MAX 10 FPGA Device Architecture

M10-ARCHITECTURE
2017.02.21

Subscribe
Send Feedback

Contents

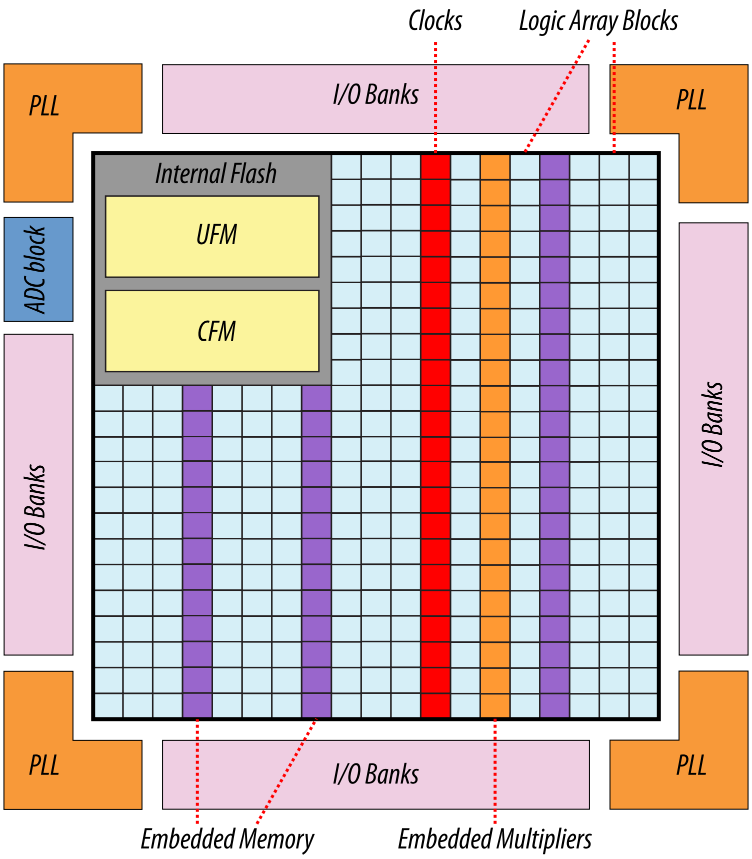
1 MAX® 10 FPGA Device Architecture

The MAX® 10 devices consist of the following:

- Logic array blocks (LABs)
- Analog-to-digital converter (ADC)
- User flash memory (UFM)
- Embedded multiplier blocks
- Embedded memory blocks (M9K)
- Clocks and phase-locked loops (PLL)
- General purpose I/O
- High-speed LVDS I/O
- External memory interfaces
- Configuration flash memory (CFM)

Figure 1. Typical Device Floorplan for MAX 10 Devices

- The amount and location of each block varies in each MAX 10 device.
- Certain MAX 10 devices may not contain a specific block.



Related Links

- MAX 10 Device Datasheet
  Provides more information about specification and performance for MAX 10 devices.
- MAX 10 FPGA Device Overview
  Provides more information about maximum resources in MAX 10 devices

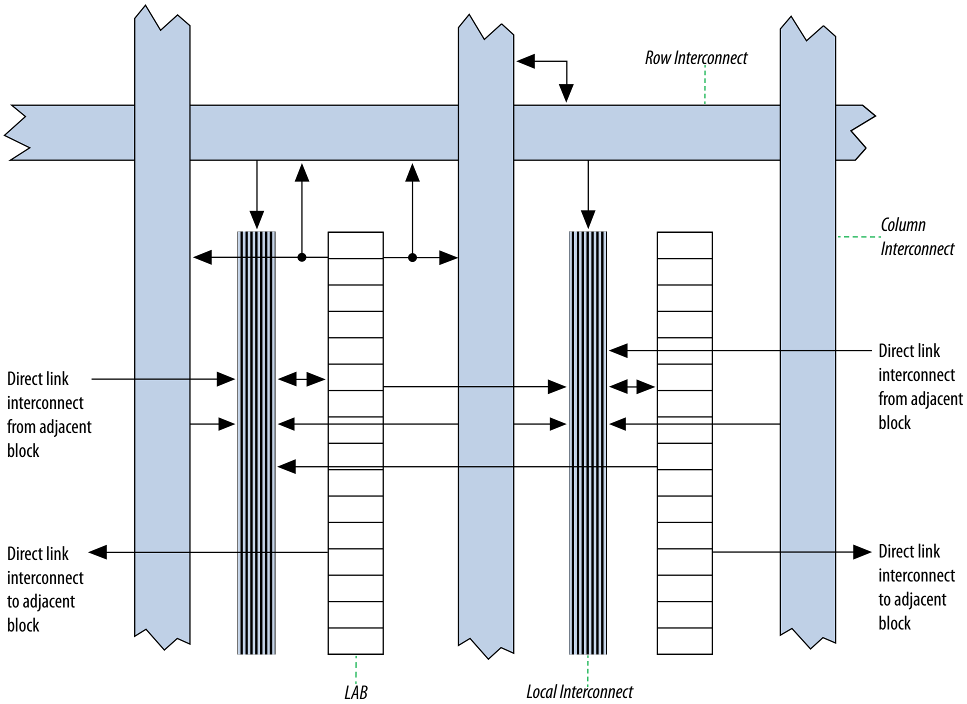
1.1 Logic Array Block

The LABs are configurable logic blocks that consist of a group of logic resources.

Each LAB consists of the following:

- 16 logic elements (LEs)—smallest logic unit in MAX 10 devices
- LE carry chains—carry chains propagated serially through each LE within an LAB
- LAB control signals—dedicated logic for driving control signals to LEs within an LAB
- Local interconnect—transfers signals between LEs in the same LAB
- Register chains—transfers the output of one LE register to the adjacent LE register in an LAB

Figure 2. LAB Structure of MAX 10 Devices


Row Interconnect
Column Interconnect
Direct link interconnect from adjacent block
Direct link interconnect from adjacent block
Direct link interconnect to adjacent block
Direct link interconnect to adjacent block
LAB
Local Interconnect

The Intel® Quartus® Prime Compiler places associated logic in an LAB or adjacent LABs, allowing the use of local and register chain connections for performance and area efficiency.

1.1.1 LAB Interconnects

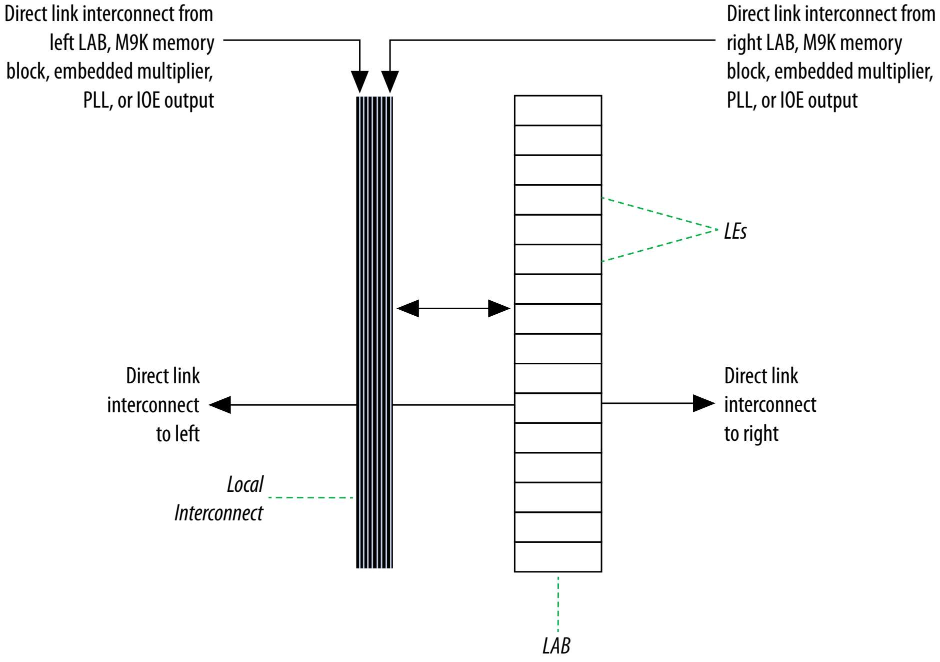
The LAB local interconnect is driven by column and row interconnects and LE outputs in the same LAB.

The direct link connection minimizes the use of row and column interconnects to provide higher performance and flexibility. The direct link connection enables the neighboring elements from left and right to drive the local interconnect of an LAB. The elements are:

- LABs
- PLLs
- M9K embedded memory blocks
- Embedded multipliers

Each LE can drive up to 48 LEs through local and direct link interconnects.

Figure 3. LAB Local and Direct Link Interconnects for MAX 10 Devices


Direct link interconnect from left LAB, M9K memory block, embedded multiplier, PLL, or IOE output
Direct link interconnect from right LAB, M9K memory block, embedded multiplier, PLL, or IOE output
LEs
Direct link interconnect to left
Direct link interconnect to right
Local Interconnect
LAB

1.1.2 LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its LEs.

The control signals include:

- Two clock signals
- Two clock enable signals
- Two asynchronous clear signals
- One synchronous clear signal
- One synchronous load signal

Figure 4. LAB-Wide Control Signals for MAX 10 Devices

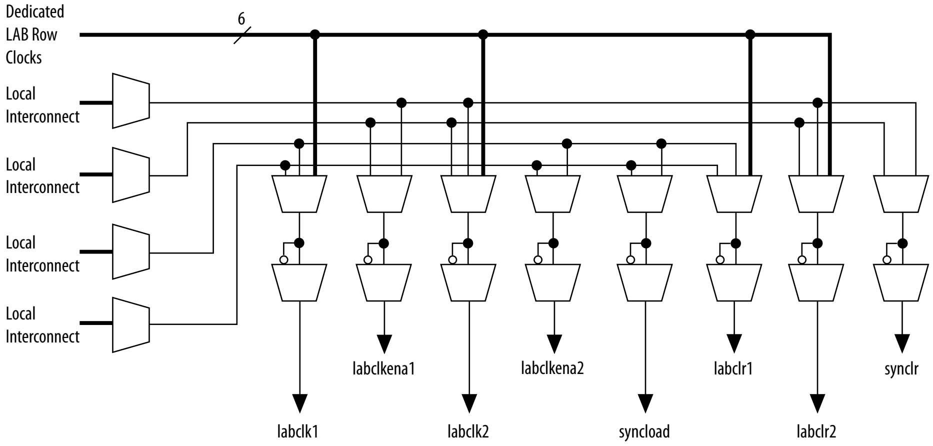


Table 1. Control Signal Descriptions for MAX 10 Devices

Control Signal	Description
`labclk1` `labclk2`	• Each LAB can use two clocks signals. The clock and clock enable signals of each LAB are linked. For example, any LE in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. • If the LAB uses both the rising and falling edges of a clock, it also uses both LAB-wide clock signals. • The LAB row clocks `[5..0]` and LAB local interconnect generate the LAB-wide clock signals. The MultiTrack interconnect inherent low skew allows clock and control signal distribution in addition to data distribution.
`labclkena1` `labclkena2`	• Each LAB can use two clock enable signals. The clock and clock enable signals of each LAB are linked. For example, any LE in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. • Deasserting the clock enable signal turns off the LAB-wide clock signal.
`labclr1` `labclr2`	Asynchronous clear signals: • LAB-wide control signals that control the logic for the clear signal of the register. • The LE directly supports an asynchronous clear function.
`syncload` `synclr`	Synchronous load and synchronous clear signals: • Can be used for implementing counters and other functions • LAB-wide control signals that affect all registers in the LAB

You can use up to eight control signals at a time. Register packing and synchronous load cannot be used simultaneously.

Each LAB can have up to four non-global control signals. You can use additional LAB control signals as long as they are global signals.

An LAB-wide asynchronous load signal to control the logic for the preset signal of the register is not available. The register preset is achieved with a NOT gate push-back technique. MAX 10 devices only support either a preset or asynchronous clear signal.

In addition to the clear port, MAX 10 devices provide a chip-wide reset pin (DEV_CLRn) to reset all registers in the device. An option set before compilation in the Intel Quartus Prime software controls this pin. This chip-wide reset overrides all other control signals.

1.1.3 Logic Elements

LE is the smallest unit of logic in the MAX 10 device family architecture. LEs are compact and provide advanced features with efficient logic usage.

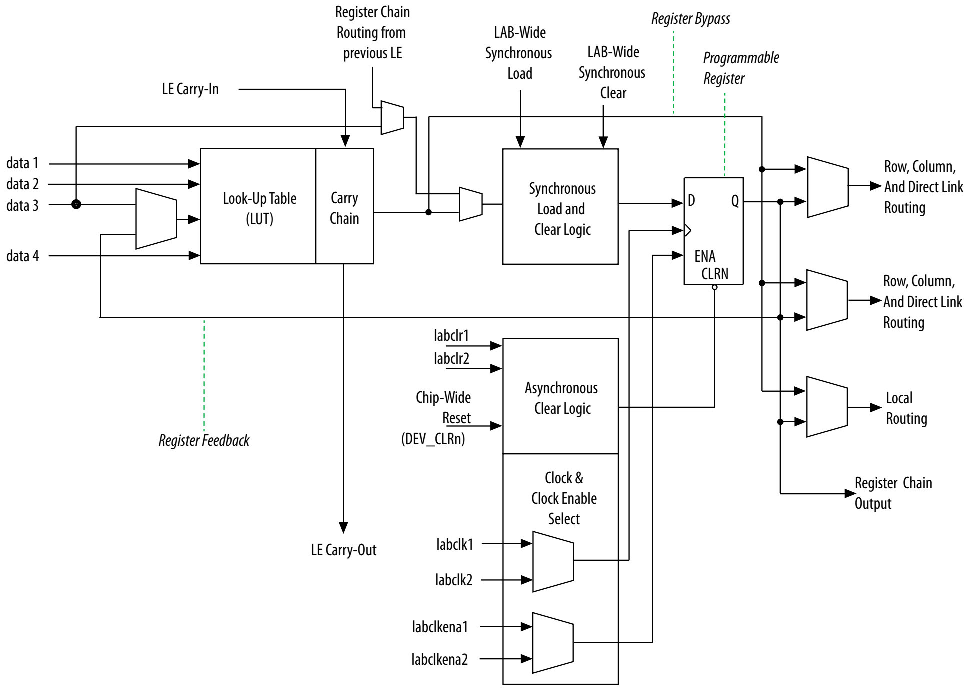
Each LE has the following features:

- A four-input look-up table (LUT) that can implement any function of four variables
- A programmable register
- A carry chain connection
- A register chain connection
- The ability to drive the following interconnects:
  - Local
  - Row
  - Column
  - Register chain
  - Direct link
- Register packing support
- Register feedback support

1.1.3.1 LE Features

LEs contain inputs, outputs, and registers to enable several features.

Figure 5. LE High-Level Block Diagram for MAX 10 Devices.


Register Chain Routing from previous LE
LAB-Wide Synchronous Load
LAB-Wide Synchronous Clear
Register Bypass
Programmable Register
LE Carry-In
data 1
data 2
data 3
data 4
Look-Up Table (LUT)
Carry Chain
Synchronous Load and Clear Logic
D
Q
ENA
CLRN
Row, Column, And Direct Link Routing
Row, Column, And Direct Link Routing
labclr1
labclr2
Chip-Wide Reset (DEV_CLRn)
Asynchronous Clear Logic
Local Routing
Register Feedback
Clock & Clock Enable Select
Register Chain Output
LE Carry-Out
labclk1
labclk2
labclkena1
labclkena2

LE Inputs

Each LE input is directed to different destinations to implement the desired logic function. In both the normal or arithmetic operating modes of the LE, there are six available inputs:

- Four data inputs from the LAB local interconnect
- One LE carry-in from the previous LE carry-chain
- One register chain connection

LE Outputs

Each LE has three general routing outputs:

- Two LE outputs drive the column or row and direct link routing connections
- One LE output drives the local interconnect resources

MAX 10 devices support register packing. With register packing, the LUT or register output drives the three outputs independently. This feature improves device utilization by using the register and the LUT for unrelated functions.

The LAB-wide synchronous load control signal is not available if you use register packing.

Register Chain Output

Each LE has a register chain output that allows registers in the same LAB to cascade together. This feature speeds up connections between LABs and optimizes local interconnect resources:

- LUTs are used for combinational functions
- Registers are used for an unrelated shift register implementation

Programmable Register

You can configure the programmable register of each LE for D, T, JK, or SR flipflop operation. Each register has the following inputs:

- Clock—driven by signals that use the global clock network, general-purpose I/O pins, or internal logic
- Clear—driven by signals that use the global clock network, general-purpose I/O pins, or internal logic
- Clock enable—driven by the general-purpose I/O pins or internal logic

For combinational functions, the LUT output bypasses the register and drives directly to the LE outputs.

Register Feedback

The register feedback mode allows the register output to feed back into the LUT of the same LE. Register feedback ensures that the register is packed with its own fan-out LUT, providing another mechanism for improving fitting. The LE can also drive out registered and unregistered versions of the LUT output.

1.1.3.2 LE Operating Modes

The LEs in MAX 10 devices operate in two modes.

- Normal mode
- Arithmetic mode

These operating modes use LE resources differently. Both LE modes have six available inputs and LAB-wide signals.

The Intel Quartus Prime software automatically chooses the appropriate mode for common functions, such as counters, adders, subtractors, and arithmetic functions, in conjunction with parameterized functions such as the library of parameterized modules (LPM) functions.

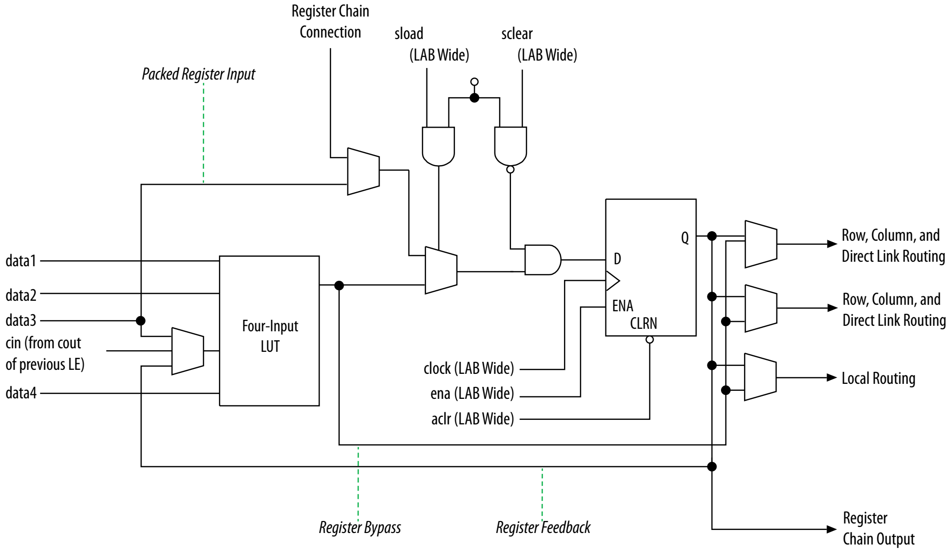
You can also create special-purpose functions that specify which LE operating mode to use for optimal performance.

1.1.3.2.1 Normal Mode

Normal mode is suitable for general logic applications and combinational functions.

In normal mode, four data inputs from the LAB local interconnect are inputs to a four-input LUT. The Intel Quartus Prime Compiler automatically selects the carry-in (cin) or the data3 signal as one of the inputs to the LUT. LEs in normal mode support packed registers and register feedback.

Figure 6. LE Operating in Normal Mode for MAX 10 devices

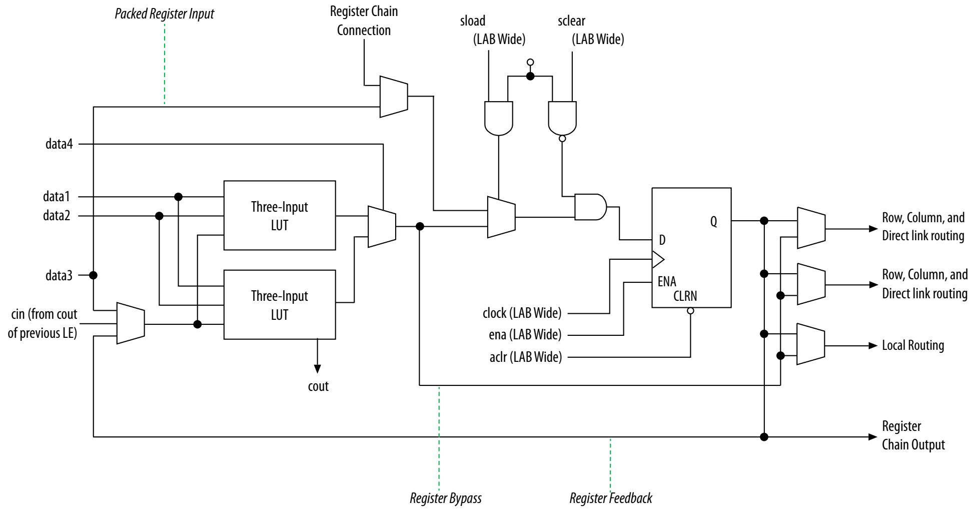


1.1.3.2.2 Arithmetic Mode

Arithmetic mode is ideal for implementing adders, counters, accumulators, and comparators.

The LE in arithmetic mode implements a two-bit full adder and basic carry chain. LEs in arithmetic mode can drive out registered and unregistered versions of the LUT output. Register feedback and register packing are supported when LEs are used in arithmetic mode.

Figure 7. LE Operating in Arithmetic Mode for MAX 10 devices



Carry Chain

The Intel Quartus Prime Compiler automatically creates carry chain logic during design processing. You can also manually create the carry chain logic during design entry. Parameterized functions, such as LPM functions, automatically take advantage of carry chains for the appropriate functions. The Intel Quartus Prime Compiler creates carry chains longer than 16 LEs by automatically linking LABs in the same column.

To enhanced fitting, a long carry chain runs vertically, which allows fast horizontal connections to M9K memory blocks or embedded multipliers through direct link interconnects. For example, if a design has a long carry chain in an LAB column next to a column of M9K memory blocks, any LE output can feed an adjacent M9K memory block through the direct link interconnect.

If the carry chains run horizontally, any LAB which is not next to the column of M9K memory blocks uses other row or column interconnects to drive a M9K memory block.

A carry chain continues as far as a full column.

1.2 Embedded Memory

The MAX 10 embedded memory block is optimized for applications such as high throughput packet processing, embedded processor program, and embedded data storage.

The MAX 10 embedded memory structure consists of 9,216-bit (including parity bits) blocks. You can use each M9K block in different widths and configuration to provide various memory functions such as RAM, ROM, shift registers, and FIFO.

MAX 10 embedded memory supports the following general features:

- 8,192 memory bits per block (9,216 bits per block including parity).
- Independent read-enable (`rden`) and write-enable (`wren`) signals for each port.
- Packed mode in which the M9K memory block is split into two 4.5 K single-port RAMs.
- Variable port configurations.
- Single-port and simple dual-port modes support for all port widths.
- True dual-port (one read and one write, two reads, or two writes) operation.
- Byte enables for data input masking during writes.
- Two clock-enable control signals for each port (port A and port B).
- Initialization file to preload memory content in RAM and ROM modes.

Related Links

MAX 10 Embedded Memory User Guide

1.3 Embedded Multiplier

You can use an embedded multiplier block in one of two operational modes, depending on the application needs:

- One 18-bit x 18-bit multiplier
- Up to two 9-bit x 9-bit independent multipliers

You can also use embedded multipliers of the MAX 10 devices to implement multiplier adder and multiplier accumulator functions. The multiplier portion of the function is implemented using embedded multipliers. The adder or accumulator function is implemented in logic elements (LEs).

Related Links

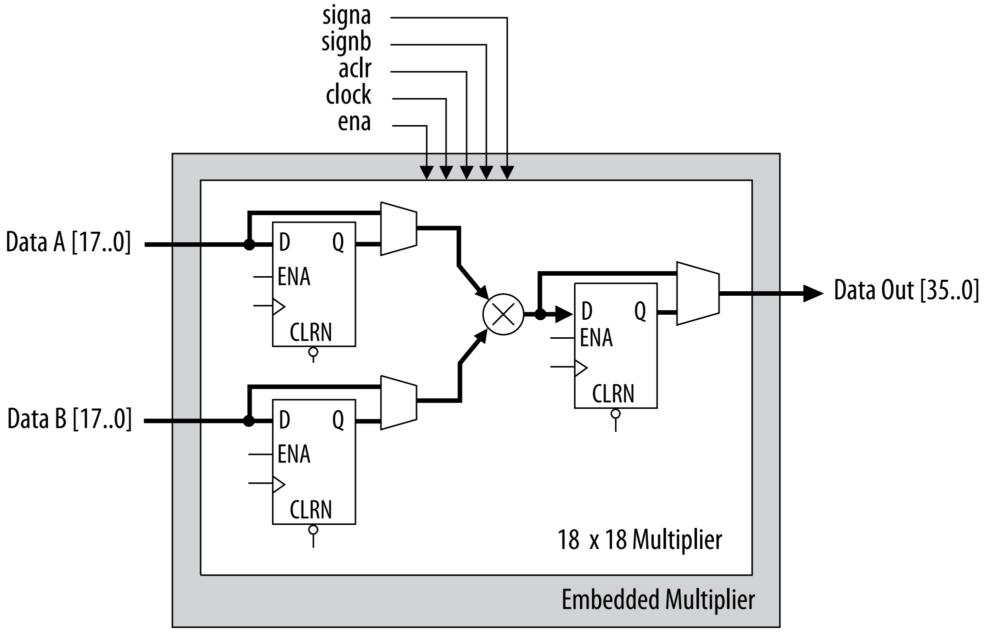
MAX 10 Embedded Multiplier User Guide

1.3.1 18-Bit Multipliers

You can configure each embedded multiplier to support a single 18 x 18 multiplier for input widths of 10 to 18 bits.

The following figure shows the embedded multiplier configured to support an 18-bit multiplier.

Figure 8. 18-Bit Multiplier Mode



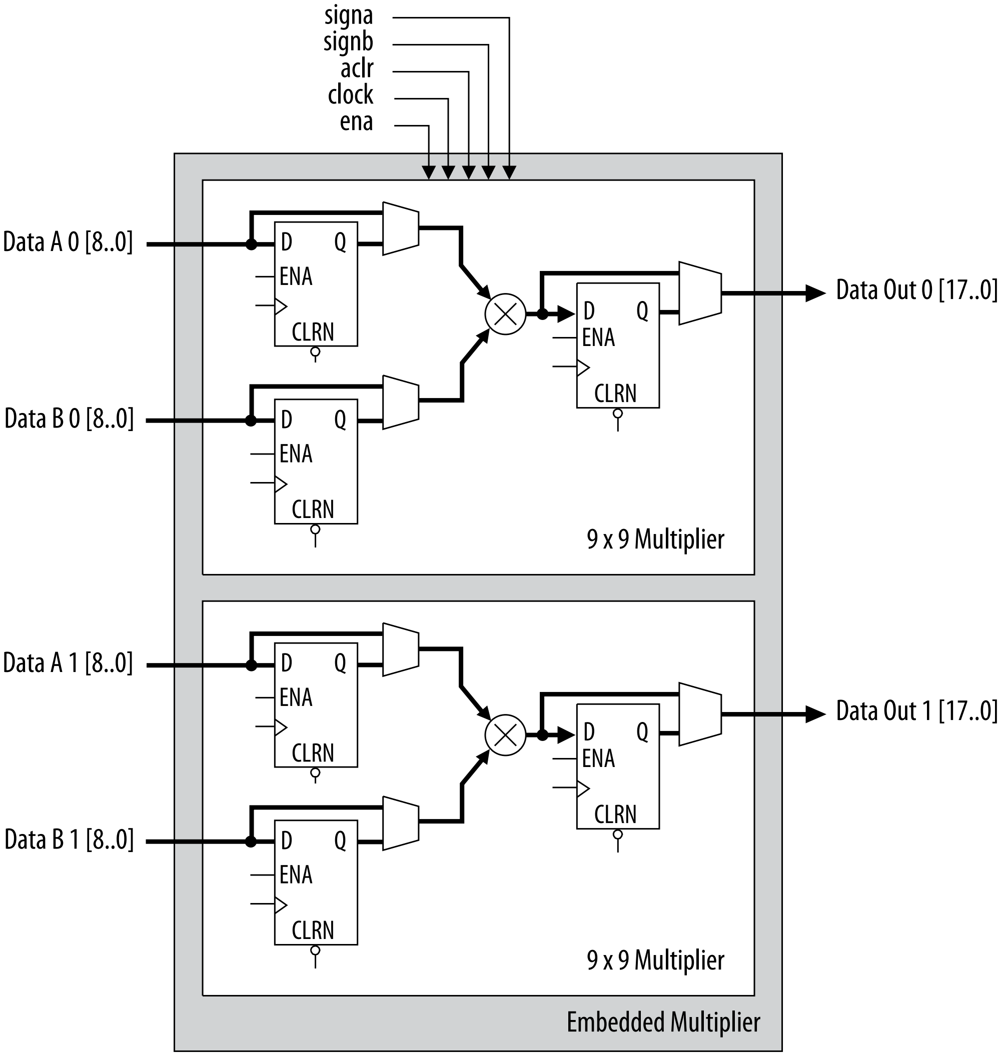
All 18-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Also, you can dynamically change the signa and signb signals and send these signals through dedicated input registers.

1.3.2 9-Bit Multipliers

You can configure each embedded multiplier to support two 9 × 9 independent multipliers for input widths of up to 9 bits.

The following figure shows the embedded multiplier configured to support two 9-bit multipliers.

Figure 9. 9-Bit Multiplier Mode



All 9-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both.

Each embedded multiplier block has only one `signa` and one `signb` signal to control the sign representation of the input data to the block. If the embedded multiplier block has two 9 × 9 multipliers the following applies:

- The `Data A` input of both multipliers share the same `signa` signal
- The `Data B` input of both multipliers share the same `signb` signal

1.4 Clocking and PLL

MAX 10 devices support global clock network (GCLK) and phase-locked loop (PLL).

Clock networks provide clock sources for the core. You can use clock networks in high fan out global signal network such as reset and clear.

PLLs provide robust clock management and synthesis for device clock management, external system clock management, and I/O interface clocking.

Related Links

MAX 10 Clock Networks and PLLs User Guide

1.4.1 Global Clock Networks

GCLKs drive throughout the entire device, feeding all device quadrants. All resources in the device, such as the I/O elements, logic array blocks (LABs), dedicated multiplier blocks, and M9K memory blocks can use GCLKs as clock sources. Use these clock network resources for control signals, such as clock enables and clears fed by an external pin. Internal logic can also drive GCLKs for internally-generated GCLKs and asynchronous clears, clock enables, or other control signals with high fan-out.

Figure 10. GCLK Network Sources for 10M02, 10M04, and 10M08 Devices

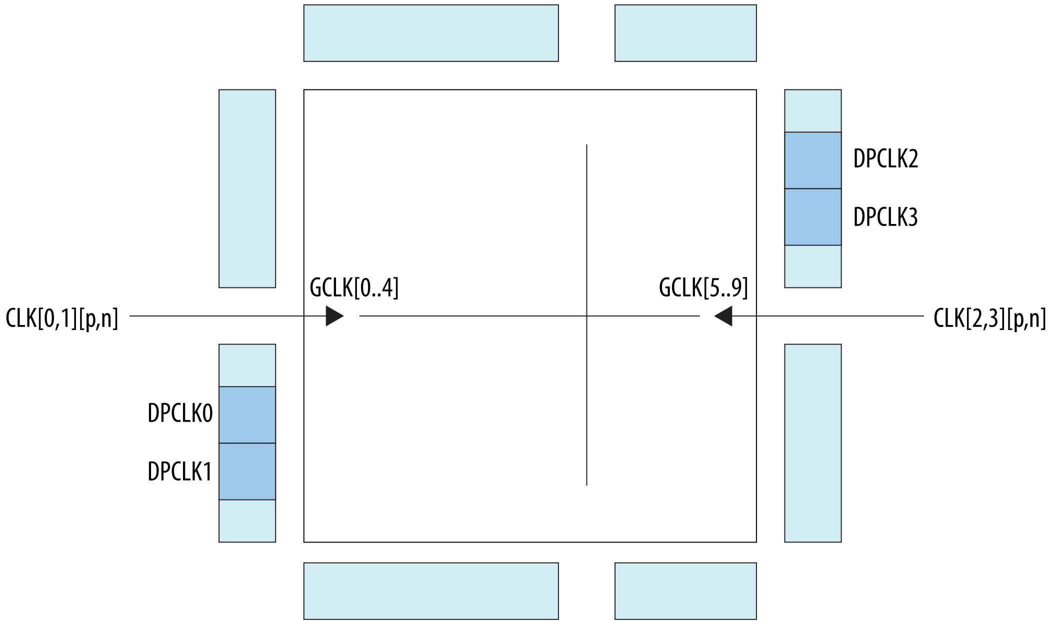
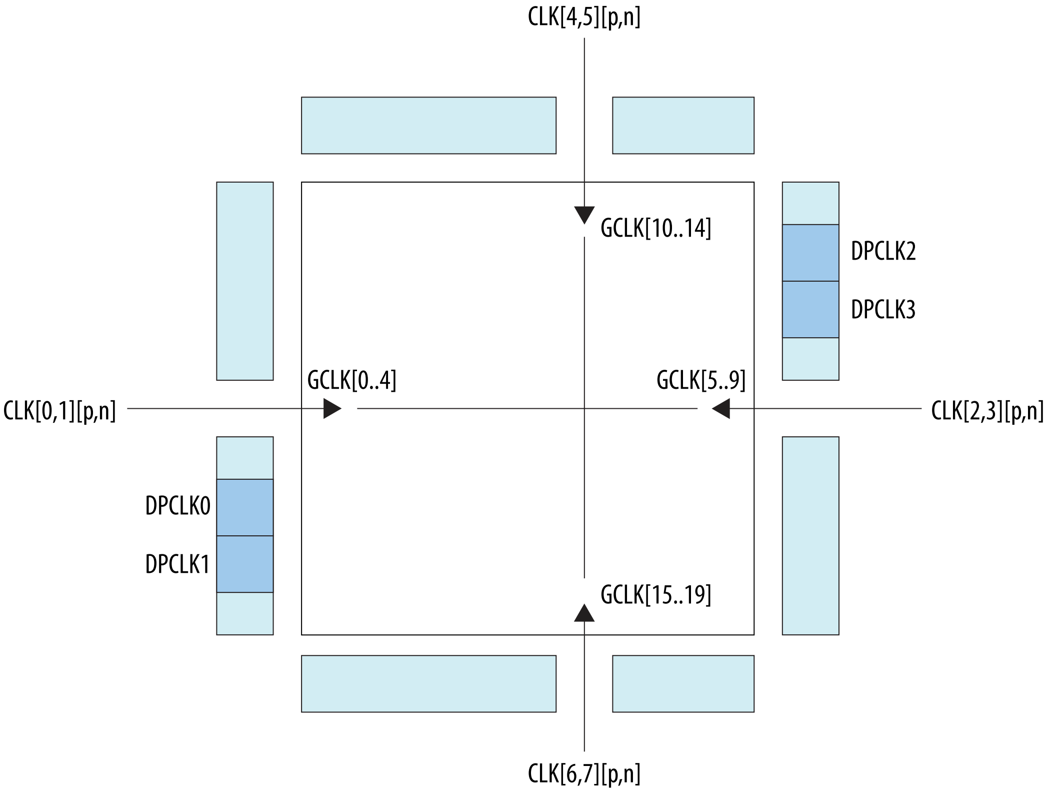


Figure 11. GCLK Network Sources for 10M16, 10M25, 10M40, and 10M50 Devices



1.4.2 Internal Oscillator

MAX 10 devices have built-in internal ring oscillator with clock multiplexers and dividers. The internal ring oscillator operates up to 232 MHz which is not accessible. This operating frequency further divides down to slower frequencies.

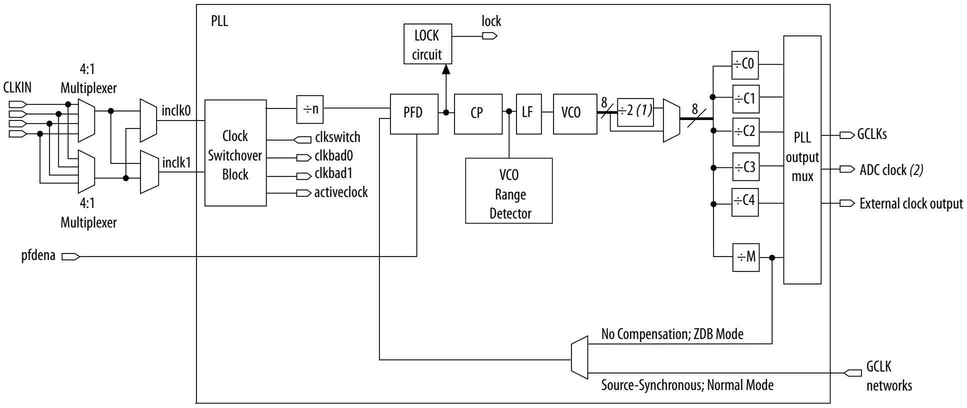
When the `oscena` input signal is asserted, the oscillator is enabled and the output can be routed to the logic array through the `clkout` output signal. When the `oscena` signal is set low, the `clkout` signal is constant high. You can analyze this delay using the TimeQuest timing analyzer.

1.4.3 PLL Block and Locations

The main purpose of a PLL is to synchronize the phase and frequency of the voltage-controlled oscillator (VCO) to an input reference clock.

Figure 12. MAX 10 PLL High-Level Block Diagram

Each clock source can come from any of the two or four clock pins located on the same side of the device as the PLL.



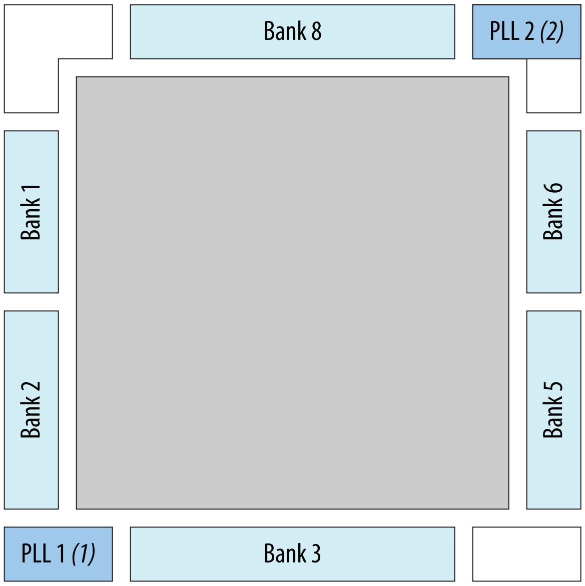
Notes:

(1) This is the VCO post-scale counter K.

(2) Only counter C0 of PLL1 and PLL3 can drive the ADC clock.

The following figures show the physical locations of the PLLs. Every index represents one PLL in the device. The physical locations of the PLLs correspond to the coordinates in the Intel Quartus Prime Chip Planner.

Figure 13. PLL Locations for 10M02 Device

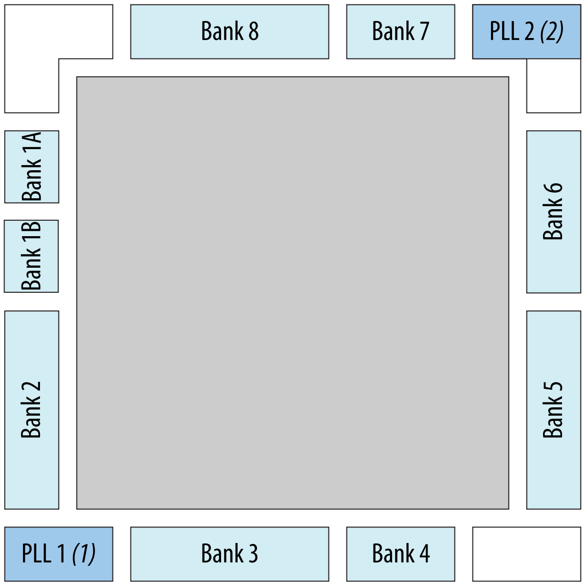


Notes:

(1) Available on all packages except V36 package.

(2) Available on U324 and V36packages only.

Figure 14. PLL Locations for 10M04 and 10M08 Devices

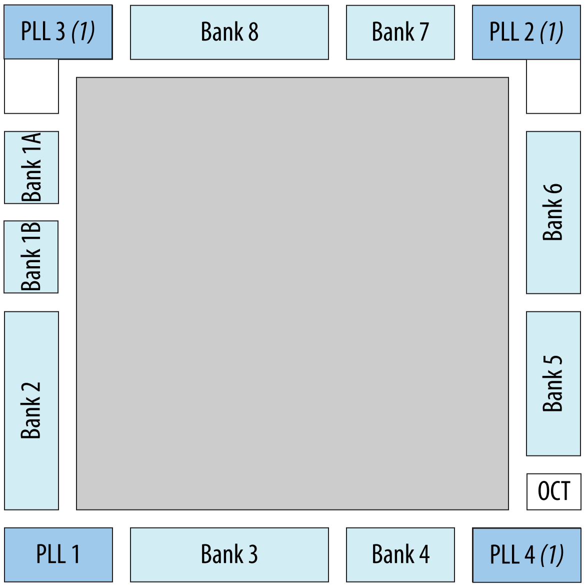


Notes:

(1) Available on all packages except V81 package.

(2) Available on F256, F484, U324, and V81 packages only.

Figure 15. PLL Locations for 10M16, 10M25, 10M40 and 10M50 Devices



Note:

(1) Available on all packages except E144 and U169 packages.

1.5 General Purpose I/O

The I/O system of MAX 10 devices support various I/O standards. In the MAX 10 devices, the I/O pins are located in I/O banks at the periphery of the devices. The I/O pins and I/O buffers have several programmable features.

Related Links

MAX 10 General Purpose I/O User Guide

1.5.1 MAX 10 I/O Banks Architecture

The I/O elements are located in a group of four modules per I/O bank:

- High speed DDR3 I/O banks—supports various I/O standards and protocols including DDR3. These I/O banks are available only on the right side of the device.
- High speed I/O banks—supports various I/O standards and protocols except DDR3. These I/O banks are available on the top, left, and bottom sides of the device.
- Low speed I/O banks—lower speeds I/O banks that are located at the top left side of the device.

For more information about I/O pins support, refer to the pinout files for your device.

1.5.2 MAX 10 I/O Banks Locations

The I/O banks are located at the periphery of the device.

For more details about the modular I/O banks available in each device package, refer to the relevant device pin-out file.

Figure 16. I/O Banks for MAX 10 02 Devices—Preliminary

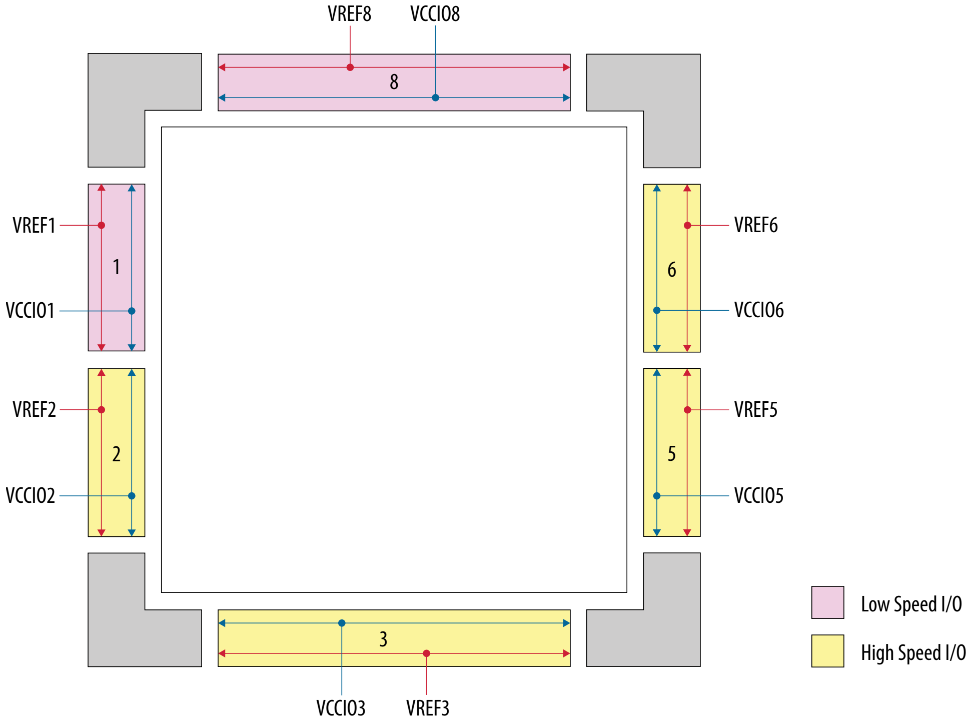


Figure 17. I/O Banks for MAX 10 04 and 08 Devices—Preliminary

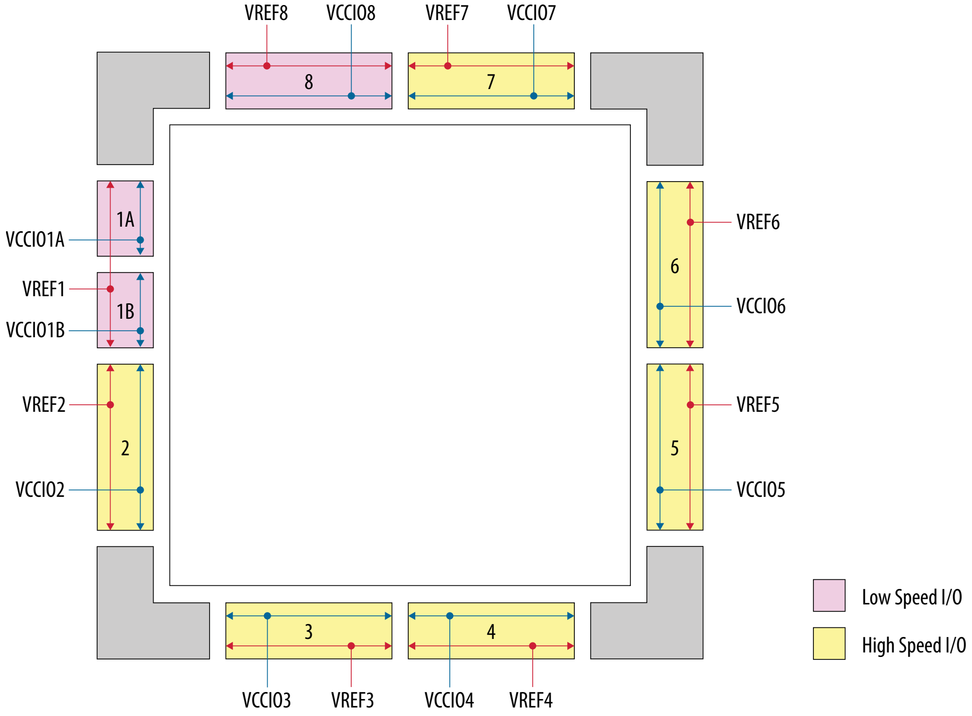
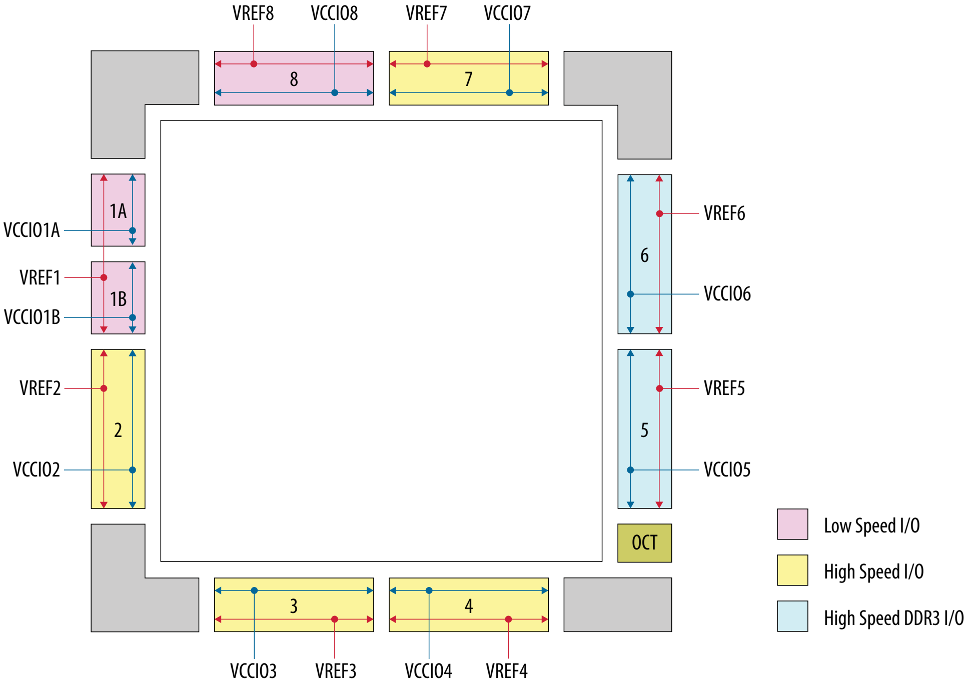


Figure 18. I/O Banks for MAX 10 16, 25, 40, and 50 Devices—Preliminary



1.6 High-Speed LVDS I/O

The MAX 10 device family supports high-speed LVDS protocols through the LVDS I/O banks and the Altera Soft LVDS IP core.

The MAX 10 devices use registers and logic in the core fabric to implement LVDS input and output interfaces.

- For LVDS transmitters and receivers, MAX 10 devices use the the double data rate I/O (DDIO) registers that reside in the I/O elements (IOE). This architecture improves performance with regards to the receiver input skew margin (RSKM) or transmitter channel-to-channel skew (TCCS).
- For the LVDS serializer/deserializer (SERDES), MAX 10 devices use logic elements (LE) registers.

Related Links

MAX 10 High-Speed LVDS I/O User Guide

1.6.1 MAX 10 High-Speed LVDS Circuitry

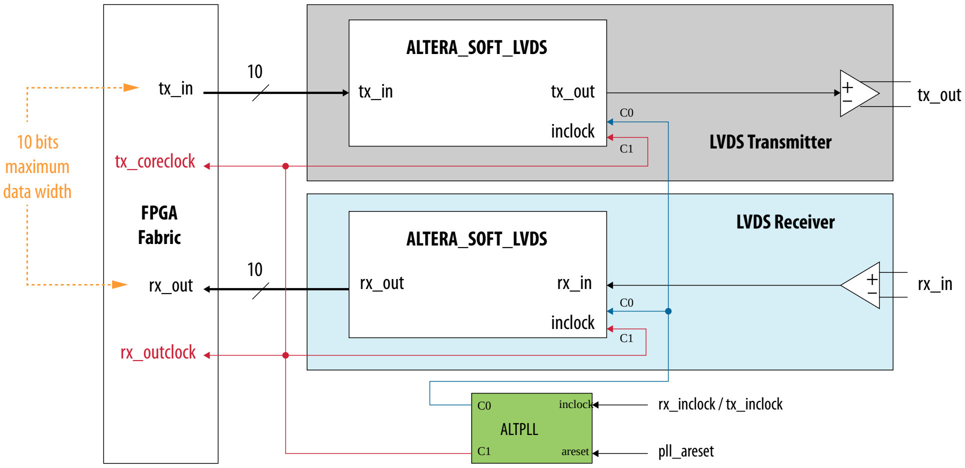
The LVDS solution uses the I/O elements and registers in the MAX 10 devices. The Altera Soft LVDS IP core implements the serializer and deserializer as soft SERDES blocks in the core logic.

The MAX 10 devices do not contain dedicated serialization or deserialization circuitry:

- You can use I/O pins and core fabric to implement a high-speed differential interface in the device.
- The MAX 10 solution uses shift registers, internal PLLs, and I/O elements to perform the serial-to-parallel and parallel-to-serial conversions of incoming and outgoing data.
- The Intel Quartus Prime software uses the parameter settings of the Altera Soft LVDS IP core to automatically construct the differential SERDES in the core fabric.

Figure 19. Soft LVDS SERDES

This figure shows a transmitter and receiver block diagram for the soft LVDS SERDES circuitry with the interface signals of the transmitter and receiver data paths.

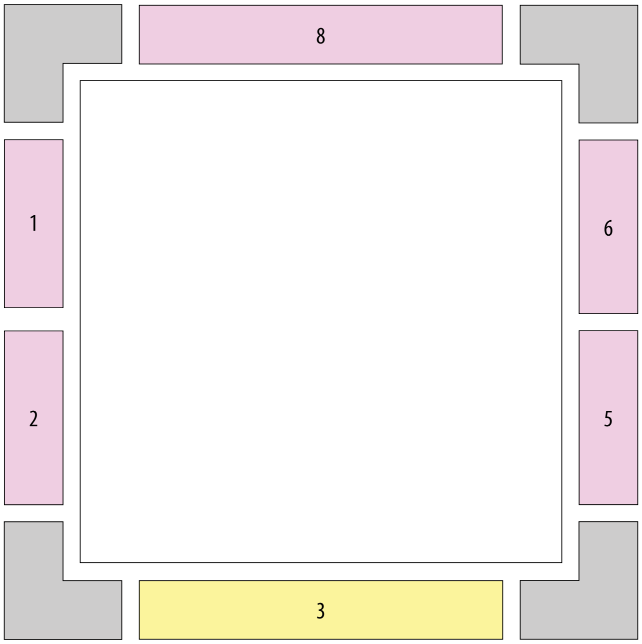


1.6.2 MAX 10 High-Speed LVDS I/O Location

The I/O banks in MAX 10 devices support true LVDS input and emulated LVDS output on all I/O banks. Only the bottom I/O banks support true LVDS output.

Figure 20. LVDS Support in I/O Banks of 10M02 Devices

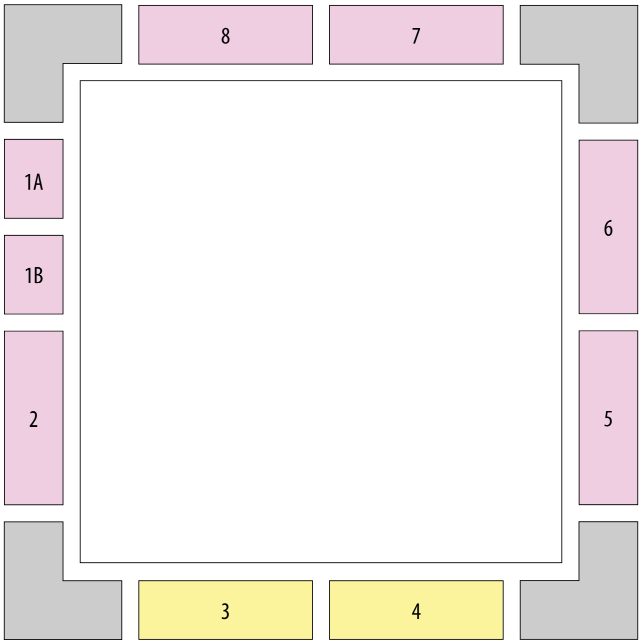
This figure shows a top view of the silicon die. Each bank is labeled with the actual bank number. LVPECL support only in banks 2 and 6.



	TX		RX	
LVDS		🟨	🟪	🟨
Emulated LVDS	🟪	🟨		
RSDS		🟨		
Emulated RSDS	🟪	🟨		
Mini-LVDS		🟨		
Emulated Mini-LVDS	🟪	🟨		
PPDS		🟨		
Emulated PPDS	🟪	🟨		
BLVDS	🟪	🟨	🟪	🟨
LVPECL			🟪	
TMDS			🟪	🟨
Sub-LVDS	🟪	🟨	🟪	🟨
SLVS	🟪	🟨	🟪	🟨
HiSpi			🟪	🟨

Figure 21. LVDS Support in I/O Banks of 10M04 and 10M08 Devices

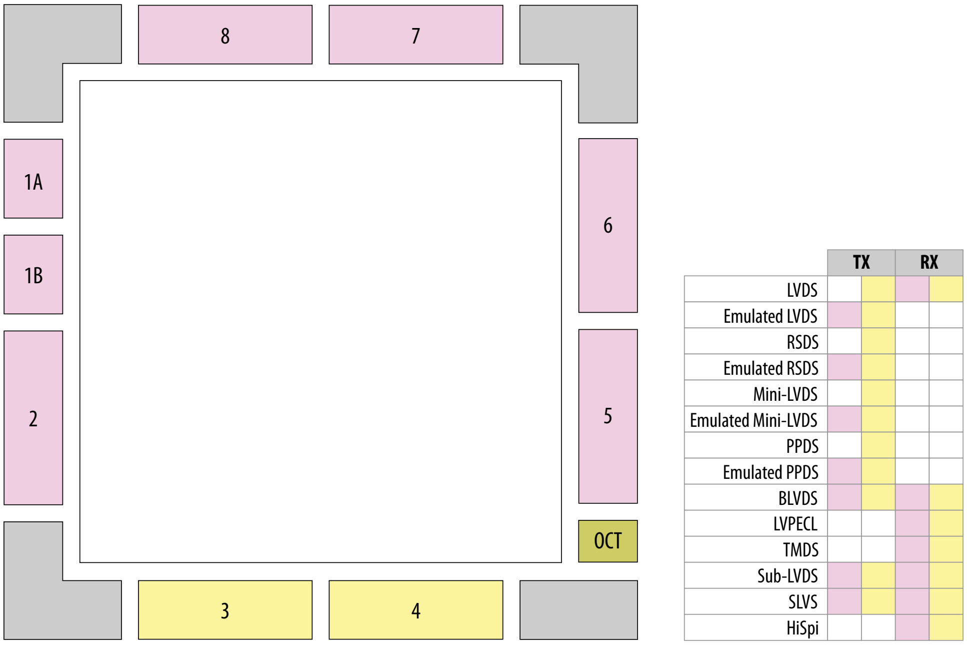
This figure shows a top view of the silicon die. Each bank is labeled with the actual bank number. LVPECL support only in banks 2 and 6.



	TX		RX	
LVDS		🟨	🟪	🟨
Emulated LVDS	🟪	🟨		
RSDS		🟨		
Emulated RSDS	🟪	🟨		
Mini-LVDS		🟨		
Emulated Mini-LVDS	🟪	🟨		
PPDS		🟨		
Emulated PPDS	🟪	🟨		
BLVDS	🟪	🟨	🟪	🟨
LVPECL			🟪	
TMDS			🟪	🟨
Sub-LVDS	🟪	🟨	🟪	🟨
SLVS	🟪	🟨	🟪	🟨
HiSpi			🟪	🟨

Figure 22. LVDS Support in I/O Banks of 10M16, 10M25, 10M40, and 10M50 Devices

This figure shows a top view of the silicon die. Each bank is labeled with the actual bank number. LVPECL support only in banks 2, 3, 6, and 8.



	TX		RX	
LVDS		🟨	🟪	🟨
Emulated LVDS	🟪	🟨		
RSDS		🟨		
Emulated RSDS	🟪	🟨		
Mini-LVDS		🟨		
Emulated Mini-LVDS	🟪	🟨		
PPDS		🟨		
Emulated PPDS	🟪	🟨		
BLVDS	🟪	🟨	🟪	🟨
LVPECL			🟪	🟨
TMDS			🟪	🟨
Sub-LVDS	🟪	🟨	🟪	🟨
SLVS	🟪	🟨	🟪	🟨
HiSpi			🟪	🟨

1.7 External Memory Interface

The MAX 10 devices are capable of interfacing with a broad range of external memory standards.

This capability allows you to use the MAX 10 devices in a wide range of applications such as image processing, storage, communications, and general embedded systems.

The external memory interface solution in MAX 10 devices consist of:

- The I/O elements that support external memory interfaces.
- The UniPHY IP core that allows you to configure the memory interfaces to support different external memory interface standards.

Related Links

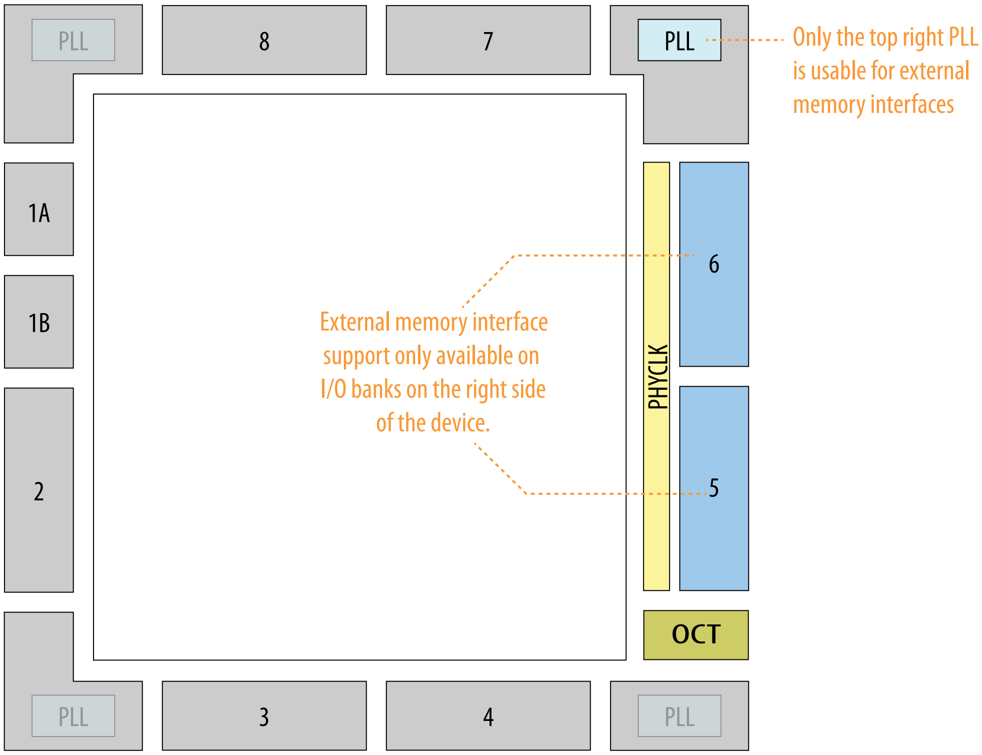
MAX 10 External Memory Interface User Guide

1.7.1 MAX 10 I/O Banks for External Memory Interface

In MAX 10 devices, external memory interfaces are supported only on the I/O banks on the right side of the device. You must place all external memory I/O pins on the I/O banks on the right side of the device.

Figure 23. I/O Banks for External Memory Interfaces

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.


PLL
8
7
PLL
Only the top right PLL is usable for external memory interfaces
1A
1B
6
External memory interface support only available on I/O banks on the right side of the device.
PHYCLK
2
5
OCT
PLL
3
4
PLL

External memory interfaces support is available only for 10M16, 10M25, 10M40, and 10M50 devices.

1.8 Analog-to-Digital Converter

MAX 10 devices feature up to two analog-to-digital converters (ADC). The ADCs provide the MAX 10 devices with built-in capability for on-die temperature monitoring and external analog signal conversion.

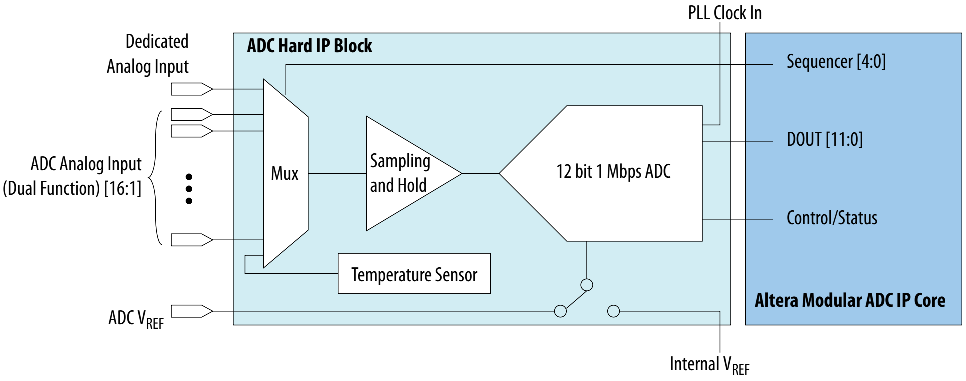
The ADC solution consists of hard IP blocks in the MAX 10 device periphery and soft logic through the Altera Modular ADC IP core.

The ADC solution provides you with built-in capability to translate analog quantities to digital data for information processing, computing, data transmission, and control systems. The basic function is to provide a 12 bit digital representation of the analog signal being observed.

The ADC solution works in two modes:

- Normal mode—monitors up to 18 single-ended external inputs with a cumulative sampling rate of one megasymbols per second (Msps).
- Temperature sensing mode—monitors internal temperature data input with a sampling rate of up to 50 kilosymbols per second (ksps).

Figure 24. ADC Hard IP Block in MAX 10 Devices



Related Links

MAX 10 Analog to Digital Converter User Guide

1.8.1 ADC Block Locations

The ADC blocks are located at the top left corner of the MAX 10 device periphery.

Figure 25. ADC Block Location in MAX 10 04 and 08 Devices

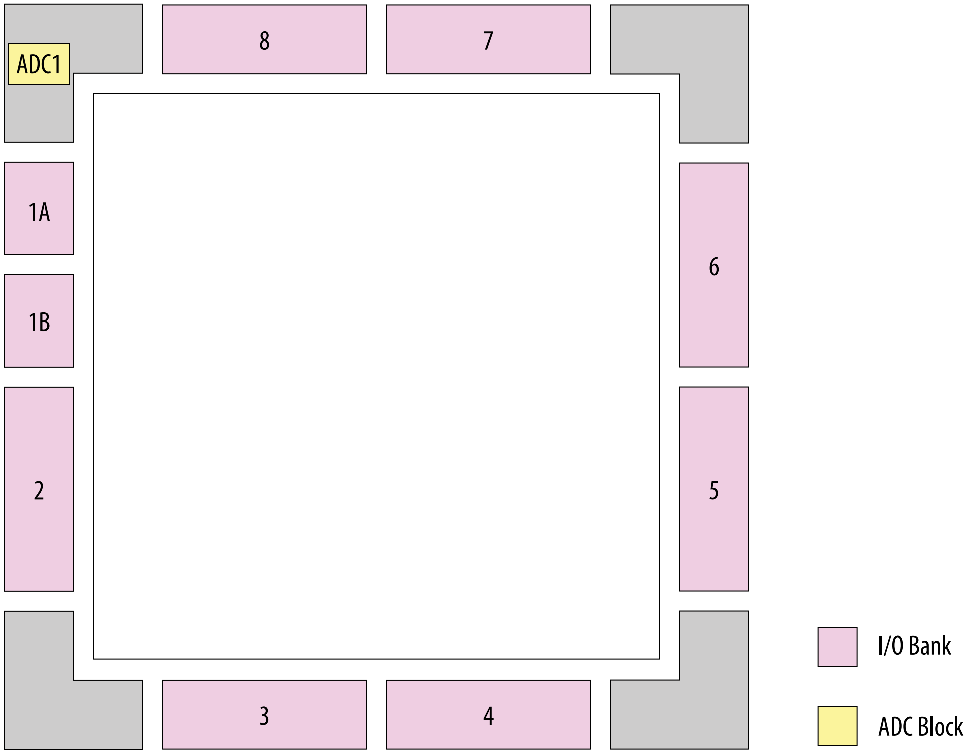


Figure 26. ADC Block Location in MAX 10 16 Devices

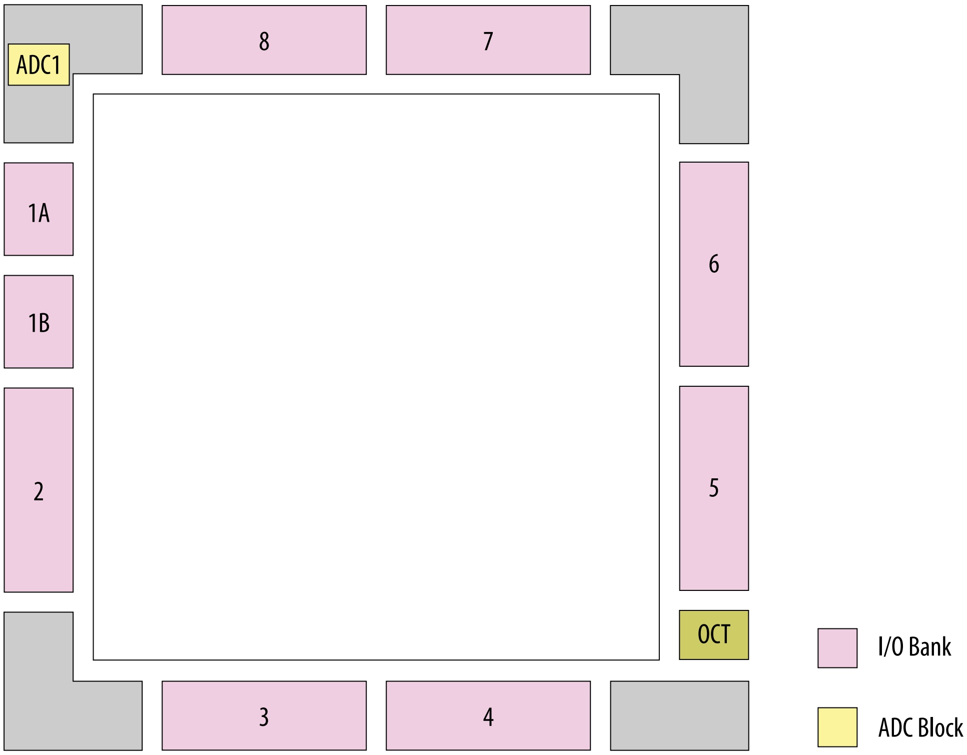
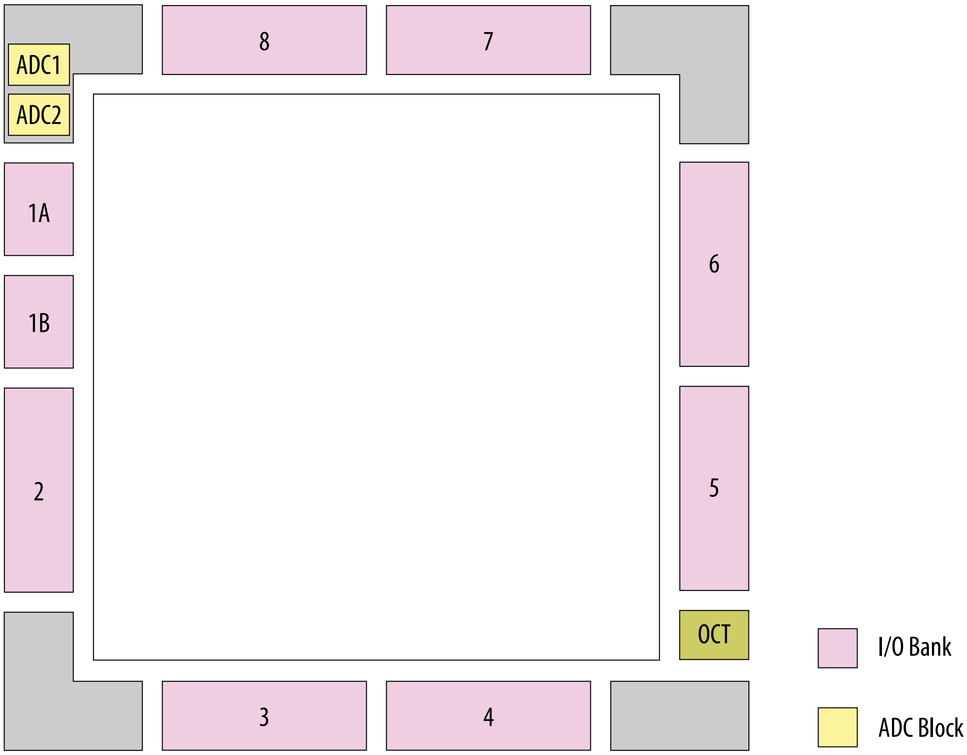


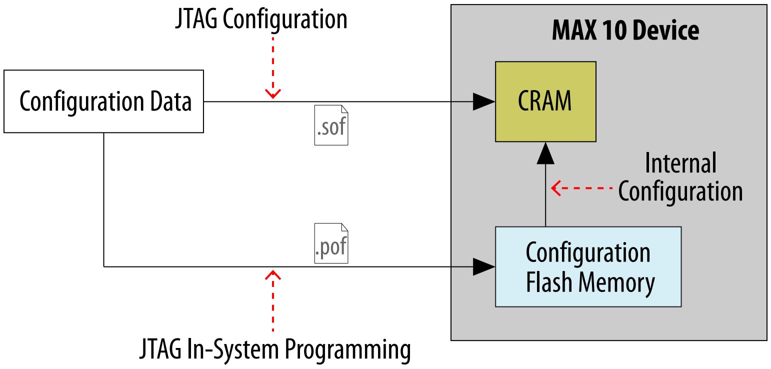
Figure 27. ADC Block Location in MAX 10 25, 40, and 50 Devices

Package E144 of these devices have only one ADC block.



1.9 Configuration Schemes

Figure 28. High-Level Overview of JTAG Configuration and Internal Configuration for MAX 10 Devices



Related Links

MAX 10 FPGA Configuration User Guide

1.9.1 JTAG Configuration

In MAX 10 devices, JTAG instructions take precedence over the internal configuration scheme.

Using the JTAG configuration scheme, you can directly configure the device CRAM through the JTAG interface—`TDI`, `TDO`, `TMS`, and `TCK` pins. The Intel Quartus Prime software automatically generates an SRAM Object File (**.sof**). You can program the **.sof** using a download cable with the Intel Quartus Prime software programmer.

1.9.2 Internal Configuration

You need to program the configuration data into the configuration flash memory (CFM) before internal configuration can take place. The configuration data to be written to CFM will be part of the programmer object file (`.pof`). Using JTAG In-System Programming (ISP), you can program the `.pof` into the internal flash.

During internal configuration, MAX 10 devices load the CRAM with configuration data from the CFM.

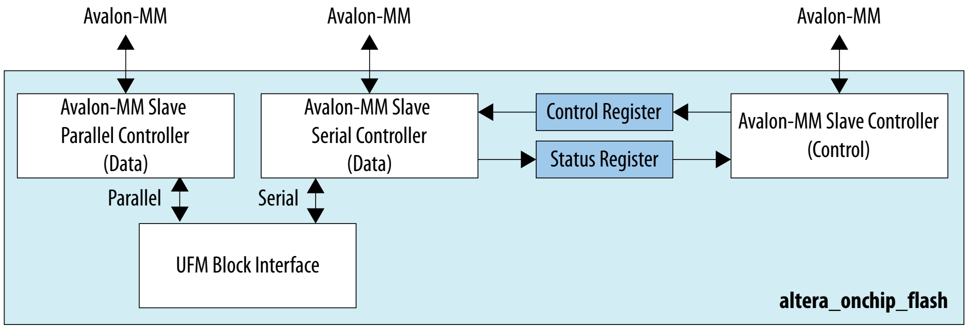
1.10 User Flash Memory

Intel MAX 10 devices feature a user flash memory (UFM) block that stores non-volatile information.

The UFM is part of the internal flash available in MAX 10 devices.

The UFM architecture of MAX 10 devices is a combination of soft and hard IPs. You can only access the UFM using the Altera On-Chip Flash IP core in the Intel Quartus Prime software.

Figure 29. Altera On-Chip Flash IP Block Diagram


Avalon-MM
Avalon-MM
Avalon-MM
Avalon-MM Slave Parallel Controller (Data)
Avalon-MM Slave Serial Controller (Data)
Control Register
Status Register
Avalon-MM Slave Controller (Control)
Parallel
Serial
UFM Block Interface
altera_onchip_flash

This IP block has two Avalon-MM slave controllers:

- Data—a wrapper of the UFM block that provides read and write accesses to the flash.
- Control—the CSR and status register for the flash, that is required only for write operations.

Related Links

MAX 10 User Flash Memory (UFM) User Guide

1.11 Power Management

MAX 10 power optimization features are as follows:

- Single-supply or dual-supply device options
- Power-on reset (POR) circuitry
- Power management controller scheme
- Hot socketing

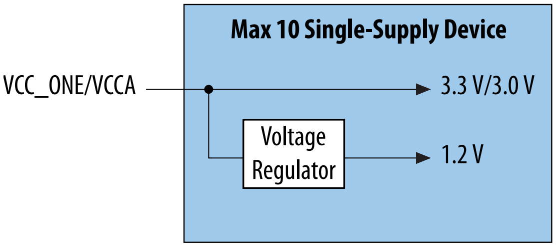
Related Links

Power Management User Guide

1.11.1 Single-Supply Device

MAX 10 single-supply devices only need either a 3.0- or 3.3-V external power supply. The external power supply serves as an input to the MAX 10 device VCC_ONE and VCCA power pins. This external power supply is then regulated by an internal voltage regulator in the MAX 10 single-supply device to 1.2 V. The 1.2-V voltage level is required by core logic operation.

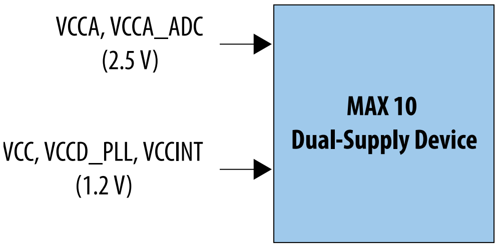
Figure 30. MAX 10 Single-Supply Device



1.11.2 Dual-Supply Device

MAX 10 dual-supply devices require 1.2 V and 2.5 V for the device core logics and periphery operations.

Figure 31. MAX 10 Dual-Supply Device



1.11.3 Power Management Controller Scheme

The power management controller scheme allows you to allocate some applications in sleep mode during runtime. This enables you to to turn off portions of the design, thus reducing dynamic power consumption. You can re-enable your application with a fast wake-up time of less than 1 ms.

1.11.4 Hot Socketing

The MAX 10 device offers hot socketing, which is also known as hot plug-in or hot swap, and power sequencing support without the use of any external devices. You can insert or remove the MAX 10 device on a board in a system during system operation. This does not affect the running system bus or the board that is inserted into the system.

The hot-socketing feature removes some encountered difficulties when using the MAX 10 device on a PCB that contains a mixture of devices with different voltage levels.

With the MAX 10 device hot-socketing feature, you no longer need to ensure a proper power-up sequence for each device on the board. MAX 10 device hot-socketing feature provides:

- Board or device insertion and removal without external components or board manipulation
- Support for any power-up sequence
- Non-intrusive I/O buffers to system buses during hot insertion

1.12 Document Revision History for MAX 10 FPGA Device Architecture

Date	Version	Changes
February 2017	2017.02.21	Rebranded as Intel.
August 2016	2016.08.11	Removed content duplication in *Embedded Multiplier*.
May 2016	2016.05.13	• Added internal oscillator architectural information. • Updated section name from *Clock Networks and PLL* to *Clocking and PLL*. • Added high-speed LVDS circuity information. • Added power management controller scheme and hot socketing information.
May 2015	2015.05.04	• Removed 'Internal Configuration' figure. • Added 'Overview of of JTAG Configuration and Internal Configuration for MAX 10 Devices' figure in 'Configuration'.
December 2014	2014.12.15	• Updated Altera On Chip Flash IP core block diagram for user flash memory. • Updated links.
September 2014	2014.09.22	Initial release.