус прерывания при ошибке 3 узла
NODE2FAULT	3	Статус прерывания при ошибке 2 узла
NODE1FAULT	2	Статус прерывания при ошибке 1 узла
NODE0FAULT	1	Статус прерывания при ошибке 0 узла
BUSTIMOUT	0	Статус прерывания по таймауту шины
–	31-12	Зарезервировано

MIS – регистр статуса маскированных прерываний

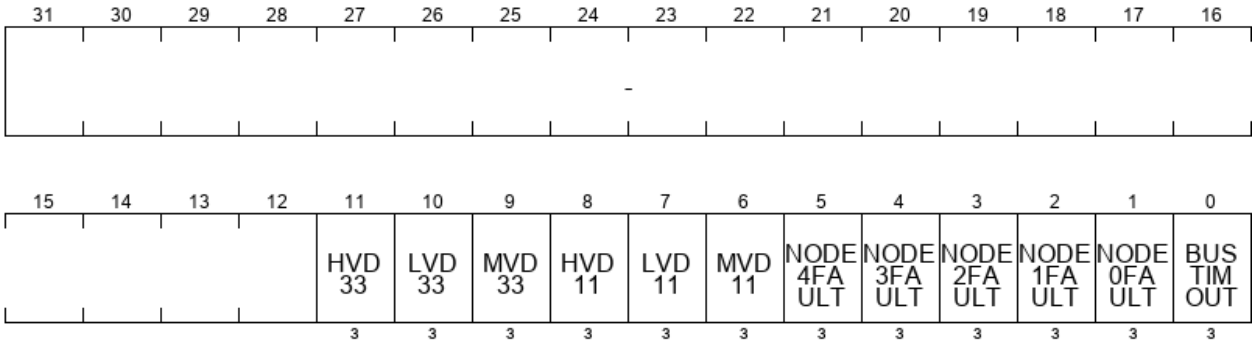
Смещение: + 38h
Сброс: 0000_0000h



Поле	Биты	Описание
HVD33	11	Статус прерывания при превышении напряжения на линии питания 3,3В
LVD33	10	Статус прерывания при понижении напряжения на линии питания 3,3В
MVD33	9	Статус прерывания при определении минимального уровня напряжения на линии питания 3,3В
HVD11	8	Статус прерывания при превышении напряжения на линии внутреннего питания 1,1В
LVD11	7	Статус прерывания при понижении напряжения на линии внутреннего питания 1,1В
MVD11	6	Статус прерывания при определении минимального уровня напряжения на линии внутреннего питания 1,1В
NODE4FAULT	5	Статус прерывания при ошибке 4 узла
NODE3FAULT	4	Статус прерывания при ошибке 3 узла
NODE2FAULT	3	Статус прерывания при ошибке 2 узла
NODE1FAULT	2	Статус прерывания при ошибке 1 узла
NODE0FAULT	1	Статус прерывания при ошибке 0 узла
BUSTIMOUT	0	Статус прерывания по таймауту шины
–	31-12	Зарезервировано

ICLR – регистр сброса прерываний

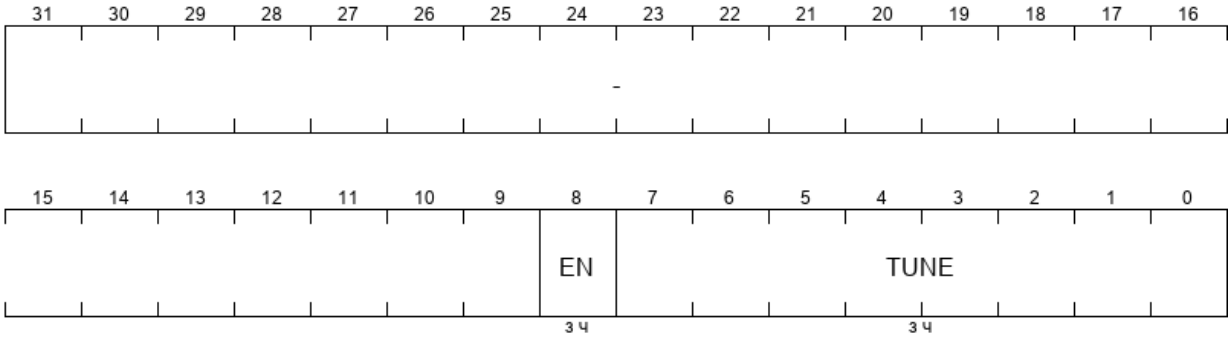
Смещение: + 38h
Сброс: 0000_0000h



Поле	Биты	Описание
HVD33	11	Сброс прерывания при превышении напряжения на линии питания 3,3В
LVD33	10	Сброс прерывания при понижении напряжения на линии питания 3,3В
MVD33	9	Сброс прерывания при определении минимального уровня напряжения на линии питания 3,3В
HVD11	8	Сброс прерывания при превышении напряжения на линии внутреннего питания 1,1В
LVD11	7	Сброс прерывания при понижении напряжения на линии внутреннего питания 1,1В
MVD11	6	Сброс прерывания при определении минимального уровня напряжения на линии внутреннего питания 1,1В
NODE4FAULT	5	Сброс прерывания при ошибке 4 узла
NODE3FAULT	4	Сброс прерывания при ошибке 3 узла
NODE2FAULT	3	Сброс прерывания при ошибке 2 узла
NODE1FAULT	2	Сброс прерывания при ошибке 1 узла
NODE0FAULT	1	Сброс прерывания при ошибке 0 узла
BUSTIMOUT	0	Сброс прерывания по таймауту шины
–	31-12	Зарезервировано

RCCTRL – регистр управления внутренним RC-генератором

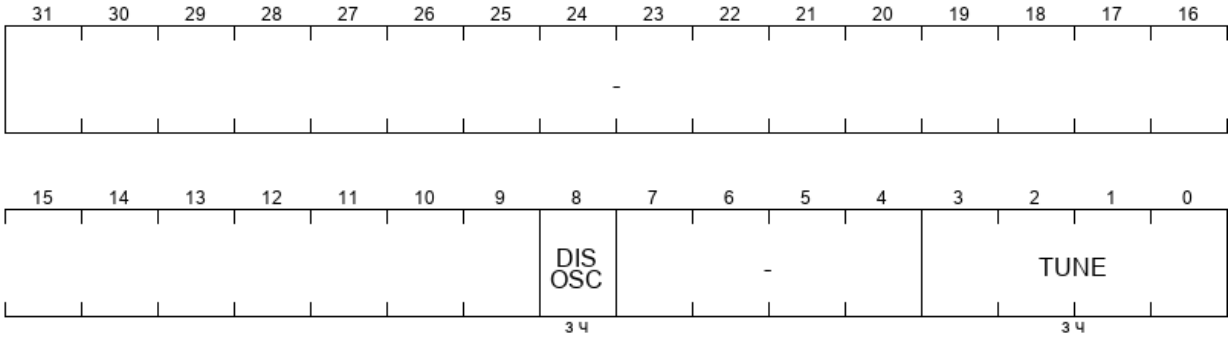
Смещение: + 40h
Сброс: 0000_0142h



Поле	Биты	Описание
EN	8	Бит включения внутреннего RC-генератора
TUNE	7-0	Поле значения подстройки внутреннего RC-генератора (см таблицу 6.1).
–	31-9	Зарезервировано

LDOCTRL – регистр управления внутренним LDO

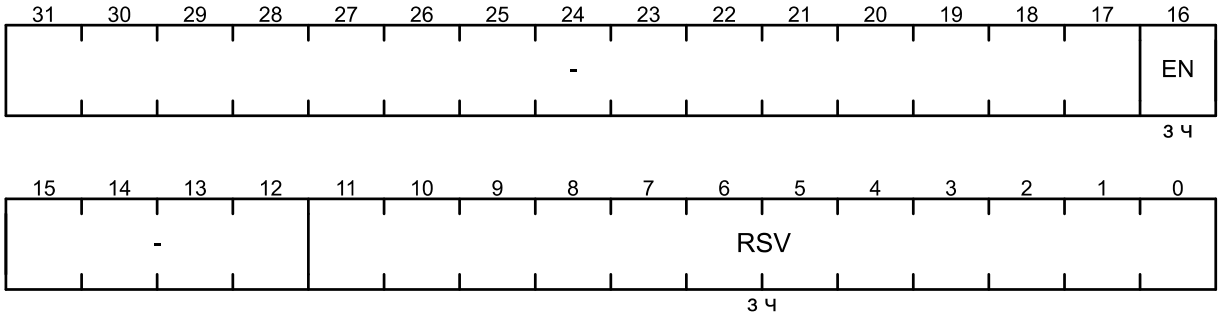
Смещение: + 44h
Сброс: 0000_0000h



Поле	Биты	Описание	
DISOSC	8	Бит выключения внутреннего осциллятора на 100 кГц	
TUNE	3-0	Поле подстройки выходного напряжения LDO	
		Код	Выходное напряжение, В
		0000b	1,103
		0001b	1,118
		0010b	1,134
		0011b	1,151
		0100b	1,168
		0101b	1,185
		0110b	1,203
		0111b	1,222
		1000b	0,992
		1001b	1,005
		1010b	1,018
		1011b	1,030
		1100b	1,044
		1101b	1,058
		1110b	1,073
		1111b	1,087
–	31-9, 7-4	Зарезервировано	

BGPCTRL – регистр управления источником опорного напряжения блока FLASH

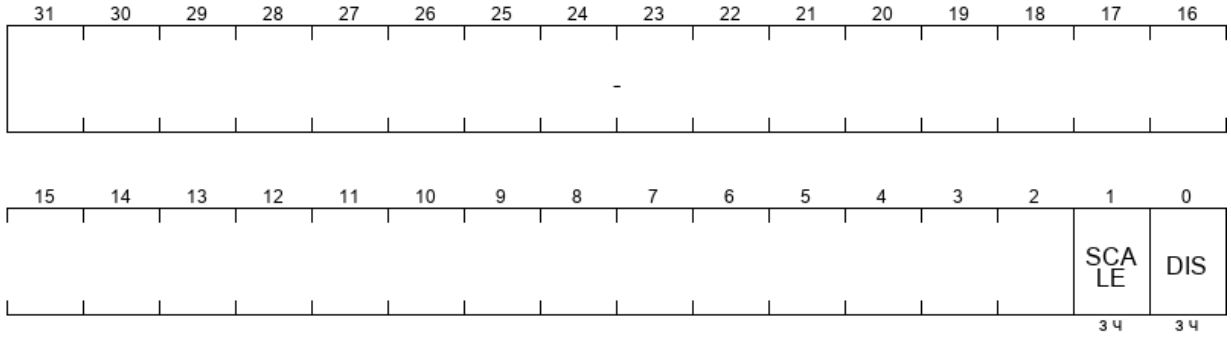
Смещение: + 48h
Сброс: 0001_0800h



Поле	Биты	Описание
EN	16	Бит включения источника опорного напряжения BANDGAP
RSV	11-0	Зарезервированное поле. Всегда должно содержать значение 0x800
–	31-17, 15-0	Зарезервировано

TSENSCTRL – регистр настройки температурного датчика

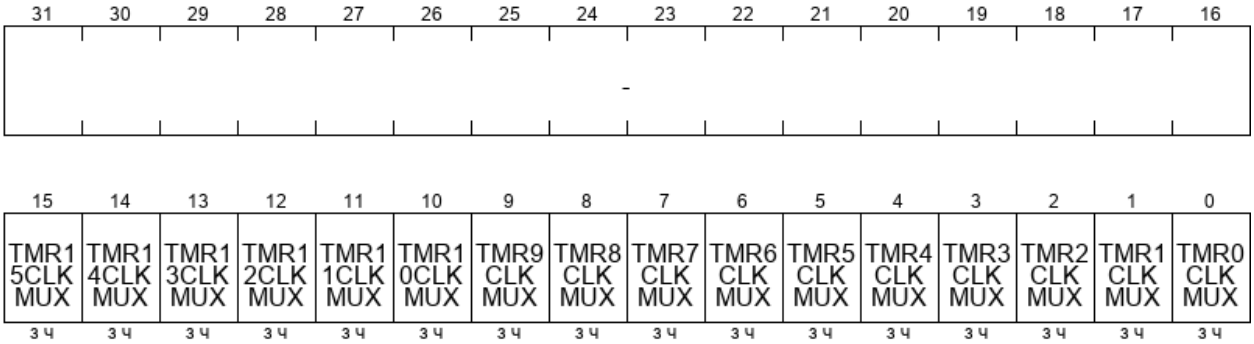
Смещение: + 4Ch
Сброс: 0000_0001h



Поле	Биты	Описание	
SCALE	1	Выбор схемы датчика	
		0	СТАТ2 на двух последовательных PN-переходах
		1	СТАТ1 на одном PN-переходе
DIS	0	Бит выключения датчика температуры. Включение осуществляется записью нуля	
–	31-2	Зарезервировано	

TMRCFG – регистр выбора источника внешней синхронизации блоков TMR

Смещение: + 50h
Сброс: 0000_0000h

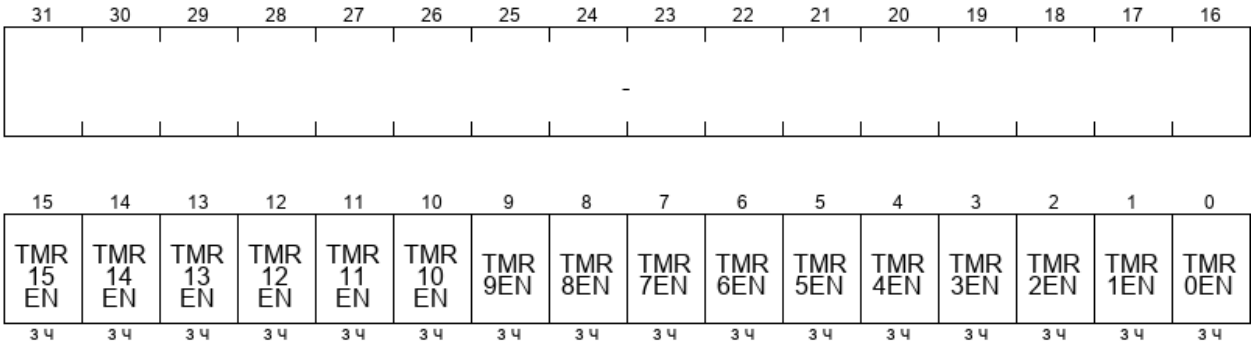


Поле	Биты	Описание	
TMR15CLKMUX	15	Бит выбора источника для TMR15	
		0	Сигнал TMR15_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR14
TMR14CLKMUX	14	Бит выбора источника для TMR14	
		0	Сигнал TMR14_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR13
TMR13LKMUX	13	Бит выбора источника для TMR13	
		0	Сигнал TMR13_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR12
TMR12CLKMUX	12	Бит выбора источника для TMR12	
		0	Сигнал TMR12_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR11
TMR11CLKMUX	11	Бит выбора источника для TMR11	
		0	Сигнал TMR11_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR10
TMR10CLKMUX	10	Бит выбора источника для TMR10	
		0	Сигнал TMR10_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR9
TMR9CLKMUX	9	Бит выбора источника для TMR9	
		0	Сигнал TMR9_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR8
TMR8CLKMUX	8	Бит выбора источника для TMR8	
		0	Сигнал TMR8_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR7
TMR7CLKMUX	7	Бит выбора источника для TMR7	
		0	Сигнал TMR7_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR6
TMR6CLKMUX	6	Бит выбора источника для TMR6	
		0	Сигнал TMR6_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR5

Поле	Биты	Описание	
TMR5CLKMUX	5	Бит выбора источника для TMR5	
		0	Сигнал TMR5_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR4
TMR4CLKMUX	4	Бит выбора источника для TMR4	
		0	Сигнал TMR4_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR3
TMR3CLKMUX	3	Бит выбора источника для TMR3	
		0	Сигнал TMR3_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR2
TMR2CLKMUX	2	Бит выбора источника для TMR2	
		0	Сигнал TMR2_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR1
TMR1CLKMUX	1	Бит выбора источника для TMR1	
		0	Сигнал TMR1_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR0
TMR0CLKMUX	0	Бит выбора источника для TMR0	
		0	Сигнал TMR0_IO с внешней ножки микроконтроллера
		1	Сигнал с выхода предыдущего таймера TMR15
–	31-16	Зарезервировано	

TMREN – регистр разрешения счета таймеров

Смещение: + 54h
Сброс: 0000_0000h

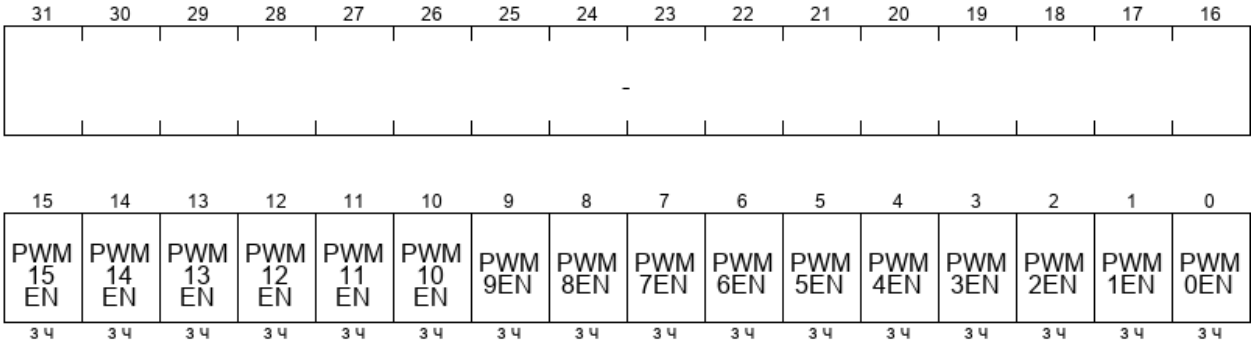


Поле	Биты	Описание
TMR15EN	15	Бит разрешения счета для TMR15
TMR14EN	14	Бит разрешения счета для TMR14
TMR13EN	13	Бит разрешения счета для TMR13
TMR12EN	12	Бит разрешения счета для TMR12
TMR11EN	11	Бит разрешения счета для TMR11
TMR10EN	10	Бит разрешения счета для TMR10
TMR9EN	9	Бит разрешения счета для TMR9
TMR8EN	8	Бит разрешения счета для TMR8
TMR7EN	7	Бит разрешения счета для TMR7
TMR6EN	6	Бит разрешения счета для TMR6

Поле	Биты	Описание
TMR5EN	5	Бит разрешения счета для TMR5
TMR4EN	4	Бит разрешения счета для TMR4
TMR3EN	3	Бит разрешения счета для TMR3
TMR2EN	2	Бит разрешения счета для TMR2
TMR1EN	1	Бит разрешения счета для TMR1
TMR0EN	0	Бит разрешения счета для TMR0
–	31-16	Зарезервировано

PWMEN – регистр разрешения счета блоков ШИМ

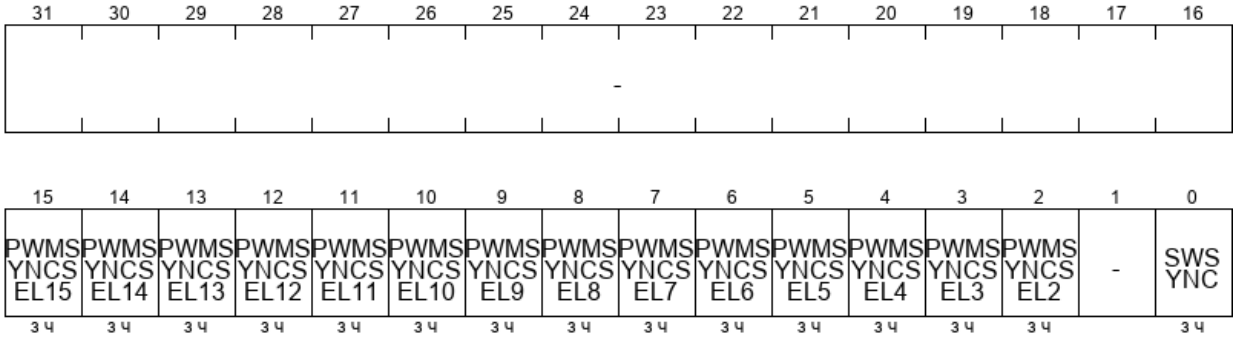
Смещение: + 58h
Сброс: 0000_0000h



Поле	Биты	Описание
PWM15EN	15	Бит разрешения счета для PWM15
PWM14EN	14	Бит разрешения счета для PWM14
PWM13EN	13	Бит разрешения счета для PWM13
PWM12EN	12	Бит разрешения счета для PWM12
PWM11EN	11	Бит разрешения счета для PWM11
PWM10EN	10	Бит разрешения счета для PWM10
PWM9EN	9	Бит разрешения счета для PWM9
PWM8EN	8	Бит разрешения счета для PWM8
PWM7EN	7	Бит разрешения счета для PWM7
PWM6EN	6	Бит разрешения счета для PWM6
PWM5EN	5	Бит разрешения счета для PWM5
PWM4EN	4	Бит разрешения счета для PWM4
PWM3EN	3	Бит разрешения счета для PWM3
PWM2EN	2	Бит разрешения счета для PWM2
PWM1EN	1	Бит разрешения счета для PWM1
PWM0EN	0	Бит разрешения счета для PWM0
–	31-16	Зарезервировано

PWMSYNC – регистр настройки синхронизации блоков PWM

Смещение: + 5Ch
Сброс: 0h



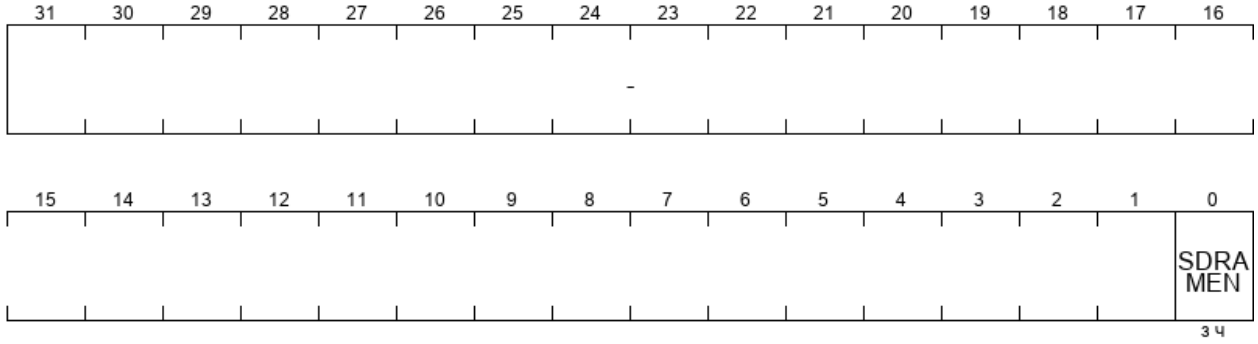
Поле	Биты	Описание	
PWMSYNCSEL15	15	Бит управления схемой синхронизации ШИМ15	
		0	Выход синхронизации SYNCO блока PWM14
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL14	14	Бит управления схемой синхронизации ШИМ14	
		0	Выход синхронизации SYNCO блока PWM13
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL13	13	Бит управления схемой синхронизации ШИМ13	
		0	Выход синхронизации SYNCO блока PWM12
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL12	12	Бит управления схемой синхронизации ШИМ12	
		0	Выход синхронизации SYNCO блока PWM11
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL11	11	Бит управления схемой синхронизации ШИМ11	
		0	Выход синхронизации SYNCO блока PWM10
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL10	10	Бит управления схемой синхронизации ШИМ10	
		0	Выход синхронизации SYNCO блока PWM9
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL9	9	Бит управления схемой синхронизации ШИМ9	
		0	Выход синхронизации SYNCO блока PWM8
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL8	8	Бит управления схемой синхронизации ШИМ8	
		0	Выход синхронизации SYNCO блока PWM7
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL7	7	Бит управления схемой синхронизации ШИМ7	
		0	Выход синхронизации SYNCO блока PWM6
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL6	6	Бит управления схемой синхронизации ШИМ6	
		0	Выход синхронизации SYNCO блока PWM5
		1	Выход синхронизации SYNCO блока PWM0

Поле	Биты	Описание	
PWMSYNCSEL5	5	Бит управления схемой синхронизации ШИМ5	
		0	Выход синхронизации SYNCO блока PWM4
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL4	4	Бит управления схемой синхронизации ШИМ4	
		0	Выход синхронизации SYNCO блока PWM3
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL3	3	Бит управления схемой синхронизации ШИМ3	
		0	Выход синхронизации SYNCO блока PWM2
		1	Выход синхронизации SYNCO блока PWM0
PWMSYNCSEL2	2	Бит управления схемой синхронизации ШИМ2	
		0	Выход синхронизации SYNCO блока PWM1
		1	Выход синхронизации SYNCO блока PWM0
SWSYNC	0	Бит программной эмуляции импульса синхронизации	
		0	Нет реакции
		1	Генерируется импульс синхронизации на входе SYNCI блока ШИМ0 (имитация внешней синхронизации)
–	31-16, 1	Зарезервировано	

Примечание – Для блока PWM1 источником сигнала синхронизации может быть только выход синхронизации SYNCO блока PWM0.

EMMUXCTRL – регистр переключателя внешней памяти SDRAM/EMC

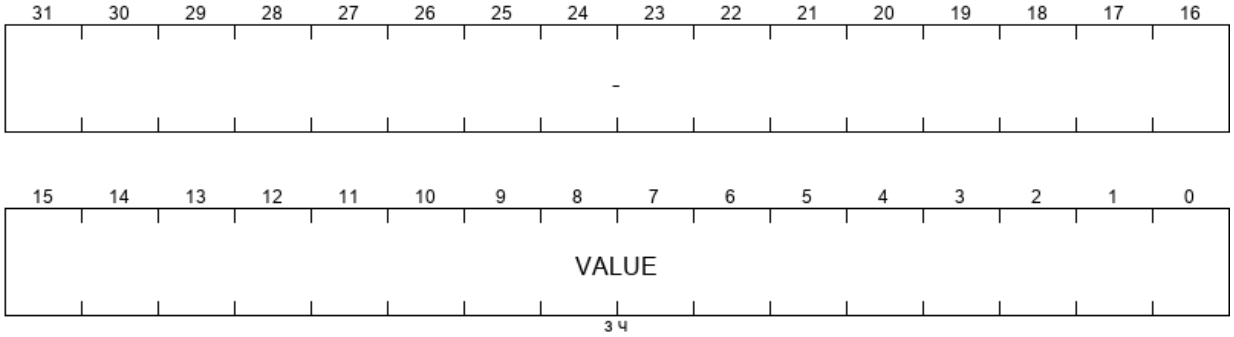
Смещение: + 60h
Сброс: 0000_0000h



Поле	Биты	Описание	
SDRAMEN	0	Бит переключения между использованием типа контроллера внешней памяти	
		0	NOR/NAND Flash (EMC)
		1	SDRAM
–	31-3	Зарезервировано	

TIMEOUT – регистр значения таймаута для сброса

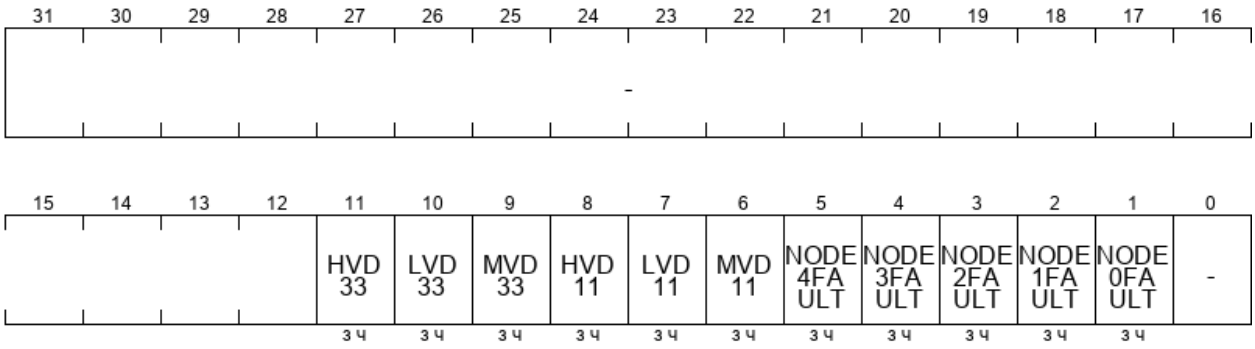
Смещение: + 64h
Сброс: 0000_FFFFh



Поле	Биты	Описание
VALUE	15-0	Поле содержит значение времени таймаута для сброса после возникновения неисправности
–	31-16	Зарезервировано

FRSTEN – регистр разрешения сброса при возникновении ошибок

Смещение: + 68h
Сброс: 0000_0000h

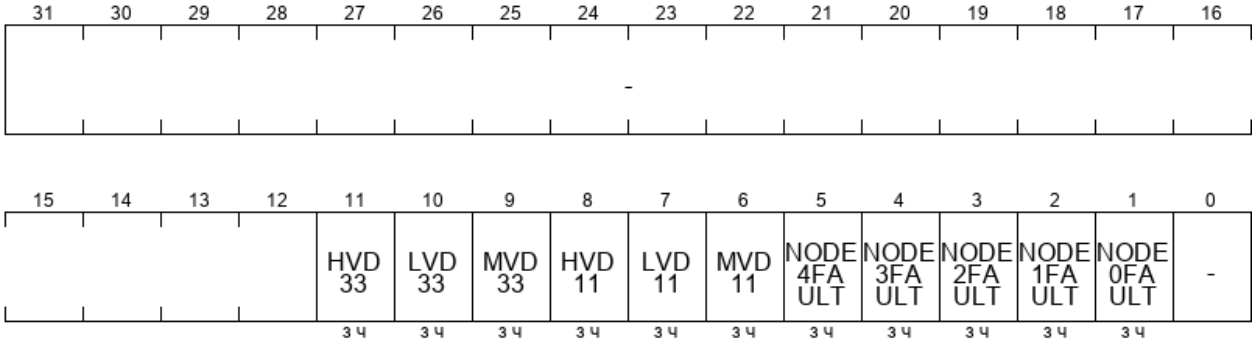


Поле	Биты	Описание
HVD33	11	Разрешение сброса при превышении напряжения на линии питания 3,3В
LVD33	10	Разрешение сброса при понижении напряжения на линии питания 3,3В
MVD33	9	Разрешение сброса при определении минимального уровня напряжения на линии питания 3,3В
HVD11	8	Разрешение сброса при превышении напряжения на линии внутреннего питания 1,1В
LVD11	7	Разрешение сброса при понижении напряжения на линии внутреннего питания 1,1В
MVD11	6	Разрешение сброса при определении минимального уровня напряжения на линии внутреннего питания 1,1В

Поле	Биты	Описание
NODE4FAULT	5	Разрешение сброса при ошибке 4 узла
NODE3FAULT	4	Разрешение сброса при ошибке 3 узла
NODE2FAULT	3	Разрешение сброса при ошибке 2 узла
NODE1FAULT	2	Разрешение сброса при ошибке 1 узла
NODE0FAULT	1	Разрешение сброса при ошибке 0 узла
–	31-12, 0	Зарезервировано

FSTAT – регистр статуса сброса при возникновении ошибок

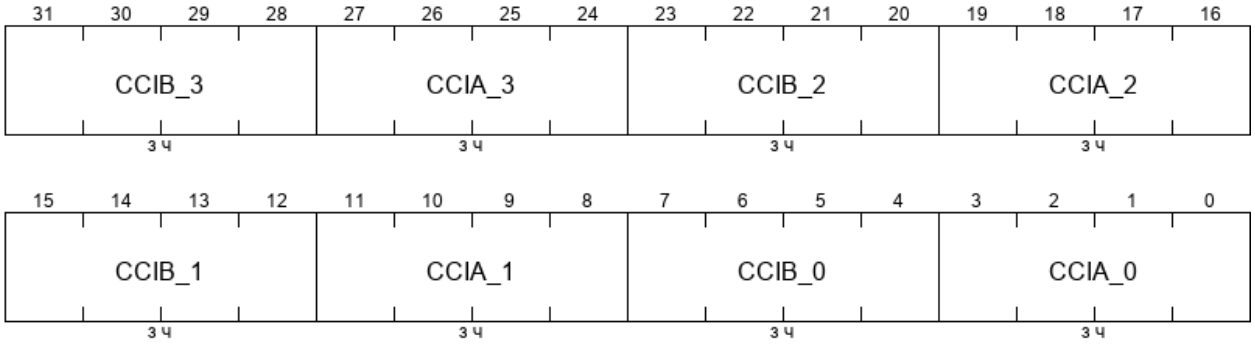
Смещение: + 6Ch
Сброс: 0000_0000h



Поле	Биты	Описание
HVD33	11	Произошел сброс при превышении напряжения на линии питания 3,3В
LVD33	10	Произошел сброс при понижении напряжения на линии питания 3,3В
MVD33	9	Произошел сброс при определении минимального уровня напряжения на линии питания 3,3В
HVD11	8	Произошел сброс при превышении напряжения на линии внутреннего питания 1,1В
LVD11	7	Произошел сброс при понижении напряжения на линии внутреннего питания 1,1В
MVD11	6	Произошел сброс при определении минимального уровня напряжения на линии внутреннего питания 1,1В
NODE4FAULT	5	Произошел сброс при ошибке 4 узла
NODE3FAULT	4	Произошел сброс при ошибке 3 узла
NODE2FAULT	3	Произошел сброс при ошибке 2 узла
NODE1FAULT	2	Произошел сброс при ошибке 1 узла
NODE0FAULT	1	Произошел сброс при ошибке 0 узла
–	31-12, 0	Зарезервировано

TMRMUX – массив регистров выбора сигналов, используемых в качестве событий захвата для таймеров

Смещение: TMRMUX + (4*d)h, где d = 0, …, 15
Сброс: 0000_0000h



Поле	Биты	Описание	
CCIB_3	31-28	Поле выбора источника сигнала для захвата по входу CCIB CAPCOM3	
		0x0	Внешний вывод TMR(n)_IO, где n – номер таймера, для которого производится настройка
		0x1	Внешний вывод TMR(n+1)_IO, где n – номер таймера, для которого производится настройка
		0x2	Внешний вывод TMR(n+2)_IO, где n – номер таймера, для которого производится настройка
		0x3	Внешний вывод TMR(n+3)_IO, где n – номер таймера, для которого производится настройка
		0x4	Импульс запуска АЦП от портов GPIOA или GPIOB
		0x5	Импульс запуска АЦП от портов GPIOC или GPIOD
		0x6	Импульс запуска АЦП от портов GPIOE или GPIOF
		0x7	Импульс запуска АЦП от порта GPIOG
		0x8	Импульс запуска АЦП от канала A блоков PWM0-PWM3
		0x9	Импульс запуска АЦП от канала A блоков PWM4-PWM7
		0xA	Импульс запуска АЦП от канала A блоков PWM8-PWM11
		0xB	Импульс запуска АЦП от канала A блоков PWM12-PWM15
		0xC	Импульс запуска АЦП от канала B блоков PWM0-PWM3
		0xD	Импульс запуска АЦП от канала B блоков PWM4-PWM7
		0xE	Импульс запуска АЦП от канала B блоков PWM8-PWM11
		0xF	Импульс запуска АЦП от канала B блоков PWM12-PWM15

Поле	Биты	Описание	
CCIA_3	27-24	Поле выбора источника сигнала для захвата по входу CCIA CAPCOM3	
		0x0	Внешний вывод TMR(n)_IO, где n – номер таймера, для которого производится настройка
		0x1	Внешний вывод TMR(n+1)_IO, где n – номер таймера, для которого производится настройка
		0x2	Внешний вывод TMR(n+2)_IO, где n – номер таймера, для которого производится настройка
		0x3	Внешний вывод TMR(n+3)_IO, где n – номер таймера, для которого производится настройка
		0x4	Импульс запуска АЦП от портов GPIOA или GPIOB
		0x5	Импульс запуска АЦП от портов GPIOC или GPIOD
		0x6	Импульс запуска АЦП от портов GPIOE или GPIOF
		0x7	Импульс запуска АЦП от порта GPIOG
		0x8	Импульс запуска АЦП от канала A блоков PWM0-PWM3
		0x9	Импульс запуска АЦП от канала A блоков PWM4-PWM7
		0xA	Импульс запуска АЦП от канала A блоков PWM8-PWM11
		0xB	Импульс запуска АЦП от канала A блоков PWM12-PWM15
		0xC	Импульс запуска АЦП от канала B блоков PWM0-PWM3
		0xD	Импульс запуска АЦП от канала B блоков PWM4-PWM7
		0xE	Импульс запуска АЦП от канала B блоков PWM8-PWM11
		0xF	Импульс запуска АЦП от канала B блоков PWM12-PWM15
CCIB_2	23-20	Поле выбора источника сигнала для захвата по входу CCIB CAPCOM2	
		0x0	Внешний вывод TMR(n)_IO, где n – номер таймера, для которого производится настройка
		0x1	Внешний вывод TMR(n+1)_IO, где n – номер таймера, для которого производится настройка
		0x2	Внешний вывод TMR(n+2)_IO, где n – номер таймера, для которого производится настройка
		0x3	Внешний вывод TMR(n+3)_IO, где n – номер таймера, для которого производится настройка
		0x4	Импульс запуска АЦП от портов GPIOA или GPIOB
		0x5	Импульс запуска АЦП от портов GPIOC или GPIOD
		0x6	Импульс запуска АЦП от портов GPIOE или GPIOF
		0x7	Импульс запуска АЦП от порта GPIOG
		0x8	Импульс запуска АЦП от канала A блоков PWM0-PWM3
		0x9	Импульс запуска АЦП от канала A блоков PWM4-PWM7
		0xA	Импульс запуска АЦП от канала A блоков PWM8-PWM11
		0xB	Импульс запуска АЦП от канала A блоков PWM12-PWM15
		0xC	Импульс запуска АЦП от канала B блоков PWM0-PWM3
		0xD	Импульс запуска АЦП от канала B блоков PWM4-PWM7
		0xE	Импульс запуска АЦП от канала B блоков PWM8-PWM11
		0xF	Импульс запуска АЦП от канала B блоков PWM12-PWM15

Поле	Биты	Описание	
CCIA_2	19-16	Поле выбора источника сигнала для захвата по входу CCIA CAPCOM2	
		0x0	Внешний вывод TMR(n)_IO, где n – номер таймера, для которого производится настройка
		0x1	Внешний вывод TMR(n+1)_IO, где n – номер таймера, для которого производится настройка
		0x2	Внешний вывод TMR(n+2)_IO, где n – номер таймера, для которого производится настройка
		0x3	Внешний вывод TMR(n+3)_IO, где n – номер таймера, для которого производится настройка
		0x4	Импульс запуска АЦП от портов GPIOA или GPIOB
		0x5	Импульс запуска АЦП от портов GPIOC или GPIOD
		0x6	Импульс запуска АЦП от портов GPIOE или GPIOF
		0x7	Импульс запуска АЦП от порта GPIOG
		0x8	Импульс запуска АЦП от канала A блоков PWM0-PWM3
		0x9	Импульс запуска АЦП от канала A блоков PWM4-PWM7
		0xA	Импульс запуска АЦП от канала A блоков PWM8-PWM11
		0xB	Импульс запуска АЦП от канала A блоков PWM12-PWM15
		0xC	Импульс запуска АЦП от канала B блоков PWM0-PWM3
		0xD	Импульс запуска АЦП от канала B блоков PWM4-PWM7
		0xE	Импульс запуска АЦП от канала B блоков PWM8-PWM11
		0xF	Импульс запуска АЦП от канала B блоков PWM12-PWM15
CCIB_1	15-12	Поле выбора источника сигнала для захвата по входу CCIB CAPCOM1	
		0x0	Внешний вывод TMR(n)_IO, где n – номер таймера, для которого производится настройка
		0x1	Внешний вывод TMR(n+1)_IO, где n – номер таймера, для которого производится настройка
		0x2	Внешний вывод TMR(n+2)_IO, где n – номер таймера, для которого производится настройка
		0x3	Внешний вывод TMR(n+3)_IO, где n – номер таймера, для которого производится настройка
		0x4	Импульс запуска АЦП от портов GPIOA или GPIOB
		0x5	Импульс запуска АЦП от портов GPIOC или GPIOD
		0x6	Импульс запуска АЦП от портов GPIOE или GPIOF
		0x7	Импульс запуска АЦП от порта GPIOG
		0x8	Импульс запуска АЦП от канала A блоков PWM0-PWM3
		0x9	Импульс запуска АЦП от канала A блоков PWM4-PWM7
		0xA	Импульс запуска АЦП от канала A блоков PWM8-PWM11
		0xB	Импульс запуска АЦП от канала A блоков PWM12-PWM15
		0xC	Импульс запуска АЦП от канала B блоков PWM0-PWM3
		0xD	Импульс запуска АЦП от канала B блоков PWM4-PWM7
		0xE	Импульс запуска АЦП от канала B блоков PWM8-PWM11
		0xF	Импульс запуска АЦП от канала B блоков PWM12-PWM15

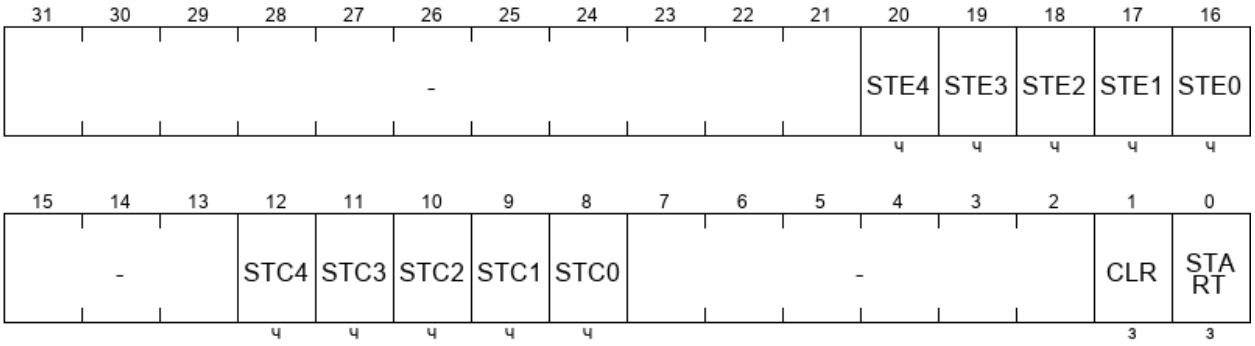
Поле	Биты	Описание	
CCIA_1	11-8	Поле выбора источника сигнала для захвата по входу CCIA CAPCOM1	
		0x0	Внешний вывод TMR(n)_IO, где n – номер таймера, для которого производится настройка
		0x1	Внешний вывод TMR(n+1)_IO, где n – номер таймера, для которого производится настройка
		0x2	Внешний вывод TMR(n+2)_IO, где n – номер таймера, для которого производится настройка
		0x3	Внешний вывод TMR(n+3)_IO, где n – номер таймера, для которого производится настройка
		0x4	Импульс запуска АЦП от портов GPIOA или GPIOB
		0x5	Импульс запуска АЦП от портов GPIOC или GPIOD
		0x6	Импульс запуска АЦП от портов GPIOE или GPIOF
		0x7	Импульс запуска АЦП от порта GPIOG
		0x8	Импульс запуска АЦП от канала A блоков PWM0-PWM3
		0x9	Импульс запуска АЦП от канала A блоков PWM4-PWM7
		0xA	Импульс запуска АЦП от канала A блоков PWM8-PWM11
		0xB	Импульс запуска АЦП от канала A блоков PWM12-PWM15
		0xC	Импульс запуска АЦП от канала B блоков PWM0-PWM3
		0xD	Импульс запуска АЦП от канала B блоков PWM4-PWM7
		0xE	Импульс запуска АЦП от канала B блоков PWM8-PWM11
		0xF	Импульс запуска АЦП от канала B блоков PWM12-PWM15
CCIB_0	7-4	Поле выбора источника сигнала для захвата по входу CCIB CAPCOM0	
		0x0	Внешний вывод TMR(n)_IO, где n – номер таймера, для которого производится настройка
		0x1	Внешний вывод TMR(n+1)_IO, где n – номер таймера, для которого производится настройка
		0x2	Внешний вывод TMR(n+2)_IO, где n – номер таймера, для которого производится настройка
		0x3	Внешний вывод TMR(n+3)_IO, где n – номер таймера, для которого производится настройка
		0x4	Импульс запуска АЦП от портов GPIOA или GPIOB
		0x5	Импульс запуска АЦП от портов GPIOC или GPIOD
		0x6	Импульс запуска АЦП от портов GPIOE или GPIOF
		0x7	Импульс запуска АЦП от порта GPIOG
		0x8	Импульс запуска АЦП от канала A блоков PWM0-PWM3
		0x9	Импульс запуска АЦП от канала A блоков PWM4-PWM7
		0xA	Импульс запуска АЦП от канала A блоков PWM8-PWM11
		0xB	Импульс запуска АЦП от канала A блоков PWM12-PWM15
		0xC	Импульс запуска АЦП от канала B блоков PWM0-PWM3
		0xD	Импульс запуска АЦП от канала B блоков PWM4-PWM7
		0xE	Импульс запуска АЦП от канала B блоков PWM8-PWM11
		0xF	Импульс запуска АЦП от канала B блоков PWM12-PWM15

Поле	Биты	Описание	
CCIA_0	3-0	Поле выбора источника сигнала для захвата по входу CCIA CAPCOM0	
		0x0	Внешний вывод TMR(n)_IO, где n – номер таймера, для которого производится настройка
		0x1	Внешний вывод TMR(n+1)_IO, где n – номер таймера, для которого производится настройка
		0x2	Внешний вывод TMR(n+2)_IO, где n – номер таймера, для которого производится настройка
		0x3	Внешний вывод TMR(n+3)_IO, где n – номер таймера, для которого производится настройка
		0x4	Импульс запуска АЦП от портов GPIOA или GPIOB
		0x5	Импульс запуска АЦП от портов GPIOC или GPIOD
		0x6	Импульс запуска АЦП от портов GPIOE или GPIOF
		0x7	Импульс запуска АЦП от порта GPIOG
		0x8	Импульс запуска АЦП от канала A блоков PWM0-PWM3
		0x9	Импульс запуска АЦП от канала A блоков PWM4-PWM7
		0xA	Импульс запуска АЦП от канала A блоков PWM8-PWM11
		0xB	Импульс запуска АЦП от канала A блоков PWM12-PWM15
		0xC	Импульс запуска АЦП от канала B блоков PWM0-PWM3
		0xD	Импульс запуска АЦП от канала B блоков PWM4-PWM7
		0xE	Импульс запуска АЦП от канала B блоков PWM8-PWM11
		0xF	Импульс запуска АЦП от канала B блоков PWM12-PWM15
–	31-8	Зарезервировано	

Примечание – Для таймеров TMR13, TMR14, TMR15 при выборе источника сигнала для захвата существует переход. Например, для TMR14 источниками сигнала для захвата могут быть внешние выводы TMR14, TMR15, TMR0, TMR1.

NODESTEST – регистр запуска и статуса самотестирования узлов

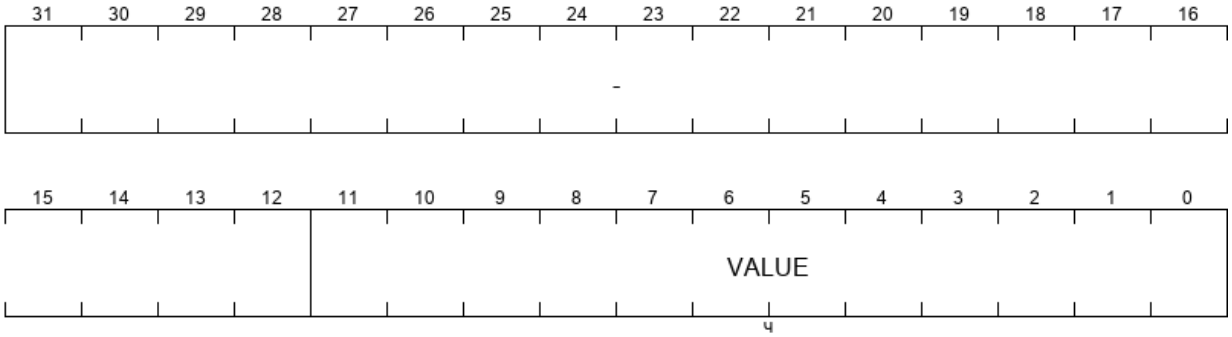
Смещение: + 100h
Сброс: 0000_0000h



Поле	Биты	Описание
STE4	20	Самотестирование узла 4 завершено с ошибкой
STE3	19	Самотестирование узла 3 завершено с ошибкой
STE2	18	Самотестирование узла 2 завершено с ошибкой
STE1	17	Самотестирование узла 1 завершено с ошибкой
STE0	16	Самотестирование узла 0 завершено с ошибкой
STC4	12	Самотестирование узла 4 выполнено
STC3	11	Самотестирование узла 3 выполнено
STC2	10	Самотестирование узла 2 выполнено
STC1	9	Самотестирование узла 1 выполнено
STC0	8	Самотестирование узла 0 выполнено
CLR	1	Сброс статусных битов
START	0	Запуск самотестирования узлов
–	31-21, 15-12, 7-2	Зарезервировано

NODE0ECNT – регистр счетчика ошибок узла 0

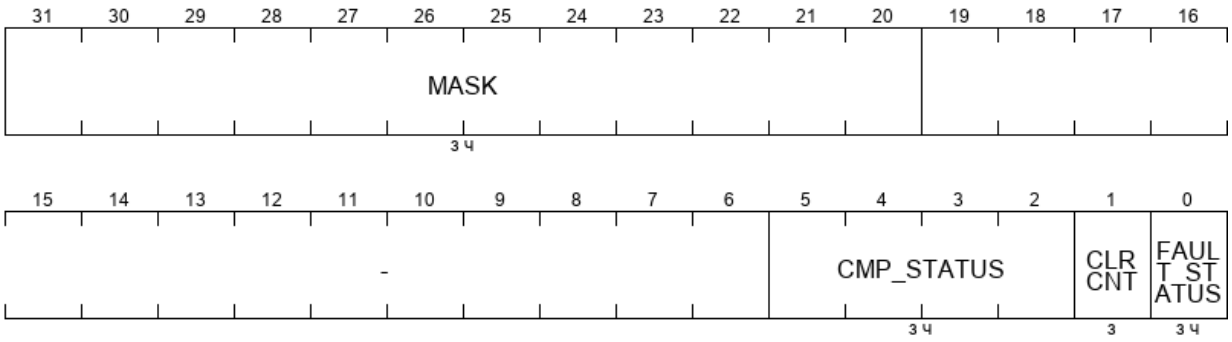
Смещение: + 110h
Сброс: 0000_0000h



Поле	Биты	Описание
VALUE	11-0	Количество ошибок в текущем узле
–	31-12	Зарезервировано

NODE0CTRL – регистр управления узлом 0

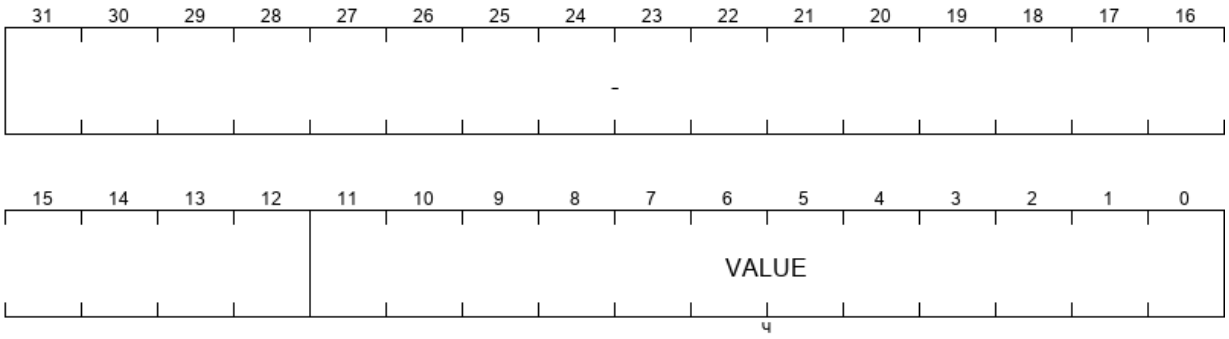
Смещение: + 114h
Сброс: FFF0_0000h



Поле	Биты	Описание
MASK	31-20	Поле выбора максимального количества ошибок узла
CMP_STATUS	5-2	Статус компаратора узла
CLRCNT	1	Бит сброса счетчика ошибок узла
FAULT_STATUS	0	Бит статуса неисправности узла
–	19-6	Зарезервировано

NODE1ECNT – регистр счетчика ошибок узла 1

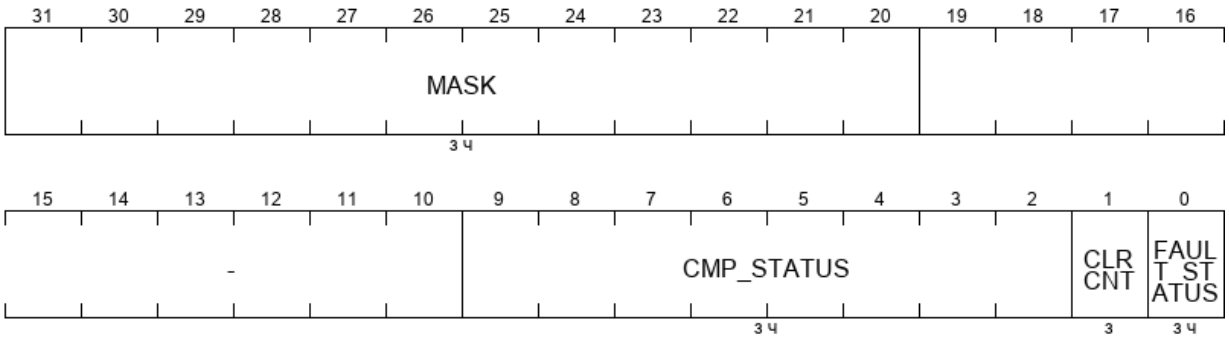
Смещение: + 118h
Сброс: 0000_0000h



Поле	Биты	Описание
VALUE	11-0	Количество ошибок в текущем узле
–	31-12	Зарезервировано

NODE1CTRL – регистр управления узлом 1

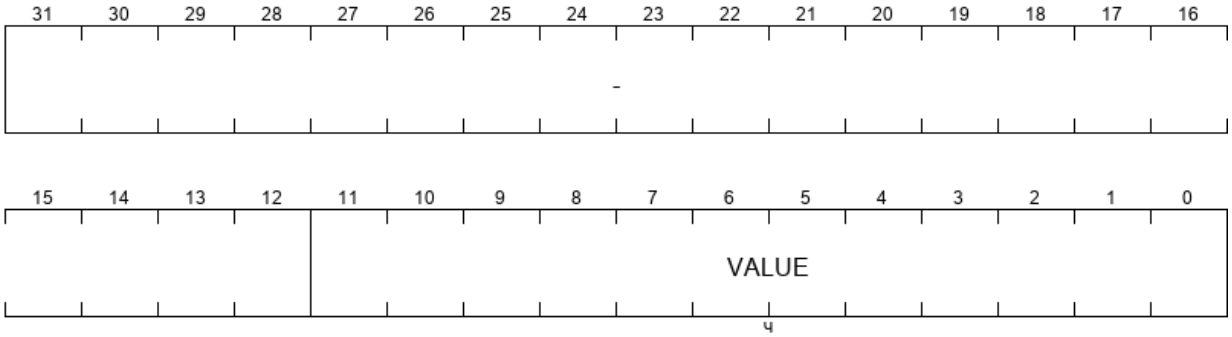
Смещение: + 11Ch
Сброс: FFF0_0000h



Поле	Биты	Описание
MASK	31-20	Поле выбора максимального количества ошибок узла
CMP_STATUS	9-2	Статус компаратора узла
CLRCNT	1	Бит сброса счетчика ошибок узла
FAULT_STATUS	0	Бит статуса неисправности узла
–	19-10	Зарезервировано

NODE2ECNT – регистр счетчика ошибок узла 2

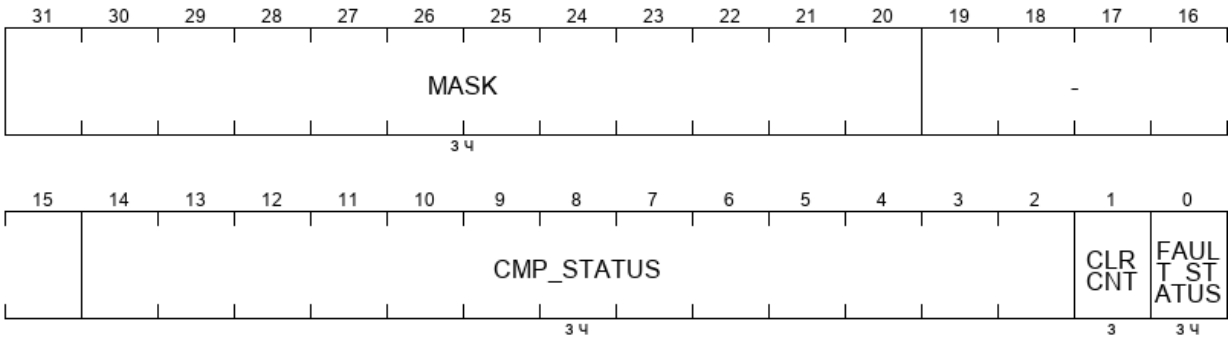
Смещение: + 120h
Сброс: 0000_0000h



Поле	Биты	Описание
VALUE	11-0	Количество ошибок в текущем узле
–	31-12	Зарезервировано

NODE2CTRL – регистр управления узлом 2

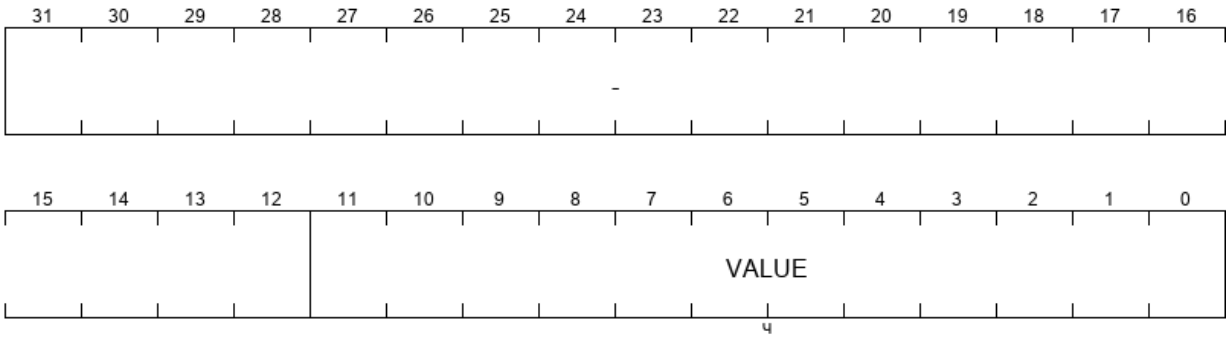
Смещение: + 124h
Сброс: FFF0_0000h



Поле	Биты	Описание
MASK	31-20	Поле выбора максимального количества ошибок узла
CMP_STATUS	14-2	Статус компаратора узла
CLRCNT	1	Бит сброса счетчика ошибок узла
FAULT_STATUS	0	Бит статуса неисправности узла
–	19-15	Зарезервировано

NODE3ECNT – регистр счетчика ошибок узла 3

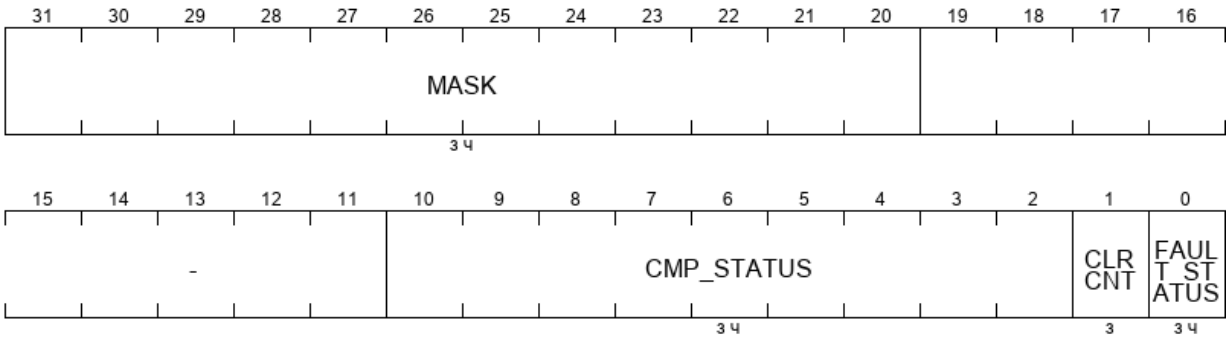
Смещение: + 128h
Сброс: 0000_0000h



Поле	Биты	Описание
VALUE	11-0	Количество ошибок в текущем узле
–	31-12	Зарезервировано

NODE3CTRL – регистр управления узлом 3

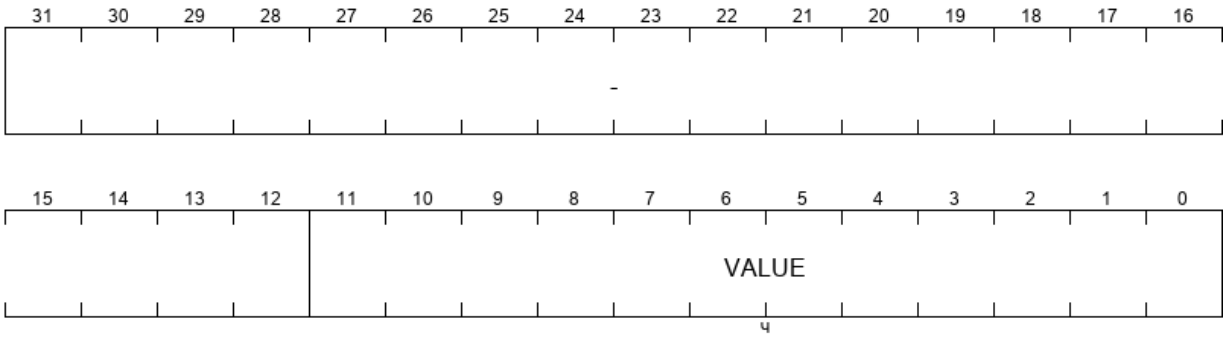
Смещение: + 12Ch
Сброс: FFF0_0000h



Поле	Биты	Описание
MASK	31-20	Поле выбора максимального количества ошибок узла
CMP_STATUS	10-2	Статус компаратора узла
CLRCNT	1	Бит сброса счетчика ошибок узла
FAULT_STATUS	0	Бит статуса неисправности узла
–	19-11	Зарезервировано

NODE4ECNT – регистр счетчика ошибок узла 4

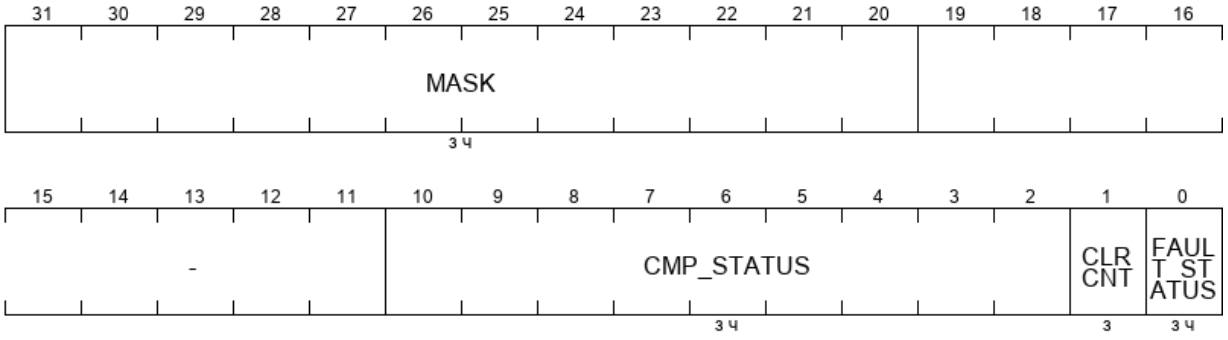
Смещение: + 130h
Сброс: 0000_0000h



Поле	Биты	Описание
VALUE	11-0	Количество ошибок в текущем узле
–	31-12	Зарезервировано

NODE4CTRL – регистр управления узлом 4

Смещение: + 134h
Сброс: FFF0_0000h



Поле	Биты	Описание
MASK	31-20	Поле выбора максимального количества ошибок узла
CMP_STATUS	10-2	Статус компаратора узла
CLRCNT	1	Бит сброса счетчика ошибок узла
FAULT_STATUS	0	Бит статуса неисправности узла
–	19-11	Зарезервировано

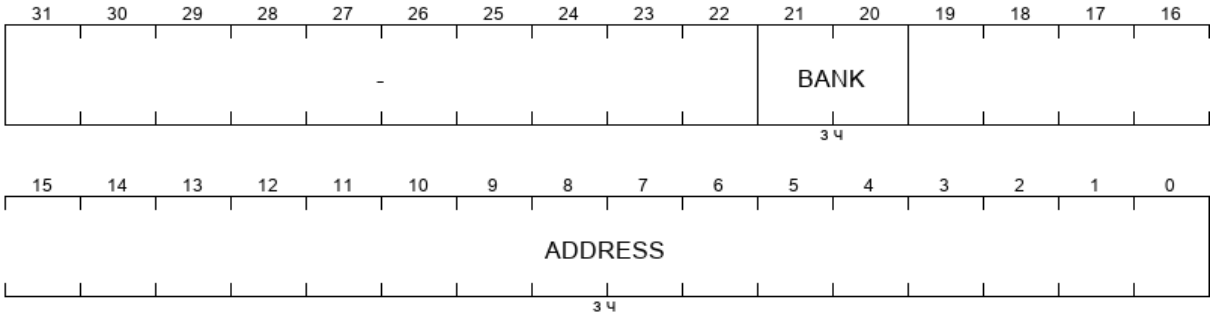
A.3 Регистры контроллера FlashM-памяти

Базовый адрес: 5000_2000h

Смещение: + 10h (DATA) Регистры данных
+ 170h (ERRMASK) Регистры маски данных

ADDR – регистр адреса Flash-памяти

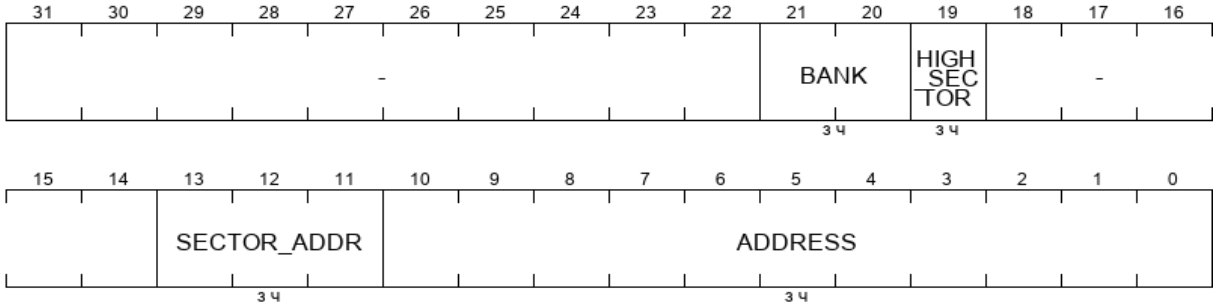
Смещение: + 0h
Сброс: 0h



Поле	Биты	Описание
BANK	21-20	Номер банка памяти. Старшие два бита адреса
ADDRESS	19-0	Адрес, используемый при командах записи, чтения и постраничного стирания. Должен быть выровнен по 16 байт. Не выровненные адреса выравниваются автоматически (биты 3-0 не имеют значения)
–	31-22	Зарезервировано

NVR_ADDR – регистр адреса для доступа к NVR области Flash-памяти

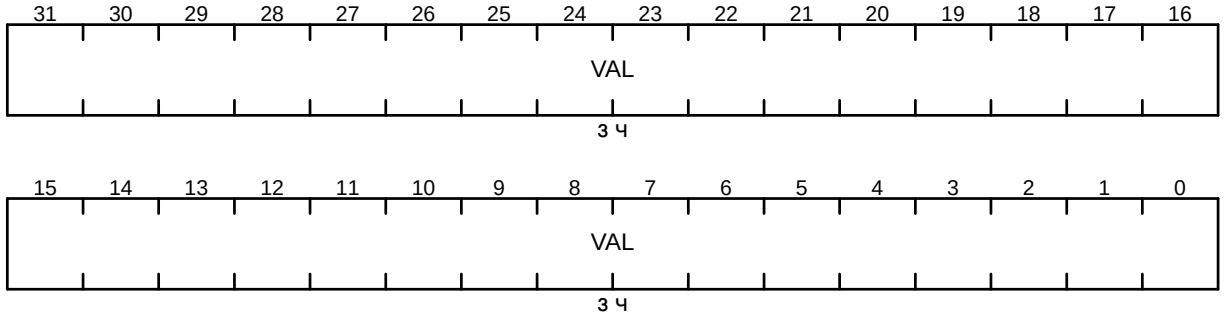
Смещение: + 0h
Сброс: 0h



Поле	Биты	Описание
BANK	21-20	Номер банка памяти. Старшие два бита адреса
HIGH_SECTOR	19	Значение адреса сектора (старшая часть) для выбранного блока Flash-памяти
SECTOR_ADDR	13-11	Значение адреса сектора (младшая часть) для выбранного блока Flash-памяти
ADDRESS	10-0	Адрес, используемый при командах записи, чтения и постраничного стирания. Должен быть выровнен по 16 байт. Не выровненные адреса выравниваются автоматически (биты 3-0 не имеют значения)
–	31-22, 18-14	Зарезервировано. Биты 18-14 не имеют значения

DATA – массив регистров данных Flash-памяти

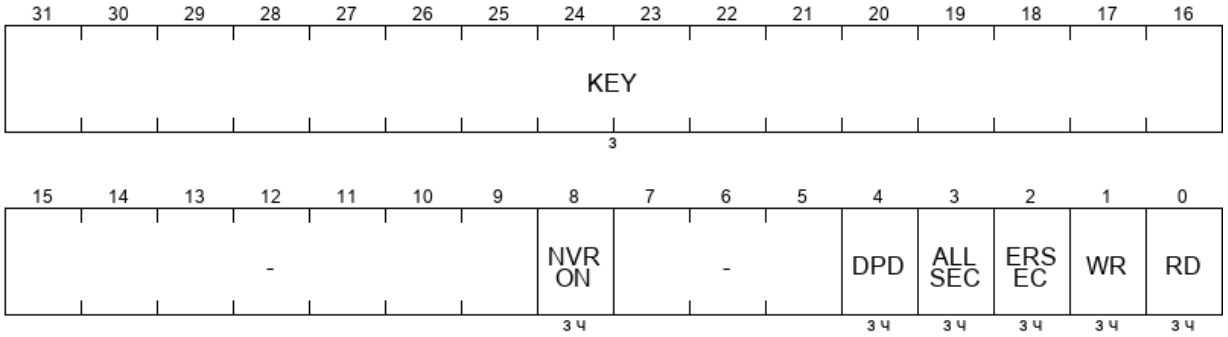
Смещение: DATA + (4*d)h, где d = 0, …, 3
Сброс: FFFF_FFFFh



Поле	Биты	Описание
VAL	31-0	32-разрядные регистры слов данных. Все слова данных должны быть загружены в регистры до установки бита команды записи. Читаемые данные будут доступны в регистрах после сброса флага BUSY

CMD – регистр команд Flash-памяти

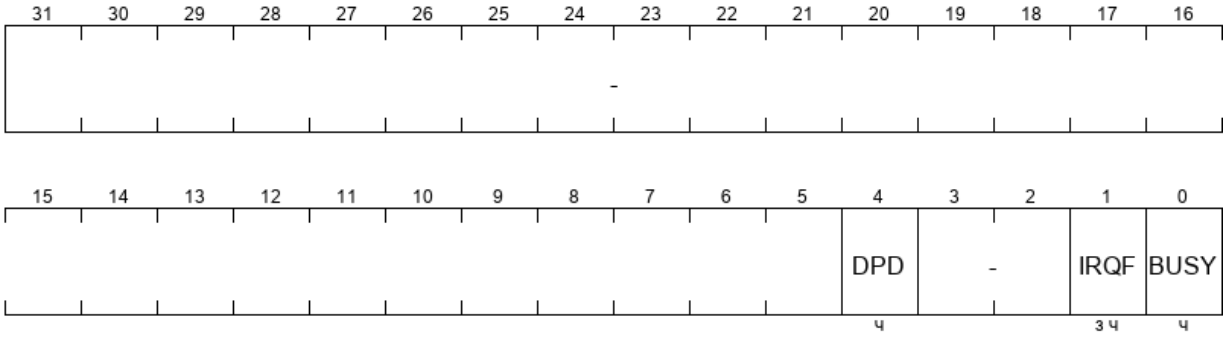
Смещение: + 40h
Сброс: C0DE_0000h



Поле	Биты	Описание
KEY	31-16	Код запуска команды. Все команды для вступления в силу должны сопровождаться записью в поле KEY значения C0DEh. Команды должны выполняться по одной, т. е. запись следующей команды разрешена, только после завершения предыдущей. При одновременной записи нескольких команд будет выполнена та, номер бита которой меньше. Чтение поля KEY всегда возвращает DEC0h
NVRON	8	Бит модификации команды для работы с NVR областью
		0 Команда выполняется для основной области Flash-памяти
		1 Команда выполняется для NVR области Flash-памяти
DPD	4	Бит установки/сброса режима PowerDown
ALLSEC	3	Бит активации команды стирания всех страниц. При установленных битах ALLSEC и ERSEC происходит полное стирание области
ERSEC	2	Бит активации команды стирания страницы области. Адрес страницы вычисляется на основе значения регистра ADDR
WR	1	Бит активации команды записи данных DATAd (d от 0 до 3), начиная с адреса ADDR в области
RD	0	Бит активации команды чтения данных в DATAd (d от 0 до 3), начиная с адреса ADDR в области
–	15-9, 7-4	Зарезервировано

STAT – регистр статуса Flash-памяти

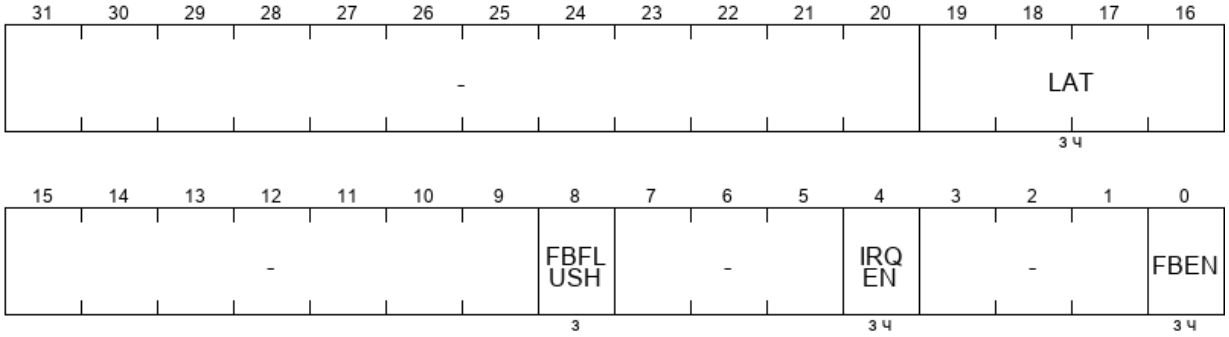
Смещение: + 44h
Сброс: 0h



Поле	Биты	Описание	
DPD	4	Бит статуса режима PowerDown	
IRQF	1	Флаг прерывания по окончании выполнения команды. Устанавливается только если бит IRQEN установлен	
		0	Нет информации
		1	Команда выполнена
		Сбрасывается записью «1»	
BUSY	0	Статус работы контроллера Flash-памяти	
		0	Нет активной команды
		1	Выполняется команда
		Примечание – В связи с особенностями пересинхронизации, при работе на высоких частотах ядра необходимо добавлять задержку между записью регистра CMD и чтением флага BUSY, например, 5 NOP команд	
–	31-5, 3-2	Зарезервировано	

CTRL – регистр настройки контроллера Flash-памяти

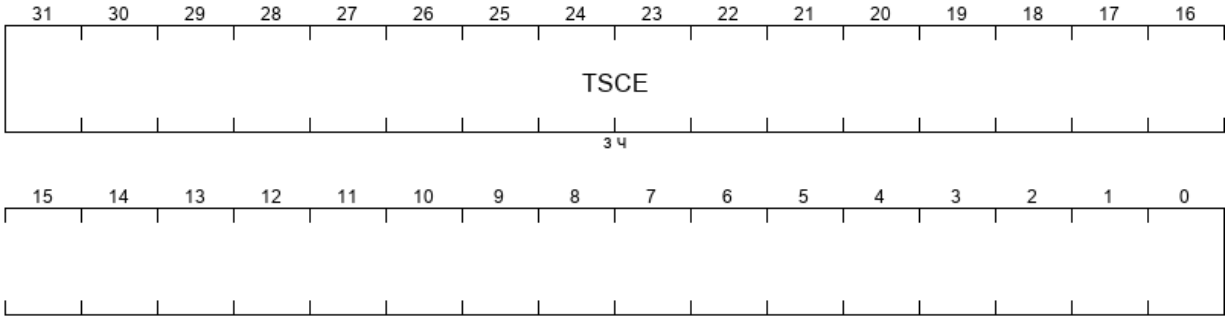
Смещение: + 48h
Сброс: 0001_0000h



Поле	Биты	Описание
LAT	19-16	Поле задания количества дополнительных тактов ожидания при чтении из Flash-памяти
FBFLUSH	8	Бит очистки внутреннего буфера предвыборки
IRQEN	4	Бит разрешения прерываний
FBEN	0	Бит включения буфера предвыборки
–	31-20, 15-9, 7-5, 3-1	Зарезервировано

FTCETR – регистр установки времени стирания Flash

Смещение: + 100h
Сброс: 0000_EA60h

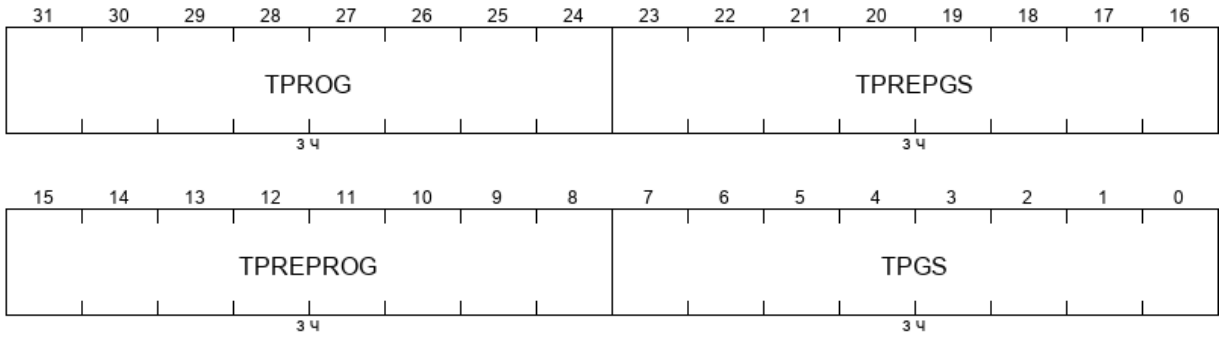


Поле	Биты	Описание
TSCE	31-0	Поле содержит значение времени стирания в циклах clk. Запись заблокирована, когда STAT->BUSY = 1

Примечание – В обычном режиме работы не следует изменять значение по сбросу.

FTCPRT – регистр установки времени программирования Flash

Смещение: + 104h
Сброс: 4820_1030h

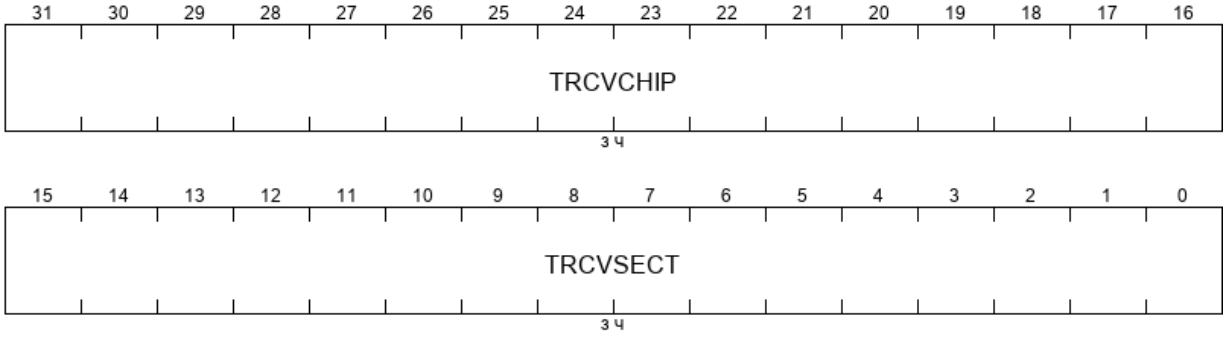


Поле	Биты	Описание
TPROG	31-24	Поле содержит значение времени в циклах clk для соблюдения тайминга PROG. Запись заблокирована, когда STAT->BUSY = 1
TPREPGS	23-16	Поле содержит значение времени в циклах clk для соблюдения тайминга PREPGS. Запись заблокирована, когда STAT->BUSY = 1
TPREPROG	15-8	Поле содержит значение времени в циклах clk для соблюдения тайминга PREPROG. Запись заблокирована, когда STAT->BUSY = 1
TPGS	7-0	Поле содержит значение времени в циклах clk для соблюдения тайминга TPGS. Запись заблокирована, когда STAT->BUSY = 1

П р и м е ч а н и е – В обычном режиме работы не следует изменять значение по сбросу.

FTCERTR – регистр установки времени восстановления после стирания Flash

Смещение: + 108h
Сброс: 0320_00C8h

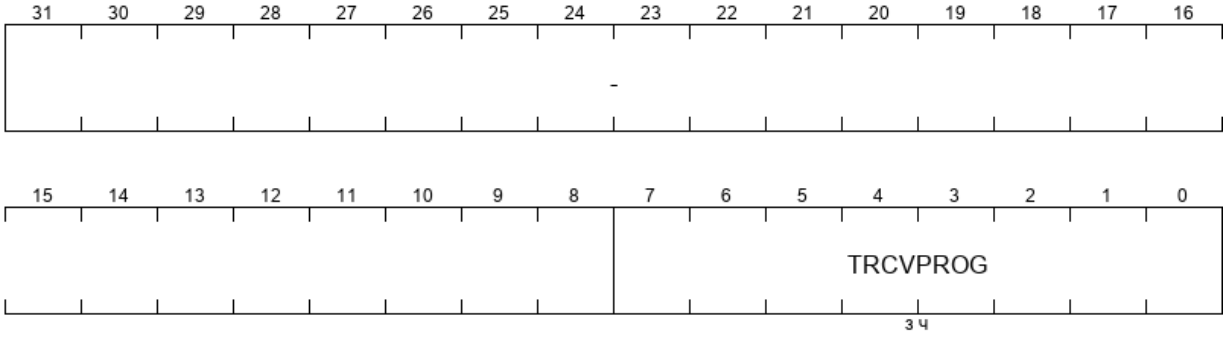


Поле	Биты	Описание
TRCVCHIP	31-16	Поле содержит значение времени восстановления после стирания всей Flash-памяти в циклах clk. Запись заблокирована, когда STAT->BUSY = 1
TRCVSECT	15-0	Поле содержит значение времени восстановления после стирания сектора памяти в циклах clk. Запись заблокирована, когда STAT->BUSY = 1

П р и м е ч а н и е – В обычном режиме работы не следует изменять значение по сбросу.

FTCPRTR – регистр установки времени восстановления после программирования Flash

Смещение: + 10Ch
Сброс: 0000_0014h

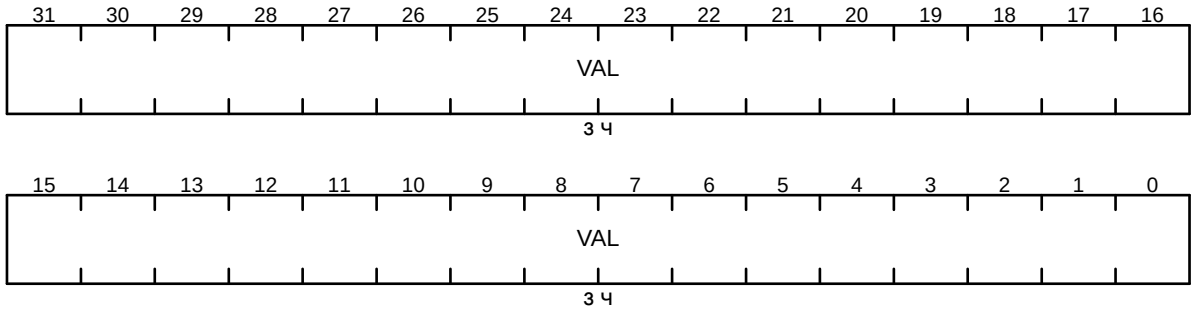


Поле	Биты	Описание
TRCVPROG	7-0	Поле содержит значение времени восстановления после программирования в циклах clk. Запись заблокирована, когда STAT->BUSY = 1
–	31-8	Зарезервировано

П р и м е ч а н и е – В обычном режиме работы не следует изменять значение по сбросу.

ERRMASK – массив регистров маски данных Flash-памяти

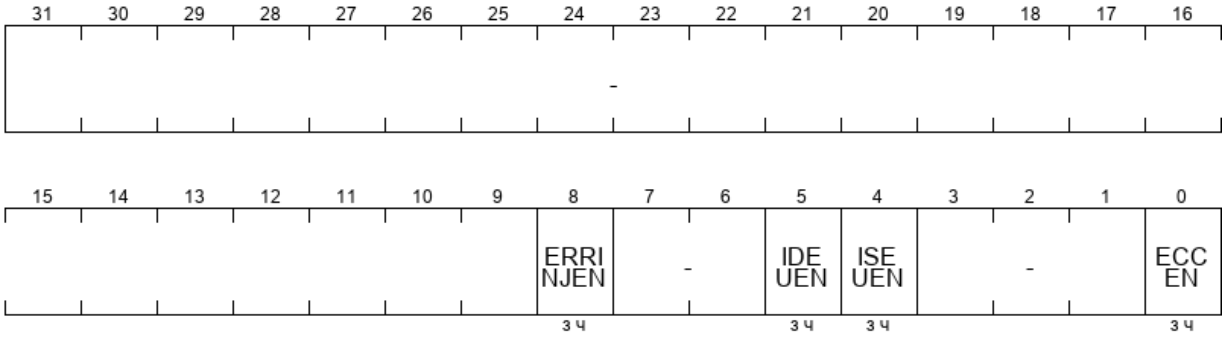
Смещение: ERRMASK + (4*d)h, где d = 0, …, 3
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Поле содержит маску данных. Соответствующий бит записываемого значения инвертируется.

FECCCR – регистр управления кодом коррекции ошибок Flash

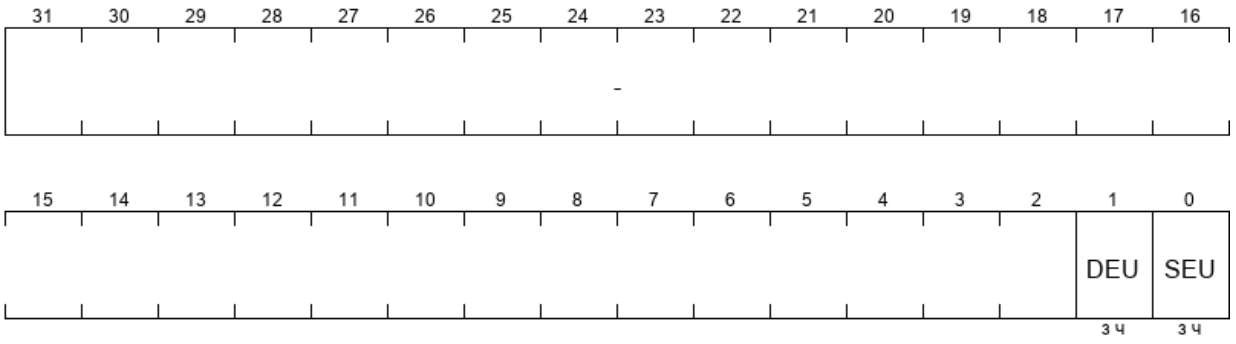
Смещение: + 180h
Сброс: 0000_0000h



Поле	Биты	Описание
ERRINJEN	8	Бит включения внедрения ошибок на шину записи данных
IDEUEN	5	Бит разрешения прерывания при обнаружении двойных ошибок
ISEUEN	4	Бит разрешения прерывания при обнаружении одиночных ошибок
ECCEN	0	Бит разрешения обнаружения и коррекции ошибок ECC
–	31-9, 7-6, 3-1	Зарезервировано

FECCISTAT – регистр статуса прерываний ECC Flash

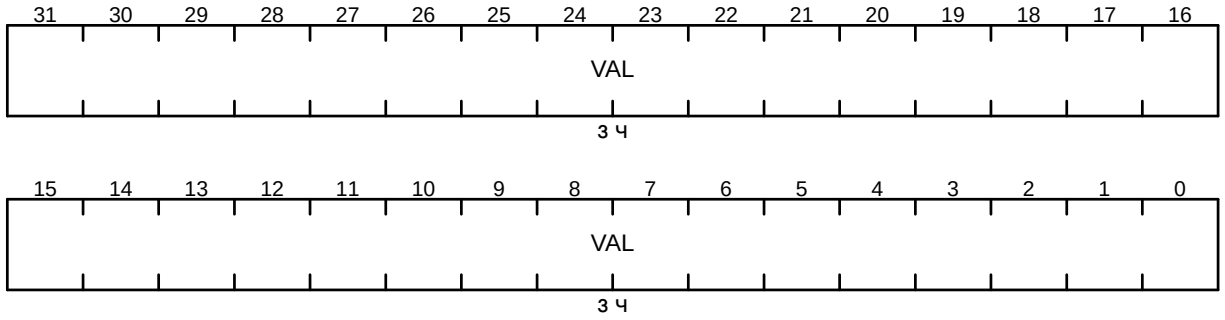
Смещение: + 184h
Сброс: 0000_0000h



Поле	Биты	Описание
DEU	1	Бит статуса обнаружения двойных ошибок
SEU	0	Бит статуса обнаружения одиночных ошибок
–	31-2	Зарезервировано

FECCSEC – регистр счетчика одиночных ошибок ECC Flash

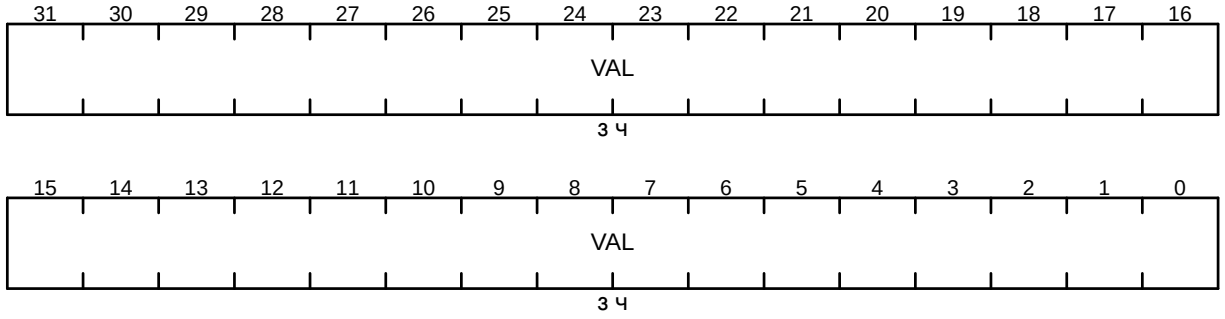
Смещение + 190h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Поле содержит счетчик одиночных ошибок ECC

FECCDEC – регистр счетчика двойных ошибок ECC Flash

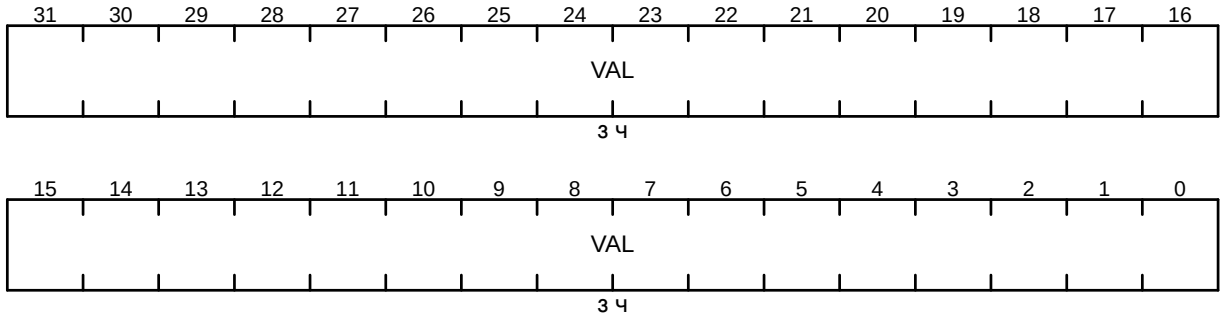
Смещение + 194h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Поле содержит счетчик двойных ошибок ECC

FECCSEA – регистр адреса одиночной ошибки ECC Flash

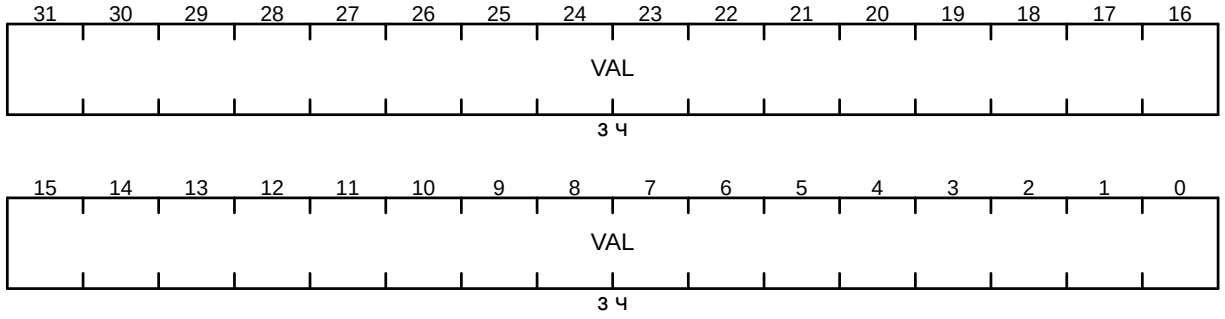
Смещение + 198h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Поле содержит адрес последнего слова данных, в котором обнаружена одиночная ошибка ECC

FECCDEA – регистр адреса двойной ошибки ECC Flash

Смещение + 19Ch
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Поле содержит адрес последнего слова данных, в котором обнаружена двойная ошибка ECC

A.4 Регистры контроллера FlashD-памяти

Базовый адрес: 5000_C000h

Смещение: + 10h (DATA) Регистры данных
+ 170h (ERRMASK) Регистры маски данных

ADDR – регистр адреса Flash-памяти

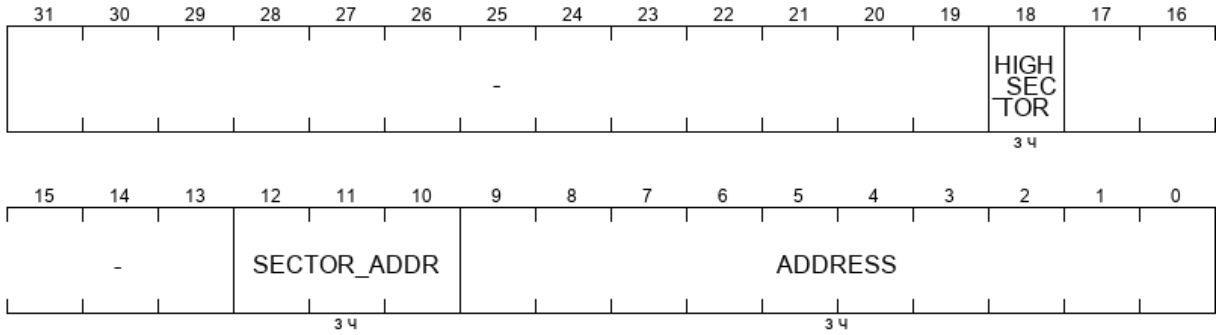
Смещение: + 0h
Сброс: 0h



Поле	Биты	Описание
ADDRESS	18-0	Адрес, используемый при командах записи, чтения и постраничного стирания. Должен быть выровнен по 8 байт. Не выровненные адреса выравниваются автоматически (биты 2-0 не имеют значения)
–	31-19	Зарезервировано

OTP_ADDR – регистр адреса для доступа к OTP области Flash-памяти

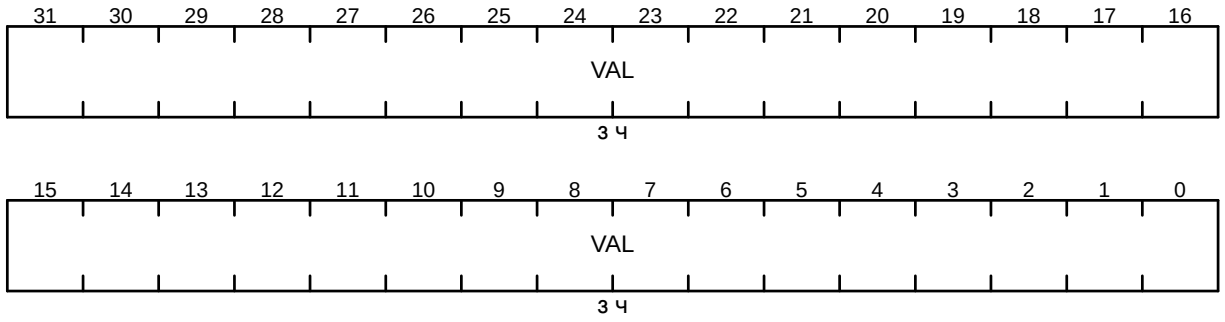
Смещение: + 0h
Сброс: 0h



Поле	Биты	Описание
HIGH_SECTOR	18	Значение адреса сектора (старшая часть) для выбранного блока Flash-памяти
SECTOR_ADDR	12-10	Значение адреса сектора (младшая часть) для выбранного блока Flash-памяти
ADDRESS	9-0	Адрес, используемый при командах записи, чтения и постраничного стирания. Должен быть выровнен по 8 байт. Не выровненные адреса выравниваются автоматически (биты 2-0 не имеют значения)
–	31-19, 17-13	Зарезервировано

DATA – массив регистров данных Flash-памяти

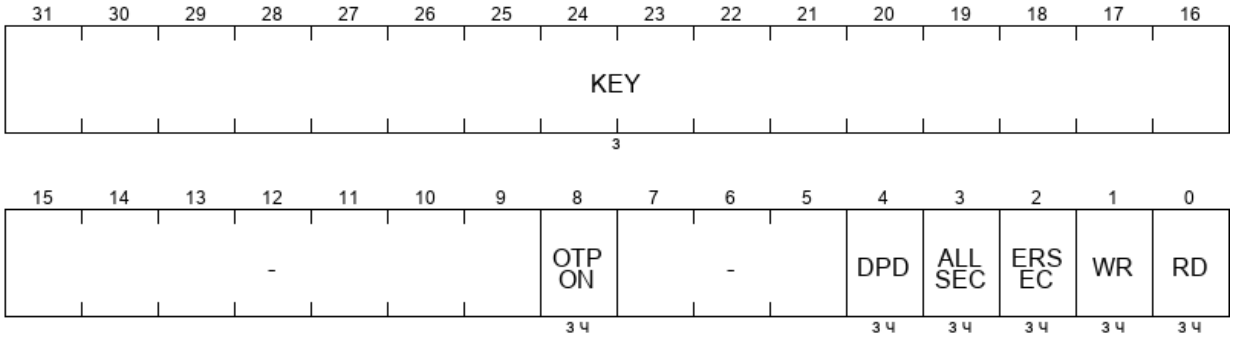
Смещение: DATA + (4*d)h, где d = 0, 1
Сброс: FFFF_FFFFh



Поле	Биты	Описание
VAL	31-0	32-разрядные регистры слов данных. Все слова данных должны быть загружены в регистры до установки бита команды записи. Читаемые данные будут доступны в регистрах после сброса флага BUSY

CMD – регистр команд Flash-памяти

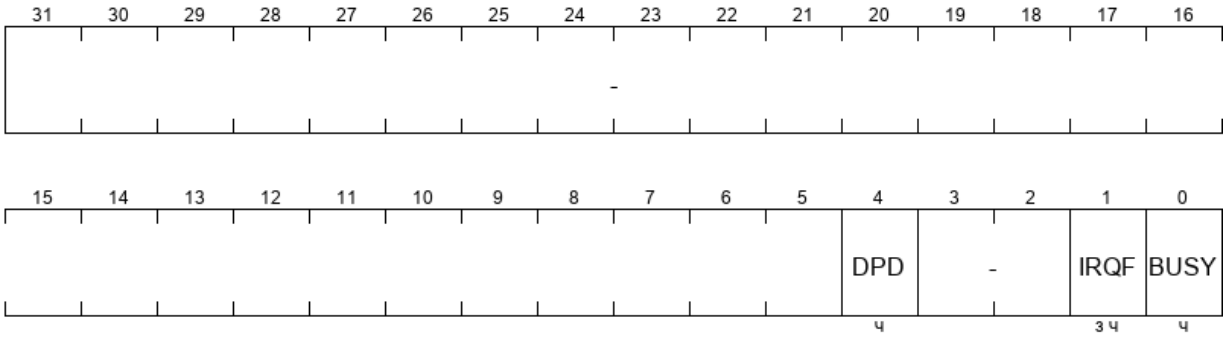
Смещение: + 40h
Сброс: C0DE_0000h



Поле	Биты	Описание
KEY	31-16	Код запуска команды. Все команды для вступления в силу должны сопровождаться записью в поле KEY значения C0DEh. Команды должны выполняться по одной, т. е. запись следующей команды разрешена, только после завершения предыдущей. При одновременной записи нескольких команд будет выполнена та, номер бита которой меньше. Чтение поля KEY всегда возвращает DEC0h
OTPON	8	Бит модификации команды для работы с OTP областью
		0 | Команда выполняется для основной области Flash-памяти
		1 | Команда выполняется для OTP области Flash-памяти.
DPD	4	Бит установки/сброса режима PowerDown
ALLSEC	3	Бит активации команды стирания всех страниц. При установленных битах ALLSEC и ERSEC происходит полное стирание области
ERSEC	2	Бит активации команды стирания страницы области. Адрес страницы вычисляется на основе значения регистра ADDR
WR	1	Бит активации команды записи данных DATAd (d от 0 до 1), начиная с адреса ADDR в области
RD	0	Бит активации команды чтения данных в DATAd (d от 0 до 1), начиная с адреса ADDR в области
–	15-9, 7-5	Зарезервировано

STAT – регистр статуса Flash-памяти

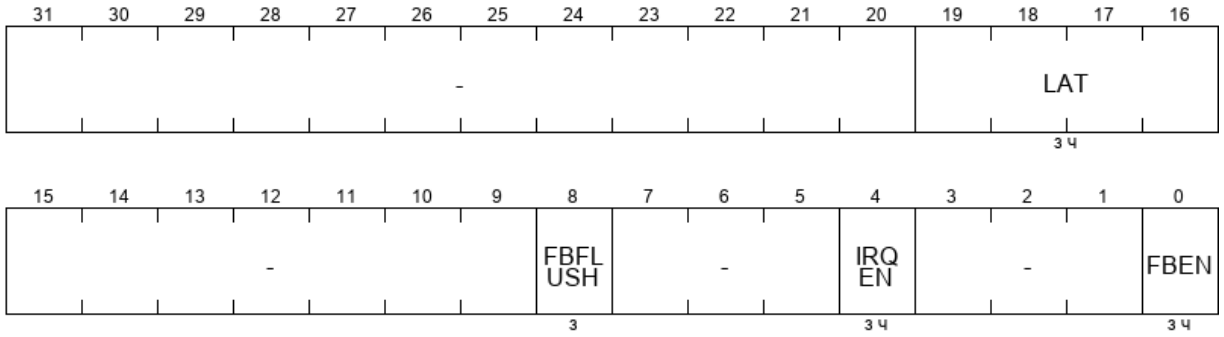
Смещение: + 44h
Сброс: 0h



Поле	Биты	Описание	
DPD	4	Бит статуса режима PowerDown	
IRQF	1	Флаг прерывания по окончании выполнения команды. Устанавливается только если бит IRQEN установлен	
		0	Нет информации
		1	Команда выполнена
		Сбрасывается записью «1»	
BUSY	0	Статус работы контроллера Flash-памяти	
		0	Нет активной команды
		1	Выполняется команда
		Примечание – В связи с особенностями пересинхронизации, при работе на высоких частотах ядра необходимо добавлять задержку между записью регистра CMD и чтением флага BUSY, например, 5 NOP команд	
–	31-5, 3-2	Зарезервировано	

CTRL – регистр настройки контроллера Flash-памяти

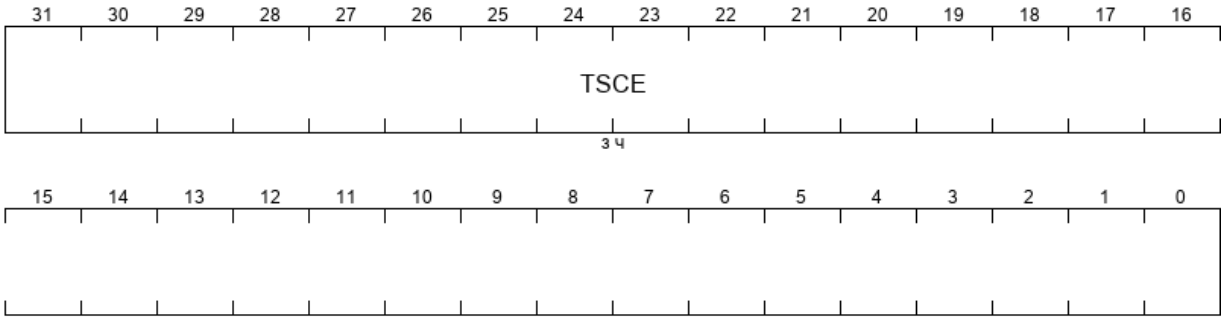
Смещение: + 48h
Сброс: 0001_0000h



Поле	Биты	Описание
LAT	19-16	Поле задания количества дополнительных тактов ожидания при чтении из Flash-памяти
FBFLUSH	8	Бит очистки внутреннего буфера предвыборки
IRQEN	4	Бит разрешения прерываний
FBEN	0	Бит включения буфера предвыборки
–	31-20, 15-9, 7-5, 3-1	Зарезервировано

FTCETR – регистр установки времени стирания Flash

Смещение: + 100h
Сброс: 0000_EA60h

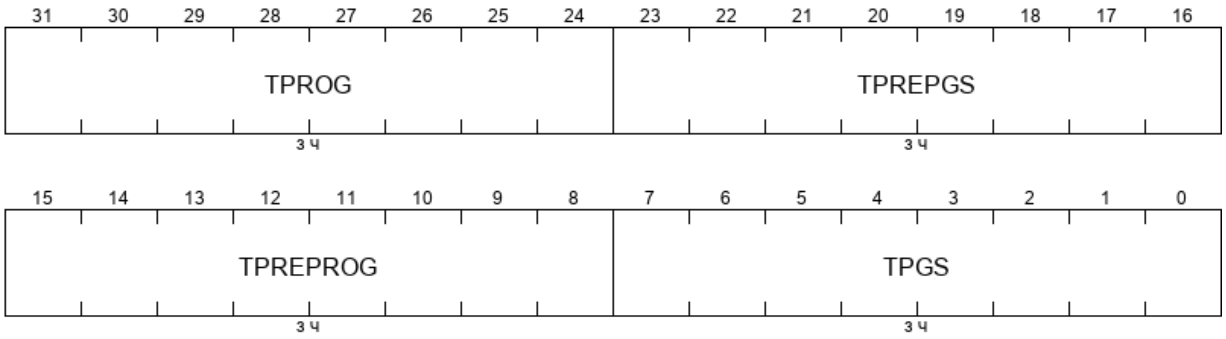


Поле	Биты	Описание
TSCE	31-0	Поле содержит значение времени стирания в циклах clk Запись заблокирована, когда STAT->BUSY = 1

П р и м е ч а н и е – В обычном режиме работы не следует изменять значение по сбросу.

FTCPRT – регистр установки времени программирования Flash

Смещение: + 104h
Сброс: 4820_1030h

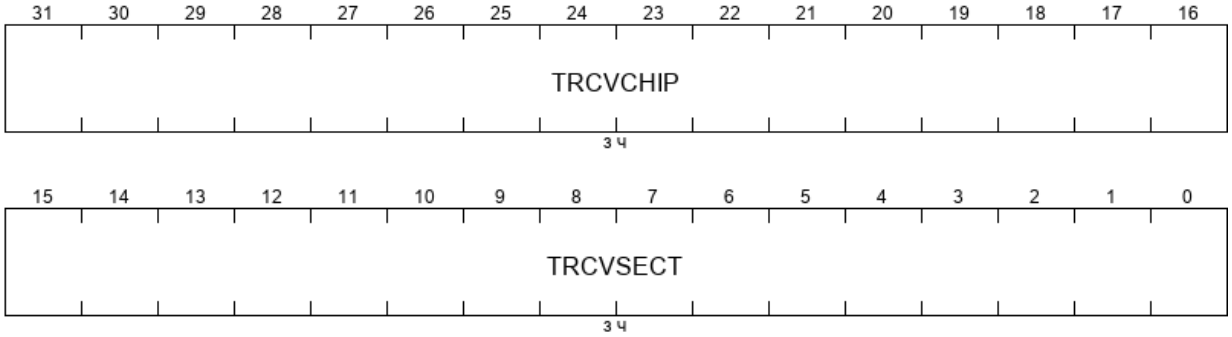


Поле	Биты	Описание
TPROG	31-24	Поле содержит значение времени в циклах clk для соблюдения тайминга PROG Запись заблокирована, когда STAT->BUSY = 1
TPREPGS	23-16	Поле содержит значение времени в циклах clk для соблюдения тайминга PREPGS Запись заблокирована, когда STAT->BUSY = 1
TPREPROG	15-8	Поле содержит значение времени в циклах clk для соблюдения тайминга PREPROG Запись заблокирована, когда STAT->BUSY = 1
TPGS	7-0	Поле содержит значение времени в циклах clk для соблюдения тайминга TPGS Запись заблокирована, когда STAT->BUSY = 1

П р и м е ч а н и е – В обычном режиме работы не следует изменять значение по сбросу.

FTCERTR – регистр установки времени восстановления после стирания Flash

Смещение: + 108h
Сброс: 0320_00C8h

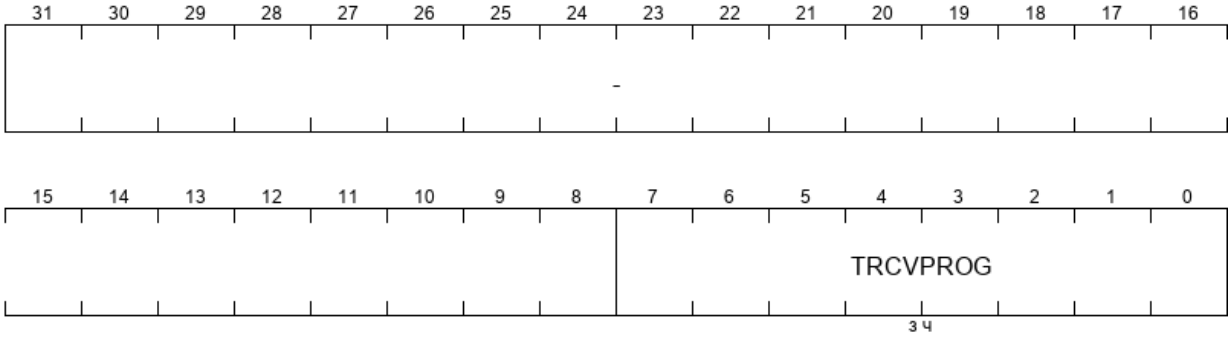


Поле	Биты	Описание
TRCVCHIP	31-16	Поле содержит значение времени восстановления после стирания всей Flash-памяти в циклах clk Запись заблокирована, когда STAT->BUSY = 1
TRCVSECT	15-0	Поле содержит значение времени восстановления после стирания сектора памяти в циклах clk Запись заблокирована, когда STAT->BUSY = 1

П р и м е ч а н и е – В обычном режиме работы не следует изменять значение по сбросу.

FTCPRTR – регистр установки времени восстановления после программирования Flash

Смещение: + 10Ch
Сброс: 0000_0014h

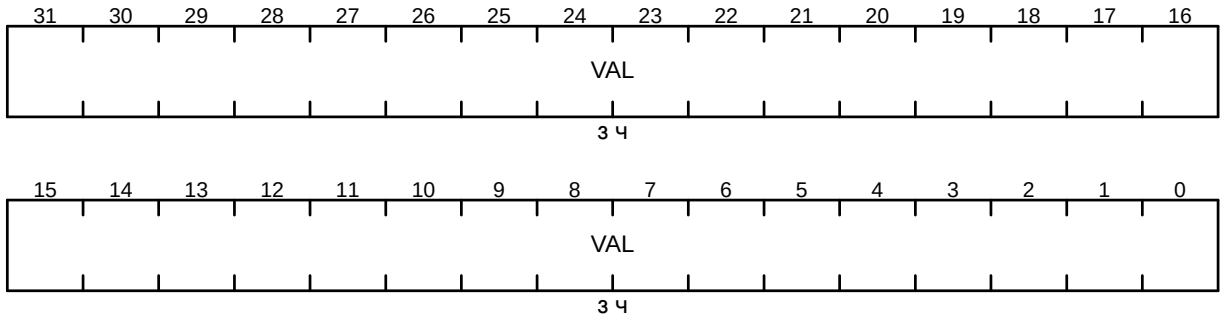


Поле	Биты	Описание
TRCVSECT	7-0	Поле содержит значение времени восстановления после программирования в циклах clk Запись заблокирована, когда STAT->BUSY = 1
–	31-8	Зарезервировано

П р и м е ч а н и е – В обычном режиме работы не следует изменять значение по сбросу.

ERRMASK – массив регистров маски данных Flash-памяти

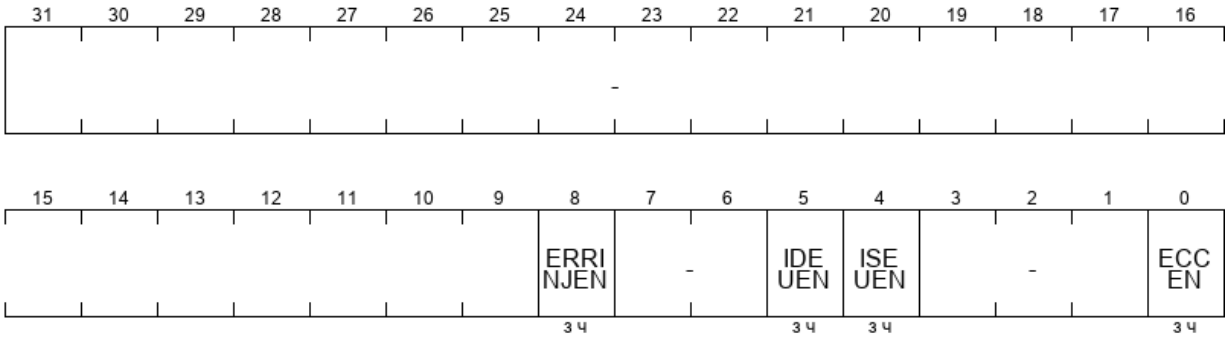
Смещение: ERRMASK + (4*d)h, где d = 0, 1
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Поле содержит маску данных. Соответствующий бит записываемого значения инвертируется.

FECCCR – регистр управления кодом коррекции ошибок Flash

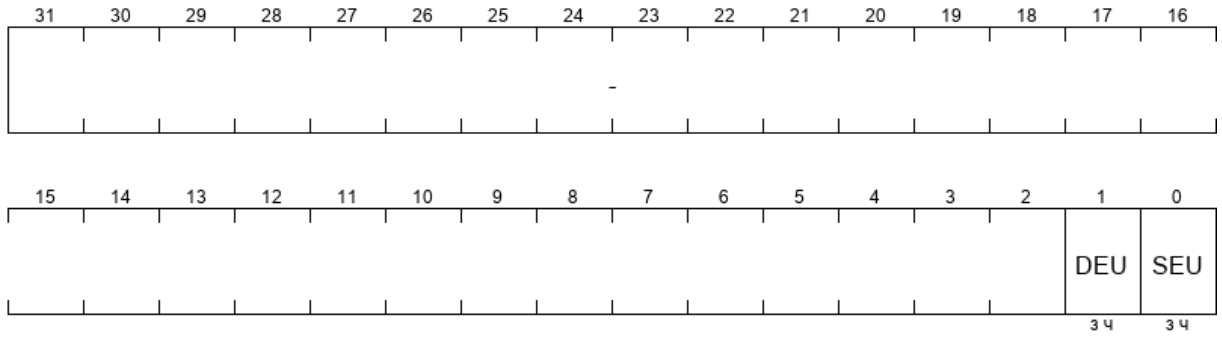
Смещение: + 180h
Сброс: 0000_0000h



Поле	Биты	Описание
ERRINJEN	8	Бит включения внедрения ошибок на шину записи данных
IDEUEN	5	Бит разрешения прерывания при обнаружении двойных ошибок
ISEUEN	4	Бит разрешения прерывания при обнаружении одиночных ошибок
ECCEN	0	Бит разрешения обнаружения и коррекции ошибок ECC
–	31-9, 7-6, 3-1	Зарезервировано

FECCISTAT – регистр статуса прерываний ECC Flash

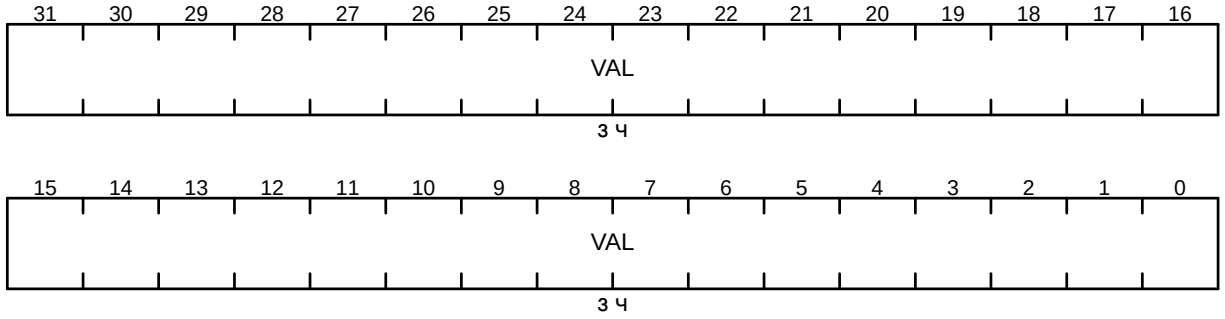
Смещение: + 184h
Сброс: 0000_0000h



Поле	Биты	Описание
DEU	1	Бит статуса обнаружения двойных ошибок
SEU	0	Бит статуса обнаружения одиночных ошибок
–	31-2	Зарезервировано

FECCSEC – регистр счетчика одиночных ошибок ECC Flash

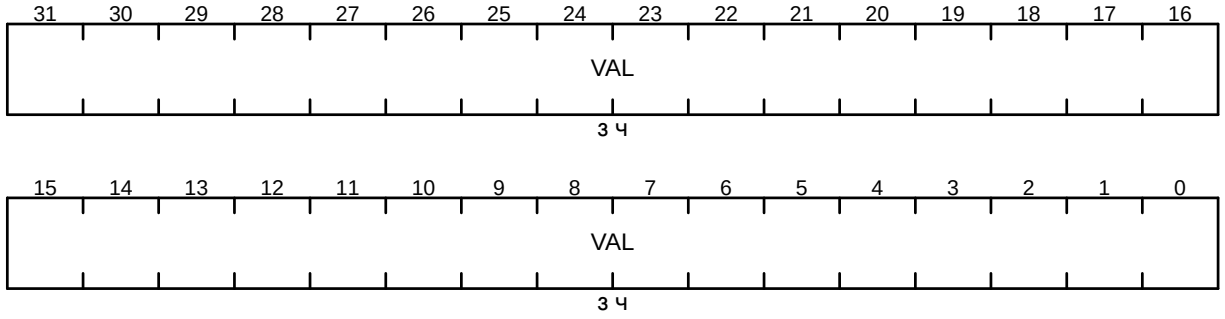
Смещение + 190h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Поле содержит счетчик одиночных ошибок ECC

FECCDEC – регистр счетчика двойных ошибок ECC Flash

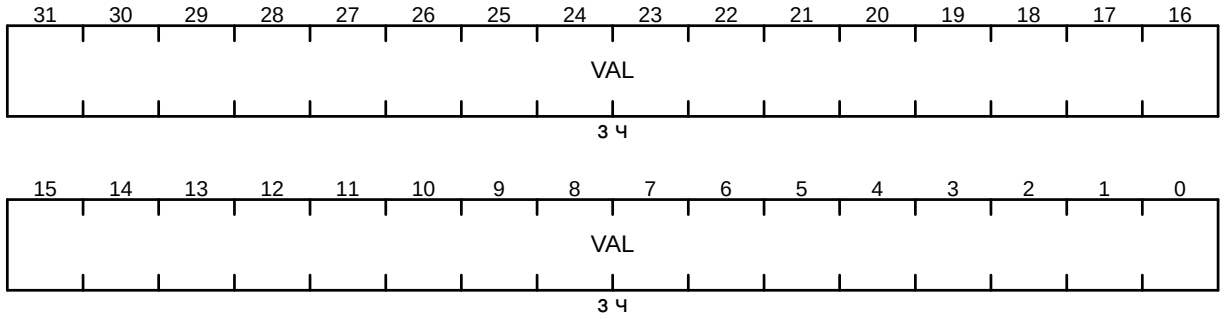
Смещение + 194h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Поле содержит счетчик двойных ошибок ECC

FECCSEA – регистр адреса одиночной ошибки ECC Flash

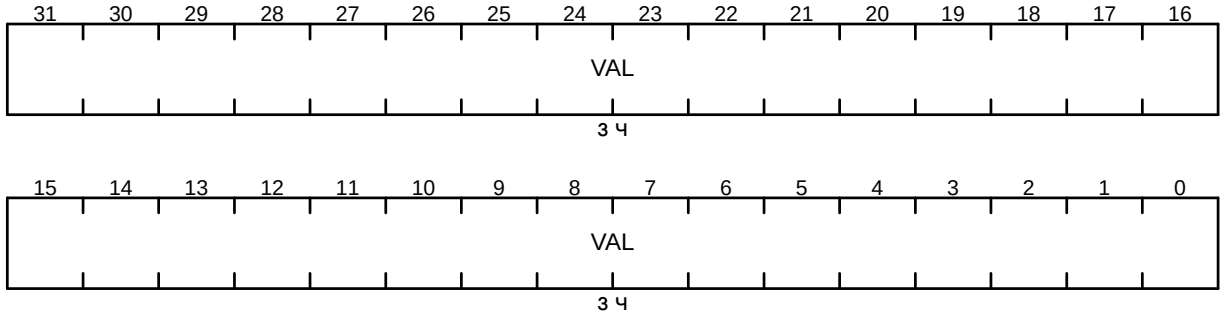
Смещение + 198h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Поле содержит адрес последнего слова данных, в котором обнаружена одиночная ошибка ECC

FECCDEA – регистр адреса двойной ошибки ECC Flash

Смещение + 19Ch
Сброс: 0h



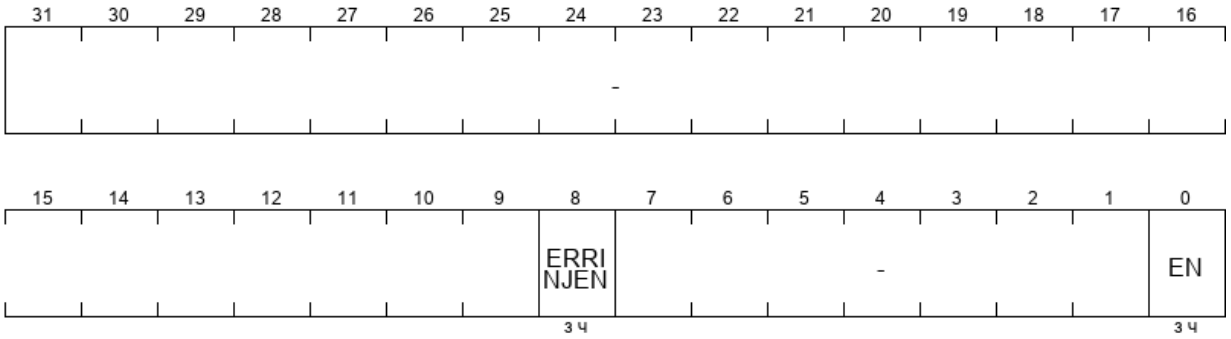
Поле	Биты	Описание
VAL	31-0	Поле содержит адрес последнего слова данных, в котором обнаружена двойная ошибка ECC

A.5 Регистры контроллера памяти SRAM с кодом коррекции Хэмминга

Базовый адрес: 5000_7000h

CFG – регистр управления контроллером памяти

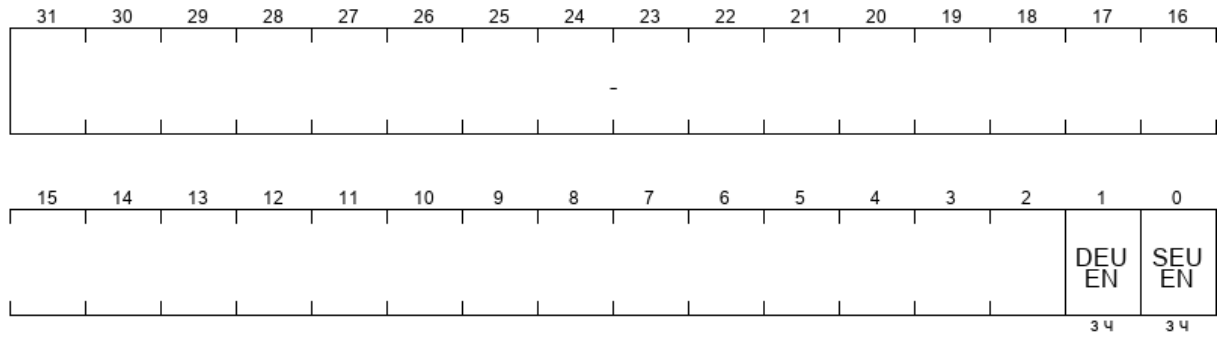
Смещение: + 00h
Сброс: 0h



Поле	Биты	Описание
ERRINJEN	8	Бит влючения инжекции ошибок при записи на шину данных
EN	0	Бит включения кода коррекции Хэмминга
–	31-9, 7-1	Зарезервировано

IRQEN – регистр разрешения прерываний

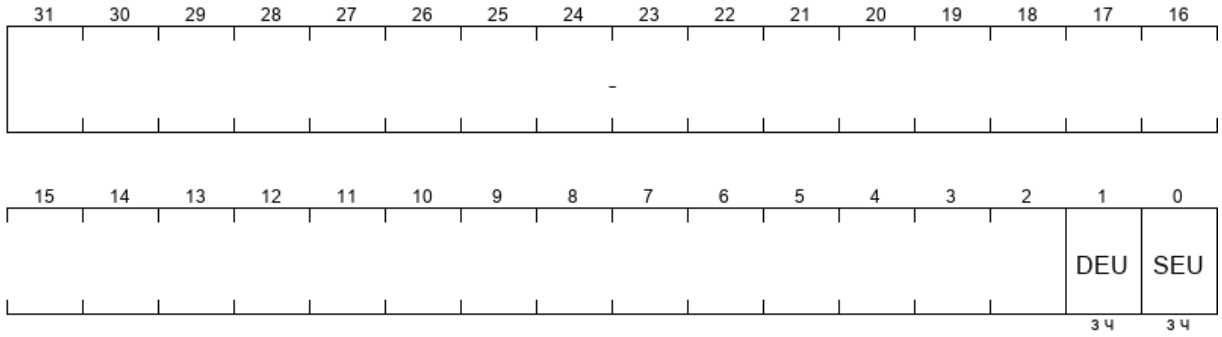
Смещение: + 04h
Сброс: 0h



Поле	Биты	Описание
DEUEN	1	Бит разрешения прерывания при обнаружении двухбитовой двойной ошибки
SEUEN	0	Бит разрешения прерывания при обнаружении однобитовой одиночной ошибки
–	31-2	Зарезервировано

IRQSTAT – регистр статуса прерываний

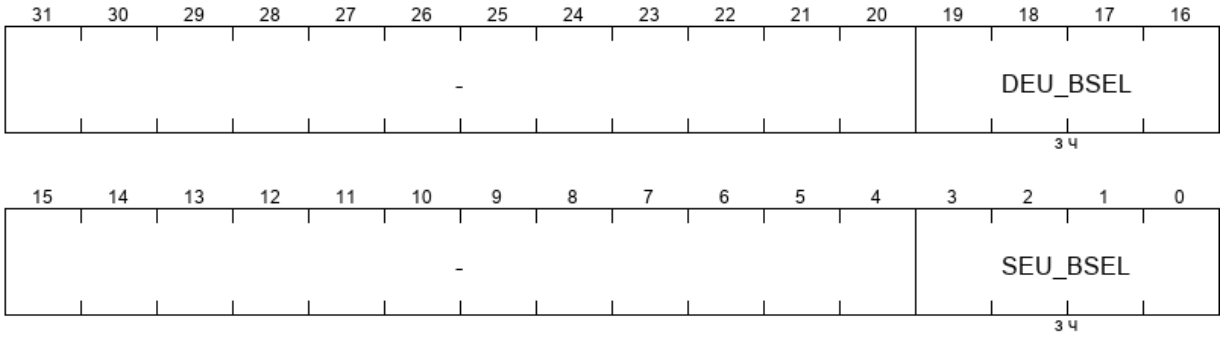
Смещение: + 08h
Сброс: 0h



Поле	Биты	Описание
DEUEN	1	Бит статуса прерывания при возникновении двухбитовой двойной ошибке
SEUEN	0	Бит статуса прерывания при возникновении однобитовой одиночной ошибки
–	31-2	Зарезервировано

ERROR_POS – регистр позиции ошибки

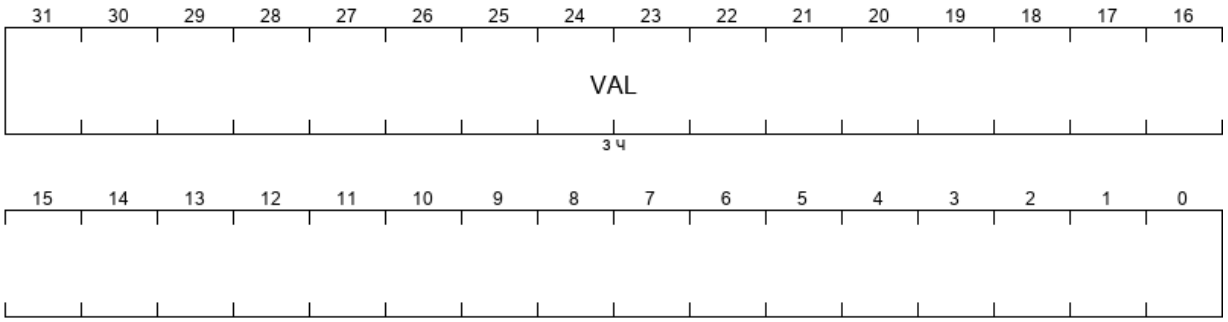
Смещение: + 0Ch
Сброс: 0h



Поле	Биты	Описание
DEU_BSEL	19-16	Позиция байта в слове, в котором произошла двухбитовая ошибка
SEU_BSEL	3-0	Позиция байта в слове, в котором произошла однобитовая ошибка
–	31-20, 15-4	Зарезервировано

SEU_CNTR – регистр счетчика однобитовых ошибок

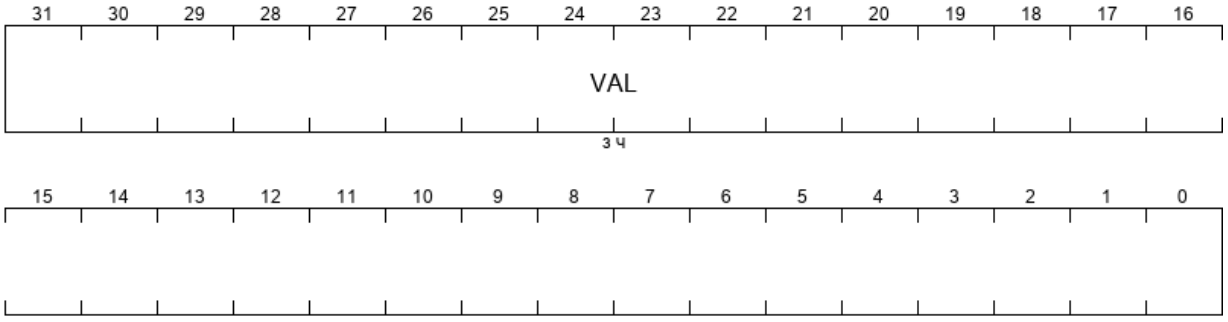
Смещение: + 10h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Значение количества обнаруженных однобитовых ошибок

DEU_CNTR – регистр счетчика двухбитовых ошибок

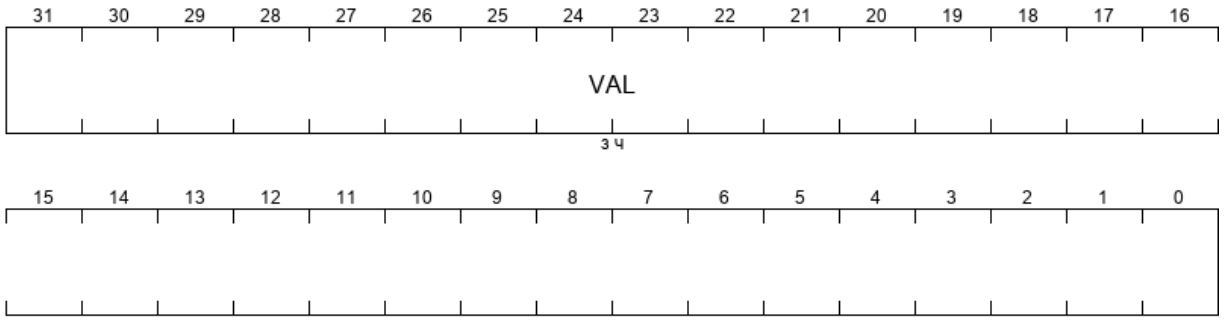
Смещение: + 14h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Значение количества обнаруженных двухбитовых ошибок

SEU_ADDR – регистр адреса однобитовой ошибки

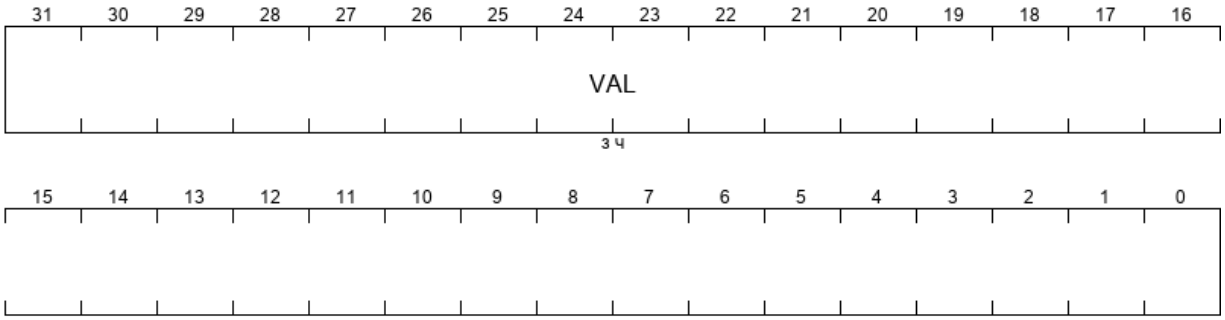
Смещение: + 18h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Последний зафиксированный адрес при возникновении однобитовой ошибки

DEU_ADDR – регистр адреса двухбитовой ошибки

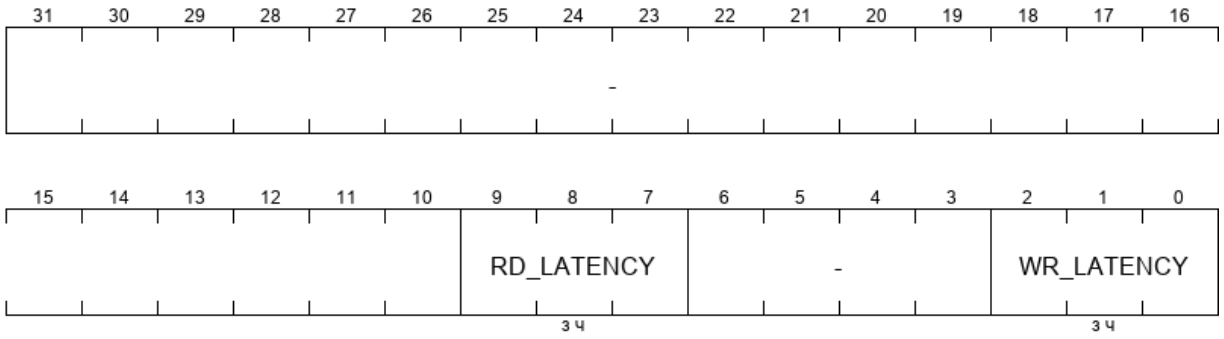
Смещение: + 1Ch
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Последний зафиксированный адрес при возникновении двухбитовой ошибки

LATENCY – регистр задержек записи и чтения

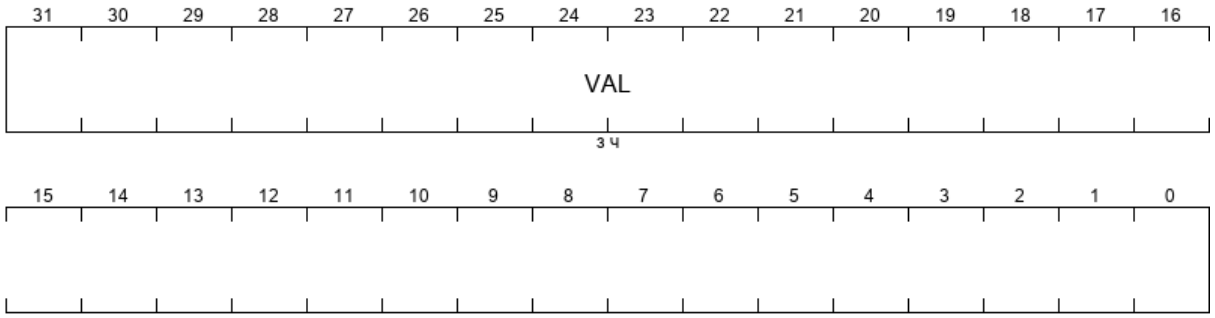
Смещение: + 20h
Сброс: 0h



Поле	Биты	Описание
RD_LATENCY	9-7	Поле задержки чтения из SRAM в тактах системной частоты
WR_LATENCY	2-0	Поле задержки записи в SRAM в тактах системной частоты
–	31-10, 6-3	Зарезервировано

ERRMASK – регистр маски данных для инжекции ошибок

Смещение: + 24h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Соответствующий бит 32-разрядной шины данных при записи инвертируется для тестирования кода коррекции Хэмминга

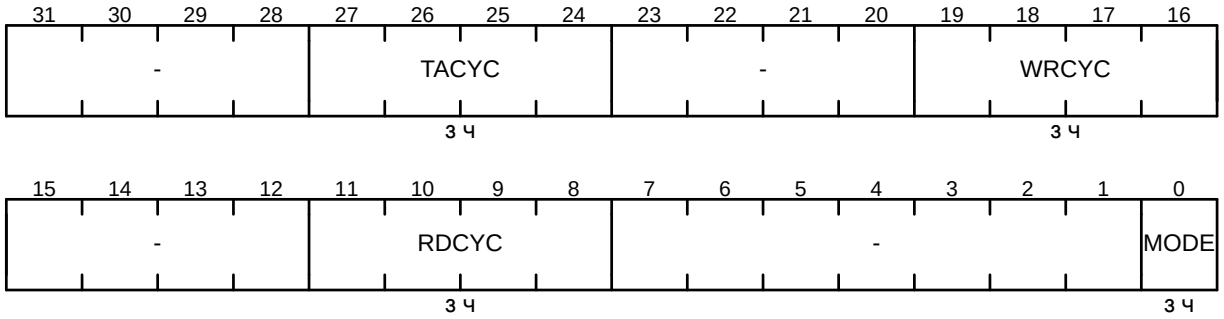
A.6 Регистры контроллера внешней памяти EMC

Базовый адрес: 5000_9000h

Примечание: d – номер окна от 0 до 7

WINCFG – массив регистров настройки контроллера внешней памяти

Смещение: (4*d)h
Сброс: 0707_0701h



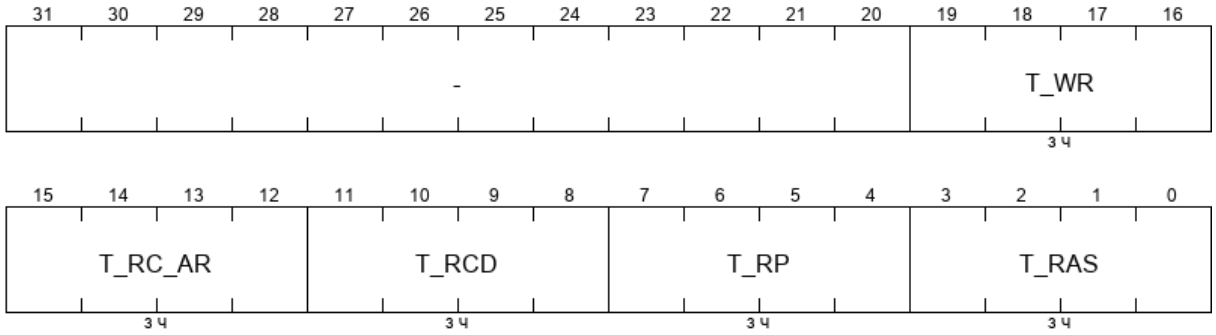
Поле	Биты	Описание	
TACYC	27-24	Длительность цикла переключения шины. Поле задает паузу между циклами чтения и записи на внешней шине в количестве тактов сигнала системной частоты. Значение 0000b соответствует одному такту, значение 1111b – шестнадцати	
WRCYC	19-16	Длительность цикла записи слова данных. Поле задает длительность цикла записи на внешней шине в количестве тактов сигнала системной частоты. Значение 0000b соответствует одному такту, значение 1111b – шестнадцати	
RDCYC	11-8	Длительность цикла чтения слова данных. Поле задает длительность цикла чтения на внешней шине в количестве тактов сигнала системной частоты. Значение 0000b соответствует одному такту, значение 1111b – шестнадцати	
MODE	0	Бит задания разрядности контроллера внешней памяти	
		0	8-разрядный режим работы
		1	16-разрядный режим работы
–	31-28, 23-20, 15-12, 7-1	Зарезервировано	

A.7 Регистры контроллера SDRAM

Базовый адрес: 5000_A000h

TIMING – регистр таймингов SDRAM

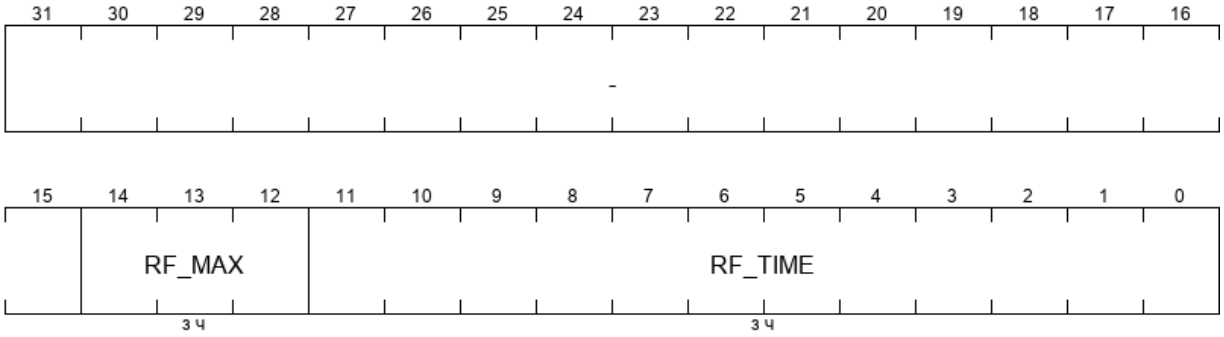
Смещение: + 00h
Сброс: 0001_7224h



Поле	Биты	Описание
T_WR	19-16	Период восстановления после записи. Указывается в тактах системной частоты
T_RC_AR	15-12	Период между активизацией двух различных строк одного и того же банка. Указывается в тактах системной частоты
T_RCD	11-8	Период между подачей адреса строки и столбца. Относится к задержке между активацией строки в памяти и доступом к столбцу. Указывается в тактах системной частоты
T_RP	7-4	Период подзарядки строки, необходимый для деактивации строки после того, как данные были прочитаны или записаны в нее. Указывается в тактах системной частоты
T_RAS	3-0	Период активности строки от момента ее активации до момента поступления команды подзарядки. Указывается в тактах системной частоты
–	31-20	Зарезервировано

REFRESH – регистр автоматического обновления

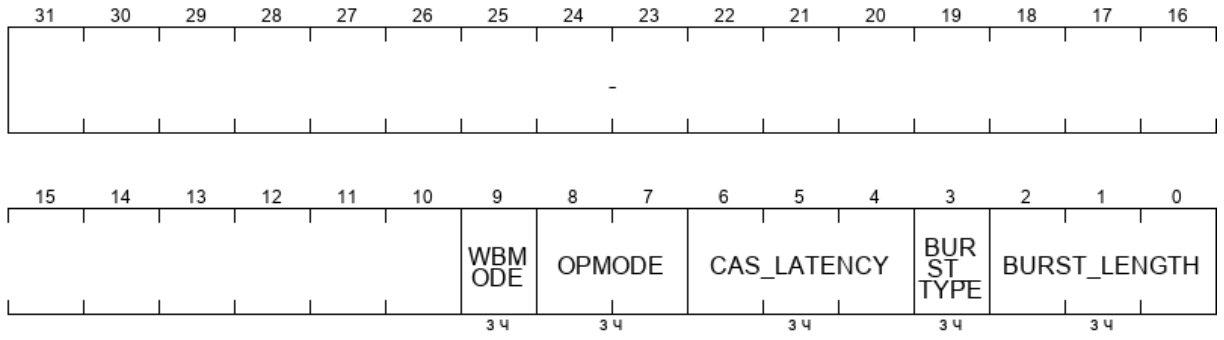
Смещение: + 04h
Сброс: 0000_6100h



Поле	Биты	Описание
RF_MAX	14-12	Максимальное количество строк, обновляемых за один раз
RF_TIME	11-0	Период между командами автоматического обновления, выдаваемыми контроллером. Указывается в тактах системной частоты
–	31-15	Зарезервировано

MODE – регистр управления режимами SDRAM

Смещение: + 08h
Сброс: 0003_0033h

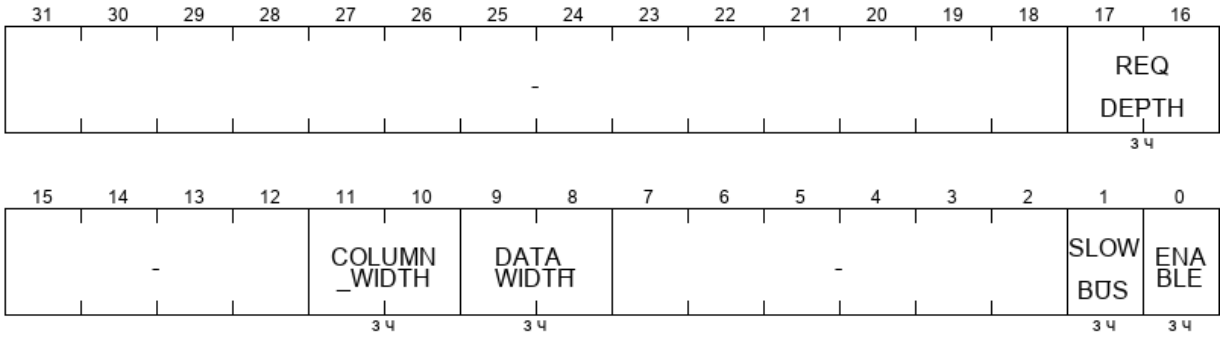


Поле	Биты	Описание
WBMODE	9	Режим пакетной записи. При установке доступ на запись осуществляется в отдельные ячейки (без пакета)
OPMODE	8-7	Режим работы. Поле должно содержать нулевые значения. Остальные комбинации значений зарезервированы.
CAS_LATENCY	6-4	Задержка CAS. Задержка в тактовых циклах между регистрацией команды чтения и доступностью первого фрагмента выходных данных

Поле	Биты	Описание	
BURST_TYPE	3	Тип пакета	
		0	последовательный
		1	с чередованием
BURST_LENGTH	2-0	Длина пакета	
		000b	1
		001b	2
		010b	4
		011b	8
		–	
		111b	Страница
–	31-10	Зарезервировано	

CONTROL – регистр управления SDRAM контроллером

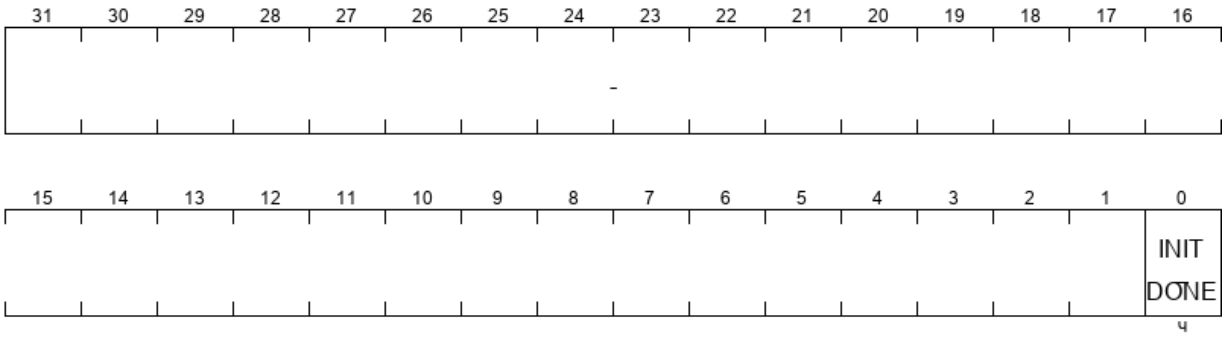
Смещение: + 0Ch
Сброс: 0003_0000h



Поле	Биты	Описание	
REQ_DEPTH	17-16	Сколько запросов должен содержать внутренний буфер	
COLUMN_WIDTH	11-10	Количество разрядов для адресации столбцов	
		00b	8 бит
		01b	9 бит
		10b	10 бит
		11b	11 бит
DATA_WIDTH	9-8	Ширина данных SDRAM памяти (размер слова данных)	
		00b	32 бита
		01b	16 бит
		10b	8 бит
		11b	Зарезервировано
SLOW_BUS	1	Данный бит должен быть установлен, если системная тактовая частота меньше частоты SDRAM	
ENABLE	0	Бит включения SDRAM контроллера	
–	31-18, 15-12, 7-2	Зарезервировано	

STATUS – регистр статуса

Смещение: + 10h
Сброс: 0000_0000h



Поле	Биты	Описание
INIT_DONE	0	Бит указывает на то, что инициализация SDRAM завершена
–	31-30	Зарезервировано

A.8 Регистры контроллера прямого доступа к памяти DMA

Базовый адрес:	5000_0000h	Регистры DMA
Смещение:	+ 000h (CH0)	Регистры канала 0
	+ 080h (CH1)	Регистры канала 1
	+ 100h (CH2)	Регистры канала 2
	+ 180h (CH3)	Регистры канала 3
	+ 200h (CH4)	Регистры канала 4
	+ 280h (CH5)	Регистры канала 5
	+ 300h (CH6)	Регистры канала 6
	+ 380h (CH7)	Регистры канала 7
	+ 400h (CH8)	Регистры канала 8
	+ 480h (CH9)	Регистры канала 9
	+ 500h (CH10)	Регистры канала 10
	+ 580h (CH11)	Регистры канала 11
	+ 600h (CH12)	Регистры канала 12
	+ 680h (CH13)	Регистры канала 13
	+ 700h (CH14)	Регистры канала 14
	+ 780h (CH15)	Регистры канала 15
	+ 800h (CH16)	Регистры канала 16
	+ 880h (CH17)	Регистры канала 17
	+ 900h (CH18)	Регистры канала 18
	+ 980h (CH19)	Регистры канала 19
	+ A00h (CH20)	Регистры канала 20
	+ A80h (CH21)	Регистры канала 21
	+ B00h (CH22)	Регистры канала 22
	+ B80h (CH23)	Регистры канала 23
	+ C00h (CH24)	Регистры канала 24
	+ C80h (CH25)	Регистры канала 25
	+ D00h (CH26)	Регистры канала 26
	+ D80h (CH27)	Регистры канала 27
	+ E00h (CH28)	Регистры канала 28
	+ E80h (CH29)	Регистры канала 29
	+ F00h (CH30)	Регистры канала 30
	+ F80h (CH31)	Регистры канала 31

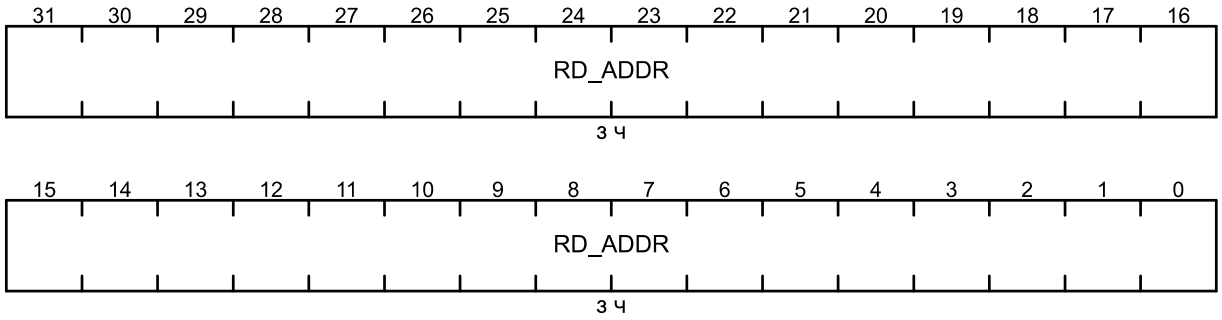
Мнемоника: CHn

Примечание – n – номер канала DMA от 0 до 31

A.8.1 Регистры каналов DMA

SRC_PTR – регистр начального адреса буфера чтения

Смещение: CHn + 00h
Сброс: 0h

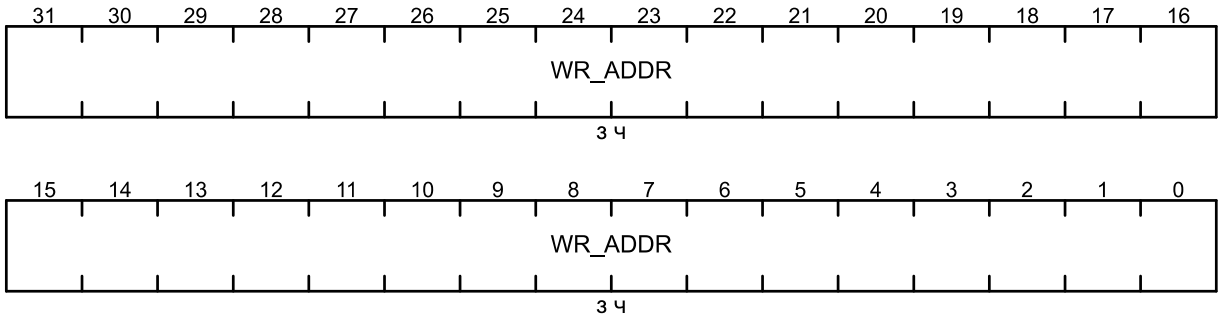


Поле	Биты	Описание
RD_ADDR	31-0	Начальный адрес буфера чтения. По этому адресу будет произведено чтение данных по шине AXI. Если адрес не выровнен по размеру FIFO канала, максимальный размер пакета будет ограничен половиной размера FIFO, вместо того чтобы быть ограниченным размером FIFO.

Примечание – При использовании списков дескрипторов DMA этот регистр будет перезаписан содержимым следующего дескриптора.

DST_PTR – регистр начального адреса буфера записи

Смещение: CHn + 04h
Сброс: 0h

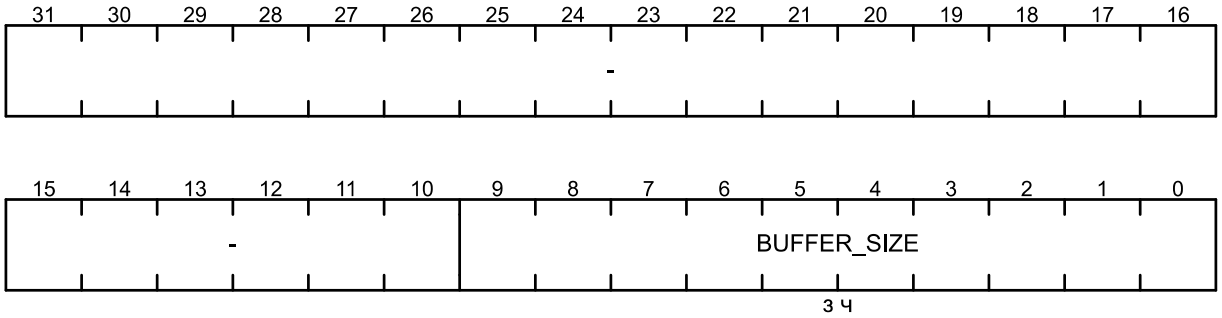


Поле	Биты	Описание
WR_ADDR	31-0	Начальный адрес буфера записи. По этому адресу будет произведена запись данных по шине AXI. Если адрес не выровнен по размеру FIFO канала, максимальный размер пакета будет ограничен половиной размера FIFO, вместо того чтобы быть ограниченным размером FIFO.

Примечание – При использовании списков дескрипторов DMA этот регистр будет перезаписан содержимым следующего дескриптора.

NDTL – регистр размера буфера

Смещение: CHn + 08h
Сброс: 0h

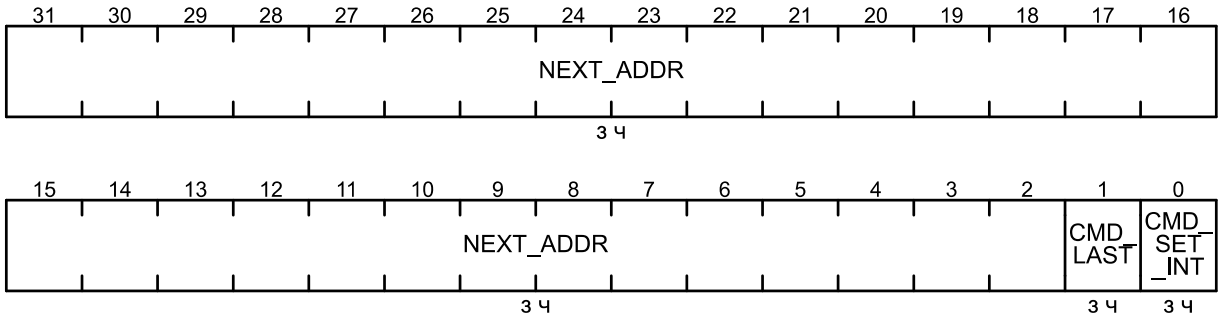


Поле	Биты	Описание
BUFFER_SIZE	9-0	Количество передач. Значение указано в байтах и не имеет ограничений по размеру
–	31-10	Зарезервировано

Примечание – При использовании списков дескрипторов DMA этот регистр будет перезаписан содержимым следующего дескриптора.

CONFIG – регистр настроек команды канала

Смещение: CHn + 0Ch
Сброс: 0h

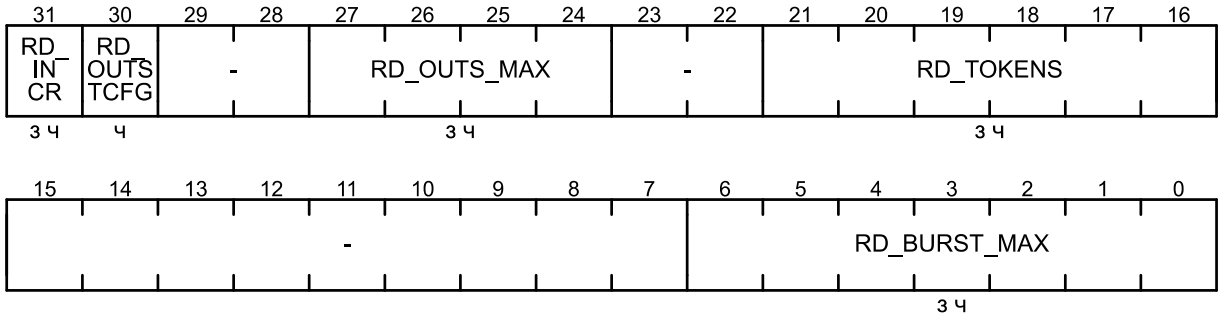


Поле	Биты	Описание
NEXT_ADDR	31-2	Старшая часть адреса следующего дескриптора. Адрес должен быть выровнен по границе 4 байта. Следующий дескриптор будет считан по шине AXI, если бит CMD_LAST сброшен
CMD_LAST	1	Если данный бит установлен, канал остановится после передачи всего буфера. Если нет, то следующий дескриптор будет загружен с адреса, указанного в поле NEXT_ADDR
CMD_SET_INT	0	Если данный бит установлен, канал выдаст прерывание, как только будет передан весь буфер

Примечание – При использовании списков дескрипторов DMA этот регистр будет перезаписан содержимым следующего дескриптора.

STATIC0 – регистр статической конфигурации канала

Смещение: CHn + 10h
Сброс: 8001_0000h

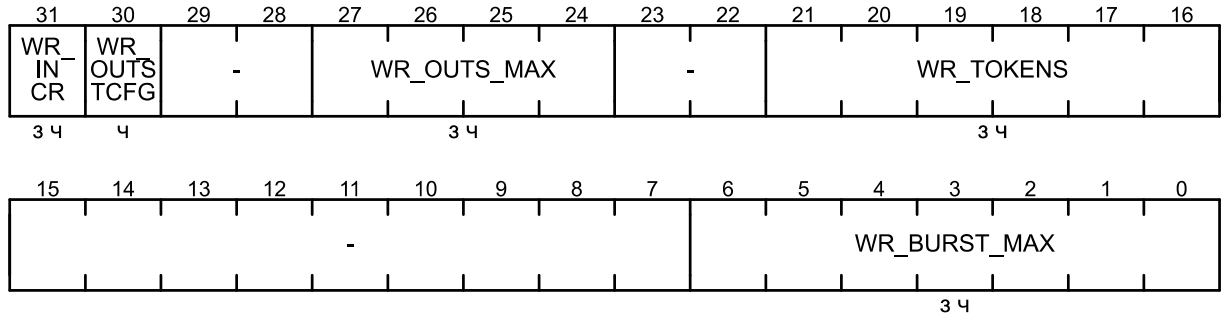


Поле	Биты	Описание
RD_INCR	31	При установке данного бита контроллер DMA инкрементирует адрес следующего пакета при чтении. Данный бит должен быть установлен при работе с памятью. При работе с периферийными буферами FIFO со статическим адресом данный бит должен быть сброшен. Значение по умолчанию – 1.
RD_OUTSTCFG	30	Функционал не реализован. Внимание! Установка данного бита может привести к возникновению ошибки передачи. Пока данный бит установлен, контроллер DMA будет выдавать на шину AXI запросы чтения даже если буфер FIFO канала заполнен, в расчете на то, что какая-то часть данных будет выгружена из буфера до того, как поступят новые считанные данные по шине AXI. Если данный расчет не оправдается, то буфер FIFO будет переполнен, данные потеряны и будет вызвано прерывание переполнения OVERFLOW
RD_OUTS_MAX	27-24	Максимальное количество незавершенных доступов чтения по шине AXI. Ограничение этого значения глубиной команды для ведомого устройства предотвратит остановку командной шины.
RD_TOKENS	21-16	Количество доступов чтения AXI, которые будут выполнены до того, как канал освободит командную шину AXI. Значение по умолчанию - 1
RD_BURST_MAX	6-0	Максимальное количество байт в пакете чтения. Возможные значения: 1, 2, 4×N, где N – целое число от 1 до 16. Если канал работает с буфером периферийного устройства, количество должно соответствовать размеру буфера FIFO периферийного устройства. При копировании в память рекомендуется выбрать значение кратное 4, максимальное-64
–	29-28, 23-22, 15-7	Зарезервировано

Примечание – Эти параметры не следует изменять в процессе работы канала. Данный регистр используется равнозначно регистру записи в совместном режиме.

STATIC1 – регистр статической конфигурации канала

Смещение: CHn + 14h
Сброс: 8001_0000h

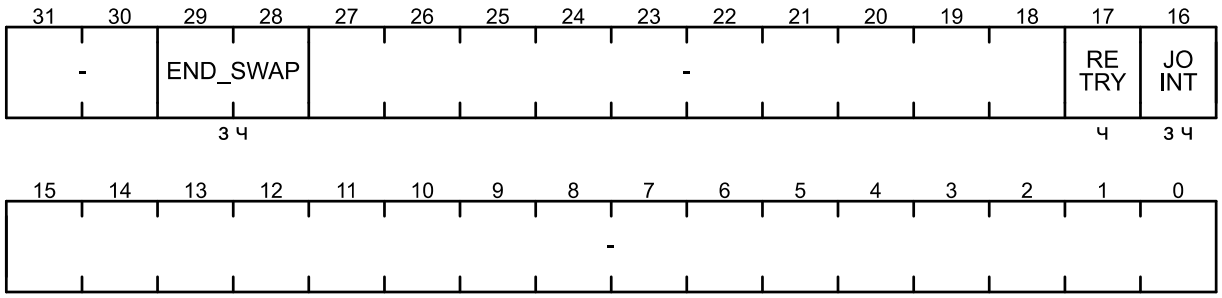


Поле	Биты	Описание
WR_INCR	31	При установке данного бита контроллер DMA инкрементирует адрес следующего пакета при чтении. Данный бит должен быть установлен при работе с памятью. При работе с периферийными буферами FIFO со статическим адресом данный бит должен быть сброшен. Значение по умолчанию – 1
WR_OUTSTCFG	30	Функционал не реализован. Внимание! Установка данного бита может привести к возникновению ошибки передачи. Если данный бит установлен, контроллеру DMA разрешено выдавать на шину AXI запрос записи сразу после подачи запроса чтения, до фактического поступления считанных данных. Это наиболее эффективно при копировании высокоприоритетных данных в память, особенно если ведомое устройство, в которое поступают данные на запись, сможет быть эффективнее за счет более ранней команды. Если выдача данных на запись произойдет до поступления считанных данных, будет вызвано прерывание UNDERFLOW (недозагрузка буфера)
WR_OUTS_MAX	27-24	Максимальное количество незавершенных команд записи по шине AXI. Ограничение этого значения глубиной команды для ведомого устройства предотвратит остановку командной шины
WR_TOKENS	21-16	Количество доступов записи AXI, которые будут выполнены до того, как канал освободит командную шину AXI. Значение по умолчанию – 1
WR_BURST_MAX	6-0	Максимальное количество байт в пакете записи. Возможные значения: 1, 2, 4×N, где N – целое число от 1 до 16. Если канал работает с буфером периферийного устройства, количество должно соответствовать размеру буфера FIFO периферийного устройства. При копировании в память рекомендуется выбрать значение кратное 4, максимальное - 64
–	29-28, 23-22, 15-7	Зарезервировано

Примечание – Эти параметры не следует изменять в процессе работы канала.

STATIC2 – регистр статической конфигурации канала

Смещение: CHn + 18h
Сброс: 0h

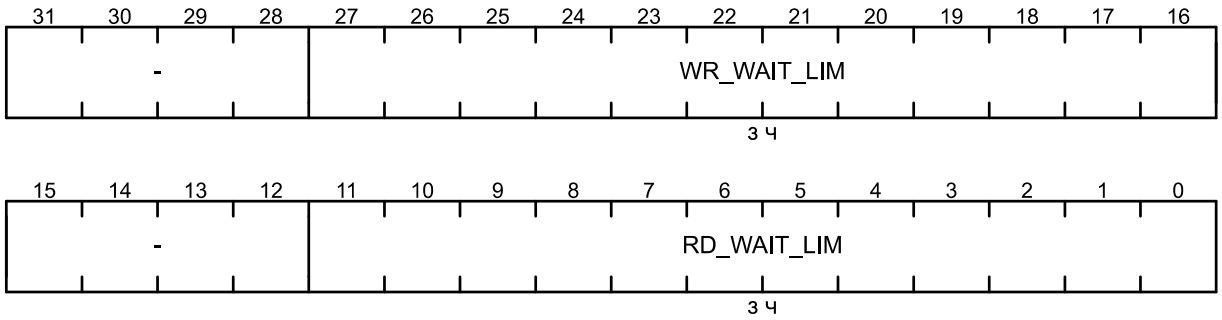


Поле	Биты	Описание
END_SWAP	29-28	Замена байтов по порядку следования. 0 – Без замены; 1 – Поменять местами байты в пределах 16 бит; 2 – Поменять местами байты в пределах 32 бит; 3 – Зарезервировано.
RETRY	17	Если данный бит установлен, канал автоматически повторит попытку передачи при возникновении переполнения или недозагрузки внутреннего буфера FIFO
JOINT	16	Если установлен, канал будет работать в совместном режиме, действует только в том случае, если установлен бит JOINT_MODE регистра CONTROL
–	31-30, 27-18, 15-0	Зарезервировано

Примечание – Эти параметры не следует изменять в процессе работы канала.

STATIC3 – регистр статической конфигурации канала

Смещение: CHn + 1Ch
Сброс: 0h

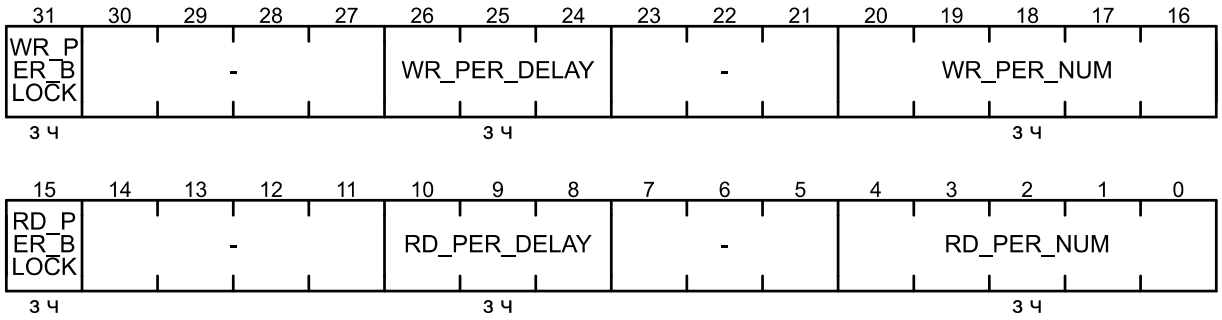


Поле	Биты	Описание
WR_WAIT_LIM	27-16	Минимальное количество циклов ожидания после освобождения канала и до того, как он начнет выдавать дополнительные команды записи. Это помогает предотвратить засорение шины AXI каналами с низким приоритетом. Значение должно быть умножено на 16. Не используется в совместном режиме
RD_WAIT_LIM	11-0	Минимальное количество циклов ожидания после освобождения канала и до того, как он начнет выдавать дополнительные команды чтения. Это помогает предотвратить засорение шины AXI каналами с низким приоритетом. Значение должно быть кратно 16. Управляет также WR_WAIT_LIM в совместном режиме
–	31-28, 15-12	Зарезервировано

Примечание – Эти параметры не следует изменять в процессе работы канала.

STATIC4 – регистр статической конфигурации канала

Смещение: CHn + 20h
Сброс: 0h

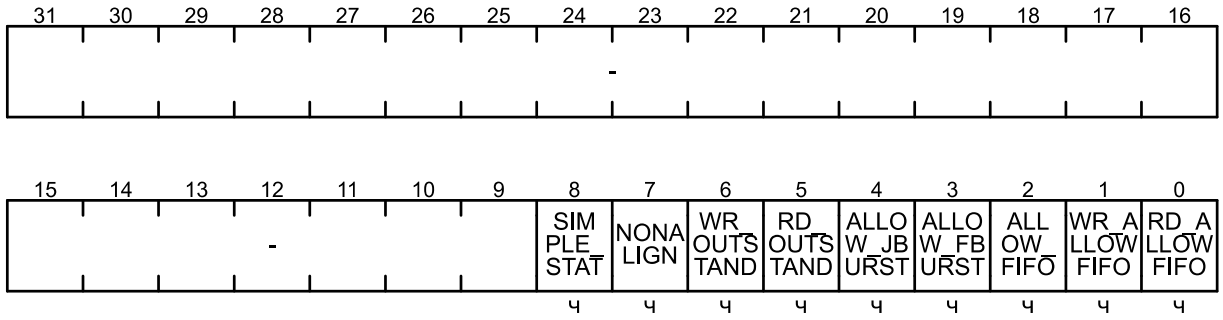


Поле	Биты	Описание
WR_PER_BLOCK	31	Если данный бит установлен, работа с периферией на запись будет производиться в блочном режиме.
WR_PER_DELAY	26-24	Количество циклов ожидания для периферийного запроса на запись для обновления периферийного устройства, после сброса сигнала записи. Определяется задержкой периферийного устройства
WR_PER_NUM	21-16	Номер аппаратного источника запроса записи (TX). Установите 0, если канал выполняет запись в память или на периферийное устройство, которое не использует периферийное управление запросом
RD_PER_BLOCK	15	Если данный бит установлен, периферийное чтение работает в блочном режиме
RD_PER_DELAY	10-8	Количество циклов ожидания для периферийного запроса на чтение для обновления периферийного устройства, после сброса сигнала чтения. Определяется задержкой периферийного устройства
RD_PER_NUM	4-0	Номер аппаратного источника запроса чтения (RX). Установите 0, если канал считывает данные из памяти или считывает данные с периферийного устройства, которое не использует периферийное управление запросом
–	30-27, 23-22, 14-11, 7-6	Зарезервировано

Примечание – Эти параметры не следует изменять в процессе работы канала.

RESTRICT – регистр статуса ограничений канала

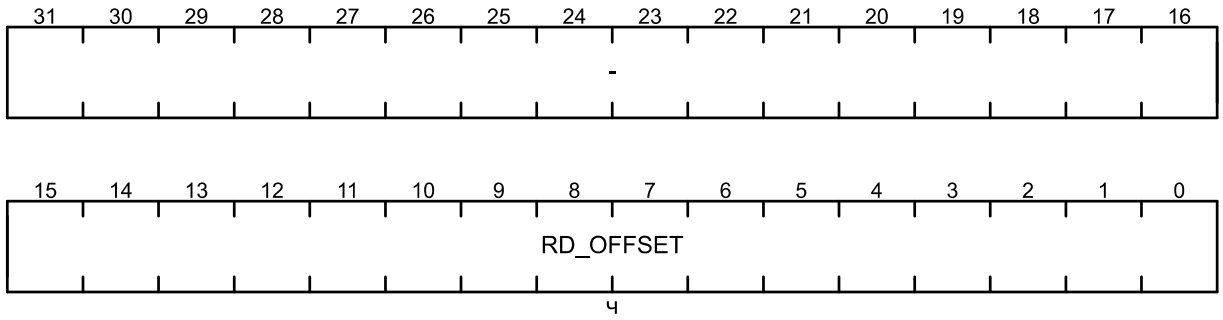
Смещение: CHn + 24h
Сброс: 0h



Поле	Биты	Описание
SIMPLE_STAT	8	Конфигурация выровнена, и периферийные устройства не используются
NONALIGN	7	Конфигурация блока не выровнена ни по начальному адресу, ни по x_size, ни по ширине кадра
WR_OUTSTAND	6	Незавершенная операция записи активна в данный момент
RD_OUTSTAND	5	Незавершенная операция чтения активна в данный момент
ALLOW_JBURST	4	Совместные пакеты активны в данный момент
ALLOW_FBURST	3	Разрешение на использование пакетов максимальной длины по 16 доступов в транзакции по шине AXI (только в совместном режиме)
ALLOW_FIFO	2	Размеры пакетов могут соответствовать размеру буфера данных, в противном случае максимальная длина пакета составит половину размера буфера данных
WR_ALLOWFIFO	1	Начальный адрес записи не ограничивает размер пакета
RD_ALLOWFIFO	0	Начальный адрес чтения не ограничивает размер пакета
–	31-9	Зарезервировано

RDOFFSET – регистр статуса смещения считывания канала

Смещение: CHn + 28h
Сброс: 0h

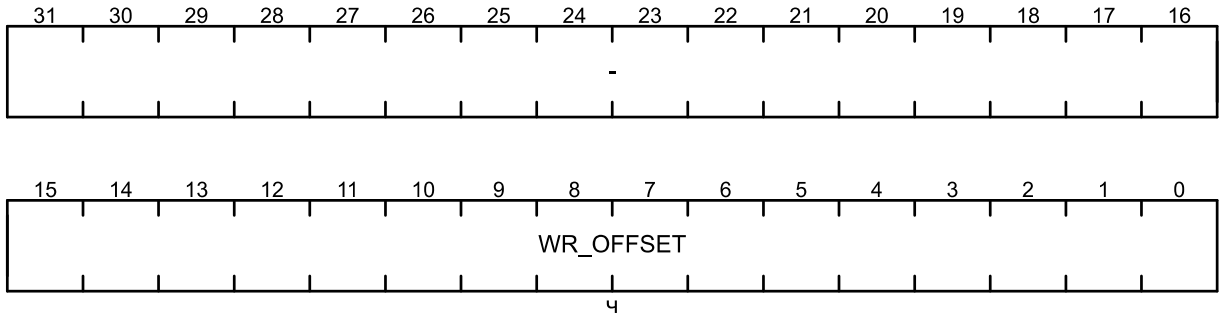


Поле	Биты	Описание
RD_OFFSET	15-0	Смещение от начала буфера. Значение указано в байтах
–	31-16	Зарезервировано

Примечание – Данный регистр доступен только для чтения.

WROFFSET – регистр статуса смещения записи канала

Смещение: CHn + 2Ch
Сброс: 0h

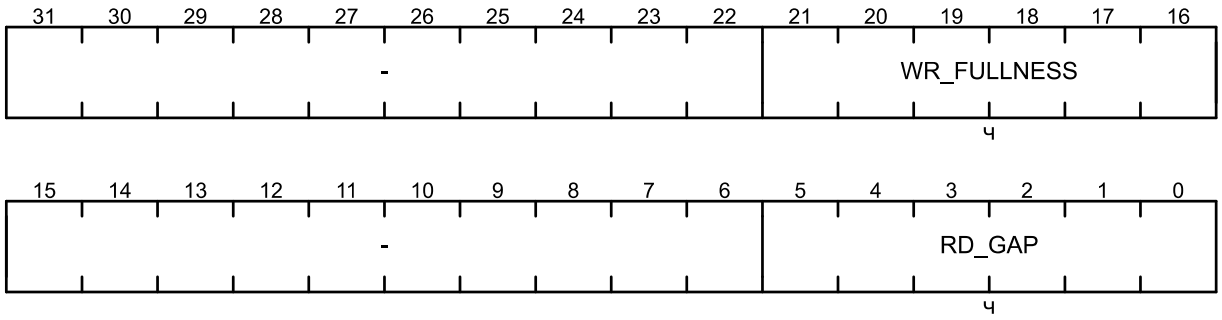


Поле	Биты	Описание
WR_OFFSET	15-0	Смещение от начала буфера. Значение указано в байтах
–	31-16	Зарезервировано

Примечание – Данный регистр доступен только для чтения.

FIFOSTAT – регистр статуса заполненности FIFO

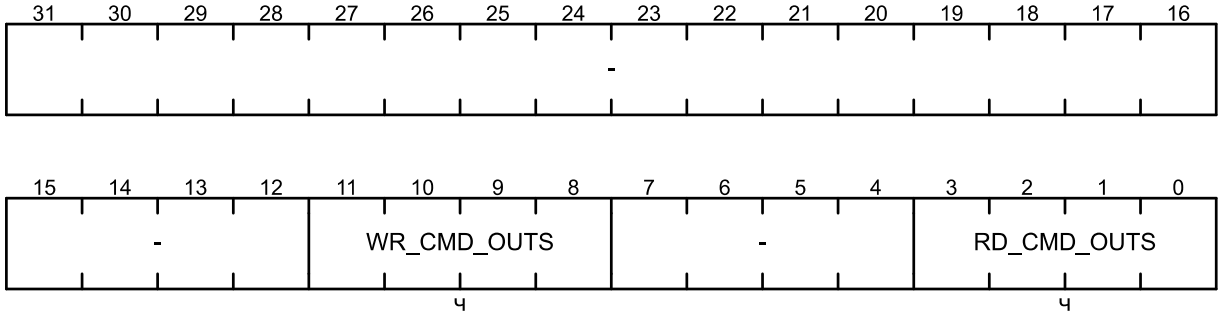
Смещение: CHn + 30h
Сброс: 0h



Поле	Биты	Описание
WR_FULLNESS	21-16	Уровень занятого пространства буфера FIFO канала при записи данных. Значение указано в байтах. Значение уменьшается при выдаче команды записи и увеличивается в соответствии со значением бита WR_OUTSTCFG регистра STATIC1. Если установлен бит WR_OUTSTCFG, то значение увеличивается после выдачи команды чтения, в противном случае оно увеличивается после считывания данных
RD_GAP	5-0	Оставшееся пространство в FIFO канала для считываемых данных. Значение указано в байтах. Значение уменьшается при выполнении команды чтения и увеличивается в соответствии со значением бита RD_OUTSTCFG регистра STATIC0. Если установлен бит RD_OUTSTCFG, то значение увеличивается после выдачи команды записи, в противном случае оно увеличивается после записи данных. Значение по сбросу равно размеру буфера FIFO
–	31-22, 15-6	Зарезервировано

CMD_OUTS – регистр статуса незавершенных доступов

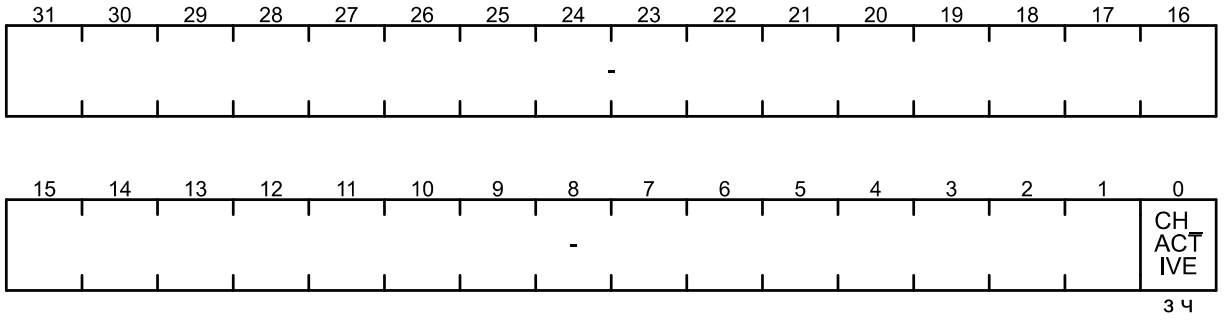
Смещение: CHn + 34h
Сброс: 0000_0000h



Поле	Биты	Описание
WR_CMD_OUTS	11-8	Номера каналов, ожидающих выполнения доступов записи по шине AXI
RD_CMD_OUTS	3-0	Номера каналов, ожидающих выполнения доступов чтения по шине AXI
–	31-12, 7-4	Зарезервировано

CH_ACTIVE – регистр включения канала

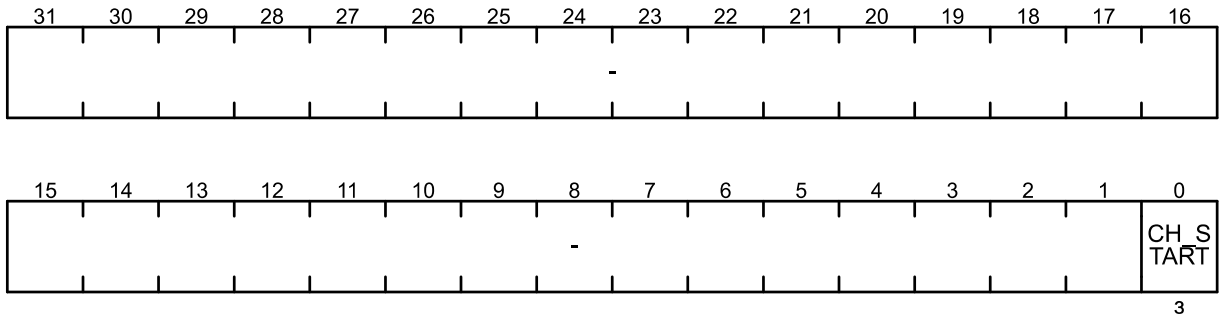
Смещение: CHn + 40h
Сброс: 0000_0001h



Поле	Биты	Описание
CH_ACTIVE	0	Бит включения канала. Часть последовательности инициализации. Также используется для приостановки и возобновления работы. Значение по умолчанию – 1
–	31-1	Зарезервировано

CH_START – регистр запуска канала

Смещение: CHn + 44h
Сброс: 0h

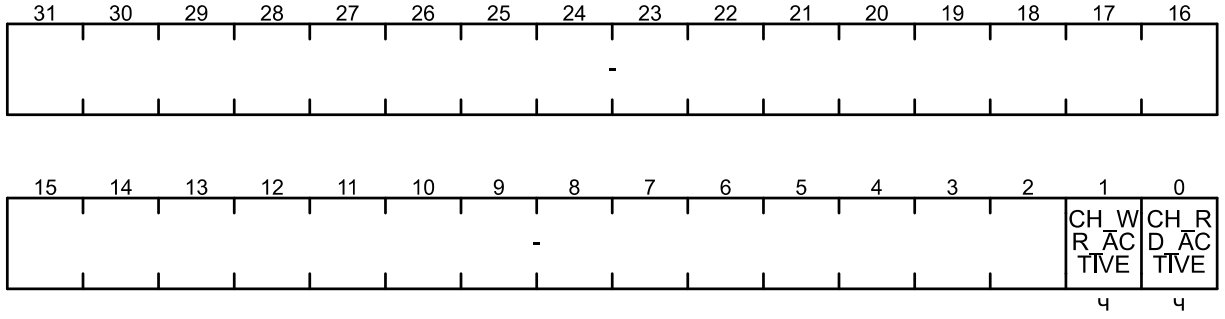


Поле	Биты	Описание
CH_START	0	Бит запуска канала. Часть последовательности инициализации
–	31-1	Зарезервировано

Примечание – Данный регистр является частью последовательности инициализации.

CH_STATUS – регистр статуса активности канала

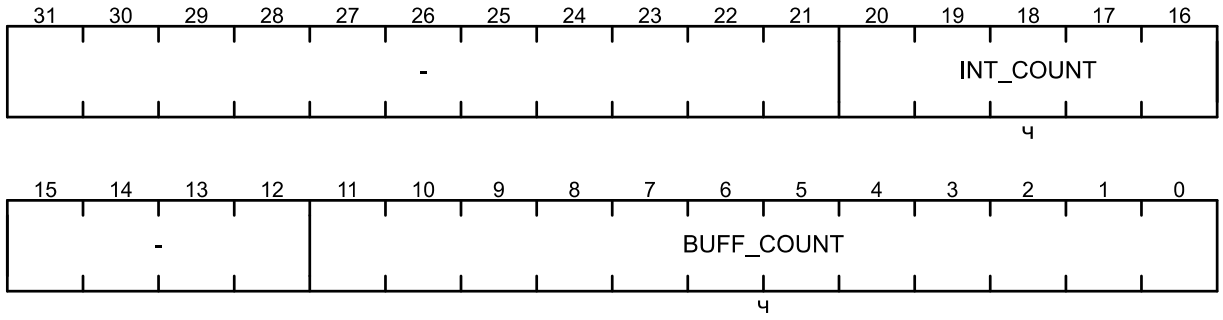
Смещение: CHn + 48h
Сброс: 0h



Поле	Биты	Описание
CH_WR_ACTIVE	1	Этот бит устанавливается, если канал включен и данные для записи в процессе передачи
CH_RD_ACTIVE	0	Этот бит устанавливается, если канал включен и данные для чтения в процессе получения
–	31-2	Зарезервировано

COUNT – регистр статуса счетчика буфера

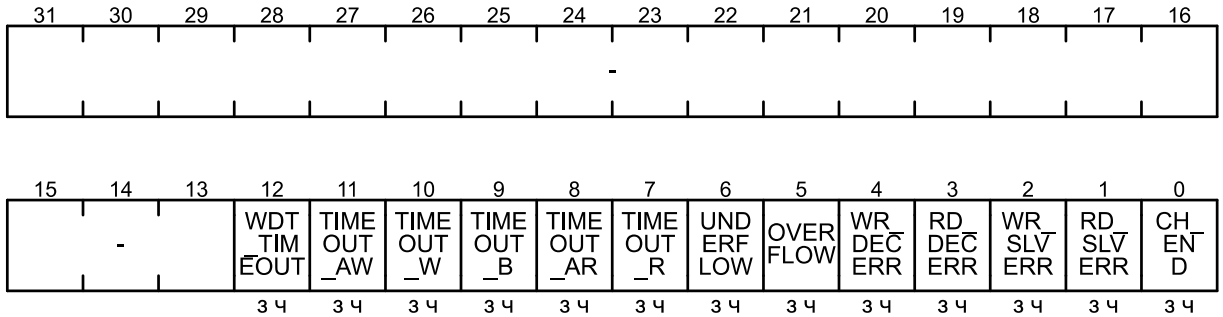
Смещение: CHn + 4Ch
Сброс: 0h



Поле	Биты	Описание
INT_COUNT	20-16	Количество необслуженных прерываний завершения работы канала. Значение увеличивается каждый раз, когда выдается прерывание завершения работы канала, и уменьшается при записи в поле CH_END регистра INT_CLEAR
BUFF_COUNT	11-0	Количество буферов, переданных по каналу с момента запуска. При использовании цепочки дескрипторов этот статус указывает, сколько дескрипторов DMA было выполнено
–	31-21, 15-12	Зарезервировано

INT_RAWSTAT – регистр статуса необработанных прерываний

Смещение: CHn + 50h
Сброс: 0h

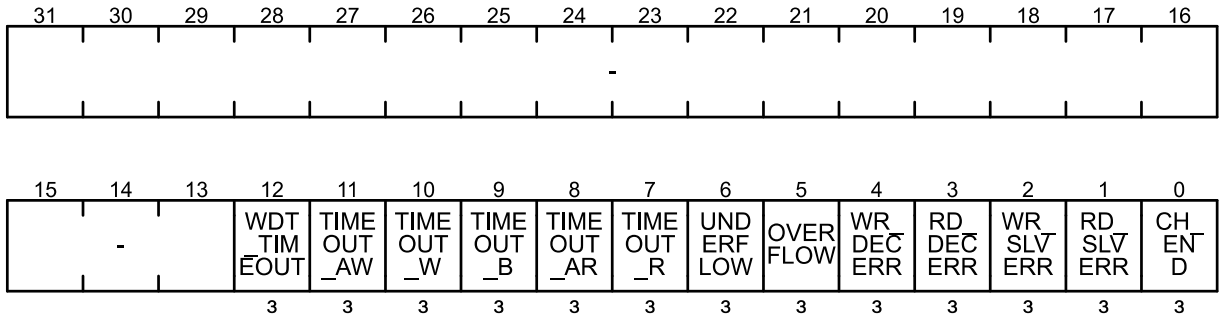


Поле	Биты	Описание
WDT_TIMEOUT	12	Бит указывает, что канал активен, но не запускал пакет в течение 2048 циклов. Прерывание может быть вызвано путем записи данного бита
TIMEOUT_AW	11	Указывает, что запись, выполненная по этому каналу, вызвала тайм-аут на командной шине записи AXI. Значение тайм-аута установлено равным 1024 циклам. Прерывание может быть вызвано путем записи данного бита

Поле	Биты	Описание
TIMEOUT_W	10	Указывает, что запись, выполненная по этому каналу, вызвала тайм-аут на шине данных записи AXI. Значение тайм-аута установлено равным 1024 циклам. Прерывание может быть вызвано путем записи данного бита
TIMEOUT_B	9	Указывает, что запись, выполненная по этому каналу, вызвала тайм-аут на шине отклика на запись AXI. Значение тайм-аута установлено равным 1024 циклам. Прерывание может быть вызвано путем записи данного бита
TIMEOUT_AR	8	Указывает, что чтение, выполненное по этому каналу, вызвало тайм-аут на командной шине чтения AXI. Значение тайм-аута установлено равным 1024 циклам. Прерывание может быть вызвано путем записи данного бита
TIMEOUT_R	7	Указывает, что чтение, выполненное по этому каналу, вызвало тайм-аут на шине данных чтения AXI. Значение тайм-аута установлено равным 1024 циклам. Прерывание может быть вызвано путем записи данного бита
UNDERFLOW	6	Указывает на недостаточный объем данных буфера FIFO, это может произойти только тогда, когда установлен бит WR_OUTSTAND_CFG регистра STATIC1, что указывает на то, что ведомое устройство было недостаточно быстрым для работы в этом режиме. Прерывание может быть вызвано путем записи данного бита
OVERFLOW	5	Указывает, что буфер FIFO был переполнен, это может произойти только тогда, когда установлен бит RD_OUTSTAND_CFG регистра STATIC0, что указывает на то, что ведомое устройство было недостаточно быстрым для работы в этом режиме. Прерывание может быть вызвано путем записи данного бита
WR_DECERR	4	Указывает, что запись, выполненная по этому каналу, вызвала ошибку записи декодирования AXI. Прерывание может быть вызвано путем записи данного бита
RD_DECERR	3	Указывает, что чтение, выполненное по этому каналу, вызвало ошибку декодирования чтения AXI. Прерывание может быть вызвано путем записи данного бита
WR_SLVERR	2	Указывает, что запись, выполненная по этому каналу, вызвала ошибку записи ведомого AXI. Прерывание может быть вызвано путем записи данного бита
RD_SLVERR	1	Указывает, что чтение, выполненное по этому каналу, вызвало ошибку чтения ведомого AXI. Прерывание может быть вызвано путем записи данного бита
CH_END	0	Указывает на необслуженное прерывание завершения работы канала. Общее количество необслуженных прерываний завершения работы канала можно прочитать в битовом поле INT_COUNT регистра COUNT. Прерывание может быть вызвано путем записи данного бита
–	31-13	Зарезервировано

INT_CLEAR – регистр очистки прерываний

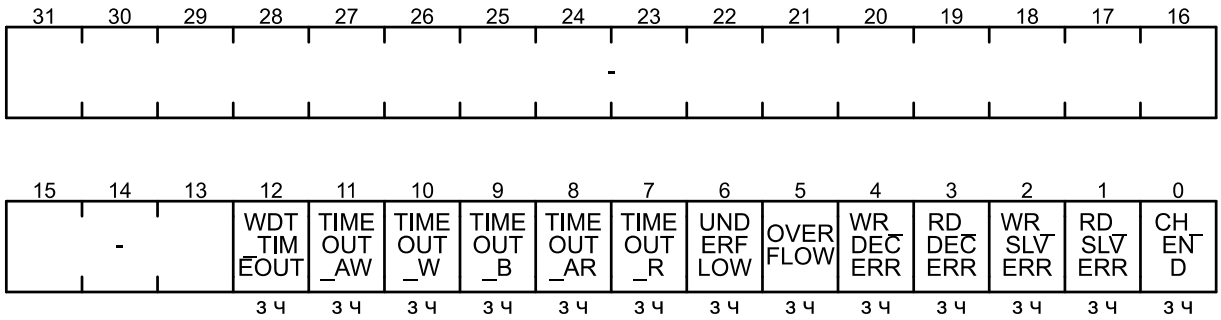
Смещение: CHn + 54h
Сброс: 0h



Поле	Биты	Описание
WDT_TIMEOUT	12	Бит очистки флага WDT_TIMEOUT регистра INT_RAWSTAT
TIMEOUT_AW	11	Бит очистки флага прерывания TIMEOUT_AW регистра INT_RAWSTAT
TIMEOUT_W	10	Бит очистки флага прерывания TIMEOUT_W регистра INT_RAWSTAT
TIMEOUT_B	9	Бит очистки флага прерывания TIMEOUT_B регистра INT_RAWSTAT
TIMEOUT_AR	8	Бит очистки флага прерывания TIMEOUT_AR регистра INT_RAWSTAT
TIMEOUT_R	7	Бит очистки флага прерывания TIMEOUT_R регистра INT_RAWSTAT
UNDERFLOW	6	Бит очистки флага прерывания UNDERFLOW регистра INT_RAWSTAT
OVERFLOW	5	Бит очистки флага прерывания OVERFLOW регистра INT_RAWSTAT
WR_DECERR	4	Бит очистки флага прерывания WR_DECERR регистра INT_RAWSTAT
RD_DECERR	3	Бит очистки флага прерывания RD_DECERR регистра INT_RAWSTAT
WR_SLVERR	2	Бит очистки флага прерывания WR_SLVERR регистра INT_RAWSTAT
RD_SLVERR	1	Бит очистки флага прерывания RD_SLVERR регистра INT_RAWSTAT
CH_END	0	Бит очистки прерывания завершения работы канала. Декрементирует значение в поле INT_COUNT регистра COUNT
–	31-13	Зарезервировано

INT_ENABLE – регистр разрешения прерываний

Смещение: CHn + 58h
Сброс: 0000_1FFFh

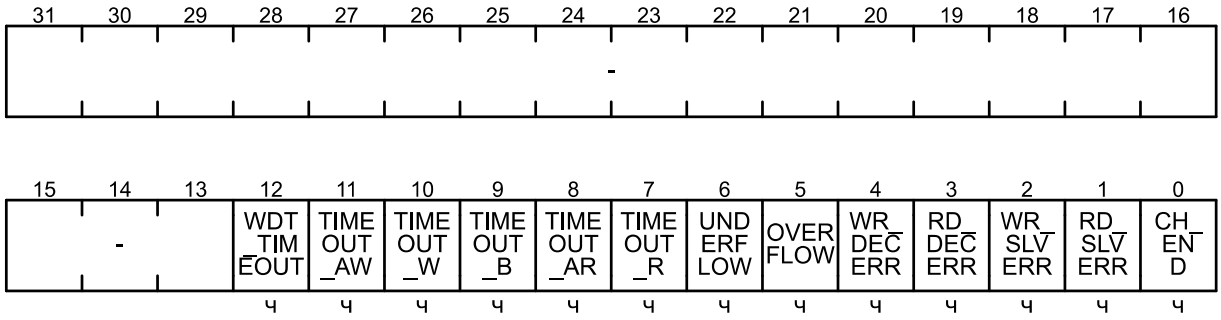


Поле	Биты	Описание
WDT_TIMEOUT	12	Разрешает прерывание WDT_TIMEOUT
TIMEOUT_AW	11	Разрешает прерывание TIMEOUT_AW
TIMEOUT_W	10	Разрешает прерывание TIMEOUT_W
TIMEOUT_B	9	Разрешает прерывание TIMEOUT_B
TIMEOUT_AR	8	Разрешает прерывание TIMEOUT_AR
TIMEOUT_R	7	Разрешает прерывание TIMEOUT_R
UNDERFLOW	6	Разрешает прерывание UNDERFLOW
OVERFLOW	5	Разрешает прерывание OVERFLOW
WR_DECERR	4	Разрешает прерывание WR_DECERR
RD_DECERR	3	Разрешает прерывание RD_DECERR
WR_SLVERR	2	Разрешает прерывание WR_SLVERR
RD_SLVERR	1	Разрешает прерывание RD_SLVERR
CH_END	0	Разрешает прерывание CH_END
–	31-13	Зарезервировано

Примечание – Каждый установленный бит позволяет соответствующему биту регистра INT_RAWSTAT отображаться в регистре INT_STATUS.

INT_STATUS – регистр статуса прерываний

Смещение: CHn + 5Ch
Сброс: 0h

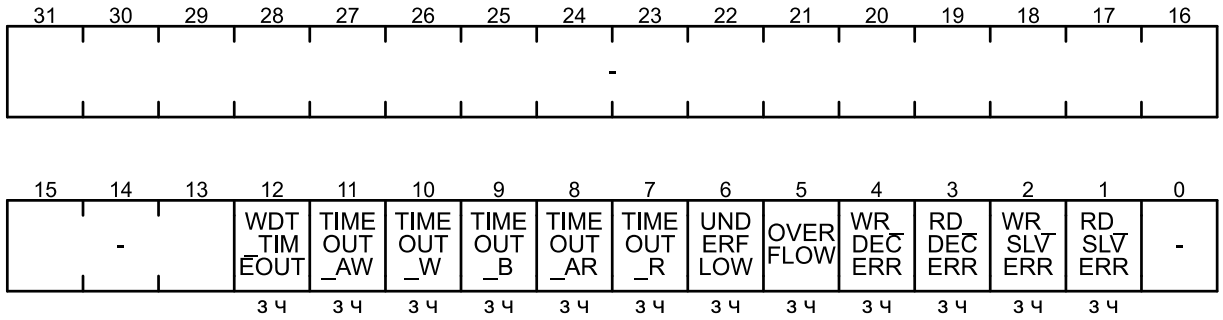


Поле	Биты	Описание
WDT_TIMEOUT	12	Флаг WDT_TIMEOUT регистра INT_RAWSTAT установлен, прерывание разрешено
TIMEOUT_AW	11	Флаг TIMEOUT_AW регистра INT_RAWSTAT установлен, прерывание разрешено
TIMEOUT_W	10	Флаг TIMEOUT_W регистра INT_RAWSTAT установлен, прерывание разрешено
TIMEOUT_B	9	Флаг TIMEOUT_B регистра INT_RAWSTAT установлен, прерывание разрешено
TIMEOUT_AR	8	Флаг TIMEOUT_AR регистра INT_RAWSTAT установлен, прерывание разрешено
TIMEOUT_R	7	Флаг TIMEOUT_R регистра INT_RAWSTAT установлен, прерывание разрешено
UNDERFLOW	6	Флаг UNDERFLOW регистра INT_RAWSTAT установлен, прерывание разрешено
OVERFLOW	5	Флаг OVERFLOW регистра INT_RAWSTAT установлен, прерывание разрешено
WR_DECERR	4	Флаг WR_DECERR регистра INT_RAWSTAT установлен, прерывание разрешено
RD_DECERR	3	Флаг RD_DECERR регистра INT_RAWSTAT установлен, прерывание разрешено
WR_SLVERR	2	Флаг WR_SLVERR регистра INT_RAWSTAT установлен, прерывание разрешено
RD_SLVERR	1	Флаг RD_SLVERR регистра INT_RAWSTAT установлен, прерывание разрешено
CH_END	0	Флаг CH_END регистра INT_RAWSTAT установлен, прерывание разрешено
–	31-13	Зарезервировано

Примечание – Данный регистр указывает, какие прерывания в данный момент выводятся.

ERR_STOP – регистр статуса остановки по ошибке канала

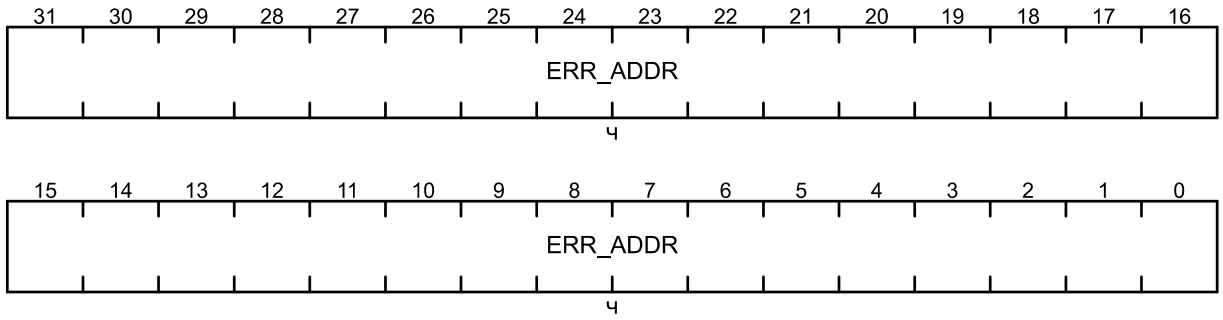
Смещение: CHn + 60h
Сброс: 0h



Поле	Биты	Описание
WDT_TIMEOUT	12	Остановка вызвана тем, что канал активен, но не запускал пакет в течение 2048 циклов
TIMEOUT_AW	11	Остановка вызвана тем, что запись, выполненная по этому каналу, вызвала тайм-аут на командной шине записи AXI
TIMEOUT_W	10	Остановка вызвана тем, что запись, выполненная по этому каналу, вызвала тайм-аут на шине данных записи AXI
TIMEOUT_B	9	Остановка вызвана тем, что запись, выполненная по этому каналу, вызвала тайм-аут на шине отклика на запись AXI
TIMEOUT_AR	8	Остановка вызвана тем, что чтение, выполненное по этому каналу, вызвало тайм-аут на командной шине чтения AXI
TIMEOUT_R	7	Остановка вызвана тем, что чтение, выполненное по этому каналу, вызвало тайм-аут на шине данных чтения AXI
UNDERFLOW	6	Остановка вызвана недостаточным объемом данных буфера FIFO, это может произойти только тогда, когда установлен бит WR_OUTSTCFG регистра STATIC1, что указывает на то, что ведомое устройство было недостаточно быстрым для работы в этом режиме
OVERFLOW	5	Остановка вызвана тем, что буфер FIFO был переполнен, это может произойти только тогда, когда установлен бит RD_OUTSTCFG регистра STATIC0, что указывает на то, что ведомое устройство было недостаточно быстрым для работы в этом режиме
WR_DECERR	4	Остановка вызвана тем, что запись, выполненная по этому каналу, вызвала ошибку записи декодирования AXI
RD_DECERR	3	Остановка вызвана тем, что чтение, выполненное по этому каналу, вызвало ошибку декодирования чтения AXI
WR_SLVERR	2	Остановка вызвана тем, что запись, выполненная по этому каналу, вызвала ошибку записи ведомого AXI
RD_SLVERR	1	Остановка вызвана тем, что чтение, выполненное по этому каналу, вызвало ошибку чтения ведомого AXI
–	31-13, 0	Зарезервировано

ERR_RD_ADDR – регистр адреса ошибочной транзакции чтения

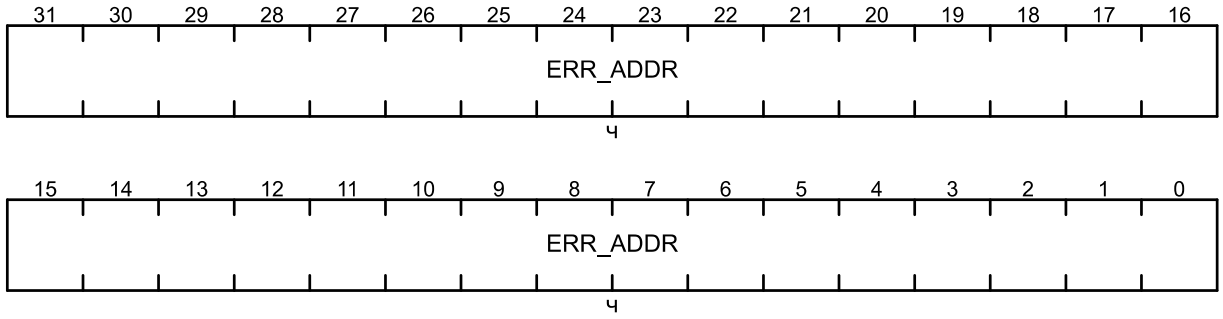
Смещение: CHn + 64h
Сброс: 0h



Поле	Биты	Описание
ERR_ADDR	31-0	Первый обнаруженный ошибочный адрес в транзакции чтения

ERR_WR_ADDR – регистр адреса ошибочной транзакции записи

Смещение: CHn + 68h
Сброс: 0h

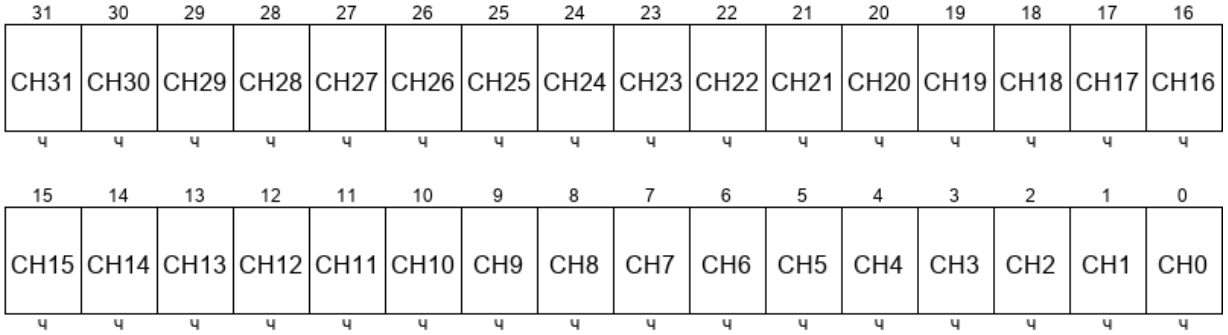


Поле	Биты	Описание
ERR_ADDR	31-0	Адрес первой транзакции записи, в которой была обнаружена ошибка

A.8.2 Общие регистры DMA

CH_INTSTAT – регистр статуса прерываний ядра DMA

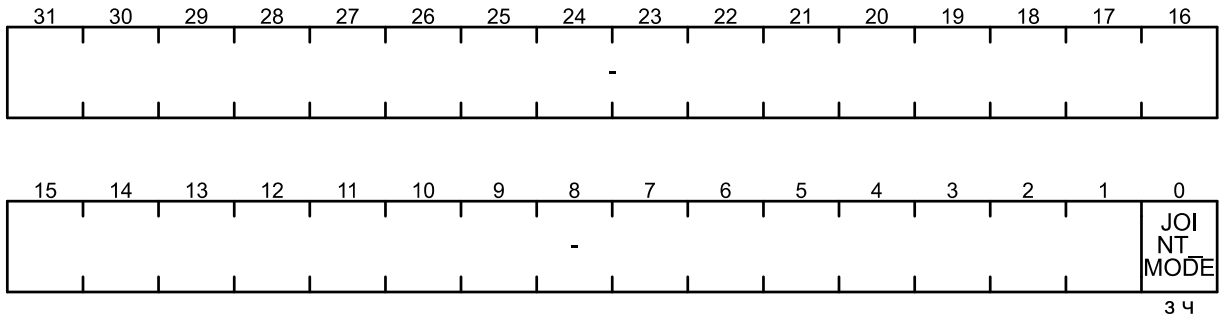
Смещение: + 1000h
Сброс: 0h



Поле	Биты	Описание
CH31	31	Прерывание вызвано каналом 31
CH30	30	Прерывание вызвано каналом 30
CH29	29	Прерывание вызвано каналом 29
CH28	28	Прерывание вызвано каналом 28
CH27	27	Прерывание вызвано каналом 27
CH26	26	Прерывание вызвано каналом 26
CH25	25	Прерывание вызвано каналом 25
CH24	24	Прерывание вызвано каналом 24
CH23	23	Прерывание вызвано каналом 23
CH22	22	Прерывание вызвано каналом 22
CH21	21	Прерывание вызвано каналом 21
CH20	20	Прерывание вызвано каналом 20
CH19	19	Прерывание вызвано каналом 19
CH18	18	Прерывание вызвано каналом 18
CH17	17	Прерывание вызвано каналом 17
CH16	16	Прерывание вызвано каналом 16
CH15	15	Прерывание вызвано каналом 15
CH14	14	Прерывание вызвано каналом 14
CH13	13	Прерывание вызвано каналом 13
CH12	12	Прерывание вызвано каналом 12
CH11	11	Прерывание вызвано каналом 11
CH10	10	Прерывание вызвано каналом 10
CH9	9	Прерывание вызвано каналом 9
CH8	8	Прерывание вызвано каналом 8
CH7	7	Прерывание вызвано каналом 7
CH6	6	Прерывание вызвано каналом 6
CH5	5	Прерывание вызвано каналом 5
CH4	4	Прерывание вызвано каналом 4
CH3	3	Прерывание вызвано каналом 3
CH2	2	Прерывание вызвано каналом 2
CH1	1	Прерывание вызвано каналом 1
CH0	0	Прерывание вызвано каналом 0

CONTROL – регистр управления ядром DMA

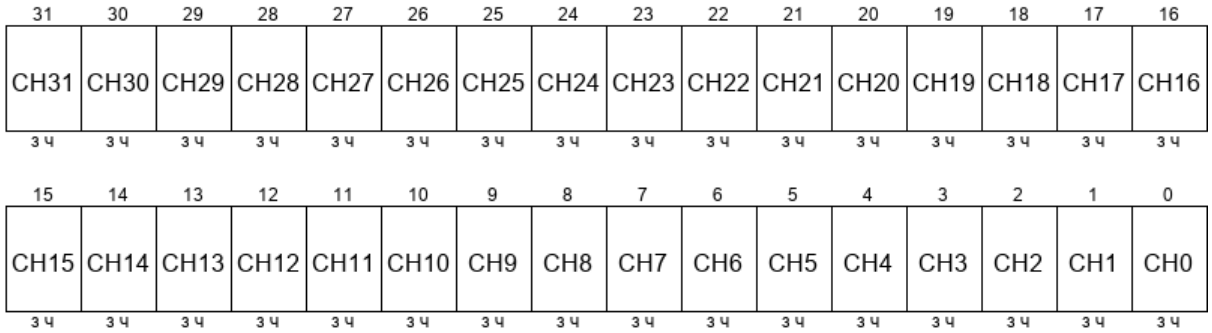
Смещение: + 1004h
Сброс: 0h



Поле	Биты	Описание	
JOINT_MODE	0	Бит включения совместного режима	
		0	Независимый режим
		1	Совместный режим
–	31-1	Зарезервировано	

RD_PRIORITY – регистр приоритетов каналов ядра DMA при чтении

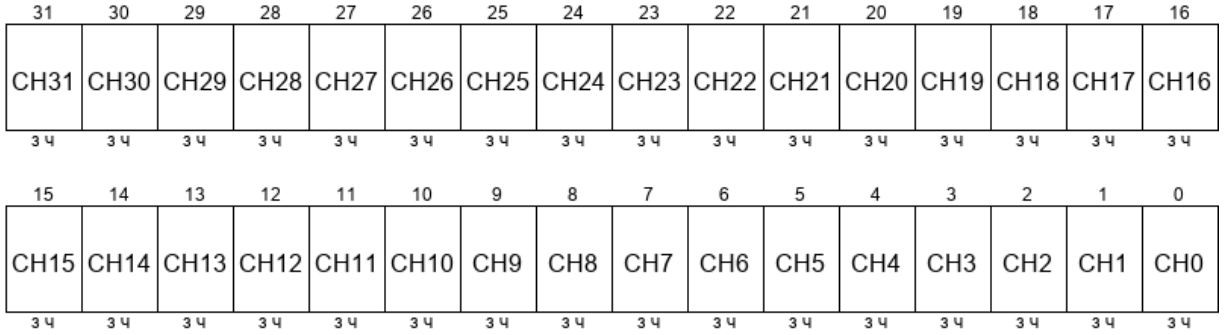
Смещение: + 1008h
Сброс: 0h



Поле	Биты	Описание
CH31	31	Установка приоритета для канала 31
CH30	30	Установка приоритета для канала 30
CH29	29	Установка приоритета для канала 29
CH28	28	Установка приоритета для канала 28
CH27	27	Установка приоритета для канала 27
CH26	26	Установка приоритета для канала 26
CH25	25	Установка приоритета для канала 25
CH24	24	Установка приоритета для канала 24
CH23	23	Установка приоритета для канала 23
CH22	22	Установка приоритета для канала 22
CH21	21	Установка приоритета для канала 21
CH20	20	Установка приоритета для канала 20
CH19	19	Установка приоритета для канала 19
CH18	18	Установка приоритета для канала 18
CH17	17	Установка приоритета для канала 17
CH16	16	Установка приоритета для канала 16
CH15	15	Установка приоритета для канала 15
CH14	14	Установка приоритета для канала 14
CH13	13	Установка приоритета для канала 13
CH12	12	Установка приоритета для канала 12
CH11	11	Установка приоритета для канала 11
CH10	10	Установка приоритета для канала 10
CH9	9	Установка приоритета для канала 9
CH8	8	Установка приоритета для канала 8
CH7	7	Установка приоритета для канала 7
CH6	6	Установка приоритета для канала 6
CH5	5	Установка приоритета для канала 5
CH4	4	Установка приоритета для канала 4
CH3	3	Установка приоритета для канала 3
CH2	2	Установка приоритета для канала 2
CH1	1	Установка приоритета для канала 1
CH0	0	Установка приоритета для канала 0

WR_PRIORITY – регистр приоритетов каналов ядра DMA при записи

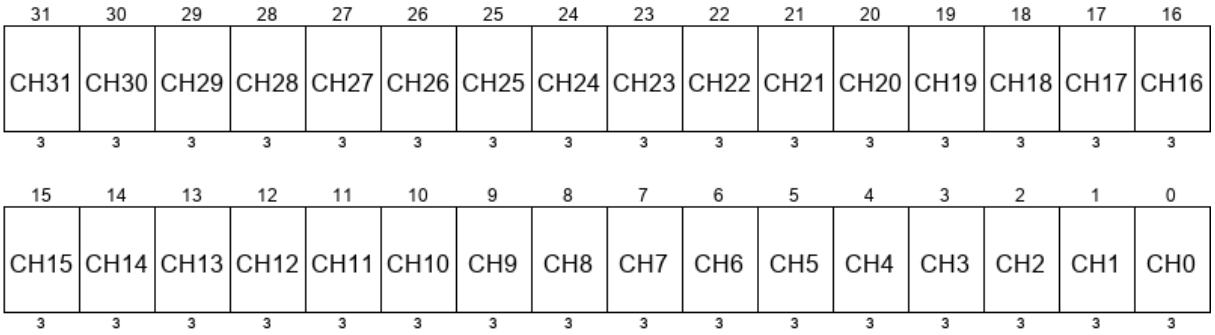
Смещение: + 100Ch
Сброс: 0h



Поле	Биты	Описание
CH31	31	Установка приоритета для канала 31
CH30	30	Установка приоритета для канала 30
CH29	29	Установка приоритета для канала 29
CH28	28	Установка приоритета для канала 28
CH27	27	Установка приоритета для канала 27
CH26	26	Установка приоритета для канала 26
CH25	25	Установка приоритета для канала 25
CH24	24	Установка приоритета для канала 24
CH23	23	Установка приоритета для канала 23
CH22	22	Установка приоритета для канала 22
CH21	21	Установка приоритета для канала 21
CH20	20	Установка приоритета для канала 20
CH19	19	Установка приоритета для канала 19
CH18	18	Установка приоритета для канала 18
CH17	17	Установка приоритета для канала 17
CH16	16	Установка приоритета для канала 16
CH15	15	Установка приоритета для канала 15
CH14	14	Установка приоритета для канала 14
CH13	13	Установка приоритета для канала 13
CH12	12	Установка приоритета для канала 12
CH11	11	Установка приоритета для канала 11
CH10	10	Установка приоритета для канала 10
CH9	9	Установка приоритета для канала 9
CH8	8	Установка приоритета для канала 8
CH7	7	Установка приоритета для канала 7
CH6	6	Установка приоритета для канала 6
CH5	5	Установка приоритета для канала 5
CH4	4	Установка приоритета для канала 4
CH3	3	Установка приоритета для канала 3
CH2	2	Установка приоритета для канала 2
CH1	1	Установка приоритета для канала 1
CH0	0	Установка приоритета для канала 0

CH_START – регистр запуска каналов DMA

Смещение: + 1010h
Сброс: 0h

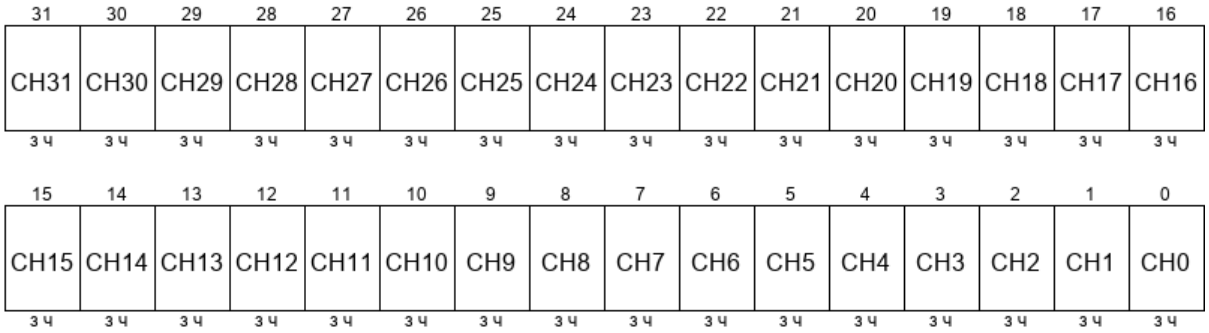


Поле	Биты	Описание
CH31	31	Запуск канала 31
CH30	30	Запуск канала 30
CH29	29	Запуск канала 29
CH28	28	Запуск канала 28
CH27	27	Запуск канала 27
CH26	26	Запуск канала 26
CH25	25	Запуск канала 25
CH24	24	Запуск канала 24
CH23	23	Запуск канала 23
CH22	22	Запуск канала 22
CH21	21	Запуск канала 21
CH20	20	Запуск канала 20
CH19	19	Запуск канала 19
CH18	18	Запуск канала 18
CH17	17	Запуск канала 17
CH16	16	Запуск канала 16
CH15	15	Запуск канала 15
CH14	14	Запуск канала 14
CH13	13	Запуск канала 13
CH12	12	Запуск канала 12
CH11	11	Запуск канала 11
CH10	10	Запуск канала 10
CH9	9	Запуск канала 9
CH8	8	Запуск канала 8
CH7	7	Запуск канала 7
CH6	6	Запуск канала 6
CH5	5	Запуск канала 5
CH4	4	Запуск канала 4
CH3	3	Запуск канала 3
CH2	2	Запуск канала 2
CH1	1	Запуск канала 1
CH0	0	Запуск канала 0

Примечание – Данный регистр допускает запуск нескольких каналов одновременно.

CH_ENABLE – регистр включения каналов DMA

Смещение: + 1014h
Сброс: 0h

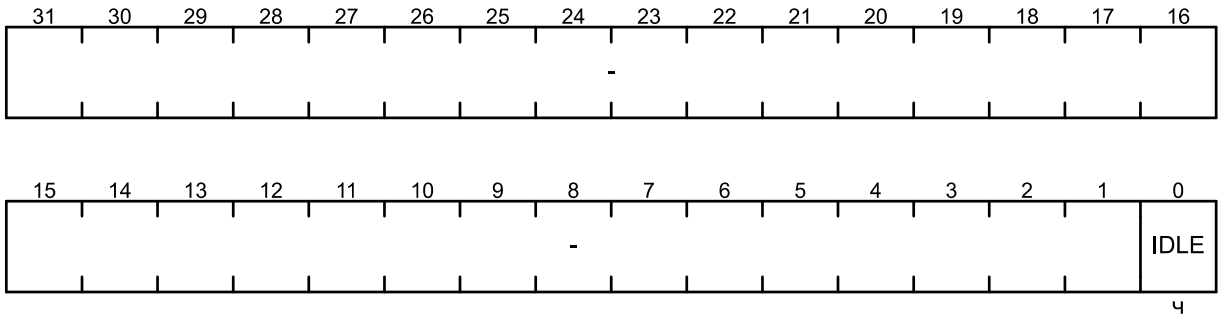


Поле	Биты	Описание
CH31	31	Включение канала 31
CH30	30	Включение канала 30
CH29	29	Включение канала 29
CH28	28	Включение канала 28
CH27	27	Включение канала 27
CH26	26	Включение канала 26
CH25	25	Включение канала 25
CH24	24	Включение канала 24
CH23	23	Включение канала 23
CH22	22	Включение канала 22
CH21	21	Включение канала 21
CH20	20	Включение канала 20
CH19	19	Включение канала 19
CH18	18	Включение канала 18
CH17	17	Включение канала 17
CH16	16	Включение канала 16
CH15	15	Включение канала 15
CH14	14	Включение канала 14
CH13	13	Включение канала 13
CH12	12	Включение канала 12
CH11	11	Включение канала 11
CH10	10	Включение канала 10
CH9	9	Включение канала 9
CH8	8	Включение канала 8
CH7	7	Включение канала 7
CH6	6	Включение канала 6
CH5	5	Включение канала 5
CH4	4	Включение канала 4
CH3	3	Включение канала 3
CH2	2	Включение канала 2
CH1	1	Включение канала 1
CH0	0	Включение канала 0

Примечание – Данный регистр позволяет включение нескольких каналов одновременно.

IDLE – регистр индикации бездействия

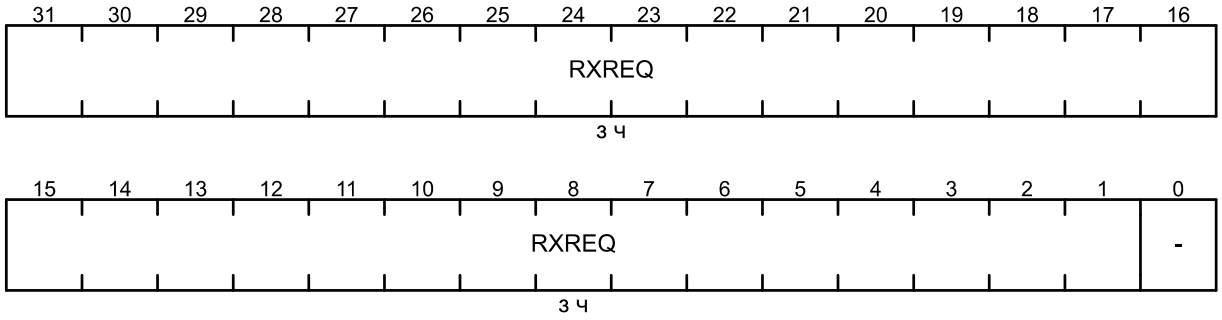
Смещение: + 1018h
Сброс: 0h



Поле	Биты	Описание
IDLE	0	Данный бит указывает, что все каналы закончили работу и все транзакции по шине завершены
–	31-1	Зарезервировано

PER_RX0_CTRL – регистр ручного управления запросом на обработку приема данных от периферии

Смещение: + 1020h
Сброс: 0h

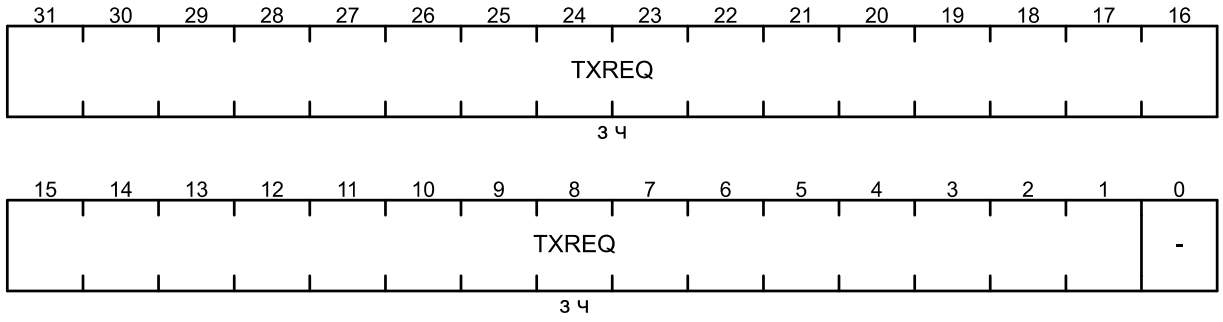


Поле	Биты	Описание
RXREQ	31-1	Позволяет непосредственно управлять периферийным запросом на чтение. Особенно полезно, когда программа хочет прочитать результаты, не дожидаясь их получения
–	0	Зарезервировано

Примечание – Бит 0 зарезервирован для неуправляемых передач.

PER_TX0_CTRL – регистр управления запросом на обработку передачи данных к периферии

Смещение: + 1024h
Сброс: 0h

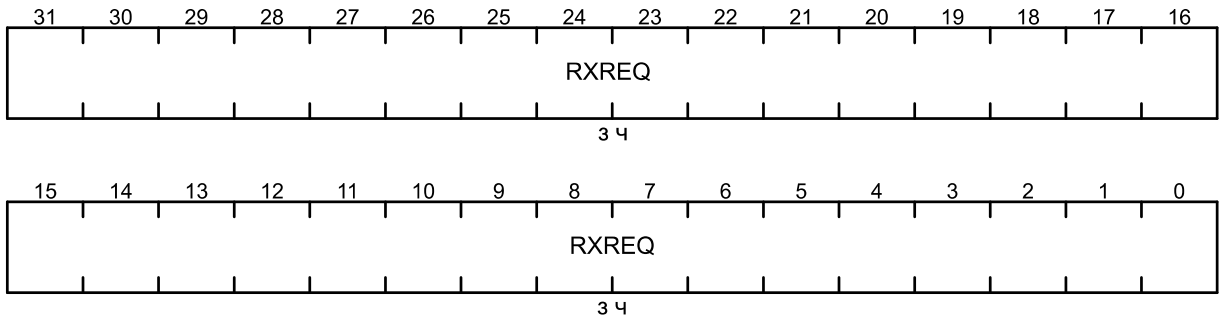


Поле	Биты	Описание
TXREQ	31-1	Позволяет непосредственно управлять периферийным запросом на запись. Особенно полезно, когда программа хочет записать регистры конфигурации в медленное устройство
–	0	Зарезервировано

Примечание – Бит 0 зарезервирован для неуправляемых передач.

PER_RX1_CTRL – регистр ручного управления запросом на обработку приема данных от периферии

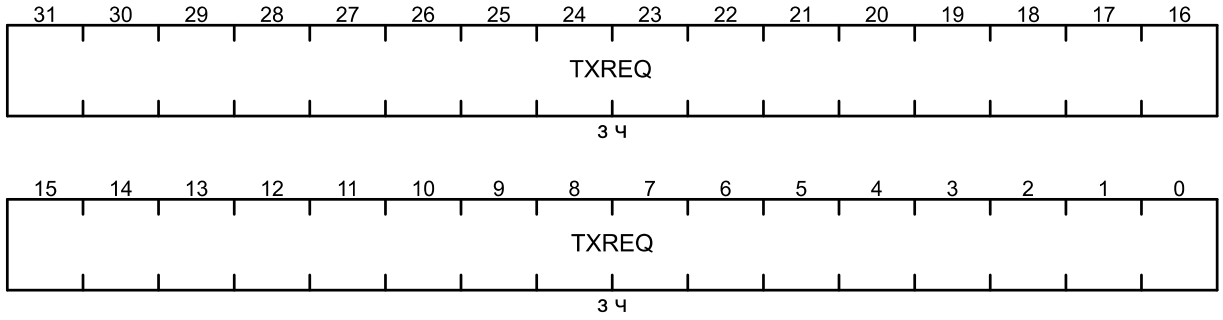
Смещение: + 1028h
Сброс: 0h



Поле	Биты	Описание
RXREQ	31-0	Позволяет непосредственно управлять периферийным запросом на чтение. Особенно полезно, когда программа хочет прочитать результаты, не дожидаясь их получения

PER_TX1_CTRL – регистр управления запросом на обработку передачи данных к периферии

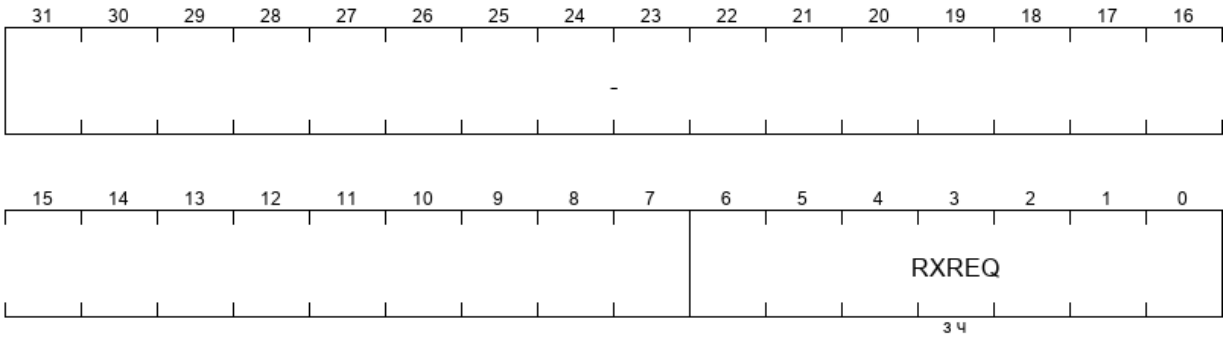
Смещение: + 102Ch
Сброс: 0h



Поле	Биты	Описание
TXREQ	31-0	Позволяет непосредственно управлять периферийным запросом на запись. Особенно полезно, когда программа хочет записать регистры конфигурации в медленное устройство

PER_RX2_CTRL – регистр ручного управления запросом на обработку приема данных от периферии

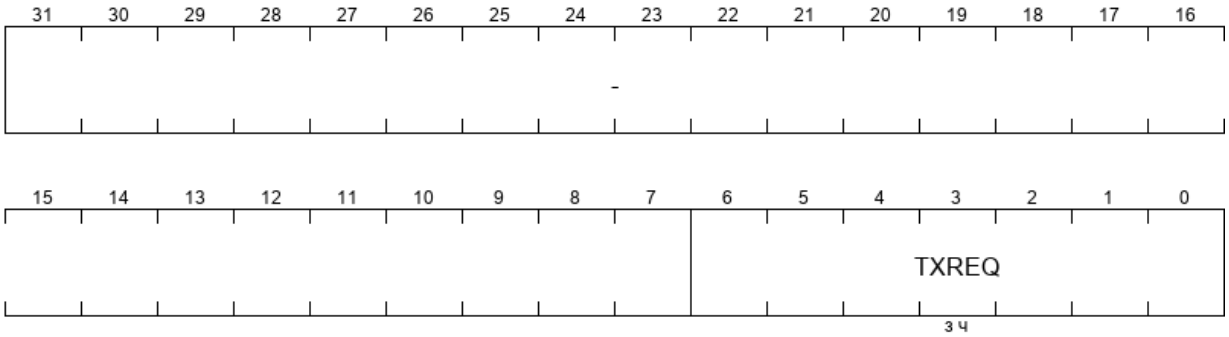
Смещение: + 1030h
Сброс: 0h



Поле	Биты	Описание
RX_REQ	6-0	Позволяет непосредственно управлять периферийным запросом на чтение. Особенно полезно, когда программа хочет прочитать результаты, не дожидаясь их получения
–	31-7	Зарезервировано

PER_TX2_CTRL – регистр управления запросом на обработку передачи данных к периферии

Смещение: + 1034h
Сброс: 0h



Поле	Биты	Описание
TX_REQ	6-0	Позволяет непосредственно управлять периферийным запросом на запись. Особенно полезно, когда программа хочет записать регистры конфигурации в медленное устройство
–	31-7	Зарезервировано

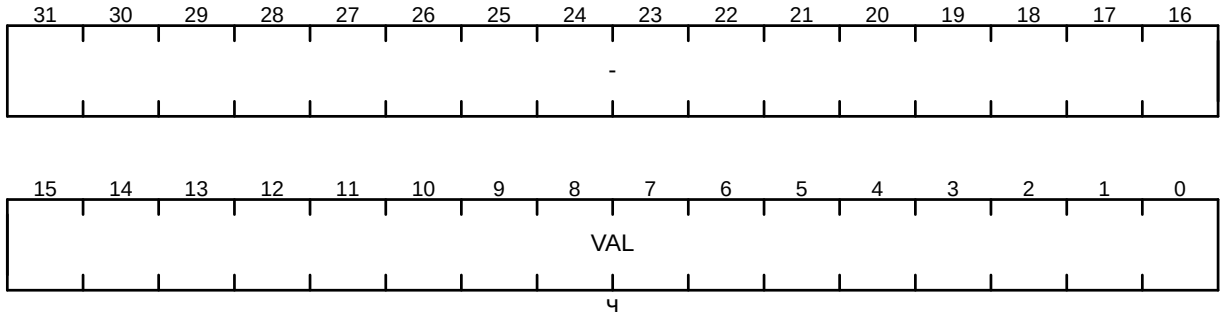
A.9 Регистры портов ввода-вывода GPIO

Базовый адрес:	4000_0000h	Регистры порта GPIOA
	4000_1000h	Регистры порта GPIOB
	4000_2000h	Регистры порта GPIOC
	4000_3000h	Регистры порта GPIOD
	4000_4000h	Регистры порта GPIOE
	4000_5000h	Регистры порта GPIOF
	4000_6000h	Регистры порта GPIOG
	4000_7000h	Регистры порта GPIOH
Смещение:	+ 400h (MASKLB)	Массив регистров масок младшего байта порта
	+ 800h (MASKHB)	Массив регистров масок старшего байта порта
Мнемоника:	GPIOp	

Примечание – p – имя порта A, B, C, D, E, F, G, H;
n – порядковый номер вывода порта от 0 до 15.

DATA – регистр входных данных порта

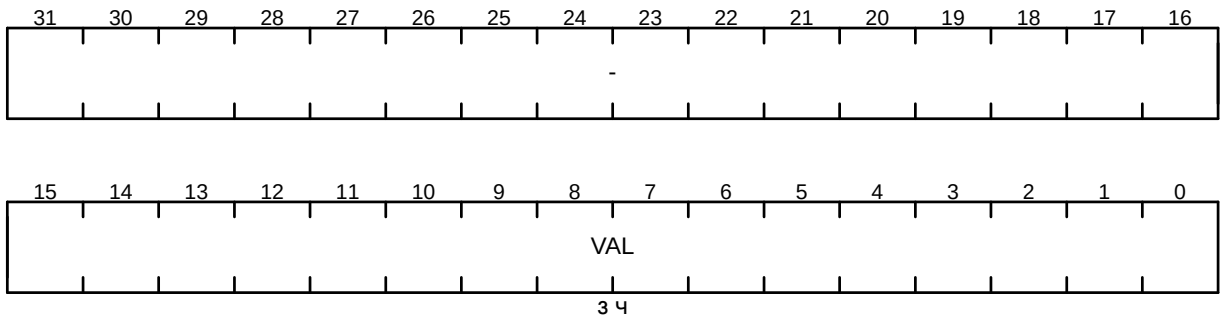
Смещение: + 00h
Сброс: 0000_xxxxh



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает текущее состояние порта
		Запись	Не выполняется
–	31-16	Зарезервировано	

DATAOUT – регистр выходных данных порта

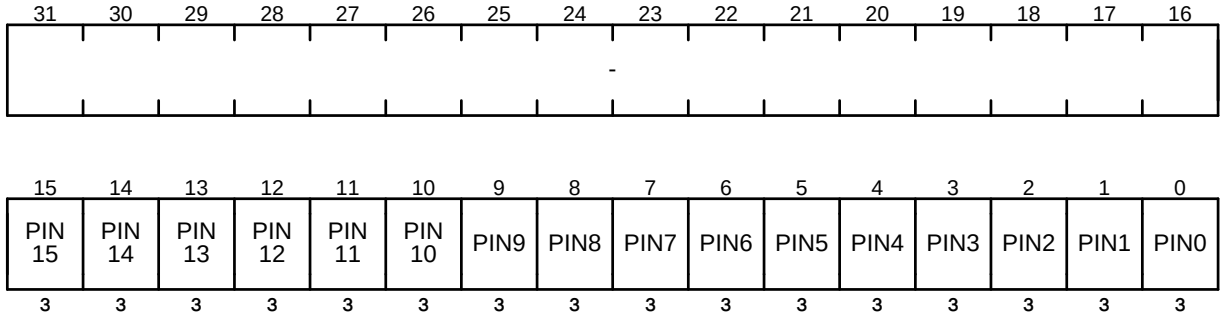
Смещение: + 04h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние порта
		Запись	Устанавливает на выводах порта уровень сигнала, соответствующий записанному в биты значению
–	31-16	Зарезервировано	

DATAOUTSET – регистр установки битов порта

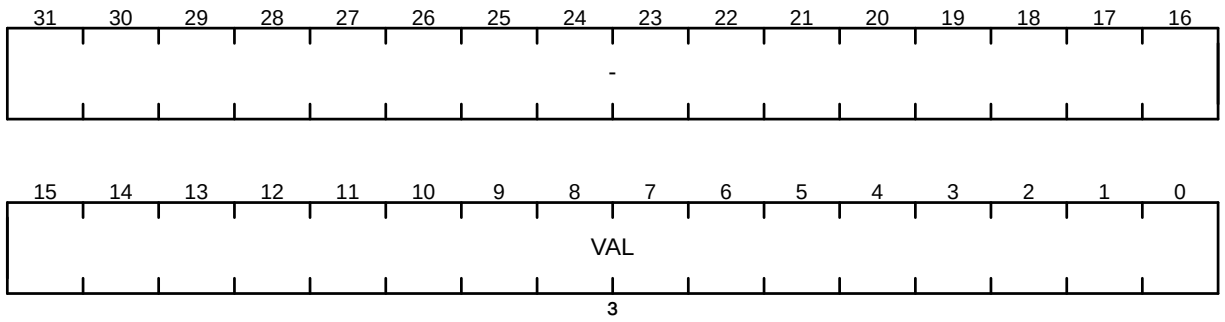
Смещение: + 08h
Сброс: 0h



Поле	Биты	Описание	
PINn	15-0	Чтение	Возвращает состояние порта (регистр DATAOUT)
		Запись нуля	Не выполняется
		Запись единицы	Устанавливает соответствующий бит n в регистре DATAOUT. Как следствие, будет наблюдаться высокий уровень сигнала на выводе n порта, если управление его состоянием было разрешено с помощью записи единицы в регистр OUTENSET и в регистре OUTMODE остался выбран активный режим выхода порта
–	31-16	Зарезервировано	

DATAOUTCLR – регистр сброса битов порта

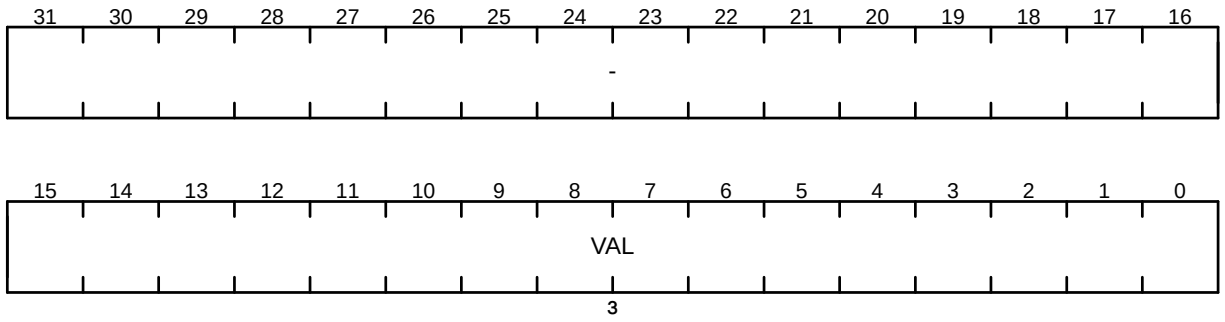
Смещение: + 0Ch
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние порта (регистр DATAOUT). Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам сбрасывает биты в регистре DATAOUT и, как следствие, устанавливает низкий уровень сигнала на выводах порта, если управление их состоянием было разрешено с помощью регистра OUTENSET
–	31-16	Зарезервировано	

DATAOUTTGL – регистр переключения битов порта

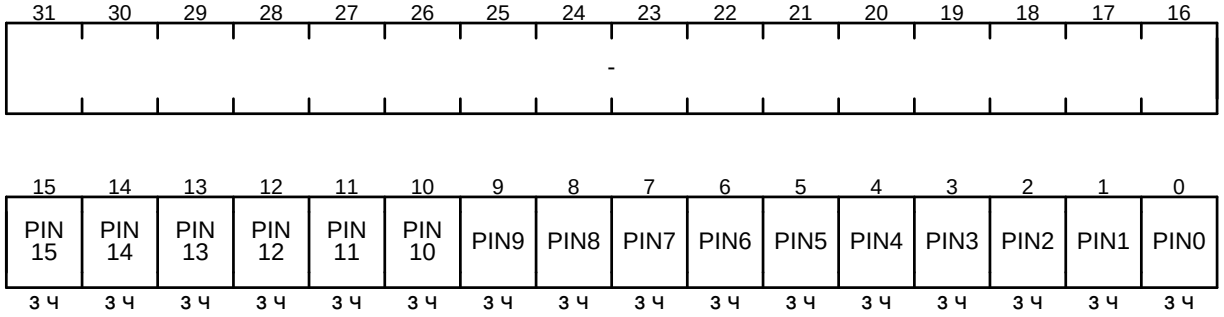
Смещение: + 10h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние порта (регистр DATAOUT). Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам изменяет состояние битов в регистре DATAOUT на противоположное и, как следствие, состояние сигналов на выводе порта, если управление их состоянием было разрешено с помощью регистра OUTENSET
–	31-16	Зарезервировано	

PULLMODE – регистр выбора режима подтяжки порта

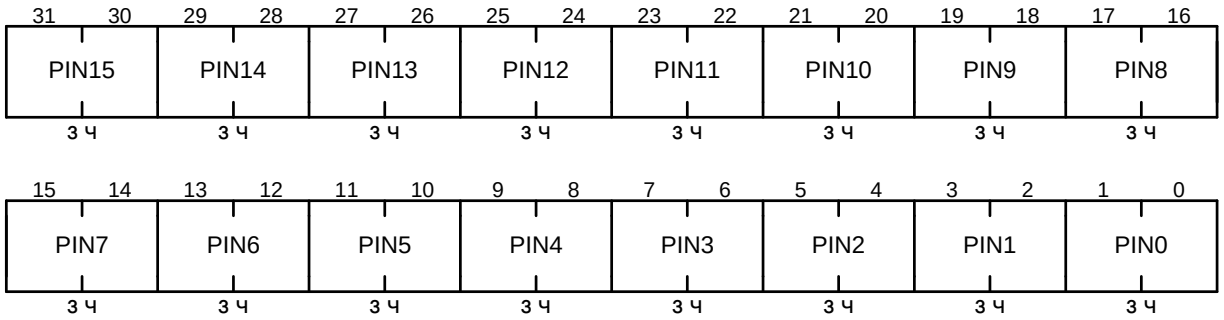
Смещение: + 20h
Сброс: 0h



Поле	Биты	Описание	
PINn	15-0	Поле выбора режима подтяжки вывода n	
		0	Подтяжка отключена
		1	Подтяжка к уровню логической единицы (pull-up)
–	31-16	Зарезервировано	

OUTMODE – регистр выбора режима выхода порта

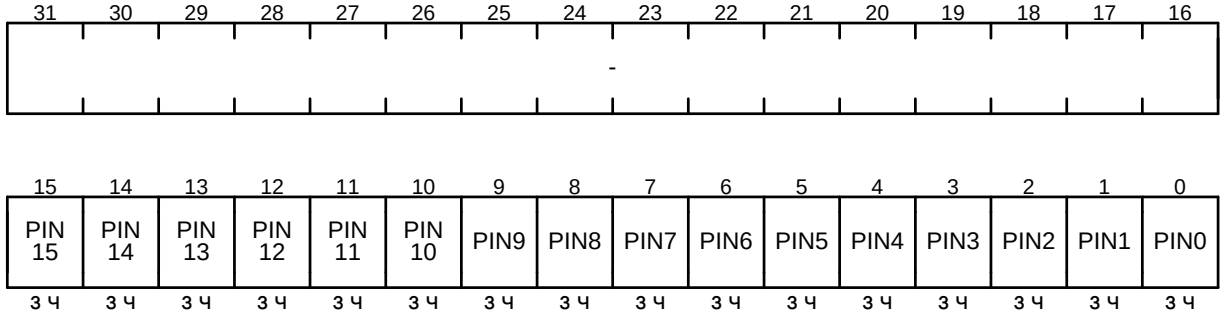
Смещение: + 24h
Сброс: 0h



Поле	Биты	Описание	
PINn	31-0	Поле выбора режима выхода n	
		00	Активный режим выхода (Push-Pull). Двунаправленное управление уровнем сигнала
		01	Выход с открытым стоком (Open Drain). Низкий уровень сигнала
		10	Выход с открытым истоком (Open Source). Высокий уровень сигнала
		11	Зарезервировано

OUTENSET – регистр разрешения управления выходом порта

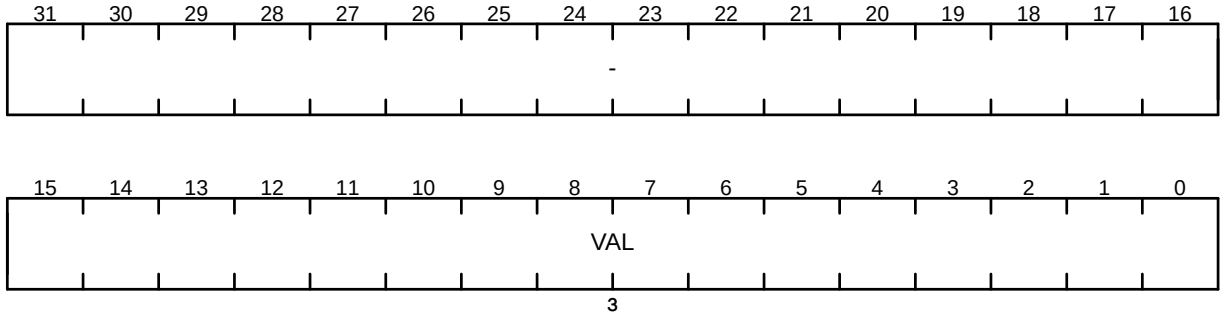
Смещение: + 2Ch
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Источник выходного сигнала (соответствующий бит регистра DATAOUT) отключен от вывода n
			1	Состояние вывода n может задаваться источником выходного сигнала (соответствующий бит регистра DATAOUT)
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает управлять состоянием вывода с помощью соответствующего бита регистра DATAOUT	
–	31-16	Зарезервировано		

OUTENCLR – регистр запрещения управления выходом порта

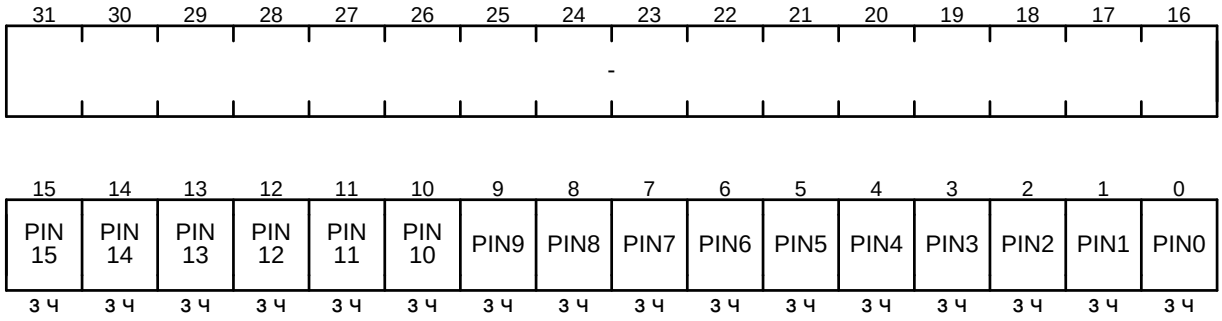
Смещение: + 30h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра OUTENSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра.
		Запись	Соответственно установленным битам сбрасывает соответствующие биты регистра OUTENSET и запрещает управлять состоянием выводов с помощью соответствующих битов регистра DATAOUT
–	31-16	Зарезервировано	

ALTFUNCSET– регистр включения альтернативной функции порта

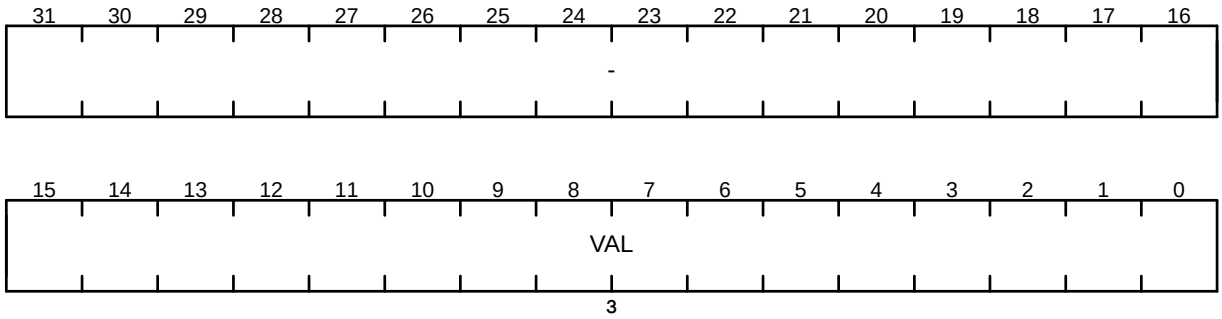
Смещение: + 34h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Вывод n в режиме входа/выхода общего назначения
			1	Вывод n в режиме альтернативной функции
		Запись нуля	Не выполняется	
		Запись единицы	Включает режим альтернативной функции вывода n	
–	31-16	Зарезервировано		

ALTFUNCCLR – регистр выключения альтернативной функции порта

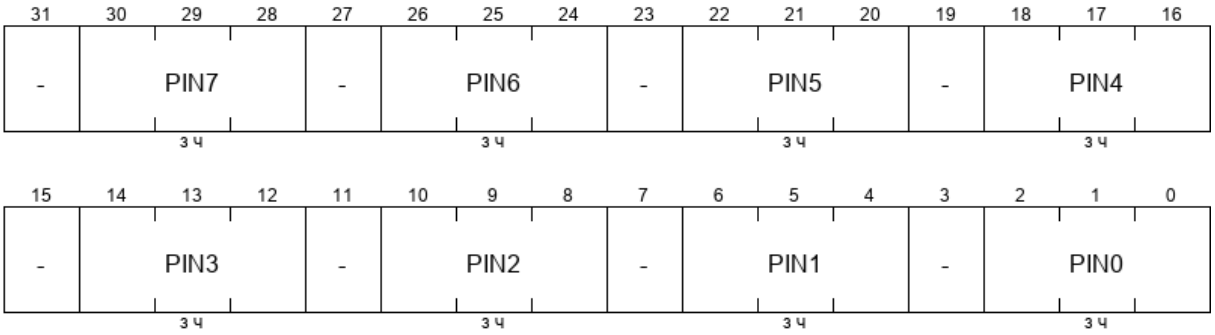
Смещение: + 38h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра ALTFUNCSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам включает режим входов/выходов общего назначения на выводах порта
–	31-16	Зарезервировано	

ALTFUNCNUM0 – регистр 0 выбора альтернативной функции выводов порта

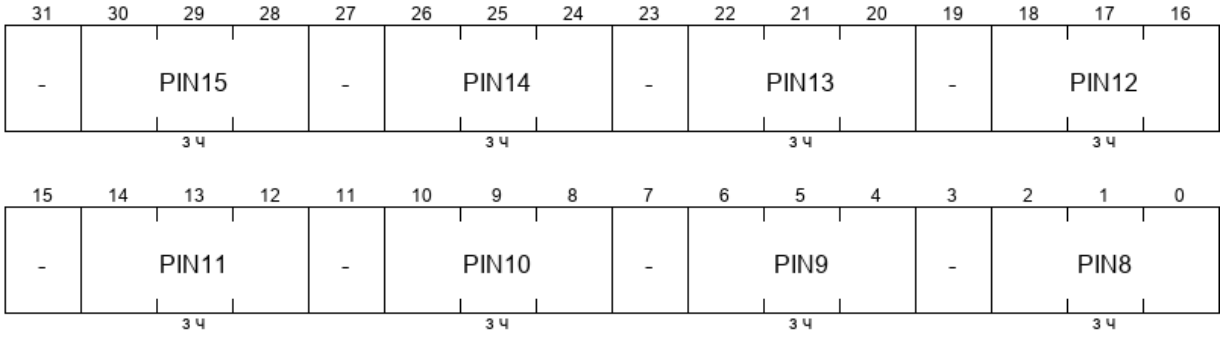
Смещение: + 3Ch
Сброс: 0h



Поле	Биты	Описание	
PINn	30-28, 26-24, 22-20, 18-16, 14-12, 10-8, 6-4, 2-0	Поле выбора альтернативной функции вывода n	
		0h	Альтернативная функция вывода не выбрана
		1h	Альтернативная функция 1
		2h	Альтернативная функция 2
		3h	Альтернативная функция 3
		4h	Альтернативная функция 4
		5h	Альтернативная функция 5
–	31, 27, 23, 19, 15, 11, 7, 3	Зарезервировано	

ALTFUNCNUM1 – регистр 1 выбора альтернативной функции выводов порта

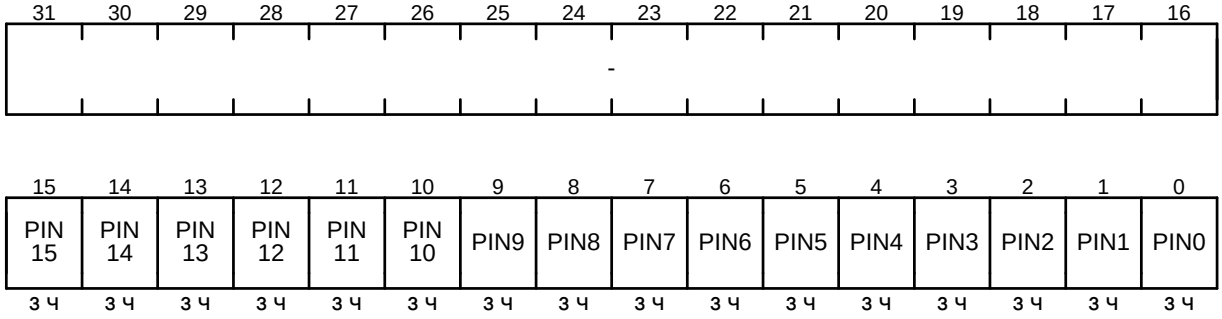
Смещение: + 40h
Сброс: 0h



Поле	Биты	Описание	
PINn	30-28, 26-24, 22-20, 18-16, 14-12, 10-8, 6-4, 2-0	Поле выбора альтернативной функции вывода n	
		0h	Альтернативная функция вывода не выбрана
		1h	Альтернативная функция 1
		2h	Альтернативная функция 2
		3h	Альтернативная функция 3
		4h	Альтернативная функция 4
		5h	Альтернативная функция 5
–	31, 27, 23, 19, 15, 11, 7, 3	Зарезервировано	

SYNCSET – регистр включения дополнительной синхронизации входов портов

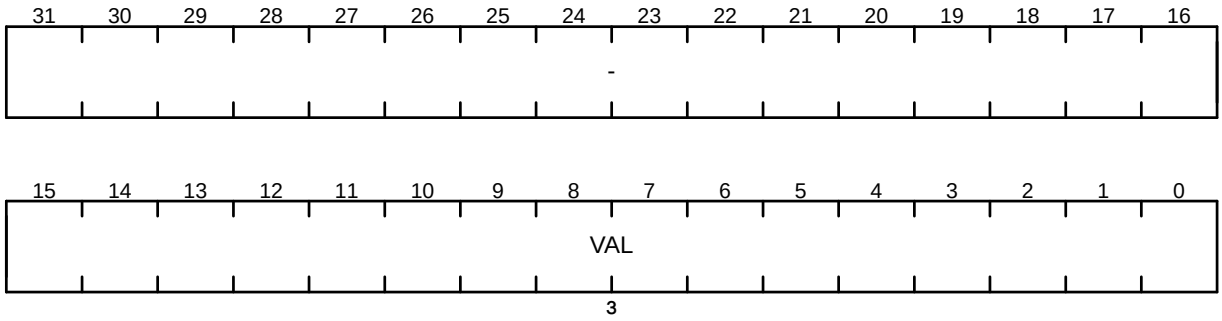
Смещение: + 44h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Сигнал с вывода n передается в регистр DATA после двухтактной синхронизации (базовая)
			1	Сигнал с вывода n передается в регистр DATA после четырехтактной синхронизации (базовая и дополнительная)
		Запись нуля	Не выполняется	
		Запись единицы	Подключает дополнительную схему для получения четырехтактной синхронизации	
–	31-16	Зарезервировано		

SYNCCLR – регистр выключения дополнительной синхронизации входов портов

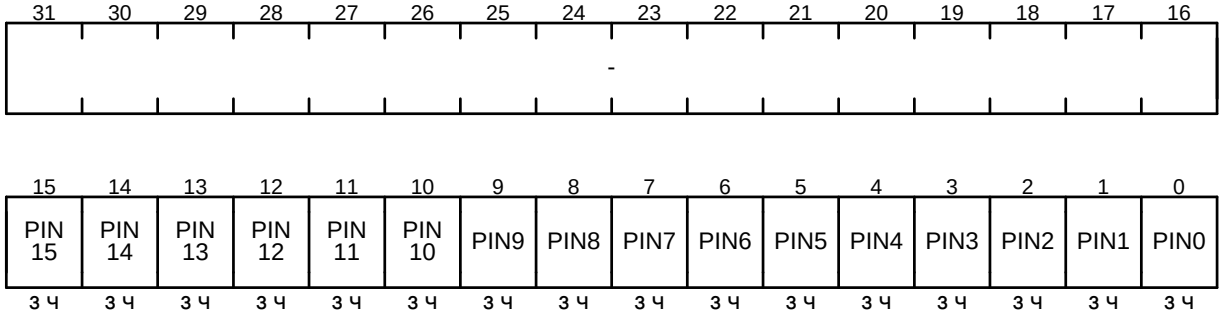
Смещение: + 48h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра SYNCSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам отключает дополнительные схемы синхронизации выводов
–	31-16	Зарезервировано	

QUALSET – регистр включения фильтров портов

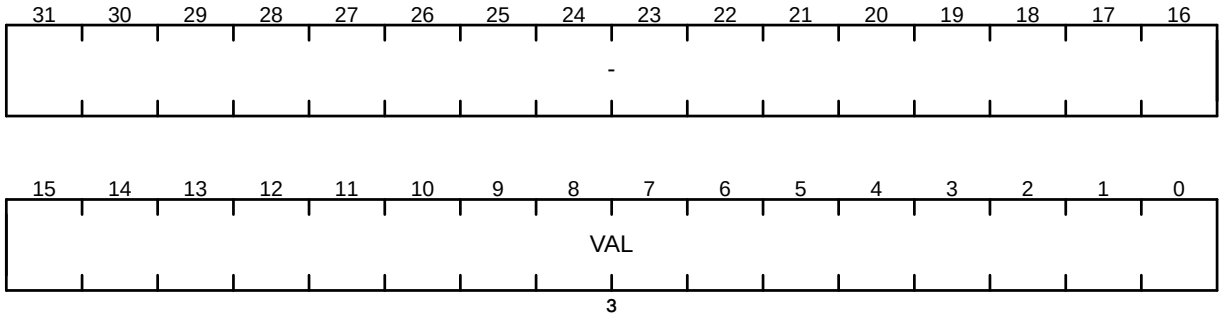
Смещение: + 4Ch
Сброс: 0h



Поле	Биты	Описание	
PINn	15-0	Чтение	0 Входной фильтр вывода n отключен
			1 Входной фильтр вывода n включен
		Запись нуля	Не выполняется
		Запись единицы	Подключает входной фильтр вывода n
–	31-16	Зарезервировано	

QUALCLR – регистр отключения фильтров портов

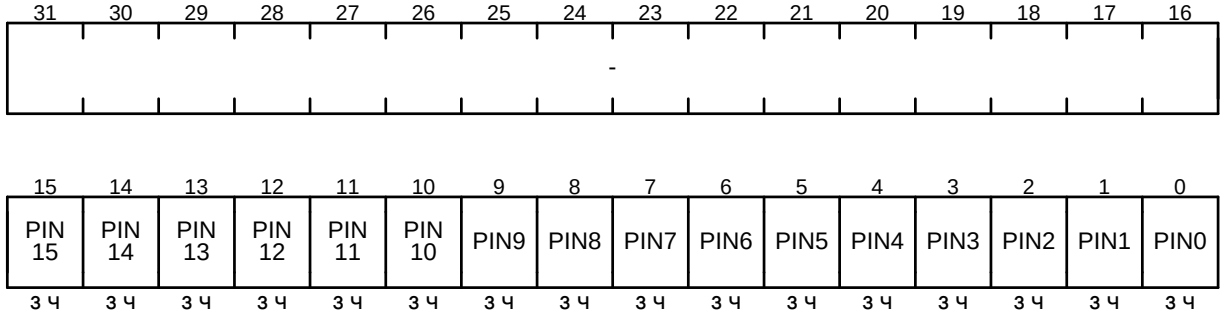
Смещение: + 50h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра QUALSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам отключает входные фильтры выводов
–	31-16	Зарезервировано	

QUALMODESET – регистр режима фильтра порта

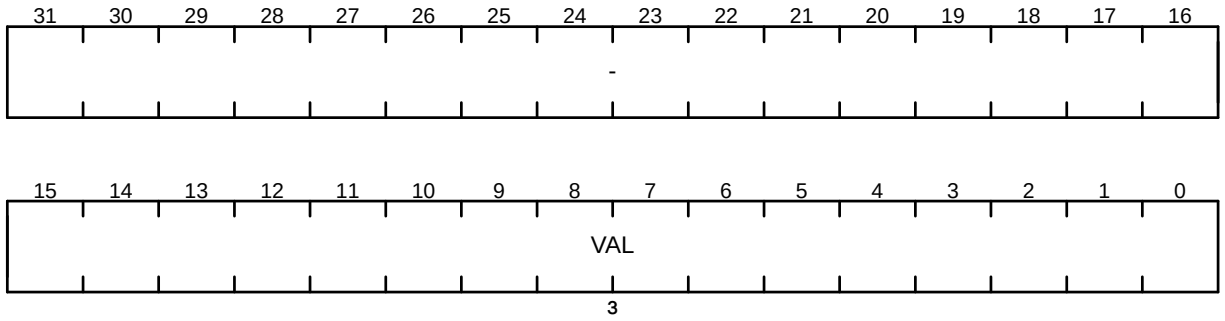
Смещение: + 54h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Включен режим измерения уровня сигнала на выводе n по трем отсчетам
			1	Включен режим измерения уровня сигнала на выводе n по шести отсчетам
		Запись нуля	Не выполняется	
		Запись единицы	Включает режим измерения уровня сигнала на выводе n по шести отсчетам	
–	31-16	Зарезервировано		

QUALMODECLR – регистр сброса режима фильтра порта

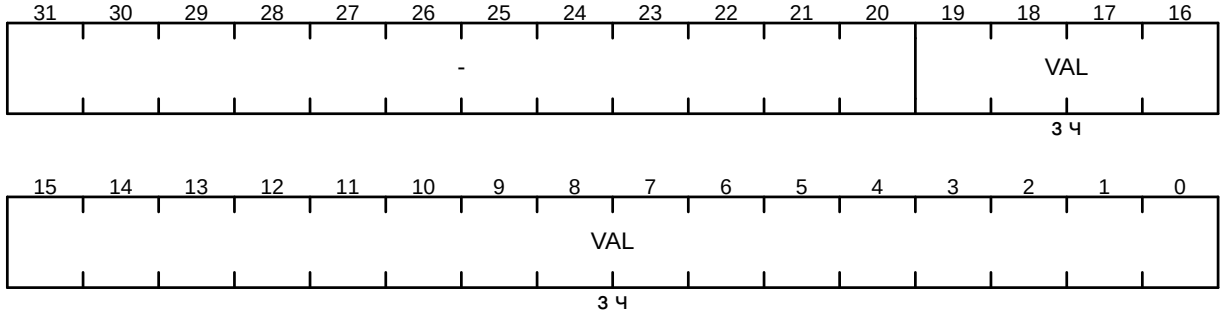
Смещение: + 58h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра QUALMODESET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам сбрасывает биты в регистре QUALMODESET и включает режим измерения уровня сигнала на выводах по трем отсчетам
–	31-16	Зарезервировано	

QUALSAMPLE – регистр настройки фильтра порта

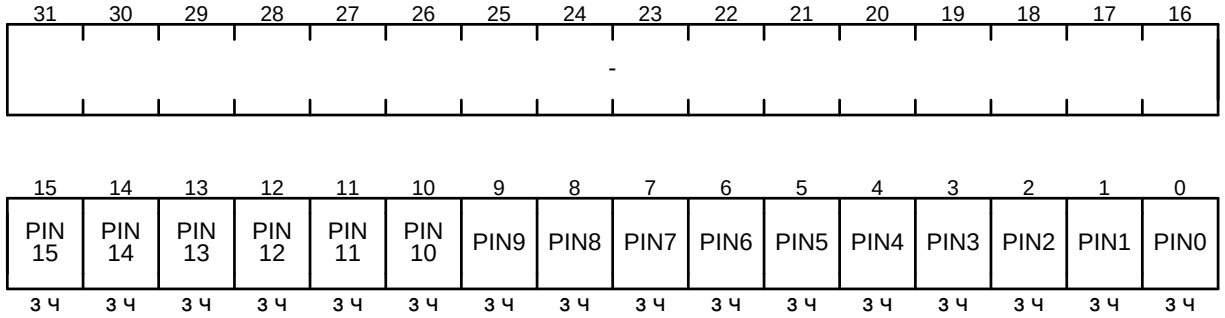
Смещение: + 5Ch
Сброс: 0h



Поле	Биты	Описание
VAL	19-0	Временной интервал (в тактах FCLK) между отсчетами при измерениях уровней сигналов на выводах порта. Заданное значение является единым для всех выводов порта
–	31-20	Зарезервировано

INTENSET – регистр разрешения прерываний порта

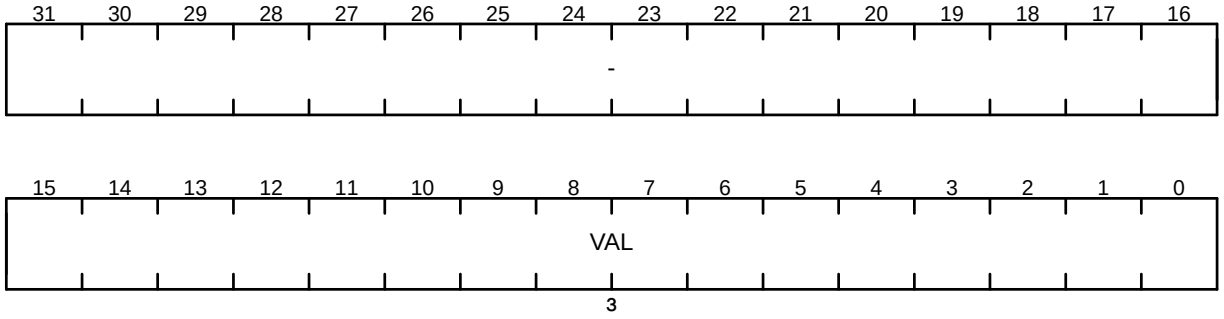
Смещение: + 60h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Прерывания вывода n запрещены
			1	Прерывания вывода n разрешены
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает прерывания вывода n	
–	31-16	Зарезервировано		

INTENCLR – регистр сброса разрешения прерываний порта

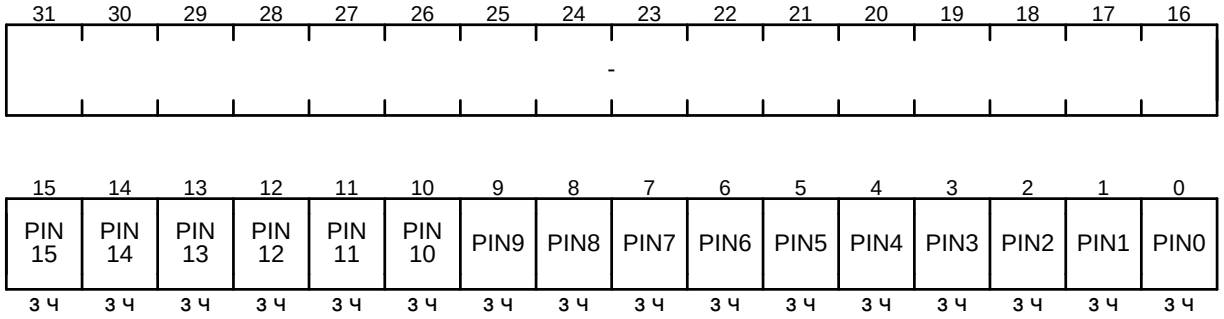
Смещение: + 64h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра INTENSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам сбрасывает биты в регистре INTENSET и запрещает прерывания выводов
–	31-16	Зарезервировано	

INTTYPESET – регистр типа прерываний порта

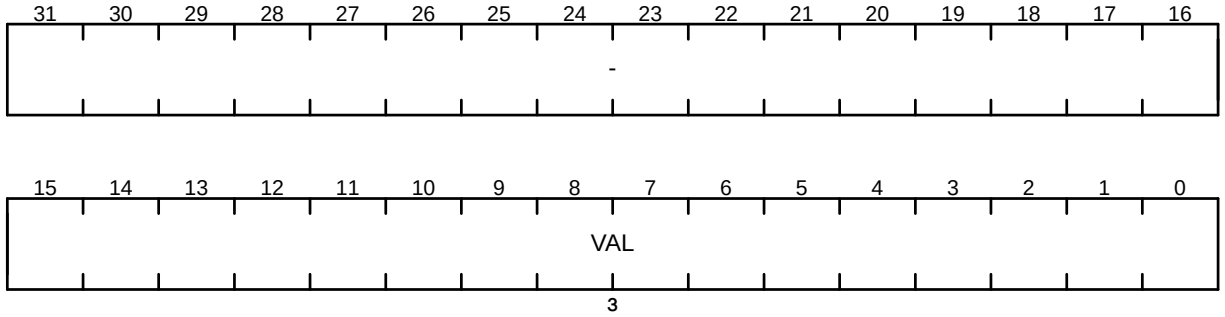
Смещение: + 68h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Прерывания по уровню сигнала на выводе n
			1	Прерывания по фронту сигнала на выводе n
		Запись нуля	Не выполняется	
		Запись единицы	Выбирает прерывания по фронту сигнала на выводе n	

INTTYPECLR – регистр сброса типа прерываний порта

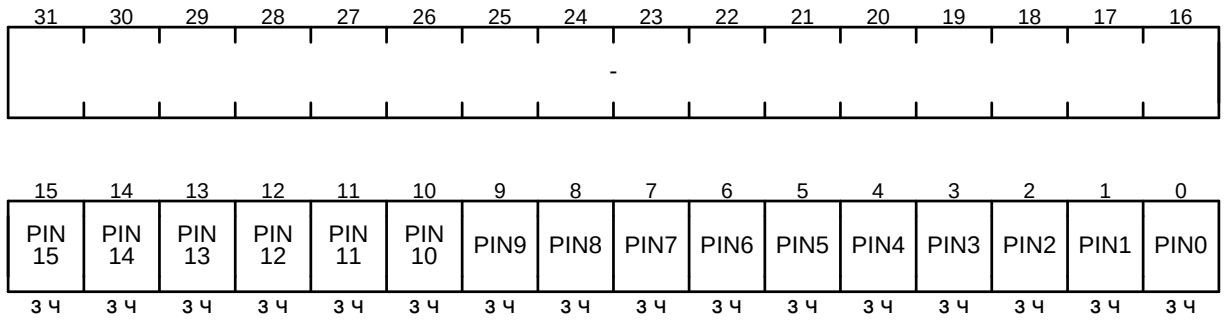
Смещение: + 6Ch
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра INTTYPESET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам сбрасывает биты в регистре INTTYPESET и выбирает прерывания по уровню сигнала на выводах
–	31-16	Зарезервировано	

INTPOLSET – регистр полярности события прерывания порта

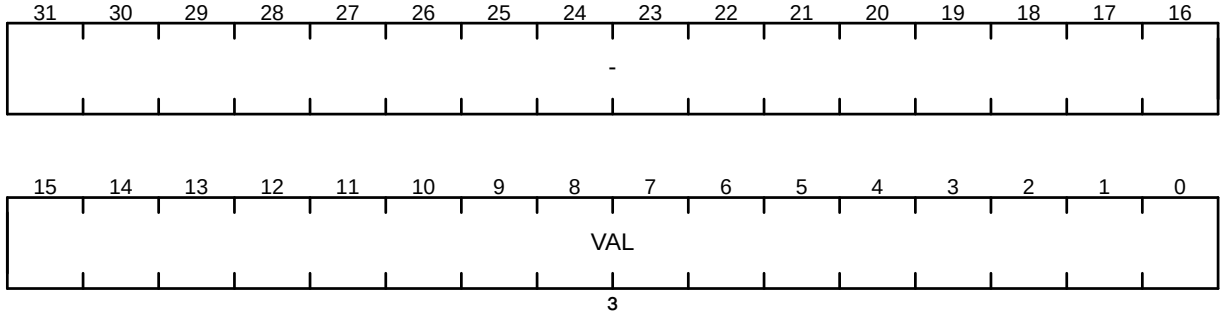
Смещение: + 70h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Прерывания по низкому уровню или отрицательному фронту сигнала на выводе n
			1	Прерывания по высокому уровню или положительному фронту сигнала на выводе n
		Запись нуля	Не выполняется	
		Запись единицы	Выбирает прерывания по высокому уровню или положительному фронту	
–	31-16	Зарезервировано		

INTPOLCLR – регистр сброса полярности события прерывания порта

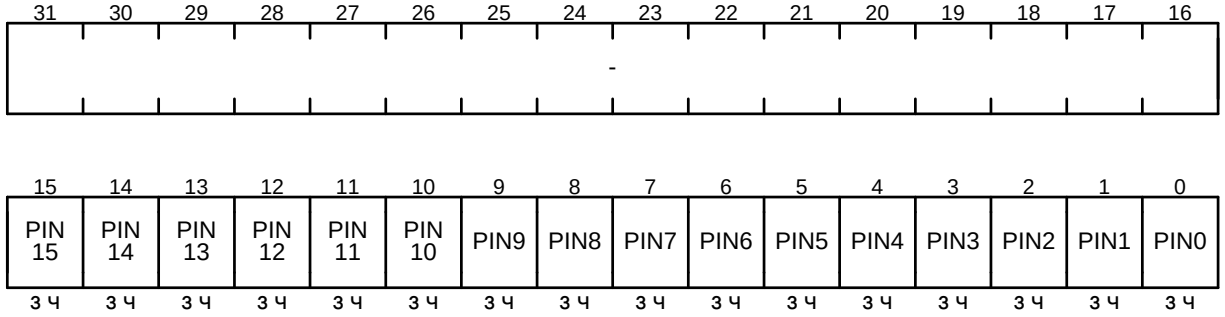
Смещение: + 74h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра INTPOLSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам сбрасывает биты в регистре INTPOLCLR и выбирает прерывания по низкому уровню или отрицательному фронту
–	31-16	Зарезервировано	

INTEDGESET – регистр включения прерывания по любому перепаду

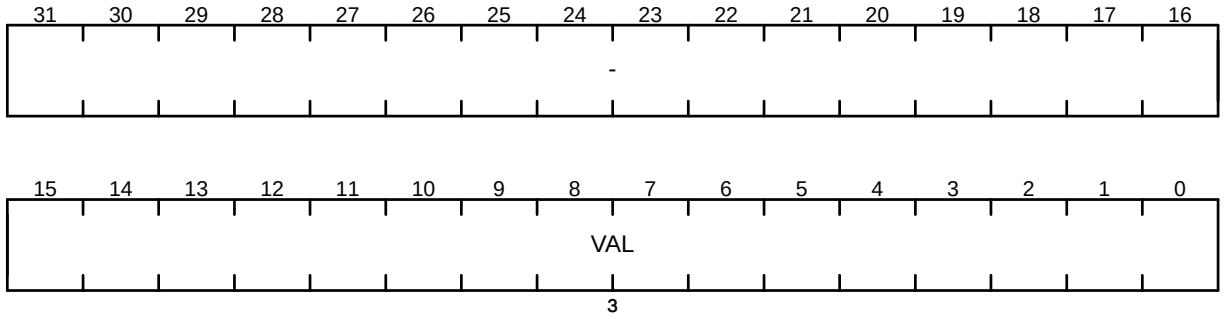
Смещение: + 78h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Прерывания по любому фронту сигнала на выводе n запрещены
			1	Прерывания по любому фронту сигнала на выводе n разрешены
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает прерывания по любому фронту сигнала на выводе n	
–	31-16	Зарезервировано		

INTEDGECLR – регистр отключения прерывания по любому перепаду

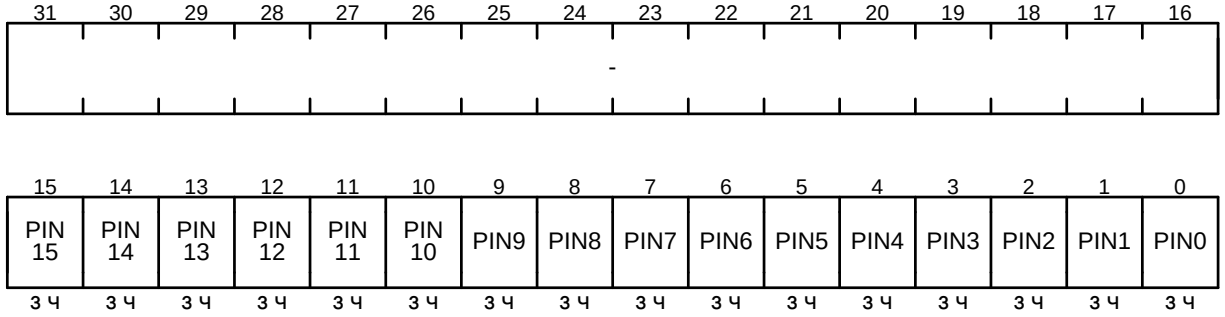
Смещение: + 7Ch
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра INTEDGESET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам сбрасывает биты в регистре INTEDGESET и запрещает прерывания по любому фронту сигнала на выводах
–	31-16	Зарезервировано	

INTSTATUS – регистр состояния и сброса прерываний порта

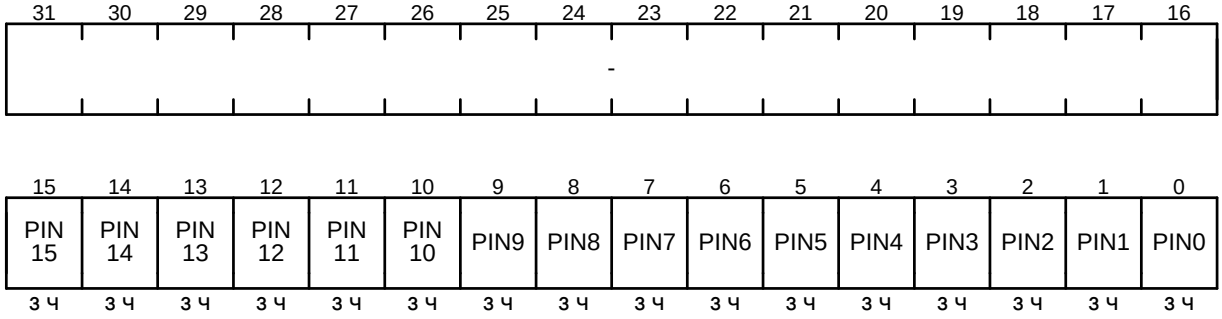
Смещение: + 80h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Нет прерывания
			1	Флаг запроса на прерывание. Бит не сбрасывается аппаратно
		Запись нуля	Не выполняется	
		Запись единицы	Обнуляет бит PINn и таким образом сбрасывает флаг прерывания, если он был установлен	
–	31-16	Зарезервировано		

DMATXREQSET – регистр включения генерации запросов DMA TX по прерыванию порта

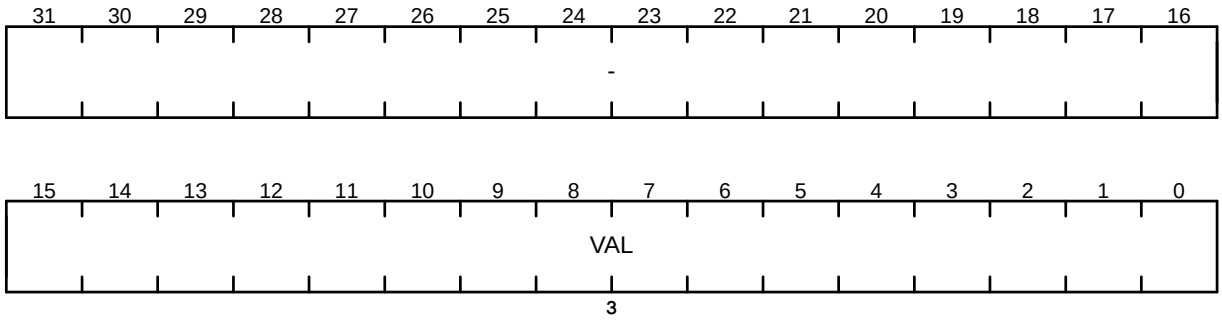
Смещение: + 84h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Запрос DMA TX не генерируется
			1	Запрос DMA TX генерируется по прерыванию (в том числе немаскированному)
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает генерирование запросов DMA TX по прерыванию вывода n (в том числе немаскированному)	
–	31-16	Зарезервировано		

DMATXREQCLR – регистр отключения генерации запросов DMA TX по прерыванию порта

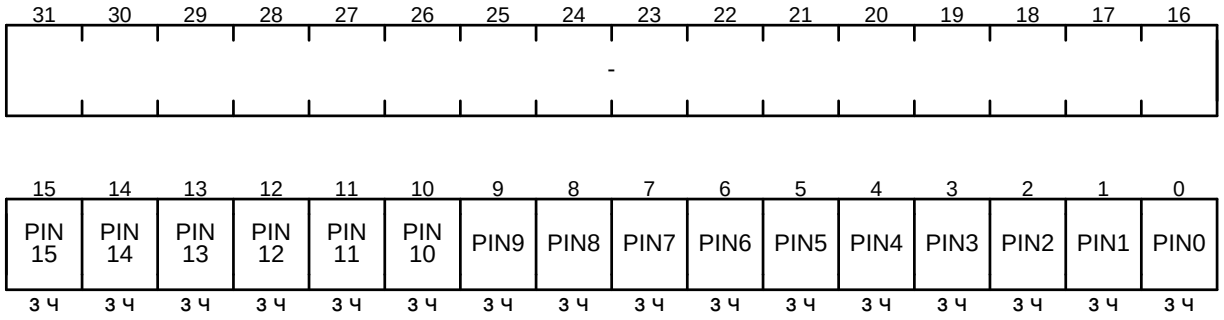
Смещение: + 88h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра DMATXREQSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра.
		Заспись	Соответственно установленным битам сбрасывает биты в регистре DMATXREQSET и запрещает генерирование запросов DMA TX по прерыванию выводов
–	31-16	Зарезервировано	

ADCSOCSET – регистр включения генерации запросов начала преобразования АЦП по прерыванию порта

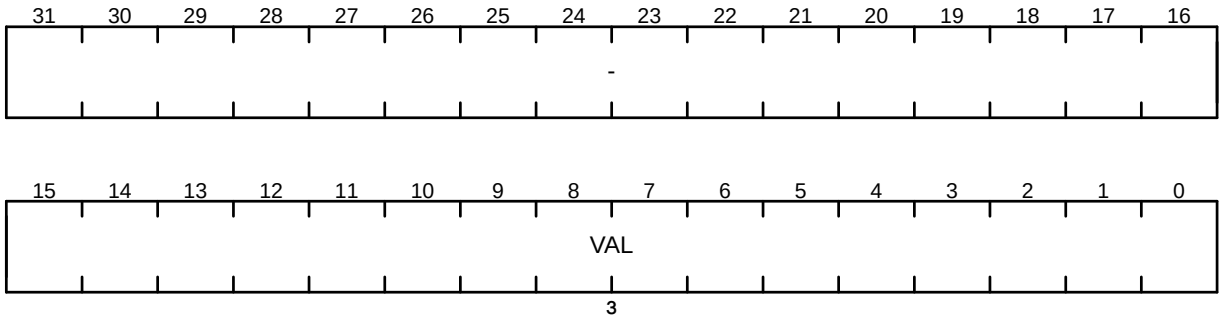
Смещение: + 8Ch
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Запрос начала преобразования АЦП не генерируется
			1	Запрос начала преобразования АЦП генерируется по прерыванию (в том числе немаскированному)
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает генерирование запросов начала преобразования АЦП по прерыванию вывода n (в том числе немаскированному)	
–	31-16	Зарезервировано		

ADCSOCCLR – регистр отключения генерации запросов начала преобразования АЦП по прерыванию порта

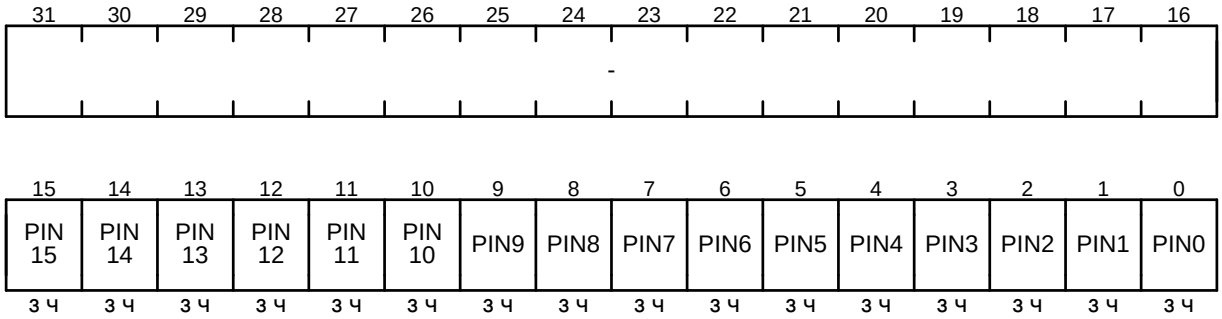
Смещение: + 90h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра ADCSOCSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам сбрасывает биты в регистре ADCSOCSET и запрещает генерирование запросов АЦП по прерыванию выводов
–	31-16	Зарезервировано	

RXEVSET – регистр включения генерации запросов RXEV к ядру по прерыванию порта

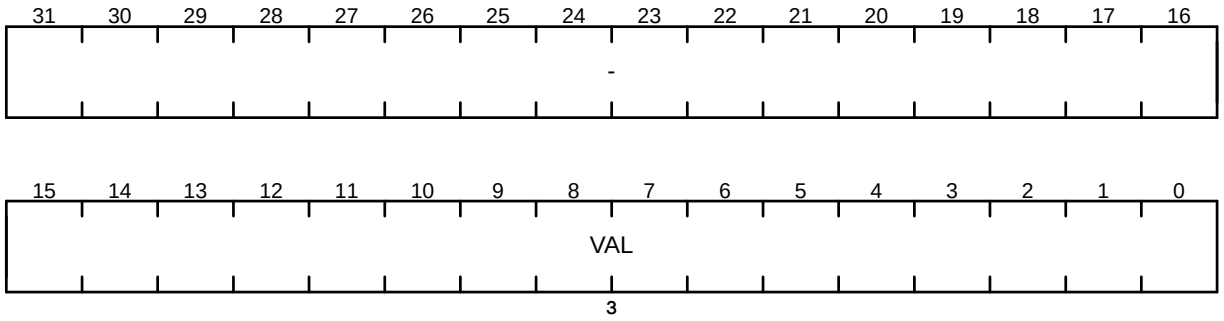
Смещение: +94h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Запрос RXEV к ядру не генерируется
			1	Запрос RXEV к ядру генерируется по прерыванию (в том числе немаскированному)
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает генерирование запросов RXEV к ядру по прерыванию вывода n (в том числе немаскированному)	
–	31-16	Зарезервировано		

RXEVCLR – регистр отключения генерации запросов RXEV к ядру по прерыванию порта

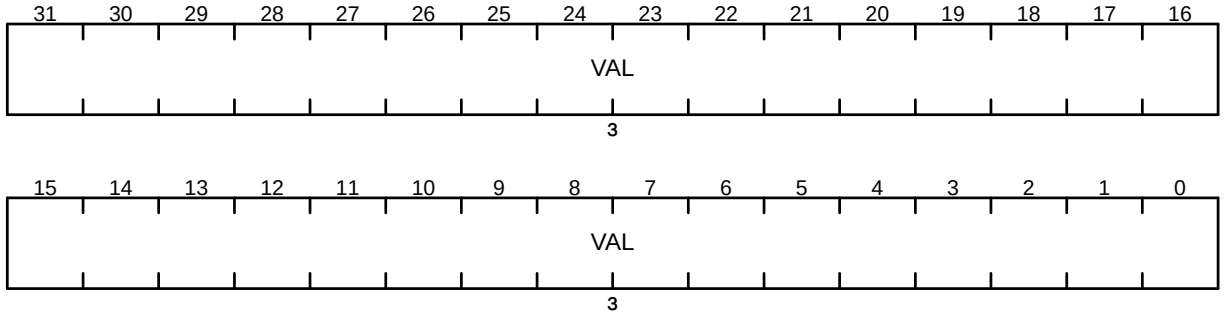
Смещение: + 98h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра RXEVSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись	Соответственно установленным битам сбрасывает биты n в регистре RXEVSET и запрещает генерирование запросов RXEV к ядру по прерываниям выводов
–	31-16	Зарезервировано	

LOCKKEY – регистр ключа блокировки

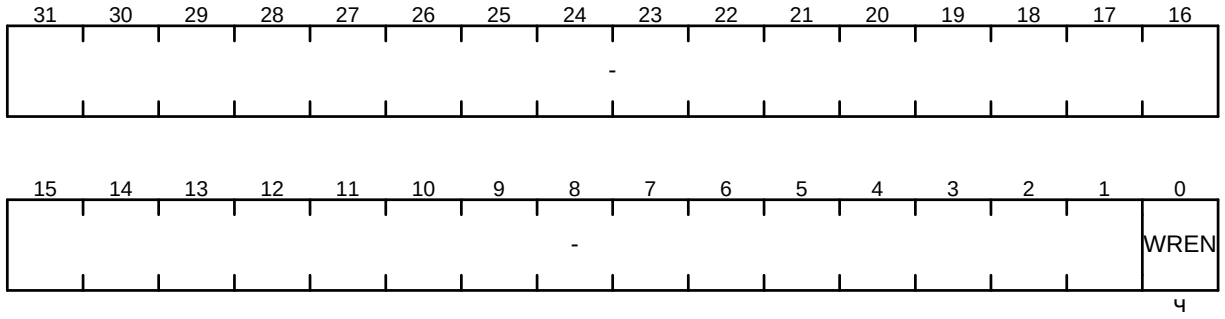
Смещение: + 9Ch
Сброс: 0h



Поле	Биты	Описание	
VAL	31-0	Значение ключа – ADEADBEEh	
		Чтение	Не выполняется
		Запись любого значения отличного от ключа	Блокирует доступ к регистрам LOCKSET и LOCKCLR
		Запись значения ключа	Разрешает доступ к регистрам LOCKSET и LOCKCLR

LOCKSTAT– регистр статуса блокировки

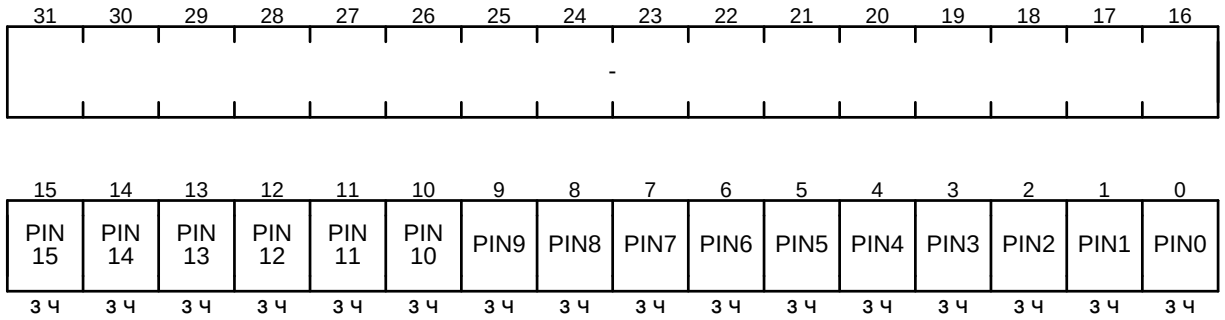
Смещение: + 9Ch
Сброс: 0h



Поле	Биты	Описание	
WREN	0	Индикатор доступа регистров LOCKSET и LOCKCLR для записи	
		0	Запрещено
		1	Разрешено
		Бит недоступен для записи	
–	31-1	Зарезервировано	

LOCKSET – регистр включения блокировки изменения конфигурации вывода

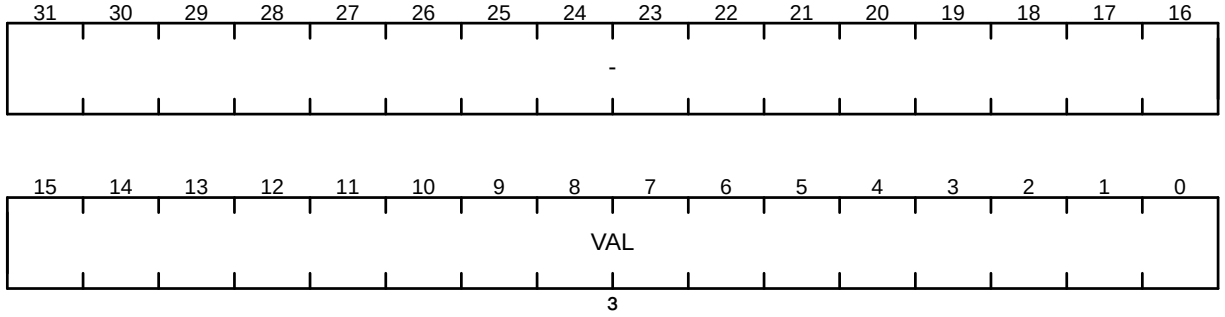
Смещение: + A0h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Блокировка отключена
			1	Блокировка включена
		Запись нуля	Не выполняется	
		Запись единицы (возможна после установки ключа в регистре LOCKKEY)	Блокирует изменение конфигурации вывода n. Соответствующие биты n регистров становятся недоступными для записи. Исключения: - регистр флагов прерываний INTSTATUS; - регистр временного интервала измерений QUALSAMPLE	
–	31-16	Зарезервировано		

LOCKCLR – регистр отключения блокировки изменения конфигурации вывода

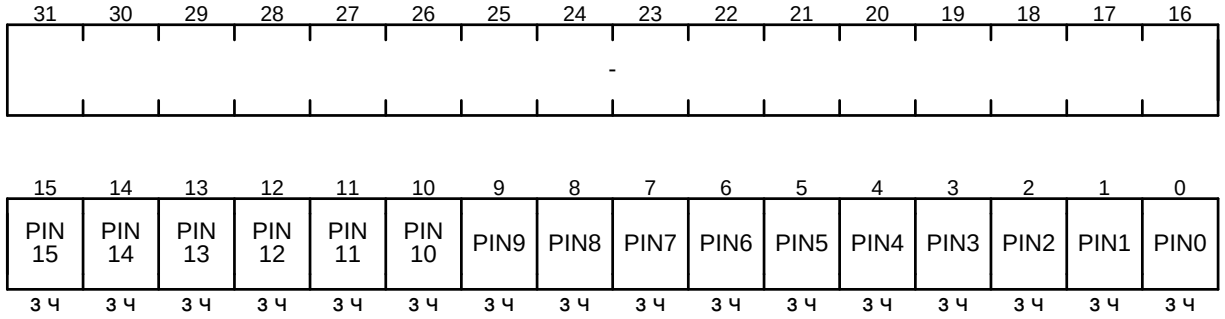
Смещение: + A4h
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра LOCKSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Запись (возможна после установки ключа в регистре LOCKKEY)	Соответственно установленным битам сбрасывает биты в регистре LOCKSET и разрешает изменения конфигурации выводов
–	31-16	Зарезервировано	

DMARXREQSET – регистр включения генерации запросов DMA RX по прерыванию порта

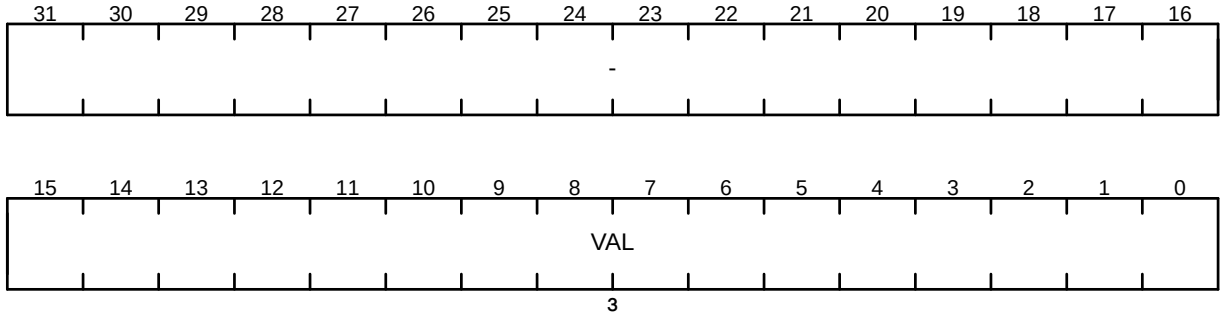
Смещение: + A8h
Сброс: 0h



Поле	Биты	Описание		
PINn	15-0	Чтение	0	Запрос DMA RX не генерируется
			1	Запрос DMA RX генерируется по прерыванию (в том числе немаскированному)
		Запись нуля	Не выполняется	
		Запись единицы	Разрешает генерирование запросов DMA RX по прерыванию вывода n (в том числе немаскированному)	
–	31-16	Зарезервировано		

DMARXREQCLR – регистр отключения генерации запросов DMA RX по прерыванию порта

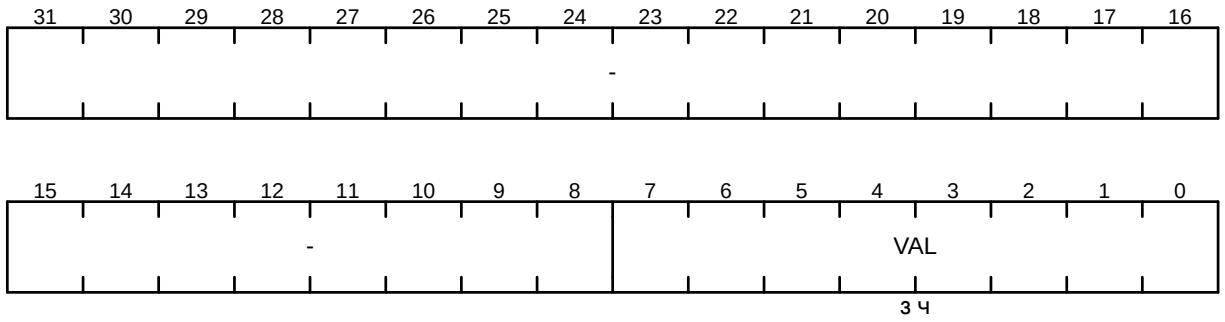
Смещение: + ACh
Сброс: 0h



Поле	Биты	Описание	
VAL	15-0	Чтение	Возвращает состояние регистра DMARXREQSET. Примечание – Использование операции чтение-модификация-запись может привести к нежелательным результатам, влияющим на другие биты регистра
		Заспись	Соответственно установленным битам сбрасывает биты в регистре DMARXREQSET и запрещает генерирование запросов DMA RX по прерыванию выводов
–	31-16	Зарезервировано	

MASKLB – регистр массива масок младшего байта порта

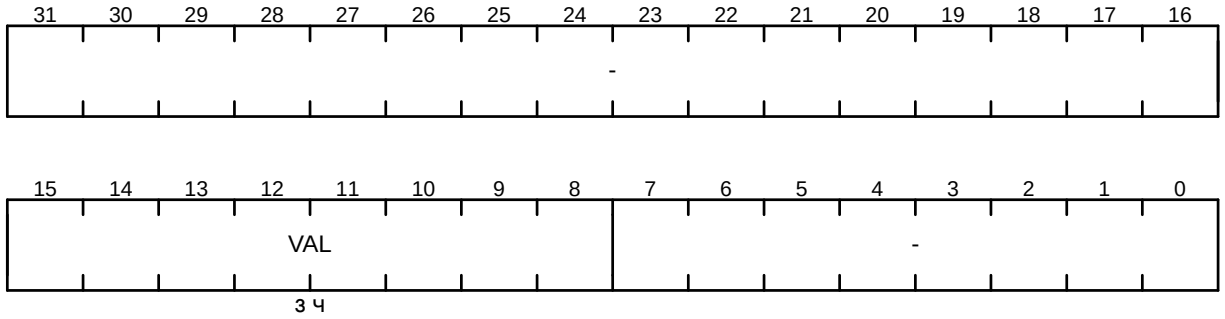
Адрес: GPIOp + MASKLB + (4*i)h, i = 0h, …, FFh
Сброс: 000000xxh



Поле	Биты	Описание
VAL	7-0	Доступ по маске для младших восьми бит порта
–	31-8	Зарезервировано

MASKHB – регистр массива масок старшего байта порта

Адрес: GPIOp + MASKHB + (4*i)h, i = 0h, …, FFh
Сброс: 0000_xx00h



Поле	Биты	Описание
VAL	15-8	Доступ по маске для старших восьми бит порта
–	31-16, 7-0	Зарезервировано

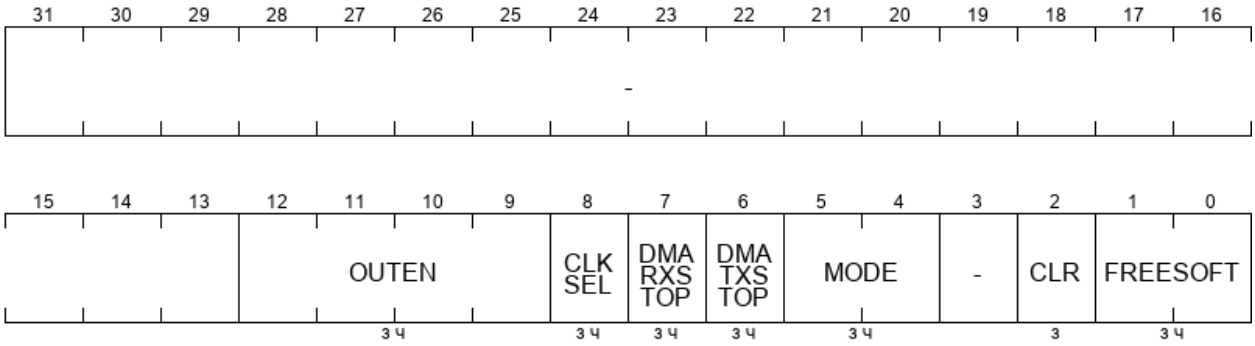
A.10 Регистры 32-разрядного таймера TMR

Базовый адрес:	5100_8000h	Регистры блока TMR0
	5100_9000h	Регистры блока TMR1
	5100_A000h	Регистры блока TMR2
	5100_B000h	Регистры блока TMR3
	5100_C000h	Регистры блока TMR4
	5100_D000h	Регистры блока TMR5
	5100_E000h	Регистры блока TMR6
	5100_F000h	Регистры блока TMR7
	5101_0000h	Регистры блока TMR8
	5101_1000h	Регистры блока TMR9
	5101_2000h	Регистры блока TMR10
	5101_3000h	Регистры блока TMR11
	5101_4000h	Регистры блока TMR12
	5101_5000h	Регистры блока TMR13
	5101_6000h	Регистры блока TMR14
	5101_7000h	Регистры блока TMR15
Смещение:	+ 100h (CAPCOM0)	Регистры захвата и сравнения CAPCOM0
	+ 110h (CAPCOM1)	Регистры захвата и сравнения CAPCOM1
	+ 120h (CAPCOM2)	Регистры захвата и сравнения CAPCOM2
	+ 130h (CAPCOM3)	Регистры захвата и сравнения CAPCOM3
Мнемоника:	CAPCOMn	

Примечания: n – номер блока захвата и сравнения от 0 до 3 для блоков TMR0 - TMR3; Блоки TMR4 – TMR15 имеют только один блок захвата и сравнения CAPCOM0.

CTRL – регистр конфигурации таймера

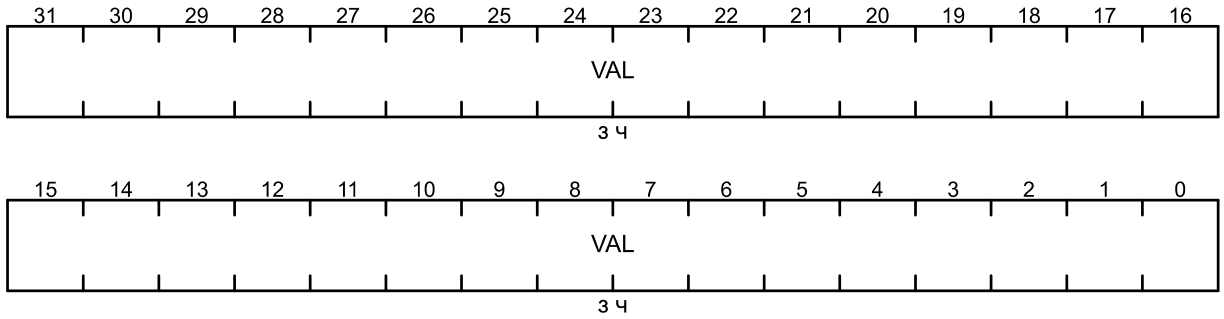
Смещение: + 00h
Сброс: 0h



Поле	Биты	Описание	
OUTEN	12-9	Разрешение работы на выход для блоков CAPCOM. Каждый бит поля отвечает за переключение одного из блоков CAPCOM. Например, запись в поле OUTEN 0xF разрешит работу для всех блоков, а запись 0x1 разрешит работу блока CAPCOM0.	
CLKSEL	8	Бит выбора тактирования таймера	
		0	тактовый сигнал SYSCLK
		1	внешний источник синхронизации
DMARXSTOP	7	Бит разрешения остановки счетчика таймера после завершения всех RX транзакций DMA	
DMATXSTOP	6	Бит разрешения остановки счетчика таймера после завершения всех TX транзакций DMA	
MODE	5-4	Выбор режима счета	
		0	останов
		1	счет вверх (многократно от 0 до значения PERIOD)
		2	непрерывно (многократно от 0 до значения FFFF_FFFFh)
		3	вверх/вниз (многократно от 0 вверх до значения PERIOD и обратно до 0).
CLR	2	Бит сброса счётчика (а также делителя CLKDIV и направления счета в режиме вверх/вниз)	
FREESOFT	1-0	Выбор режима эмуляции. Выбор поведения счетчика во время событий эмуляции	
		0	Остановка таймера при следующем такте CLK
		1	Остановка таймера при перегрузке
		2	Бесконечный отсчет
		3	Зарезервировано
–	31-13, 3	Зарезервировано	

COUNT – регистр счетчика таймера

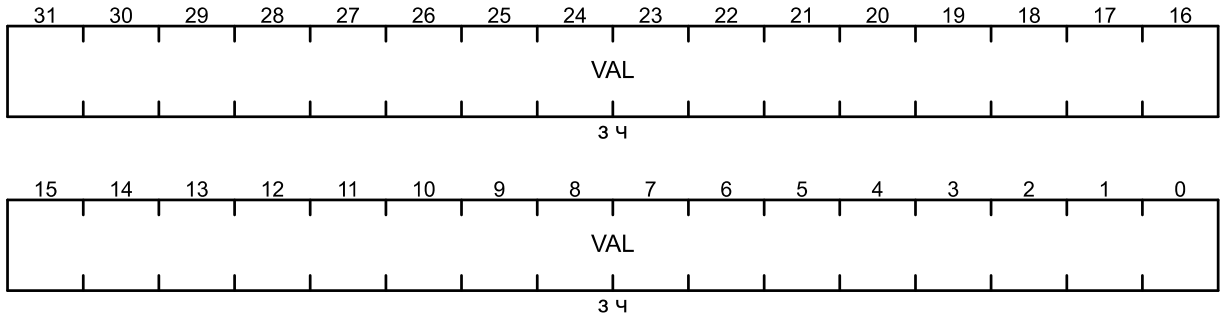
Смещение: + 04h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	32-разрядное текущее значение счетчика таймера

CLKDIV – регистр делителя тактового сигнала таймера

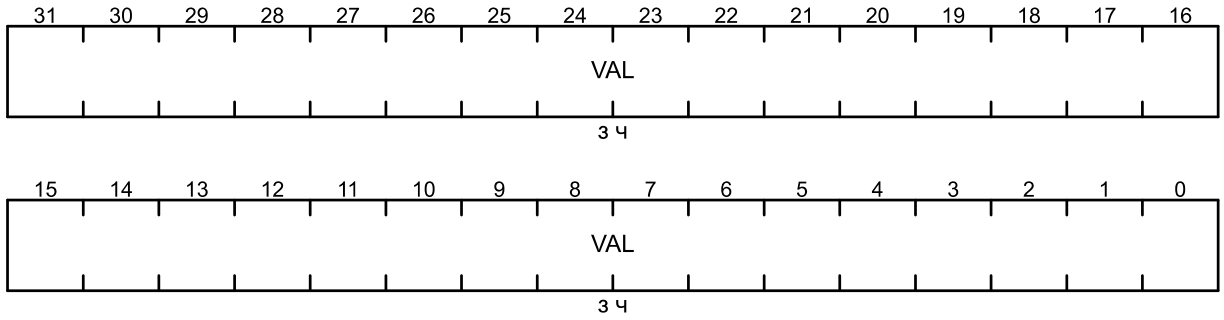
Смещение: + 08h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	32-разрядное значение делителя тактового сигнала

PERIOD – регистр периода счета таймера

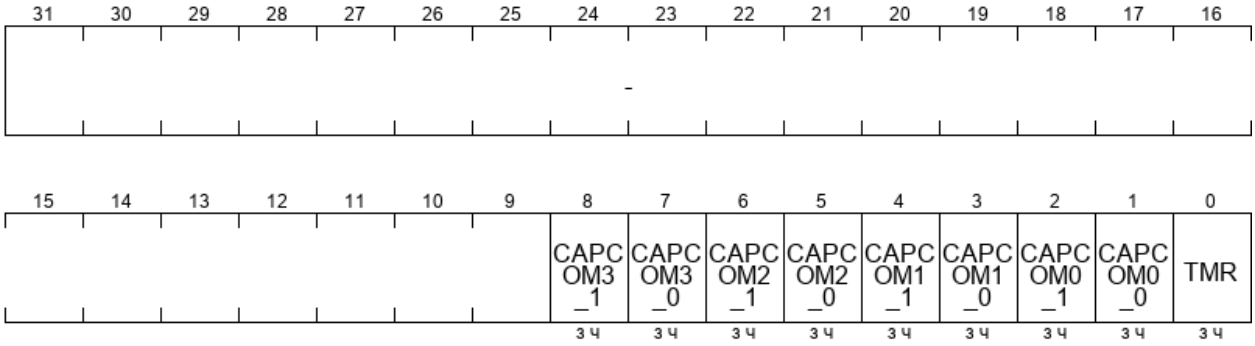
Смещение: + 0Ch
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	32-разрядное значение периода счета таймера в режимах «вверх» и «вверх/вниз»

IM – регистр маски прерываний

Смещение: + 10h
Сброс: 0h

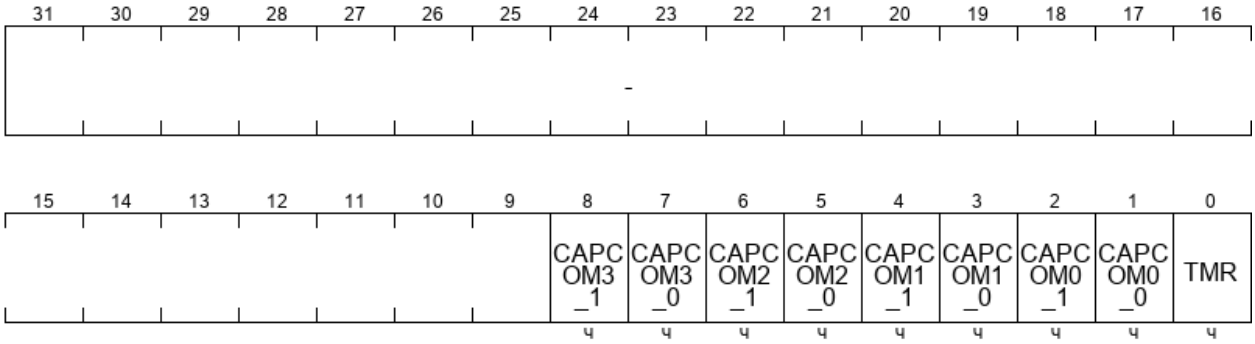


Поле	Биты	Описание
CAPCOM3_1	8	Маска линии прерывания 1 блока CAPCOM3
CAPCOM3_0	7	Маска линии прерывания 0 блока CAPCOM3
CAPCOM2_1	6	Маска линии прерывания 1 блока CAPCOM2
CAPCOM2_0	5	Маска линии прерывания 0 блока CAPCOM2
CAPCOM1_1	4	Маска линии прерывания 1 блока CAPCOM1
CAPCOM1_0	3	Маска линии прерывания 0 блока CAPCOM1
CAPCOM0_1	2	Маска линии прерывания 1 блока CAPCOM0
CAPCOM0_0	1	Маска линии прерывания 0 блока CAPCOM0
TMR	0	Маска линии прерывания таймера
–	31-9	Зарезервировано

Примечание – Маска прерываний формируется установкой бит. Необходимо установить биты тех событий, прерывания от которых нужно разрешить.

RIS – регистр состояния немаскированных прерываний

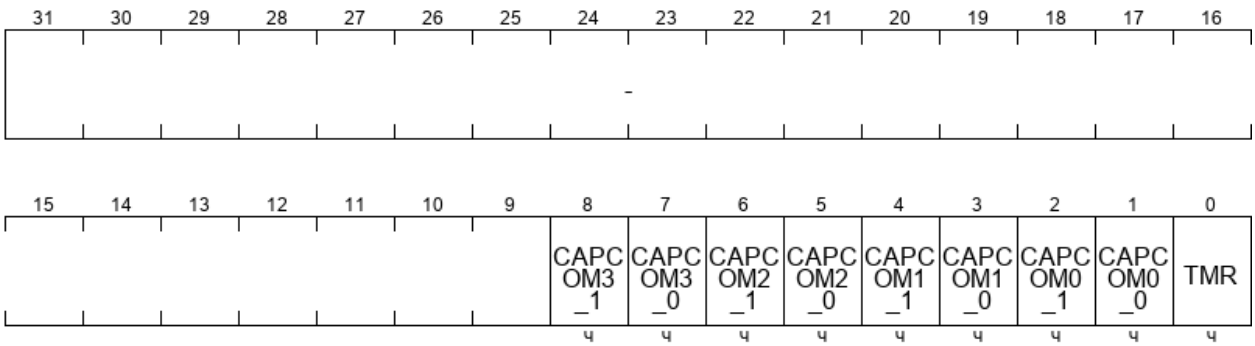
Смещение: + 14h
Сброс: 0h



Поле	Биты	Описание
CAPCOM3_1	8	Статус прерывания 1 блока CAPCOM3
CAPCOM3_0	7	Статус прерывания 0 блока CAPCOM3
CAPCOM2_1	6	Статус прерывания 1 блока CAPCOM2
CAPCOM2_0	5	Статус прерывания 0 блока CAPCOM2
CAPCOM1_1	4	Статус прерывания 1 блока CAPCOM1
CAPCOM1_0	3	Статус прерывания 0 блока CAPCOM1
CAPCOM0_1	2	Статус прерывания 1 блока CAPCOM0
CAPCOM0_0	1	Статус прерывания 0 блока CAPCOM0
TMR	0	Статус прерывания таймера
–	31-9	Зарезервировано

MIS – регистр состояния прерываний с маскированием

Смещение: + 18h
Сброс: 0h

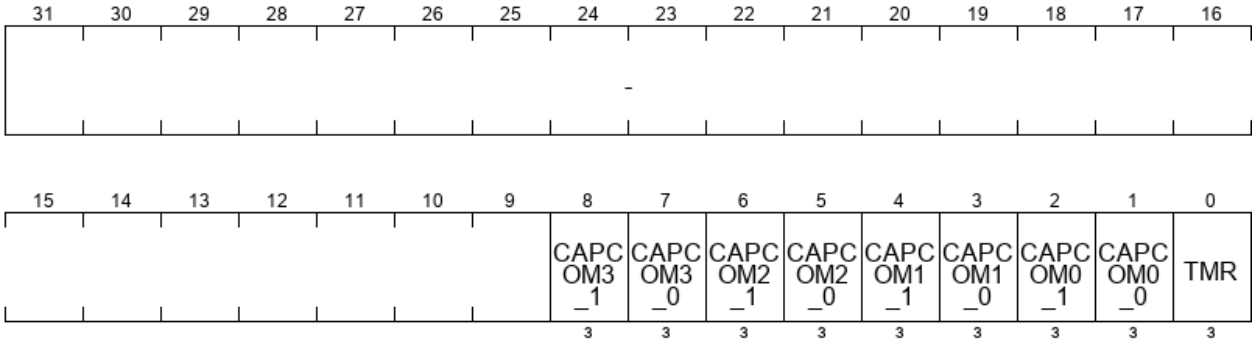


Поле	Биты	Описание
CAPCOM3_1	8	Статус прерывания 1 блока CAPCOM3
CAPCOM3_0	7	Статус прерывания 0 блока CAPCOM3

Поле	Биты	Описание
CAPCOM2_1	6	Статус прерывания 1 блока CAPCOM2
CAPCOM2_0	5	Статус прерывания 0 блока CAPCOM2
CAPCOM1_1	4	Статус прерывания 1 блока CAPCOM1
CAPCOM1_0	3	Статус прерывания 0 блока CAPCOM1
CAPCOM0_1	2	Статус прерывания 1 блока CAPCOM0
CAPCOM0_0	1	Статус прерывания 0 блока CAPCOM0
TMR	0	Статус прерывания таймера
–	31-9	Зарезервировано

IC – регистр сброса прерываний

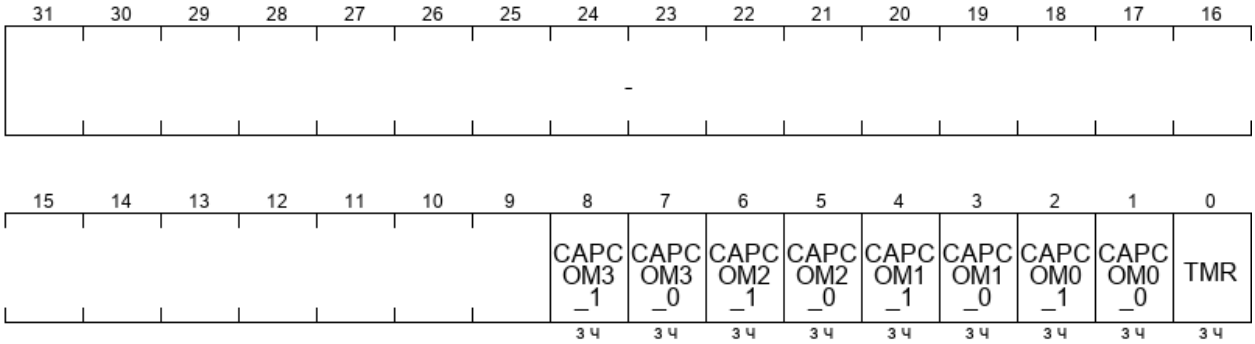
Смещение: + 1Ch
Сброс: 0h



Поле	Биты	Описание
CAPCOM3_1	8	Сброс флага маскированного и немаскированного прерывания 1 блока CAPCOM3
CAPCOM3_0	7	Сброс флага маскированного и немаскированного прерывания 0 блока CAPCOM3
CAPCOM2_1	6	Сброс флага маскированного и немаскированного прерывания 1 блока CAPCOM2
CAPCOM2_0	5	Сброс флага маскированного и немаскированного прерывания 0 блока CAPCOM2
CAPCOM1_1	4	Сброс флага маскированного и немаскированного прерывания 1 блока CAPCOM1
CAPCOM1_0	3	Сброс флага маскированного и немаскированного прерывания 0 блока CAPCOM1
CAPCOM0_1	2	Сброс флага маскированного и немаскированного прерывания 1 блока CAPCOM0
CAPCOM0_0	1	Сброс флага маскированного и немаскированного прерывания 0 блока CAPCOM0
TMR	0	Сброс флага маскированного и немаскированного прерывания таймера
–	31-9	Зарезервировано

DMA_TXIM – регистр маски TX запросов DMA

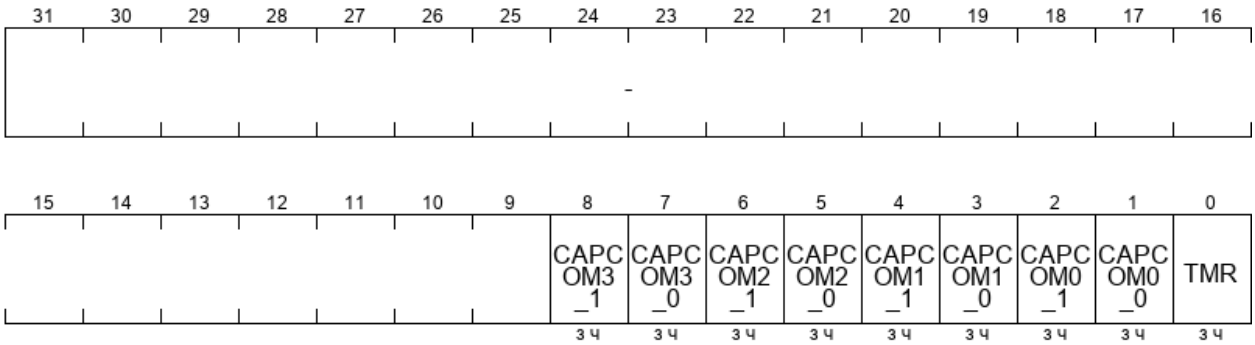
Смещение: + 20h
Сброс: 0h



Поле	Биты	Описание
CAPCOM3_1	8	Маска линии запроса DMA TX 1 от блока CAPCOM3
CAPCOM3_0	7	Маска линии запроса DMA TX 0 от блока CAPCOM3
CAPCOM2_1	6	Маска линии запроса DMA TX 1 от блока CAPCOM2
CAPCOM2_0	5	Маска линии запроса DMA TX 0 от блока CAPCOM2
CAPCOM1_1	4	Маска линии запроса DMA TX 1 от блока CAPCOM1
CAPCOM1_0	3	Маска линии запроса DMA TX 0 от блока CAPCOM1
CAPCOM0_1	2	Маска линии запроса DMA TX 1 от блока CAPCOM0
CAPCOM0_0	1	Маска линии запроса DMA TX 0 от блока CAPCOM0
TMR	0	Маска линии запроса DMA TX от таймера
–	31-9	Зарезервировано

DMA_RXIM – регистр маски RX запросов DMA

Смещение: + 24h
Сброс: 0h

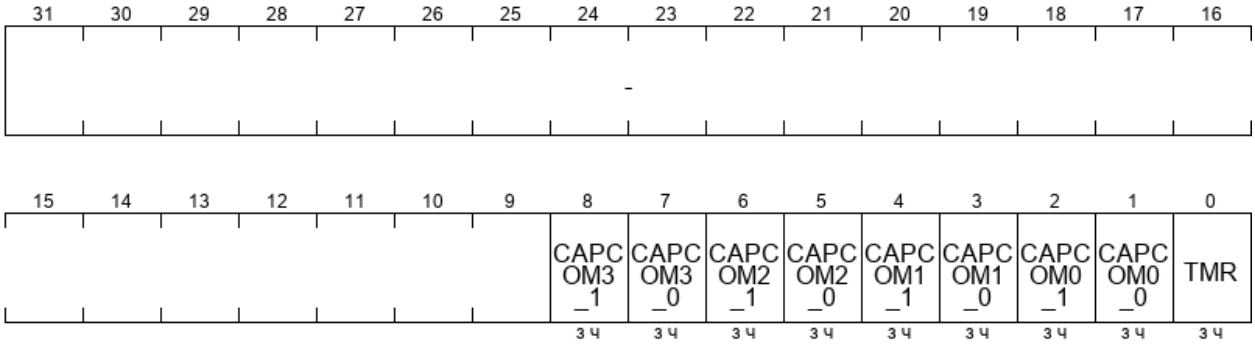


Поле	Биты	Описание
CAPCOM3_1	8	Маска линии запроса DMA RX 1 от блока CAPCOM3
CAPCOM3_0	7	Маска линии запроса DMA RX 0 от блока CAPCOM3
CAPCOM2_1	6	Маска линии запроса DMA RX 1 от блока CAPCOM2
CAPCOM2_0	5	Маска линии запроса DMA RX 0 от блока CAPCOM2

Поле	Биты	Описание
CAPCOM1_1	4	Маска линии запроса DMA RX 1 от блока CAPCOM1
CAPCOM1_0	3	Маска линии запроса DMA RX 0 от блока CAPCOM1
CAPCOM0_1	2	Маска линии запроса DMA RX 1 от блока CAPCOM0
CAPCOM0_0	1	Маска линии запроса DMA RX 0 от блока CAPCOM0
TMR	0	Маска линии запроса DMA RX от таймера
–	31-9	Зарезервировано

EXTEVT_IM – регистр маски запросов внешних событий

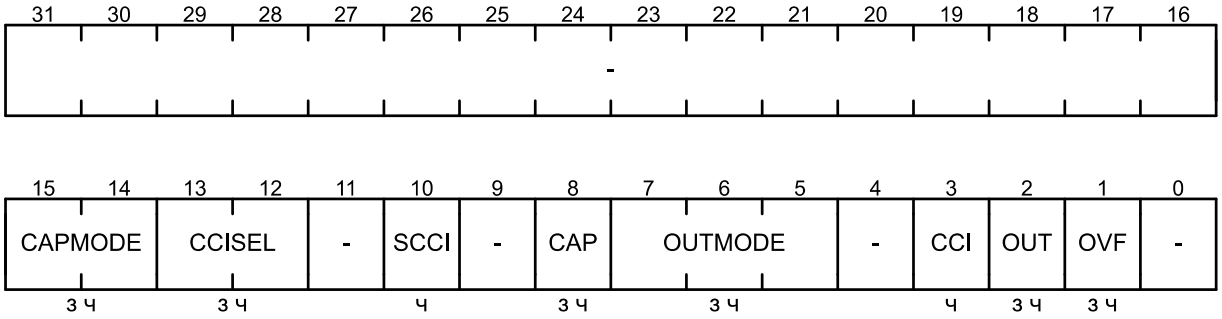
Смещение: + 28h
Сброс: 0h



Поле	Биты	Описание
CAPCOM3_1	8	Маска линии запроса на внешнее событие 1 от блока CAPCOM3
CAPCOM3_0	7	Маска линии запроса на внешнее событие 0 от блока CAPCOM3
CAPCOM2_1	6	Маска линии запроса на внешнее событие 1 от блока CAPCOM2
CAPCOM2_0	5	Маска линии запроса на внешнее событие 0 от блока CAPCOM2
CAPCOM1_1	4	Маска линии запроса на внешнее событие 1 от блока CAPCOM1
CAPCOM1_0	3	Маска линии запроса на внешнее событие 0 от блока CAPCOM1
CAPCOM0_1	2	Маска линии запроса на внешнее событие 1 от блока CAPCOM0
CAPCOM0_0	1	Маска линии запроса на внешнее событие 0 от блока CAPCOM0
TMR	0	Маска линии запроса на внешнее событие от таймера
–	31-9	Зарезервировано

CAPCOM_CTRL – регистр настройки захвата и сравнения

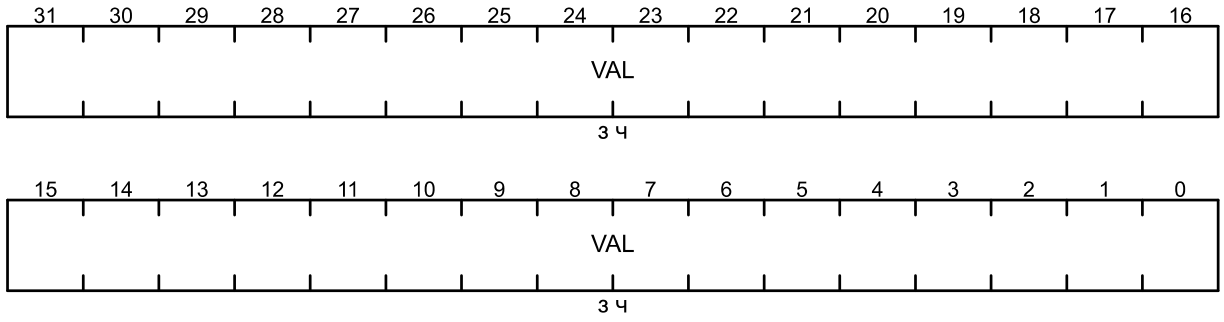
Смещение: CAPCOMn + 00h
Сброс: 0000_0020h



Поле	Биты	Описание
CAPMODE	15-14	Выбор режима захвата
		0 захвата нет
		1 захват по переднему фронту тактового сигнала
		2 захват по заднему фронту тактового сигнала
		3 захват по обоим фронтам тактового сигнала
CCISEL	13-12	Выбор входа захвата/сравнения
		0 CCIA
		1 CCIB
		2 Уровень логического нуля (Low)
		3 Уровень логической единицы (High)
SCCI	10	Бит состояния синхронизированного входа захвата/сравнения. Примечание – выбранный входной сигнал защелкивается по сигналу EQUx и может быть прочитан через этот бит.
CAP	8	Выбор режима захвата/сравнения
		0 режим сравнения
		1 режим захвата (CAPMODE)
OUTMODE	7-5	Выбор режима выхода. Значение по умолчанию – 1
		0 значение бита OUT
		1 установка (Set);
		2 переключение/сброс (Toggle/Reset);
		3 установка/сброс (Set/Reset);
		4 переключение (Toggle);
		5 сброс (Reset);
		6 переключение/установка (Toggle/Set);
		7 сброс/установка (Reset/Set).
CCI	3	Флаг состояния входа захвата/сравнения. Отображает значение выбранного входного сигнала
OUT	2	Выбор состояния выхода. Для режима выхода (OUTMODE = 0) бит напрямую определяет значение выходного сигнала.
		0 низкий уровень выходного сигнала
		1 высокий уровень выходного сигнала
OVF	1	Флаг переполнения захвата
		0 переполнение отсутствует
		1 произошло переполнение
		Данный флаг должен быть сброшен программно, записью нуля либо чтением регистра CAPCOM_VAL1.
–	31-16, 11, 9, 4, 0	Зарезервировано

CAPCOM_VAL0 – регистр 0 значения захвата и сравнения

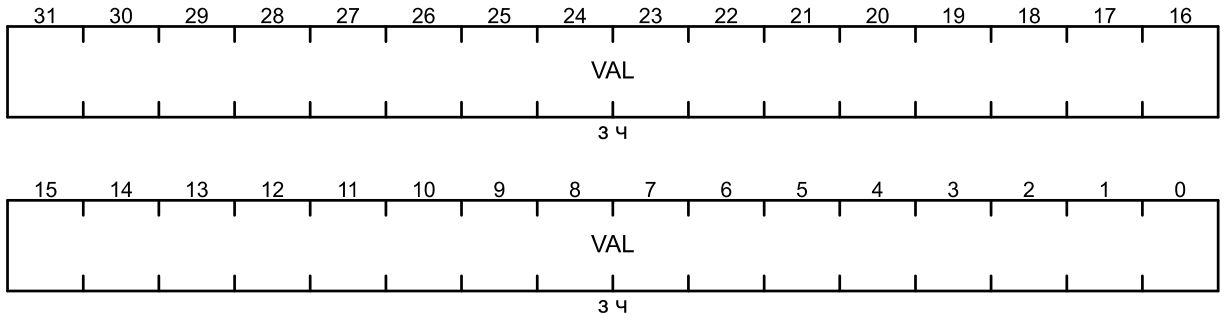
Смещение: CAPCOMn + 04h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	32-разрядное значение регистра захвата и сравнения

CAPCOM_VAL1 – регистр 1 значения захвата и сравнения

Смещение: CAPCOMn + 08h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	32-разрядное значение регистра захвата и сравнения

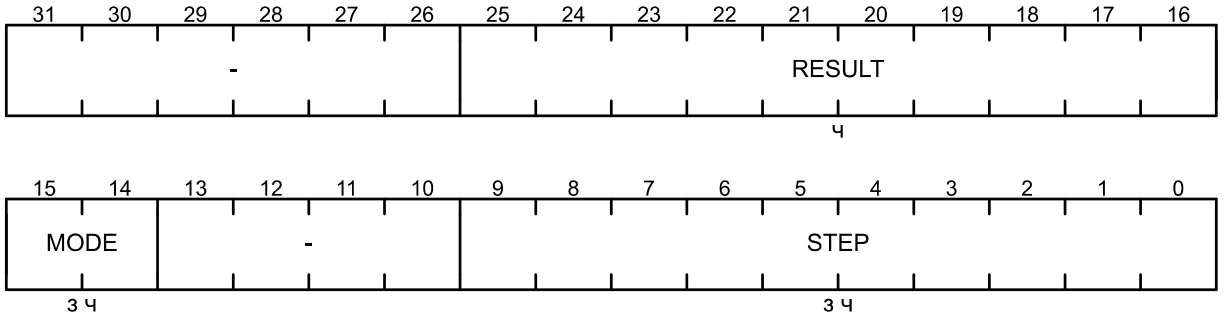
A.11 Регистры блока MSC

Базовый адрес: 5201_C000h

Смещение: + 040h (U_ITEMS) Регистры объектов канала приема

FDR – Регистр дробного делителя входного тактового сигнала

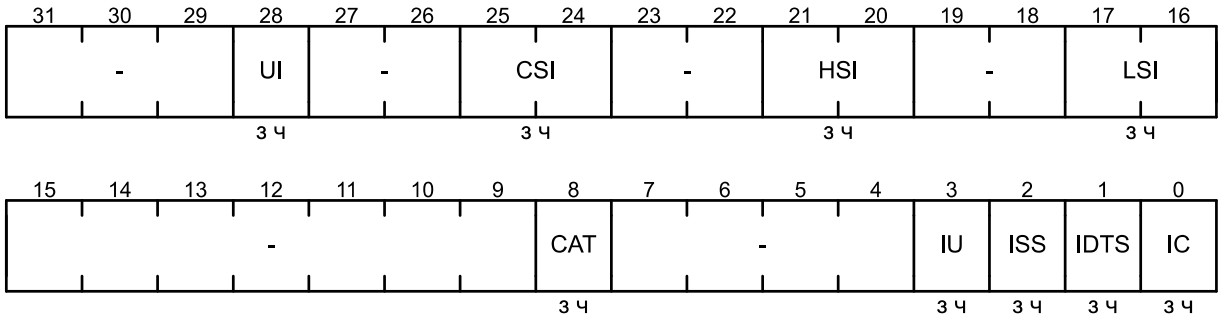
Смещение: + 00h
Сброс: 0h



Поле	Биты	Описание	
RESULT	25-16	Счетчик делителя частоты	
MODE	15-14	Поле задания режима делителя частоты	
		00b	Счетчик выключен
		01b	Нормальный режим работы. Частота тактового сигнала $f_{MSC} = f_{PCLK} / (1024 - STEP)$. Сигнал сброса внешнего делителя в состоянии логического нуля. При активации режима поле RESULT загружается значением 3FFh. Далее периодически загружается значением из STEP
		10b	Режим дробного деления. Частота тактового сигнала $f_{MSC} = f_{PCLK} * STEP / 1024$. Сигнал сброса внешнего делителя в состоянии логического нуля. При активации режима поле RESULT загружается значением 3FFh. Далее периодически загружается значением из STEP
		11b	Режим работы «на проход» (без деления) Синхросигнал f_{MSC} соответствует входному тактовому сигналу. Поле RESULT не меняется
STEP	9-0	Шаг делителя. Поле хранит значение, которое загружается в RESULT при переполнении счетчика делителя. В режиме дробного деления данное значение каждый такт добавляется к текущему значению RESULT	
–	31-26, 13-10	Зарезервировано	

IO_CTRL – Регистр конфигурации входов/выходов

Смещение: + 04h
Сброс: 0h

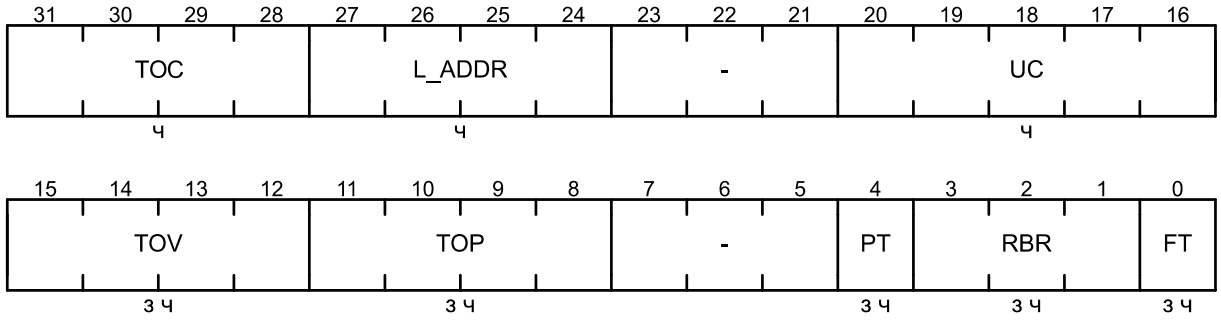


Поле	Биты	Описание
UI	28	Индекс активного входа канала приема Данное поле определяет с которого из входов данные от ведомых устройств будут подаваться в модуль MSC
CSI	25-24	Индекс активного выхода выбора ведомого команды Данное поле определяет на который из выходов выбора ведомого будет выставлен активный уровень при передаче кадра команды
HSI	21-20	Индекс активного выхода выбора ведомого старшей части данных Данное поле определяет на который из выходов выбора ведомого будет выставлен активный уровень при передаче старшей части кадра данных
LSI	17-16	Индекс активного выхода выбора ведомого младшей части данных Данное поле определяет на который из выходов выбора ведомого будет выставлен активный уровень при передаче младшей части кадра данных
CAT	8	Непрерывное тактирование Если бит сброшен, то тактирование осуществляется только во время активной фазы кадра данных или команд (не во время передачи кадров пассивного времени). В противном случае тактирование осуществляется вне зависимости от состояния передающего канала
IU	3	Инвертировать вход канала приема Если бит установлен, то значение линии входных данных, которое поступает на фильтр от высокочастотных выбросов, будет предварительно инвертировано
ISS	2	Инвертировать выходы выбора ведомого Установление данного бита изменяет активный уровень для сигналов выбора ведомого с логической единицы на логический ноль. Значение данного бита управления буферизируется на время передачи кадра данных или команды. Это означает, что измененное значение бита ISS будет применено только после завершения передачи текущего кадра

Поле	Биты	Описание
IDTS	1	Инвертировать выход передаваемых данных Если бит установлен, то значение линии выходных данных будет инвертировано
IC	0	Инвертировать выход тактирования Если бит установлен, то значение входного тактового сигнала интерфейса MSB (MicroSecond Bus) будет инвертировано
–	31-29, 27-26, 23-22, 19-18, 15-9, 7-4	Зарезервировано

U_CTRL – Регистр управления каналом приема

Смещение: + 08h
Сброс: 0h

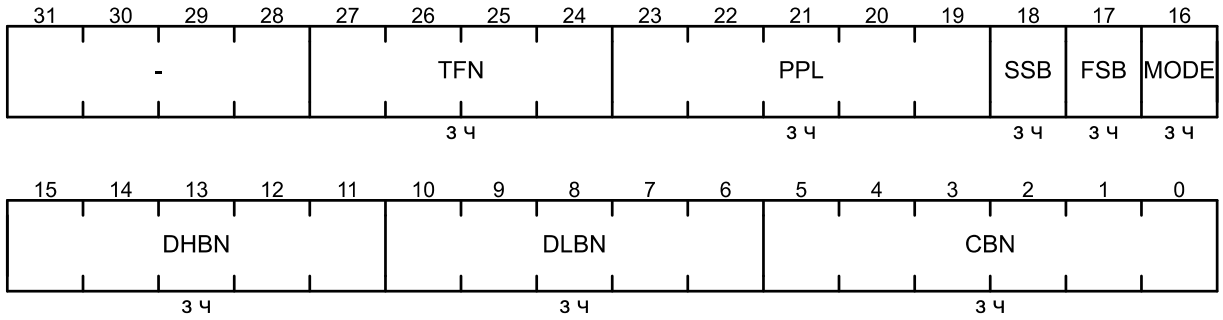


Поле	Биты	Описание
TOC	31-28	Значение счетчика таймера тайм-аута Текущее значение декрементирующегося счетчика таймера тайм-аута
L_ADDR	27-24	Адрес последнего полученного кадра Для кадров
UC	20-16	Счетчик бит канала приема Это поле отображает количество оставшихся бит до ожидаемого завершения кадра
TOV	15-12	Значение таймера тайм-аута приема Значение из данного поля попадает в счетчик таймера тайм-аута канала приема при каждой его перезагрузке (производится автоматически при обнаружении стартового бита или при перезаписи полей TOP, TOV данного регистра). Тайм-аут наступит, если прошло более $(2^{(TOP-1)} * (TOV + 1)$ промежутков времени, которых достаточно для приема одного бита, а таймер не был отключен или перезагружен. Запись в данное поле приводит к перезагрузке таймера тайм-аута

Поле	Биты	Описание	
TOP	11-8	Предварительный делитель таймера тайм-аута приема Определяет во сколько раз медленнее производится декремент таймера тайм-аута чем время прием одного бита (задается значением поля RBR данного регистра). Запись в данное поле приводит к перезагрузке таймера тайм-аута	
		0h	Таймер тайм-аута остановлен
		1h	1
		2h	2
		3h	4
		4h	8
		5h	16
		6h	32
		7h	64
		8h	128
		9h	256
		Ah	512
		Bh	1024
		Ch	2048
		Dh	4096
		Eh	8192
		Fh	16384
PT	3	Тип четности Определяет режим вычисления бита четности в канале приема	
		0	Режим дополнения до четного. Бит четности будет установлен, если суммарное количество установленных бит в полях адреса и данных нечетное
		1	Режим дополнения до нечетного. Бит четности будет установлен, если суммарное количество установленных бит в полях адреса и данных четное
RBR	3-1	Скорость канала приема Значение данного поля задает ожидаемую длительность принимаемых бит	
		000b	Канал приема не функционирует
		001b	Скорость приема в бодах = F_{MSC} / 4
		010b	Скорость приема в бодах = F_{MSC} / 8
		011b	Скорость приема в бодах = F_{MSC} / 16
		100b	Скорость приема в бодах = F_{MSC} / 32
		101b	Скорость приема в бодах = F_{MSC} / 64
		110b	Скорость приема в бодах = F_{MSC} / 128
		111b	Скорость приема в бодах = F_{MSC} / 256
FT	0	Тип кадра Определяет планируемый формат принимаемого кадра	
		0	12-битный кадр
		1	16-битный кадр (содержит 4-битное поле адреса)
–	31-28, 23-21, 7-5	Зарезервировано	

D_CTRL – Регистр управления передающим каналом

Смещение: + 0Ch
Сброс: 0h

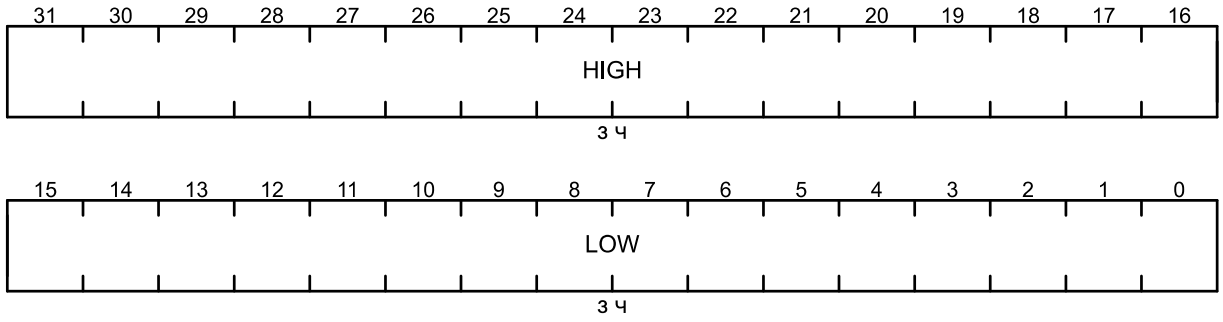


Поле	Биты	Описание	
TFN	27-24	Количество кадров пассивного времени Определяет какое количество кадров пассивного времени будет вставлено между двумя кадрами данных в режиме повторяющейся передачи данных (TM = 1)	
		0h	Без дополнительных кадров пассивного времени
		1h	Вставить 1 кадр пассивного времени
		2h	Вставить 2 кадра пассивного времени
		…	…
		Fh	Вставить 15 кадров пассивного времени
PPL	23-19	Длительность пассивной фазы кадра данных Данное поле определяет длительность пассивной фазы кадра данных	
		00h	Длительность пассивной фазы (2 * t_{FCL})
		01h	Длительность пассивной фазы (2 * t_{FCL})
		02h	Длительность пассивной фазы (2 * t_{FCL})
		03h	Длительность пассивной фазы (3 * t_{FCL})
		…	…
		1Fh	Длительность пассивной фазы (31 * t_{FCL})
SSB	18	Добавить второй бит выбора кадра данных Значение поля определяет будет ли отправлен опциональный бит выбора кадра данных перед передачей старшей части кадра данных	
		0	Не добавлять дополнительный бит
		1	Оправить бит логического 0 перед передачей содержимого старшей части сдвигового регистра
FSB	17	Добавить первый бит выбора кадра данных Значение поля определяет будет ли отправлен опциональный бит выбора кадра данных перед передачей младшей части кадра данных	
		0	Не добавлять дополнительный бит
		1	Оправить бит логического 0 перед передачей содержимого младшей части сдвигового регистра
MODE	16	Режим передачи Данный бит определяет режим работы передающего канала	
		0	Старт передача после программного запроса (Triggered mode)
		1	Режим повторяющейся передачи данных (Data Repetition mode)

Поле	Биты	Описание	
DHBN	15-11	Количество бит старшей части кадра данных Определяет количество бит из поля HIGH регистра D_DATA, которые будут оправлены в активную фазу передачи кадра данных	
		00h	Ни одного бита; вся активная фаза SRH будет пропущена (значения поля FSB данного регистра игнорируется)
		01h	Один младший бит (SRH[0])
		…	…
		10h	Все шестнадцать бит (SRH[0:15])
		> 10h	Значения зарезервированы
DLBN	10-6	Количество бит младшей части кадра данных Определяет количество бит из поля LOW регистра D_DATA, которые будут оправлены в активную фазу передачи кадра данных	
		00h	Ни одного бита; вся активная фаза SRL будет пропущена (значения поля FSB данного регистра игнорируется)
		01h	Один младший бит (SRL[0])
		…	…
		10h	Все шестнадцать бит (SRL[0:15])
		> 10h	Значения зарезервированы
CBN	5-0	Количество бит кадра команды Определяет количество бит из регистра D_CMD, которые будут оправлены в активную фазу передачи кадра команд	
		00h	Ни одного бита
		01h	Один младший бит (SRL[0])
		…	
		10h	Шестнадцать младших бит (SRL[0:15])
		11h	Семнадцать младших бит (SRL[0:15], SRH[0])
		…	…
		20h	Все тридцать два бита (SRL[0:15], SRH[0:15])
		> 20h	Значения зарезервированы
–	31-26, 13-10	Зарезервировано	

D_DATA – Регистр передаваемых данных

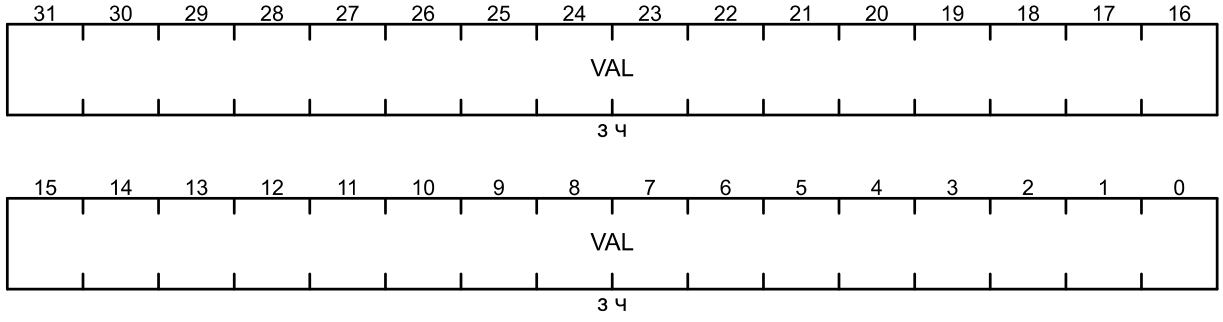
Смещение: + 10h
Сброс: 0h



Поле	Биты	Описание
HIGH	31-16	Биты старшей части Биты, которые будут отправлены в активную фазу сдвига старшей части (SRH) кадра данных
LOW	15-0	Биты младшей части Биты, которые будут отправлены в активную фазу сдвига младшей части (SRL) кадра данных

D_CMD – Регистр передаваемой команды

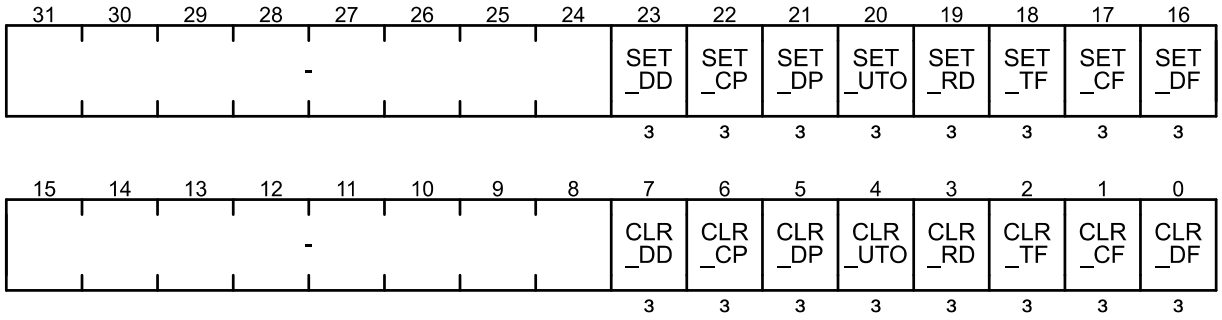
Смещение: + 14h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Значение Биты, которые будут отправлены в активную фазу кадра команды. Каждый доступ к записи в регистр D_CMD инициирует передачу кадра команды через установку бита D_STAT.CP

FLAG_CTRL – Регистр управления флагами

Смещение: + 18h
Сброс: 0h

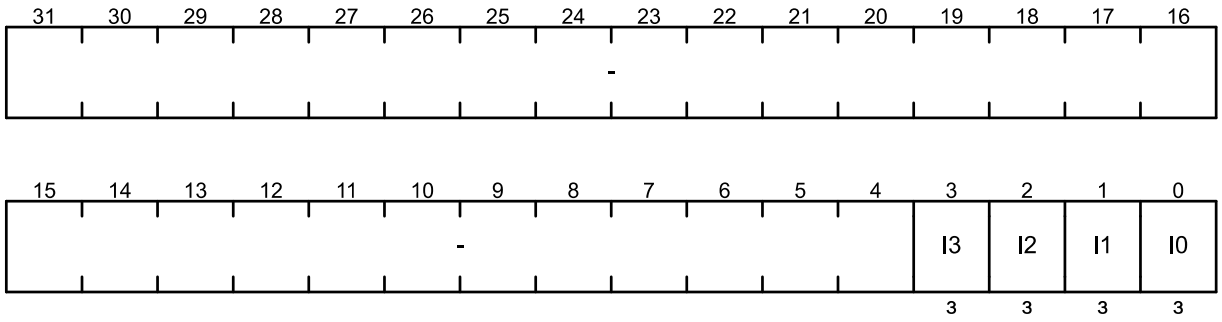


Поле	Биты	Описание
SET_DD	23	Установить бит отключения передающего канала Запись логической 1 приведет к установке бита DD в регистре D_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
SET_CP	22	Установить бит ожидания передачи кадра команды Запись логической 1 приведет к установке бита CP в регистре D_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
SET_DP	21	Установить бит ожидания передачи кадра данных Запись логической 1 приведет к установке бита DP в регистре D_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
SET_UTO	20	Установить флаг тайм-аута канала приема Запись логической 1 приведет к установке бита UTO в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
SET_RD	19	Установить флаг получения кадра данных Запись логической 1 приведет к установке бита RD в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
SET_TF	18	Установить флаг завершения передачи кадра пассивного времени Запись логической 1 приведет к установке бита TF в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
SET_CF	17	Установить флаг завершения передачи кадра команды Запись логической 1 приведет к установке бита CF в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
SET_DF	16	Установить флаг передачи кадра данных Запись логической 1 приведет к установке бита DF в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0

Поле	Биты	Описание
CLR_DD	7	Сбросить бит отключения передающего канала Запись логической 1 приведет к сбросу бита DD в регистре D_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
CLR_CP	6	Сбросить бит ожидания передачи кадра команды Запись логической 1 приведет к сбросу бита CP в регистре D_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
CLR_DP	5	Сбросить бит ожидания передачи кадра данных Запись логической 1 приведет к сбросу бита DP в регистре D_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
CLR_UTO	4	Сбросить флаг тайм-аута канала приема Запись логической 1 приведет к сбросу бита UTO в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
CLR_RD	3	Сбросить флаг получения кадра данных Запись логической 1 приведет к сбросу бита RD в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
CLR_TF	2	Сбросить флаг завершения передачи кадра пассивного времени Запись логической 1 приведет к сбросу бита TF в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
CLR_CF	1	Сбросить флаг завершения передачи кадра команды Запись логической 1 приведет к сбросу бита CF в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
CLR_DF	0	Сбросить флаг передачи кадра данных Запись логической 1 приведет к сбросу бита DF в регистре IRQ_STAT. Запись логического 0 игнорируется. Всегда считывается как 0
–	31-24, 15-8	Зарезервировано

UI_CLR – Регистр сброса битов валидности в записях канала приема

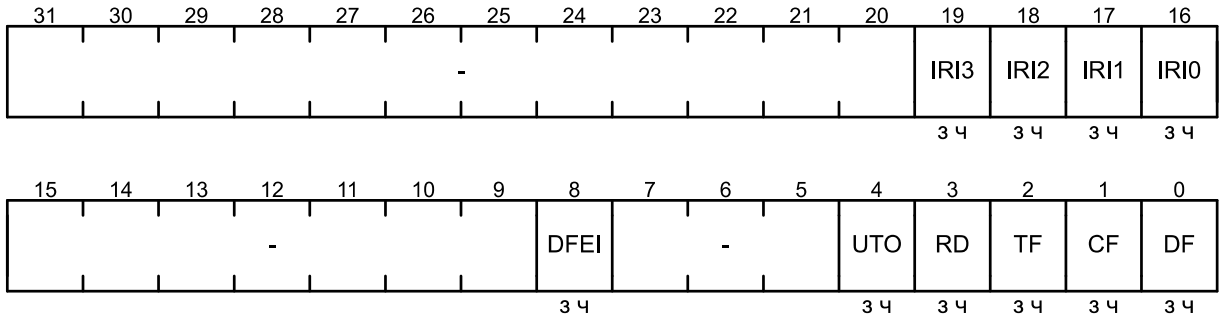
Смещение: + 1Ch
Сброс: 0h



Поле	Биты	Описание
I3	3	Инвалидировать запись 3 Запись логической 1 приведет к сбросу бита валидности в записи 2 канала приема. Запись логического 0 игнорируется. Всегда считывается как 0
I2	2	Инвалидировать запись 2 Запись логической 1 приведет к сбросу бита валидности в записи 2 канала приема. Запись логического 0 игнорируется. Всегда считывается как 0
I1	1	Инвалидировать запись 1 Запись логической 1 приведет к сбросу бита валидности в записи 1 канала приема. Запись логического 0 игнорируется. Всегда считывается как 0
I0	0	Инвалидировать запись 0 Запись логической 1 приведет к сбросу бита валидности в записи 0 канала приема. Запись логического 0 игнорируется. Всегда считывается как 0
–	31-4	Зарезервировано

IRQ_EN – Регистр конфигурации прерываний

Смещение: + 20h
Сброс: 0h

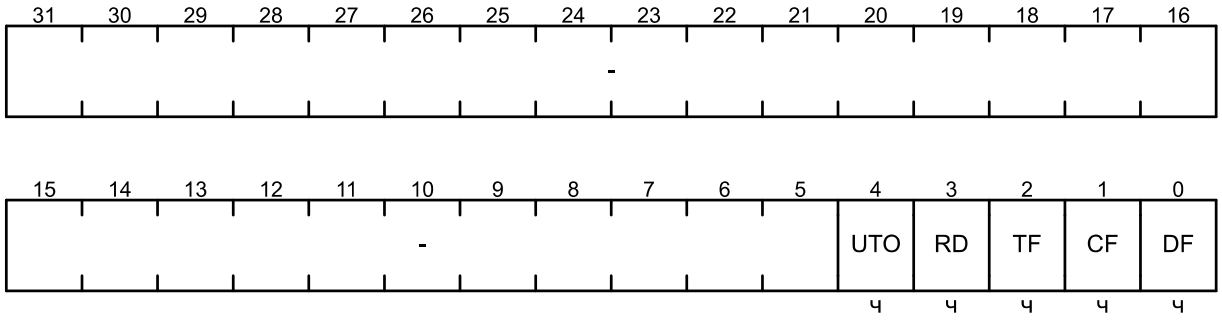


Поле	Биты	Описание	
IRI3	19	Игнорировать получение кадра, который сохраняется в U_ITEM[3] Установка данного бита блокирует установку бита RD в регистре IRQ_STAT при получении кадра, который сохраняется как запись 3 канала приема	
IRI2	18	Игнорировать получение кадра, который сохраняется в U_ITEM[2] Установка данного бита блокирует установку бита RD в регистре IRQ_STAT при получении кадра, который сохраняется как запись 2 канала приема	
IRI1	17	Игнорировать получение кадра, который сохраняется в U_ITEM[1] Установка данного бита блокирует установку бита RD в регистре IRQ_STAT при получении кадра, который сохраняется как запись 1 канала приема	
IRI0	16	Игнорировать получение кадра, который сохраняется в U_ITEM[0] Установка данного бита блокирует установку бита RD в регистре IRQ_STAT при получении кадра, который сохраняется как запись 0 канала приема	
DFEI	8	Раннее прерывание при передаче кадра данных Данный бит управляет моментом установки флага DF в регистре IRQ_STAT. Для того чтобы флаг DF мог быть сформирован необходимо передавать не пустой кадр данных (в регистре D_CTRL хотя бы одно из полей DLBN, DHBN должно содержать не нулевое значение)	
		0	Последний бит данных был выставлен в качестве выходных данных
		1	Первый бит данных был выставлен в качестве выходных данных
UTO	4	Разрешение прерывания при тайм-ауте канала приема Установка данного бита разрешает формирование запроса на прерывание при установке флага UTO в регистре IRQ_STAT	
RD	3	Разрешение прерывания при приеме кадра данных Установка данного бита разрешает формирование запроса на прерывание при установке флага RD в регистре IRQ_STAT	

Поле	Биты	Описание
TF	2	Разрешение прерывания после передачи кадра пассивного времени Установка данного бита разрешает формирование запроса на прерывание при установке флага TF в регистре IRQ_STAT
CF	1	Разрешение прерывания при передаче кадра команды Установка данного бита разрешает формирование запроса на прерывание при установке флага CF в регистре IRQ_STAT
DF	0	Разрешение прерывания при передаче кадра данных Установка данного бита разрешает формирование запроса на прерывание при установке флага DF в регистре IRQ_STAT
–	31-26, 13-10	Зарезервировано

IRQ_STAT – Регистр флагов прерываний

Смещение: + 24h
Сброс: 0h

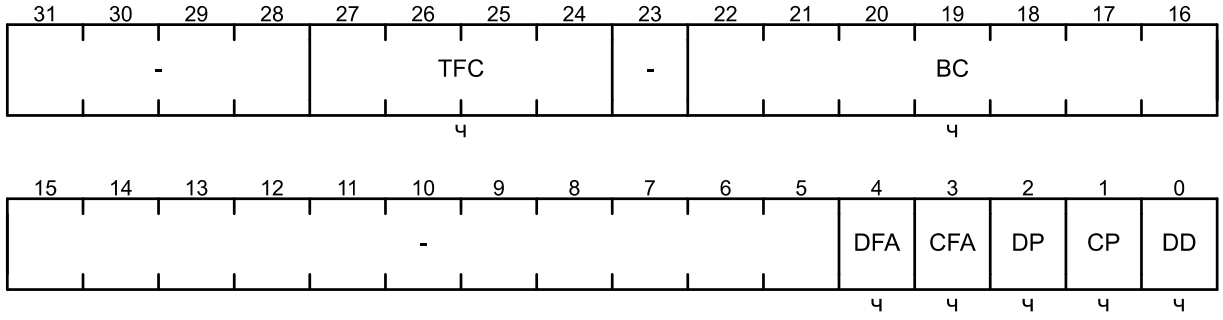


Поле	Биты	Описание
UTO	4	Флаг тайм-аута канала приема Флаг аппаратно устанавливается при достижении условия обратного переполнения счетчика тайм-аута канала приема. Флаг может быть программно установлен или сброшен записью логической 1 в бит SET_UTO или CLR_UTO регистра FLAG_CTRL
RD	3	Флаг приема кадра данных Флаг аппаратно устанавливается при завершении приема кадра данных, адрес которого попадает в один из разрешенных диапазонов. Флаг может быть программно установлен или сброшен записью логической 1 в бит SET_RD или CLR_RD регистра FLAG_CTRL
TF	2	Флаг завершения передачи кадра пассивного времени Флаг аппаратно устанавливается в конце пассивной фазы передачи кадра пассивного времени (присутствуют только в режиме повторяющейся передачи данных при не нулевом значении поля D_CTRL.TFN). Флаг может быть программно установлен или сброшен записью логической 1 в бит SET_TF или CLR_TF регистра FLAG_CTRL

Поле	Биты	Описание
CF	1	Флаг передачи кадра команды Флаг аппаратно устанавливается при завершении активной фазы передачи кадра команды. Флаг может быть программно установлен или сброшен записью логической 1 в бит SET_CF или CLR_CF регистра FLAG_CTRL
DF	0	Флаг передачи кадра данных Флаг аппаратно устанавливается при достижении ожидаемого этапа передачи кадра данных (передачи первого или последнего бита данных). Флаг может быть программно установлен или сброшен записью логической 1 в бит SET_DF или CLR_DF регистра FLAG_CTRL
–	31-5	Зарезервировано

D_STAT – Регистр статуса передающего канала

Смещение: + 28h
Сброс: 0h

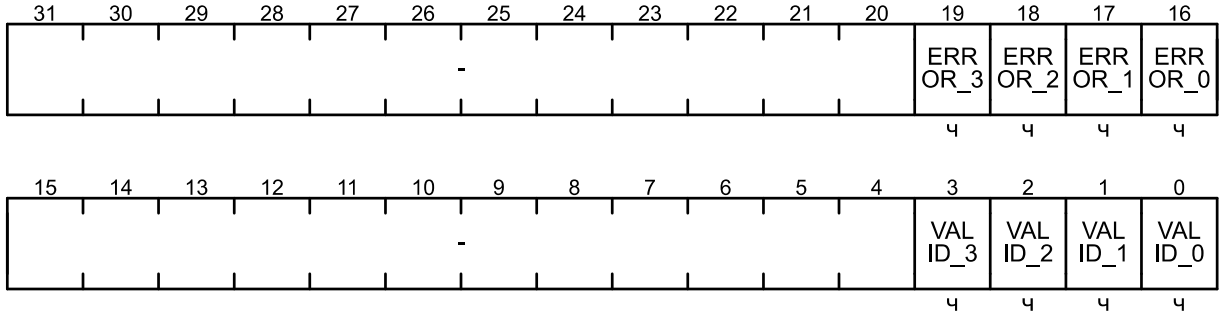


Поле	Биты	Описание	
TFC	27-24	Счетчик кадров пассивного времени В режим повторяющейся передачи данных (Data Repetition mode) это поле показывает номер текущего кадра пассивного времени. В режиме старта передачи после программного запроса (Triggered mode) поле TFC остается равным 0	
		0h	Передается кадр данных (если данное действие не было отложено из-за передачи кадра команды)
		1h	Передается 1-й кадр пассивного времени
		2h	Передается 2-й кадр пассивного времени
		…	…
		Fh	Передается 15-й кадр пассивного времени

Поле	Биты	Описание	
BC	22-16	Счетчик передаваемых бит Это битовое поле указывает количество полных периодов выходного тактового сигнала MSC передающего канала прошло с момента начала текущего кадра. Счетчик сбрасывается в последний такт кадра.	
		00h	Ни одного полного такта не прошло с момента сброса счетчика
		01h	1 полный такт с начала кадра
		02h	2 полных такта с начала кадра
		...	...
		40h	64 полных такта с начала кадра
		> 40h	Значения, превышающие 64, не достижимы. Длительность кадра данных максимальной длины составляет (1 + 16 + 1 + 16 +31) периодов выходного тактового сигнала
DFA	4	Ведется передача кадра данных Установленный бит свидетельствует о том, что ведется передача кадра данных	
CFA	3	Ведется передача кадра команды Установленный бит свидетельствует о том, что ведется передача кадра команды	
DP	2	Ожидается передача кадра данных В режиме старта передачи после программного запроса (D_CTRL.TM равен 0) бит DP устанавливается записью логической 1 в бит FLAG_CTRL.SET_DP. В режим повторяющейся передачи данных бит DP устанавливается аппаратно в последнем кадре пассивного времени. При старте кадра данных данный бит аппаратно сбрасывается	
CP	1	Ожидается передача кадра команды Бит CP устанавливается записью логической 1 в бит FLAG_CTRL.SET_CP. При старте кадра команды данный бит аппаратно сбрасывается	
DD	0	Передающий канал отключен Значение данного бита характеризует состояние передающего канала	
		0	Разрешена работа передающего канала
		1	Передающий канал отключен. Запуск передачи новых кадров не производится. Передача уже начатого кадра будет завершена
–	31-28, 23, 15-5	Зарезервировано	

UI_STAT – Регистр статуса записей канала приема

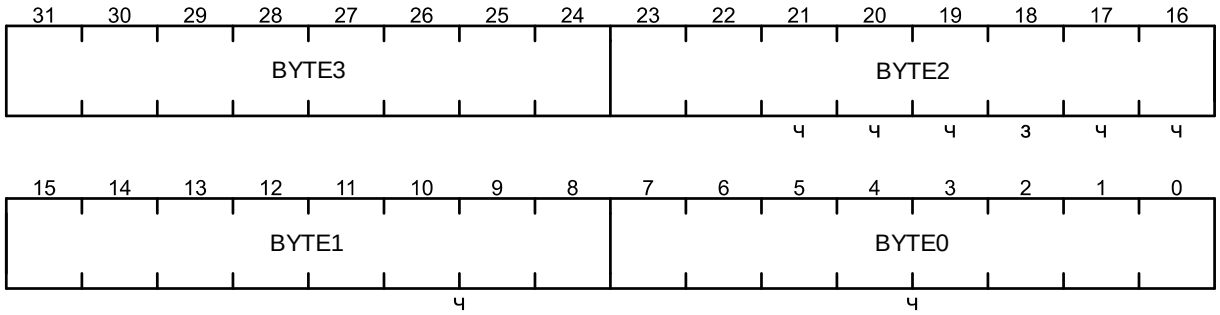
Смещение: + 2Ch
Сброс: 0h



Поле	Биты	Описание
ERROR_3	19	Кадр, сохраненный как запись 3, принят с ошибкой Установленный бит обозначает наличие по крайней мере одного установленного бита ошибки в записи 3
ERROR_2	18	Кадр, сохраненный как запись 2, принят с ошибкой Установленный бит обозначает наличие по крайней мере одного установленного бита ошибки в записи 2
ERROR_1	17	Кадр, сохраненный как запись 1, принят с ошибкой Установленный бит обозначает наличие по крайней мере одного установленного бита ошибки в записи 1
ERROR_0	16	Кадр, сохраненный как запись 0, принят с ошибкой Установленный бит обозначает наличие по крайней мере одного установленного бита ошибки в записи 0
VALID_3	3	Валидность записи 3 Значение бита валидности в записи 3 канала приема
VALID_2	2	Валидность записи 2 Значение бита валидности в записи 2 канала приема
VALID_1	1	Валидность записи 1 Значение бита валидности в записи 1 канала приема
VALID_0	0	Валидность записи 0 Значение бита валидности в записи 0 канала приема
–	31-20, 15-4	Зарезервировано

U_DATA – Регистр данных канала приема

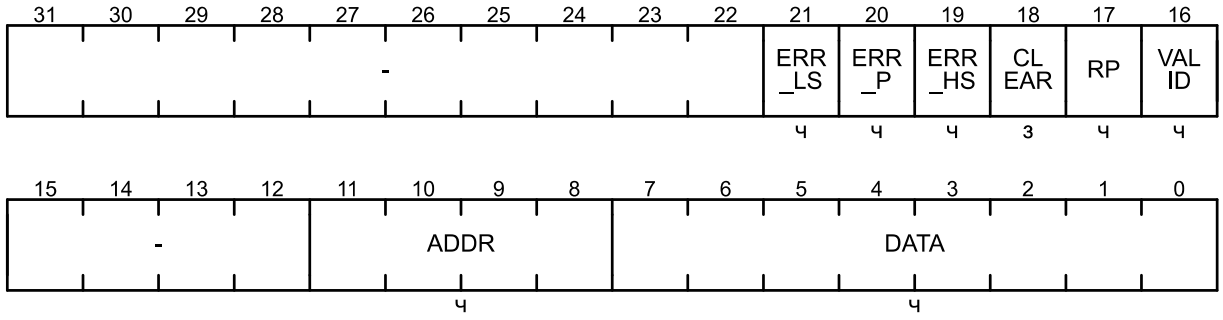
Смещение: + 30h
Сброс: 0h



Поле	Биты	Описание
BYTE3	31-24	Байт данных 3 В данное поле отображается полученный байт данных из U_ITEM[3]
BYTE2	23-16	Байт данных 2 В данное поле отображается полученный байт данных из U_ITEM[2]
BYTE1	15-8	Байт данных 1 В данное поле отображается полученный байт данных из U_ITEM[1]
BYTE0	7-0	Байт данных 0 В данное поле отображается полученный байт данных из U_ITEM[0]

U_ITEM – Регистры записей канала приема

Смещение: U_ITEMS + (4*i)h, где i = 0, …, 3
Сброс: 400h * i, где i = 0, …, 3



Поле	Биты	Описание
ERR_LS	21	Ошибка приема логического нуля в завершающих битах Установленный бит свидетельствует о том, что хотя бы в одном из завершающий бит на момент выборки был уровень логического 0
ERR_P	20	Ошибка бита четности Установленный бит свидетельствует о том, что рассчитанное значение бита четности не совпало с полученным
ERR_HS	19	Ошибка приема логической единицы в стартовом бите Установленный бит свидетельствует о том, что к моменту выборки стартового бита на входе наблюдался уровень логической 1
CLEAR	18	Сбросить бит валидности Запись логической 1 в данный бит приведет к сбросу бита VALID данной записи канала приема. Запись логического 0 игнорируется
RP	17	Значение полученного бита четности Данное содержит значение бита четности в полученном кадре данных
VALID	16	Запись валидна Установка данного бита осуществляется аппаратно после заполнения остальных полей регистра записи канала приема. Бит может быть сброшен записью логической 1 в поле CLEAR или в соответствующую номеру записи позицию регистра UI_CLR. В случае одновременного возникновения аппаратного запроса на установку и программного запроса на сброс, данный бит будет сброшен
ADDR	11-8	Поле адреса Данное поле содержит полученный адрес принимаемого 16-битного кадра данных. Все 12-битные кадры данных попадают в U_ITEM[0] с ADDR равным 0. Старшие биты поля ADDR соответствуют индексу записи канала приема
DATA	7-0	Поле данных Данное поле содержит полученный байт данных
–	31-22, 15-12	Зарезервировано

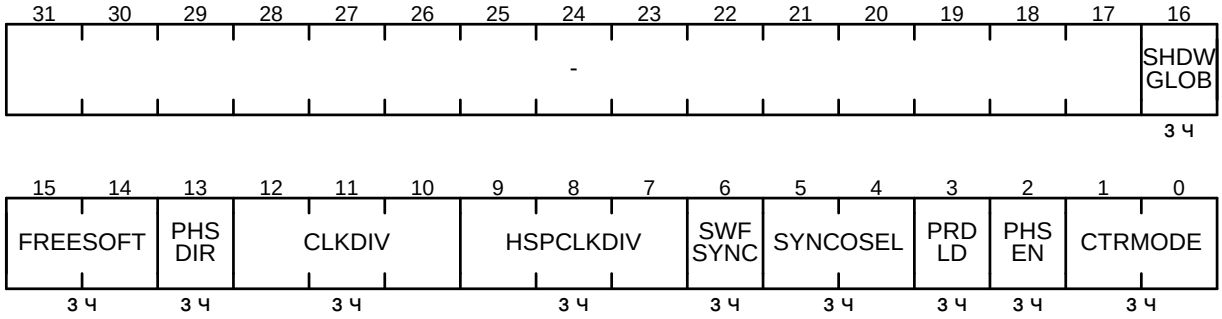
A.12 Регистры блоков ШИМ

Базовый адрес:	5000_D000h	Регистры блока ШИМ 0
	5000_E000h	Регистры блока ШИМ 1
	5000_F000h	Регистры блока ШИМ 2
	5001_0000h	Регистры блока ШИМ 3
	5001_1000h	Регистры блока ШИМ 4
	5001_2000h	Регистры блока ШИМ 5
	5001_3000h	Регистры блока ШИМ 6
	5001_4000h	Регистры блока ШИМ 7
	5001_5000h	Регистры блока ШИМ 8
	5001_6000h	Регистры блока ШИМ 9
	5001_7000h	Регистры блока ШИМ 10
	5001_8000h	Регистры блока ШИМ 11
	5001_9000h	Регистры блока ШИМ 12
	5001_A000h	Регистры блока ШИМ 13
	5001_B000h	Регистры блока ШИМ 14
	5001_C000h	Регистры блока ШИМ 15

TBCTL – регистр управления таймером

Смещение: + 00h

Сброс: 0001_8000h

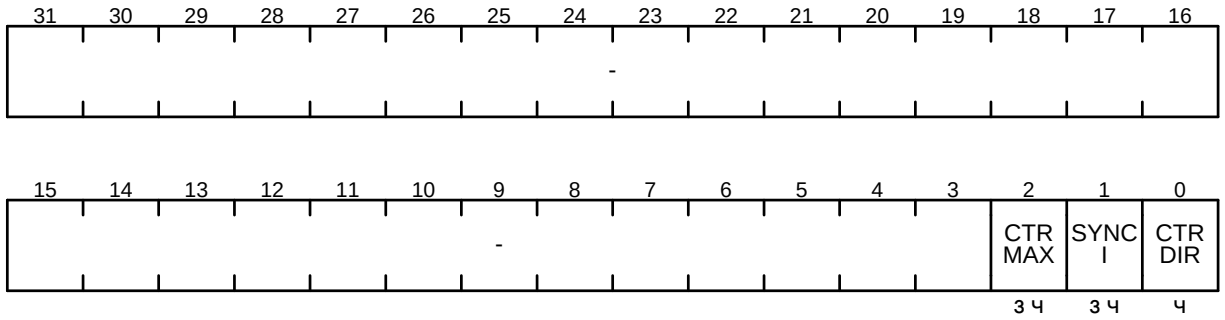


Поле	Биты	Описание	
SHDWGLOB	16	Глобальное разрешение всех теневых загрузок	
		0	Все теневые регистры пишутся, но события перезаписи в активные блокируются
		1	Теневая загрузка происходит согласно настройкам соответствующих полей
FREESOFT	15-14	Поле задания поведения счетчика ШИМ после перехода в режим останова во время отладки	
		00	Счетчик будет остановлен на следующий такт TBCLK
		01	Счетчик будет остановлен по достижении события: - TBCTR = TBPRD (при счете вверх); - TBCTR = 0000h (при счете вниз или вверх-вниз)
		1x	Счетчик продолжит работу
PHSDIR	13	Бит задания фазового направления (используется только при двунаправленном счете). Задает направление счета после синхронизации. Загружается вместе с регистром фазы TBPHS	
		0	Вниз
		1	Вверх

Поле	Биты	Описание	
CLKDIV	12-10	Поле задания первого делителя тактовой частоты	
		000	1
		001	1/2
		010	1/4
		011	1/8
		100	1/16
		101	1/32
		110	1/64
		111	1/128
HSPCLKDIV	9-7	Поле задания второго делителя тактовой частоты. Конечное значение делителя является произведением значений делителей, задаваемых полями CLKDIV и HSPCLKDIV	
		000	1
		001	1/2
		010	1/4
		011	1/6
		100	1/8
		101	1/10
		110	1/12
		111	1/14
SWFSYNC	6	Бит программной эмуляции появления синхроимпульса	
		0	Нет действий
		1	Запись единицы вызывает появление синхроимпульса в цепи PWM_SYNCI
SYNCOSEL	5-4	Поле выбора источника для выходного сигнала синхронизации PWM_SYNCO	
		00	PWM_SYNCI
		01	CTR = 0000h
		10	CTR = CMPB
		11	Выдача синхроимпульса запрещена
PRDLD	3	Бит управления загрузкой регистра TBPRD	
		0	Режим отложенной загрузки регистра TBPRD разрешен
		1	Запись в TBPRD будет произведена сразу в активный регистр
PHSEN	2	Бит разрешения загрузки счетчика таймера	
		0	Запрещено
		1	Разрешена загрузка счетчика TBCTR значением регистра фазы TBPHS при получении события синхронизации (импульс на входе PWM_SYNCI или запись в бит SWFSYNC)
CTRMODE	1-0	Поле задания направления счета	
		00	Вверх
		01	Вниз
		10	Вверх-вниз
		11	Счетчик остановлен
–	31-17	Зарезервировано	

TBSTS – регистр статуса таймера

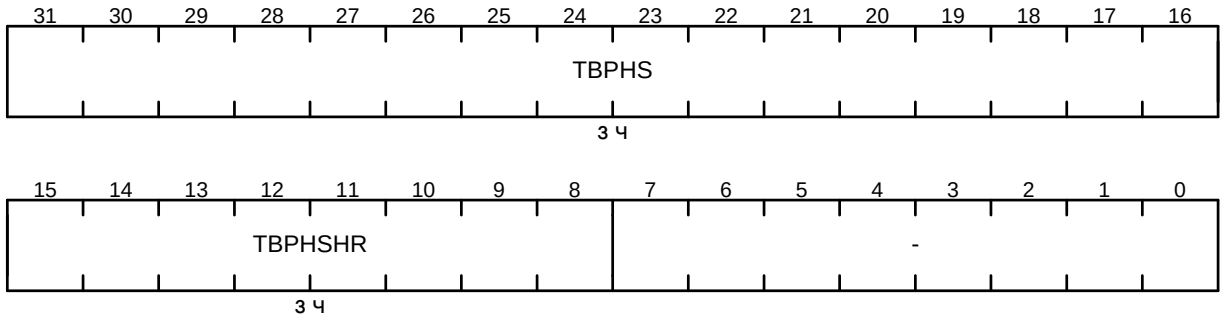
Смещение: + 04h
Сброс: 0h



Поле	Бит	Описание	
CTRMAX	2	Флаг достижения счетчиком таймера своего максимального значения FFFFh	
		0	Значение не достигнуто или флаг был сброшен
		1	Значение было достигнуто
		Запись единицы сбрасывает флаг	
SYNCI	1	Флаг синхронизации	
		0	Синхронизация не достигнута или флаг был сброшен
		1	Синхронизация произошла
		Запись единицы сбрасывает флаг	
CTRDIR	0	Флаг направление счета таймера	
		0	Вниз
		1	Вверх
–	31-3	Зарезервировано	

TBPHS – регистр фазы таймера

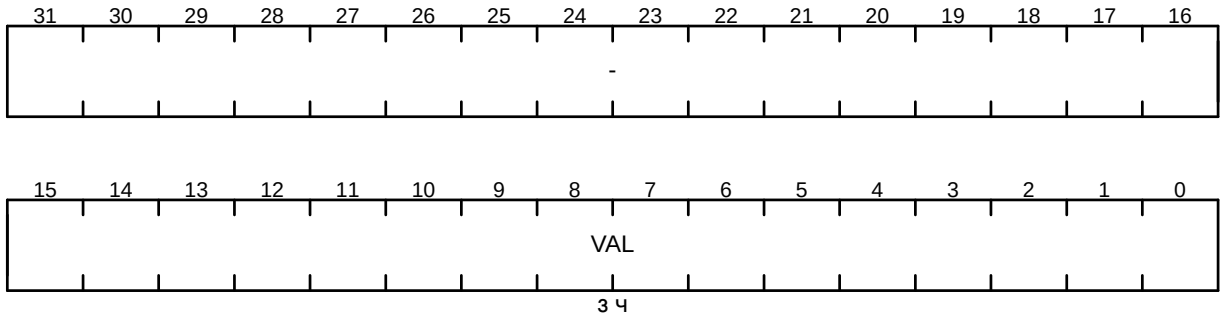
Смещение: + 08h
Сброс: 0h



Поле	Биты	Описание
TBPHS	31-16	Поле задания начальной фазы таймера при получении сигнала синхронизации
TBPHSHR	15-8	Поле дополнительных разрядов начальной фазы таймера. Доступно в блоке ШИМ высокого разрешения. Если бит PHSEN сброшен, то синхронизация отключена, и таймер не будет загружаться значением TBPHS. Если бит PHSEN установлен, то по получении события синхронизации в счетчик таймера TBCTR будет загружено значение TBPHS
–	7-0	Зарезервировано

TBCTR – регистр текущего значения таймера

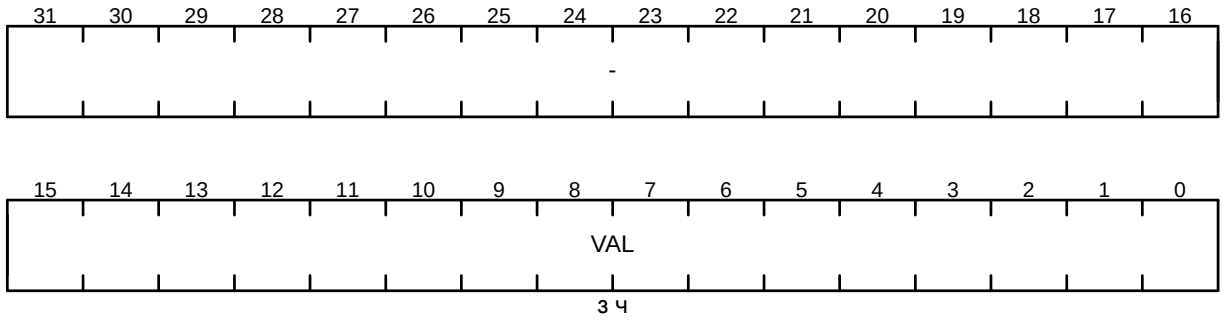
Смещение: + 0Ch
Сброс: 0h



Поле	Биты	Описание
VAL	15-0	Текущее значение счетчика таймера. Запись в регистр изменяет значение таймера. Запись происходит асинхронно с TBLK и не использует отложенный механизм загрузки
–	31-16	Зарезервировано

TBPRD – регистр периода таймера

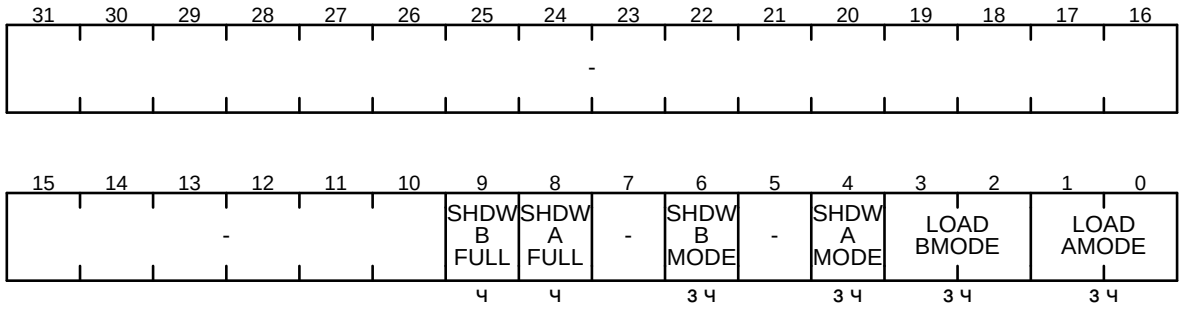
Смещение: + 10h
Сброс: 0h



Поле	Биты	Описание
VAL	15-0	Период таймера (максимальное значение счета таймера). Отложенная загрузка в этот регистр программируется битом PRDLD регистра TBCTL. По умолчанию бит PRDLD сброшен и запись в регистр TBPRD приводит к записи в теневой регистр. Активный регистр будет загружен по событию TBCTR = Zero. Если бит PRDLD установлен, то запись выполняется напрямую в активный регистр
–	31-16	Зарезервировано

CMPCTL – регистр управления компаратором

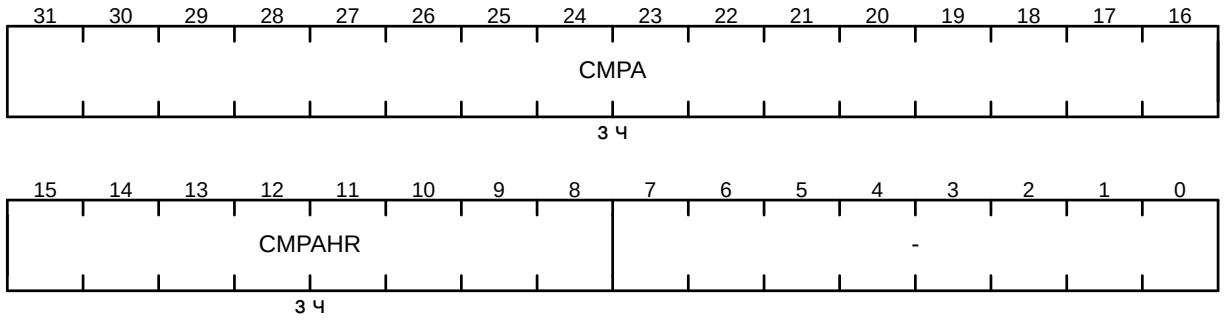
Смещение: + 14h
Сброс: 0h



Поле	Биты	Описание	
SHDWBFULL	9	Флаг отложенной загрузки в регистр CMPB	
		0	Нет действий
		1	Активный регистр загружен значением из теневого регистра
SHDWAFULL	8	Флаг отложенной загрузки в регистр CMPA	
		0	Нет действий
		1	Активный регистр загружен значением из теневого регистра
SHDWBMODE	6	Бит управления загрузкой регистра CMPB	
		0	Значение, записываемое в регистр CMPB, размещается в теневом регистре (отложенная загрузка)
		1	Производится загрузка напрямую в активный регистр
SHDWAMODE	4	Бит управления загрузкой регистра CMPA	
		0	Значение, записываемое в регистр CMPA, размещается в теневом регистре (отложенная загрузка)
		1	Производится загрузка напрямую в активный регистр
LOADBMODE	3-2	Поле задания события загрузки отложенного значения в регистр CMPB (при условии, что бит SHDWBMODE сброшен)	
		00	CTR = Zero
		01	CTR = PRD
		10	CTR = Zero или CTR = PRD
		11	Загрузка запрещена
LOADAMODE	1-0	Поле задания события загрузки отложенного значения в регистр CMPA (при условии, что бит SHDWAMODE сброшен)	
		00	CTR = Zero
		01	CTR = PRD
		10	CTR = Zero или CTR = PRD
		11	Загрузка запрещена
–	31-10, 7, 5	Зарезервировано	

CMPA – регистр порога срабатывания A

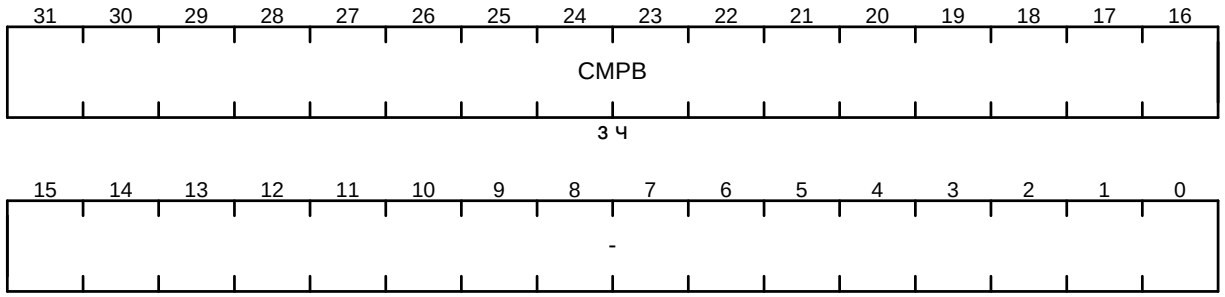
Смещение: + 18h
Сброс: 0h



Поле	Биты	Описание
CMPA	31-16	Активное значение порога срабатывания канала A, которое сравнивается со значением счетчика таймера. При совпадении значений формируется событие CTR = CMPA, которое влияет на поведение сигналов на линиях PWMx_A и PWMx_B
CMPAHR	15-8	Дополнительные младшие биты значения порога срабатывания канала A (используются только для блока ШИМ высокого разрешения). Отложенная загрузка включается и работает также как и для поля CMPA
–	7-0	Зарезервировано

CMPB – регистр порога срабатывания B

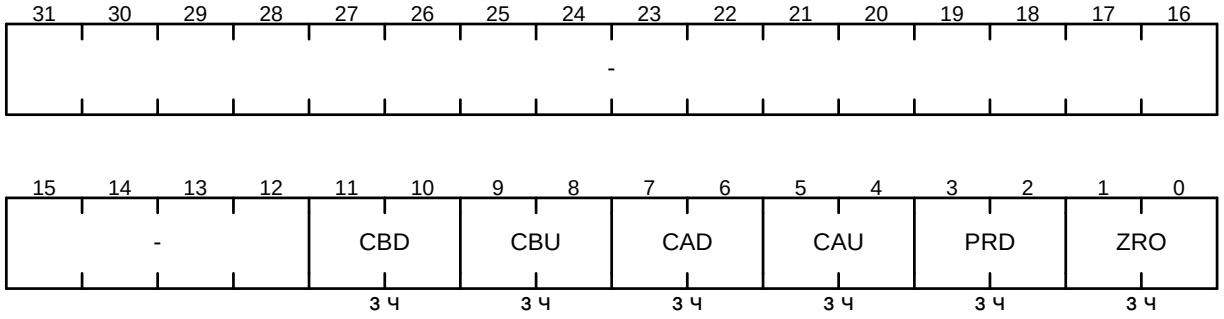
Смещение: + 1Ch
Сброс: 0h



Поле	Биты	Описание
CMPB	31-16	Активное значение порога срабатывания канала B, которое сравнивается со значением счетчика таймера. При совпадении значений формируется событие CTR = CMPB, которое влияет на поведение сигналов на линиях PWMx_A и PWMx_B
–	15-0	Зарезервировано

AQCTLA – регистр обработчика для выхода A

Смещение: + 20h
Сброс: 0h

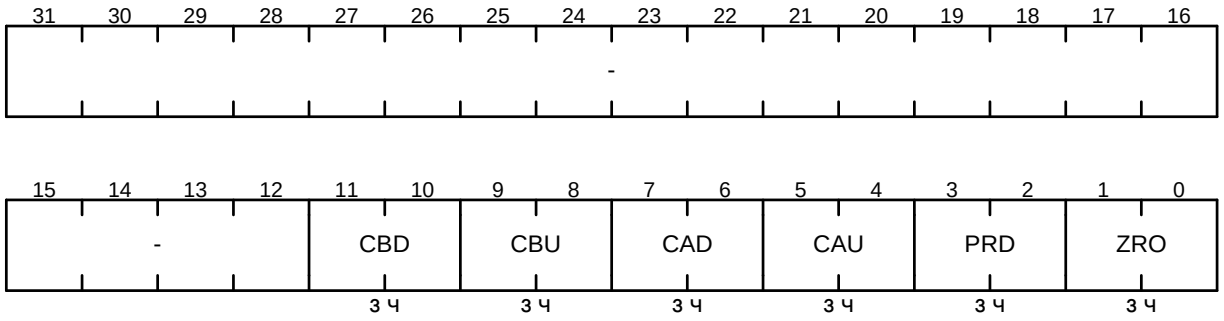


Поле	Биты	Описание
CBD	11-10	Действие на выводе PWMx_A при CTR = CMPB при счете вниз
CBU	9-8	Действие на выводе PWMx_A при CTR = CMPB при счете вверх
CAD	7-6	Действие на выводе PWMx_A при CTR = CMPA при счете вниз
CAU	5-4	Действие на выводе PWMx_A при CTR = CMPA при счете вверх
PRD	3-2	Действие на выводе PWMx_A при CTR = PRD
ZRO	1-0	Действие на выводе PWMx_A при CTR = Zero
–	31-12	Зарезервировано

Примечание – Для каждого события может быть задано одно из четырех действий:
00 – нет реакции;
01 – переключение вывода PWMx_A в ноль;
10 – переключение вывода PWMx_A в единицу;
11 – инверсия вывода PWMx_A.

AQCTLB – регистр обработчика для выхода B

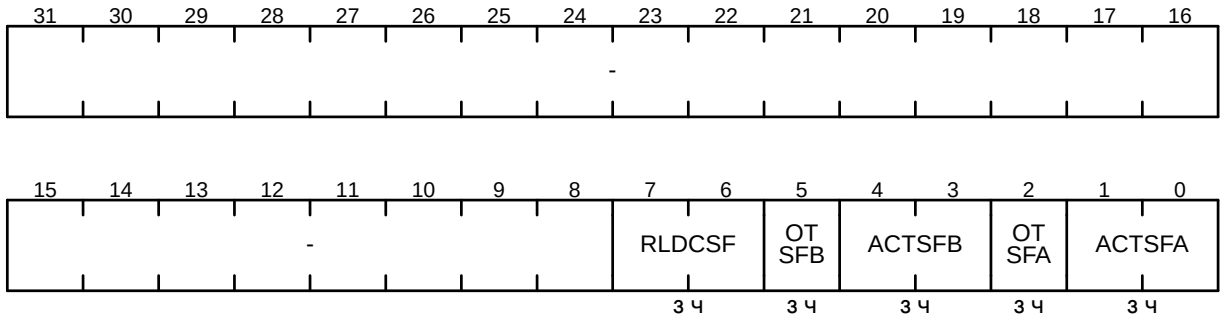
Смещение: + 24h
Сброс: 0h



Примечание – Назначения битовых полей аналогичны назначению полей регистра AQCTLA, с той разницей, что относятся они к выводу PWMx_B.

AQSFRC – регистр программного управления однократным действием

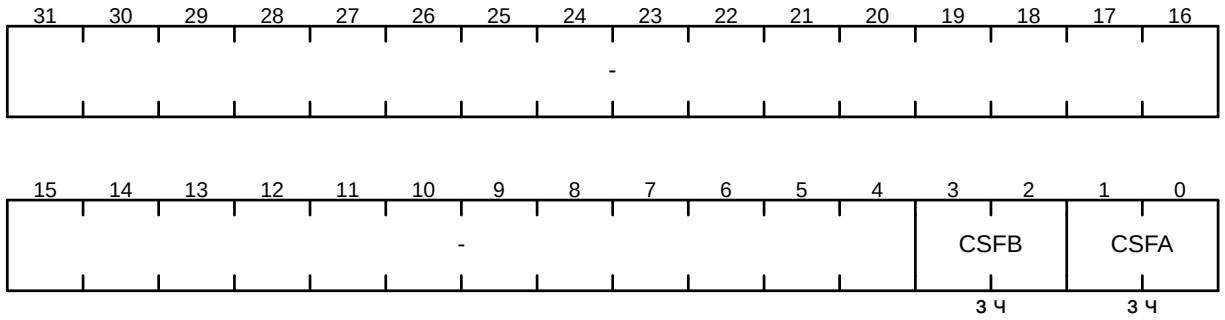
Смещение: + 28h
Сброс: 0h



Поле	Биты	Описание	
RLDCSF	7-6	Управление событием теневой загрузки полей OTSFA, OTSFB, а также полей CSFA, CSFB регистра AQCSFRC	
		00	CTR = Zero
		01	CTR = PRD
		10	CTR = Zero или CTR = PRD
		11	Без теневой загрузки, прямая запись в регистр
OTSFB	5	Запись единицы приводит к однократному переключению вывода в состояние, согласно ACTSFB	
ACTSFB	4-3	Выбор действия с выходным сигналом на выводе	
		00	Нет действий
		01	PWMx_B = 0
		10	PWMx_B = 1
		11	Инверсия вывода PWMx_B
OTSFA	2	Запись единицы приводит к однократному переключению вывода в состояние, согласно ACTSFA	
ACTSFA	1-0	Выбор действия с выходным сигналом на выводе	
		00	Нет действий
		01	PWMx_A = 0
		10	PWMx_A = 1
		11	Инверсия вывода PWMx_A
–	31-8	Зарезервировано	

AQCSFRC – регистр обработчика для циклического программного управления

Смещение: + 2Ch
Сброс: 0h

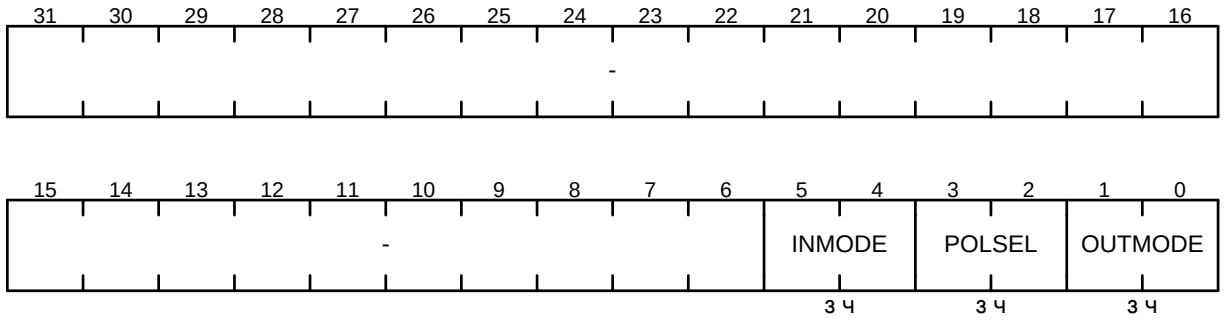


Поле	Биты	Описание
CSFB/ CSFA	3-2/ 1-0	Поле задания циклического воздействия на выход PWMx_B/PWMx_A
–	31-4	Зарезервировано

Примечание – Может быть задано одно из четырех воздействий:
00, 11 – нет реакции;
01 – значение 0 на выходе PWMx_B/PWMx_A;
10 – значение 1 на выходе PWMx_B/PWMx_A;

DBCTL – регистр управления генератором «мертвого» времени ШИМ

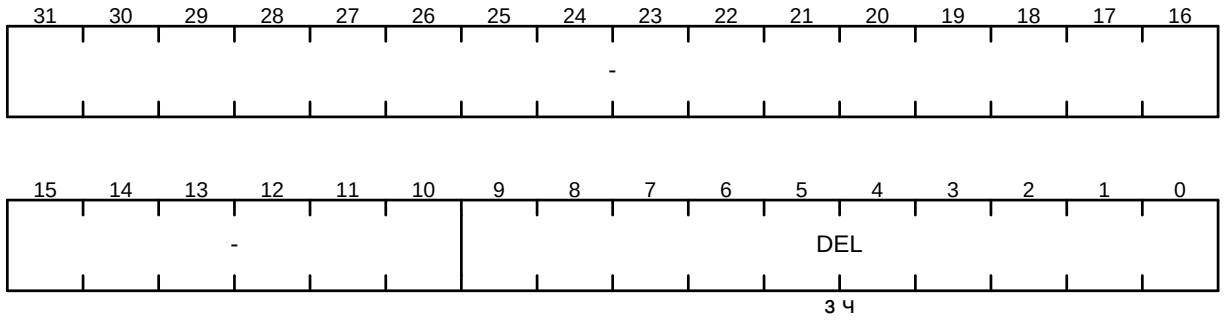
Смещение: + 30h
Сброс: 0h



Поле	Биты	Описание	
INMODE	5-4	Поле выбора источника для контроля по фронту и срезу. Старший бит поля управляет ключом S5, младший – ключом S4, см. рисунок 20.12	
		00b	Входной сигнал PWMx_A используется для контроля по переднему и заднему фронтам
		01b	Входной сигнал PWMx_A используется для контроля по заднему фронту, а сигнал PWMx_B – по переднему
		10b	Входной сигнал PWMx_A используется для контроля по переднему фронту, а сигнал PWMx_B – по заднему
		11b	Входной сигнал PWMx_B используется для контроля по переднему и заднему фронтам
POLSEL	3-2	Поле задания полярности сигнала на выходе. Старший бит поля управляет ключом S3, а младший – ключом S2, см. рисунок 20.12	
		00b	Инверсия запрещена
		01b	Инверсия только на выходе PWMx_A
		10b	Инверсия только на выходе PWMx_B
		11b	Инверсия на выходах PWMx_A и PWMx_B
OUTMODE	1-0	Поле выбора фронта, для которого включена задержка («мертвое» время). Старший бит поля управляет ключом S1, а младший – ключом S0, см. рисунок 20.12	
		00b	Не задано
		01b	Задний фронт выхода PWMx_B
		10b	Передний фронт выхода PWMx_A
		11b	Передний фронт выхода PWMx_A и задний фронт выхода PWMx_B
–	31-6	Зарезервировано	

DBRED – регистр управления «мертвым» временем переднего фронта

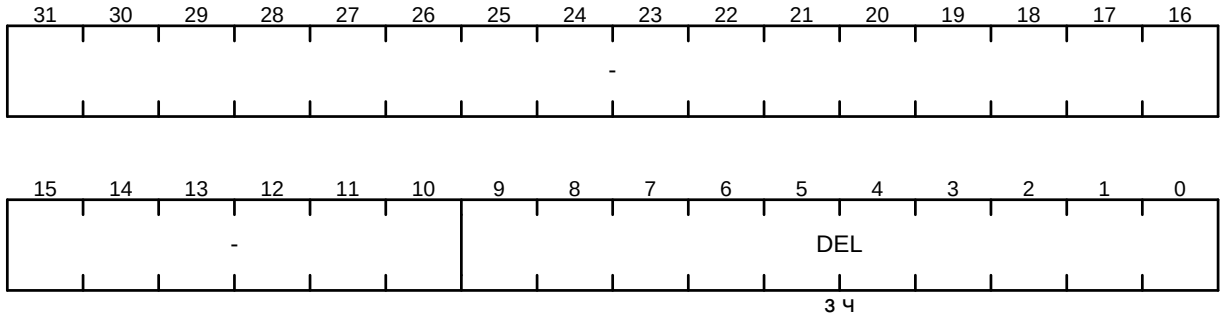
Смещение: + 34h
Сброс: 0h



Поле	Биты	Описание
DEL	9-0	Величина задержки переднего фронта для генератора «мертвого» времени ШИМ (в периодах тактового сигнала PCLK)
–	31-10	Зарезервировано

DBFED – регистр управления «мертвым» временем заднего фронта

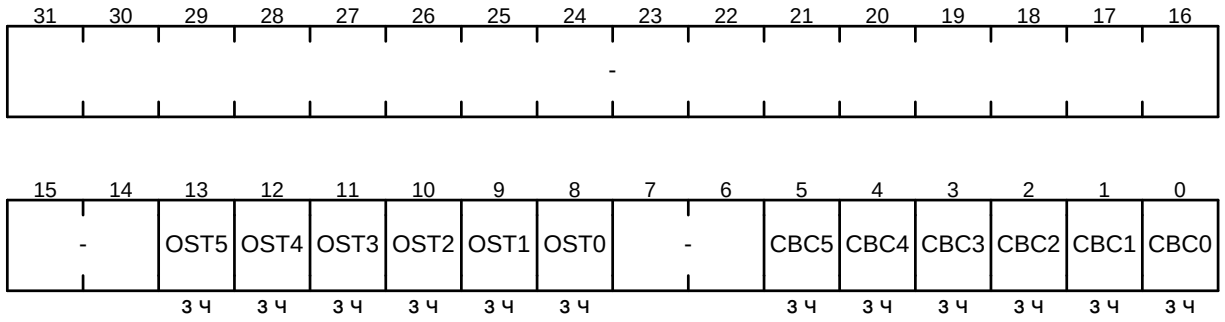
Смещение: + 38h
Сброс: 0h



Поле	Биты	Описание
DEL	9-0	Величина задержки заднего фронта для генератора «мертвого» времени ШИМ (в периодах тактового сигнала PCLK)
–	31-10	Зарезервировано

TZSEL – регистр источника сигнала аварии

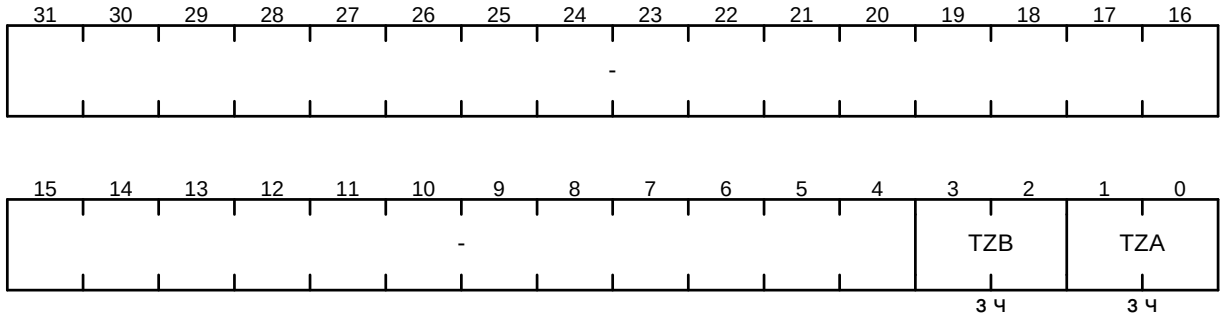
Смещение: + 3Ch
Сброс: 0h



Поле	Биты	Описание	
OSTn	13-8	Бит разрешения источника сигнала аварии с вывода PWM_TZn (n от 0 до 5) в однократном режиме	
		0	Запрещено
		1	Разрешено
CBCn	5-0	Бит разрешения источника сигнала аварии с вывода PWM_TZn (номер вывода от 0 до 5) в циклическом режиме	
		0	Запрещено
		1	Разрешено
–	31-14, 7-6	Зарезервировано	

TZCTL – регистр управления детектором сигнала аварии

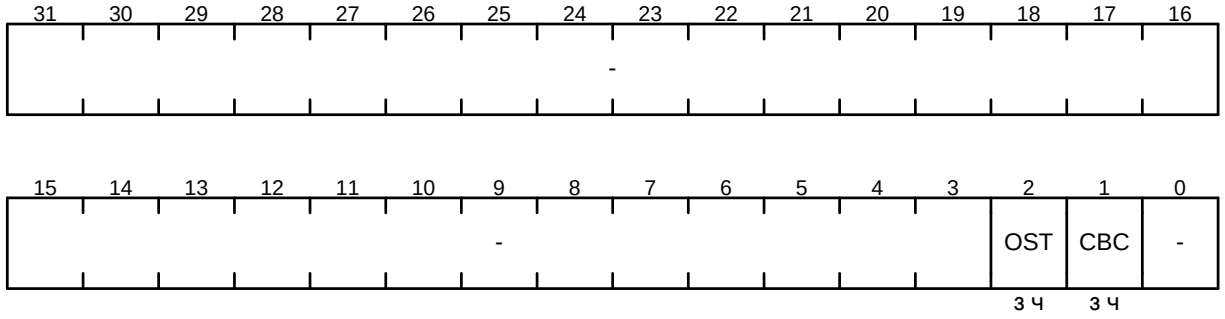
Смещение: + 40h
Сброс: 0h



Поле	Биты	Описание	
TZB/ TZA	3-2/ 1-0	Поле задания поведения вывода PWMx_B/PWMx_A в случае получения сигнала аварии. Источник сигнала аварии при этом определяется регистром TZSEL	
		00b	Переключение в третье состояние
		01b	Переключение в единицу
		10b	Переключение в ноль
		11b	Нет действий
–	31-4	Зарезервировано	

TZEINT – регистр маски прерывания детектора сигнала аварии

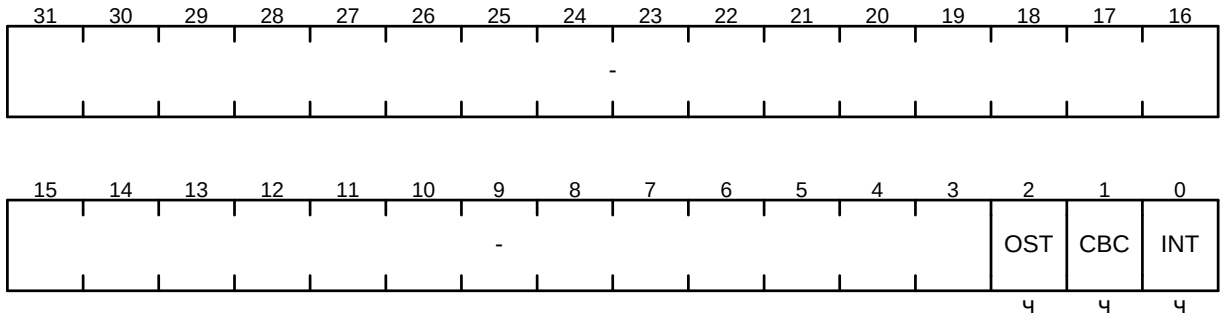
Смещение: + 44h
Сброс: 0h



Поле	Биты	Описание	
OST	2	Бит разрешения генерации прерывания в однократном режиме обработки аварии	
		0	Запрещено
		1	Разрешено
CBC	1	Бит разрешения генерации прерывания в циклическом режиме обработки аварии	
		0	Запрещено
		1	Разрешено
–	31-3, 0	Зарезервировано	

TZFLG – регистр флагов прерывания детектора сигнала аварии

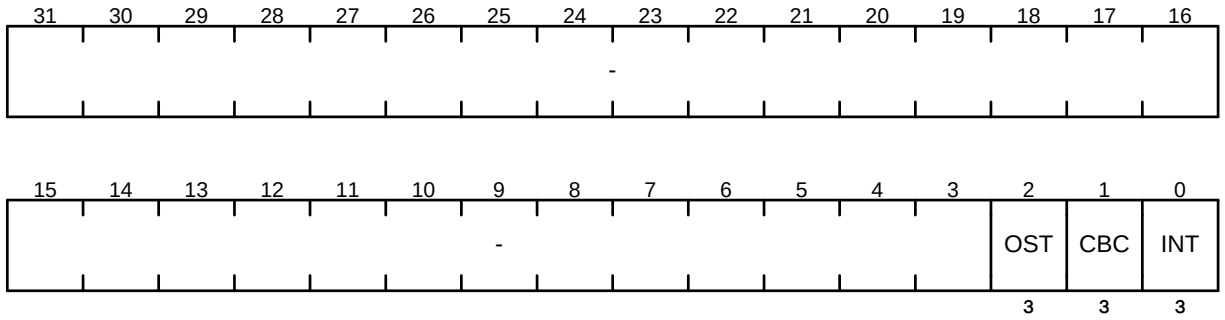
Смещение: + 48h
Сброс: 0h



Поле	Биты	Описание	
OST	2	Флаг прерывания в однократном режиме обработки аварии	
		0	Нет прерывания
		1	Запрос на прерывание
		При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова	
CBC	1	Флаг прерывания в циклическом режиме обработки аварии	
		0	Нет прерывания
		1	Запрос на прерывание
		При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова	
INT	0	Флаг внешнего прерывания PLIC	
		0	Нет прерывания
		1	Запрос на прерывание
		Если флаг был сброшен, а один из флагов CBC или OST установлен, флаг установится снова	
–	31-3	Зарезервировано	

TZCLR – регистр сброса флагов прерываний детектора сигнала аварии

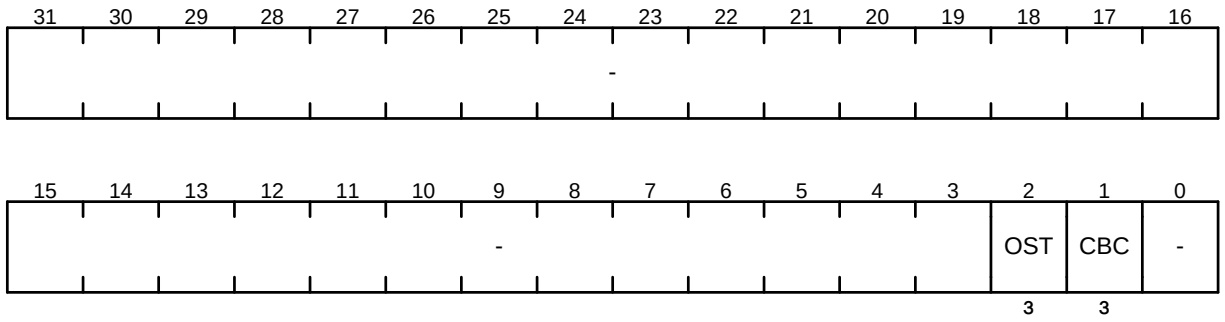
Смещение: + 4Ch
Сброс: 0h



Поле	Биты	Описание
OST	2	Бит сброса флага прерывания в однократном режиме обработки аварии. Запись единицы сбрасывает бит OST в регистре TZFLG
CBC	1	Бит сброса флага прерывания в циклическом режиме обработки аварии. Запись единицы сбрасывает бит CBC в регистре TZFLG
INT	0	Бит сброса флага внешнего прерывания PLIC. Запись единицы сбрасывает бит INT в регистре TZFLG
–	31-3	Зарезервировано

TZFRC – регистр программной эмуляции сигнала аварии

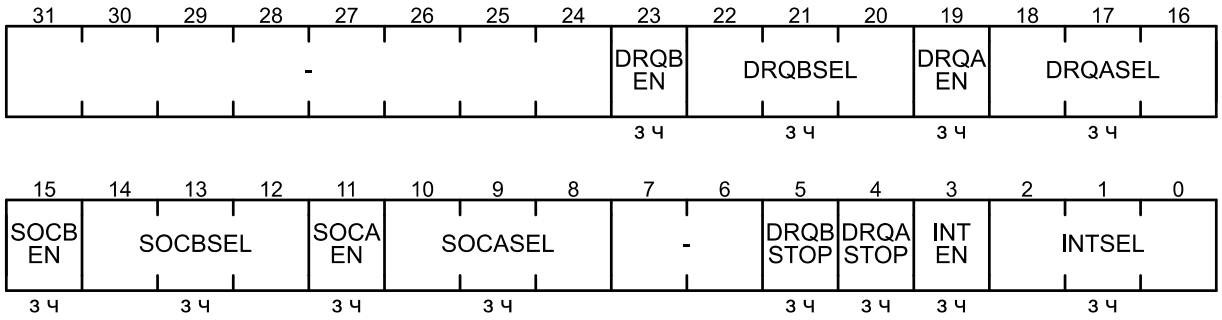
Смещение: + 50h
Сброс: 0h



Поле	Биты	Описание
OST	2	Бит программной генерации сигнала аварии в однократном режиме. Запись единицы устанавливает бит OST в регистре TZFLG. Чтение бита возвращает ноль
CBC	1	Бит программной генерации сигнала аварии в циклическом режиме. Запись единицы устанавливает бит CBC в регистре TZFLG. Чтение бита возвращает ноль
–	31-3, 0	Зарезервировано

ETSEL – регистр источника триггера событий

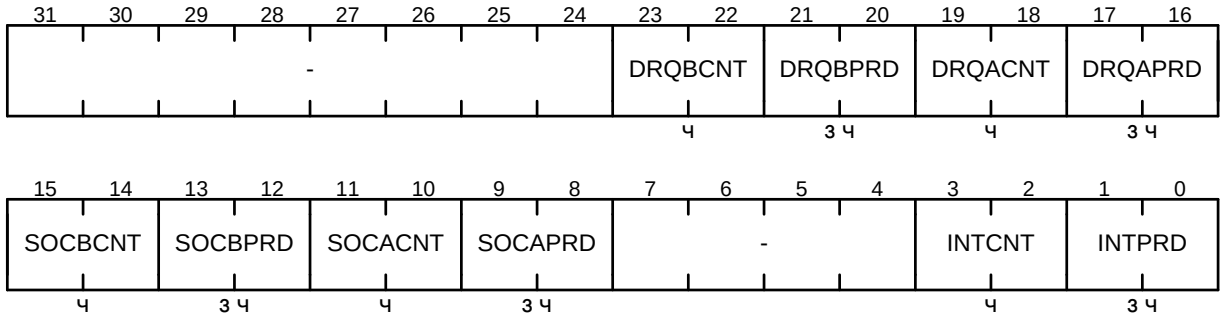
Смещение: + 54h
Сброс: 0h



Поле	Биты	Описание	
DRQBEN	23	Бит разрешения генерации запроса DMA RX (PWM_DRQB)	
		0	Запрещено
		1	Разрешено
DRQAEN	19	Бит разрешения генерации запроса DMA TX (PWM_DRQA)	
		0	Запрещено
		1	Разрешено
SOCBEN/ SOCAEN	15/ 11	Бит разрешения генерации внешнего сигнала PWM_SOCB/ PWM_SOCA для запуска АЦП	
		0	Запрещено
		1	Разрешено
DRQBSEL/ DRQASEL/ SOCBSEL/ SOCASEL/ INTSEL	22-20/ 18-16/ 14-12/ 10-8/ 2-0	Поле выбора события, по которому будет сформирован импульс DMA_REQB/DMA_REQA/PWM_SOCB/PWM_SOCA/PWM_INT	
		000	CTR = CMPA независимо от направления счета
		001	CTR = Zero
		010	CTR = PRD
		011	CTR = CMPB независимо от направления счета
		100	CTR = CMPA при счете вверх
		101	CTR = CMPA при счете вниз
		110	CTR = CMPB при счете вверх
		111	CTR = CMPB при счете вниз
DRQBSTOP	5	Автоматическая очистка запроса DMA RX (PWM B), когда канал завершает все транзакции	
DRQASTOP	4	Автоматическая очистка запроса DMA TX (PWM A), когда канал завершает все транзакции	
INTEN	3	Бит разрешения генерации внешнего прерывания PWM_INT	
		0	Запрещено
		1	Разрешено
–	31-24, 7-6	Зарезервировано	

ETPS – регистр предделителя триггера событий

Смещение: + 58h
Сброс: 0h

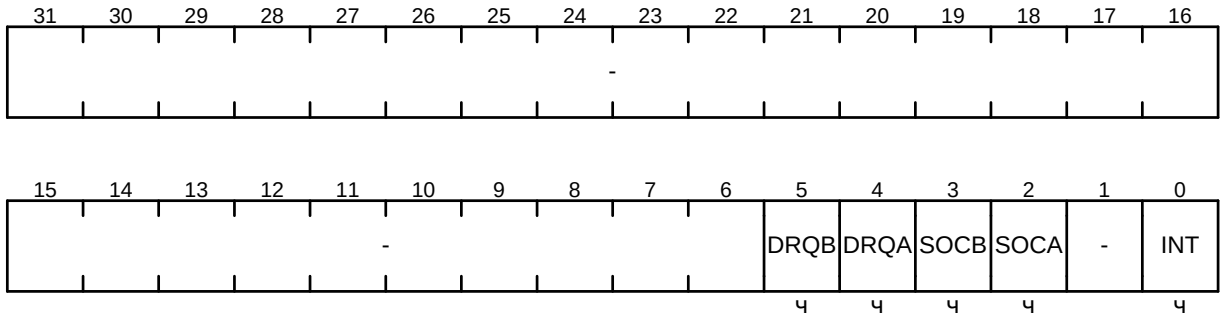


Поле	Биты	Описание	
DRQBCNT/ DRQACNT/ SOCBCNT/ SOCACNT	23-22/ 19-18/ 15-14/ 11-10	Счетчик событий сигнала DMA_REQB/DMA_REQA/PWM_SOCB/ PWM_SOCA. Счетчик автоматически сбрасывается, когда сформировано прерывание и перестает считать, когда достигает значения DRQBPRD/ DRQAPRD/SOCBPRD/SOCAPRD соответственно.	
DRQBPRD/ DRQAPRD	21-20/ 17-16	Поле задания количества событий, выбранных полем DRQBSEL/ DRQASEL регистра ETSEL, по которым будет сформирован сигнал запуска DMA_REQB/DMA_REQA. Для разрешения генерации сигнала нужно установить бит DRQBEN/ DRQAEN регистра ETSEL. Сигнал будет сформирован, даже если флаг DRQB/DRQA (регистр ETFLG) предыдущего сигнала не был сброшен. Как только сигнал DMA_REQB/DMA_REQA отправлен, счетчик DRQBCNT/DRQACNT автоматически сбрасывается	
		00	Выдача сигнала по каждому событию
		01	Выдача сигнала каждые два события
		10	Выдача сигнала каждые три события
		11	Выдача сигнала каждые четыре события
SOCBPRD/ SOCAPRD	13-12/ 9-8	Поле задания количества событий, заданных полем SOCBSEL/ SOCASEL регистра ETSEL, по которым будет сформирован сигнал запуска АЦП PWM_SOCB/PWM_SOCA. Для разрешения генерации сигнала нужно установить бит SOCBEN/ SOCAEN регистра ETSEL. Сигнал будет сформирован, даже если флаг SOCB/SOCA (регистр ETFLG) предыдущего сигнала не был сброшен. Как только сигнал PWM_SOCB/PWM_SOCA отправлен, счетчик SOCBCNT/SOCACNT автоматически сбрасывается	
		00	Выдача сигнала по каждому событию
		01	Выдача сигнала каждые два события
		10	Выдача сигнала каждые три события
		11	Выдача сигнала каждые четыре события
INTCNT	3-2	Значение счетчика событий прерываний. Счетчик автоматически сбрасывается, когда сформировано прерывание и перестает считать, когда достигает значения INTPRD	

Поле	Биты	Описание	
INTPRD	1-0	Поле задания количества событий, выбранных полем INTSEL регистра ETSEL, по которым будет сформировано внешнее прерывание PWM_INT. Для разрешения генерации прерывания нужно установить бит INTEN в регистре ETSEL. Если флаг прерывания INT (регистр ETFLG) установлен от предыдущего прерывания, то текущее прерывание не будет активировано до сброса этого флага (сбрасывается записью единицы в бит INT регистра ETCLR). Такой механизм позволяет обрабатывать одно прерывание, в то время как другое ждет своей очереди	
		00	Прерывания по каждому событию (INTCNT = 00b)
		01	Прерывания каждые два события (INTCNT = 01b)
		10	Прерывания каждые три события (INTCNT = 10b)
		11	Прерывания каждые четыре события (INTCNT = 11b)
–	31-24, 7-4	Зарезервировано	

ETFLG – регистр флагов триггера событий

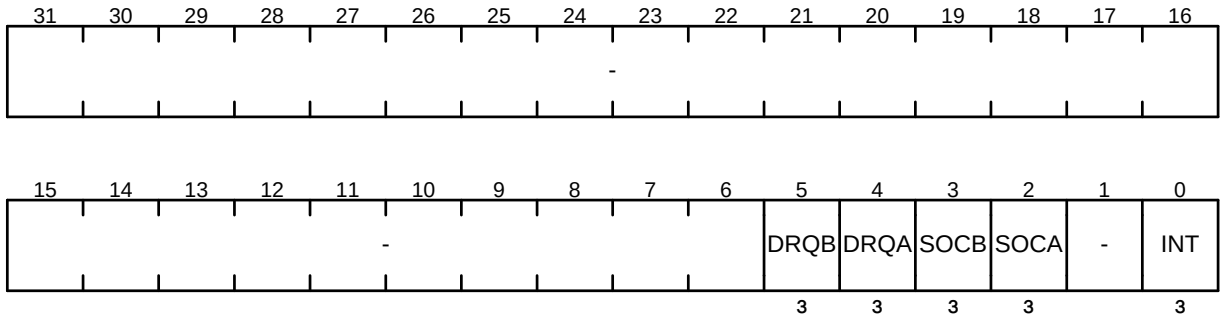
Смещение: + 5Ch
Сброс: 0h



Поле	Биты	Описание	
DRQB/ DRQA	5/ 4	Флаг запроса DMA_REQB/DMA_REQA	
		0	Не установлено или сброшено
		1	Установлено
SOCB/ SOCA/ INT	3/ 2/ 0	Флаг внешнего сигнала АЦП PWM_SOCB/PWM_SOCA/PWM_INT	
		0	Не установлено или сброшено
		1	Установлено
–	31-6, 1	Зарезервировано	

ETCLR – регистр сброса флагов триггера событий

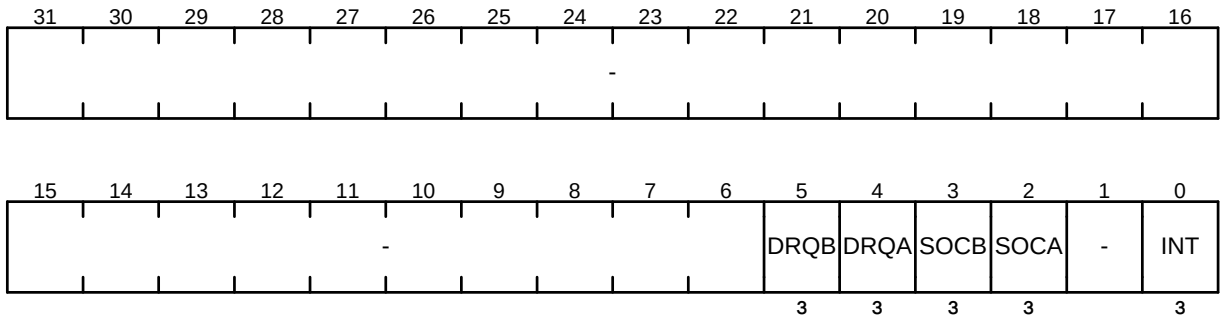
Смещение: + 60h
Сброс: 0h



Поле	Биты	Описание	
DRQB/ DRQA	5/ 4	Бит сброса флага DRQB/DRQA в регистре ETFLG	
		0	Нет действий
		1	Запись единицы сбрасывает флаг
SOCB/ SOCA/ INT	3/ 2/ 0	Бит сброса флага SOCB/SOCA/INT в регистре ETFLG	
		0	Нет действий
		1	Запись единицы сбрасывает флаг
–	31-6, 1	Зарезервировано	

ETFRC – регистр программной эмуляции флагов триггера событий

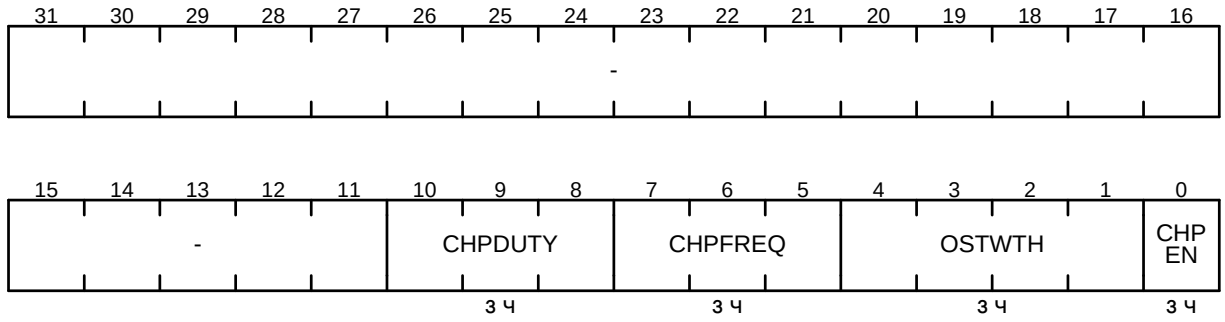
Смещение: + 64h
Сброс: 0h



Поле	Биты	Описание	
DRQB/ DRQA	5/ 4	Бит программной установки флага DRQB/DRQA в регистре ETFLG	
		0	Нет действий
		1	Запись единицы устанавливает флаг
SOCB/ SOCA/ INT	3/ 2/ 0	Бит программной установки флага SOCB/SOCA/INT в регистре ETFLG	
		0	Нет действий
		1	Запись единицы устанавливает флаг
–	31-4, 1	Зарезервировано	

PCCTL - регистр управления модулятором

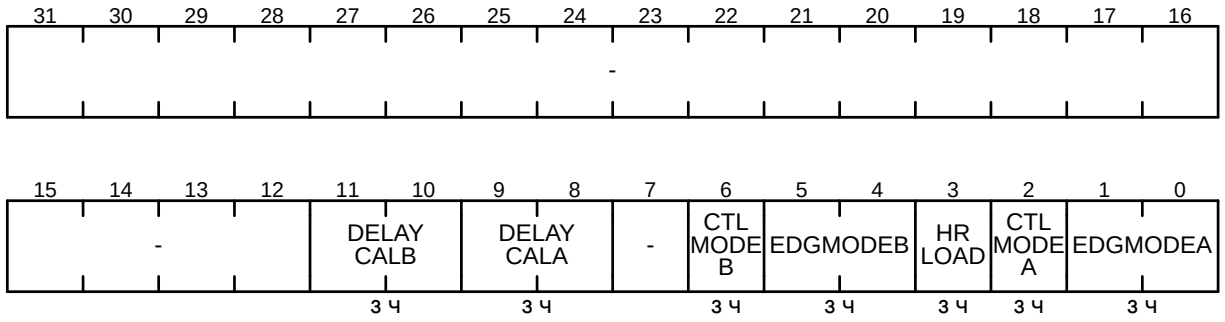
Смещение: + 68h
Сброс: 0h



Поле	Биты	Описание	
CHPDUTY	10-8	Поле задания коэффициента заполнения второго и последующих импульсов	
		000	1/8 (13,5 %)
		001	2/8 (25,0 %)
		…	…
		110	7/8 (87,5 %)
		111	Зарезервировано
CHPFREQ	7-5	Поле выбора делителя частоты синхронизации для задания частоты второго и последующих импульсов	
		000	1
		001	1/2
		…	…
		110	1/7
		111	1/8
OSTWTH	4-1	Поле задания ширины первого импульса	
		0h	$1 \times T_{PCLK} \times 8$
		1h	$2 \times T_{PCLK} \times 8$
		…	…
		Eh	$15 \times T_{PCLK} \times 8$
		Fh	$16 \times T_{PCLK} \times 8$
CHPEN	0	Бит разрешения работы модулятора	
		0	Запрещено
		1	Разрешено
–	31-11	Зарезервировано	

HRCTL – регистр управления блоком ШИМ высокого разрешения

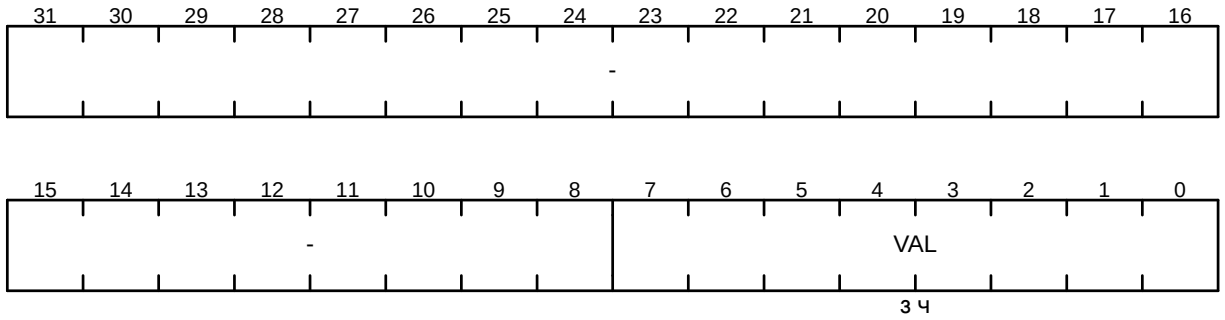
Смещение: + 6Ch
Сброс: 0h



Поле	Биты	Описание	
DELAYCALB/ DELAYCALA	11-10/ 9-8	Флаги для оценки реального времени работы линии задержки на выводе PWMx_B/PWMx_A	
		00	Величина выбранной задержки меньше одного периода системной частоты
		01	Величина выбранной задержки больше или равна одному периоду системной частоты
		10	Зарезервировано
		11	Величина выбранной задержки больше или равна двум периодам системной частоты
CTLMODEB/ CTLMODEA	6/ 2	Бит выбора регистра для задания значения задержки на выводе PWMx_B/PWMx_A	
		0	Поле CMPAHR
		1	Поле TBPHSHR
EDGEMODEB/ EDGEMODEA	5-4/ 1-0	Поле выбора фронта сигнала ШИМ, который сдвигается линией задержки на выводе PWMx_B/PWMx_A	
		00 b	Зарезервировано
		01 b	Передний
		10 b	Задний
		11 b	Передний и задний
HRLOAD	3	Бит выбора события, по которому производится загрузка отложенного значения в активный регистр CMPAHR	
		0	Событие CTR = Zero
		1	Событие CTR = PRD. При этом режим отложенной загрузки доступен, только если бит CTLMODE сброшен, а загрузка поля CMPA осуществляется аналогичным образом (в регистре CMPCTL поле LOADMODE = 00/01)
–	31-12, 7	Зарезервировано	

FWDTH – регистр ширины фильтрации

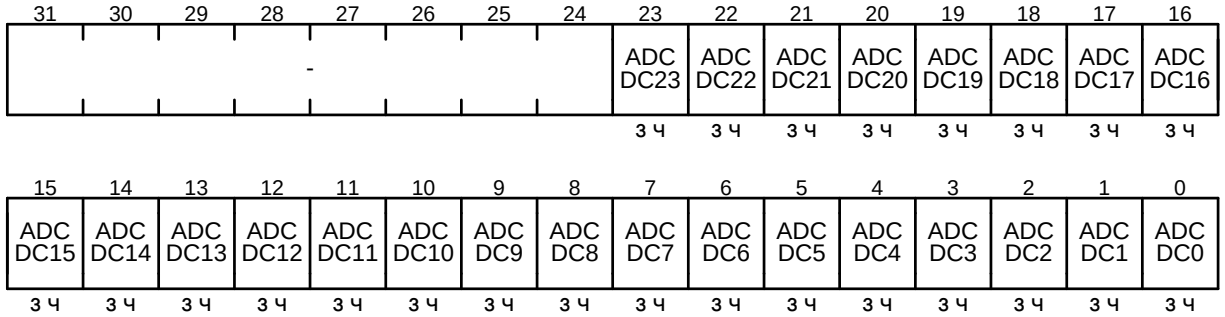
Смещение: + 70h
Сброс: 0h



Поле	Биты	Описание	
VAL	7-0	Поле задания ширины фильтрации коротких импульсов в тактах PCLK. Если f_{PCLK} = 100 МГц, то фильтруются импульсы до 2,55 мкс с шагом 0,01 мкс	
		0h	Фильтр выключен
		1h	1 такт PCLK
		…	…
		FFh	255 тактов PCLK
–	31-8	Зарезервировано	

HDSEL – регистр источника сигнала события удержания

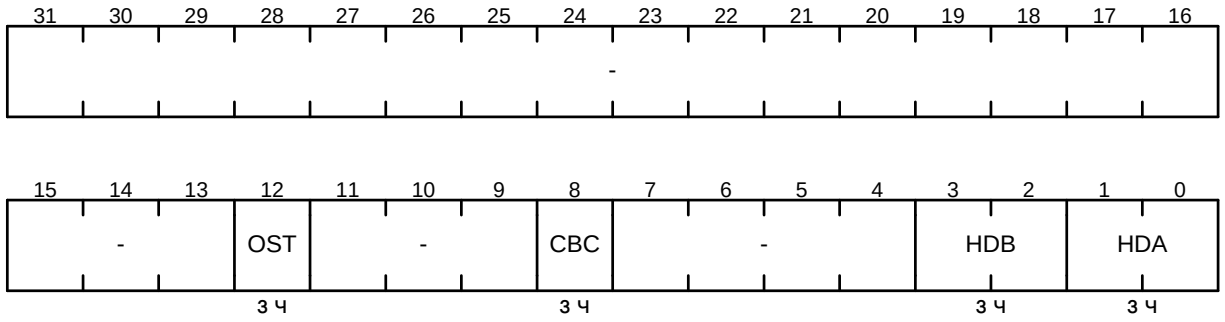
Смещение: + 74h
Сброс: 0h



Поле	Биты	Описание	
ADCDCm	23-0	Бит выбора цифрового компаратора m (m от 0 до 23) блока АЦП, с выхода которого берется сигнал для формирования события удержания	
		0	Не выбрано
		1	Выбрано
–	31-24	Зарезервировано	

HDCTL – регистр управления детектором событий удержания

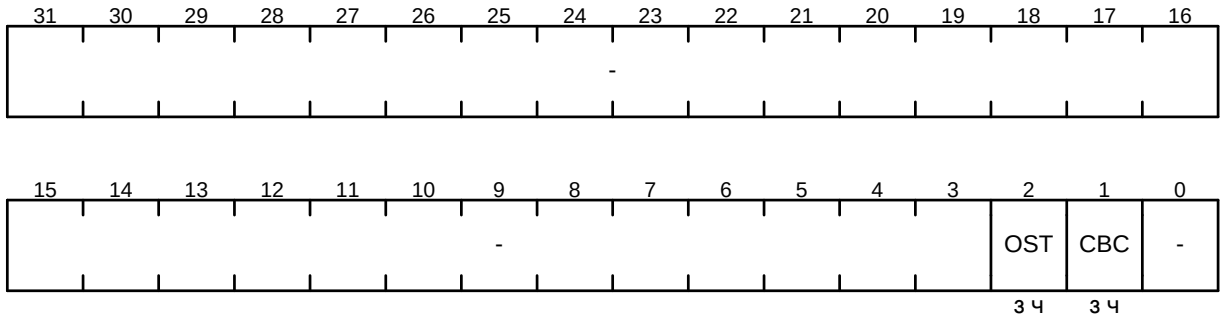
Смещение: + 7Ch
Сброс: 0h



Поле	Биты	Описание	
OST	12	Бит разрешения события по источнику ADCDC/SDFM в однократном режиме обработки аварии	
		0	Запрещено
		1	Разрешено
CBC	8	Бит разрешения события по источнику ADCDC/SDFM в циклическом режиме обработки аварии	
		0	Запрещено
		1	Разрешено
HDB/ HDA	3-2/ 1-0	Поле задания поведения сигнала PWMx_B/PWMx_A в случае сбоя (аварии). (Источник сбоя определяется регистром HDSEL)	
		00b	Зарезервировано
		01b	Переключается в состояние единицы
		10b	Переключается в состояние нуля
		11b	Остается без изменений
–	31-13, 11-9, 7-4	Зарезервировано	

HDEINT – регистр маски прерывания порогового выключателя

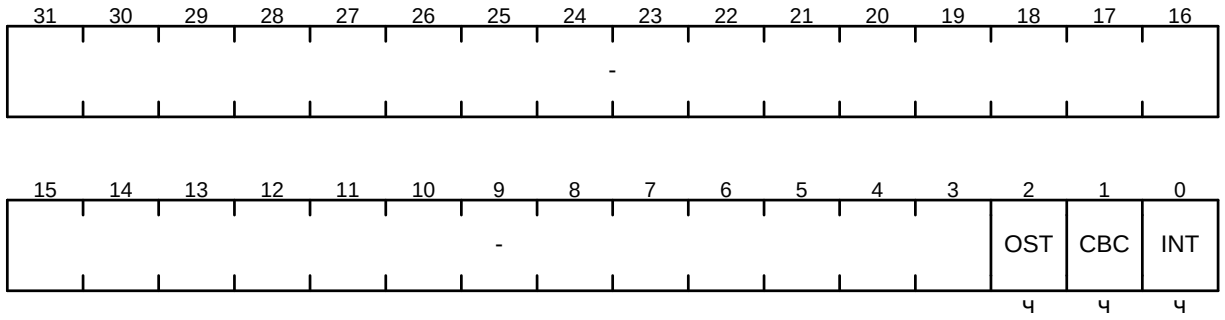
Смещение: + 80h
Сброс: 0h



Поле	Биты	Описание	
OST	2	Бит разрешения генерации прерывания в однократном режиме	
		0	Запрещено
		1	Разрешено
CBC	1	Бит разрешения генерации прерывания в циклическом режиме	
		0	Запрещено
		1	Разрешено
–	31-3, 0	Зарезервировано	

HDFLG – регистр флагов прерывания порогового выключателя

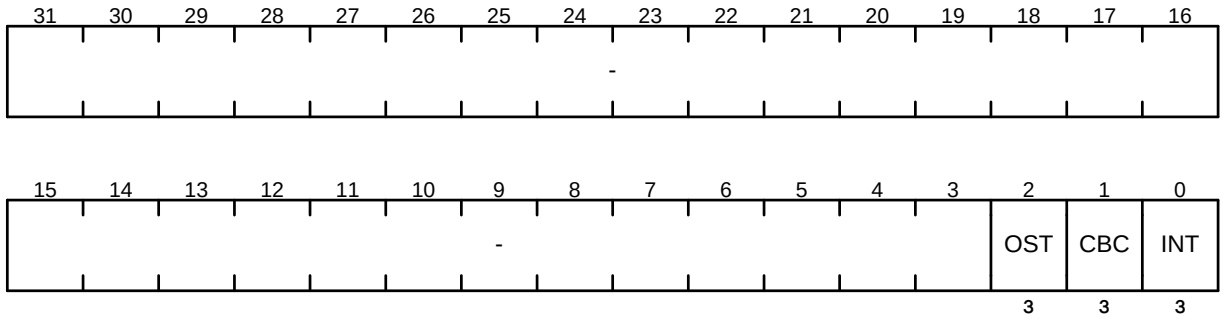
Смещение: + 84h
Сброс: 0h



Поле	Биты	Описание
OST	2	Флаг прерывания в однократном режиме. Примечание – При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова
CBC	1	Флаг прерывания в циклическом режиме. Примечание – При этом действие на выходе продолжается вплоть до обнуления счетчика таймера, если сигнал аварии не перестал быть активным к этому моменту. Если флаг сброшен, а источник сигнала аварии остался, флаг установится снова
INT	0	Флаг внешнего прерывания PLIC. Если флаг был сброшен, а один из флагов CBC или OST установлен, флаг INT установится снова
–	31-3	Зарезервировано

HDCLR – регистр сброса флагов порогового выключателя

Смещение: + 88h
Сброс: 0h

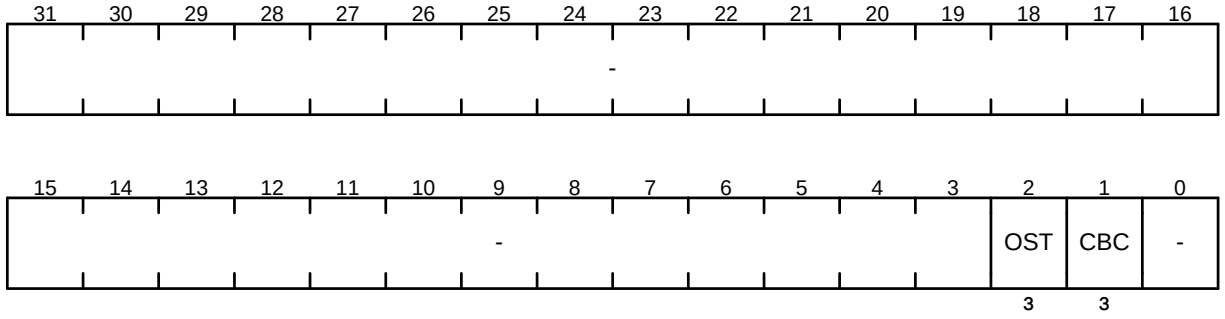


Поле	Бит	Описание
OST	2	Бит сброса флага прерывания в однократном режиме
CBC	1	Бит сброса флага прерывания в циклическом режиме
INT	0	Бит сброса флага внешнего прерывания PLIC
–	31-3	Зарезервировано

Примечание – Запись единицы в бит регистра сбрасывает соответствующий бит в регистре HDFLG

HDFRC – регистр программной активации порогового выключателя

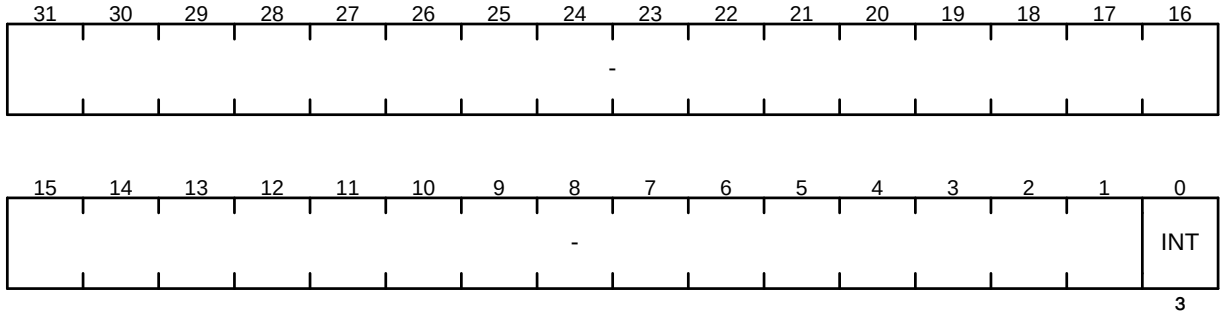
Смещение: + 8Ch
Сброс: 0h



Поле	Бит	Описание	
OST	2	Бит активации порогового выключателя в однократном режиме обработки аварии	
		0	Нет действий
		1	Запись единицы активирует выключатель и устанавливает флаг OST в регистре HDFLG
CBC	1	Бит активации порогового выключателя в циклическом режиме обработки аварии	
		0	Нет действий
		1	Запись единицы активирует выключатель и устанавливает флаг CBC в регистре HDFLG
–	31-3, 0	Зарезервировано	

HDINTCLR – регистр сброса прерывания порогового выключателя

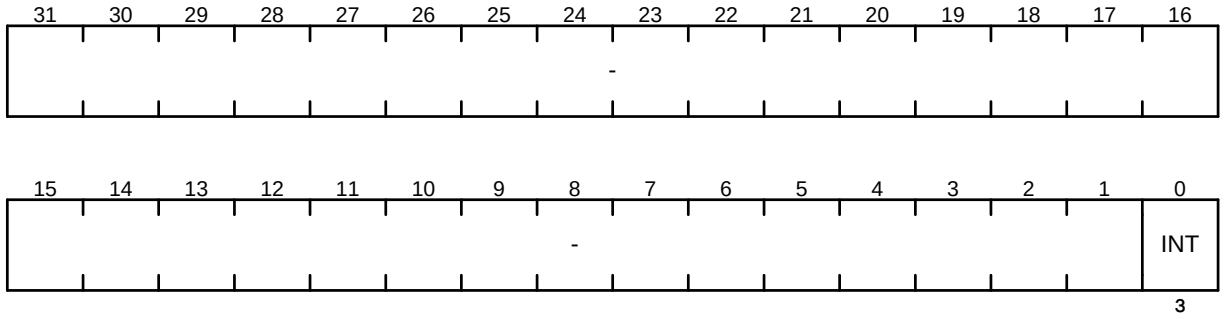
Смещение: + 90h
Сброс: 0h



Поле	Биты	Описание
INT	0	Бит сброса прерывания. Сброс запроса на прерывание должен производиться программой обработки прерывания, во избежание повторного запуска обработчика
–	31-1	Зарезервировано

TZINTCLR – регистр сброса прерывания детектора событий аварии

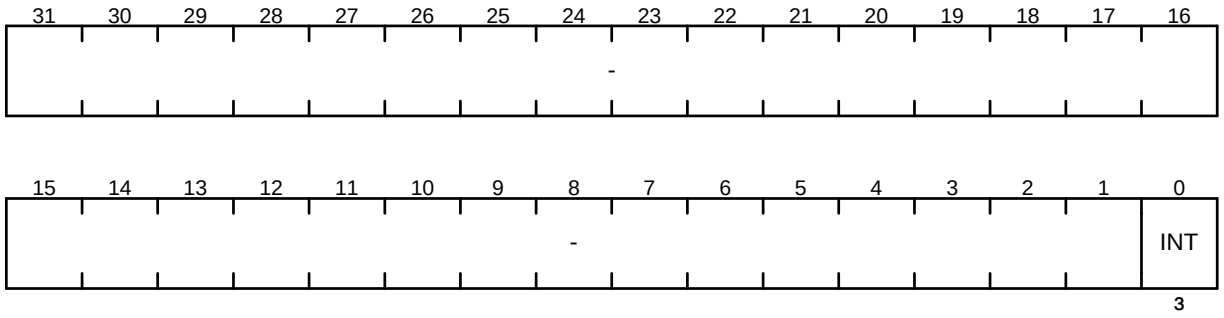
Смещение: + 94h
Сброс: 0h



Примечание – Назначение и функционирование аналогично регистру HDINTCLR.

INTCLR – регистр сброса прерывания таймера блока ШИМ

Смещение: + 98h
Сброс: 0h



Примечание – Назначение и функционирование аналогично регистру HDINTCLR.

А.13 Регистры блока АЦП

Базовый адрес: 4102_0000h

Смещение:

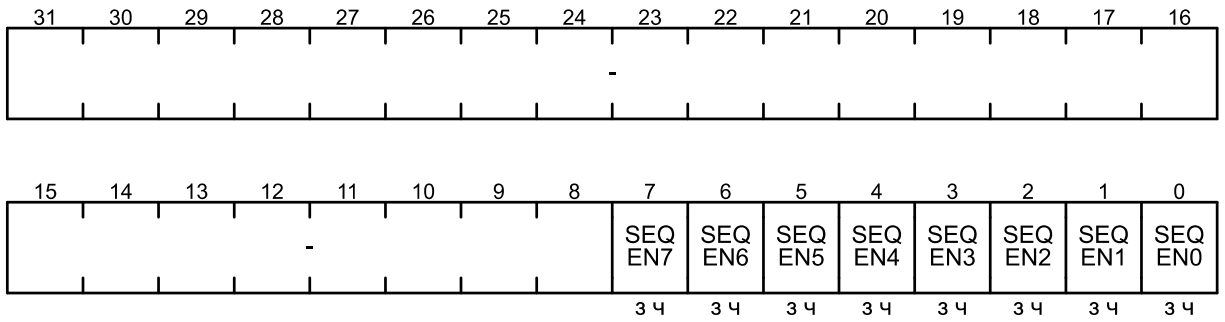
+ 040h (SEQ0)	Регистры секвенсора 0
+ 07Ch (SEQ1)	Регистры секвенсора 1
+ 0B8h (SEQ2)	Регистры секвенсора 2
+ 0F4h (SEQ3)	Регистры секвенсора 3
+ 130h (SEQ4)	Регистры секвенсора 4
+ 16Ch (SEQ5)	Регистры секвенсора 5
+ 1A8h (SEQ6)	Регистры секвенсора 6
+ 1E4h (SEQ7)	Регистры секвенсора 7
+ 400h (DC0)	Регистры цифрового компаратора 0
+ 40Ch (DC1)	Регистры цифрового компаратора 1
+ 418h (DC2)	Регистры цифрового компаратора 2
+ 424h (DC3)	Регистры цифрового компаратора 3
+ 430h (DC4)	Регистры цифрового компаратора 4
+ 43Ch (DC5)	Регистры цифрового компаратора 5
+ 448h (DC6)	Регистры цифрового компаратора 6
+ 454h (DC7)	Регистры цифрового компаратора 7
+ 460h (DC8)	Регистры цифрового компаратора 8
+ 46Ch (DC9)	Регистры цифрового компаратора 9
+ 478h (DC10)	Регистры цифрового компаратора 10
+ 484h (DC11)	Регистры цифрового компаратора 11
+ 490h (DC12)	Регистры цифрового компаратора 12
+ 49Ch (DC13)	Регистры цифрового компаратора 13
+ 4A8h (DC14)	Регистры цифрового компаратора 14
+ 4B4h (DC15)	Регистры цифрового компаратора 15
+ 4C0h (DC16)	Регистры цифрового компаратора 16
+ 4CCh (DC17)	Регистры цифрового компаратора 17
+ 4D8h (DC18)	Регистры цифрового компаратора 18
+ 4E4h (DC19)	Регистры цифрового компаратора 19
+ 4F0h (DC20)	Регистры цифрового компаратора 20
+ 4FCh (DC21)	Регистры цифрового компаратора 21
+ 508h (DC22)	Регистры цифрового компаратора 22
+ 514h (DC23)	Регистры цифрового компаратора 23
+ 540h (ACTL)	Массив регистров настройки АЦП
+ 580h (CHCTL)	Массив регистров настройки каналов
+680h (CHDELAY)	Массив регистров настройки количества дополнительных циклов для каждого канала

Мнемоника: SEQs;
DCd

Примечание – s – номер секвенсора 0 – 7;
d – номер по порядку цифрового компаратора от 0 до 23;
m – номер модуля АЦП от 0 до 3;
n – номер канала АЦП от 0 до 63.

SEQEN – регистр включения секвенсоров

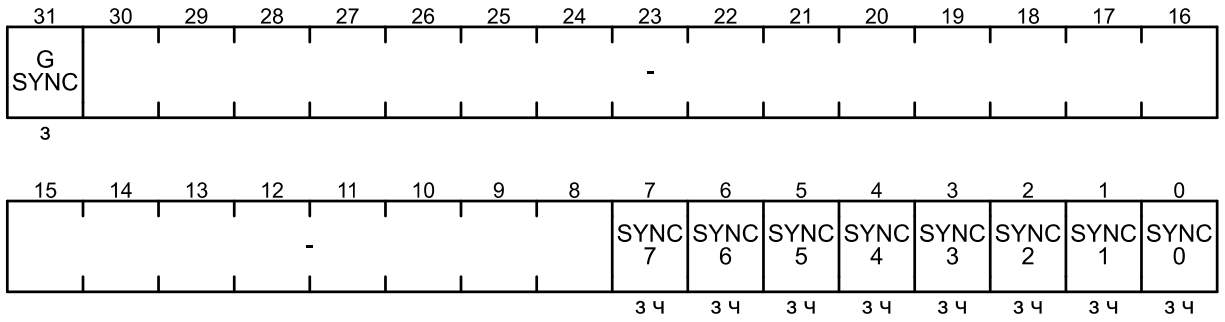
Смещение: + 00h
Сброс: 0000_0000h



Поле	Биты	Описание	
SEQENs	7-0	Бит разрешения работы секвенсора s	
		0	Запрещено
		1	Разрешено
–	31-8	Зарезервировано	

SEQSYNC – регистр программной синхронизации секвенсоров

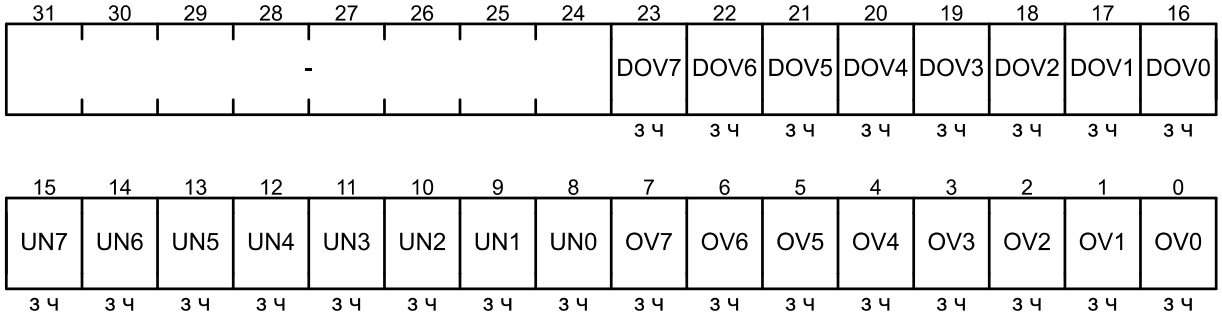
Смещение: 04h
Сброс: 0000_0000h



Поле	Биты	Описание	
GSYNC	31	Бит запуска секвенсоров. Запись единицы запускает секвенсоры, работа которых разрешена и для которых установлены биты SYNCs	
SYNCs	7-0	Бит разрешения запуска секвенсора s	
		0	Запрещено
		1	Разрешено (если установлен бит SEQENs в регистре SEQEN)
–	30-8	Зарезервировано	

FSTAT – регистр флагов буферов результатов и блока DMA

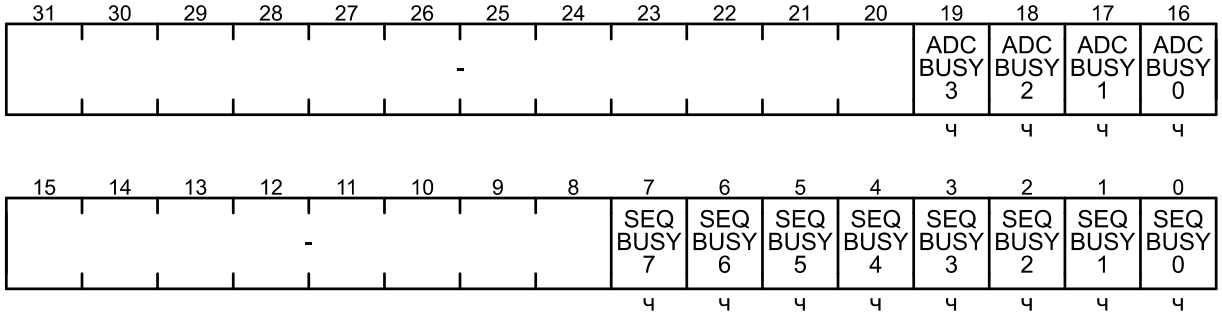
Смещение: 08h
Сброс: 0000_0000h



Поле	Биты	Описание	
DOVs	23-16	Флаг ошибки DMA	
		0	Нет ошибки
		1	При наличии обрабатываемого запроса DMA от секвенсора s, пришел еще один запрос, который не может быть обработан
		Флаг сбрасывается записью единицы	
UNs	15-8	Флаг пустоты буфера секвенсора s	
		0	Буфер не пуст
		1	Буфер пуст
		Флаг может быть сброшен программно записью единицы	
OVs	7-0	Флаг заполнения буфера секвенсора s	
		0	В буфере есть как минимум одна свободная ячейка
		1	Буфер заполнен. Все последующие записи в буфер блокируются до появления как минимум одной свободной ячейки
		Флаг сбрасывается записью единицы	
–	31-24	Зарезервировано	

BSTAT – регистр флагов занятости

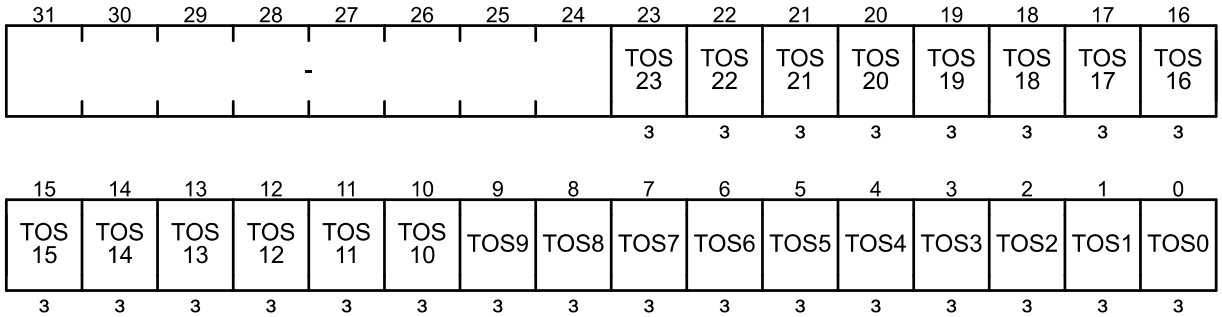
Смещение: 0Ch
Сброс: 0000_0000h



Поле	Биты	Описание	
ADCBUSYs	19-16	Флаг занятости модуля АЦП s	
		0	АЦП выключен или в режиме ожидания запроса
		1	АЦП проводит измерения по активным запросам
SEQBUSYs	7-0	Флаг занятости секвенсора s	
		0	Секвенсор выключен или в режиме ожидания сигнала запуска
		1	Секвенсор производит запуск/перезапуск или выполняет задержку перезапуска
–	31-20, 15-8	Зарезервировано	

DCTRIG – регистр сброса флагов выходных триггеров компараторов

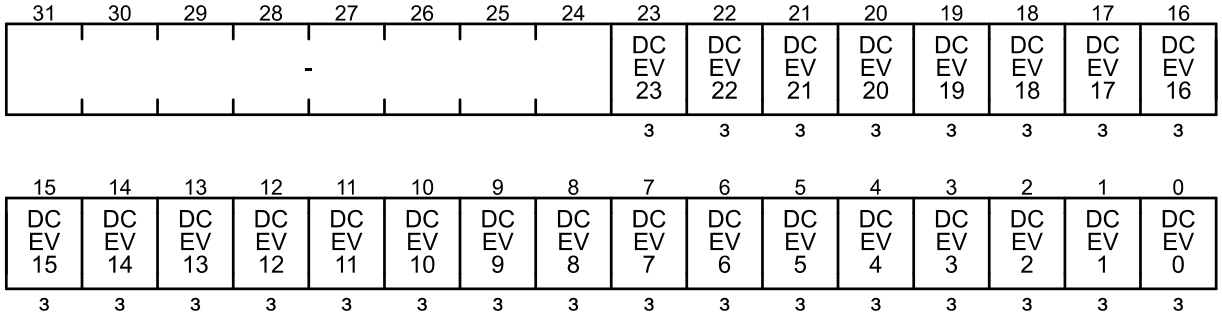
Смещение: 10h
Сброс: 0000_0000h



Поле	Биты	Описание	
TOSd	23-0	Флаг состояния выходного триггера компаратора d	
		0	Триггер не сработал
		1	Триггер сработал
		Флаг сбрасывается записью единицы	
–	31-24	Зарезервировано	

DCEV – регистр сброса флагов компараторов

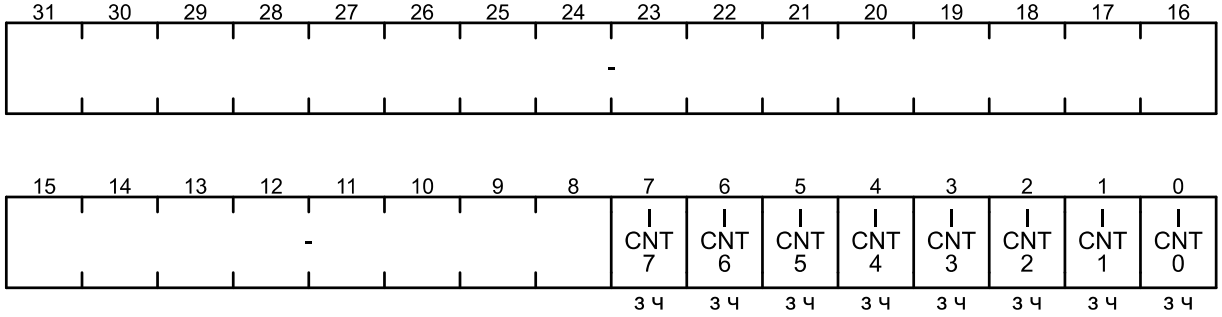
Смещение: 14h
Сброс: 0000_0000h



Поле	Биты	Описание	
DCEVd	23-0	Флаг события сравнения компаратора d	
		0	Сравнение не выполнялось
		1	Сравнение выполнялось
		Флаг сбрасывается записью единицы	
–	31-24	Зарезервировано	

CICNT – регистр настройки режима сброса счетчика прерываний

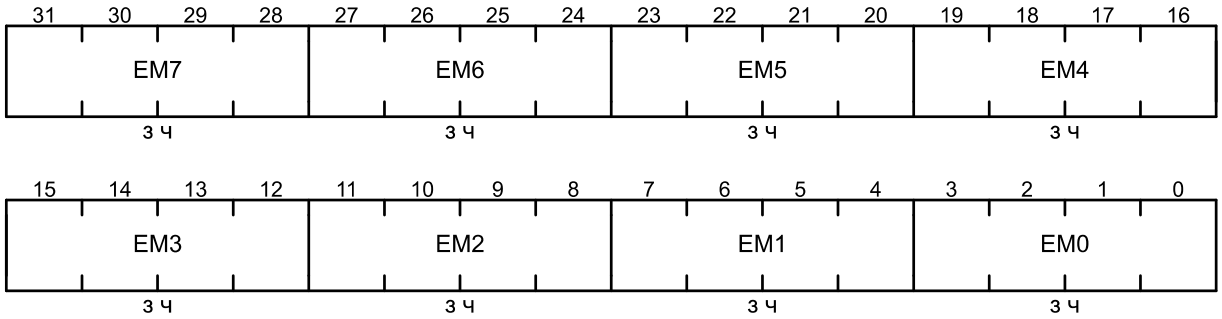
Смещение: 18h
Сброс: 0000_0000h



Поле	Биты	Описание	
ICNTs	7-0	Бит выбора режима сброса счетчика, используемого для генерации прерываний секвенсора s	
		0	Счетчик будет сбрасываться по запуску секвенсора
		1	Запрет сброса счетчика по запуску секвенсора
–	31-8	Зарезервировано	

EMUX – регистр выбора событий запуска секвенсоров

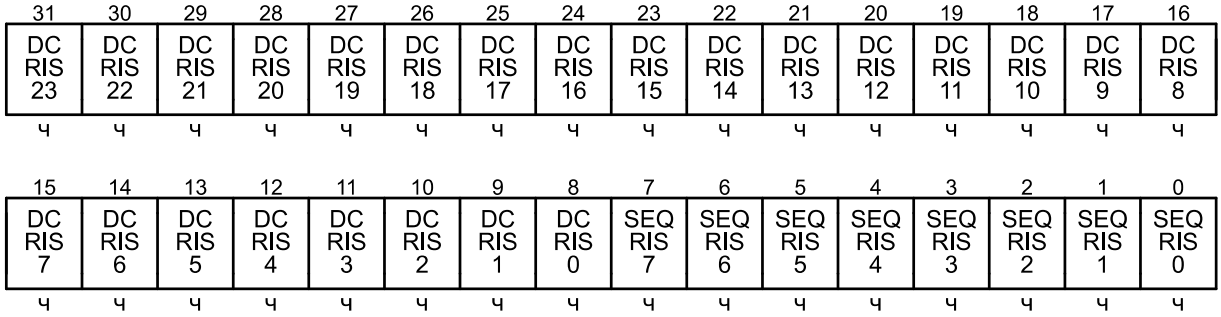
Смещение: 1Ch
Сброс: 0000_0000h



Поле	Биты	Описание	
EMs	31-0	Поле выбора события для запуска секвенсора s	
		0h	Установка бита GSYNC в регистре SEQSYNC
		1h	Сигнал от любого из GPIO
		2h	Сигнал от ACMP0
		3h	Сигнал от ACMP1
		4h	Сигнал от ACMP2
		5h	Сигнал от блоков TMR0-TMR3
		6h	Сигнал от блоков TMR4-TMR7
		7h	Сигнал от блоков TMR8-TMR11
		8h	Сигнал от блоков TMR12-TMR15
		9h	Сигнал от блоков PWM0-PWM2 канал A
		Ah	Сигнал от блоков PWM0-PWM2 канал B
		Bh	Сигнал от блоков PWM3-PWM5 канал A
		Ch	Сигнал от блоков PWM3-PWM5 канал B
		Dh	Сигнал от блоков PWM6-PWM8 канал A
		Eh	Сигнал от блоков PWM6-PWM8 канал B
		Fh	Циклическая работа. Активируется после установки бита GSYNC в регистре SEQSYNC

RIS – регистр флагов немаскированных прерываний

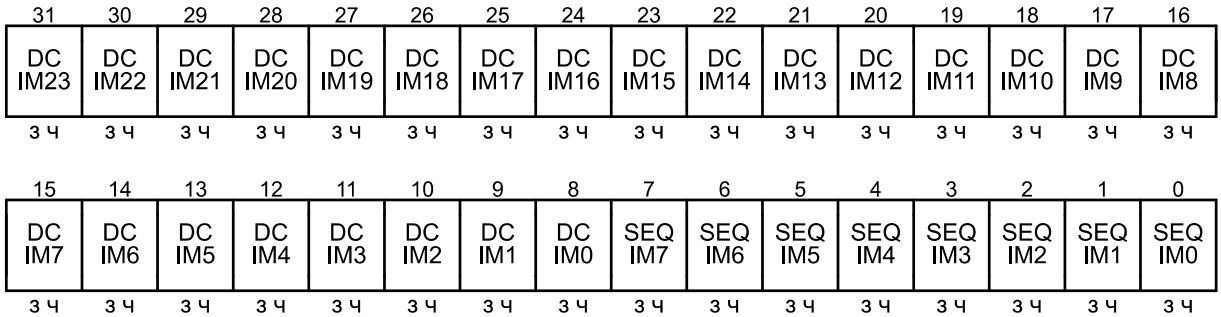
Смещение: 20h
Сброс: 0000_0000h



Поле	Биты	Описание	
DCRISd	31-8	Флаг прерывания компаратора d	
		0	Нет прерывания или флаг сброшен
		1	Поступил запрос на прерывание
SEQRISs	7-0	Флаг прерывания секвенсора s	
		0	Нет действий
		1	Запрос секвенсора завершился и счетчик прерываний досчитал до значения ICNT регистра SCCTL

IM – регистр маски прерываний

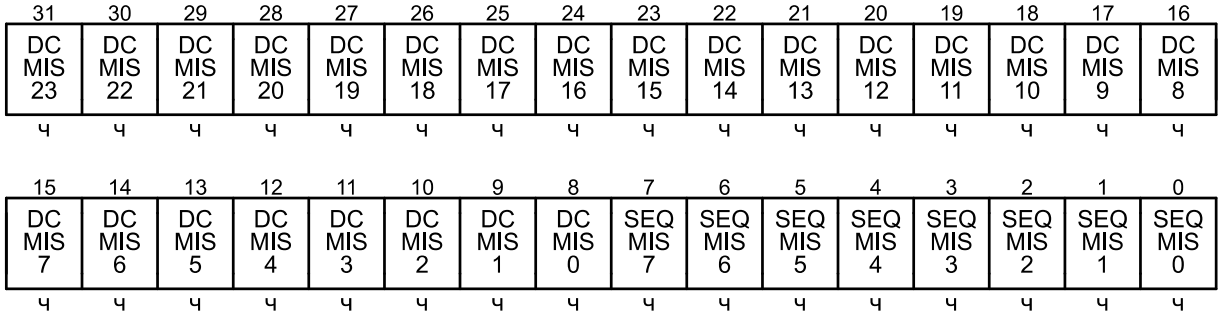
Смещение: 24h
Сброс: 0000_0000h



Поле	Биты	Описание	
DCIMd	31-8	Маска прерывания компаратора d	
		0	Маскировано
		1	Разрешено
SEQIMs	7-0	Маска прерывания секвенсора s	
		0	Маскировано
		1	Разрешено

MIS – регистр флагов маскированных прерываний

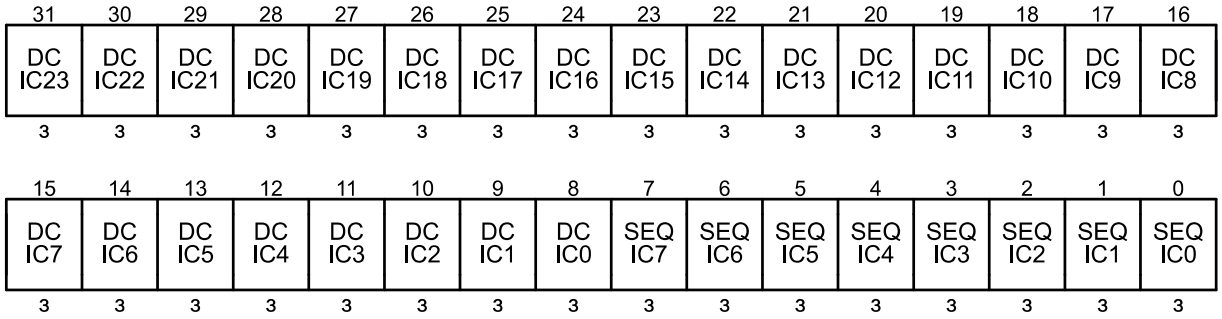
Смещение: 28h
Сброс: 0000_0000h



Поле	Биты	Описание	
DCMISd	31-8	Флаг маскированного прерывания компаратора d	
		0	Нет прерывания или флаг сброшен
		1	Поступил запрос на прерывание
SEQMISs	7-0	Флаг маскированного прерывания секвенсора s	
		0	Нет действий
		1	Запрос секвенсора завершился и счетчик прерываний досчитал до значения ICNT регистра SCCTL

IC – регистр сброса флагов прерываний

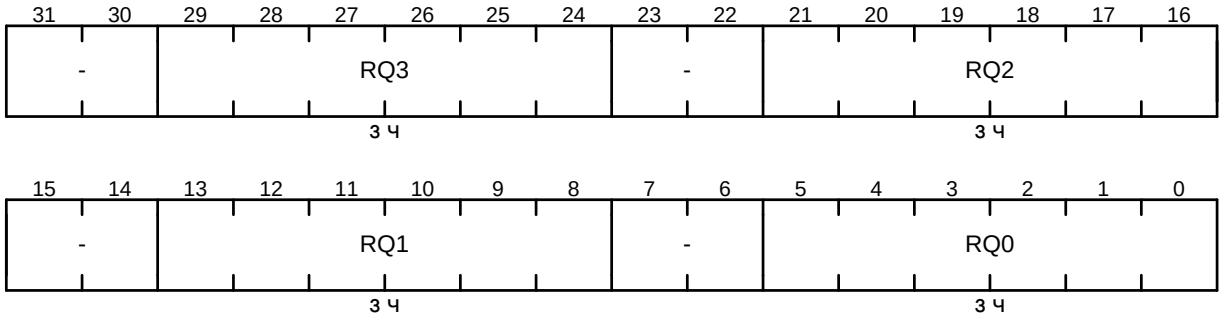
Смещение: 2Ch
Сброс: 0000_0000h



Поле	Биты	Описание	
DCICd	31-8	Сброс маскированного и немаскированного флага прерывания компаратора d	
		0	Нет действий
		1	Сброс флагов
SEQICs	7-0	Сброс маскированного и немаскированного флага прерывания секвенсора s	
		0	Нет действий
		1	Сброс флагов

SRQSEL0 – регистр 0 выбора каналов для запросов секвенсора

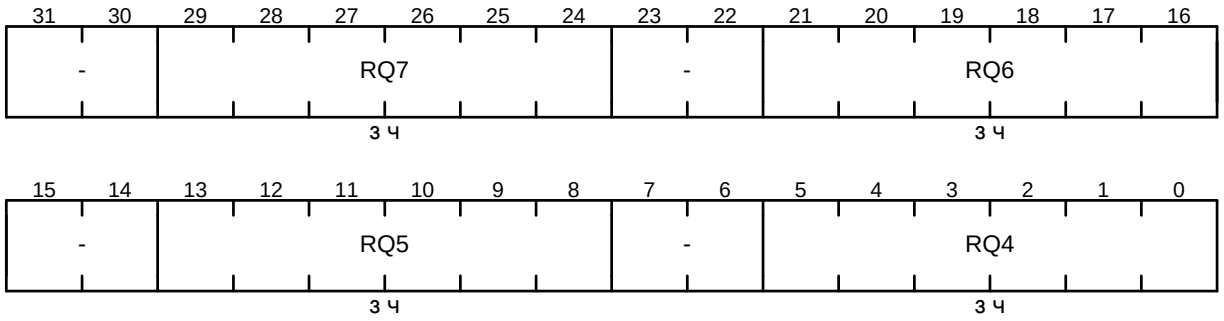
Смещение: SEQs + 00h
Сброс: 0000_0000h



Поле	Биты	Описание
RQn	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 0 до 3) секвенсора s. Допустимые значения 0 – 63
–	31-30, 23-22, 15-14, 5-0	Зарезервировано

SRQSEL1 – регистр 1 выбора каналов для запросов секвенсора

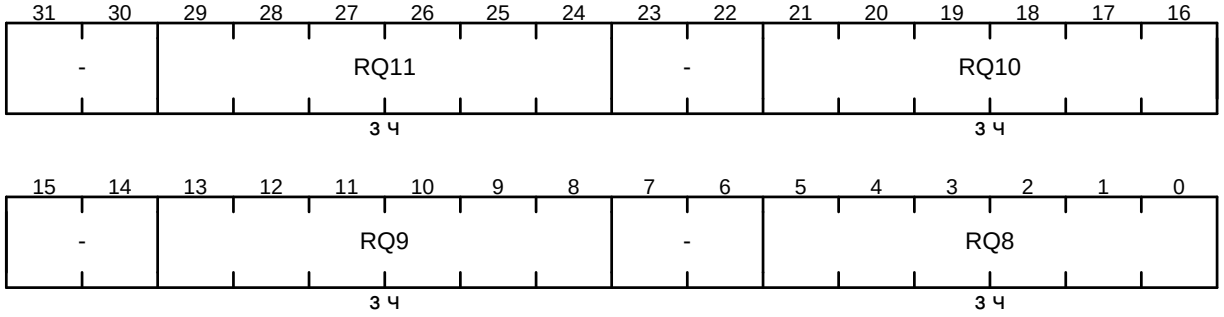
Смещение: SEQs + 04h
Сброс: 0000_0000h



Поле	Биты	Описание
RQi	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 4 до 7) секвенсора s. Допустимые значения 0 – 63
–	31-30, 23-22, 15-14, 7-6	Зарезервировано

SRQSEL2 – регистр 2 выбора каналов для запросов секвенсора

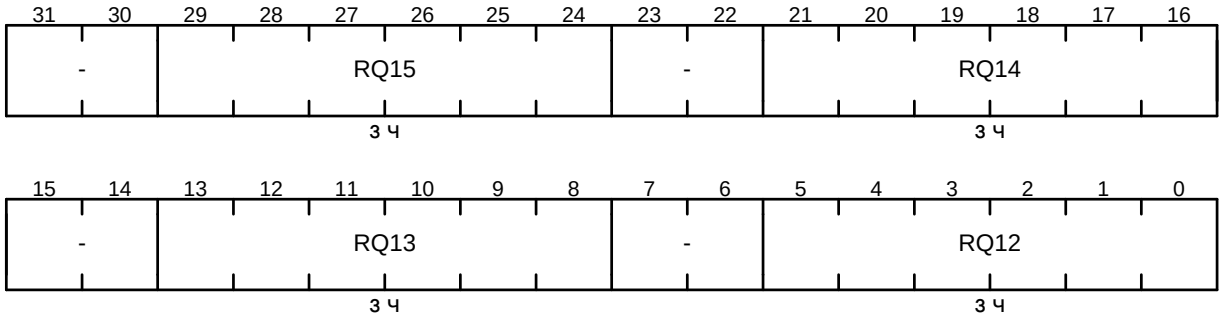
Смещение: SEQs + 08h
Сброс: 0000_0000h



Поле	Биты	Описание
RQi	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 8 до 11) секвенсора s. Допустимые значения 0 – 63
–	31-30, 23-22, 15-14, 7-6	Зарезервировано

SRQSEL3 – регистр 3 выбора каналов для запросов секвенсора

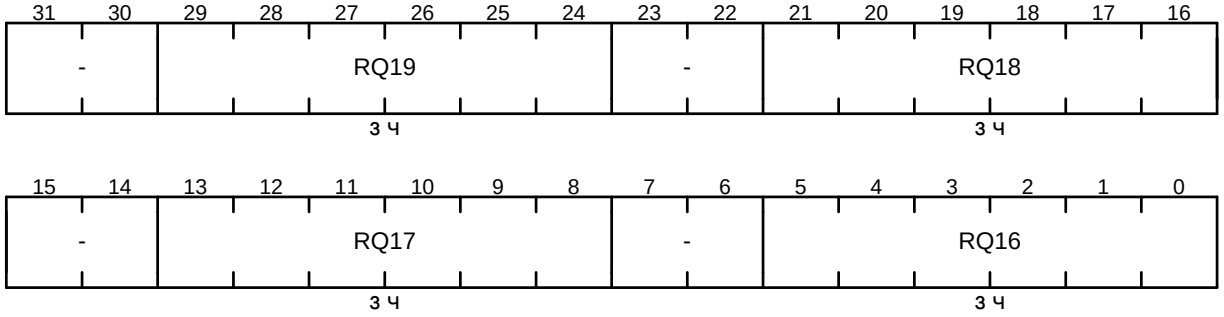
Смещение: SEQs + 0Ch
Сброс: 0000_0000h



Поле	Биты	Описание
RQi	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 12 до 15) секвенсора s. Допустимые значения 0 – 63.
–	31-30, 23-22, 15-14, 7-6	Зарезервировано

SRQSEL4 – регистр 4 выбора каналов для запросов секвенсора

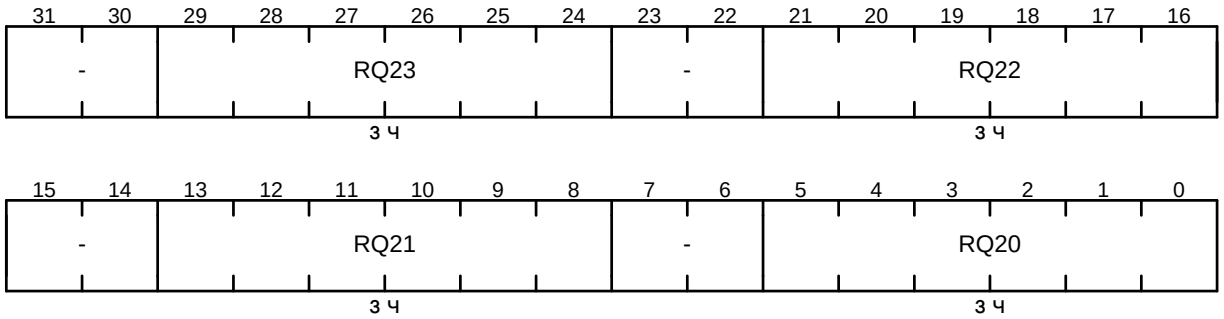
Смещение: SEQs + 10h
Сброс: 0000_0000h



Поле	Биты	Описание
RQi	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 16 до 19) секвенсора s. Допустимые значения 0 – 63
–	31-30, 23-22, 15-14, 7-6	Зарезервировано

SRQSEL5 – регистр 5 выбора каналов для запросов секвенсора

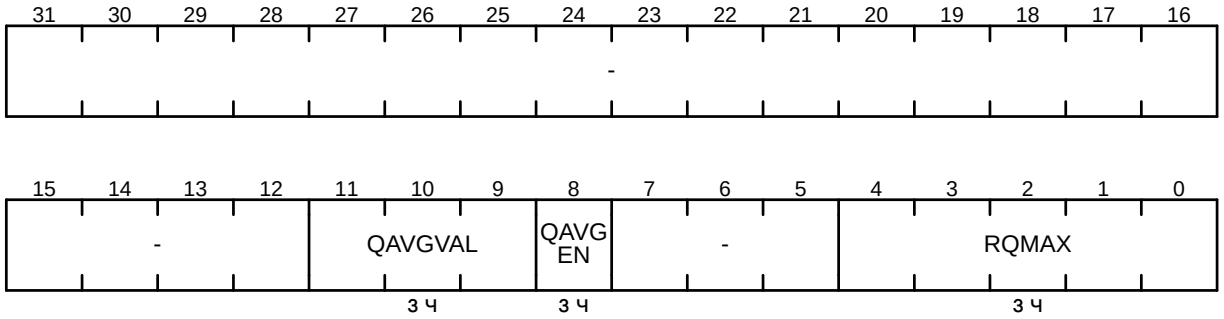
Смещение: SEQs + 14h
Сброс: 0000_0000h



Поле	Биты	Описание
RQi	29-24, 21-16, 13-8, 5-0	Номер канала АЦП для запроса i (i от 20 до 23) секвенсора s. Допустимые значения 0 – 63
–	31-30, 23-22, 15-14, 7-6	Зарезервировано

SRQCTL – регистр управления очередью запросов секвенсора

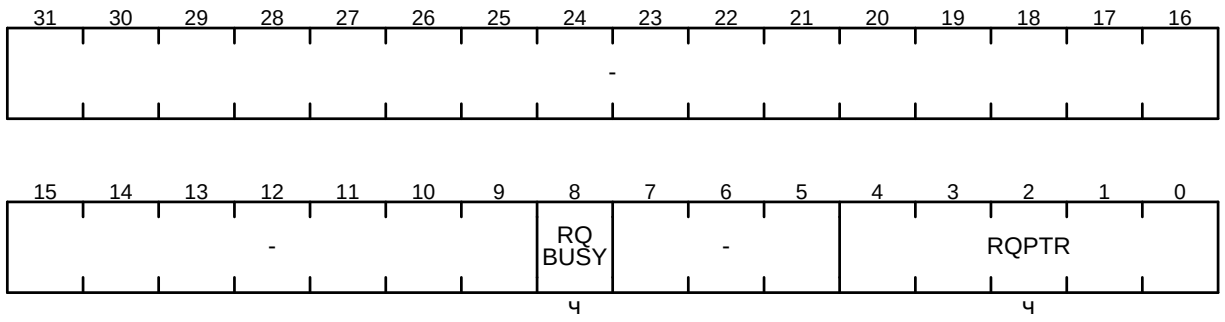
Смещение: SEQs + 18h
Сброс: 0000_0000h



Поле	Биты	Описание	
QAVGVAL	11-9	Поле задания количества опросов для усреднения сканированием	
		000	Зарезервировано
		001	2
		010	4
		011	8
		100	16
		101	32
		110	64
		111	Зарезервировано
QAVGEN	8	Бит управления режимом усреднения сканированием	
		0	Усреднение сканированием очереди запросов отключено
		1	Усреднение сканированием очереди запросов включено
RQMAX	4-0	Поле задания «глубины» очереди запросов секвенсора s (номер последнего элемента)	
–	31-12, 7-5	Зарезервировано	

SRQSTAT – регистр статуса очереди запросов секвенсора

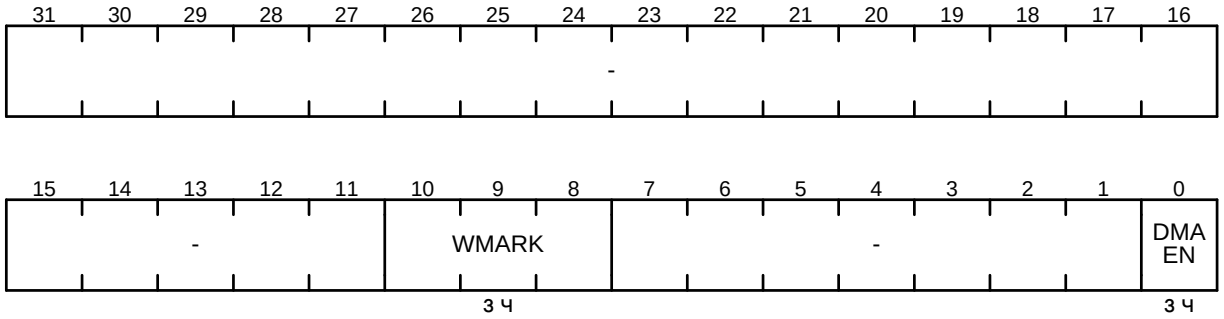
Смещение: SEQs + 1Ch
Сброс: 00000000h



Поле	Биты	Описание	
RQBUSY	8	Флаг активного запроса секвенсора	
		0	Текущий запрос неактивен
		1	Запрос на измерение выставлен и находится в состоянии обработки или ожидания
RQPTR	4-0	Номер текущего запроса в очереди	
–	31-9, 7-5	Зарезервировано	

SDMACTL – регистр управления запросами DMA секвенсора

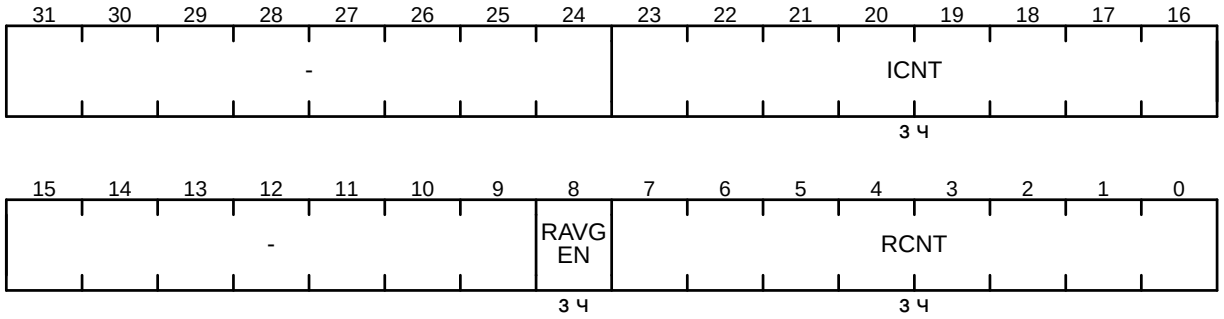
Смещение: SEQs + 20h
Сброс: 0000_0000h



Поле	Биты	Описание	
WMARK	10-8	Поле задания количества результатов измерений, записанных в буфер секвенсора, по достижении которого вызывается DMA	
		000	Зарезервировано
		001	Одна запись в буфер
		010	2
		011	4
		100	8
		101	16
		110	32
		111	Зарезервировано
DMAEN	0	Бит разрешения использования блока DMA	
		0	Запрещено
		1	Разрешено
–	31-11, 7-1	Зарезервировано	

SCCTL – регистр управления счетчиками прерывания и перезапуска секвенсора

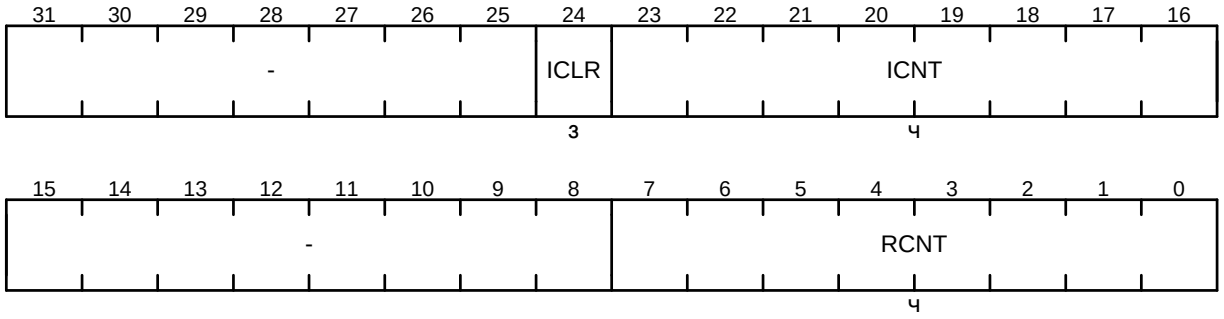
Смещение: SEQs + 24h
Сброс: 0000_0000h



Поле	Биты	Описание	
ICNT	23-16	Цикличность прерываний. Поле задания количества запросов секвенсором s модуля АЦП, по достижении которого генерируется прерывание. Значение 00h означает выставление прерывания по каждому запросу модуля АЦП, значение FFh – каждые 256 запросов	
RAVGEN	8	Бит разрешения режима усреднения результатов по перезапускам	
		0	Запрещено
		1	Разрешено
		Примечание – Для корректной работы этого режима поле RCNT регистра SCCTL должно содержать любое значение, соответствующее $2^p - 1$, где p = 1, ..., 8	
RCNT	7-0	Количество перезапусков очереди запросов секвенсора s. Поле задания количества перезапусков очереди запросов секвенсора s после его запуска по событию. Значение 00h соответствует режиму без перезапусков, значение 01h – один перезапуск, FFh – 255 перезапусков	
–	31-24, 15-9	Зарезервировано	

SCVAL – регистр состояния счетчиков прерывания и перезапуска секвенсора

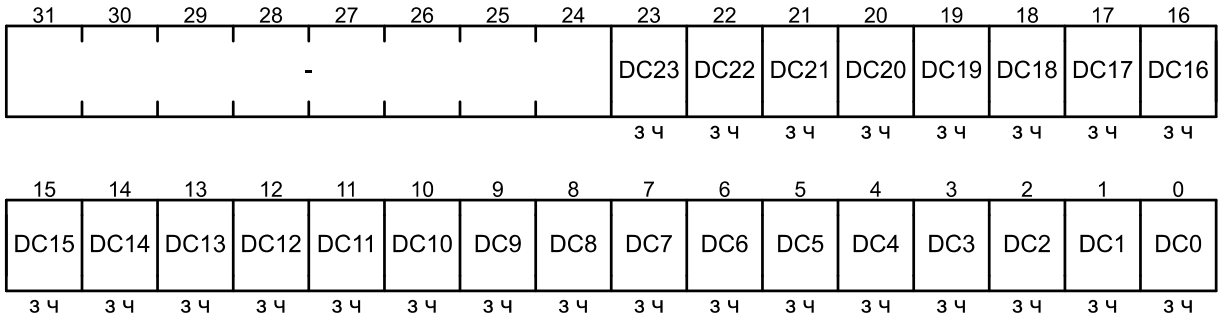
Смещение: SEQs + 28h
Сброс: 0000_0000h



Поле	Биты	Описание	
ICLR	24	Бит сброса счетчика запросов для генерации прерывания	
		0	Нет действий
		1	Сброс счетчика ICNT
ICNT	23-16	Текущее состояние счетчика запросов, используемого для генерации прерываний	
RCNT	7-0	Текущее количество совершенных перезапусков	
–	31-25, 15-8	Зарезервировано	

SDC – регистр выбора компаратора секвенсором

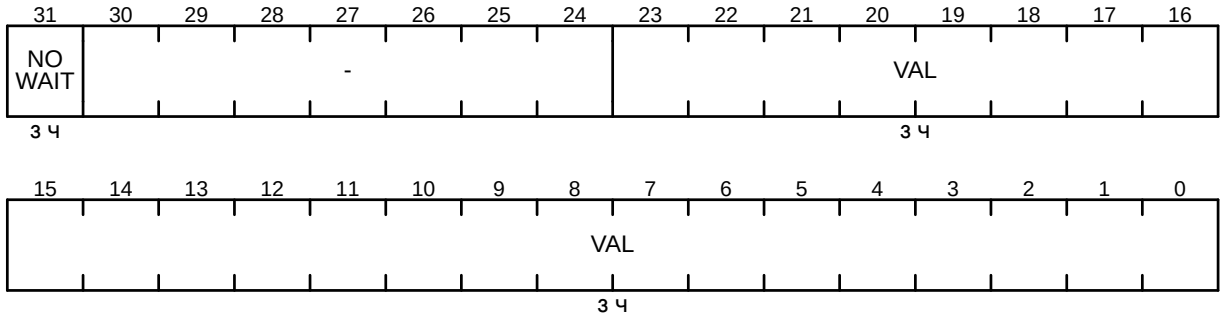
Смещение: SEQs + 2Ch
Сброс: 0000_0000h



Поле	Биты	Описание	
DCd	23-0	Бит разрешения работы компаратора d секвенсором	
		0	Запрещено
		1	Разрешено
–	31-24	Зарезервировано	

SRTMR – регистр задержки перезапусков секвенсора

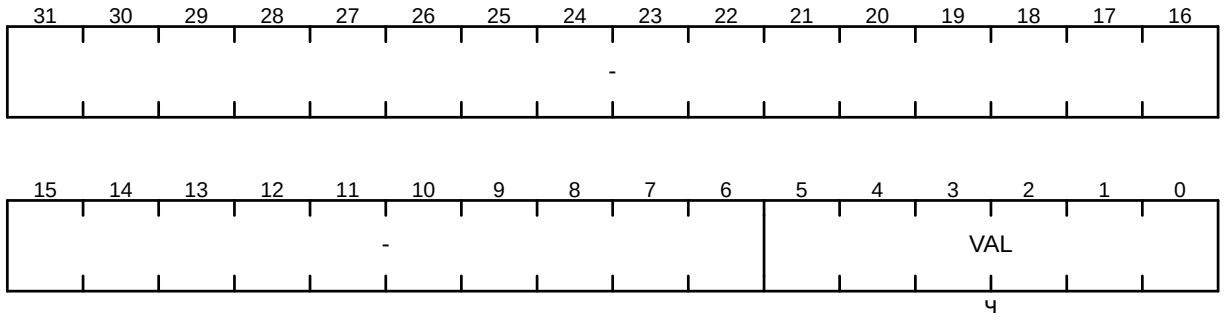
Смещение: SEQs + 30h
Сброс: 0000_0000h



Поле	Биты	Описание	
NOWAIT	31	Бит управления теневой загрузкой значения задержки перезапуска	
		0	Значение обновится по ближайшему событию запуска секвенсора
		1	Значение обновится по ближайшему событию перезапуска
VAL	23-0	Поле задания задержки перезапуска очереди секвенсора. Значение 000000h задает немедленный перезапуск (если он включен)	
–	30-24	Зарезервировано	

SFLOAD – регистр загрузки буфера секвенсора

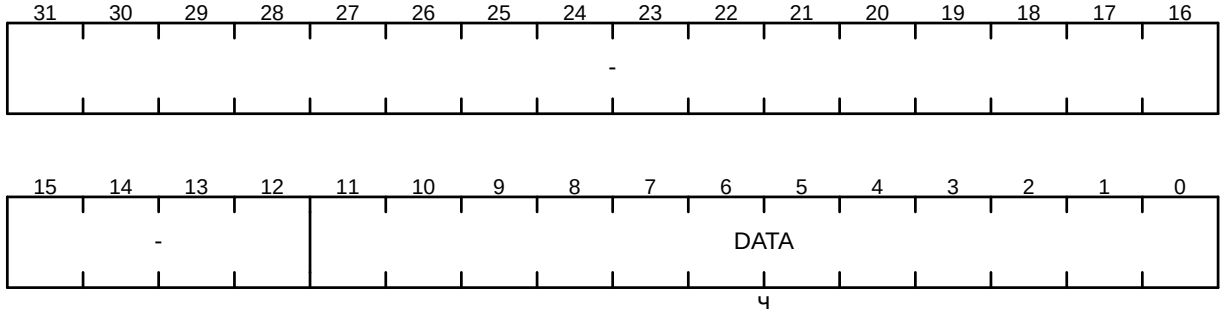
Смещение: SEQs + 34h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	5-0	Значение количества результатов измерений, сохраненных в буфере секвенсора
–	31-6	Зарезервировано

SFIFO – регистр результата измерения секвенсора

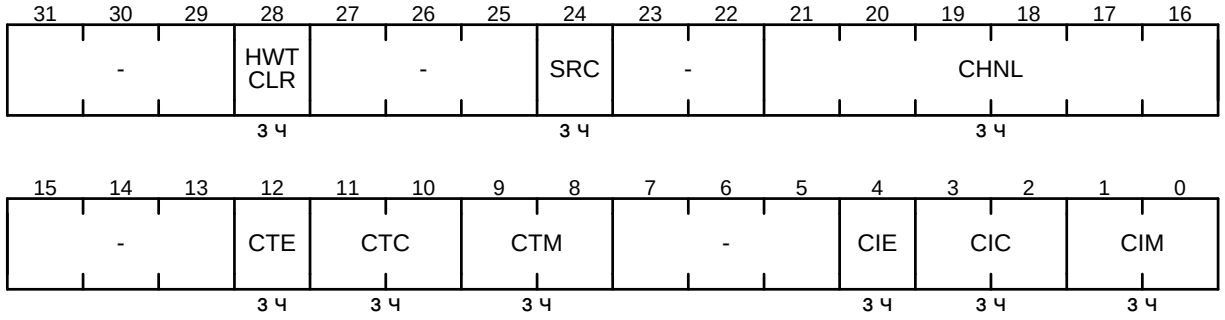
Смещение: SEQs + 38h
Сброс: 0000_0000h



Поле	Биты	Описание
DATA	11-0	Результат измерения. Чтение поля DATA возвращает результат измерения из буфера секвенсора
–	31-12	Зарезервировано

DCTL – регистр управления компаратора

Смещение: DCd + 00h
Сброс: 0000_0000h



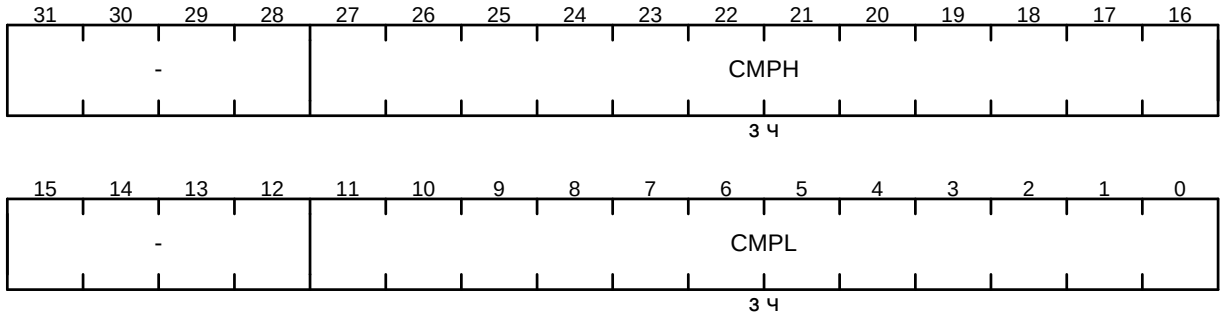
Поле	Биты	Описание	
HWTCLR	28	Управление аппаратным сбросом выходного триггера компаратора	
		0	Сброс происходит через флаги DCTRIG
		1	Сброс происходит аппаратно при первом отрицательном сравнении
SRC	24	Выбор источника значения для сравнения	
		0	Окончание измерения АЦП
		1	Запись результата в FIFO секвенсором
CHNL	21-16	Номер канала. Поле выбирает канал, результат измерения которого будет передан на компаратор	
CTE	12	Бит разрешения срабатывания выходного триггера компаратора	
		0	Запрещено
		1	Разрешено

Поле	Биты	Описание

CTC	11-10	Поле задания условия срабатывания выходного триггера. Если для значения, полученного в результате измерения, выполняется условие, то состояние триггера единица, в противном случае – ноль	
		00	Измерение ≤ CMPL
		01	CMPL ≤ Измерение ≤ CMPH
		10	CMPH ≤ Измерение
		11	Зарезервировано
		Параметры CMPL и CMPH задаются в регистре DCMP	
CTM	9-8	Поле задания режима срабатывания выходного триггера	
		00	Многократный
		01	Однократный
		10	Многократный с гистерезисом
		11	Однократный с гистерезисом
CIE	4	Бит разрешения прерывания компаратора	
		0	Запрещено
		1	Разрешено. Прерывание генерируется каждый раз при одновременном выполнении условий CIC и CIM
CIC	3-2	Поле задания условия генерирования прерывания	
		00	Измерение ≤ CMPL
		01	CMPL ≤ Измерение ≤ CMPH
		10	CMPH ≤ Измерение
		11	Зарезервировано
CIM	1-0	Поле задания режима генерирования прерывания	
		00	Многократный
		01	Однократный
		10	Многократный с гистерезисом
		11	Однократный с гистерезисом
–	31-29, 27-25, 23-22, 15-13, 7-5	Зарезервировано	

DCMP – регистр диапазона компаратора

Смещение: DCd + 04h
Сброс: 0000_0000h

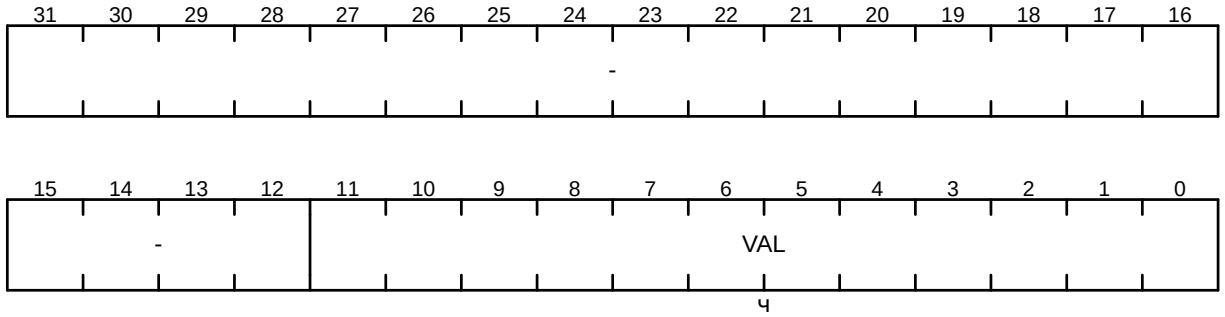


Поле	Биты	Описание
CMPH	27-16	Поле значения верхнего порога диапазона измерений
CMPL	11-0	Поле значения нижнего порога диапазона измерений
–	31-28, 15-12	Зарезервировано

Примечание – Всегда должно выполняться условие CMPL ≤ CMPH.

DDATA – регистр результата измерения компаратора

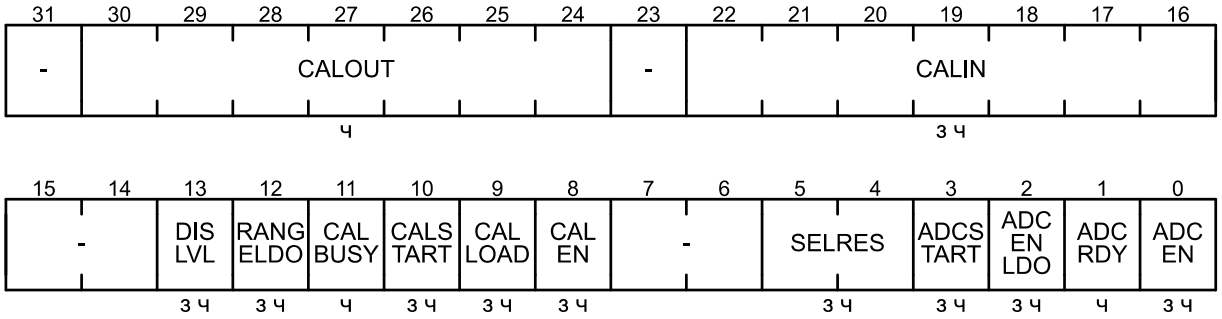
Смещение: DCd + 08h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	11-0	Значение результата измерения, которое последним использовалось компаратором при проверке на соответствие условиям CTC и CTM (см. регистр DCTL)
–	31-12	Зарезервировано

ACTL – массив регистров управления модулем АЦП

Смещение: ACTL + (4*m)h
Сброс: 0000_0000h

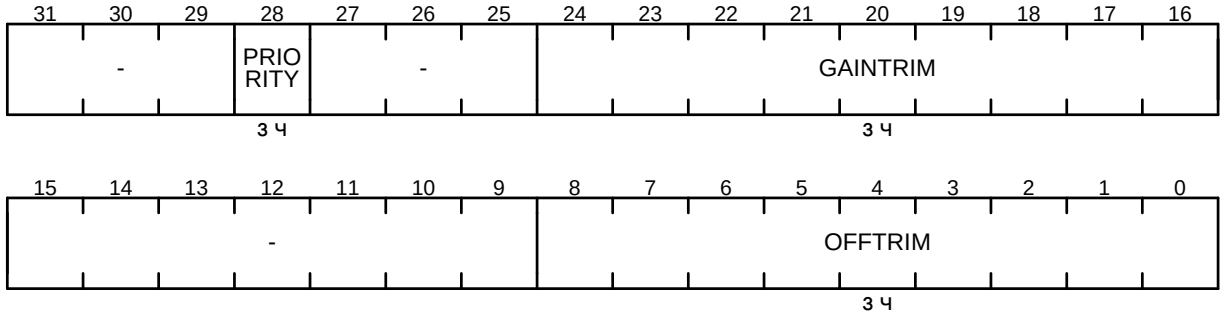


Поле	Биты	Описание	
CALOUT	30-24	Поправочное значение, полученное в результате калибровки. Диапазон значений от -64 до 63, величина в дополнительном коде	
CALIN	22-16	Поправочное значение для ручного занесения в схему калибровки с помощью бита CALLOAD. Диапазон значений от -64 до 63, величина вносится в дополнительном коде	
DISLVL	13	Отключение регуляторов уровня. По умолчанию данный бит установлен для развязки питания и уменьшения энергопотребления блока АЦП. При работе необходимо сбросить	
RANGELDO	12	Выбор поддиапазона напряжения LDO. Рекомендуется работа в поддиапазоне 1	
		0	поддиапазон от 0,81 В до 1,0 В
		1	поддиапазон от 1,0 В до 1,32 В
CALBUSY	11	Флаг активности процедуры калибровки. Сбрасывается автоматически	
		0	Калибровка не проводится
		1	Проводится калибровка
CALSTART	10	Старт проведения схемы внутренней калибровки АЦП	
		0	Нет реакции
		1	Старт калибровки
CALLOAD	9	Внесение значения поля CALIN во внутреннюю схему калибровки	
		0	Нет реакции
		1	Загрузка значения
CALEN	8	Включение схемы внутренней калибровки	
		0	Выключено
		1	Включено
SELRES	5-4	Выбор разрядности результата модуля АЦП	
		00b	6 бит
		01b	8 бит
		10b	10 бит
		11b	12 бит

Поле	Биты	Описание	
ADCSTART	3	Запуск АЦП. Запускается процедура инициализации, которая завершается установкой ADCRDY	
		0	Модуль АЦП не запущен
		1	Модуль АЦП запущен
ADCENLDO	2	Включение LDO АЦП	
ADCRDY	1	Флаг готовности АЦП к проведению измерений	
		0	АЦП выключен (ADCEN=0) либо в состоянии инициализации
		1	АЦП включен и готов к преобразованиям
ADCEN	0	Включение АЦП	
–	31, 23, 15-14, 7-6	Зарезервировано	

CHCTL – массив регистров настройки каналов

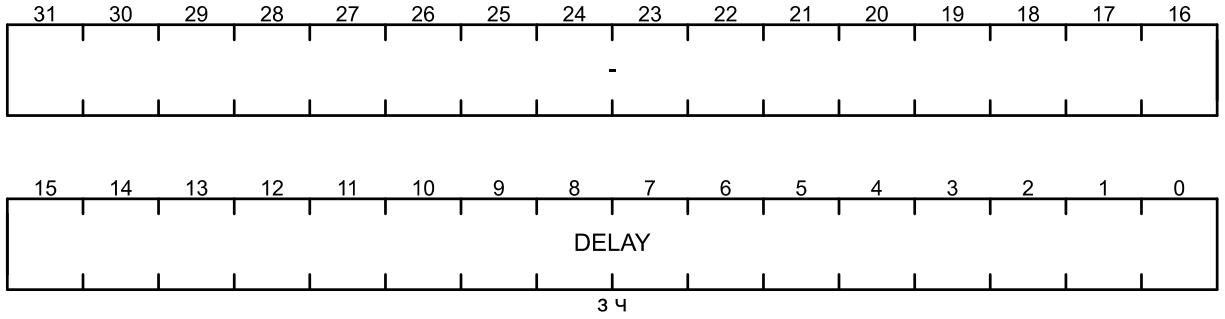
Смещение: CHCTL + (4*n)h
Сброс: 0000_0000h



Поле	Биты	Описание	
PRIORITY	28	Бит задания приоритета канала	
		0	Канал имеет стандартный приоритет
		1	Канал имеет повышенный приоритет
GAINTRIM	24-16	Поле задания коэффициента для коррекции усиления. Диапазон значений от –256 до 255 в 12-разрядном режиме, величина вносится в дополнительном коде: 100h соответствует -256, 000h – 0, 0FFh – 255. При других разрядностях результата диапазон сужается – подробнее в таблице 18.2.	
OFFTRIM	8-0	Поле задания коэффициента для коррекции смещения нуля. Диапазон значений от –256 до 255 в 12-разрядном режиме, величина вносится в дополнительном коде: 100h соответствует - 256, 000h – 0, 0FFh – 255. При других разрядностях результата диапазон сужается – подробнее в таблице 18.2.	
–	31-29, 27-25, 15-9	Зарезервировано	

CHDELAY – массив регистров настройки количества дополнительных циклов для каждого канала

Смещение: CHDELAY + (4*n)h
Сброс: 0000_0000h



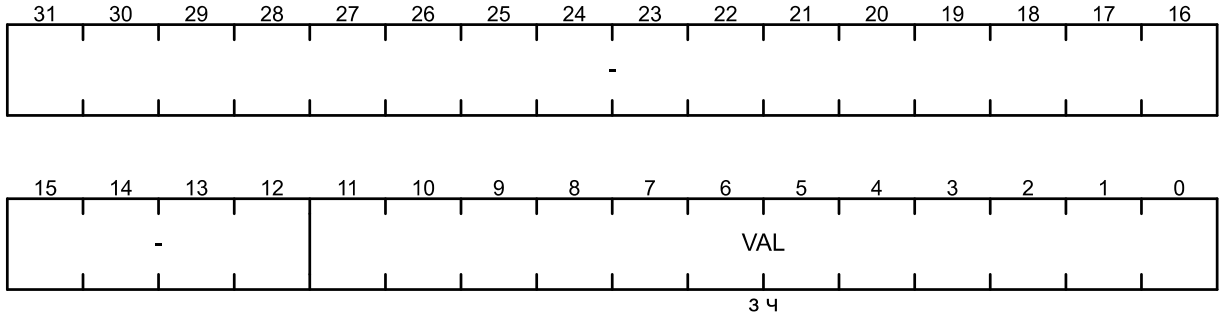
Поле	Биты	Описание
DELAY	15-0	Поле задания количества дополнительных циклов (время, на которое канал измерения подключается к внутренней зарядной емкости АЦП) для канала n

A.14 Регистры блока ЦАП

Базовый адрес: 5100_4000h Регистры блока DAC0
5100_5000h Регистры блока DAC1

DATA12 – регистр 12-разрядных выходных данных

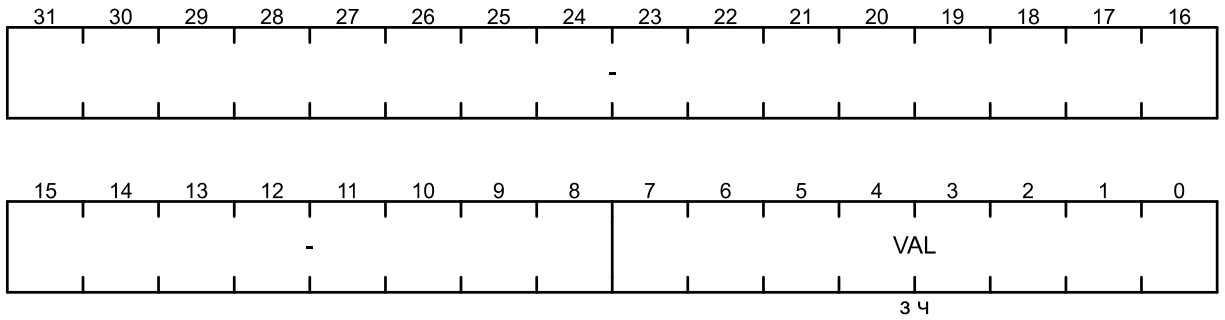
Смещение: + 00h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	11-0	Поле для записи 12-разрядных значений
–	31-12	Зарезервировано

DATA8 – регистр 8-разрядных выходных данных

Смещение: + 04h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	7-0	Поле для записи 8-разрядных значений
–	31-8	Зарезервировано

DATA_DUAL – регистр выходных данных в совместном режиме

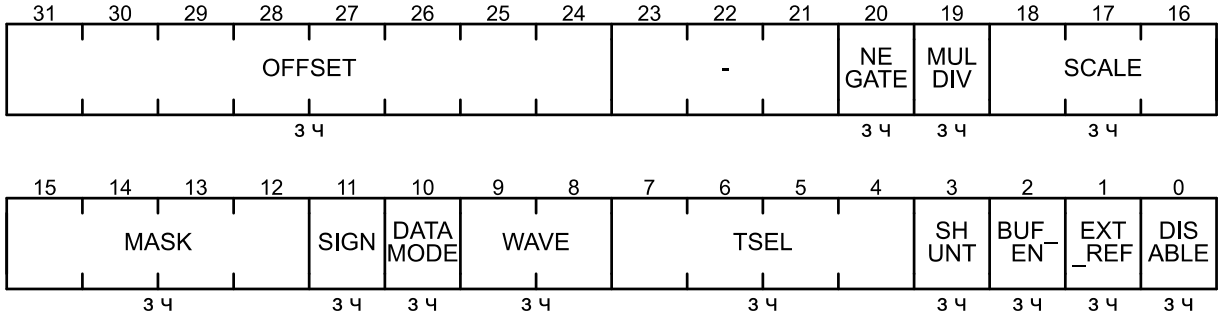
Смещение: + 08h
Сброс: 0000_0000h



Поле	Биты	Описание
DAC1	27-16	Поле записи значений для DAC1
DAC0	11-0	Поле записи значений для DAC0
–	31-28, 15-12	Зарезервировано

CONTROL – регистр управления ЦАП

Смещение: + 0Ch
Сброс: 0000_0001h



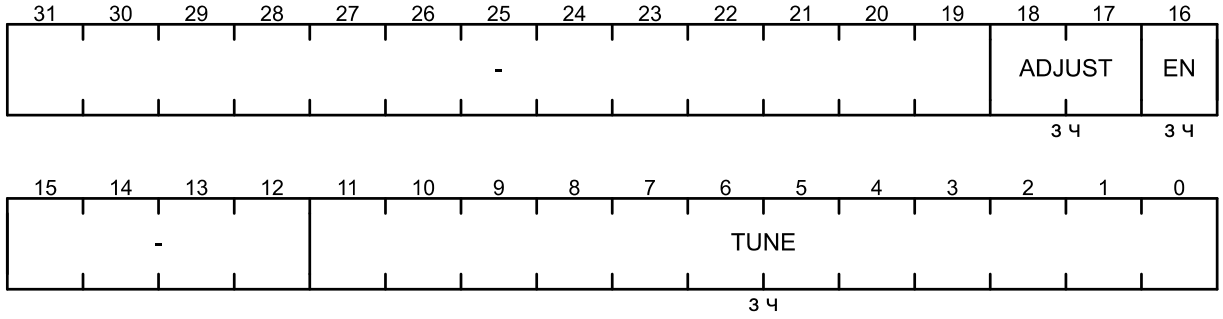
Поле	Биты	Описание	
OFFSET	31-24	Поле задания смещения	
NEGATE	20	Бит конвертации значения в дополнительный код	
MULDIV	19	Бит переключения между умножением и делением входного значения на степень числа 2, указанной в поле SCALE	
		0	Деление
		1	Умножение

Поле	Биты	Описание	
SCALE	18-16	Поле содержит значение степени числа 2 для умножения/деления входного значения. Результирующее значение рассчитывается как 2^{SCALE}	
		0	Отсутствие умножения/деления
		1	Умножение/деление на 2
		2	Умножение/деление на 4
		3	Умножение/деление на 8
		4	Умножение/деление на 16
		5	Умножение/деление на 32
		6	Умножение/деление на 64
		7	Умножение/деление на 128
MASK	15-12	Поле задания маски в режиме генерации шума (noise wave mode). Также поле используется для задания амплитуды в режиме генерации треугольной волны (triangle wave mode)	
		0000	Бит 0 LFSR размаскирован / Амплитуда равна 1
		0001	Биты [1:0] LFSR размаскированы / Амплитуда равна 3
		0010	Биты [2:0] LFSR размаскированы / Амплитуда равна 7
		0011	Биты [3:0] LFSR размаскированы / Амплитуда равна 15
		0100	Биты [4:0] LFSR размаскированы / Амплитуда равна 31
		0101	Биты [5:0] LFSR размаскированы / Амплитуда равна 63
		0110	Биты [6:0] LFSR размаскированы / Амплитуда равна 127
		0111	Биты [7:0] LFSR размаскированы / Амплитуда равна 255
		1000	Биты [8:0] LFSR размаскированы / Амплитуда равна 511
		1001	Биты [9:0] LFSR размаскированы / Амплитуда равна 1023
		1010	Биты [10:0] LFSR размаскированы / Амплитуда равна 2047
		>1011	Биты [11:0] LFSR размаскированы / Амплитуда равна 4096
SIGN	11	Бит включения знакового режима ЦАП	
		0	Беззнаковый режим. Данные воспринимаются как значения без знака.
		1	Знаковый режим. Данные воспринимаются как значения со знаком.
DATAMODE	10	Бит переключения между независимым и сдвоенным режимами работы ЦАП	
		0	Независимый режим
		1	Сдвоенный режим
WAVE	9-8	Поле включения режимов генерации волн каналом ЦАП	
		0	Генерация волн отключена
		1	Включение режима генерации шума
		2	Включение режима генерации треугольной волны
		3	Включение режима генерации волны по образцу

Поле	Биты	Описание	
TSEL	7-4	Поле выбора события для запуска ЦАП	
		0	Программный запрос. ЦАП будет запущен при установке бита SWTRIG
		1	Внутренний аппаратный запрос. ЦАП будет запущен при установке бита ENABLE регистра HWTRIG
		2	Запрос от одного из таймеров TMR32_0 – TMR32_3
		3	Запрос от одного из таймеров TMR32_4 – TMR32_7
		4	Запрос от одного из таймеров TMR32_8 – TMR32_11
		5	Запрос от одного из таймеров TMR32_12 – TMR32_15
		6	Запрос от портов GPIOA или GPIOB
		7	Запрос от портов GPIOC или GPIOD
		8	Запрос от портов GPIOE или GPIOF
		9	Запрос от порта GPIOG
		10	Запрос от канала A ШИМ0 – ШИМ15
		11	Запрос от канала B ШИМ0 – ШИМ15
		12-15	Зарезервировано
BUF_EN	2	Бит включения выходного буфера	
EXT_REF	1	Бит включения использования внешнего источника опорного напряжения	
		0	Используется внутренний источник опорного напряжения
		1	Используется источник опорного напряжения AVCC (3.3В)
DISABLE	0	Бит отключения модуля ЦАП. Значение по умолчанию – 1	
		0	Модуль ЦАП активен
		1	Модуль ЦАП выключен
–	23-21	Зарезервировано	

BGCTRL – регистр внутреннего источника опорного напряжения

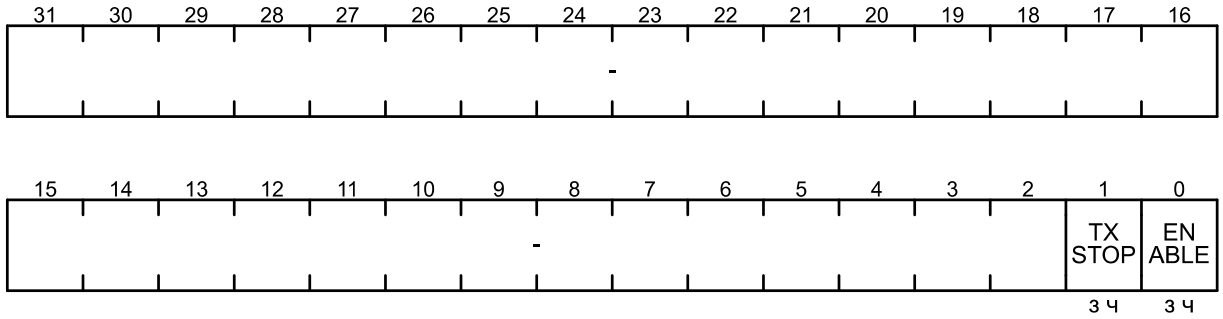
Смещение: + 10h
Сброс: 0000_0800h



Поле	Биты	Описание	
ADJUST	18-17	Поле регулировки выходного напряжения внутреннего источника опорного напряжения	
		0	Выходное напряжение 1,5 В
		1	Выходное напряжение 1,8 В
		2	Выходное напряжение 2,048 В
		3	Выходное напряжение 2,5 В
EN	16	Бит включения внутреннего источника опорного напряжения	
TUNE	11-0	Поле регулировки источника опорного напряжения. Значение по умолчанию – 2048	
–	31-19, 15-12	Зарезервировано	

DMACTRL – регистр управления DMA

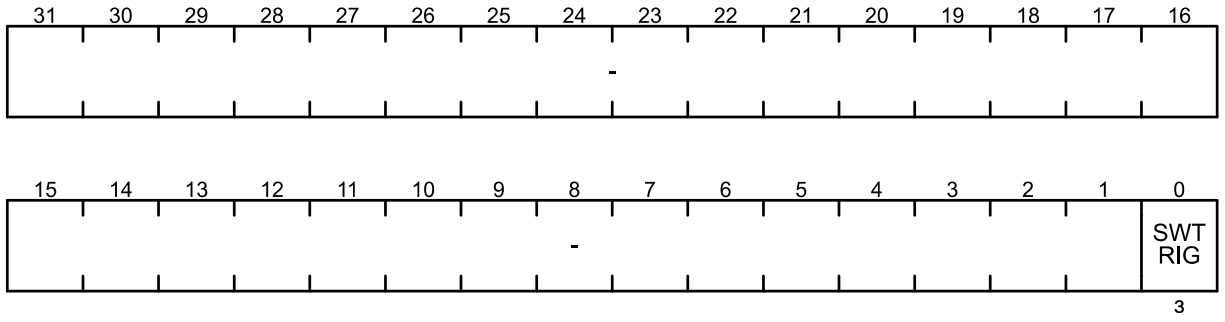
Смещение: + 14h
Сброс: 0000_0000h



Поле	Биты	Описание
TXSTOP	1	Бит автоматической очистки запроса DMA TX по окончанию всех транзакций. Если данный бит установлен, тогда при окончании загрузки всех данных, генерация запросов прекратится и бит ENABLE сбросится автоматически
ENABLE	0	Включение генерации запросов DMA. Для генерации запросов поле TSEL регистра CONTROL не должно быть равно нулю
–	31-2	Зарезервировано

SWTRIG – регистр управления программным запросом

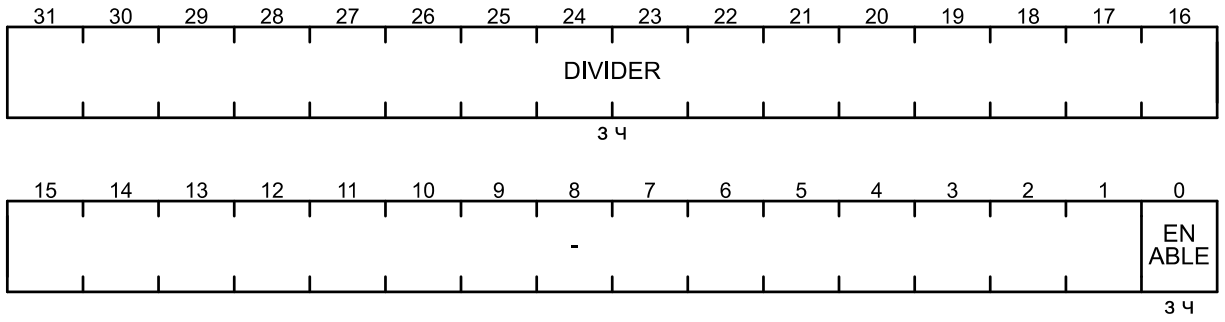
Смещение: + 18h
Сброс: 0000_0000h



Поле	Биты	Описание
SWTRIG	0	Бит программного запроса на запуск преобразования
–	31-1	Зарезервировано

HWTRIG – регистр управления генератором аппаратных запросов

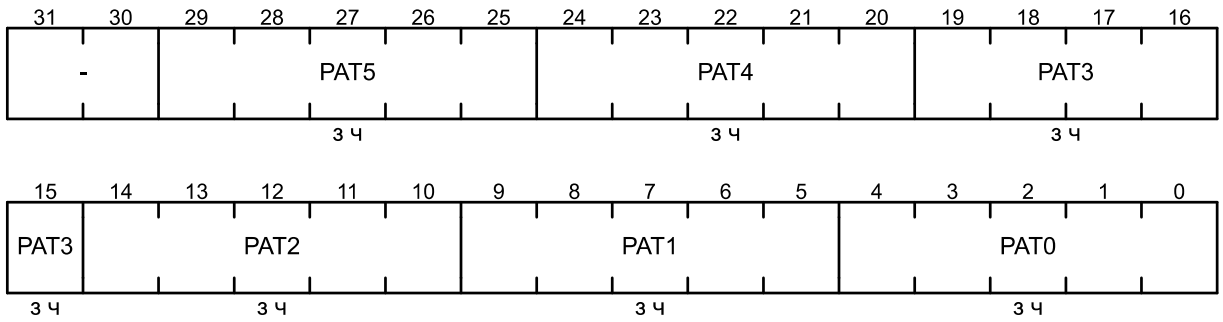
Смещение: + 1Ch
Сброс: 0010_0000h



Поле	Биты	Описание
DIVIDER	31-16	Поле записи значения делителя частоты. Значение по умолчанию – 16
ENABLE	0	Бит запуска генератора запросов
–	15-1	Зарезервировано

PATDORL – регистр младшей части отсчетов для генерации сигнала по образцу

Смещение: + 20h
Сброс: 3568_B0C0h

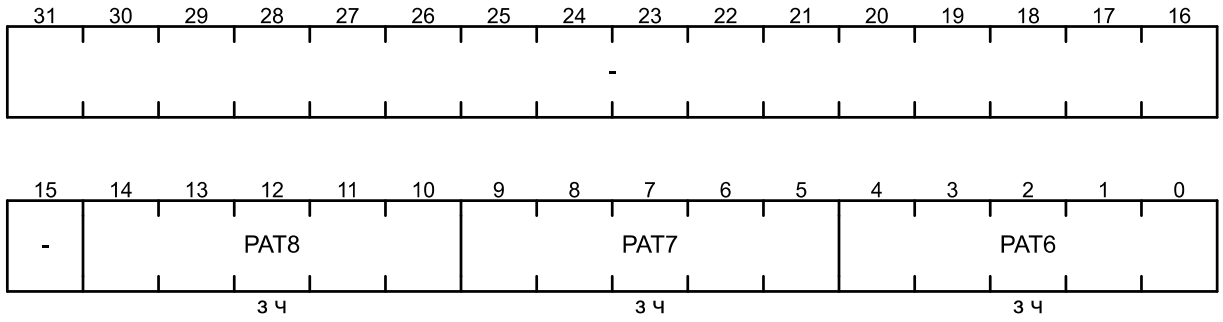


Поле	Биты	Описание
PAT5	29-25	Отсчет №5. Значение по умолчанию – 26
PAT4	24-20	Отсчет №4. Значение по умолчанию – 22
PAT3	19-15	Отсчет №3. Значение по умолчанию – 17
PAT2	14-10	Отсчет №2. Значение по умолчанию – 12
PAT1	9-5	Отсчет №1. Значение по умолчанию – 6
PAT0	4-0	Отсчет №0. Значение по умолчанию – 0
–	31-30	Зарезервировано

Примечание – По умолчанию отсчеты заполнены значениями для генерации синусоидального периодического сигнала.

PATDORH – регистр старшей части отсчетов для генерации сигнала по образцу

Смещение: + 24h
Сброс: 0000_7FDDh

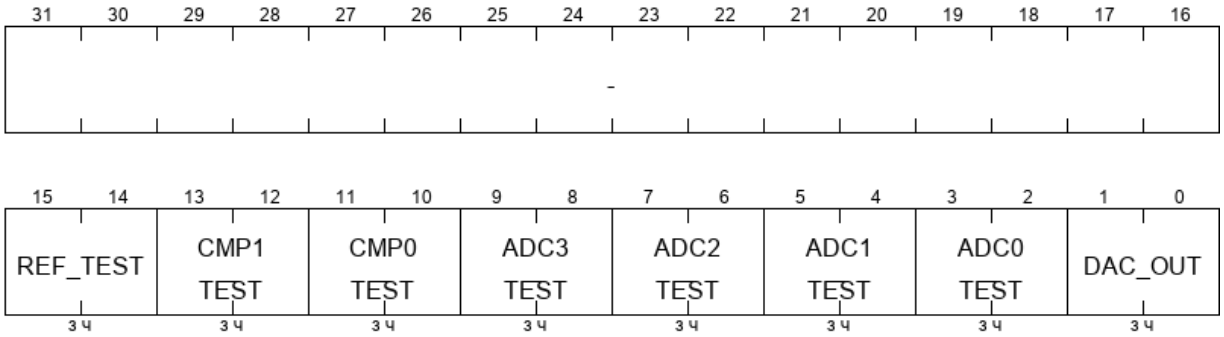


Поле	Биты	Описание
PAT8	14-10	Отсчет №8. Значение по умолчанию – 31
PAT7	9-5	Отсчет №7. Значение по умолчанию – 30
PAT6	4-0	Отсчет №6. Значение по умолчанию – 29
–	31-15	Зарезервировано

Примечание – По умолчанию отсчеты заполнены значениями для генерации синусоидального периодического сигнала.

TEST – регистр функционального тестирования ЦАП

Смещение: + 28h
Сброс: 0000_3FFCh



Поле	Биты	Описание	
REF_TEST	15-14	Выбор источника опорного напряжения	
		00b	Выход в третьем состоянии (Z)
		01b	Внешний источник опорного напряжения
		10b	Зарезервировано
		11b	Внутренний источник опорного напряжения
CMP1_TEST	31-12	Выбор выходного сигнала, подключаемого к блоку CMP1	
		00b	Нормальный режим (выход ЦАП)
		01b	Подключение выхода к напряжению питания VCC (PULLUP)
		10b	Подключение выхода к напряжению питания GND (PULLDOWN)
		11b	Выход в третьем состоянии (Z)
CMP0_TEST	11-10	Выбор выходного сигнала, подключаемого к блоку CMP0	
		00b	Нормальный режим (выход ЦАП)
		01b	Подключение выхода к напряжению питания VCC (PULLUP)
		10b	Подключение выхода к напряжению питания GND (PULLDOWN)
		11b	Выход в третьем состоянии (Z)
ADC3_TEST	9-8	Выбор выходного сигнала, подключаемого к модулю ADC3 блока АЦП	
		00b	Нормальный режим (выход ЦАП)
		01b	Подключение выхода к напряжению питания VCC (PULLUP)
		10b	Подключение выхода к напряжению питания GND (PULLDOWN)
		11b	Выход в третьем состоянии (Z)

Поле	Биты	Описание	
ADC2_TEST	7-6	Выбор выходного сигнала, подключаемого к модулю ADC2 блока АЦП	
		00b	Нормальный режим (выход ЦАП)
		01b	Подключение выхода к напряжению питания VCC (PULLUP)
		10b	Подключение выхода к напряжению питания GND (PULLDOWN)
		11b	Выход в третьем состоянии (Z)
ADC1_TEST	5-4	Выбор выходного сигнала, подключаемого к модулю ADC1 блока АЦП	
		00b	Нормальный режим (выход ЦАП)
		01b	Подключение выхода к напряжению питания VCC (PULLUP)
		10b	Подключение выхода к напряжению питания GND (PULLDOWN)
		11b	Выход в третьем состоянии (Z)
ADC0_TEST	3-2	Выбор выходного сигнала, подключаемого к модулю ADC0 блока АЦП	
		00b	Нормальный режим (выход ЦАП)
		01b	Подключение выхода к напряжению питания VCC (PULLUP)
		10b	Подключение выхода к напряжению питания GND (PULLDOWN)
		11b	Выход в третьем состоянии (Z)
DAC_OUT	1-0	Выбор сигнала, подаваемого на выход блока ЦАП	
		00b	Нормальный режим (выход ЦАП)
		01b	Подключение выхода к напряжению питания VCC (PULLUP)
		10b	Подключение выхода к напряжению питания GND (PULLDOWN)
		11b	Выход в третьем состоянии (Z)
–	31-16	Зарезервировано	

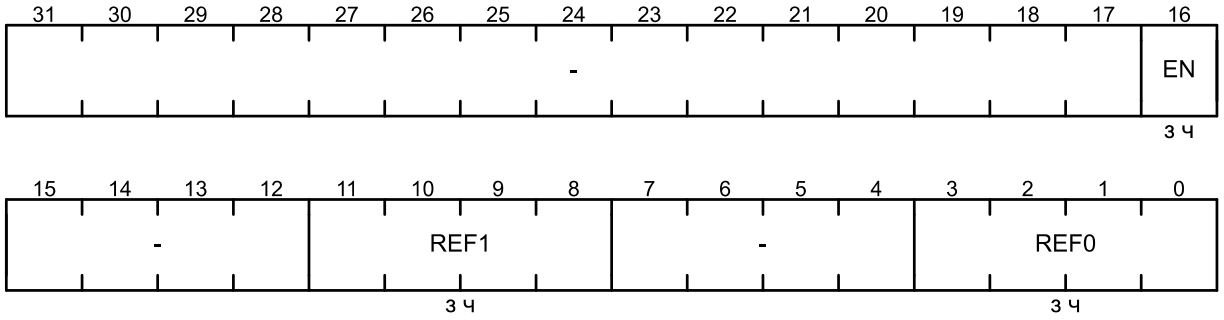
A.15 Регистры управления аналоговым компаратором

Базовый адрес:

5100_0000h	ACMP0
5100_1000h	ACMP1
5100_2000h	ACMP2
5100_3000h	ACMP3

DACCTL – регистр настройки ЦАП компаратора

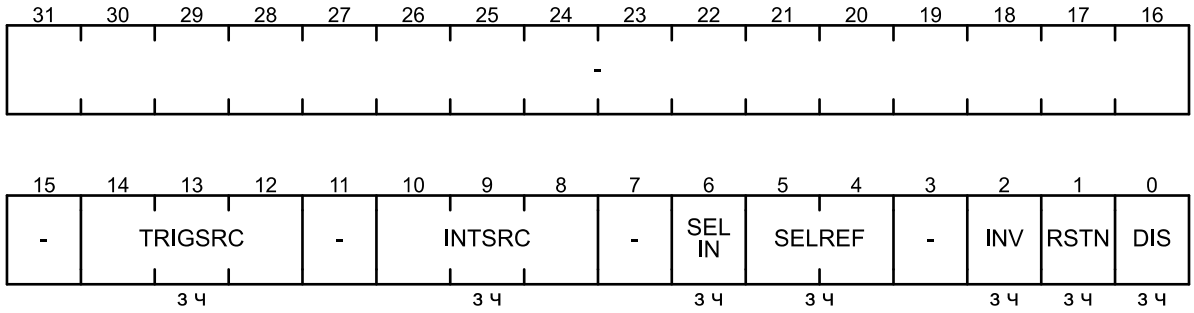
Смещение: + 0h
Сброс: 0h



Поле	Биты	Описание
EN	16	Включение ЦАП
		0 Включен
		1 Выключен
REF1	11-8	Уровень напряжения на выходе ЦАП REF1
REF0	3-0	Уровень напряжения на выходе ЦАП REF0
–	31-17, 15-12, 7-4	Зарезервировано

ACMPCTL – регистр настройки компаратора

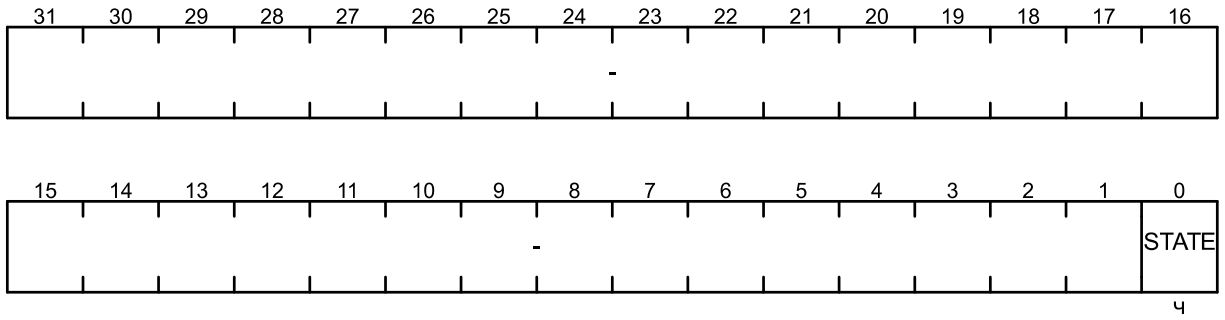
Смещение: + 04h
Сброс: 0000_0003h



Поле	Биты	Описание	
TRIGSRC	14-12	Выбор события для формирования сигнала TRIG0	
		0	Сигнал TRIG0 не формируется
		1	Положительный фронт
		2	Отрицательный фронт
		3	Любой фронт
		4	Уровень логического нуля
		5	Уровень логической единицы
INTSRC	10-8	Выбор события для формирования прерывания	
		0	Прерывание запрещено
		1	Положительный фронт
		2	Отрицательный фронт
		3	Любой фронт
		4	Уровень логического нуля
		5	Уровень логической единицы
SELIN	6	Выбор сигнала, подключаемого на положительный вход компаратора	
		0	Внешний сигнал IN_P
		1	Тестовый сигнал CMPn_TEST с выхода ЦАП
SELREF	5-4	Выбор сигнала, подключаемого на отрицательный вход компаратора	
		0	Вывод микроконтроллера IN0_N
		1	Выход ЦАП REF0
		2	Выход ЦАП REF1
INV	2	Инверсия выходного сигнала компаратора	
		0	На выходе прямой сигнал
		1	На выходе инверсный сигнал
RSTN	1	Сброс компаратора	
		0	Состояние сброса
		1	Состояние работы
DIS	0	Инверсный бит разрешения работы компаратора	
		0	Разрешено
		1	Запрещено
–	31-15, 11, 7, 3	Зарезервировано	

ACSTATUS – регистр статуса компаратора

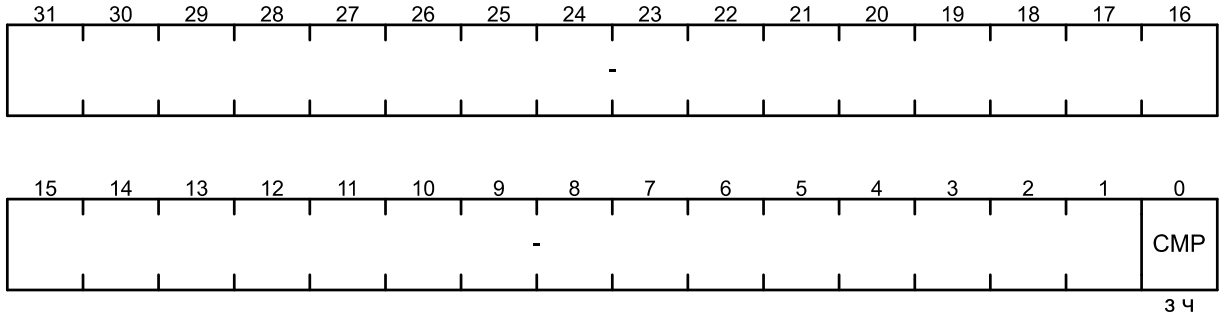
Смещение: + 0Ch
Сброс: 0h



Поле	Биты	Описание
STATE	0	Состояние выхода компаратора
–	31-1	Зарезервировано

INTEN – регистр разрешения прерываний компаратора

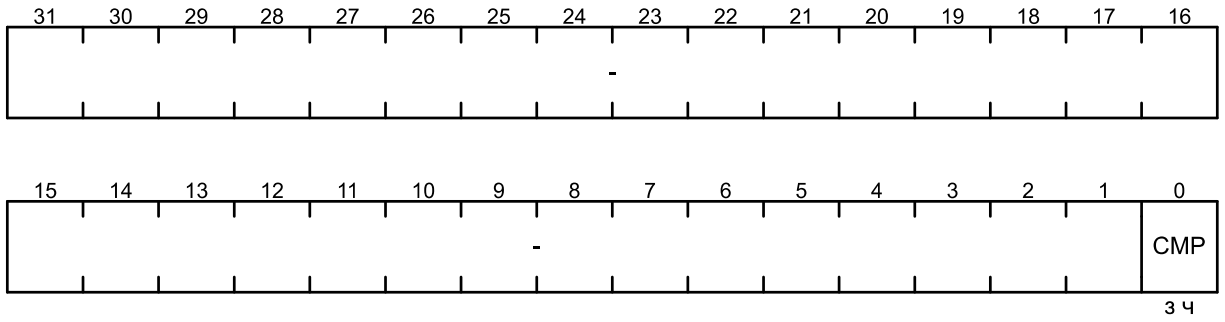
Смещение: + 10h
Сброс: 0h



Поле	Биты	Описание
CMP	0	Разрешение прерывания от компаратора
–	31-1	Зарезервировано

RIS – регистр состояний прерываний

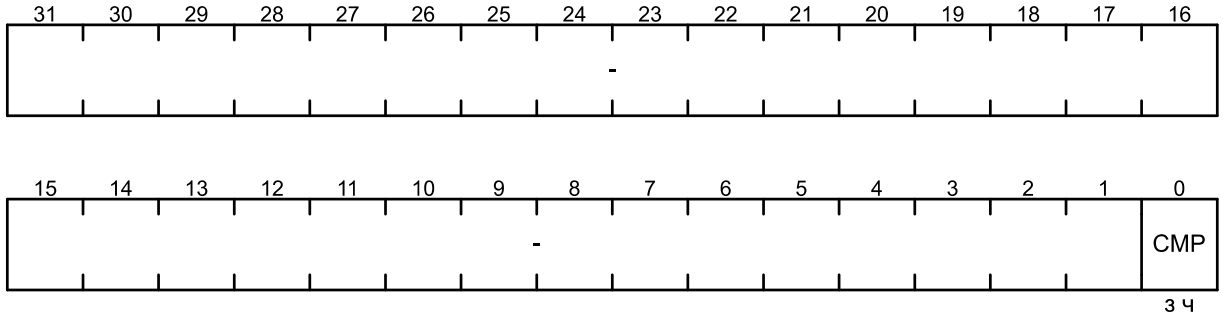
Смещение: + 14h
Сброс: 0h



Поле	Биты	Описание
CMP	0	Запрос на прерывание от компаратора
–	31-1	Зарезервировано

MIS – регистр состояний прерываний с маскированием

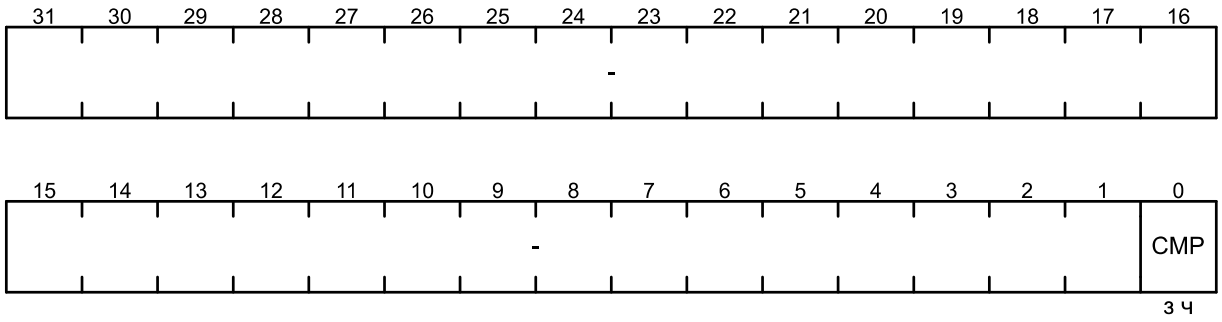
Смещение: + 18h
Сброс: 0h



Поле	Биты	Описание
CMP	0	Запрос на прерывание от компаратора
–	31-1	Зарезервировано

ICLR – регистр сброса прерываний

Смещение: + 1Ch
Сброс: 0h



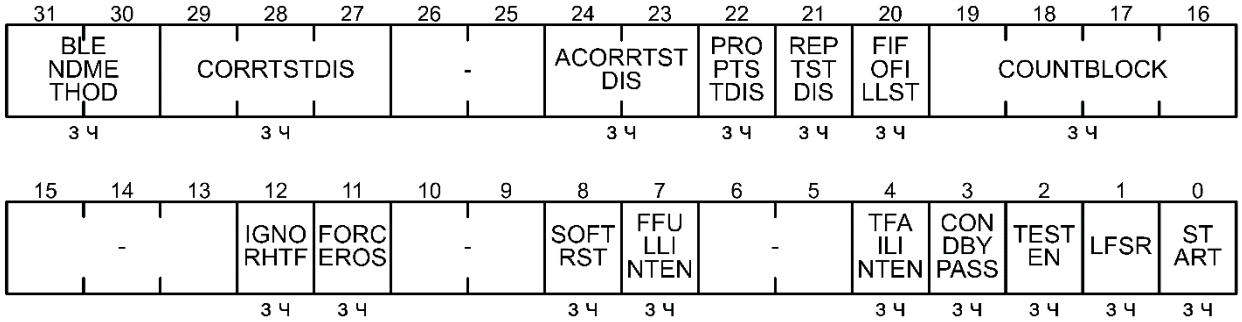
Поле	Биты	Описание
CMP	0	Сброс прерывания от компаратора
–	31-1	Зарезервировано

A.16 Регистры генератора случайных чисел ИГСЧ

Базовый адрес: 5000_8000h

CR – регистр конфигурации блока ИГСЧ

Смещение: + 00h
Сброс: 0004_0000h

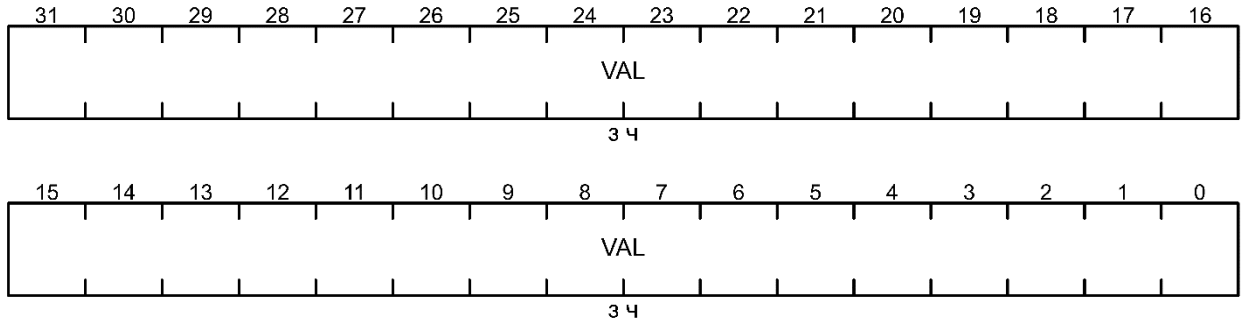


Поле	Биты	Описание	
BLENDMETHOD	31-30	Выбор метода смешивания	
		00	Объединение
		01	XOR_LEVEL_1
		10	XOR_LEVEL_2
		11	Исключение систематической ошибки Фон-Неймана
CORRTSTDIS	29-27	Отключение отдельных проверок со смещением в тесте на корреляцию – разные кольцевые генераторы (RO):	
		xx1	по сравнению с такой же выборкой другого кольцевого генератора RO→(0) нулевое смещение
		x1x	по сравнению с предыдущей выборкой другого кольцевого генератора RO→(-1) смещение -1
		1xx	по сравнению со следующей выборкой другого кольцевого генератора RO→(+1) смещение +1
ACORRTSTDIS	24-23	Отключение отдельных проверок со смещением в тесте на автокорреляцию – один и тот же кольцевой генератор (RO):	
		x1	по сравнению со следующим образцом того же самого RO→(0) нулевое смещение
		1x	по сравнению с более поздним образцом того же самого RO→(+1) смещение +1
PROPTSTDIS	22	Все тесты на пропорции отключаются с помощью этого бита	
REPTSTDIS	21	Все тесты на повторения отключаются с помощью этого бита	
FIFOFILLST	20	Разрешение записи выборок в FIFO во время запуска	
COUNTBLOCK	19-16	Число 128-битных блоков, используемых в функции последующей обработки. Нулевое значение не допускается	
IGNORHTF	12	Результаты тестов работоспособности во время запуска и во время работы не влияют на смену состояния КА (конечного автомата). Он также обходит фазу StartUp КА.	

Поле	Биты	Описание	
FORCEROS	11	Принудительный запуск генераторов, когда FIFO заполнен	
SOFTRST	8	Очистка содержимого информационного канала и переход КА в состояние сброса	
		0	Нормальный режим
		1	Компоненты информационного канала – тесты работоспособности, функция обработки и FIFO – очищаются. Этот бит не очищается автоматически. Регистры конфигурации сохраняют свои значения. Отправляет КА в состояние сброса.
FFULLINTEN	7	Включить прерывание, если FIFO заполнен.	
TFAILINTEN	4	Включить прерывание, если какой-либо из тестов работоспособности провален	
CONDBYPASS	3	0	Используется функция обработки (обычный режим)
		1	Функция обработки пропускается (для непосредственного наблюдения за источником энтропии)
TESTEN	2	Выбор входных данных для функции обработки и непрерывных тестов:	
		0	Источник шума (обычный режим)
		1	Регистр с тестовыми данными (тестовый режим)
LFSR	1	Выбор между ИГСЧ с асинхронными независимыми генераторами (при значении «0») и псевдослучайным генератором с синхронными генераторами для целей моделирования (при значении «1»).	
START	0	Запуск/включение ИГСЧ	
–	26-25, 15-13, 10-9, 6-5	Зарезервировано	

FIFOLEV – регистр уровня заполнения FIFO

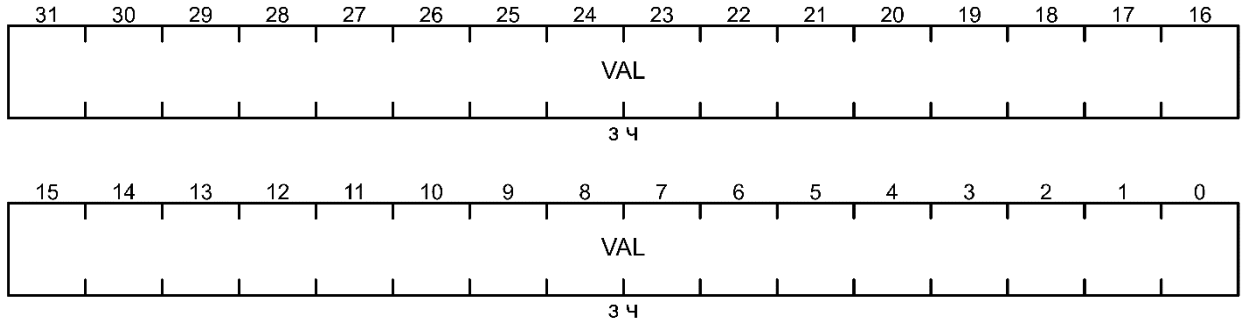
Смещение: + 04h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Количество доступных 32-битных случайных слов в буфере FIFO. Любая запись в этот регистр очищает флаг FifoFull в регистре состояния.

FIFOTHR – регистр порогового значения FIFO

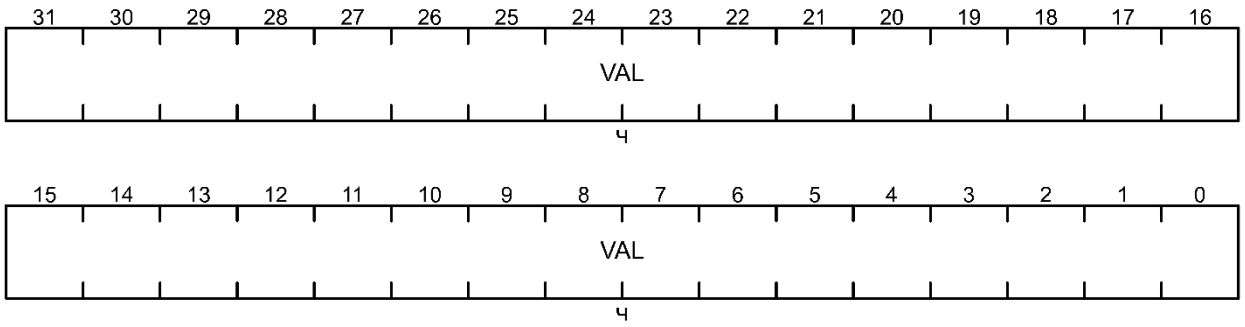
Смещение: + 08h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Пороговый уровень буфера FIFO, ниже которого модуль начинает пополнять FIFO Выражен в количестве 128-битных блоков.

FIFODEP – регистр максимального количества слов в FIFO

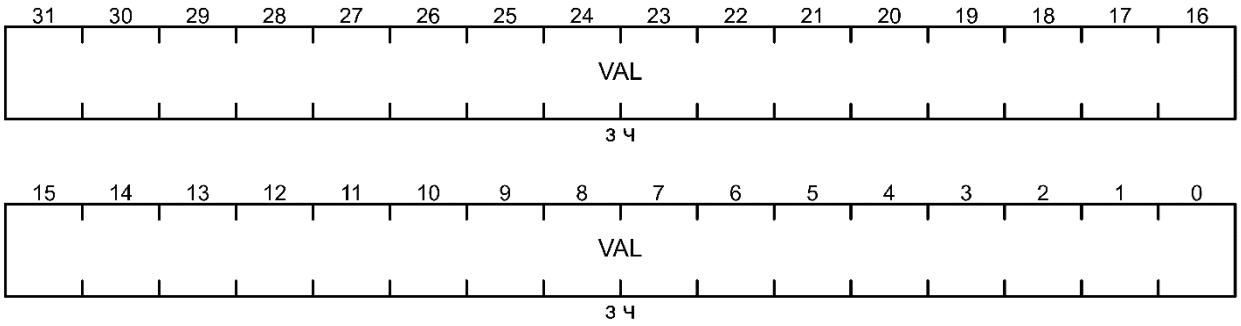
Смещение: + 0Ch
Сброс: 0000_0100h



Поле	Биты	Описание
VAL	31-0	Максимальное количество 32-разрядных слов, которые могут быть сохранены в FIFO. При чтении возвращается значение 256.

KEY0 – регистр ключа 0

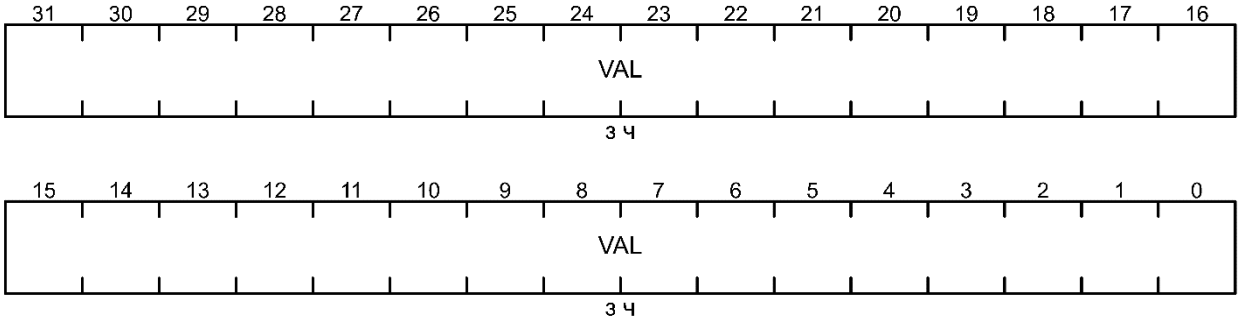
Смещение: + 10h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Ключ 0. Байты ключа AES [0-3] (Набор битов четырех регистров ключей формирует 128-битный ключ AES, используемый для функции обработки)

KEY1 – регистр ключа 1

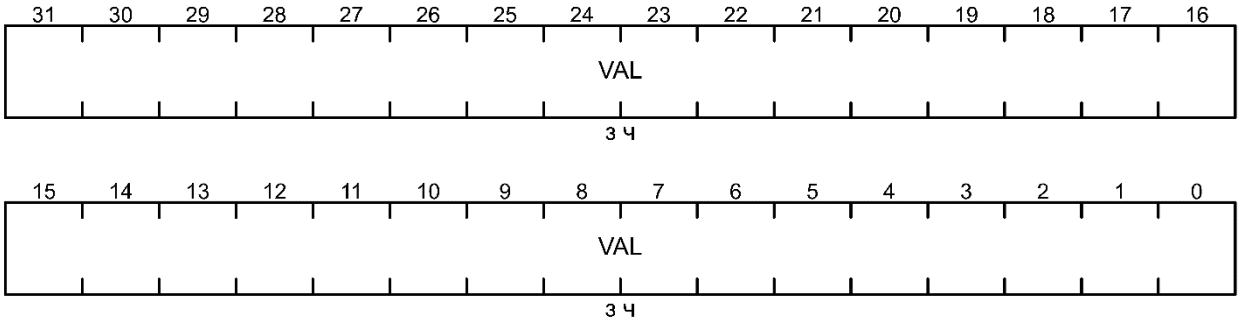
Смещение: + 14h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Ключ 1. Байты ключа AES [4-7] (Набор битов четырех регистров ключей формирует 128-битный ключ AES, используемый для функции обработки)

KEY2 – регистр ключа 2

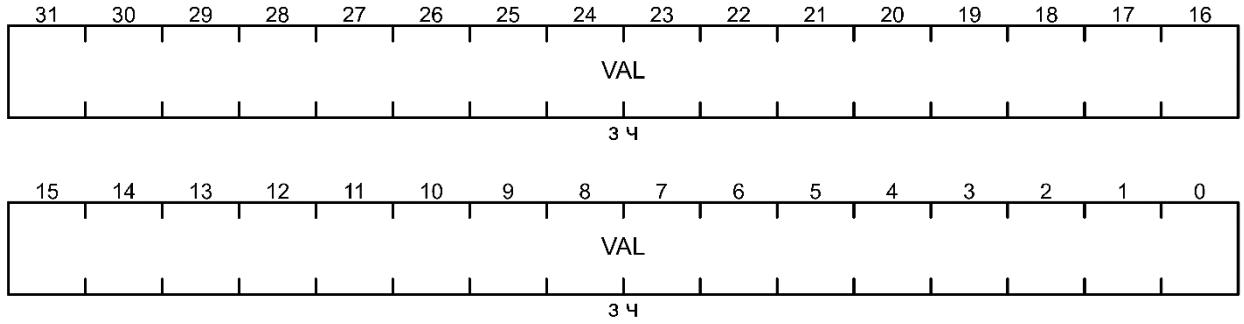
Смещение: + 18h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Ключ 2. Байты ключа AES [8-11] (Набор битов четырех регистров ключей формирует 128-битный ключ AES, используемый для функции обработки)

KEY3 – регистр ключа 3

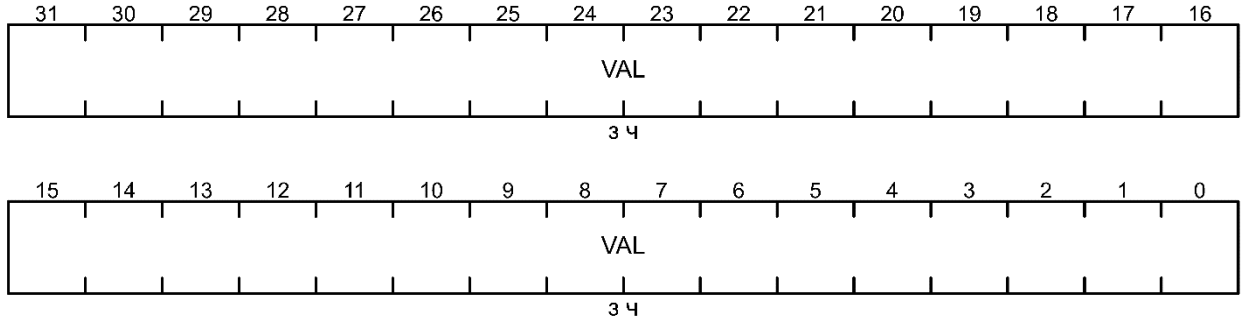
Смещение: + 1Ch
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Ключ 3. Байты ключа AES [12-15] (Набор битов четырех регистров ключей формирует 128-битный ключ AES, используемый для функции обработки)

TESTDATA – регистр с тестовыми данными

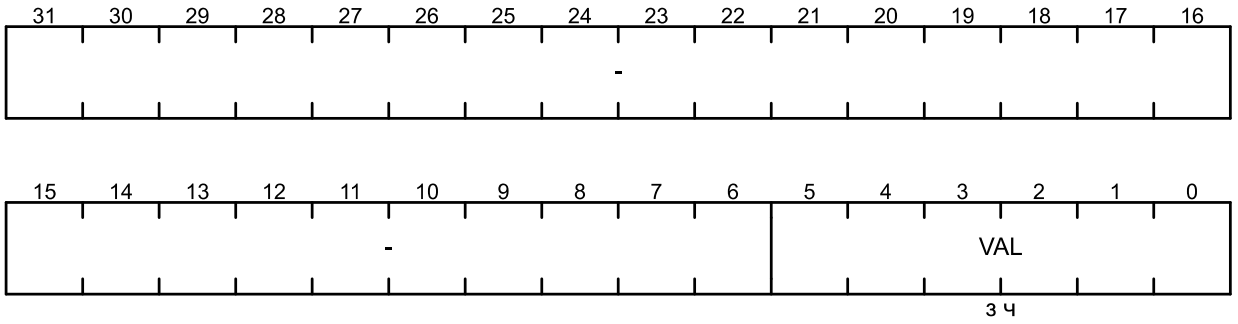
Смещение: + 20h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Передача известных данных в функцию обработки или в непрерывные тесты

REPTSTCUTOFF – регистр предельного значения для теста на повторение

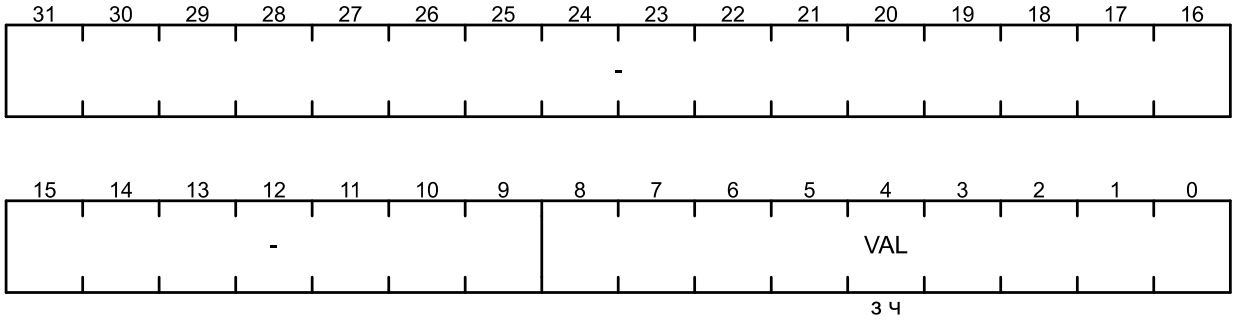
Смещение: + 24h
Сброс: 0000_0006h



Поле	Биты	Описание
VAL	5-0	Предельное значение для теста на повторение
-	31-6	Зарезервировано

PROPTESTCUTOFF - регистр предельного значения для теста на пропорции

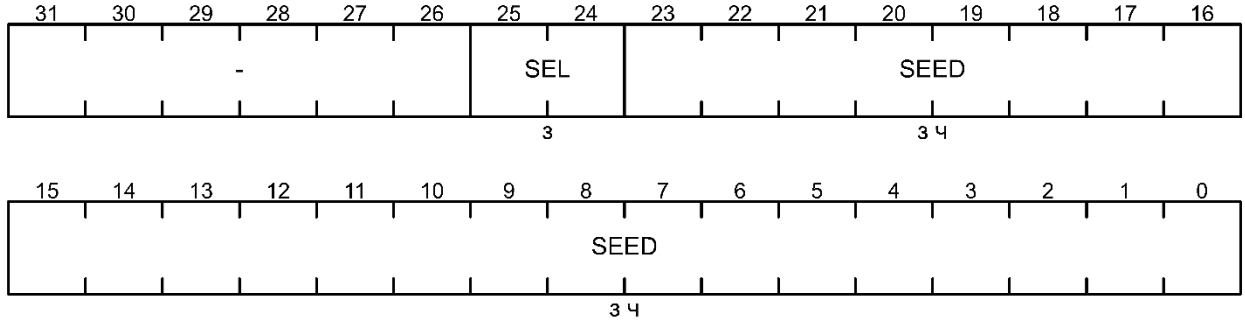
Смещение: + 28h
Сброс: 0000_003Eh



Поле	Биты	Описание
VAL	8-0	Предельное значение для тестов на пропорции
-	31-9	Зарезервировано

LFSRSEED – регистр сдвига с линейной обратной связью

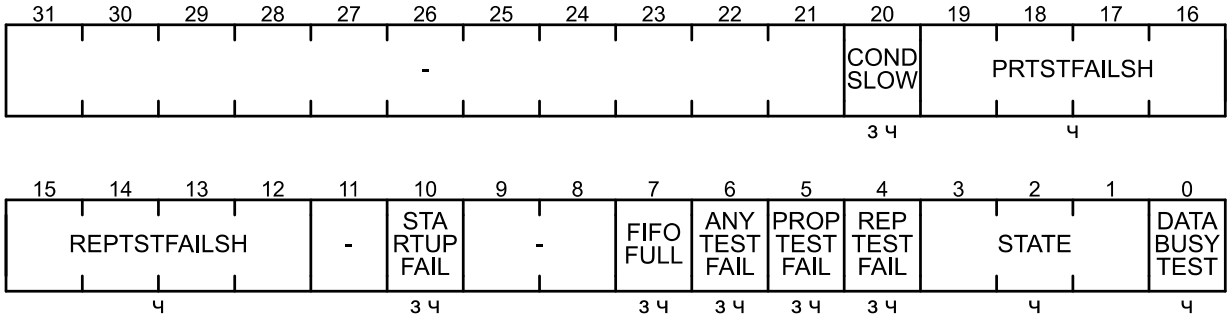
Смещение: + 2Ch
Сброс: 00FF_FFFFh



Поле	Биты	Описание
SEL	25-24	Общий индекс LFSR, для которого следует использовать значение инициализации
SEED	23-0	Начальное значение LFSR при инициализации
-	31-26	Зарезервировано

STAT – регистр состояния ИГСЧ

Смещение: + 30h
Сброс: 0000_0000h

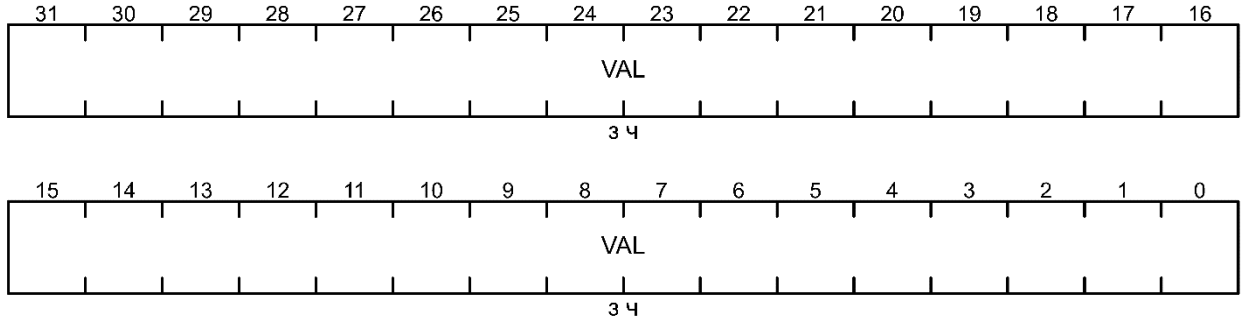


Поле	Биты	Описание
CONDSLOW	20	Функция обработки прорабатывает данные медленнее, чем они ей выдаются
PRTSTFAILSH	19-16	Доля проваленных тестов на пропорции
REPTSTFAILSH	15-12	Доля проваленных тестов на повторения
STARTUPFAIL	10	Сбой теста (тестов) запуска
FIFOFULL	7	Статус полного буфера FIFO
ANYTESTFAIL	6	Любой из включенных тестов работоспособности завершился провалом
PROPTESTFAIL	5	Провал теста на пропорции
REPTESTFAIL	4	Провал теста (тестов) на повторения

Поле	Биты	Описание	
STATE	3-1	Состояние контрольного автомата КА	
		000	Сброс
		001	Запуск
		010	Холостой ход / FIFO заполнен
		011	Не используется
		100	Заполннение FIFO
		101	Ошибка
		110	Не используется
		111	Не используется
DATABUSYTEST	0	Устанавливается во время процесса записи данных в регистр TESTDATA	
-	31-21, 11, 9-8	Зарезервировано	

WARMPERIOD – регистр времени прогрева

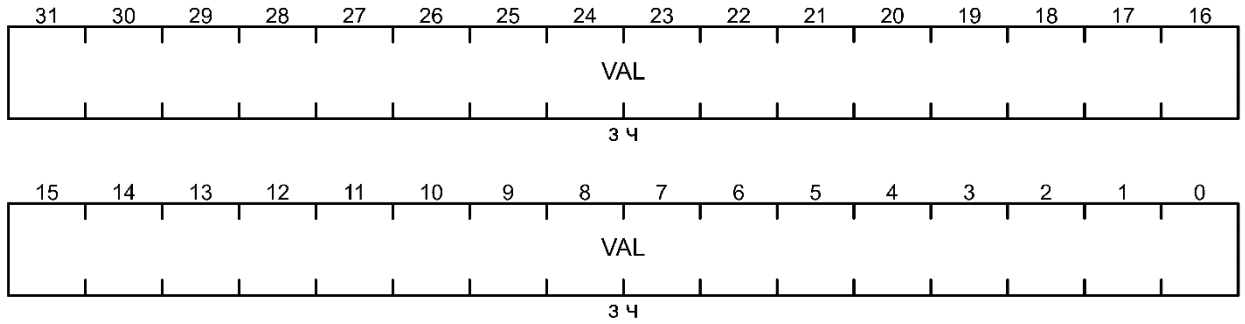
Смещение: + 34h
Сброс: 0000_0200h



Поле	Биты	Описание
VAL	31-0	Количество тактов, ожидаемых во время процесса прогрева

DISOSC – регистр отключения колец генератора

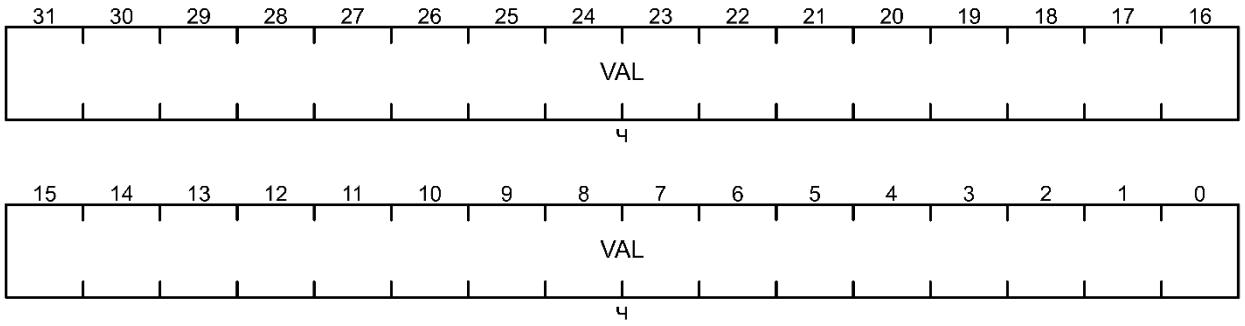
Смещение: + 38h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Независимое отключение кольца генератора от 0 до N-1. По одному биту на каждое кольцо.

SAMPERIOD – регистр периода дискретизации

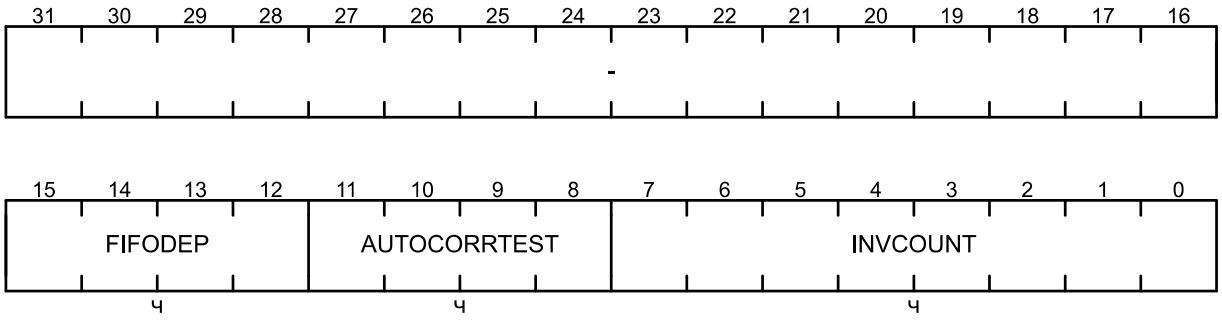
Смещение: + 44h
Сброс: 0000_0FFFh



Поле	Биты	Описание
VAL	31-0	Количество тактов ожидания между моментами выборки (дискретизации).

HWCFG – регистр аппаратной конфигурации

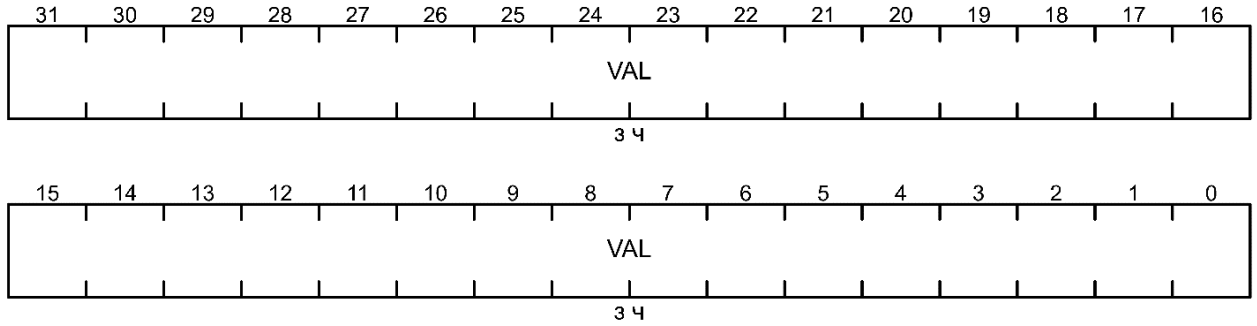
Смещение: + 58h
Сброс: 0000_8105h



Поле	Биты	Описание
FIFODEPTH	15-12	Значение глубины буфера FIFO. При чтении возвращается значение 8
AUTOCORRTEST	11-8	Количество тестов на автокорреляцию. При чтении возвращается значение 1
INVCOUNT	7-0	Количество инверторов. При чтении возвращается значение 5
-	31-16	Зарезервировано

COOLDPERIOD – регистр значения периода ожидания в холодном режиме

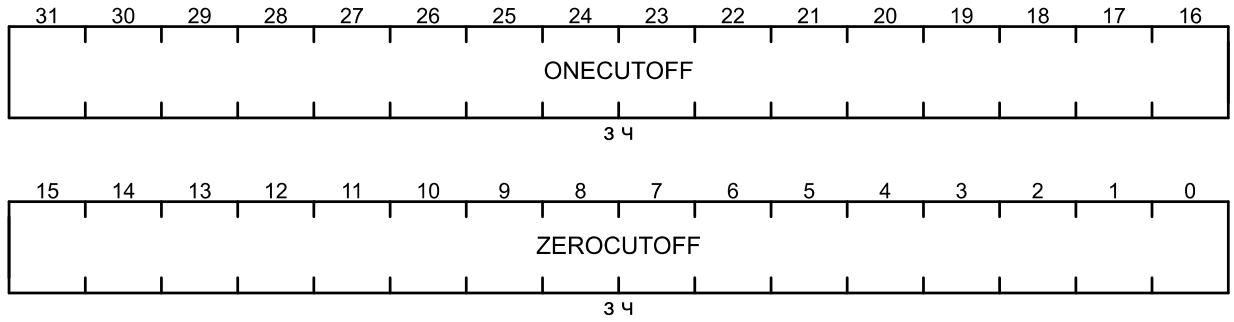
Смещение: + 5Ch
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Количество тактов во время цикла ожидания в холодном режиме

AUTOCORRTESTCUTOFF0 – регистр граничных значений для теста на автокорреляцию

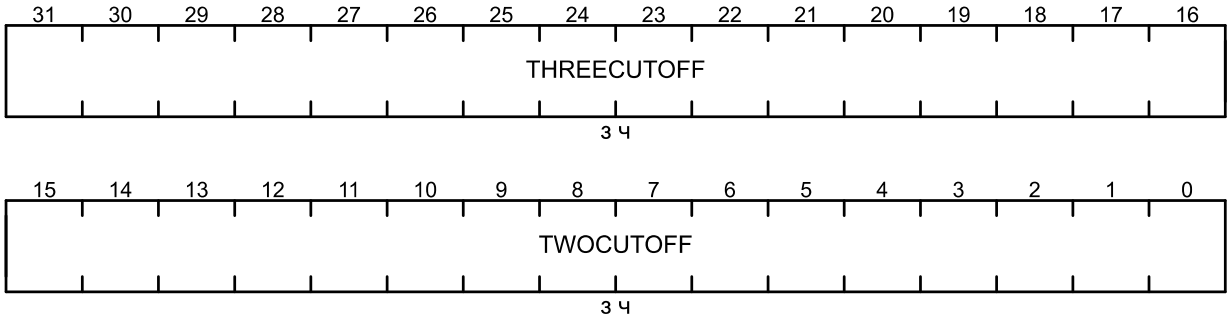
Смещение: + 60h
Сброс: 007F_007Fh



Поле	Биты	Описание
ONECUTOFF	31-16	Граничное значение для задержки выборки +1 в тесте на автокорреляцию
ZEROCUTOFF	15-0	Граничное значение для задержки выборки 0 в тесте на автокорреляцию

AUTOCORRTESTCUTOFF1 - регистр граничных значений для теста на автокорреляцию

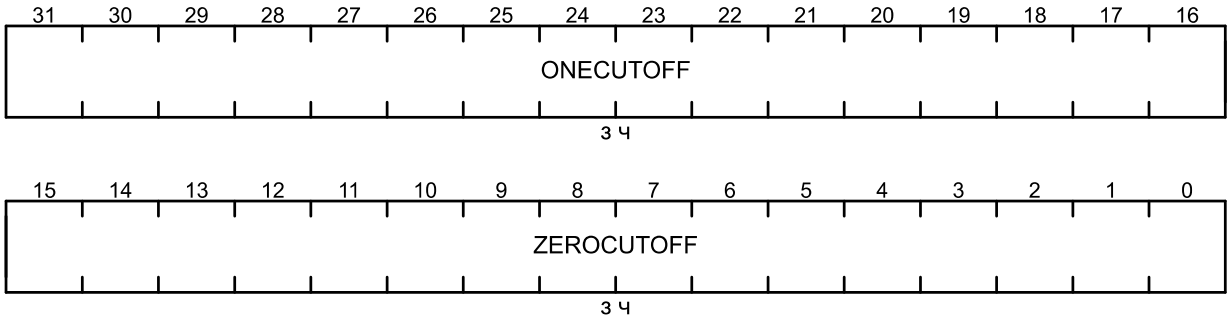
Смещение: + 64h
Сброс: 007F_007Fh



Поле	Биты	Описание
THREECUTOFF	31-16	Граничное значение для задержки выборки +3 в тесте на автокорреляцию
TWOCUTOFF	15-0	Граничное значение для задержки выборки +2 в тесте на автокорреляцию

CORRTESTCUTOFF0 - регистр граничных значений для теста на корреляцию

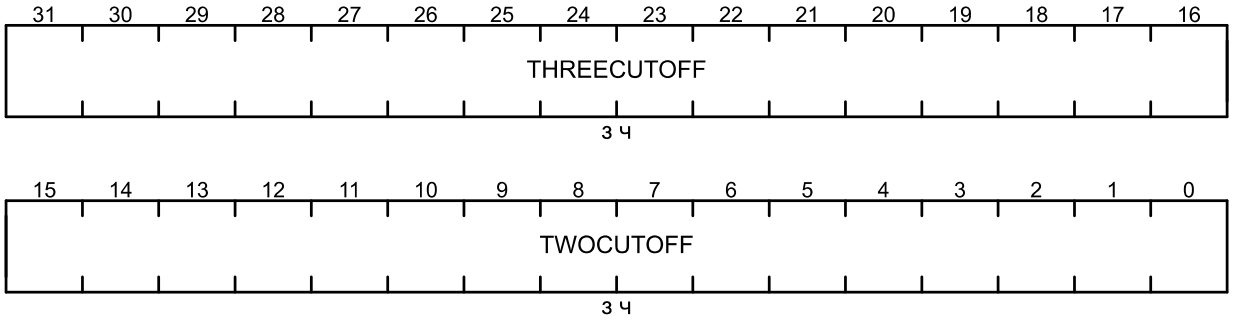
Смещение: + 68h
Сброс: 007F_007Fh



Поле	Биты	Описание
ONECUTOFF	31-16	Граничное значение для сдвига/задержки выборки +/-1 в тесте на корреляцию
ZEROCUTOFF	15-0	Граничное значение для сдвига/задержки выборки 0 в тесте на корреляцию

CORRTESTCUTOFF1 - регистр граничных значений для теста на корреляцию

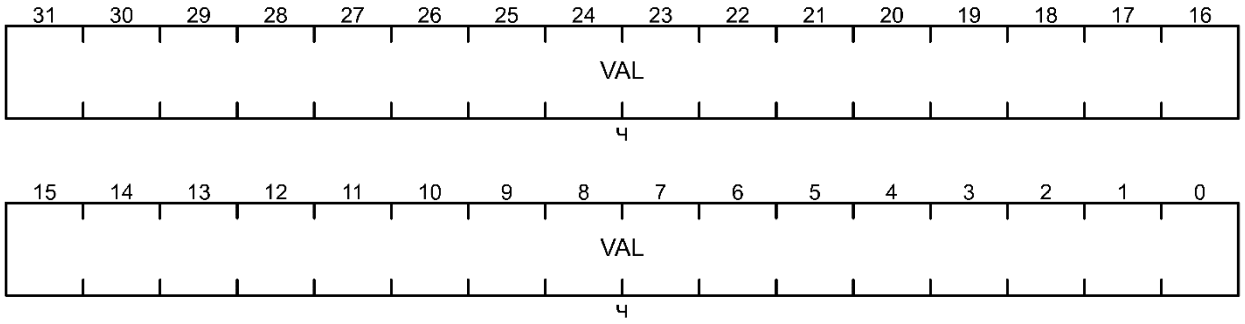
Смещение: + 6Ch
Сброс: 007F_007Fh



Поле	Биты	Описание
THREECUTOFF	31-16	Граничное значение для сдвига/задержки выборки +/-3 в тесте на корреляцию
TWOCUTOFF	15-0	Граничное значение для сдвига/задержки выборки +/-2 в тесте на корреляцию

AUTOCORRTESTFAILED – регистр, хранящий значения индексов кольцевых генераторов, проваливших тест на автокорреляцию

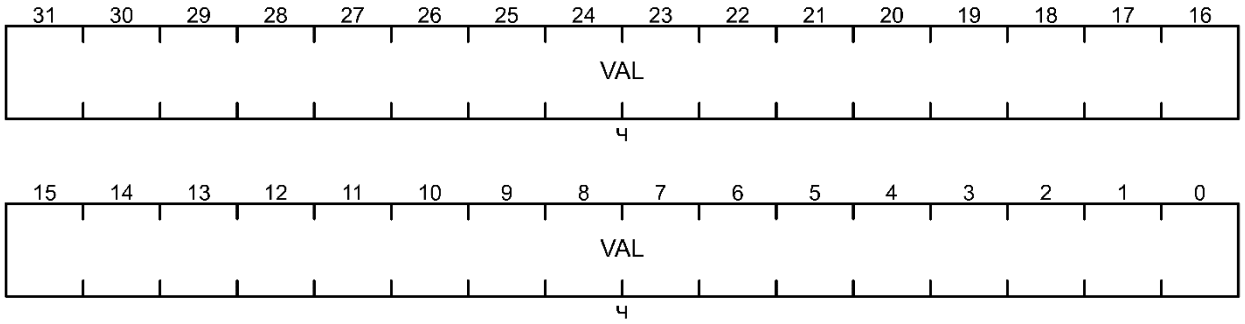
Смещение: + 70h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Кольцевой генератор/генераторы, провалившие тест на автокорреляцию

CORRTESTFAILED - регистр, хранящий значения индексов кольцевых генераторов, проваливших тест на корреляцию

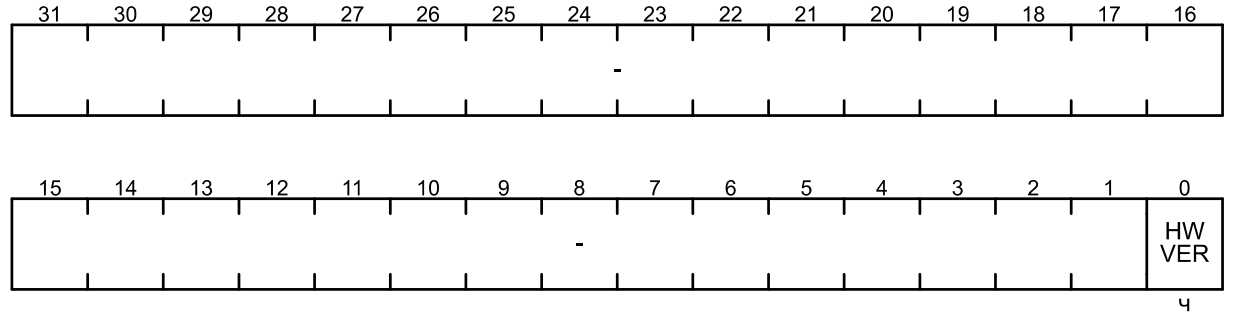
Смещение: + 74h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Кольцевые генераторы, провалившие тест на корреляцию

HWVERSION – регистр версии оборудования

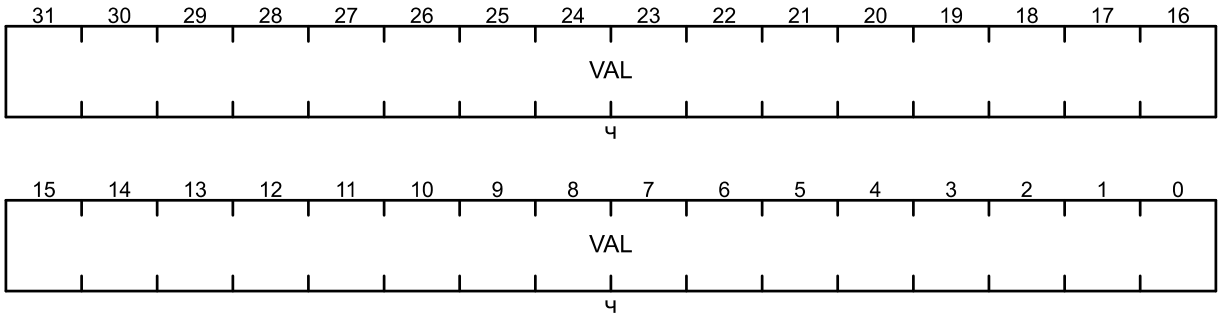
Смещение: + 7Ch
Сброс: 0000_0001h



Поле	Биты	Описание
HWVER	0	Установлено в «1» для этой версии
-	31-1	Зарезервировано

FIFO<i> – регистр буфера FIFO

Смещение: +80 +(4*i)h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Ячейка буфера FIFO

Примечание – i – номер регистра FIFO от 0 до 31.

A.17 Регистры блока HASH

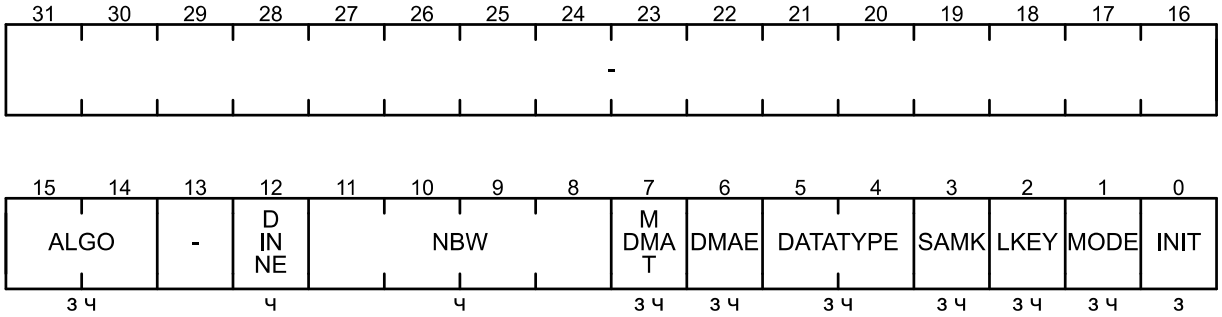
Базовый адрес: 4106_0000h

Смещение: + 040h (HR) Регистры хеш-суммы

Примечание – n – номер регистра хеш-суммы 0 – 7

CR - регистр управления

Смещение: + 00h
Сброс: 0000_0000h



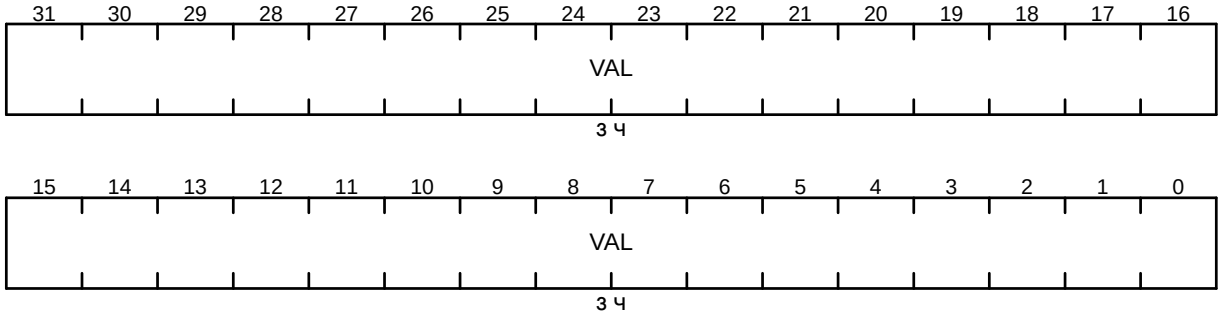
Поле	Биты	Описание	
ALGO	15-14	Выбор алгоритма для хеш-функции:	
		0	SHA-1
		1	MD5
		2	SHA-224
		3	SHA-256
DINNE	12	Регистр DIN не пуст Этот бит устанавливается, когда регистр DIN содержит допустимые данные (означает, что он хотя бы единожды был записан). Очищается либо установкой бита INIT (при инициализации) или при установке бита DCAL (окончание обработки сообщения) в «1».	
		0	Нет допустимых данных в DIN регистре
		1	Входной буфер содержит хотя бы одно слово данных

Поле	Биты	Описание	
NBW	11-8	Количество слов в FIFO Это битовое поле отображает количество слов, в сообщении которые были помещены во входное FIFO. NBW инкрементируется (+1) когда происходит запись в DIN регистр при установленном DINNE в «1». Поле сбрасывается в «0000» при установке бита INIT в «1» или, когда начинается окончательный подсчёт хеш-суммы сообщения (DCAL устанавливается в «1» или DMA заканчивает последнюю транзакцию).	
		0	DINNE=0: Нет доступных слов в буфере DIN (буфер пуст, ни в DIN регистре, ни во входном FIFO нет доступных данных)
			DINNE=1: 1 слово было записано в буфер DIN (DIN регистр содержит 1 слово, входное FIFO пусто)
		1	2 слова записано в буфер DIN (DIN регистр и входное FIFO содержат по 1 слову)
		…	
		15	16 слов было записано в буфер DIN
MDMAT	7	Множественная передача ПДП. Бит устанавливается при подсчёте хеш-сумм больших файлов используя механизм прямого доступа к памяти.	
		0	DCAL автоматически устанавливается после окончания передачи данных с использованием ПДП
		1	DCAL автоматически не устанавливается после окончания передачи данных с использованием ПДП
DMAE	6	Включение механизма DMA 0: Передачи DMA выключены. 1: Включение передач DMA. Запрос DMA будет послан сразу, как только хеш-процессор будет готов принять данные.	
DATATYPE	5-4	Выбор формата данных Определяет формат данных для загрузки в регистр DIN:	
		0	32-битные данные. Эти данные пишутся в регистр DIN и напрямую используются в вычислении, без изменения порядка следования битов.
		1	16-битные данные, полуслова. Данные, записываемые в регистр DIN подразумеваются как 2 полуслова, и меняются местами до вычисления.
		2	8-битные данные, байты. Данные, записываемые в регистр DIN подразумеваются как 4 байта, и меняются местами до вычисления.
		3	битовые данные или битовая строка. Данные, записываемые в регистр DIN подразумеваются как 32 бита (первый бит строки на позиции 0), и меняются местами до вычисления (первый бит строки на позиции 31).

Поле	Биты	Описание	
SAMK	3	Одинаковый ключ Бит позволяет хеш-процессору использовать одинаковые внутренние и внешние ключи при использовании длинного ключа.	
		0	Используются разные ключи
		1	Используются одинаковые ключи
LKEY	2	Выбор длинного ключа В режиме HMAC бит определяет выбор длины ключа.	
		0	Короткий ключ (менее 64 байт)
		1	Длинный ключ (более 64 байт)
MODE	1	Выбор режима Бит выбирает HASH или HMAC режим алгоритма:	
		0	Режим HASH
		1	Режим HMAC. Бит LKEY должен быть установлен если длина используемого ключа больше 64 байт. Бит SAMK должен быть установлен если для длинного ключа используется одинаковый внутренний и внешний ключи.
INIT	0	Инициализация хеш-процессора. Запись этого бита в «1» производит сброс хеш-процессора и производит захват основных параметров (ALGO, MODE, LKEY и т. д.), таким образом хеш-процессор становится готов для вычисления новой хеш-суммы сообщения.	
–	31-16, 13	Зарезервировано	

DIN – регистр ввода данных

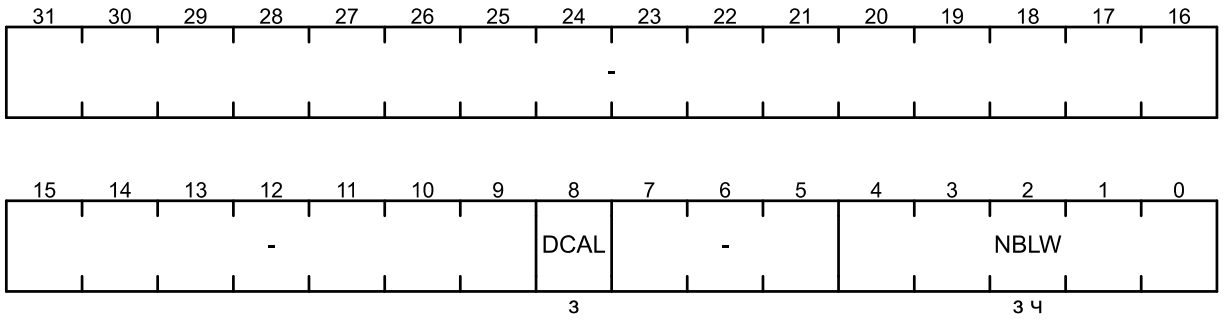
Смещение: + 04h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Значение данных для ввода сообщения блоками по 512 бит. Биты DATATYPE должны быть предварительно настроены в регистре CR, чтобы получить правильное представление сообщения. Когда в регистр DIN записан блок из 16 слов, запускается вычисление промежуточной хеш-суммы: - путем записи новых данных в регистр DIN (первое слово следующего блока), если DMA не используется; - автоматически, если используется DMA. Когда последний блок был записан в регистр DIN, запускается окончательный расчет хеш-суммы (включая заполнение): - путем записи бита DCAL в 1 в регистре STR; - автоматически, если используется DMA и бит MDMAT установлен в «0». Когда выполняется расчет хеш-суммы сообщения (промежуточный или окончательный), любой новый доступ для записи в регистр DIN задерживается до тех пор, пока не завершится вычисление. При чтении регистра DIN осуществляется доступ к последнему слову, записанному в этом месте (ноль после сброса).

STR – регистр запуска

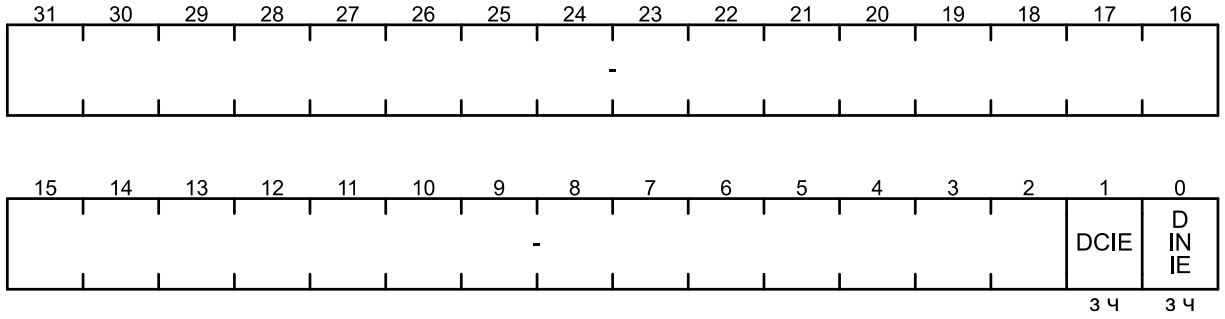
Смещение: + 08h
Сброс: 0000_0000h



Поле	Биты	Описание	
DCAL	8	Вычисление хеш-суммы Запись 1 в этот бит запускает заполнение сообщения с использованием ранее записанного значения NBLW и запускает вычисление окончательной хеш-суммы сообщения со всеми словами данных, записанными во входное FIFO с момента последней записи бита INIT в 1. Чтение этого бита всегда возвращает 0.	
NBLW	4-0	количество допустимых битов в последнем слове сообщения в организации битовой строки хеш-процессора. При записи в эти биты с DCAL равным «0», они принимают значение с шины данных AHB: Когда эти биты пишутся, с DCAL равным «1», битовое поле не изменяется. Их чтение возвращает последнее значение, записанное в NBLW. Примечание. Эти биты необходимо настроить до установки бита DCAL, иначе они не будут учитываться. В частности, невозможно одновременно настроить NBLW и установить DCAL.	
		0	Все 32 бита последних данных, записанных в организацию битовой строки хеш-процессора (после замены данных), действительны.
		1	Действителен только бит [31] последних данных, записанных в организации битовой строки хеш-процессора (после замены данных).
		2	Действительны только биты [31:30] последних данных, записанных в организации битовой строки хеш-процессора (после замены данных).
		3	Действительны только биты [31:29] последних данных, записанных в организации битовой строки хеш-процессора (после замены данных).
		…	
		31	Действительны только бит [0] последних данных, записанных в организации битовой строки хеш-процессора (после замены данных).
-	31-9, 7-5	Зарезервировано	

IMR – регистр разрешения прерывания

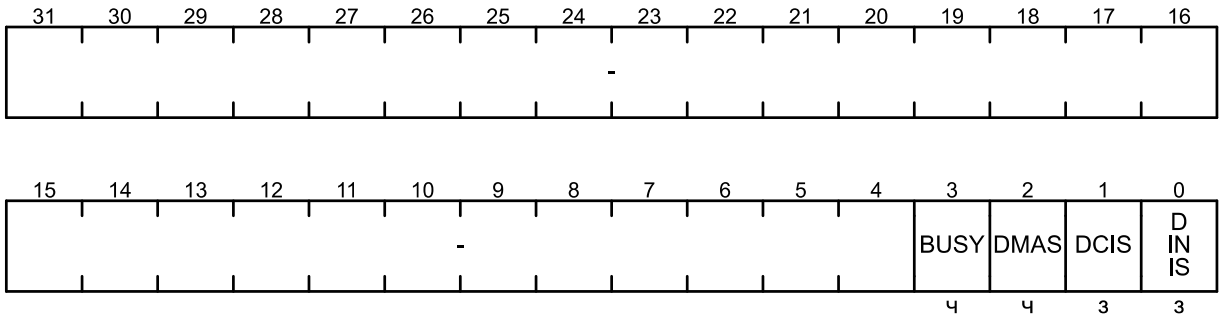
Смещение: + 0Ch
Сброс: 0000_0000h



Поле	Биты	Описание
DCIE	1	Разрешение прерывания по окончанию вычисления хеш-суммы сообщения
DINIE	0	Разрешение прерывания по готовности принятия нового блока данных
–	31-2	Зарезервировано

SR – регистр статуса

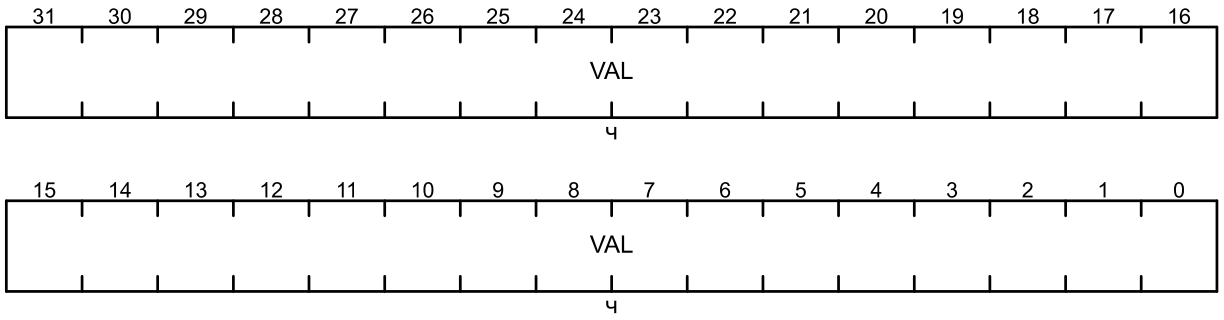
Смещение: + 10h
Сброс: 0000_0000h



Поле	Биты	Описание	
BUSY	3	Бит занятости	
		0	блок данных в данный момент не обрабатывается
		1	хеш-процессор обрабатывает блок данных
DMAS	2	Статус ПДП Этот бит предоставляет информацию об активности интерфейса ПДП. Он устанавливается при записи в CR.DMAE и сбрасывается, когда DMAE=0 и передача ПДП не выполняется. С этим битом не связано прерывание.	
		0	интерфейс ПДП отключен (DMAE=0) и передача не ведется
		1	интерфейс ПДП включен (DMAE=1) или идет передача
DCIE	1	Статус прерывания завершения вычисления сводки сообщения Этот бит устанавливается аппаратно при вычислении хеш-суммы (обработано все сообщение). Он очищается записью его в «0» или записью бита INIT в «1» в регистре CR.	
		0	Нет доступных результатов в регистрах HRx
		1	Расчет завершен, результат доступен в регистрах HRx. Прерывание генерируется, если бит DCIE установлен в регистре IMR.
DINIE	0	Состояние прерывания ввода данных. Этот бит устанавливается аппаратно, когда входной буфер готов принять новый блок (16 ячеек свободны). Он очищается записью в «0» или записью в регистр DIN. 0: Во входном буфере свободно менее 16 ячеек. 1: Во входной буфер можно ввести новый блок. Прерывание генерируется, если бит DINIE установлен в регистре IMR.	
–	31-4	Зарезервировано	

HR – массив регистров хеш-суммы

Смещение: HR + (4*n)h
Сброс: 0000_0000h



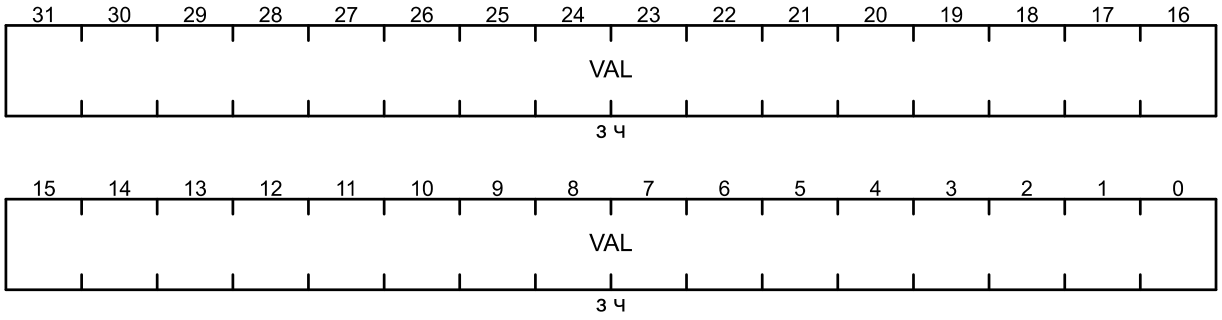
Поле	Биты	Описание
VAL	31-0	Регистры содержат результат вычисления хеш-суммы: 1. В регистры HR0, HR1, HR2, HR3 и HR4 помещается результат вычисления хеш-суммы с использованием алгоритма SHA1. Обратите внимание, что в этом случае регистры с HR5 по HR7 не используются и читаются как ноль. 2. В регистры HR0, HR1, HR2 и HR3 помещается результат вычисления хеш-суммы с использованием алгоритма MD5. Обратите внимание, что в этом случае регистры с HR4 по HR7 не используются и читаются как ноль. 3. В регистры от HR0 до HR6 помещается результат вычисления хеш-суммы с использованием алгоритма SHA224. Обратите внимание, что в этом случае регистр HR7 не используется и читается как ноль. 4. В регистры от HR0 до HR7 помещается результат вычисления хеш-суммы с использованием алгоритма SHA256. Если происходит чтение к одному из этих регистров, когда ядро хеш-процессора вычисляет промежуточную или окончательную хеш-сумму сообщения (то есть, когда бит DCAL был записан в «1»), то чтение приостанавливается (возникает состояние ожидания на шине AHB) до завершения расчётов.

A.18 Регистры блока подсчета циклического избыточного кода CRC

Базовый адрес: 4105_0000h Регистры CRC

DR - регистр данных

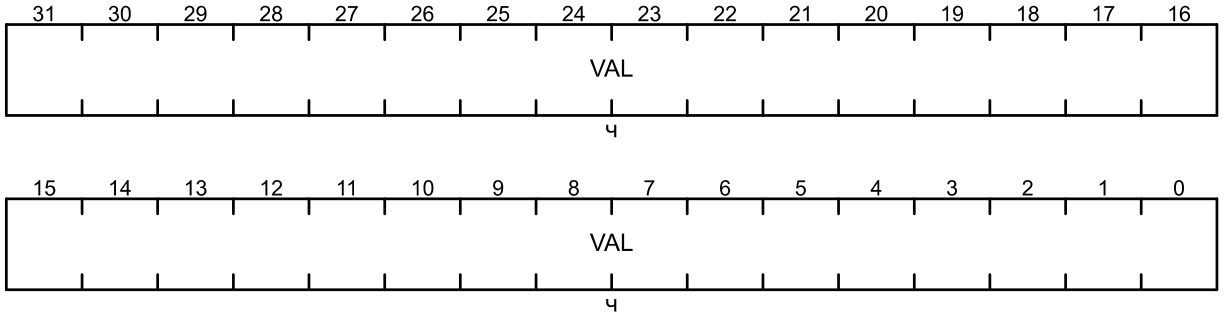
Смещение: 0h
Сброс: FFFF_FFFFh



Поле	Биты	Описание
VAL	31-0	Данные для записи в вычислительный блок CRC записываются в регистр. Вычисленное значение CRC возвращаются при чтении регистра. Если размер данных меньше 32-бит, младшие значащие биты используются для чтения/записи корректного значения.

POST – регистр пост-обработки

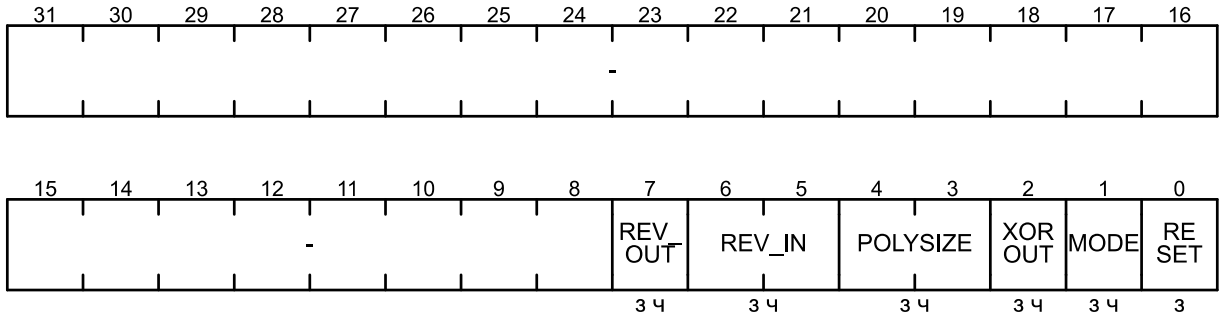
Смещение: 04h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Значение пост-обработки посчитанного значения CRC результата согласно установленным битам в регистре управления CR.

CR – регистр управления

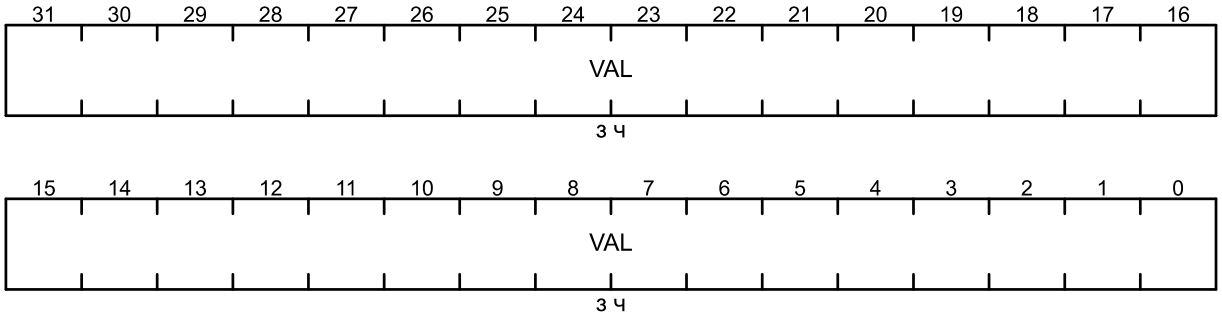
Смещение: 08h
Сброс: 0h



Поле	Биты	Описание	
REV_OUT	7	Обратить выходные данные. Этот бит управляет порядком следования бит выходных данных.	
		0	Порядок битов не изменяется
		1	Бит-реверсный формат выходных данных
REV_IN	6-5	Обращение входных данных Эти биты управляют порядком следования бит входных данных	
		0	Порядок битов не изменяется
		1	Бит-реверсный формат, внутри каждого байта входных данных
		2	Бит-реверсный формат, внутри каждого полуслова входных данных
		3	Бит-реверсный формат, внутри слова входных данных
POLYSIZE	4-3	Размер полинома. Эти биты управляют размером порождающего полинома	
		0	32 – бит
		1	16 – бит
		2	8 – бит
		3	7 – бит
XOROUT	2	Операция «Исключающее ИЛИ» (XOR) над выходными данными «Исключающее ИЛИ» производится над выходными данными после обращения, если оно имело место быть. Пост-обработка.	
		0	Не выполняется
		1	Выходные данные подвергаются операции XOR со значением 0xFFFFFFFF
MODE	1	Режим работы блока подсчёта CRC. Бит определяет возможный режим работы блока подсчёта CRC:	
		0	Входные данные для расчёта поступают напрямую
		1	Каждая операция записи в регистр данных создаёт комбинацию (операция «Исключающее ИЛИ» (XOR)) с предыдущим содержимым регистра данных CRC
RESET	0	Бит сброса блока подсчёта CRC и установки значения регистра данных DR значением сохранённым в INIT.	
-	31-8	Зарезервировано	

INIT – регистр начального значения

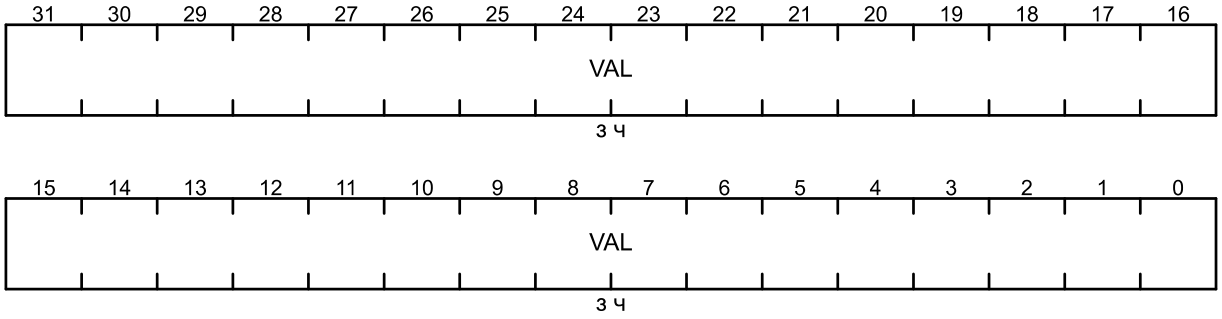
Смещение: 0Ch
Сброс: FFFF_FFFFh



Поле	Биты	Описание
VAL	31-0	Значение для установки начального значения CRC.

POL – регистр полинома

Смещение: 10h
Сброс: 04C1_1DB7h



Поле	Биты	Описание
VAL	31-0	Значение коэффициентов порождающего полинома, который используется при подсчёте CRC.

A.19 Регистры модуля CRYPTO

Описание модуля CRYPTO приведено в разделе 26.

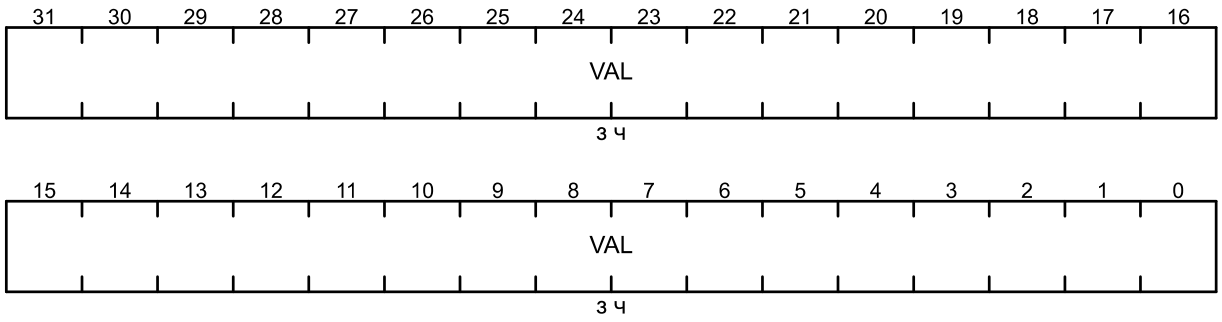
Базовый адрес: 4107_0000h

Смещение:		
	+ 00h (IV)	Массив регистров вектора инициализации
	+ 10h (TEXT_IN)	Массив регистров входного текста
	+ 20h (KEY)	Массив регистров раундовых ключей
	+ 70h (TEXT_OUT)	Массив регистров выходного текста
	+ 90h (GCM_HASH)	Массив регистров хеш-ключа
	+ A0h (GCM_TAG)	Массив регистров аутентификационного тега

Примечание – n – номер регистра от 0 до 3;
k – номер регистра ключа от 0 до 7.

IV – массив регистров вектора инициализации

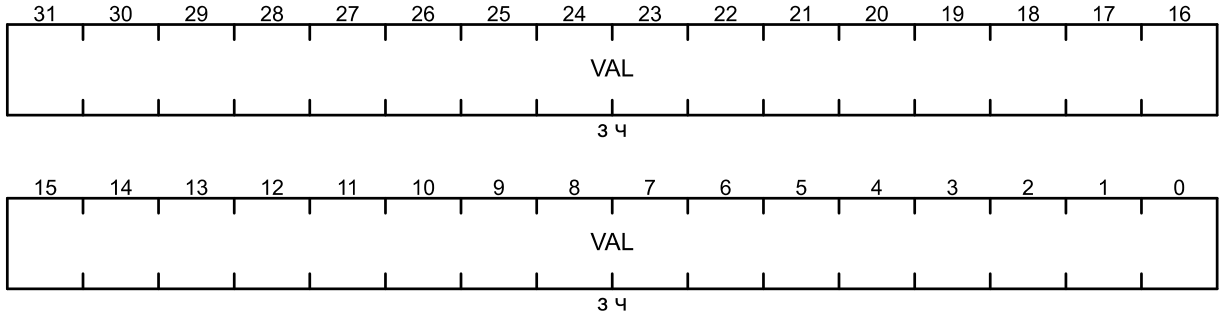
Смещение: IV +(4*n)h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Для алгоритмов AES-128, AES-256, «Кузнечик» биты с 127 – (32*n) по 96 – (32*n) инициализационного вектора. Для алгоритма «Магма» при n от 0 до 1 биты с 63 – (32*n) по 32 – (32*n) инициализационного вектора. Значения регистров IV_2, IV_3 будут игнорироваться.

TEXT_IN – массив регистров входного текста

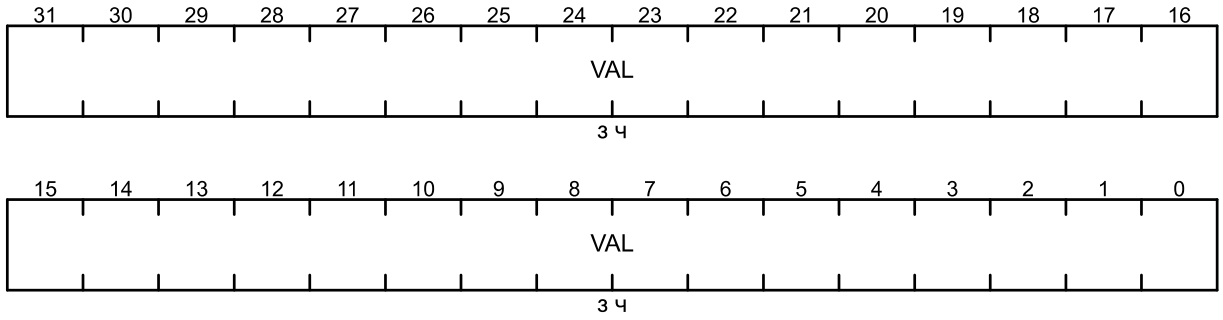
Смещение: TEXT_IN +(4*n)h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Для алгоритмов AES-128, AES-256, «Кузнечик» биты с 127 – (32*n) по 96 – (32*n) входного текста. Для алгоритма «Магма» при n от 0 до 1 биты с 63 – (32*n) по 32 – (32*n) входного текста. Значения регистров TEXT_IN_2, TEXT_IN_3 будут игнорироваться.

KEY – массив регистров криптографического ключа

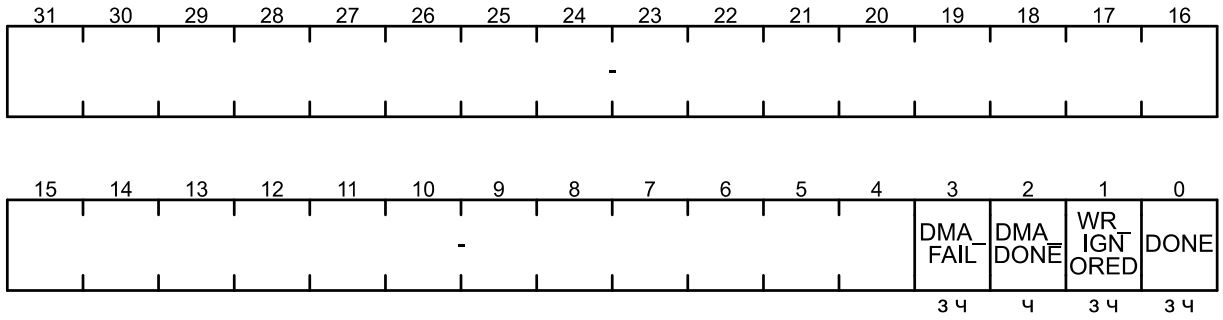
Смещение: KEY + (4*k)h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Для алгоритма AES-128 при k от 0 до 3 биты с 127 – (32*k) по 96 – (32*k) криптографического ключа. Значения регистров KEY_4, …, KEY_7 не используются для формирования раундовых ключей данного алгоритма. Для алгоритмов AES-256, «Кузнечик», «Магма» биты с 255 – (32*k) по 224 – (32*n) криптографического ключа.

IRQ_ENABLE – регистр разрешения прерываний

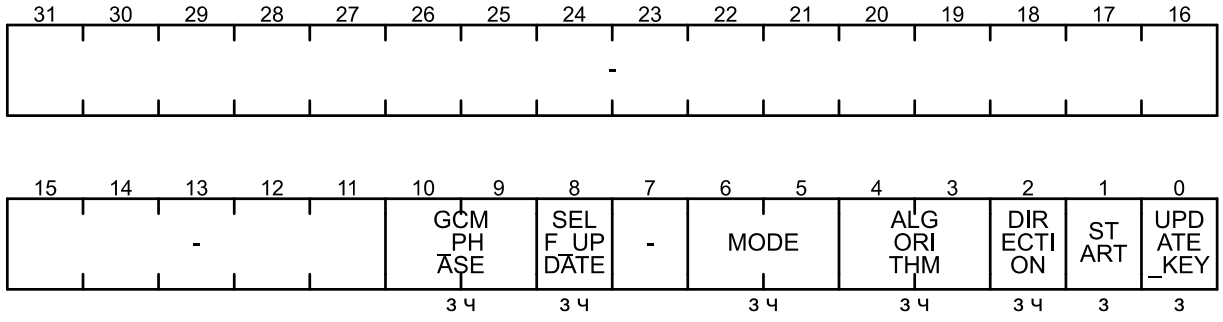
Смещение: + 40h
Сброс: 0000_0000h



Поле	Биты	Описание
DMA_FAIL	3	Если бит установлен, то функционал бита IRQ.DMA_FAIL разрешен.
DMA_DONE	2	Если бит установлен, то функционал бита IRQ.DMA_DONE разрешен. Бит доступен только для чтения и отображает значение бита INTERRUPT управляющего слова дескриптора DMA.
WR_ IGNORED	1	Если бит установлен, то функционал бита IRQ.WR_IGNORED разрешен.
DONE	0	Если бит установлен, то функционал бита IRQ.DONE разрешен.
-	31-4	Зарезервировано

CONTROL – регистр управления функционированием

Смещение: + 44h
Сброс: 0000_0000h

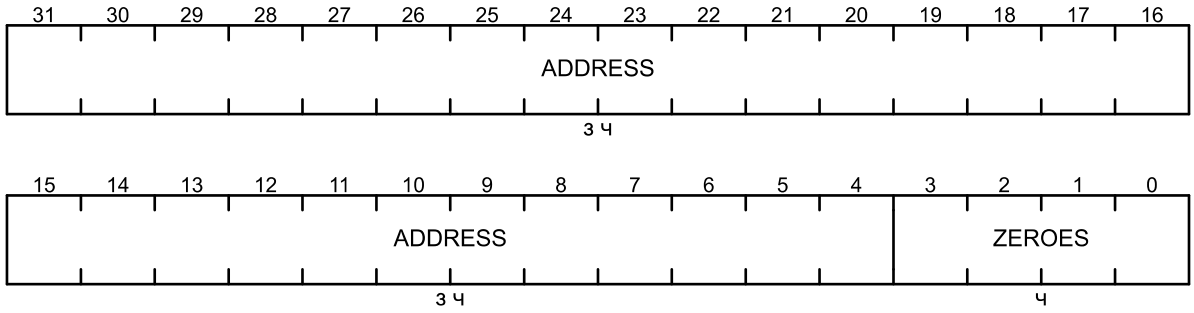


Поле	Биты	Описание	
GCM_PHASE	10-9	Фаза обработки пакета в режиме GALOIS_COUNTER_MODE. Если режим выполнения криптографической операции не GALOIS_COUNTER_MODE, то значение поля игнорируется.	
		0	GCM_INIT;
		1	GCM_HEADER;
		2	GCM_PAYLOAD;
		3	GCM_LAST_BLOCK.

Поле	Биты	Описание	
SELF_UPDATE	8	Разрешение автоматического обновления содержимого регистров IV_* при завершении выполнения криптографической операции.	
GCM_PHASE	10-9	Фаза обработки пакета в режиме GALOIS_COUNTER_MODE. Если режим выполнения криптографической операции не GALOIS_COUNTER_MODE, то значение поля игнорируется.	
MODE	6-5	Выбор режима выполнения криптографической операции:	
		0	ELECTRONIC_CODEBOOK (ECB);
		1	CIPHER_BLOCK_CHAINING (CBC);
		2	COUNTER_MODE (CTR);
		3	GALOIS_COUNTER_MODE (GCM).
ALGORITHM	4-3	Выбор алгоритма криптографической операции	
		0	AES_128
		1	AES_256
		2	MAGMA
		3	KUZNECHIK
DIRECTION	2	Направление криптографической операции (0 – Encryption, 1 - Decryption)	
START	1	Запрос на выполнение криптографической операции. Непосредственный старт может быть отложен до завершения обновления раундовых ключей (спровоцированное установленным битом UPDATE_KEY или выполненное из-за отсутствия раундовых ключей для выбранного криптографического алгоритма). При чтении всегда возвращает 0.	
UPDATE_KEY	0	Принудительно выполнить обновление раундовых ключей шифрования на основе текущего значения регистров {KEY_0, ..., KEY_7} или {KEY_0, ..., KEY_3} для алгоритма AES-128. При чтении всегда возвращает 0.	
-	31-11, 7	Зарезервировано	

BASE_DESCRIPTOR – регистр адреса базового дескриптора

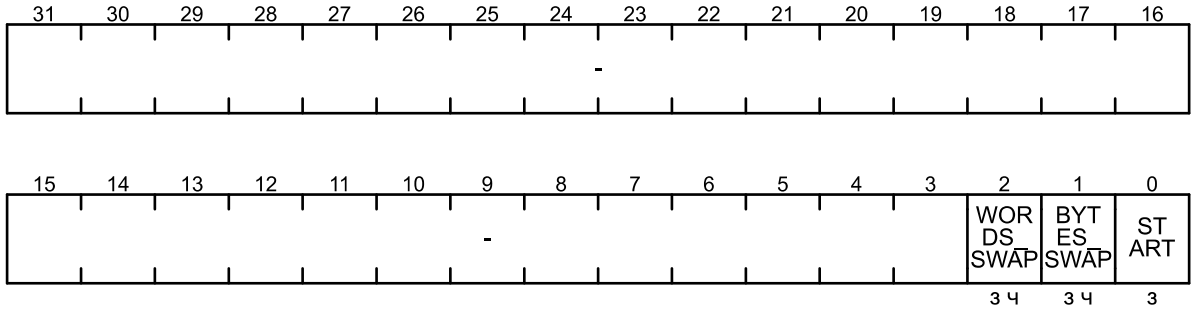
Смещение: + 48h
Сброс: 0000_0000h



Поле	Биты	Описание
ADDRESS	31-4	Адрес, который указывает на первый дескриптор DMA в цепочке, который будет прочитан из памяти после записи 1 в бит START регистра DMA_CONTROL
ZEROES	4-0	Младшие биты адреса, должны быть равны 0. Для обеспечения корректности записываемого адреса дескриптора доступны только для чтения.

DMA_CONTROL – регистр управления функционированием DMA

Смещение: + 4Ch
Сброс: 0000_0000h

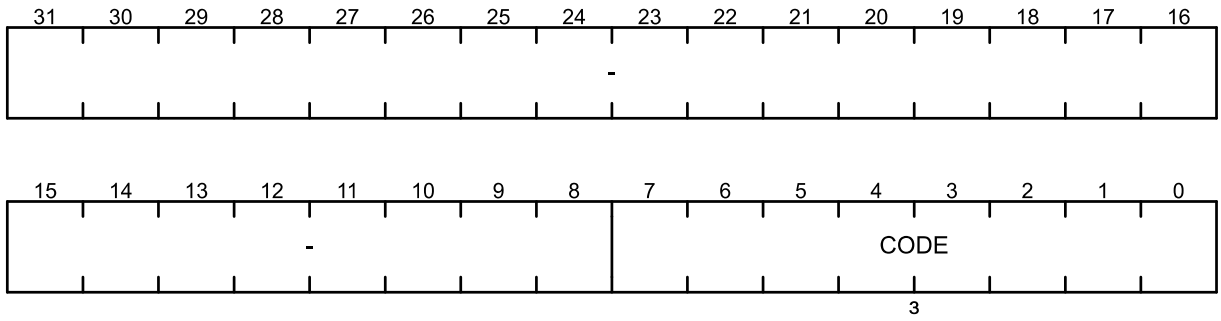


Поле	Биты	Описание
WORDS_SWAP	2	Если бит сброшен, то заполнение регистров TEXT_IN_* считанными из памяти словами входного текста идет в порядке увеличения индекса регистра. Если бит установлен, то последовательность заполнения регистров TEXT_IN_* меняется на обратную. Аналогичное влияние оказывается на пересылку выходных слов текста из регистров TEXT_OUT_* в память. Данный бит не меняет порядок считывания слов дескриптора.

Поле	Биты	Описание
BYTES_SWAP	1	Если бит установлен, то перед записью в регистры TEXT_IN_* в прочитанном слове будет изменен порядок следования байт. Пересылаемые из TEXT_OUT_* слова будут выставлены на шину AHB с инвертированным порядком байт. Данный бит не оказывает влияние на считываемые слова дескриптора
START	0	Установленный бит запускает последовательность операций в режиме прямого доступа к памяти. При чтении всегда возвращает 0.
-	31-3	Зарезервировано

TERMINATE – регистр экстренного завершения операции

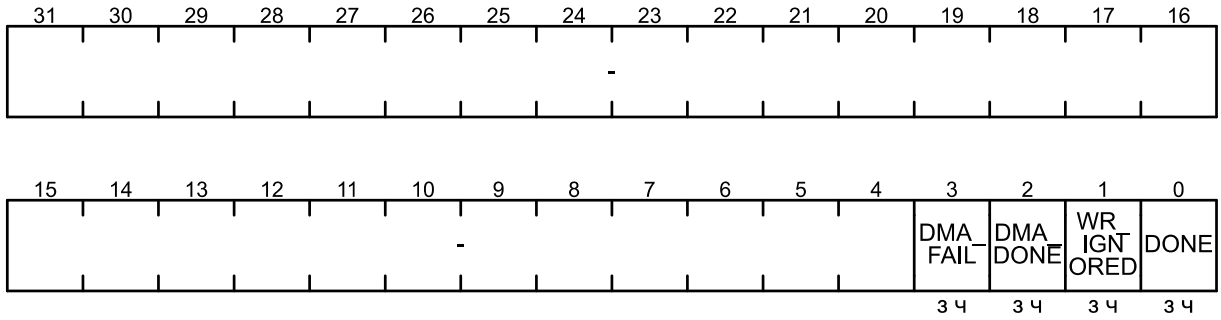
Смещение: + 50h
Сброс: 0000_0000h



Поле	Биты	Описание
CODE	7-0	Запись в данное поле значения 0xD0 приведет к сбросу внутреннего состояния модулей криптографических операций и хранилища криптографического ключа. Запись других значений будет проигнорирована. При чтении всегда возвращает 0.
-	31-8	Зарезервировано

IRQ – регистр прерываний, ожидающих обработки

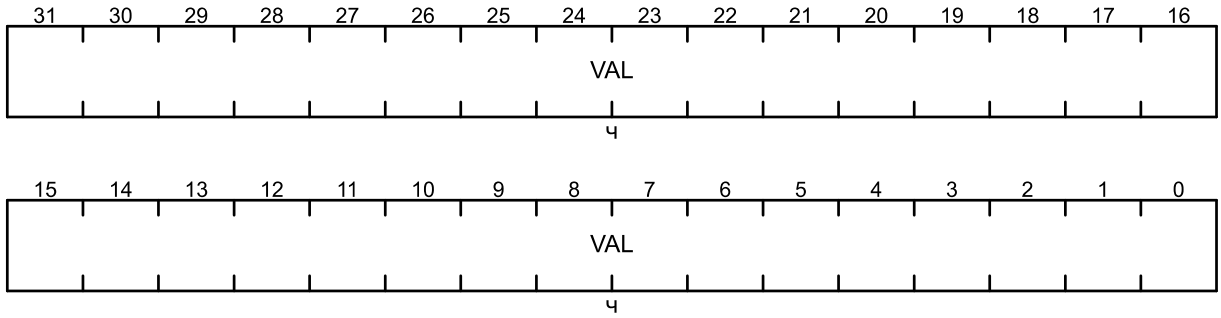
Смещение: + 60h
Сброс: 0000_0000h



Поле	Биты	Описание
DMA_FAIL	3	Бит устанавливается, если установлен соответствующий бит регистра IRQ_ENABLE, при преждевременном завершении работы DMA из-за обнаружения ошибки. Бит может быть сброшен записью 1.
DMA_DONE	2	Бит устанавливается, если бит INTERRUPT управляющего слова текущего дескриптора DMA был установлен (его состояние отображается в соответствующий бит регистра IRQ_ENABLE), при успешном завершении выполнения дескриптора DMA (поля STATUS.BAD_DESCR и STATUS.AHB_ERROR содержат нулевое значение). Бит может быть сброшен записью 1.
WR_ IGNORED	1	Бит устанавливается, если установлен соответствующий бит регистра IRQ_ENABLE, при выполнении записи в регистр CONTROL, TEXT_IN_* или IV_* пока STATUS.READY сброшен. Бит может быть сброшен записью 1.
DONE	0	Бит устанавливается, если установлен соответствующий бит регистра IRQ_ENABLE, при переходе в режим готовности (переключение STATUS.READY в 1). Бит может быть сброшен записью 1.
-	31-4	Зарезервировано

TEXT_OUT – массив регистров выходного текста

Смещение: TEXT_OUT +(4*n)h
Сброс: 0000_0000h

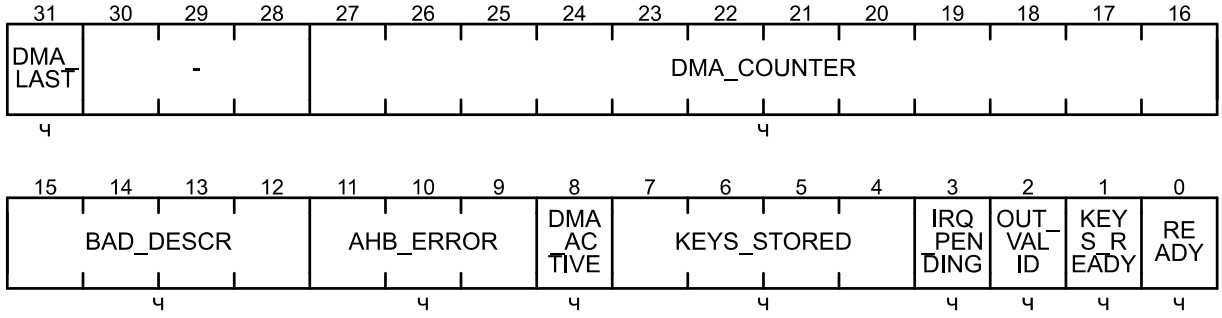


Поле	Биты	Описание
VAL	31-0	Для алгоритмов AES-128, AES-256, «Кузнечик» биты с 127 – (32*n) по 96 – (32*n) выходного текста. Для алгоритма «Магма» при n от 0 до 1 биты с 63 – (32*n) по 32 – (32*n) выходного текста. Значения регистров TEXT_OUT_2, TEXT_OUT_3 следует игнорировать.

Примечание – В TEXT_OUT_* не хранится результат криптографической операции, отображаемое значение зависит от состояния полей MODE, DIRECTION и ALGORITHM регистра CONTROL, см. пункт «Общие положения» подраздела «26.4 Режимы работы».

STATUS – регистр статуса операций

Смещение: + 80h
Сброс: 0000_0001h



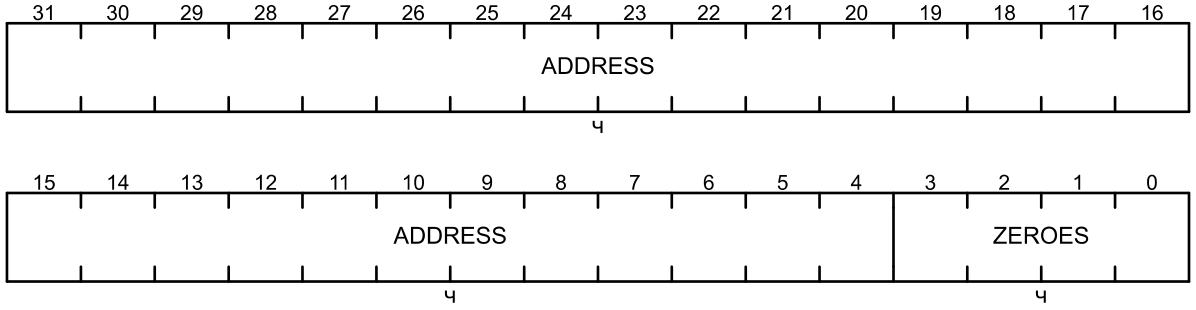
Поле	Биты	Описание
DMA_LAST	31	Поле отображает значение бита LAST слова управления DMA последнего прочитанного дескриптора. Бит установлен, если прочитанный дескриптор является последним в цепочке дескрипторов. Значение обновляется при считывании нового дескриптора. Бит сбрасывается при экстренном прекращении выполнения операции.

<table>
<tr><th>Поле</th><th>Биты</th><th colspan="2">Описание</th></tr>
<tr><td>DMA_
COUNTER</td><td>27-16</td><td colspan="2">Поле отображает количество обработанных DMA блоков текст. Значение обнуляется при считывании нового дескриптора. Значение сбрасывается при экстренном прекращении выполнения операции.</td></tr>
<tr><td>BAD_
DESCR</td><td>15-12</td><td colspan="2">Поле хранит сведения об обнаруженных некорректных словах в прочитанном дескрипторе DMA. После записи в регистр DMA_CONTROL любого значения поле будет переведено в состояние NO_ERROR.
Список возможных значений поля:</td></tr>
<tr><td></td><td></td><td>0000b</td><td>NO_ERROR</td></tr>
<tr><td></td><td></td><td>0001b</td><td>CONTROL</td></tr>
<tr><td></td><td></td><td>0010b</td><td>TEXT_IN_ADDRESS</td></tr>
<tr><td></td><td></td><td>0100b</td><td>TEXT_OUT_ADDRESS</td></tr>
<tr><td></td><td></td><td>1000b</td><td>NEXT_DESCRIPTOR</td></tr>
<tr><td>AHB_ERROR</td><td>11-9</td><td colspan="2">Поле хранит сведения об области памяти, на которую пришелся ответ ERROR на транзакцию по шине AHB. После записи в регистр DMA_CONTROL любого значения поле будет переведено в состояние NO_ERROR.
Список возможных значений:</td></tr>
<tr><td></td><td></td><td>000b</td><td>NO_ERROR</td></tr>
<tr><td></td><td></td><td>001b</td><td>READ_DESCRIPTOR</td></tr>
<tr><td></td><td></td><td>010b</td><td>READ_TEXT_IN</td></tr>
<tr><td></td><td></td><td>100b</td><td>WRITE_TEXT_OUT</td></tr>
<tr><td>DMA_
ACTIVE</td><td>8</td><td colspan="2">Если бит установлен, то конечный автомат DMA занят выполнением дескриптора. Бит READY будет находиться в сброшенном состоянии, пока DMA_ACTIVE установлен. Бит сбрасывается после выполнения цепочки дескрипторов, обнаружения некорректного дескриптора, получения ответа ERROR по шине AHB или при экстренном прекращении выполнения операции.</td></tr>
<tr><td>KEYS_
STORED</td><td>7-4</td><td colspan="2">Каждый бит данного поля соответствует заданному криптографическому алгоритму. Биты нумеруются справа налево. 0 – AES-128, 1 – AES-256, 2 – MAGMA, 3 – KUZNECHIK.</td></tr>
<tr><td>IRQ_
PENDING</td><td>3</td><td colspan="2">Если бит установлен, то регистр IRQ содержит хотя бы 1 не сброшенный бит.</td></tr>
<tr><td>TEXT_OUT_
VALID</td><td>2</td><td colspan="2">Если бит установлен, то на выходе криптографического подмодуля находится результат шифрования/дешифрования входных данных (TEXT_IN_*, IV_* или TEXT_IN_* ^ IV_* в зависимости от алгоритма и режима выполнения криптографической операции).</td></tr>
<tr><td>KEYS_
READY</td><td>1</td><td colspan="2">Если бит установлен, криптографический ключ текущего активного криптографического алгоритма сгенерированы. Бит сбрасывается при старте обновления криптографического ключа или при экстренном прекращении выполнения операции.</td></tr>
</table>

Поле	Биты	Описание
READY	0	Если бит установлен, то модуль готов к выполнению очередной криптографической операции и/или обновлению раундовых ключей шифрования. Бит сбрасывается при старте обновления раундовых ключей, выполнения криптографической операции и считывания первого дескриптора цепочки. Бит устанавливается после завершения всех запланированных действий или при экстренном прекращении выполнения операции.
-	30-28	Зарезервировано

CURRENT_DESCRIPTOR – регистр адреса активного дескриптора

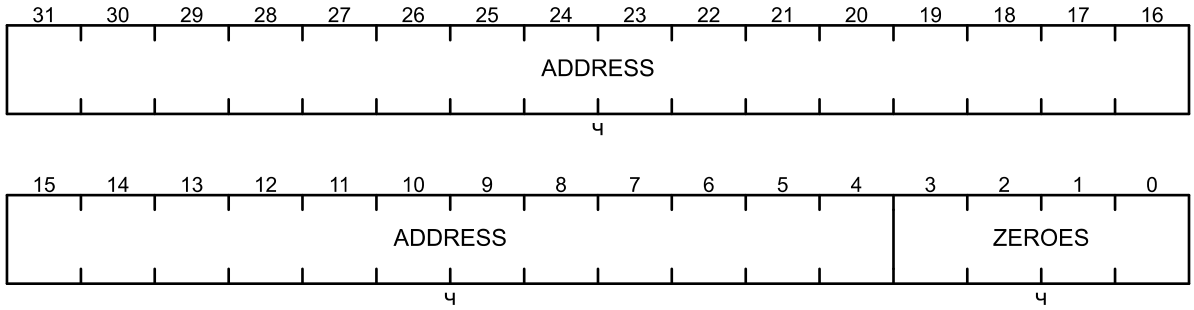
Смещение: + 84h
Сброс: 0000_0000h



Поле	Биты	Описание
ADDRESS	31-4	Адрес, который указывает на текущий выполняемый дескриптор. При записи 1 в бит START регистра DMA_CONTROL в данный регистр попадает значение из регистра BASE_DESCRIPTOR. Если значение бита LAST слова управления DMA текущего дескриптора сброшен, то при успешном завершении текущего дескриптора в CURRENT_DESCRIPTOR заносится значение NEXT_DESCRIPTOR
ZEROES	4-0	Младшие биты адреса, должны быть равны 0.

NEXT_DESCRIPTOR – регистр адреса следующего дескриптора

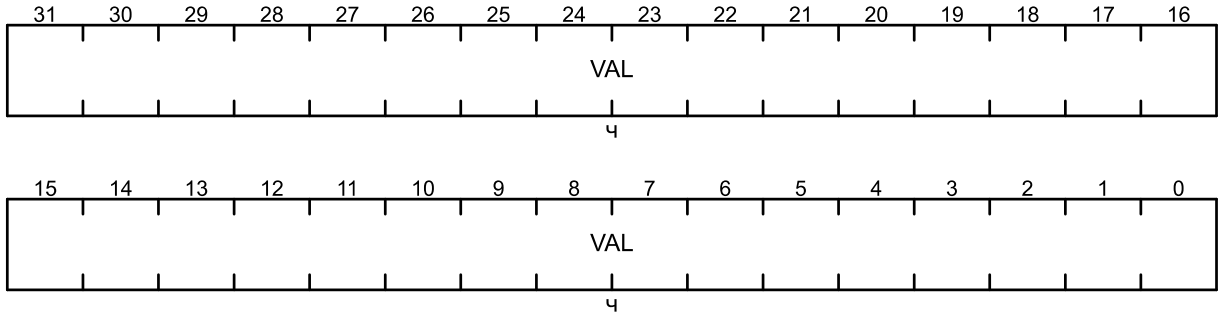
Смещение: + 88h
Сброс: 0000_0000h



Поле	Биты	Описание
ADDRESS	31-4	Адрес, который указывает на следующий дескриптор цепочки. Значение регистра будет сброшено, когда в слове управления текущего дескриптора DMA будет установлен бит LAST
ZEROES	4-0	Младшие биты адреса, должны быть равны 0.

GCM_HASH – массив регистров хеш-ключа

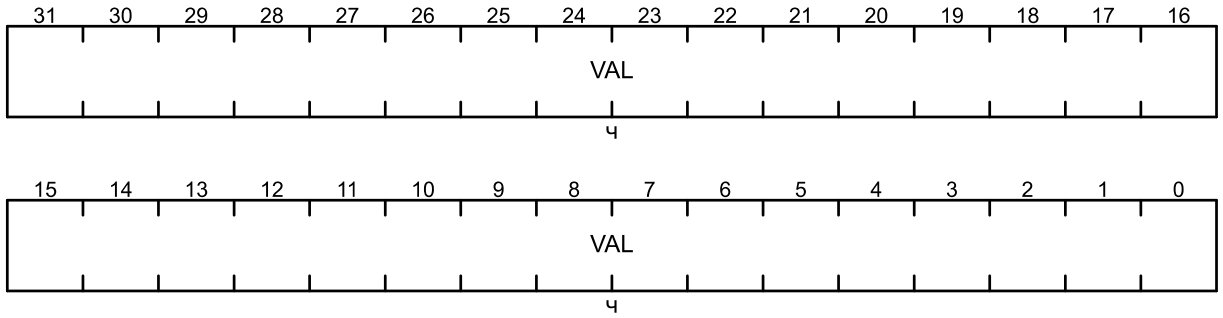
Смещение: GCM_HASH +(4*n)h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Биты с 127 – (32*n) по 96 – (32*n) хеш-ключа, используемого в криптографическом режиме GCM.

GCM_TAG – массив регистров аутентификационного тега

Смещение: GCM_TAG +(4*n)h
Сброс: 0000_0000h



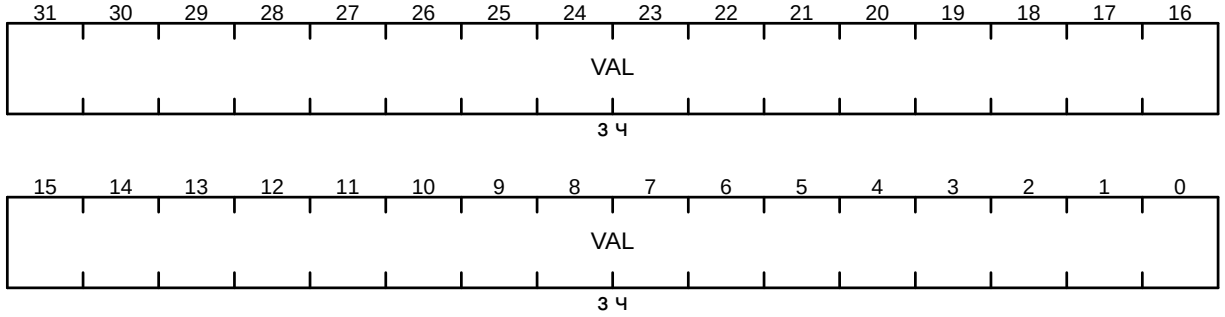
Поле	Биты	Описание
VAL	31-0	Биты с 127 – (32*n) по 96 – (32*n) аутентификационного тега в криптографическом режиме GCM.

A.20 Регистры сторожевого таймера WDT

Базовый адрес: 5000_6000h Регистры WDT

LOAD – регистр загрузки сторожевого таймера

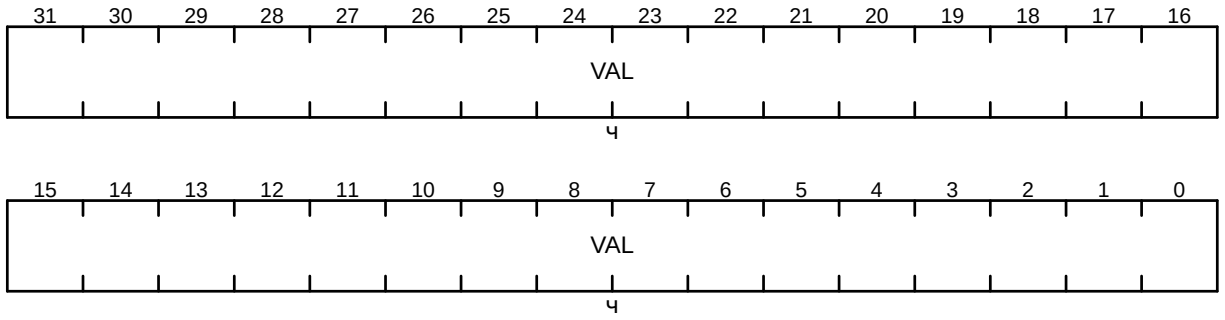
Смещение: + 00h
Сброс: FFFF_FFFFh



Поле	Биты	Описание
VAL	31-0	32-разрядный регистр, хранящий начальное значение счетчика. Когда происходит запись в этот регистр, счетчик сразу инициируется этим новым значением. Минимальное допустимое значение 0000_0001h

VALUE – регистр значения счетчика сторожевого таймера

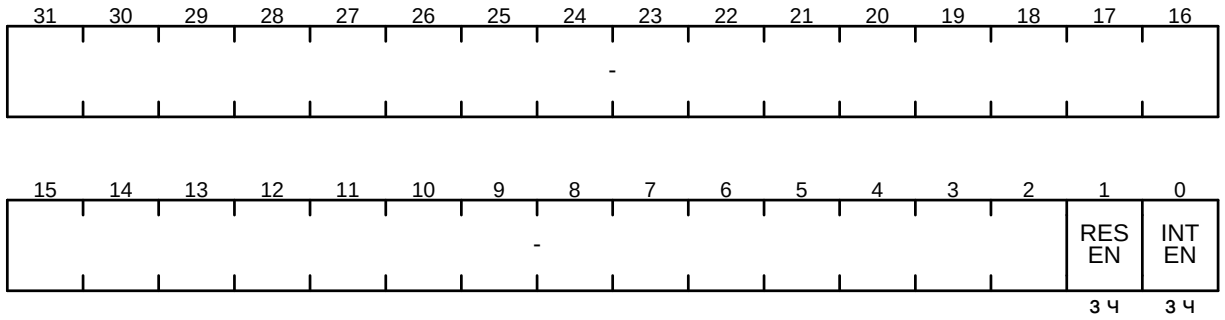
Смещение: + 04h
Сброс: FFFF_FFFFh



Поле	Биты	Описание
VAL	31-0	32-разрядный регистр текущего значения счетчика

CTRL – регистр управления сторожевого таймера

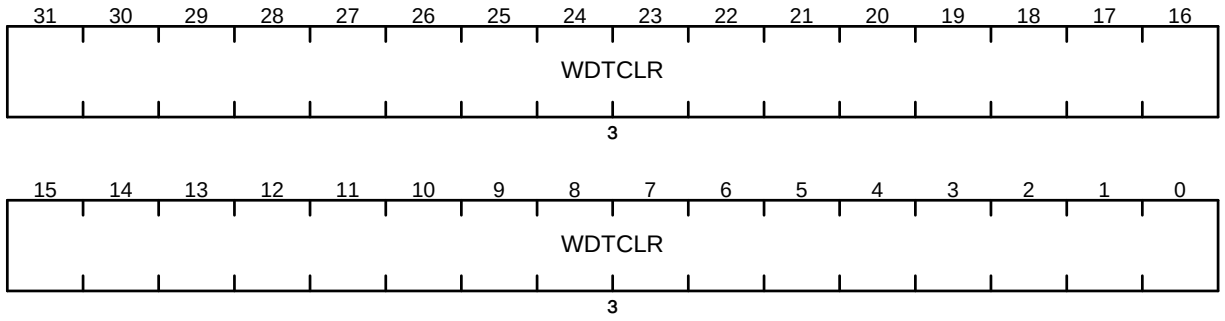
Смещение: + 08h
Сброс: 0h



Поле	Биты	Описание	
RESEN	1	Бит разрешения сброса микроконтроллера по сторожевому таймеру. Работает по функции «Логическое И» с битом INTEN регистра WDTCTRL	
		0	Сброс бита выключает сброс
		1	Установка бита включает сброс
INTEN	0	Бит включения счета и разрешения прерывания сторожевого таймера	
		0	Сброс бита выключает счетчик и снимает прерывание
		1	Установка бита включает счетчик и генерирует прерывание. Если счетчик был включен на момент установки бита, то он инициируется значением из регистра LOAD
–	31-2	Зарезервировано	

INTCLR – регистр сброса сторожевого таймера

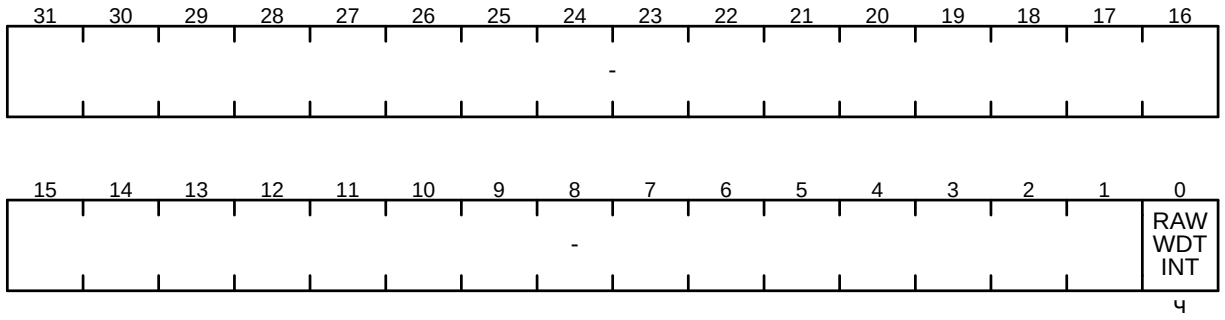
Смещение: + 0Ch
Сброс: 0h



Поле	Биты	Описание
WDTCLR	31-0	32-разрядный регистр сброса сторожевого таймера. Запись любого значения в этот регистр приводит к сбросу прерывания сторожевого таймера и загрузке счетчика значением из регистра LOAD

RIS – регистр прерывания сторожевого таймера

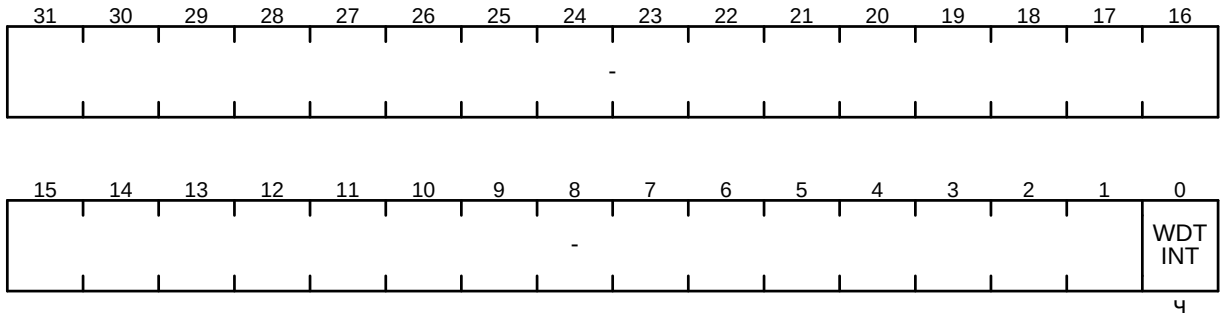
Смещение: + 10h
Сброс: 0h



Поле	Биты	Описание	
RAWWDTINT	0	Индикатор состояния немаскированного бита прерывания	
		0	Сброшено
		1	Установлено
–	31-1	Зарезервировано	

MIS – регистр маскированного прерывания сторожевого таймера

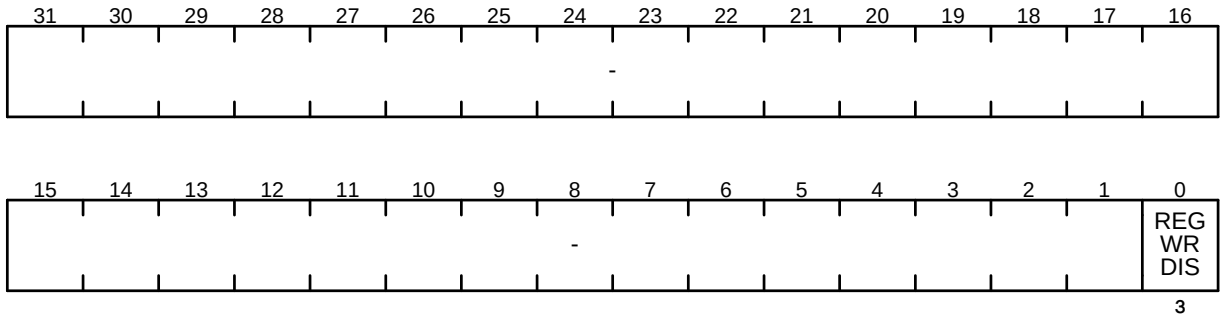
Смещение: + 14h
Сброс: 0h



Поле	Биты	Описание	
WDTINT	0	Индикатор состояния маскированного бита прерывания. Сигнализирует о появлении маскированного прерывания от счетчика. Состояние бита WDTINT – это «Логическое И» битов RAWWDTINT, INTEN	
		0	Сброшено
		1	Установлено
–	31-1	Зарезервировано	

LOCK – регистр блокировки сторожевого таймера

Смещение: + C00h
Сброс: 0h



Поле	Биты	Описание	
REGWRDIS	0	Бит запрета записи во все регистры сторожевого таймера (кроме регистра LOCK). Функция необходима для предотвращения отключения сторожевого таймера сбойными программами	
		0	Разрешено (по умолчанию). Для сброса бита следует записать в регистр LOCK значение 1ACC_E551h
		1	Запрещено. Для установки бита следует записать в регистр LOCK любое значение, кроме 1ACC_E551h
–	31-1	Зарезервировано	

A.21 Регистры блока RTC

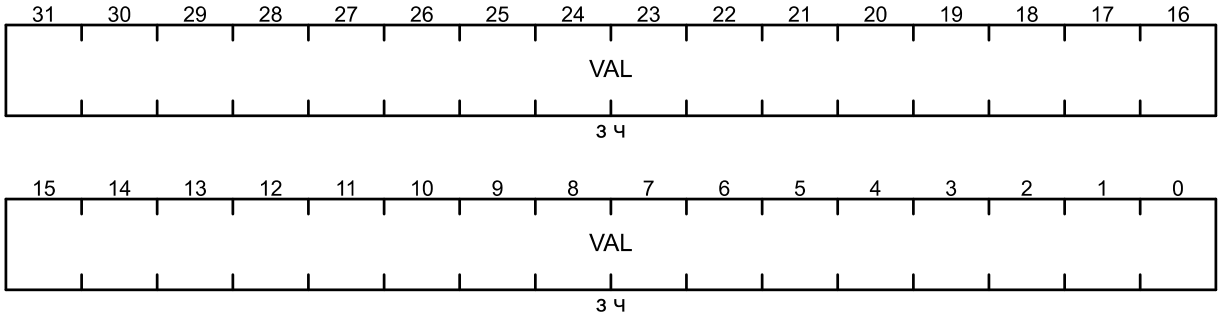
Базовый адрес: 5000_5000h

Смещение: + 020h (REG) Массив регистров общего назначения

Примечание – n – номер регистра 0 – 15.

TIME – регистр счетчика времени RTC

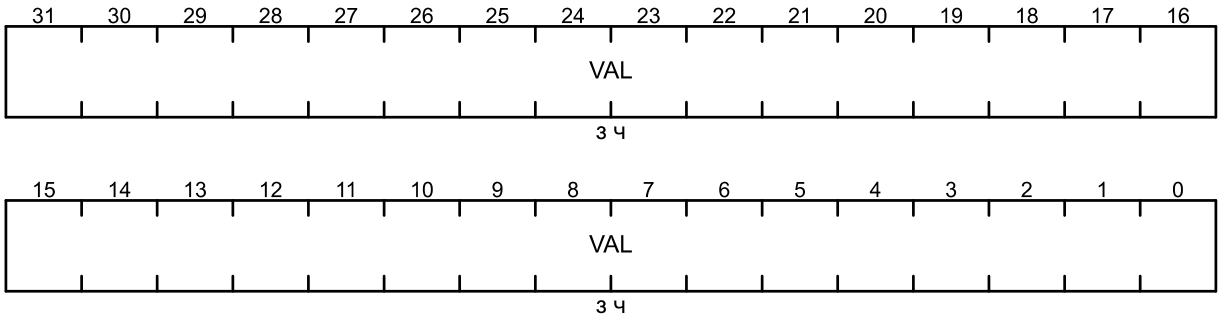
Смещение: + 0h
Сброс: 0h



Поле	Биты	Описание
VAL	31-0	Текущее значение времени в секундах в 32 битном формате

ALARM1 – регистр первого счетчика пробуждения блока RTC

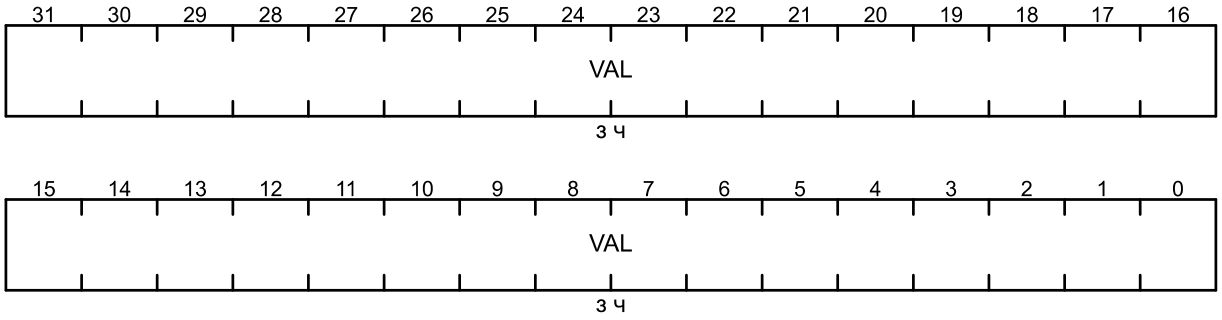
Смещение: + 04h
Сброс: FFFF_FFFFh



Поле	Биты	Описание
VAL	31-0	Пороговое значение срабатывания будильника 1

ALARM2 – регистр второго счетчика пробуждения блока RTC

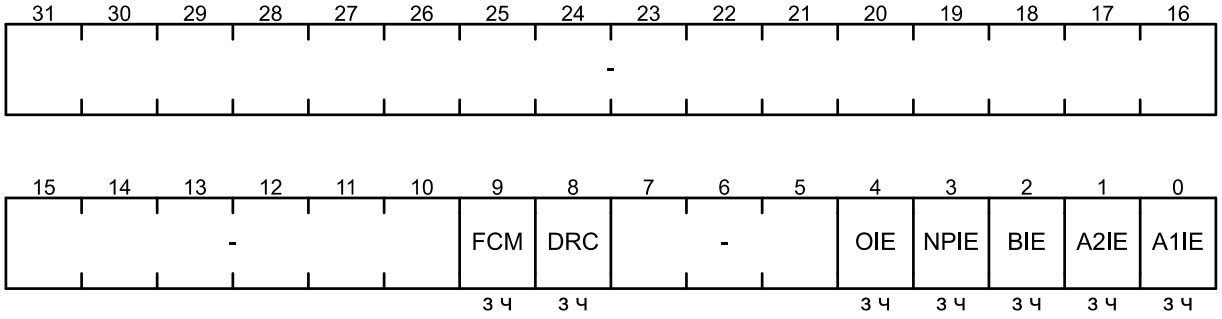
Смещение: + 08h
Сброс: FFFF_FFFFh



Поле	Биты	Описание
VAL	31-0	Пороговое значение срабатывания будильника 2

CONTROL – управляющий регистр блока RTC

Смещение: + 0Ch
Сброс: 0h

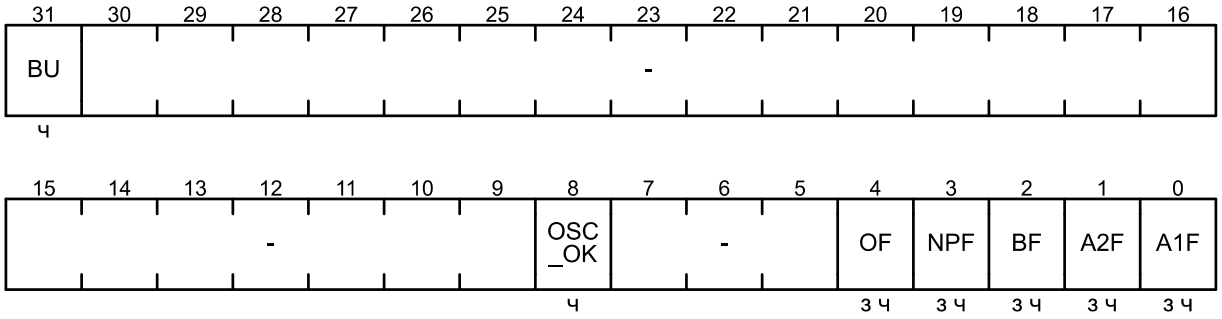


Поле	Биты	Описание
FCM	9	При установке бита для тактирования регистра времени вместо односекундного тактового сигнала (clk_1s) будет использоваться 32 кГц тактовый сигнал (clk_out)
DRC	8	Бит отключения RC-генератора. Когда бит сброшен, работает RC-генератор. При установке бита RC-генератор останавливается, а в качестве опорного тактового сигнала блока RTC используется кварцевый генератор
OIE	3	Разрешение прерывания при переполнении регистра счетчика времени. При установленном бите разрешается генерация прерывания по установке флага OF регистра STATUS
NPIE	3	Разрешение прерывания при переключении на основное питание. При установленном бите разрешается генерация прерывания по установке флага NPF регистра STATUS
BIE	2	Разрешение прерывания при переключении на батарейное питание. При установленном бите разрешается генерация прерывания по установке флага BF регистра STATUS
A2IE	1	Разрешение прерывания при срабатывании будильника 2. При установленном бите разрешается генерация прерывания по установке флага A2F регистра STATUS

Поле	Биты	Описание
A1IE	0	Разрешение прерывания при срабатывании будильника 1. При установленном бите разрешается генерация прерывания по установке флага A1F регистра STATUS
–	31-10, 7-3	Зарезервировано

STATUS – регистр статуса блока RTC

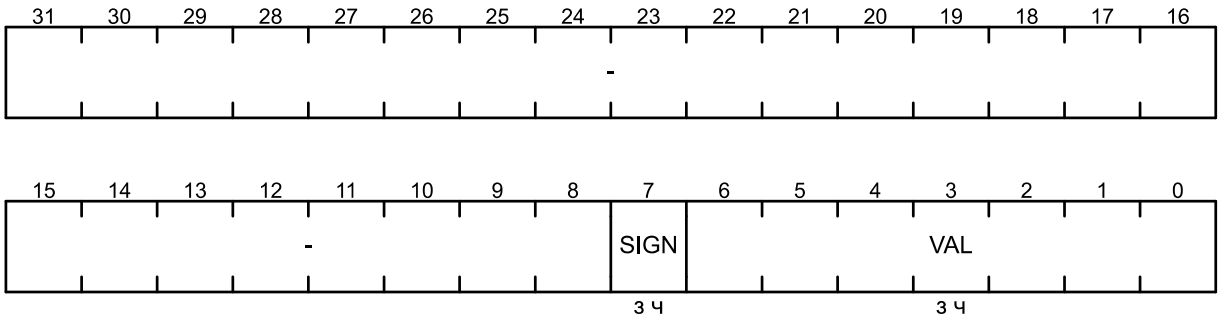
Смещение: + 10h
Сброс: 0h



Поле	Биты	Описание
BU	31	Флаг статуса использования батарейного питания. Установленный бит означает, что в качестве источника питания используется напряжение VBAT
OSCOK	8	Осциллятор работает в штатном режиме. Установленный бит указывает на штатную работу осциллятора XTAL со стабильной генерацией тактового сигнала
OF	4	Флаг переполнения регистра счетчика времени. Установленный бит указывает на то, что произошло переполнение регистра TIME. Сбрасывается записью логической единицы
NPF	3	Флаг статуса переключения на основное питание. Установленный бит указывает на переключение напряжения питания с VBAT на DVCC. Сбрасывается записью логической единицы
BF	2	Флаг статуса переключения на батарейное питание. Установленный бит указывает на переключение напряжения питания с DVCC на VBAT. Сбрасывается записью логической единицы
A2F	1	Флаг статуса срабатывания будильника 2. Устанавливается при равенстве значений регистров ALARM2 и TIME. Сбрасывается записью логической единицы.
A1F	0	Флаг статуса срабатывания будильника 1. Устанавливается при равенстве значений регистров ALARM1 и TIME. Сбрасывается записью логической единицы
–	30-9, 7-4	Зарезервировано

TRIM_RC – регистр подстройки RC генератора

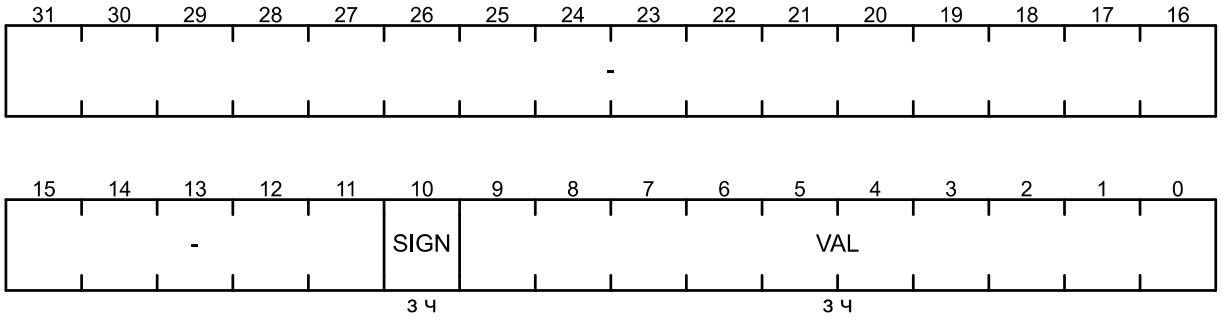
Смещение: + 14h
Сброс: 0h



Поле	Биты	Описание
SIGN	7	Бит знака подстроечного значения. Если бит равен 0 – число положительное, 1 – число отрицательное
VAL	6-0	Значение подстройки RC-генератора в дополнительном коде (диапазон значений от -128 до 127)
–	31-8	Зарезервировано

TRIM_CLK1S – регистр подстройки односекундного периода RTC

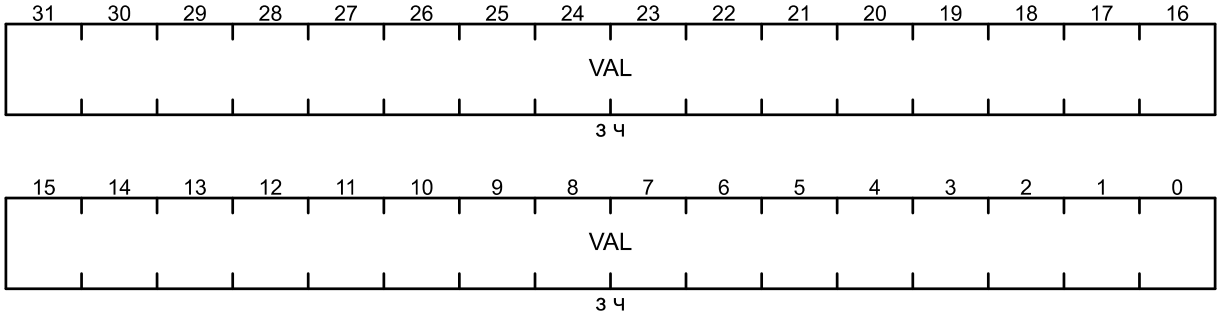
Смещение: + 18h
Сброс: 0h



Поле	Биты	Описание
SIGN	10	Бит знака подстроечного значения. Если бит равен 0 – число положительное, 1 – число отрицательное
VAL	9-0	Значение подстройки тактового сигнала clk_1s в дополнительном коде (диапазон значений от -1024 до 1023). Подстройка производится каждые 64 секунды
–	31-11	Зарезервировано

GPR – массив регистров общего назначения

Смещение: REG + (4*n)h
Сброс: 0h



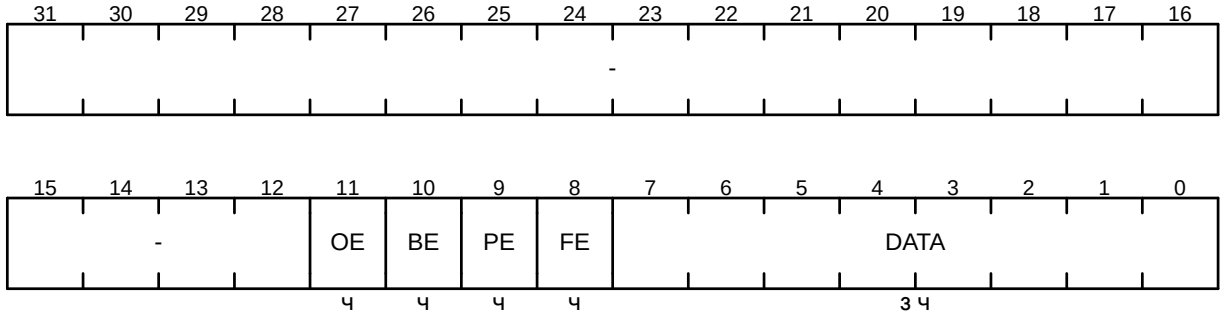
Поле	Биты	Описание
VAL	31-0	Значение регистра n общего назначения содержащее 32-битное слово. Данные будут сохранены при наличии напряжения VCC либо VBAT

A.22 Регистры приемопередатчика UART

Базовый адрес:	5201_0000h	Регистры приемопередатчика UART0
	5201_1000h	Регистры приемопередатчика UART1
	5201_2000h	Регистры приемопередатчика UART2
	5201_3000h	Регистры приемопередатчика UART3
	5201_4000h	Регистры приемопередатчика UART4
	5201_5000h	Регистры приемопередатчика UART5
	5201_6000h	Регистры приемопередатчика UART6
	5201_7000h	Регистры приемопередатчика UART7

DR – регистр данных

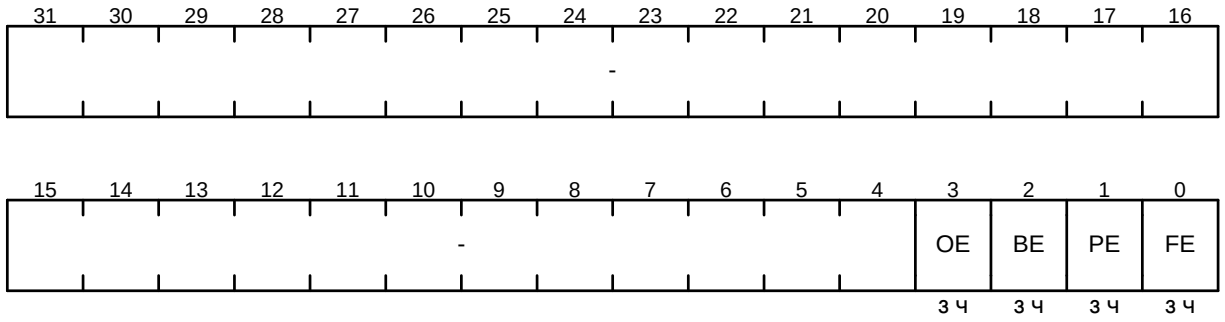
Смещение: + 00h
Сброс: 0h



Поле	Биты	Описание
OE, BE, PE, FE	11, 10, 9, 8	См. далее описание бит в регистре RSR
DATA	7-0	Поле данных. Результатом записи в поле DATA является размещение байта в буфере передатчика, а результатом чтения – считывание байта из буфера приемника
–	31-12	Зарезервировано

RSR – регистр состояния приемника и сброса ошибки приемника

Смещение: + 04h
Сброс: 0h

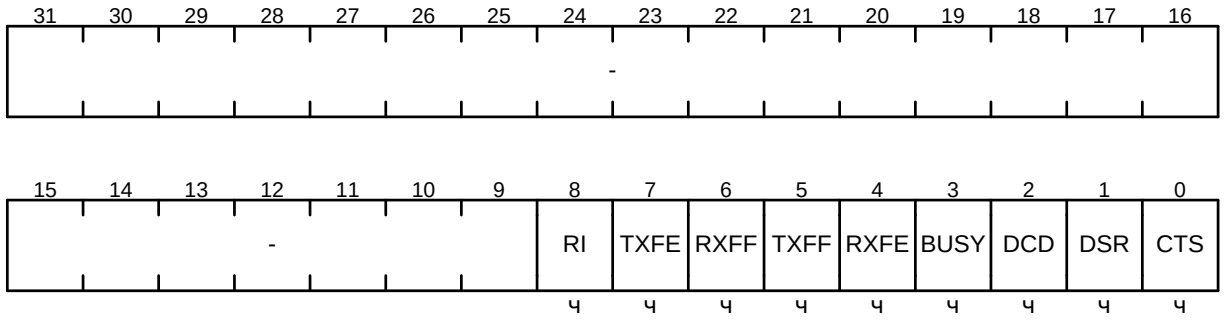


Поле	Биты	Описание	
OE	3	Флаг переполнения буфера приемника	
		0	В буфере есть свободное место или бит был сброшен после записи в регистр RSR. Содержимое буфера остается верным, так как перезаписан был только сдвиговый регистр. Центральный процессор должен считать данные для того, чтобы освободить буфер
		1	Буфер заполнен, а данные продолжают поступать
BE	2	Флаг разрыва линии	
		0	Нормальная работа или бит был сброшен после записи в регистр RSR
		1	Обнаружен признак разрыва линии, то есть наличие низкого логического уровня на входе приемника в течение времени, большего, чем длительность передачи полного кадра данных (включая стартовый, стоповый биты и бит проверки на четность). При включенном режиме FIFO данная ошибка ассоциируется с последним байтом, поступившим в буфер. В случае обнаружения разрыва линии в буфер загружается только один нулевой кадр. Прием данных возобновляется только после перехода линии в логическую единицу и последующего обнаружения корректного стартового бита
PE	1	Флаг ошибки контроля четности	
		0	Нормальная работа или бит был сброшен после записи в регистр RSR
		1	Четность принятого кадра данных не соответствует установкам битов EPS и SPS в регистре управления линией LCRH. При включенном режиме FIFO данная ошибка ассоциируется с байтом, находящимся на вершине буфера
FE	0	Флаг ошибки в структуре кадра	
		0	Нормальная работа или бит был сброшен после записи в регистр RSR
		1	В принятом символе не обнаружен корректный стоповый бит (единица). При включенном режиме FIFO данная ошибка ассоциируется с байтом, находящимся на вершине буфера
–	31-4	Зарезервировано	

Примечание – Все флаги сбрасываются записью единицы.

FR – регистр флагов

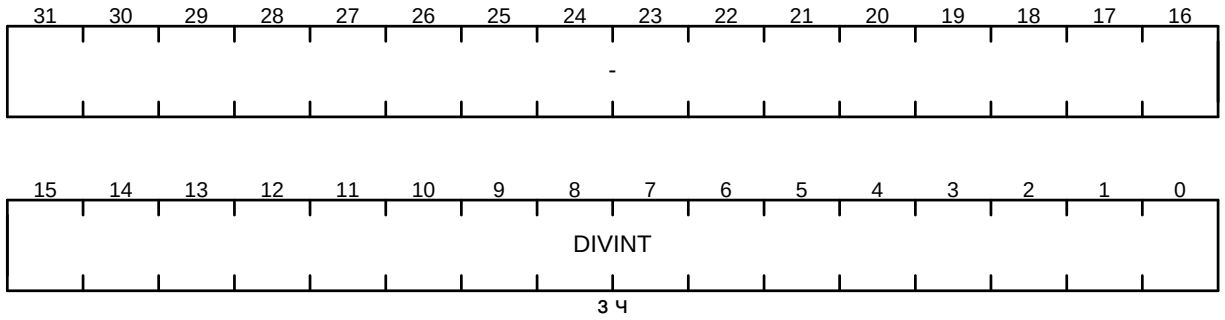
Смещение: + 18h
Сброс: 0000_0197h



Поле	Биты	Описание	
RI	8	Флаг состояния линии UARTx_RI	
		0	Линия неактивна
		1	Линия активна
TXFE/ RXFE	7/ 4	Флаг пустоты буфера передатчика/приемника. Установка флага зависит от состояния бита FEN регистра LCRH	
		0	Буфер не пуст
		1	Буфер пуст
		Примечание – Бит TXFE/RXFE не дает никакой информации о наличии данных в сдвиговом передающем регистре.	
RXFF/ TXFF	6/ 5	Флаг заполнения буфера приемника/передатчика. Установка флага зависит от состояния бита FEN регистра LCRH (т. е. включен режим FIFO или нет)	
		0	Буфер не заполнен
		1	Если режим FIFO запрещен, бит устанавливается, когда буферный регистр приемника/передатчика занят. Если режим FIFO разрешен бит устанавливается, если заполнен буфер приемника/ передатчика
BUSY	3	Бит занятости блока UART	
		0	Блок не занят
		1	Блок передает данные на линию. Бит остается установленным до тех пор, пока данные, включая стоповые биты, не будут полностью переданы. Также бит устанавливается при наличии данных в буфере передатчика, вне зависимости от состояния приемопередатчика (даже если он запрещен)
DCD	2	Флаг состояния линии UARTx_DCD	
		0	Линия неактивна
		1	Линия активна
DSR	1	Флаг состояния линии UARTx_DSR	
		0	Линия неактивна
		1	Линия активна
CTS	0	Флаг состояния линии UARTx_CTS	
		0	Линия неактивна
		1	Линия активна
–	31-9	Зарезервировано	

IBRD – регистр целой части делителя скорости обмена данными

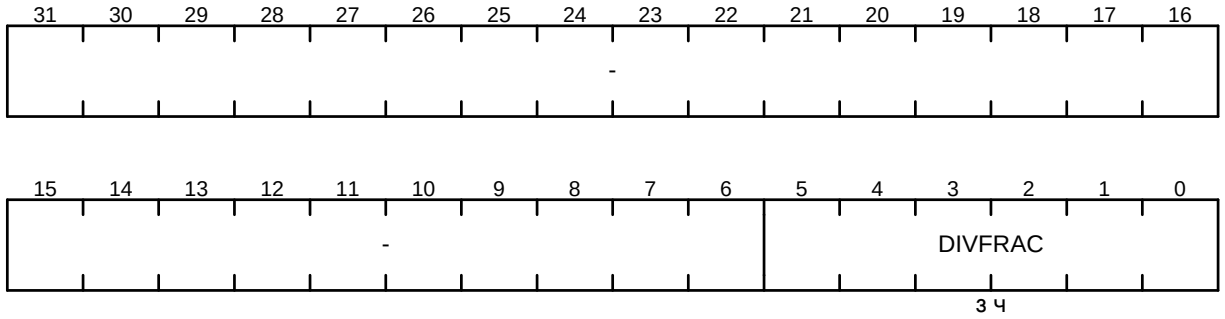
Смещение: + 24h
Сброс: 0h



Поле	Биты	Описание
DIVINT	15-0	Целая часть коэффициента деления частоты для формирования тактового сигнала передачи данных. Минимальное значение 0001h
–	31-16	Зарезервировано

FBRD – регистр дробной части делителя скорости обмена данными

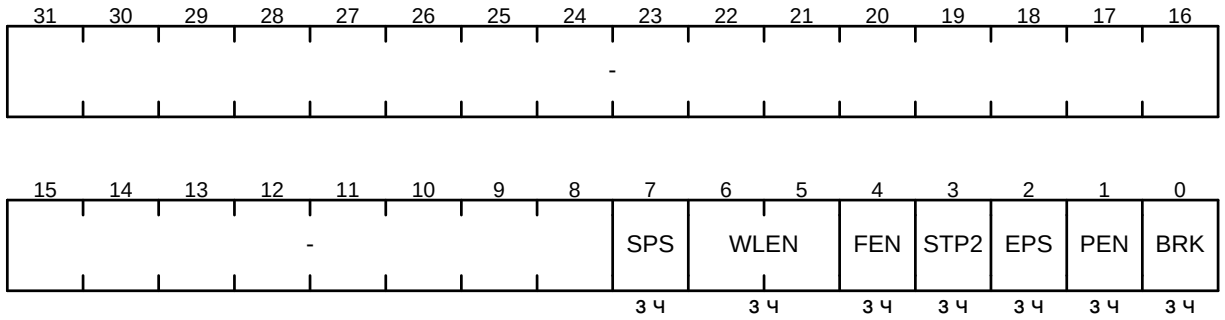
Смещение: + 28h
Сброс: 0h



Поле	Биты	Описание
DIVFRAC	5-0	Дробная часть коэффициента деления частоты для формирования тактового сигнала передачи данных. При DIVINT = FFFFh, значение DIVFRAC может быть только 00h. Примечание – Невыполнение этого условия приведет к прерыванию приема/передачи
–	31-6	Зарезервировано

LCRH – регистр управления линией

Смещение: + 2Ch
Сброс: 0h



Поле	Биты	Описание	
SPS	7	Бит разрешения передачи бита четности с фиксированным значением	
		0	Запрещено
		1	На месте бита четности передается инверсное значение бита EPS, оно же проверяется при приеме данных. При EPS = 0 на месте бита четности передается единица. При EPS = 1 на месте бита четности передается ноль
WLEN	6-5	Поле количества передаваемых/принимаемых информационных бит	
		00	5 бит
		01	6 бит
		10	7 бит
		11	8 бит
FEN	4	Бит включения режима FIFO буфера приемника и передатчика	
		0	Выключено
		1	Включено
STP2	3	Бит выбора режима передачи стопового бита	
		0	Один стоповый бит
		1	Два стоповых бита (следует помнить, что приемник не проверяет наличие дополнительного стопового бита в кадре)
EPS	2	Бит паритета. Определяет четность числа, до которого дополняется количество единиц в информационной части кадра	
		0	Нечетное число
		1	Четное число
PEN	1	Бит включения проверки четности	
		0	Выключено. Кадр не содержит бита четности
		1	Включено. Бит четности передается в кадре и проверяется
BRK	0	Флаг разрыва линии	
		0	Нормальная работа
		1	По завершении передачи текущего символа на выходе передатчика устанавливается низкий уровень сигнала. Для правильного выполнения этой операции программное обеспечение должно обеспечить передачу сигнала разрыва в течение, как минимум, времени передачи двух информационных кадров
–	31-8	Зарезервировано	

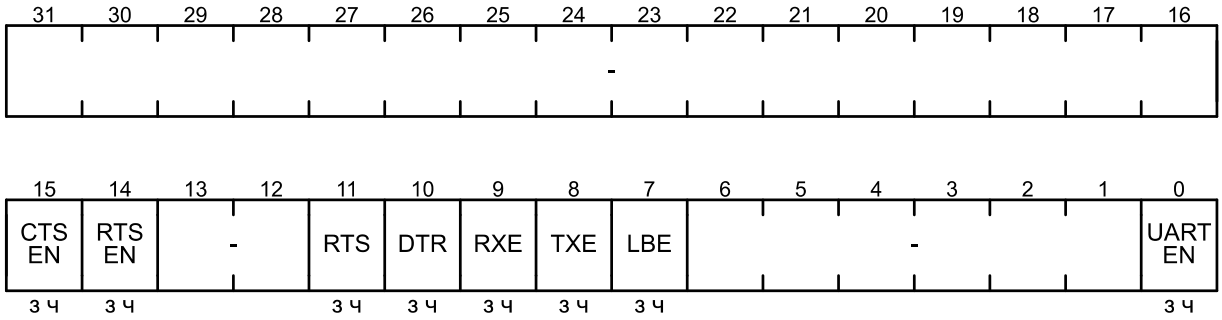
Примечание – Дополнительная информация о бите паритета кадра в таблице А.22.1.

Таблица А.22.1 – Зависимость бита паритета в кадре от состояния битов регистра LCRH

Биты регистра LCRH			Наличие и состояние бита паритета
SPS	EPS	PEN	
Не важно	Не важно	0	Не передается, не проверяется
0	0	1	Проверка нечетности слова данных
0	1	1	Проверка четности слова данных
1	0	1	Бит четности постоянно равен единице
1	1	1	Бит четности постоянно равен нулю

CR – регистр управления

Смещение: + 30h
Сброс: 300h

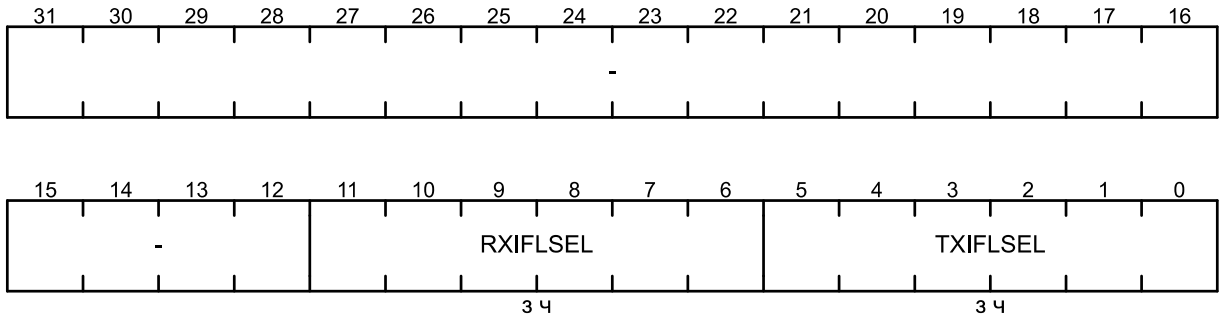


Поле	Биты	Описание	
CTSEN	15	Бит разрешения аппаратного управления потоком данных по линии CTS	
		0	Запрещено
		1	Разрешено. Данные передаются в линию только при активном значении сигнала UARTx_CTS
RTSEN	14	Бит разрешения аппаратного управления потоком данных по линии UARTx_RTS	
		0	Запрещено
		1	Разрешено. Запрос данных от внешнего устройства осуществляется только при наличии свободного места в буфере приемника
RTS	11	Бит программного управления состоянием линии модема UARTx_RTS	
		0	Высокий уровень
		1	Низкий уровень
DTR	10	Бит программного управления состоянием линии модема UARTx_DTR	
		0	Высокий уровень
		1	Низкий уровень
RXE	9	Бит разрешения приема	
		0	Запрещено
		1	Разрешено

Поле	Биты	Описание	
TXE	8	Бит разрешения передачи	
		0	Запрещено
		1	Разрешено
LBE	7	Бит включения режима «петли» – выход передатчика соединяется со входом приёмника	
		0	Запрещено
		1	Разрешено
UARTEN	0	Бит разрешения работы приемопередатчика	
		0	Запрещено
		1	Разрешено
–	31-16, 13-12, 6-1	Зарезервировано	

IFLS – регистр порога прерывания по заполнению буфера в режиме FIFO

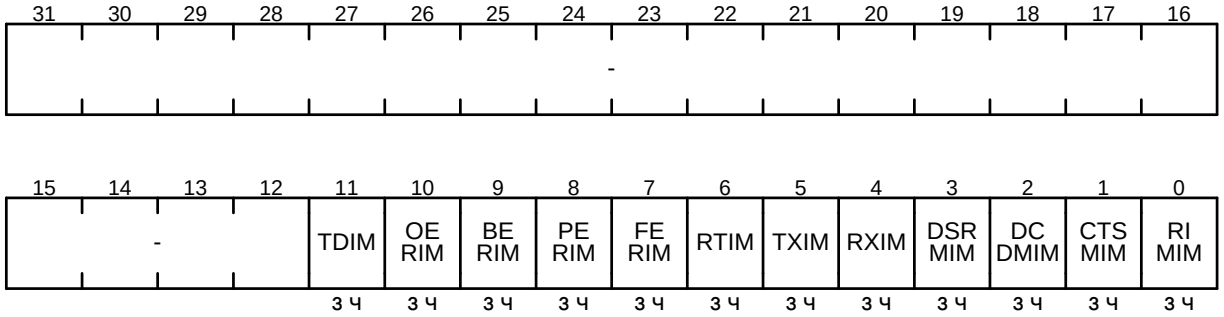
Смещение: + 34h
Сброс: 104h



Поле	Биты	Описание
RXIFLSEL	11-6	Порог заполнения буфера приемника, по достижении которого будет генерироваться прерывание. Диапазон допустимых значений: 1 – 32
TXIFLSEL	5-0	Порог опустошения буфера передатчика, по достижении которого будет генерироваться прерывание. Диапазон допустимых значений: 1 – 32
–	31-6	Зарезервировано

IMSC – регистр маски прерываний

Смещение: + 38h
Сброс: 0h

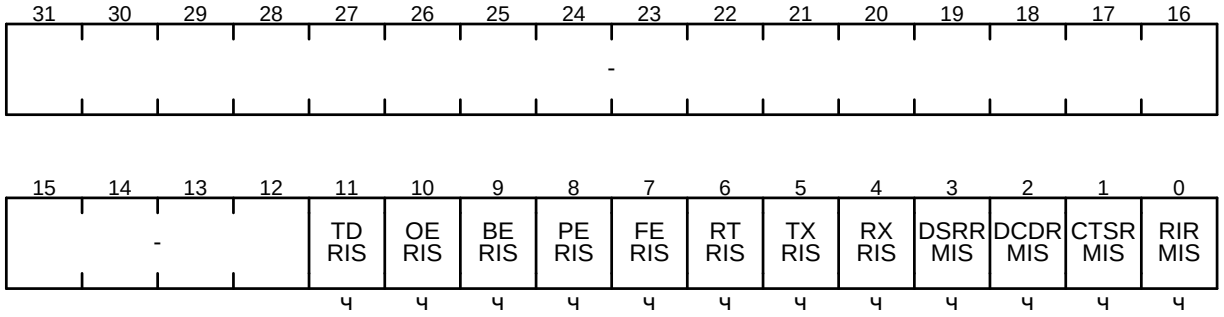


Поле	Биты	Описание
TDIM	11	Окончание передачи в линии
OERIM	10	Переполнение буфера приемника
BERIM	9	Разрыв линии
PERIM	8	Ошибка контроля четности
FERIM	7	Ошибка в структуре кадра
RTIM	6	Таймаут приема данных
TXIM	5	Порог опустошения буфер передатчика
RXIM	4	Порог переполнения буфера приемника
DSRMIM	3	Флаг изменения состояния линии UARTx_DSR
DCDMIM	2	Флаг изменения состояния линии UARTx_DCD
CTSMIM	1	Флаг изменения состояния линии UARTx_CTS
RIMIM	0	Флаг изменения состояния линии UARTx_RI
–	31-12	Зарезервировано

Примечание – Маска прерываний формируется установкой бит. Необходимо установить биты тех событий, прерывания от которых нужно разрешить.

RIS – регистр состояния прерываний

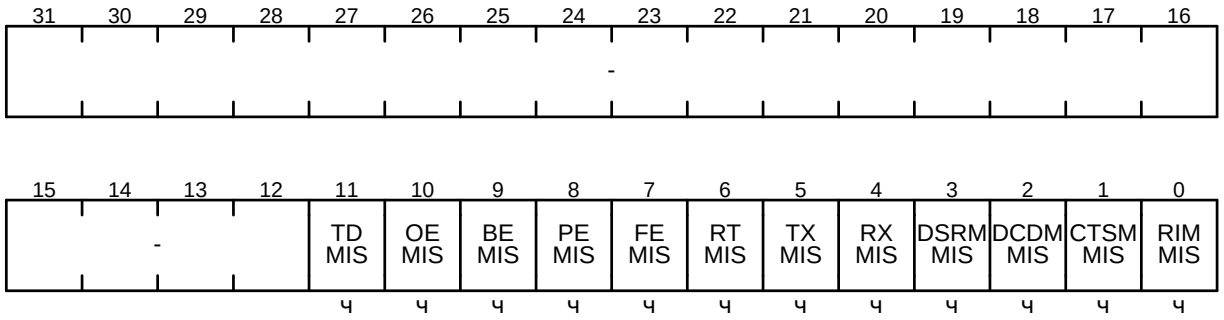
Смещение: + 3Ch
Сброс: 0h



Поле	Биты	Описание
TDRIS	11	Флаг окончания передачи в линии
OERIS	10	Флаг переполнения буфера приемника
BERIS	9	Флаг разрыва линии
PERIS	8	Флаг ошибки контроля четности
FERIS	7	Флаг ошибки в структуре кадра
RTRIS	6	Флаг таймаута приема данных. Примечание – Данный статусный бит не устанавливается, если не установлен бит разрешения RTIM регистра IMSC. Таким образом, биты RTRIS и RTMIS регистров RIS и MIS будут установлены синхронно.
TXRIS	5	Флаг порога опустошения буфер передатчика
RXRIS	4	Флаг порога переполнения буфера приемника
DSRRMIS	3	Флаг изменения состояния линии UARTx_DSR
DCDRMIS	2	Флаг изменения состояния линии UARTx_DCD
CTSRMIS	1	Флаг изменения состояния линии UARTx_CTS
RIRMIS	0	Флаг изменения состояния линии UARTx_RI
–	31-12	Зарезервировано

MIS – регистр состояния прерываний с маскированием

Смещение: + 40h
Сброс: 0h

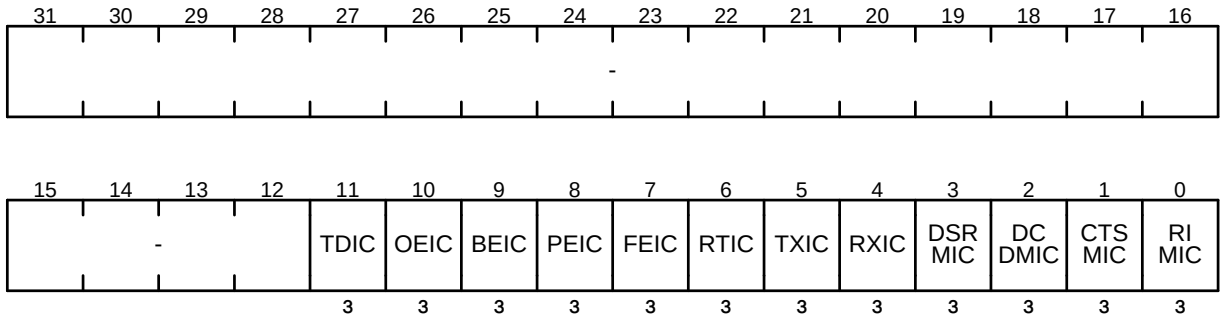


Поле	Биты	Описание
TDMIS	11	Флаг окончания передачи в линии
OEMIS	10	Флаг переполнения буфера приемника
BEMIS	9	Флаг разрыва линии
PEMIS	8	Флаг ошибки контроля четности
FEMIS	7	Флаг ошибки в структуре кадра
RTMIS	6	Флаг таймаута приема данных
TXMIS	5	Флаг порога опустошения буфера передатчика
RXMIS	4	Флаг порога переполнения буфера приемника
DSRMMIS	3	Флаг изменения состояния линии UARTx_DSR
DCDMMIS	2	Флаг изменения состояния линии UARTx_DCD
CTSMMIS	1	Флаг изменения состояния линии UARTx_CTS
RIMMIS	0	Флаг изменения состояния линии UARTx_RI
–	31-12	Зарезервировано

Примечание – Устанавливаются флаги, которые закрыты маской регистра IMSC.

ICR – регистр сброса прерываний

Смещение: + 44h
Сброс: 0h

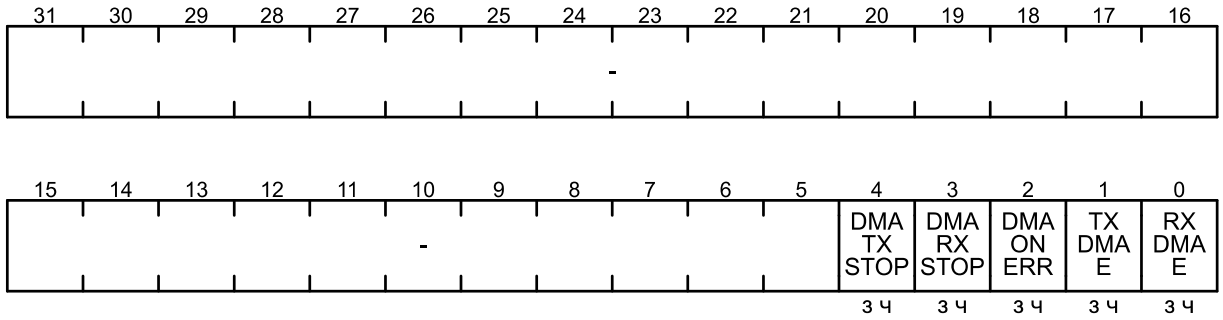


Поле	Биты	Описание
TDIC	11	Бит сброса флага окончания передачи в линии
OEIC	10	Бит сброса флага переполнения буфера приемника
BEIC	9	Бит сброса флага разрыва линии
PEIC	8	Бит сброса флага ошибки контроля четности
FEIC	7	Бит сброса флага ошибки в структуре кадра
RTIC	6	Бит сброса флага таймаута приема данных
TXIC	5	Бит сброса флага порога опустошения буфера передатчика
RXIC	4	Бит сброса флага порога переполнения буфера приемника
DSRMIC	3	Бит сброса флага изменения состояния линии UARTx_DSR
DCDMIC	2	Бит сброса флага изменения состояния линии UARTx_DCD
CTSMIC	1	Бит сброса флага изменения состояния линии UARTx_CTS
RIMIC	0	Бит сброса флага изменения состояния линии UARTx_RI
–	31-12	Зарезервировано

Примечание – Запись единиц в биты регистра сбрасывает соответствующие им флаги в регистрах RIS и MIS, а также прерывания, вызвавшие установку этих флагов.

DMACR – регистр управления прямым доступом к памяти

Смещение: + 48h
Сброс: 0h



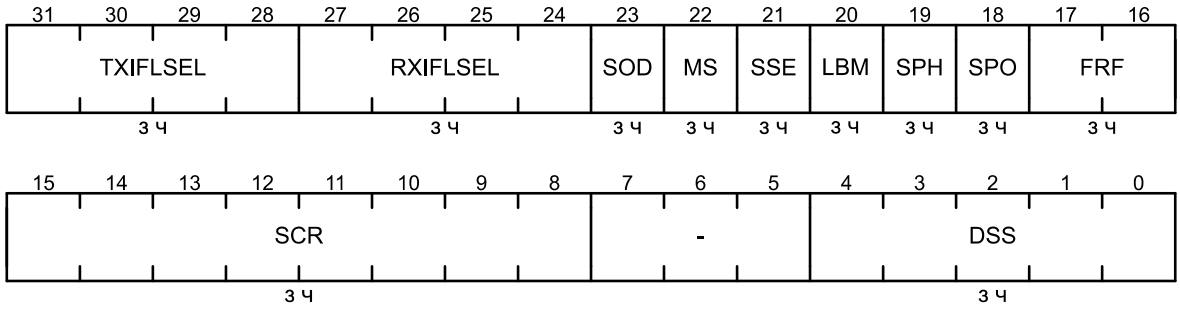
Поле	Биты	Описание
DMATXSTOP	4	Автоматическое снятие запроса DMA TX и сброс бита TXDMAE, когда канал DMA завершает все транзакции
DMARXSTOP	3	Автоматическое снятие запроса DMA RX и сброс бита RXDMAE, когда канал DMA завершает все транзакции
DMAONERR	2	Блокирование запросов UARTRXDMASREQ и UARTRXDMABREQ от приемника в случае прерывания по ошибке
		0 Выключено
		1 Включено
TXDMAE	1	Бит разрешения формирования запросов блока DMA для обслуживания буфера передатчика
		0 Запрещено
		1 Разрешено
RXDMAE	0	Бит разрешения формирования запросов блока DMA для обслуживания буфера приемника
		0 Запрещено
		1 Разрешено
–	31-3	Зарезервировано

A.23 Регистры контроллера интерфейса SPI

Базовый адрес:	5200_8000h	Регистры контроллера SPI0
	5200_9000h	Регистры контроллера SPI1
	5200_A000h	Регистры контроллера SPI2
	5200_B000h	Регистры контроллера SPI3
	5200_C000h	Регистры контроллера SPI4
	5200_D000h	Регистры контроллера SPI5
	5200_E000h	Регистры контроллера SPI6
	5200_F000h	Регистры контроллера SPI7

CR – регистр управления

Смещение: + 00h
Сброс: 4400_0000h

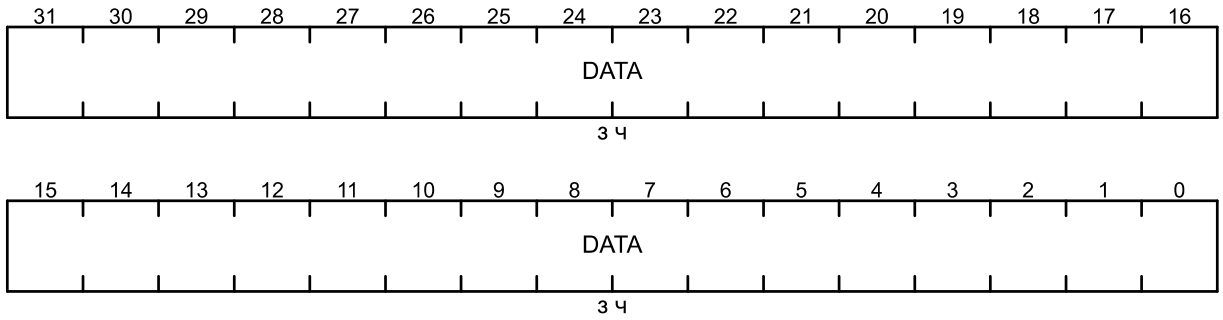


Поле	Биты	Описание	
TXIFLSEL	31-28	Величина порога опустошения передающего FIFO. При опустошении до уровня порога или ниже может быть сгенерировано прерывание или запрос DMA. Допустимый диапазон значений 0h-8h (по умолчанию – 4h). Для генерации прерываний/запросов DMA значение поле должно быть отличным нуля	
RXIFLSEL	27-24	Величина порога наполнения принимающего FIFO. При заполнении до уровня порога или выше может быть сгенерировано прерывание или запрос DMA. Допустимый диапазон значений 0h-8h (по умолчанию – 4h). Для генерации прерываний/запросов DMA значение поле должно быть отличным от нуля	
SOD	23	Бит запрета передачи данных. В режиме мастера значение бита игнорируется. В режиме ведомого бит контролирует выход данных. Пока бит сброшен передача и прием данных разрешены. Установка бита блокирует передачу данных и переводит вывод SPIx_TX в состояние слабой логической единицы, при этом прием тактового сигнала и прием данных не блокируется	
MS	22	Бит выбора режима работы	
		0	Мастер
		1	Ведомый
SSE	21	Бит разрешения работы приемопередатчика	
		0	Запрещено
		1	Разрешено

Поле	Биты	Описание	
LBM	20	Режим «обратной петли» (тестовый)	
		0	Обычный режим работы
		1	Режим «обратной петли» (выводы SPIx_RX и SPIx_TX замкнуты внутри микроконтроллера)
SPH	19	Фаза сигнала SPIx_SCK (только для протокола обмена SPI)	
		0	Выборка данных по переднему фронту синхросигнала, а установка – по заднему
		1	Выборка данных по заднему фронту синхросигнала, а установка – по переднему
SPO	18	Полярность сигнала SPIx_SCK (только для протокола обмена SPI)	
		0	В режиме ожидания линия SPIx_SCK удерживается в состоянии логического нуля
		1	В режиме ожидания линия SPIx_SCK удерживается в состоянии логической единицы
FRF	17-16	Поле выбора протокола обмена информацией	
		00	SPI
		01	SSI
		10	Microwire
		11	Зарезервировано
SCR	15-8	Коэффициент деления второго делителя. Может принимать значения 00h до FFh Примечание – Если запись поля SCR выполнялась с использованием побитового доступа, тогда все остальные биты регистра обнуляются и требуется их настройка	
DSS	4-0	Размер слова данных	
		0h-2h	Зарезервировано
		03h	4 бита
		04h	5 бит
		...	…
		1Eh	31 бит
		1Fh	32 бит
–	20, 7-5	Зарезервировано	

DR – регистр данных

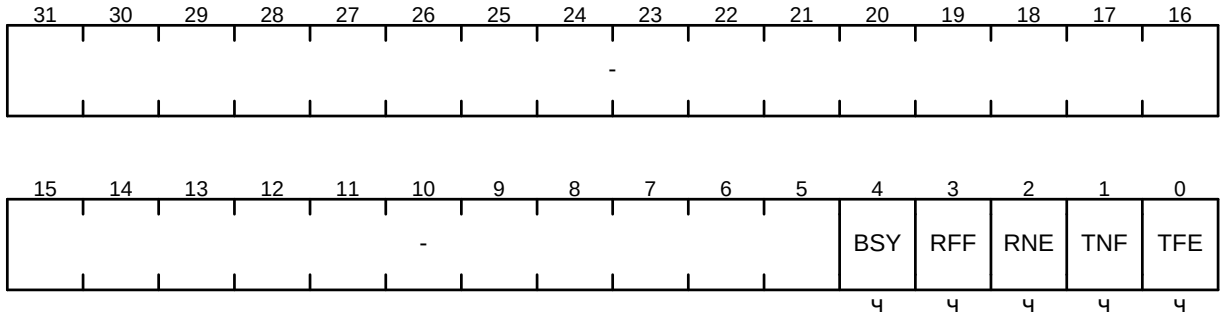
Смещение: + 08h
Сброс: 0h



Поле	Биты	Описание
DATA	31-0	32-разрядный буфер FIFO приемника и передатчика. Данные для передачи записываются в регистр. Если размер данных менее 32 бит, они должны быть выравнены по правой границе. Принятые данные автоматически выравниваются по правой границе. Принятые данные возвращаются при чтении регистра

SR – регистр состояния

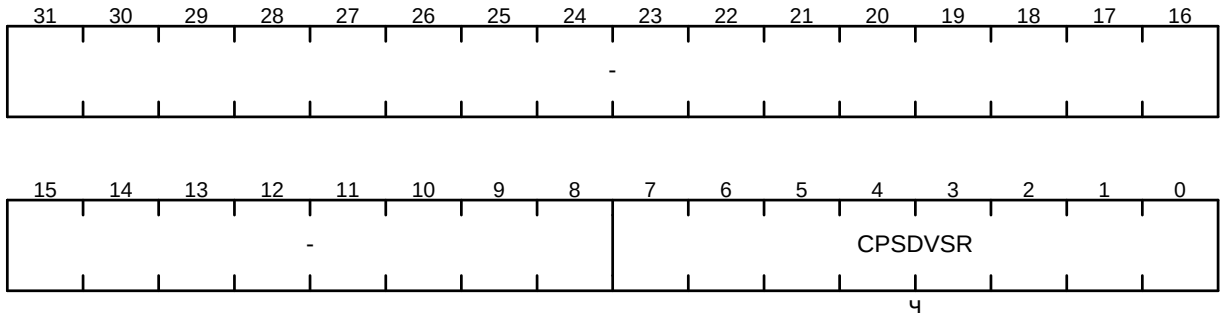
Смещение: + 0Ch
Сброс: 3h



Поле	Биты		Описание
BSY	4	0	Приемопередатчик не активен
		1	Передача/прием данных, либо буфер FIFO не пуст
RFF	3	0	Буфер FIFO приемника не заполнен
		1	Буфер FIFO приемника заполнен
RNE	2	0	Буфер FIFO приемника пуст
		1	Буфер FIFO приемника не пуст
TNF	1	0	Буфер FIFO передатчика заполнен
		1	Буфер FIFO передатчика не заполнен
TFE	0	0	Буфер FIFO передатчика не пуст
		1	Буфер FIFO передатчика пуст
–	31-5	Зарезервировано	

CPSR – регистр делителя тактовой частоты

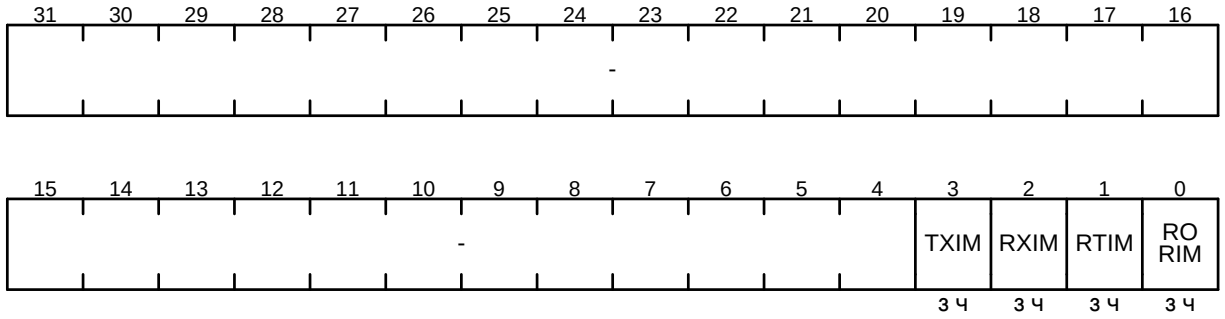
Смещение: + 10h
Сброс: 0h



Поле	Биты	Описание
CPSDVSR	7-0	Коэффициент деления первого делителя. Может принимать четные значения от 02h до FEh
–	31-8	Зарезервировано

IMSC – регистр маски прерываний

Смещение: + 14h
Сброс: 0h

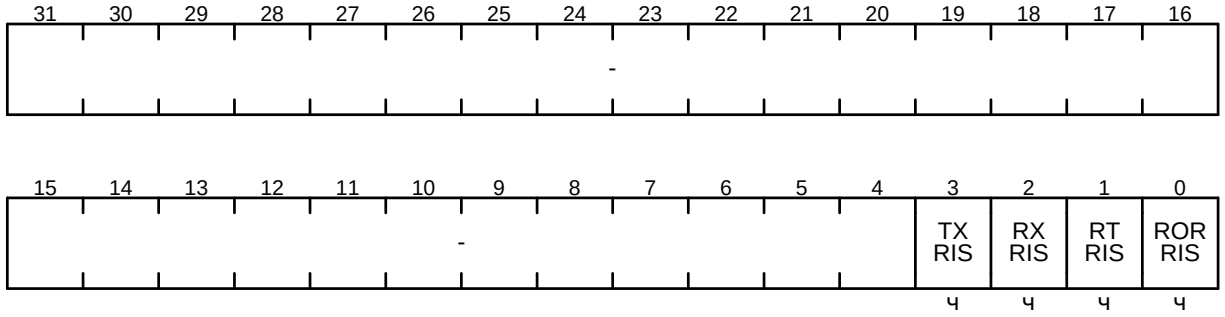


Поле	Биты	Описание
TXIM	3	Буфер передатчика опустошен до величины порога или ниже
RXIM	2	Буфер приемника заполнен на величину порога или выше
RTIM	1	Таймаут приема данных
RORIM	0	Переполнение буфера приемника
–	31-4	Зарезервировано

Примечание – Установка/сброс бит формирует маску. По умолчанию все биты сброшены, и установка флагов запрещена. Необходимо установить биты тех событий, прерывания от которых нужно разрешить.

RIS – регистр состояния прерываний

Смещение: + 18h
Сброс: 8h

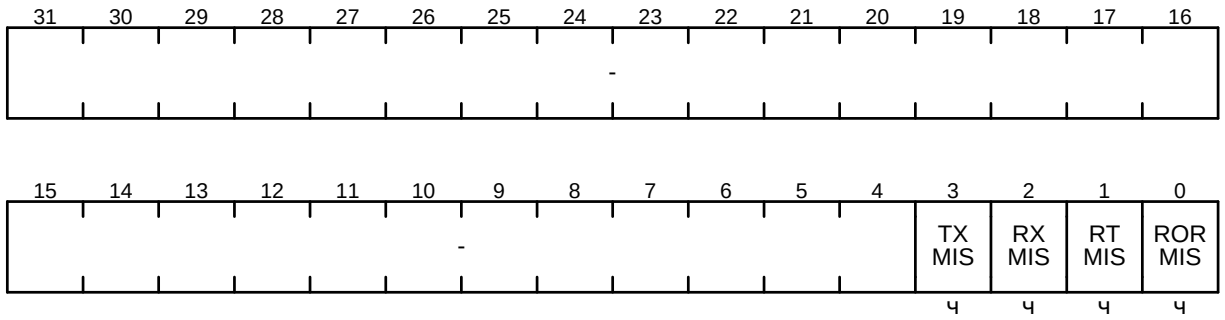


Поле	Биты	Описание
TXRIS	3	Буфер передатчика опустошен до величины порога или ниже
RXRIS	2	Буфер приемника заполнен на величину порога или выше
RTRIS	1	Таймаут приема данных
RORRIS	0	Переполнение буфера приемника
–	31-4	Зарезервировано

Примечание – При возникновении прерываний устанавливаются соответствующие им немаскируемые флаги. Биты RTRIS также сбрасываются после чтения буфера приемника.

MIS – регистр состояния прерываний с маскированием

Смещение: + 1Ch
Сброс: 0h

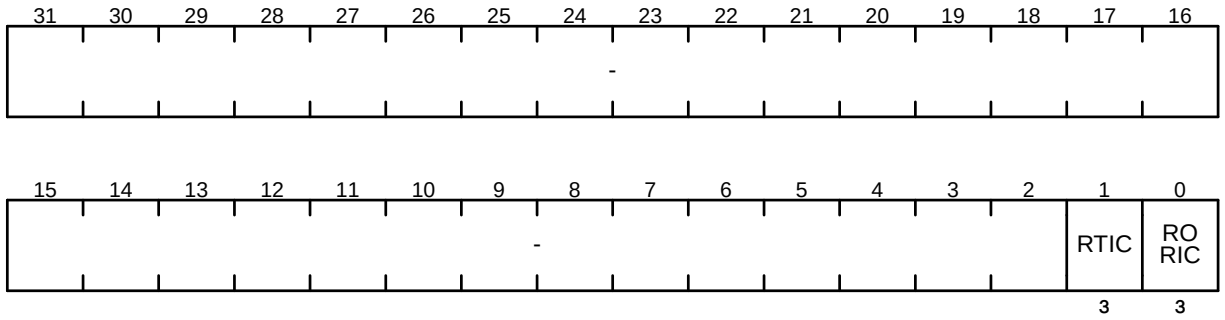


Поле	Биты	Описание
TXMIS	3	Буфер передатчика опустошен до величины порога или ниже
RXMIS	2	Буфер приемника заполнен на величину порога или выше
RTMIS	1	Таймаут приема данных
RORMIS	0	Переполнение буфера приемника
–	31-4	Зарезервировано

Примечание – В регистре устанавливаются только те флаги, которые закрыты маской регистра IMSC. Бит RTMIS также сбрасывается после чтения буфера приемника.

ICR – регистр сброса прерываний

Смещение: + 20h
Сброс: 0h

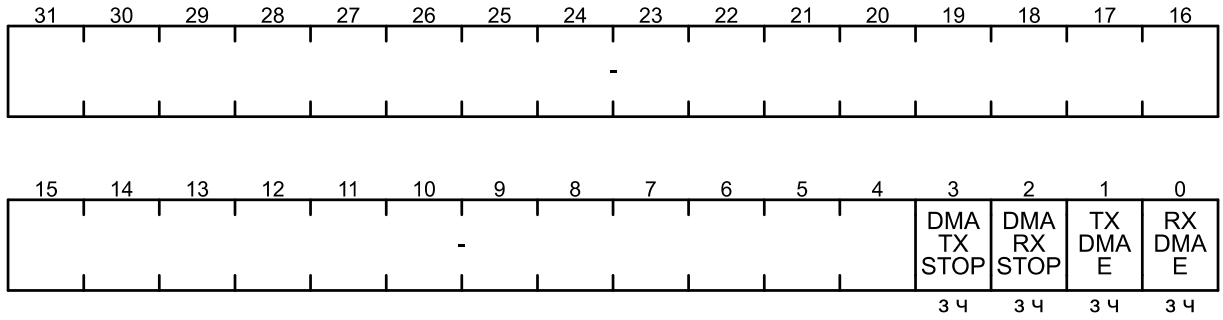


Поле	Биты	Описание
RTIC	1	Таймаут приема данных
RORIC	0	Переполнение буфера приемника
–	31-4	Зарезервировано

Примечание – Запись единиц в биты регистра сбрасывает соответствующие им флаги в регистрах RIS и MIS, а также прерывания, вызвавшие установку этих флагов.

DMACR – регистр управления прямым доступом к памяти

Смещение: + 24h
Сброс: 0h



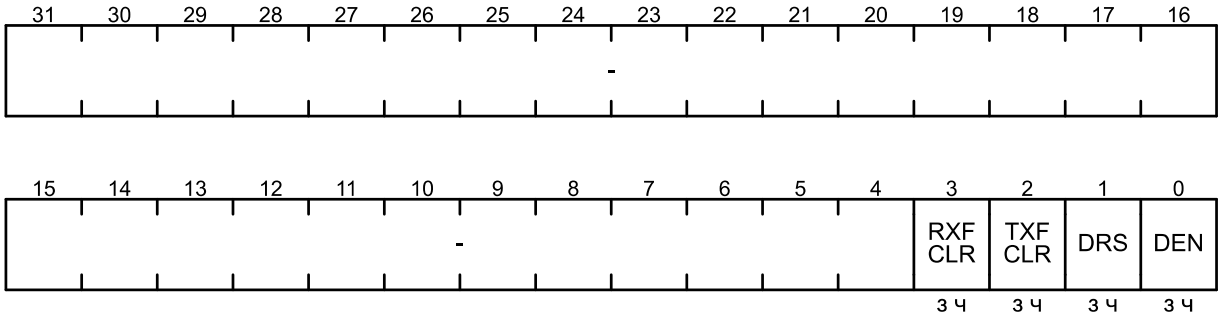
Поле	Биты	Описание	
DMATXSTOP	4	Автоматическое снятие запроса DMA TX и сброс бита TXDMAE, когда канал DMA завершает все транзакции	
DMARXSTOP	3	Автоматическое снятие запроса DMA RX и сброс бита RXDMAE, когда канал DMA завершает все транзакции	
TXDMAE	1	Бит разрешения использования контроллера DMA при передаче	
		0	Не используется
		1	Разрешено формирование запросов контроллера DMA для обслуживания буфера FIFO передатчика
RXDMAE	0	Бит разрешения использования контроллера DMA при приеме	
		0	Не используется
		1	Разрешено формирование запросов контроллера DMA для обслуживания буфера FIFO приемника
–	31-2	Зарезервировано	

A.24 Регистры контроллера интерфейса QSPI

Базовый адрес: 4103_0000h Регистры контроллера QSPI0
4104_0000h Регистры контроллера QSPI1

HCR – регистр управления

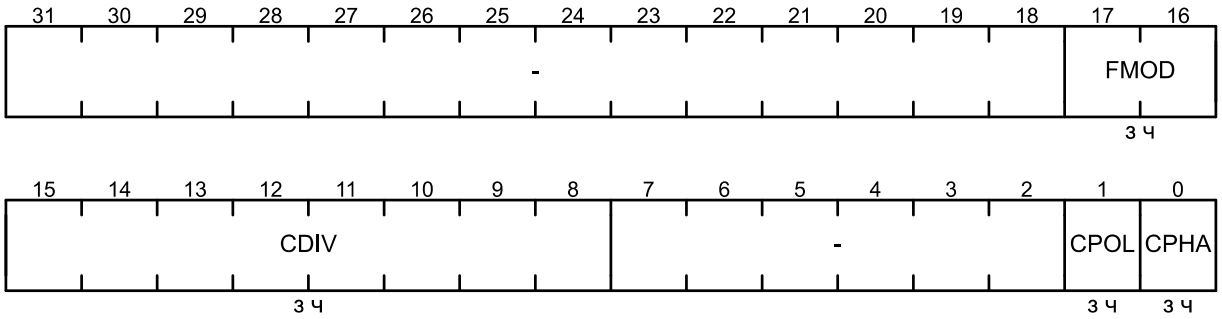
Смещение: + 00h
Сброс: 00h



Поле	Биты	Описание
RXCLR	3	Бит очистки приемного буфера FIFO
TXCLR	2	Бит очистки передающего буфера FIFO
DRS	1	Бит сброса устройства (останова активной передачи)
DEN	0	Бит разрешения работы
–	31-4	Зарезервировано

DCR – регистр конфигурации устройства

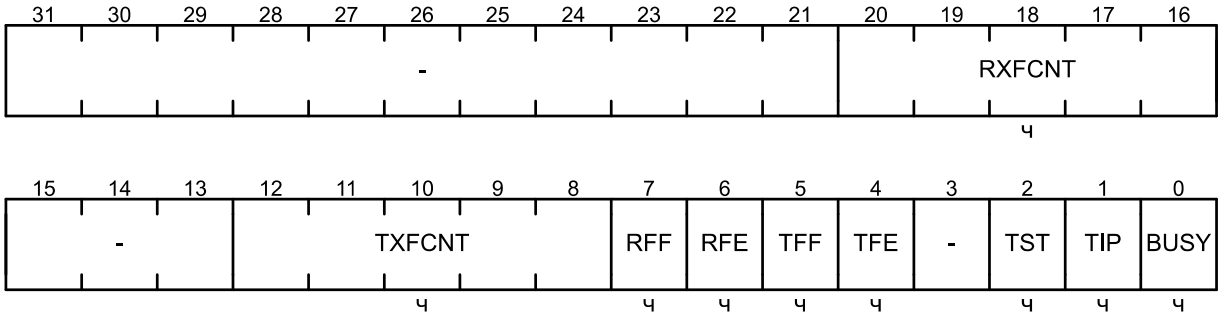
Смещение: + 04h
Сброс: 0000_0100h



Поле	Биты	Описание	
FMOD	17-16	Выбор режима работы	
		0	Стандартный режим SPI
		1	Зарезервировано
		2	Интерфейс памяти QSPI
		3	Зарезервировано
CDIV	15-8	Значение делителя	
CPOL	1	Полярность сигнала QSPI_SCK	
CPHA	0	Фаза сигнала QSPI_SCK	
–	31-19, 7-2	Зарезервировано	

DSR – регистр статуса

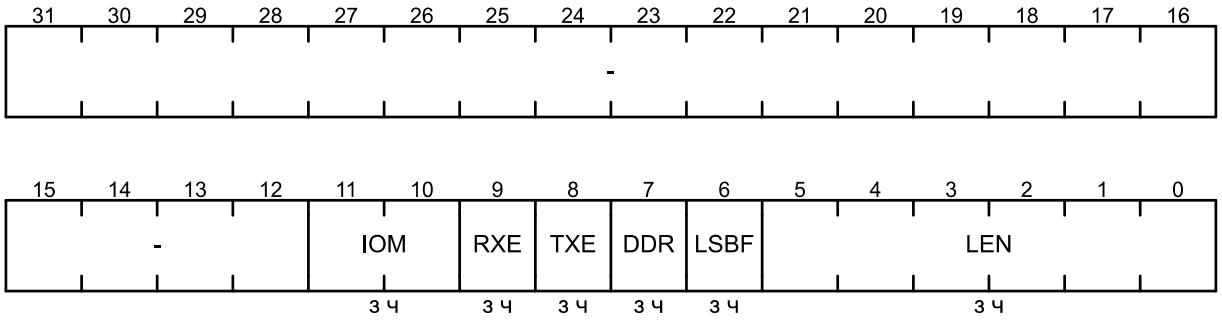
Смещение: + 08h
Сброс: 0000_0050h



Поле	Биты	Описание
RXFCNT	20-16	Количество элементов в буфере FIFO приемника
TXFCNT	12-8	Количество элементов в буфере FIFO передатчика
RFF	7	Буфер FIFO приемника заполнен
RFE	6	Буфер FIFO приемника пуст
TFF	5	Буфер FIFO передатчика заполнен
TFE	4	Буфер FIFO передатчика пуст
TST	2	Передача приостановлена
TIP	1	Передача в процессе
BUSY	0	0 Приемопередатчик не активен
		1 Передача/прием данных
–	31-21, 15-13, 3	Зарезервировано

TCR – регистр конфигурации передачи

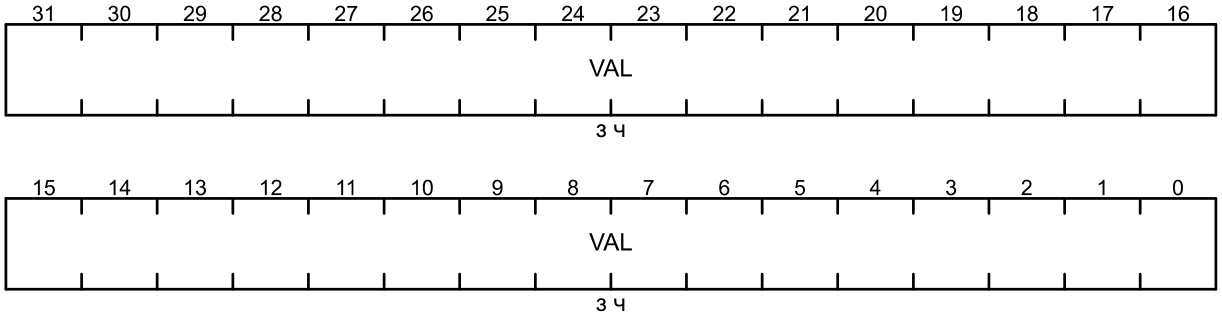
Смещение: + 0Ch
Сброс: 00h



Поле	Биты	Описание	
IOM	11-10	Выбор количества линий данных	
		0	Одна линия передачи (Single)
		1	
		2	Две линии передачи (Double)
		3	Четыре линии передачи (Quad)
RXE	9	Бит разрешения приема	
TXE	8	Бит разрешения передачи	
DDR	7	0	Обычная скорость передачи данных (изменение данных по одному фронту QSPI_SCK)
		1	Двойная скорость передачи данных (изменение данных по обоим фронтам QSPI_SCK)
LSBF	6	0	Передача старшим битом вперед
		1	Передача младшим битом вперед
LEN	5-0	Размер слова данных (количество бит). Значение от 1 до 32	
–	31-12	Зарезервировано	

TDR – регистр передачи данных

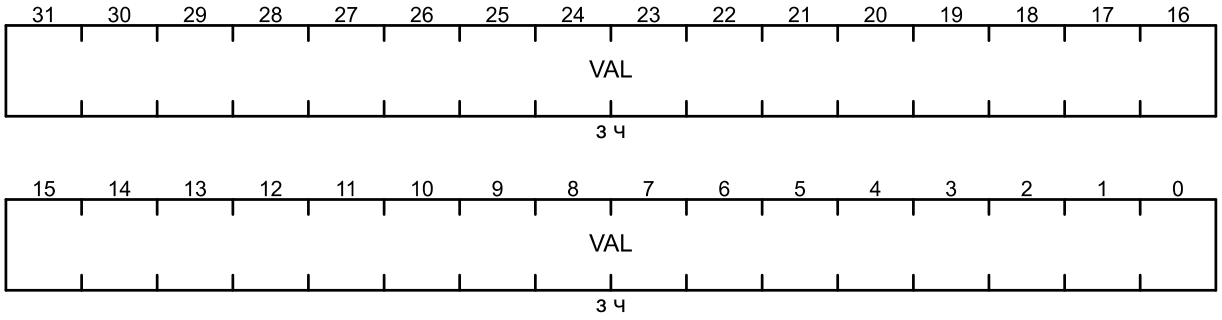
Смещение: + 10h
Сброс: 00h



Поле	Биты	Описание
VAL	31-0	32-разрядный буфер FIFO приемника и передатчика. Данные для передачи записываются в регистр. Принятые данные возвращаются при чтении регистра.

TDS – регистр количества слов данных

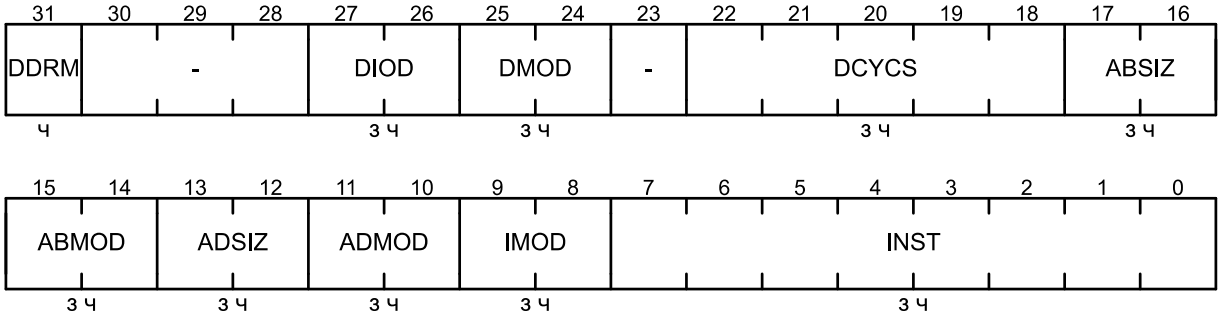
Смещение: + 14h
Сброс: 00h



Поле	Биты	Описание
VAL	31-0	Количество слов данных для приема/передачи

QCC – регистр конфигурации обмена QSPI

Смещение: + 18h
Сброс: 00h

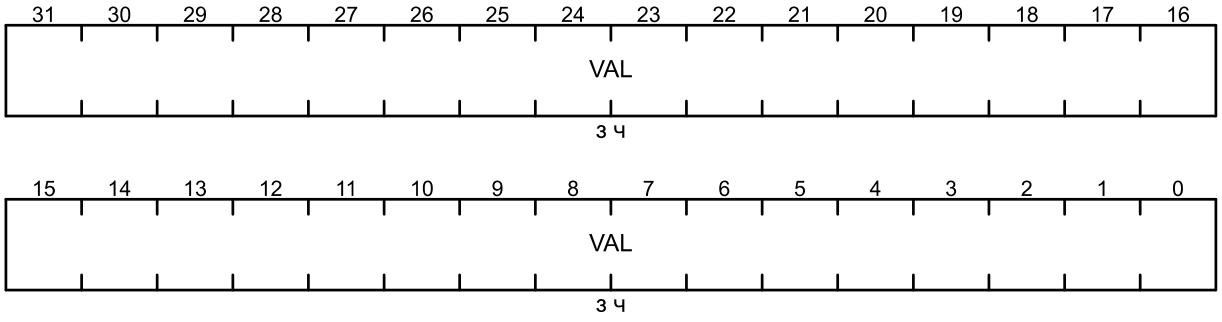


Поле	Биты	Описание	
DDRM	31	Выбор режима передачи по обоим фронтам. Бит оказывает влияние в режиме QSPI только для адреса, дополнительных данных и данных.	
		0	Обычная скорость передачи данных (изменение данных по одному фронту QSPI_SCK)
		1	Двойная скорость передачи данных (изменение данных по обоим фронтам QSPI_SCK)
DIOD	27-26	Выбор направления передачи данных	
		0	Передача данных во внешнее устройство
		1	Прием данных из внешнего устройства
		2	Зарезервировано
		3	Зарезервировано
DMOD	25-24	Выбор режима данных	
		0	Отсутствует передача данных
		1	Передача данных по одной линии данных (SIO)
		2	Передача данных по двум линиям данных (DIO)
		3	Передача данных по четырем линиям данных (QIO)
DCYCS	22-18	Количество циклов ожидания (число периодов QSPI_CLK). Значения от 0 до 31	

Поле	Биты	Описание	
ABSIZ	17-16	Выбор размера дополнительных данных	
		0	8 бит
		1	16 бит
		2	24 бита
		3	32 бита
ABMOD	15-14	Выбор режима передачи дополнительных данных	
		0	Отсутствует передача дополнительных данных
		1	Передача дополнительных данных по одной линии данных (SIO)
		2	Передача дополнительных данных по двум линиям данных (DIO)
		3	Передача дополнительных данных по четырем линиям данных (QIO)
ADSIZ	13-12	Выбор размера адреса	
		0	8-битный адрес
		1	16-битный адрес
		2	24-битный адрес
		3	32-битный адрес
ADMOD	11-10	Выбор режима передачи адреса	
		0	Отсутствует передача адреса
		1	Передача адреса по одной линии данных (SIO)
		2	Передача адреса по двум линиям данных (DIO)
		3	Передача адреса по четырем линиям данных (QIO)
IMOD	9-8	Выбор режима инструкции	
		0	Нет иструкции
		1	Инструкция для режима одной линии данных (SIO)
		2	Инструкция для режима двух линий данных (DIO)
		3	Инструкция для режима четырех линий данных (QIO)
INST	7-0	Код инструкции, отправляемый во внешнее устройство по QSPI	
–	30-28, 23	Зарезервировано	

QAD – регистр адреса

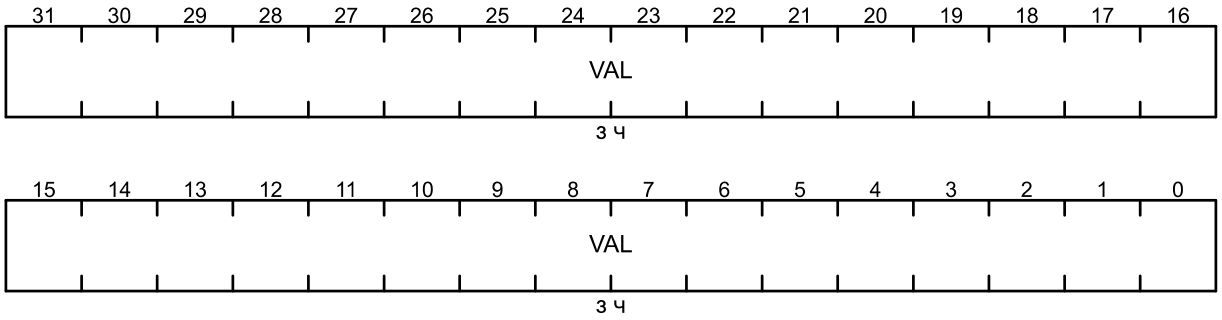
Смещение: + 1Ch
Сброс: 00h



Поле	Биты	Описание
VAL	31-0	Значение адреса, передаваемого во внешнюю Flash-память

QAB – регистр дополнительных данных

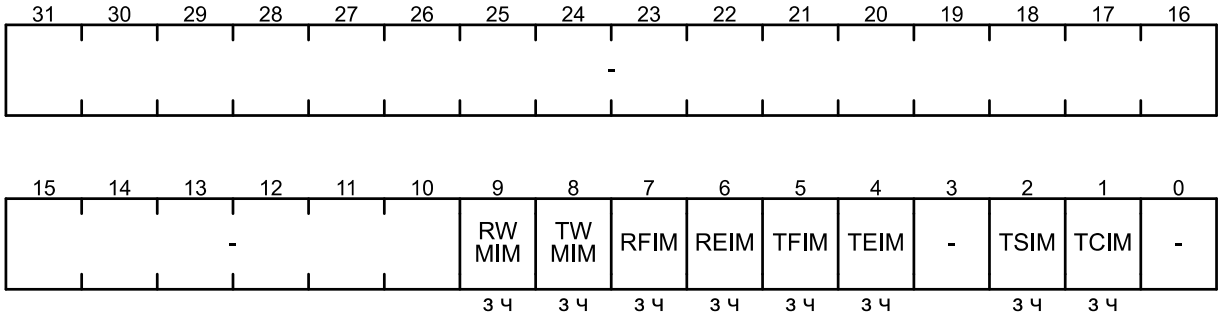
Смещение: + 20h
Сброс: 00h



Поле	Биты	Описание
VAL	31-0	Значение дополнительных данных, передаваемых во внешнее устройство

IMR – регистр маски прерываний

Смещение: + 24h
Сброс: 00h



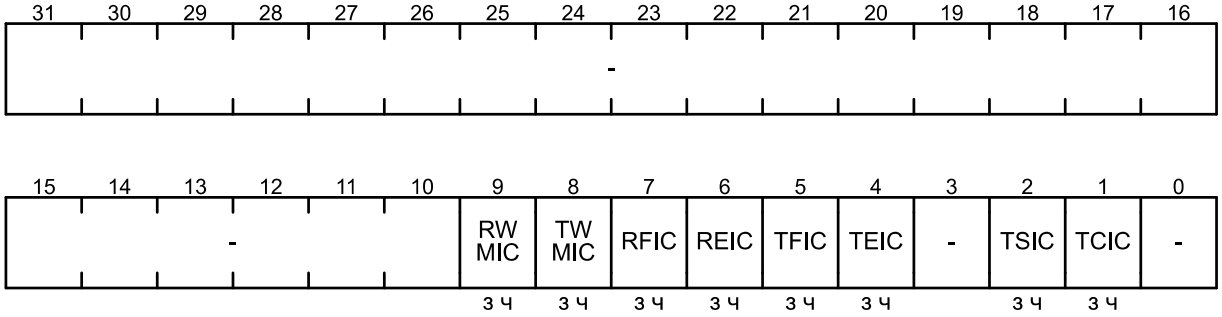
Поле	Биты	Описание
RWMIM	9	Порог переполнения буфера приемника
TWMIM	8	Порог опустошения буфер передатчика
RFIM	7	Буфер FIFO приемника заполнен
REIM	6	Буфер FIFO приемника пуст
TFIM	5	Буфер FIFO передатчика заполнен
TEIM	4	Буфер FIFO передатчика пуст
TSIM	2	Останов передачи
TCIM	1	Завершение передачи
–	31-10, 3, 0	Зарезервировано

Примечание – Установка/сброс бит формирует маску. По умолчанию все биты сброшены, и установка флагов запрещена. Необходимо установить биты тех событий, прерывания от которых нужно разрешить.

ICR – регистр сброса прерываний

Смещение: + 28h

Сброс: 00h

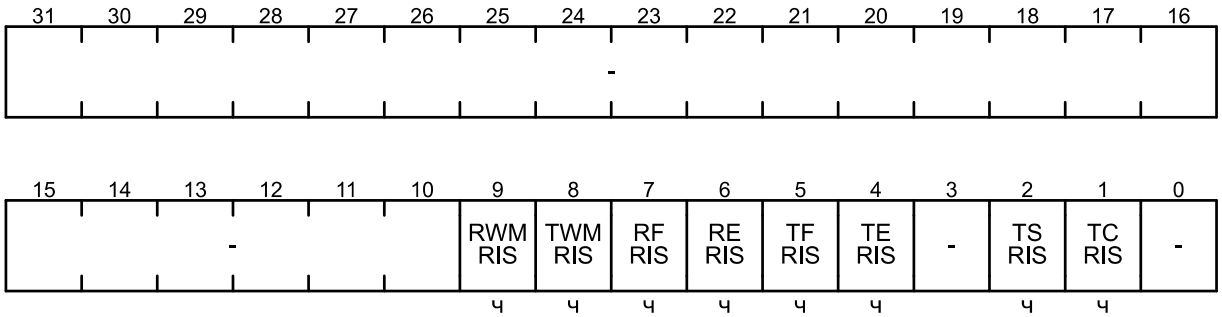


Поле	Биты	Описание
RWIC	9	Бит сброса прерывания по достижению порога переполнения буфера приемника
TWIC	8	Бит сброса прерывания по достижению порога опустошения буфер передатчика
RFIC	7	Бит сброса прерывания по заполнению буфера FIFO приемника
REIC	6	Бит сброса прерывания по опустошению буфера FIFO приемника
TFIC	5	Бит сброса прерывания по заполнению буфера FIFO передатчика
TEIC	4	Бит сброса прерывания по опустошению буфера FIFO передатчика
TSIC	2	Бит сброса прерывания по остановке передачи
TCIC	1	Бит сброса прерывания по завершению передачи
–	31-10, 3, 0	Зарезервировано

RIS – регистр состояния прерываний

Смещение: + 2Ch

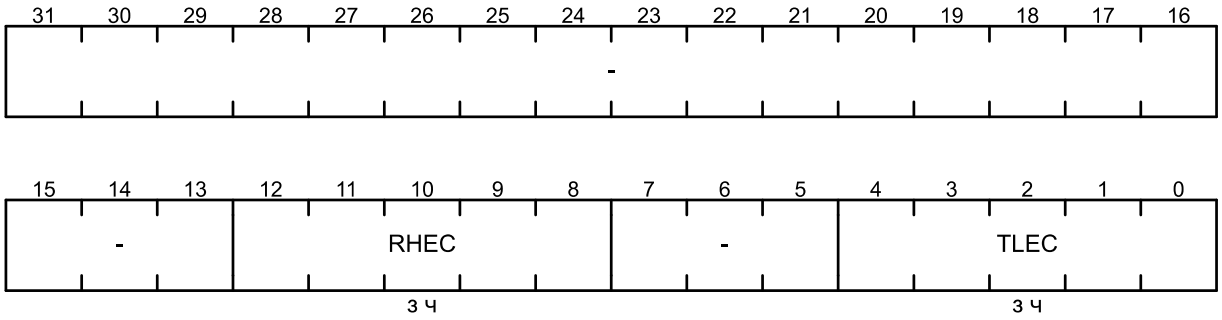
Сброс: 00h



Поле	Биты	Описание
RWRIS	9	Флаг порогового заполнения буфера приемника
TWRIS	8	Флаг порогового опустошения буфера передатчика
RFRIS	7	Флаг заполнения буфера приемника
RERIS	6	Флаг опустошения буфера приемника
TFRIS	5	Флаг заполнения буфера передатчика
TERIS	4	Флаг опустошения буфера передатчика
TSRIS	2	Флаг останова передачи
TCRIS	1	Флаг завершение передачи
–	31-10, 3, 0	Зарезервировано

FWM – регистр пороговых значений FIFO

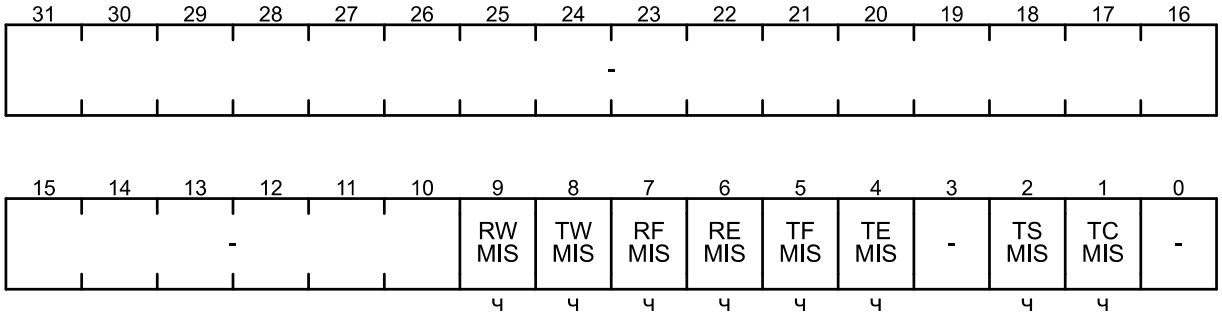
Смещение: + 30h
Сброс: 0000_0E02h



Поле	Биты	Описание
RHEC	12-8	Уровень порогового заполнения буфера приемника. При достижении заданного значения будет выставлен флаг RWRIS регистра RIS
TLEC	4-0	Уровень порогового опустошения буфера передатчика. При достижении заданного значения будет выставлен флаг TWRIS регистра RIS
–	31-13, 7-6	Зарезервировано

MIS – регистр состояния прерываний с маскированием

Смещение: + 34h
Сброс: 00h



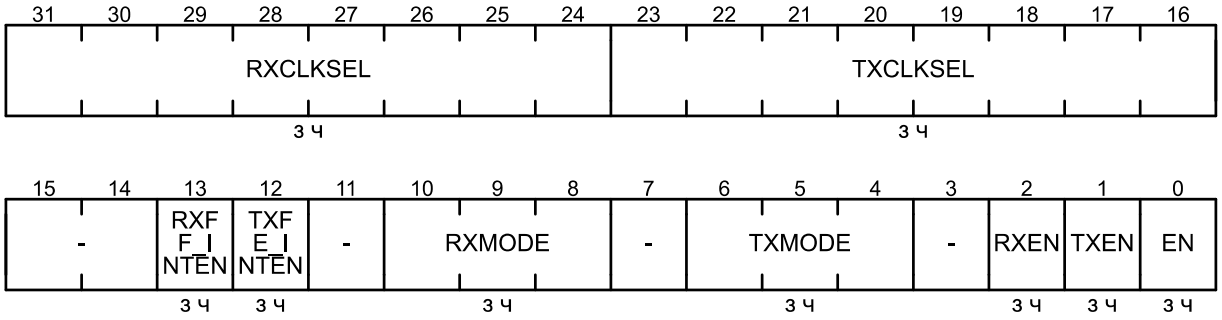
Поле	Биты	Описание
RWMIS	9	Флаг порогового заполнения буфера приемника
TWMIS	8	Флаг порогового опустошения буфера передатчика
RFMIS	7	Флаг заполнения буфера приемника
REMIS	6	Флаг опустошения буфера приемника
TFMIS	5	Флаг заполнения буфера передатчика
TEMIS	4	Флаг опустошения буфера передатчика
TSMIS	2	Флаг останова передачи
TCMIS	1	Флаг завершение передачи
–	31-10, 3, 0	Зарезервировано

A.25 Регистры контроллера интерфейса I2S

Базовый адрес: 5100_5000h Регистры блока I2S0
5100_6000h Регистры блока I2S1

CTRL – регистр управления

Смещение: + 00h
Сброс: 0000_0000h

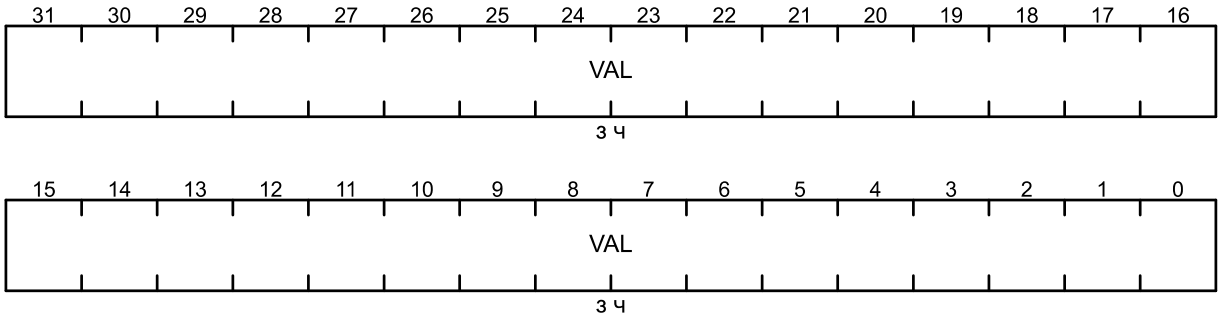


Поле	Биты	Описание	
RXCLKSEL	31-24	Источник тактовых импульсов приемника:	
		0x00	приемник в режиме «ведущий»
		0x01	приемник в режиме «ведомый» от внешних сигналов
		0x02	приемник в режиме «ведомый» от сигналов «ведущего» 1-го блока I2S
		0x03	приемник в режиме «ведомый» от сигналов «ведущего» 2-го блока I2S
		0x04 – 0xFF	Зарезервировано
TXCLKSEL	23-16	Источник тактовых импульсов передатчика:	
		0x00	передатчик в режиме «ведущий»
		0x01	передатчик в режиме «ведомый» от внешних сигналов
		0x02	передатчик в режиме «ведомый» от сигналов «ведущего» 1-го блока I2S
		0x03	передатчик в режиме «ведомый» от сигналов «ведущего» 2-го блока I2S
		0x04 – 0xFF	Зарезервировано
RXFF_INTEN	13	Прерывание по заполнению приемного буфера:	
		1	разрешено
		0	запрещено
TXFE_INTEN	12	Прерывание по отсутствию данных в передающем буфере:	
		1	разрешено
		0	запрещено

Поле	Биты	Описание	
RXMODE	10-8	Режим работы приемника:	
		000	I2S Philips
		001	MSB-justified протокол
		010	LSB-justified протокол
		011	PCM протокол (Short frame sync)
		111	PCM протокол (Long frame sync)
		100 – 110	Зарезервировано
TXMODE	6-4	Режим работы передатчика:	
		000	I2S Philips
		001	MSB-justified протокол
		010	LSB-justified протокол
		011	PCM протокол (Short frame sync)
		111	PCM протокол (Long frame sync)
		100 – 110	Зарезервировано
RXEN	2	Разрешение работы приемника	
		1	включен
		0	выключен
TXEN	1	Разрешение работы передатчика	
		1	включен
		0	выключен
EN	0	Включение блока	
		1	включен
		0	выключен
-	15-14, 11,7, 3	Зарезервировано	

CLKDIV1 – регистр делителя частоты для передатчика

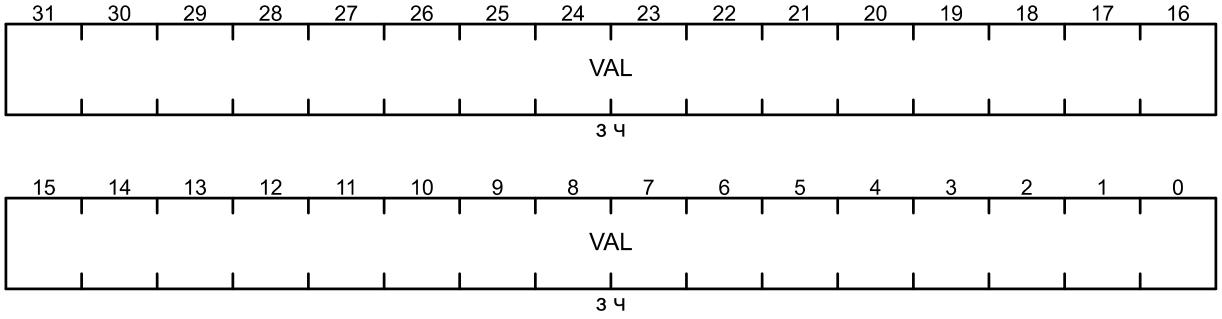
Смещение: + 04h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Коэффициент деления системной частоты для получения частоты BCLK (в режиме ведущего). Получается по формуле: $F_{sys}/(2*F_{sck})$ Возможные значения 1 – $(2^{32}-1)$

CLKDIV2 – регистр делителя частоты для приемника

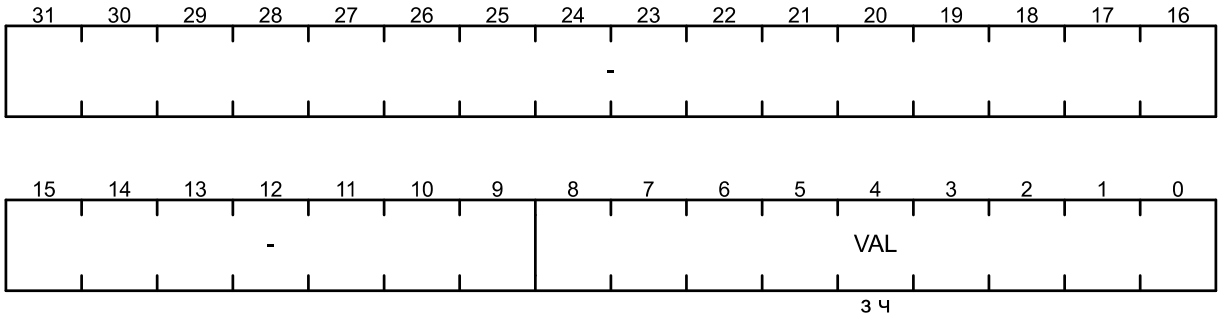
Смещение: + 08h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Коэффициент деления системной частоты для получения частоты BCLK (в режиме ведущего). Получается по формуле: $F_{sys}/(2*F_{sck})$ Возможные значения 1 – (2^{32}-1)

FRMCTRL1 – регистр управления кадром передатчика

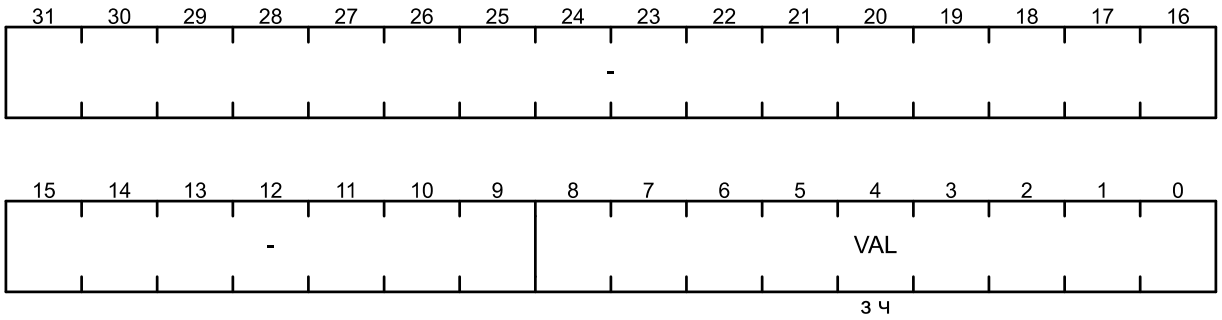
Смещение: + 0Ch
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	8-0	Длина слова на передачу в битах. Возможные значения 1-256. В режиме «ведущего» определяет длину полупериода LRCLK, в режиме «ведомого» - размер пакета
-	31-9	Зарезервировано

FRMCTRL2 – регистр управления кадром приемника

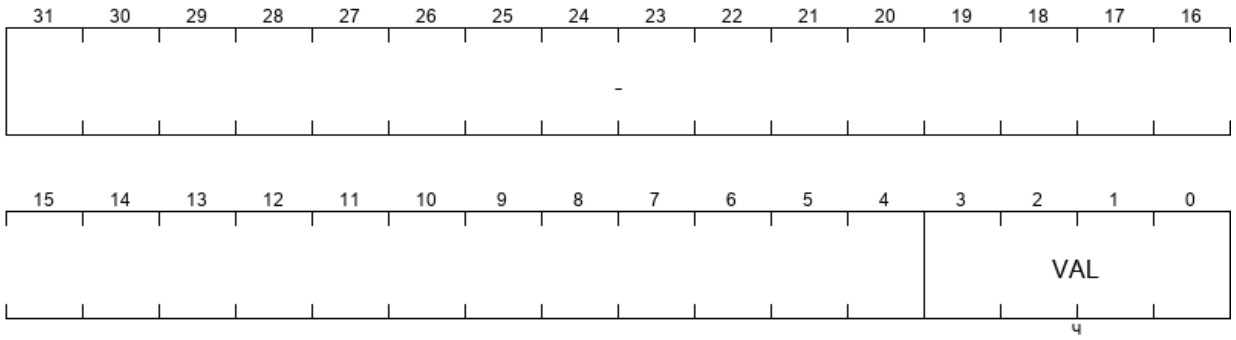
Смещение: + 10h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	8-0	Длина слова на прием в битах. Возможные значения 1-256. В режиме «ведущего» определяет длину полупериода LRCLK, в режиме «ведомого» - размер пакета
-	31-9	Зарезервировано

RECBUFUSED – регистр количества доступных ячеек приемного буфера

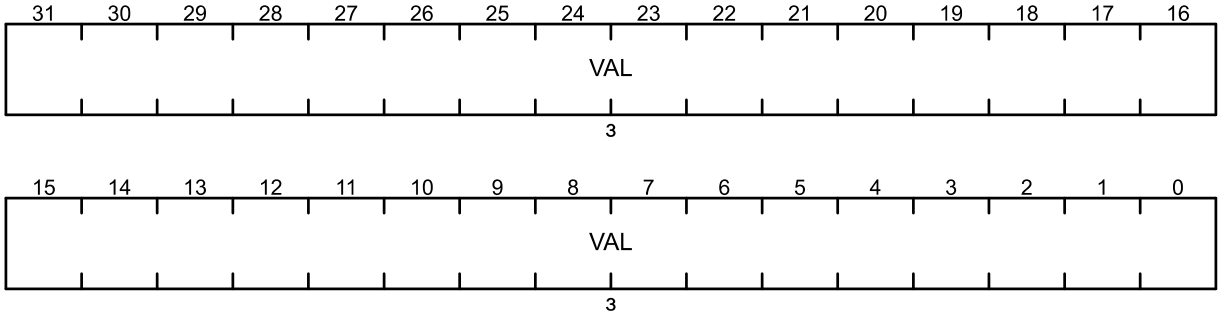
Смещение: + 14h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	3-0	Количество отсчетов, доступных на чтение в приемном буфере. Возможные значения 0 (нет данных) – длина буфера (буфер заполнен)
-	31-4	Зарезервировано

TRANSDATA – регистр данных на передачу

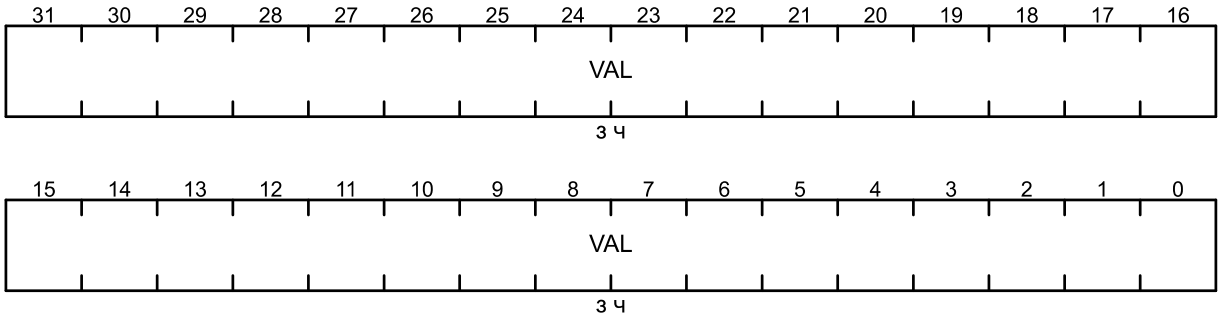
Смещение: + 18h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Буфер, хранящий отсчеты на передачу. Отсчеты для левого и правого каналов идут последовательно. Первый записанный отсчет считается отсчетом левого канала

MCLKDIV – регистр делителя тактовой частоты MCLK

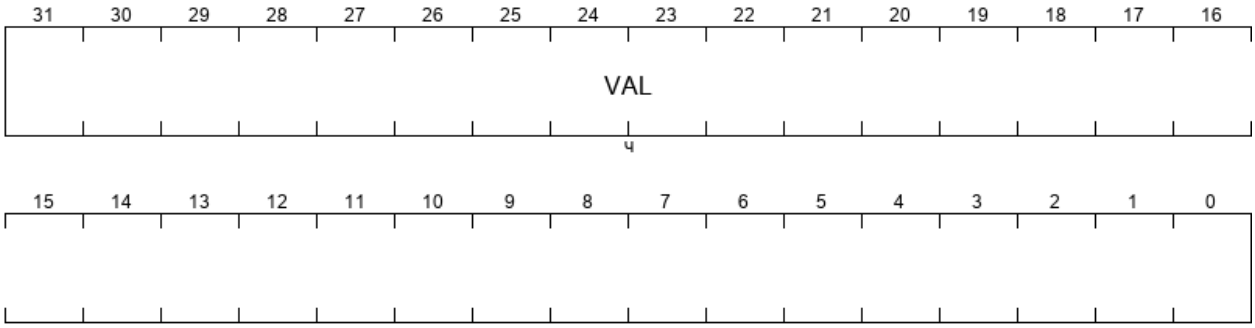
Смещение: + 1Ch
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Коэффициент деления системной частоты для получения частоты MCLK (в режиме ведущего). Получается по формуле: $F_{sys}/(2*F_{sck})$ Возможные значения 1 – $(2^{32}-1)$. Значение 0x00 отключает генерацию.

RECDATA – регистр принятых данных приемника

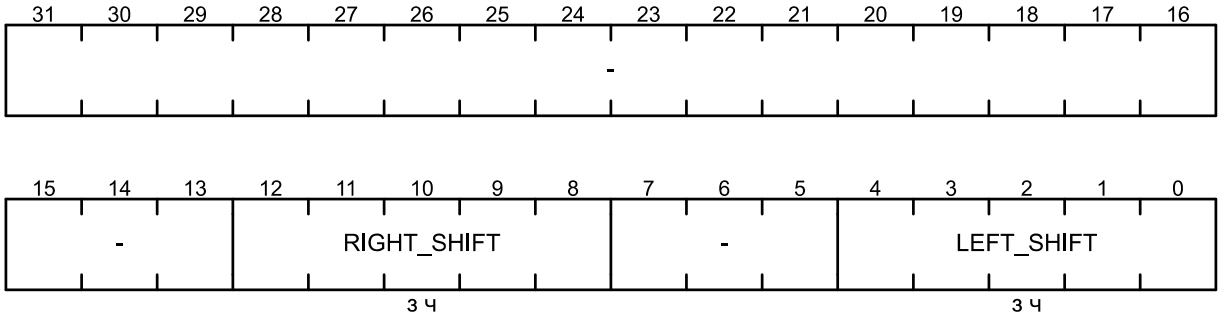
Смещение: + 20h
Сброс: 0000_0000h



Поле	Биты	Описание
VAL	31-0	Буфер FIFO, хранящий принятые отсчеты

SHIFTDATA – регистр сдвига данных

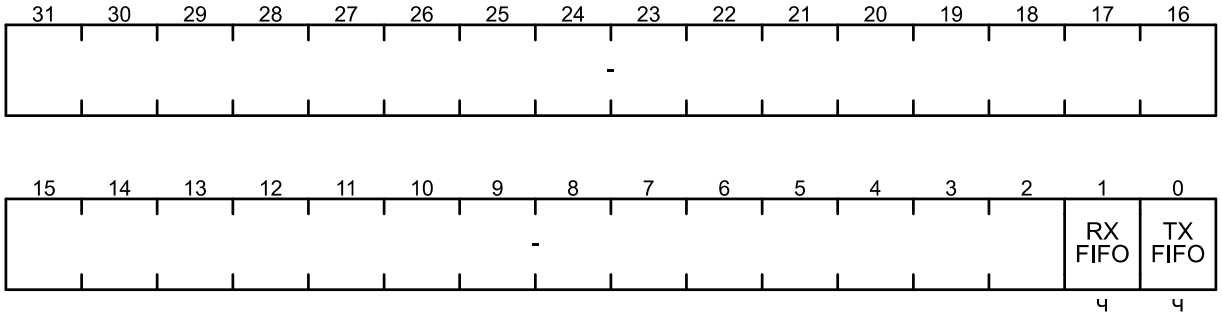
Смещение: + 24h
Сброс: 0000_0000h



Поле	Биты	Описание
RIGHT_SHIFT	12-8	Количество отсчетов для сдвига вправо 32-битных данных, записываемых в приемный буфер в режиме I2S и LSB-justified. Возможные значения 0-31
LEFT_SHIFT	4-0	Количество отсчетов для сдвига влево 32-битных данных, записываемых в передающий буфер в режиме I2S и MSB-justified. Возможные значения 0-31
-	31-13, 7-5	

STATE – регистр состояния FIFO

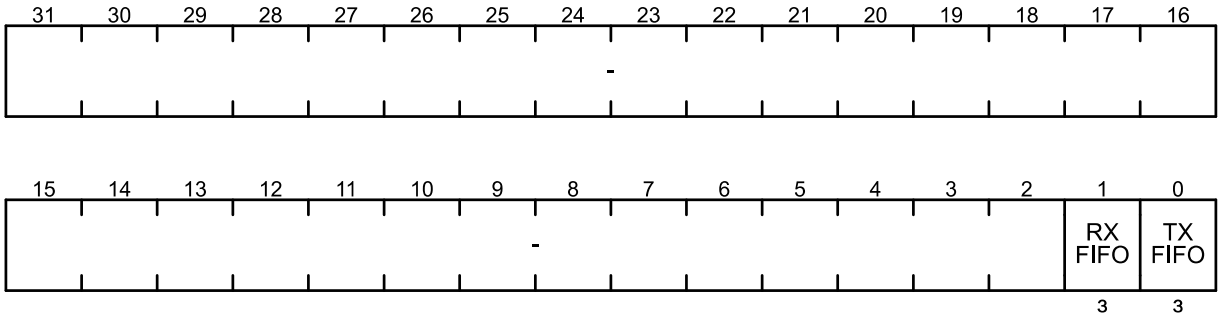
Смещение: + 28h
Сброс: 0000_0000h



Поле	Биты	Описание	
RXFIFO	1	Состояние приемного FIFO	
		1	возникала ситуация, когда во избежание переполнения приемный буфер отбрасывал самый старый пакет
		0	переполнение не возникало
TXFIFO	0	Состояние передающего FIFO	
		1	возникала ситуация, когда передающий буфер был пуст в момент выставления данных в канал
		0	передающий буфер никогда не опустошался
-	31-2	Зарезервировано	

STATECLR – регистр очистки флагов регистра состояний

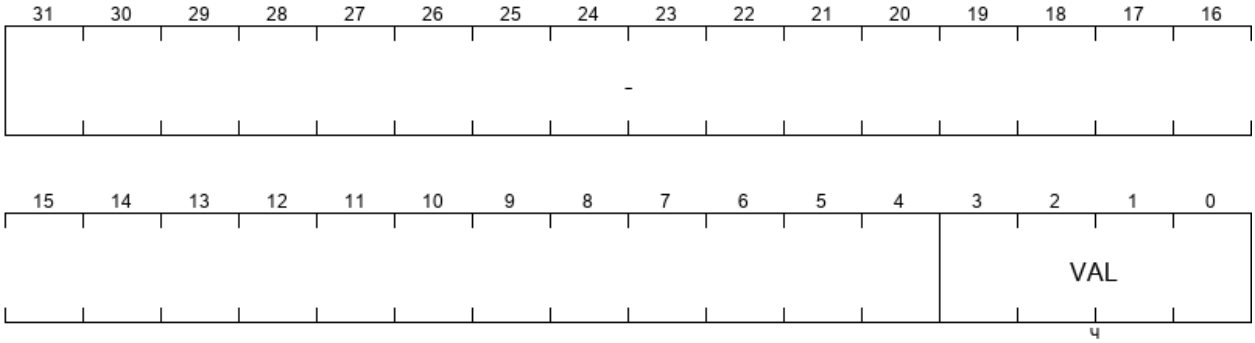
Смещение: + 2Ch
Сброс: 0000_0000h



Поле	Биты	Описание
RXFIFO	1	Очищается флаг RXFIFO регистра STATE
TXFIFO	0	Очищается флаг TXFIFO регистра STATE
-	31-2	Зарезервировано

TRBUFUSED – регистр счетчика заполненных ячеек буфера передатчика

Смещение: + 30h
Сброс: 0000_0000h



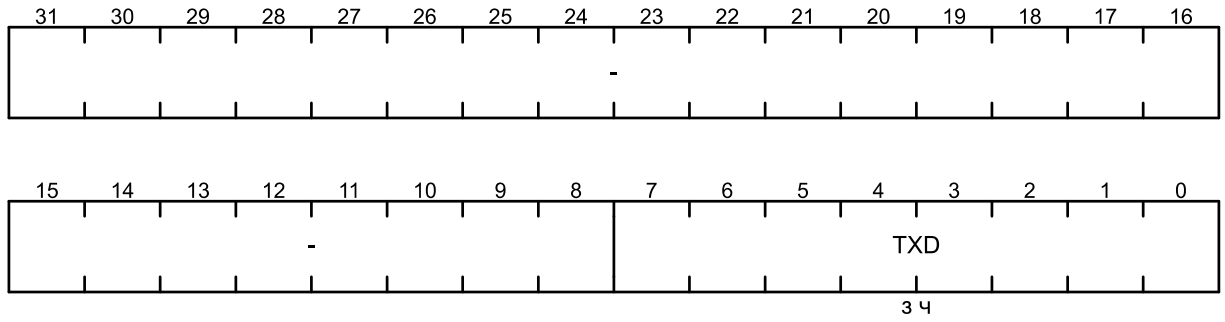
Поле	Биты	Описание
VAL	3-0	Количество заполненных ячеек в передающем буфере. Возможные значения от 0 (нет данных) – длина буфера (буфер заполнен)
-	31-4	Зарезервировано

A.26 Регистры контроллера интерфейса LIN

Базовый адрес:	5200_0000h	Регистры блока LIN0
	5200_1000h	Регистры блока LIN1
	5200_2000h	Регистры блока LIN2
	5200_3000h	Регистры блока LIN3
	5200_4000h	Регистры блока LIN4
	5200_5000h	Регистры блока LIN5
	5200_6000h	Регистры блока LIN6
	5200_7000h	Регистры блока LIN7

DTX – регистр передаваемых данных

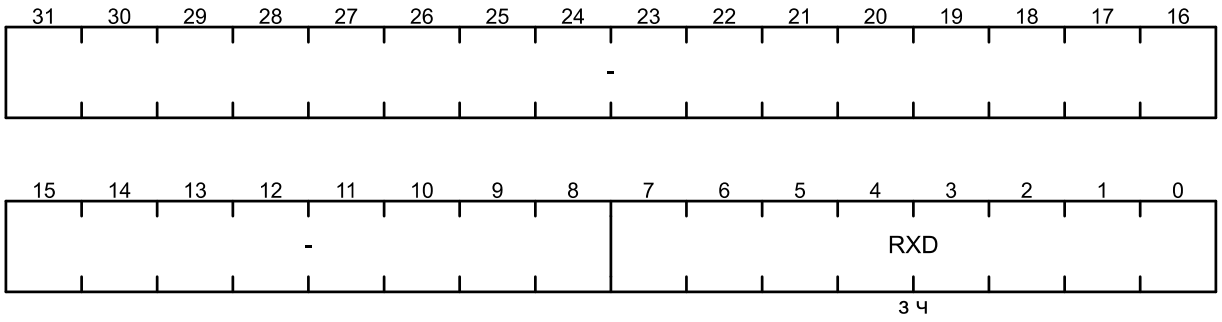
Смещение: 00h + (4*N)h, где N от 0 до 7
Сброс: 00h



Поле	Биты	Описание
TXD	7-0	Данные, отправляемые в следующем пакете. В зависимости от настроек протокола может заполняться от 1 до 8 регистров. Заполнение остальной части не обязательно. Количество отправляемых данных определяется в PID или полем UDL регистра CR
–	31-8	Зарезервировано

DRX – регистр принятых данных

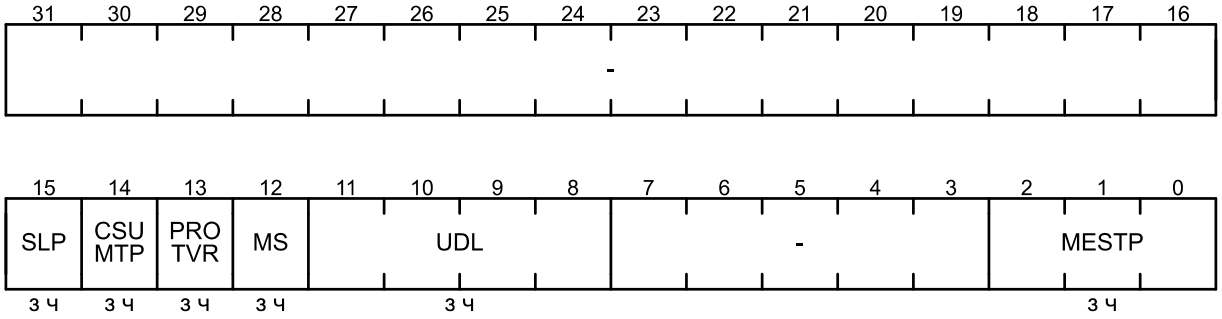
Смещение: 20h + (4*N)h, где N от 0 до 7
Сброс: 00h



Поле	Биты	Описание
RXD	7-0	Содержит последние полученные данные. Количество перезаписанных байт определяется PID принятого пакета, типом протокола и полем UDL регистра CR
–	31-8	Зарезервировано

CR – регистр управления

Смещение: + 40h
Сброс: 00h



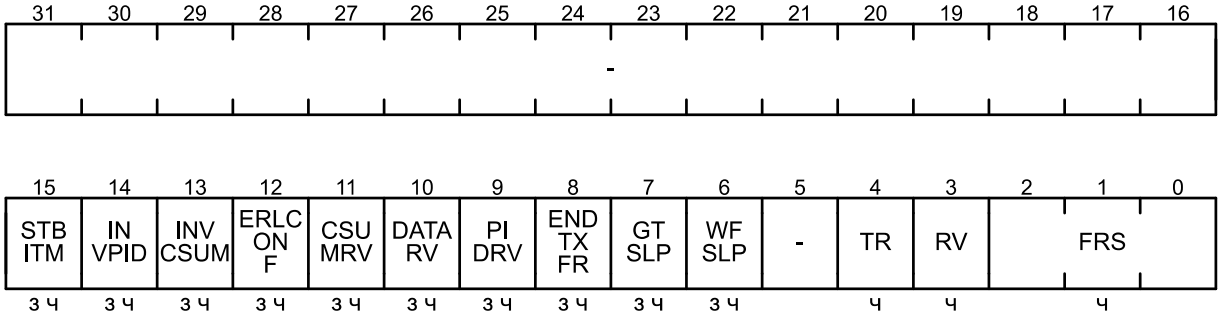
Поле	Биты	Описание
SLP	15	Бит состояния сна 0 – устройство активно 1 – устройство в режиме сна
CSUMTP	14	Бит типа контрольной суммы 0 – обычная контрольная сумма 1 – расширенная контрольная сумма
PROTVR	13	Бит типа протокола 0 – соответствие версии 1.X 1 – соответствие версии 2.X
MS	12	Бит типа устройства ведущий (master) / ведомый (slave) 0 – ведомый (slave); 1 – ведущий (master)

Поле	Биты	Описание	
UDL	11-8	Количество полей данных в одном кадре сообщения	
MESTP	2-0	Тип кадра сообщения	
		001b	Состояние IDLE (только в режиме transmit)
		010b	Состояние HEADER
		011b	Состояние HEADER+RESPONCE
		100b	Состояние RESPONSE RECEIVE
		101b	Состояние RESPONSE TRANSMIT
		111b	Послать сигнал WAKE
-	31-16, 7-3	Зарезервировано	

SR – регистр статуса

Смещение: + 44h

Сброс: 00h

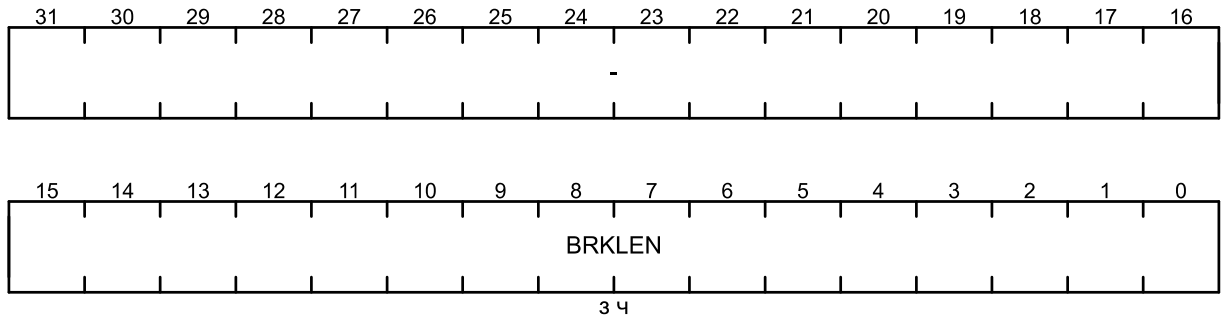


Поле	Биты	Описание
STBITM	15	Бит отсутствия стопового символа. Устанавливается при отсутствии в поле стопового бита, при разрешении данного прерывания в регистре IM. Сбрасывается вручную
INVPID	14	Бит ошибки PID. Устанавливается при ошибке четности в PID, при разрешении данного прерывания в регистре IM. Сбрасывается вручную
INVCSUM	13	Бит ошибки контрольной суммы. Устанавливается, когда принятая сумма, отличается от суммы, рассчитанной на основе принятых полей, при разрешении данного прерывания в регистре IM. Сбрасывается вручную
ERLCONF	12	Бит конфликта линии. Устанавливается, если при передаче рецессивного бита, линия находится в доминантном состоянии, при разрешении данного прерывания в регистре IM. Сбрасывается вручную
CSUMRV	11	Бит принятия контрольной суммы. Устанавливается при принятии поля контрольной суммы, при разрешении данного прерывания в регистре IM. Сбрасывается вручную
DATARV	10	Бит принятия данных. Устанавливается при принятии поля данных, при разрешении данного прерывания в регистре IM. Сбрасывается вручную

Поле	Биты	Описание	
PIDRV	9	Бит принятия PID. Устанавливается при принятии поля PID, при разрешении данного прерывания в регистре IM. Сбрасывается вручную	
ENDTXFR	8	Бит конца передачи. Устанавливается при отправке последнего символа текущей посылки, при разрешении данного прерывания в регистре IM. Сбрасывается вручную	
GTSLP	7	Бит устанавливается, если принятый кадр сообщения является сигналом ко сну, при разрешении данного прерывания в регистре IM. Сбрасывается вручную	
WFSLP	6	Бит устанавливается, когда во время сна принят сигнал пробуждения, при разрешении данного прерывания в регистре IM. Сбрасывается вручную	
TR	4	Бит активности передающей части интерфейса 0 – передатчик не активен 1 – идет передача	
RV	3	Бит активности приемной части интерфейса 0 – приемник не активен 1 – идет прием	
FRS	2-0	Текущее состояние устройства	
		001b	Состояние IDLE (только в режиме transmit)
		010b	Состояние BREAK
		011b	Состояние PID
		100b	Состояние DATA
		101b	Состояние CSUM
		111b	Состояние посылки сигнала WAKE
-	31-16, 5	Зарезервировано	

BL – регистр настройки длины поля BREAK

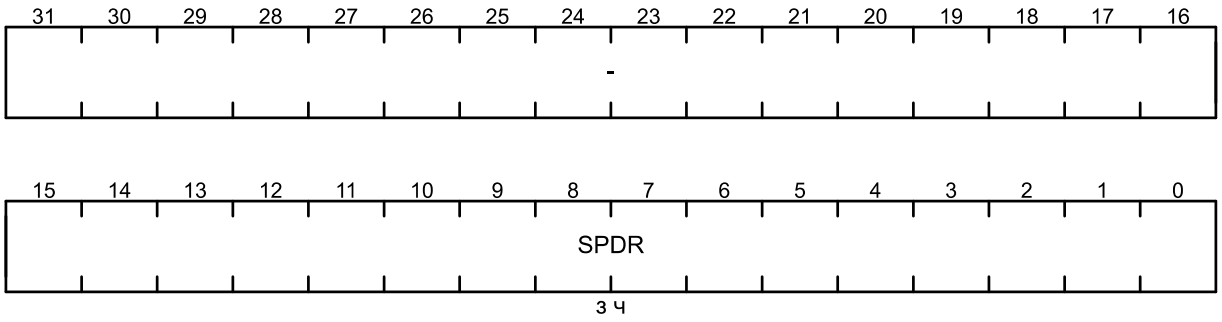
Смещение: + 48h
Сброс: 00h



Поле	Биты	Описание
BRKLEN	15-0	Устройство в сети является подчиненным. Регистр содержит минимальную длину поля разрыва синхронизации, выраженную в периодах тактового сигнала
–	31-16	Зарезервировано

SRR – регистр настройки скорости передачи

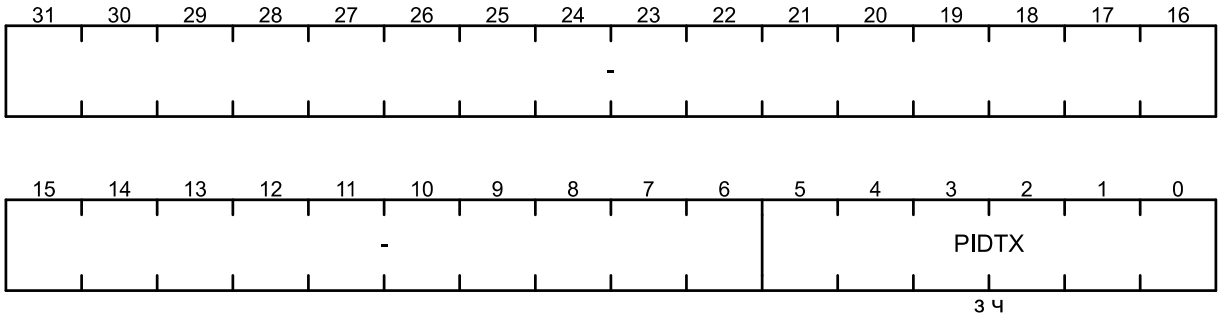
Смещение: + 4Ch
Сброс: 00h



Поле	Биты	Описание
SPDR	15-0	Определение скорости передачи главного устройства. Заполняется при ведущем устройстве в сети. Для подчиненных устройств значение регистра игнорируется
–	31-16	Зарезервировано

PTX – регистр отправляемого PID

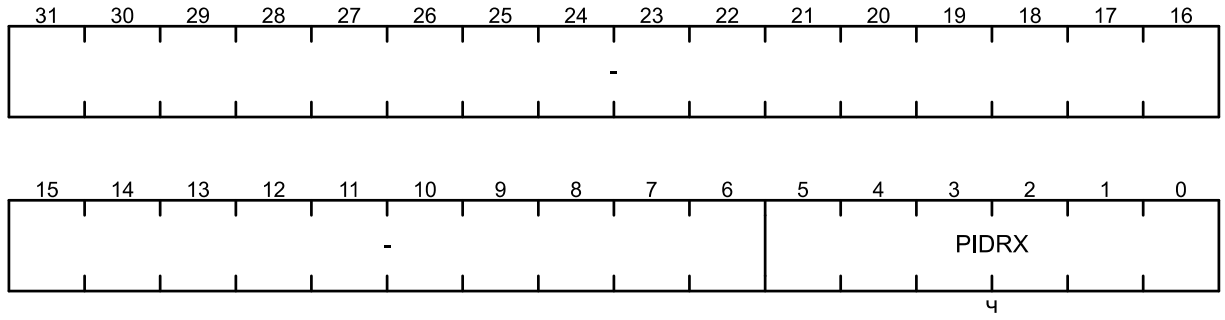
Смещение: + 50h
Сброс: 00h



Поле	Биты	Описание
PIDTX	5-0	В данный регистр загружается PID, который будет отправлен в следующем кадре данных. Загружаемое значение не зависит от версии протокола и типа контрольной суммы. При отправке два неиспользованных разряда будут автоматически заменены на биты четности
–	31-6	Зарезервировано

PRX – регистр принятого PID

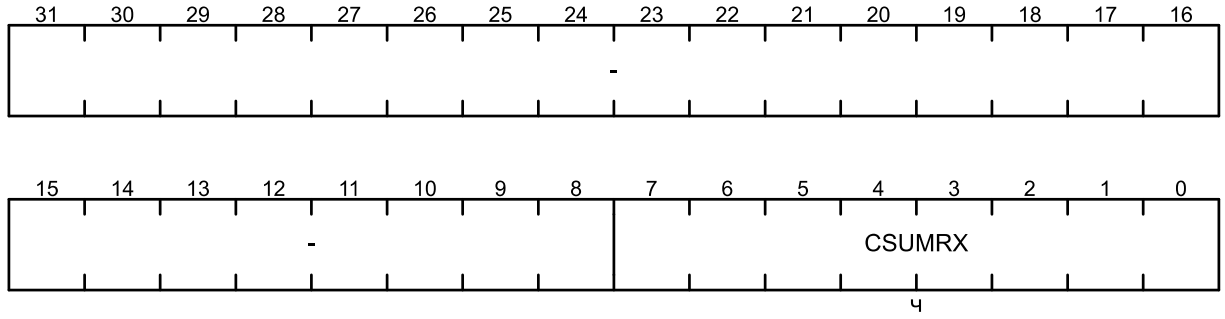
Смещение: + 50h
Сброс: 00h



Поле	Биты	Описание
PIDRX	5-0	Содержит значение принятого PID. Обновление происходит одновременно с выработкой сигнала PID received регистра SR1. Биты четности в данный регистр не попадают, в случае нарушения четности будет установлен бит Invalid PID регистра SR1
–	31-6	Зарезервировано

CSRX – регистр принятой контрольной суммы

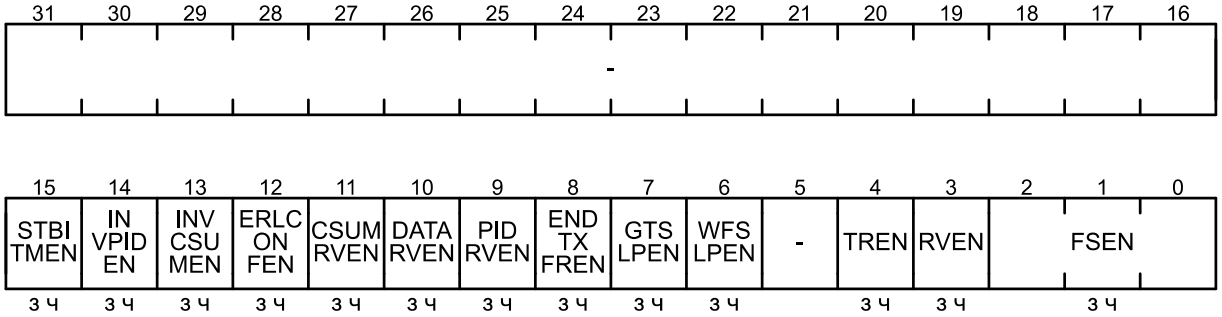
Смещение: + 58h
Сброс: 00h



Поле	Биты	Описание
CSUMRX	7-0	Содержит значение принятой контрольной суммы. Обновление происходит одновременно с выработкой сигнала CSumm received регистра SR. Соответствие принятой контрольной суммы и вычисленной на основе принятых данных определяет бит Invalid CSumm регистра SR
–	31-8	Зарезервировано

IM – регистр маски прерываний

Смещение: + 5Ch
Сброс: 00h

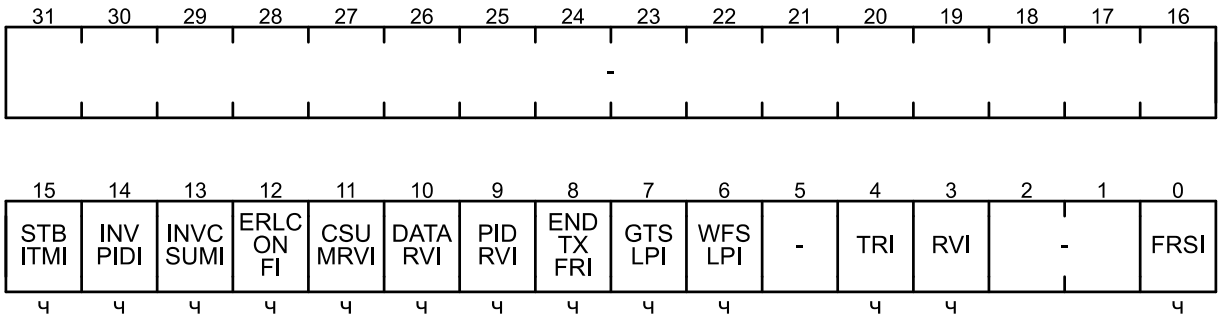


Поле	Биты	Описание
STBITMEN	15	Разрешение прерывания по отсутствию стопового бита
INVPIDEN	14	Разрешение прерывания по приему PID с нарушенной четностью
INVCSUMEN	13	Разрешение прерывания по приему неверной контрольной суммы
ERLCONFEN	12	Разрешает прерывание при возникновении конфликта линии
CSUMRVEN	11	Разрешает прерывание по приему поля контрольной суммы
DATARVEN	10	Разрешает прерывание по приему поля данных
PIDRVEN	9	Разрешает прерывание по приему PID поля
ENDTXFREN	8	Разрешает прерывание при завершении передачи очередного кадра сообщения
GTSLPEN	7	Разрешает прерывание по получению сигнала сна
WFSLPEN	6	Разрешает прерывание по получению сигнала пробуждения
TREN	4	Разрешает прерывание по активизации передающей части устройства
RVEN	3	Разрешает прерывание по активизации приемной части устройства
FSEN	2-0	Разрешает прерывание при переходе устройства на работу с установленным полем кадра сообщения
–	31-16, 5	Зарезервировано

Примечание – Установка/сброс бит формирует маску. По умолчанию все биты сброшены, и установка флагов запрещена. Необходимо установить биты тех событий, прерывания от которых нужно разрешить.

MIS – регистр статуса маскированных прерываний

Смещение: + 60h
Сброс: 00h



Поле	Биты	Описание
STBITMI	15	Бит устанавливается по отсутствию стопового бита и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
INVPIDI	14	Бит устанавливается по приему PID с нарушенной четностью и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
INVCSUMI	13	Бит устанавливается по приему неверной контрольной суммы и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
ERLCONFI	12	Бит устанавливается при возникновении конфликта линии и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
CSUMRVI	11	Бит устанавливается по приему поля контрольной суммы и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
DATARVI	10	Бит устанавливается по приему поля данных и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
PIDRVI	9	Бит устанавливается по приему PID поля и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
ENDTXFRI	8	Бит устанавливается при завершении передачи очередного кадра сообщения и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1

Поле	Биты	Описание
GTSLPI	7	Бит устанавливается по получению сигнала сна и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
WFSLPI	6	Бит устанавливается по получению сигнала пробуждения и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
TRI	4	Бит устанавливается при переходе устройства в режим передачи пробуждения и установленного соответствующего бита маски прерываний регистра IM. Сбрасывается записью 1
RVI	3	Бит устанавливается при переходе устройства в режим пробуждения и установленного соответствующего бита маски прерываний регистра IM Сбрасывается записью 1
FRSI	0	Бит устанавливается при переходе устройства на работу с полем кадра сообщения, соответствующему установленной маски прерываний регистра IM (биты [2:0]). Сбрасывается записью 1
–	31-16, 5, 2-1	Зарезервировано

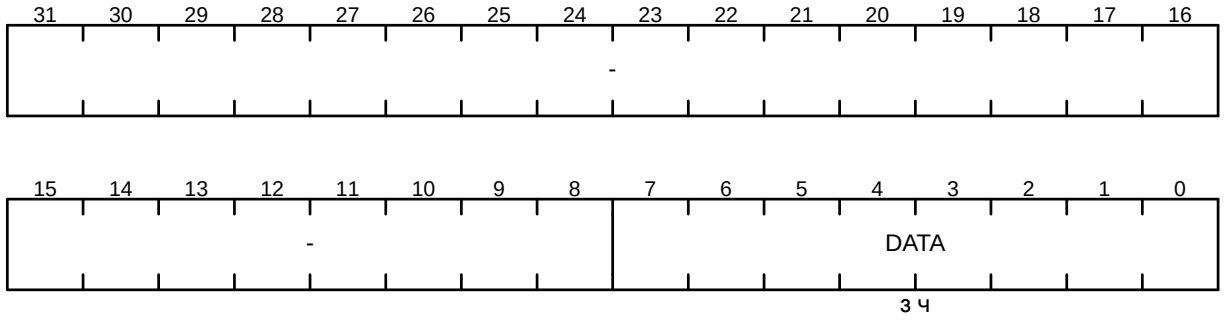
A.27 Регистры контроллера интерфейса I2C

Базовый адрес:	5201_8000h	Регистры блока I2C0
	5201_9000h	Регистры блока I2C1
	5201_A000h	Регистры блока I2C2
	5201_B000h	Регистры блока I2C3
Смещение:	+ 028h (EADDR)	Массив регистров дополнительного адреса

Примечание – n – номер регистра 0 – 7.

SDA – регистр данных

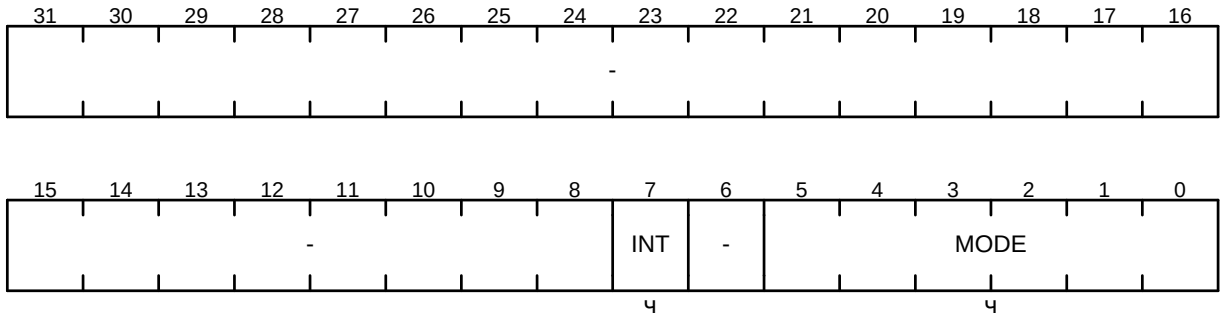
Смещение: + 00h
Сброс: 0xxh



Поле	Биты	Описание
DATA	7-0	Поле данных
–	31-8	Зарезервировано

ST – регистр состояния

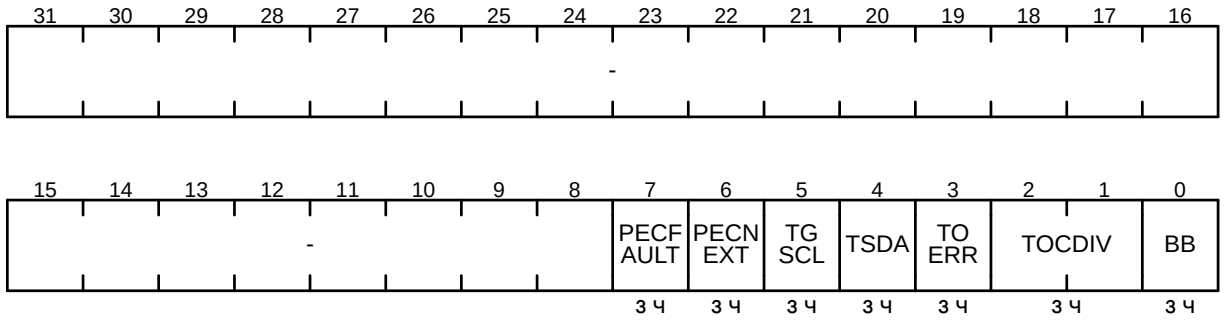
Смещение: + 04h
Сброс: 0h



Поле	Биты	Описание
INT	7	Флаг прерывания. Устанавливается после девятого такта сигнала SCL (когда SCL = 0) в любое запрограммированное время. Условия выставления флага INT: - во время приема/передачи, как в режиме мастера, так и в режиме ведомого; - при совпадении адреса (адреса ведомого, адреса отклика или адреса общего вызова) содержимое регистра SDA должно контролироваться программно для определения типа полученного адреса; - после успешного формирования стартового состояния или состояния повторного старта; - в случае неквитирования переданной информации; - при потере арбитража во время передачи последнего бита; - при обнаружении валидного состояния останова или состояния повторного старта; - при обнаружении ошибки на шине. Пока установлен флаг INT, на линии SCL удерживается низкий уровень сигнала. Флаг INT может быть сброшен установкой бита CLRST в регистре CTL0 или выключением модуля I2C (обнуление бита ENABLE в регистре CTL1). Условия выставления флага INT (не влияющие на уровень сигнала на линии SCL): - простой на линии SCL; - состояние останова в режиме ведомого (MODE = 1Ch); - потеря арбитража, вследствие чего ведомый переключился в безадресный режим (MODE = 03h или MODE = 23h); - неквитированная передача байта данных (MODE = 17h)
MODE	5-0	Код состояния. Возникновение того или иного состояния в течение функционирования модуля I2C сопровождается записью соответствующего кода в поле MODE
–	31-8, 6	Зарезервировано

CST – регистр управления и статуса

Смещение: + 08h
Сброс: 0h

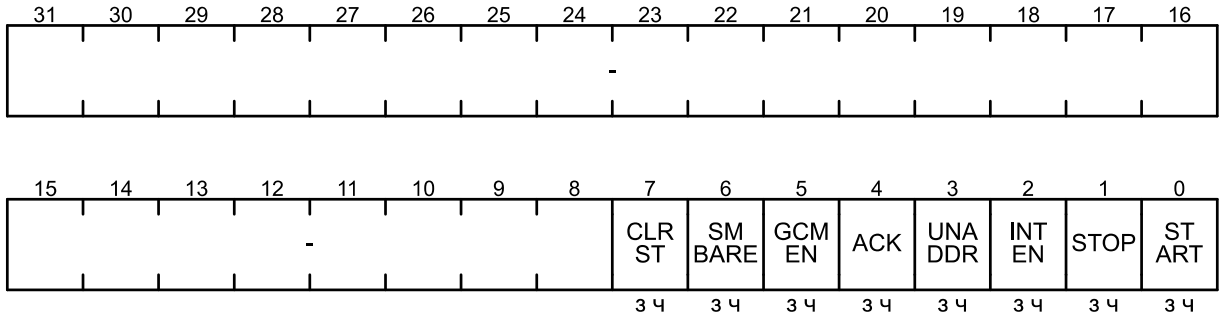


Поле	Биты	Описание
PECFAULT	7	Флаг ошибки. Устанавливается в случае, если после расчета контрольной суммы для пакета данных и сравнения ее с полученной суммой, значение во внутреннем регистре ошибок не нулевое
PECNEXT	6	Бит управления отправкой байта контрольной суммы. Установка бита указывает на то, что следующий передаваемый байт будет байтом CRC (байт контрольной суммы). Реакция на установку бита PECNEXT зависит от режима работы. В режиме мастера передатчика установка бита PECNEXT вызовет загрузку результата вычисления CRC в регистр SDA. После сброса флага INT начнется передача байта CRC. В режиме приемника установка этого бита будет указывать логике управления на то, что следующий байт, который будет принят, будет байтом CRC. В режиме ведомого приемника модуль I2C автоматически будет квитировать или не квитировать прием байта CRC, в зависимости от того, будет ли выявлена ошибка пакета данных или нет. В режиме мастера приемника по окончании приема байта CRC будет отправлено значение бита ACK регистра CTL0
TGSCL	5	Бит переключения SCL. Бит позволяет переключать вывод I2C_SCL во время восстановления после ошибки. Когда на выводе I2C_SDA – низкий уровень сигнала, запись «1» в бит TGSCL переключит вывод SCL на один такт. Когда на выводе I2C_SDA высокий уровень сигнала, запись «1» в бит TGSCL игнорируется. Бит очищается аппаратно по окончании такта
TSDA	4	Бит тестирования SDA. Содержит текущее значение SDA. Этот бит можно использовать для отслеживания окончания процесса восстановления после ошибки, в течение которого ведомый постоянно поддерживает низкий уровень сигнала на выводе SDA
TOERR	3	Флаг ошибки простоя на шине. Если TOERR = 1b, это указывает на то, что на линии SCL был обнаружен простой. Флаг TOERR выставляется по обнулению основного счетчика времени простоя и может быть сброшен записью «1» в бит CLRST регистра CTL0

Поле	Биты	Описание	
TOCDIV	2-1	Поле коэффициента делителя. Устанавливает коэффициент деления системного тактового сигнала, подаваемого на предделитель времени простоя линии SCL	
		00	Тактовый сигнал отсутствует
		01	Деление на 4
		10	Деление на 8
		11	Деление на 16
BB	0	Флаг занятости шины. Если BB = 1b, это указывает на то, что шина занята. Устанавливается, как только шина переходит в активное состояние (одновременное появление низкого уровня сигнала на выводах I2C_SDA и I2C_SCL или хотя бы на одном из них) или в стартовое состояние. Сбрасывается при выключении интерфейса I2C либо при обнаружении состояния останова	
–	31-8	Зарезервировано	

CTL0 – регистр управления 0

Смещение: + 0Ch
Сброс: 0h

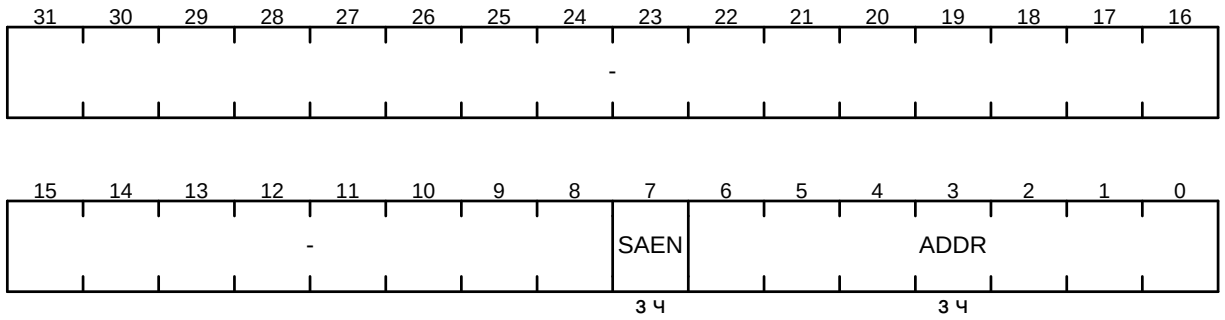


Поле	Биты	Описание	
CLRST	7	Бит сброса флага прерывания INT	
		Чтение	Возвращает ноль
		Запись нуля	Не выполняется
		Запись единицы	Сбрасывает флаг INT в регистре ST
SMBARE	6	Бит управления реакцией на получение адреса отклика	
		0	Полученный адрес не проверяется на совпадение с адресом отклика
		1	Адрес, полученный сразу после старта, проверяется на совпадение с адресом отклика (0001100b)
		Бит очищается при выходе ведомого из режима IDLE	
GCMEN	5	Бит управления реакцией на получение адреса общего вызова	
		0	Полученный адрес не проверяется на совпадение с адресом общего вызова
		1	Адрес, полученный сразу после старта, проверяется на совпадение с адресом общего вызова (000_0000b)
		Бит очищается при выходе ведомого из режима IDLE	

Поле	Биты	Описание
ACK	4	Бит квитирования приема. В режиме передатчика не используется. В режиме приемника (мастера/ведомого) содержит значение, которое передается в течение цикла отклика на запрос передатчика подтвердить прием. Передача нуля по окончании передачи байта (квитирование) означает, что данные успешно получены. Передача единицы (неквитирование) означает, что приемник не может продолжать работу по каким-либо причинам. Бит ACK очищается аппаратно по окончании цикла отклика
UNADDR	3	Бит разрешения ответа на любой полученный адрес в режиме ведомого
INTEN	2	Бит разрешения прерывания
		0 Запрещено
		1 Разрешено
STOP	1	Бит останова. В режиме мастера установка бита STOP генерирует состояние останова, которое завершает или прерывает текущую передачу. После прекращения передачи бит STOP очищается аппаратно
START	0	Бит старта. Этот бит устанавливается, когда требуется сформировать стартовое состояние на шине. Бит START очищается аппаратно по окончании цикла стартового состояния, а также при обнаружении ошибки на шине (состояние с кодом 1Fh)
–	31-8	Зарезервировано

ADDR – регистр собственного адреса

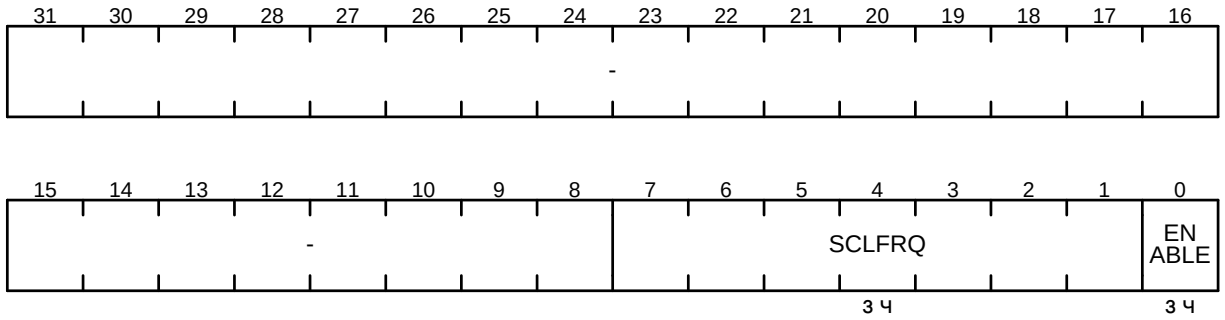
Смещение: + 10h
Сброс: 0h



Поле	Биты	Описание
SAEN	7	Бит разрешения распознавания адреса
		0 Безадресный режим
		1 Включена функция распознавания принятого адреса
ADDR	6–0	Поле собственного 7-битного адреса. При работе в режиме ведомого первые 7 бит, принятые после стартового состояния, сравниваются со значением ADDR. Если обнаружено совпадение и установлен бит SAEN, ведомый переходит в режим приемника или передатчика (в зависимости от состояния бита направления R/W#)
–	31-8	Зарезервировано

CTL1 – регистр управления 1

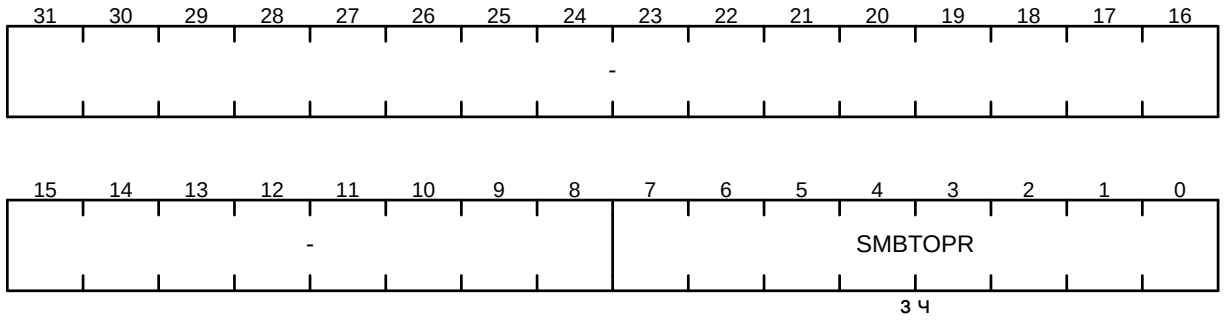
Смещение: + 14h
Сброс: 0h



Поле	Биты	Описание
SCLFRQ	7-1	Младшие разряды поля выбора частоты f_{SCL} сигнала на выводе I2Cx_SCL в режиме мастера. Длительности высокого T_{SCLH} и низкого T_{SCLL} уровней сигнала SCL зависят от тактовой частоты f_{PCLK} модуля I2C и рассчитываются по формуле: $T_{SCLH} = T_{SCLL} = 2 \times SCLFRQ \times (1/f_{PCLK})$. (A.27.1) Таким образом, частота сигнала на выводе I2Cx_SCL равна $f_{SCL} = 1/(T_{SCLH} + T_{SCLL})$. (A.27.2) В поле SCLFRQ можно записать любое значение в диапазоне от 4h до 7FFFh (старшие разряды находятся в регистре CLT3). При попытке записи любого значения меньше 4h, оно будет записано со смещением 4h. Например, при записи числа 2h, к нему будет аппаратно добавлено смещение 4h и, в итоге, в поле SCLFRQ окажется значение 6h
ENABLE	0	Бит включения модуля I2C
		0 – Модуль выключен. Тактирование не осуществляется. Регистры CTL0, ST, CST сброшены
		1 – Модуль включен
–	31-8	Зарезервировано

TOPR – регистр загрузки предделителя

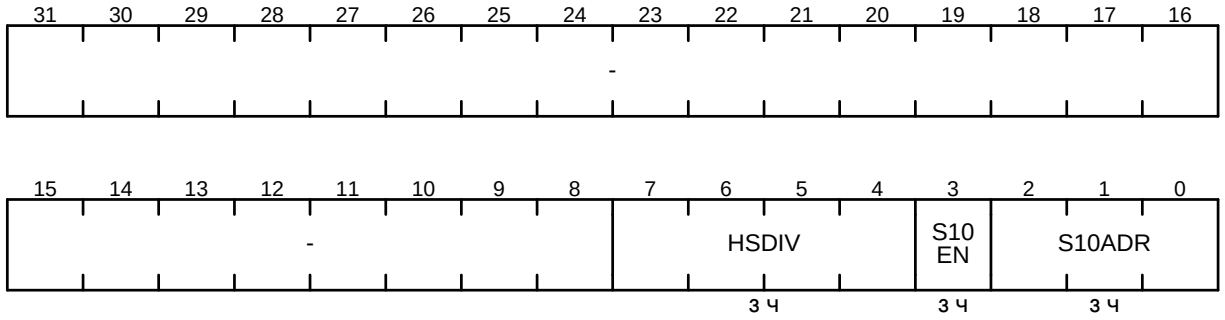
Смещение: + 18h
Сброс: 0h



Поле	Биты	Описание
SMBTOPR	7-0	Поле значения перезагрузки предделителя
–	31-8	Зарезервировано

CTL2 – регистр управления 2

Смещение: + 1Ch
Сброс: 0h

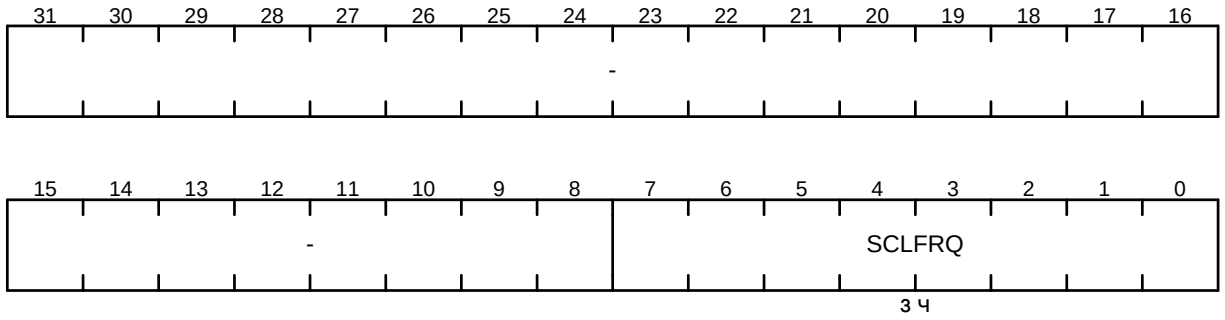


Поле	Биты	Описание
HSDIV	7-4	Младшие разряды поля выбора частоты f_{SCL} сигнала на выводе I2C_SCL в режиме HS мастера. Длительности высокого(T_{HSCLH}) и низкого(T_{HSCLL}) уровней сигнала на выводе I2C_SCL зависят от тактовой частоты f_{PCLK} модуля I2C и рассчитываются по формулам $T_{HSCLH} = HSDIV \times (1/ f_{PCLK})$, (A.27.3) $T_{HSCLL} = 2 \times HSDIV \times (1/ f_{PCLK})$. (A.27.4) Таким образом, частота сигнала на выводе I2C_SCL равна $f_{SCL} = 1/(T_{HSCLH} + T_{HSCLL})$. (A.27.5) В поле HSDIV можно записать любое значение в диапазоне от 2h до 1000h (старшие разряды находятся в регистре CTL4). При попытке записи любого значения меньше 2 в поле HSDIV, оно будет записано со смещением 2. Например, при записи числа 1 к нему будет аппаратно добавлено смещение 2 и, в итоге, в поле HSDIV окажется значение 3

Поле	Биты	Описание	
S10EN	3	Бит разрешения 10-битной адресации ведомого	
		0	Запрещено
		1	Разрешено при условии, что установлен бит SAEN в регистре ADDR
S10ADR	2-0	Поле старших битов 10-битного адреса ведомого. Поле содержит старшие три разряда адреса ведомого при 10-битной адресации. Первый принятый байт адреса сравнивается со значением [11110b, S10ADR[2:1]], второй байт адреса – со значением [S10ADR[0], ADDR]	
–	31-8	Зарезервировано	

CTL3 – регистр управления 3

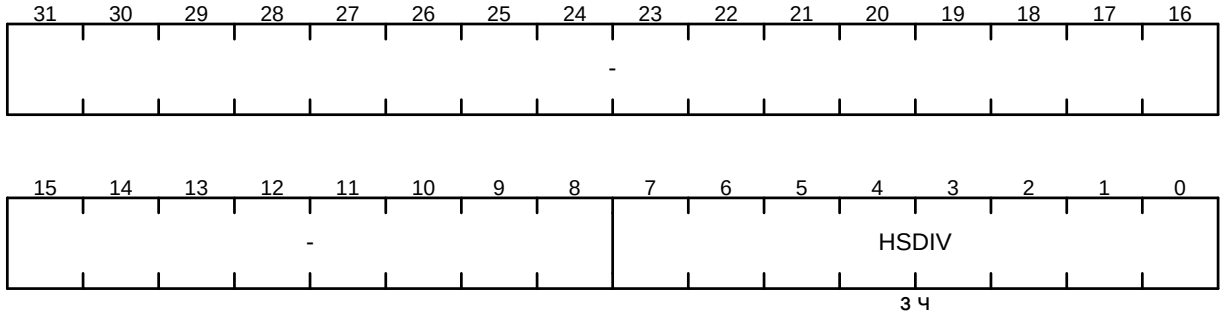
Смещение: + 20h
Сброс: 0h



Поле	Биты	Описание
SCLFRQ	7-0	Старшие разряды делителя SCLFRQ. См. основное описание поля в регистре CTL1
–	31-8	Зарезервировано

CTL4 – регистр управления 4

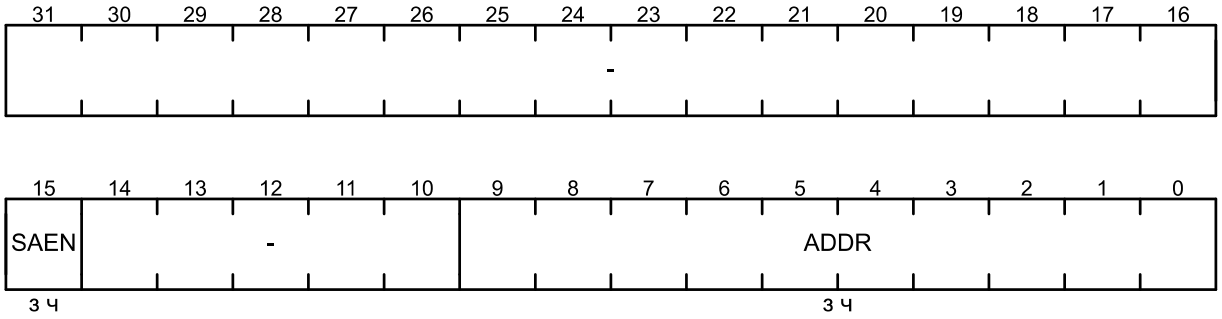
Смещение: + 24h
Сброс: 0h



Поле	Биты	Описание
HSDIV	7-0	Старшие разряды делителя HSDIV. См. основное описание поля в регистре CTL2
–	31-8	Зарезервировано

EADDR – массив регистров дополнительного адреса

Смещение: EADDR + (4*n)h
Сброс: 0h



Поле	Биты	Описание
SAEN	15	Бит включения распознавания дополнительного адреса
ADDR	9-0	Дополнительный адрес
–	31-16, 14-10	Зарезервировано

Примечание – n - номер регистра дополнительного адреса от 0 до 7.

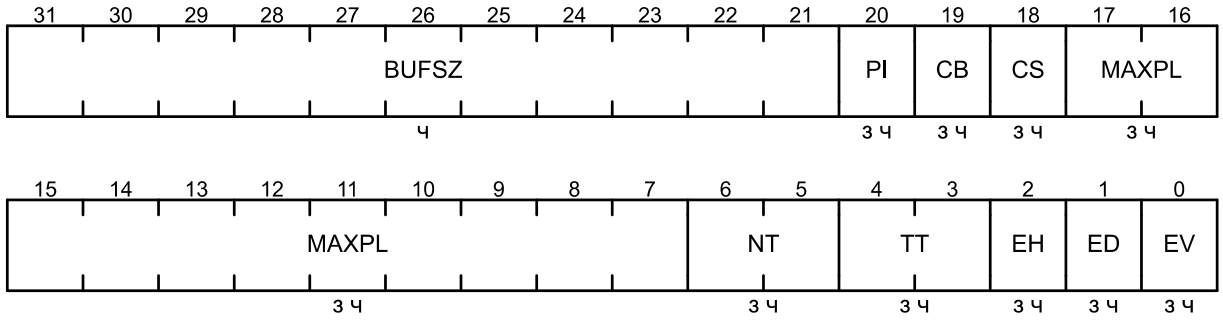
A.28 Регистры контроллера USB 2.0 DC

Базовый адрес:	4108_0000h	Регистры блока USB DC0
	4109_0000h	Регистры блока USB DC1
Смещение:	+ 000h (EPO0)	Регистры конечной точки 0 типа OUT
	+ 010h (EPO1)	Регистры конечной точки 1 типа OUT
	+ 020h (EPO2)	Регистры конечной точки 2 типа OUT
	+ 030h (EPO3)	Регистры конечной точки 3 типа OUT
	+ 100h (EPI0)	Регистры конечной точки 0 типа IN
	+ 110h (EPI1)	Регистры конечной точки 1 типа IN
	+ 120h (EPI2)	Регистры конечной точки 2 типа IN
	+ 130h (EPI3)	Регистры конечной точки 3 типа IN
Мнемоника:	EPOn;	
	EPIn;	

Примечание – n – номер конечной точки от 0 до 3

CTRL – управляющий регистр конечной точки

Смещение: EPOn/EPIn + 00h
Сброс: 0h

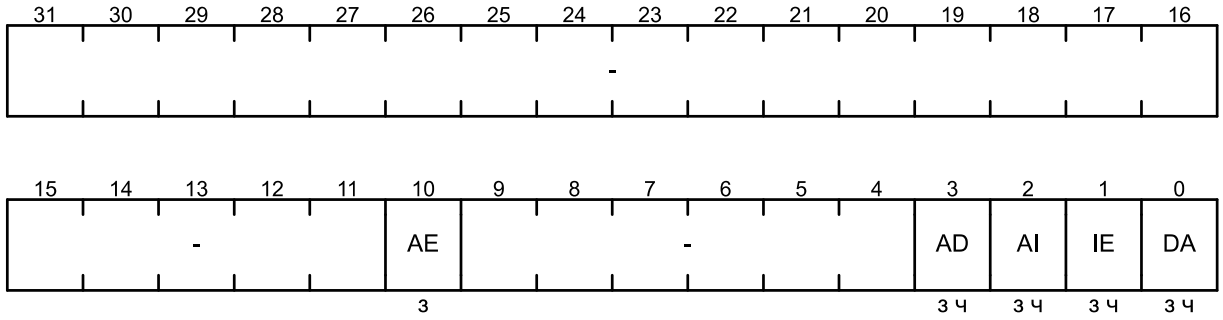


Поле	Биты	Описание
BUFSZ	31-21	Размер буфера данных в байтах одного аппаратного буферного слота. У каждой конечной точки имеется два буферных слота.
PI	20	Бит разрешения прерывания по принятию/передаче пакета. Для конечной точки OUT разрешает генерацию прерывания для каждого пакета, полученного по USB для этой конченой точки (пакет был сохранен во внутренних буферах конечной точки). Для конечной точки IN разрешает генерацию прерывания каждый раз, когда пакет передается по USB и внутренний буфер очищается. Значение по сбросу – 0.
CB	19	Бит очистки буферов. Очищает все буферы конечной точки, содержащие данные, если буфер в данный момент не активен.

Поле	Биты	Описание
CS	18	Бит управления состоянием STALL. При установке данного бита конечная точка будет возвращать пакет STALL на этапах данных и статуса в управляющих передачах. Данный бит автоматически очищается при получении следующего Setup-токена. Используется только тогда, когда конечная точка настроена как CONTROL EP.
MAXPL	17-7	Поле установки значения максимальной полезной нагрузки для конечной точки (максимальный размер одного пакета, отправляемого в/из конечной точки). Не всегда используются все биты поля. Данное поле не сбрасывается.
NT	6-5	Поле устанавливает количество дополнительных транзакций на микрокадр для конечных точек highspeed и на кадр для конечных точек full-speed. Доступно только для изохронных конечных точек. Данное поле не сбрасывается.
TT	4-3	Поле выбора типа обмена для конечной точки. При использовании типа передачи CONTROL, только для конечных точек типа OUT должен быть установлен тип передачи CONTROL, а конечная точка типа IN с тем же номером будет использоваться автоматически. Важно не использовать конечные точки OUT, у которых нет соответствующей конечной точки IN в качестве управляющей конечной точки CONTROL. Данное поле не сбрасывается. Принимаемые значения: 0 – Конечная точка управления (CONTROL) 1 – Изохронная конечная точка (ISOCHRONOUS) 2 – Конечная точка BULK 3 – Конечная точка прерывания (INTERRUPT)
EH	2	Бит перевода конечной точки в режим останова (HALT). Если данный бит установлен, все передачи к этой конечной точке будут получать handshake-пакет STALL. Значение по сбросу – 0.
ED	1	Бит отключения конечной точки. Если данный бит установлен, все передачи к этой конечной точке будут получать handshake-пакет NAK. Значение по сбросу – 0.
EV	0	Бит разрешения работы конечной точки. Если данный бит сброшен, все передачи к этой конечной точке будут проигнорированы и handshake-пакет не будет отправлен. Значение по сбросу – 0.

DMACTRL – управляющий регистр DMA конечной точки

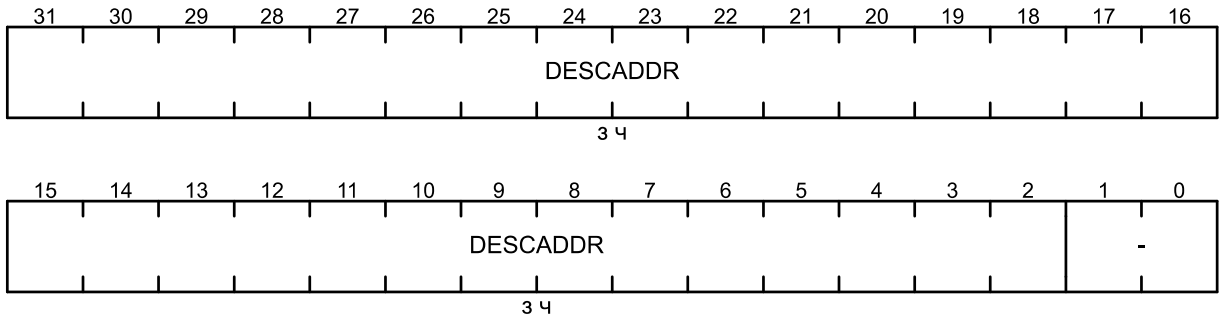
Смещение: EPOn/EPIn + 04h
Сброс: 0h



Поле	Биты	Описание
AE	10	Бит ошибки AHB. Для этой конечной точки произошла ошибка AHB
AD	3	Бит прекращения работы DMA. Отключает обработку дескриптора (сбрасывает бит DA) и прерывает текущую передачу DMA, если она активна
AI	2	Бит разрешения генерации прерывания при обнаружении ошибки на шине AHB для данной конечной точки
IE	1	Бит разрешения прерываний DMA. При установленном бите каждый раз, когда данные были получены или переданы в/из дескриптора будет генерироваться прерывание.
DA	0	Бит доступности дескрипторов. Устанавливается для указания DC контроллеру, что один или несколько дескрипторов были включены. Установленный бит позволяет считать дескриптор, расположенный по адресу, указанному в поле DESCADDR регистра DMADESC
–	31-11, 9-4	Зарезервировано

DMADESC – регистр адреса дескриптора конечной точки

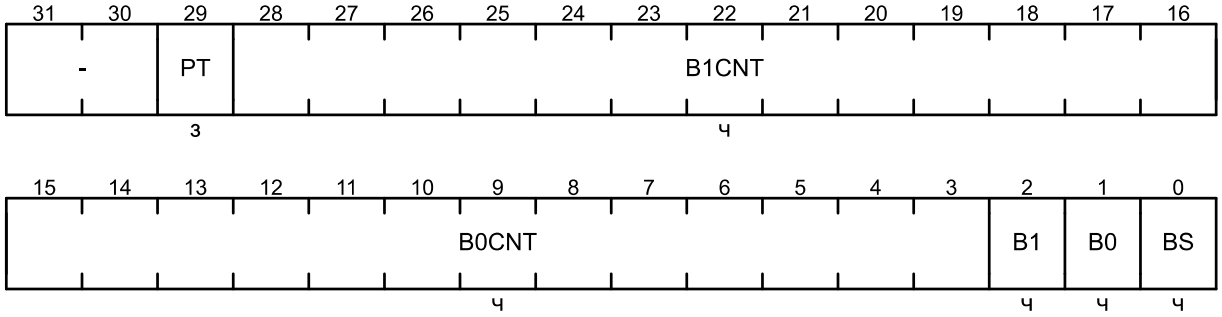
Смещение: EPOn/EPIn + 08h
Сброс: 0h



Поле	Биты	Описание
DESCADDR	31-2	Поле адреса таблицы дескрипторов. Содержит адрес следующего дескриптора в таблице дескрипторов. Данное поле не сбрасывается.
–	1-0	Зарезервировано

STAT – регистр статуса конечной точки

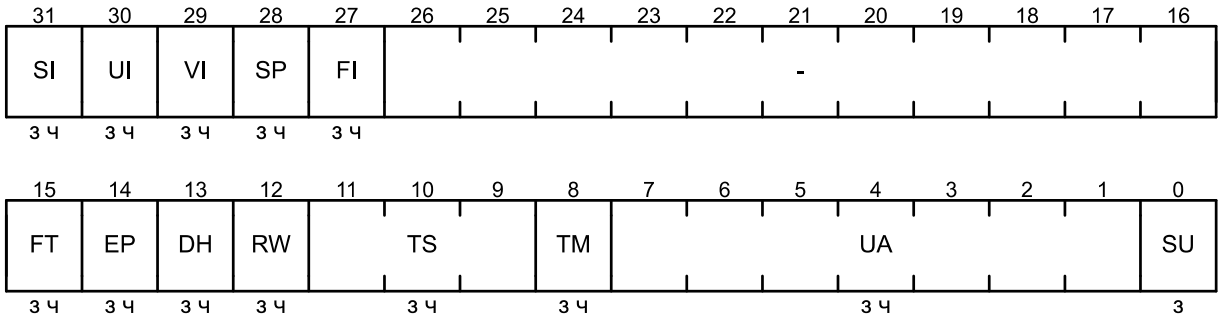
Смещение: EPOn/EPIn + 0Ch
Сброс: 0h



Поле	Биты	Описание
PR/PT	29	Для EPO PR – флаг принятого конечной точкой пакета. Устанавливается каждый раз, когда пакет был получен и сохранен во внутренних буферах. Для EPI PT – флаг переданного конечной точкой пакета. Устанавливается, когда пакет был передан и очищен из внутренних буферов. Бит очищается при записи 1.
B1CNT	28-16	Счетчик байтов первого внутреннего буфера. Количество байт в первом буфере.
B0CNT	15-3	Счетчик байтов нулевого внутреннего буфера. Количество байт в нулевом буфере.
B1	2	Флаг доступности данных в первом буфере. Устанавливается, когда первый буфер содержит доступные данные.
B0	1	Флаг доступности данных в нулевом буфере. Устанавливается, когда нулевой буфер содержит доступные данные.
BS	0	Флаг переключения внутренних буферов. Указывает буфер, выбранный в настоящий момент.
–	31-30	Зарезервировано

GCTRL – управляющий регистр DC контроллера

Смещение: + 200h
Сброс: 0h



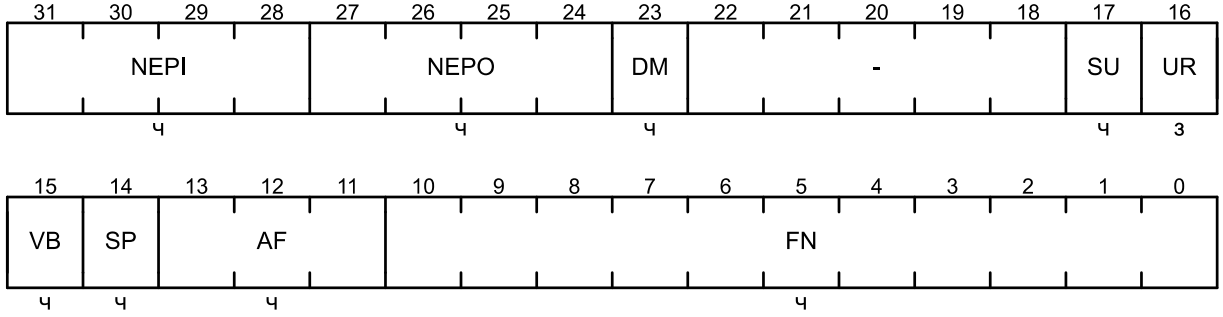
Поле	Биты	Описание
SI	31	Бит разрешения генерации прерывания при изменении статуса режима приостановки (SUSPEND).
UI	30	Бит разрешения генерации прерывания при получении сброса USB.
VI	29	Бит разрешения генерации прерывания при изменении статуса VBUS.
SP	28	Бит разрешения генерации прерывания при изменении скоростного режима, например при переходе FS → HS.
FI	27	Бит разрешения генерации прерывания при получении нового токена Start of frame (SOF).
FT	15	Бит включения режима функционального тестирования. Включает режим, который сокращает все таймеры, такие как reset и chirp до 8-ми тактовых импульсов.
EP	14	Бит включения подтяжки Pull-Up на линии D+, для оповещения хоста о подключении DC контроллера к линии.
DH	13	Бит выключения режима High-speed. При установке данного бита, DC контроллер не будет отправлять handshake-пакет high-speed, что означает работу DC контроллера только в режиме full-speed.
RW	12	Бит запуска сигнала удаленного пробуждения (REMOTE WAKEUP). Он самоочищается и будет сброшен после завершения передачи удаленного пробуждения, если в данный момент DC контроллер находился в режиме SUSPEND. При установке бита не в режиме SUSPEND, данный бит немедленно сбросится. Если данный бит уже установлен, запись в него будет проигнорирована. Значение по сбросу – 0.
TS	11-9	Поле выбора тестового режима, может принимать значения:

0	Зарезервировано
1	Test_J
2	Test_K
3	Test_SE0_NAK
4	Test_Packet
5-7	Зарезервировано

Поле	Биты	Описание
TM	8	Бит включения тестового режима. Стоит обратить внимание, что тестовый режим нельзя установить без сброса или цикла включения питания, и его нельзя установить, если установлен бит DH.
UA	7-1	Поле адреса, присвоенного DC контроллеру на шине USB.
SU	0	Бит установки USB адреса. При установке данного бита, USB адрес, сохраненный в поле UA, будет задан DC контроллеру.
–	26-16	Зарезервировано

GSTAT – регистр статуса DC контроллера

Смещение: + 204h
Сброс: 0h



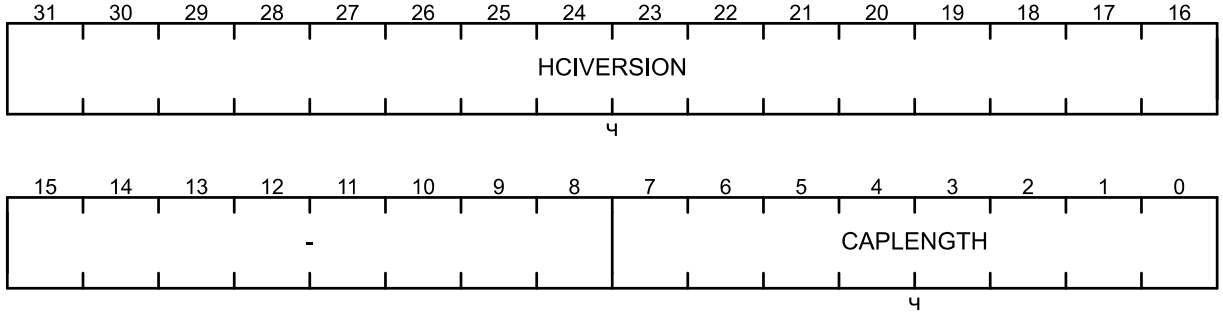
Поле	Биты	Описание
NEPI	31-28	Поле количества реализованных конечных точек типа IN. Содержит значение количества N-1 (включая EP0).
NEPO	27-24	Поле количества реализованных конечных точек типа OUT. Содержит значение количества N-1 (включая EP0).
DM	23	Флаг режима работы с процессорной шиной: 0 – При передаче данных по процессорной шине DC контроллер использует режим slave 1 – При передаче данных по процессорной шине DC контроллер использует режим master (DMA)
SU	17	Флаг установленного режима приостановки (SUSPEND). Принимаемые значения: 0 – Устройство приостановлено (suspended) 1 – Устройство не приостановлено (not suspended)
UR	16	Флаг сброса USB. Устанавливается при каждом обнаружении сброса по линии USB. Очищается записью единицы.
VB	15	Флаг VBUSVALID. Устанавливается в 1 при обнаружении допустимого уровня напряжения на линии USB VBUS.
SP	14	Флаг текущего скоростного режима устройства на шине USB. Принимаемые значения: 0 – Режим high-speed (HS) 1 – Режим full-speed (FS)
AF	13-11	Поле количества дополнительных кадров, полученных с текущим номером кадра
FN	10-0	Поле номера кадра. Содержит значение последнего полученного токена SOF.
–	22-18	Зарезервировано

A.29 Регистры контроллера USB 2.0 HC

Базовый адрес: 4108_1000h Регистры блока USB HC0
4109_1000h Регистры блока USB HC1

CAPLENGTH – регистр характеристик USB HC

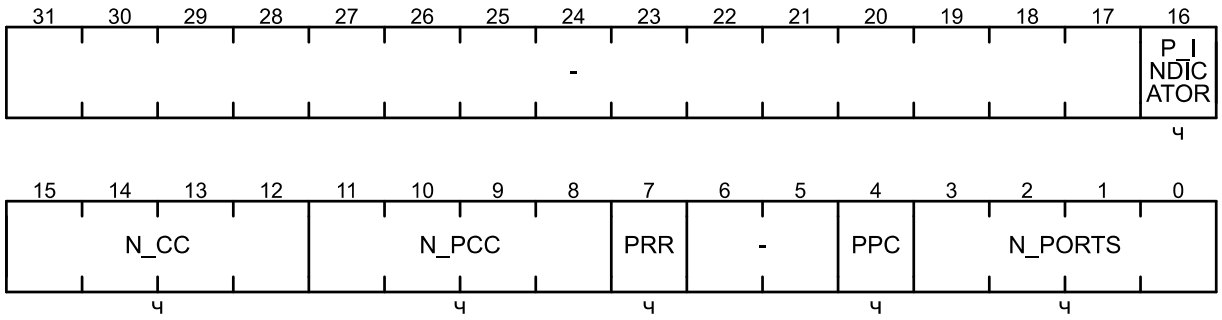
Смещение: + 00h
Сброс: 0100_0014h



Поле	Биты	Описание
HCIVERSION	31-16	Данное поле содержит номер версии хост-контроллера USB
CAPLENGTH	7-0	Данное поле содержит смещение в байтах от базового адреса хост-контроллера EHCI до первого рабочего (operation) регистра (другими словами, количество байт, используемых для реализации набора регистров возможностей (capability)). Данное значение может использоваться для смещения в рабочие регистры.
–	15-8	Зарезервировано

HCSPARAMS – регистр структурных параметров USB HC

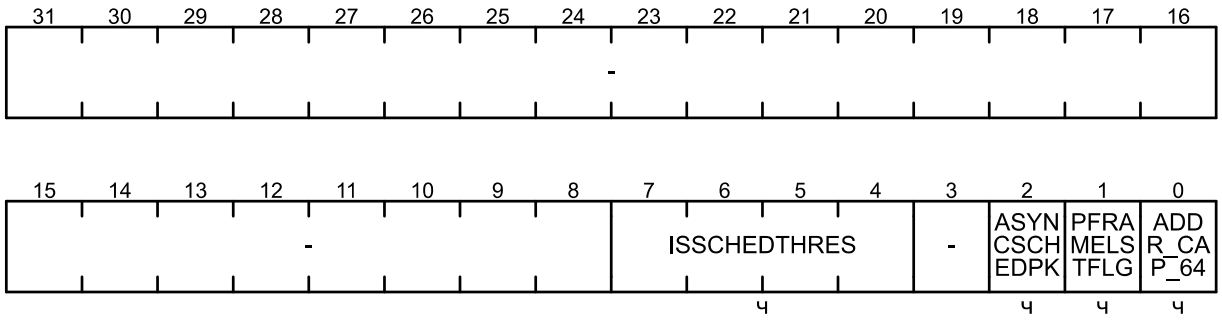
Смещение: + 04h
Сброс: 0000_1111h



Поле	Биты	Описание
P_INDICATOR	16	Индикатор порта. Данный бит указывает, поддерживают ли порты контроллера управление индикатором порта.
N_CC	15-12	Количество вспомогательных контроллеров. Данное поле отображает количество вспомогательных UHCI контроллеров, встроенных в данный хост-контроллер, для поддержки LS и FS устройств USB версии 1.1.
N_PCC	11-8	Количество портов на вспомогательный контроллер. Данное поле указывает количество портов, поддерживаемых каждым дополнительным контроллером.
PRR	7	Правила маршрутизации портов. Данный бит указывает, какое из двух соглашений было использовано при маршрутизации вспомогательного контроллера.
PPC	4	Управление питанием порта. Данный бит используется для индикации возможности управления питанием порта хост-контроллером.
N_PORTS	3-0	Данное поле отображает количество физических портов, реализованных в данном хост-контроллере.
–	31-17, 6-5	Зарезервировано

HCCPARAMS – регистр функциональных параметров USB HC

Смещение: + 08h
Сброс: 0000_0086h

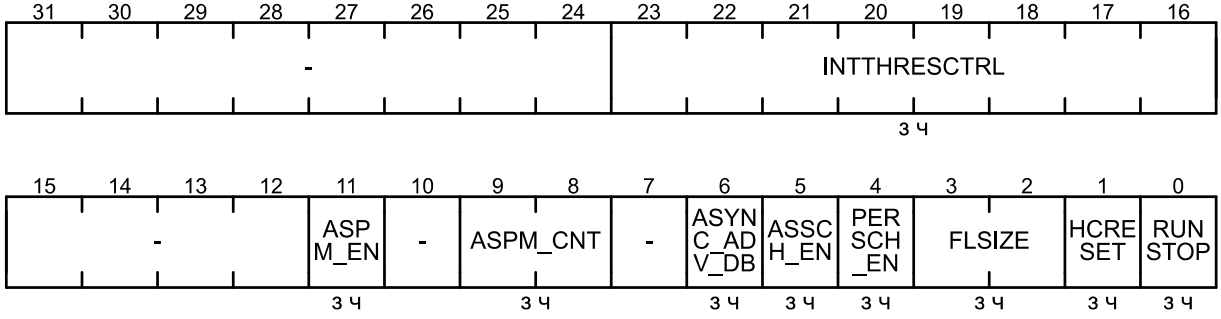


Поле	Биты	Описание
ISO_SCHED_ THRES	7-4	Граница изохронного списка. Данное поле используется для информирования, когда, относительно места в изохронном списке, где в данный момент находится хост-контроллер, он может безопасно обновить изохронный список. Каждый изохронный дескриптор передачи содержит 8 микрокадров транзакций. Объем кэширования указывается в этом поле, поскольку его необходимо учитывать, когда системное программное обеспечение USB обновляет записи в списке, чтобы опередить хост-контроллер. Два важных случая: 1) Бит 7 равен 0. Тогда младшие биты [6:4] указывают количество микрокадров транзакций, кэшированных хост-контроллером. 2) Бит 7 равен 1. Это указывает, что хост-контроллер кэширует структуры данных дескриптора для всего кадра (8 микрокадров).
ASYNC_SCHED_ PARK	2	Флаг доступного режима Asynchronous Schedule Park. Если данный бит равен единице, хост-контроллер поддерживает функцию остановки для высокоскоростных заголовков очереди в асинхронном списке. Функция может быть отключена или включена и установлена на определенный уровень с помощью полей ASPM_EN и ASPM_CNT регистра EHCCMD.
P_FRAME_LST_ FLAG	1	Флаг программируемого списка кадров. Указывает, допускает ли хост-контроллер размер списка кадров, отличный от стандартных 1024 записей.
		0 — Хост-контроллер не допускает размер списка кадров, отличный от 1024-х записей
		1 — Хост-контроллер позволяет программировать размер списка кадров с помощью поля FLSIZE регистра EHCCMD. Список кадров по-прежнему должен начинаться с адреса, выровненного по границе 4 КБ.

Поле	Биты	Описание	
ADDR_CAP_64	0	Флаг возможности 64-разрядной адресации. Указывает, способен ли хост-контроллер использовать 64-разрядные указатели адресов во время взаимодействия со структурами данных памяти.	
		0	Хост-контроллер не способен генерировать адреса, превышающие 32 бита, при обращении к памяти в качестве ведущего.
		1	Хост-контроллер способен генерировать адреса размером более 32 бит при обращении к памяти в качестве ведущего. Это означает, что он ожидает использования 64-разрядных структур данных, описанных в спецификации EHCI.
–	31-8, 3	Зарезервировано	

EHCCMD – управляющий регистр EHC USB

Смещение: + 14h
Сброс: 0008_0B00h



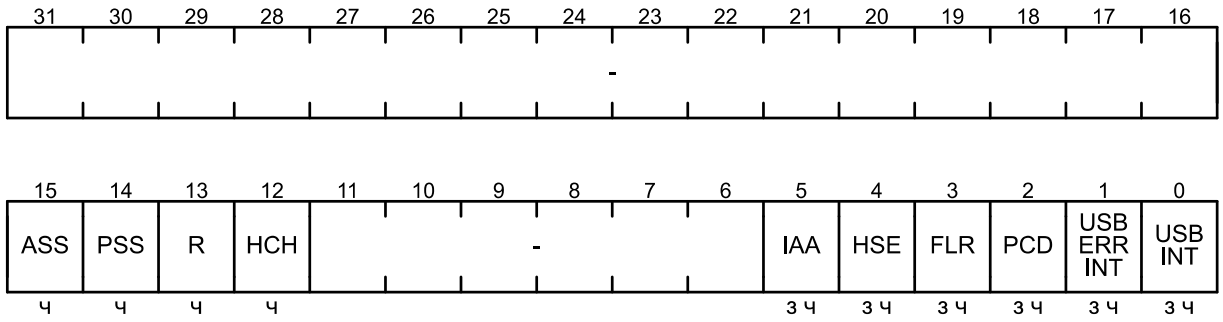
Поле	Биты	Описание	
INT_THRES_ CNTRL	23-16	Порог прерывания. Значение, указываемое в поле, задает максимальную частоту, с которой хост-контроллер может генерировать прерывания. Допустимые коды приведены ниже:	
		00h	Зарезервировано
		01h	Прерывание после 1 микрокадра
		02h	Прерывание после 2-х микрокадров
		04h	Прерывание после 4-х микрокадров
		08h	Прерывание после 8-ми микрокадров (Значение по умолчанию. Соответствует 1 мс).
		10h	Прерывание после 16-ти микрокадров (2 мс)
		20h	Прерывание после 32-х микрокадров (4 мс)
		40h	Прерывание после 64-х микрокадров (8 мс)
ASPM_EN	11	Включение режима Asynchronous Schedule Park. По умолчанию установлен. Используется для включения/выключения режима Park Mode.	
		0	Режим Park Mode отключен
		1	Режим Park Mode включен

Поле	Биты	Описание	
ASPM_CNT	9-8	Счетчик режима Asynchronous Schedule Park. Данное поле содержит подсчет количества последовательных транзакций, которые хост-контроллеру разрешено выполнять из одного заголовка высокоскоростной асинхронной очереди, прежде чем продолжить прохождение по асинхронной очереди. Допустимые значения от 1h до 3h (по умолчанию 3h). Не следует записывать в данное поле нулевое значение при установленном бите ASPM_EN, поскольку это приведет к неопределенному поведению.	
ASYNC_ADV_DB	6	Прерывание при продвижении по асинхронной очереди. Данный бит используется для сигнализирования хосту вызвать прерывание при следующем продвижении по асинхронной очереди (бит IAA в регистре EHCINTR также должен быть установлен для запуска прерывания). Хост-контроллер очищает этот бит после инициирования прерывания. При отключенной асинхронной очереди не допускается установка данного бита, поскольку это приведет к неопределенному результату.	
ASSCH_EN	5	Включение асинхронной очереди.	
		0	Не обрабатывать асинхронную очередь (по умолчанию)
		1	Обрабатывать асинхронную очередь, используя регистр ASYNCLISTADDR в качестве указателя
PERSCH_EN	4	Включение периодической очереди.	
		0	Не обрабатывать периодическую очередь (по умолчанию)
		1	Обрабатывать периодическую очередь, используя регистр PERIODICLISTBASE в качестве указателя
FLSIZE	3-2	Размер списка кадров	
		0	1024 элемента в списке кадров (по умолчанию)
		1	512 элементов в списке кадров
		2	256 элементов в списке кадров
		3	Зарезервировано
HCRESET	1	Сброс хост-контроллера. Бит позволяет выполнить программный сброс. При установке сбрасываются внутренние конвейеры хост-контроллера и внутренние конечные автоматы, завершаются все выполняемые транзакции USB. Не вызывает сброс на шине USB и не влияет на регистры конфигурации PCI.	
		0	Если позже читается как «0», сброс выполнен
		1	Выполнить сброс хост-контроллера

Поле	Биты	Описание	
RUNSTOP	0	Запуск/остановка. Данный бит используется для запуска и остановки хост-контроллера при обработке им обоих очередей. При остановке контроллер устанавливает бит HCH регистра EHCSTS. Хост-контроллер должен остановиться в течение 16 микрокадров с момента обнаружения нулевого значения бита RUNSTOP. Состояние остановки проверяется считывнием бита HCH.	
		0	Прекращение обработки очередей. Если очереди в данный момент обрабатываются, при установке нуля хост-контроллер завершает текущую и все активные конвейерные транзакции и останавливается.
		1	Запуск и продолжение обработки всех очередей, включенных в битах 4 и 5 данного регистра
–	31-24, 15-12, 10, 7	Зарезервировано	

EHCSTS – регистр статуса EHC USB

Смещение: + 18h
Сброс: 0000_1000h

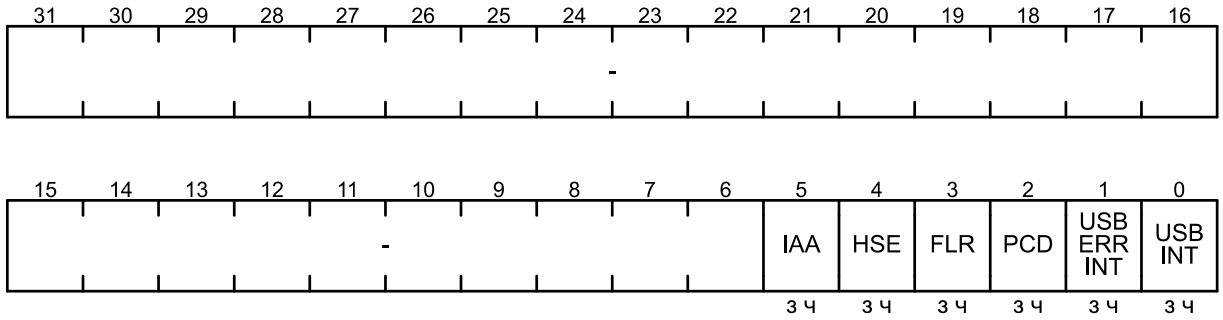


Поле	Биты	Описание	
ASS	15	Бит статуса асинхронной очереди. Бит отображает текущий реальный статус асинхронной очереди. От хост-контроллера не требуется немедленно отключать или включать асинхронную очередь, когда программно изменяется бит ASSCH_EN в регистре USBCMD.	
		0	Асинхронная очередь выключена
		1	Асинхронная очередь включена
PSS	14	Бит статуса периодической очереди. Бит отображает текущий реальный статус периодической очереди. От хост-контроллера не требуется немедленно отключать или включать периодическую очередь, когда программно изменяется бит PERSCH_EN в регистре USBCMD.	
		0	Периодическая очередь выключена
		1	Периодическая очередь включена
R	13	Бит статуса рекламации. Используется для обнаружения пустой асинхронной очереди.	

Поле	Биты	Описание
HCH	12	Бит статуса остановки. Используется хост-контроллером для сообщения о фактическом внутреннем статусе работы хост-контроллера. Хост-контроллер устанавливает этот бит после того, как он прекратил работу в результате сброса бита RUNSTOP программно, либо аппаратно (например, при внутренней ошибке).
IAA	5	Бит статуса прерывания при асинхронном продвижении. Системное программное обеспечение может заставить хост-контроллер выдать прерывание при следующем продвижении хост-контроллера по асинхронной очереди, записав единицу в бит ASYNC_ADV_DB регистра USBCMD.
HSE	4	Бит статуса системной ошибки хоста. PCI условия, устанавливающие этот бит, включают: PCI Parity error, PCI Master Abort, и PCI Target Abort. При возникновении этой ошибки, хост-контроллер очищает бит RUNSTOP в командном регистре EHCCMD, чтобы предотвратить дальнейшее выполнение запланированных дескрипторов передач.
FLR	3	Бит обнаружения переключения списка кадров. Устанавливается аппаратно хост-контроллером каждый раз при переключении списка кадров с максимального значения на 0. Обычно это происходит после завершения выполнения 1024 записей, но если используется программируемый размер списка кадров (поле FLSIZE регистра EHCCMD), то переход может произойти на границах 1024, 512 или 256 элементов. Сбрасывается записью единицы.
PCD	2	Бит обнаружения смены порта. Хост-контроллер устанавливает этот бит, когда любой порт, для которого бит обладателя порта PO, имеющий до этого нулевое значение, меняет свое состояние с нуля на единицу.
USBERRINT	1	Бит статуса прерывания по ошибке USB. Хост-контроллер устанавливает этот бит, когда завершение транзакции USB приводит к возникновению ошибки. Если в дескрипторе передачи, на котором произошло прерывание с ошибкой, также был установлен бит IOC, устанавливаются как этот бит, так и бит USBINT.
USBINT	0	Бит статуса USB-прерывания. Хост-контроллер устанавливает этот бит при завершении транзакции USB, что приводит к удалению дескриптора передачи, у которого был установлен бит IOC. Также хост-контроллер устанавливает этот бит при обнаружении короткого пакета (фактическое количество полученных байт оказалось меньше ожидаемого количества).
–	31-16, 11-6	Зарезервировано

EHCINTR – регистр разрешения прерываний EHC USB

Смещение: + 1Ch
Сброс: 0000_0000h

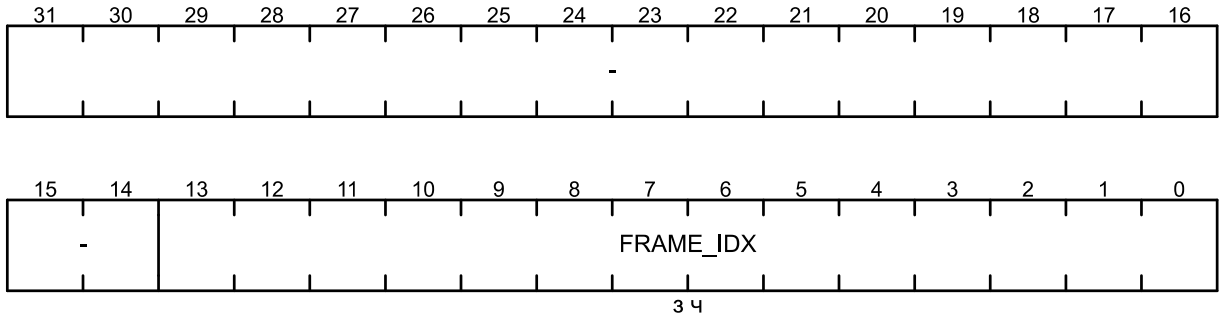


Поле	Биты	Описание
IAA	5	Бит разрешения генерации прерывания при продвижении по асинхронной очереди. Используется совместно с битом IAA регистра USBSTS. Если бит установлен, тогда в любой момент, при установке хост-контроллером бита IAA регистра USBSTS, будет сгенерировано прерывание. Флаг прерывания снимается при записи единицы в бит IAA регистра USBSTS.
HSE	4	Бит разрешения генерации прерывания при системной ошибке хоста. Используется совместно с битом HSE регистра USBSTS. Если бит установлен, тогда в любой момент, при установке хост-контроллером бита HSE регистра USBSTS, будет сгенерировано прерывание. Флаг прерывания снимается при записи единицы в бит HSE регистра USBSTS.
FLR	3	Бит разрешения генерации прерывания при переключении списка кадров. Используется совместно с битом FLR регистра USBSTS. Если бит установлен, тогда в любой момент, при установке хост-контроллером бита FLR регистра USBSTS, будет сгенерировано прерывание. Флаг прерывания снимается при записи единицы в бит FLR регистра USBSTS.
PCD	2	Бит разрешения генерации прерывания при обнаружении смены порта. Используется совместно с битом PCD регистра USBSTS. Если этот бит установлен, тогда в любой момент, при установке хост-контроллером бита PCD регистра USBSTS, будет сгенерировано прерывание. Флаг прерывания снимается при записи единицы в бит PCD регистра USBSTS.

Поле	Биты	Описание
USBERRINT	1	Бит разрешения генерации прерывания при ошибке USB. Используется совместно с битом USBERRINT регистра USBSTS. Если этот бит установлен, тогда в любой момент, при установке хост-контроллером бита USBERRINT регистра USBSTS, будет сгенерировано прерывание. Флаг прерывания снимается, при записи единицы в бит USBERRINT регистра USBSTS.
USBINT	0	Бит разрешения прерывания USB. Используется совместно с битом USBINT регистра USBSTS. Если этот бит установлен, тогда в любой момент, при установке хост-контроллером бита USBINT регистра USBSTS, будет сгенерировано прерывание. Флаг прерывания снимается при записи единицы в бит USBINT регистра USBSTS.
–	31-6	Зарезервировано

FRINDEX – регистр индекса кадра в периодическом списке EHC USB

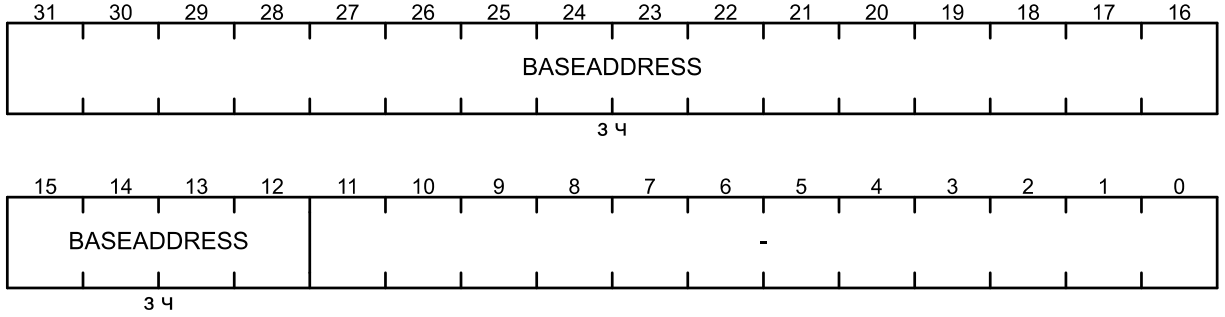
Смещение: + 20h
Сброс: 0000_0000h



Поле	Биты	Описание	
FRAME_IDX	13-0	Поле индекса кадра. В зависимости от используемого размера списка кадров, старшие биты в этой группе указывают текущий номер записи в списке кадров. Три младших бита отражают смещение микрокадра внутри записи списка кадров. Назначение бит в зависимости от размера списка кадров:	
		1024	Биты [12-03] – запись в списке кадров; Биты [2-0] – номер микрокадра.
		512	Биты [11-03] – запись в списке кадров; Биты [2-0] – номер микрокадра.
		256	Биты [10-03] – запись в списке кадров; Биты [2-0] – номер микрокадра.
–	31-14	Зарезервировано	

PERIODICLISTBASE – регистр начального адреса периодического списка кадров EHC USB

Смещение: + 28h
Сброс: 0000_0000h

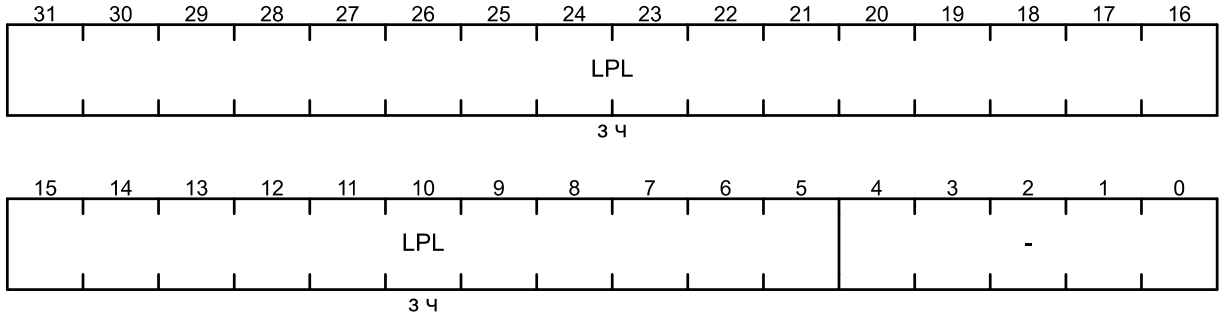


Поле	Биты	Описание
BASEADDRESS	31-12	Базовый адрес периодического списка кадров
–	11-0	Зарезервировано

Примечание– Хост-контроллер использует содеримое данного регистра в качестве базового адреса периодического списка (выровненного по границе 4КБ). Далее хост-контроллер последовательно проходит по списку, используя текущее содержимое регистра FRINDEX в качестве смещения.

ASYNCLISTADDR – регистр указателя на адрес текущего элемента в асинхронном списке EHC USB

Смещение: + 2Ch
Сброс: 0000_0000h

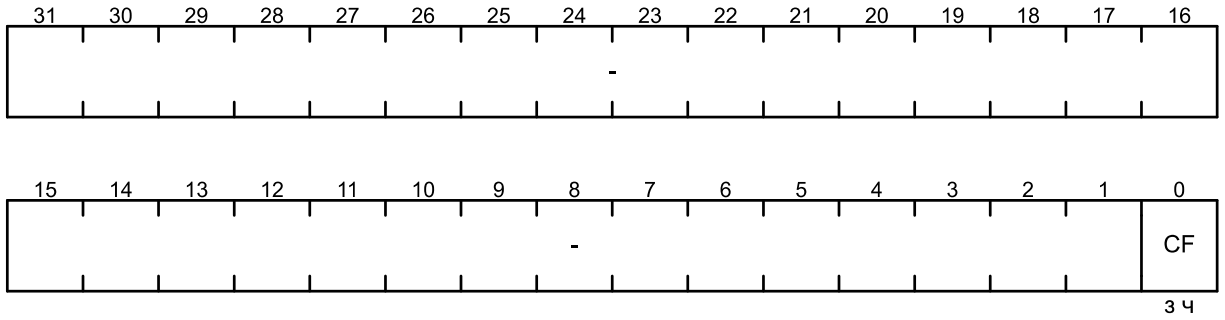


Поле	Биты	Описание
LPL	31-5	Указатель на текущий элемент в асинхронной очереди
–	4-0	Зарезервировано

Примечание – Поскольку предполагается, что каждый адрес начальной записи в очереди выровнен по границе 32 байта, младшие 5 бит этого регистра не могут меняться и всегда считываются нулями.

CONFIGFLAG – регистр флага конфигурации EHC USB

Смещение: + 54h
Сброс: 0000_0000h

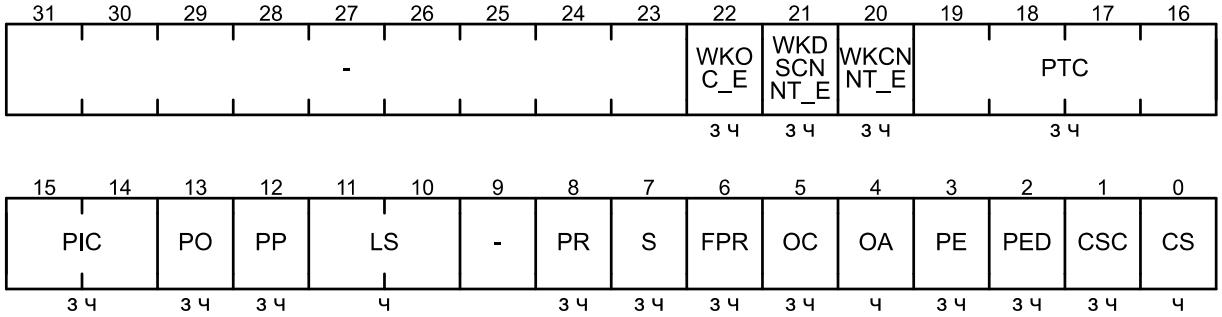


Поле	Биты	Описание	
CF	0	Конфигурация EHC. Определяет схему логики маршрутизации портов по умолчанию	
		0	По умолчанию для каждого порта маршрутизация осуществляется к вспомогательному контроллеру UHCI.
		1	По умолчанию для каждого порта маршрутизация осуществляется к контроллеру EHCI. Если подключенное устройство окажется версии USB 1.1, управление будет передано вспомогательному контроллеру UHCI.
–	31-1	Зарезервировано	

Примечание – Установка данного бита, как правило, является последним действием при настройке хост-контроллера.

EHCPORTSC – регистр статуса и управления портом EHC USB

Смещение: + 58h
Сброс: 0000_2000h



Поле	Биты	Описание	
WKOC_E	22	Бит включения пробуждения при перегрузке по току. Устанавливает возможность генерации события пробуждения портом в случае перегрузки по току на шине USB. При отключенном питании порта данный бит сброшен.	
		0	Порт не генерирует событие пробуждения при перегрузке по току
		1	Порт генерирует событие пробуждения при перегрузке по току
WKDSCNNT_E	21	Бит включения пробуждения при отключении. Устанавливает возможность генерации события пробуждения портом в случае отключения устройства от шины USB. При отключенном питании порта данный бит сброшен.	
		0	Порт не генерирует событие пробуждения при отключениях от шины USB
		1	Порт генерирует событие пробуждения при отключениях от шины USB
WKCNNT_E	20	Бит включения пробуждения при подключении. Устанавливает возможность генерации события пробуждения портом в случае подключения устройства к шине USB. При отключенном питании порта данный бит сброшен.	
		0	Порт не генерирует событие пробуждения при подключениях к шине USB
		1	Порт генерирует событие пробуждения при подключениях к шине USB
PTC	19-16	Поле управления тестированием порта. Значение в поле определяет выбранный тестовый режим:	
		0000b	Тестовый режим: отключен (по умолчанию)
		0001b	Тестовый режим: Test J_STATE
		0010b	Тестовый режим: Test K_STATE
		0011b	Тестовый режим: Test SEO_NAK
		0100b	Тестовый режим: Test Packet
		0101b	Тестовый режим: Test FORCE_ENABLE

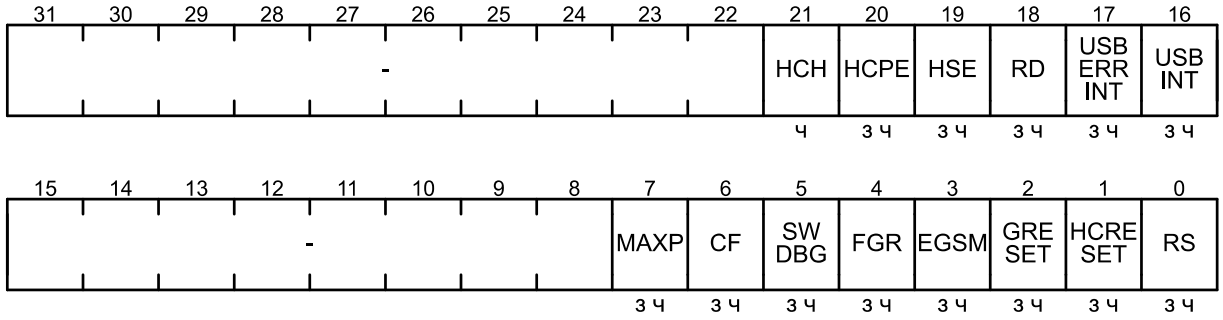
Поле	Биты	Описание	
PIC	15-14	Поле управления индикатором порта. Поле не используется, если бит P_INDICATOR регистра HCSPARAMS равен нулю (индикаторы не поддерживаются). Наборы состояний индикаторов:	
		00b	Выключить все индикаторы (по умолчанию)
		01b	Включить желтый индикатор
		10b	Включить зеленый индикатор
		11b	Зарезервировано
PO	13	Бит выбора контроллера. Допускается программная установка бита, чтобы передать право управления вспомогательному контроллеру UHCI, при обнаружении не высокоскоростного подключенного устройства.	
		0	Порт под управлением контроллера EHCI
		1	Порт под управлением контроллера UHCI
PP	12	Бит питания порта. Используется совместно с битом PPC регистра HCSPARAMS. Когда на питаемом порту обнаружено состояние перегрузки по току, с помощью бита PP питание порта может быть отключено.	
		0	Питание порта выключено
		1	Питание порта включено
LS	11-10	Поле статуса линии. Используется для определения текущего состояния дифференциальной пары сигналов USB, чтобы определить, является ли подключенное устройство низкоскоростным (LS) или нет. Эти биты должны быть считаны до сброса порта. Бит 11 отражает состояние линии D+, а бит 10 отражает состояние линии D-. Коды в битах выглядят следующим образом:	
		00b	Состояние SE0: устройство не является низкоскоростным, выполните сброс EHCI.
		01b	Состояние JSTATE: устройство не является низкоскоростным, выполните сброс EHCI.
		10b	Состояние KSTATE: устройство является низкоскоростным, освободите доступ к порту (с помощью установки бита PO).
		11b	Состояние UNDEFINED: устройство не является низкоскоростным, выполните сброс EHCI.
PR	8	Бит сброса порта. Используется для инициирования и подтверждения события сброса USB-порта. Спецификация USB 2.0 предполагает последовательность, которую необходимо соблюдать при выполнении сброса: – Контроллер должен быть остановлен (бит HCH регистра EHCSTS равен 1) перед попыткой выполнения сброса; – Бит PR программно устанавливается, чтобы инициировать сброс, и не сбрасывается, для завершения последовательности сброса до истечения указанного времени; – Далее можно программно опрашивать (считывать) этот бит до тех пор, пока не будет обнаружено, что он равен нулю (что указывает на завершение процедуры сброса).	

Поле	Биты	Описание
S	7	Бит приостановки порта. При отключенном питании порта данный бит сброшен. Примечания: – Когда порт находится в состоянии приостановки, нисходящий трафик блокируется, за исключением сброса порта. – Находясь в этом состоянии, порт все еще способен обнаруживать команду возобновления. – При программной установке этого бита, для приостановки порта, сначала будет завершена текущая транзакция, а затем произойдет приостановка. Можно опрашивать этот бит до тех пор, пока не будет считано нулевое значение, указывающее на завершение операции приостановки. – Этот бит будет автоматически сброшен при сбросе (изменении бита PR с 0 на 1) или возобновлении (изменении бита FPR с 1 на 0).
FPR	6	Бит принудительного возобновления работы порта. Принудительное возобновление передачи сигналов через порт может быть выполнено как программно, так и самим хост-контроллером: – Программная установка данного бита, для инициирования возобновления работы на приостановленном порту; – Хост-контроллер устанавливает этот бит, если он обнаруживает переход от состояния J к состоянию K на USB-порту, во время приостановки. В этом случае хост-контроллер также устанавливает бит PCD регистра EHCSTS. Это, в свою очередь, возможно сконфигурировать таким образом, чтобы вызвать генерацию прерывания (бит PCD регистра EHCINTR). При отключенном питании порта данный бит сброшен.
OC	5	Бит изменения перегрузки по току. Устанавливается, если происходит изменение бита OA. По сути, этот бит фиксирует любое событие изменения состояния перегрузки, в то время как бит OA сообщает только о текущем состоянии перегрузки. Данный бит можно сбросить программно, записав в него единицу.
OA	4	Флаг активной перегрузки по току. Устанавливается при обнаружении события перегрузки по току на USB-порту. Флаг самостоятельно очищается при пропадании перегрузки.
PE	3	Бит изменения включения/выключения порта. Устанавливается, если статус включения/выключения порта изменился из-за ошибки. Данный бит можно сбросить программно, записав в него единицу. При отключенном питании порта данный бит сброшен.
PED	2	Бит включения/выключения порта. Устанавливается хост-контроллером, когда последовательность сброса завершена и проверяется, что порт подключен к высокоскоростному устройству. После включения порт может быть отключен из-за неисправности, либо программно. При отключенном питании порта данный бит сброшен.

Поле	Биты	Описание
CSC	1	Бит изменения статуса подключения. Устанавливается хост-контроллером, если происходит изменение текущего статуса подключения порта. Очищается записью единицы. При отключенном питании порта данный бит сброшен.
CS	0	Флаг текущего статуса подключения. Устанавливается хост-контроллером, если к этому порту подключено устройство, в противном случае он сбрасывается. При отключенном питании порта данный бит сброшен.
–	31-23, 9	Зарезервировано

UHCCMDSTS – управляющий и статусный регистр UHC USB

Смещение: + 800h
Сброс: 0000_2000h



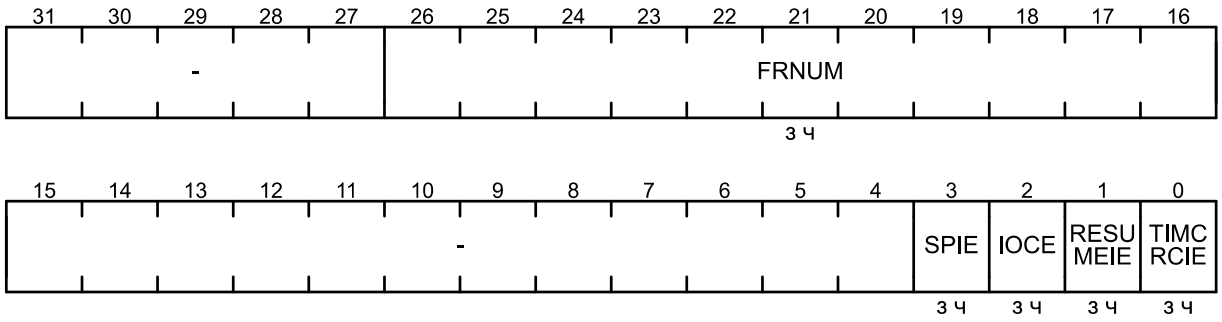
Поле	Биты	Описание
HCH	21	Флаг статуса остановки хост-контроллера. Хост-контроллер устанавливает этот бит после того, как он прекратил работу в результате сброса бита запуска/остановки (RS) программным способом, либо аппаратно хост-контроллером.
HCPE	20	Бит ошибки процесса работы хоста. Хост-контроллер устанавливает этот бит, когда обнаруживает фатальную ошибку и указывает, что он столкнулся с ошибкой проверки согласованности при обработке дескриптора передачи. При возникновении этой ошибки хост-контроллер сбрасывает бит запуска/остановки (RS), чтобы предотвратить дальнейшее выполнение очереди. В системе генерируется аппаратное прерывание.
HSE	19	Бит системной ошибки хоста. Хост-контроллер устанавливает этот бит, когда во время доступа к хост-системе возникает серьезная ошибка, связанная с модулем хост-контроллера. PCI условия, устанавливающие этот бит, включают: ошибку четности PCI, PCI Master Abort, PCI Target Abort. При возникновении этой ошибки хост-контроллер очищает бит запуска/остановки (RS), чтобы предотвратить дальнейшее выполнение запланированных дескрипторов передачи. В системе генерируется аппаратное прерывание.

Поле	Биты	Описание
RD	18	Бит обнаружения команды восстановления. Хост-контроллер устанавливает этот бит, когда получает сигнал RESUME от USB-устройства. Это допустимо только в том случае, если хост-контроллер находится в состоянии глобальной приостановки (установлен бит EGSM).
USBERRINT	17	Бит прерывания по ошибке USB. Хост-контроллер устанавливает этот бит, когда завершение транзакции USB приводит к возникновению ошибки. Если в дескрипторе передачи, на котором произошло прерывание с ошибкой, также был установлен бит IOC, устанавливаются как этот бит, так и бит USBINT.
USBINT	16	Бит USB-прерывания. Хост-контроллер устанавливает этот бит, когда причиной прерывания является завершение транзакции USB, в дескрипторе передачи которой был установлен бит IOC. Хост-контроллер также устанавливает этот бит при обнаружении короткого пакета. Сброс бита осуществляется записью единицы.
MAXP	7	Бит максимального размера пакета. Выбирает максимальный размер пакета, который может быть использован для заполнения полосы пропускания в режиме FullSpeed в конце кадра. Это значение используется хост-контроллером для определения, следует ли ему инициировать новую транзакцию, основываясь на времени, оставшемся в счетчике SOF. 0 — 32 байта 1 — 64 байта
CF	6	Бит конфигурации. Устанавливается программно, в качестве последнего действия в процессе настройки хост-контроллера. Бит не влияет на аппаратную часть, а используется только для информирования программной части.
SWDBG	5	Бит программной отладки. В режиме программной отладки хост-контроллер очищает бит запуска/остановки RS после завершения каждой транзакции USB. Следующая транзакция выполняется при повторной установке бита RS. Изменение бита SWDBG допускается только когда хост-контроллер находится в остановленном состоянии. 0 — Обычный режим 1 — Режим отладки
FGR	4	Бит принудительного запуска глобального возобновления. При установке бита хост-контроллер отправляет сигнал глобального возобновления на шину USB. Сбрасывается программно по истечении 20 мс, чтобы прекратить отправку сигнала глобального возобновления. К этому времени все USB-устройства должны быть готовы к работе с шиной. Изменение бита с 1 на 0 приводит к отправке портом низкоскоростного сигнала EOP. Бит будет оставаться установленным до завершения EOP.

Поле	Биты	Описание
EGSM	3	Бит принудительного запуска режима глобальной приостановки. При установке бита хост-контроллер переходит в состояние глобальной приостановки, в течение которого не выполняется никаких USB-транзакций. В состоянии приостановки хост-контроллер может получать сигналы возобновления с шины USB и системные прерывания. Для выхода из режима глобальной приостановки данный бит может быть программно сброшен. При программном сбросе бита FGR данный бит также должен быть сброшен. Перед установкой данного бита необходимо убедиться, что бит RS сброшен.
GRESET	2	Бит запуска глобального сброса. При установке бита хост-контроллер отправляет сигнал глобального сброса на шину USB и затем сбрасывает всю свою логику, включая внутренние регистры хаба. Данный бит сбрасывается программно по истечении как минимум 10 мс (как указано спецификации USB). Примечание: аппаратный сброс микроконтроллера приводит к тому же эффекту, что и глобальный сброс (бит GRESET), за исключением того, что хост-контроллер не отправляет команду глобального сброса по шине USB.
HCRESET	1	Бит сброса хост-контроллера. При установке бита модуль хост-контроллера сбрасывает свои внутренние таймеры, счетчики, конечные автоматы и т.д. в их начальное значение. Любая транзакция, выполняемая в данный момент по USB, немедленно прекращается. Бит очищается хост-контроллером по завершении процесса сброса.
RS	0	Бит запуска/остановки хост-контроллера (Run/Stop). Пока бит установлен хост-контроллер продолжает обрабатывать очередь. При сбросе хост-контроллер завершает текущую транзакцию на шине USB и затем останавливается. Бит HCH указывает, когда хост-контроллер завершил транзакцию и был остановлен. Хост-контроллер сбрасывает этот бит при возникновении следующих фатальных ошибок: сбой проверки соответствия, ошибки шины AHB.
–	31-22, 15-8	Зарезервировано

UHCIEFRNUM – регистр разрешения прерываний и количества кадров UHC USB

Смещение: + 804h
Сброс: 0000_0000h

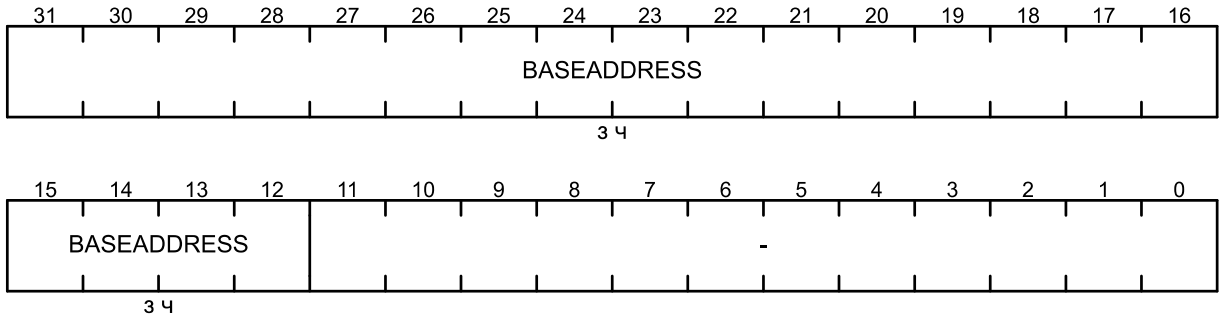


Поле	Биты	Описание	
FRNUM	26-16	Номер кадра/текущий индекс в списке кадров. Биты [26:16] определяют значение номера кадра и используются в SOF-пакете кадра. Значение в поле увеличивается в конце каждого кадра (примерно каждую 1 мс). Кроме того, биты [25:16] используются как текущий индекс в списке кадров.	
SPIE	3	Разрешение прерывания при коротком пакете.	
		0	Запрещено
		1	Разрешено
IOCE	2	Разрешение прерывания при завершении транзакции.	
		0	Запрещено
		1	Разрешено
RESUMEIE	1	Разрешение прерывания при возобновлении работы.	
		0	Запрещено
		1	Разрешено
TIMCRCIE	0	Разрешение прерывания по таймауту/CRC.	
		0	Запрещено
		1	Разрешено
–	31-27, 15-4	Зарезервировано	

Примечание – Источники прерываний, которые отключены в этом регистре, по-прежнему отображаются в регистре состояния, что позволяет программно опрашивать их состояние.

FRBASEADD – регистр базового адреса списка кадров UHC USB

Смещение: + 808h
Сброс: 0000_0000h

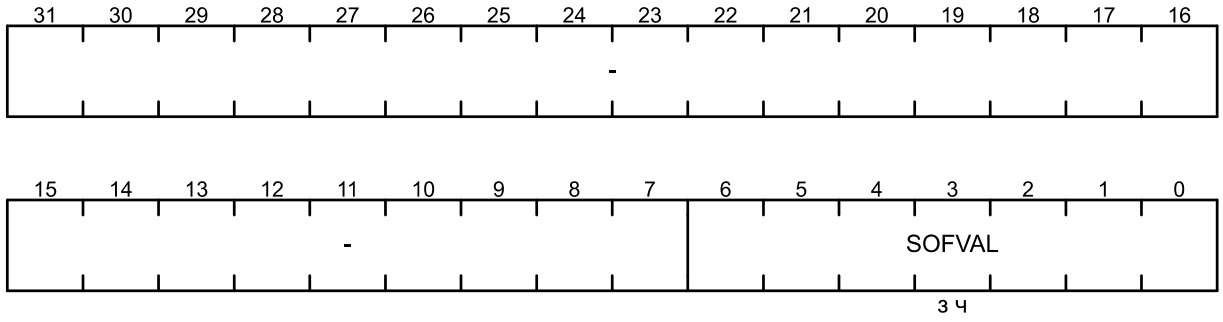


Поле	Биты	Описание
BASEADDRESS	31-12	Начальный адрес списка кадров в системной памяти. Содержимое данного поля объединяется со счетчиком номера кадра, что позволяет хост-контроллеру последовательно перемещаться по списку кадров.
–	11-0	Зарезервировано

Примечание – Младшие 12 бит имеют нулевое значение, для выравнивания регистра по границе 4 Кбайт.

SOFMOD – регистр изменения времени начала кадра UHC USB

Смещение: + 80Ch
Сброс: 0000_0040h

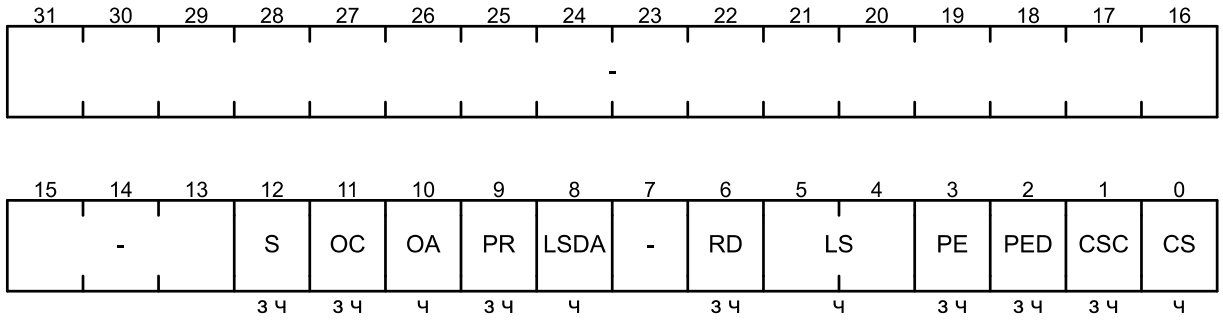


Поле	Биты	Описание
SOFVAL	6-0	Значение корректировки SOF. Время цикла SOF (количество тактовых периодов счетчика SOF для генерации длины кадра SOF) равно 11936 + значение в этом поле. Значение по умолчанию – 64. При записи нового значения, оно вступает в силу с начала следующего кадра.
–	31-7	Зарезервировано

Примечание – Данный регистр сбрасывается при сбросе хост-контроллера или глобальном сбросе.

UHCPORTSC – регистр статуса и управления порта UHC USB

Смещение: + 810h
Сброс: 0000_0080h



Поле	Биты	Описание
S	12	Приостановка. Бит не должен устанавливаться, если активен режим глобальной приостановки (установлен бит EGSM регистра UHCCMDSTS). В стостоянии приостановки передача данных на порту блокируется, за исключением глобального сброса и сброса порта. Если в момент записи бита была активна транзакция, то порт будет приостановлен по завершению текущей транзакции, поэтому может возникнуть задержка в приостановке порта. В состоянии приостановки порт способен обнаруживать команду возобновления.
OC	11	Бит изменения состояния перегрузки по току. Устанавливается, если происходит изменение бита OA. По сути, этот бит фиксирует любое изменение состояния перегрузки, в то время как бит OA сообщает только о текущем состоянии перегрузки. Данный бит можно сбросить программно, записав в него единицу.
OA	10	Флаг активной перегрузки по току. Устанавливается при обнаружении события перегрузки по току на USB-порту. Данный бит сам очищается при пропадании перегрузки.
PR	9	Бит сброса порта. В состоянии сброса порт отключается и отправляет сигнал сброса по шине USB.
LSDA	8	Флаг подключенного низкоскоростного устройства
RD	6	Бит обнаружения режима возобновления. Программно устанавливается для запуска сигнала возобновления. Хост-контроллер устанавливает этот бит, если обнаружен переход от состояния J к состоянию K, в тот момент, когда порт находится в состоянии приостановки.
LS	5-4	Биты статуса линии. Данные биты отражают логические уровни сигнальных линий D+ (бит 4) и D- (бит 5).
PE	3	Бит изменения состояния включения/выключения порта. Очищается записью единицы.

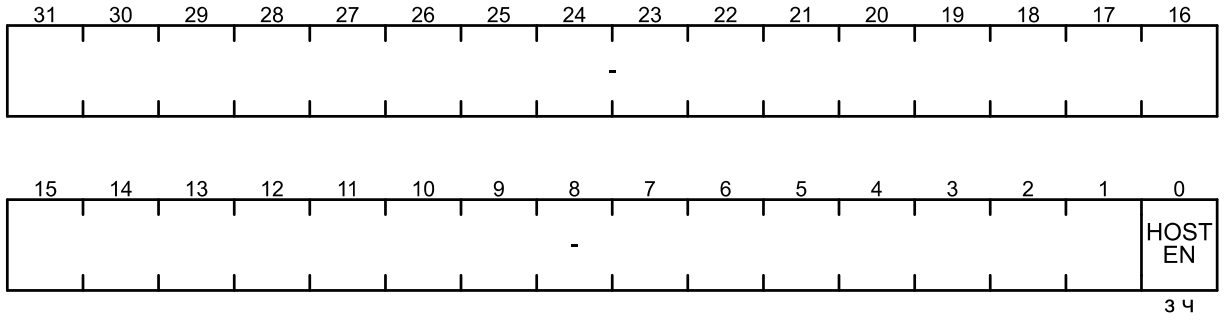
Поле	Биты	Описание
PED	2	Бит включения/выключения порта. Порт может быть включен только программно. Отключение порта может быть как аппаратным (из-за неисправности), так и программным. Поскольку статус бита не изменяется до тех пор, пока состояние порта фактически не изменится, может возникнуть задержка при отключении или включении порта, если в данный момент активна транзакция на USB-шине.
CSC	1	Бит изменения статуса подключения. Указывает на изменение текущего статуса подключения порта. Очищается записью единицы.
CS	0	Флаг текущего статуса подключения. Установленный бит означает, что к порту в данный момент подключено устройство. Значение бита отражает текущее состояние порта и может не соответствовать напрямую событию, вызвавшему установку бита изменения статуса подключения (бит CSC).
–	31-13, 7	Зарезервировано

A.30 Регистры управления USBCTR (PHY)

Базовый адрес: 4108_0800h Регистры блока USBCTR0 (PHY)
4109_0800h Регистры блока USBCTR1 (PHY)

HOSTEN – регистр включения хоста

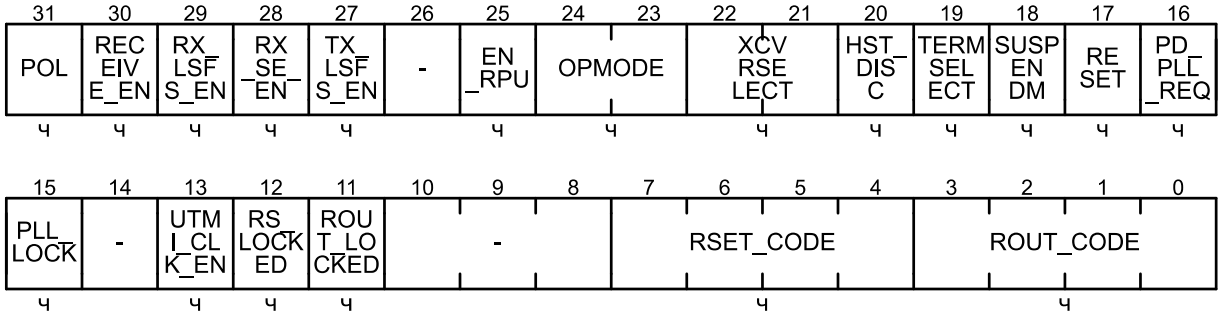
Смещение: + 00h
Сброс: 0h



Поле	Биты	Описание
HOSTEN	0	Конфигурирование USB-выхода в качестве хоста
–	31-1	Зарезервировано

PHYSTAT – регистр статуса PHY

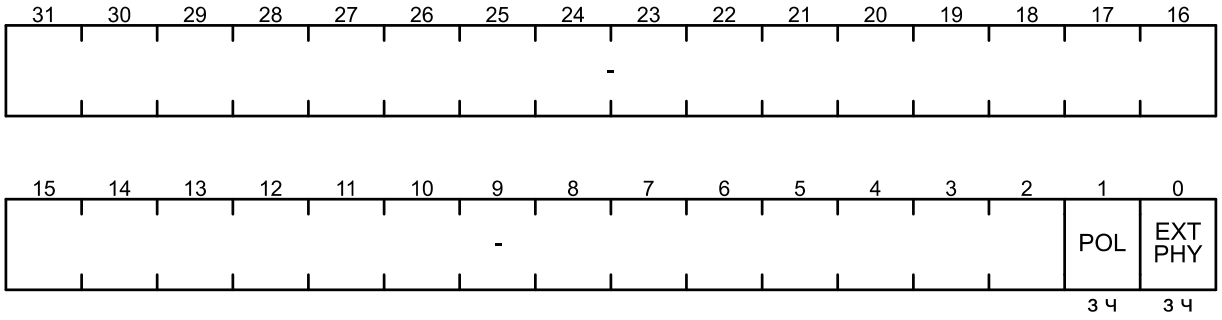
Смещение: + 04h
Сброс: 0h



Поле	Биты	Описание
POL	31	Состояние на входе POLARITY
RECEIVE_EN	30	Состояние приемника в цифровой части:
		0 заблокирован на время передачи пакета
		1 разблокирован
RX_LSFS_EN	29	Текущее состояние сигнала RX_LSFS_EN
RX_SE_EN	28	Текущее состояние сигнала RX_SE_EN
TX_LSFS_EN	27	Текущее состояние сигнала TX_LSFS_EN
EN_RPU	25	Текущее состояние сигнала EN_RPU
OPMODE	24-23	Текущее состояние сигнала OpMode[1:0]
XCVRSELECT	22-21	Текущее состояние сигнала XcvrSelect[1:0]
HST_DISC	20	Текущее состояние сигнала HostDisconnect
TERMSELECT	19	Текущее состояние сигнала TermSelect
SUSPENDM	18	Текущее состояние сигнала SuspendM
RESET	17	Текущее состояние сигнала Reset
PD_PLL_REQ	16	Текущее состояние сигнала PD_pll_req
PLL_LOCK	15	Текущее состояние сигнала PllLock
UTMI_CLK_EN	13	Признак выдачи тактового сигнала CLK на интерфейс UTMI
RS_LOCKED	12	Состояние на выходе rsource_locked
ROUT_LOCKED	11	Состояние на выходе rout_locked
RSET_CODE	7-4	Текущий код подстройки задающего сопротивления в источнике тока
ROUT_CODE	3-0	Текущий код подстройки выходного сопротивления передатчика
–	26, 14, 10-8	Зарезервировано

PHYCFG0 – регистр 0 конфигурации PHY

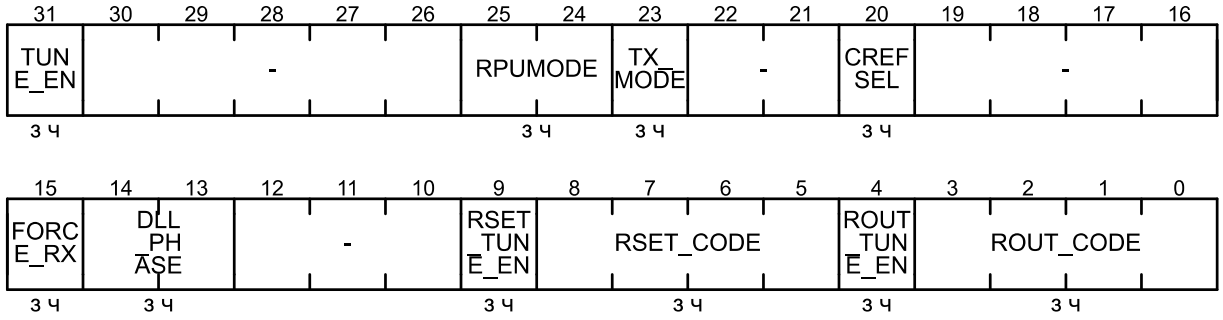
Смещение: + 08h
Сброс: 0h



Поле	Биты	Описание	
POL	1	Смена функционального назначения выводов DP и DM:	
		0	функционируют в соответствии с обозначением;
		1	DP функционирует в качестве DM, а DM в качестве DP.
EXTPHY	0	Устанавливается при использовании внешнего устройства USB PHY	
–	31-2	Зарезервировано	

PHYCFG1 – регистр 1 конфигурации PHY

Смещение: + 0Ch
Сброс: 0h



Поле	Биты	Описание	
TUNE_EN	31	Разрешение подстройки.	
		0	Значение по умолчанию
		1	используются значения на шине cfg_bus

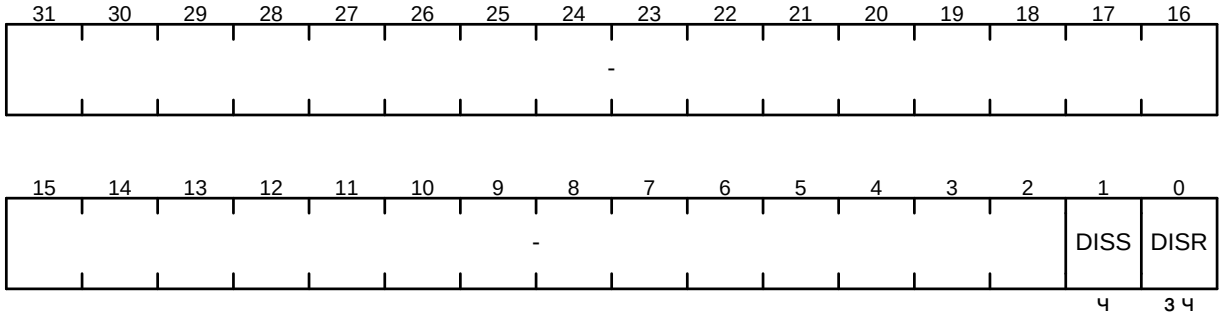
Поле	Биты	Описание	
RPUMODE	25-24	Режим компенсации RPU в передатчике:	
		00b	по умолчанию
		01b	зарезервировано;
		10b	на входе TX_RPU_COMP_EN всегда 0;
		11b	на входе TX_RPU_COMP_EN всегда 1.
TX_MODE	23	Режим работы передатчика:	
		0	с отключением в паузе между пакетами;
		1	без отключения в паузе между пакетами.
CREFSEL	20	Выбор типа источника опорного тока:	
		0	с использованием внутреннего задающего резистора
		1	с использованием внешнего задающего резистора
FORCE_RX	15	Отладочный сигнал:	
		0	нормальный режим;
		1	не отключать приемник после приема пакета
DLL_PHASE	14-13	Точка захвата бита в режимах LS/FS при приеме:	
		00b	80% битового интервала;
		01b	40% битового интервала;
		10b	60% битового интервала;
		11b	100% битового интервала
RSET_TUNE_EN	9	Автоматическая калибровка задающего резистора в источнике тока:	
		0	включена;
		1	выключена.
RSET_CODE	8-5	Код ручной подстройки задающего резистора в источнике тока, используемый при установленном бите RSOURCE_TUNE_EN	
ROUT_TUNE_EN	4	Автоматическая калибровка выходного сопротивления передатчика:	
		0	включена;
		1	выключена.
ROUT_CODE	3-0	Код ручной подстройки выходного сопротивления передатчика, используемый при установленном бите ROUT_TUNE_EN	
–	30-26, 22-21, 19-16, 12-10	Зарезервировано	

A.31 Регистры контроллера интерфейса CAN 2.0B

Базовый адрес:	4101_0000h	Контроллер CAN
Смещение:	+ 100h (LIST)	Регистры свободного списка
	+ 140h (MSPND)	Регистры ожидающих обработки прерываний
	+ 180h (MSID)	Регистры индексов сообщений
	+ 200h (Node_0)	Регистры узла 0
	+ 300h (Node_1)	Регистры узла 1
	+ 400h (Node_2)	Регистры узла 2
	+ 500h (Node_3)	Регистры узла 3
	+ 1000h + 20h*m (Msg_m)	Регистры объекта сообщений m
Мнемоника:	Node_n; Msg_m	
Примечание:	ln – номер списка от 0 до 7; n – номер узла 0, 1, 2 или 3; m – номер объекта сообщения от 0 до 255; mo – номер объекта сообщения в шестнадцатеричном формате от 0h до FFh (что соответствует диапазону от 0 до 255); pn – номер регистра ожидающих обработки прерываний от 0 до 7.	

CLC – регистр управления частотой

Смещение: + 00h
Сброс: 3h

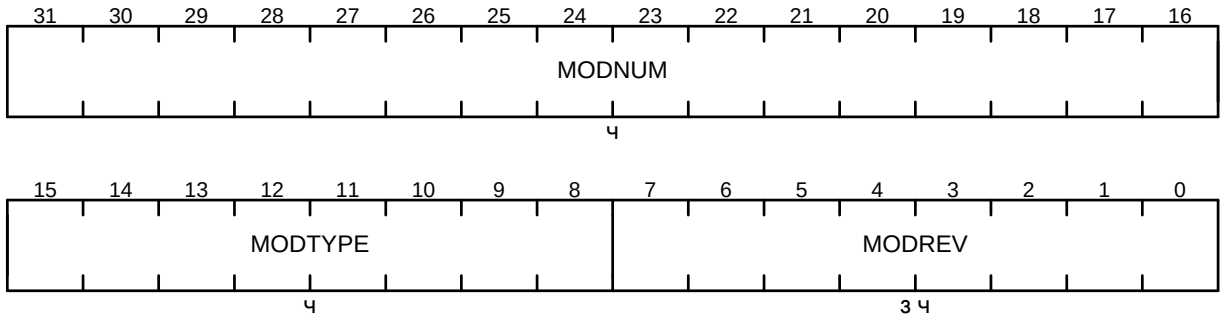


Поле	Биты	Описание	
DISS	1	Бит состояния контроллера CAN	
		0	Включен
		1	Выключен
DISR	0	Бит выключения контроллера CAN	
		0	Запрос отсутствует
		1	Запрос активен
–	31-2	Зарезервировано	

Примечание – Когда контроллер CAN находится в выключенном состоянии, только регистр CLC доступен для записи и чтения, доступ к остальным регистрам не возможен.

ID – регистр идентификации

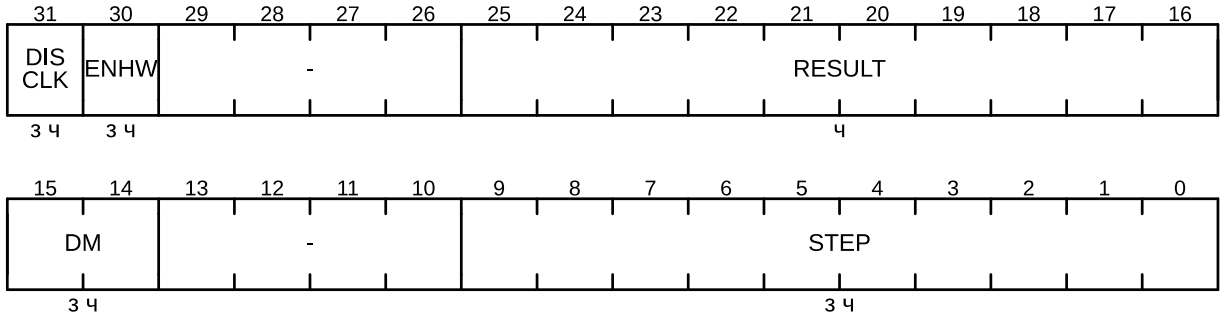
Смещение: + 08h
Сброс: 2B_C051h



Поле	Биты	Описание
MODNUM	31-16	Идентификационный номер контроллера CAN
MODTYPE	15-8	Разрядность контроллера CAN (C0h – 32-битная разрядность)
MODREV	7-0	Число модификаций контроллера CAN

FDR – регистр делителя

Смещение: + 0Ch
Сброс: 0h

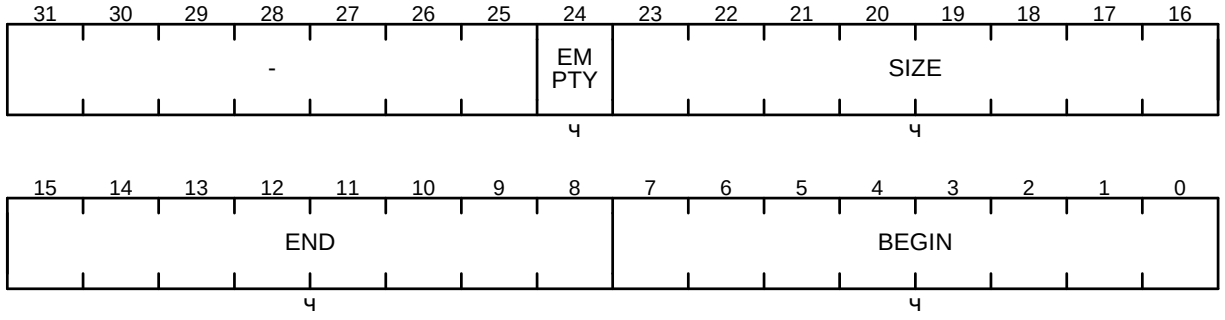


Поле	Биты	Описание	
DISCLK	31	Бит запрета внутреннего тактирования	
		0	Генерирование сигнала FOUT разрешено установленным значением поля DM
		1	Генерирование сигнала FOUT запрещено
ENHW	30	Бит контроля синхронизации. Это бит аппаратно удерживается в сброшенном состоянии и не может быть установлен	
RESULT	25-16	Результат. В нормальном режиме данное поле хранит значение инкрементирующегося на 1 счетчика, который при переполнении загружается значением STEP. В режиме дробного делителя поле содержит результат суммирования предыдущего значения RESULT со значением STEP	

Поле	Биты	Описание	
DM	15-14	Поле задания режима делителя частоты	
		00b	Счетчик выключен
		01b	Нормальный режим работы. Синхросигнал FOUT формируется. Сигнал сброса внешнего делителя в состоянии логического нуля. При активации режима поле RESULT загружается значением 3FFh. Далее периодически загружается значением из STEP
		10b	Режим дробного деления. Синхросигнал FOUT формируется. Сигнал сброса внешнего делителя в состоянии логического нуля. При активации режима поле RESULT загружается значением 3FFh. Далее периодически загружается значением из STEP
		11b	Синхросигнал FOUT не генерируется. Сигнал сброса внешнего делителя в состоянии логической единицы. Поле RESULT не меняется
STEP	9-0	Шаг делителя. В нормальном режиме поле хранит значение, которое загружается в RESULT при переполнении счетчика делителя. В режиме дробного деления значение данного поля прибавляется к RESULT каждый такт системной частоты	
–	29-26, 10	Зарезервировано	

LIST – массив регистров свободного списка

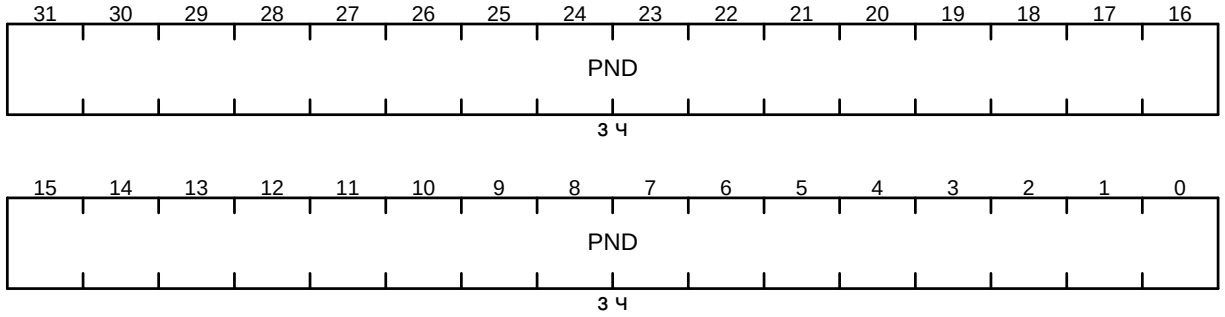
Смещение: LIST + 4×ln
Сброс: FF_FF00h (для списка 0), 100_0000h (для списков 1 – 7)



Поле	Биты	Описание	
EMPTY	24	Индикатор пустого списка	
		0	В списке есть как минимум один элемент
		1	Список пуст
SIZE	23-16	Размер списка. Количество элементов (объектов сообщений) в списке. Значение поля SIZE всегда на единицу меньше числа элементов. Если список пуст, SIZE = 00h	
END	15-8	Номер объекта сообщения, находящегося последним в списке. Поле может принимать значения от 00h до FFh, согласно количеству объектов сообщений (256)	
BEGIN	7-0	Номер объекта сообщения, находящегося первым в списке. Поле может принимать значения от 00h до FFh, согласно количеству объектов сообщений	
–	31-25	Зарезервировано	

MSPND – массив регистров ожидающих обработки прерываний

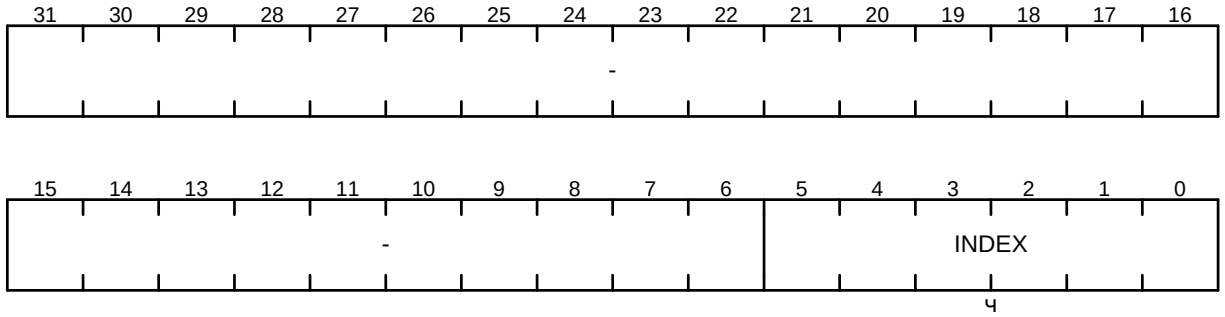
Смещение: MSPND + 4pn
Сброс: 0h



Поле	Биты	Описание
PND	31-0	Поле ожидающих обработки сообщений. При возникновении прерывания, объект сообщения устанавливает бит в одном из регистров MSPND. Позиция бита в регистре определяется битами 4-0 поля MPN регистра MOIPR. Выбор регистра MSPND определяется битами 7-5 поля MPN регистра MOIPR. Биты устанавливаются только аппаратно. Установленные биты сбрасываются программно записью нуля. Запись единицы не имеет эффекта.

MSID – массив регистров индекса сообщения

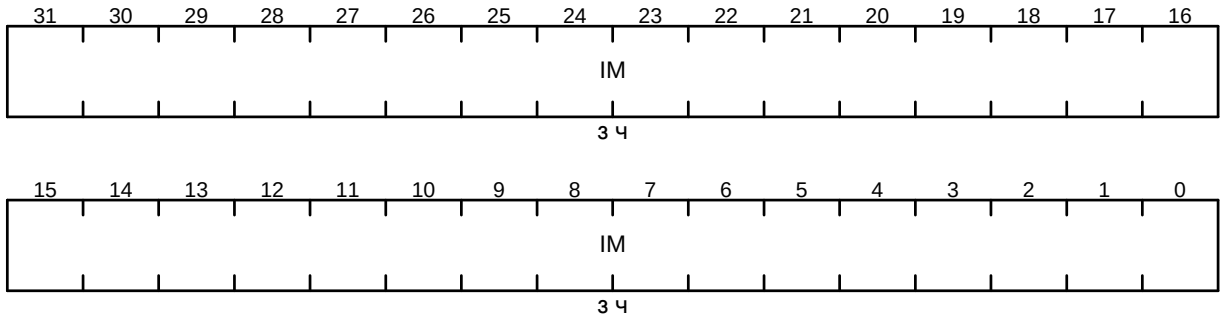
Смещение: MSID + 4×pn
Сброс: 20h



Поле	Биты	Описание
INDEX	5-0	Поле номера ожидающего обработки бита. Если в регистре MSPND есть установленные биты, которые не маскируются соответствующими битами регистра MSIMASK, то поле INDEX будет указывать на самый младший из них. Если в регистре MSPND нет установленных битов или они замаскированы, то в поле INDEX будет находиться значение 20h
–	31-6	Зарезервировано

MSIMASK – регистр маски индекса сообщения

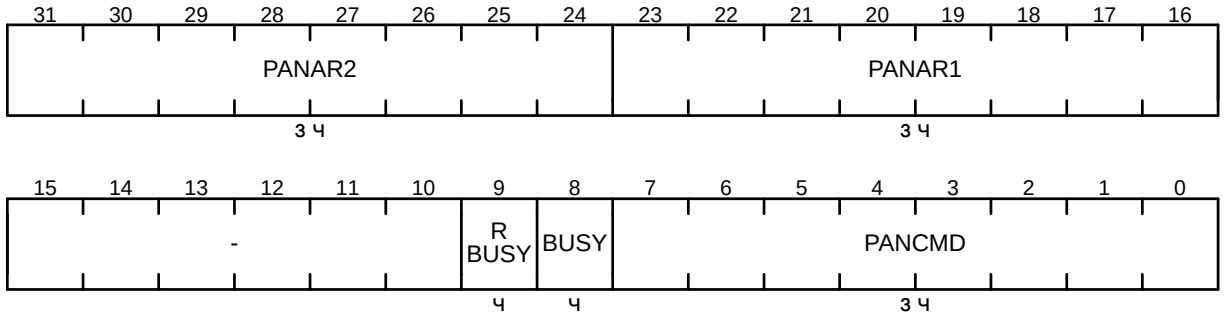
Смещение: + 1C0h
Сброс: 0h



Поле	Биты	Описание
IM	31-0	Маска для битов ожидающих обработки сообщений. Учитывается состояние только тех бит регистров MSPND, для которых в поле IM установлены соответствующие биты

PANCTR – регистр панели команд

Смещение: + 1C4h
Сброс: 301h



Поле	Биты	Описание	
PANAR2	31-24	Второй аргумент команды (см. таблицу А.31.1)	
PANAR1	23-16	Первый аргумент команды (см. таблицу А.31.1)	
RBUSY	9	Флаг занятости результата команды	
		0	Контроллером списков не запланировано обновление полей PANAR1 и PANAR2
		1	Запущена команда списка, итог выполнения которой будет записан в поля PANAR1 и PANAR2, но сам результат еще не готов
BUSY	8	Флаг занятости панели команд	
		0	Панель завершила выполнение команды и готова к записи новой
		1	Панель находится в процессе выполнения операции
PANCMD	7-0	Поле команды (см. таблицу А.31.1). После выполнения команды в это поле автоматически заносится значение 00h (нет операции)	
–	15-10	Зарезервировано	

Таблица А.31.1 – Коды команд работы со списками

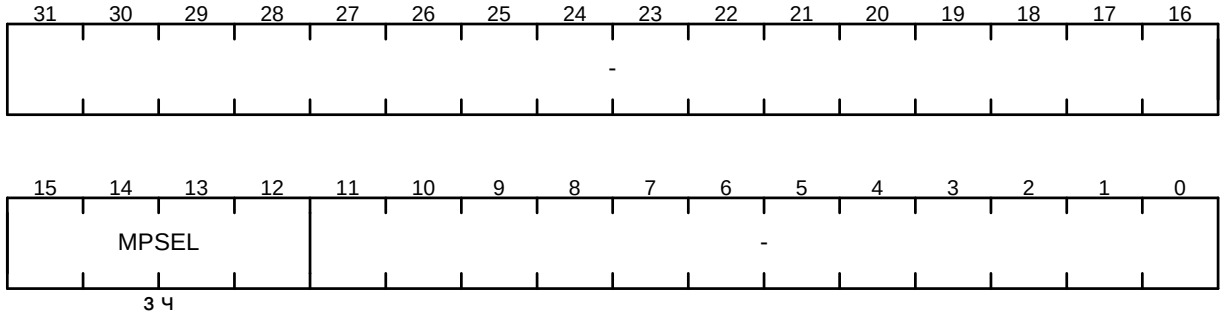
Поле PANCMD	Поле PANAR2	Поле PANAR1	Описание команды
00h	–	–	Нет операции. Никаких действий не выполняется
01h	Результат: бит 7 – ошибка, биты 6-0 – не определены	–	Инициализация списков. Запуск инициализации для очистки битовых полей CTRL и LIST всех объектов сообщений. Регистры LIST0 – LIST7 устанавливаются в свои значения после сброса. Это приводит к переносу всех объектов сообщений в список № 0 (список нераспределенных объектов сообщений). Инициализация списков требует, чтобы биты INIT и CCE регистра NCR были установлены для всех узлов CAN. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – инициализация завершена успешно; - 1 – инициализация не выполнена, поскольку не все биты INIT и CCE были установлены. Команда инициализации списков автоматически запускается при каждом сбросе контроллера CAN, за исключением случая, когда все регистры объектов сообщений уже сброшены
02h	Аргумент: номер списка	Аргумент: номер объекта сообщения	Статическое распределение – размещение объекта сообщения в список. Объект сообщения переносится из текущего списка в конец списка, номер которого указан в поле PANAR2. Эта команда также используется для освобождения объекта сообщения, т. е. переноса его в список № 0 (если поле PANAR2 равно 00h)
03h	Аргумент: номер списка. Результат: бит 7 – ошибка, биты 6-0 – не определены	Результат: номер объекта сообщения	Динамическое распределение – размещение первого объекта сообщения списка №0 в конец списка, номер которого указан в поле PANAR2. Номер объекта сообщения возвращается полем PANAR1. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – операция выполнена; - 1 – операция не выполнена – список № 0 пуст
04h	Аргумент: номер объекта назначения	Аргумент: номер переноси-мого объекта сообщения	Статическая вставка перед объектом сообщения – перенос объекта сообщения с номером PANAR1 из текущего списка в список, к которому принадлежит объект сообщения с номером PANAR2, в позицию перед объектом назначения

Продолжение таблицы А.31.1

Поле PANCMD	Поле PANAR2	Поле PANAR1	Описание команды
05h	Аргумент: номер объекта назначения Результат: бит 7 – ошибка, биты 6-0 – не определены	Результат: номер добавленного объекта сообщения	Динамическая вставка перед объектом сообщения – перемещение первого объекта сообщения списка №0 в список, к которому принадлежит объект сообщения с номером PANAR2, в позицию перед объектом назначения. Номер добавленного объекта сообщения возвращается полем PANAR1. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – операция выполнена; - 1 – операция не выполнена – список № 0 пуст
06h	Аргумент: номер объекта назначения	Аргумент: номер переноси-мого объекта сообщения	Статическая вставка позади объекта сообщения – перенос объекта сообщения с номером PANAR1 из текущего списка в список, к которому принадлежит объект сообщения с номером PANAR2, в позицию после объекта назначения
07h	Аргумент: номер объекта назначения Результат: бит 7 – ошибка, биты 6-0 – не определены	Результат: номер добавленного объекта сообщения	Динамическая вставка позади объекта сообщения – перемещение первого объекта сообщения списка №0 в список, к которому принадлежит объект сообщения с номером PANAR2, в позицию после объекта назначения. Номер добавленного объекта сообщения возвращается полем PANAR1. Бит 7 поля PANAR2 сигнализирует о результате операции: - 0 – операция выполнена; - 1 – операция не выполнена – список № 0 пуст
08h – FFh	–	–	Зарезервировано

MCR – регистр управления

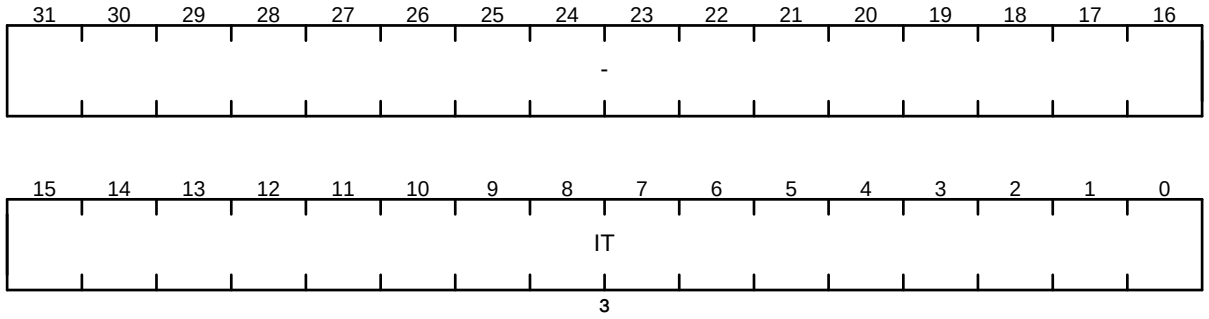
Смещение: + 1C8h
Сброс: 0h



Поле	Биты	Описание
MPSEL	15-12	Селектор, который позволяет выбирать позицию бита ожидающего обработки сообщения после приёма/передачи, используя комбинацию битовых полей RXINP, TXINP и MPN регистра MOIPRn. Пояснения приведены на рисунках 38.26 и 38.27
–	31-16, 11-0	Зарезервировано

MITR – регистр генерации прерываний

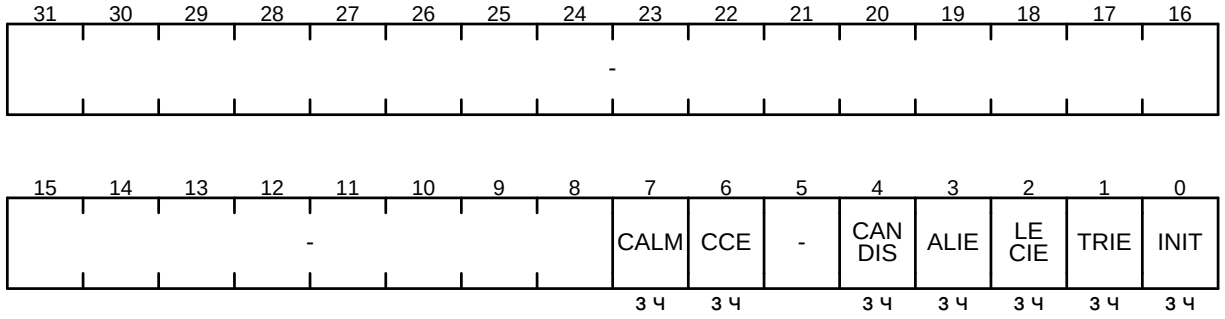
Смещение: + 1CCh
Сброс: 0h



Поле	Биты	Описание
IT	15-0	Поле генератора прерываний. Каждый бит поля связан с одной из линий прерываний. Номера битов от 0 до 15 соответствуют номерам линий прерываний. Для того, чтобы программно сгенерировать одно или несколько прерываний, следует установить соответствующие биты. Всегда считывается как 0
–	31-16	Зарезервировано

NCR – регистр управления узла

Смещение: Node_n + 00h
Сброс: 1h

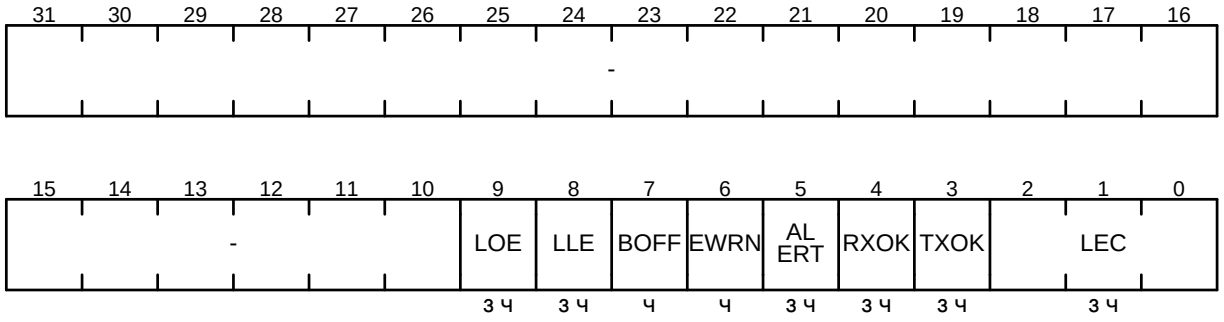


Поле	Биты	Описание	
CALM	7	Бит включения режима анализа узла	
		0	Режим выключен
		1	Установка бита включает режим анализа узла. В этом режиме сообщения могут только приниматься, бит подтверждения не посылается после успешного приема сообщения, флаг активной ошибки посылается рецессивным вместо доминантного. На линии отправки сообщений поддерживается высокий уровень сигнала
		Бит может быть установлен только, если установлен бит INIT	
CCE	6	Бит разрешения изменения конфигурации узла. Управляет доступом к регистрам NBTR, NPCR и NECNT	
		0	Только чтение
		1	Полный доступ
CANDIS	4	Бит выключения узла	
		0	Сброс бита включает узел
		1	Установка бита выключает узел. Сначала узел переходит в состояние «простоя» или «отключен от шины», далее аппаратно устанавливается бит INIT, и, если разрешено, генерируется прерывание ALERT
ALIE	3	Бит разрешения прерывания ALERT от узла	
		0	Запрещено
		1	Разрешено
LECIE	2	Бит разрешения прерывания от узла при обновлении кода последней ошибки, когда значение LEC регистра NSR не равно 0	
		0	Запрещено
		1	Разрешено
TRIE	1	Бит разрешения прерывания от узла по окончании передачи/приема	
		0	Запрещено
		1	Разрешено

Поле	Биты	Описание	
INIT	0	Инициализация узла. Данный бит автоматически устанавливается при переходе узла в состояние «отключен от шины»	
		0	Сброс бита разрешает участие узла в работе CAN шины. Узел ожидает последовательность из 11 рецессивных бит на шине и включается в трафик. Если на момент сброса бита INIT узел находился в состоянии «отключен от шины», начинается процесс выхода из этого состояния в следующем порядке – получение 128 последовательностей из 11 идущих подряд рецессивных бит, выход из состояния «отключен от шины», включение в трафик
		1	Установка бита INIT прекращает участие узла в трафике. Все текущие передачи останавливаются, линии передач переходят в рецессивное состояние. Если на момент установки бита INIT узел находился в состоянии «отключен от шины», процесс выхода из этого состояния продолжается до его завершения. Далее узел остается неактивным до тех пор, пока установлен бит INIT
–	31-8, 5	Зарезервировано	

NSR – регистр состояния узла

Смещение: Node_n + 04h
Сброс: 0h

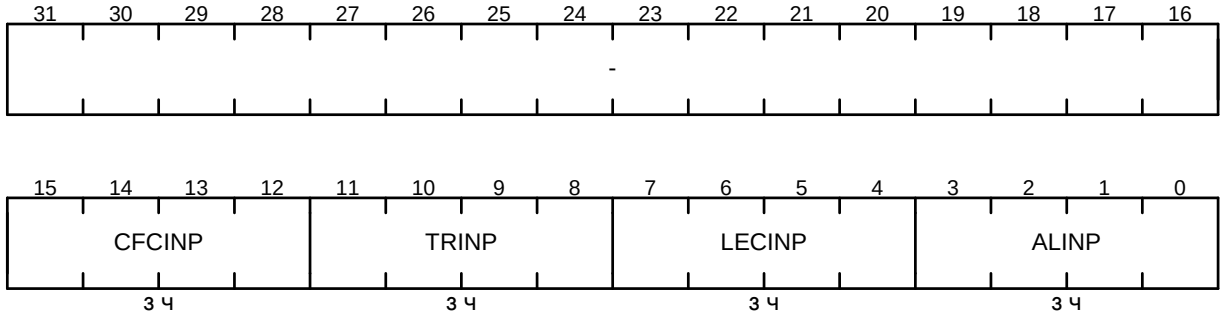


Поле	Биты	Описание	
LOE	9	Флаг ошибки номера списка	
		0	Ошибок не обнаружено
		1	Обнаружена ошибка при фильтрации принимаемого сообщения. В регистре MOSTAT объекта сообщения обнаружен неверный номер списка
		Бит должен сбрасываться программно записью нуля	
LLE	8	Флаг ошибки количества элементов списка	
		0	Ошибок не обнаружено
		1	Обнаружена ошибка при фильтрации принимаемого сообщения. Количество элементов списка, принадлежащего узлу, отличается от указанного в поле SIZE соответствующего регистра списка
		Бит должен сбрасываться программно записью нуля	
BOFF	7	Флаг состояния «отключен от шины»	
		0	Узел не находится в состоянии «отключен от шины»
		1	Узел находится в состоянии «отключен от шины»
EWRN	6	Флаг превышения порогового значения ошибок	
		0	Лимит ошибок еще не достигнут
		1	По крайней мере, один из счетчиков ошибок (REC, TEC) достиг лимита ошибок, заданного полем EWRNLVL регистра NECNT узла
ALERT	5	Флаг предупреждения ALERT	
		0	Нет событий
		1	Произошло одно или несколько не взаимоисключающих событий: - изменение значения бита BOFF; - изменение значения бита EWRN - установка бита LOE; - установка бита LLE; - аппаратная установка бита INIT
		Бит должен сбрасываться программно записью нуля	
RXOK	4	Флаг успешного приема сообщения	
		0	Полученных сообщений нет
		1	Сообщение получено
		Бит должен сбрасываться программно записью нуля	
TXOK	3	Флаг успешной передачи сообщения	
		0	Переданных сообщений нет
		1	Сообщение передано без ошибок с получением подтверждения
		Бит должен сбрасываться программно записью нуля	

Поле	Биты	Описание	
LEC	2-0	Поле кода последней из обнаруженных ошибок работы узла	
		000	Ошибок нет
		001	Ошибка стаффинга (STUFF ERROR). Может быть обнаружена во время передачи шестого бита из последовательности шести одинаковых бит в поле сообщения, которое должно быть кодировано методом разрядного заполнения (заключается в том, что после передачи пяти битов одинаковой полярности, шестой бит должен иметь противоположную полярность и вставляться передатчиком в поток данных автоматически, приемник пропускает этот бит)
		010	Ошибка формата (FORM ERROR). Ошибка формата обнаруживается, если битовое поле фиксированного формата содержит один или несколько недопустимых битов. Исключениями являются получение приемником доминантного бита в течение последнего бита поля «конец кадра» и обнаружение любым узлом доминантного значения на месте последнего бита разделителя ошибки либо разделителя перегрузки
		011	Ошибка подтверждения (ACKNOWLEGMENT ERROR). Обнаруживается передатчиком всякий раз, когда он не обнаруживает доминантный бит АСК в «области подтверждения»
		100	Ошибка символа при передаче рецессивного бита (BIT 1 ERROR). Узел, передающий рецессивный «1» бит, обнаружил на шине доминантный «0» бит, исключая биты поля арбитража, бит АСК в «области подтверждения», передачу пассивного флага ошибки
		101	Ошибка символа при передаче доминантного бита (BIT 0 ERROR). Данный код устанавливается в двух случаях: - во время передачи сообщения (или бита подтверждения, флага активной ошибки, флага перезагрузки), узел передает доминантный бит «0», но на шине обнаруживается рецессивный «1» бит; - во время выхода из состояния «отключен от шины» при каждом обнаружении последовательности из 11 рецессивных битов. ЦП может использовать этот код как указание на то, что шина не находится под постоянным воздействием помех
		110	Ошибка циклического избыточного кода (CRC ERROR). Передатчик по установленному алгоритму вычисляет значение контрольной суммы (CRC) для передаваемых данных и вставляет ее в сообщение. Приемник, после получения данных, вычисляет CRC по тому же алгоритму, что и передатчик, и сравнивает вычисленное значение с принятым значением. В случае несовпадения фиксируется ошибка
		111	Разрешенный для программной записи код
		Поле LEC обновляется аппаратно после каждой передачи (успешной или с ошибкой) – в нём устанавливается код ошибки, либо 000b, если её не было. Прерывание по LEC происходит каждый раз при обнаружении ошибки при условии, что оно разрешено битом LECIE регистра NCR. Программно в поле можно записать только 111b, если требуется. Это значение будет находиться в поле до завершения текущей передачи	
–	31-10	Зарезервировано	

NIPR – регистр указателя прерываний узла

Смещение: Node_n + 08h
Сброс: 0h

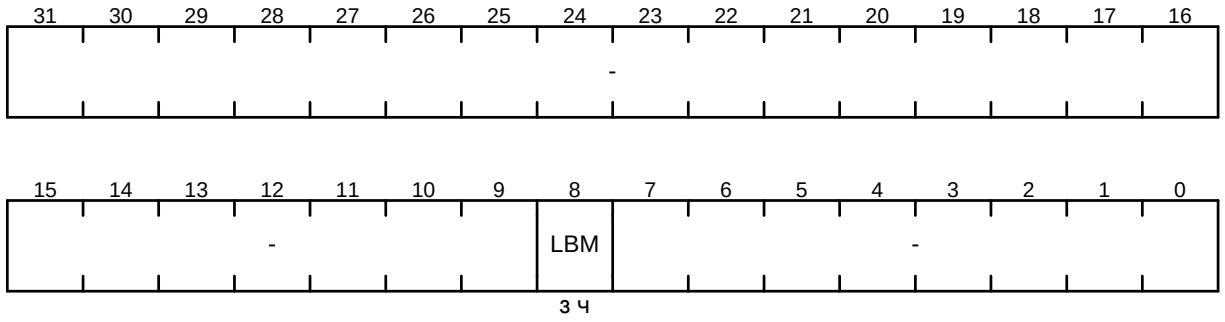


Поле	Биты	Описание
CFCINP	15-12	Указатель для прерывания при переполнении счетчика кадров узла
TRINP	11-8	Указатель для прерывания по окончании передачи/приема сообщения
LECINP	7-4	Указатель для прерывания при записи кода последней ошибки
ALINP	3-0	Указатель для прерывания ALERT
–	31-16	Зарезервировано

Примечание – Каждый из указателей задает номер одной из 16 линий прерываний. Значение 0h соответствует нулевой линии, значение 1h – линии 1 и так далее до Fh, которое соответствует линии 15.

NPCR – регистр управления портом узла

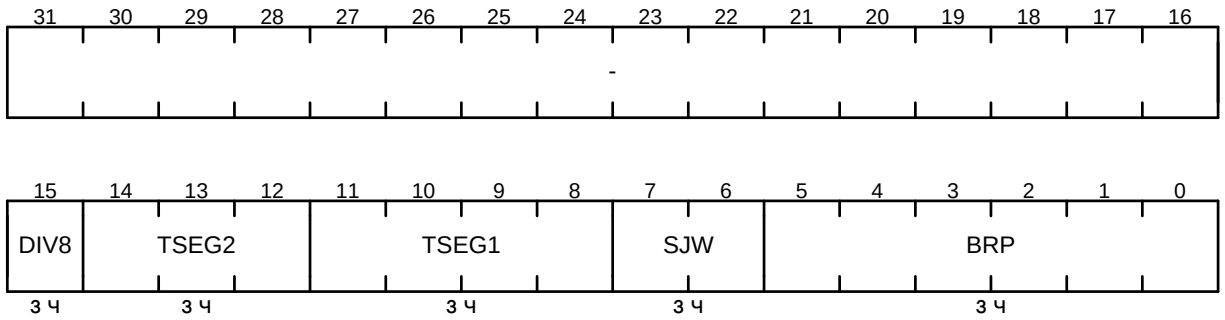
Смещение: Node_n + 0Ch
Сброс: 0h



Поле	Биты	Описание	
LBM	8	Бит включения режима обратной петли (Loop-Back)	
		0	Выключено
		1	Включено. В этом режиме узел подсоединяется к внутренней виртуальной CAN шине. Все узлы с включенным режимом обратной петли объединяются виртуальной CAN шиной и могут взаимодействовать друг с другом. При этом на внешних выводах узлов, соединенных с внешней физической CAN шиной, поддерживается рецессивный уровень сигнала, т. е. узлы не активны
–	31-9, 7-0	Зарезервировано	

NBTR – регистр синхронизации битов

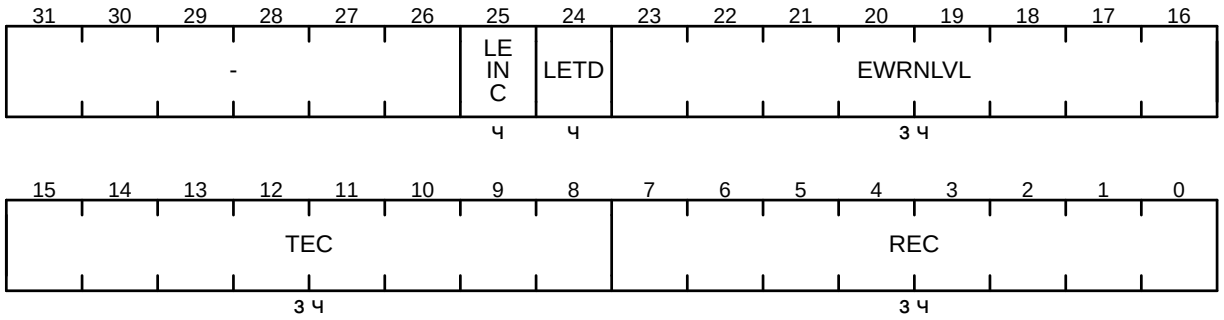
Смещение: Node_n + 10h
Сброс: 0h



Поле	Биты	Описание	
DIV8	15	Делитель частоты на восемь	
		0	Выключен
		1	Включен
TSEG2	14-12	Параметр 2. Временной промежуток от точки выборки до точки передачи, определяемый пользователем. Длительность сегмента равна tq × (TSEG2 + 1) и может быть уменьшена за счет ресинхронизации. Диапазон допустимых значений от 0h до 7h	
TSEG1	11-8	Параметр 1. Временной промежуток от сегмента синхронизации до точки выборки, определяемый пользователем и включающий в себя сегмент распространения. Длительность равна tq × (TSEG1 + 1) и может быть увеличена за счет ресинхронизации. Допустимые значения от 1h до Fh	
SJW	7-6	Ширина перехода повторной синхронизации. Длительность равна tq × (SJW + 1). Диапазон допустимых значений от 0h до значения TSEG2 включительно	
BRP	5-0	Предделитель скорости передачи. Длительность одного кванта времени (в тактах частоты): - (BRP + 1), если DIV8 = 0; - 8 × (BRP + 1), если DIV8 = 1	
–	31-16	Зарезервировано	

NECNT – регистр счетчика ошибок узла

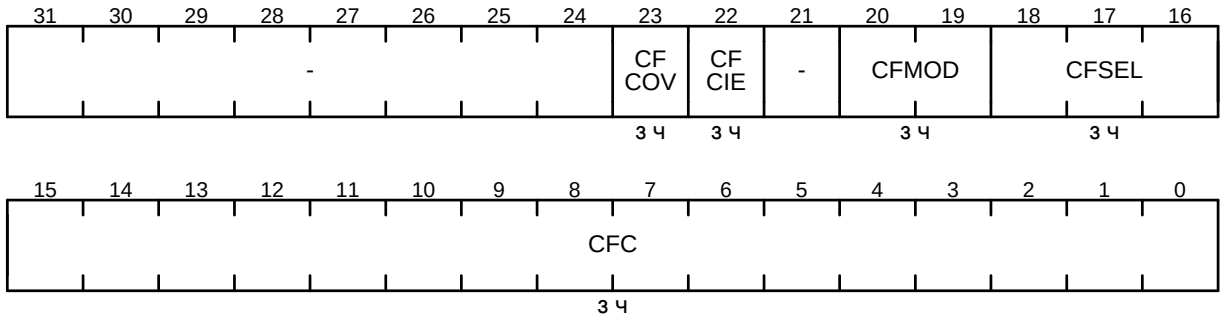
Смещение: Node_n + 14h
Сброс: 60_0000h



Поле	Биты	Описание	
LEINC	25	Индикатор инкрементирования при последней ошибке	
		0	Обнаруженная ошибка привела к инкрементированию счетчика ошибок на единицу
		1	Обнаруженная ошибка привела к инкрементированию счетчика ошибок на восемь
LETD	24	Флаг направления передачи при последней ошибке	
		0	Последняя ошибка обнаружена при приеме сообщения, произошло инкрементирование поля REC
		1	Последняя ошибка обнаружена при передаче сообщения, произошло инкрементирование поля TEC
EWRNLVL	23-16	Поле задания лимита ошибок, по достижении которого выставляется флаг EWRN в регистре NSR (по умолчанию, количество ошибок – 96)	
TEC	15-8	Поле счетчика ошибок передачи сообщений	
REC	7-0	Поле счетчика ошибок приема сообщений	
–	31-26	Зарезервировано	

NFCR – регистр счетчика сообщений узла

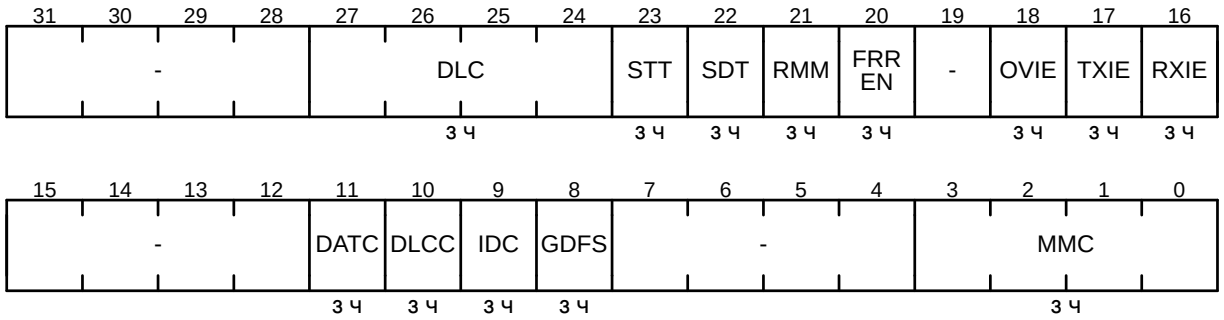
Смещение: Node_n + 18h
Сброс: 0h



Поле	Биты	Описание	
CFCOV	23	Флаг переполнения счетчика сообщений	
		0	Счетчик не переполнен
		1	Счетчик переполнился. В режиме синхросчетчика этот флаг устанавливается при изменении поля CFC, и, если установлен бит CFCIE, формируется прерывание
		Бит сбрасывается программно	
CFCIE	22	Бит разрешения прерывания от счетчика сообщений	
		0	Запрещено
		1	Разрешено
CFMOD	20-19	Режим работы счетчика сообщений	
		00	Счетчик сообщений инкрементируется после каждого успешного приема/передачи сообщения
		01-11	Зарезервировано. Не использовать!
CFSEL	18-16	Поле задания условия инкрементирования счетчика сообщений	
		**1	При получении сообщения, которое предназначено другому устройству в сети CAN (не соответствующий какому-либо объекту сообщения)
		1	При получении сообщения, имеющего соответствующий объект сообщения
		1**	При успешной отправке сообщения
		000	Зарезервировано. Не использовать!
		Состояние бита, отмеченного символом «*», не важно в данном контексте. Условия могут комбинироваться между собой (например, 110b или 101b)	
CFC	15-0	Поле счетчика сообщений. Хранит значение счетчика сообщений при CFMOD = 00b	
–	31-24, 21	Зарезервировано	

MOFCR – регистр управления функционированием объекта сообщения

Смещение: Msg_mo + 00h
Сброс: 0h



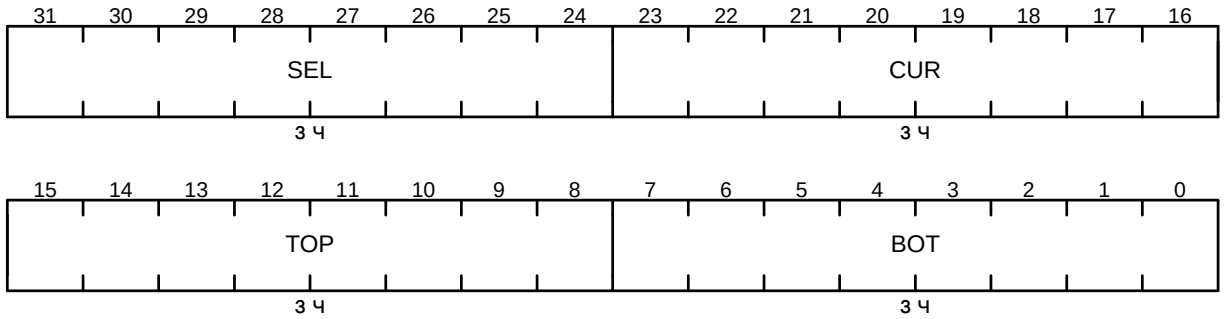
Поле	Биты	Описание	
DLC	27–24	Код длины данных. Показывает количество байт данных, находящихся в объекте сообщения. Диапазон – значение от 0 до 8. Если значение поля DLC больше 8, это автоматически указывает на 8 байт. Значение поля DLC полученного сообщения сохраняется таким, каким было получено	
STT	23	Бит задания однократной пересылки данных	
		0	Нет действий
		1	Если бит установлен, тогда бит TXRQ регистра MOSTAT сбрасывается после начала передачи объекта сообщения m. В связи с этим, в случае неудачной передачи, повторной передачи сообщения не будет
SDT	22	Бит задания однократного участия объекта сообщения m в пересылке	
		0	Нет действий
		1	Если объект сообщения m не является объектом FIFO, тогда бит MSGVAL регистра MOSTAT сбрасывается после успешного приема данных. Если объект сообщения m является базовым объектом FIFO, то бит MSGVAL сбрасывается, когда указатель на текущий объект CUR достигает значения SEL в регистре MOFGPR
RMM	21	Бит включения удаленного мониторинга объекта передачи	
		0	Выключено. Идентификатор, бит IDE и поле DLC объекта сообщения m остаются без изменений до получения корректного кадра удаленного запроса
		1	Включено. Идентификатор, бит IDE и поле DLC корректного кадра удаленного запроса копируются в объект передачи m для мониторинга входящего кадра удаленного запроса
		Состояние бита оказывает влияние только на объекты передач	
FRREN	20	Бит разрешения удаленного запроса. Определяет, будет ли устанавливаться бит TXRQ в объекте сообщения m или в другом объекте сообщения, на который указывает CUR	
		0	Бит TXRQ объекта сообщения m устанавливается после получения корректного кадра удаленного запроса
		1	Бит TXRQ другого объекта сообщения (на который указывает CUR) устанавливается после получения им корректного кадра удаленного запроса

Поле	Биты	Описание	
OVIE	18	Бит разрешения прерывания по заполнению FIFO объекта сообщения m. Прерывание генерируется, когда указатель CUR (указатель на текущий объект) достигает значения SEL регистра MOFGPR	
		0	Запрещено
		1	Разрешено
		Если объект сообщения m является объектом приема FIFO, то поле TXINP (регистр MOIPR объекта сообщения m) определяет, которая из линий прерываний используется. Если объект сообщения m является объектом передачи FIFO, то поле RXINP (регистр MOIPR) задает используемую линию прерывания. Для всех других режимов объекта сообщения состояние бита OVIE не важно	
TXIE	17	Бит разрешения прерывания по окончании передачи сообщения	
		0	Запрещено
		1	Разрешено. Прерывание генерируется, если сообщение из объекта сообщения m было успешно передано. Поле TXINP (регистр MOIPR) определяет, которая из линий прерывания используется
RXIE	16	Бит разрешения прерывания по окончании приема сообщения	
		0	Запрещено
		1	Разрешено. Прерывание генерируется, если сообщение было успешно принято объектом сообщения m (напрямую или через шлюз). Поле RXINP (регистр MOIPR) определяет, которая из линий прерывания используется
DATC	11	Индикатор копирования данных	
		0	Данные не копируются
		1	Данные в регистрах MODATAH и MODATAL объекта-источника шлюза (после сохранения принятого кадра в источнике) копируются через шлюз в объект-приемник
		Бит DATC используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует	
DLCC	10	Индикатор копирования кода длины данных DLC	
		0	Код не копируется
		1	Код длины данных объекта-источника шлюза (после сохранения принятого кадра в источнике) копируется через шлюз в объект-приемник
		Бит DLCC используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует	
IDC	9	Индикатор копирования идентификатора	
		0	Идентификатор не копируется
		1	Идентификатор объекта-источника шлюза (после сохранения принятого кадра в источнике) копируется через шлюз в объект-приемник
		Бит IDC используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует	
GDFS	8	Индикатор отправки кадра через шлюз	
		0	Состояние бита TXRQ объекта-приемника остается без изменений
		1	Бит TXRQ объекта-приемника устанавливается после внутренней передачи из объекта-источника
		Бит GDFS используется только объектом-источником в режиме шлюза. Во всех остальных случаях бит не функционирует	

Поле	Биты	Описание	
MMC	3-0	Задание режима объекта сообщения m	
		0000	Стандартный объект сообщения
		0001	Базовый объект приемной структуры FIFO
		0010	Базовый объект передающей структуры FIFO
		0011	Вспомогательный объект передающей структуры FIFO
		0100	Объект-источник шлюза
		Остальные комбинации зарезервированы	
–	31-28, 19, 15-12, 7-4	Зарезервировано	

MOFGPR – регистр указателя FIFO или шлюза объекта сообщения

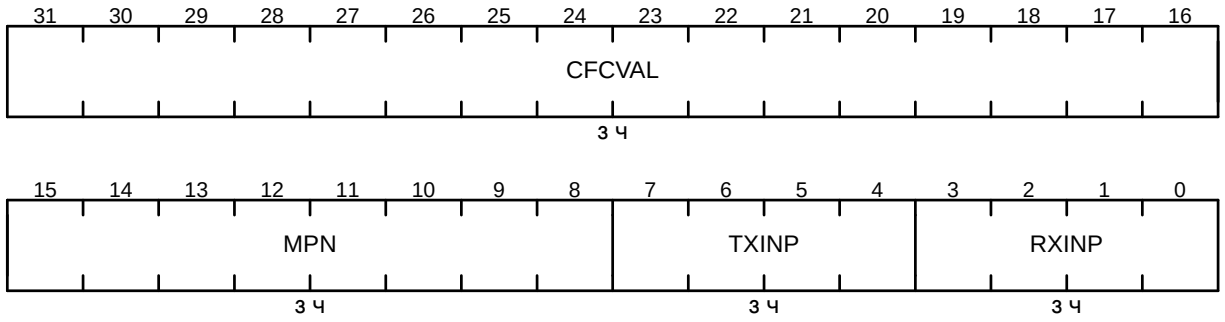
Смещение: Msg_mo + 04h
Сброс: 0h



Поле	Биты	Описание
SEL	31-24	Указатель объекта сообщения. Второй (программный) указатель в дополнение к аппаратному указателю CUR при работе с FIFO. Поле SEL используется для общего мониторинга (генерирование прерываний FIFO)
CUR	23-16	Указатель на текущий объект в пределах FIFO или шлюза. После каждой операции FIFO или передачи через шлюз указатель CUR обновляется – в него заносится номер следующего объекта сообщения в списке (поле PNEXT регистра MOSTAT) – до тех пор, пока не будет достигнут верхний элемент FIFO (поле TOP), после чего CUR сбрасывается, и в него загружается номер нижнего элемента списка (из поля BOT)
TOP	15-8	Указатель верхнего элемента FIFO. В поле находится номер последнего элемента
BOT	7-0	Указатель нижнего элемента FIFO. В поле находится номер первого элемента

MOIPR – регистр указателя прерываний объекта сообщения

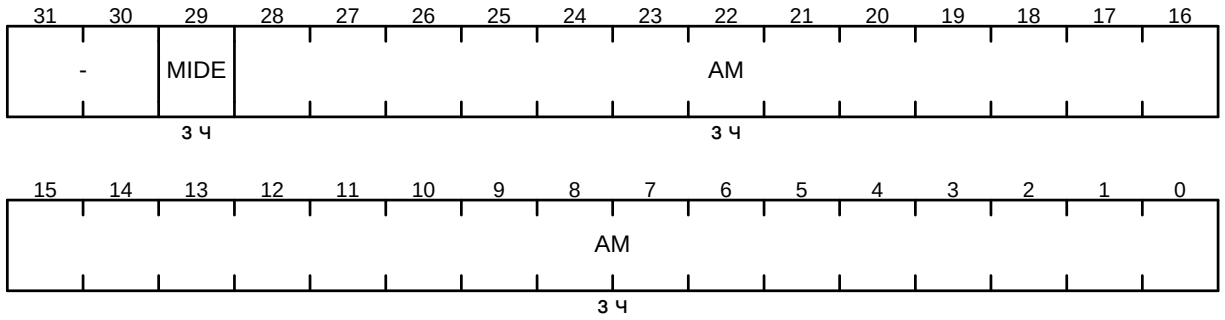
Смещение: Msg_mo + 08h
Сброс: 0h



Поле	Биты	Описание
CFCVAL	31-16	Количество кадров. Каждый раз после записи принятого сообщения в объект сообщения m или успешной передачи объекта сообщения m, значение счетчика кадров CFC (регистр NFCR) копируется в CFCVAL
MPN	15-8	Номер бита ожидающего обработки сообщения. Указывает позицию бита, соответствующего объекту сообщения m в регистре MSPND
TXINP	7-4	Указатель линии прерываний для прерывания после передачи. Всего доступно 16 линий прерывания с номерами от 0 до 15. Значение 0000b, записанное в TXINP, выбирает нулевую линию прерываний, 0001b – первую и так далее до 1111b, которое соответствует линии 15. Дополнительно бит TXINP используется для выбора позиции бита ожидающего обработки объекта сообщения m
RXINP	3-0	Указатель линии прерываний для прерывания после приема. Всего доступно 16 линий прерывания с номерами от 0 до 15. Значение 0000b, записанное в RXINP, выбирает нулевую линию прерываний, 0001b – первую и так далее до 1111b, которое соответствует линии 15. Дополнительно бит RXINP используется для выбора позиции бита ожидающего обработки объекта сообщения m

MOAMR – регистр маски объекта сообщения

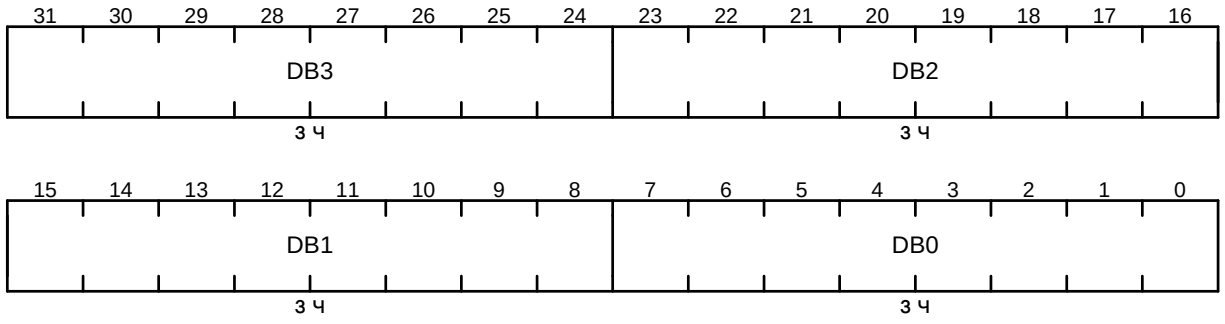
Смещение: Msg_mo + 0Ch
Сброс: 3FFF_FFFFh



Поле	Биты	Описание	
MIDE	29	Маска бита IDE сообщения	
		0	Объект сообщения m может принимать как стандартные, так и расширенные кадры
		1	Объект сообщения m может принимать только те кадры, у которых состояние бита IDE совпадает с его битом IDE
AM	28-0	Маска идентификатора. При приеме расширенного сообщения используется вся маска. При приеме стандартного сообщения используются биты 28-18, при этом состояние битов 17-0 не важно	
–	31-30	Зарезервировано	

MODATAL – младший регистр данных объекта сообщения

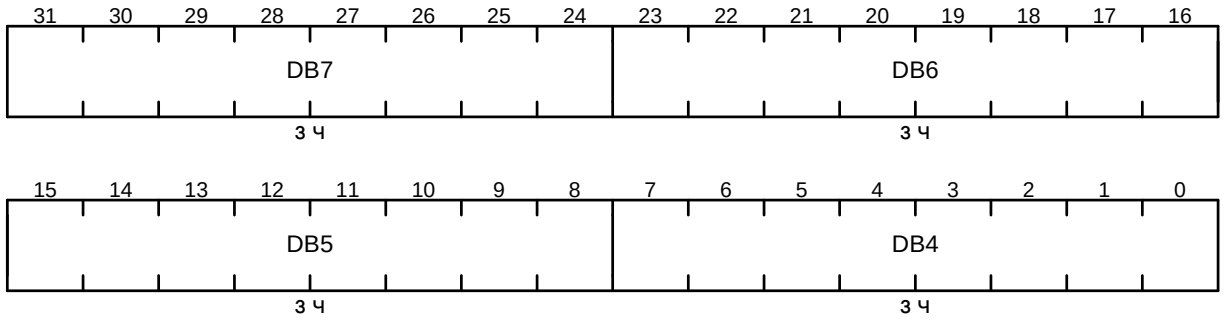
Смещение: Msg_mo + 10h
Сброс: 0h



Поле	Биты	Описание
DB3	31-24	Третий байт данных
DB2	23-16	Второй байт данных
DB1	15-8	Первый байт данных
DB0	7-0	Нулевой байт данных

MODATAH – старший регистр данных объекта сообщения

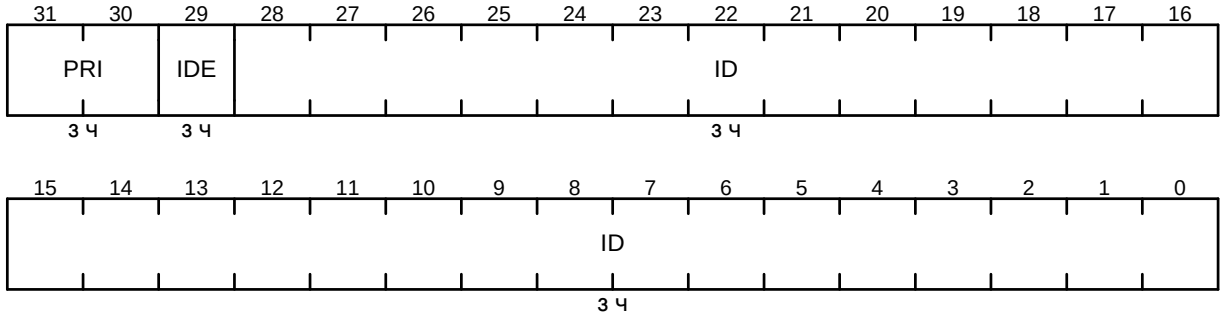
Смещение: Msg_mo + 14h
Сброс: 0h



Поле	Биты	Описание
DB7	31-24	Седьмой байт данных
DB6	23-16	Шестой байт данных
DB5	15-8	Пятый байт данных
DB4	7-0	Четвертый байт данных

MOAR – регистр арбитража объекта сообщения

Смещение: Msg_mo + 18h
Сброс: 0h



Поле	Биты	Описание	
PRI	31-30	Класс приоритета. Поле определяет один из четырех классов (0, 1, 2 и 3) приоритета объекта сообщения m. Нулевой класс устанавливает наивысший приоритет. Объекты сообщений с нулевым классом всегда выигрывают арбитраж при передаче и приеме сообщений. Фильтрация сообщений на основе идентификатора (маскируемого) и позиции в списке организуются только для объектов сообщений с равным приоритетом. Кроме этого, поле PRI определяет метод фильтрации	
		00	Зарезервировано
		01	Фильтрация в зависимости от положения объекта сообщения в списке. Объект сообщения m получает приоритет на передачу сообщения только в случае, если перед ним в списке нет других объектов сообщений с установленными битами MSGVAL, TXEN0 и TXEN1

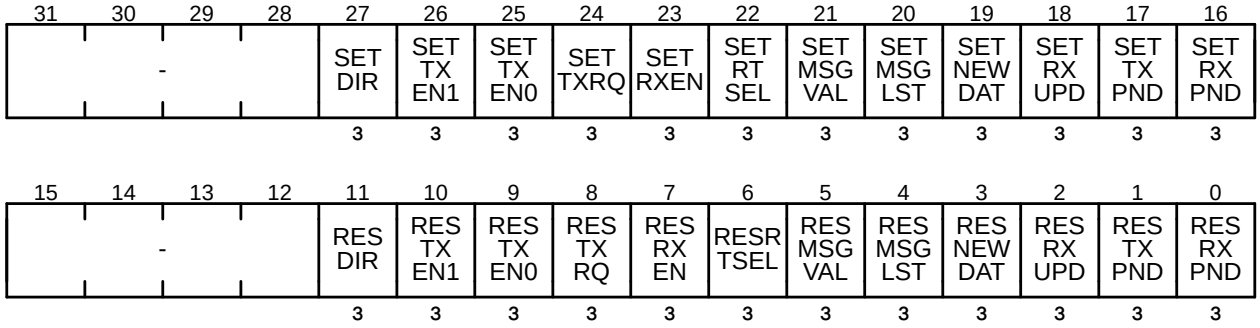
Поле	Биты	Описание	
		10	Фильтрация в зависимости от значения идентификатора. Объект сообщения m получает приоритет на передачу сообщения только в случае, если в списке нет других объектов сообщений с «Идентификатор + IDE + DIR» более высокого приоритета (согласно правилам арбитража в таблице А.31.2)
		11	Фильтрация в зависимости от положения объекта сообщения в списке (как при PRI = 01b)
IDE	29	Бит расширения идентификатора объекта сообщения m	
		0	Объект сообщения m оперирует с кадрами со стандартным 11-битным идентификатором
		1	Объект сообщения m оперирует с кадрами с расширенным 29-битным идентификатором
ID	28-0	Идентификатор объекта сообщения m. При оперировании с расширенными кадрами используются биты 28-0. При оперировании со стандартными кадрами используются биты 28-18, при этом состояние битов 17–0 не важно	

Таблица А.31.2 – Распределение приоритета между объектами сообщений согласно правилам арбитража

Установки для объектов сообщений 0 и 1, которые участвуют в арбитраже (приоритет объекта 0 выше приоритета объекта 1)	Пояснение
MOAR0[28:18] < MOAR1[28:18] (11-битный стандартный идентификатор объекта 0 меньше по числовому значению, чем 11-битный идентификатор объекта 1)	Стандартный кадр с идентификатором, имеющим меньшее значение, обладает более высоким приоритетом
MOAR0[28:18] = MOAR1[28:18]. В регистре MOAR0 бит IDE = 0. В регистре MOAR1 бит IDE = 1	При равенстве значений стандартных идентификаторов, стандартный кадр имеет приоритет перед расширенным
MOAR0[28:18] = MOAR1[28:18]. Биты IDE обоих объектов сброшены. В регистре MOSTAT0 бит DIR = 1. В регистре MOSTAT1 бит DIR = 0	При равенстве значений идентификаторов стандартный кадр данных имеет приоритет перед стандартным кадром удаленного запроса
MOAR0[28:0] = MOAR1[28:0] Биты IDE обоих объектов установлены. В регистре MOSTAT0 бит DIR = 1. В регистре MOSTAT1 бит DIR = 0	При равенстве значений идентификаторов расширенный кадр данных имеет приоритет перед расширенным кадром удаленного запроса
MOAR0[28:0] < MOAR1[28:0] Биты IDE обоих объектов установлены. (29-битный идентификатор объекта 0 меньше по числовому значению, чем 29-битный идентификатор объекта 1)	Расширенный кадр с идентификатором, имеющим меньшее значение, обладает более высоким приоритетом

MOCTR – регистр управления объектом сообщения

Смещение: Msg_mo + 1Ch
Сброс: 0h



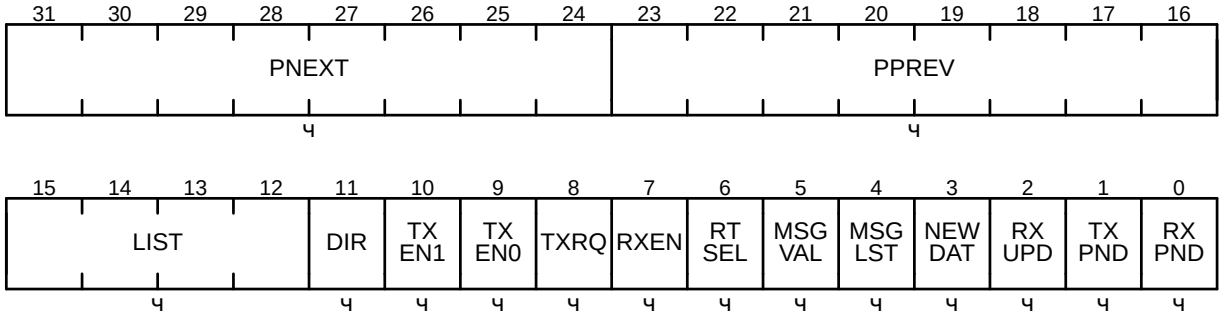
Поле	Биты	Описание
SETDIR, RESDIR	27, 11	Установка/сброс бита DIR
SETTXEN1, RESTXEN1	26, 10	Установка/сброс бита TXEN1
SETTXEN0, RESTXEN0	25, 9	Установка/сброс бита TXEN0
SETTXRQ, RESTXRQ	24, 8	Установка/сброс бита TXRQ
SETRXEN, RESRXEN	23, 7	Установка/сброс бита RXEN
SETRTSEL, RESRTSEL	22, 6	Установка/сброс бита RTSEL
SETMSGVAL, RESMSGVAL	21, 5	Установка/сброс бита MSGVAL
SETMSGLST, RESMSGLST	20, 4	Установка/сброс бита MSGLST
SETNEWDAT, RESNEWDAT	19, 3	Установка/сброс бита NEWDAT
SETRXUPD, RESRXUPD	18, 2	Установка/сброс бита RXUPD
SETTXPND, RESTXPND	17, 1	Установка/сброс бита TXPND
SETRXPND, RESRXPND	16, 0	Установка/сброс бита RXPND
–	31-28, 15-12	Зарезервировано. При чтении возвращаются нули. При записи следует писать 0h

Примечание – Биты с префиксом SET и RES работают попарно. Комбинация состояний этих бит оказывает влияние на соответствующий бит регистра MOSTAT:

- SET* = 1, RES* = 0 устанавливает бит *;
- SET* = 0, RES* = 1 сбрасывает бит *;
- SET* = RES* = 0 или SET* = RES* = 1 не изменяет состояние бита *.

MOSTAT – регистр состояния объекта сообщения

Смещение: Msg_mo + 1Ch
Сброс:
- 0100_0000h для объекта сообщения 0;
- [mo + 1][mo - 1][0000]h для объектов сообщений с номерами от 1 до 254, (соответственно mo от 01h до FEh);
- FFFE_0000h для объекта сообщения 255



Поле	Биты	Описание	
PNEXT	31-24	Указатель на следующий элемент списка. В поле находится номер объекта сообщения, расположенного выше по списку относительно текущего	
PPREV	23-16	Указатель на предыдущий элемент списка. В поле находится номер объекта сообщения, расположенного ниже по списку относительно текущего	
LIST	15-12	Номер списка, которому принадлежит объект сообщения m. Поле обновляется аппаратно при распределении/перераспределении объекта сообщения	
DIR	11	Направление передачи сообщения	
		0	Объект для приема сообщения. При установленном бите TXRQ планируется передача кадра удаленного запроса с идентификатором объекта сообщения m. При приеме кадра данных с соответствующим идентификатором его содержимое сохраняется в объекте сообщения m
		1	Объект для передачи сообщения. При установленном бите TXRQ планируется передача кадра данных сформированного из объекта сообщения m. При приеме кадра удаленного запроса с соответствующим идентификатором будет установлен бит TXRQ данного регистра
TXEN1	10	Бит разрешения передачи кадра	
		0	Передача кадра, сформированного из объекта сообщения m, запрещена
		1	Передача кадра разрешена. Объект сообщения m может быть передан, только если установлены оба бита – TXEN1 и TXEN0. Контроллер CAN использует бит TXEN1 для выбора активного объекта передачи сообщения из FIFO

Поле	Биты	Описание	
TXEN0	9	Бит разрешения передачи кадра	
		0	Передача кадра, сформированного из объекта сообщения m, запрещена
		1	Передача кадра разрешена. Объект сообщения m может быть передан, только если установлены оба бита – TXEN0 и TXEN1. Можно программно очищать бит TXEN0 для запрета передачи сообщения, которое в настоящий момент формируется, или для запрета автоматической передачи ответа на удаленный запрос
TXRQ	8	Бит запроса на передачу	
		0	Передача объекта сообщения m не требуется
		1	Установлен запрос на передачу объекта сообщения m
		Запрос на передачу кадра будет удовлетворен только в случае, если установлены биты TXRQ, TXEN0, TXEN1 и MSGVAL. Бит TXRQ устанавливается аппаратно при получении кадра удаленного запроса. Бит сбрасывается аппаратно при успешном завершении передачи, и если при этом не был повторно программно установлен бит NEWDAT	
RXEN	7	Бит разрешения приема	
		0	Прием кадра в объект сообщения m запрещен
		1	Объект сообщения может принимать сообщения
		Состояние бита учитывается только при фильтрации принимаемых сообщений	
RTSEL	6	Индикатор возможности приема/передачи	
		0	Объект сообщения не может принимать/передавать сообщения
		1	Объект сообщения может принимать/передавать сообщения
		Прием кадра. Бит RTSEL устанавливается аппаратно после того, как выбран объект сообщения m для сохранения только что принятого кадра. Прежде, чем записать принятые данные в объект сообщения m, аппаратная часть проверяет состояние бита RTSEL. ЦП может сбросить этот бит, чтобы запретить запись принятого кадра в объект сообщения m. Передача кадра. Бит RTSEL устанавливается аппаратно после того, как объект сообщения m выбран в качестве следующего передаваемого кадра. Аппаратная часть перед началом передачи проверяет: установлен ли бит RTSEL и сброшен ли бит NEWDAT. Бит RTSEL должен оставаться установленным до окончания передачи. Проверка состояния бита RTSEL производится только при попытке изменения содержимого объекта сообщения m во избежание одновременного выполнения операций передачи кадра и его изменения. Бит не участвует в фильтрации сообщений и не сбрасывается аппаратно	
MSGVAL	5	Бит активности объекта сообщения m	
		0	Не активен
		1	Активен
		Только те объекты сообщений, для которых установлен этот бит, могут использоваться для операций приема и передачи	

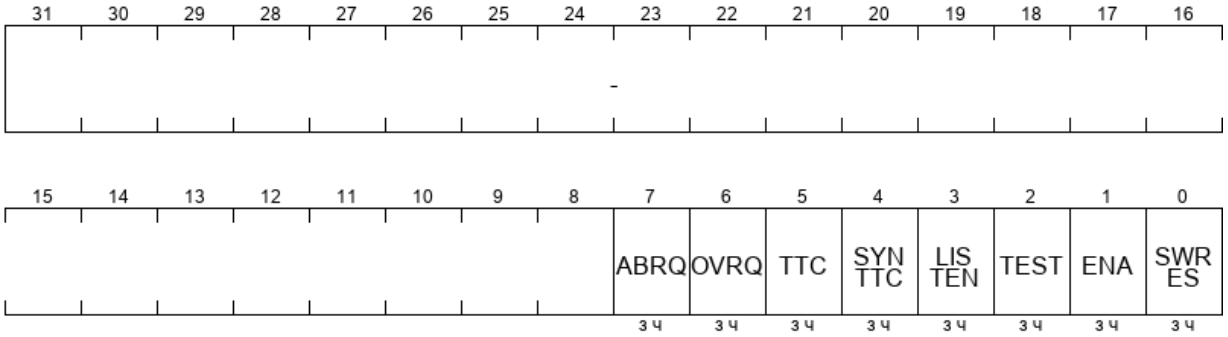
Поле	Биты	Описание	
MSGLST	4	Бит потери сообщения	
		0	Ни одно сообщение не потеряно
		1	Принятое сообщение потеряно вследствие того, что контроллер CAN установил бит NEWDAT по окончании приема сообщения при том, что флаг NEWDAT уже был установлен вследствие полученного ранее сообщения
NEWDAT	3	Индикатор новых данных	
		0	С момента сброса бита NEWDAT никаких изменений объекта сообщения m не происходило
		1	Объект сообщения был изменен. Бит устанавливается аппаратно после того, как принятое сообщение было сохранено в объекте сообщения m. Бит сбрасывается аппаратно после начала передачи объекта сообщения m. Бит NEWDAT следует устанавливать программно после того, как новые данные для передачи будут сохранены в объекте сообщения m для предотвращения автоматического сброса бита TRXQ в конце текущей передачи
RXUPD	2	Индикатор изменений	
		0	Нет текущих изменений
		1	Идентификатор сообщения, поле длины данных DLC и данные в объекте сообщения изменяются
TXPND	1	Индикатор окончания передачи	
		0	Переданных сообщений нет
		1	Сообщение объекта m было успешно передано
RXPND	0	Индикатор окончания приема	
		0	Принятых сообщений нет
		1	Сообщение было успешно принято объектом сообщения m (напрямую или через копирование объекта в режиме шлюза). Бит должен сбрасываться программно

A.32 Регистры контроллера интерфейса CANFD

Базовый адрес:	4100_0000h	Контроллер CANFD0
	4100_1000h	Контроллер CANFD1
	4100_2000h	Контроллер CANFD2
	4100_3000h	Контроллер CANFD3
Смещение:	+ C0h (MSG)	Регистры данных объектов сообщений

CONTROL – общий регистр управления

Смещение: + 40h
Сброс: 0h

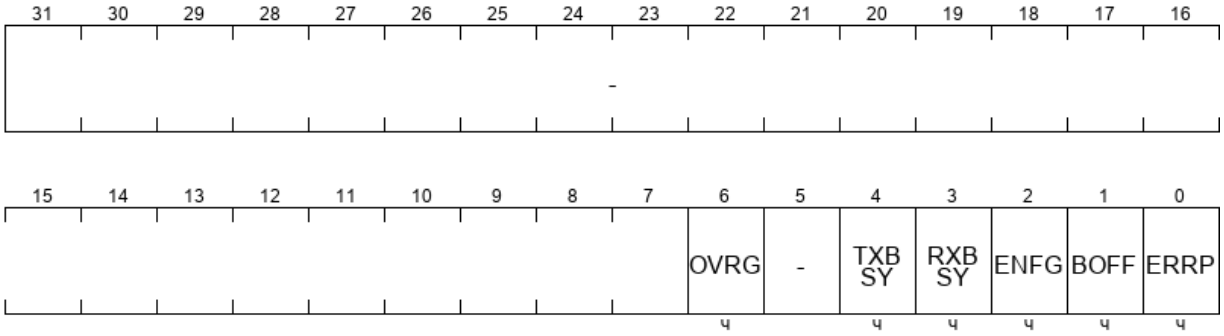


Поле	Биты	Описание	
ABRQ	7	Бит запроса на прерывание	
		0	нет запроса
		1	запрос на прерывание
OVRQ	6	Запрос кадра перегрузки	
		0	нет запроса
		1	запрос кадра перегрузки
TTC	5	Связь с запуском по времени	
		0	выключен
		1	режим TTC
SYNTTC	4	Синхронизация TTC	
		0	захват таймера TTC происходит по биту SOF
		1	захват таймера TTC происходит по биту EOF
LISTEN	3	Режим прослушивания	
		0	выключен
		1	режим прослушивания
TEST	2	Тестовый режим	
		0	выключен
		1	тестовый режим, зануляющий внутренние сигналы TXOK и ack_err. Бит используется в тестовых целях, для игнорирования статуса передачи кадра. В частности: не генерируется прерывание после успешной отправки пакета, не сбрасывается 1 в CANEN, не формируется кадр ошибки при отсутствии ответа от приёмника (ack_err)

Поле	Биты	Описание	
ENA	1	Управление режимом ожидания	
		0	текущая передача прекращается и канал CAN замораживается. Передатчик постоянно обеспечивает рецессивный уровень. Приемник не работает.
		1	включение канала CAN (после ожидания 11 рецессивных бит)
SWRES	0	Программный сброс	
		0	нет сброса
		1	сброс, связан по «или» с аппаратным сбросом
–	31-8	Зарезервировано	

STATUS – общий регистр состояния

Смещение: + 44h
Сброс: 0h

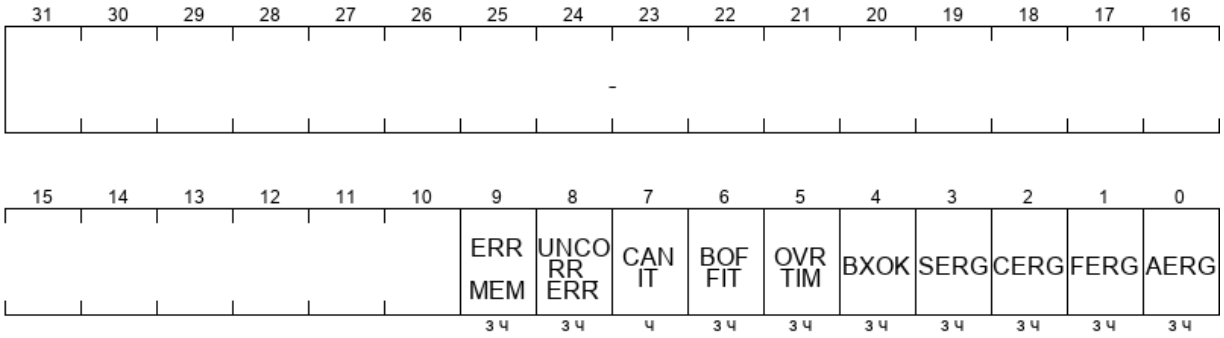


Поле	Биты	Описание	
OVRTIM	6	Флаг фрейма перегрузки	
		0	фрейм перегрузки отсутствует
		1	устанавливается до тех пор, пока созданный фрейм не будет отправлен. Этот флаг не генерирует прерывание
TXBSY	4	Флаг занятости передатчика	
		0	передатчик не занят
		1	передатчик занят, устанавливается аппаратно пока передается фрейм или поле ACK. Этот флаг не генерирует прерывание
RXBSY	3	Флаг занятости приемника	
		0	приемник не занят
		1	устанавливается пока принимается фрейм или отслеживается. Этот флаг не генерирует прерывание
ENFG	2	Флаг включения	
		0	контроллер отключен
		1	контроллер включен. Флаг отображает реальное состояние контроллера, так как бит включения срабатывает только через 11 рецессивных бит. Этот флаг не генерирует прерывание

Поле	Биты	Описание	
BOFF	1	Режим отключения шины	
		0	обычный режим
		1	режим выключения шины. Только вход в режим отключения шины генерирует прерывание BOFFIT
ERRP	0	Ошибка пассивного режима	
		0	пассивный режим без ошибок
		1	ошибка пассивного режима. Этот флаг не генерирует прерывание
–	31-7, 5	Зарезервировано	

IF – общий регистр статуса прерываний

Смещение: + 48h
Сброс: 0h

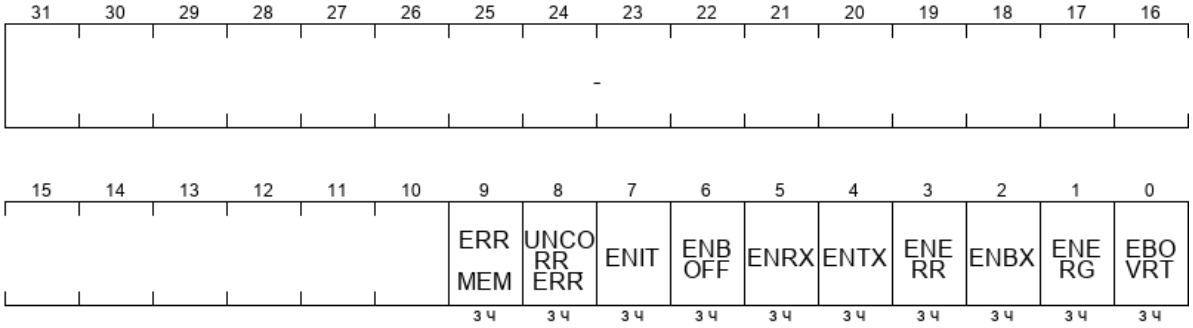


Поле	Биты	Описание	
ERR_MEM	9	Флаг ошибки в одном бите данных памяти сообщений. Запись логической единицы сбрасывает этот флаг прерывания.	
		0	прерывания нет;
		1	была выявлена и исправлена ошибка в одном бите при обращении к памяти сообщений.
UNCORR_ERR	8	Флаг неисправимой ошибки в двух или более битах данных памяти сообщений. Запись логической единицы сбрасывает этот флаг прерывания.	
		0	прерывания нет;
		1	были выявлены ошибки в двух или более битах данных памяти сообщений.
CANIT	7	Общий флаг прерывания	
		0	прерывания нет
		1	прерывание CAN: отображение всех прерываний контроллера CAN, за исключением прерывания OVRTIM. Этот бит может быть использован для метода опроса

Поле	Биты	Описание	
BOFFIT	6	Флаг прерывания отключения шины. Запись логической единицы сбрасывает этот флаг прерывания. Флаг BOFIT устанавливается только тогда, когда CAN переходит в режим отключения шины (при выходе из пассивного режима ошибки).	
		0	прерывания нет.
		1	прерывание выключения шины, когда CAN переходит в режим выключения шины.
OVRTIM	5	Таймер переполнения CAN. Запись логической единицы сбрасывает этот флаг прерывания.	
		0	прерывания нет.
		1	прерывание переполнения таймера CAN: устанавливается, когда таймер CAN переключается с 0xFFFF на 0.
BXOK	4	Прерывание общего приема. Запись логической единицы сбрасывает этот флаг прерывания. Флаг BXOK может быть снят только в том случае, если все поля CONMOB MObs были переписаны ранее.	
		0	прерывания нет.
		1	прерывание пакетного приема: устанавливается по завершении приема всеми активными MObs.
SERG	3	Ошибка общего характера. Запись логической единицы сбрасывает этот флаг прерывания.	
		0	прерывания нет.
		1	прерывание с ошибкой заполнения – обнаружение более 5 последовательных битов с одинаковой полярностью.
CERG	2	Общая ошибка CRC. Запись логической единицы сбрасывает этот флаг прерывания.	
		0	прерывания нет.
		1	прерывание с ошибкой CRC – проверка CRC в удаленном сообщении не соответствует полю CRC.
FERG	1	Ошибка формы общего характера. Запись логической единицы сбрасывает этот флаг прерывания.	
		0	прерывания нет.
		1	прерывание ошибки формы – одно или несколько нарушений фиксированной формы в разделителе CRC, разделителе подтверждения или EOF.
AERG	0	Общая ошибка подтверждения. Запись логической единицы сбрасывает этот флаг прерывания.	
		0	прерывания нет.
		1	прерывание с ошибкой подтверждения – не обнаружено доминирующего бита в слоте подтверждения.
–	31-10	Зарезервировано	

IE – общий регистр включения прерываний

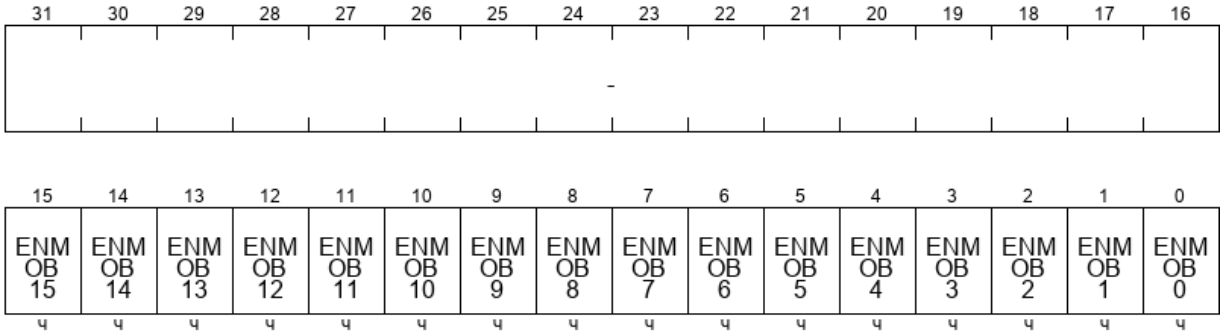
Смещение: + 4Ch
Сброс: 0h



Поле	Биты	Описание	
ERR_MEM	9	Включение прерывания по флагу ошибки в одном бите данных памяти сообщений	
		0	прерывание отключено.
		1	прерывание включено.
UNCORR_ERR	8	Включение прерывания по ошибке в двух и более битах данных памяти сообщений	
		0	прерывание отключено.
		1	прерывание включено.
ENIT	7	Включение всех прерываний (за исключением прерывания переполнения таймера CAN)	
		0	прерывание отключено.
		1	CANIT прерывание включено.
ENBOFF	6	Включение прерывания выключенной шины	
		0	прерывание отключено.
		1	прерывание включено.
ENRX	5	Включение прерывания по приему	
		0	прерывание отключено.
		1	прерывание включено.
ENTX	4	Включение прерывания по передаче	
		0	прерывание отключено.
		1	прерывание включено.
ENERR	3	Включение прерывания с ошибками объектов сообщений MOb	
		0	прерывание отключено.
		1	прерывание включено.
ENBX	2	Включение прерывания общего приема	
		0	прерывание отключено.
		1	прерывание BXOK включено.
ENERG	1	Включение прерывания общих ошибок	
		0	прерывание отключено.
		1	прерывание включено.
EBOVRT	0	Включение прерывания переполнения таймера CAN	
		0	прерывание отключено.
		1	прерывание включено.
–	31-10	Зарезервировано	

MOBEN –регистр включенных объектов сообщений

Смещение: + 50h
Сброс: 0h

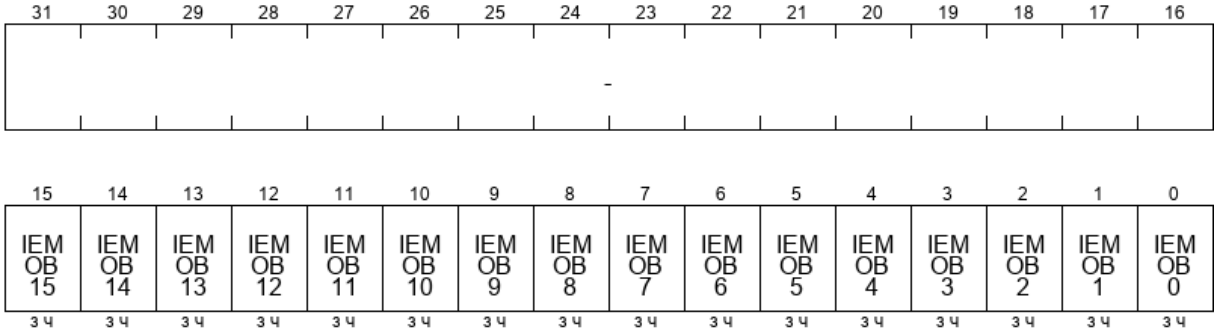


Поле	Биты	Описание
ENMOB15	15	Бит доступности объекта сообщения 15
ENMOB14	14	Бит доступности объекта сообщения 14
ENMOB13	13	Бит доступности объекта сообщения 13
ENMOB12	12	Бит доступности объекта сообщения 12
ENMOB11	11	Бит доступности объекта сообщения 11
ENMOB10	10	Бит доступности объекта сообщения 10
ENMOB9	9	Бит доступности объекта сообщения 9
ENMOB8	8	Бит доступности объекта сообщения 8
ENMOB7	7	Бит доступности объекта сообщения 7
ENMOB6	6	Бит доступности объекта сообщения 6
ENMOB5	5	Бит доступности объекта сообщения 5
ENMOB4	4	Бит доступности объекта сообщения 4
ENMOB3	3	Бит доступности объекта сообщения 3
ENMOB2	2	Бит доступности объекта сообщения 2
ENMOB1	1	Бит доступности объекта сообщения 1
ENMOB0	0	Бит доступности объекта сообщения 0
–	31-16	Зарезервировано

Примечание – Соответствующий бит ENMOB устанавливается, когда объект сообщения включен (значение MOBCTRL.CONMOB не нулевое). При установке бита TXOK или RXOK регистра MOBSTATUS, соответствующий бит объекта сообщения ENMOB сбрасывается.

MOBIE –регистр включения прерываний объектов сообщений

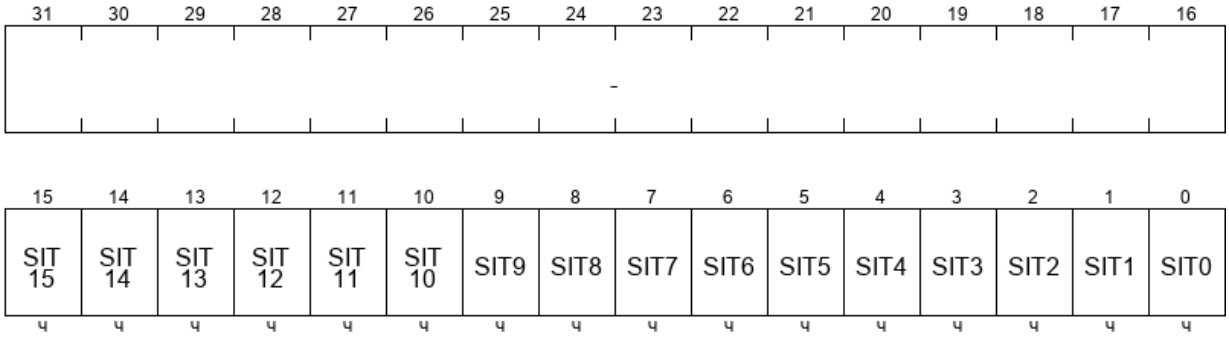
Смещение: + 54h
Сброс: 0h



Поле	Биты	Описание
IEMOB15	15	Бит включения прерывания объекта сообщения 15
IEMOB14	14	Бит включения прерывания объекта сообщения 14
IEMOB13	13	Бит включения прерывания объекта сообщения 13
IEMOB12	12	Бит включения прерывания объекта сообщения 12
IEMOB11	11	Бит включения прерывания объекта сообщения 11
IEMOB10	10	Бит включения прерывания объекта сообщения 10
IEMOB9	9	Бит включения прерывания объекта сообщения 9
IEMOB8	8	Бит включения прерывания объекта сообщения 8
IEMOB7	7	Бит включения прерывания объекта сообщения 7
IEMOB6	6	Бит включения прерывания объекта сообщения 6
IEMOB5	5	Бит включения прерывания объекта сообщения 5
IEMOB4	4	Бит включения прерывания объекта сообщения 4
IEMOB3	3	Бит включения прерывания объекта сообщения 3
IEMOB2	2	Бит включения прерывания объекта сообщения 2
IEMOB1	1	Бит включения прерывания объекта сообщения 1
IEMOB0	0	Бит включения прерывания объекта сообщения 0
–	31-16	Зарезервировано

MOBIF –регистр статуса прерываний объектов сообщений

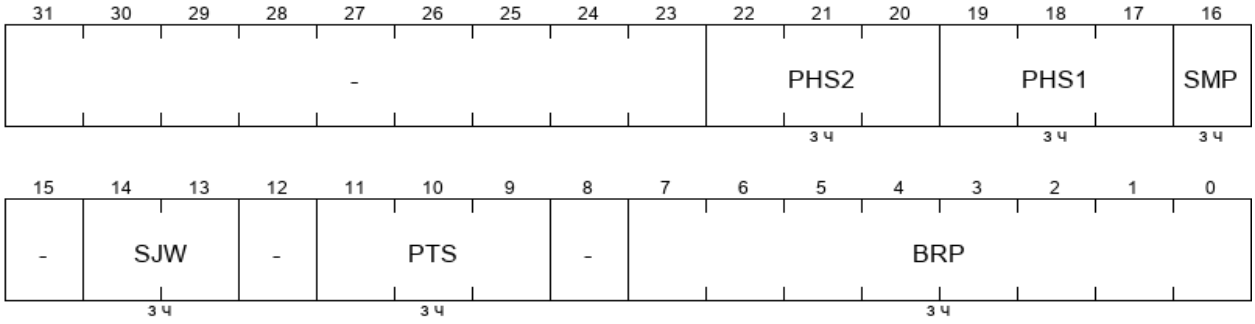
Смещение: + 58h
Сброс: 0h



Поле	Биты	Описание
SIT15	15	Флаг статуса прерывания объекта сообщения 15
SIT14	14	Флаг статуса прерывания объекта сообщения 14
SIT13	13	Флаг статуса прерывания объекта сообщения 13
SIT12	12	Флаг статуса прерывания объекта сообщения 12
SIT11	11	Флаг статуса прерывания объекта сообщения 11
SIT10	10	Флаг статуса прерывания объекта сообщения 10
SIT9	9	Флаг статуса прерывания объекта сообщения 9
SIT8	8	Флаг статуса прерывания объекта сообщения 8
SIT7	7	Флаг статуса прерывания объекта сообщения 7
SIT6	6	Флаг статуса прерывания объекта сообщения 6
SIT5	5	Флаг статуса прерывания объекта сообщения 5
SIT4	4	Флаг статуса прерывания объекта сообщения 4
SIT3	3	Флаг статуса прерывания объекта сообщения 3
SIT2	2	Флаг статуса прерывания объекта сообщения 2
SIT1	1	Флаг статуса прерывания объекта сообщения 1
SIT0	0	Флаг статуса прерывания объекта сообщения 0
–	31-16	Зарезервировано

BITTIMING – регистр синхронизации

Смещение: + 5Ch
Сброс: 0h

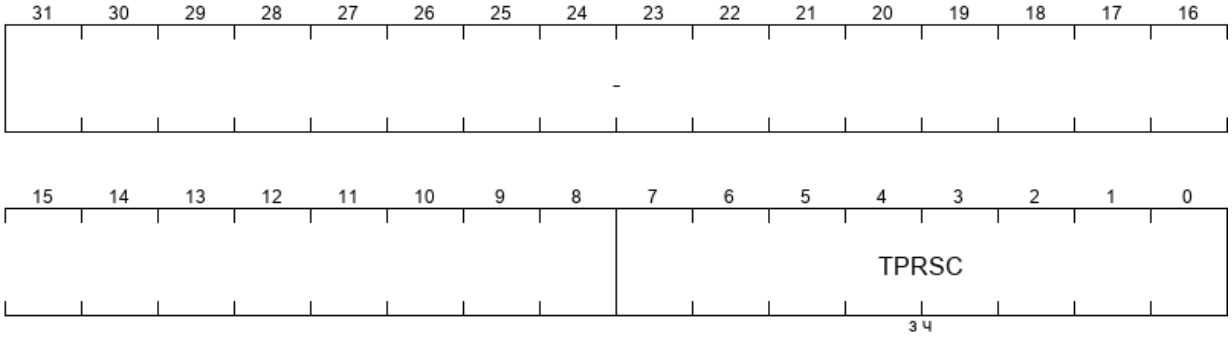


Поле	Биты	Описание	
PHS2	22-20	Сегмент фазы PHS2. Эта фаза используется для компенсации ошибок на границе фазы. Этот сегмент может быть сокращен на ширину импульса повторной синхронизации. Должно выполняться условие PHS2≤PHS1. Расчитывается как T_{PHS2} = Tscl x (PHS2 [2:0] + 1).	
PHS1	19-17	Сегмент фазы PHS1. Эта фаза используется для компенсации ошибок на границе фазы. Этот сегмент может быть увеличен на ширину импульса повторной синхронизации. T_{PHS1} = Tscl x (PHS1 [2:0] + 1).	
SMP	16	Точка выборки. Эта опция позволяет фильтровать возможные шумы на входном выводе Tx CAN.	
		0	выборка будет производиться один раз в настроенной пользователем точке выборки – SP.
		1	при трехточечной конфигурации выборки первая выборка будет происходить за два такта TclkIO до настроенной пользователем точки выборки – SP, снова за один такт TclkIO до SP и, наконец, в SP. Затем уровень битов будет определен большинством из трех выборок. При установке SMP в «1», поле BRP не должно содержать нулевое значение
SJW	14-13	Ширина импульса повторной синхронизации. Чтобы компенсировать фазовые сдвиги между тактовыми генераторами различных контроллеров шины, контроллер должен повторно синхронизироваться на любом соответствующем фронте сигнала текущей передачи. Ширина импульса синхронизации определяет максимальное количество тактов. Битовый период может быть сокращен или удлинен повторной синхронизацией. Расчитывается как T_{SJW} = Tscl x (SJW [1:0] +1)	
PTS	11-9	Временной сегмент распространения. Эта часть битового времени используется для компенсации физического времени задержки в сети. Это в два раза больше суммы времени распространения сигнала по шине, задержки входного компаратора и задержки выходного драйвера. Расчитывается как T_{PTS} = Tscl x (PTS [2:0] + 1)	

Поле	Биты	Описание
BRP	7-0	Скорость передачи. Биты, задающие значение «Baud Rate», тем самым задающие период системного тактового сигнала CAN Tsck.
–	31-23, 15, 12, 8	Зарезервировано

TIMERCTRL – регистр управления таймером

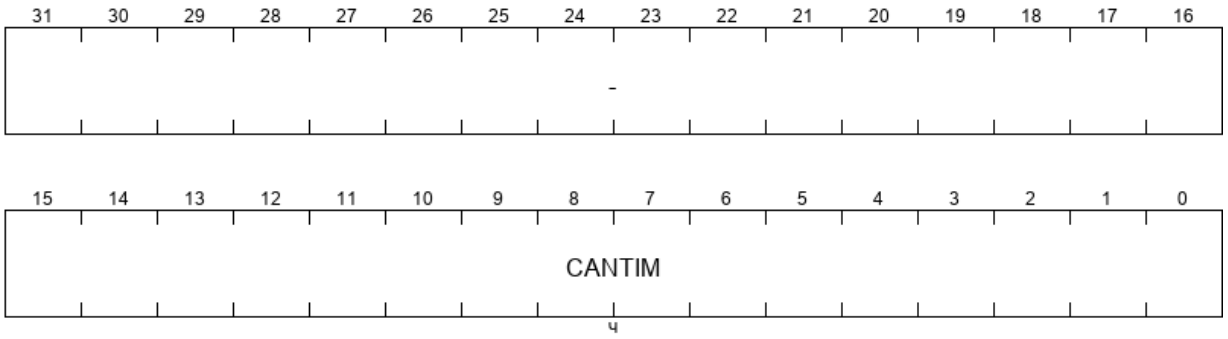
Смещение: + 60h
Сброс: 0h



Поле	Биты	Описание
TPRSC	7-0	Предварительный делитель таймера CAN от 0 до 255. Он обеспечивает синхронизацию с таймером CAN, если включен контроллер CAN. TclkTIMERVAL = TclkIO x 8 x (TPRSC [7:0] + 1)
–	31-8	Зарезервировано

TIMERVAL – регистр таймера

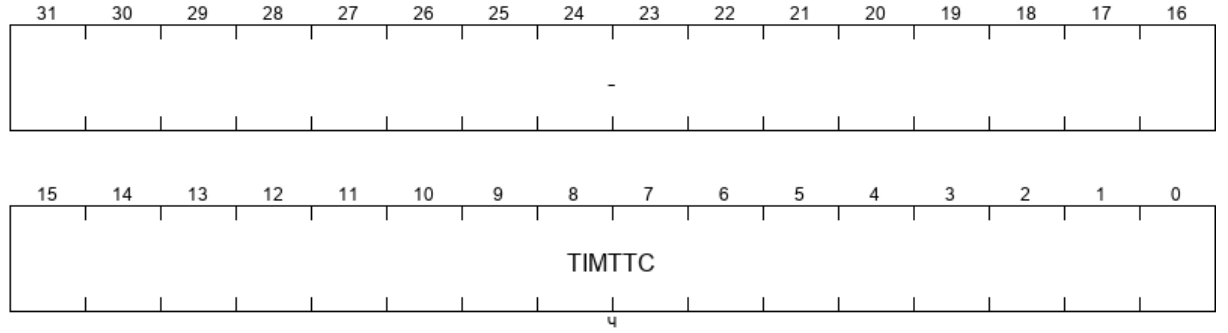
Смещение: + 64h
Сброс: 0h



Поле	Биты	Описание
CANTIM	15-0	Счетчик таймера CAN в диапазоне от 0 до 65535.
–	31-16	Зарезервировано

TTCVAL – регистр таймера TTC

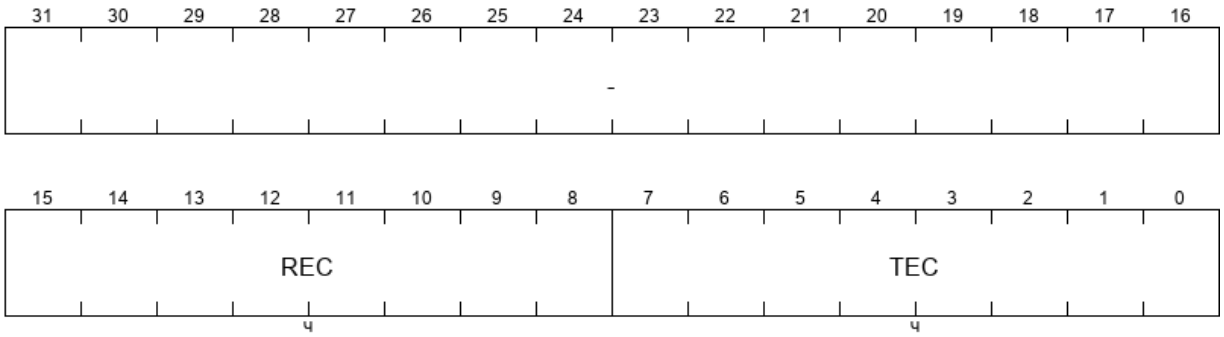
Смещение: + 68h
Сброс: 0h



Поле	Биты	Описание
TIMTTC	15-0	Счетчик таймера TTC CAN в диапазоне от 0 до 65535.
–	31-16	Зарезервировано

TREC – регистр счетчика ошибок передач

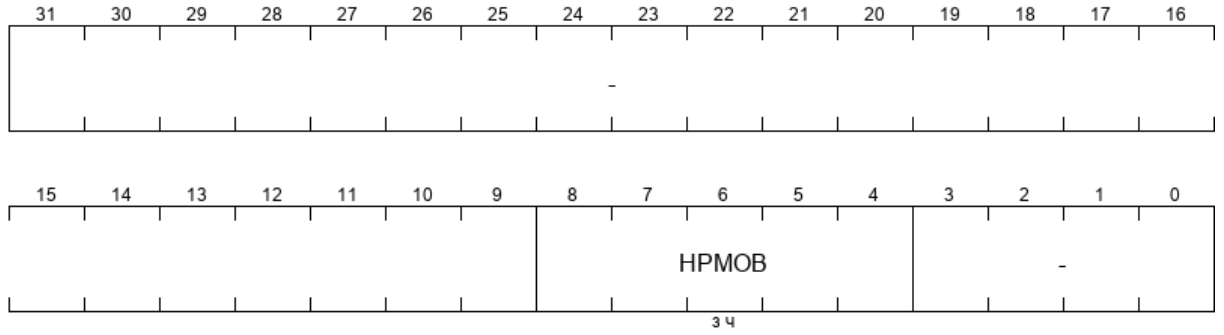
Смещение: + 6Ch
Сброс: 0h



Поле	Биты	Описание
REC	15-8	Счетчик ошибок приема CAN от 0 до 255.
TEC	7-0	Счетчик ошибок передачи CAN от 0 до 255.
–	31-16	Зарезервировано

HPRIMOB – регистр объекта сообщения с наивысшим приоритетом

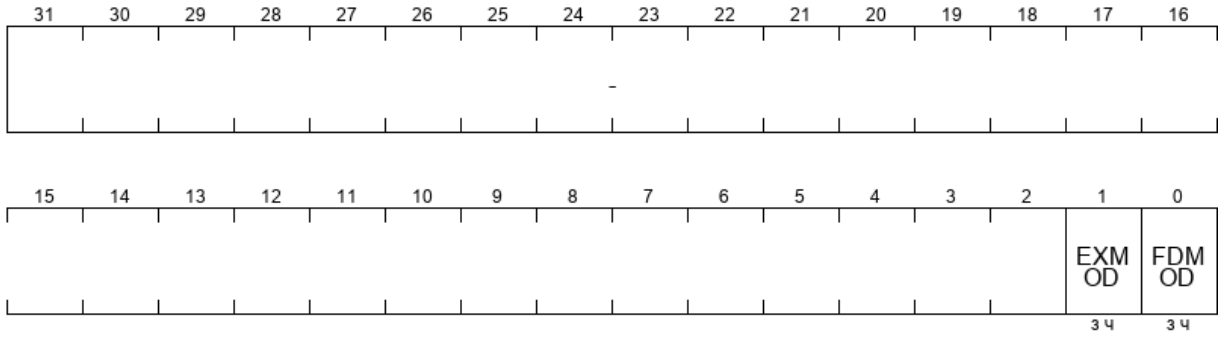
Смещение: + 70h
Сброс: 0000_01F0h



Поле	Биты	Описание
HPMOB	8-4	Номер объекта сообщения с наивысшим приоритетом
–	31-9, 3-0	Зарезервировано

FD_CTRL –регистр управления FD режимом

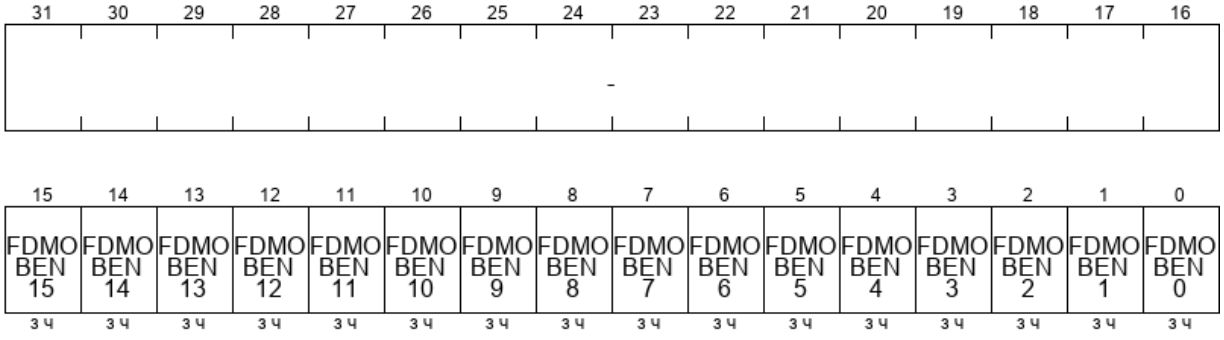
Смещение: + 78h
Сброс: 0h



Поле	Биты	Описание	
EXMOD	1	Бит включения CAN FD ISO	
		0	CAN формирует пакеты согласно формату non-ISO
		1	CAN формирует пакеты согласно формату ISO
FDMOD	0	Бит включения режима FD	
		0	CAN работает в режиме 2.0
		1	CAN работает в режиме FD
–	31-2	Зарезервировано	

MOBFDEN – регистр включения режима FD для каждого объекта сообщений

Смещение: + 7Ch
Сброс: 0h

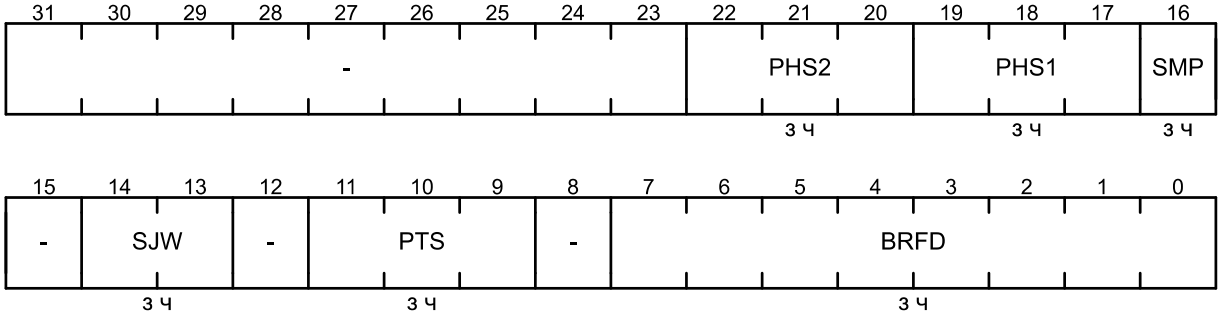


Поле	Биты	Описание
FDMOBEN15	15	Бит включения FD режима для объекта сообщения 15
FDMOBEN14	14	Бит включения FD режима для объекта сообщения 14
FDMOBEN13	13	Бит включения FD режима для объекта сообщения 13
FDMOBEN12	12	Бит включения FD режима для объекта сообщения 12
FDMOBEN11	11	Бит включения FD режима для объекта сообщения 11
FDMOBEN10	10	Бит включения FD режима для объекта сообщения 10

Поле	Биты	Описание
FDMOBEN9	9	Бит включения FD режима для объекта сообщения 9
FDMOBEN8	8	Бит включения FD режима для объекта сообщения 8
FDMOBEN7	7	Бит включения FD режима для объекта сообщения 7
FDMOBEN6	6	Бит включения FD режима для объекта сообщения 6
FDMOBEN5	5	Бит включения FD режима для объекта сообщения 5
FDMOBEN4	4	Бит включения FD режима для объекта сообщения 4
FDMOBEN3	3	Бит включения FD режима для объекта сообщения 3
FDMOBEN2	2	Бит включения FD режима для объекта сообщения 2
FDMOBEN1	1	Бит включения FD режима для объекта сообщения 1
FDMOBEN0	0	Бит включения FD режима для объекта сообщения 0
–	31-16	Зарезервировано

FDBITTIMING – регистр таймингов синхронизации в режиме FD

Смещение: + 80h
Сброс: 0h

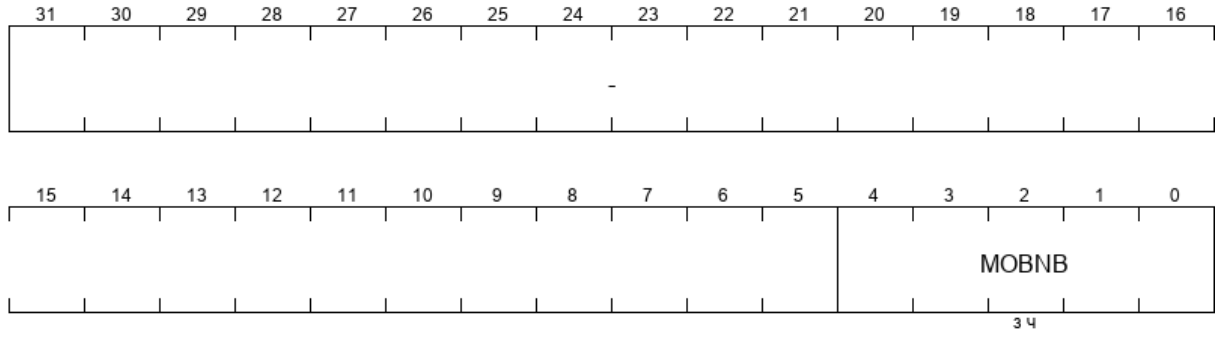


Поле	Биты	Описание	
PHS2	22-20	Сегмент фазы PHS2. Эта фаза используется для компенсации ошибок на границе фазы. Этот сегмент может быть сокращен на ширину импульса повторной синхронизации. Должно выполняться условие PHS2≤PHS1. Расчитывается как T_{PHS2} = Tscl x (PHS2 [2:0] + 1).	
PHS1	19-17	Сегмент фазы PHS1. Эта фаза используется для компенсации ошибок на границе фазы. Этот сегмент может быть увеличен на ширину импульса повторной синхронизации. T_{PHS1} = Tscl x (PHS1 [2:0] + 1).	
SMP	16	Точка выборки. Эта опция позволяет фильтровать возможные шумы на входном выводе Tx CAN.	
		0	выборка будет производиться один раз в настроенной пользователем точке выборки – SP.
		1	при трехточечной конфигурации выборки первая выборка будет происходить за два такта TclkIO до настроенной пользователем точки выборки – SP, снова за один такт TclkIO до SP и, наконец, в SP. Затем уровень битов будет определен большинством из трех выборок. При установке SMP в «1», поле BRFD не должно содержать нулевое значение

Поле	Биты	Описание
SJW	14-13	Ширина ипульса повторной синхронизации. Чтобы компенсировать фазовые сдвиги между тактовыми генераторами различных контроллеров шины, контроллер должен повторно синхронизироваться на любом соответствующем фронте сигнала текущей передачи. Ширина импульса синхронизации определяет максимальное количество тактов. Битовый период может быть сокращен или удлинен повторной синхронизацией. Расчитывается как T_{SJW} = Tscl x (SJW [1:0] +1)
PTS	11-9	Временной сегмент распространения. Эта часть битового времени используется для компенсации физического времени задержки в сети. Это в два раза больше суммы времени распространения сигнала по шине, задержки входного компаратора и задержки выходного драйвера. Расчитывается как T_{PTS} = Tscl x (PTS [2:0] + 1)
BRFD	7-0	Биты, задающие значение «Baud Rate» части пакета CAN, передающейся в режиме FD. Примечание – Если значение в поле BRFD регистра FDBITTIMING больше, чем значение в поле BRP регистра BITTIMING, то для BRFD берется значение из BRP. При этом бит BRS будет иметь доминантный уровень. Контроль соотношения длительности фаз квантования возлагается на пользователя.
–	31-23, 15, 12, 8	Зарезервировано

MOBPAGE – регистр страницы объекта сообщения

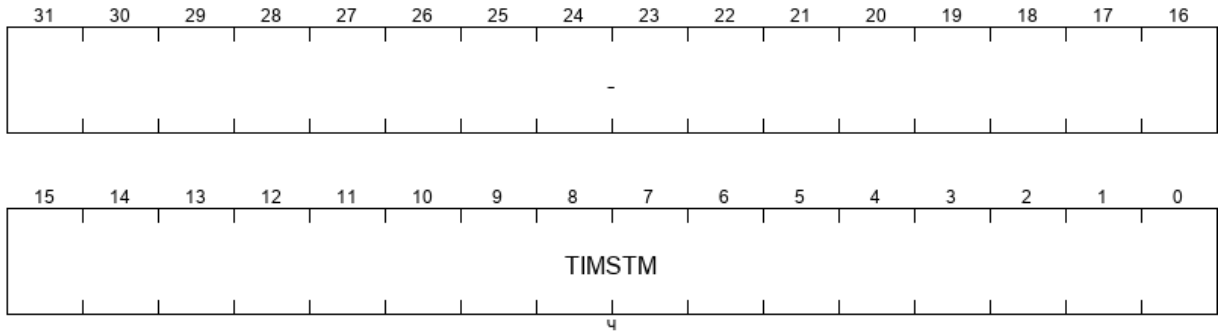
Смещение: + A8h
Сброс: 0h



Поле	Биты	Описание
MOBNB	4-0	Выбор номера объекта сообщения MOb, доступные номера от 0 до 15. Для того, чтобы регистр MOBPAGE не указывал на конкретный объект сообщения MOb (его номер), бит поля MOBNB[4] должен быть установлен в 1, тогда как остальные биты MOBNB[3:0] должны быть сброшены.
–	31-5	Зарезервировано

TIMESTAMP – регистр временной метки

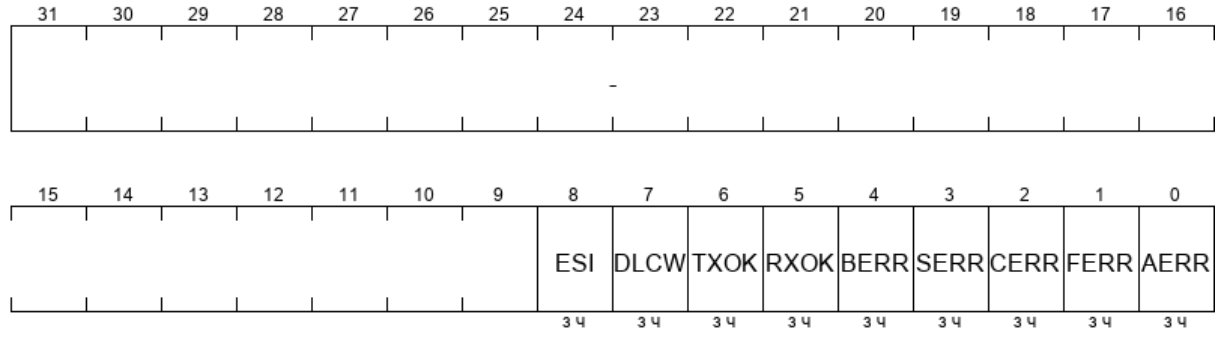
Смещение: + ACh
Сброс: 0h



Поле	Биты	Описание
TIMSTM	15-0	Счетчик временной метки с диапазоном значений от 0 до 65535.
–	31-16	Зарезервировано

MOBSTATUS – регистр статуса объектов сообщений

Смещение: + B0h
Сброс: 0h



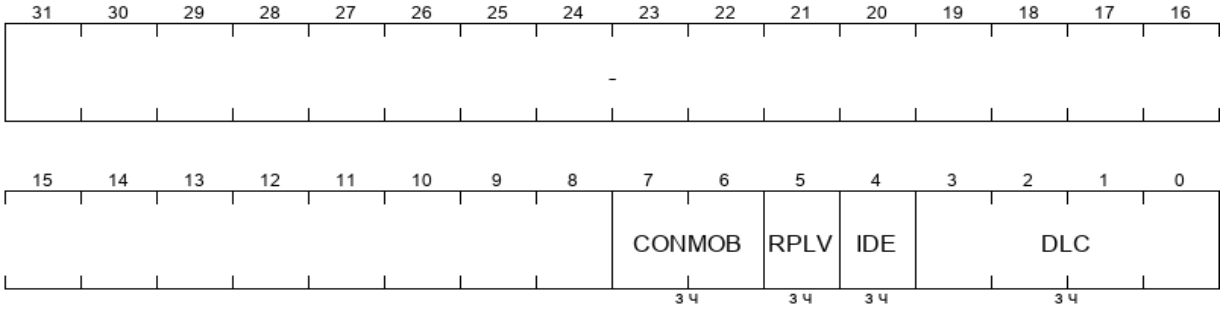
Поле	Биты	Описание	
ESI	8	Флаг индикации состояния ошибки	
		0	Кадр CANFD был передан узлом в состоянии активной ошибки ошибкой. Или был передан кадр CAN2.0;
		1	Кадр CANFD был передан узлом в состоянии пассивной ошибки
DLCW	7	Входящее сообщение не содержит ожидаемого DLC. Независимо от типа кадра, поле DLC регистра MOBCTRL обновляется полученным DLC. Этот флаг сбрасывается путем записи 1.	

Поле	Биты	Описание
TXOK	6	Успешная передача. Установка данного флага будет сопровождаться установкой бита SIT[n] в регистре MOBIF, где n – номер соответствующего объекта сообщения Mob, и вызовом прерывания, при условии установленного бита ENTX регистра IE. Бит TXOK сбрасывается путем записи 1. TxOK устанавливается в конце поля EOF. Когда контроллер готов отправить кадр, если в качестве передатчиков включены два или более объекта сообщения, сначала указывается младший индекс объекта сообщения MOb (от 0 до 15).
RXOK	5	Успешный прием. Установка данного флага будет сопровождаться установкой бита SIT[n] в регистре MOBIF, где n – номер соответствующего объекта сообщения Mob, и вызовом прерывания, при условии установленного бита ENRX регистра IE. Бит RXOK сбрасывается путем записи 1. RXOK устанавливается в конце 6-го бита поля EOF. В случае двух или более обращений к принимающему объекту сообщения сначала обновляется нижний индекс объекта сообщения MOb (от 0 до 15).
BERR	4	Битовая ошибка при передаче. Значение отслеживаемого бита отличается от значения отправленного бита. Исключения: получение доминантного бита, вместо отслеживаемого рецессивного бита, во время процедуры арбитража, а также обнаружение доминантного бита во время слота подтверждения. Установка данного флага будет сопровождаться установкой бита SIT[n] в регистре MOBIF, где n – номер соответствующего объекта сообщения Mob, и вызовом прерывания, при условии установленного бита ENERR регистра IE. Бит BERR сбрасывается путем записи 1.
SERR	3	Ошибка заполнения. Обнаружение более пяти последовательных битов с одинаковой полярностью. Установка данного флага будет сопровождаться установкой бита SIT[n] в регистре MOBIF, где n – номер соответствующего объекта сообщения Mob, и вызовом прерывания, при условии установленного бита ENERR регистра IE. Бит SERR сбрасывается путем записи 1.
CERR	2	Ошибка CRC. Получатель выполняет проверку CRC для каждого принятого сообщения от начала кадра до поля данных. Если эта проверка не совпадает с полем CRC, устанавливается ошибка CRC. Установка данного флага будет сопровождаться установкой бита SIT[n] в регистре MOBIF, где n – номер соответствующего объекта сообщения Mob, и вызовом прерывания, при условии установленного бита ENERR регистра IE. Бит CERR сбрасывается путем записи 1.

Поле	Биты	Описание
FERR	1	Ошибка формы. Возникает в результате одного или нескольких нарушений фиксированной формы в следующих битовых полях: – Разделитель CRC. – Разделитель подтверждения. – EOF. Установка данного флага будет сопровождаться установкой бита SIT[n] в регистре MOBIF, где n – номер соответствующего объекта сообщения Mob, и вызовом прерывания, при условии установленного бита ENERR регистра IE. Бит FERR сбрасывается путем записи 1.
AERR	0	Ошибка подтверждения. Не обнаружено доминантного бита в слоте подтверждения. Установка данного флага будет сопровождаться установкой бита SIT[n], где n – номер соответствующего объекта сообщения Mob, и вызовом прерывания, при условии установленного бита ENERR регистра IE. Бит AERR должен быть очищен путем записи 1.
–	31-9	Зарезервировано

MOBCTRL –регистр управления объектами сообщений

Смещение: + B4h
Сброс: 0h

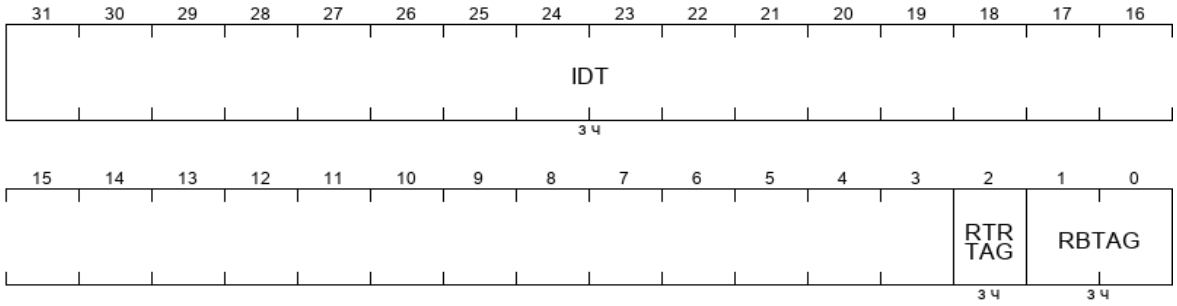


Поле	Биты	Описание	
CONMOB	7-6	Биты конфигурации объектов сообщений. Эти биты устанавливают тип обмена, который должен быть выполнен. Биты CONMOB не очищаются после выполнения обмена. Пользователь должен переписать конфигурацию, чтобы подготовить новый обмен данными. Эта операция необходима для того, чтобы иметь возможность сбросить флаг BXOK. Запись ненулевого значения устанавливает соответствующий бит регистра MOBEN	
		00b	отключить
		01b	включить передачу
		10b	включить прием
		11b	включить прием буфера кадров. Этот режим не влияет на прием, однако позволяет контроллеру отслеживать успешное получение кадров, и в случае, когда каждый настроенный в этом режиме объект сообщения MOb принял кадр, генерируется прерывание BXOK

Поле	Биты	Описание	
RPLV	5	Ответ действителен. Используется в режиме автоматического ответа после получения удаленного кадра.	
		0	ответ не готов
		1	ответ готов и действителен
IDE	4	Расширение идентификатора. IDE-бит удаленного устройства или фрейма данных для отправки. Этот бит обновляется соответствующим значением отправленного или полученного кадра данных.	
		0	стандарт CAN rev 2.0 A (длина идентификаторов = 11 бит)
		1	стандарт CAN rev 2.0 B (длина идентификаторов = 29 бит)
DLC	3-0	Код длины данных. Количество байт в поле данных сообщения. Поле DLC удаленного устройства или фрейма данных для отправки. Диапазон DLC составляет от 0 до 8. Если поле DLC содержит значение больше 8, тогда используется таблица кодировки для режима FD (см. таблицу 39.1). Это поле обновляется соответствующим значением отправленного или полученного фрейма данных. Если ожидаемое DLC отличается от входящего DLC, в регистре MOBSTATUS появится предупреждение DLCW.	
–	31-8	Зарезервировано	

IDTAG – регистр тега идентификатора

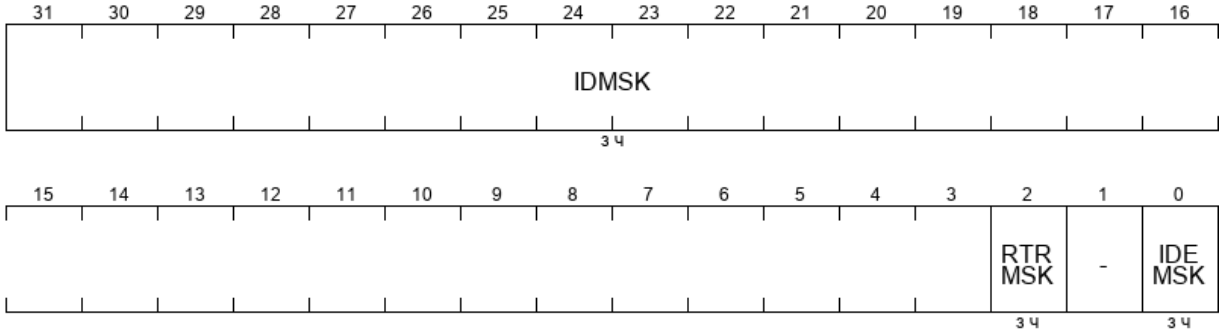
Смещение: + B8h
Сброс: 0h



Поле	Биты	Описание
IDT	31-3	Идентификационный тег. Поле идентификатора удаленного устройства или фрейма данных для отправки. Это поле обновляется соответствующим значением удаленного или полученного фрейма данных.
RTRTAG	2	Тег запроса на удаленную передачу. Этот тег обновляется соответствующим значением отправленного или полученного фрейма данных. В случае режима автоматического ответа этот бит автоматически сбрасывается перед отправкой ответа. В случае передачи FD кадра, значение бита игнорируется.
RBTAG	1-0	Зарезервированные биты тега удаленного устройства. Этот тег обновляется соответствующим значением отправленного или полученного фрейма данных.

IDMASK – регистр маски идентификатора

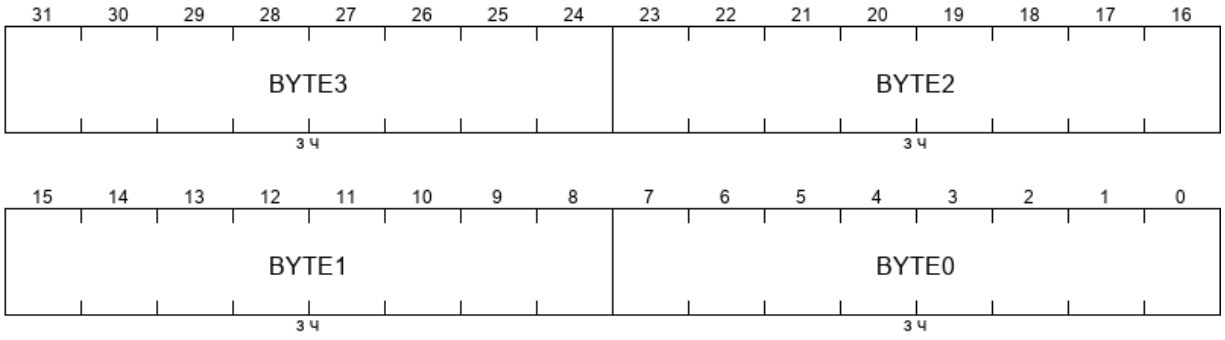
Смещение: + BCh
Сброс: 0h



Поле	Биты	Описание
IDMSK	31-3	Маска идентификатора. Сравниваются только биты, в которых установлены 1. Биты с установленным 0 не сравниваются и считаются верными.
RTRMSK	2	Маска запроса на удаленную передачу
IDEMSK	0	Маска расширения идентификатора
–	1	Зарезервировано

MESSAGE – массив регистров данных объектов сообщений

Смещение: MSG + 4h*n, где n – номер ячейки данных от 0 до 15.
Сброс: 0h



Поле	Биты	Описание
BYTE3	31-24	Байт 3 памяти данных объекта сообщения
BYTE2	23-16	Байт 2 памяти данных объекта сообщения
BYTE1	15-8	Байт 1 памяти данных объекта сообщения
BYTE0	7-0	Байт 0 памяти данных объекта сообщения

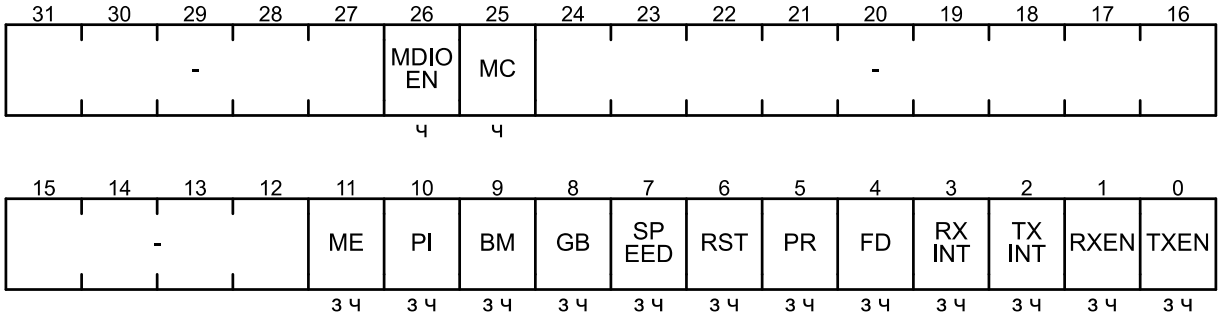
Примечание – В каждом регистре содержится 4 байта данных. Каждый объект сообщения может содержать до 64 байт данных. Номер объекта сообщения MOb определяется содержимым поля MOBNB[3:0] регистра MOBPAGE.

A.33 Регистры контроллера Ethernet 10/100/1000

Базовый адрес: 5000_B000h

CTRL – регистр управления

Смещение: + 00h
Сброс: 00h

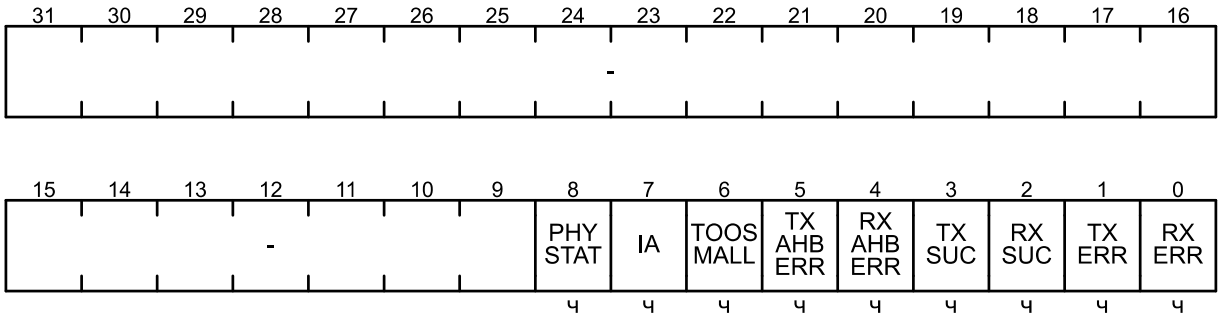


Поле	Биты	Описание	
MDIOEN	26	MDIO прерывания включены. Значение установлено в «1», если контроллер Ethernet поддерживает прерывания MDIO	
MC	25	Доступность многоадресной рассылки. Значение установлено в «1», если контроллер Ethernet поддерживает прием многоадресной рассылки адресов	
ME	11	Бит разрешения приема многоадресной рассылки. Значение по сбросу – 0.	
PI	10	Бит разрешения прерывания при изменении состояния PHY	
BM	9	Бит включения пакетного режима (Burstmode). Режим работы не определен, если для других скоростных режимов установлено значение 1. Значение по сбросу – 0.	
		0	В этом скоростном режиме всегда используются обычные передачи с установленным расширением.
		1	При передаче используется пакетный режим в полудуплексном режиме со скоростью 1000 Мбит/с (бит GB=1, бит FD=0)
GB	8	Бит включения гигабитного режима. Значение по сбросу – 0.	
		0	Режим скорости выбирается битом 7 (SPEED)
		1	Режим скорости 1000 Мбит
SPEED	7	Бит установления текущего режима скорости. Данный бит не должен быть установлен одновременно с битом 8 (GB). Значение по сбросу – 0.	
		0	Режим скорости 10 Мбит
		1	Режим скорости 100 Мбит
RST	6	Сброс. Запись единицы в этот бит сбрасывает контроллер Ethernet. После выполнения сброса данный бит очищается самостоятельно. После установки не следует выполнять никаких других действий, кроме опроса данного бита то тех пор, пока он не будет очищен.	

Поле	Биты	Описание	
PR	5	Неразборчивый режим. Если данный бит установлен, контроллер Ethernet работает в неразборчивом режиме, т. е. будет получать все пакеты независимо от адреса назначения. Значение по сбросу – 0.	
FD	4	Полный дуплекс. Установка данного бита переводит контроллер Ethernet в полнодуплексный режим. Значение по сбросу – 0.	
		0	Режим полудуплекс
		1	Режим полный дуплекс
RXINT	3	Включение прерывания приемника. При установке данного бита прерывание будет генерироваться каждый раз при получении пакета. Прерывание генерируется независимо от того, был ли пакет получен успешно или завершен с ошибкой. Значение по сбросу – 0.	
TXINT	2	Включение прерывания передатчика. При установке данного бита прерывание будет генерироваться каждый раз при передаче пакета. Прерывание генерируется независимо от того, был ли пакет передан успешно или завершен с ошибкой. Значение по сбросу – 0.	
RXEN	1	Бит включения приема. Должен устанавливаться каждый раз при включении новых дескрипторов. Пока бит установлен, контроллер Ethernet будет считывать новые дескрипторы. Как только будет обнаружен отключенный дескриптор, он остановится до тех пор, пока бит RXEN не будет установлен снова. Данный бит должен быть установлен после включения новых дескрипторов. Значение по сбросу – 0.	
TXEN	0	Бит включения передачи. Должен устанавливаться каждый раз при включении новых дескрипторов. Пока бит установлен, контроллер Ethernet будет считывать новые дескрипторы. Как только будет обнаружен отключенный дескриптор, он остановится до тех пор, пока бит TXEN не будет установлен снова. Данный бит должен быть установлен после включения новых дескрипторов. Значение по сбросу – 0.	
–	31-27, 24-12	Зарезервировано	

STATUS – регистр состояния

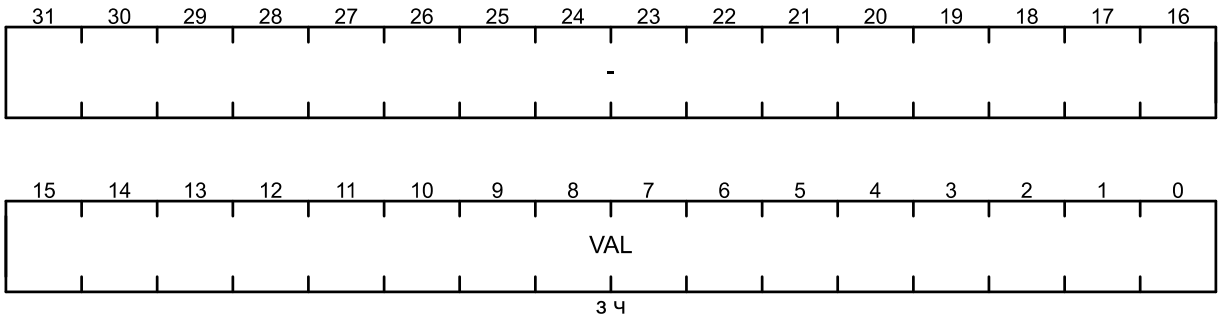
Смещение: + 04h
Сброс: 00h



Поле	Биты	Описание
PHYSTAT	8	Бит изменения состояния PHY. Устанавливается каждый раз при обнаружении изменения статуса PHY
IA	7	Получение пакета с адресом, не принятым MAC. Очищается записью единицы. Значение по сбросу – 0.
TOOSMALL	6	Получение пакета, размер которого меньше минимального размера. Очищается записью единицы. Значение по сбросу – 0.
TXAHBERR	5	Ошибка AHB передатчика. В модуле DMA передатчика обнаружена ошибка AHB. Очищается записью единицы.
RXAHBERR	4	Ошибка AHB приемника. В модуле DMA приемника обнаружена ошибка AHB. Очищается записью единицы.
TXSUC	3	Успешная передача. Пакет был передан без ошибок. Очищается записью единицы.
RXSUC	2	Успешный прием. Пакет был получен без ошибок. Очищается записью единицы.
TXERR	1	Ошибка передатчика. Был передан пакет, который завершился с ошибкой. Очищается записью единицы.
RXERR	0	Ошибка приемника. Был получен пакет, который завершился с ошибкой. Очищается записью единицы.
–	31-9	Зарезервировано

MACMSB – регистр старших байт MAC-адреса

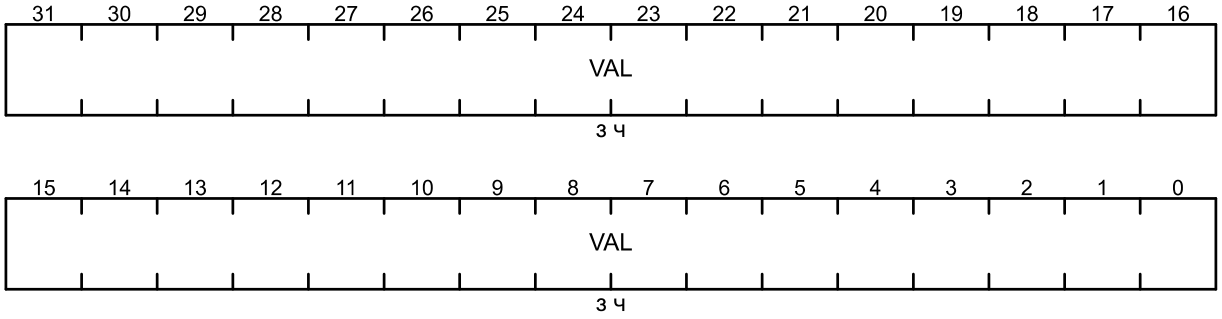
Смещение: + 08h
Сброс: 00h



Поле	Биты	Описание
VAL	15-0	Два старших байта MAC-адреса.
-	31-16	Зарезервировано

MACLSB – регистр младших байт MAC-адреса

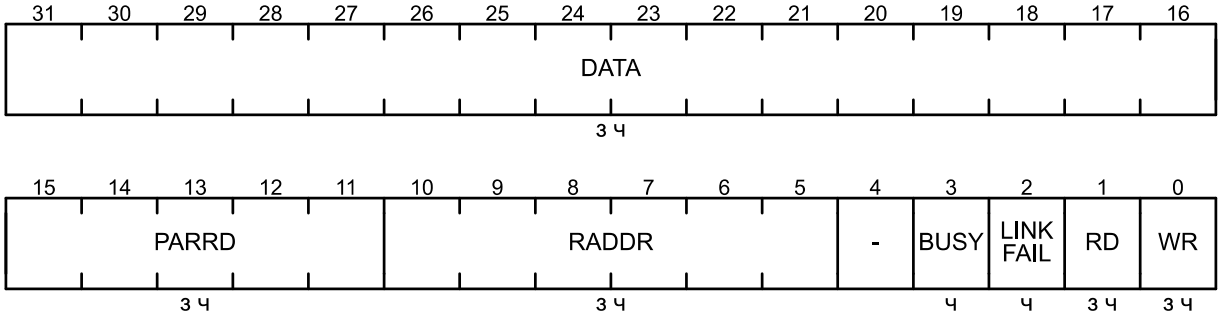
Смещение: + 0Ch
Сброс: 00h



Поле	Биты	Описание
VAL	31-0	4 младших байта MAC-адреса.

MDIOCSR – регистр управления/состояния MDIO

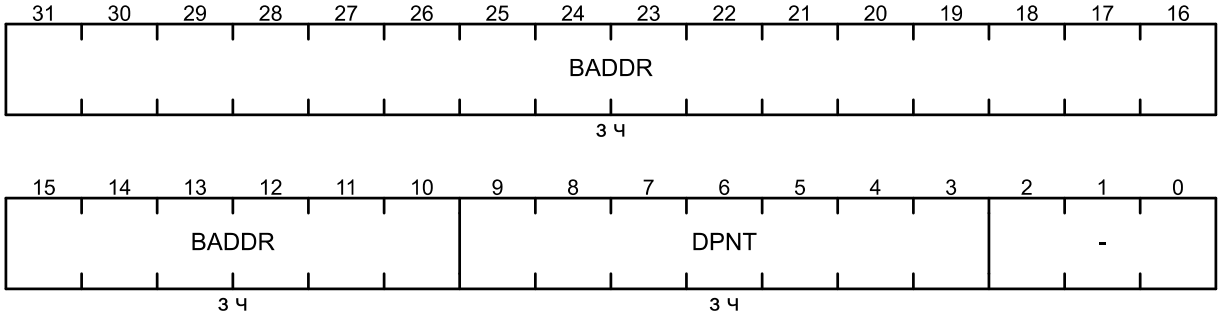
Смещение: + 10h
Сброс: 00h



Поле	Биты	Описание
DATA	31-16	Поле данных. Поле содержит данные, считанные во время операции чтения. Во время передачи, передаваемые данные также берутся из этого поля. Значение по сбросу – 0.
PADDR	15-11	Поле адреса PHY. Данное поле содержит адрес PHY, к которому следует обращаться во время операции записи или чтения. Значение по сбросу – 0.
RADDR	10-6	Поле адреса регистра. Это поле содержит адрес регистра, к которому следует обращаться во время операции записи или чтения. Значение по сбросу – 0.
BUSY	3	Бит занятости. При выполнении операции этот бит устанавливается в единицу. Как только операция завершена и канал управления свободен, этот бит сбрасывается. Значение по сбросу – 0.
LINKFAIL	2	Бит сбоя соединения. При завершении операции (BUSY = 0), этот бит устанавливается, если не было обнаружено функциональное соединение. Значение по сбросу – 0.
RD	1	Бит запуска операции чтения в интерфейсе управления. Данные сохраняются в поле DATA. Значение по сбросу – 0.
WR	0	Бит запуска операции записи в интерфейсе управления. Данные берутся из поля DATA. Значение по сбросу – 0.
–	5-4	Зарезервировано

TXBASE – регистр базового адреса таблицы дескрипторов передатчика

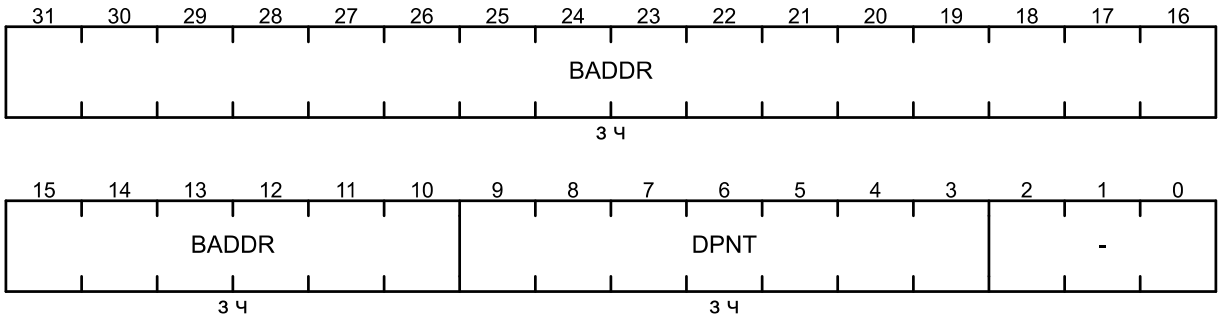
Смещение: + 14h
Сброс: 00h



Поле	Биты	Описание
BADDR	31-10	Поле содержит базовый адрес таблицы дескрипторов передатчика. Нет сброса
DPNT	9-3	Поле с указателем на отдельные дескрипторы. Автоматически инкрементируется MAC Ethernet
–	2-0	Зарезервировано

RXBASE – регистр базового адреса таблицы дескрипторов приемника

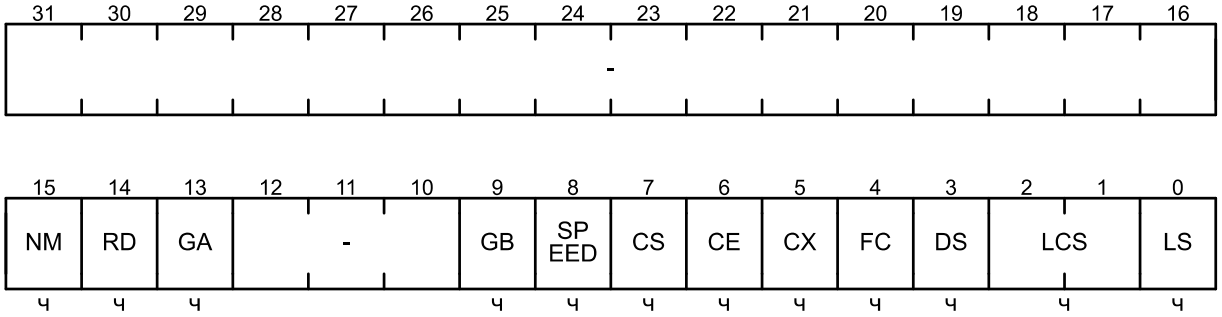
Смещение: + 18h
Сброс: 00h



Поле	Биты	Описание
BADDR	31-10	Поле содержит базовый адрес таблицы дескрипторов приемника. Нет сброса
DPNT	9-3	Поле с указателем на отдельные дескрипторы. Автоматически инкрементируется MAC Ethernet
–	2-0	Зарезервировано

RGMII_STAT – регистр статуса RGMII

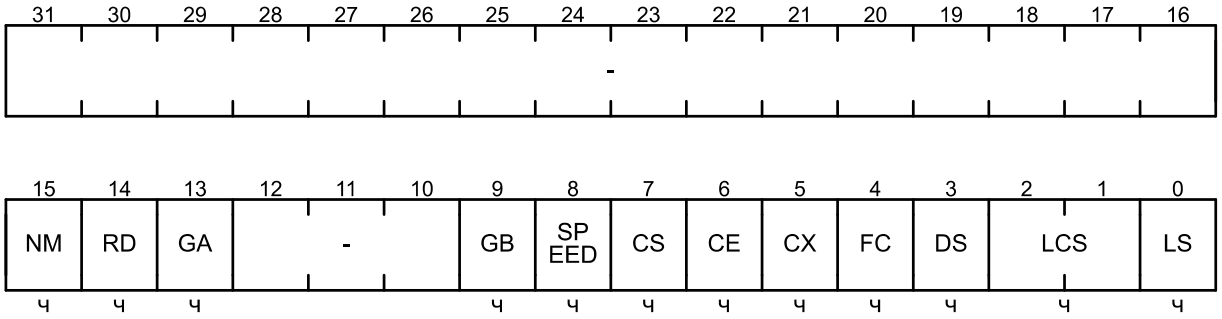
Смещение: + 40h
Сброс: 00h



Поле	Биты	Описание	
NM	15	Режим работы без тактового мультиплексора. Если данный бит установлен, внутренняя тактовая логика не используется	
RD	14	Включена отладка RAM. Если данный бит установлен, доступна отладка RAM	
GA	13	Доступен гигабитный MAC. Если данный бит установлен, MAC имеет возможность работы на скорости 1000 Мбит	
GB	9	Гигабитный режим. Установленный бит указывает, что текущий режим скорости MAC составляет 1000 Мбит. Если данный бит сброшен, скоростной режим определяется с помощью бита SPEED	
SPEED	8	Режим скорости. Указывает текущий скоростной режим MAC. Не должен быть установлен одновременно с установленным битом GB.	
		0	режим 10 Мбит
		1	режим 100 Мбит
CS	7	Обнаружение несущей частоты	
CE	6	Ошибка расширения кадров несущей	
CX	5	Расширение кадров несущей	
FC	3	Обнаружение ложной несущей частоты	
DS	4	Статус дуплекса	
		0	полудуплекс
		1	полный дуплекс
LCS	2-1	Индикация тактовой частоты RXC	
		00b	2,5 МГц
		01b	25 МГц
		10b	125 МГц
		11b	Зарезервировано
LS	0	Статус линка	
		0	отключен
		1	включен
–	31-16, 12-10	Зарезервировано	

RGMII_IF – регистр статуса флагов прерываний RGMII

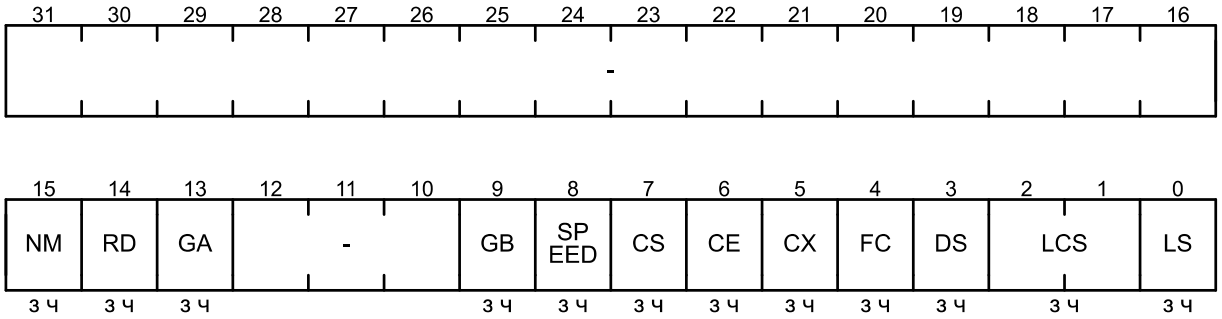
Смещение: + 44h
Сброс: 00h



Поле	Биты	Описание
NM	15	Бит статуса прерывания по изменению режима работы тактового мультиплексора
RD	14	Бит статуса прерывания по включению отладки RAM
GA	13	Бит статуса прерывания доступности гигабитного режима MAC
GB	9	Бит статуса прерывания при переходе в гигабитный режим
SPEED	8	Бит статуса прерывания при переходе в режим 100 Мбит
CS	7	Бит статуса прерывания при обнаружении несущей частоты
CE	6	Бит статуса прерывания при ошибке расширения кадров
CX	5	Бит статуса прерывания при расширении кадров
FC	3	Бит статуса прерывания при обнаружении ложной несущей частоты
DS	4	Бит статуса прерывания при изменении статуса дуплекса
LCS	2-1	Бит статуса прерывания при изменении тактовой частоты RXC
LS	0	Бит статуса прерывания при изменении статуса линка
–	31-16, 12-10	Зарезервировано

RGMII_IM – регистр маски флагов прерываний RGMII

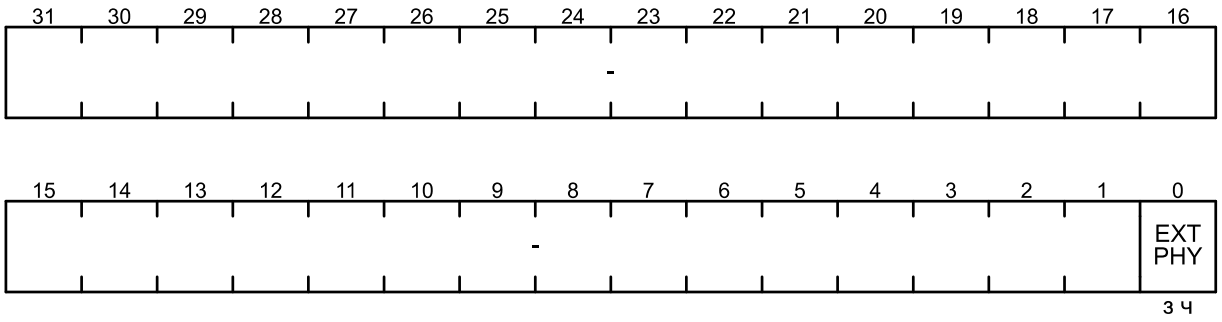
Смещение: + 48h
Сброс: 00h



Поле	Биты	Описание
NM	15	Бит маски прерывания по изменению режима работы тактового мультиплексора
RD	14	Бит маски прерывания по включению отладки RAM
GA	13	Бит маски прерывания доступности гигабитного режима MAC
GB	9	Бит маски прерывания при переходе в гигабитный режим
SPEED	8	Бит маски прерывания при переходе в режим 100 Мбит
CS	7	Бит маски прерывания при обнаружении несущей частоты
CE	6	Бит маски прерывания при ошибке расширения кадров
CX	5	Бит маски прерывания при расширении кадров
FC	3	Бит маски прерывания при обнаружении ложной несущей частоты
DS	4	Бит маски прерывания при изменении статуса дуплекса
LCS	2-1	Бит маски прерывания при изменении тактовой частоты RXC
LS	0	Бит маски прерывания при изменении статуса линка
–	31-16, 12-10	Зарезервировано

PHYCFG – регистр управления внешним PHY

Смещение: + 7Ch
Сброс: 0000_0001h



Поле	Биты	Описание	
EXTPHY	0	Использование внешнего PHY	
		0	используется внутренний PHY
		1	используется внешняя микросхема PHY
–	31-1	Зарезервировано	

Приложение Б
(обязательное)
Коды состояний функционирования блока I2C

В таблицах Б.1 – Б.11 представлена информация о соответствии кодов и операций.

Условные обозначения, принятые в данных таблицах:

- [ADR, 0], [ADR, 1] – 8-разрядное значение, состоящее из 7-разрядного адреса ADR и бита направления передачи R/W#, значение которого «0» или «1» указывается непосредственно;
- DAT – байт данных;
- код мастера – 8-разрядное значение 0000_1xxxb, где «xxx» – уникальный код каждого мастера в системе нескольких устройств;
- «с ACK» – выражение, обозначающее, что после передачи адреса/байта в ответ на запрос подтверждения передачи (бит ACK) передатчик получает подтверждение передачи от ведомого (квитирование);
- «с NACK» – выражение, обозначающее, что после передачи адреса/байта в ответ на запрос подтверждения передачи (бит ACK) передатчик получает неподтверждение передачи от ведомого (неквитирование);
- X – бит может быть установленным (1b) или сброшенным (0b), в зависимости от режима работы, состояния и дальнейших действий модуля I2C.

Таблица Б.1 – Исключительные состояния

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
00h	IDLE	–	–	–	–	–	Ожидать завершения текущей передачи байта
1Fh	Ошибка на шине	–	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)

Таблица Б.2 – Режим FS мастера передатчика (дополнительно см. таблицу Б.4)

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
1	2	3	4	5	6	7	8
01h	Старт	Код мастера	1	0	0	0	Передать код мастера и перейти в режим HS (0Ch/21h)
		[ADR, 0]					Передать адрес ведомого (04h/05h)
02h	Повторный старт	[ADR, 0]	1	0	0	0	Передать адрес ведомого (04h/05h)
		[ADR, 1]					Передать адрес ведомого, после чего перейти в режим приемника (08h/09h)

Окончание таблицы Б.2

1	2	3	4	5	6	7	8
03h	Потеря арбитража, мастер перешел в режим безадресного ведомого	–	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
04h	Отправлен адрес ведомого с ACK	DAT	1	0	0	0	Передать байт данных (06h/07h)
		–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
05h	Отправлен адрес ведомого с NACK	–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
06h	Отправлен байт данных с ACK	DAT	1	0	0	0	Передать байт данных (06h/07h)
		–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
07h	Отправлен байт данных с NACK	–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)

Таблица Б.3 – Режим FS мастера приемника

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
1	2	3	4	5	6	7	8
08h	Отправлен адрес ведомого с ACK	–	1	0	0	0	Получить байт данных, квитировать прием (0Ah)
			1	1	0	0	Получить байт данных, не квитировать прием (0Bh)
09h	Отправлен адрес ведомого с NACK	–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
0Ah	Принят байт данных и квити-рован	DAT	1	0	0	0	Получить байт данных, квитировать прием (0Ah)
			1	1	0	0	Получить байт данных, не квитировать прием (0Bh)

Окончание таблицы Б.3

1	2	3	4	5	6	7	8
0Bh	Принят байт данных и не квити-рован	DAT	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)

Таблица Б.4 – Режим FS мастера передатчика (дополнительно см. таблицу Б.2)

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
0Ch	Отправлен код мастера, обнаружена ошибка (ACK)	–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)

Таблица Б.5 – Режим FS ведомого приемника (дополнительно см. таблицу Б.7)

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
10h	Принят адрес и квити-рован	–	1	0	0	0	Получить байт данных, квитировать прием (12h)
			1	1	0	0	Получить байт данных, не квитировать прием (13h)
11h	Принят адрес пос-лепотери арбитража и квити-рован	–	1	0	0	0	Получить байт данных, квитировать прием (12h)
			1	1	0	0	Получить байт данных, не квитировать прием (13h)
12h	Принят байт данных и квити-рован	DAT	1	0	0	0	Получить байт данных, квитировать прием (12h)
			1	1	0	0	Получить байт данных, не квитировать прием (13h)
13h	Принят байт данных и не квити-рован	DAT	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	0	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)

Таблица Б.6 – Режим FS ведомого передатчика

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
1	2	3	4	5	6	7	8
14h	Принят адрес и квити-рован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (16h/17h)
15h	Принят адрес после потери арбитража и квити-рован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (16h/17h)
16h	Отправлен байт данных с ACK	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (16h/17h)
17h	Отправлен байт данных с NACK	–	1	X	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	X	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)
18h	Принят адрес отклика и квити-рован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (1Ah/1Bh)
19h	Принят адрес отклика после потери арбитража и квити-рован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (1Ah/1Bh)
1Ah	Отправлен байт данных в ответ на получение адреса отклика с ACK	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (1Ah/1Bh)

Окончание таблицы Б.6

1	2	3	4	5	6	7	8
1Bh	Отправлен байт данных в ответ на получение адреса отклика с NACK	–	1	X	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	X	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)

Таблица Б.7 – Режим FS ведомого приемника (дополнительно см. таблицу Б.5)

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
1Ch	Стоп	–	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	0	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)
1Dh	Принят адрес общего вызова и квити-рован	–	1	0	0	0	Получить байт данных, квитировать прием (12h)
			1	1	0	0	Получить байт данных, не квитировать прием (13h)
1Eh	Принят адрес общего вызова после потери арбитража и квити-рован	–	1	0	0	0	Получить байт данных, квитировать прием (12h)
			1	1	0	0	Получить байт данных, не квитировать прием (13h)

Таблица Б.8 – Режим HS мастера передатчика

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
21h	Успешно отправлен код мастера, мастер перешел в режим HS	–	1	0	0	1	Сделать повторный старт (22h)
22h	Повторный старт	[ADR, 0]	1	0	0	0	Передать адрес ведомого (28h/29h)
		[ADR, 1]					Передать адрес ведомого, после квитирования/не квитирования переключиться в режим мастера приемника (28h/29h)
23h	Потеря арбитража, мастер перешел в режим HS безадресного ведомого	–	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
24h	Отправлен адрес ведомого с ACK	DAT	1	0	0	0	Передать байт данных (26h/27h)
		–	1	0	0	1	Сделать повторный старт (22h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
25h	Отправлен адрес ведомого с NACK	–	1	0	0	1	Сделать повторный старт (22h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
26h	Отправлен байт данных с ACK	DAT	1	0	0	0	Передать байт данных (26h/27h)
		–	1	0	0	1	Сделать повторный старт (22h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
27h	Отправлен байт данных с NACK	–	1	0	0	1	Сделать повторный старт (22h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)

Таблица Б.9 – Режим HS мастера приемника

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
28h	Отправлен адрес ведомого с ACK	–	1	0	0	0	Получить байт данных, квитировать прием (2Ah)
			1	1	0	0	Получить байт данных, не квитировать прием (2Bh)
29h	Отправлен адрес ведомого с NACK	–	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)
2Ah	Принят байт данных и квити-рован	DAT	1	0	0	0	Получить байт данных, квитировать прием (2Ah)
			1	1	0	0	Получить байт данных, не квитировать прием (2Bh)
2Bh	Принят байт данных и не квити-рован	DAT	1	0	0	1	Сделать повторный старт (02h)
			1	0	1	0	Остановить передачу (00h)
			1	0	1	1	Остановить передачу, а затем сделать повторный старт (01h)

Таблица Б.10 – Режим HS ведомого приемника

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
30h	Принят адрес и квити-рован	–	1	0	0	0	Получить байт данных, квитировать прием (32h)
			1	1	0	0	Получить байт данных, не квитировать прием (33h)
32h	Принят байт данных и квити-рован	DAT	1	0	0	0	Получить байт данных, квитировать прием (32h)
			1	1	0	0	Получить байт данных, не квитировать прием (33h)
33h	Принят байт данных и не квити-рован	DAT	1	0	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	0	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)

Таблица Б.11 – Режим HS ведомого передатчика

Код	Описание состояния	Регистр SDA	Биты регистра CTL0				Возможные дальнейшие действия и коды результатов их выполнения
			CLRST	ACK	STOP	START	
34h	Принят адрес и квити-рован	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (36h/37h)
36h	Отправлен байт данных с ACK	DAT	1	X	0	0	Передать байт данных, квитировать/ не квитировать (36h/37h)
37h	Отправлен байт данных с NACK	–	1	X	0	0	Функционировать в режиме безадресного ведомого (00h)
			1	X	0	1	Функционировать в режиме безадресного ведомого, сделать старт после освобождения шины (00h, 01h)

Приложение В
(обязательное)
Назначение выводов в корпусе

Функции выводов микроконтроллера К1921ВГ1Т указаны на рисунке В.1, функции выводов микроконтроллера К1921ВГ10Я указаны на рисунке В.2, альтернативные функции выводов GPIO микроконтроллеров К1921ВГ1Т, К1921ВГ10Я приведены в таблице В.1.

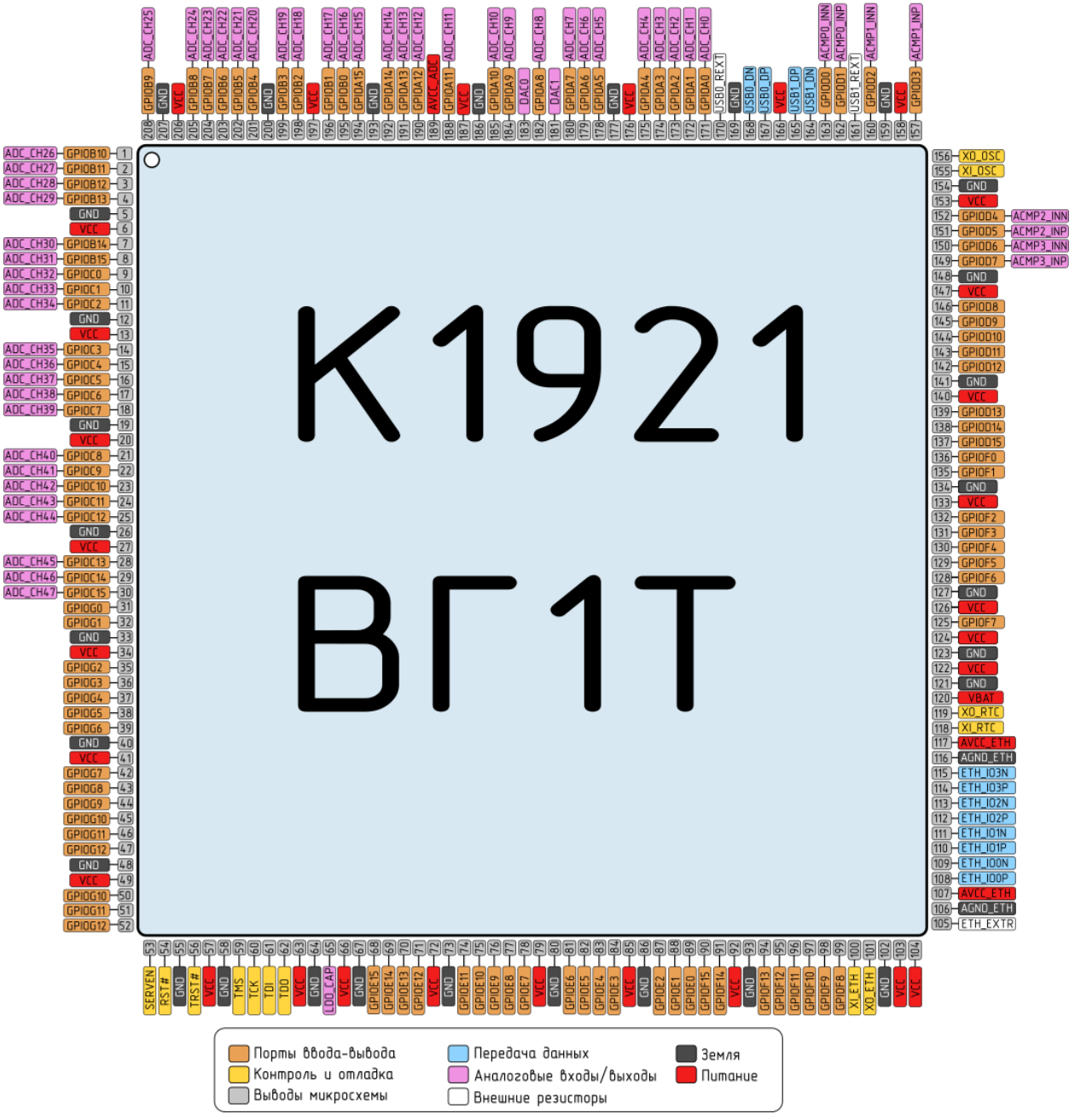


Рисунок В.1 – Назначение выводов микроконтроллера К1921ВГ1Т в корпусе LQFP-208

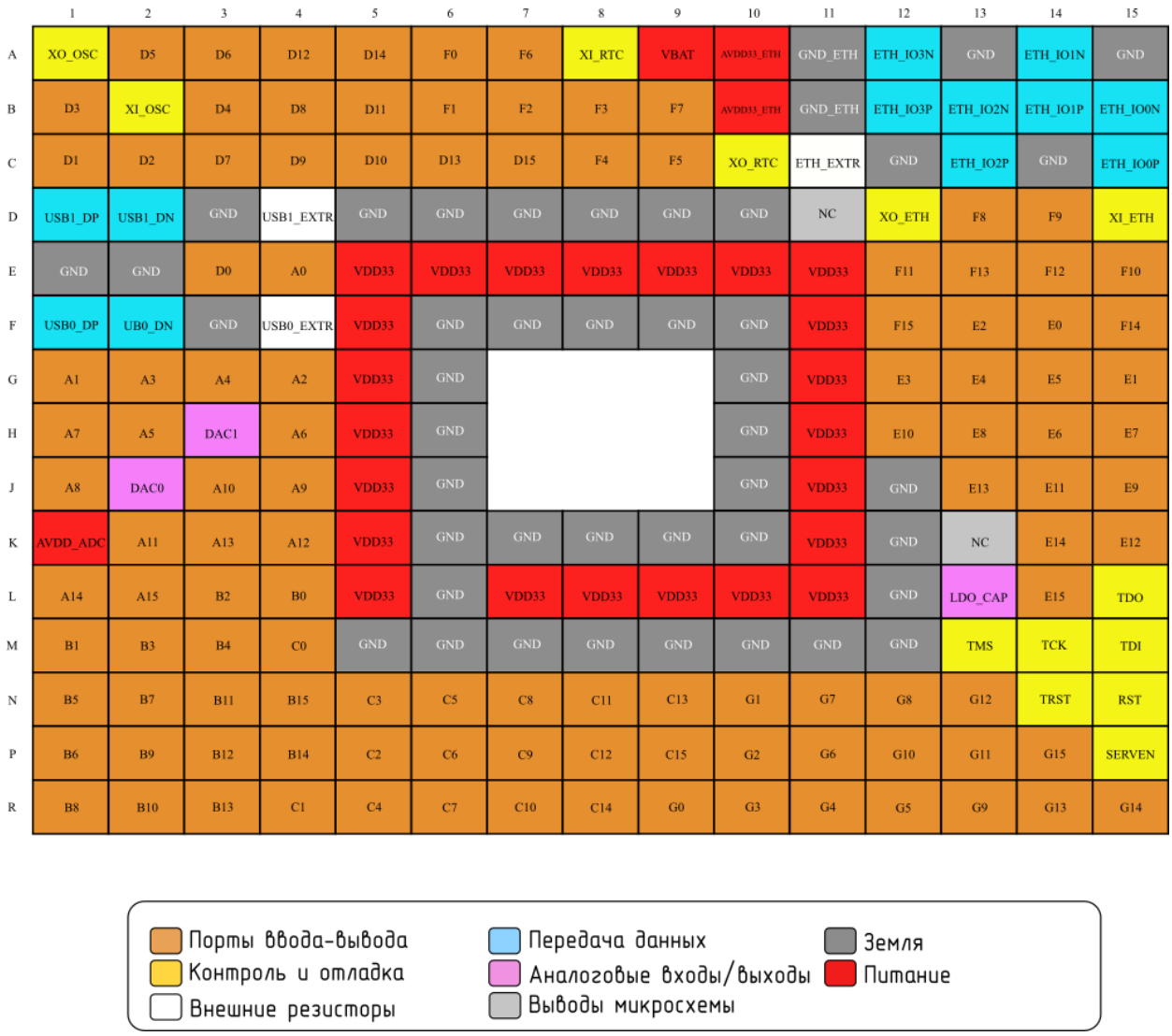


Рисунок В.2 – Назначение выводов микроконтроллера К1921ВГ10Я в корпусе BGA-216

Таблица В.1 – Альтернативные функции портов GPIO К1921ВГ1Т и К1921ВГ10Я

Вывод GPIO	Альтернативная функция				
	1	2	3	4	5
GPIOA					
A0	SPI0_CLK	TMR14_IO	CANFD0_RX	USB1_RESET	TMR0_IO0
A1	SPI0_FSS	TMR15_IO	CANFD0_TX	USB1_CLK	TMR0_IO1
A2	SPI0_RX	CLKOUT	CANFD1_RX	USB1_XCVRSELECT0	TMR0_IO2
A3	SPI0_TX	EXTCLK	CANFD1_TX	USB1_XCVRSELECT1	TMR0_IO3
A4	SPI1_CLK	PWM0_A	CANFD2_RX	USB1_TERMSELECT	TMR1_IO0
A5	SPI1_FSS	PWM0_B	CANFD2_TX	USB1_SUSPENDM	TMR1_IO1
A6	SPI1_RX	PWM1_A	CANFD3_RX	USB1_OPMODE0	TMR1_IO2
A7	SPI1_TX	PWM1_B	CANFD3_TX	USB1_OPMODE1	TMR1_IO3
A8	SPI2_CLK	PWM2_A	I2S1_BCLKT	USB1_LINESTATE0	TMR2_IO0
A9	SPI2_FSS	PWM2_B	I2S1_LRCLKT	USB1_LINESTATE1	TMR2_IO1
A10	SPI2_RX	PWM3_A	I2S1_BCLKR	USB1_TXVALID	TMR2_IO2
A11	SPI2_TX	PWM3_B	I2S1_LRCLKR	USB1_TXREADY	TMR2_IO3
A12	SPI3_CLK	PWM4_A	I2S1_SDI	USB1_RXACTIVE	TMR3_IO0
A13	SPI3_FSS	PWM4_B	I2S1_SDO	USB1_RXVALID	TMR3_IO1
A14	SPI3_RX	PWM5_A	I2S1_MCLK	USB1_RXERROR	TMR3_IO2
A15	SPI3_TX	PWM5_B	TMR0_IO0	USB1_VBUSVALID	TMR3_IO3
GPIOB					
B0	CAN0_RX	SPI4_CLK	UART0_RX	USB1_CHRGVBUS	TMR4_IO
B1	CAN0_TX	SPI4_FSS	UART0_TX	USB1_DISCHRGVBUS	TMR5_IO
B2	CAN1_RX	SPI4_RX	LIN0_RX	USB1_DPPULLDOWN	TMR6_IO
B3	CAN1_TX	SPI4_TX	LIN0_TX	USB1_DMPULLDOWN	TMR7_IO
B4	CAN2_RX	SPI5_CLK	UART1_RX	USB1_IDPULLUP	TMR8_IO
B5	CAN2_TX	SPI5_FSS	UART1_TX	USB1_HOSTDISCONNECT	TMR9_IO
B6	CAN3_RX	SPI5_RX	LIN1_RX	USB1_DATA0	TMR10_IO
B7	CAN3_TX	SPI5_TX	LIN1_TX	USB1_DATA1	TMR11_IO
B8	ACMP0_OUT	SPI6_CLK	UART2_RX	USB1_DATA2	TMR12_IO
B9	ACMP1_OUT	SPI6_FSS	UART2_TX	USB1_DATA3	TMR13_IO
B10	ACMP2_OUT	SPI6_RX	LIN2_RX	USB1_DATA4	TMR14_IO
B11	ACMP3_OUT	SPI6_TX	LIN2_TX	USB1_DATA5	TMR15_IO
B12	TMR0_IO0	SPI7_CLK	UART3_RX	USB1_DATA6	TMR12_IO
B13	TMR1_IO0	SPI7_FSS	UART3_TX	USB1_DATA7	TMR13_IO
B14	I2C1_SDA	SPI7_RX	LIN3_RX	UART0_CTS	TMR0_IO0
B15	I2C1_SCL	SPI7_TX	LIN3_TX	UART0_RTS	TMR0_IO1
GPIOC					
C0	PWM0_A	QSPI1_CLK	TMR11_IO	MSC_DO	TMR0_IO2
C1	PWM0_B	QSPI1_FSS	TMR12_IO	MSC_CLK	TMR0_IO3
C2	PWM1_A	QSPI1_IO0	TMR1_IO0	MSC_DI0	TMR15_IO
C3	PWM1_B	QSPI1_IO1	TMR2_IO0	MSC_SS0	TMR1_IO1
C4	PWM2_A	QSPI1_IO2	TMR3_IO0	MSC_SS1	TMR1_IO2
C5	PWM2_B	QSPI1_IO3	TMR4_IO	MSC_SS2	TMR1_IO3
C6	PWM3_A	TMR10_IO	ACMP0_OUT	I2C2_SDA	TMR2_IO0
C7	PWM3_B	TMR11_IO	ACMP1_OUT	I2C2_SCL	TMR2_IO1
C8	PWM4_A	TMR12_IO	SPI0_CLK	–	TMR2_IO2
C9	PWM4_B	I2S0_BCLKT	SPI0_FSS	–	TMR2_IO3
C10	PWM5_A	I2S0_LRCLKT	SPI0_RX	–	TMR3_IO0
C11	PWM5_B	I2S0_BCLKR	SPI0_TX	–	TMR3_IO1
C12	PWM_TZ0	I2S0_LRCLKR	SPI1_CLK	–	TMR3_IO2
C13	PWM_TZ1	I2S0_SDI	SPI1_FSS	USB0_CLK	TMR3_IO3
C14	PWM_TZ2	I2S0_SDO	SPI1_RX	I2C3_SDA	TMR4_IO
C15	PWM_TZ3	I2S0_MCLK	SPI1_TX	I2C3_SCL	TMR5_IO

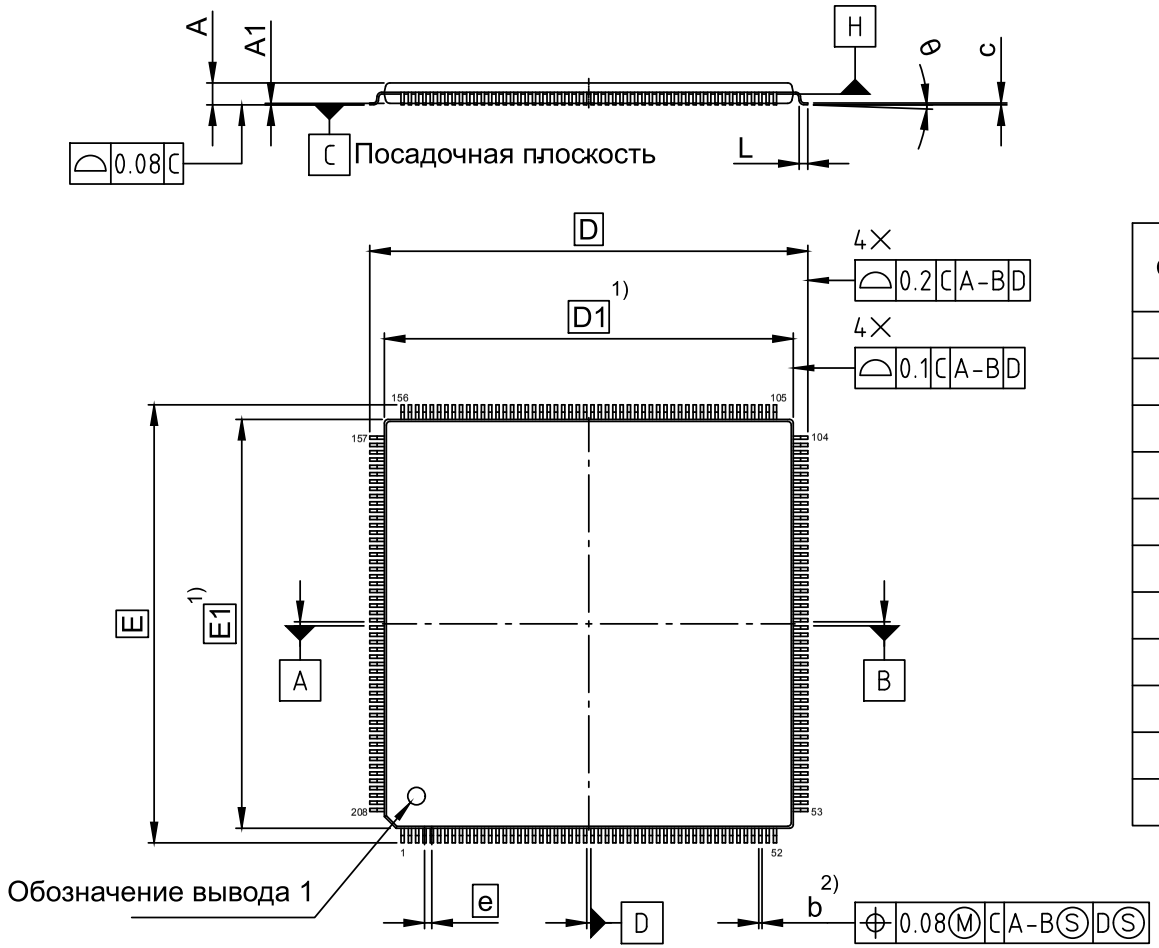
Продолжение таблицы B.1

Вывод GPIO	Альтернативная функция				
	1	2	3	4	5
GPIOD					
D0	PWM14_A	I2C1_SDA	TMR5_IO	UART1_CTS	TMR6_IO
D1	PWM14_B	I2C1_SCL	TMR6_IO	UART1_RTS	TMR7_IO
D2	PWM15_A	I2C0_SDA	TMR7_IO	–	TMR8_IO
D3	PWM15_B	I2C0_SCL	TMR8_IO	GMII_TXCLK	TMR9_IO
D4	PWM_SYNC	ACMP2_OUT	TMR9_IO	GMII_GTXCLK	TMR10_IO
D5	PWM_TZ4	ACMP3_OUT	TMR10_IO	GMII_RXCLK	TMR11_IO
D6	EXTCLK	TMR6_IO	QSPI0_CLK	–	TMR12_IO
D7	CLKOUT	SDR_CLKOUT	QSPI0_FSS	–	TMR14_IO
D8	CANFD0_RX	MEM_ADDR0	QSPI0_IO0	GMII_RXDV	TMR13_IO
D9	CANFD0_TX	MEM_ADDR1	QSPI0_IO1	GMII_RXER	TMR15_IO
D10	CANFD1_RX	MEM_ADDR2	QSPI0_IO2	GMII_MDIO	TMR1_IO0
D11	CANFD1_TX	MEM_ADDR3	QSPI0_IO3	GMII_TXEN	TMR13_IO
D12	CANFD2_RX	MEM_ADDR4	SPI2_CLK	GMII_TXER	TMR0_IO0
D13	CANFD2_TX	MEM_ADDR5	SPI2_FSS	GMII_CRS	TMR0_IO1
D14	CANFD3_RX	MEM_ADDR6	SPI2_RX	GMII_COL	TMR0_IO2
D15	CANFD3_TX	MEM_ADDR7	SPI2_TX	GMII_MDC	TMR0_IO3
GPIOE					
E0	UART4_RX	MEM_ADDR8	SPI6_CLK	GMII_TXD0	TMR1_IO0
E1	UART4_TX	MEM_ADDR9	SPI6_FSS	GMII_TXD1	TMR1_IO1
E2	UART5_RX	MEM_ADDR10	SPI6_RX	GMII_TXD2	TMR1_IO2
E3	UART5_TX	MEM_ADDR11	SPI6_TX	GMII_TXD3	TMR1_IO3
E4	LIN4_RX	MEM_ADDR12	PWM15_A	GMII_TXD4	TMR2_IO0
E5	LIN4_TX	MEM_ADDR13 (SDR_BA0)	PWM15_B	GMII_TXD5	TMR2_IO1
E6	LIN5_RX	MEM_ADDR14 (SDR_BA1)	PWM14_A	GMII_TXD6	TMR2_IO2
E7	LIN5_TX	MEM_ADDR15 (SDR_CKE)	PWM14_B	GMII_TXD7	TMR2_IO3
E8	UART6_RX	MEM_ADDR16 (SDR_WEN)	PWM_TZ5	GMII_RXD0	TMR3_IO0
E9	UART6_TX	MEM_ADDR17 (SDR_RASN)	TMR13_IO	GMII_RXD1	TMR3_IO1
E10	UART7_RX	MEM_ADDR18 (SDR_CASN)	TMR14_IO	GMII_RXD2	TMR3_IO2
E11	UART7_TX	MEM_ADDR19 (SDR_CSN)	TMR15_IO	GMII_RXD3	TMR3_IO3
E12	LIN6_RX	MEM_ADDR20 (SDR_DQ0)	SPI3_CLK	GMII_RXD4	TMR4_IO
E13	LIN6_TX	MEM_ADDR21 (SDR_DQ1)	SPI3_FSS	GMII_RXD5	TMR5_IO
E14	LIN7_RX	MEM_ADDR22 (SDR_DQ2)	SPI3_RX	GMII_RXD6	TMR6_IO
E15	LIN7_TX	MEM_ADDR23 (SDR_DQ3)	SPI3_TX	GMII_RXD7	TMR7_IO

Продолжение таблицы В.1

Вывод GPIO	Альтернативная функция				
	1	2	3	4	5
GPIOF					
F0	QSPI0_CLK	SDR_DATA16	LIN4_RX	USB0_DATA0	TMR8_IO
F1	QSPI0_FSS	SDR_DATA17	LIN4_TX	USB0_DATA1	TMR9_IO
F2	QSPI0_IO0	SDR_DATA18	LIN5_RX	USB0_DATA2	TMR10_IO
F3	QSPI0_IO1	SDR_DATA19	LIN5_TX	USB0_DATA3	TMR11_IO
F4	QSPI0_IO2	EMC_CEN0 (SDR_DATA20)	UART6_RX	USB0_DATA4	TMR12_IO
F5	QSPI0_IO3	EMC_CEN1 (SDR_DATA21)	UART6_TX	USB0_DATA5	TMR13_IO
F6	I2C0_SDA	EMC_CEN2 (SDR_DATA22)	UART7_RX	USB0_DATA6	TMR14_IO
F7	I2C0_SCL	EMC_CEN3 (SDR_DATA23)	UART7_TX	USB0_DATA7	TMR15_IO
F8	TMR2_IO0	EMC_CEN4 (SDR_DATA24)	SPI4_CLK	–	TMR0_IO0
F9	TMR3_IO0	EMC_CEN5 (SDR_DATA25)	SPI4_FSS	USB0_RESET	TMR0_IO1
F10	TMR4_IO	EMC_CEN6 (SDR_DATA26)	SPI4_RX	USB0_XCVRSELECT0	TMR0_IO2
F11	TMR5_IO	EMC_CEN7 (SDR_DATA27)	SPI4_TX	USB0_XCVRSELECT1	TMR0_IO3
F12	TMR6_IO	EMC_WEN (SDR_DATA28)	SPI5_CLK	USB0_TERMSELECT	TMR1_IO0
F13	TMR7_IO	EMC_LBN (SDR_DATA29)	SPI5_FSS	USB0_SUSPENDM	TMR1_IO1
F14	TMR8_IO	EMC_UBN (SDR_DATA30)	SPI5_RX	USB0_OPMODE0	TMR1_IO2
F15	TMR9_IO	EMC_OEN (SDR_DATA31)	SPI5_TX	USB0_OPMODE1	TMR1_IO3
GPIOG					
G0	PWM6_A	MEM_DATA0	CAN0_RX	USB0_LINESTATE0	TMR2_IO0
G1	PWM6_B	MEM_DATA1	CAN0_TX	USB0_LINESTATE1	TMR2_IO1
G2	PWM7_A	MEM_DATA2	CAN1_RX	USB0_TXVALID	TMR2_IO2
G3	PWM7_B	MEM_DATA3	CAN1_TX	USB0_TXREADY	TMR2_IO3
G4	PWM8_A	MEM_DATA4	UART4_RX	USB0_RXACTIVE	TMR3_IO0
G5	PWM8_B	MEM_DATA5	UART4_TX	USB0_RXVALID	TMR3_IO1
G6	PWM9_A	MEM_DATA6	UART5_RX	USB0_RXERROR	TMR3_IO2
G7	PWM9_B	MEM_DATA7	UART5_TX	USB0_VBUSVALID	TMR3_IO3
G8	PWM10_A	MEM_DATA8	CAN2_RX	USB0_CHRGVBUS	TMR4_IO
G9	PWM10_B	MEM_DATA9	CAN2_TX	USB0_DISCHRGVBUS	TMR5_IO
G10	PWM11_A	MEM_DATA10	CAN3_RX	USB0_DPPULLDOWN	TMR6_IO
G11	PWM11_B	MEM_DATA11	CAN3_TX	USB0_DMPULLDOWN	TMR7_IO
G12	PWM12_A	MEM_DATA12	LIN6_RX	USB0_IDPULLUP	TMR8_IO
G13	PWM12_B	MEM_DATA13	LIN6_TX	USB0_HOSTDISCONNECT	TMR9_IO
G14	PWM13_A	MEM_DATA14	LIN7_RX	MSC_DI1	TMR10_IO
G15	PWM13_B	MEM_DATA15	LIN7_TX	MSC_SS3	TMR11_IO

Примечание – Для выбора альтернативной функции вывода следует записать ее номер в поле соответствующего вывода регистра ALTFUNCNUM0/ALTFUNCNUM1. Для выводов 0-7 используется регистр ALTFUNCNUM0, для выводов 8-15 используется регистр ALTFUNCNUM1.



Символ	Размеры		
	мин.	номин.	макс.
A	-	-	1.7
A1	0.05	-	0.15
b	0.17	0.22	0.27
c	0.09	-	0.2
D	30 BSC		
D1	28 BSC		
e	0.5 BSC		
E	30 BSC		
E1	28 BSC		
L	0.45	0.6	0.75
θ	0°	-	8°

1) Размеры D1 и E1 не включают выступ формы. Допустимый выступ составляет 0,25 мм с каждой стороны.
2) Размер b не включает в себя демпфирующий выступ. Из-за наличия демпфирующих выступов ширина направляющей не должна превышать максимальную ширину b более, чем на 0,08 мм.
Все размеры указаны в мм.

Рисунок Г.1 – Габаритный чертеж корпуса LQFP-208

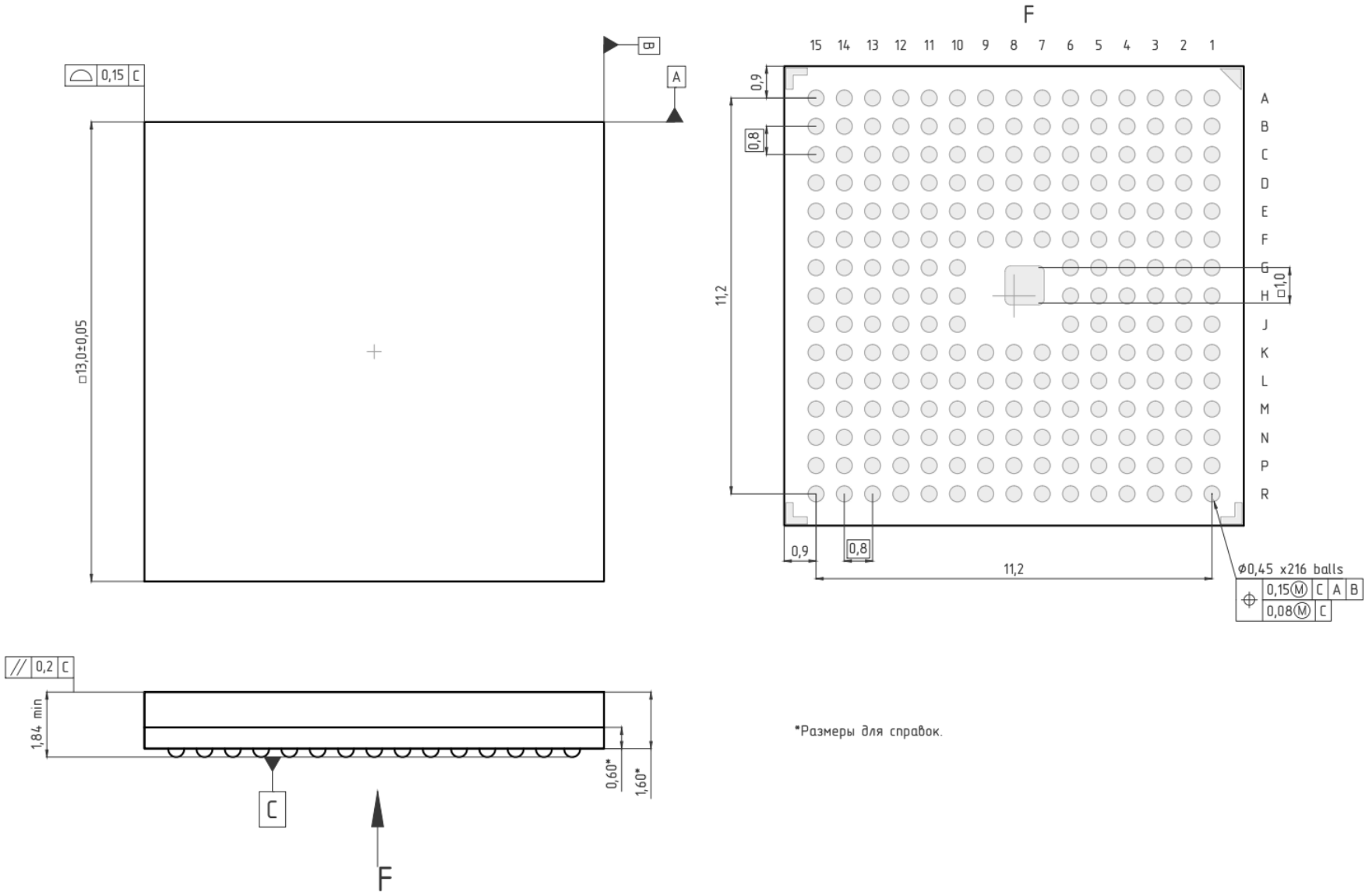


Рисунок Г.2 – Габаритный чертеж корпуса BGA-216

Приложение Д
(справочное)
Примеры схем включения микроконтроллеров

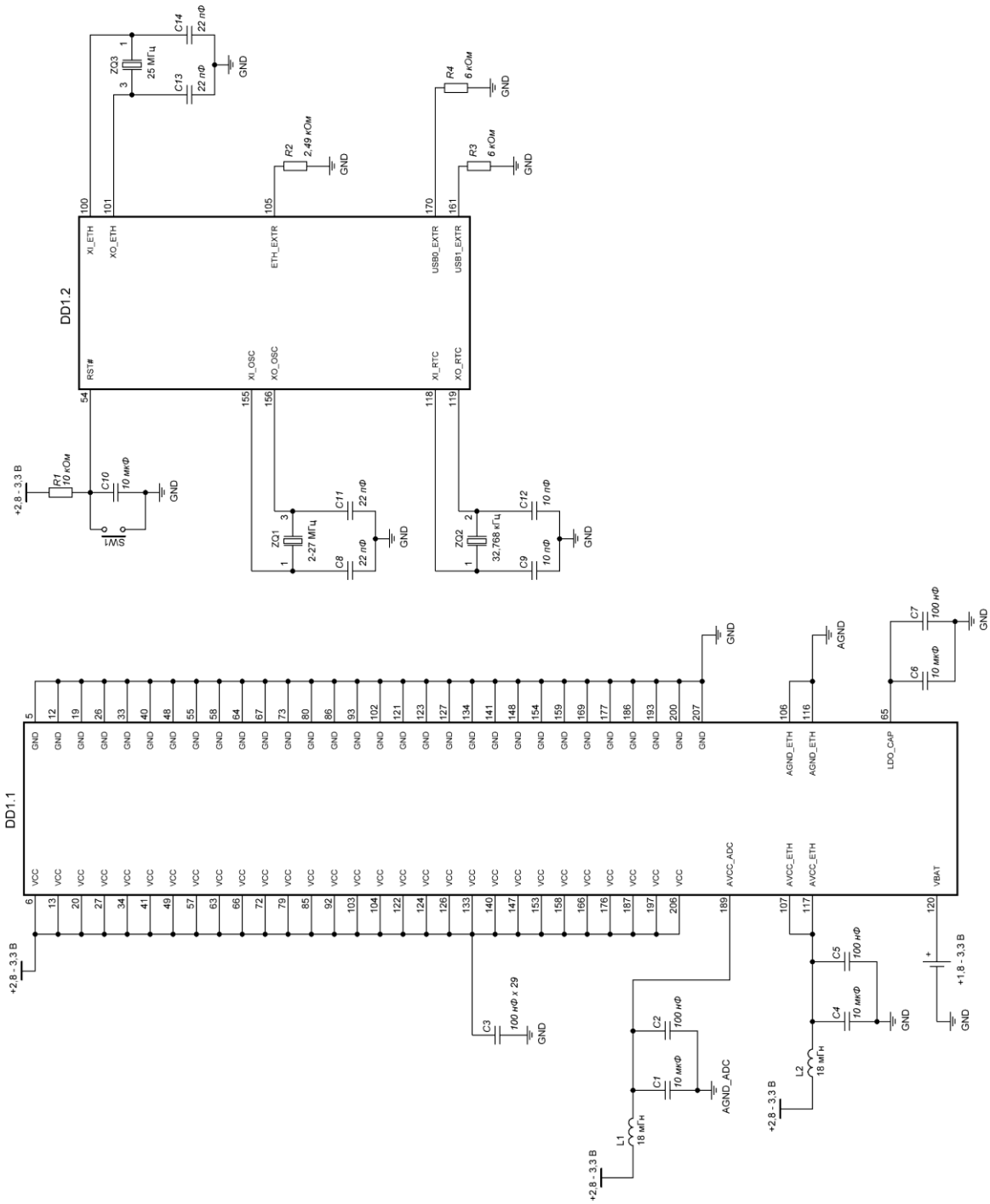


Рисунок Д.1 – Пример схемы включения микроконтроллера К1921ВГ1Т

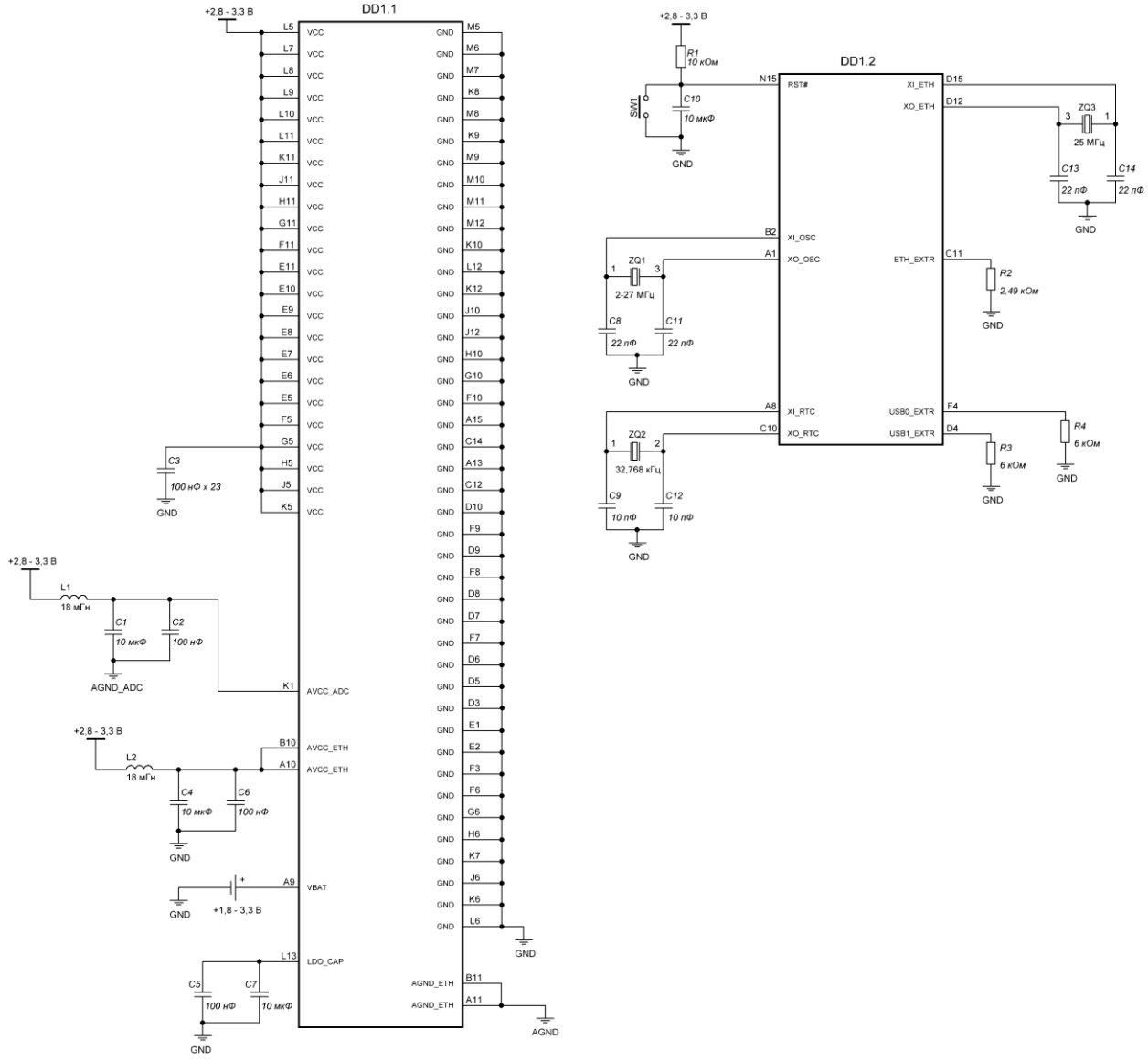


Рисунок Д.2 – Пример схемы включения микроконтроллера К1921ВГ10Я