

Приложение А
Содержание

Описание DCS_Electric.....	3
Настройка рабочей станции.....	3
Рекомендуемые системные требования:	3
Предварительная работа:.....	3
Маршрут проектирования программируемых схем	5
Проектирование аналоговой части	6
Создание электрической схемы	6
Основные блоки схемы	8
FPGA – Конфигурируемая пользователем цифровая схема	8
TB – Блок тестирования	11
ADC (x2) – 12-ти разрядный АЦП.....	11
DAC (x2) – 12-ти разрядный ЦАП.....	11
DPOP (x2) – силовые высоковольтные контактные площадки.....	12
SPI – интерфейс	13
MUX1in16 (x4) – мультиплексор 16:1	14
GAL – глобальная шина	15
DIOP (x26) – аналоговые площадки ввода/вывода из цифровой части	15
DIOP_OR (x4) – аналоговые площадки ввода/вывода из цифровой части с возможностью регулирования направления сигнала с помощью цифровой логики.	16
INPUT – вход схемы	16
OUTPUT – прямой выход схемы	16
OUTPUT_ADR_x2 – выход схемы через аналоговый буфер с усилением в два раза	17
OUTPUT_ADR – выход схемы через аналоговый буфер	17
OUTPUT_DDR – выход схемы через цифровой буфер	17
FPGA_connector – элемент для связи ПЛИС с динамическими ключами аналоговой части (включая блок PAM)	17
PAM – программируемый усилительный блок.....	19
PAM_CAP_FULL_DIFF – полностью дифференциальный усилитель с шунтирующими емкостями в обратных связях универсального усилительного блока PAM	20
PAM_COMP – компаратор на основе универсального усилительного блока PAM.	20
PAM_COMP_HYST – компаратор с гистерезисом на основе универсального усилительного блока PAM	20
PAM_INTGRTR – интегратор на основе универсального усилительного блока PAM.....	20
PAM_DIFF – полностью дифференциальный ОУ на основе универсального усилительного блока PAM	21
PAM_NEG_CAP_FB – ОУ с ООС и с шунтирующей емкостью на основе универсального усилительного блока PAM	21
PAM_NEG_FB – ОУ с ООС на основе универсального усилительного блока PAM	21
PAM_OA – ОУ на основе универсального усилительного блока PAM	21
PAM_OA_shunt – ОУ с шунтирующей емкостью на основе универсального усилительного блока PAM.....	22
RES – программируемый резистор из блока пассивных компонентов.....	22
SPI_source – источник сигналов (SS, MOSI, CLK, SCK) для управления SPI-интерфейсом в режиме «ведомый».....	22

SPM – блок свободной конфигурации	23
SWITCH – динамический аналоговый ключ	24
extern – блок имитации внешних соединений.....	24
VDD – вывод питания из блока пассивных компонентов.....	25
VSS – вывод «земли» из блока пассивных компонентов	25
CAP – конденсатор	25
ipulse – источник прямоугольных импульсов тока	26
isource – источник постоянного тока	26
vpulse – источник прямоугольных импульсов относительно VSS.....	26
vpulse_clear – источник прямоугольных импульсов	27
vsource – источник постоянного напряжения относительно VSS	27
vpwl – источник напряжения, задаваемый по точкам относительно VSS	27
vpwl_clear – источник напряжения, задаваемый по точкам.....	28
vsin – источник синусоидальных импульсов относительно VSS.....	28
vsource_clear – источник постоянного напряжения	28
vsin_clear – источник синусоидальных импульсов	28
5400TP094_core – блок параметров моделирования	29
Автоматическая трассировка схемы.....	30
Учёт трассировочных ключей и паразитных элементов	30
Сбалансированные цепи	30
Критические цепи	30
Ошибки при построении схем и методы их устранения.....	31
Ошибка количества используемых блоков.	31
Ошибка отсутствия названия основного модуля в параметрах конфигурации Fpga settings.	31
Ошибка установки номера входа/выхода.....	32
Отсутствие блока FPGA.....	32
Моделирование аналоговой части.....	33
Проектирование цифровой части.....	36
Проектирование цифровых схем с использованием языка Verilog	36
Смешанное моделирование (долгое).....	38
Логическое моделирование (v1.5+).....	38
Статический временной анализ (v1.5+).....	39
Проектирование цифровых схем при помощи встроенной цифровой библиотеки.....	40
Смешанное моделирование цифровой части.....	43
Интерфейс топологического редактирования X-PLACE	47
Пример работы программы в ручном режиме	47
Настройки отображения.....	49
Полное смешанное моделирование	49
Дополнительные настройки статического временного анализа	50
Основные опции запуска CVA:	50
Доступные временные ограничения	51
Функции, позволяющие получить списки имен по требуемому критерию.....	58
Лист регистрации изменений	59

Описание DCS_Electric

Программное обеспечение используется для проектирования, моделирования и конфигурирования схем. Программное обеспечение предназначено для работы с конфигурируемыми микросхемами 5400TP035 (ПАИС), 5400TP094 (ПАЦИС), 5400TP045A-031 (ПАМС), 5400TP055A-008 (ПИКЛ).



Рисунок 1. Программное обеспечение DCS_Electric

Настройка рабочей станции

Рекомендуемые системные требования:

- операционная система: Windows 7, Windows 8, Windows 10;
- оперативная память 8 ГБ;
- 20 ГБ свободного места на жёстком диске.

Предварительная работа:

- 1) Скопировать папку с программой на локальный диск.
- 2) Установить *Java* из папки *.Vninstall* на диск *C:*
- 3) Запустить файл *electric.bat* из папки с программой.
- 4) Загрузить настройки (выполняется один раз при первом запуске программы):

File → *Preferences* → *Import*

Путь к файлу *.\electric\Prefs\Cadence_style_PDC.xml*

Важно! Путь, где находится программа должен содержать только латинские символы. Папка, где находится программа, не должна требовать прав администратора.

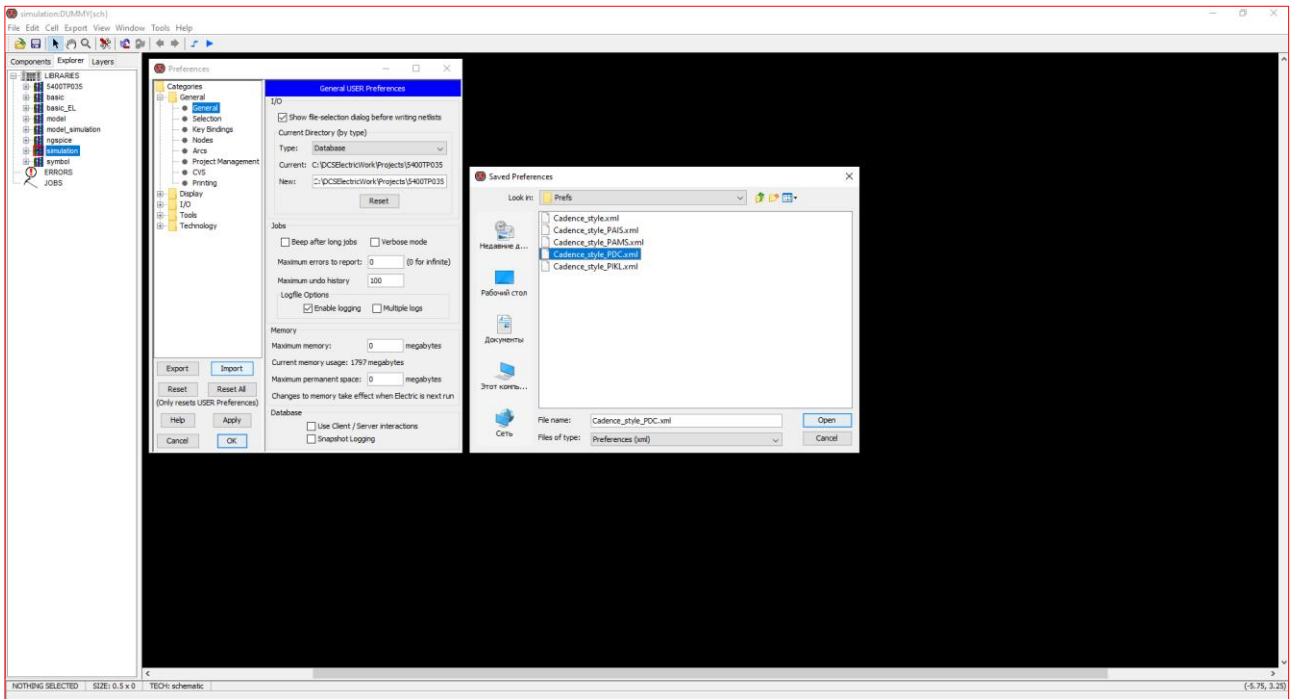


Рисунок 2. Окно загрузки настроек при первом запуске программы

5) Перезапустить программу.

6) Подключить скрипт для работы блока *SPI_source*. (Опционально)

Tools → *Languages* → *Manage Scripts* → *Add Script*

Добавить скрипт, находящийся по пути:

...DCS_Electric\electric\BeanShellScripts\form_digital_source_in_spi.bsh

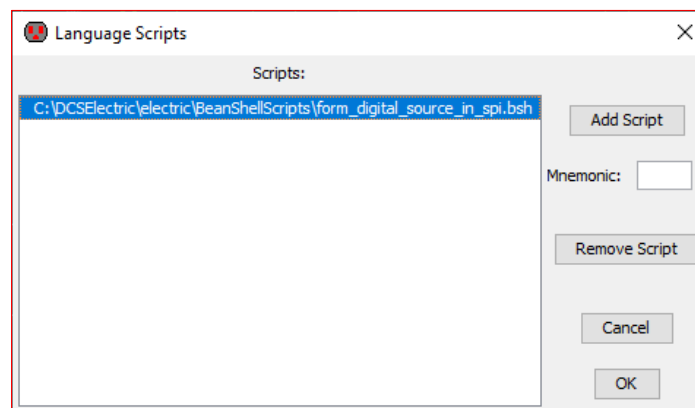


Рисунок 3. Окно управления скриптами

Маршрут проектирования программируемых схем

Общий вид маршрута проектирования программируемых схем приведён на рисунке 4.



Рисунок 4. Маршрут проектирования программируемых схем

САПР DCS_Electric позволяет спроектировать схему, используя готовые функциональные блоки, провести моделирование и создать конфигурационную последовательность для прошивки в микросхему.

Для прошивки микросхемы в ОЗУ или ПЗУ используется программа DCSProg-2 совместно с программатором (подробнее см. Руководство пользователя 5400TP094 Приложение Б).

Проектирование аналоговой части

Создание электрической схемы

1) Открыть проект (*File* → *Open Library*)

.DCSElectric\Projects\5400TP094\simulation.jelib

2) Создать новую схему (правой кнопкой мыши по библиотеке *simulation* → *Create New Cell*).

В открывшемся окне в поле *Name* ввести название схемы, в поле *View* выбрать *schematic*.

Функциональные блоки для проектирования схемы расположены в библиотеке *symbol* (Рисунок 7). Чтобы перенести компоненты из библиотеки в рабочее пространство, нажмите левой кнопкой мыши на нужный блок и, не отпуская кнопку, перетащите в рабочее пространство.

3) Собрать конечную схему, замыкая нужные ключи и выставляя необходимые параметры. Чтобы замкнуть ключ следует нажать на один из контактов ключа левой кнопкой мыши, а затем на другой контакт ключа правой кнопкой мыши.

Для коммутации блоков между собой следует нажать на один из контактов блока левой кнопкой мыши, а затем на контакт другого блока правой кнопкой мыши, либо произвести коммутацию вручную путем последовательной отрисовки проводника в необходимых областях. Для этого нужно нажать левую кнопку мыши на начальную точку, затем перевести курсор в необходимую область и нажать правую кнопку, повторять операцию до тех пор, пока не будет осуществлена коммутация нужных блоков.

Чтобы установить параметр необходимо нажать клавишу *Ctrl* и левой кнопкой мыши выделить данный параметр. После того как параметр выделен, отпустите клавишу *Ctrl* и дважды нажмите левой кнопкой мыши по параметру. Введите необходимый параметр и нажмите клавишу *Enter*.

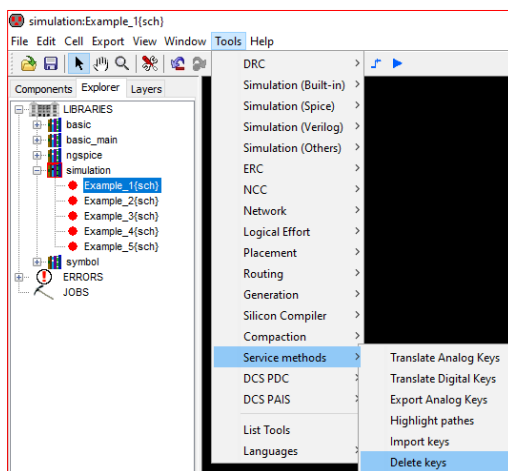


Рисунок 5. Размещение скрипта *Delete keys* в DCSElectric

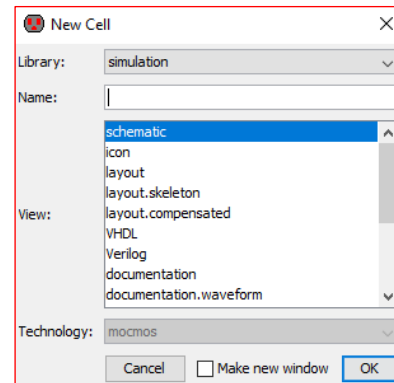


Рисунок 6. Окно создания новой схемы

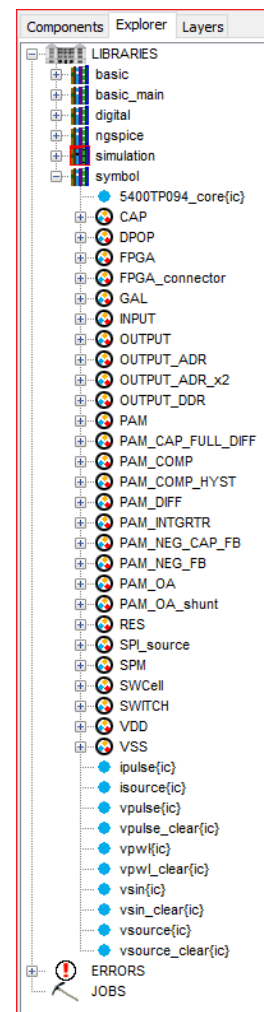


Рисунок 7. Расположение блоков в библиотеке *symbol*

Важно! Для одновременного удаления ключей из отдельного элемента или всей схемы существует скрипт *Delete keys*. Для его использования необходимо выделить, либо отдельный элемент левой кнопкой мыши, либо всю схему для удаления всех замкнутых ключей (*Ctrl+A*), затем выполнить команду *Tools* → *Service methods* → *Delete keys* (Рисунок 5). Для удаления одного ключа необходимо зажать клавишу *Ctrl* и нажать левой кнопкой мыши на ключ, который необходимо удалить. После того как ключ будет выделен белым цветом нажать клавишу *Backspace* или *Delete*. Для выделения нескольких ключей необходимо использовать комбинацию *Ctrl+Shift+ЛКМ*.

Примечание: в процессе проектирования рекомендуется периодически нажимать кнопку *F8* (исправление связей). При нажатии проверяются все связи в схеме и удаляются лишние соединительные элементы.

Зажатая клавиша *CTRL* позволяет выделять элементы, находящиеся на заднем плане.

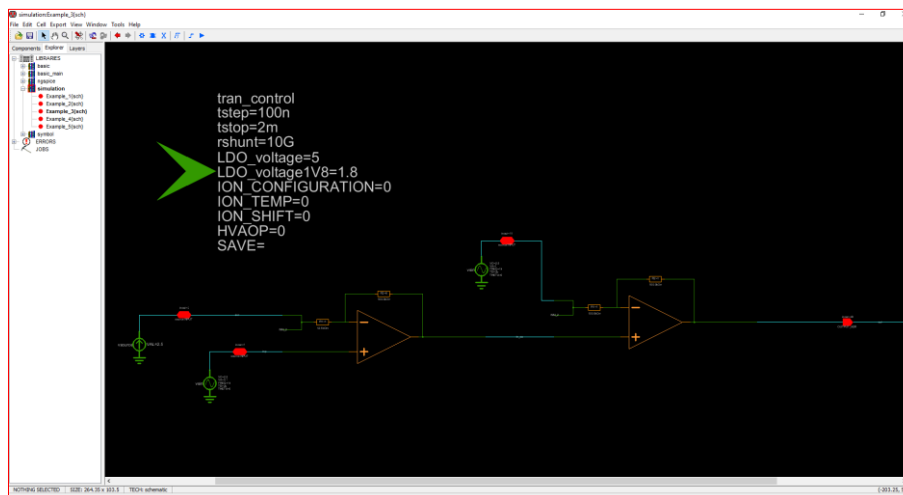


Рисунок 8. Окно проектирования «The Electric VLSI Design System»

Навигация в графическом интерфейсе программы:

- Приближение и отдаление активного поля
 - Клавиша «*E*» – приближение
 - Клавиша «*W*» – отдаление
 - Клавиша «*Z*» – масштабирование области
 - Клавиша «*Ctrl*» + прокрутка колеса мыши
 - Клавиша «*F*» – масштабирование и центрирование всей схемы
- Перемещение по полю
 - Нажать колесо мыши, перемещаться по полю
 - Нажать на значок «*Toggle Pan*» в поле инструментов и, зажав левую кнопку мыши, перемещаться по полю
 - «*Num2*» – перемещение по рабочей области вниз
 - «*Num4*» – перемещение по рабочей области влево
 - «*Num6*» – перемещение по рабочей области вправо
 - «*Num8*» – перемещение по рабочей области вверх
- Отмена действия
 - Сочетаний клавиш «*Ctrl*» и «*Z*»
 - Нажать на значок «*Undo*» в поле инструментов

Основные блоки схемы

FPGA – Конфигурируемая пользователем цифровая схема

В представленной библиотеке *symbol* отсутствует отдельный элемент, соответствующий исключительно блоку FPGA. Вся цифровая часть микросхемы представлена одним элементом с именем FPGA.

Важно! Конфигурация периферии и вспомогательных блоков ПЛИС осуществляется в процессе аналоговой автоматической трассировки.

Блок содержит:

FPGA – программируемый логический блок;

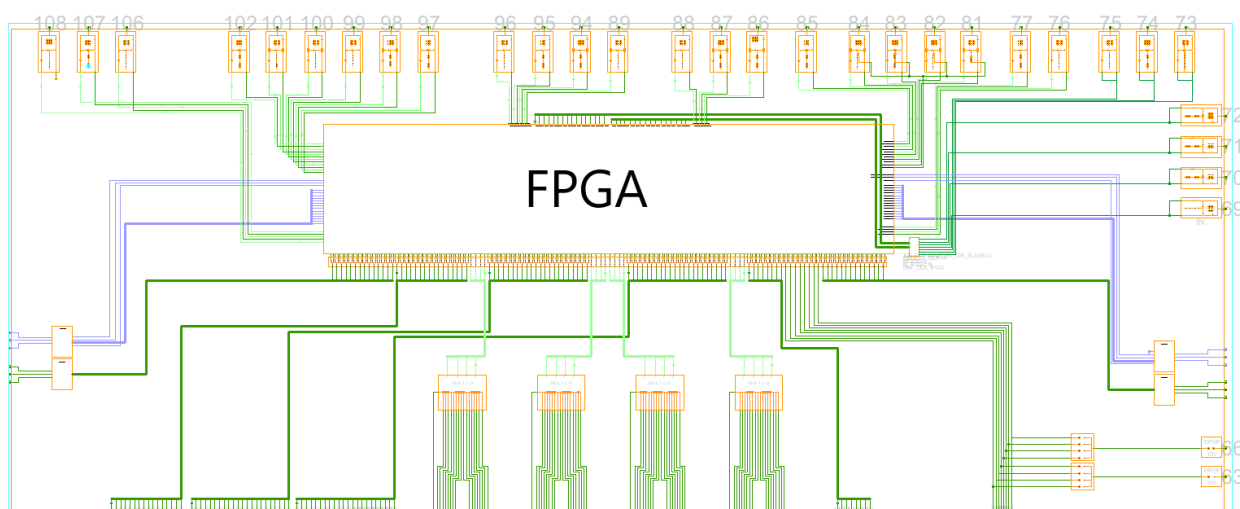


Рисунок 9. Графическое отображение элемента FPGA

В блоке FPGA под ПЛИС присутствует блок, задающий направление цифровых сигналов. Направление определяется вручную путем замыкания ключей.

Если ни один ключ не замкнут, как показано на рисунке 10, то связь «А» подтягивается к «0».



Рисунок 10. Блок, задающий направление цифрового сигнала без замкнутых ключей

В случае, если ключи замкнуты, как показано на рисунке 11, но сигнала не поступает, то вывод «А» будет иметь X-состояние.



а) направление сигнала – в ПЛИС

б) направление сигнала – из ПЛИС

Рисунок 11. Блок, задающий направление цифрового сигнала с замкнутыми ключами

Таблица 1. Таблица назначения выводов блока FPGA

Условное наименование вывода блока FPGA	Последовательное подключение	Назначение вывода
FROM_PAD <1 – 6>		Цифровой вход 102 – 97
TO_PAD <1 – 6>		Цифровой выход 102 – 97
TO_PAD <7>	STARTADC	Управляющий сигнал START ADC 1
TO_PAD <8>	CLKADC	Тактовый сигнал CLK ADC 1
FROM_PAD <9>	READADC	Управляющий сигнал READY ADC 1
FROM_PAD <10 – 23>	DADC <13 – 0>	13-й бит ADC 1 – 0-й бит ADC 1
FROM_PAD_CLK <1>		Вход 107 для глобального тактового сигнала 1 FPGA
TO_PAD_CLK <1>		Выход 107 глобального тактового сигнала 1 FPGA
FROM_PAD_CLK <2>		Вход 106 для глобального тактового сигнала 2 FPGA
TO_PAD_CLK <2>		Выход 106 для глобального тактового сигнала 2 FPGA
FROM_PAD_RST		Вход 108 сброса FPGA
TO_PAD <24 – 37>	DDAC <0 – 13>	0-й бит DAC 1 – 13-й бит DAC 1
FROM_PAD <38 – 53>		Вывод FPGA на аналоговую часть
TO_PAD <38 – 53>		Вывод FPGA на аналоговую часть
TO_PAD <54 – 57>		Управляющие выводы мультиплексора 1 в 16
FROM_PAD <58 – 79>		Вывод FPGA на аналоговую часть
TO_PAD <58 – 79>		Вывод FPGA на аналоговую часть
TO_PAD <80–87>		Управляющие выводы мультиплексора 1 в 16
FROM_PAD <88 – 109>		Вывод FPGA на аналоговую часть

Условное наименование вывода блока FPGA	Последовательное подключение	Назначение вывода
TO_PAD <88 – 109>		Вывод FPGA на аналоговую часть
TO_PAD <110 – 113>		Управляющие выводы мультиплексора 1 в 16
TO_PAD <122 – 125>		Вывод FPGA на аналоговую часть/вывод на драйвер силового ключа 63
TO_PAD <126 – 129>		Вывод FPGA на аналоговую часть/вывод на драйвер силового ключа 66
FROM_PAD <114 – 129>		Вывод FPGA на аналоговую часть
TO_PAD <114 – 121>		Вывод FPGA на аналоговую часть
TO_PAD <130 – 143>	DDAC <13 – 0>	13-й бит DAC 2 – 0-й бит DAC 2
RST_EXT		Вход 73 сброса SPI-интерфейса
START		Вход 74 Start SPI в режиме Master
CLK_EXT		Вход 75 тактового сигнала для блока SPI (возможно внешнее подключение к тактовому сигналу ПЛИС)
SS		Вход 69 сигнала Slave Select (Chip Select) SPI блок в режиме Master output, в режиме Slave input
SCK		Вход 70 SCK-тактового сигнала общения Master и Slave для SPI, в режиме Master output; в режиме Slave input
MOSI		Вход 71 Master output Slave input (SPI), в режиме Master output, в режиме Slave input
MISO		Вход 72 Master input Slave output (SPI), в режиме Master input, в режиме Slave output
FROM_PAD_CLK <4>		Вход 76 глобального тактового сигнала 4 FPGA
TO_PAD_CLK <4>		Выход 76 глобального тактового сигнала 4 FPGA
FROM_PAD_CLK <3>		Вход 77 глобального тактового сигнала 3 FPGA
TO_PAD_CLK <3>		Выход 77 глобального тактового сигнала 3 FPGA
FROM_PAD <144 – 157>	DADC <0 – 13>	0-й бит ADC 2 – 13-й бит ADC 2
FROM_PAD <158>		Управляющий сигнал READY ADC 2
TO_PAD <159>	CLKADC	Тактовый сигнал CLK ADC 2
TO_PAD <160>	STARTADC	Управляющий сигнал START ADC 2
TO_PAD <161>		Вывод не поддерживается
FROM_PAD <162 – 169>		Цифровой вход 81 – 88
TO_PAD <186 – 201>		Выход блока SPI
FROM_PAD <170 – 185>		Вход блока SPI
FROM_PAD <202 – 205>		Цифровой вход 89, 94 – 96
TO_PAD <162>		Цифровой выход 81 – 88
TO_PAD <202 – 205>		Цифровой выход 89, 94 – 96

Важно!

FROM_PAD <XXX> эквивалентен IN_<XXX> на символе блока FPGA.

TO_PAD <XXX> эквивалентен OUT_<XXX> на символе блока FPGA.

ТВ – Блок тестирования

В блок ТВ записывается Verilog код для тестирования и верификации логики ПЛИС. Данный блок используется в процессе моделирования симулятором Icarus Verilog. Рекомендуется результаты моделирования выгружать в файл `sim_results.vcd` для автоматического запуска `gtkwave` (средство просмотра осциллограмм).

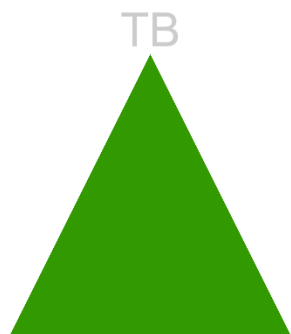


Рисунок 12 Блок тестирования

```

timescale 1ns/1ps

module test;

initial
begin
  $dumpfile("sim_results.vcd");
  $dumpvars(0, test);
  $display("Hello, World");
  $finish ;
end

endmodule

```

Рисунок 13 Пример заполнения блока ТВ

ADC (x2) – 12-ти разрядный АЦП

13 – 14-ти разрядный АЦП применяется опционально.

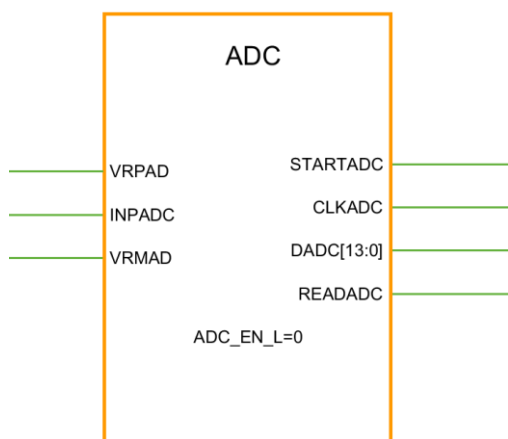


Рисунок 14. Графические изображения АЦП

DAC (x2) – 12-ти разрядный ЦАП

13 – 14-ти разрядный ЦАП применяется опционально.

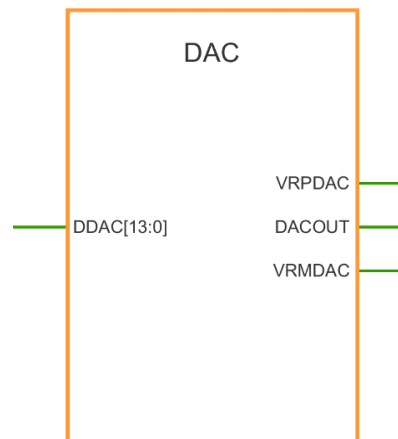


Рисунок 15. Графическое изображение ЦАП

Параметры `ADC_EN_(L/R)` необходимы для включения и выключения АЦП и могут принимать значения:

- «0» – блок выключен;
- «1» – блок включен.

DPOP (x2) – силовые высоковольтные контактные площадки

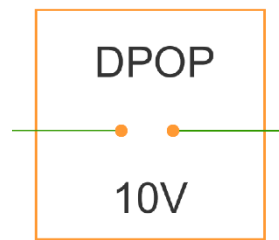


Рисунок 16. Графическое изображение блока DPOP

Для использования силовой высоковольтной контактной площадки необходимо подать на вход цифровые сигналы из выводов ПЛИС, показанных на рисунке 17.

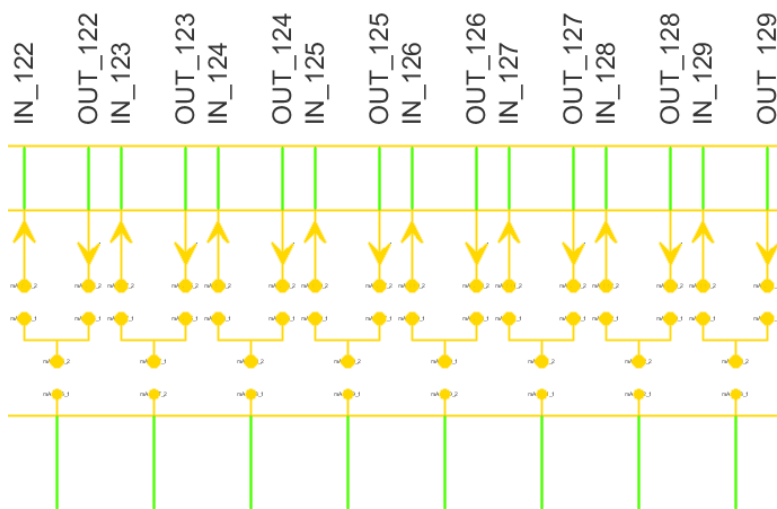


Рисунок 17. Вывод ПЛИС, ведущие к высоковольтным контактным площадкам

После определения выходных площадок ПЛИС требуется замкнуть соответствующий ключ мультиплексора, показанного на рисунке 18, для подачи цифрового сигнала на DPOP.

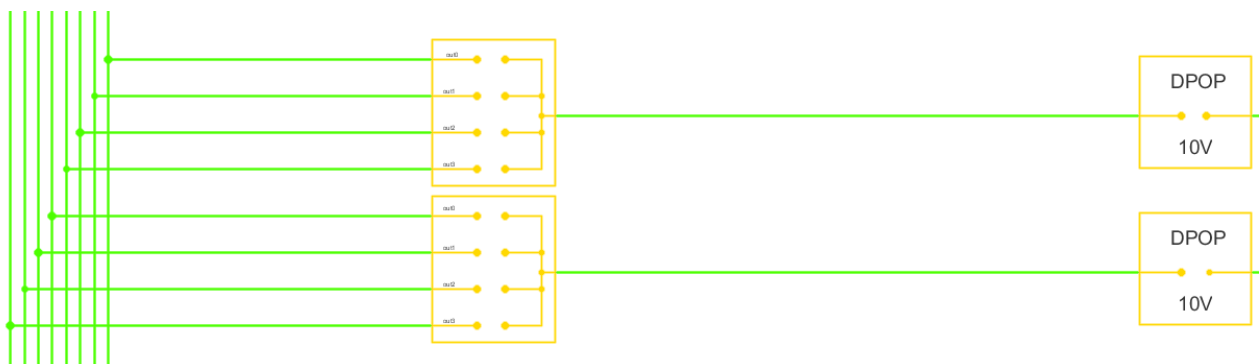


Рисунок 18. Мультиплексоры, определяющие цифровые сигналы на вход высоковольтных контактных площадок

Затем требуется замкнуть ключ включения блока DPOP, изображенный на рисунке 16.

SPI – интерфейс

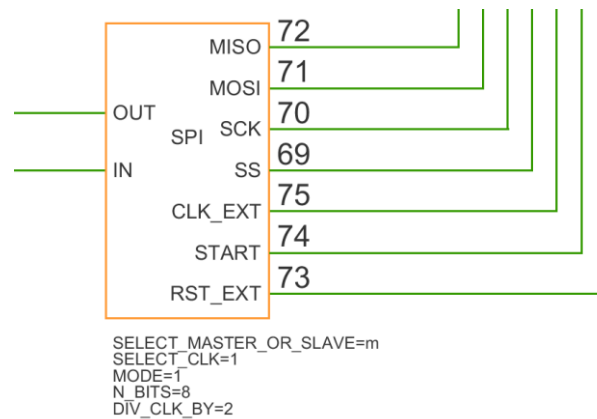
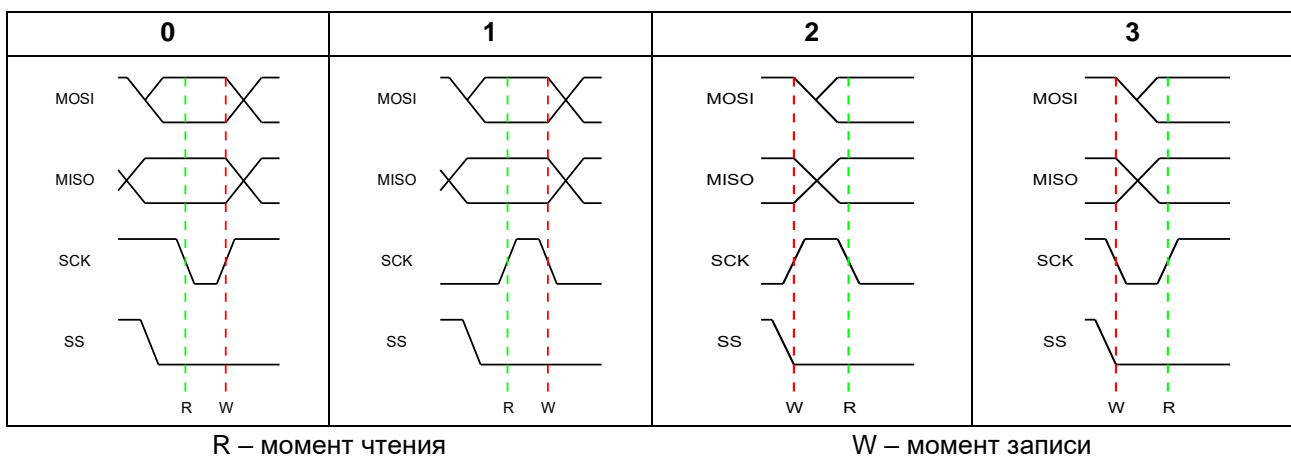


Рисунок 19. Графическое изображение блока SPI

SPI – интерфейс как составная часть блока FPGA имеет конфигурируемые пользователем параметры:

- DIV_CLK_BY– встроенный делитель частоты. Доступные параметры:
 - 2 – деление частоты на 2;
 - 4 – деление частоты на 4;
 - 8 – деление частоты на 8.
- SELECT_MASTER_OR_SLAVE – выбор режима работы SPI - интерфейса. Доступные параметры:
 - m, master – ведущий;
 - s, slave – ведомый;
 - n – отсутствие выбора ведущего или ведомого.
- SELECT_CLK – выбор канала подачи частоты. Доступные параметры:
 - 1 – подача частоты через FPGACLK1 (вывод ПЛИС);
 - 2 – подача частоты через FPGACLK2 (вывод ПЛИС);
 - 3 – подача частоты через FPGACLK3 (вывод ПЛИС);
 - 4 – подача частоты через FPGACLK4 (вывод ПЛИС);
 - ext – подача частоты через внешний вывод CLKEXTSPI.
- MODE – выбор режима работы SPI-интерфейса. Доступные параметры:



- N_BITS – выбор размера посылки. Доступные параметры:
 - 8 – размер посылки - 8 бит;
 - 16 – размер посылки - 16 бит.

MUX1in16 (x4) – мультиплексор 16:1.

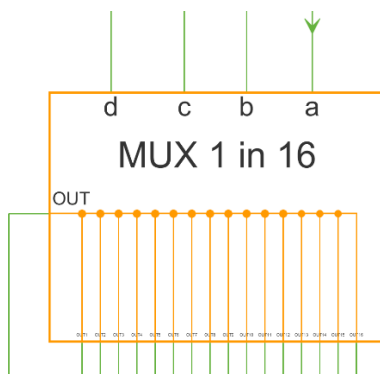


Рисунок 20. Графическое изображение MUX16in1

Таблица 2. Таблица истинности мультиплексора 16 в 1

a	b	c	d	Открытый канал
0	0	0	0	IN1
0	0	0	1	IN2
0	0	1	0	IN3
0	0	1	1	IN4
0	1	0	0	IN5
0	1	0	1	IN6
0	1	1	0	IN7
0	1	1	1	IN8
1	0	0	0	IN9
1	0	0	1	IN10
1	0	1	0	IN11
1	0	1	1	IN12
1	1	0	0	IN13
1	1	0	1	IN14
1	1	1	0	IN15
1	1	1	1	IN16

GAL – глобальная шина

Шина необходима для исключения ключей и паразитных элементов. Шина обладает существенной ёмкостью, что не позволяет использовать её для пропускания высокочастотных сигналов. В своем составе микросхема имеет 8 глобальных шин.

Допустимые параметры для определения портов GAL: 4, 6, 12, 13, 25, 27, 31, 33.

Каждый GAL напрямую связан с определенным каналом каждого из аналоговых мультиплексоров 16:1 (**MUX1in16**) блока FPGA. Соответствие портов GAL с каналами мультиплексора приведены в таблице 3.

Таблица 3. Таблица соответствия портов GAL, соединенных с каналами мультиплексоров напрямую

Порт GAL	Входной канал мультиплексоров
4	1-й канал
6	2-й канал
12	5-й канал
13	6-й канал
25	9-й канал
27	10-й канал
31	13-й канал
33	14-й канал

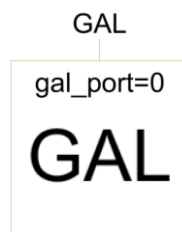


Рисунок 21. Графическое изображение элемента GAL

DIOP (x26) – аналоговые площадки ввода/вывода из цифровой части

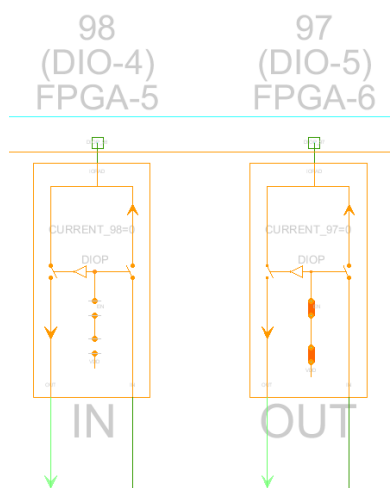


Рисунок 22. Графическое изображение DIOP (DCS_Electric v1.5+)

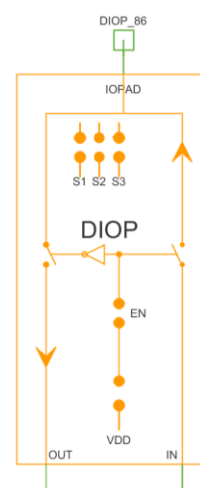


Рисунок 23. Графическое изображение DIOP (DCS_Electric v1.4)

Важно! По умолчанию все площадки DIOP запрограммированы на вход. В случае, если необходимо запрограммировать DIOP на выход, то нужно замкнуть все 5 ключей DIOP.

До версии 1.4 включительно: Ключи S1, S2, S3 необходимы для токовой настройки DIOP. Ключи VDD и EN необходимы для смены направления сигнала со входа на выход.

Версия 1.5+: Токовая настройка DIOP задаётся параметром CURRENT. Возможные значения параметра: от 0 до 7, где 0 – отсутствие тока, 1 – минимальный ток, 7 – максимальный ток. Ключи VDD и EN необходимы для смены направления сигнала со входа на выход. Направление работы DIOP динамически отображается под блоком. IN – площадка является входом в ПЛИС. OUT – площадка является выходом ПЛИС.

DIOP_OR (x4) – аналоговые площадки ввода/вывода из цифровой части с возможностью регулирования направления сигнала с помощью цифровой логики.

Управляющий сигнал поступает на все 4 элемента DIOP_OR одновременно.

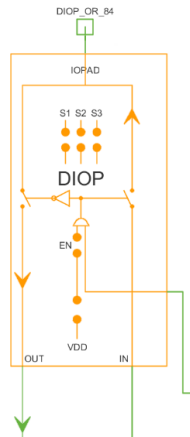


Рисунок 24. Графическое изображение DIOP_OR

VDD	EN	OR	Направление
0	0	0	IN
0	1	0	X
1	0	0	X
1	1	0	OUT
0	0	1	OUT
0	1	1	X
1	0	1	X
1	1	1	OUT

X – некорректное состояние.

Важно! Направление работы определяется не только ключами, но и сигналом из ПЛИС.

INPUT – вход схемы

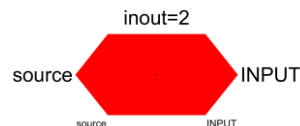


Рисунок 25. Графическое отображение элемента INPUT

source – порт для подключения источника сигнала из библиотеки *symbol* (имитирует внешние входные сигналы);

INPUT – порт входной площадки INPUT для подключения к внутренней схеме.

OUTPUT – прямой выход схемы

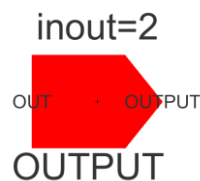


Рисунок 26. Графическое изображение элемента OUTPUT

OUTPUT_ADR_x2 – выход схемы через аналоговый буфер с усилением в два раза

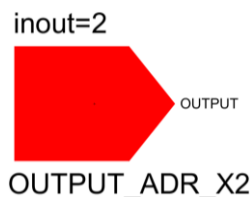


Рисунок 27. Графическое отображение элемента OUTPUT_ADR_x2

OUTPUT_ADR – выход схемы через аналоговый буфер

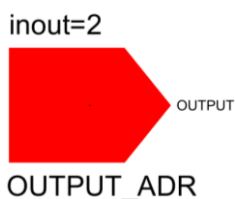


Рисунок 28. Графическое отображение элемента OUTPUT_ADR

OUTPUT_DDR – выход схемы через цифровой буфер

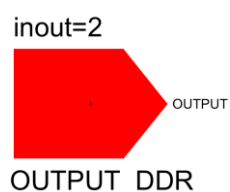


Рисунок 29. Графическое отображение элемента OUTPUT_DDR

Важно! Для подключения внутренней схемы к выходным площадкам OUTPUT_ADR и OUTPUT_DDR необходимо соединить шину с центральным безымянным портом площадки. Вывод OUTPUT необходим для моделирования поведения сигнала после выходного буфера.

FPGA_connector – элемент для связи ПЛИС с динамическими ключами аналоговой части (включая блок PAM)

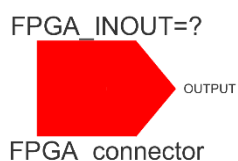


Рисунок 30. Графическое изображение элемента FPGA_connector

Блок FPGA содержит выводы для управления блоками аналоговой части, такими, как PAM, SWITCH. Выводы некоторых аналоговых блоков соединены с определенными выводами блока FPGA без возможности конфигурации. Для автоматического определения доступных на ПЛИС выводов используется блок FPGA_connector.

Рассмотрим работу блока FPGA_connector на примере подключения его к блоку SWITCH:

Блок FPGA_connector подключается к выводу dCtrl. После процесса автоматической трассировки (см. пункт «**Автоматическая трассировка схемы**») параметр FPGA_INOUT примет значение соответствующего вывода ПЛИС, как показано на рисунке 31:

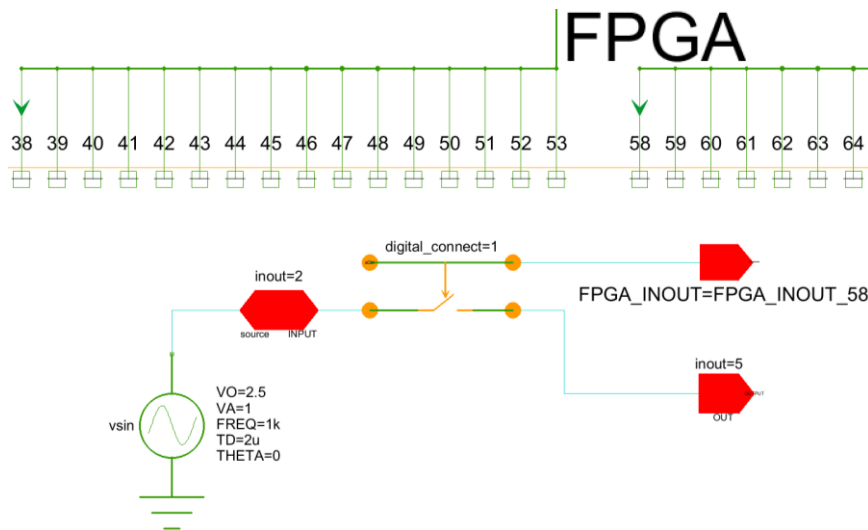


Рисунок 31. Тестовая схема с использованием блока FPGA_connector

Для моделирования необходимо обеспечить соединение блока FPGA_connector с соответствующим выводом блока FPGA, как показано на рисунке 32:

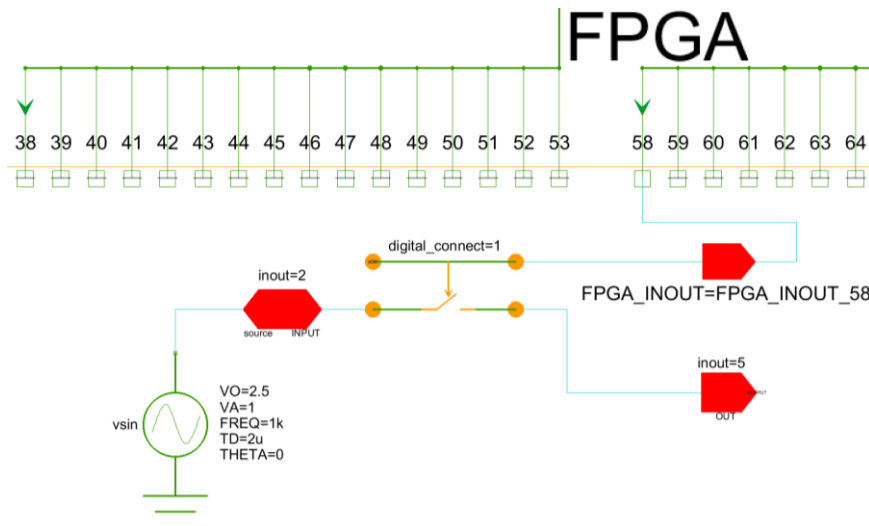


Рисунок 32. Соединение блоков FPGA_connector и FPGA в тестовой схеме

РАМ – программируемый усилительный блок

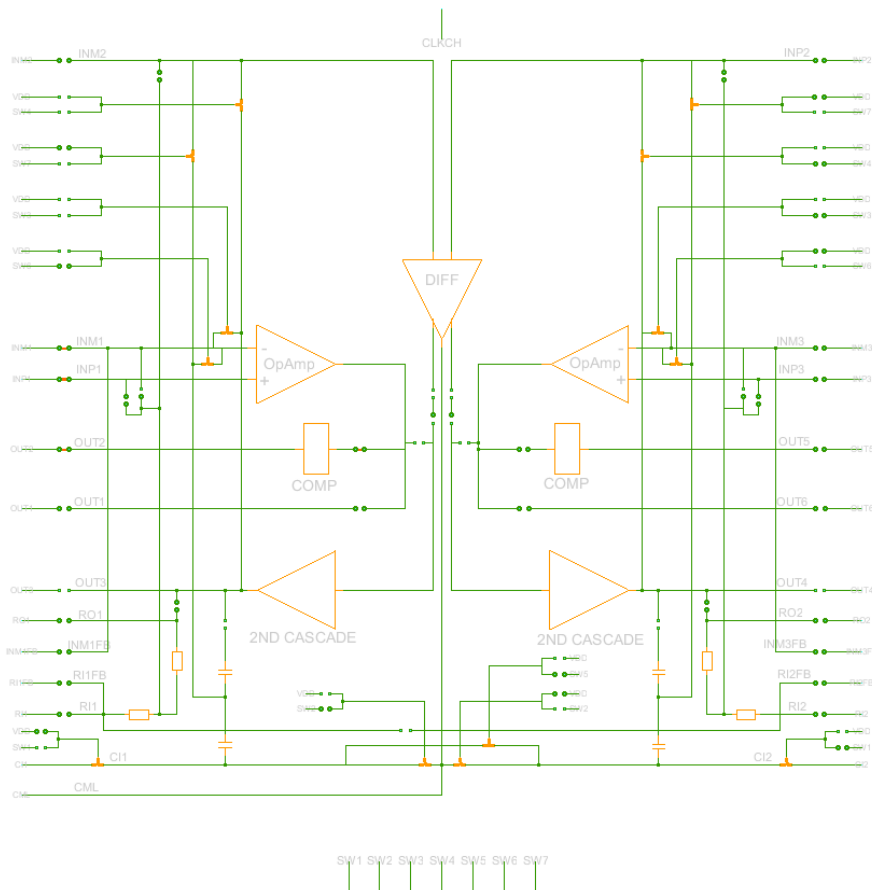


Рисунок 33. Графическое отображение элемента РАМ

Программируемый усилительный блок РАМ и практически все его последующие конфигурации имеют в своем составе конфигурированные резисторы и конденсаторы. Пользователь имеет возможность установки коэффициентов резисторов и конденсаторов для получения требуемого соотношения.

Конфигурируемые резисторы имеют доступный диапазон коэффициентов 1-8. Для удобства, при установке определенного значения коэффициента будут показаны значения сопротивлений, при которых достигается требуемое соотношение.

Конфигурируемые емкости имеют тот же принцип установки значений. Ниже представлена таблица соответствия коэффициентов и значений емкостей.

Таблица 4. Таблица соответствия коэффициента конфигурируемой емкости и ее значения.

Коэффициент	Значение емкости, пФ	Коэффициент	Значение емкости, пФ
1	0,251	9	2,259
2	0,502	10	2,51
3	0,753	11	2,761
4	1,004	12	3,012
5	1,255	13	3,263
6	1,506	14	3,514
7	1,757	15	3,765
8	2,008	16	4,016

PAM_CAP_FULL_DIFF – полностью дифференциальный усилитель с шунтирующими емкостями в обратных связях универсального усилительного блока PAM

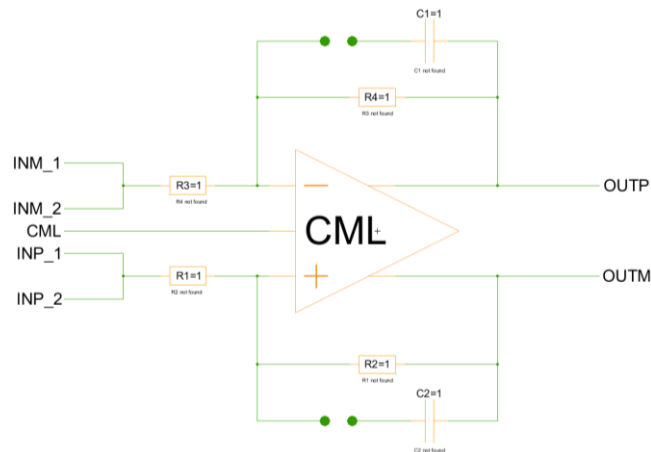


Рисунок 34. Графическое отображение элемента PAM_CAP_FULL_DIFF

PAM_COMP – компаратор на основе универсального усилительного блока PAM.



Рисунок 35 Графическое отображение элемента PAM_COMP

Важно! На основе одного блока PAM можно построить 2 компаратора. PAM_COMP использует только левую часть PAM, выход которого соединён с аналоговой частью трассировки. Для использования правого компаратора, выход которого соединён с цифровой частью следует использовать исходный блок PAM.

PAM_COMP_HYST – компаратор с гистерезисом на основе универсального усилительного блока PAM

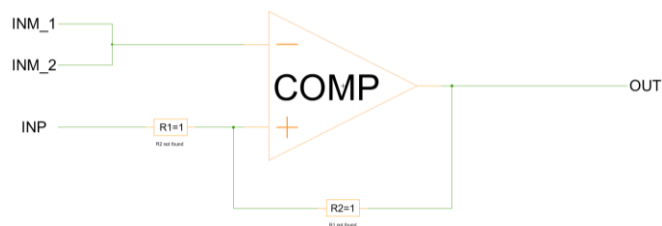


Рисунок 36. Графическое отображение элемента PAM_COMP_HYST

Важно! Блок формируется как единственная возможная конфигурация блока PAM – операционный усилитель с положительной обратной связью.

PAM_INTGRTR – интегратор на основе универсального усилительного блока PAM

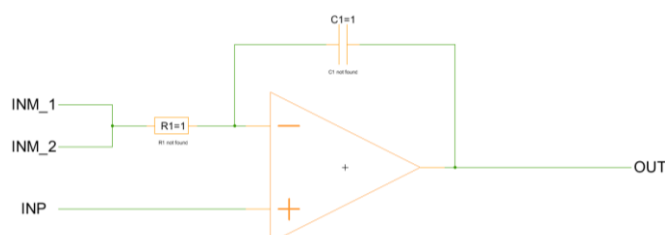


Рисунок 37. Графическое отображение элемента PAM_INTGRTR

PAM_DIFF – полностью дифференциальный ОУ на основе универсального усилительного блока PAM

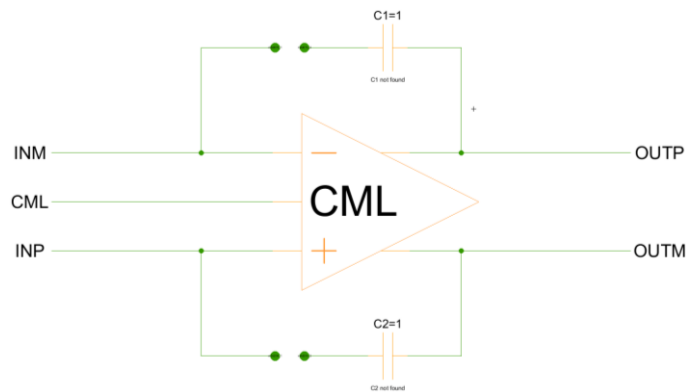


Рисунок 38. Графическое отображение элемента PAM_DIFF

PAM_NEG_CAP_FB – ОУ с ООС и с шунтирующей емкостью на основе универсального усилительного блока PAM

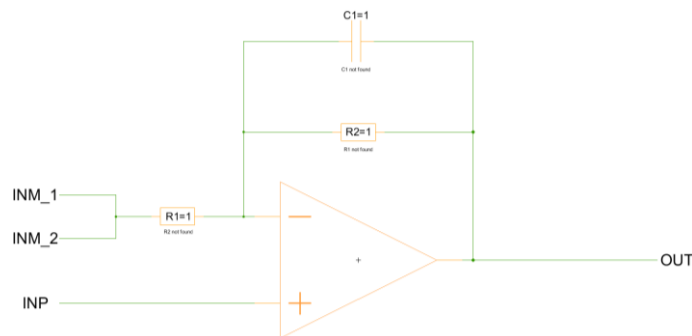


Рисунок 39. Графическое отображение элемента PAM_NEG_CAP_FB

PAM_NEG_FB – ОУ с ООС на основе универсального усилительного блока PAM

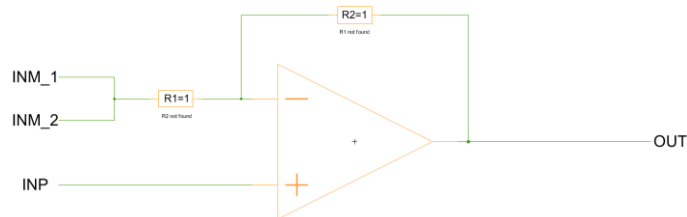


Рисунок 40. Графическое отображение элемента PAM_NEG_FB

PAM_OA – ОУ на основе универсального усилительного блока PAM

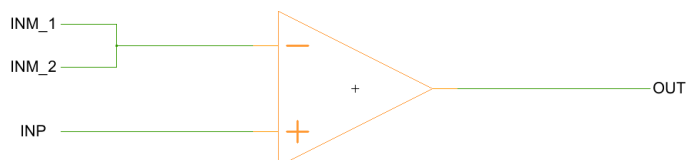


Рисунок 41. Графическое отображение элемента PAM_OA

PAM_OA_shunt – ОУ с шунтирующей емкостью на основе универсального усилительного блока PAM

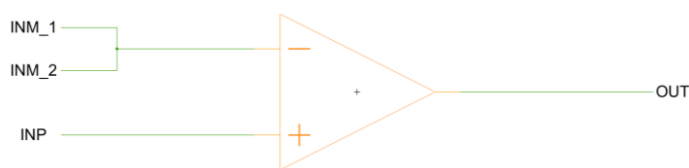


Рисунок 42. Графическое отображение элемента PAM_OA_shunt

RES – программируемый резистор из блока пассивных компонентов

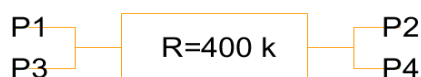


Рисунок 43. Графическое отображение элемента RES

R – значение сопротивления (принимает значения 80, 160, 240, 320, 400 кОм)

SPI_source – источник сигналов (SS, MOSI, CLK, SCK) для управления SPI-интерфейсом в режиме «ведомый»

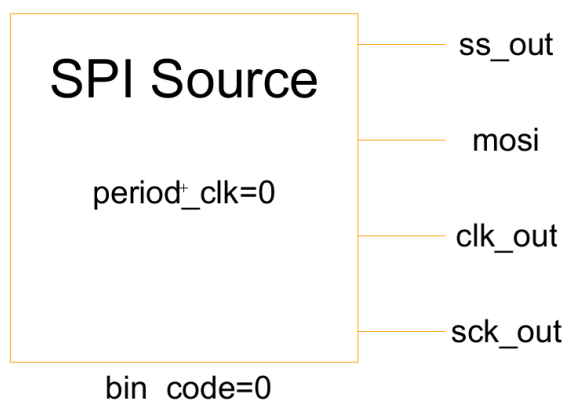


Рисунок 44. Графическое отображение элемента SPI_source

Пример корректной подачи сигнала для управления SPI-интерфейсом в режиме «ведомый»

Источник формирования сигналов SS, SCK, MOSI, CLK имеет следующие порты:

- ss_out – выходной сигнал ss;
- clk_out – выходной сигнал clk;
- sck_out – выходной сигнал sck;
- mosi – выходной сигнал mosi.

Параметры источника:

- bin_code – формирование сигнала mosi в бинарном виде;
- period_clk – Устанавливает период сигнала CLK.

Примечание: Пример бинарного кода для передачи блоку последовательных данных: «10001010010101;10111010011101».

После каждой посылки из 16 бит или команды необходимо ставить знак «;».

Для дублирования команды необходимо использовать знак «:».

Перед моделированием источника SPI_source необходимо сначала запустить скрипт, выполнив команду:

Tools -> Languages -> form_digital_source_in_spi.bsh

SPM – блок свободной конфигурации

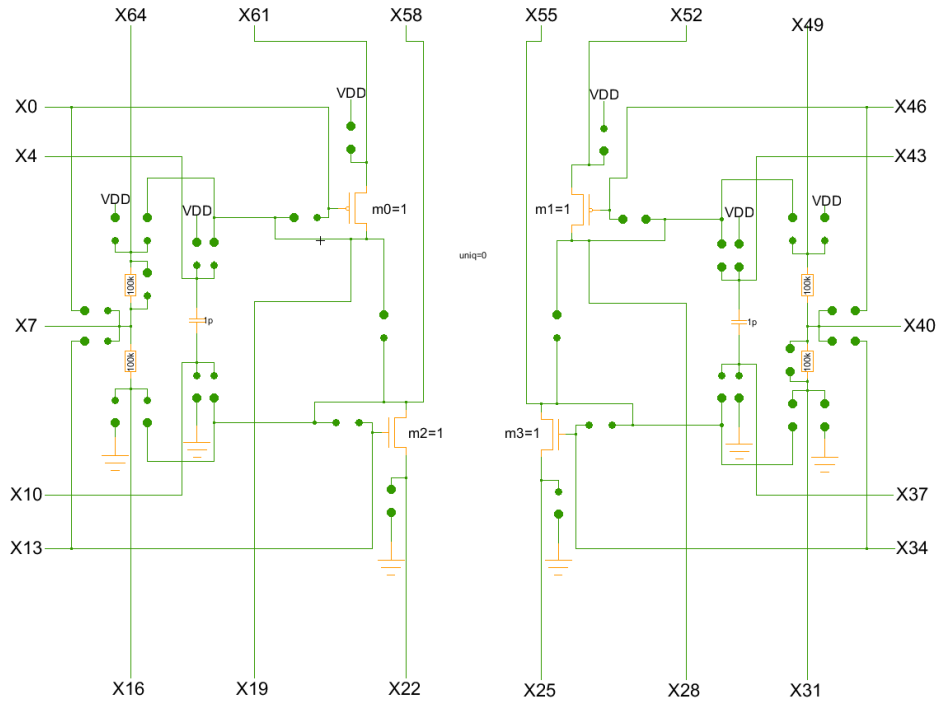
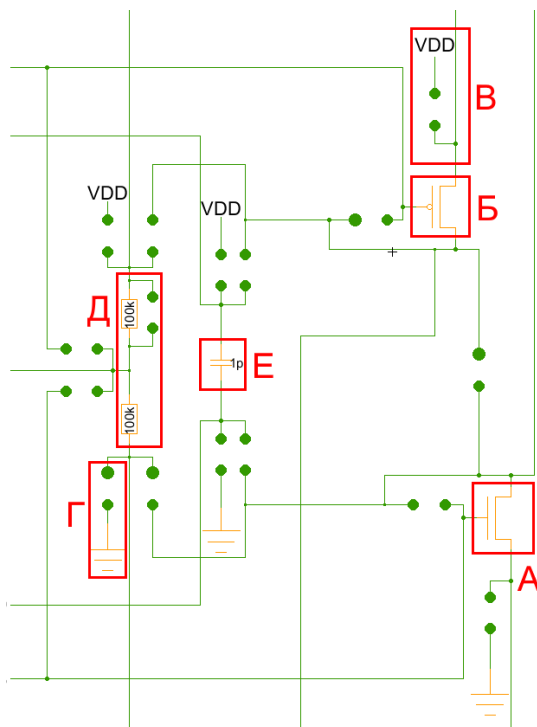


Рисунок 45. Графическое отображение элемента SPM



- А – n-канальный МОП-транзистор;
- Б – p-канальный МОП-транзистор;
- В – вывод подключения к VDDA;
- Г – вывод подключения к VSSA;
- Д – резисторы 100 кОм;
- Е – конденсатор 1 пФ.

Рисунок 46. Схема ячейки блока свободной конфигурации SPM

SWITCH – динамический аналоговый ключ.

Ключом можно управлять как из аналоговой, так и из цифровой части.

Примечание: кроме этого, ключ может быть использован для передачи напрямую сигнала из цифровой части в аналоговую или из аналоговой части в цифровую.

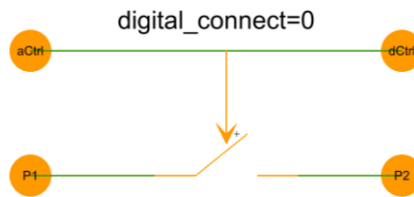


Рисунок 47. Графическое изображение динамического аналогового ключа

digital_connect – параметр, позволяющий включить либо отключить управление ключом из цифровой части. Доступные параметры:

- 0 – управление ключом из цифровой части отключено;
- 1 – управление ключом из цифровой части включено.

extern – блок имитации внешних соединений



Рисунок 48. Графическое изображение элемента extern

Алгоритм автоматической трассировки работает по принципу считывания соседних элементов. При соединении элементов, которые не могут быть соединены внутри микросхемы (например, выходной аналоговый буфер ПАЦИС и вход АЦП) будет выведена ошибка автоматической трассировки. Для моделирования такого соединения необходимо разместить между данными элементами блок extern. Пример использования блока extern приведён на рис. 49.

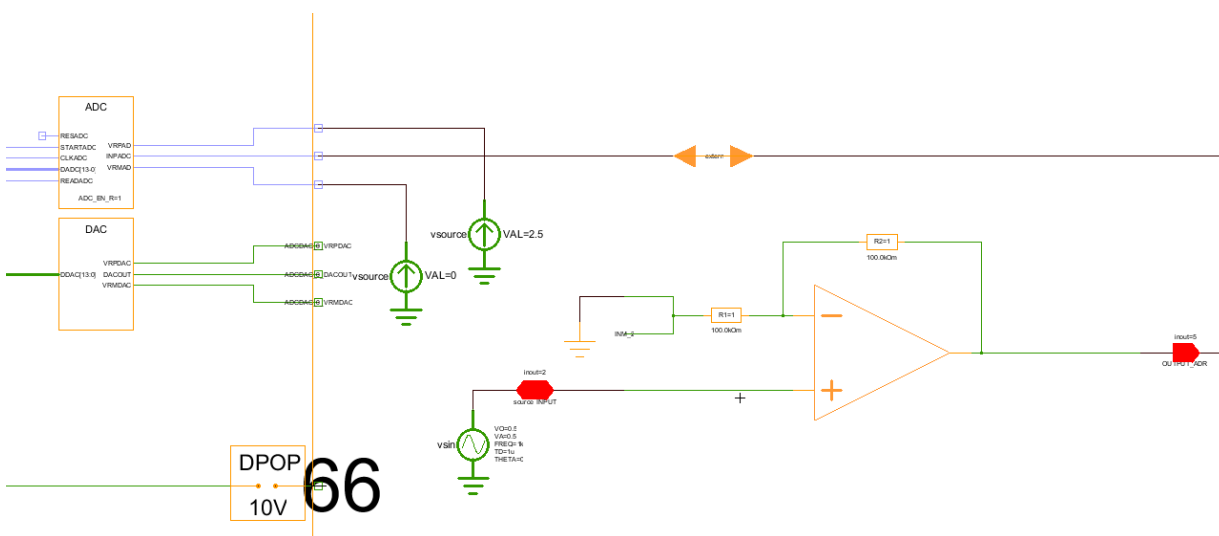


Рисунок 49. Пример применения блока extern

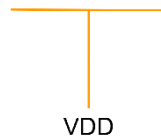
VDD – вывод питания из блока пассивных компонентов

Рисунок 50. Графическое отображение элемента VDD

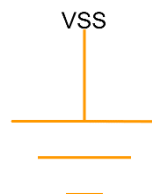
VSS – вывод «земли» из блока пассивных компонентов

Рисунок 51. Графическое отображение элемента VSS

CAP – конденсатор

Емкость 8 пФ из блока пассивных компонентов



Рисунок 52. Графическое отображение элемента CAP

Важно! С помощью источников напряжения следует задать внешние воздействия, для этого нужно перенести компоненты из библиотеки *symbol* в рабочее пространство. Источники напряжения необходимо подключать к порту *source* блока ввода INPUT. Выход блока INPUT нужно коммутировать с тем элементом, на который необходимо подать входной сигнал с используемого источника.

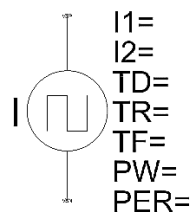
ipulse – источник прямоугольных импульсов тока

Рисунок 53. Графическое отображение элемента ipulse

I1 – значение тока нижнего уровня	TF – время среза
I2 – значение тока верхнего уровня	PW – ширина импульса
TD – время задержки	PER – период
TR – время фронта	

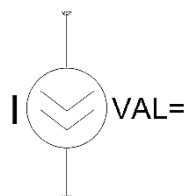
isource – источник постоянного тока

Рисунок 54. Графическое отображение элемента isource

VAL – значение постоянного тока

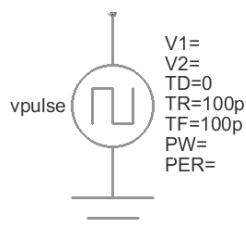
vpulse – источник прямоугольных импульсов относительно VSS

Рисунок 55. Графическое отображение элемента vpulse

V1 – значение напряжения нижнего уровня	TF – время среза
V2 – значение напряжения верхнего уровня	PW – ширина импульса
TD – время задержки	PER – период
TR – время фронта	

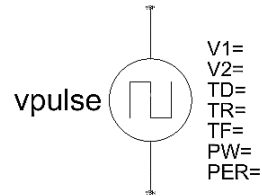
vpulse_clear – источник прямоугольных импульсов

Рисунок 56. Графическое отображение элемента vpulse_clear

V1 – значение напряжения нижнего уровня	TF – время среза
V2 – значение напряжения верхнего уровня	PW – ширина импульса
TD – время задержки	PER – период
TR – время фронта	

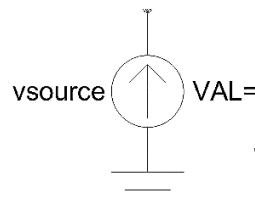
vsorce – источник постоянного напряжения относительно VSS

Рисунок 57. Графическое отображение элемента vsorce

VAL – значение постоянного напряжения

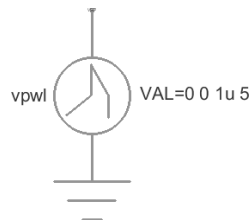
vpwl – источник напряжения, задаваемый по точкам относительно VSS

Рисунок 58. Графическое отображение элемента vpwl

VAL = T1 V1 T2 V2 T3 ...

T1 – время 1 точки	T2 – время 2 точки
V1 – значение напряжения в точке T1	V2 – значение напряжения в точке T2

vpwl_clear – источник напряжения, задаваемый по точкам

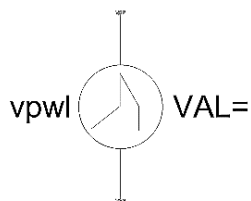


Рисунок 59. Графическое отображение элемента vpwl_clear

VAL = T1 V1 T2 V2 T3 ...

T1 – время 1 точки	T2 – время 2 точки
V1 – значение напряжения в точке T1	V2 – значение напряжения в точке T2

vsin – источник синусоидальных импульсов относительно VSS

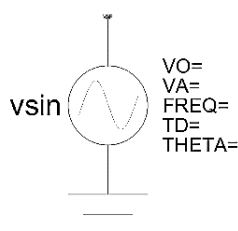


Рисунок 60. Графическое отображение элемента vsin

VO – напряжение смещения	FREQ – частота
VA – амплитуда	TD – время задержки
THETA – коэффициент затухания	

vsource_clear – источник постоянного напряжения

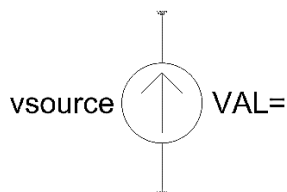


Рисунок 61. Графическое отображение элемента v_source_clear

VAL – значение постоянного напряжения

vsin_clear – источник синусоидальных импульсов

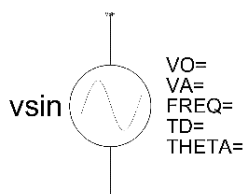



Рисунок 62. Графическое отображение элемента vsin_clear

VO – напряжение смещения	FREQ – частота
VA – амплитуда	TD – время задержки
THETA – коэффициент затухания	

5400TP094_core – блок параметров моделирования


```

tran_control
tstep=100n
tstop=2m
rshunt=10G
LDO_voltage=5
LDO_voltage1V8=1.8
ION_CONFIGURATION=0
ION_TEMP=0
ION_SHIFT=0
HVAOP=0
SAVE=

```

Рисунок 63. Графическое отображение элемента 5400TP094_core

Блок 5400TP094_core отвечает за настройку параметров моделирования и конфигурирования. Он должен обязательно присутствовать в каждой схеме.

tstep – максимальный шаг моделирования (чем меньше, тем точнее моделирование).

tstop – время моделирования.

rshunt – значение сопротивления резистора, добавленного между каждым выводом и «землей» для улучшения сходимости расчетов.

LDO_voltage – напряжение аналогового ядра микросхемы (принимает значение 5,0 В).

LDO_voltage1V8 – напряжение цифрового ядра микросхемы (принимает значение 1,8 В).

ION_CONFIGURATION – параметр, отвечающий за источник опорного напряжения на требуемом выводе. Доступные параметры:

- 0 – отсутствие ИОН на доступных выводах;
- AOP_7 – Напряжение ИОН 1,0 В на выводе AP7;
- PRE – ИОН 1,0 В используется в качестве входа независимого блока VDD1V8OUT, питание которого задает пользователь на выводе VDDAEXT.

ION_TEMP – температурная настройка ИОН (принимает только целые значения от 0 до 8).

ION_SHIFT – настройка ИОН (принимает только целые значения от 0 до 63). Средним значением является 31.

HVAOP – управление высоковольтными усилителями в 4 раза. Доступные параметры:


- 0 – оба высоковольтных усилителя отключены;
- 1 – высоковольтный усилитель 1 включен;
- 2 – высоковольтный усилитель 2 включен;
- 3 – оба высоковольтных усилителя включены.

Для работы высоковольтных усилителей требуется подать питание 9,0 В на вывод V10VA.

SAVE – опция ngspice, которая обеспечивает сохранение только написанных цепей в процессе моделирования. Используется для уменьшения размера файла с результатами моделирования. Для стандартного моделирования поле требуется оставить пустым.

Опция SAVE=all позволяет сохранить все внутренние и внешние цепи. Более подробную информацию можно посмотреть в ngspice manual «15.6.1. SAVE: Name vector(s) to be saved in raw file». Пример использования: «SAVE=inp inm out».

Автоматическая трассировка схемы

Для создания конфигурационной последовательности необходимо выполнить команду: *Tools* → *DCS PDC* → *Autotracing* (кнопка  на панели инструментов).

После завершения процесса автоматической трассировки программа выдаст сообщение:

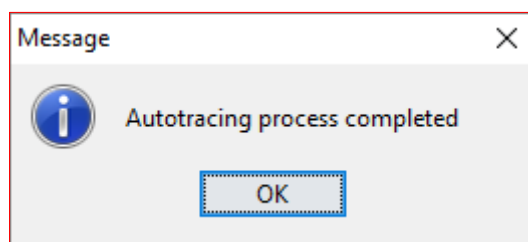


Рисунок 64. Сообщение об успешной трассировке выбранной схемы

В случае получения сообщения об ошибке см. пункт «**Ошибки при построении схем и методы их устранения**».

Всплывающее окно с сообщением об успешной трассировке схемы также означает, что конфигурационная последовательность аналоговой части экспортирована в текстовый файл *analog_config.txt*, находящимися по пути *\DCSElectric\config\analog_config.txt*

Учёт трассировочных ключей и паразитных элементов

В процессе автоматической трассировки происходит размещение элементов в структуру микросхемы 5400TP094 и прокладываются связи между ними. В этот момент учитывается влияние трассировочных ключей и паразитных элементов. Значения их сопротивлений и емкостей записываются во внутреннюю структуру блоков. После автоматической трассировки моделирование результатов выполняется уже с учетом влияния паразитных элементов и ключей.

Сбалансированные цепи

Опция сбалансированных цепей сделана специально для проектирования прецизионных схем. Сбалансированные цепи, принадлежащие одной группе, трассируются с использованием одинакового количества соединительных ключей. Для того, чтобы пометить цепь как сбалансированную, требуется зайти в настройки выбранной связи (нажатием кнопки *q*) и переименовать её в соответствии с шаблоном (*balance<group>name*).

Например, связи «*balance<1>out1*» и «*balance<1>out2*» принадлежат одной группе, а связи «*balance<2>out3*» и «*balance<2>out4*» принадлежат другой группе.

Критические цепи

Иногда при проектировании требуется минимизировать сопротивление и ёмкость некоторых связей. Цепям, которые были помечены пользователем, как критические ставится наивысший приоритет в процессе автоматической трассировки.

Для того, чтобы пометить цепь как критическую, требуется зайти в настройки выбранной связи (нажатием кнопки *q*) и добавить в её название «*critical_*», например «*out*» → «*critical_out*».

Примечание: Опции сбалансированных и критических цепей являются экспериментальными и находятся в процессе отладки.

Ошибки при построении схем и методы их устранения

В момент создания конфигурационной последовательности (*Tools* → *DCS PDC* → *Autotracing*) проводится полная проверка схемы, реализованной пользователем. При некорректном построении схемы или несоответствии внутренней структуре выводится одна из ошибок.

Ошибка количества используемых блоков.

Ошибка возникает при использовании чрезмерного количества блоков в одной схеме. Допустимое количество блоков:

Название блока	Количество	Название блока	Количество
PAM_COMP, PAM_DIFF, PAM_CAP_FULL_DIFF	14	PAM (кроме PAM_COMP, PAM_DIFF, PAM_CAP_FULL_DIFF)	28
ADC	2	VDD	48
DAC	2	VSS	48
FPGA	1	резисторы	24
SPI	1	конденсаторы	24
5400TP094_core	1	SWITCH	24
INPUT, OUTPUT_ADR, OUTPUT_DDR	18	SPM	8

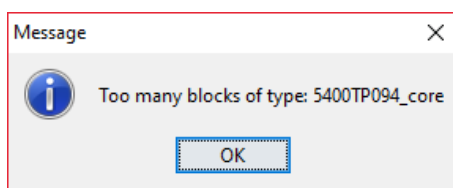


Рисунок 65. Ошибка количества используемых блоков

Решение: уменьшить количество блоков, указанных в тексте ошибки.

Ошибка отсутствия названия основного модуля в параметрах конфигурации Fpga settings.



Ошибка возникает при отсутствии названия модуля в параметрах конфигурации *Fpga settings* и последующем выполнении команды *Tools* → *DCS scripts* → *Verilog synthesis*, либо нажатии кнопки  на панели инструментов.



Рисунок 66. Сообщение об отсутствии названия основного модуля в параметрах конфигурации FPGA_settings

Решение: выполнить команду *Tools* → *DCS PDC* → *Fpga settings*, либо нажать кнопку  на панели инструментов и заполнить поле.

Ошибка установки номера входа/выхода

Ошибка возникает при неверном численном значении входа/выхода схемы в блоках *INPUT*, *OUTPUT_ADR*, *OUTPUT_DDR*.

Допустимые значения: 2, 5, 11, 26, 32, 35, 38, 40, 41, 45, 49, 50, 131, 132, 136, 140, 141, 143.

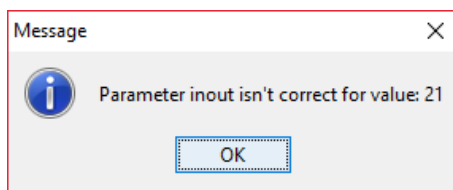


Рисунок 67. Сообщение об использовании некорректного вывода микросхемы

Решение: установить допустимые значения номера входа/выхода.

Отсутствие блока FPGA

Ошибка возникает при отсутствии в рабочем пространстве блока FPGA.

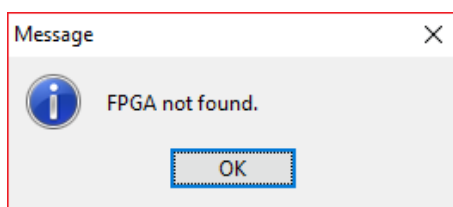


Рисунок 68. Сообщение об отсутствии блока FPGA в схеме

Решение: вынести в рабочее поле блок FPGA из библиотеки *symbol*.

Моделирование аналоговой части

После создания первоначальной схемы имеется возможность провести моделирование для подтверждения правильности работы.

1) Установить параметры источников напряжения.

Для этого дважды нажать на параметр левой кнопкой мыши и вписать значение. Значения параметра вводится без указания единиц измерения. Чтобы ввести десятичную приставку, используются следующие обозначения: фемто-f, пико-p, нано-n, микро-u, милли-m, кило-K, мега-Mег, гига-G, тера-T.

2) Установить параметры моделирования.

Необходимо перенести из библиотеки *symbol* блок «5400TP094_core». Чтобы перенести блок в рабочее пространство нажмите левой кнопкой мыши на нужный блок и, не отпуская кнопку, перетащите в рабочее пространство.

3) Обозначить выводы, которые необходимо контролировать (INP, CML, INM и т.д.).

Для обозначения вывода необходимо зайти в его свойства (клавиша «q» или двойное нажатие левой кнопкой мыши по проводу) и в поле «Name» ввести название.

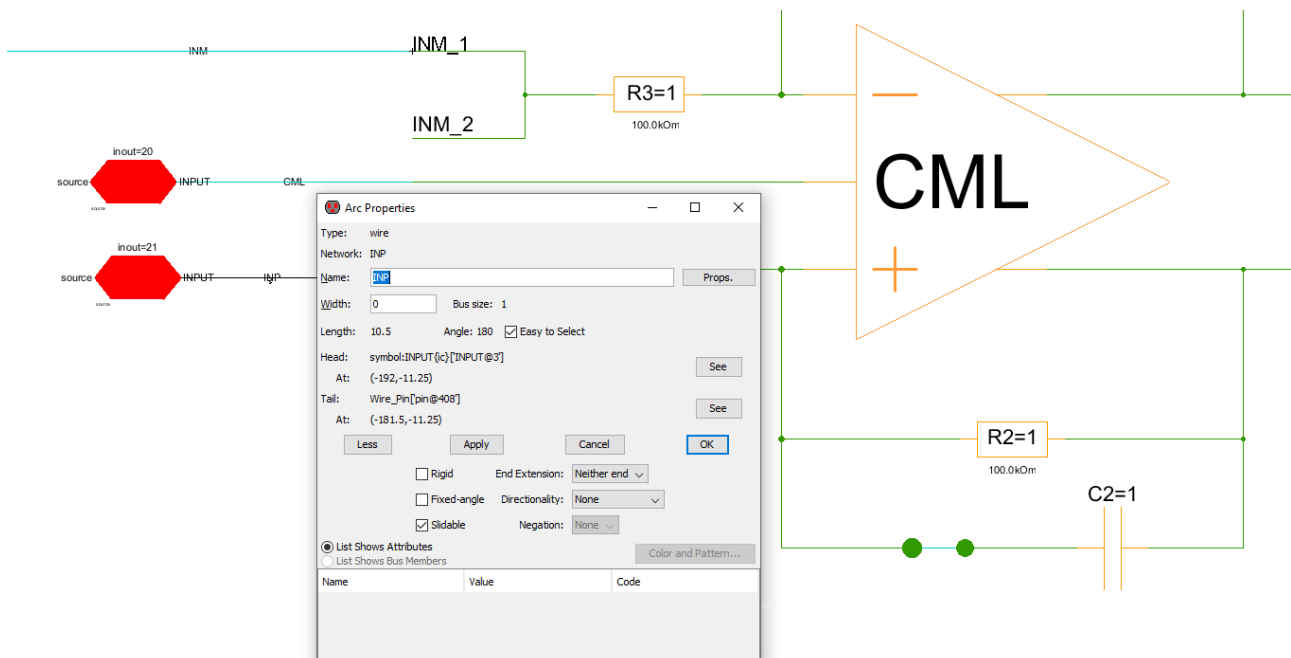


Рисунок 69. Обозначение выводов после построения схемы

4) Сохранить проект (*File* → *Save*).

5) Запустить моделирование:

Tools → *Simulation (Spice)* → *Simulate* (кнопка ► на панели инструментов).

После завершения процесса моделирования откроется окно *LTSpice IV*.

Для вывода результатов на экран выбрать пункт *Plot Settings* → *Add trace* и в появившемся окне указать нужные выводы (INP, CML, INM и т.д.). Выбор проводника осуществляется при помощи поисковой строки «*Only list traces matching*», где вводятся названия выводов. Например, если необходимо посмотреть сигнал на выходе схемы, то в поисковой строке необходимо ввести out и нужный проводник будет обозначаться как «v(out)» (рисунок 70).

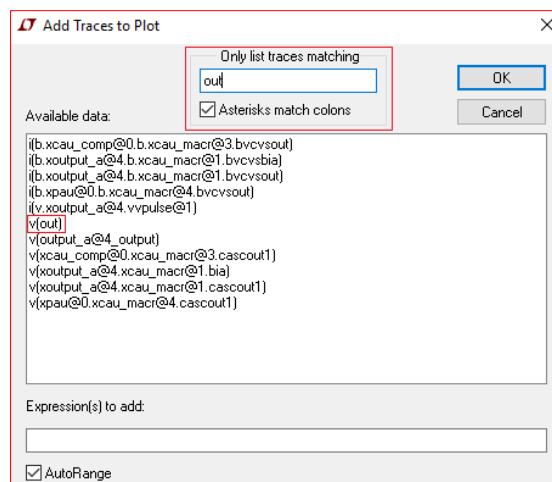


Рисунок 70. Окно вывода результатов моделирования

Для просмотра промежуточных точек (заранее не названных) необходимо зайти в свойства (клавиша «q» или двойное нажатие левой кнопкой мыши по проводу) и найти обозначение в поле «*Network*» (обратите внимание, параметр «*Network*» и «*Name*» могут различаться).

Вывод результатов проводится аналогично при помощи поисковой строки «*Only list traces matching*». В поисковой строке необходимо ввести 498 и нужный проводник будет обозначаться как v(net@498).

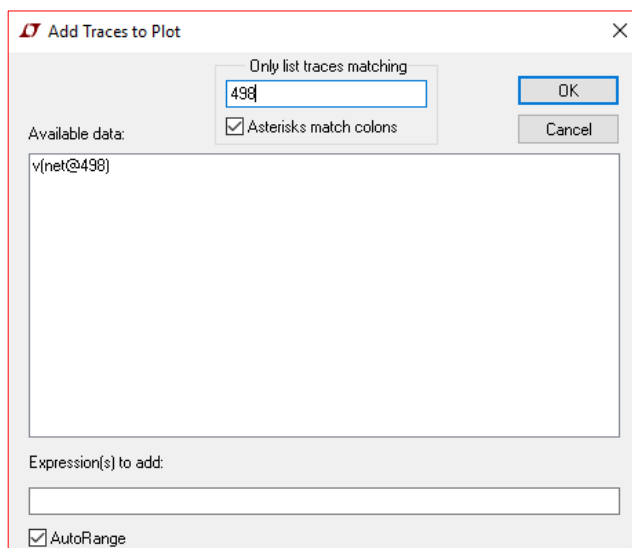


Рисунок 71. Вывод результатов моделирования при помощи поисковой строки

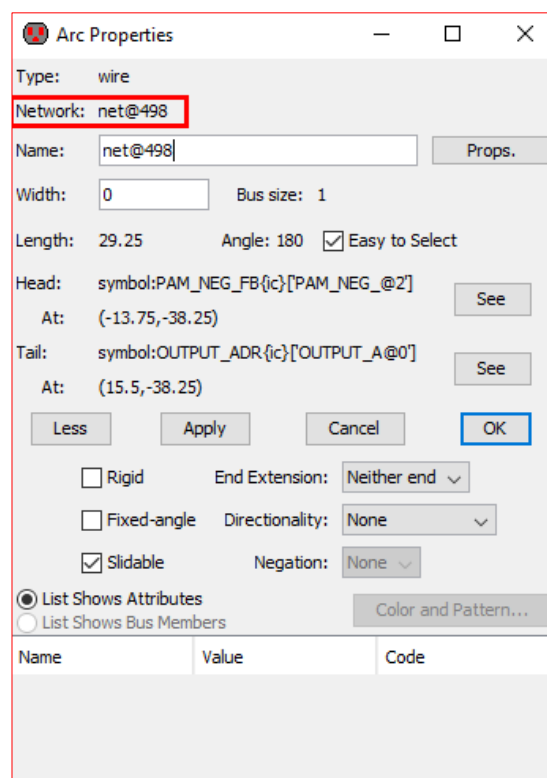


Рисунок 72. Свойство провода для вывода результатов моделирования

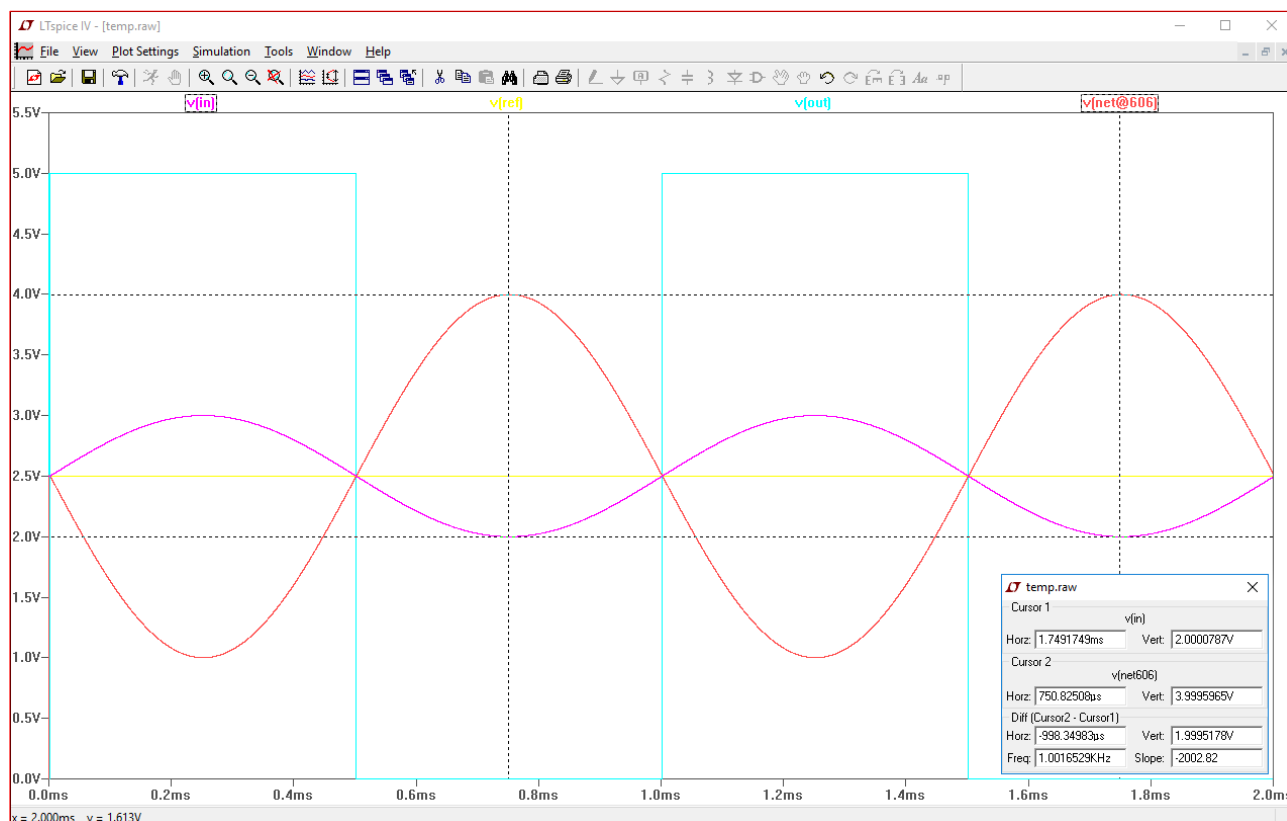


Рисунок 73. Результаты моделирования с использованием маркеров

Некоторые инструменты программы моделирования LTspice IV:

Увеличение интересующей области – нажать левую кнопку мыши, и не отпуская, выделить интересующую область.

Возврат масштаба к начальному – нажать кнопку «*Zoom full extents*» в панели инструментов.

Добавление координатной плоскости: *Plot Settings* → *Add Plot Pane*, также добавить координатную плоскость можно установленной горячей клавишей «*}*».

Вывод маркеров – нажать левой кнопкой мыши по названию проводника.

Удаление маркера – нажать клавишу «*Delete*» и левой кнопкой мыши выбрать название проводника.

Проектирование цифровой части

Проектирование цифровых схем с использованием языка Verilog

Для проектирования цифровых схем с использованием Verilog необходимо вынести блок FPGA в рабочее пространство, войти внутрь блока, выделив его и нажав «1». Далее будет доступно поле для Verilog-кода. Пример показан на рисунке 74.

```

module counter (C, CLR, Q);
input C, CLR;
output [1:0] Q;
reg [1:0] tmp;

always @(posedge C or posedge CLR)
begin
if (CLR)
tmp <= 4'b00;
else
tmp <= tmp + 1'b1;
end
assign Q = tmp;


endmodule

```

Рисунок 74. Пример реализации Verilog-кода для блока FPGA

Важно! При написании Verilog-кода рекомендуется использовать только один фронт тактового сигнала.

Для выхода из поля Verilog-кода нужно нажать кнопку  на панели инструментов.

Затем необходимо задать конфигурацию блока FPGA, выполнив команду *Tools* → *DCS PDC* → *Fpga settings*, либо нажав кнопку  на панели инструментов. В открывшемся окне в графе *Название модуля верхнего уровня* необходимо ввести название главного модуля Verilog-кода.

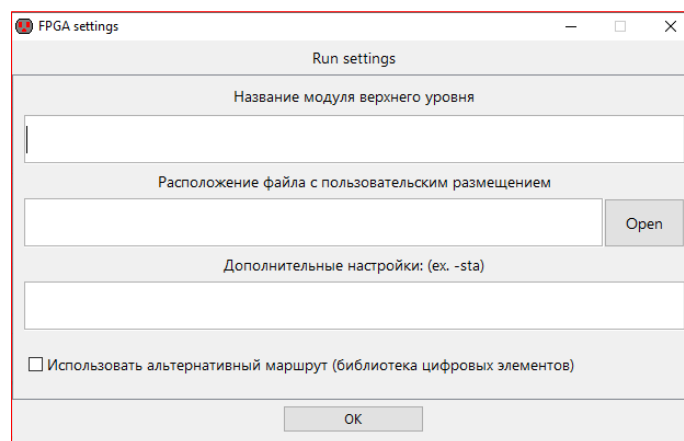


Рисунок 75. Окно конфигурации блока FPGA

При проектировании Verilog можно вручную заполнить tcl-файл, содержащий информацию о всех входах и выходах цифровой схемы. По умолчанию файл находится по пути: *DCSElectric* → *Projects* → *5400TP094* → *simulation* → *Verilog.inout.tcl*.

Пример заполнения файла показан на рисунке 76.

```

1  ###Inout file
2  □ set xc(inout) {
3      { din[0] FROM_PAD1 }
4      { din[1] FROM_PAD2 }
5      { din[2] FROM_PAD3 }
6      { din[3] FROM_PAD4 }
7      { din[4] FROM_PAD5 }
8      { din[5] FROM_PAD6 }
9  }
10 □ set xc(inout_type) {
11     { din[0] xa_ib a }
12     { din[1] xa_ib a }
13     { din[2] xa_ib a }
14     { din[3] xa_ib a }
15     { din[4] xa_ib a }
16     { din[5] xa_ib a }
17 }
18 ###END
19

```

Рисунок 76. Пример заполнения файла с информацией о входах и выходах цифровой схемы

В DCSElectric v1.2 и выше можно использовать планировщик выводов (pinplanner) для автоматического создания файла с описанием выводов. Для этого необходимо запустить планировщик, выполнив команду *Tools → DCS PDC → XPlace pinplanner*. В случае наличия сформированного файла Verilog.v в составе проекта, он будет автоматически подгружен в планировщик. В ином случае требуется выбрать исходный Verilog-файл командой *File -> Open circuit*.

Важно: Каждая цифровая площадка обладает 4-мя именами. Например:

- 1) 102 – номер вывода микросхемы;
- 2) (DIO-0) – название вывода микросхемы (цифровой площадки);
- 3) IN_1/OUT_1 – название порта, соединяющего блок FPGA и DIO;
- 4) FROM_PAD1/TO_PAD1 – порт входа/выхода внутреннего блока FPGA;

В планировщике используются порты входа/выхода блока FPGA т.е. при выборе порта [FROM_PAD1 | TO_PAD1] сигнал идёт к цифровой площадке DIO-0, которая выходит на вывод микросхемы с номером 102.

Примечание: для выбора площадки тактового сигнала нужно в меню I/O group вместо PAD выбрать IOB_CLK.

Для формирования .inout.tcl файла требуется выполнить команду *File -> export inout.tcl*.

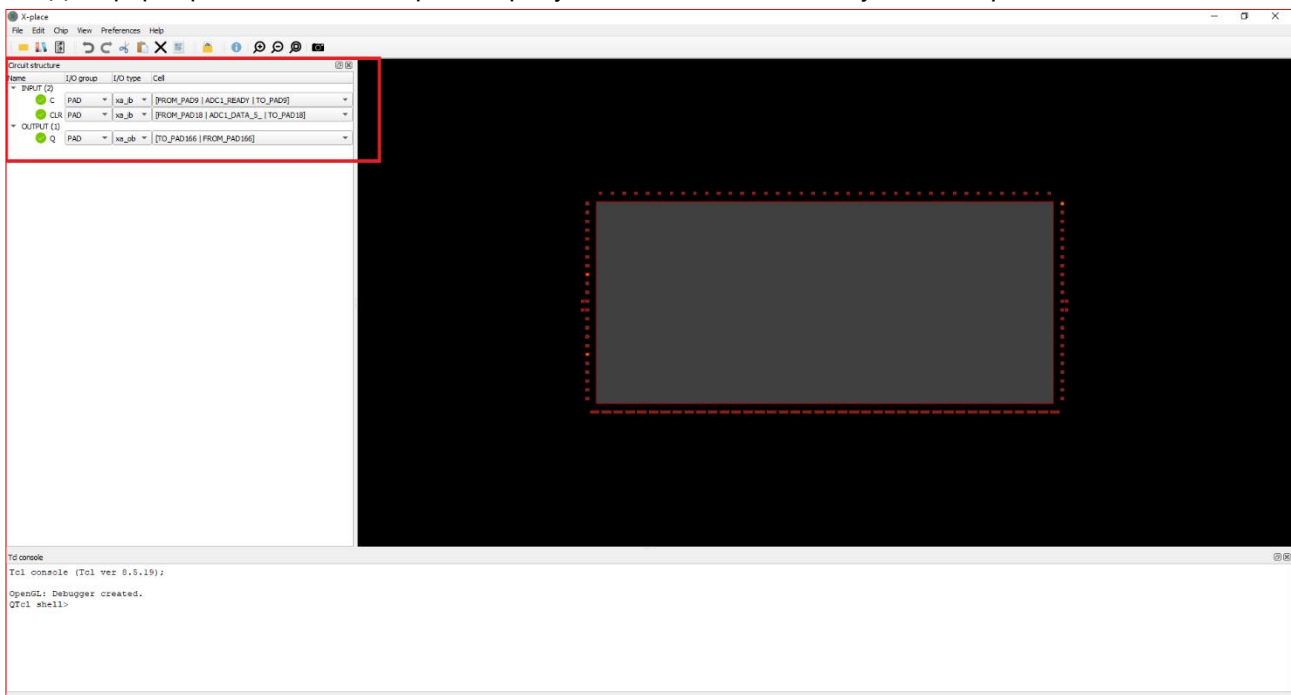

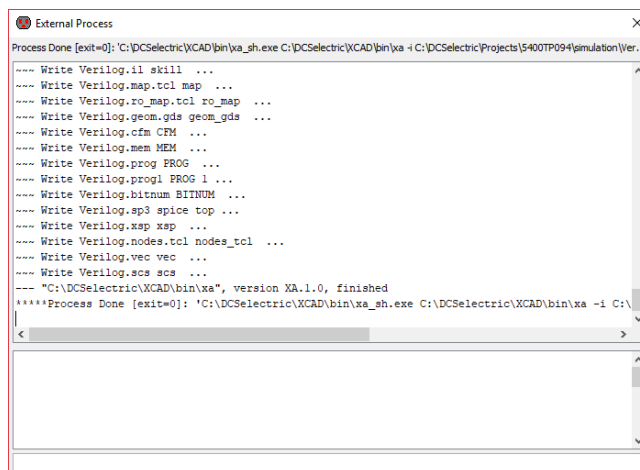


Рисунок 77. Работа с планировщиком выводов

Для использования полученного файла размещения нужно ввести путь к файлу `name_of_file.inout.tcl` в поле «Расположение файла с пользовательским размещением» в окне `FPGA settings`.

Далее необходимо синтезировать автоматически сформированный Verilog-код, выполнив команду `Tools → DCS PDC → Verilog synthesis`, либо нажав кнопку  на панели инструментов. После успешного синтеза в окне `External Process` будет соответствующее уведомление об окончании процесса, как показано на рисунке 78 (Все ошибки и предупреждения выводятся в нижней части окна `External Process`)



```

External Process
Process Done [exit=0]: 'C:\DCSElectric\XCAD\bin\xa_sh.exe C:\DCSElectric\XCAD\bin\xa -i C:\DCSElectric\Projects\5400TP094\simulation\Ver...
---- Write Verilog.il skill ...
---- Write Verilog.map.tcl map ...
---- Write Verilog.ro_map.tcl ro_map ...
---- Write Verilog.geom.gds geom_gds ...
---- Write Verilog.cfm CFM ...
---- Write Verilog.mem MEM ...
---- Write Verilog.prog PROG ...
---- Write Verilog.prog1 PROG 1 ...
---- Write Verilog.bitnum BITNUM ...
---- Write Verilog.sp3 spice top ...
---- Write Verilog.xsp xsp ...
---- Write Verilog.nodes.tcl nodes_tcl ...
---- Write Verilog.vec vec ...
---- Write Verilog.scs scs ...
---- "C:\DCSElectric\XCAD\bin\xa", version XA.1.0, finished
****Process Done [exit=0]: 'C:\DCSElectric\XCAD\bin\xa_sh.exe C:\DCSElectric\XCAD\bin\xa -i C:\
  
```


Рисунок 78. Окно вывода процесса

Окончание синтеза также означает, что конфигурационная последовательность **цифровой части** создана и автоматически экспортирована в текстовый файл `digital_config.txt`, находящимися по пути `.DCSElectric\config\digital_config.txt`.

Смешанное моделирование (долгое)

После создания первоначальной схемы возможно провести моделирование для подтверждения правильности работы.

- 1) Установить параметры моделирования (`5400TP094_core`);
- 2) Обозначить выводы, которые необходимо контролировать;
- 3) Запустить моделирование с помощью команды:

`Tools → Simulation (Spice) → Simulate` (кнопка  на панели инструментов).

После завершения процесса моделирования откроется окно `LTSpice IV`.

Работа с `LTSpice IV` подробно описана в пункте «Моделирование аналоговой части».

Логическое моделирование (v1.5+)

В состав DCSElectric включён симулятор Icarus Verilog для моделирования RTL и gtkwave для просмотра результатов. Программы находятся по пути: `.liverilog/*`. Их можно использовать как отдельно, так и в составе DCSElectric.

Для моделирования с использованием Icarus Verilog нужно:

- 1) Вынести элемент FPGA в рабочее поле схемы из библиотеки `simulation`;
- 2) Зайти внутрь блока (выделить блок, нажать клавишу «1») и добавить Verilog описание, пример Verilog можно посмотреть в описании блока;
- 3) Вынести блок ТВ в рабочее поле схемы из библиотеки `simulation`;

4) Зайти внутрь блока (выделить блок, нажать клавишу «1») и добавить Verilog описание блока тестирования, пример Verilog можно посмотреть в описании блока.

5) Вернуться в общую схему и запустить скрипт *Tools* → *DCS PDC* → *Icarus Verilog*.

6) В консоли будут выводиться сообщения и по завершению моделирования можно будет посмотреть результаты. Для выгрузки результатов рекомендуется использовать конструкцию:

```
$dumpfile("sim_results.vcd");  
$dumpvars(0, test);
```

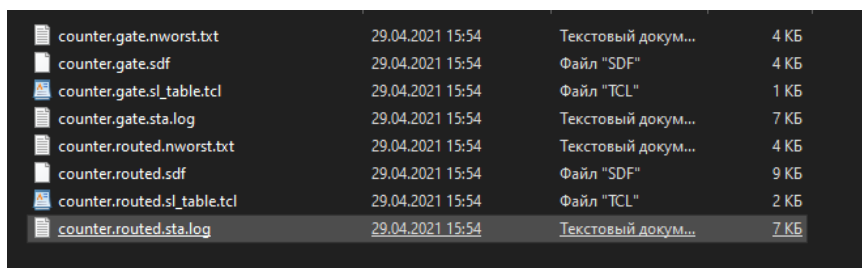
Рисунок 79. Команда для вывода результатов в файл *sim_results.vcd*

Вместо *test* следует использовать название модуля, в котором написан *testbench*. При выгрузке результатов моделирования в файл *sim_results.vcd* окно *gtkwave* для просмотра результатов будет открыто автоматически.

Статический временной анализ (v1.5+)

В окне *FPGA settings* есть поле «Дополнительные настройки», которое позволяет использовать некоторые дополнительные возможности цифрового маршрута, одно из них – статический анализ.

Для наиболее простого использования статического анализа можно добавить в поле дополнительные настройки параметр: «-sta». После команды Verilog Synthesis с этим параметром в папке *simulation* будут автоматически сформированы результаты статического анализа.



counter.gate.nworst.txt	29.04.2021 15:54	Текстовый докум...	4 КБ
counter.gate.sdf	29.04.2021 15:54	Файл "SDF"	4 КБ
counter.gate.sl_table.tcl	29.04.2021 15:54	Файл "TCL"	1 КБ
counter.gate.sta.log	29.04.2021 15:54	Текстовый докум...	7 КБ
counter.routed.nworst.txt	29.04.2021 15:54	Текстовый докум...	4 КБ
counter.routed.sdf	29.04.2021 15:54	Файл "SDF"	9 КБ
counter.routed.sl_table.tcl	29.04.2021 15:54	Файл "TCL"	2 КБ
counter.routed.sta.log	29.04.2021 15:54	Текстовый докум...	7 КБ

Рисунок 80. Результаты статического анализа

Для установки тактовых сигналов, определения временных ограничений и изменения других настроек СВА следует использовать дополнительные опции запуска, которые описаны в разделе «Дополнительные настройки статического временного анализа».

Проектирование цифровых схем при помощи встроенной цифровой библиотеки

- 1) Создать новую схему (правой кнопкой мыши по библиотеке *simulation* → *Create New Cell*)
- 2) В открывшемся окне в поле *Name* ввести название схемы, в поле *View* выбрать *schematic*.

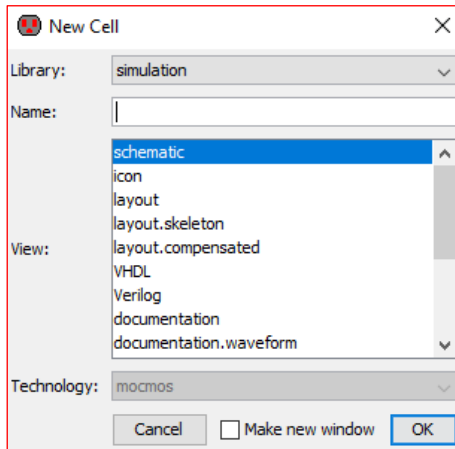


Рисунок 81. Окно создания новой схемы

Блоки, из которых проектируется схема, расположены в библиотеке *digital*. Чтобы перенести компоненты из библиотеки в рабочее пространство, нажмите левой кнопкой мыши на нужный блок и, не отпуская кнопку, перетащите в рабочее пространство.

- 3) Собрать схему с помощью элементов из библиотеки *digital*.

Примечание: Источники сигналов для имитации внешних воздействий подключаются из библиотеки *symbol*.

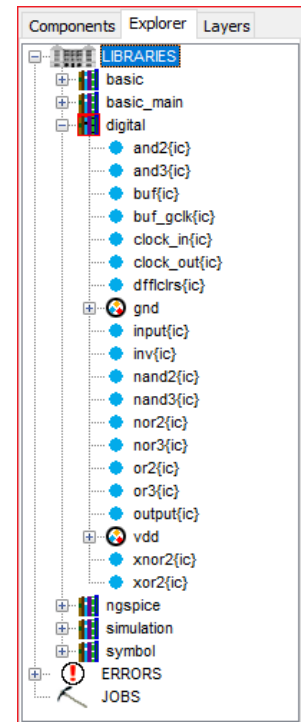


Рисунок 82. Расположение блоков библиотеки *digital*

Описание элементов *digital.jelib*

clock_in – вход тактовой частоты цифровой схемы



Рисунок 83. Графическое отображение элемента *clock_in*

clock_out – выход тактовой частоты цифровой схемы



Рисунок 84. Графическое отображение элемента *clock_out*

input – вход цифровой схемы

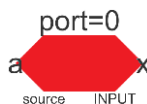


Рисунок 85. Графическое отображение элемента *input*

output – выход цифровой схемы

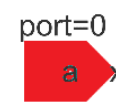


Рисунок 86. Графическое отображение элемента *output*

and2 – логический элемент 2И

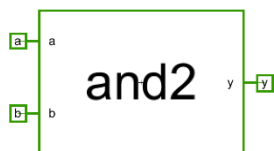


Рисунок 87. Графическое отображение элемента and2

buf – буфер



Рисунок 88. Графическое отображение элемента buf

dffcllr – D-триггер

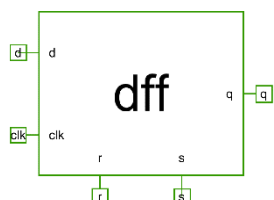


Рисунок 89. Графическое отображение элемента ха_dffcllr

buf_gclk – буфер тактовой частоты

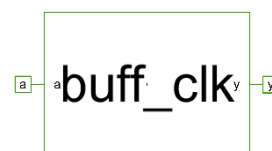


Рисунок 90. Графическое отображение элемента buf_gclk

gnd – цифровая «земля»



Рисунок 91. Графическое отображение элемента gnd

vdd – цифровое питание

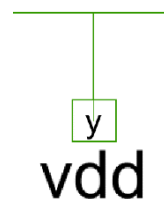


Рисунок 92. Графическое отображение элемента vdd

nor3 – логический элемент 3ИЛИ-НЕ

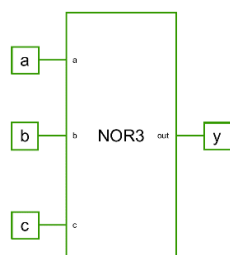


Рисунок 93. Графическое отображение элемента nor3

nand3 – логический элемент 3И-НЕ

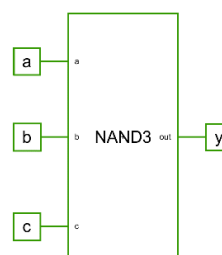


Рисунок 94. Графическое отображение элемента nand3

nand2 – логический элемент 2И-НЕ

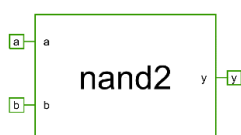


Рисунок 95. Графическое отображение элемента nand2

inv – инвертор



Рисунок 96. Графическое отображение элемента inv

and3 – логический элемент 3И

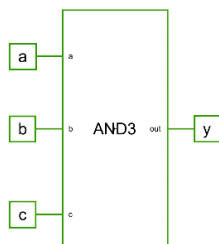


Рисунок 97. Графическое отображение элемента and3

or3 – логический элемент 3ИЛИ

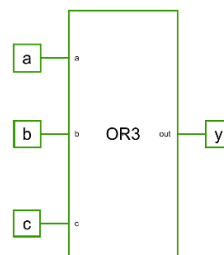


Рисунок 98. Графическое отображение элемента or3

or2 – логический элемент 2ИЛИ

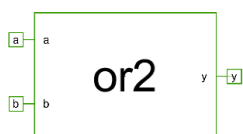


Рисунок 99. Графическое отображение элемента or2

nor2 – логический элемент 2ИЛИ-НЕ

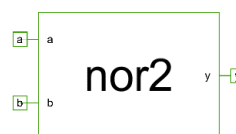


Рисунок 100. Графическое отображение элемента nor2

xnor2 – логический элемент 2 ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ

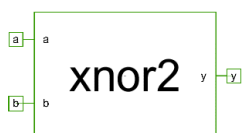


Рисунок 101. Графическое отображение элемента xnor2

xor2 – логический элемент 2 ИСКЛЮЧАЮЩЕЕ ИЛИ

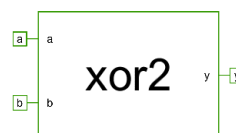


Рисунок 102. Графическое отображение элемента xor2

Важно! Порты ввода и вывода библиотеки *digital* можно назначать вручную на необходимую или более удобную площадку путем изменения параметра «port» (port=25 для вывода IN_25 или OUT_25), либо позволить программе автоматически определить входные и выходные порты, для этого необходимо оставить значение порта по умолчанию – «port=0».

Моделирование и синтез цифровой части

После создания первоначальной схемы возможно провести моделирование для подтверждения правильности работы.

- 1) Установить параметры моделирования.
- 2) Обозначить выходы, которые необходимо контролировать.

Важно!: Все элементы библиотеки *digital* и связанные с ними сигналы являются цифровыми. Для вывода результатов моделирования необходимо перевести сигнал в аналоговый вид. Входные и выходные площадки являются мостами между цифровой и аналоговой частью. Так, на рисунке 99 выходы «set», «reset», «out_a» являются аналоговыми и могут быть выведены в результатах моделирования.

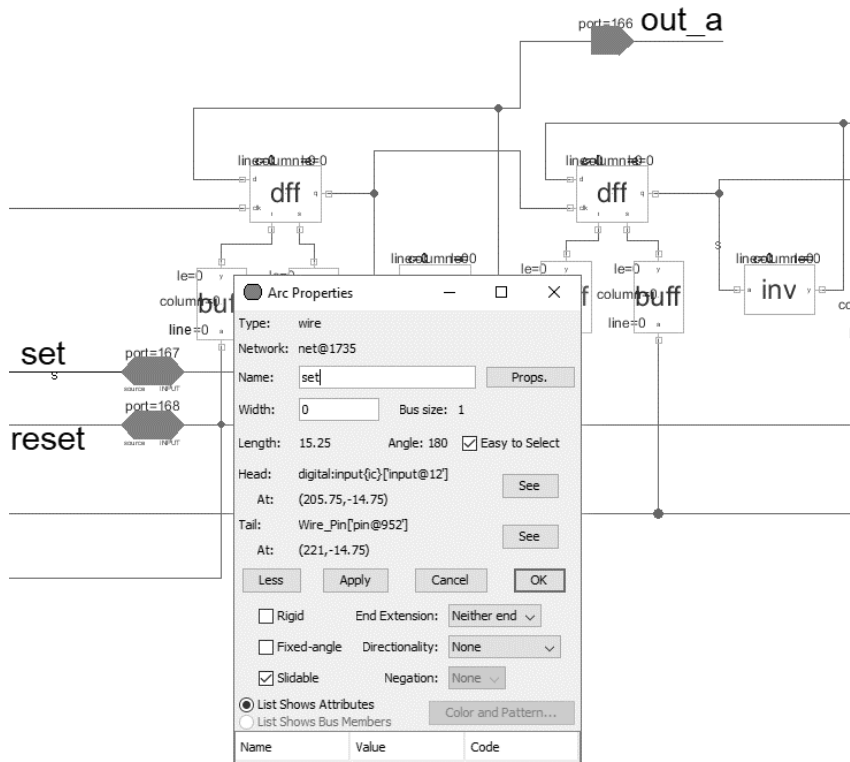




Рисунок 103. Обозначение выводов после построения схемы

- 3) Сохранить проект (*File* → *Save*)
- 4) На этом моменте можно запустить моделирование построенной схемы с помощью команды *Tools* → *Simulation (Spice)* → *Simulate* (кнопка  на панели инструментов). Более подробно в п.8.
- 5) Для синтеза сформировать .tcl-файл

Формирование файла происходит с помощью команды *Tools* → *DCS PDC* → *Create digital circuit*.

Далее выполнить команду *Tools* → *DCS PDC* → *Fpga settings*, либо нажать кнопку  на панели инструментов.

Для использования определенных пользователем выводов ПЛИС в открывшемся окне в пункте «*Расположение файла с пользовательским размещением*» требуется установить галочку и указать путь к.tcl-файлу, содержащему информацию о входах и выходах спроектированной схемы. По умолчанию файл находится по пути *.DCSElectric\Projects\5400TP094\simulation\Verilog.inout.tcl*. Также необходимо поставить галочку в пункте «*Использовать альтернативный маршрут*». Нажать *OK*.

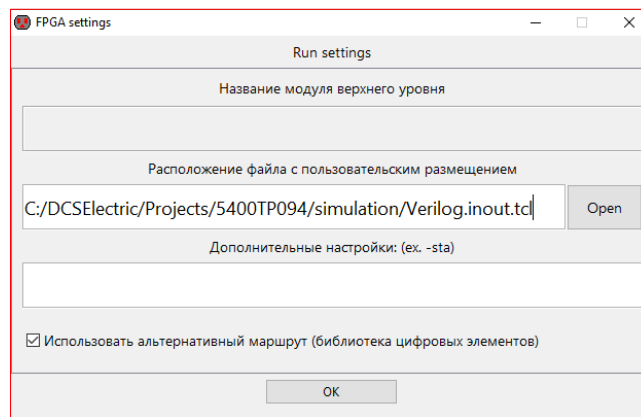



Рисунок 104. Окно конфигурации блока FPGA

6) Создать новую схему (правой кнопкой мыши по библиотеке *simulation* → *Create New Cell*) и вытащить блок FPGA из библиотеки *symbol* в рабочее поле.

7) Синтезировать схему

Выполнить команду *Tools* → *DCS PDC* → *Verilog synthesis*, либо нажать кнопку  на панели инструментов. После успешного синтеза в окне *External Process* будет соответствующее уведомление об окончании процесса, как показано на рисунке 105 (Все ошибки и предупреждения выводятся в нижней части окна *External Process*).

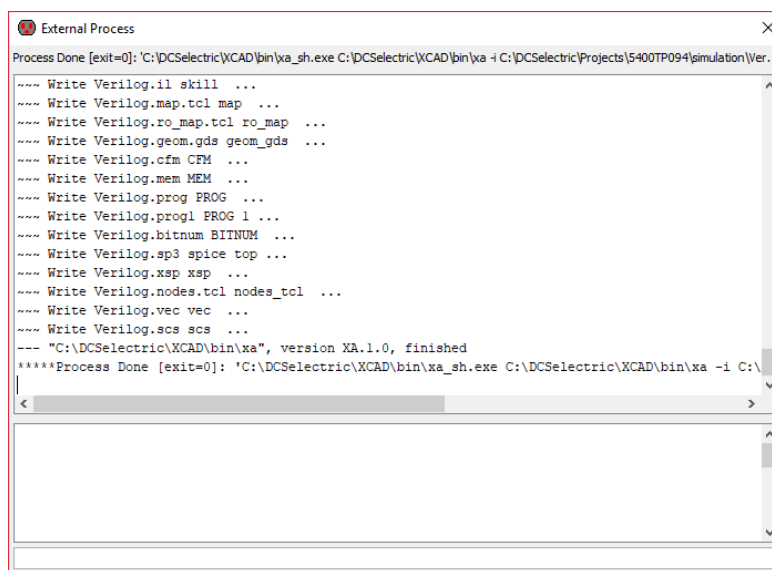



Рисунок 105. Окно вывода процесса

Окончание синтеза также означает, что конфигурационная последовательность **цифровой части** создана и автоматически экспортирована в текстовый файл *digital_config.txt*, находящимися по пути *.\config\digital_config.txt*.

8) Затем рекомендуется выполнить команду *Tools* → *DCS PDC* → *Start XPlace*, либо нажать на кнопку  на панели инструментов для ознакомления с размещением элементов на кристалле ПЛИС.

Далее появится окно *X-place* с реализованной конфигурацией блока FPGA, представленной на рисунке 106:

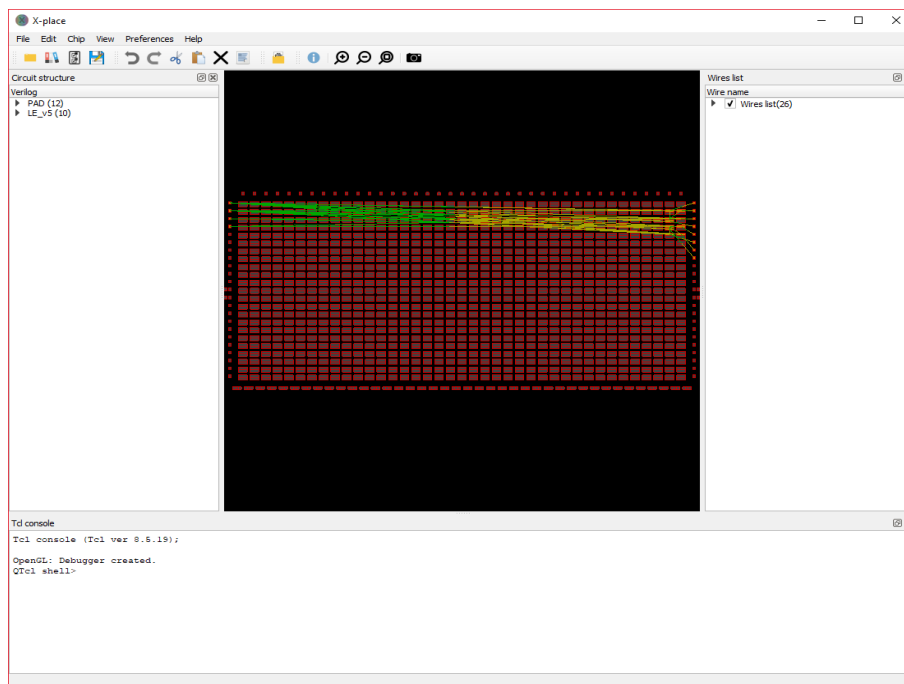



Рисунок 106. Окно X-place

9) Далее следует запустить моделирование с помощью команды

Tools → *Simulation (Spice)* → *Simulate* (кнопка  на панели инструментов). После завершения процесса моделирования откроется окно *LTSpice IV*.

Для вывода результатов на экран требуется выбрать пункт *Plot Settings* → *Add trace* и в появившемся окне указать нужные выходы. Выбор проводника осуществляется при помощи поисковой строки «*Only list traces matching*», где вводятся названия выводов. Для просмотра сигнала на выходе схемы, то в поисковой строке необходимо ввести *out* и нужный проводник будет обозначаться как «*v(out)*».

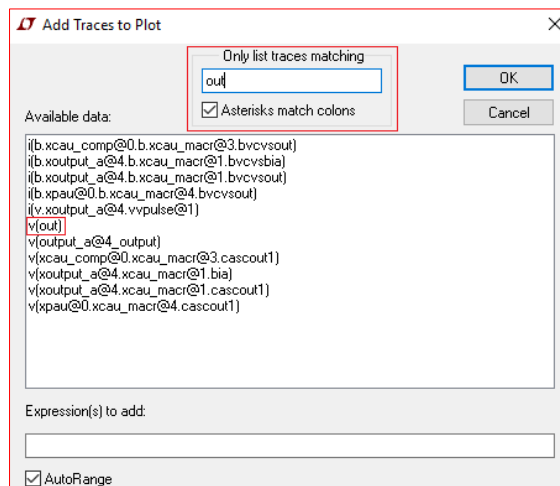


Рисунок 107. Окно вывода результатов моделирования

Для просмотра промежуточных точек (заранее не названных) необходимо зайти в свойства (клавиша «*q*» или двойное нажатие левой кнопкой мыши по проводу) и найти обозначение в поле «*Network*» (обратите внимание, параметр «*Network*» и «*Name*» могут различаться).

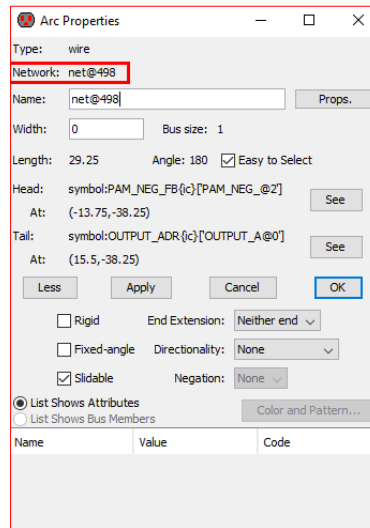


Рисунок 108. Свойство провода для вывода результатов моделирования

Вывод результатов проводится аналогично при помощи поисковой строки «*Only list traces matching*».

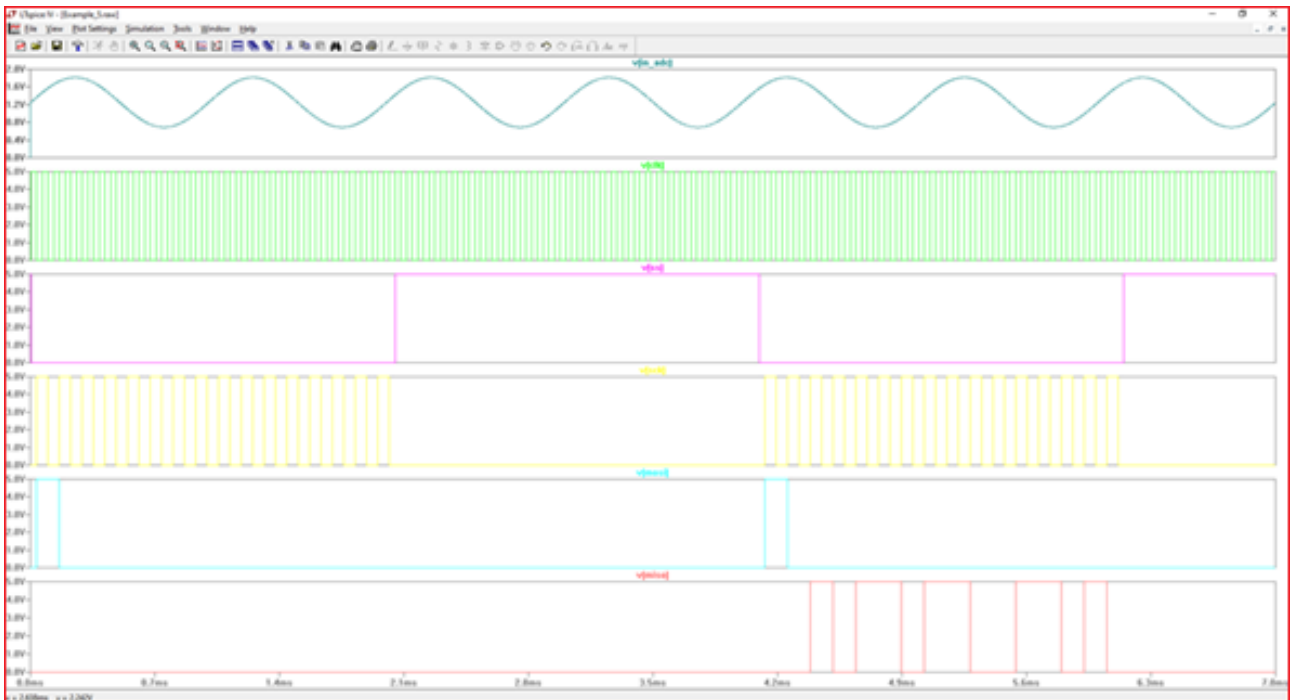


Рисунок 109. Результаты смешанного моделирования

Некоторые инструменты программы моделирования LTspice IV:

Увеличение интересующей области – нажать левую кнопку мыши, и не отпуская, выделить интересующую область.

Возврат масштаба к начальному – нажать кнопку «*Zoom full extents*» в панели инструментов.

Добавление координатной плоскости: *Plot Settings* → *Add Plot Pane*, также добавить координатную плоскость можно установленной горячей клавишей «*}*».

Вывод маркеров – нажать левой кнопкой мыши по названию проводника.

Удаление маркера – нажать клавишу «*Delete*» и левой кнопкой мыши выбрать название проводника.

Интерфейс топологического редактирования X-PLACE

X-place – графический интерфейс, предназначенный для просмотра и редактирования размещения функциональных и периферийных блоков на кристалле ПЛИС.

В обычном случае кристалл, схема и размещение будут автоматически загружены.

Окно программы выглядит следующим образом:

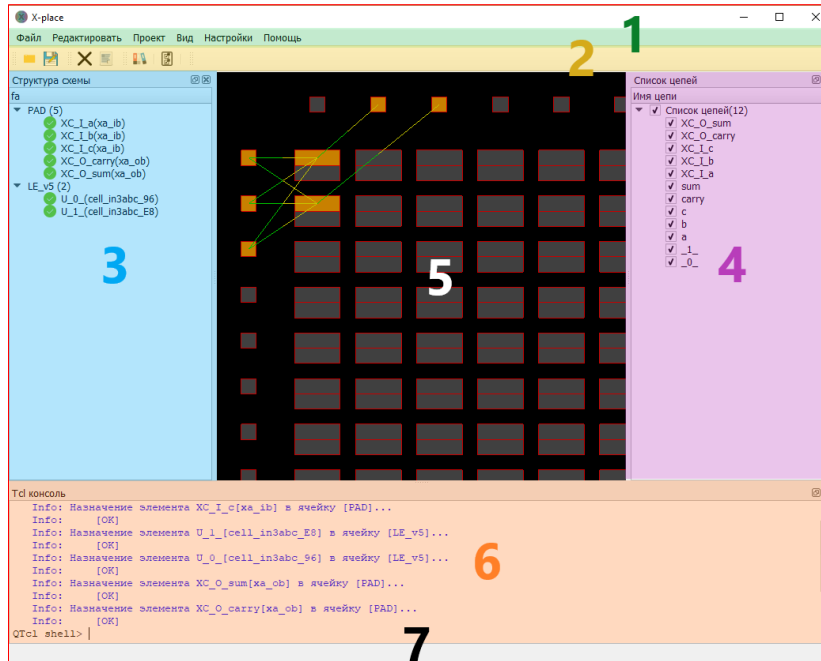


Рисунок 110. Структура главного окна программы

- | | | |
|-------------------------|-------------------------------|------------------------|
| 1 – главное меню | 3 – обозреватель списка цепей | 5 – окно «Tcl консоль» |
| 2 – панели инструментов | 4 – рабочее окно | 6 – статусная строка |

Пример работы программы в ручном режиме

При запуске программы X-place появляется главное окно, в котором пока нет данных ни о кристалле, ни о схеме.

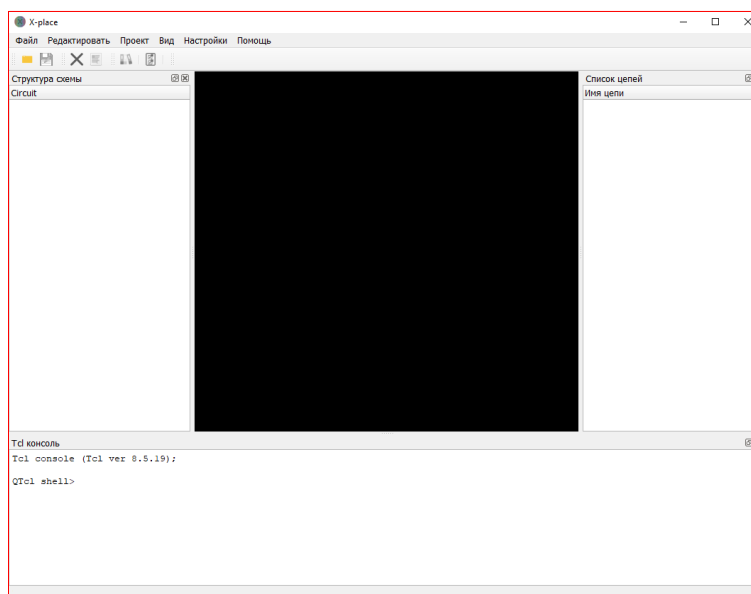


Рисунок 111. Вид главного окна при запуске

Следующим шагом необходимо загрузить кристалл ПЛИС. Сделать это можно следующим образом: открыть меню «Проект» из панели меню и выбрать необходимый кристалл в меню «Кристалл», где кристаллы разбиты по семействам. После этого начнется загрузка кристалла, который в результате будет отображен в графической области главного окна.

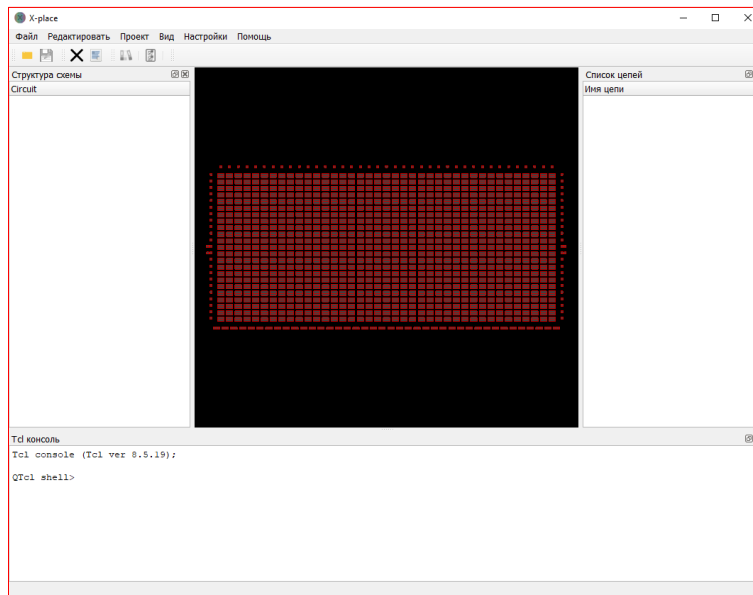


Рисунок 112. Главное окно с загруженным графическим представлением кристалла ПЛИС

Далее следует загрузить схему, которая будет размещена на кристалле. Сделать это можно через меню «Файл – Открыть схему». Откроется окно файлового обозревателя, где следует выбрать файл схемы с расширением «*.tcl». Для примера, на рисунке 113 изображено окно после загрузки файла «fa.syn_xcy.tcl» с описанием схемы «fa» в терминах ХА.

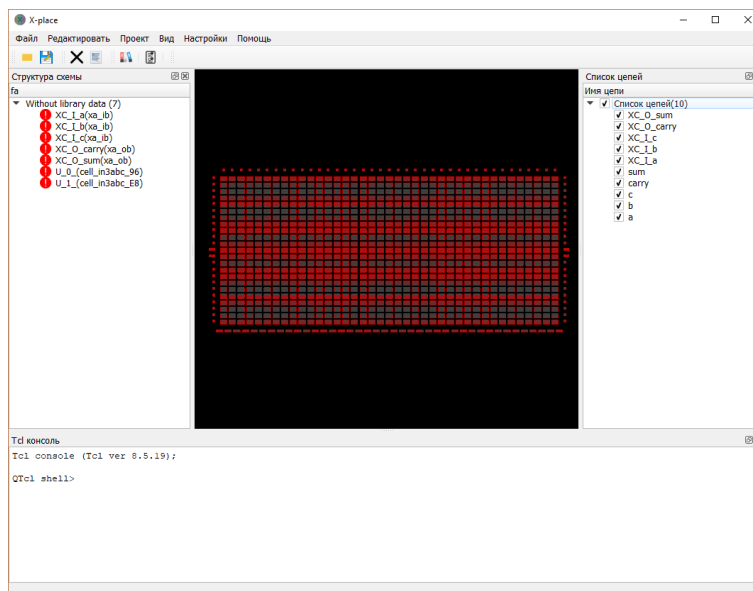


Рисунок 113. Окно программы после загрузки схемы

Настройки отображения

Общие настройки отображения («Вид – Отображаемые объекты») позволяют настроить (включить/отключить) отображение:

- Заполненные ячейки;
- Кристалл (не затрагивает занятые ячейки);
- Цепи (не затрагивает отображение цепей выделенных ячеек);
- Подсвеченные цепи выделенных объектов;
- Настройки режима отображения цепей позволяют выбрать один из двух режимов отображения:
- Прямые соединения – из каждого размещенного элемента выходят соединяющие линии напрямую к элементам, связанным с данным;
- Центрированные соединения – межсоединение представляет собой группу линий с общим центром масс, откуда эти линии выходят по направлению к подсоединенным элементам.
- Настройки режима отображения подсвеченных цепей позволяет выбрать режим подсвечивания соединений выделенных объектов:
- Только подсоединенные элементы – подсвечиваются только те линии, которые связывают данный элемент с другими;
- Полностью подсветить цепь – межсоединение подсвечивается полностью, то есть не только соединения данного элемента с другими, но и все другие направления, куда распространяется это межсоединение.

Полное смешанное моделирование

Программа DCSElectric поддерживает смешанное моделирование всех элементов присутствующих в схеме.

Аналоговая часть моделируется симулятором ngspice.

Для моделирования цифровой части используется xspice, который связывается с аналоговой частью через аналого-цифровые и цифро-аналоговые мосты.

Макромодели ADC, DAC и SPI также написаны на языке симулятора ngspice и поддерживают смешанное моделирование.

Последовательность действий для смешанного моделирования представлена ниже:

- Проектирование аналоговой схемы;
- Проектирование и синтез цифровой схемы;
- Проведение автоматической трассировки;
- Моделирование.

Дополнительные настройки статического временного анализа

Для выполнения статического временного анализа (СВА) проектируемой схемы необходимо добавить опцию **-sta** в команду запуска программы. Для установки тактовых сигналов, определения временных ограничений и изменения других настроек СВА следует использовать дополнительные опции запуска.

Основные опции запуска СВА:

Управление процессом статического временного анализа	
-sta	выполнение статического временного анализа
-top <top_name>	определение имени топового модуля;
-clk <list>	определение списка тактовых сигналов в проекте;
-per <list>	определение значений периодов тактовых сигналов;
-max_delay <value>	определение значения максимальной задержки от входа до выхода;
-sta_num_crit <value>	определение количества критических путей, выведенных в отчет;
-sta_sdc <file_path>	определение полного пути до файла временными ограничениями проекта;
-sta_cap <file_path>	определение полного пути до файла с емкостями цепей;
-sta_spef <file_path>	определение полного пути до файла паразитными элементами;

Результаты СВА генерируются по завершению работы программы в папку *<top_name>.STA.reports*.

Результатом проведенного СВА схемы до этапа трассировки является следующий набор файлов:

<top_name>.gate.nworst.txt – временной отчет для заданного количества наиболее медленных критических путей;

<top_name>.sl_table.tcl – файл с таблицей значений slack для всех элементов логических элементов схемы;

<top_name>.sdf – файл с задержками и заданными временными ограничениями.

Результатом проведенного СВА схемы после этапа трассировки является следующий набор файлов:

<top_name>.routed.nworst.txt – временной отчет для заданного количества наиболее медленных критических путей;

<top_name>.routed.sl_table.tcl – файл с таблицей значений slack для всех элементов логических элементов схемы.

<top_name>.routed.sdf – файл с задержками и заданными временными ограничениями.

Доступные временные ограничения

create_clock	set_multicycle_path
set_clock_latency	set_input_transition
set_clock_uncertainty	set_input_delay
set_clock_transition	set_output_delay
set_propagated_clock	set_max_delay
create_generated_clock	set_min_delay
set_false_path	set_load
set_disable_timing	set_case_analysis

create_clock – добавить к анализу тактовый сигнал, его источник, период, время нарастания и спада сигнала.

create_clock -period <T_per> [-waveform <T_rise> <T_fall>] <source>,

-period <T_per>	<p>задать период задаваемого тактового сигнала <T_per> – время периода тактового сигнала, задается в нс или MGH</p>
-waveform <T_rise> <T_fall>	<p>задать коэффициент заполнения тактового сигнала. По умолчанию, коэффициент заполнения – 50%, что эквивалентно <T_rise> = 0 нс, <T_fall> = (<T_per>/2) нс. <T_rise> <T_fall> – время начала нарастания и начала спада сигнала.</p>
<source>	<p>источник создаваемого синхросигнала (порт, пин). По умолчанию тактовый сигнал будет назван по имени источника. При желании можно указать имя тактового с помощью опции -name..</p>

Пример использования команды:

```
create_clock -period 5 [get_ports CLK1]
create_clock -period 5 -waveform {1 2} [get_ports CLK2]
```

Результат определения коэффициента заполнения тактового сигнала по умолчанию и с использованием опции -waveform показан на рисунке.

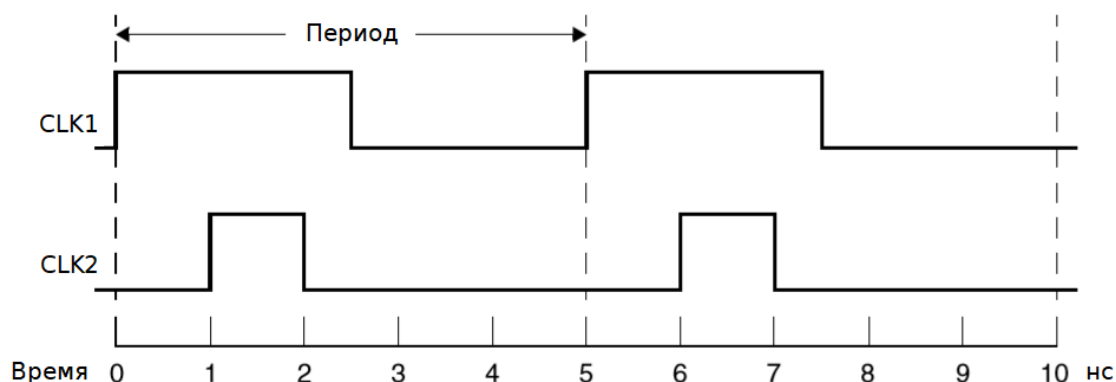


Рисунок 114. Диаграммы тактовых сигналов

set_clock_latency – установить время задержки тактового сигнала.

```
set_clock_latency [-rise/-fall] [-min][-max] [-source] [-late] [-early]
<delay_value> <name>,
```

-rise	задать задержку только при нарастании тактового сигнала;
-fall	задать задержку только при убывании тактового сигнала;
-min	задать только минимальную задержку тактового сигнала;
-max	задать только максимальную задержку тактового сигнала;
-source	задать только задержку задержка источника;
-late	задать только позднее время прибытия от источника;
-early	задать только раннее время прибытия от источника;
<delay_value>	значение времени задержки;
<name>	установить имя тактового сигнала, для которого задается задержка.

Задержка clock_latency состоит из двух:

source latency – задержка источника – время, требуемое синхросигналу, для распространения от источника до точки, указанной в проекте.

network_latency – задержка сети – время, требуемое синхросигналу, чтобы дойти от точки, указанной в проекте, до тактового вывода последовательностного элемента.

В данном случае задержкой источника является задержка ЯВВ, добавляющейся в проект автоматически. Её задержка определяется программой из характеризованной библиотеки.

Задержка сети определяется программой за счет оценки реальных трассировочных элементов, используемых в схеме для коммутации межсоединений. Их задержка также передается из характеризованной библиотеки.

Пример использования команды:

```
set_clock_latency -rise -max 0.34 CLK1
```

set_clock_uncertainty – установить время, на которое отклоняется тактовый сигнал от идеального при распространении на различные приемники после трассировки.

```
set_clock_uncertainty [-setup/-hold] [-rise/-fall]
-from <clk_names> -to <clk_names> <uncertainty>,
```

-setup	задать погрешность только для времени установки;
-hold	задать погрешность только для времени удержания;
-rise	задать погрешность во время нарастания тактового сигнала;
-fall	задать погрешность во время спада тактового сигнала;
-from <clk_names>	задать список тактовых сигналов, являющихся начальными точками анализируемого пути;
-to <clk_names>	задать список тактовых сигналов, являющихся конечными точками анализируемого пути;
<uncertainty>	задать значение погрешности.

В данном случае данная погрешность определяется программой за счет оценки задержки реальных трассировочных элементов из характеризованной библиотеки.

Пример использования команды:

```
set_clock_uncertainty -rise -from CLK1 -to CLK2 0.34
```

set_clock_transition – установить время переключения тактового сигнала.

```
set_clock_transition [-rise/-fall] <transition_time>, <clock_name>
```

-rise	задать только время нарастания тактового сигнала;
-fall	задать только время спада тактового сигнала;
<transition_time>	задать значение времени переключения;
<clock_name>	задать список тактовых сигналов, которым будет установлено заданное значение времени переключения.

Пример использования команды:

```
set_clock_transition -rise 0.15 CLK1
```

set_propagated_clock – распространить задержки тактового сигнала и автоматически определить задержку на каждом тактовом пине регистра с учетом добавленных паразитных элементов.

```
set_propagated_clock <clock_name>,
```

Пример использования команды:

```
set_propagated_clock CLK1
```

create_generated_clock – добавить к анализу тактовый сигнал, производный от основного тактового сигнала, деленный, помноженный или инвертированный внутренним элементом схемы.

```
create_generated_clock -name <clock_name> -source <master_name>
[-divide_by divide_factor | -multiply_by multiply_factor]
[-duty_cycle percent] [-invert] <source>,
```

-name <clk_name>	имя создаваемого синхросигнала;
-source <master_name>	имя источника основного синхросигнала (порт, пин);
-divide_by factor	коэффициент деления сигнала;
-multiply_by factor	коэффициент умножения сигнала;
-duty_cycle percent	коэффициент заполнения в процентах;
-invert	инверсия, создаваемого синхросигнала;
<source>	источник создаваемого синхросигнала (порт, пин);

На рисунке 115 представлена схема, выполняющая деление тактового сигнала на 3.

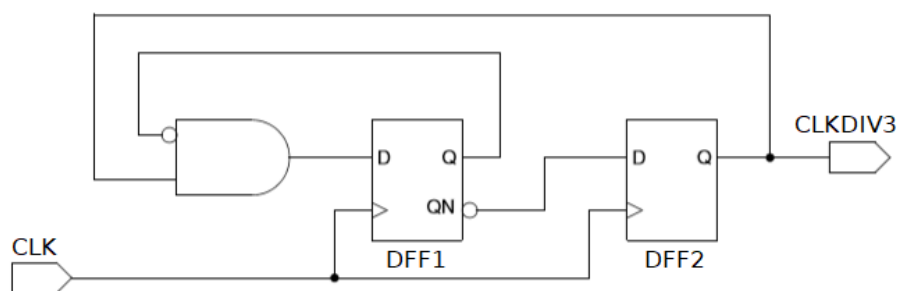


Рисунок 115. Схема деления тактового сигнала

Основной тактовый сигнал CLK и производный от него CLKDIV3 задаются программе следующим образом:

```
create_clock period 2 -name CLK [get_ports CLK]
create_generated_clock -name CLKDIV3 -source [get_ports CLK]
-divide_by 3 -duty_cycle 67 [get_pins DFF2/Q]
```

Формы заданных тактовых сигналов представлены на рисунке 116.

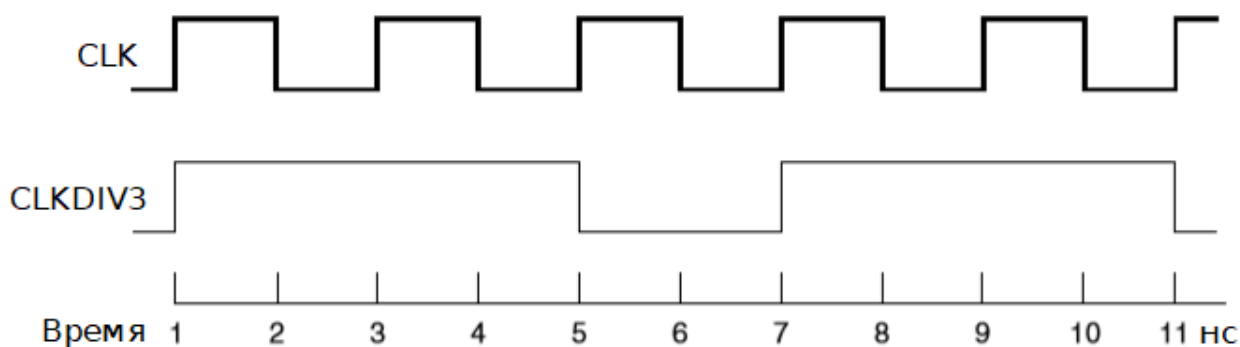


Рисунок 116 – Временные диаграммы основного и производного тактовых сигналов

set_false_path – исключить заданный путь из временного анализа. Команда определяет заданные пути ложными и удаляет временные ограничения на этих путях. Ложные временные пути - это пути, по которым не распространяется изменение логического уровня.

```
set_false_path [-setup/-hold] [-rise/-fall] -through {node list}
-from {node list} -to {node list},
```

-setup	исключить из временного анализа проверку времени установки заданного сигнала;
-hold	исключить из временного анализа проверку времени удержания заданного сигнала;
-rise	исключить из временного анализа элементы, тактирующиеся нарастающим фронтом заданного тактового сигнала;
-fall	исключить из временного анализа элементы, тактирующиеся убывающим фронтом заданного тактового сигнала;
-through {list}	задать список пинов, портов или ячеек, через которые должны проходить ложные пути;
-from {list}	задать список начальных точек исключаемого временного пути – тактовые сигналы, входные порты, входные тактовые пины последовательностных ячеек;
-to {list}	задать список конечных точек исключаемого временного пути – тактовые сигналы, выходные порты, выходные пины данных последовательностных ячеек;

Пример использования команды:

```
set_false_path -to [get_clocks CLK]
```

Данная команда объявляет пути, тактируемые в конечных точках сигналом CLK, на рисунке 117 ложными, т.е. исключает из временного анализа все пути.

```
set_false_path -fall -to [get_clocks CLK]
```

Данная команда объявляет пути, тактируемые в конечных точках убывающим фронтом сигнала CLK, на рисунке 117 ложными, т.е. исключает из временного анализа пути В и С.

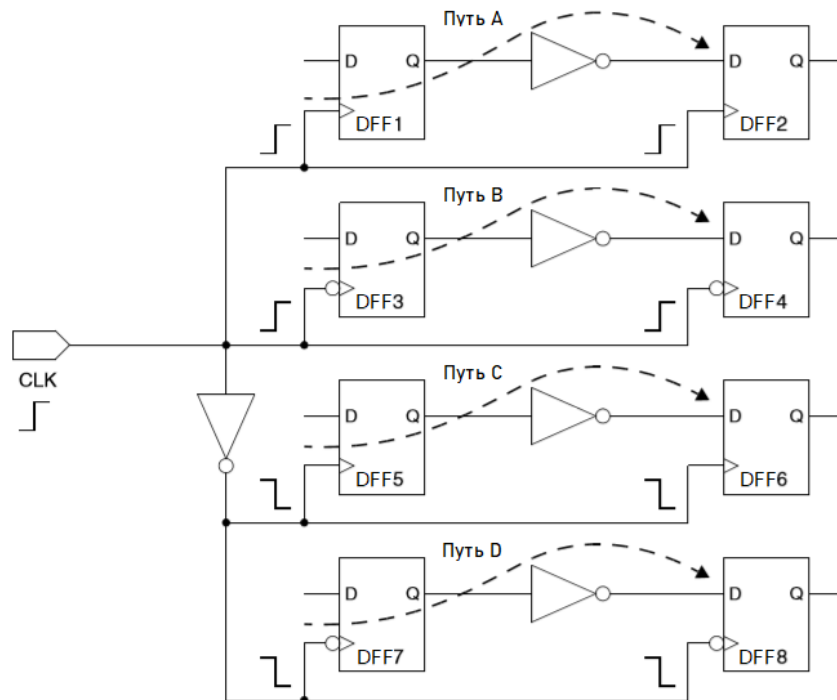


Рисунок 117 – Схема, поясняющая использование ложных путей

set_disable_timing – удалить из временного анализа все объекты (ячейки, пины или порты) на заданном пути.

Эта команда полностью удаляет из временного анализа встретившиеся на пути объекты, в отличие от команды `set_false_path`, которая исключает только конкретный путь, а не вычисление задержки объектов этого пути. Если все пути через определенный пин – ложные, то применение `set_disable_timing` более эффективно, чем `set_false_path`.

```
set_disable_timing {[-from pin_name] | [-to pin_name]} <cell_list>,
```

-from pin_name	задать начальный пин исключаемого временного пути;
-to pin_name	задать конечный пин исключаемого временного пути;
	Для работы команды необходимо задать одновременно как пин начала пути, так и пин, показывающий конец пути.
<cell_list>	задать список ячеек, удаляемых из временного анализа для заданного пути;

Пример использования команды:

```
set_disable_timing -from DFF1/D -to DFF2/Q {DFF1 DFF2}
```

set_multicycle_path – установить путь, занимающий несколько тактов, для проверки времени установки и времени удержания.

```
set_multicycle_path ncycles [-setup/-hold] [-rise/-fall]
-through {node list} -from {node list} -to {node list},
```

ncycles	количество циклов тактового сигнала, которое требуется данным для проверки времени установки и удержания;
-setup	изменить количество циклов тактового сигнала только для проверки времени установки;
-hold	изменить количество циклов тактового сигнала только для проверки времени удержания. По умолчанию время удержания проверяется на ncsycles - 1 такте;
-rise	изменить количество циклов нарастания тактового сигнала;
-fall	изменить количество циклов убывания тактового сигнала;
-through {list}	задать список пинов, портов или ячеек, через которые проходит устанавливаемый путь;
-from {list}	задать список начальных точек устанавливаемого пути – тактовые сигналы, входные порты, входные тактовые пины последовательностных ячеек;
-to {list}	задать список конечных точек устанавливаемого пути – тактовые сигналы, выходные порты, выходные пины данных последовательностных ячеек;

Пример использования команды:

```
set_multicycle_path 4 -setup -from [get_ports CLK]
```

set_input_transition – установить время переключения на входах схемы.

```
set_input_transition <value> <input name list> [r / f],
```

<value>	значение времени переключения;
<input name list>	список входов, которым будет присвоено заданное значение;
r	установить время нарастания сигнала на заданных входах;
f	установить время спада сигнала на заданных входах;
	По умолчанию, если не заданы опции r/f, заданное время переключения установится как для нарастания, так и для спада.

Пример использования команды:

```
set_input_transition 1.5 [get_inputs]
```

set_input_delay – установить время задержки прибытия входного сигнала относительно основного тактового сигнала.

```
set_input_delay <input_delay value> <net name list>
[-clock <related clock>],
```

<input_delay value>	значение времени задержки
<net name list>	задать список входов, которым будет присвоено заданное значение
-clock <related clock>	установить имя тактового сигнала, относительно которого приходит входной сигнал

Пример использования команды:

```
set_input_delay 0.344 {IN1 IN2} [-clock CLK]
```

set_output_delay – установить время задержки прибытия выходного сигнала до последовательностной логики вне проекта относительно основного тактового сигнала.

```
set_output_delay <output delay value> <net name list>,
[<related clock>],
```

<output_delay value>	значение времени задержки
<net name list>	задать список входов, которым будет присвоено заданное значение
-clock <related clock>	установить имя тактового сигнала, относительно которого приходит выходной сигнал

Пример использования команды:

```
set_output_delay 0.7 {OUT} [-clock CLK]
```

set_max_delay – задать максимальное время задержки сигнала для временных путей, заканчивающихся в заданном списке выходов.

```
set_max_delay <delay_value> <name list> [r / f],
```

<delay_value>	значение времени задержки;
<name list>	список выходов, которым будет присвоено заданное значение;
r	установить максимальную задержку для нарастающего сигнала;
f	установить максимальную задержку для убывающего сигнала;
	По умолчанию, если не заданы опции r/f, заданное время задержки установится как для нарастающего, так и для убывающего сигнала.

Пример использования команды:

```
set_max_delay 10.5 [get_outputs] r
```

set_min_delay – задать минимальное время задержки сигнала для временных путей, заканчивающихся в заданном списке выходов.

```
set_min_delay <delay value> <name list> [r / f],
```

<delay_value>	значение времени задержки;
<name list>	список выходов, которым будет присвоено заданное значение;
r	установить максимальную задержку для нарастающего сигнала;
f	установить максимальную задержку для убывающего сигнала;
	По умолчанию, если не заданы опции r/f, заданное время задержки установится как для нарастающего, так и для убывающего сигнала.

Пример использования команды:

```
set_min_delay 6.5 [get_outputs] r
```

set_load – задать дополнительную емкость списку цепей.

```
set_load <cap_value> <names_list> replace,
```

<cap_value> значение емкости;
 <names_list> задать список цепей, которым будет установлена дополнительная емкость;
 заменить существующую на заданных цепях емкость. По умолчанию, значение,
 replace установленное данной командой, добавляется к существующей емкости, а не
 заменяет её;

Пример использования команды:

```
set_load 0.5 [get_outputs] replace
```

set_case_analysis – задать постоянное логическое значение сигнала или нарастающий/убывающий фронт и исключить из временного анализа связанный с ними путь.

```
set_case_analysis {0 | 1 | zero | one | rise | fall} <objects_list>,
```

0 / zero	задать порту или пину значение «0»;
1 / one	задать порту или пину значение «1»;
rise	задать порту или пину постоянный нарастающий сигнал;
fall	задать порту или пину постоянный убывающий сигнал;
<objects_list>	задать список портов или пинов, для которых будет задано постоянное значение;

Функции, позволяющие получить списки имен по требуемому критерию

get_inputs	получить список входов;
get_outputs	получить список выходов;
get_nodes	получить список всех цепей;
get_nodes <internal>	получить список внутренних цепей;
get_cells	получить список всех ячеек;

