

Продолжение. Начало в № 11 `2008

## Программа схемотехнического моделирования SwitcherCAD III

Михаил ПУШКАРЕВ

В этой части статьи рассмотрены синтаксис языка LTspice и директивы моделирования.

### Кратко о LTspice

LTspice — инструмент моделирования схем в SwitcherCAD III. Это программа моделирования схем с поддержкой рисования. LTspice-симулятор первоначально базировался на Berkeley SPICE 3F4/5. Симулятор претерпел полную переработку с целью улучшения работы, обнаружения ошибок и доведен до состояния промышленного стандарта в полупроводниковых и поведенческих моделях. Наряду с расширением возможностей аналогового SPICE-моделирования добавлена возможность цифрового моделирования, чтобы сделать LTspice симулятором аналогового и смешанного моделирования промышленного уровня для большого класса схем импульсных регуляторов и фильтров на переключаемых конденсаторах.

Многие продукты Linear Technology смоделированы с использованием составляющих собственность блоков и (или) составляющих собственность языка, которые объединяют реалистичное поведение с традиционными макромоделями. Это дает возможность быстро смоделировать систему питания и изготовить прототип.

LTspice может использоваться как SPICE-симулятор схем общего назначения. Новые схемы можно создать с использованием подключенной программы рисования. Команды моделирования и параметры помещаются на схему как текст с использованием синтаксиса SPICE. Щелчком мыши на цепи схемы в процессе моделирования или после моделирования на экран выводятся графики напряжений и токов.

LTspice — зарегистрированная торговая марка Linear Technology Corporation.

### Описание схемы

Схемы описываются текстом списка соединений (netlist). Список соединений состоит из списка элементов схемы и их узлов, определенных моделей и других SPICE-команд.

Обычно список соединений формируется при графическом вводе схемы. Чтобы начать новую схему, выберите в меню **File>Open**. Появится окно файлового браузера. Выберите любую имеющуюся схему и сохраните ее

под новым именем или создайте новый чистый файл схемы под новым именем. LTspice использует различные типы файлов и документов. Описание схемы сохраняется в файле с расширением .asc. Команды рисования схемы находятся в меню **Edit**. Клавиши быстрого вызова показаны в списке команд редактора схем.

Когда схема моделируется, информация для списка соединений извлекается из графической информации для создания файла с именем схемы, но с расширением .net. LTspice читает его как netlist.

Можно также открывать, моделировать и редактировать текстовый список соединений, созданный вручную или сгенерированный извне. Файлы с расширениями .net, .cir и .sp распознаются LTspice как список соединений.

### Общая структура и соглашения

Анализируемая схема описывается текстовым файлом, называемым списком соединений. Первая строка в списке соединений игнорируется, так как по предположению является комментарием. Последняя строка списка соединений обычно просто строка .END, но она может быть опущена. Любые строки после строки .END игнорируются.

Порядок строк между комментарием и концом не регламентирован. Строки могут быть комментариями, параметрами элементов схемы или директивами моделирования. Начнем с примера:

```
* Первая строка игнорируется
* Схема представляет собой резистивно-емкостной фильтр
* с входным сигналом прямоугольной формы 1 МГц
R1 n1 n2 1k; резистор 1 кОм между узлами n1 и n2
C1 n2 0 100p; конденсатор 100 пФ между узлами n2 и «землей»
V1 n1 0 PULSE(0 1 0 0.5u 1u); прямоугольный сигнал 1 МГц
.tran 3u; анализ переходного процесса продолжительностью 3 мкс
.end
```

Первые две строки — комментарии. Любая строка, начинающаяся с «\*», является комментарием и игнорируется. Строка, начинающаяся с «R1», объявляет, что есть резистор сопротивлением 1 кОм, включенный между узлами n1 и n2. Отметим, что при этом может использоваться точка с запятой «;», что-

бы начать комментарий в середине строки. Строка, начинающаяся с «C1», объявляет, что есть конденсатор емкостью 100 пФ между узлом n2 и «землей». Узел «0» — глобальная «земля» схемы.

Случайные знаки, пробелы и табуляции игнорируются.

Первый отличный от пробела знак в строке (табл. 6) определяет тип элемента схемы.

Числа могут быть выражены не только в научном написании, например, 1e12, но и с ис-

Таблица 6. Первый символ в строке

Символ, начинающий строку	Тип строки
*	Комментарий
A	Устройство со специальными функциями
B	Источник с произвольным поведением
C	Конденсатор
D	Диод
E	Источник напряжения, управляемый напряжением
F	Источник тока, управляемый током
G	Источник тока, управляемый напряжением
H	Источник напряжения, управляемый током
I	Независимый источник тока
J	Полевой транзистор с управляющим p-n-переходом (JFET)
K	Взаимная индуктивность
L	Индуктивность
M	МОП-транзистор (MOS)
O	Линия передачи с потерями
Q	Биполярный транзистор
R	Резистор
S	Ключ, управляемый напряжением
T	Линия передачи без потерь
U	Распределенная RC-цепь
V	Независимый источник напряжения
W	Ключ, управляемый током
X	Макромодель
Z	Полевой транзистор с затвором Шоттки (MESFET)
.	Директива моделирования, например: .options reltol=1e-4
+	Продолжение предыдущей строки. «+» не учитывается и продолжение строки считается частью предшествующей строки

Таблица 7. Масштабные коэффициенты

Суффикс	Масштабный коэффициент
T	1×10 <sup>12</sup>
G	1×10 <sup>9</sup>
Meg	1×10 <sup>6</sup>
K	1×10 <sup>3</sup>
Mil	25,4×10 <sup>-4</sup>
M	1×10 <sup>-3</sup>
u (или μ)	1×10 <sup>-4</sup>
n	1×10 <sup>-9</sup>
p	1×10 <sup>-12</sup>
f	1×10 <sup>-15</sup>

пользованием инженерных множителей. Таким образом, 1000 или 1e3 может также быть записано как 1 К. В таблице 7 перечислены масштабные коэффициенты.

Не специфицированные буквы, следующие после числа или инженерного множителя, игнорируются. Следовательно, 10, 10 V, 10 Volts и 10 Hz представляют одно и то же число, и М, МА, МSec и МMhos представляют один и тот же масштабный коэффициент (0,001). Распространенная ошибка состоит в изображении резистора со значением сопротивления 1М, подразумевая резистор в один мегаом, однако, 1М интерпретируется как резистор с сопротивлением один миллиом. Этим обеспечивается совместимость со стандартными SPICE-правилами.

LTspice воспринимает числа, написанные в форме 6K34, как 6,34 К. Это справедливо для любого из перечисленных множителей. Эта возможность может быть выключена следующим путем **Tools>Control Panel>SPICE** с отключением **Accept 3K4 as 3.4K**.

Имена узлов могут быть произвольным набором символов. Узел глобальной «земли» схемы — «0», хотя есть и «GND» — специальный синоним. Поскольку узлы обозначаются строкой символов, то «0» и «00» — разные цепи.

Во всех последующих разделах угловые скобки, помещенные вокруг полей данных, должны быть заполнены соответствующей информацией, например, <srcname> будет именем некоего определенного источника. Квадратные скобки указывают, что вложенные в поле данные являются дополнительными и необязательными.

## Синтаксис компонента

Компоненты, поддерживаемые в LTSpice, и их синтаксис приведены в таблице 8.

## Директивы моделирования. Команды с точкой

Для начала моделирования необходимо определить не только схему, но и вид анализа, который будет выполнен. Есть шесть различных типов анализа: линеаризованный анализ на переменном токе в режиме малого сигнала (**linearized small signal AC**), анализ на постоянном токе со ступенчатым изменением параметра (**DC sweep**), анализ шумов (**noise**), расчет рабочей точки на постоянном токе (**DC operating point**), нахождение передаточной функции на постоянном токе в режиме малого сигнала (**small signal DC transfer function**) и анализ переходного процесса (**transient analysis**). Один из этих типов анализа должен быть правильно описан.

Схема обычно изображается графически, а команды помещаются в схему как текст. Все такие команды начинаются с точки и поэтому названы «команды с точкой».

Таблица 8. Синтаксис компонентов

Устройство со специальными функциями	Axx <узел 1> <узел 2> <узел 3> <узел 4> <узел 5> <узел 6> <узел 7> <узел 8> <имя модели> [дополнительные параметры]
Источник с произвольным поведением	Vxx <+узел> <-узел> <V=... или I=...>
Конденсатор	Cxx <+узел> <-узел> <емкость> [[C=<начальное напряжение>] [Rser=<значение>] [Lser=<значение>] [Rpar=<значение>] [Cpar=<значение>] [m=<значение>] [дополнительные параметры]]
Диод	Dxx <узел анода> <узел катода> <имя модели> [коэффициент кратности area]
Источник напряжения, управляемый напряжением	Exx <+узел> <-узел> <+управляющий узел> <-управляющий узел> <коэффициент передачи>
Источник тока, управляемый током	Fxx <+узел> <-узел> <имя управляющего источника напряжения> <коэффициент передачи>
Источник тока, управляемый напряжением	Gxx <+узел> <-узел> <+управляющий узел> <-управляющий узел> <коэффициент передачи>
Источник напряжения, управляемый током	Hxx <+узел> <-узел> <имя управляющего источника напряжения> <коэффициент передачи>
Независимый источник тока	Ixx <+узел> <-узел> <значение тока>
Полевой транзистор с управляющим p-n-переходом (JFET)	Jxx <узел стока> <узел затвора> <узел истока> <имя модели> [коэффициент кратности area] [проводимость] [IC=<Vds,Vgs>] [temp=<температура>]
Взаимная индуктивность	Kxx L1 L2 L3... <коэффициент связи>
Индуктивность	Lxx <+узел> <-узел> <индуктивность> [[C=<начальный ток>] [Rser=<значение>] [Rpar=<значение>] [Cpar=<значение>] [m=<значение>] [дополнительные параметры]]
МОП-транзистор (MOSFET)	Mxx <узел стока> <узел затвора> <узел истока> <узел подложки> <имя модели> [L=<длина канала>] [W=<ширина канала>] [AD=<площадь стока>] [AS=<площадь истока>] [PD=<периметр стока>] [PS=<периметр истока>] [NRD=<значение>] [NRS=<значение>] [проводимость] [IC=<Vds,Vgs,Vbs>] [temp=<температура>]
Линия передачи с потерями	Oxx <+узел порта L> <-узел порта L> <+узел порта R> <-узел порта R> <имя модели>
Биполярный транзистор	Qxx <узел коллектора> <узел базы> <узел эмиттера> [узел подложки] <имя модели> [коэффициент кратности area] [проводимость] [IC=<Vbe,Vce>] [temp=<температура>]
Резистор	Rxx <узел1> <узел2> <сопротивление> [дополнительные параметры]
Ключ, управляемый напряжением	Sxx <узел1> <узел2> <+управляющий узел> <-управляющий узел> <имя модели> [сопротивление замкнутого ключа, сопротивление разомкнутого ключа]
Линия передачи без потерь	Txx <+узел порта L> <-узел порта L> <+узел порта R> <-узел порта R> <Z0=<значение> TD=<значение>
Однородная RC-линия с распределенными параметрами	Uxx <узел1> <узел2> <узел общий> <имя модели> L=<длина, м> [N=<количество сегментов>]
Независимый источник напряжения	Vxx <+узел> <-узел> <значение напряжения>
Ключ, управляемый током	Wxx <узел1> <узел2> <имя управляющей ветви> <модель> [сопротивление замкнутого ключа, сопротивление разомкнутого ключа]
Макромодель	Xxx n1 n2 n3... <имя макромодели>
Полевой транзистор с затвором Шоттки (MESFET)	Zxx <узел стока> <узел затвора> <узел истока> <имя модели> [коэффициент кратности area] [проводимость] [IC=<Vds,Vgs>]

## .AC — анализ на переменном токе в режиме малого сигнала, линеаризованный в окрестности рабочей точки по постоянному току (расчет частотных характеристик)

Для малых приращений сигнала (линейных) на переменном токе LTspice вычисляет комплексные значения узловых напряжений и токов ветвей в функции частоты. Сначала определяется рабочая точка схемы по постоянному току. Затем находятся линеаризованные для малого сигнала модели всех нелинейных компонентов схемы в этой рабочей точке. И, наконец, при использовании независимых источников напряжения и тока как управляющие сигналы результирующая линеаризованная схема обчисляется в заданном диапазоне частот.

Этот метод применяется при анализе фильтров и цепей, устойчивости и шумов.

Синтаксис:

```
.ac <oct, dec, lin> <количество точек> <начальная частота> <конечная частота>
```

Частота изменяется от <начальной частоты> до <конечной частоты>. Ключевые слова **oct**, **dec** устанавливают логарифмический характер изменения частоты октавами и декадами соответственно, **lin** — линейное изменение частоты. Ключевое слово <количество точек> определяет количество точек на октаву, декаду или весь диапазон частот.

Директива .AC вносится в лист схемы по команде **Edit>SPICE Analysis** с последующим вы-

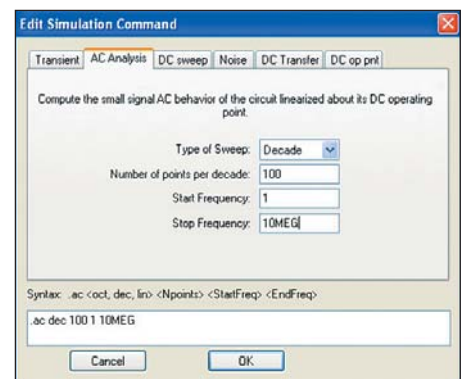


Рис. 31. Задание параметров .AC-анализа

бором в диалоговом окне пункта **AC Analysis** и заполнением чекбоксов соответственно заданию. Диалоговое окно показано на рис. 31.

Один из примеров анализа на переменном токе содержится в файле ...LTC\SwCADIII\examples\Educational\phono.asc.

## .BACKANNO — связь имен выводов макромодели с токами портов

Синтаксис:

```
.backanno
```

Эта директива автоматически включается в каждый список соединений, генерируемый SwitcherCAD III из схемы. Она заставляет LTspice включить в .raw файл информацию о связи токов портов с именами выводов. Это

позволяет отображать ток через вывод компонента щелчком мыши на выводе компонента.

### **.DC — анализ режима по постоянному току с вариацией параметров источников**

По этой директиве выполняется анализ по постоянному току со ступенчатым изменением параметра источника постоянного напряжения или тока. Одновременно может быть использовано до трех источников. Директива, в частности, применяется для вычисления передаточной функции усилителя по постоянному току или отображения характеристик для проверки модели транзистора.

Синтаксис:

```
.dc < имя 1-го источника> < начальное значение> <конечное значение> < приращение> [< имя 2-го источника> < начальное значение> <конечное значение> < приращение>]
```

<Имя источника> — имя независимого источника напряжения или тока, выходной сигнал которого задается степенями <приращение> от <начального значения> до <конечного значения>.

Это не документировано разработчиком, но возможно в качестве одного из источников указывать температуру. При вариации температуры в качестве имени источника указывается ключевое слово TEMP.

Параметры моделирования задаются с использованием диалогового окна, вызываемого по команде меню **Edit>SPICE Analysis** с выбором пункта **DC sweep**, показанного на рис. 32.

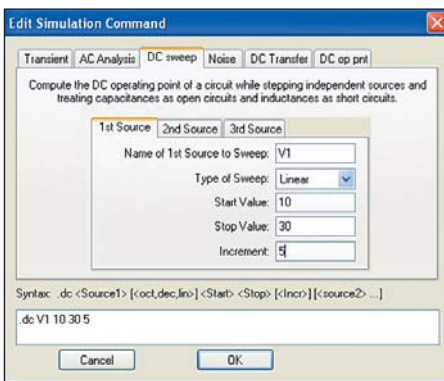


Рис. 32. Задание параметров .DC-анализа

Для каждого от одного до трех независимых источников необходимо указать его имя, характер изменения его выходного сигнала (линейный, логарифмический на октаву или декаду, задаваемый в виде списка значений), начальное и конечное значения, шаг изменения или количество точек.

В качестве выходного сигнала выступают разность узловых потенциалов или ток через ветвь с включенным в нее резистором.

В файле ...LTC\SwCADIII\examples\Educational\stepmodelparam.asc показан пример ис-

пользования .DC-директивы для демонстрации выходных характеристик биполярного транзистора.

### **.END — конец списка соединений**

Эта директива отмечает конец текстового списка соединений. Все строки после нее игнорируются. Не нужно помещать директиву как текст на схеме, поскольку экстрактор списка соединений делает это самостоятельно.

### **.ENDS — конец описания макромодели**

Синтаксис:

```
.ends [имя макромодели]
```

Эта директива отмечает конец описания макромодели. (См. .SUBCKT для дополнительной информации.) Необязательный параметр [имя макромодели] используется для удобства чтения и анализа текстового файла.

### **.FOUR — вычисление Фурье-компонентов после .TRAN-анализа**

Синтаксис:

```
.four <частота первой гармоники> [количество гармоник] [количество периодов] <выходные данные 1> [<выходные данные 2> ...]
```

Например:

```
.four 1kHz V(out)
```

Эта команда выполняется после анализа переходного процесса. Результат выполнения команды печатается в файл **.log**. Для просмотра результата используется команда меню **View>Spice Error Log**. В большинстве случаев для быстрого преобразования Фурье лучше воспользоваться возможностями окна графиков по команде меню **View>FFT**.

Задается частота первой гармоники и количество анализируемых гармоник. По умолчанию рассчитываются первые 9 гармоник.

Фурье-анализ выполняется на отрезке времени, отстоящем от конечного времени, Tend, более чем на период, по меньшей мере, на один период, если после количества гармоник указано количество периодов. Если количество периодов задано равным -1, Фурье-анализ выполняется по всему временному отрезку.

Пример использования директивы .FOUR приведен в файле ...LTC\SwCADIII\examples\Educational\audioamp.asc.

### **.FUNC — функции, определяемые пользователем**

Синтаксис:

```
.func <имя>([аргументы]) {выражение}
```

Например:

```
.func Pythag(x,y) {sqrt(x*x+y*y)}
```

Директива .FUNC позволяет создание пользовательских функций для использования в собственных параметризованных схемах и поведенческих источниках. Этим обеспечивается понятная ассоциативная связь имени с функцией и параметризованной макромоделью, чтобы абстрактные схемы можно было сохранить в библиотеках.

Директива .FUNC может быть включена в описание макромодели, чтобы ограничить использование функции этой макромоделью и макромоделями, входящими в нее.

Чтобы вводить параметр и оценивать выражение пользовательской функции, выражение заключается в фигурные скобки. Вложенное выражение будет заменено числом с плавающей точкой.

Пример использования директив .FUNC и .PARAM:

```
* Example deck using a .func statement
.func myfunc(x,y) {sqrt(x*x+y*y)}
.param u=100 v=600
V1 a 0 pulse(0 1 0 1n 1n .5u 1u)
R1 a b {myfunc(u,v/3)}
C1 b 0 100p
.tran 3u
.end
```

Подстановка параметров делается до начала моделирования.

### **.FERRET — загрузка файла данных с использованием URL**

Эта команда позволяет загружать файлы пакетным способом, указывая URL. Это удобно, когда не хочется указывать пути к каждому файлу через интернет-браузер. Загруженный файл будет в том же каталоге, что и исходная схема или список соединений. Эта команда не имеет никакого воздействия на моделирование:

```
* example deck
.ferret http://tspice.linear.com/software/scad3.pdf
.end
```

### **.GLOBAL — назначение глобальных узлов**

Синтаксис:

```
.global <узел1> [узел2 |узел3] [...]
```

Например:

```
.global VDD VCC
```

Команда .GLOBAL позволяет объявить, что определенные узлы, упомянутые в макромоделях, не локальные для макромодели узлы, а абсолютно глобальные узлы.

Общим глобальным для схемы является узел «O», для этого не требуется использовать директиву .GLOBAL. Узлы с именами в форме «\$G\_» также являются глобальными без упоминания в директиве .GLOBAL.

### .IC — задание начальных условий

Директива .IC задает начальные условия для анализа переходных процессов. Могут быть заданы узловые напряжения и токи через катушки индуктивности. Решение для постоянного тока выполняется с использованием начальных условий как ограничений. Если катушки индуктивности в решении для постоянного тока в других SPICE-программах обычно рассматриваются коротко замкнутыми, в LTspice, если начальный ток определен, они рассматриваются как источники тока бесконечного импеданса.

Синтаксис:

```
.ic [V(<узел>)=<напряжение>] [I(<катушка>)=<ток>]
```

Например:

```
.ic V(in)=2 V(out)=5 V(vc)=1.8 I(L1)=300m
```

### .INCLUDE — подключение текстовых файлов

Синтаксис:

```
.include <имя файла>
```

Эта директива подключает названный файл, как будто этот файл напечатан в списке соединений вместо команды .INCLUDE. Она используется для подключения библиотек моделей или макромоделей.

В имени файла должен быть указан полный путь к нему. Иначе LTspice смотрит сначала в каталоге <SwCADIII>\lib\sub, а затем в каталоге, который содержит запрашиваемый список соединений, где <SwCADIII> — каталог, содержащий исполняемый scad3.exe, типично устанавливается как C:\Program Files\LTCSwCADIII.

Недопустимо игнорировать расширение имени файла. Следует использовать **.inc myfile.lib**, а не **.inc myfile**, если файл называется myfile.lib.

Можно определить url как имя файла следующим образом:

```
.inc http://www.company.com/models/library.lib
```

Файл library.lib будет передан в каталог схемы и подключен. Во избежание повторной загрузки файла при следующем сеансе моделирования можно отредактировать директиву .INC.

```
.inc library.lib
```

Отметим, что если запрашиваемый url не существует, большинство серверов сети не показывают ошибку, но показывают html web-страницу в вашем web-браузере, которая объясняет ошибку. LTspice может не всегда воспринимать эти страницы как указание на ошибку, таким образом, можно получить некое загадочное сообщение об ошибке, когда идет процесс моделирования с включением html web-страницы как директивы SPICE.

Если полученный документ представляет собой .pdf-файл, моделирование прервется после загрузки. Например, после загрузки файла .pdf будет получен следующий комментарий:

```
* Dummy simulation to download the help file.
* The simulation will abort with an error, but
* you'll be left with the file scad3.pdf in the
* same directory containing the netlist.
.inc http://ltspice.linear.com/software/scad3.pdf
.end
```

Директива .INCLUDE использована в примере ...LTCSwCADIII\examples\Educational\logamp.asc для подключения макромоделей компонента opamp, в атрибутах которого нет указания на имя макромоделей или файл, ее содержащий.

### .LIB — подключение библиотеки

Синтаксис:

```
.lib <имя файла>
```

Эта директива подключает модель и описание макромоделей в названном файле, как будто этот файл напечатан в списке соединений вместо команды .lib. Если модель (модели) компонентов в подключаемом файле имеет альтернативные описания в библиотеке LTspice, то последние игнорируются.

В имени файла должен быть указан полный путь к нему. Иначе LTspice смотрит сначала в каталоге <SwCADIII>\lib\sub, а затем в каталоге, который содержит запрашиваемый список соединений, где <SwCADIII> — каталог, содержащий исполняемый scad3.exe, типично устанавливается как C:\Program Files\LTCSwCADIII.

Недопустимо игнорировать расширение имени файла. Необходимо указать **.lib myfile.lib**, а не **.lib myfile**, если файл называется myfile.lib.

Можно определить url как имя файла следующим образом:

```
.lib http://www.company.com/models/library.mod
```

Файл library.mod будет передан в каталог схемы и подключен. Во избежание повторной загрузки файла при следующем сеансе моделирования можно отредактировать указание .lib:

```
.lib library.mod
```

## Зашифрованные библиотеки

LTspice может генерировать и читать библиотеки, зашифрованные в специальной форме. Это позволяет одному пользователю готовить библиотеку для другого пользователя, которую можно использовать в моделировании, не показывая ее содержимое. Резонно попытаться сделать зашифрованную библиотеку, которую исключительно сложно расшифровывать, но такое решение нельзя считать абсолютным, если только не решить проблему непосредственно в программном обеспечении.

Чтобы подготовить зашифрованную библиотеку, нужно вызвать LTspice из командной строки с выбором опции «-encrypt». Сначала следует сделать копию библиотеки, поскольку она будет заменена зашифрованной версией. Нет никакой возможности преобразовать зашифрованную библиотеку обратно в текстовый формат. Уточним эти два шага:

1. Сделайте резервные копии библиотеки.
2. В командной строке напечатайте:

```
scad3.exe -encrypt <filename>
```

Файл <filename> будет заменен зашифрованной версией. Процесс шифрования займет несколько минут.

По окончании этого процесса вы будете иметь зашифрованный ASCII-файл. Можно добавить уведомление об авторском праве выше строки «\*Begin:», но первые 9 строк файла должны остаться неизменными, и каждая строка уведомления об авторском праве, добавленная вами, должна начинаться символом «\*».

Итак, приведем пример зашифрованного файла, созданный LTspice:

```
* LTspice Encrypted File
*
* This encrypted file has been supplied by a 3rd
* party vendor that does not wish to publicize
* the technology used to implement this library.
*
* Permission is granted to use this file for
* simulations but not to reverse engineer its
* contents.
*
* Begin:
50 3E 46 0F FA 6E 67 FF B8 4D D9 62 14 32 60 24
36 71 35 0B 66 4F AD 52 B8 F5 9E 22 9F C0 18 8B
FB FE 1D...
```

Можете изменить файл следующим образом:

```
* LTspice Encrypted File
*
* This encrypted file has been supplied by a 3rd
* party vendor that does not wish to publicize
* the technology used to implement this library.
*
* Permission is granted to use this file for
* simulations but not to reverse engineer its
* contents.
*
* Copyright © 2005 Acme SPICE Modeling
* For additional information, see
* www.acmespicemodels.com
*
* Begin:
50 3E 46 0F FA 6E 67 FF B8 4D D9 62 14 32 60 24
36 71 35 0B 66 4F AD 52 B8 F5 9E 22 9F C0 18 8B
FB FE 1D...
```

### **.LOADBIAS** — загрузка предварительно выполненного расчета по постоянному току

Синтаксис:

```
.loadbias <имя файла>
```

Команда **.LOADBIAS** — дополнение к команде **.SAVEBIAS**. Сначала выполните моделирование по команде **.SAVEBIAS**. Затем измените команду **.SAVEBIAS** на команду **.LOADBIAS**.

### **.MEASURE** — измерение заданных пользователем электрических величин

Есть два различных варианта директив **.MEASURE**. В первом случае обращаются к точке на оси абсцисс (независимая переменная, откладываемая по горизонтальной оси, ось времени для **.TRAN**-анализа), во втором случае обращаются к диапазону по оси абсцисс. Первый вариант используется, чтобы найти значение данных или выражения в заданной точке на оси абсцисс, или когда выполняется необходимое условие. Используется следующий синтаксис:

```
.meas[sure] [ACIDCIOPITRANITFNOISE] <имя результата>
[<FINDIDERIVIPARAM> <выражение>] [WHEN <выражение>]
[AT=<выражение>] [TD = <значение1>] [<RISE|FALL|CROSS> =
<отсчет1>] [LAST]
```

Здесь [ACIDCIOPITRANITFNOISE] — вид анализа, к которому применяется директива; FIND (найти); DERIV (производная); PARAM (использование результатов предшествующих измерений); WHEN (условие); AT (координата точки измерения на оси абсцисс); TD (задержка в единицах измерения по оси абсцисс относительно момента выполнения условия); RISE (увеличение); FALL (уменьшение); CROSS (пересечение); LAST (последний случай).

В первой позиции можно указать вид анализа, к которому применяется директива **.MEAS**. Это позволяет использовать соответствующие директивы **.MEAS** только в определенных видах анализа. Обязательно следует присваивать имя результату, который может быть использован в качестве параметра в других директивах **.MEAS**. Далее — примеры директивы **.MEAS** с обращением к одной точке на оси абсцисс:

```
.MEAS TRAN res1 FIND V(out) AT=5m
```

Результат — значение  $V(out)$  для  $t = 5$  мс, именованный как **res1**.

```
.MEAS TRAN res2 FIND V(out)*I(Vout) WHEN V(x)=3*V(y)
```

Результат — значение выражения  $V(out)*I(Vout)$ , когда условие  $V(x)=3*V(y)$  встречается в первый раз. Он будет именован как **res2**.

```
.MEAS TRAN res3 FIND V(out) WHEN V(x)=3*V(y) cross=3
```

Результат — значение  $V(out)$ , когда условие  $V(x)=3*V(y)$  встречается в третий раз. Он будет именован как **res3**.

```
.MEAS TRAN res4 FIND V(out) WHEN V(x)=3*V(y) rise=last
```

Результат — значение  $V(out)$ , для которого условие  $V(x)=3*V(y)$  встречается в последний раз, либо когда  $V(x)$  максимально близко к  $3*V(y)$ . Он будет именован как **res4**.

```
.MEAS TRAN res5 FIND V(out) WHEN V(x)=3*V(y) cross=3 TD=1m
```

Результат — значение  $V(out)$  через 1 мс после того, как условие  $V(x)=3*V(y)$  встречается в третий раз. Он будет именован как **res5**.

```
.MEAS TRAN res6 PARAM 3*res1/res2
```

Результат — значение  $3*res1/res2$ . Такая форма используется, чтобы определить значение выражения, составленного из результатов других **.MEAS**-утверждений. Не предполагается, что в оцениваемом выражении присутствуют непосредственные результаты моделирования, например,  $V(3)$ , если только это не данные из последней точки моделирования. Результат именован как **res6**.

В приведенных примерах обращения к одной точке на оси абсцисс результат соответствует данным ординаты (зависимые переменные). Если информация об ординате не требуется, результатом **.MEAS**-утверждения будет точка на оси абсцисс, в которой выполняется условие измерений:

```
.MEAS TRAN res6 WHEN V(x)=3*V(y)
```

Результат — это момент времени, когда условие  $V(x)=3*V(y)$  встречается впервые. Он именован как **res6**.

Второй вариант директивы **.MEAS** обращается к диапазону по абсциссе. Используется следующий синтаксис:

```
.meas [ACIDCIOPITRANITFNOISE] <имя результата>
[<AVG|MAX|MIN|PP|RMS|INTEG> <выражение>] [TRIG <lhs1>
[[VAL] = <rhs1>] [TD = <значение1>] [<RISE|FALL|CROSS> =
<отсчет1>] [TARG <lhs2> [[VAL] = <rhs2>] [TD = <значение2>]
[<RISE|FALL|CROSS> = <отсчет2>]
```

Диапазон по абсциссе ограничен точками, определенными как «TRIG» и «TARG». По умолчанию точка TRIG устанавливает начало моделирования и может быть опущена. Точно так же точка TARG по умолчанию соответствует данным в конце моделирования. Если все три из TRIG, TARG и WHEN точки опущены, то утверждение **.MEAS** действует на весь диапазон данных. Типы измерений,

**Таблица 9.** Вычислительные операции, выполняемые по диапазону данных

Ключевое слово	Выполняемая по диапазону операция
AVG	Вычисление среднего значения <выражение>
MAX	Нахождение максимального значения <выражение>
MIN	Нахождение минимального значения <выражение>
PP	Нахождение размаха от пика до пика <выражение>
RMS	Вычисление среднеквадратичного значения <выражение>
INTEG	Интегрирование <выражение>

которые могут быть выполнены в интервале, перечислены в таблице 9.

Если никакое измерение не задано, результат директивы **.MEAS** — расстояние по абсциссе между точками TRIG и TARG. Далее — пример директивы **.MEAS** по диапазону:

```
.MEAS TRAN res7 AVG V(NS01) TRIG V(NS05) VAL=1.5 TD=1.1u
FALL=1 TARG V(NS03) VAL=1.5 TD=1.1u FALL=1
```

Результат — среднее значение напряжения  $V(NS01)$  на отрезке времени от 1-го уменьшения  $V(NS05)$  до 1,5 В с задержкой 1,1 мкс и до 1-го уменьшения  $V(NS03)$  до 1,5 В с задержкой 1,1 мкс. Результат именован как **res7**.

Для **.AC**-анализа условные выражения в комплексных данных упрощаются до вещественной части с преобразованием в выражение для амплитуды. Например:

```
.MEAS AC res8 when V(out)=1/sqrt(2)
```

Результат **res8** — частота, на которой величина  $V(out)$  равна 0.7071067811865475.

Кроме того, результат директивы **.MEAS** может использоваться в другой директиве **.MEAS**. В следующем примере рассчитана полоса пропускания на уровне 3 дБ:

```
.MEAS AC tmp max mag(V(out)); найдем амплитуду и запомним как tmp
.MEAS AC BW trig mag(V(out))=tmp/sqrt(2) rise=1 targ
mag(V(out))=tmp/sqrt(2) fall=last
```

Результат — разность частот между двумя точками, лежащими на 3 дБ ниже максимального значения сигнала. Отметим, что данные, получаемые в **.AC**-анализе, комплексные, и таковы же результаты по директиве **.MEASURE**. Тем не менее, равенство оперирует только с реальной частью комплексного числа, то есть выражение  $\text{mag}(V(out))=\text{tmp}/\sqrt{2}$  эквивалентно  $\text{Re}(\text{mag}(V(out)))=\text{Re}(\text{tmp}/\sqrt{2})$ .

Операции AVG, RMS и INTEG в **.NOISE**-анализе несколько иные, чем в анализах других типов, так как **.NOISE** связан с квадратом частоты. Таким образом, и AVG, и RMS дают среднеквадратичное напряжение шумов, а INTEG дает интегрированный полный шум. Следовательно, если вы добавляете SPICE-директивы:

```
.MEAS NOISE out_totn INTEG V(onoise)
.MEAS NOISE in_totn INTEG V(inoise)
```

к .NOISE-анализу, в .log-файле будут напечатаны значения полных интегрированных среднеквадратичных шумов, приведенных к выходу и входу.

Директива .MEAS выполняется после окончания моделирования. Это позволяет записывать сценарии директив .MEAS и выполнять их над набором данных. Для этого сделайте окно графиков активным и выполните команду меню **File>Execute .MEAS Script**. Другое последствие .MEAS-директив, выполняемых после моделирования, — то, что точность вычислений ограничена точностью данных графика после сжатия. Можно изменить параметры настройки сжатия для получения более точных результатов по .MEAS-директиве.

Не следует задавать условие в виде равенства, типа «when <cond1> = <cond2>». Это объясняется тем, что равенство чисел в форме с плавающей запятой никогда не должно требоваться из-за конечной точности, используемой при хранении чисел.

Пример использования директивы .MEASURE приведен в файле ...LTC\SwCADIII\examples\Educational\MeasureBW.asc.

### .MODEL — определение SPICE-модели

Директива определяет модель для диода, транзистора, ключа, линии передачи с потерями или однородной RC-линии.

Некоторые элементы схемы, например, транзисторы, имеют много параметров. Вместо того чтобы определять каждый параметр для каждого транзистора, транзисторы объединены именем модели и имеют общие параметры. Транзисторы одной модели могут иметь различные размеры, и электрическое поведение масштабируется в соответствии с размером.

Синтаксис:

```
.model <имя модели> <имя типа модели> [(<список параметров>)]
```

Например:

```
.model DIODE D (Is=1E-10)
```

Таблица 10. Имена типов моделей компонентов

Имя типа модели	Тип компонента
CSW	Ключ, управляемый током
D	Диод
LTRA	Линия передачи с потерями
NJF	Полевой транзистор с каналом <i>л</i> -типа
NMF	MESFET с каналом <i>л</i> -типа
NMOS	МОП-транзистор с каналом <i>л</i> -типа
NPN	Биполярный <i>р-р-л</i> -транзистор
PJF	Полевой транзистор с каналом <i>р</i> -типа
PMF	MESFET с каналом <i>р</i> -типа
PMOS	МОП-транзистор с каналом <i>р</i> -типа
PNP	Биполярный <i>р-р-р</i> -транзистор
SW	Ключ, управляемый напряжением
URC	Однородная RC-линия с распределенными параметрами
VDMOS	Мощный вертикальный МОП-транзистор по технологии двойной диффузии

Имя модели должно быть уникальным. Таким образом, два различных типа элементов схемы, типа диода и транзистора, не могут иметь одного имени модели. Список параметров зависит от ее типа. В таблице 10 приведен перечень типов моделей.

### .NET — расчет параметров цепей в .AC-анализе

Эта директива используется при .AC-анализе в режиме малого сигнала, чтобы вычислить входной и выходной адмиттанс, импеданс, *Y*-, *Z*-, *H*- и *S*-параметры четырехполюсника. Эту методику можно также использовать для вычисления входного адмиттанса и импеданса двухполюсника. Директива должна использоваться с директивой .AC, определяющей девиацию частоты при анализе цепи.

Синтаксис:

```
.net [V(out[,ref])|I(Rout)] <Vin|Iin> [Rin=<значение>] [Rout=<значение>]
```

Входной порт цепи определяется либо независимым источником напряжения, <Vin>, либо независимым источником тока, <Iin>. Соответственно, выходной порт определяется либо как узел, V(out), либо как резистор, I(Rout). К портам подключаются резисторы Rin и Rout. Если не указано определенно, подключаемые импедансы по умолчанию 1 Ом, кроме случаев источника напряжения с определенным Rser или выходного порта, определенного как резистор. В этих двух случаях подключаемые резисторы по умолчанию являются импедансами устройства. Импедансы подключаемых по директиве .NET устройств действительны только для вычислений по этой директиве, но не для узловых напряжений и токов обычного .AC-анализа. То есть директива .NET не навязывает подключаемые к цепи импедансы при расчете напряжений и токов в процессе .AC-анализа.

Файл примера — ...LTC\SwCADIII\examples\Educational\S-param. Рекомендуется использовать источник напряжения, V4, с установкой требуемого импеданса источника Rser, и резистор, Rout, подключенный к выходу, с простой формой записи директивы .net I(Rout) V4. При этом нет необходимости задавать значения Rin или Rout в директиве .NET, достаточно задать характеристики устройств, подключенных к входу и выходу.

Доступ к графикам зависимостей малосигнальных параметров по команде **Plot Settings>Visible Traces** или **View>Add Trace**.

### .NODESET — задание начального режима по постоянному току

Директива .NODESET устанавливает начальные значения узловых потенциалов в рабочей точке по постоянному току. Если схема имеет множество возможных состояний по постоянному току, например, триггер, процесс итерации решения по постоянному току может никогда не сойтись. Директиву

.NODESET можно использовать, чтобы установить схему в то или иное состояние. В основном, после того как выполнен проход решения с напряжением, определенным в директиве .NODESET, ограничение удаляется в последующих итерациях.

Синтаксис:

```
.nodeset V(узел1)=<напряжение> [V(узел2)=<напряжение [...]>]
```

### .NOISE — расчет спектральной плотности шумов

Синтаксис:

```
.noise V(<узел1>[,<узел2>]) <имя источника> <oct, dec, lin> <количество точек> <начальная частота> <конечная частота>
```

Это — анализ в частотной области, в котором вычисляются джонсоновский тепловой шум, дробовый и фликкер-шумы. Выходные данные — значения квадратного корня из спектральной плотности шумов.

V(<узел1>[,<узел2>]) — участок цепи, для которого вычисляется полный выходной шум. Это может быть выражено как V(n1, n2), чтобы представить напряжение между двумя узлами. <Имя источника> — имя независимого источника напряжения или тока, к входу которого пересчитывается шум. Это фиктивный не шумящий источник сигнала. Параметры <oct, dec, lin>, <количество точек>, <начальная частота> и <конечная частота> определяют интересующий частотный диапазон и разрешение таким же образом, как и при использовании в директиве .AC. Параметры анализа вводятся с помощью диалогового окна (рис. 33), вызываемого по команде меню **Edit>Spice Analysis**, с выбором пункта **Noise**.

График выходных данных V(onoise) — спектральная плотность шумового напряжения в узле (узлах), определенных как выход в вышеупомянутом синтаксисе. Если источник входного сигнала — источник напряжения, то график V(inoise) — приведенная к входу плотность напряжения шумов. Если источник входного сигнала — источник тока, то график inoise — шум, приведенный к току источника входного сигнала. Может быть

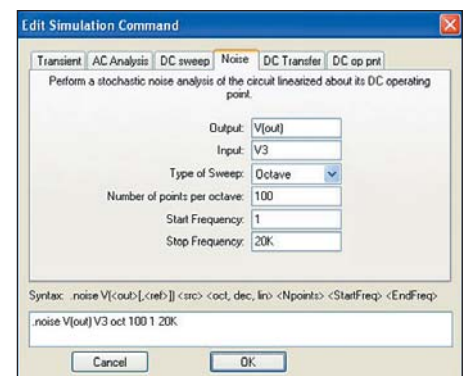


Рис. 33. Задание параметров .NOISE-анализа

Таблица 11. Параметры моделирования

Параметр	Тип данных	Значение по умолчанию	Наименование
ABSTOL	число	1 nA	Абсолютная погрешность расчета токов
BAURATE	число	нет	Используется при просмотре графиков. Задаёт режим сглаживания битовых переходов
CHGTOL	число	10 фК	Абсолютная погрешность расчета заряда
CSHUNT	число	0	Дополнительная емкость, добавляемая к каждому узлу относительно «земли»
CSHUNTINTERN	число	cshunt	Дополнительная емкость, добавляемая к каждому внутреннему узлу устройства относительно «земли»
DEFAD	число	0	Площадь стока МОП-транзистора по умолчанию
DEFAS	число	0	Площадь истока МОП-транзистора по умолчанию
DEFL	число	100 мкм	Длина канала МОП-транзистора по умолчанию
DEFW	число	100 мкм	Ширина канала МОП-транзистора по умолчанию
DELAY	число	0	Используется при просмотре графиков. Перемещает позиции битов в графике
FASTACCESS	флаг	false	Преобразование в формат файла fastaccess в конце моделирования
FLSGLOADS	флаг	false	Включение внешних источников тока в режим нагрузки
GMIN	число	1.00E-12	Проводимость, добавленная к каждому p-n-переходу для облегчения преобразований
GMINSTEPS	число	25	Устанавливается в нуль, чтобы предотвратить перебор параметра при начальном решении по постоянному току
GSHUNT	число	0	Дополнительная проводимость, добавляемая к каждому узлу относительно «земли»
ITL1	число	100	Максимальное количество итераций в расчете по постоянному току
ITL2	число	50	Максимальное количество итераций при расчете передаточной функции по постоянному току
ITL4	число	10	Максимальное количество итераций при переходе к следующему моменту времени при анализе переходного процесса
ITL6	число	25	Устанавливается в нуль, чтобы предотвратить ступенчатое изменение параметра источника при начальном решении по постоянному току
MAXCLOCKS	число	бескон.	Максимальное сохраняемое количество тактов
MAXSTEP	число	бескон.	Максимальный шаг при анализе переходного процесса
MEASCPLXfmt	слово	bode	Формат комплексных чисел для результатов директивы .MEASURE. Один из возможных: "polar", "cartesian" или "bode"
MEASDGT	число	6	Разрядность выходных данных для директивы .MEASURE.
METHOD	слово	trap	Выбор метода численного интегрирования
MINLOCKS	число	10	Минимальное сохраняемое количество тактов
MINDELTA GMIN	число	1.00E-04	Установка ограничения для завершения подбора gmin
NOMARCH	флаг	false	Отключение вывода графиков в процессе моделирования
NOOPITER	флаг	false	Переход к непосредственному управлению шагом gmin
NUMDGT	число	6	Исторически "numdgt" использовался для установки разрядности выходных данных. В LTspice, если "numdgt" больше 6, для зависимых переменных данных используется двойная точность
PIVREL	число	1.00E-03	Относительная величина элемента строки матрицы, необходимая для его выделения в качестве ведущего элемента
PIVTOL	число	1.00E-13	Абсолютная величина элемента строки матрицы, необходимая для его выделения в качестве ведущего элемента
PLOTABSTOL	число	1 nA	Установка абсолютной погрешности по току при сжатии графиков
PLOTRELTOL	число	0,0025	Установка относительной погрешности при сжатии графиков
PLOTVNTOL	число	10 мкВ	Установка абсолютной погрешности по напряжению при сжатии графиков
PLOTWINSIZE	число	300	Количество точек данных в одном окне после сжатия. Устанавливается в нуль, чтобы отменить сжатие
RELTOL	число	0,001	Относительная погрешность при расчете напряжений и токов
SRCSTEPMETHOD	число	0	Алгоритм ступенчатого изменения параметра источника
SRCSTEPS	число	25	Альтернативное имя для itl6
SSTOL	число	0,001	Относительная погрешность детектирования установленного режима
STARTLOCKS	число	5	Количество тактов ожидания перед поиском установленного режима
TEMP	число	27 °C	По умолчанию температура элементов схемы, для которых не задано определенное значение температуры
TNOM	число	27 °C	Номинальное значение температуры
TOPOLOGYCHECK	число	1	Устанавливается в нуль, чтобы отменить проверку плавающих узлов, контуров источников напряжения и топологии витков трансформаторов
TRTOL	число	1	Установка погрешности анализа переходного процесса. Этот параметр — коэффициент, определяющий фактическую ошибку округления
TRYTOCOMPACT	число	1	Представление в компактной форме информации о токах и напряжениях в длинных линиях с потерями
VNTOL	число	1 мкВ	Абсолютная погрешность при расчете напряжений

отображен шумовой вклад каждого компонента. Эти вклады пересчитываются к выходу. Можно пересчитать их к входу, поделив данные графика на усиление.

В окне графиков можно суммировать шумы в полосе пропускания, нажав на имя соответствующего графика правой кнопкой мыши при нажатой клавише Ctrl.

Пример использования директивы .NOISE приведен в файле ...LTC\SwCADIII\examples\Educational\noise.asc.

### .OP — определение режима по постоянному току в рабочей точке

Выполняется решение для постоянного тока с отключенными емкостями и коротко замкнутыми индуктивностями. Обычно

решение для постоянного тока выполняется как часть другого анализа, чтобы найти рабочую точку схемы. Используйте директиву .OP, если нужно только найти рабочую точку. Результаты появятся в диалоговом окне. Решение по постоянному току появится в управляющей строке, если указать на узел или цепь с током после .OP-моделирования.

В файле ...LTC\SwCADIII\examples\Educational\DcopPnt.asc приведен пример использования директивы .OP.

### .OPTIONS — установка параметров моделирования

Все доступные параметры моделирования приведены в таблице 11.

### .PARAM — параметры, определяемые пользователем

Директива .PARAM позволяет создавать определяемые пользователем переменные. Она обеспечивает понятную ассоциативную связь имени с параметризованной макромоделью, чтобы абстрактные схемы можно было сохранить в библиотеках.

Директива .PARAM может быть включена в описание макромодели, чтобы ограничить действие значения параметра этой макромоделью и макромоделями, включенными в эту макромодель.

Для подстановки параметра в оцениваемое выражение заключите выражение в фигурные скобки. Вложенное выражение будет заменено величиной с плавающей точкой.

Далее приведен пример использования как собственно директивы .PARAM, так и непосредственного ввода параметров в строке вызываемой макромодели:

```
*
* This is the circuit definition
.params x=y z=1k*tan(pi/4+.1)
X1 a b 0 divider top=x bot=z
V1 a 0 pulse(0 1 0 .5µ .5µ 0 1µ)
* this is the definition of the subcircuit
.subckt divider n1 n2 n3
r1 n1 n2 {top}
r2 n2 n3 {bot}
.ends
*
.tran 3µ
.end
```

Доступные математические функции и операции собраны в таблицах 12, 13.

В файле ...LTC\SwCADIII\examples\Educational\Cohn.asc приведен пример использования директивы .PARAM.

### .SAVE — ограничение количества сохраняемых данных

В процессе некоторых видов анализа, особенно при моделировании во временной области, может быть сгенерирован большой объем данных. Объем выходных данных может быть ограничен при использовании директивы .SAVE, чтобы сохранить только определенные интересующие узловые напряжения и токи в устройстве.

Синтаксис:

```
.save V(out) [V(in) [I(L1) [I(S2)]]] [dialogbox]
```

Директива .save I(Q2) сохраняет токи базы, коллектора и эмиттера биполярного транзистора Q2. Чтобы сохранить ток через один вывод транзистора, следует указать Ic(Q2).

Чтобы сохранить данные графиков, соответствующих образцу, можно использовать символы группировки «\*» и «?». Например, «.save V(\*) Id(\*)» позволит сохранить каждое значение напряжения и каждое значение тока.

Если присутствует ключевое слово dialogbox, то появляется диалоговое окно со списком всех доступных по умолчанию узловых напряжений и токов, позволяющее пользова-

Таблица 12. Математические функции

Функция	Определение
<b>ABS(x)</b>	Абсолютное значение x
<b>ACOS(x)</b>	Действительная часть $\arccos(x)$ , например, $\arccos(-5)$ возвращает 3.14159, а не 3.14159+2.29243i
<b>ARCOS(x)</b>	Синоним для $\arccos(x)$
<b>ACOSH(x)</b>	Действительная часть $\operatorname{acosh}(x)$ , например, $\operatorname{acosh}(5)$ возвращает 0, а не 1.0472i
<b>ASIN(x)</b>	Действительная часть арксинуса x, например, $\arcsin(-5)$ возвращает -1.57080, а не -1.57080+2.29243i
<b>ARCSIN(x)</b>	Синоним для $\arcsin(x)$
<b>ASINH(x)</b>	Гиперболический арксинус
<b>ATAN(x)</b>	Арктангенс x
<b>ARCTAN(x)</b>	Синоним для $\operatorname{atan}(x)$
<b>ATAN2(y, x)</b>	Арктангенс (x/y)
<b>ATANH(x)</b>	Гиперболический арктангенс
<b>BUF(x)</b>	$\operatorname{buf}(x)=1$ , если $x>0$ ; $\operatorname{buf}(x)=0$ , если $x\leq 0$ ,5
<b>CBRT(x)</b>	Кубический корень из x
<b>CEIL(x)</b>	Целое число, равное или большее x
<b>COS(x)</b>	Косинус x
<b>COSH(x)</b>	Гиперболический косинус x
<b>DDT(x)</b>	Первая производная $dx/dt$
<b>DELAY(x,y)</b>	Задержка сигнала на время y
<b>EXP(x)</b>	Экспоненциальная функция x
<b>FABS(x)</b>	То же самое, что и $\operatorname{abs}(x)$
<b>FLAT(x)</b>	Случайное число между -x и x с нормальным распределением
<b>FLOOR(x)</b>	Целое число, равное или меньше x
<b>GAUSS(x)</b>	Случайное число с распределением Гаусса с сигмой x
<b>HUPOT(x,y)</b>	$\sqrt{x^2+y^2}$
<b>IDT(x)</b>	Интеграл $\int x(t)$
<b>IF(x, y, z)</b>	y, если x истинно, иначе z
<b>INT(x)</b>	Округление x до целого числа
<b>INV(x)</b>	$\operatorname{inv}(x)=0$ , если $x>0$ ,5; $\operatorname{inv}(x)=1$ , если $x\leq 0$ ,5
<b>LIMIT(x,y,z)</b>	x, если $y<x<z$ ; y, если $x=y$ ; z, если $x>y$
<b>LN(x)</b>	Натуральный логарифм x
<b>LOG(x)</b>	Альтернатива $\ln(x)$
<b>LOG10(x)</b>	Десятичный логарифм x
<b>MAX(x,y)</b>	Максимальное из x или y
<b>MC(x,y)</b>	Случайное число между $x*(1+y)$ и $x*(1-y)$ с нормальным распределением
<b>MIN(x,y)</b>	Минимальное из x или y
<b>POW(x, y)</b>	Действительная часть $x^{**y}$ , например, $\operatorname{pow}(-0.5, 1.5)$ возвращает 0, а не 0,353553i
<b>PWR(x, y)</b>	$\operatorname{abs}(x)^{**y}$
<b>PWRS(x, y)</b>	$\operatorname{sgn}(x)*\operatorname{abs}(x)^{**y}$
<b>RAND(x)</b>	Случайные числа между 0 и 1 в зависимости от значения x
<b>RANDOM(x)</b>	Функция, подобная $\operatorname{rand}(x)$ , но с более плавными переходами между значениями
<b>ROUND(x)</b>	Ближайшее целое число к x
<b>SGN(x)</b>	Знак x
<b>SIN(x)</b>	Синус x
<b>SINH(x)</b>	Гиперболический синус x
<b>SQRT(x)</b>	Действительная часть квадратного корня x, например, $\operatorname{sqrt}(-1)$ возвращает 0, не 0,707107i
<b>TABLE(x,a,b,c,d,...)</b>	Табличная зависимость функции от x. Координаты точек (x, y) в виде таблицы, в промежутках между точками линейная аппроксимация
<b>TAN(x)</b>	Тангенс x
<b>TANH(x)</b>	Гиперболический тангенс x
<b>U(x)</b>	Ступенчатая функция: $u(x)=1$ , если $x>0$ ; $u(x)=0$ , если $x\leq 0$
<b>URAMP(x)</b>	Функция ограничения: $u\operatorname{ramp}(x)=x$ , если $x>0$ ; $u\operatorname{ramp}(x)=0$ , если $x\leq 0$
<b>WHITE(x)</b>	Случайные числа между -0,5 и 0,5 с плавными переходами между значениями, даже более плавными, чем в $\operatorname{random}()$

тею составить список сохраняемых данных. Если список соединений был сгенерирован из схемы, то, указав на выбранные ранее узлы и устройства и нажав кнопку мыши, можно отобразить графики в окне.

Если указать для сохранения только один график, то после запуска процесса моделирования он будет отображаться в окне графиков по ходу процесса, если только не дезактивирована ячейка **Marching Waveforms** в окне **Operation** панели управления.

Таблица 13. Математические операции

Символ операции	Назначение
&	Преобразование, эквивалентное логическому И (AND)
	Преобразование, эквивалентное логическому ИЛИ (OR)
^	Преобразование, эквивалентное логическому исключающему ИЛИ (XOR)
>	ИСТИНА (TRUE), если выражение слева больше выражения справа, иначе ЛОЖЬ (FALSE)
<	ИСТИНА (TRUE), если выражение слева меньше выражения справа, иначе ЛОЖЬ (FALSE)
>=	ИСТИНА (TRUE), если выражение слева больше или равно выражению справа, иначе ЛОЖЬ (FALSE)
<=	ИСТИНА (TRUE), если выражение слева меньше или равно выражению справа, иначе ЛОЖЬ (FALSE)
+	Сложение с плавающей точкой
-	Вычитание с плавающей точкой
*	Умножение с плавающей точкой
/	Деление с плавающей точкой
**	Возведение в степень, возвращается только действительная часть, например, $-2^{**1,5}$ возвращает 0, а не 2,82843i
!	Преобразование, эквивалентное логической инверсии
@	Оператор выбора шага

### .SAVEBIAS — сохранение рабочей точки на диске

Синтаксис:

```
.savebias <имя файла> [internal] [temp=<значение>] [time=< значение >] [repeat] [step=< значение >] [DC1=< значение >] [DC2=< значение >] [DC3=< значение >]
```

Это команда записи на диск текстового файла, который при последующем моделировании загружается командой **.LOADBIAS**. Для схемы, имеющей сложный расчет режима по постоянному току в рабочей точке, можно сохранить решение на диске, чтобы сэкономить время на расчете режима по постоянному току при последующих анализах.

Ключевое слово **INTERNAL** добавляется, чтобы указать, что внутренние узловые потенциалы некоторых устройств также должны быть сохранены с сохранением более полного решения по постоянному току.

Чтобы сохранить определенную рабочую точку по постоянному току при расчете переходных процессов **.TRAN**, необходимо указать момент времени. Первое решение для момента времени, следующего за указанным, будет сохранено. Модификатор **REPEAT** заставляет записывать решение по постоянному току через каждый период после указанного момента времени. В файл записывается только последнее решение по постоянному току. Можно указать **DC1**, **DC2** и **DC3**, чтобы извлечь единственную рабочую точку из **.DC**-анализа с вариацией параметров.

По директиве **.SAVEBIAS** записывается текстовый файл в формате директивы **.NODESET**. Отметим, что директивы **.NODESET** — только рекомендации для решения. Таким образом, вычислитель начнет повторять решение с узловыми напряжениями, заданными в директиве **.NODESET**, но продолжит повторять вычисления, пока не будет найдено правильное решение. Если нужно повторно запустить **.TRAN**-решение из рабочей точки по постоянному току, можно изменить файл **.NODESET** на **.IC**, чтобы принудить вычислитель начать решение с этой рабочей точки по постоянному току.

Поскольку суммарное состояние всех реактивных элементов схемы не сохраняется в **.SAVEBIAS** файле, успешное решение в этой технике достижимо не всегда.

### .STEP — ступенчатое изменение параметра

Эта команда заставляет неоднократно выполнять анализ, ступенчато изменяя температуру, параметр модели, глобальный параметр или параметр независимого источника. Ступени могут быть линейными, логарифмическими или определенными как список значений.

Например:

```
.step oct v1 1 20 5
```

Ступенчатое изменение напряжения независимого источника напряжения **V1** от 1 до 20 В с логарифмическим изменением ступеней с 5 точками на октаву.

Например:

```
.step I1 10u 100u 10u
```

Изменение тока независимого источника тока **I1** от 10 до 100 мкА с шагом 10 мкА.

Например:

```
.step param RLOAD LIST 5 10 15
```

Трехкратное выполнение моделирования с глобальным параметром **Rload**, равным 5, 10 и 15.

Например:

```
.step NPN 2N2222(VAF) 50 100 25
```

Ступенчатое изменение параметра **VAF** модели **NPN** от 50 до 100 с шагом 25.

Например:

```
.step temp -55 125 10
```

Изменение температуры от -55 до 125 °С с шагом 10 °С.

Вариации шага могут быть вложениями глубиной до трех уровней.

Например:

```
.step I1 10u 100u 10u temp list 25 100
```

Изменение тока независимого источника тока **I1** от 10 до 100 мкА с шагом 10 мкА при двух значениях температуры: 25 и 100 °С.

### .SUBCKT — описание макромодели

Для упрощения описания повторяющиеся схемы могут быть оформлены в виде макромодели со своим описанием и использо-



ваться в одной схеме многократно. Прежде чем начнется моделирование, список соединений схемы будет расширен с заменой каждого обращения к макромодели элементами схемы из описания макромодели. Нет никаких ограничений по размеру или сложности макромоделей.

Описание макромодели должно заканчиваться директивой `.ENDS`. Для удобства работы с текстовым файлом в дальнейшем целесообразно строку дополнить именем макромодели.

Пример использования макромодели:

```
*
* Описание схемы
X1 a b 0 divider
V1 a 0 pulse(0 1 0 .5µ .5µ 0 1µ)

* Описание макромодели
.subckt divider n1 n2 n3
r1 n1 n2 1k
r2 n2 n3 1k
.ends
.tran 3µ
.end
```

Расширение после запуска моделирования:

```
* Расширение X1 до набора из двух резисторов
r:1:1 a b 1k
r:1:2 b 0 1k
*
v1 a 0 pulse(0 1 0 .5µ .5µ 0 1µ)
.tran 3µ
.end
```

Отметим, что для элементов схемы, образованных расширением макромодели, созданы уникальные имена, основанные на имени макромодели и именах элементов описания макромодели.

### **.TEMP** — температурный анализ

Это устаревшая форма для команды ступенчатого изменения температуры. Все виды анализа выполняются для каждой указанной температуры.

Синтаксис:

```
.TEMP <температура1> <температура2> ...
```

Что эквивалентно:

```
.STEP TEMP LIST <температура1> <температура2> ...
```

### **.TF** — определение малосигнальной передаточной функции по постоянному току

В этом режиме определяется малосигнальная передаточная функция по постоянному току для узлового напряжения или тока ветви при малых изменениях параметра независимого источника. Параметры моделирования можно задать через диалоговое окно, показанное на рис. 34.

Синтаксис:

```
.TF V(<узел>[, <опорный узел>]) <независимый источник>
.TF I(<источник напряжения>) <независимый источник>
```

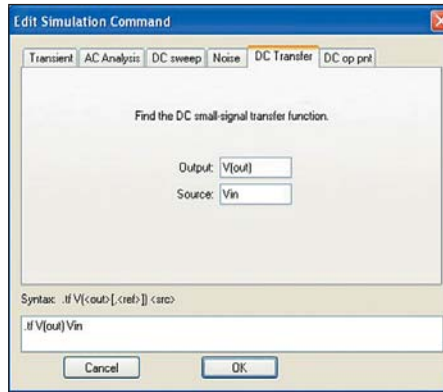


Рис. 34. Задание параметров .TF-анализа

Например:

```
.TF V(out) Vin
.TF V(5,3) Vin
.TF I(Vload) Vin
```

### **.TRAN** — анализ переходных процессов

По этой директиве выполняется анализ переходного процесса. Это наиболее информативное моделирование схемы. В процессе анализа вычисляется, что происходит в схеме после подачи питания. В качестве испытательных сигналов часто применяются независимые источники.

Синтаксис:

```
.TRAN <Tstep> <Tstop> [Tstart [dTmax]] [modifiers]
.TRAN <Tstop> [modifiers]
.TRAN <шаг вывода данных> <конечное время> [начальный момент времени вывода данных [максимальный шаг]] [модификаторы]
.TRAN <конечное время> [модификаторы]
```

Первая запись представляет собой традиционную форму для SPICE команды `.TRAN`. `Tstep` — шаг вывода графиков. `LTspice` использует сжатие графиков, так что этот параметр не имеет большого значения и может быть опущен или установлен в ноль. `Tstop` — продолжительность моделирования. Анализ переходного процесса всегда начинается с момента времени, равного нулю. Однако, если определено `Tstart`, данные для графика между нулем и `Tstart` не сохраняются. Это — средство управления размером файлов графиков, позволяющее иг-

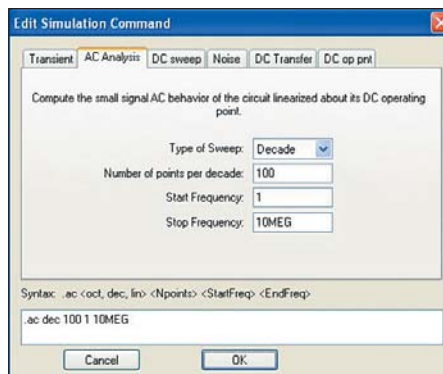


Рис. 35. Задание параметров .TRAN-анализа

норировать переходные процессы при включении. Последний параметр `dTmax` является максимальным шагом интегрирования. Если заданы `Tstart` или `dTmax`, то должен быть задан и `Tstep`. Параметры моделирования легко задаются с использованием диалогового окна, изображенного на рис. 35.

В строку `.TRAN` могут быть помещены несколько модификаторов из перечисленных далее.

### UIC

Использование начальных условий. Обычно расчет рабочей точки по постоянному току выполняется перед началом анализа переходного процесса. Директива подавляет эту инициализацию. Начальные условия для некоторых элементов схемы могут быть определены случайным образом. `UIC` — не особенно рекомендуемая возможность `SPICE`. Пропуск расчета рабочей точки по постоянному току пункта ведет к нереализуемому физически начальному условию. Рассмотрим, например, подключение емкости к источнику напряжения. Узловое напряжение равно нулю, если не задано иное. Тогда в первый момент бесконечный ток обязан заряжать конденсатор. Симулятор не может найти достаточно короткий шаг времени и выдает сообщение **time step too small convergence fail**.

### STARTUP

Модификатор подобен оригинальному `SPICE UIC`. Он означает, что независимые источники должны отключаться на первые 20 мс моделирования. Однако анализ рабочей точки по постоянному току выполнен с использованием ограничений, заданных директивой `.IC`.

### STEADY

Останавливает моделирование по достижению установившегося режима. Это требуется для создания отчета о расчете эффективности. Обнаружение установившегося режима записано в макромоделах ВИП. Обычное решение — обнаружение нулевого значения в выходном токе усилителя ошибки, усредненном по периоду тактовой частоты. Алгоритм предполагает анализ выходного сигнала усилителя ошибки. Доля импульсного тока, считающаяся нулевым током, задается параметром `SSTOL`.

Автоматическое обнаружение установившегося режима может быть неудачным вследствие неоптимального критерия его детектирования. Можно в интерактивном режиме определить установившийся режим следующим образом: как только моделирование начинается, выполните команду меню **Simulate>Efficiency Calculation>Mark Start**. При выполнении этой команды в первый раз вы указываете `LTspice`, что собираетесь вручную определять пределы интегрирования. После того как схема достигла установившегося режима, выполните эту команду снова. Это очистит историю и перезапустит расчет

потерь. Затем, по меньшей мере через 10 тактовых импульсов выполните команду `Simulate>Efficiency Calculation>Mark End`. Каждый раз при выполнении команды `Simulate>Efficiency Calculation>Mark Start` повторно начинается расчет потерь и очищается история графиков. Это хороший метод воспрепятствовать чрезмерному увеличению файла и замедлению рисования. Рекомендуется периодически выполнять `Simulate>Efficiency Calculation>Mark Start` всякий раз, когда ясно, что накоплено существенное количество данных, которые нецелесообразно включать в расчет потерь.

Используйте директиву `.IC` для определения узловых напряжений и токов через катушки индуктивности, чтобы уменьшить длительность анализа переходного процесса, предшествующего установившемуся режиму.

### NODISCARD

Не удалять часть моделирования переходного процесса, прежде чем будет достигнут установившийся режим.

### STEP

Расчет отклика схемы на ступенчатое воздействие. Эта функция работает с использованием в качестве нагрузки источника тока со ступенчатым изменением тока. Процедура включает в себя следующие этапы:

1. Определяется установившийся режим и очищается история, если не используется `NODISCARD`.

2. Нагрузка ступенчато изменяется к следующему значению из списка со скоростью 20 А/мкс.

3. Определяется установившийся режим.

4. Нагрузка ступенчато изменяется к следующему значению из списка или отключается, если его нет.

В сложной схеме автоматически изменение тока нагрузки можно и не обнаружить. В подобном случае лучше использовать команду `.TRAN` для запуска моделирования переходного процесса и наблюдать начало и конец периода желаемого отклика на скачок нагрузки. Используйте команду `PWL`, чтобы программировать выходной ток нагрузки и переключать его на различные уровни в желаемые моменты времени. Например:

```
PWL(0 0.5 1m 0.5 1.01m 0.1 3m 0.1 3.01m 0.5)
Ток нагрузки начинается с 0,5 А в момент времени 0,
пребывает равным 0,5 А в течение 1 мс,
переключается на 0,1 А в момент времени 1,01 мс,
пребывает равным 0,1 А до 3 мс,
переключается на 0,5А в 3,01 мс и остается равным 0,5 А.
```

`PWL` может иметь почти неограниченную последовательность пар (время, значение).

### .WAVE —

*запись выбранных узлов в .wav-файл*

`LTspice` может записывать звуковые файлы `.wav`. Эти файлы можно слушать или использовать в качестве входных для другого моделирования.

Синтаксис:

```
.wave <имя файла .wav> <Nbits> <SampleRate> V(out) [V(out2) ...]
```

Например:

```
.wave C:\output.wav 16 44.1K V(left) V(right)
```

Здесь `<имя файла .wav>` — полный путь к `.wav`-файлу или относительный путь, определяемый из каталога, содержащего симулируемую схему или список соединений. `<Nbits>` — число битов в выборке. Допустимый диапазон — от 1 до 32 битов. `<SampleRate>` — количество выборок в секунду. Допустимый диапазон — от 1 до 4 294 967 295 выборок в секунду. Далее в синтаксисе — перечисление узловых напряжений, значения которых нужно сохранить. Каждый узел будет независимым каналом в `.wav`-файле. Число каналов может быть от одного до 65 535. Можно записать ток в устройстве, например, `Ib(Q1)`, или узловое напряжение. `.wav`-аналог цифрового преобразователя имеет полномасштабный диапазон от  $-1$  до  $+1$  вольт или ампера.

В файле `... \LTC\SwCADIII\examples\Educational\wavein.asc` приведен пример использования директивы `.WAVE`.

В следующей статье — примеры моделирования схем с использованием программы. ■