



**ОАО “НИИМЭ и “Микрон”**

**Руководство пользователя  
к системе управления моделированием  
ArtistKit для технологии  
“Микрон HCMOS10\_LP\_7M\_2.5V” 90нм.**

## Оглавление

<b>1 Введение .....</b>	<b>4</b>
1.1 Поддерживаемые платформы .....	4
1.2 Поддерживаемые симуляторы .....	4
1.3 Методология нетлизования и интеграции .....	4
<b>2 Управление углами моделирования .....</b>	<b>5</b>
2.1 Введение .....	5
2.2 Методология .....	5
2.3 Пример с симулятором Spectre .....	6
2.3.1 Device Library Files .....	6
2.3.2 Module Library Files .....	7
2.3.3 Device Library файлы, поддерживающие некоррелированные худшие случаи (uncorrelated worst-cases) .....	7
2.3.4 Device Library файлы, поддерживающие предопределённые, частично коррелированные худшие случаи (Pre-Defined Partially Correlated Worst-cases) .....	8
2.3.5 Common Library файлы, поддерживающие предопределённые, частично коррелированные худшие случаи (Pre-Defined Partially Correlated Worst-cases) .....	12
2.3.6 Device Library файлы, поддерживающие предопределённые, полностью коррелированные худшие случаи (Pre-Defined Totally Correlated Worst-cases) .....	14
2.3.7 Common Library файлы, поддерживающие предопределённые, полностью коррелированные худшие случаи (Pre-Defined Totally Correlated Worst-cases) .....	17
2.3.8 Device Library файлы, поддерживающие определённые пользователем, полностью коррелируемые худшие случаи (User-Defined Totally Correlated Worst-cases) .....	19
2.3.9 Common Library файлы, поддерживающие определённые пользователем, полностью коррелированные худшие случаи (User-Defined Totally Correlated Worst-cases) .....	22
2.3.10 Модельный файл .....	23
2.4 Выбор углов моделирования в интерактивном режиме .....	23
2.5 Управление сценариями .....	29
2.6 Сохранение состояния установленных углов (Setup Corners State) .....	30
2.7 Загрузка состояния установленных углов (Setup Corners State) .....	31
2.8 Запуск моделирования .....	32
2.9 Создание комментариев в окне waveform .....	35
2.10 Управление библиотеками параметров моделей. ....	37
2.10.1 Создание новой библиотеки. ....	38
2.10.2 Изменение пути библиотеки. ....	39
2.10.3 Удаление библиотеки .....	39
2.11 Добавление пользовательского файла .....	40
2.11.1 Определение и редактирование пользовательского файла. ....	40
2.11.2 Команда Include On .....	41
2.11.3 Команда Include Off .....	41
<b>3 Настройка и запуск анализа Monte Carlo .....</b>	<b>42</b>
3.1. Введение .....	42
3.2 Методология .....	42
3.3 Пример со Spectre .....	43
3.3.1 Device Library файлы .....	43
3.3.2 Model File .....	45
3.4 Выбор статистических углов и эффектов рассогласования .....	46
3.5 Запуск анализа Монте-Карло со spectre .....	48

<b>4 Управление пользовательскими углами моделирования .....</b>	<b>50</b>
4.1 Введение .....	50
4.2 Методология .....	50
4.3 Выбор пользовательских углов моделирования .....	51
4.4 Управление пользовательскими углами моделирования .....	54
4.4.1 Определение нового угла моделирования.....	55
4.4.2 Изменение угла моделирования .....	55
4.4.3 Удаление угла моделирования.....	56
4.4.4 Загрузка угла моделирования.....	56
4.5 Создание комментариев к окну waveform .....	57
4.6 Сохранение состояния (State) в окне Setup User-Defined Corners .....	58
4.7 Загрузка состояния (State) в окне Setup User-Defined Corners.....	59
<b>5 Импорт списков соединений (import netlists).....</b>	<b>61</b>
5.1 Введение .....	61
5.2 Методология .....	61
5.3 Импорт списка соединений с текстовым представлением .....	64
5.4 Включение/ выключение импорта списка соединений.....	65
<b>6 Работа в Hierarchy Editor (HED).....</b>	<b>67</b>
6.1 Введение .....	67
6.2 Работа в Hierarchy Editor .....	67
<b>7 Настройка и запуск смешанного моделирования .....</b>	<b>70</b>
7.1 Введение .....	70
7.2 Создание конфигурации.....	70
7.3 Открытие конфигурированного схмотехнического представления из CIW окна .....	71
7.4 Опции разбиения (Partitioning Options).....	71
7.5 Открытие сессии смешанного моделирования.....	72
7.6 Настройка моделей.....	73
7.7 Настройка аналоговых воздействий (Analog Stimuli).....	73
7.8 Настройка цифровых воздействий (digital stimuli) .....	73
7.9 Запуск моделирования .....	76
<b>8 Контактная информация .....</b>	<b>77</b>

## 1 Введение

ArtistKit - это продукт, используемый в Cadence Analog Design Environment (ADE, также известном как Analog Artist) и в редакторе *Hierarchy Editor Simulation Environment*. Этот продукт решает следующие основные задачи:

- обеспечение комплексного решения для управления моделями устройств
  - настройка углов моделирования
  - определение и настройка пользовательских (user-defined) углов моделирования
  - моделирование по углам (corner simulation)
  - настройка статистических моделей и анализ методом Монте-Карло (monte-carlo analysis)
- обеспечение методики импорта внешних списков соединений (netlists) для моделирования схемы с учетом паразитных параметров (post-layout simulation).
- обеспечение некоторых значений по умолчанию для программной среды моделирования (симулятор по умолчанию (default simulator), управление видами (Switch View List, Stop View List), меню и опции программной среды и т.д.).

### 1.1 Поддерживаемые платформы

- ix86-linux Red Hat release 7.2 & Red Hat Enterprise 3 (Taroon)

### 1.2 Поддерживаемые симуляторы

- *Eldo* симулятор от Mentor-Graphics
- *Spectre* симулятор от Cadence
- *Star-Hspice* симулятор от Synopsys

### 1.3 Методология нетлистования и интеграции

ADE поддерживает два метода нетлистования:

- обычный *Spice Socket*, базирующийся на *Cadence Spice (cdsSpice)*
- прямое нетлистование (*Direct Netlisting*)

ArtistKit поддерживает:

- Метод прямого нетлистования (*Direct Netlisting*) для *Spectre*.
- Для Eldo применимы оба метода *Direct Netlisting* (для *eldoD*) и *Spice Socket* (*eldo*).
- Оба метода *Direct Netlisting* и *Spice Socket* применимы для *Star-Hspice*. ADE обращается к этим симуляторам как *hspiceD* и *hspiceS* соответственно.

## 2 Управление углами моделирования

### 2.1 Введение

Глава описывает, как использовать пользовательский графический интерфейс *Setup Corners*, определяющий параметры моделей и выбор углов моделирования перед моделированием в среде Analog Design Environment (ADE).

### 2.2 Методология

Какой бы ни использовался метод нетлистования, ADE предлагает несколько возможностей включить файл в конечный список соединений. В случае метода *Spice Socket*, этот файл может и не быть проанализирован с помощью *cdsSpice*, поэтому, во всех случаях, может быть использован родной язык симулятора. Таким образом, в любом случае для управления параметрами моделирования необходимо:

- иметь данные моделирования в подходящем формате ( *Library Files* ) для заданного симулятора. Эти данные расположены в *Library* как Design Kit (DK).
- использовать пользовательский графический интерфейс *Setup Corners* продукта *ArtistKit* для генерации файла ( *Model File* ) с постоянным именем, содержащим все указания симулятору для получения ожидаемой модели.
- указать ADE систематически включать этот файл (*Model File*) в конечный список соединений (final netlist).

Методология, используемая для определения электрических и статистических моделей устройств, различается от одной технологии к другой.

Для технологии 90 нм методология позволяет описать:

- Не коррелированные худшие случаи.
- Предопределенные частично коррелированные худшие случаи, коррелирующие подмножества семейств устройств.
- Предопределенные полностью коррелированные статистические модели, коррелирующие все семейства устройств (используемые во время анализа Монте-Карло, как описано в главе 3 ).
- Определяемые пользователем полностью коррелированные худшие случаи, коррелирующие все семейства устройств.

Примечание 1: Предоставленные средства проектирования могут поддерживать комбинацию некоррелированных худших случаев и частично коррелированных худших случаев.

Примечание 2: Статистические модели и определяемые пользователем худшие случаи коррелируют все устройства вместе, что влечет за собой наиболее точное моделирование процесса.

Примечание 3: Для целей моделирования некоторые средства проектирования определяют модели устройства в нескольких модулях. Некоторые семейства устройств могут быть поделены между несколькими модулями: например, МОП-транзистор и RF МОП-транзистор часто описываются в разных модулях, хотя они должны варьироваться вместе в условиях моделирования углов. Такая совместимость поддерживается продуктом ArtistKit.

## 2.3 Пример с симулятором Spectre

### 2.3.1 Device Library Files

Большинство информации о моделях сгруппировано по типу семейства и хранится в *Device Library Files*, используя расширение файла *.scs* (spectre control syntax):

<LIBRARY\_PATH>/svtgp.scs

<LIBRARY\_PATH>/rpolyp.scs

...

Внутри каждого такого файла данные сгруппированы по типу угла моделирования, используя выражения *library-endlibrary/section-endsection*:

```
library <family>

    section <corner>

        [<данные> которые зависят от поддерживаемой методологии]

    endsection <corner>

endlibrary <family>
```

### 2.3.2 Module Library Files

Когда *spectreHDL* или *Verilog-a* описания необходимы для выражения некой специфической особенности модели, модули описываются в файлах с расширением *.module* или *.va*, которые включены в файл с расширением *.scs*:

```
<LIBRARY_PATH>/svtgp.module

<LIBRARY_PATH>/rpolyp.va ...
```

### 2.3.3 Device Library файлы, поддерживающие некоррелированные худшие случаи (*uncorrelated worst-cases*)

Для этих случаев информация определяется внутри выражений *section-<corners>-endsection*. Также внутри данных выражений может содержаться ссылка на Module Library файлы.

Пример для **Device Library** файла *rpolyp.scs*, который определяет некоррелированные углы (corners): *typ*, *rmin*, *rmax* для *rpolyp* семейства:

```
library rpolyp
section typ
subckt RPO1PAM1 PLUS MINUS B ...
ends RPO1PAM1
subckt RPO1PAM2 PLUS MINUS B ...
ends RPO1PAM2
ahdl_include "rpolyp.module"
endsection typ
section rmin
subckt RPO1PAM1 PLUS MINUS B ...
ends RPO1PAM1
```

```
subckt RPO1PAM2 PLUS MINUS B ...
ends RPO1PAM2
ahdl_include "rpolyp.module"
endsection rmin
section rmax
subckt RPO1PAM1 PLUS MINUS B ...
ends RPO1PAM1
subckt RPO1PAM2 PLUS MINUS B ...
ends RPO1PAM2
ahdl_include "rpolyp.module"
endsection rmax
endlibrary
```

### 2.3.4 Device Library файлы, поддерживающие предопределённые, частично коррелированные худшие случаи (Pre-Defined Partially Correlated Worst-cases)

В этом случае информация (*<data>*) определяется внутри выражений *section* *<corners>* *endsection* и содержит:

- Установленные параметры углов (*<corner>*), выражающие отклонение (*deviation*) параметров процесса по отношению к их средним значениям и соответствующие данному семейству (*<family>*). Данное отклонение (*deviation*) обычно выражается количественно через показатели сигма (*sigmas*). В дальнейшем, будем подразумевать под  $D_i$  отклонение (*deviation*)  $i$ -ого параметра угла (*<corner>*) семейства (*<family>*).

Отметим, что семейство (*<family>*) часто определяет более одного семейства в понимании методологии некоррелированных худших случаев (*uncorrelated worst-cases methodology*): это позволяет определить более реалистично расчет углов для корреляции между семействами. Именно поэтому методология называется методологией частично коррелированных худших случаев (*Partially Correlated Worst-cases*).

- Рекурсивные ссылки к секциям (*sections*) в этом же файле (*Library File*):
  - секция, относящаяся к значениям параметра угла (*<corner>*), выражающим количественно отклонение реального численного значения параметра процесса



(<process>) (PARAM address). Назовём  $P_i$  параметром процесса (process parameter)  $i$ -ого семейства (<family>),  $T_i$  и  $\sigma_i$  - типичное значение и стандартное отклонение  $i$ -ого параметра процесса.

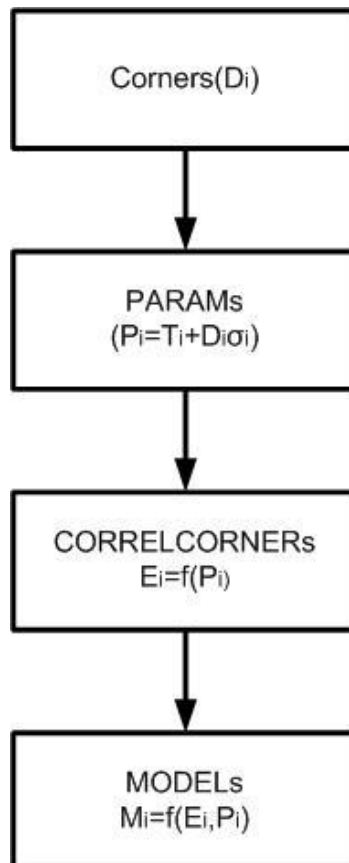
$$P_i = T_i + D_i \sigma_i$$

- секция, показывающая отношение электрических (<electrical>) параметров семейства с параметрами процесса (<process>) текущего семейства или с параметрами процесса так называемых общих семейств (<common> families) (описано далее) через корреляционные выражения (<family>\_CORRESLCORNER address). Назовём  $E_i$   $i$ -ым электрическим (<electrical>) параметром семейства (<family>).

$$E_i = f(P_j), \text{ где } j = 1, \dots, i, \dots$$

- секция, определяющая описание моделей самих устройств исходного семейства (MODEL address). Данные модели образуются с помощью установок значений предопределённых параметров (pre-define parameter), одни и те же модели могут быть образованы с помощью установки значений параметров, определяемых пользователем (user-define parameter) или установки распределений (distribution) параметров процесса (описано далее). Обозначим  $M_i$   $i$ -ым параметром модели (<model>) семейства (<family>).

$$M_i = f(E_i, P_j)$$



*Рис. 2.1 Секции Library File.*

Пример файла svtgp.scs, который определяет углы TT, FF и SS для семейства svtgp (p и n коррелированы):

**library svtgp****section TT**

```
parameters nsigma_nsvtgp_dmu = 0
parameters nsigma_psvtgp_dmu = 0 ...
include "svtgp.scs" section=PARAM
include "svtgp.scs" section=CORRELCORNER
include "svtgp.scs" section=MODEL
```

**endsection TT****section FF**

```
parameters nsigma_nsvtgp_dmu = 2
parameters nsigma_psvtgp_dmu = 2 ...
include "svtgp.scs" section=PARAM
include "svtgp.scs" section=CORRELCORNER
include "svtgp.scs" section=MODEL
```

**endsection FF****section SS**

```
parameters nsigma_nsvtgp_dmu = -2
parameters nsigma_psvtgp_dmu = -2 ...
include "svtgp.scs" section=PARAM
include "svtgp.scs" section=CORRELCORNER
include "svtgp.scs" section=MODEL
```

**endsection SS****section PARAM**

```
parameters nsvtgp_dmu = 0.0 + nsigma_nsvtgp_dmu*1.250e-02
parameters psvtgp_dmu = 0.0 + nsigma_nsvtgp_dmu*1.250e-02 ...
```

**endsection PARAM****section CORRELCORNER**

```
parameters nsvtgp_dx1 = poly_cdvar
parameters psvtgp_dx1 = poly_cdvar
parameters psvtgp_drshgu = ppoly_drshu ...
```

**endsection CORRELCORNER****section MODEL**

```
subck nsvtgp D G S B ...
parameters dmdu = nsvtgp_dmdu
parameters x1    = -1.5e-0.8 +nsvtgp_dx1
param rshgu      = 150 + nsvtgp_drshgu ...
ends nsvtgp

subck psvtgp D G S B ...
parameters dmdu = psvtgp_dmdu
parameters x1    = -1.5e-0.8 +psvtgp_dx1
param rshgu      = 710 + psvtgp_drshgu ...
ends psvtgp

endsection MODEL
endlibrary svtgp
```

Пример файла rpolyp.scs, который определяет углы typ, rmin и rmax для семейства rpolyp:

```
library rpolyp

section typ
parameters nsigma_ppoly_drshu_rpolyp_drho = 0
parameters nsigma_rpolyp_ddw = 0 ...
include "rpolyp.scs" PARAM
include "rpolyp.scs" CORRELCORNER
include "rpolyp.scs" MODEL
endsection typ

section rmin
parameters nsigma_ppoly_drshu_rpolyp_drho = 2
parameters nsigma_rpolyp_ddw = 2 ...
include "rpolyp.scs" PARAM
include "rpolyp.scs" CORRELCORNER
include "rpolyp.scs" MODEL
endsection rmin

section rmax
parameters nsigma_ppoly_drshu_rpolyp_drho = -2
parameters nsigma_rpolyp_ddw = -2 ...
include "rpolyp.scs" PARAM
include "rpolyp.scs" CORRELCORNER
include "rpolyp.scs" MODEL
```

**endsection rmax**

**section PARAM**

parameters **ppoly\_drshu\_rpolyp\_drho** = 0 + **nsigma\_ppoly\_drho**\*13.333

parameters rpolyp\_ddw = 0 + nsigma\_rpolyp\_ddw\*4e-9 ...

**endsection PARAM**

**section CORRELCORNER**

parameters **rpolyp\_drho** = ppolyp\_drshu\_rpolyp\_drho

**endsection CORRELCORNER**

**section MODEL**

subck rpporpo plus minus b ...

parameters rho = 320 + **rpolyp\_drho**

parameters dw = 39e-9+rpolypddw ...

ends rpporpo

**endsection MODEL**

**endlibrary rpolyp**

### 2.3.5 Common Library файлы, поддерживающие предопределённые, частично коррелированные худшие случаи (*Pre-Defined Partially Correlated Worst-cases*)

Некоторые из параметров, определяющие процесс корреляции (process correlations) (например *polygp\_cdvar*, *npoly\_drshu* и *ppoly\_drshu* для транзисторов), могут быть не определены в Device Library Files. Если эти параметры встречаются более чем в одном типе приборов, то они группируются в так называемые **Common Library** файлы.

<LIBRARY\_PATH>/common\_poly.scs

<LIBRARY\_PATH>/common\_gol.scs ...

Common Library файл имеет схожую структуру с Device Library файлом, но

- в нём не определены никакие модели (отсутствует MODEL section)
- корреляция не обязательна (CORREL address может быть пустым)

Пример **Common Library** файла *common\_poly.scs*:

**library common\_poly**

**section TT**

parameters nsigma\_polygp\_cdvar = 0

parameters nsigma\_npoly\_drshu = 0

parameters **nsigma\_ppoly\_drshu** = 0 ...

include "common\_poly.scs" PARAM

```
include "common_poly.scs" CORRELCORNER
```

```
endsection TT
```

```
section FF
```

```
parameters nsigma_polygp_cdvar = -2
```

```
parameters nsigma_npoly_drshu = -3
```

```
parameters nsigma_ppoly_drshu = -3 ...
```

```
include "common_poly.scs" PARAM
```

```
include "common_poly.scs" CORRELCORNER
```

```
endsection FF
```

```
section SS
```

```
parameters nsigma_polygp_cdvar = 2
```

```
parameters nsigma_npoly_drshu = 3
```

```
parameters nsigma_ppoly_drshu = 3 ...
```

```
include "common_poly.scs" PARAM
```

```
include "common_poly.scs" CORRELCORNER
```

```
endsection SS
```

```
section PARAM
```

```
parameters polygp_cdvar = 0.0 + nsigma_nsigma_polygp_cdvar*3.000e-09
```

```
parameters npoly_drshu = 0 + nsigma_npoly_drshu*16.667
```

```
parameters ppoly_drshu = 0 + nsigma_ppoly_drshu*33.333 ...
```

```
endsection PARAM
```

```
section CORRELCORNER
```

```
endsection CORRELCORNER
```

```
endlibrary common_poly
```

Отметим, что *svtgp* транзистор и *rpolyr* резистор не коррелированы вместе. Они могли бы быть коррелированы только с помощью заменяющего равенства

```
parameters rpolyr_drho = ppoly_drshu_rpolyr_drho ...
```

в *rpolyr\_CORRELCORNER* section с помощью

```
parameters rpolyr_drho = ppoly_drshu
```

В действительности *ppoly\_drshu* параметр определен в *common\_poly.scs* файле и коррелирует оба семейства. Разработчик имеет выбор сделать транзистор и резистор некоррелированными с помощью выбора угла (corner selection): когда выбирается предустановочный худший случай для транзистора (mosfet pre-defined worst-case), то нет

воздействия на резисторы, когда выбирается предустановочный худший случай для резистора (resistor pre-defined worst-case), то нет воздействия на транзисторы.

### 2.3.6 Device Library файлы, поддерживающие предопределённые, полностью коррелированные худшие случаи (*Pre-Defined Totally Correlated Worst-cases*).

В данном случае информация *<data>* определена внутри выражений *section-<corner>-endsection* и содержит:

- Специальные статистические углы (*statistical corners*) (начинаются со строки *stat*). По сравнению с предопределёнными углами (*pre-defined corners*) эти углы не принимают фиксированные значения параметров процесса, но определяют статистические распределения (*statistical distributions*). Используем те же обозначения, как и в предыдущем разделе (распределение (*distrib*) может быть Гауссовское, равномерное, логарифмическое и т.д.)

$$P_i = \text{distrib}(\mu_i, \sigma_i)$$

- Рекурсивные ссылки на другие секции в этом же библиотечной файле:
  - секции, показывающие отношение параметров процесса с электрическими параметрами через корреляционные равенства (*CORREL address*).Отметим, что данные корреляции могут отличаться от определённых с помощью предустановленных частично коррелируемых худших случаев (*Pre-Defined Partially Correlated Worst-Cases*): *CORREL* вместо *CORRELCORNER*. Используем те же условные обозначения, как и в предыдущем разделе,

$$E_i = f(P_j), \text{ где } j = 1, \dots, i, \dots$$

- секции, определяющие описания самих моделей всех приборов данной библиотеки (*MODEL address*). Эти описания такие же, какие используются для определённых с помощью предустановленных частично коррелируемых худших случаев (*Pre-Defined Partially Correlated Worst-Cases*).

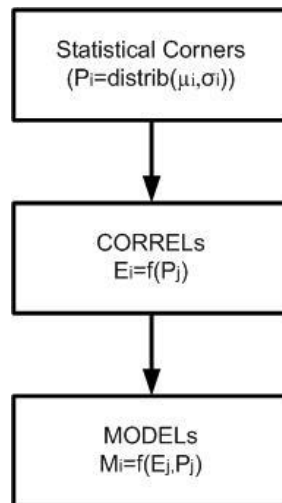


Рис. 2.2 Секции Library File.

Пример файла *svtgp.scs*, который определяет углы *statcrolles* и *statmultifab* для семейства *svtgp* (р и п коррелированы):

**library svtgp**

**section statcrolles**

parameters nsvtgp\_dmu = 0.0

parameters psvtgp\_dmu = 0.

statistics {

process {

    vary nsvtgp\_dmu dist=gauss std=0.0125 percent=no

    vary psvtgp\_dmu dist=gauss std=0.0125 percent=no ...

}

}

include "svtgp.scs" section=CORREL

include "svtgp.scs" MODEL

**endsection statcrolles**

**section statmultifab**

parameters nsvtgp\_dmu = 0.0

parameters psvtgp\_dmu = 0.

statistics {

process {

    vary nsvtgp\_dmu dist=unif N=0.05 percent=no

    vary psvtgp\_dmu dist=unif N=0.05 percent=no ...

}

}

include "svtgp.scs" section=CORREL

```
include "svtgp.scs" MODEL
endsection statmultifab
section CORREL
parameters nsvtgp_dx1 = polygp_cdvar
parameters psvtgp_dx1 = polygp_cdvar
parameters nsvtgp_drshgu = npoly_drshu
parameters psvtgp_drshgu = ppoly_drshu ...
endsection CORREL
section MODEL
subck nsvt D G S B ...
parameters dm_u = nsvtgp_dm_u
parameters x1 = -1.5e-08+nsvtgp_dx1
parameters rshgu = 150+nsvtgp_drshgu ...
ends nsvt
subck psvt D G S B ...
parameters dm_u = psvtgp_dm_u
parameters x1 = -1.5e-08+psvtgp_dx1
parameters rshgu = 710+psvtgp_drshgu ...
ends psvt
endsection MODEL
endlibrary svtgp
```

Пример для *rpoly.scs*, который определяет углы *statcrolles* и *statmultifab* для семейства *rpoly*:

```
library rpolyp
section statcrolles
parameters rpolyp_ddw = 0.0
statistics {
  process {
    vary rpolyp_ddw dist=gauss std=4e-09 percent=no
    ...
  }
}
include "rpoly.scs" section=CORREL
include "rpoly.scs" MODEL
endsection statcrolles
```



**section statmultifab**

parameters rpolyp\_ddw = 0.0

parameters psvtgp\_dmu = 0.

statistics {

process {

    vary rpolyp\_ddw dist=unif N=1.28e-08 percent=no

    ...

  }

}

include "rpolyp.scs" section=CORREL

include "rpolyp.scs" MODEL

**endsection statmultifab****section CORREL**

parameters rpolyp\_drho = ppoly\_drshu ...

**endsection CORREL****section MODEL**

subck rpporpo plus minus b ...

param rho = 320 + rpolyp\_drho

param dw = 39e-9+rpolyp\_ddw ...

ends rpporpo

**endsection MODEL****endlibrary rpolyp**

### ***2.3.7 Common Library файлы, поддерживающие предопределённые, полностью коррелированные худшие случаи (Pre-Defined Totally Correlated Worst-cases).***

Common Library файл имеет схожую структуру с Device Library файл, но

- в нём не определены никакие модели (отсутствует MODEL section)
- корреляция не обязательна (CORREL address может быть пустым)

Пример **Common Library** файла *common\_poly.scs*:

**library common\_poly****section statcrolles**

parameters polygp\_cdvar = 0.0

parameters npoly\_drshu = 0.0

parameters ppoly\_drshu = 0.0

```
statistics {  
  process {  
    vary polygp_cdvar dist=gauss std=3e-09 percent=no  
    vary npoly_drshu dist=gauss std=16.667 percent=no  
    vary ppoly_drshu dist=gauss std=33.333 percent=no ...  
  }  
}  
include "common_poly.scs" PARAM  
include "common_poly.scs" CORREL  
endsection statcrolles  
section statmultifab  
parameters polygp_cdvar = 0.0  
parameters npoly_drshu = 0.0  
parameters ppoly_drshu = 0.0  
statistics {  
  process {  
    vary polygp_cdvar dist=unif N=9e-09 percent=no  
    vary npoly_drshu dist=unif N=66.667 percent=no  
    vary ppoly_drshu dist=unif N=100 percent=no ...  
  }  
}  
include "common_poly.scs" PARAM  
include "common_poly.scs" CORREL  
endsection statmultifab  
section CORRELCORNER  
endsection CORRELCORNER  
endlibrary common_poly
```

Отметим, что svtgp транзистор и rpolyp резистор не коррелируются вместе через *ppoly\_drshu* параметр. В svtgp\_CORREL секции мы получаем

```
parameters psvtgp_drshgu ='ppoly_drshu'
```

В polyp\_CORELL секции мы получаем

```
parameters rpolyp_drho = ppoly_drshu
```

### 2.3.8 Device Library файлы, поддерживающие определённые пользователем, полностью коррелируемые худшие случаи (*User-Defined Totally Correlated Worst-cases*).

Современные средства проектирования дают пользователю возможность настраивать предопределённые худшие случаи (*Pre-Defined Worst-cases*) и затем определять так называемые определяемые пользователем худшие случаи (*User-Defined Worst-cases*). Таким образом, в случае статистических моделей, определённые пользователем худшие случаи (*User-Defined Worst-cases*) коррелированы **всеми** семействами приборов (*devices families*).

В данном случае информация определяется внутри выражений section <corner> endsection и содержит:

- Специальные углы (называемые USER), выражающие отклонение параметров процесса семейства по отношению к определенному пользователем отклонению. Назовём  $UD_j$  - j-ым определенным пользователем отклонением. Реальное отклонение  $D_i$  i-ого параметра процесса семейства определяется как:

$$D_i = k_i UD_j$$

где  $k_i$  определяет значение отклонения i-ого параметра процесса семейства (возможные значения -1, 1).

Отметим, по сравнению с предустановленными углами, определяемые пользователем углы могут выражать отклонение как число с плавающей запятой, а не только целого числа.

- Рекурсивные ссылки на другие секции в этом же библиотечном файле:
  - секция, относящаяся к значениям параметра семейства (<family>), выражающим показатель отклонения реальных численных значений параметров процесса ((PARAMUSER) address). Обозначим  $P_i$  i-ым параметром процесса семейства,  $T_i$  и  $\sigma_i$  - это среднее значение и стандартное отклонение соответственно:

$$P_i = T_i + D_i \sigma_i$$

$$P_i = T_i + k_i UD_j \sigma_i$$

Отметим, что выражения могут отличаться от определённых с помощью предопределённых, частично коррелируемых худших случаев. Address PARAMUSER вместо PARAM для того, чтобы перекоррелировать все семейства приборов.

- секция, показывающая отношение электрических параметров к параметрам процесса текущего семейства или параметров процесса, так называемых общих

(<common>) семейств (описано далее) через корреляционные выражения (<family>\_CORREL address).

Отметим, что эта корреляция использует статистические модели, которые прозывают коррелировать все семейства приборов.

Определим  $E_i$   $i$ -ым электрическим (<electrical>) параметром семейства (<family>)

$$E_i = f(P_j), \text{ где } j = 1, \dots, i, \dots$$

- секция, определяющая описания самих моделей всех приборов данного семейства (MODEL address). Назовём  $M_i$   $i$ -ым параметром модели семейства.

$$M_i = f(E_i, P_i)$$

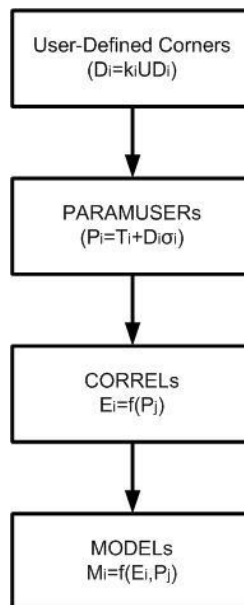


Рис. 2.3 Секции Library File.

Пример файла svtgp.scs, который определяет угол USER для семейства svtgp (два определенных пользователем отклонения (user-defined deviation)):

**library svtgp**

**section USER**

parameters nsigma\_nsvtgp\_dmu = 1\*min(4.0,max, (-4.0, nstgp\_user))

parameters nsigma\_psvtgp\_dmu = 1\*min(4.0,max, (-4.0, nstgp\_user))

...

include "svtgp.scs" PARAM

include "svtgp.scs" CORRELCORNER

include "svtgp.scs" MODEL

**endsection USER**

**section PARAMUSER**

```
parameters nsvtgp_dmu = 0.0 + nsigma_nsvtgp_dmu*1.250e-02
```

```
parameters psvtgp_dmu = 0.0 + nsigma_psvtgp_dmu*1.250e-02
```

```
endsection PARAMUSER
```

```
section CORREL
```

```
parameters nsvtgp_dx1 = polygp_cdvar
```

```
parameters psvtgp_dx1 = polygp_cdvar
```

```
parameters nsvtgp_drshgu = npoly_drshu
```

```
parameters psvtgp_drshgu = ppoly_drshu ...
```

```
endsection CORREL
```

```
section MODEL
```

```
subck nsvtgp D G S B ...
```

```
parameters dmu = nsvtgp_dmu
```

```
parameters x1 = -1.5e-08+nsvtgp_dx1
```

```
parameters rshgu = 150+nsvtgp_drshgu ...
```

```
ends nsvtgp
```

```
subck psvt D G S B ...
```

```
parameters dmu = psvtgp_dmu
```

```
parameters x1 = -1.5e-08+psvtgp_dx1
```

```
parameters rshgu = 710+psvtgp_drshgu ...
```

```
ends psvt
```

```
endsection MODEL
```

```
endlibrary svtgp
```

Пример файла rpolyp.scs, который определяет углы USER для семейства rpolyp (только одно определенное пользователем отклонение (user-defined deviation))

```
library rpolyp
```

```
section USER
```

```
parameters nsigma_rpolyp_ddw = 1*min(3.0,max(3.0, rpolyp_user))
```

```
...
```

```
include "rpolyp.scs" section=PARAMUSER
```

```
include "rpolyp.scs" section=CORREL
```

```
include "rpolyp.scs" section=MODEL
```

```
endsection USER
```

```
section PARAMUSER
```

```
parameters rpolyp_ddw = 0.0 + nsigma_rpolyp_ddw * 4e-9 ...
```

```
endsection PARAMUSER
```

## section CORREL

parameters **rpolyp\_drho** = **ppoly\_drshu** ...

## endsection CORREL

## section MODEL

subck rpporpo plus minus b ...

param rho = 320 + **rpolyp\_drho**

param dw = 39e-9+**rpolyp\_ddw** ...

ends rpporpo

## endsection MODEL

## endlibrary rpolyp

**2.3.9 Common Library файлы, поддерживающие определённые пользователем, полностью коррелированные худшие случаи (User-Defined Totally Correlated Worst-cases).**

Некоторые из параметров, определяющие процесс корреляции (process correlations) (например *polygp\_cdvar*, *npoly\_drshu* и *ppoly\_drshu* для транзисторов), могут быть не определены в Device Library Files. Если эти параметры встречаются более чем в одном типе приборов, то они группируются в так называемые **Common Library Files**.

<LIBRARY\_PATH>/common\_poly.scs

<LIBRARY\_PATH>/common\_go1.scs ...

Common Library файл имеет схожую структуру с Device Library файлом, но

- в нём не определены никакие модели (отсутствует MODEL section)
- корреляция не обязательна (CORREL address может быть пустым)

Пример **Common Library** файла *common\_poly.scs*:

## library common\_poly

## section USER

parameters **nsigma\_polygp\_cdvar** = 1\*min(3.0,max(-3.0,**poly\_cd\_user**))

parameters **nsigma\_npoly\_drshu** = 1\*min(3.0,max(-3.0,**npoly\_drshu\_user**))

parameters **nsigma\_ppoly\_drshu** = 1\*min(3.0,max(-3.0,**ppoly\_drshu\_user**))

...

include "common\_poly.scs" PARAM

include "common\_poly.scs" CORRELCORNER

## endsection USER

## section PARAMUSER

parameters **polygp\_cdvar**= 0.0 + **nsigma\_polygp\_cdvar**\*3.000e-09

```
parameters npoly_drshu = 0 + nsigma_npoly_drshu * 16.67
parameters ppoly_drshu = 0 + nsigma_ppoly_drshu * 33.333 ...
endsection PARAMUSER
section CORREL
endsection CORREL
endlibrary common_poly
```

Отметим, что статистические модели: svtpg транзисторы и rpolyp резисторы коррелируются вместе через параметр **ppoly\_drshu**. В svtpg\_CORREL секции мы получаем

```
parameters psvtpg_drshgu = `ppoly_drshu`
```

в rpolyp\_CORREL секции мы получаем

```
parameters rpolyp_drho =ppoly_drshu
```

### 2.3.10 Модельный файл

Интерфейс *Setup Corners* будет генерить модельный файл, называемый corner.scs.

Пример модельного файла для spectre, полученного с помощью интерфейса:

```
include "<simRunDir>/paramuser.scs"
include "<LIBRARY_PATH>/common_poly.scs" section=USER
include "<LIBRARY_PATH>/svtpg.scs" section=USER
include "<LIBRARY_PATH>/rpoly.scs" section=typ
...
include <LIBRARY_PATH>/veriloga.scs section=typ
```

Отметим, что важен порядок *include* выражений в модельном файле: ссылка на Common Library файлы предшествует Device Library файлам.

## 2.4 Выбор углов моделирования в интерактивном режиме

1. В окне моделирования ADE выберите *Tools->Setup Corners*. Вам откроется следующее окно (рис. 2.4).

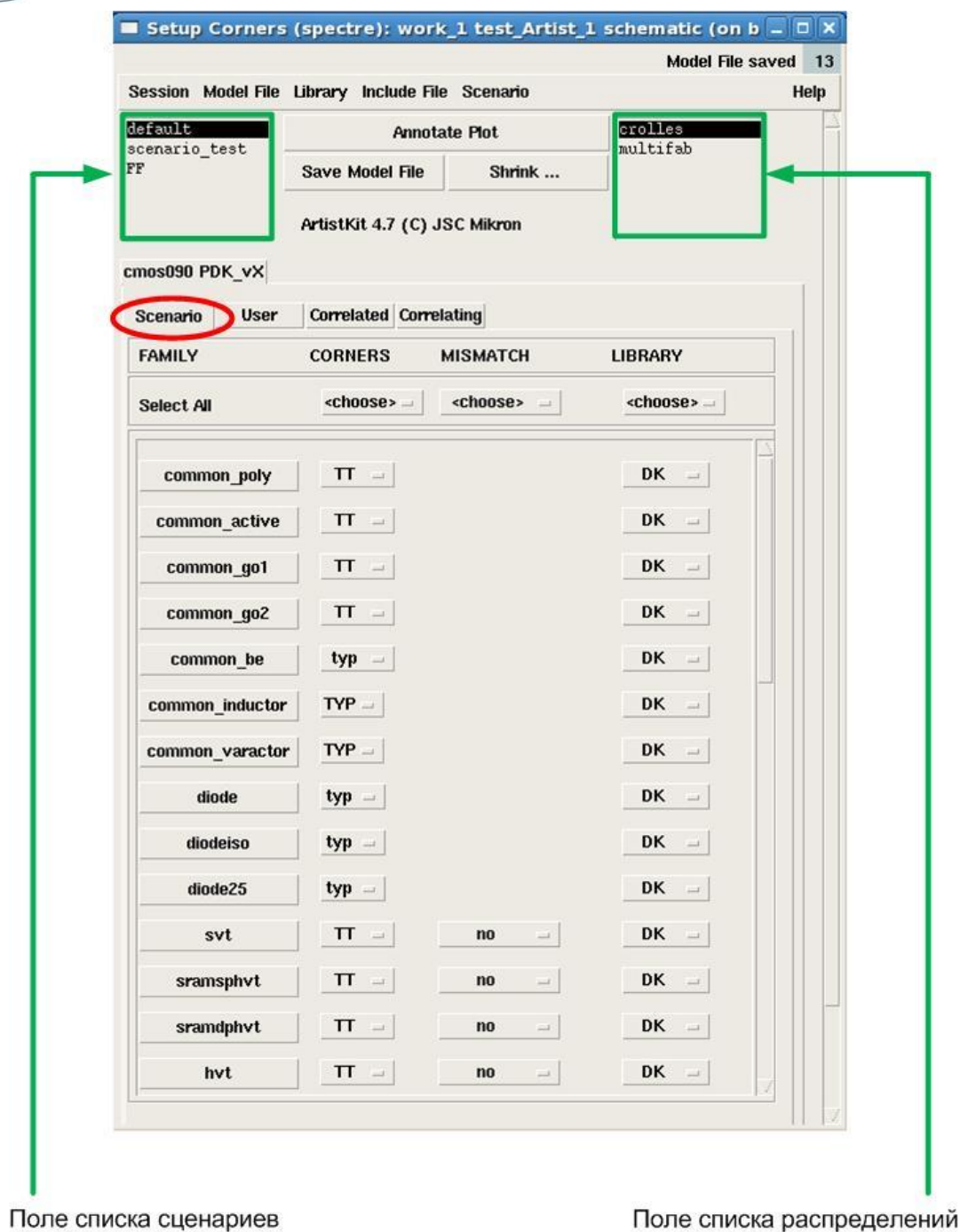


Рис. 2.4 Окно Setup Corners.

В этом окне располагается четыре вкладки:

- Во вкладке *Scenario* (рис. 2.4), для каждого компонента семейства *FAMILY* доступны углы моделирования *CORNERS* и библиотека *LIBRARY* (раздел 2.11), которая содержит *Library Files*. В некоторых семействах может присутствовать меню *MISMATCH* (см. главу 3). Если вы открыли *Setup Corners* впервые, то в списке сценариев выбран *defaults*. При сценарии *defaults*:

- все углы моделирования установлены в значение *typ* или *TT*.



- для всех семейств *MISMATCH* установлен в значение *no*.
- для всех семейств выбрана библиотека (*LIBRARY*) *DK(DesignKit)*.

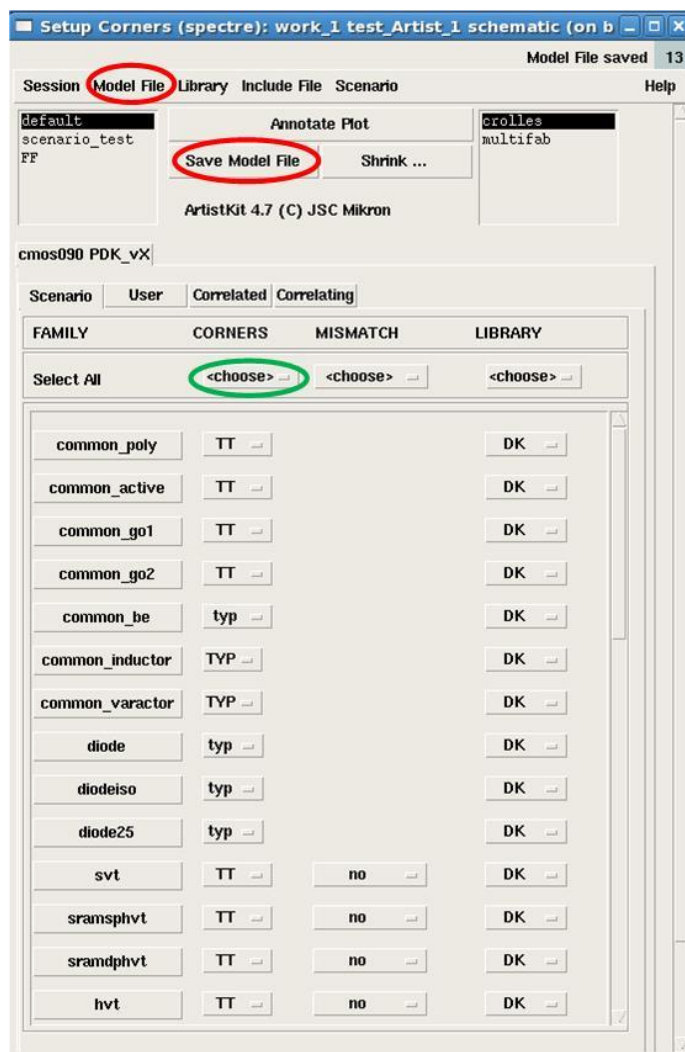


Рис. 2.5 Окно *Setup Corners* (сохранение модельного файла).

*Setup Corners* также позволяет определить пользовательские (*User-Defined*) библиотеки. Если пользовательские библиотеки уже определены, они отображаются в падающем списке *FAMILY* и *CORNERS*.

Падающее меню *<choose>* позволяет задать одинаковые значения *CORNERS*, *MISMATCH*, *LIBRARY* для всех семейств (рис. 2.5).

- Во вкладке *User* (рис. 2.6), для каждого компонента *FAMILY* существуют параметры процесса, которые могут меняться для создания специальных случаев моделирования углов, определяемых пользователем.

Редактирование параметров возможно только для тех семейств, у которых во вкладке *Scenario* выбран угол (*CORNERS*) *USER*.

Все семейства (*FAMILY*) и приборы (*DEVICES*) отображены с левой стороны окна. Выбрав необходимое семейство (например, svt на рис. 2.6) вы имеете возможность редактировать значение параметра отклонения *NSIGMA*. Также отображаются минимальное (*NSIGMA-MIN*) и максимальное (*NSIGMA-MAX*) значения отклонения (*NSIGMA*).

- Во вкладках *Correlated* и *Correlating* (рис. 2.7, 2.8), существуют два взаимодополняющих отображения корреляций между *DK Device Family* и *DK Common Family*.

Вкладка *Correlated* описывает корреляцию семейств приборов (*Device Family*) и показывает:

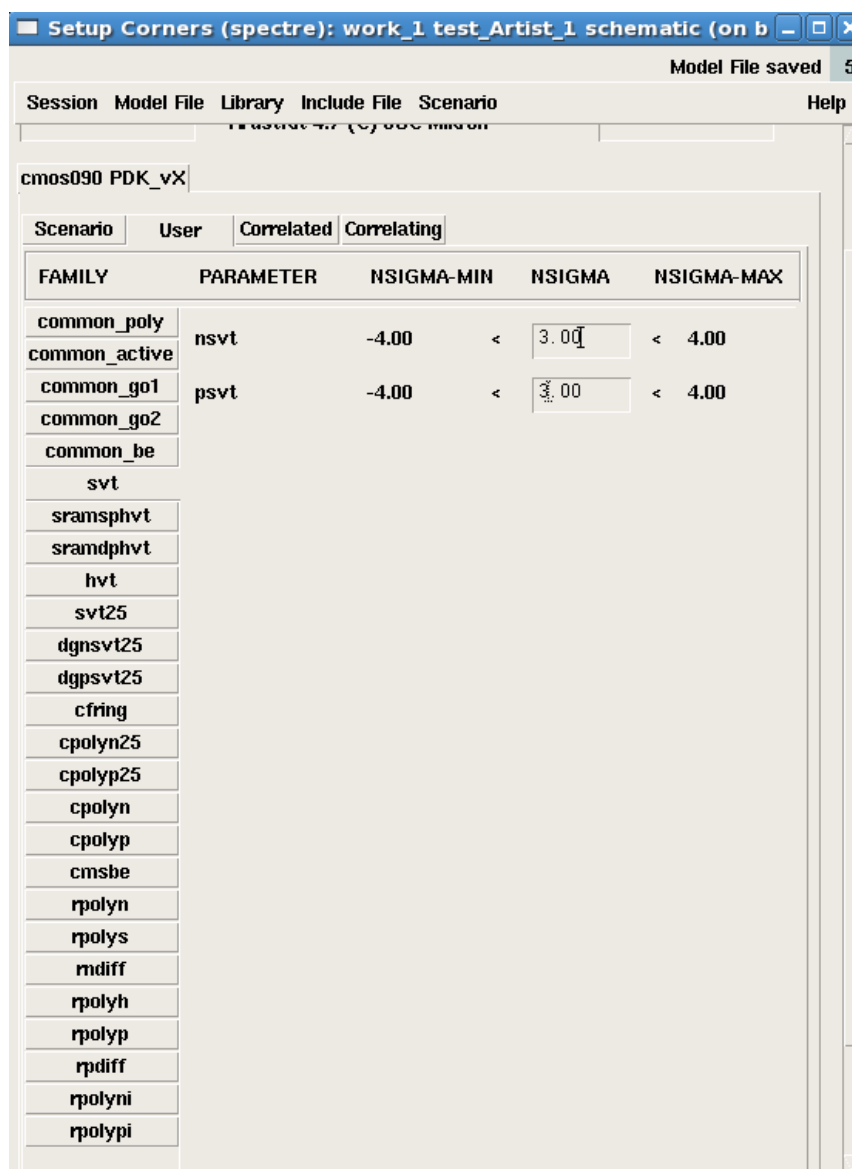


Рис. 2.6 Вкладка User.

- Коррелирует ли семейство приборов (*Device Family*) с каким либо общим семейством (*Common Family*) (yes/no)?

- Список общих семейств (*Common Family*), с которым коррелирует семейство приборов (*Device Family*). (Например, на рисунке 2.7 семейство приборов svt коррелирует с тремя общими семействами (*Common Family*): common\_active, common\_gol, common\_poly).
- Коррелируется ли семейство по углам (*On Corners(Pre-Defined Partially Correlated Worst-Cases)*)(yes/no)?
- Коррелируется ли семейство по углам *On stat&user corners* (статистические модели (*statistical model*) и модели, определённые пользователем (*User-Defined Totally Correlated Worst-Cases*)(yes/no)?

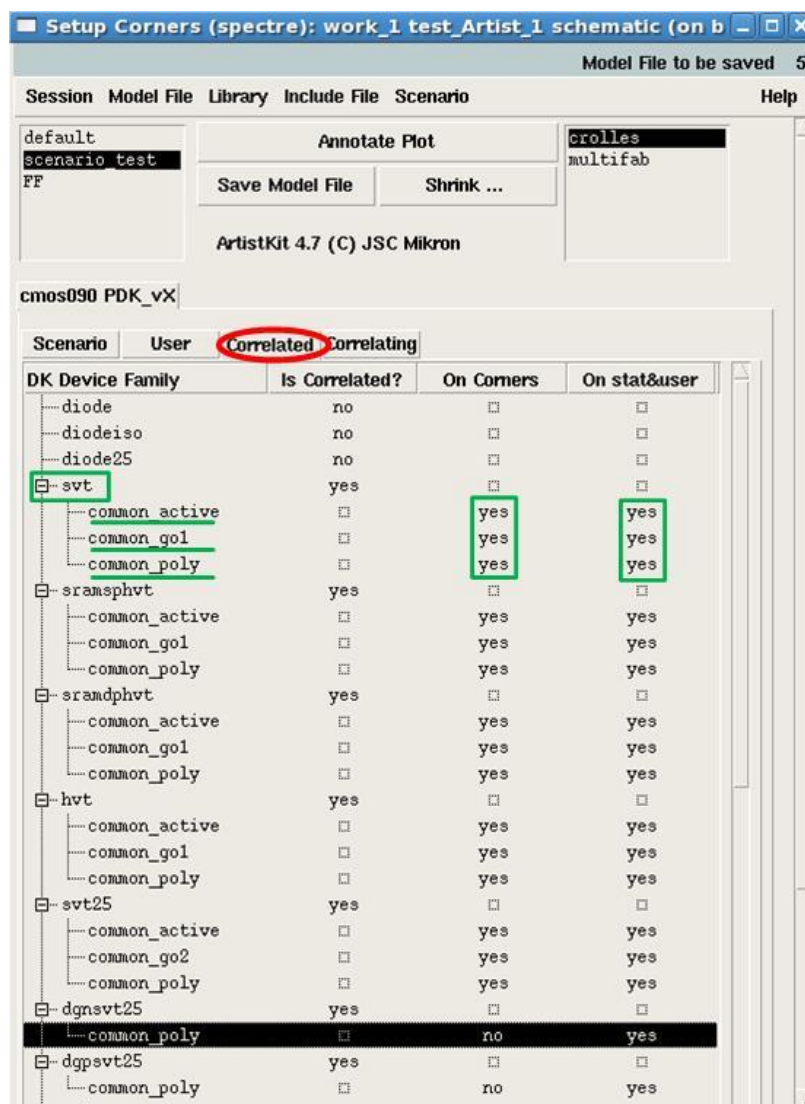


Рис. 2.7 Вкладка Correlated.

Вкладка *Correlating* показывает корреляцию семейств приборов (*Device Family*) и показывает:

- Список семейств приборов (*Device Family*), с которыми коррелирует общее семейство (*Common Family*). (Например, на рисунке 2.8 семейство *common\_be* коррелирует с двумя семействами приборов (*Device Family*): *cfring*, *cmsbe*).

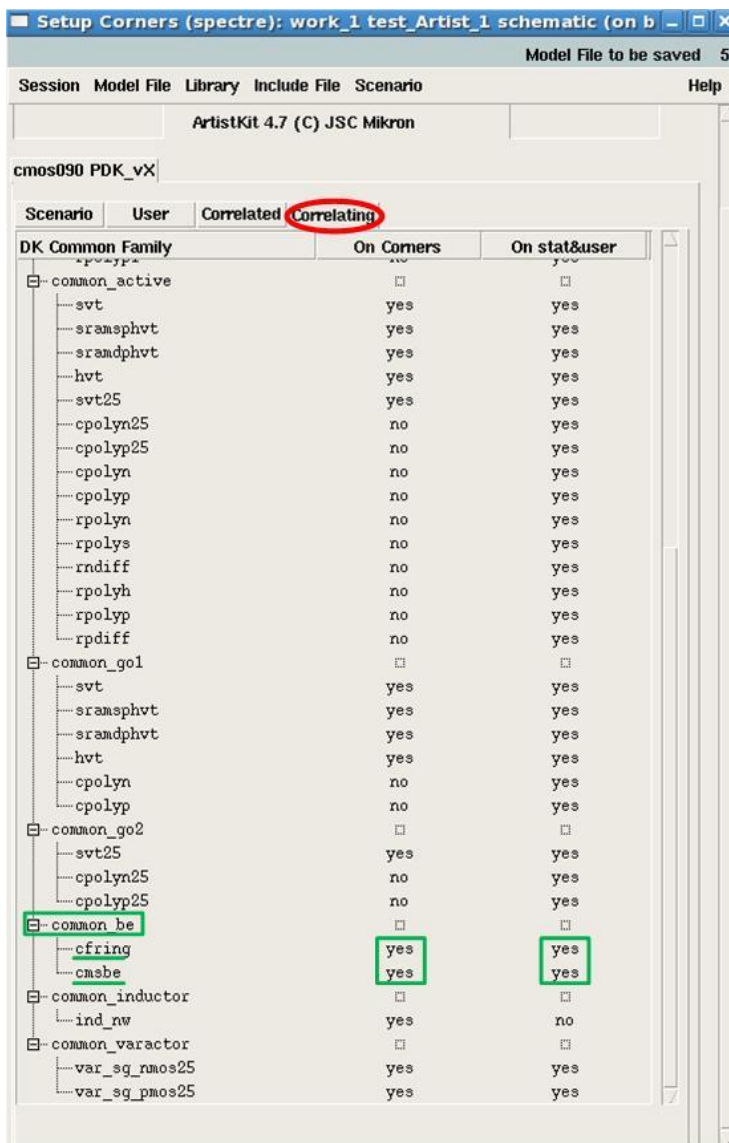


Рис. 2.8 Вкладка *Correlating*.

- Коррелируется ли семейство по углам *On Corners*(yes/no)?
- Коррелируется ли семейство по углам *On stat&user corners*(yes/no)?

## 2.5 Управление сценариями

Сценарий определяют несколько составляющих, таких как заданный набор углов моделирования (*CORNERS*), *MISMATCH* и библиотеки *LIBRARY*. Сценарий с параметрами по умолчанию (default) доступен, когда впервые вызывается окно *Setup Corners*.

Доступные сценарии отображаются в окне *Setup Corners* в поле списка сценариев. При выборе одного сценария из списка для каждого семейства *FAMILY* обновляются поля *CORNERS*, *MISMATCH* и *LIBRARY*.

### Создание нового Scenario

- Выберите необходимые углы (*CORNERS*) для всех семейств.
- Выберите необходимые значения параметра *MISMATCH*.
- Выберите необходимую библиотеку *LIBRARY*.
- Выберите меню *Scenario->New...* в появившемся окне (рис. 2.9)
- Введите оригинальное имя нового сценария и нажмите *OK*. Новый сценарий появился в поле списка сценариев. Если вы выберете любой другой сценарий из списка сценариев, а затем выберете ваш новый сценарий, то вы увидите, что введенные вами *CORNERS*, *MISMATCH* и *LIBRARY* сохранились.



Рис. 2.9 Окно создания нового сценария.

### Изменение сценария

- Выберите сценарий в поле списка сценариев.
- Выберите необходимые *CORNERS*, *MISMATCH* и *LIBRARY*.
- Выберите меню *Scenario->Change*. Выбранный сценарий сохранил введенные изменения.

### Удаление сценария

- Выберите сценарий в поле списка сценариев.
- Выберите меню *Scenario->Delete*.
- Подтвердите удаление сценария, нажав кнопку *OK* в появившемся окне. Удаленный сценарий исчезнет из списка сценариев.

### Отображение сценария

- Выберите сценарий в поле списка сценариев.
- Выберите меню *Scenario->Print*. Появившееся окно содержит информацию о сценарии: углы, значение отклонения (NSIGMA) и используемую библиотеку для каждого семейства. Если оставить окно открытым и снова нажать *Scenario->Print*, то информация о сценарии добавится в текстовое окно.
- Выберите в появившемся меню *File->Save As...* Введите имя файла. Это позволит сохранить сценарий в файле. Далее вы сможете загрузить этот файл с помощью меню *Scenario->Load...*(описано ниже).

### Загрузка сценария

- Выберите меню *Scenario->Load....*
- Введите имя файла сценария. Файл должен генериться с помощью меню *Scenario->Print* или должен быть определен в похожем формате.
- Нажмите *OK*. Загруженный сценарий появился в поле списка сценариев и содержит в себе загруженные *CORNERS*, *MISMATCH*, *LIBRARY* и значения отклонения *NSIGMA*.

## 2.6 Сохранение состояния установленных углов (Setup Corners State)

Сохранение состояния установленных углов (*Setup Corners State*) заключается в переводе данных об углах в файл состояния, также как в ADE. Это позволяет этим данным быть загруженными с любой другой сессии *Setup Corners*. Файл состояния может быть загружен для другой ячейки и/или для другого симулятора.

- Выберите команду меню *Session -> Save State....*
- В появившемся окне (рис. 2.10) введите имя состояния или выберите из списка уже существующих состояний (*Existing States*).
- Нажмите *OK*.

Данные сессии сохранились в структуре директории сохранения ADE:

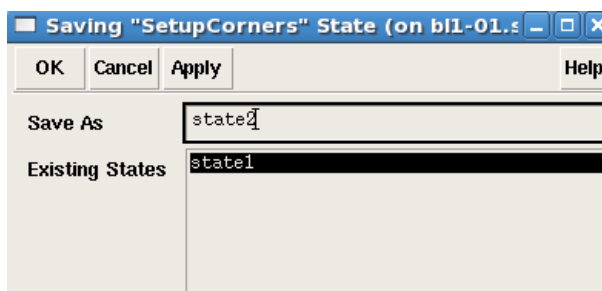


Рис. 2.10 Форма сохранения состояния установленных углов (*Setup Corners State*).

<state\_save\_dir>/<libName>/<cellName>/<simulator>/

.setupCornerScenario/<stateName>

где

<state\_save\_dir> это директория сохранения состояния ADE.

<libName> это название библиотеки проекта в окне ADE.

<cellName> это название ячейки проекта в окне ADE.

<simulator> это название симулятора в окне ADE .

<stateName> это название режима сессии *Setup Corners*.

## 2.7 Загрузка состояния установленных углов (Setup Corners State)

Загрузка состояния установленных углов ( *Setup Corners State*) заключается в обновлении окна *Setup Corners* с помощью данных, хранящихся в файле состояния. Состояние установленных углов (*Setup Corners State*) создается путем выбора команды *Session->Save State...*, как было описано выше.

- Выберите команду *Session -> Load State...* Загрузка состояния заменит значения текущего состояния теми значениями, которые определены в выбранном состоянии: поэтому для Вас появляется подсказка диалогового окна об итоговом сохранении текущего состояния до того, как будет загружено следующее состояние.
- Выберите *Да (yes)* или *Нет (no)* в диалоговом окне. Если Вы выбрали *Да*, то пожалуйста перейдите к разделу 2.6.
- В появившейся форме (рис. 2.11) выберите поля *Library*, *Cell* и *Simulator*. Доступные состояния для *Library*, *Cell* и *Simulator* отражены в поле списка *State Name*.



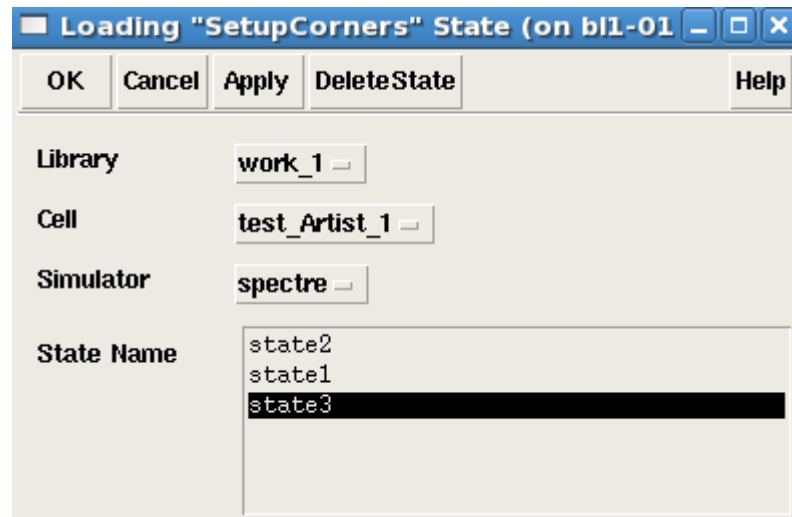


Рис. 2.11 Форма загрузки состояния установленных углов (Setup Corners State).

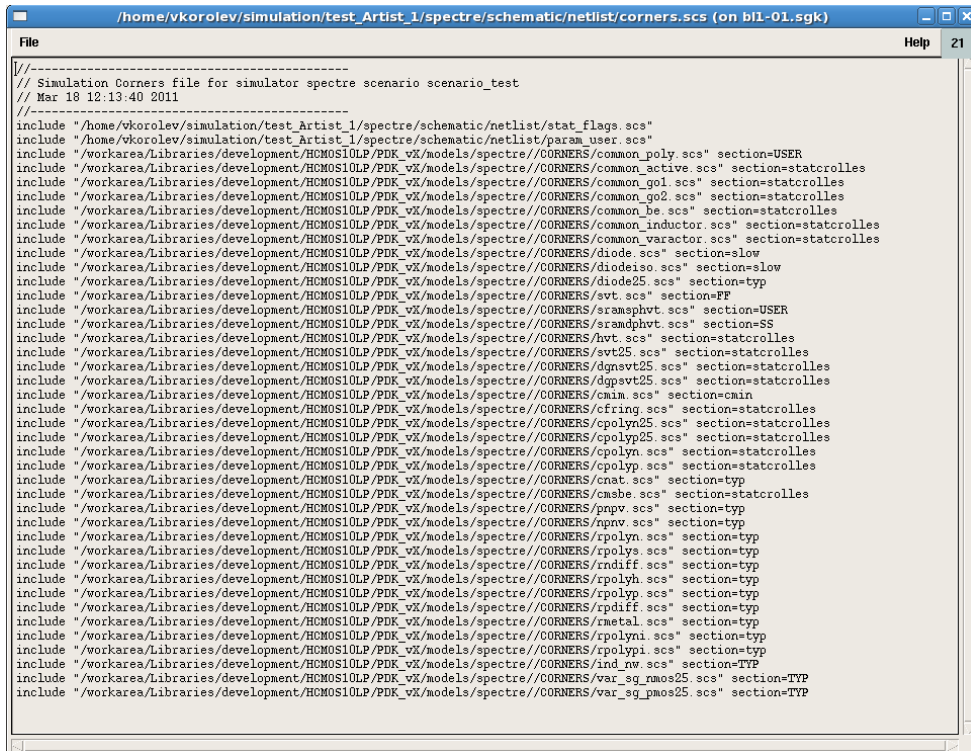
- Выберите название состояния *State Name*, которое Вы желаете загрузить. Существует возможность удалить выбранное состояние нажатием кнопки *DeleteState*.
- Нажмите *OK*. Окно *Setup Corners* обновлено.

Примечание: Следует учесть, что могут появиться некоторые предупреждения, если имеются несоответствия между загружаемым Вами состоянием и текущей сессией *Setup Corners* (несоответствия в названии семейств, углов, технологии, несоответствия между моделями пользователя и т.д. ).

## 2.8 Запуск моделирования

- Выберите один сценарий в поле списка сценариев. Окно *Setup Corners* обновляется соответствующими данными.
- Если вы открыли *Setup Corners* впервые, то модельный файл отсутствует. Необходимо сохранить выбранные углы моделирования, нажав кнопку “*Save Model File*” в верхней части окна, либо выбрав пункт меню *Model File->Save* (рис. 2.4).  
После сохранения углов моделирования в рабочей директории проекта (Project Directory) формируется файл “*corners.scs*” (для *spectre*) или “*corners*” (для остальных симуляторов), содержащий подключение модельных файлов в соответствии с выбранными углами моделирования (рис. 2.12).





```

File /home/vkorolev/simulation/test_Artist_1/spectre/schematic/netlist/corners.scs (on bl1-01.sgk) Help 21
//-----
// Simulation Corners file for simulator spectre scenario scenario_test
// Mar 18 12:13:40 2011
//-----
include "/home/vkorolev/simulation/test_Artist_1/spectre/schematic/netlist/stat_flags.scs"
include "/home/vkorolev/simulation/test_Artist_1/spectre/schematic/netlist/param_user.scs"
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/common_poly.scs" section=USER
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/common_active.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/common_g01.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/common_g02.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/common_be.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/common_inductor.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/common_varactor.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/diode.scs" section=slow
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/diodeiso.scs" section=slow
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/diode25.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/svt.scs" section=FF
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/sramphvt.scs" section=USER
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/srandphvt.scs" section=SS
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/hvt.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/svt25.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/dgnsvt25.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/dgnsvt25.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/dgnsvt25.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/cmk.scs" section=cmkn
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/cfring.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/cpolyn25.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/cpolyn.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/cpolyp.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/cnat.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/casbe.scs" section=staticrolles
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/pnpv.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/pnpv.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/rpolyn.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/rpolys.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/rndiff.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/rpolys.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/rpolyp.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/rpdif.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/rmetal.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/rpolyni.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/rpolypi.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/ind_rw.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/var_sg_rmos25.scs" section=typ
include "/workarea/Libraries/development/HCMOS10LP/PDK_vx/models/spectre/CORNERS/var_sg_rmos25.scs" section=typ

```

Рис. 2.12 Содержимое файла "corners.scs".

Другой файл под названием *models* управляет содержащимся в нем файлом *corners* и выбираемым пользовательским файлом, описанным ниже ( раздел 2.12, «Добавление пользовательского файла»). Файл *models* также используется для ввода в действие некоторых стандартных опций и параметров симулятора, которые скрыты от пользователя благодаря содержащемуся в файле *models* файлу *hiddenOptions*. Заметьте, что для *spectre* все имена файлов имеют расширение .scs.

Файл *models* для *eldo*, *eldoD* и *hspiceD/hspiceS*:

```

.include hiddenOptions

.include corners

.include setupCornersIncludeFile

```

Файл *models.scs* для *spectre*:

```

include "hiddenOptions.scs"

include "corners.scs"

include "setupCornersIncludeFile.scs"

```

Этот файл *models* предписан к использованию в ADE в качестве составляющей списка соединений. Для проверки этого утверждения рассмотрим два случая:

- ✓ работая с методом *Spice Socket* (например, для *eldo*), выберите команду *Setup ->Environment*, и в поле *Include File* подключается файл *models* (рис.2.13).
- ✓ работая с методом прямого нетлистования (*Direct Netlisting*) (например, для *spectre*) выберите команду *Setup->Simulation Files*, и в поле *Definition Files* подключается файл *models.scs* (рис.2.14).

Выбор команды меню *Model File -> None* возвращает состояние *Corners* к состоянию *No Corners*, означающее, что *Model File* удален и в списке соединений отсутствует какая-либо модель.

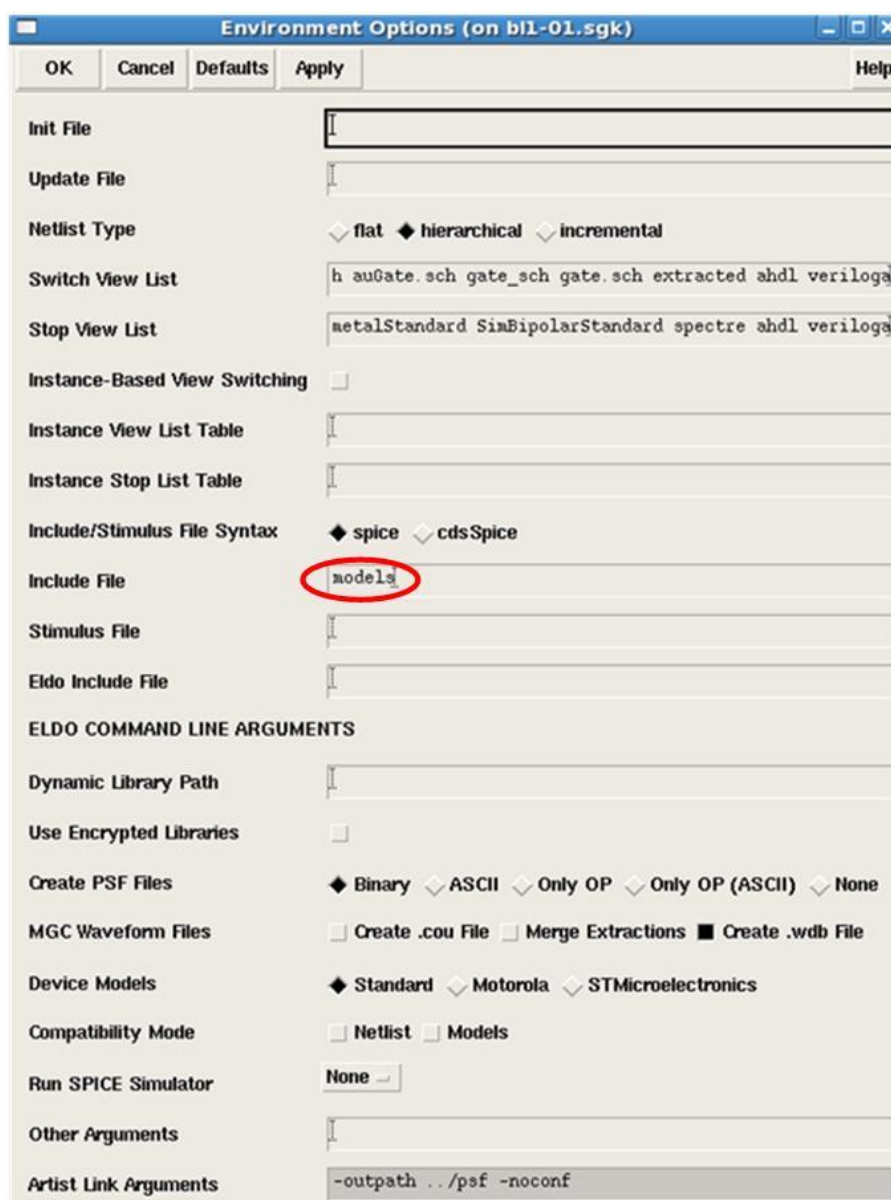


Рис. 2.13 Подключение файла "models".



Рис. 2.14 Подключение файла “models.scs”.

- Выберите команду меню *Model File - > Show. Model File*, создаваемый командой *Model File -> Save*, отображается в окне просмотра файлов (рис. 2.12)
- Настройте и запустите моделирование из окна ADE.

Используя формы и меню *Analog Artist*, задайте условия своего моделирования (переменные проекта, температура, входные сигналы, тип анализа и т.д.) и список соединений и промоделируйте вашу схему.

Вы можете использовать команду меню *Simulation->Run* или эквивалентный значок.

- Закройте окно, выбрав команду меню *Model File -> Close*.

Текущее состояние выбранных углов моделирования сохранено в директории запуска моделирования в файл под названием *corner\_map*. Этот *map*-файл вызывается при запуске новой рабочей сессии.

## 2.9 Создание комментариев в окне waveform

Окно *Setup Corners* позволяет Вам с легкостью делать комментарии на графиках, содержащие некоторую информацию об условиях моделирования. Вы можете подписывать температуру, переменные проекта, имя текущего сценария и углы моделирования.

- Выберите углы моделирования, как было описано в разделе 2.4, оставляя открытым окно *Setup Corners*.

- Настройте и запустите моделирование. Используя формы и меню Analog Artist, создайте условия своего моделирования ( переменные проекта, температура, входной сигнал, анализ,..), список соединений и промоделируйте вашу схему.
- Выведите результаты моделирования в окно изменения амплитуды сигнала во времени в Analog Artist. Для этого используйте команды меню Analog Artist *Outputs* и *Results* -> *Plot Outputs*.
- Нажмите кнопку *Annotate Plot* в окне *Setup Corners* (рис.2.15). Появится окно со списком (рис.2.16), позволяющее выбрать один или несколько пунктов среди температуры, переменных проекта, сценария и углов.

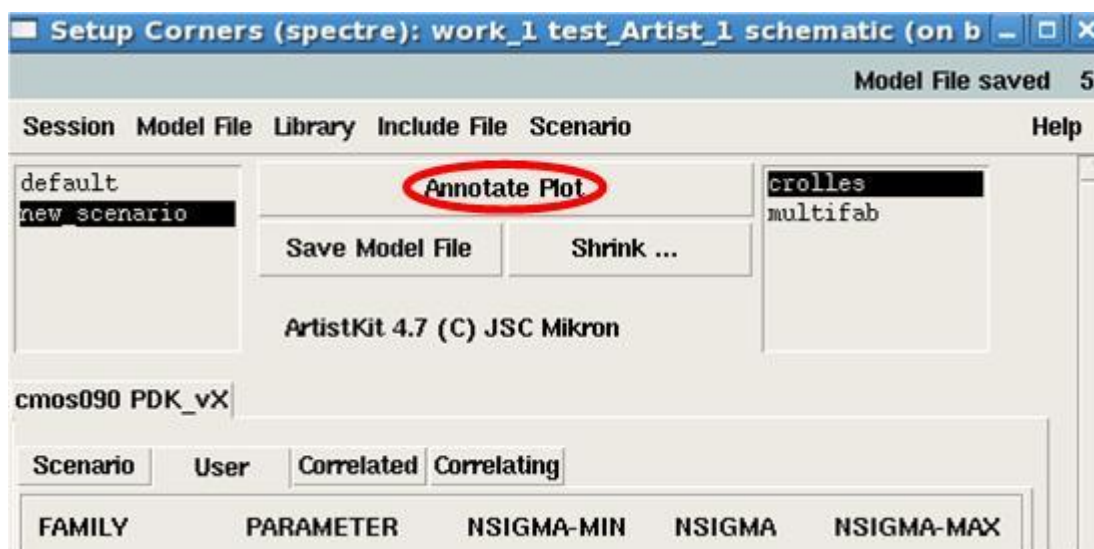


Рис. 2.15 Создание комментариев к окну waveform .

- Выберите один или несколько пунктов в окне со списком и нажмите *OK*.

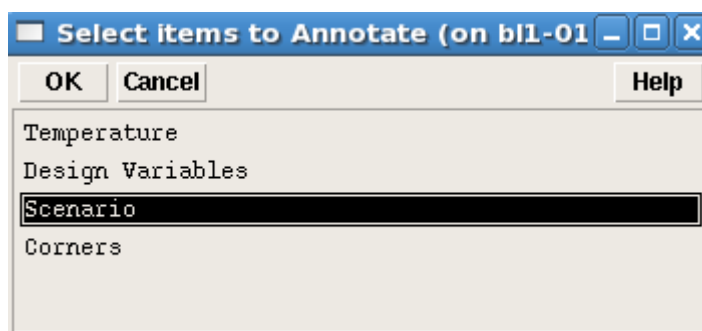


Рис. 2.16 Окно выбора комментариев к окну waveform .

Текстовые комментарии отображаются в текущем окне waveform. Вы можете редактировать и изменять эти комментарии (содержание, ориентация, шрифт, размер,

цвет...) с помощью команды меню *Annotation->Edit* (для *AWD*) в окне *waveform*. Также Вы можете перемещать эти комментарии по данному окну.

## 2.10 Управление библиотеками параметров моделей.

Библиотека моделей (*Library*) определяется с помощью имени и пути в окне *Setup Corners*. Имена библиотек возможно выбрать, используя циклическое меню *LIBRARY* каждого семейства *FAMILY*. Их пути используются при генерации *Model File* нажатием кнопки *Save Model File*.

При вызове окна *Setup Corners* впервые, две библиотеки *Libraries* уже предопределены:

- *DK*: эта библиотека является Design Kit и содержит все файлы для поддерживаемых симуляторов, как было сказано в разделе 2.4. Путь к этой библиотеке ищется по умолчанию в текущем загруженном Design Kit. Вы можете переписать этот путь, определив переменную в Вашем файле *.cdsinit*:

-> *UARTeldoRoot, UARTeldoDRoot*

-> *UARTspectreRoot*

-> *UARThspiceSRoot, UARThspiceDRoot*

переопределяя библиотеку *DK* соответственно для симуляторов *eldo*, *eldoD*, *spectre*, *hspiceS* и *hspiceD*. Обратите внимание, что при загрузке нескольких Design Kits эти переменные должны иметь несколько путей.

- *UserLib*: эта библиотека предназначена для того, чтобы дать Вам возможность описывать Ваши собственные моделируемые параметры, когда по какой-либо причине они могут отличаться от текущих значений Design Kit. Путь к этой библиотеке определен по умолчанию как *~/models*. Вы можете переписать этот путь, определив переменную *UARTCornerUserLibPath* в Вашем файле *.cdsinit*.

Таким образом, Вы можете убедиться, что для каждого меню *LIBRARY* начальными возможными вариантами являются *DK* и *UserLib*, где *DK* установлен по умолчанию.

Существует возможность определять дополнительные библиотеки для средств проектирования. Для понимания предназначения каждой библиотеки смотрите

документацию к Вашим средствам проектирования (например, передача непрерывных и дискретных моделей в DK в другую библиотеку (*Library*)).

Также есть возможность определить Ваши собственные библиотеки, их имена и пути, определяя следующие переменные (в Вашем файле *.cdsinit* )

– *UARTeldoCornerLibrary* = '((*t\_name t\_path*) (*t\_name t\_path*) ...)

– *UARTeldoDCornerLibrary* = '((*t\_name t\_path*) ...)

– *UARTspectreCornerLibrary* = '((*t\_name t\_path*) ...)

– *UARThspiceSCornerLibrary* = '((*t\_name t\_path*) ...)

соответственно для симуляторов *eldo*, *eldoD*, *spectre* и *hspiceS*. Например:

*UARTeldoCornerLibrary*='("WL" "/users/ne1/0.13u/eldo/wl")')

Для каждого компонента *FAMILY* Вы можете вывести содержание текущей библиотеки *LIBRARY* путем нажатия кнопки соответствующего названия семейства. Файлы библиотек, как было сказано в разделе 2.4, отображаются в окне просмотра файлов.

Конечно же, существует возможность управлять библиотеками, используя интерфейс меню *Library* в окне *Setup Corners*.

### 2.10.1 Создание новой библиотеки.

- Выберите команду меню *Library* -> *New* (рис.2.17).

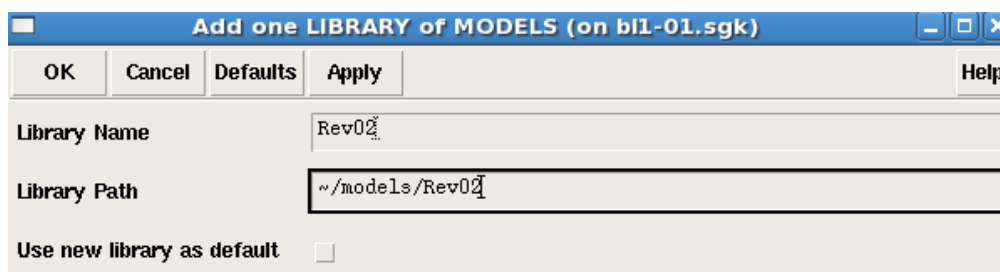


Рис. 2.17 Окно создания новой библиотеки.

Поле *Library Name* определяет имя новой библиотеки, которая вызывается в окне *Setup Corners*.

Поле *Library Path* определяет полный путь к новой библиотеке. Этот путь должен быть действительным, иначе библиотека не будет зарегистрирована.

Поле *Use new library as default* определяет, будет ли новая библиотека использоваться по умолчанию и заменит библиотеку DK или нет.

- Заполните поля и нажмите *OK*.

Все меню *LIBRARY* обновлены: добавлен еще один возможный альтернативный вариант и в случае определения новой библиотеки по умолчанию текущее значение меню *LIBRARY* устанавливается в соответствии с новым именем библиотеки.

### 2.10.2 Изменение пути библиотеки.

- Выберите команду меню *Library -> Change* (рис. 2.18).

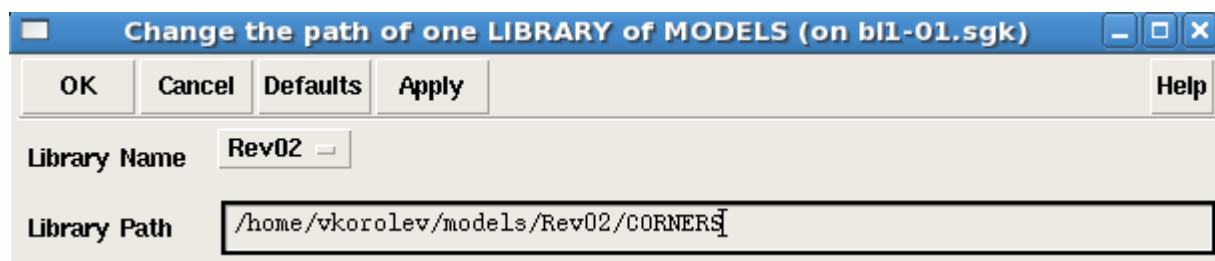


Рис. 2.18 Окно изменения пути библиотек.

Поле *Library Name* определяет имя библиотеки, чей путь необходимо изменить. Только те библиотеки, которые уже определены, могут быть выбраны здесь через меню.

Поле *Library Path* определяет путь к библиотеке. Этот путь должен быть действительным, иначе путь к библиотеке не будет изменен.

- Выберите Вашу библиотеку, введите новый путь и нажмите *OK*. Путь к выбранной библиотеке будет обновлен.

### 2.10.3 Удаление библиотеки

- Выберите команду меню *Library -> Delete* (рис. 2.19).



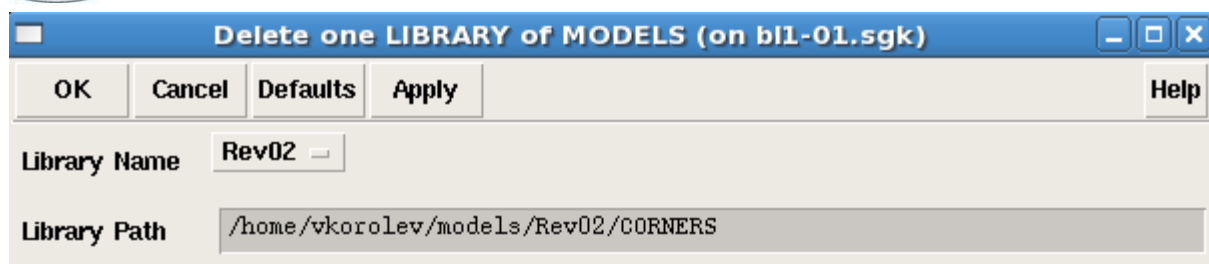


Рис. 2.19 Окно изменения пути библиотек.

Поле *Library Name* определяет имя той библиотеки, которую необходимо удалить. Только те библиотеки, которые уже определены, могут быть выбраны здесь через меню.

Поле *Library Path* определяет путь к текущей библиотеке, выбранной в поле *Library Name*. Здесь путь указан только для информации и не может быть изменен.

- Выберите имя Вашей библиотеки и нажмите *OK*.

Выбранное имя библиотеки удаляется из возможных альтернативных вариантов меню *LIBRARY*. Когда оно было текущим значением, оно заменялось библиотекой по умолчанию. Заметьте, что библиотеки *DK* и *UserLib* не могут быть удалены.

## 2.11 Добавление пользовательского файла

По какой либо причине может возникнуть необходимость включения в список соединений некоторых команд моделирования, которые в полной мере не поддерживаются продуктом ADE. Файл *models* подключает файл под названием *setupCornersIncludeFile* (раздел 2.4 «Выбор углов моделирования в интерактивном режиме»). В данном файле есть возможность подключения пользовательского файла непосредственно из окна *Setup Corners*.

### 2.11.1 Определение и редактирование пользовательского файла.

Выберите команду меню *Include File -> Edit* (рис.2.20).





Рис. 2.20 Окно редактирования пользовательского файла.

Поле *Include File Name* определяет имя файла, который Вы хотите включить в список соединений симулятора. Этот файл предполагается размещать в текущей директории моделирования.

Клавиша *Edit* позволяет редактировать файл включения с помощью текущего редактора файлов.

- Определите имя Вашего *Include File*, отредактируйте и сохраните его, затем нажмите *OK*.

Таким образом, Вы определили Ваш *Include File*.

### 2.11.2 Команда *Include On*

По умолчанию ни один файл не включен в список соединений. Для того чтобы осуществить включение файла:

- Выберите команду меню *Include File -> On*  
Тогда команда меню *Include File -> On* становится неактивной, а команда меню *Include File -> Off* становится активной. Для файла включения команды меню *Off / On* действуют как переключатели режимов активный/неактивный.
- Нажмите кнопку *Save Model File*

Откройте файл *setupCornersIncludeFile* и вы увидите подключение вашего *Include* файла. Пример подключения пользовательского файла для *spectre*.

```
// User Defined spectre Include File  
  
// Mar 22 16:33:03 2011  
  
//-----  
  
include "userIncludeFile.scs"
```

### 2.11.3 Команда *Include Off*

- Выберите команду меню *Include File -> Off*

Тогда команда меню *Include File -> Off* становится неактивной, а команда меню *Include File -> On* становится активной.

- Нажмите кнопку *Save Model File*

В файле *setupCornersIncludeFile* не добавляется больше никаких указаний по подключению файла.

## 3 Настройка и запуск анализа Monte Carlo

### 3.1. Введение

Эта глава рассказывает о настройке статистических моделей и об анализе Монте-Карло, до начала моделирования с помощью Analog Design Environment (ADE) или Analog Artist .

Эта методология поддерживается средствами проектирования, которые предоставляет статистические модели. Способ выбора статистических моделей общий для симуляторов *eldo*, *eldoD* и *spectre*.

Статистические модели и анализ Монте-Карло не поддерживаются симуляторами *hspiceD* и *hspiceS*.

Определение анализа Монте-Карло специфично для каждого симулятора:

- с помощью *eldo* используется определенный интерфейс из ArtistKit. Заметьте, что с помощью *eldo* Вы можете использовать как интерфейс ArtistKit, так и определенное приложение от Mentor Graphics .
- с помощью *spectre* используется родное приложение ADE.

### 3.2 Методология

В данной методологии используется уникальный набор моделей для описания как традиционных худших случаев, так и статистических моделей, включающих эффекты рассогласования.

Эта методология описывалась в главе 2. Она имеет некоторые преимущества, поскольку обеспечивает:

- более реалистичный учёт предопределённых углов моделирования для частично коррелированных семейств (например,  $n$  и  $p$  МОП-транзисторы),
- учёт предопределённых статистических моделей для полностью коррелированных семейств (например,  $n$  и  $p$  МОП-транзисторы вместе с резисторами *poly*),
- возможность использовать учёт настраиваемых пользователем углов моделирования также и для полных корреляций между семействами,
- возможность моделирования эффектов рассогласования (*mismatch*) для каждого из этих углов.

Нижесказанное является продолжением главы 2, объясняющим, как описывать эффекты рассогласования с использованием этой новой методологии.

Эффекты рассогласования описываются самими моделями с помощью *MODEL* (см. главу 2), включая статистические, предопределяемые изменения, а также изменения, определяемые пользователем.

Выбор эффектов рассогласования может быть осуществлен:

- семейством через семейство, используя графический интерфейс *Setup Corners*
- элемент (*instance*) через элемент (*instance*), устанавливая параметр (*mismatch*) в *Schematic*.

Когда статистические изменения для эффектов рассогласования или статистических углов моделирования, как сказано в главе 2, выбраны, то они учитываются через анализ Монте-Карло, определяемый из:

- окна *Setup Corners* с *eldo* и *eldoD*.
- *MGC Sim->Tools->Monte-Carlo* анализа с *eldoD*.
- *Tools->Monte Carlo* команды со *spectre*.

### 3.3 Пример со Spectre

#### 3.3.1 Device Library файлы

Эффекты рассогласования описываются в разделе *<family>\_MODEL*, внутри *subckt* определений (эффекты рассогласования чувствительны к размерам устройства). Установка индикатора (*mflag parameters*) в *subckt* определении используется для выбора различных методологий рассогласования. Эти индикаторы связаны с одним индивидуальным глобальным параметром через семейство приборов, называемое *<family>\_dev*, чье значение устанавливается через окно *Setup Corners* (см. ниже). Допустимыми значениями для параметров *<family>\_dev* являются 0 и 1, обозначающие следующее:

- 0 – ни одного эффекта рассогласования не учтено.
- 1 – эффекты рассогласования (*mismatch effects*) моделируются посредством локальных статистических отклонений: во время моделирования необходим анализ Монте-Карло для получения эффекта от этих отклонений.

Отметим, что параметр элемента (*instance parameter*) *mismatch* может использоваться для выбора подмножества элементов, на которые воздействуют статистические отклонения.

Пример файла *svtgp.scs*, где описывается эффект рассогласования для *Vt*:

```
library svtgp

...

section MODEL

//mismatch parameters def.

parameters DVTH0NSVTGP=0

statistics {

mismatch {

vary DVTH0NSVTGP dist=gauss std=1

}

}

subckt nsvtgp d g s b

parameters mismatch=1 w=0.15u l=0.13u mult=1 ...

parameters dvth0 = nsvtgp_vth0

parameters mcor_vt = nsvtgp_mcor_vt ...

// -----

// flag use to switch on/off mismatch effects

// -----

// svtgp_dev no stat_var
```

```
// 0          yes    no

// 1          no     yes

+ mflag_mc = ((svtgp_dev==0) ? 1 : 0)

+ mflag_corner = ((svtgp_dev==1) ? 1 : 0)

// Equations for mismatch deviations:

// -----

+ dvth0_sig=(2.7e-9/sqrt(w*1))+0.0)/sqrt(2.0*mult)

+ dvth0_mm=(dvth0_sig*DVTH0NSVTGP*mflag_mc*mismatch)

+ dvthtot = dvth0/(((dvth0==0)? 1.0:0.0)

+sqrt(pow(dvth0,2)))*sqrt((pow(dvth0_sig*mcov_vt,2))*mflag_c
corner+pow(dvth0,2))

...

parameters vth = 0.312+dvthtot ...

parameters vth0 = vth+dvth0_mm*mflag_mc ...

model nsvtgp bsim4 type=n ... vth0=vth0 ...

ends nsvtgp ...

endsection MODEL

...

endlibrary svtgp
```

### 3.3.2 Model File

Этот файл создается в интерфейсе *Setup Corners*. Файл *stat\_flags* определяет индикаторы *<family>\_dev*, участвующие в моделировании эффектов рассогласования.

```
include "<NETLIST_DIR>/stat_flags.scs"
```

```
include "<NETLIST_DIR>/paramuser.scs"
```

```
include "<LIBPATH>/common_poly.scs" section=TT  
  
include "<LIBPATH>/svtgp.scs" section=statcrolles  
  
include "<LIBPATH>/rpolyp.scs" section=typ ...
```

Пример файла *stat\_flags*:

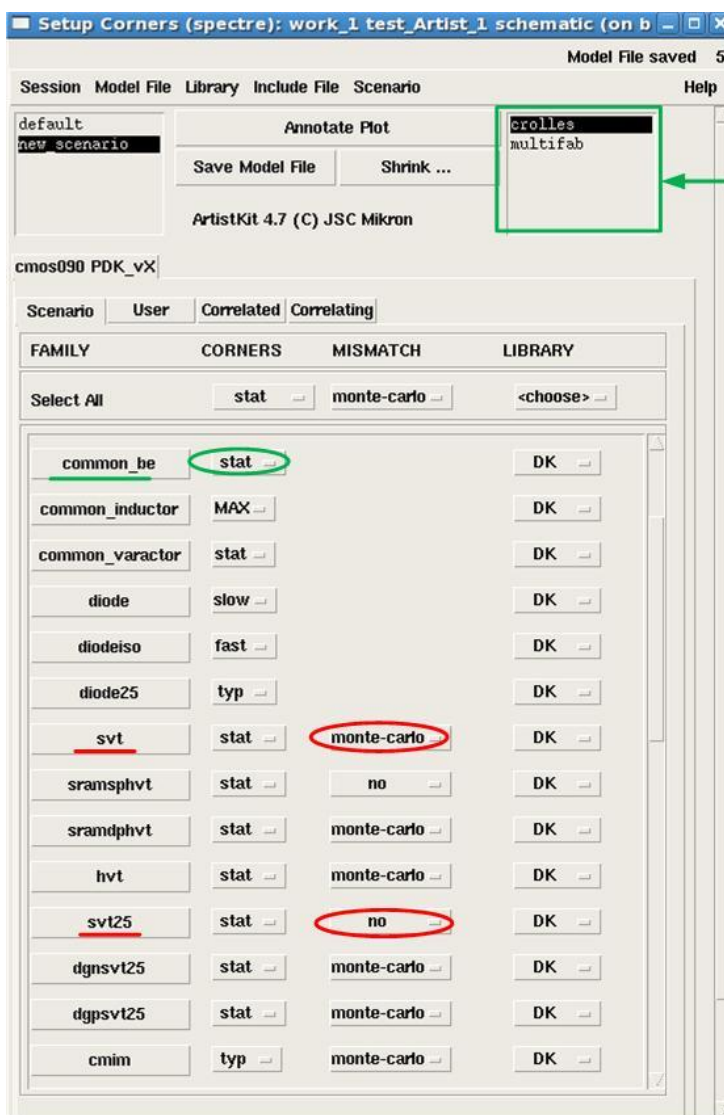
```
parameters svtgp_dev=1  
  
parameters rpolyp_dev=0 ...
```

### 3.4 Выбор статистических углов и эффектов рассогласования

- Выберите *Tools->Setup Corners* в окне симуляции ADE.

Обратите внимание, что в появившемся окне (рис. 3.1):

- некоторые семейства (device family) могут принимать значение углов - **stat**;
- как только **stat** углы будут выбраны, поле списка распределений становится значимым.
- некоторые семейства приборов имеют циклическое меню MISMATCH с вариантами выбора: *no, monte-carlo*.



Поле списка распределений  
(Distribution Lisbox Field)

Рис. 3.1 Выбор статистических углов и эффектов рассогласования.

- Выберите Ваши **stat** углы, статистические распределения и варианты рассогласований (mismatch variation)

Выбор эффектов рассогласования для семейств приборов влияет на все элементы прибора в Вашей схеме. Как было описано в предыдущих разделах, существует возможность моделировать эффекты рассогласования только в подмножествах этих элементов. В большинстве средств проектирования есть свойство, называемое рассогласование (mismatch) в виде symbol прибора, которое работает как индикатор со значениями 1 и 0 и позволяет включать эффекты рассогласования только на выбранных подмножествах элементов приборов.

- Нажмите кнопку *Save Model File*

Для Вашего симулятора создается *Model File*, как было описано в предыдущих разделах.

### 3.5 Запуск анализа Монте-Карло со spectre

- В окне *Setup Corners* выберите Ваши **stat** углы, статистические распределения, варианты рассогласований и сохраните *Model File*.

Примечание 1: При работе со *spectre* не существует клавиши *Monte-Carlo*

Примечание 2: Для некоторых средств проектирования вместо выбора рассогласований через указанные поля Вы можете использовать особое семейство под названием *mismatch*, чьи углы моделирования имеют значения *да* или *нет*.

- В окне моделирования ADE выберите *Tools -> Monte-Carlo*  
В появившемся окне анализа Монте-Карло (рис. 3.2) Вы можете:  
- определить свой анализ Монте-Карло;

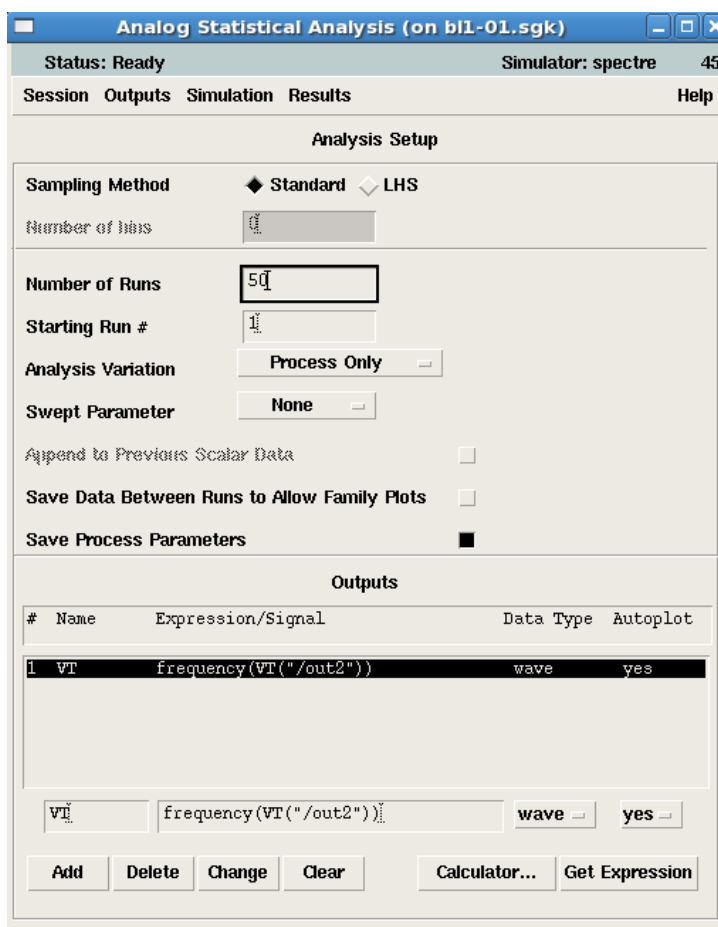


Рис. 3.2 Окно Монте-Карло анализа.



- выбрать вид анализа (*Analysis Variation*) (процесс (*Process Variation*) и/или эффекты рассогласования (*Mismatch effects*));
- определить выходы (*Outputs*) с помощью калькулятора и задать их типы (scalar или wave);
- запустить анализ Монте-Карло;
- вывести результаты с помощью графического или текстового интерфейса.

## 4 Управление пользовательскими углами моделирования

### 4.1 Введение

Эта глава описывает, как настроить Ваши пользовательские углы моделирования до того, как начать моделирование в среде Analog Design Environment (ADE).

Также в этой главе описываются функциональные возможности пользовательских углов моделирования, которые поддерживаются средствами проектирования.

Отметим, что симулятор *Hspice* здесь не поддерживается.

### 4.2 Методология

Используя преимущества интерфейса *Setup User-Defined Corners*, пользователь может создать новый угол моделирования (так называемый пользовательский угол), определив набор постоянных величин для параметров процесса.

*Модельный файл* будет включать в себя этот новый набор величин и тогда будет обращаться к обычным секциям *CORREL* и *MODEL* (см. главу 2) для моделирования статистических моделей. Обратите внимание, что семейства, поддерживаемые предыдущей методологией некоррелированных худших случаев, не включены непосредственно в пользовательские углы моделирования, но они могут также упоминаться в *Model File*.

К тому же, для целей моделирования, некоторые средства проектирования определяют модели прибора в нескольких модулях. Некоторые семейства приборов могут быть поделены между несколькими из этих модулей: например, МОП-транзистор и радиочастотный МОП-транзистор часто описываются в разных модулях, несмотря на то, что они должны изменяться вместе в условиях моделирования углов. Такая возможность теперь поддерживается ArtistKit, и параметры процесса, которые могут определяться в разных модулях для данного семейства, сгруппированы вместе графическим интерфейсом.

Покуда интерфейсы *Setup Corners* (см. главу 2 ) и *Setup User-Defined Corners* участвуют в одном и том же *Model File*, Вам будет разрешено моделирование схемы путем вызова в качестве альтернативы обычных предопределенных углов и углов, определяемых пользователем.

- **Пример со Spectre**

- **Файл пользовательских углов моделирования:** *process\_spectre.best*

Предположим, что пользовательский угол имеет название *best*: тогда угловой файл называется *process\_spectre.best* и хранится в *Library*, определенной в интерфейсе *Setup Corners* ( см. главу 2 ):

```
parameters polygp_cdvar = 0.0 + (3.0*3.000e-9)
```

```
parameters npoly_drshu = 0.0 + (3.0*16.667)
```

```
parameters ppoly_drshu = 0.0 + (3.0*33.333) ...
```

- **Model File** создается интерфейсом *Setup User-Defined Corners*:

```
include "<USER_LIBRARY_PATH>/process_spectre.best"
```

```
include "<LIBRARY_PATH>/common_poly.scs" section=CORREL
```

```
include "<LIBRARY_PATH>/svtgp.scs" section=CORREL
```

```
include "<LIBRARY_PATH>/rpolyp.scs" section=typ ...
```

#### 4.3 Выбор пользовательских углов моделирования

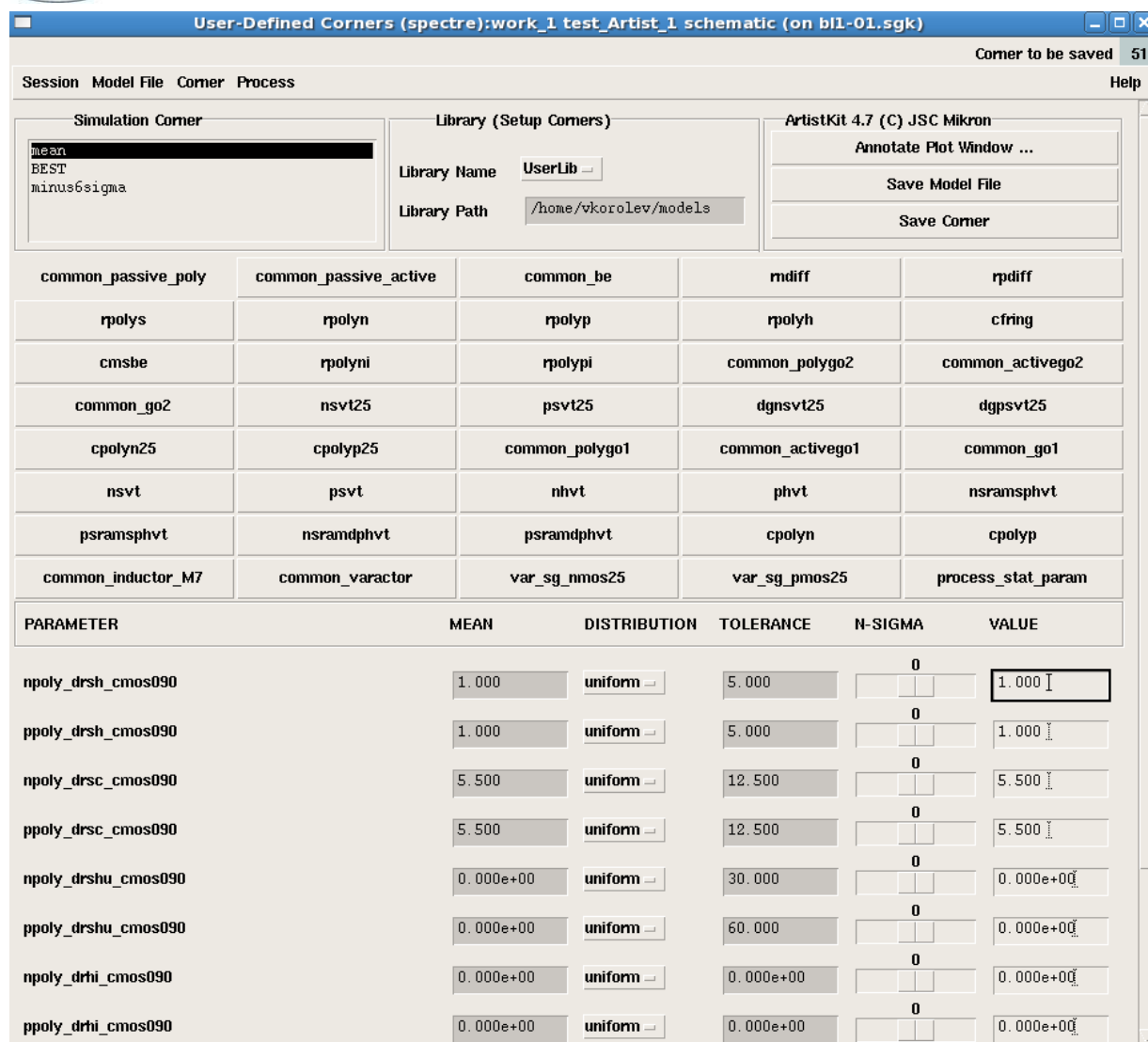
- Выберите *Tools->\*Setup User-Defined Corners* в окне Analog Artist (рис. 4.1)

Это окно отображает для каждого параметра процесса (PARAMETER) его среднее значение (MEAN), тип распределения (DISTRIBUTION) и допустимое отклонение (TOLERANCE). Параметры процесса классифицируются по семействам: при нажатии кнопки определенного семейства отображаются соответствующие параметры.

Для каждого параметра процесса текущее значение может меняться:

- с использованием непосредственно поля *VALUE*
- с помощью шкалы поля *N-SIGMA* в заданных пределах; происходит пересчет текущего значения *VALUE* как  $MEAN + N-SIGMA * TOLERANCE$ .

Примечание 1: Диапазон *N-SIGMA* колеблется в пределах [-1,+1] при равномерном распределении и в пределах [-6,+6] при гауссовом распределении. Вы можете менять



PARAMETER	MEAN	DISTRIBUTION	TOLERANCE	N-SIGMA	VALUE
npoly_drsh_cmos090	1.000	uniform	5.000	0	1.000
ppoly_drsh_cmos090	1.000	uniform	5.000	0	1.000
npoly_drsc_cmos090	5.500	uniform	12.500	0	5.500
ppoly_drsc_cmos090	5.500	uniform	12.500	0	5.500
npoly_drshu_cmos090	0.000e+00	uniform	30.000	0	0.000e+00
ppoly_drshu_cmos090	0.000e+00	uniform	60.000	0	0.000e+00
npoly_dtrhi_cmos090	0.000e+00	uniform	0.000e+00	0	0.000e+00
ppoly_dtrhi_cmos090	0.000e+00	uniform	0.000e+00	0	0.000e+00

Рис. 4.1 Окно Setup User-Defined Corners.

диапазон гауссового распределения на новую величину, определив специальную переменную *UARTCornerUserNrange* в своем файле *.cdsinit*.

Примечание 2: Когда тип распределения меняется (например от равномерного к гауссовому), то поле допустимого отклонения (TOLERANCE) обновляется (от равномерного допустимого отклонения к стандартному гауссовому отклонению), значение поля *N-SIGMA* сбрасывается до 0 и диапазон обновляется (от [-1,+1] до [-6,+6]).

Набор значений параметров процесса определяет пользовательский угол моделирования. При первом запуске интерфейса *Setup User-Defined Corners* доступен один угол моделирования под названием *mean* (средний). С этим средним углом моделирования все параметры процесса принимают свои средние значения, что эквивалентно, скажем, тому, что все поля *N-SIGMA* принимают значение 0.

Интерфейс *Setup User-Defined Corners* использует те же библиотеки, которые были определены в окне *Setup Corners* (см. главу 2) для сохранения углов моделирования.

Текущий статус отображается под заголовком окна. При первом открытии окна установлено *Corner to be saved*, и так устанавливается каждый раз, когда происходит изменение значения параметра процесса. После сохранения угла моделирования статус меняется на *Model File to be saved*. В итоге, после сохранения модельного файла статус указывает, что модельный файл сохранён.

- Выберите определенный пользователем угол из списка в поле *Simulation Corner*.

При выборе одного пункта в поле списка *Simulation Corner* обновляются все поля параметров процесса. Потом этот угол может быть изменен через параметр путем установления значений в поле *VALUE* (меняя *DISTRIBUTION*, перемещая ролик по шкале *N-SIGMA*). Параметры каждого семейства приборов могут быть отображены при нажатии соответствующих вкладок.

- Выберите библиотеку, где будет храниться пользовательский угол моделирования.

Меню поля названия библиотеки отображает библиотеки, определенные в окне *Setup Corners*. Для выбранной библиотеки отображается ее путь.

- Нажмите кнопку *Save Corner*.

Файл пользовательского угла (описанного в предыдущем разделе) сохранен по текущему адресу библиотеки. Имя созданного файла построено из названия угла и текущего симулятора:

<LIBRARY\_PATH>/process\_eldo.<cornerName>

<LIBRARY\_PATH>/process\_spectre.<cornerName>

Вы можете отобразить содержание текущего файла пользовательского угла, хранящегося в текущей библиотеке, в окне просмотра файлов с помощью команды меню *Process -> Show Corner*.

Выбор команды меню *Corner -> Save* производит то же действие, что и нажатие кнопки *Save Corner*.

Текущий статус окна меняется *Model File to be saved*.

- Нажмите кнопку *Save Model File*.

Модельный файл, как было описано в прошлом разделе, создается и хранится в текущей директории запуска моделирования. Этот файл используется точно так же как в интерфейсе *Setup Corners* (см. главу 2).

Выбор команды меню *Model File -> Save* производит то же действие, что и нажатие кнопки *Save Model File*.

Текущий статус окна указывает - *Model File to be saved*.

Выбор команды меню *Model File -> None* статус окна сбрасывается до статуса *Model file to be saved*, это означает, что *Model File* удален и в список соединений не включена ни одна модель.

- Выберите команду меню *Model File -> Show*

*Model File*, получаемый после команды *Model File -> Save* отображается в окне просмотра файлов.

- Запустите моделирование в окне *Analog Artist*

Вы можете использовать команду меню *Simulation->Run* или эквивалентный значок.

- Закройте окно, выбрав команду меню *Session -> Close*

Текущий статус выбранных параметров процесса сохраняется в директории запуска моделирования в файле под названием *usercorner\_map*. Этот *map* файл учитывается при запуске новой рабочей сессии.

#### 4.4 Управление пользовательскими углами моделирования

Данный набор значений параметров процесса определяет пользовательский угол моделирования. При первом запуске интерфейса *Setup User-Defined Corners* доступен один угол моделирования под названием *mean* (средний). С этим средним углом моделирования все параметры процесса принимают свои основные значения.

Доступные углы моделирования отображаются в окне *Setup User-Defined Corners* в поле списка *Simulation Corner*. При выборе одного угла из списка происходит обновление полей *DISTRIBUTION*, *N-SIGMA* и *VALUE* для каждого параметра (*PARAMETER*).

#### 4.4.1 Определение нового угла моделирования.

- Для каждого параметра (*PARAMETER*) каждого семейства приборов измените поле *VALUE* или выберите поле *DISTRIBUTION* и передвиньте шкалу *N-SIGMA*.
- Выберите команду меню *Corner -> New*, которая отображает следующую форму (рис. 4.2).



Рис. 4.2 Определение нового угла моделирования.

*New corner name* определяет имя нового угла моделирования, который примет текущие значения параметров процесса *N-SIGMA* и *VALUE*.

- Задайте имя для нового угла моделирования и нажмите *OK*.

В поле списка появляется новый угол моделирования, и он становится текущим. Выбор любого другого угла в списке сбросит обратно все значения полей *VALUE*, *DISTRIBUTION* и *N-SIGMA* для каждого параметра.

#### 4.4.2 Изменение угла моделирования

- Выберите угол моделирования в поле списка углов моделирования.

Поля *VALUE*, *DISTRIBUTION* и *N-SIGMA* обновляются в соответствии со значениями этого угла.

- Измените как Вам необходимо поля *VALUE* и/или *N-SIGMA*
- Выберите команду меню *Corner -> Save* или нажмите кнопку *Save Corner*.

#### 4.4.3 Удаление угла моделирования

- Выберите угол моделирования в поле списка углов моделирования.

Поля *VALUE*, *DISTRIBUTION* и *N-SIGMA* обновляются в соответствии со значениями этого угла.

- Выберите команду меню *Corner -> Delete*

Перед Вами появится диалоговое окно для подтверждения удаления.

- В диалоговом окне нажмите *Yes*.

Текущий угол удален из поля списка углов моделирования. Теперь первый угол в списке является текущим. Заметьте, что должен существовать хотя бы один угол моделирования: невозможно удалить их все.

#### 4.4.4 Загрузка угла моделирования

- Выберите команду *Corner -> Load*
- Введите имя файла в поле формы, которая высветилась перед Вами.

Файл должен был быть создан с помощью команды *Corner->Save* или должен был быть определен в соответствии с таким же форматом.

- В данной форме нажмите *OK*.

Угол моделирования, распознанный в этом файле, добавляется в поле списка углов моделирования.

Имейте в виду, что существует возможность загружать углы моделирования, которые определены как в данной ячейке и в данном симуляторе, так и в другой ячейке и в другом симуляторе.



#### 4.5 Создание комментариев к окну waveform

Окно *Setup User-Defined Corners* позволяет Вам легко добавлять информацию об условиях моделирования на графике в окне waveform. Вы можете комментировать температуру, переменные проекта и текущие пользовательские углы моделирования.

- Выберите пользовательский угол моделирования, как было описано в предыдущих разделах, и оставьте открытым окно настройки пользовательских углов.
- Настройте и запустите моделирование.

Используя меню *Analog Artist*, настройте условия Вашего моделирования (переменные проекта, температуру, входные сигналы, вид анализа,..), список соединений и промоделируйте Вашу схему.

- Выведите результаты Вашего моделирования в окне waveform.

Для этого, используйте команды меню *Analog Artist Outputs* и *Results -> Plot Outputs*.

- В окне настройки пользовательских углов, нажмите кнопку *Annotate Plot Window*.

Перед Вами появится форма (рис. 4.3) со списком, позволяющая выбрать один или несколько пунктов среди температуры, переменных проекта и пользовательских углов моделирования.

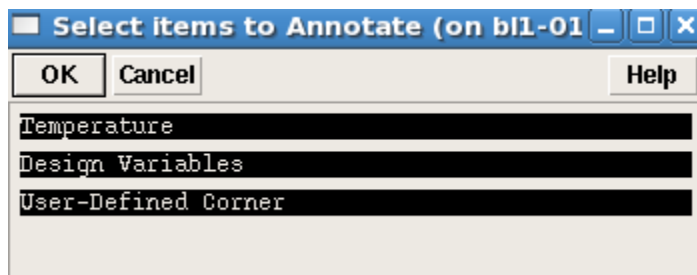


Рис. 4.3 Создание комментариев к окну waveform.

- Выберите один или несколько пунктов в списке формы и нажмите *OK*.

Текстовые комментарии отображаются в текущем окне waveform. Вы можете редактировать и изменять эти комментарии (содержание, ориентация, шрифт, размер, цвет...) с помощью команды меню *Annotation->Edit* (для AWD) в окне waveform. Также Вы можете передвигать эти комментарии по данному окну.

#### 4.6 Сохранение состояния (State) в окне Setup User-Defined Corners

Сохранение состояния (state) окна *Setup User-DefinedCorners* заключается в переводе данных всех сессий *Setup User-Defined Corners* в файл состояния, как и в ADE. Это позволяет этим данным быть перезагруженными из любой другой сессии *Setup User-Defined Corners*, а также эти данные могут передаваться в другую ячейку и/или другой симулятор.

- Выберите команду меню *Session -> Save State*.

Отобразится следующая форма (рис.4.4)

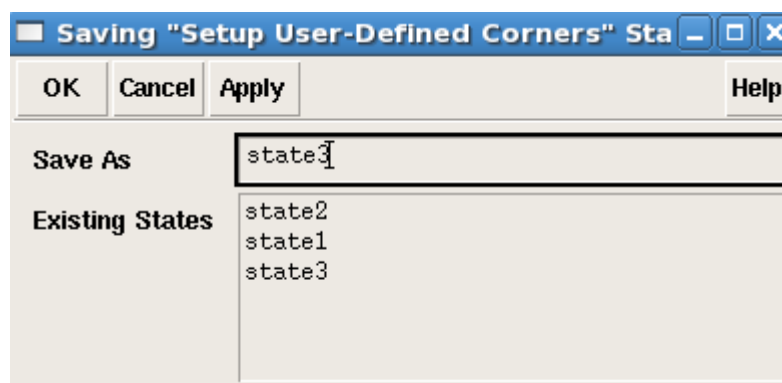


Рис. 4.4 Сохранение состояния(state).

- Введите название состояния в поле формы *Save as* или выберите одно из списка состояний в поле *Existing States*.
- Нажмите *OK*.

Данные сессии сохранились в структуре директории сохранения ADE в файле состояния, определяемом как:

```
<state_save_dir>/<libName>/<cellName>/<simulator>/  
.setupUserDefinedCorner/<stateName>
```

где

<state\_save\_dir> -это ADE State Save директория (выберите

*Session->Options* в окне ADE )

<libName> -это имя библиотеки проекта в окне ADE

<cellName> -это имя ячейки проекта в окне ADE

<simulator> -это имя симулятора в окне ADE

<stateName> -это имя *Setup Corners* состояния(state)

#### 4.7 Загрузка состояния (State) в окне Setup User-Defined Corners

Загрузка состояния в окне *Setup User-Defined Corners* заключается в обновлении окна *Setup User-Defined Corners* с помощью данных, хранящихся в файле состояния, также как в ADE. Состояние в окне *Setup User-Defined Corners* создается путем выбора команды *Session->Save State*, как было описано выше.

- Выберите команду *Session -> Load State*.

Загрузка состояния заменит значения углов моделирования текущей сессии теми значениями, которые определены в данном состоянии: поэтому для Вас появляется подсказка диалогового окна об итоговом сохранении текущего состояния режима до того, как загрузить следующий.

- Выберите *Да(yes)* или *Нет(no)* в диалоговом окне.

Если Вы выбрали *Да*, то пожалуйста перейдите к предыдущему пункту. Затем появится следующая форма, которая отображает содержание директории сохранения состояния ADE в условиях состояния *Setup User-Defined Corners*.

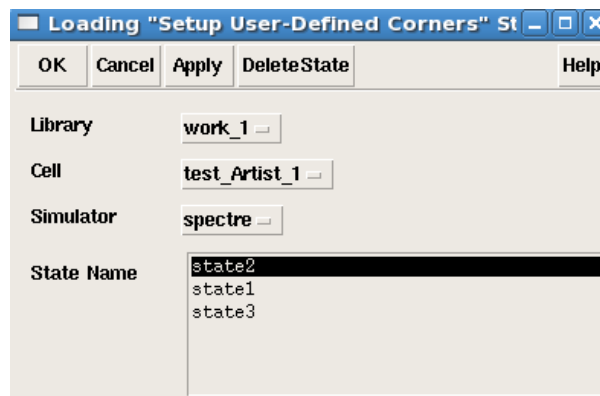


Рис. 4.4 Сохранение состояния(state).

- В этой форме выберите поля *Library*, *Cell* и *Simulator*.

Доступные режимы для *Library*, *Cell* и *Simulator* отражены в поле списка *State Name*.

- Выберите название режима *State Name*, которое Вы желаете загрузить.

Существует возможность удалить выбранный режим нажатием кнопки *DeleteState*.

- Нажмите *OK*.

Окно *Setup User-Defined Corners* обновлено.

Примечание: Следует учесть, что могут появиться некоторые предупреждения, если имеются несоответствия между загружаемым Вами режимом и текущей сессией *Setup User-Defined Corners* ( несоответствия в названии семейств, углов, технологии, несоответствия между моделями пользователя,.. )

## 5 Импорт списков соединений (import netlists)

### 5.1 Введение

Эта глава описывает, как импортировать список соединений (netlist) в принципиальную электрическую схему (schematic) путем создания специальных ячеек, видов представления ячеек (cellview) и свойств.

### 5.2 Методология

Импортирование списка соединений удобно в случае, когда проектировщик имеет только текстовое представление списка соединений одного блока схемы. Например, случай, когда список соединений был создан с помощью сторонних средств проектирования.

Назначение этой функции ArtistKit состоит в том, чтобы иметь возможность из сессии Artist Kit обращаться и присоединять такие внешние списки соединений к конечному списку соединений верхнего уровня принципиальной схемы.

Для этой цели создаются или изменяются специальные ячейки с определенными пользователем выводами (pins) и параметрами, видами представления ячеек, списками соединений и информацией о моделировании.

Во время нетлистования импортируемых блоков в директории моделирования */netlist* создается файл под названием *importNetlist*, содержащий подключаемые выражения всех импортируемых списков соединений.

Могут быть выделены два основных применения:

- импорт списка соединений, который имеет только текстовое представление списка соединений: здесь необходимо создавать новую ячейку с нуля.
- импорт списка соединений, который имеет, по крайней мере, схематическое и топологическое представления: здесь ячейка уже существует, добавляются только ее виды представления и свойства для нетлистования и моделирования.

- *Случай импорта списка соединений только с текстовым представлением*

Этот случай позволяет иллюстрировать текстовое представление списка соединений в верхнем уровне принципиальной электрической схемы (schematic); первой создается символьное представление ячейки (*symbol*), с использованием списка входных и выходных выводов, заданных пользователем.

Затем необходим специальный вид представления (*stop view*) для каждого поддерживаемого симулятора (*eldo*, *eldoD*, *spectre*, *hspiceS*, *hspiceD*), для того чтобы нетлистер Analog Artist мог добавить блок к списку импортируемых файлов.

В итоге запрашивается следующая информация *CDF* (рис. 5.1):

- два параметра, определяющие соответственно имя полного импортируемого файла (*full import file*) и имя импортируемой подсхемы (*Sub-Circuit File*). Предполагается, что импортируемый список соединений предоставляется в виде инкапсуляции подсхемы, написанной на выходном языке симулятора (*eldo*, *hspice* или *spectre*).
- некоторые дополнительные параметры, определяемые пользователем
- информация о нетлистировании и моделировании для поддерживаемых симуляторов (процедура нетлистования, упорядоченный список выводов,...)

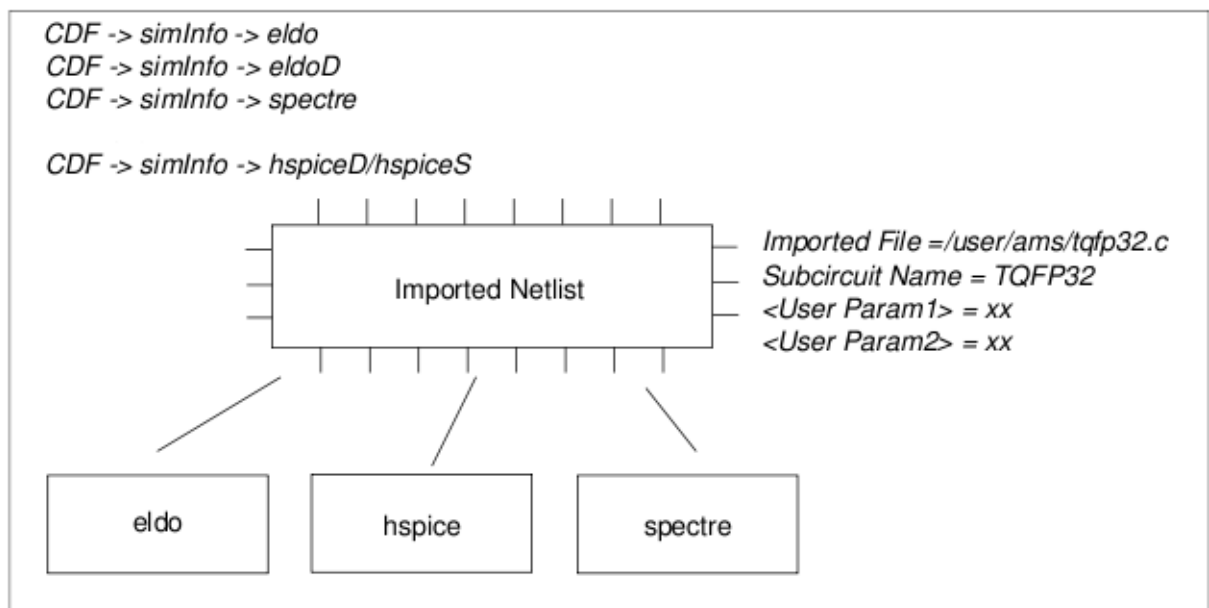


Рис.5.1 Случай импорта списка соединений с текстовым представлением.

- Случай списка соединений со схмотехническим и топологическим представлениями

Здесь ячейка уже существует в виде *symbol* и в своем обычном представлении *schematic* и *layout*. Экстрагированный список соединений создавался с использованием внешнего средства проектирования в виде инкапсуляции подсхемы, написанной на выходном языке симулятора; целью является использование этого списка соединений (с

учетом паразитных параметров) вместо принципиальной схемы (schematic) в моделировании схемы с учетом паразитных параметров (*PLS*). С помощью Hierarchy Editor может быть проведено сравнение результатов моделирования принципиальной схемы с результатами моделирования схемы с учетом паразитных параметров.

Во-первых, необходим специальный вид представления (stop view), для того чтобы нетлистер Analog Artist мог добавить список соединений к списку импортируемых файлов.

Во-вторых, требуется следующая CDF информация:

- параметр, определяющий число параллельно подключаемых блоков ( коэффициент  $m$  )
- Информация о моделировании для поддерживаемых симуляторов.

Виды представления (stop views), созданные в случае списков соединений с учетом экстракции паразитных параметров, должны быть добавлены в списки *Switch* и *Stop View Lists*, если они отличаются от тех, которые управляются *PLSKit*.

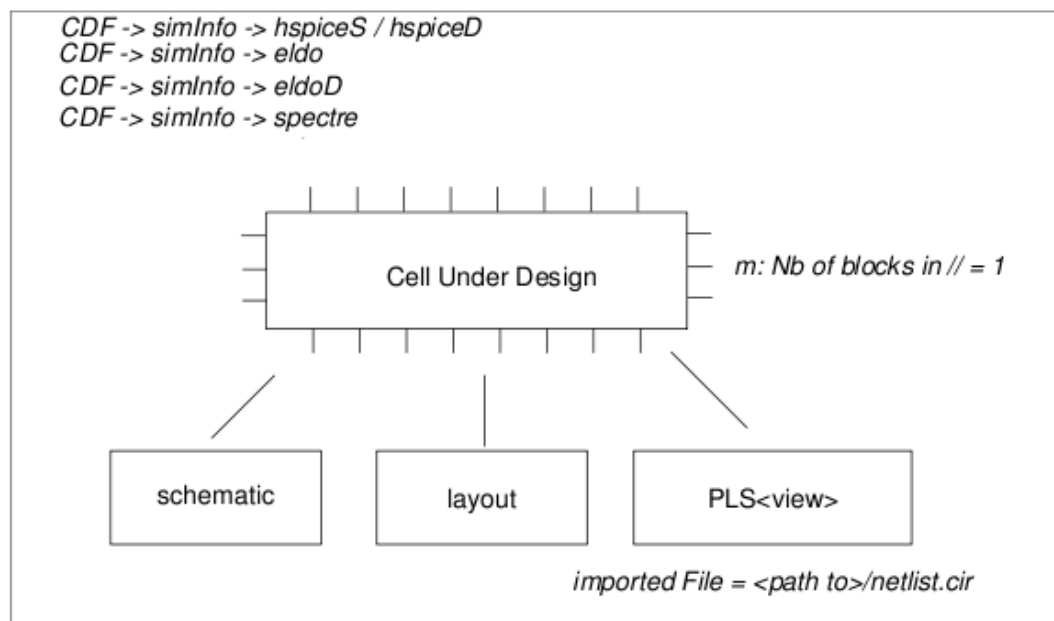


Рис.5.2 Случай списка соединений со схемотехническим и топологическим представлениями.

Эта методология используется в средстве проектирования *PLSKit* для маршрута под названием *Textual Netlist Import*.

### 5.3 Импорт списка соединений с текстовым представлением

- Откройте сессию Analog Artist на верхнем уровне Вашей принципиальной схемы
  - Выберите команду меню *Tools-> \*Import Netlist -> Create/Update Symbol* (рис.5.3)
- Появится форма импортирования списка соединений.

Поле *Library Name* определяет вашу библиотеку, которой будет принадлежать импортируемый блок. По умолчанию, это библиотека, содержащая Ваш проект.

Поле *Cell Name* определяет имя новой импортируемой ячейки. Когда ячейка уже существует, все предыдущие определения будут удалены: *CDF* информация и *Symbol*. По определению, имя ячейки носит название *Import*.

Кнопка *Browse* позволяет Вам заполнить первые два поля с помощью *Library Browser*.

Поле *File Name* определяет имя полного пути к файлу, содержащему импортируемый список соединений. По умолчанию, это файл под названием *Import* в директории, из которой Вы запускаете *Cadence Design Framework*.

Поле *Sub-circuit Name* определяет имя подсхемы, описанной в *File Name*. По умолчанию, оно принимает название ячейки. Это имя будет использоваться нетлистером для приписания подсхеме выражения “X” в случае *eldo*.

Поле *Input Terminals* определяет имена входных выводов. Эти имена должны быть скомпонованы по допустимым значениям символов симулятора и разделены хотя бы одним пробелом.

Поле *Output Terminals* определяет имена выходных выводов. Эти имена должны быть разделены хотя бы одним пробелом.

Меню *Terminal Order* предлагает два варианта:

- Использовать для нетлиستования порядок выводов по умолчанию, при котором сначала стоят входные выводы (*Input Terminals*), затем выходные (*Output Terminals*).



- Использовать пользовательский порядок, который Вы можете задать с помощью поля, находящегося ниже. Если Вы используете это пользовательское поле, то все

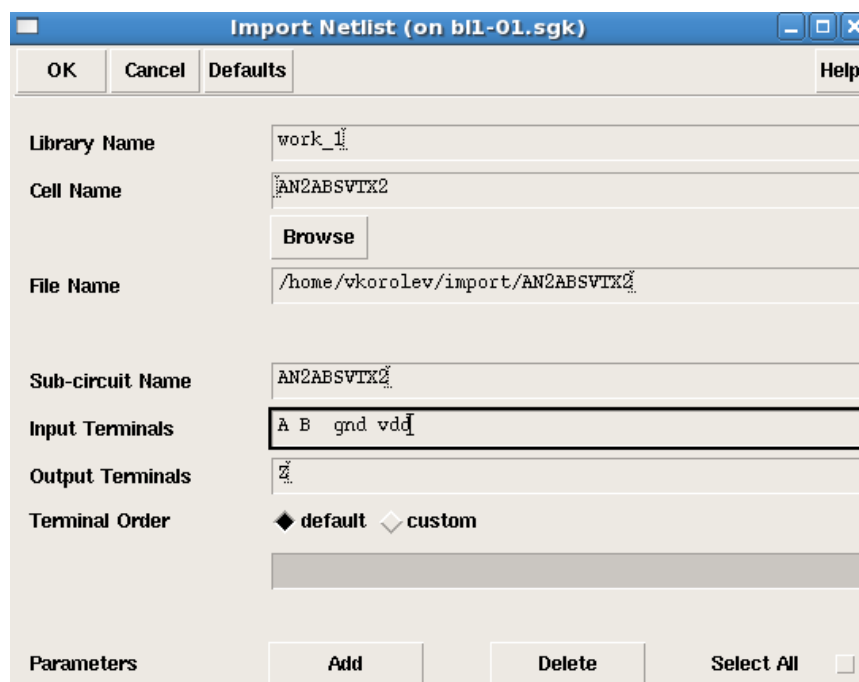


Рис.5.3 Окно импорта списка соединений.

*Input* и *Output Terminals* должны быть прописаны туда.

Кнопки *Parameters* (*Add*, *Delete* и *Select All*) управляют импортом параметров подсхемы. Для добавления параметра нажмите кнопку *Add*: это создаёт строку с тремя новыми полями внизу формы *Import Netlist*:

- поле *Parameter Name* назначает новое имя параметра
- поле *Value* назначает величину по умолчанию, которая будет храниться в CDF параметрах ячейки.
- логическая клавиша *Select All*, позволяющая проектировщику выбирать все специальные параметры до того, как нажать клавишу *Delete* для удаления.

Для удаления всех значений параметров нажмите кнопки *Select All* и *Delete*.

- Нажмите *OK*, когда все поля будут заполнены.

Это активизирует создание ячейки *symbol*, списка соединений (*stop view netlist*) (*eldo*, *eldoD*, *hspiceS*, *hspiceD*, *spectre*) и CDF параметров для Analog Artist.

#### 5.4 Включение/ выключение импорта списка соединений.

- Выберите *Tools -> \*Import Netlist -> Enable*

Это активирует функциональные возможности импорта списка соединений, означающие, что нетлистер Analog Artist будет присоединять все импортируемые списки соединений, которые найдет во время просмотра своей иерархии. Это можно понимать как обновление файла *importNetlist*.

- Запустите нетлистер ADE

Во время нетлистования файл под названием *importNetlist*, содержащий все присоединенные файлы импортируемых списков соединения, создается в директории запуска моделирования.

Для всех симуляторов, файл *importNetlist* присоединяется к конечному списку соединений через файл *setupCornersIncludeFile*, который уже используется пользовательским файлом *Setup Corners* ( см. раздел 2.4).

- Выберите *Tools -> \*Import Netlist -> Show File*

Содержание файла *importNetlist* отображается в окне просмотра файла.

- Выберите *Tools -> \*Import Netlist -> Disable*

Эта команда полезна в случае, когда возможность импорта списка соединений не должна больше приниматься во внимание: например, когда Вы не обращаетесь к какому-либо импортируемому списку соединений в Вашем проекте: в этом режиме файл *importNetlist* не будет больше присоединяться к конечному списку соединений.

Заметьте, что может быть особенно полезно включать/выключать механизм импорта списка соединений при запуске Ocean скрипта.

## 6 Работа в Hierarchy Editor (HED)

### 6.1 Введение

Эта глава содержит описание, как создавать конфигурации в Hierarchy Editor, а также как настраивать симулятор, управлять списками *Switch View*, *Stop View* и другие опции окружения, которые определяются с помощью ArtistKit.

### 6.2 Работа в Hierarchy Editor

Hierarchy Editor создаёт файл конфигурации, которые обеспечивают расширенную информацию для разделения схемы. Для создания файла конфигурации необходимы следующие шаги:

- В CIW-окне выберите *File->New->Cellview*
- В появившейся форме (рис. 6.1) выберите библиотеку вашего проекта
- В поле *Cell Name* выберите имя ячейки, которую вы хотите моделировать
- В циклическом меню *Tool* выберите *Hierarchy Editor* и нажмите *OK*.

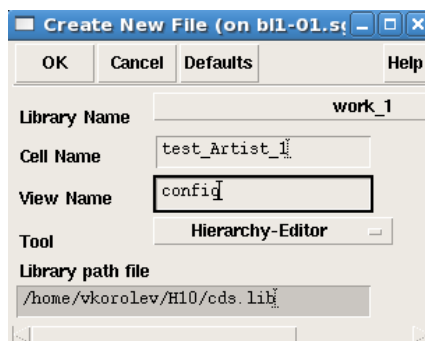


Рис.6.1 Создание вида config.

Отобразятся две формы: *Hierarchy Editor* и *New Configuration*.

- Кнопка *Use Template...* в форме *New Configuration* позволяет вам выбрать шаблон (template), совместимый с симулятором, который вы запускаете (рис. 6.2).

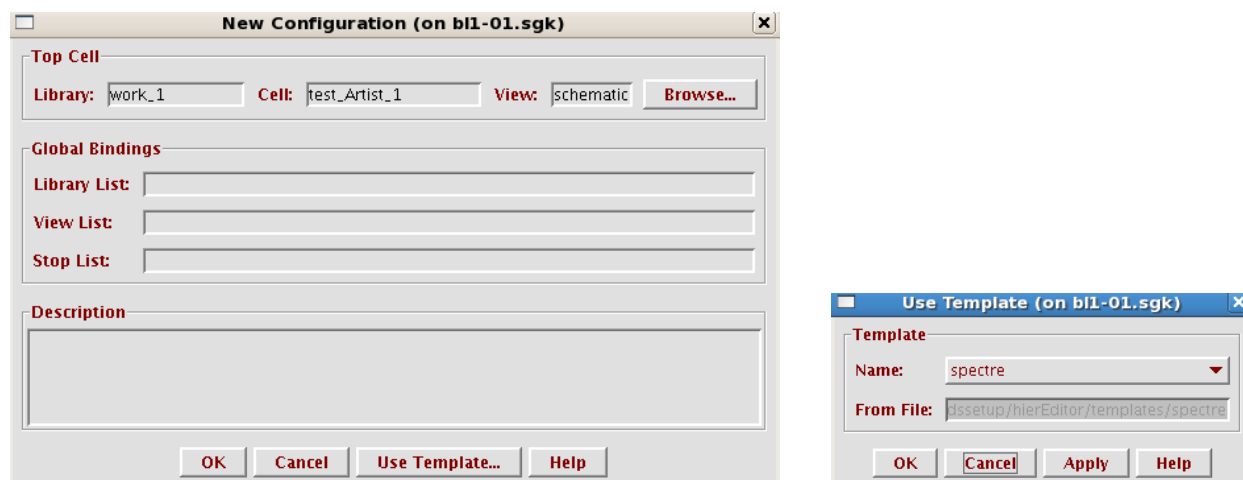


Рис.6.2 Выбор шаблона (template) для создания новой конфигурации.

После выбора шаблона поля формы “New Configuration” заполняются автоматически, как показано на рис. 6.3.

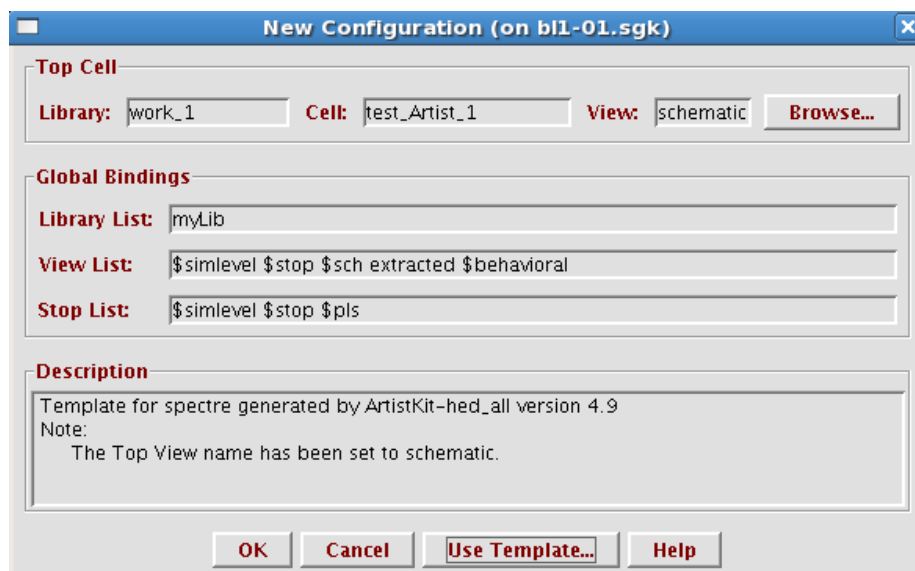


Рис. 6.3. Окно создания новой конфигурации проекта после заполнения всех полей.

Шаблоны ArtistKit содержат:

- определение уровней моделирования (Simulation Levels) с HED постоянной, называемой \$simlevel;
- определение схмотехнического представления (Schematic views) с HED постоянной, называемой \$sch;

- определение видов, полученных после экстракции паразитных параметров (Post-Layout Simulation views) с двумя HED постоянными: \$pls и \$pls\_sch (продукт PLSKit должен быть подгружен, иначе эти постоянные будут недействительны).

- В появившемся окне “Cadence Hierarchy Editor” можно изменить представление элементов вашего проекта на любой из падающего списка, после чего нажать кнопку “Update”, сохранить созданную конфигурацию (рис. 6.4).

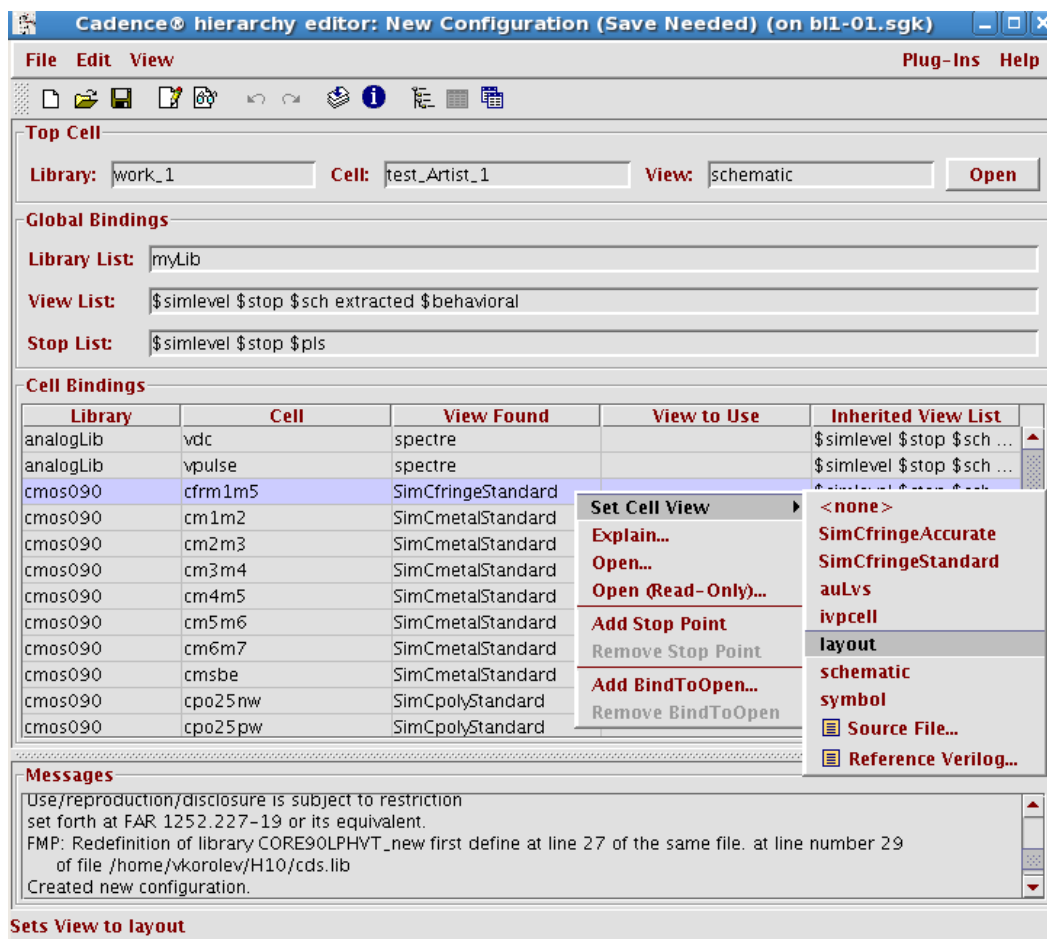


Рис. 6.4. Окно Hierarchy Editor.

- Открыть созданное на предыдущих шагах представление “config” с помощью Virtuoso Schematic Editor и вызвать из него Analog Design Environment. Далее следует убедиться, что используется действительно представление “config” (рис. 6.5). После этого запустить моделирование проекта с учетом заданных вами представлений элементов.

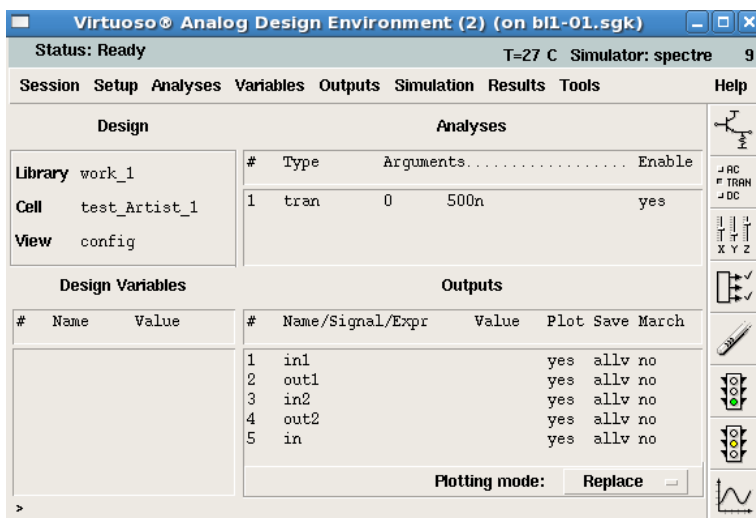


Рис. 6.5. Главное окно ADE, запущенное из представления “config”.

## 7 Настройка и запуск смешанного моделирования

### 7.1 Введение

Глава содержит информацию о смешанном моделировании в Mixed-Signal Design Enviroment (MSDE)-Artist.

Единственный поддерживаемый симулятор – spectreVerilog.

### 7.2 Создание конфигурации

Сессия смешанного моделирования должна быть определена с конфигурационным файлом Hierarchy Editor (HED) вашего проекта, который предоставляет расширенную информацию для смешанного моделирования.

Для этого необходимо совершить следующие действия:

- В CIW-окне выберите *File->New->Cellview*
- В появившейся форме выберите библиотеку вашего проекта
- В поле *Cell Name* выберите имя ячейки, которую вы хотите моделировать

- В поле View Name пропишите вид вашей конфигурации (например: *config\_spectreVerilog*)
  - В циклическом меню *Tool* выберите *Hierarchy Editor* и нажмите *OK*
- Отобразятся две формы: *New configuration* и *Cadence Hierarchy Editor*.
- Выберите *spectreVerilog* в поле выбора шаблона (*template*) *ArtistKit* (смотреть главу 6). *ArtistKit* предоставляет собственные шаблоны (*templates*) для симулятора *spectreVerilog*.

### 7.3 Открытие конфигурированного схмотехнического представления из CIW окна

- Выберите *File->Open* в CIW окне
- Выберите соответствующую библиотеку, ячейку и имя вида вашего проекта.
- Нажмите *OK*.
- В окне конфигурированного схмотехнического представления (*configured Schematic*), выберите *Tools->Mixed-Signal Opts.* для добавления *Mixed-Signal* меню в верхнюю панель.

### 7.4 Опции разбиения (Partitioning Options)

Конфигурация проекта, полученная в *Hierarchy Editor*, определяет список соединений и разбиение на части для каждой ячейки проекта. Разбиение делит все виды (*stopping view*) на аналоговую (*analog*) и цифровую (*digital*) части. Каждый нетлистер (аналоговый или цифровой) запускает соответствующую часть схемы.

Изменить эти данные для вашего проекта можно, выбрав *Mixed-Signal->Partition Options* в меню *Configured Schematic* (рис. 7.1).

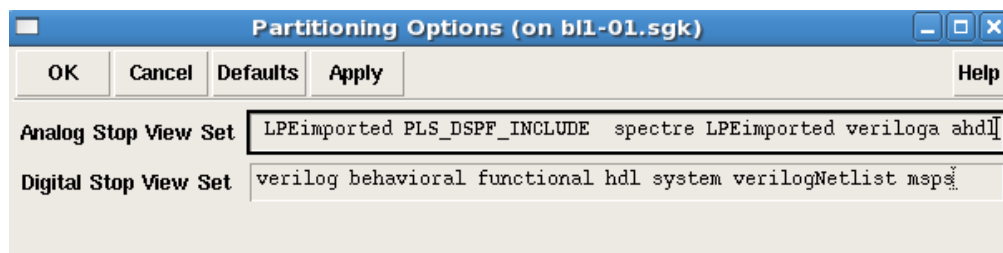
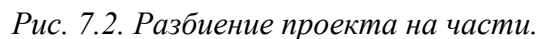


Рис. 7.1. Окно *Partitioning Options*.



## 7.5 Открытие сессии смешанного моделирования

## Выбор симулятора

- Выберите *Setup->Simulator/Directory/Host...* в окне ADE.
- Выберите *spectreVerilog* в поле *Simulator* появившейся формы и нажмите *OK* (рис. 7.3).



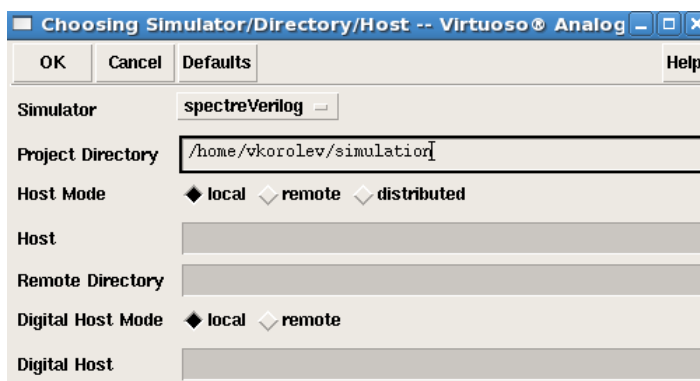


Рис. 7.3. Выбор симулятора.

## 7.6 Настройка моделей

Как и в сессии аналогового моделирования, аналоговые модели устанавливаются с использованием методологии *ArtistKit*, описанной в главе 2.

В сессии смешанного моделирования модельный файл, называемый *models*, создаётся при использовании в *ArtistKit* команды *Tools->Setup Analog Corners*.

## 7.7 Настройка аналоговых воздействий (Analog Stimuli)

Аналоговое воздействие может быть отредактировано с помощью команды *Setup->Stimuli->Analog* в окне ADE (рис.7.4).

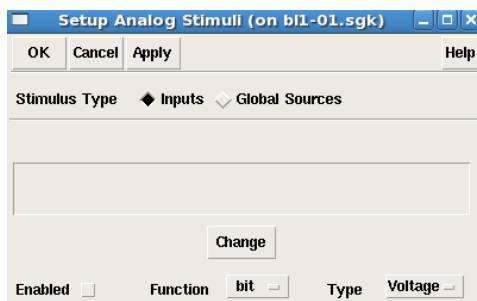


Рис. 7.4. Окно редактирования аналогового воздействия.

## 7.8 Настройка цифровых воздействий (digital stimuli)

Редактирование файла testfixture.verimix

Цифровые воздействия могут быть отредактированы с использованием команды *Setup->Stimuli->Digital* в окне ADE. Файл *testfixture.verimix* расположен в *.../netlist/digital* директории и открывается с помощью текстового редактора.

Файл уже содержит шаблон с именами действующих соединений (**actual net**) входных цифровых терминалов схемы. Опишем следующий пример:

```
// // Verimix stimulus file.
// Default verimix stimulus.
// ** Events included in loops **

initial
fork
#0 repeat (50)
begin
#0 plus = 0;
#1000 plus = 1;
#1000 ;
end
join
// ** Main flow of events **
initial
begin
#0 snall = 0;
#2000 snall = 1;

end
```

#### Работа со схемой тестовых воздействий

Если вы хотите создать схему тестовых воздействий, то вам необходимо создать ячейку с *symbol* и *behavioral textual* видами, которые содержат описание цифровых воздействий.

Для этого вы должны выполнить следующие действия:

- Создать *symbol* из *pinList* (списка выводов) (все входные цифровые терминалы вашей схемы являются выходными терминалами этого символа (*symbol*)), используя команду *Design -> Create Cellview->From Pin List*

- Создать behavioral вид на этот новый символ, используя *Design->Create CellView - > From CellView*. Затем выберите *symbol* для поля *From View Name* и *behavioral* (или *functional*) для поля *To View Name*.
- Введите текст вашего модуля воздействий, основываясь на заготовленном шаблоне. Например:

```
'timescale 1ns / 1ns
module peak_logic_stimuli (plus, snall);
output plus;
output snall;
reg plus;
reg snall;
// ** Events included in loops **
initial
fork
#0 repeat (50)
begin
#0 plus = 0;
#1000 plus = 1;
#1000 ;
end
join
// ** Main flow of events **
initial
begin
#0 snall = 0;
#2000 snall = 1;
end
endmodule
```

- Сохраните ваш тестовый модуль
- Проверьте результаты моделирования вашего модуля.

## 7.9 Запуск моделирования

### Настройка опций смешанного моделирования

Значения по умолчанию для *Verimix Convergence* опций: *DC Interval* принимает значение 10; *Max DC Iter* принимает значение 3.

Для изменения их значений для вашего проекта выберите *Simulation->Options->Mixed Signal* в окне ADE.

### Выбор анализа

В окне ADE выберите команду *Analyses->Choose* и настройте опции выбранного анализа.

Теперь моделирование может быть запущено с использованием команды *Simulation->Run*.

## 8 Контактная информация

Разработка и поддержка PDK:

[Mikron\\_Lib\\_support@mikron.ru](mailto:Mikron_Lib_support@mikron.ru)