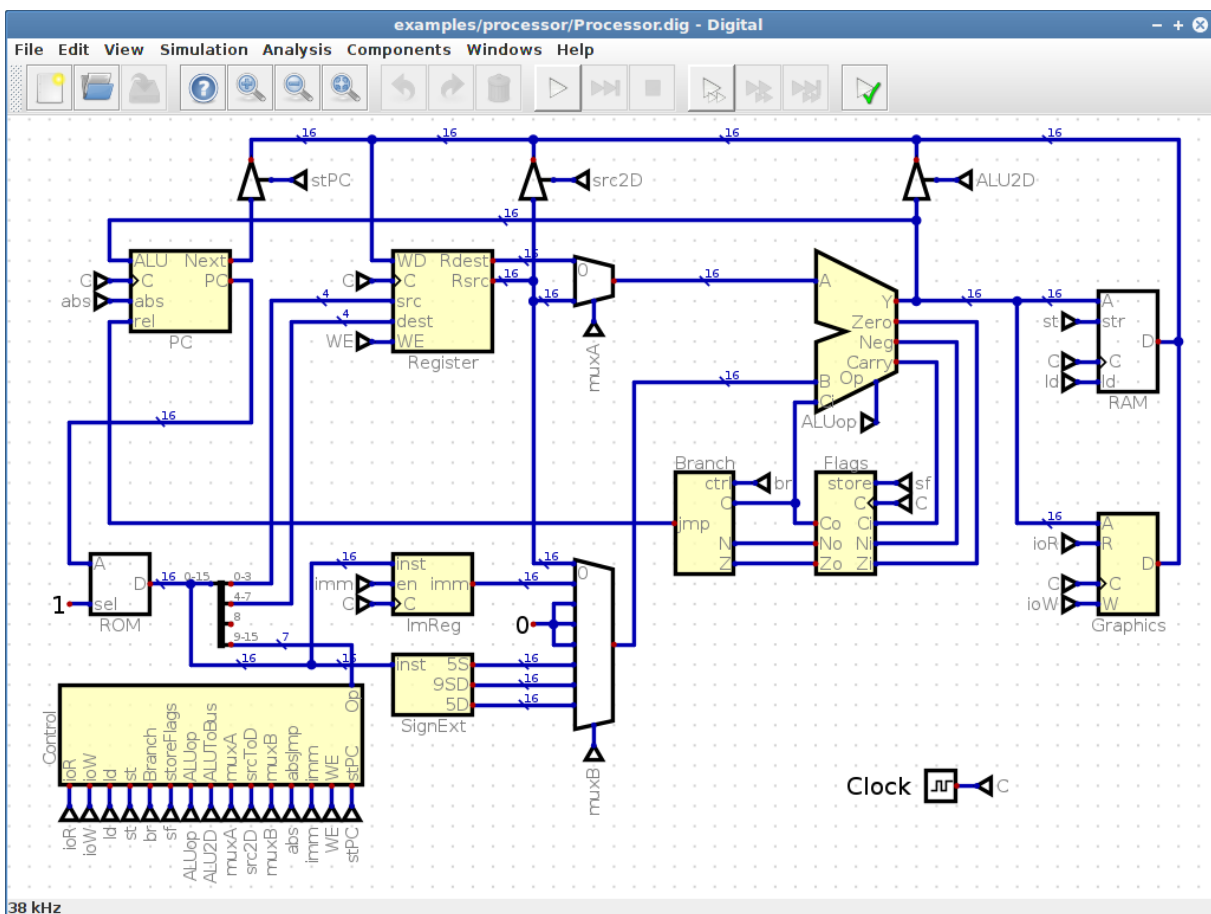


# Digital



Версия: null  
 Дата: null

# Оглавление

## А Общие

1. Digital .....	6
1.1. Введение .....	6
1.2. Первые шаги .....	6
1.3. Провода .....	8
1.4. Иерархический дизайн .....	8
2. Моделирование .....	9
2.1. Задержка распространения .....	9
3. Анализ .....	10
3.1. Анализ и синтез схем .....	10
3.2. Выражение .....	10
3.3. Диаграммы состояний .....	10
4. Оборудование .....	11
4.1. GAL16v8 и GAL22v10 .....	11
4.2. ATF150xAS .....	11
4.3. Экспорт в VHDL или Verilog .....	11
5. Пользовательские компоненты .....	12
6. Общие схемы .....	13
7. Тестирование под управлением сценария .....	14
8. Часто задаваемые вопросы .....	14
9. Сочетания клавиш .....	16

## В Настройки

## С Интерфейс командной строки

## D Компоненты

1. Логика	
1.1. И .....	23
1.2. И-НЕ .....	23
1.3. ИЛИ .....	24
1.4. ИЛИ-НЕ .....	25
1.5. XOR .....	25
1.6. XNOR .....	26
1.7. НЕ .....	27
1.8. Таблица поиска .....	27
2. ввод-вывод	
2.1. Вывод .....	28
2.2. светодиод .....	29
2.3. Ввод .....	29
2.4. Вход часов .....	30
2.5. Кнопка .....	31
2.6. DIP-переключатель .....	31
2.7. Зонд .....	32
2.8. График данных .....	32
2.9. График иницируемых данных .....	33
3. ввод-вывод - Индикаторы	
3.1. RGB-светодиод .....	33

---

3.2. Светодиод с двумя подключениями. ....	34
3.3. Кнопка со светодиодом .....	34
3.4. Семисегментный дисплей .....	35
3.5. Семисегментный шестнадцатеричный дисплей .....	36
3.6. 16-сегментный дисплей .....	36
3.7. Лампочка .....	37
3.8. Светодиодная матрица .....	37
4. ввод-вывод - Электромеханика	
4.1. Поворотный энкодер .....	38
4.2. Шаговый двигатель, униполярный .....	38
4.3. Шаговый двигатель, биполярный .....	39
5. ввод-вывод - Периферийные устройства	
5.1. Клавиатура .....	40
5.2. Терминал .....	41
5.3. Монитор VGA .....	41
5.4. MIDI .....	42
6. Провода	
6.1. Земля .....	43
6.2. Напряжение питания .....	43
6.3. Постоянное значение .....	44
6.4. Туннель .....	44
6.5. Разделитель / Слияние .....	44
6.6. Драйвер .....	45
6.7. Драйвер, инвертированный выбор .....	46
6.8. Задержка .....	47
6.9. Подтягивающий резистор .....	47
6.10. Понижающий резистор .....	48
6.11. Не подключен .....	48
7. Плексеры	
7.1. Мультиплексор .....	48
7.2. Демультиплексор .....	49
7.3. Декодер .....	50
7.4. Селектор бит .....	50
7.5. Приоритетный шифратор .....	51
8. Триггеры	
8.1. RS-триггер .....	52
8.2. RS-триггер, синхронизированный .....	53
8.3. JK-триггер .....	54
8.4. D-триггер .....	55
8.5. T-триггер .....	55
8.6. JK-триггер, асинхронный .....	56
8.7. D-триггер, асинхронный .....	57
8.8. Монофlop .....	58
9. Память - ОЗУ	
9.1. ОЗУ, отдельные порты .....	59
9.2. ОЗУ блока, разделенные порты .....	60
9.3. ОЗУ, двунаправленный порт .....	61
9.4. RAM, выбор микросхемы .....	62
9.5. Файл регистрации .....	63
9.6. ОЗУ, двойной порт .....	64

---

9.7. ОЗУ, асинхронно .....	65
9.8. Графическая RAM .....	66
10. Память - EEPROM	
10.1. EEPROM .....	67
10.2. EEPROM, разделенные порты .....	68
11. Память	
11.1. Зарегистрироваться .....	69
11.2. ПЗУ .....	70
11.3. Двойной порт ПЗУ .....	71
11.4. Счетчик .....	72
11.5. Счетчик с предустановкой .....	73
11.6. Генератор случайных чисел .....	74
12. Арифметика	
12.1. Сумматор .....	75
12.2. Вычесть .....	75
12.3. Умножить .....	76
12.4. Деление .....	77
12.5. Баррель-шифтер .....	77
12.6. Компаратор .....	78
12.7. Отрицание .....	79
12.8. Расширитель знаков .....	79
12.9. Битовый счетчик .....	80
13. Переключатели	
13.1. Переключить .....	80
13.2. Двойной переключатель .....	81
13.3. Реле .....	81
13.4. Реле двойного выброса .....	82
13.5. П-канальный полевой транзистор .....	83
13.6. N-канальный полевой транзистор .....	84
13.7. Предохранитель .....	84
13.8. Диод к VDD .....	85
13.9. Диод - земля .....	85
13.10. П-канальный полевой транзистор с плавающим затвором .....	86
13.11. N-канальный полевой транзистор с плавающим затвором .....	87
13.12. Шлюз передачи .....	87
14. Разное	
14.1. Тестовый пример .....	88
15. Разное - Украшение	
15.1. Текст .....	88
15.2. Прямоугольник .....	89
16. Разное - Общий	
16.1. Общая инициализация .....	90
16.2. Код .....	90
17. Разное - VHDL/Verilog	
17.1. Внешний .....	90
17.2. Внешний файл .....	91
17.3. Пин-контроль .....	92
18. Разное	
18.1. Питание .....	93
18.2. Двухнаправленный разделитель .....	93

---

18.3. Сброс .....	94
18.4. Разрыв .....	94
18.5. Остановить .....	95
18.6. Асинхронное время .....	95

**Е Библиотека**

## А Общие

### 1. Digital

#### 1.1. Введение

Digital - это простой симулятор, используемый для моделирования цифровых схем. Логические вентили подключены друг к другу проводами, и можно смоделировать поведение всей схемы. Пользователь может взаимодействовать с симуляцией, нажимая кнопки или задавая значения на входы схемы.

Таким образом, можно построить и смоделировать большинство базовых схем, используемых в цифровой электронике. В папке *examples* пользователи могут найти примеры, включающие функциональные 16-битный одноканальный процессор Гарварда.

Симулятор имеет два режима работы: режим редактирования и моделирования. В режиме редактирования можно вносить изменения в схему. Пользователи могут добавлять или подключать компоненты. В этом режиме имитация отключена. Режим моделирования активируется нажатием кнопки *Start* на панели инструментов. При запуске моделирования схема проверяется на целостность. Если в цепи есть ошибки, отображается соответствующее сообщение и затронутые компоненты или провода выделяются. Если в схеме нет ошибок, симуляция включена. Теперь вы можете взаимодействовать с бегущей симуляцией. В режиме моделирования невозможно изменить схему. Для этого вам необходимо активировать снова в режиме редактирования, остановив симуляцию.

#### 1.2. Первые шаги

В качестве первого примера схема должна быть построена с вентилем Иключающее ИЛИ. В главном окне меню *Компоненты* позволяет выбирать различные компоненты. Затем они размещаются на панели для рисования. Этот процесс можно отменить, нажав клавишу ESC в любое время. Начните с выбора входного компонента. Позже этим можно будет управлять в интерактивном режиме с помощью мыши.

После выбора первый вход можно разместить на панели рисования. Красная точка на символе входного компонента - это точка соединения между компонентом и проводом, который будет подключен позже. Красный цвет указывает на выход. Это означает, что порт определяет значение сигнала или может управлять проводом.

Таким же образом добавляется второй вход. Лучше всего разместить его прямо под первым входом.

После добавления входов выбирается элемент XOR. Этот элемент представляют собой фактическую логическую функцию.

Этот вентиль теперь также можно добавить в схему. Лучше всего разместить его так, чтобы последующие разводка выполняется максимально просто. Синие точки обозначают входные клеммы элемента.

Теперь выберите выход, который можно использовать для отображения состояния сигнала или для последующей передачи сигналов в схема вложения.

Он размещен таким образом, чтобы его можно было легко подключить. На выходе есть синяя точка, обозначающая входной терминал. Здесь вы можете ввести значение, которое затем экспортируется.

После того, как все компоненты выбраны и установлены на свои места, с помощью мыши соедините синий и красные точки. Убедитесь, что ровно одна красная точка соединена с любым количеством синих точек. Только использование выходов с тремя состояниями позволяет отклониться от этого правила и соединить несколько красных точек. Если все провода нарисованы, цепь замкнута.

Взаимодействие со схемой возможно при запуске моделирования. Это делается нажатием на кнопку воспроизведения, расположенную на панели инструментов. После запуска моделирования цвет проводов меняется, и входы и выходы теперь заполнены. Ярко-зеленый означает логическую «1», а темно-зеленый - логический «0». На рисунке выше все провода имеют значение «0».

Щелчком мыши можно переключать входы. Поскольку симуляция теперь активна, выход изменится в соответствии с текущими состояниями входа. Схема ведет себя как Исключающий ИЛИ, как и ожидалось.

Для дальнейшей обработки схемы необходимо сначала остановить моделирование. Самый простой способ сделать это с помощью кнопки Стоп на панели инструментов. Щелчок по компоненту правой кнопкой мыши (щелкните macOS, удерживая нажатой клавишу Control), открывается диалоговое окно, в котором отображаются свойства компонента. Этикетка «A» может быть определенным для первого ввода через это диалоговое окно.

Таким же образом можно определить метки для остальных входов и выходов. Пункт меню *Анализ* также содержит пункт меню *Анализ*. Эта функция выполняет анализ текущая цепь. Однако это возможно только при правильной маркировке всех входов и выходов.

Таблица истинности смоделированной схемы появится в новом окне. Под таблицей вы можете найти алгебраическое выражение, связанное со схемой. Если существует несколько возможных алгебраических выражений, откроется отдельное окно, в котором будут показаны все возможные выражения.

В главном меню диалогового окна таблицы есть пункт *K-Map*. Это позволяет показать таблицу истинности в виде K-карты.

Вверху этого диалогового окна есть раскрывающийся список, который позволяет выбрать желаемый выражение в К-карте. Таким образом вы можете, например, проиллюстрировать, как несколько эквивалентных могут возникнуть алгебраические выражения. Однако в этом примере есть только одно минимальное выражение. Таблицу истинности также можно изменить, щелкнув К-карту.

### 1.3. Провода

Все компоненты должны быть соединены проводами. Невозможно соединить два компонента разместив их рядом друг с другом.

Кроме того, есть только соединения между конечной точкой провода и компонентом. Если вывод компонента расположен в середине провода, соединение между компонентом не выполняется. и провод. Следовательно, провод должен фактически заканчиваться на каждом контакте, который должен быть подключен. Даже если используется туннельный компонент, между штифтом и туннельным элементом должен быть провод.

Чтобы переместить компонент, необходимо выбрать его с помощью инструмента прямоугольного выделения. вместе с подключенными проводами. Для перемещения компонента без подключенных проводов, выберите компонент щелчком мыши.

С помощью CTRL-Click можно выбрать одиночный участок провода, чтобы переместить или удалить его. Если во время рисования провода нажать клавишу D, можно нарисовать диагональный провод. Клавиша S позволяет разделить линейный сегмент на два сегмента.

### 1.4. Иерархический дизайн

Если построена сложная схема, это может быстро запутать. Чтобы отследить здесь, разные части схемы могут храниться в разных файлах. Этот механизм также делает его возможно использовать подсхему, которая была создана один раз, несколько раз в следующей схеме. Этот подход также предлагает то преимущество, что файлы могут храниться независимо друг от друга в система контроля версий и изменения могут быть отслежены.

В качестве примера рассмотрим 4-битный сумматор: сначала мы построили простой полусумматор. Он состоит из Элемент XOR и Элемент И. Сумма двух битов «А» и «В» передается на выходы «S» и «C». Эта схема хранится в файле *halfAdder.dig*.

Теперь из двух полусумматоров можно построить полный сумматор. Для этого создайте новый пустой файл и сохраните пустой файл как *fullAdder.dig* в той же папке, что и полусумматор. Тогда полусумматор можно добавить в новую схему через меню *Components* → *Custom*. Порядок контактов на упаковке полусумматора можно изменить с полусумматора в меню *Редактировать* → *Порядок входов* или *Edit* → *Порядок выходов*. Полный сумматор складывает три бита «А», «В» и «Сi» и дает сумму для выходов «S» и «Co».

Чтобы проверить правильность работы полного сумматора, необходимо добавить тестовый пример. В тестовом примере сохраняется таблица истинности, которая должна выполнять схему. Таким образом, это может быть автоматически проверил, так ли это.



Тесты могут быть выполнены с помощью редактора тестовых примеров или кнопки тестирования на панели инструментов. Ячейки таблицы, выделенные зеленым цветом, показывают, что выход схемы соответствует таблице истинности, приведенная в тестовом примере.

Теперь полные сумматоры можно собрать в так называемый сумматор с волновым переносом. В этом случае выход переноса дополнения направляется как вход переноса в добавление следующего бит более высокого порядка, как это обычно бывает при сложении карандаша и бумаги. Этот 4-битный сумматор следует проверить на правильность работы. Для этого был вставлен тестовый пример.

Этот тестовый пример выполняет 100% тест, который возможен только с относительно простыми схемами: все возможные 512 входных комбинаций применяются к схеме, и проверяется, схемы правильно. В первой строке перечислены входные и выходные сигналы. Ниже вводятся применяемые входные значения и проверяемые выходные значения указываются в строке, как в таблице истинности. Однако в этом примере требуется 512 строк. Ввод этого будет утомительным и подверженным ошибкам занятием. Автоматически генерировать нужные строки проще и надежнее. Для этого пересекаются переменные  $A$  и  $B$ , от 0 до 15. Соответствующие значения  $A$  и  $B$  затем назначаются входам « $A [n]$ » и « $B [n]$ ». Затем проверяется, выводит ли схема значение  $A + B$ . Затем снова проверяется с помощью установлен бит переноса, и в этом случае должно получиться  $A + B + 1$ . Подробная информация о синтаксисе теста представлена в диалоговом окне справки.

Если схема встроена в другую схему, только имя файла подсхемы сохраняется в схема, а не сама встроена схема. Соответствующие файлы поэтому встроенные подсхемы должны быть найдены в файловой системе во время выполнения моделирования. Чтобы как можно лучше поддерживать различные методы работы пользователей и при этом избегать сложное администрирование путей импорта и т.д., реализована несколько необычная стратегия импорта.

В файле схем хранятся только имена файлов встроенных схем, а не полный путь. Если файл необходимо открыть, во всех вложенных папках выполняется поиск файла с соответствующим именем. Если подходящий файл найден, он импортируется. Этот процесс зависит только от имени файла, который нужно быть прочитанным, а не на своем пути. Соответственно, выдается сообщение об ошибке, если имеется несколько файлов одно и то же имя в разных подпапках, поскольку в этом случае возникают двусмысленности.

Таким образом, подходящая структура проекта выглядит следующим образом: Корневой контур расположен в отдельном папка. Все импортированные схемы должны находиться в одной папке или подпапках. Все схемы должны иметь разные названия, поэтому не должно происходить, чтобы схемы с одним и тем же именем находились в разных папках.

## 2. Моделирование

### 2.1. Задержка распространения

Во время моделирования каждый логический вентиль имеет задержку распространения. Каждый компонент, найденный в библиотеке имеет одинаковую задержку распространения независимо от его сложности. Таким образом, логический элемент И имеет ту же задержку распространения, что и умножитель. Единственным исключением являются диоды, переключатели и разветвители, которые используются для создания шин данных. Эти компоненты вообще не имеют задержки распространения.

Если необходимо смоделировать логический элемент - например, множитель - с большей задержкой распространения, задержкой вентиль должен быть вставлен в схему сразу за выходом умножителя.

Если схема включена в другую родительскую схему, включенная схема сохраняет свои временные характеристики. Поэтому, если вы включите сложную схему, которая имеет большую задержку распространения, потому что входные сигналы должны пройти три гейта, пока не достигнет выхода, это поведение сохраняется при включении этой схема. Никаких дополнительных задержек в результате включения цепи не возникает. Если не все выходы схема есть такая же задержка распространения, то это также имеет место, если она включена в родительский контур. В общем, включение схемы в другую схему вообще не изменяет ее временное поведение. А включенная цепь ведет себя точно так же, как если бы все компоненты были вставлены в одну и ту же цепь уровень.

## 3. Анализ

### 3.1. Анализ и синтез схем

Цепь можно проанализировать с помощью пункта меню *Анализ*. Таблица истинности создается для чисто комбинаторные схемы. Эту таблицу истинности можно редактировать по желанию. После редактирования из этой таблицы истинности может быть сгенерирована новая схема.

Помимо чисто комбинаторных схем, также можно анализировать или генерировать последовательные схемы. Вместо простой таблицы истинности создается так называемая таблица переходов состояний. Таким образом, каждый триггер появляется на входной и выходной стороне таблицы переходов состояний. В этой таблице справа вы можете найти следующее состояние, которое будет происходить после следующего тактового сигнала. Это следующее состояние зависит от текущего состояния триггеров. в левой части таблицы. Чтобы анализ был возможен, триггеры должны быть названы.

Применяется следующее соглашение об именах: следующее состояние бита в правой части таблицы обозначается строчной буквой « $n + 1$ ». Соответствующее текущее состояние обозначается добавленным « $n$ ». Если существует переменная состояния « $A$ », « $A_n$ » указывает текущее состояние, а « $A_{n + 1}$ » указывает следующее состояние. Если в таблице истинности слева и справа присутствуют сигналы, соответствующие этому шаблону предполагается, что таблица является таблицей переходов состояний, и вместо нее создается последовательная схема. комбинаторная схема.

Следует отметить, что анализируемая схема может содержать только чисто комбинаторные элементы в дополнение к встроенным шлепанцам D и JK. Если триггер, например, сделано из элементов ИЛИ-НЕ, это Схема не распознается как триггер, поэтому анализ такой схемы невозможен.

### 3.2. Выражение

Через пункт меню *Выражение* можно ввести логическую функцию, из которой затем может быть сгенерирована цепь.

### 3.3. Диаграммы состояний

Редактор диаграмм состояний доступен через пункт меню *Конечный автомат*. Он позволяет графическое создание конечных автоматов путем рисования состояний и переходов между состояниями. Таким образом, разные выходы могут быть установлены в разные состояния. Предостав-

для переходы с условиями, можно сгенерировать входные сигналы. Установив выходные значения при переходах, можно также определить автоматы Мили.

Определенный таким образом конечный автомат затем может быть автоматически перенесен в таблицу переходов состояний, из которого на следующем этапе может быть сгенерирована схема, реализующая автомат начальных состояний. Если затем запустить моделирование этой схемы, текущее состояние также можно будет отслеживать в диаграмма состояния.

## 4. Оборудование

### 4.1. GAL16v8 и GAL22v10

В меню генерации схемы в таблице истинности также есть функции для генерации так называемых Файлы JEDEC. Это специальный формат файла, который описывает схему предохранителей PLD. Этот файл JEDEC можно записать в соответствующий PLD с помощью специального программатора. На данный момент схемы типа *GAL16v8* и *GAL22v10* или схемы предохранителей совместимы устройства поддерживаются.

### 4.2. ATF150xAS

Микросхемы в *ATF150x* Семейство - это простые CPLD, содержащие до 128 макроячеек. Они доступны в Пакет PLCC, который делает их пригодными для лабораторных упражнений: если ИС разрушается во время упражнений, его можно просто заменить. Кроме того, с *ATDH1150USB* доступен простой в использовании и недорогой программатор. Этот программист умеет программировать *ATF150x* микросхемы в системе с использованием интерфейса JTAG. Подходящая оценочная плата (*ATF15XX-DK3-U*) также доступен. Программное обеспечение *ATMISP*, который доступен на веб-сайте ATMEL/Microchip, необходим для программирования микросхем.

К сожалению, информация о карте предохранителей не является общедоступной, поэтому нет подходящего установщика для этого чипа. могут быть интегрированы в Digital, как это возможно с микросхемами *GAL16v8* и *GAL22v10*.

Следовательно, необходимо использовать установщики *fit150 [x].exe*, предоставленные ATMEL. Эти программы создают Файл *JEDEC* из подходящего файла *TT2*, который затем можно запрограммировать на микросхеме. Цифровой автоматически запускает установщик каждый раз при создании файла *TT2*. Для этого путь к Установщики *fit150 [n].exe* должны быть указаны в настройках. Созданный файл *JEDEC* затем можно открыть и запрограммировать напрямую с помощью *ATMISP*.

По юридическим причинам установщик *fit1502.exe* не может распространяться вместе с Digital. Однако это может быть находится в папке *WinCupl \ Fitters* после установки *WinCupl*. *WinCupl* доступен на веб-сайте ATMEL/Microchip. В системах Linux установщики также могут выполняться Digital, если установлено *wine*.

### 4.3. Экспорт в VHDL или Verilog

Схема может быть экспортирована в VHDL или Verilog. Создается файл, содержащий полное описание схемы. Сгенерированный код VHDL был протестирован с Xilinx Vivado и симулятор VHDL с открытым исходным кодом ghdl. Код Verilog протестирован с помощью симулятора Verilog Icarus Verilog.

Если схема содержит тестовые примеры, тестовые данные используются для создания тестового стенда HDL. Это можно использовать для проверки правильности работы схемы в моделировании HDL.

Могут быть созданы дополнительные файлы, необходимые для специальных плат. В настоящее время только BASYS3 доска и доски Mimas Mimas и Mimas V2 поддерживаются. Создается файл ограничений, содержащий назначение выводов. Описание булавок можно найти в листе технических данных платы, и его необходимо ввести как номер контакта для входов и выходов.

Для платы BASYS3, если тактовая частота схемы низкая, в HDL встроены делитель частоты. код, чтобы соответственно разделить часы на плате. Если тактовая частота, выбранная в схеме, превышает 4,7 МГц, блок MMCM Artix-7 используется для генерации часов. Это гарантирует, что ресурсы FPGA, предоставленные для распределения часов, используются. Это позволяет включенному образцу процессора работать на частоте 20 МГц, и, если вы можете обойтись без множитель, также возможен 30MHz.

Если схема должна работать на плате BASYS3, в Vivado можно создать новый проект. Сгенерированный файл VHDL и файл ограничений необходимо добавить в проект. После создания проекта можно сгенерировать поток битов и использовать Hardware-Manager. для программирования платы BASYS3.

Чтобы создать требуемый файл ограничений в дополнение к файлу HDL, соответствующая плата необходимо настроить в настройках. В поле «Toolchain Configuration» соответствующий файл XML. можно выбрать. Доступные конфигурации находятся в папке *examples/hdl* и содержат файл расширения *.config*. Если конфигурация была успешно интегрирована, появится дополнительное меню, в котором плата доступны определенные функции.

## 5. Пользовательские компоненты

Хотя у Digital есть несколько параметров, которые определяют внешний вид схемы, когда она встроена в другую, в некоторых случаях может быть полезно использовать особый компонент для подсхемы. Примером является представление ALU в процессоре, смотреть в примерах. В этой главе объясняется, как определить такой особый компоненты цепи.

В Digital нет редактора для создания особого компонента. Вместо этого требуется небольшой обходной путь для создания фигуры контура: сначала открывается компонент, который должен быть представлен специальным открытым компонентом. Затем для этой схемы создается шаблон SVG. В этом шаблоне схема представлена простым прямоугольником. Он также содержит все контакты схемы, представленные синим цветом (входы) и красные (выходы) кружочки. Чтобы узнать, какой круг принадлежит какому выводу, вы можете посмотреть идентификатор кружка в свойствах объекта. Этот идентификатор имеет вид *pin: [имя]* или *pin +: [name]*. В последнем варианте вывод снабжен ярлыком при повторном импорте в цифровой формат. Если вам не нужна такая метка, можно удалить +.

Теперь этот файл SVG можно редактировать. Наиболее подходящей является программа с открытым исходным кодом. Inkscape, которая доступна бесплатно. Выводы можно перемещать свободно, но при повторном импорте они перемещаются к следующей точке сетки.

Если необходимо использовать существующие файлы SVG, проще всего открыть созданный шаблон и вставить существующий рисунок в шаблон с помощью копирования и вставки.

Если файл был сохранен, его можно импортировать с помощью Digital. Файл открывается и все необходимая информация извлекается и сохраняется в схеме. Для дальнейшего использования схемы SVG файл больше не требуется.

Последнее замечание: SVG - очень мощный и гибкий формат файла. Его можно использовать для описания чрезвычайно сложной графики. Цифровой импортер не может импортировать все возможные файлы SVG без ошибок. Если файл не может быть импортирован или отображается не так, как ожидалось, может потребоваться некоторое экспериментирование, прежде чем будет достигнут желаемый результат.

## 6. Общие схемы

Бывает, что подсхема создана, и ее нужно использовать в разных вариантах. Например, вы можете представить специальный счетчик, который нужен для разной разрядности. Если создать подсхему на 4, 5 и 6 бит каждая, обслуживание схемы будет усложнено в будущем, так как всегда нужно работать с несколькими подсхемами, которые идентичны, за исключением одного параметра - разрядности.

Чтобы предотвратить это, можно создать общую подсхему, которая может быть параметризована. Для этого в настройках контура должен быть установлен флажок «Цепь является универсальной». Затем диалоговое окно параметров каждого компонента в этой цепи содержит дополнительное поле «общая параметризация». В это поле можно ввести программный код, который может изменять параметры составная часть. Каждый параметр имеет имя и может быть изменен как атрибут поля *this*. Названия параметров можно найти в диалоговом окне справки компонента. Если вы хотите изменить разрядность сумматора, можно использовать строку *this.Bits=1;*

Однако таким образом еще невозможно создать схему, которая может быть параметризована. По-прежнему необходимо получить доступ к параметрам, которые устанавливаются при использовании схемы. Это делается через поле «args». Если вы хотите установить разрядность снаружи, вы можете написать: *this.Bits=args.bitWidth;* Имя аргумента - здесь *bitWidth* произвольно. Если такая подсхема используется, этот аргумент должен быть установлен.

Если схема используется и открывается диалоговое окно параметров встроенной схемы, она также имеет поле «общая параметризация». Здесь используемую разрядность можно установить с помощью инструкции *bitWidth:=5;*

Если общий контур должен быть запущен напрямую, это невозможно сразу, так как требуемые отсутствуют аргументы, которые необходимо указать при встраивании схемы. Эти недостающие аргументы приведет к появлению соответствующих сообщений об ошибках. Поэтому, чтобы упростить тестирование схемы, компонент *Generic Initialization* может быть добавлен в схему. В этом компоненте вы можете установить аргументы, которые будут поступать от схемы встраивания. Таким образом, обычная схема также может быть смоделирована напрямую. Если схема встроена, этот компонент игнорируется. Он нужен только для непосредственного запуска симуляции.

При определенных обстоятельствах может быть полезно не только изменить атрибуты компонентов схему, но добавить совершенно новые компоненты и провода в зависимости от переданных аргументов. Для этого можно использовать компонент *Code*. Если он добавлен в схему, содержащийся Код будет выполнен при запуске симуляции. Здесь провод можно добавить с помощью функции *addWire ([x1], [y1], [x2], [y2])* и с помощью функции *addComponent ([name], [x], [y])* новый компонент *[name]* может быть добавлен в позицию *([x], [y])*. Возвращаемое значение функции *addComponent ([Name], [x], [y])* позволяет установить параметры составная часть.

Схема примера *examples/generic/modify/Conway/GenericConway.dig* показывает, как более сложные Схема может быть собрана таким образом.

Другой способ создания схемы - это рекурсия: можно, в зависимости от аргументов, заменить одну схема другим. Для этого доступна функция *setCircuit([Name])*. Если он вызывается в части определения подсхемы, вставляемая схема может быть заменена по другой схеме. Это позволяет рекурсивно определять схему. Как и в другом программировании языков, должно быть обеспечено подходящее условие прекращения.

Папка *examples/generic* содержит пример счетчика кода Грея, разрядность которого может быть настроенным. Здесь счетчик кода Грея создается путем рекурсивного добавления дополнитель-

ных битов к начальной схеме до тех пор, пока не будет достигнуто необходимое количество бит счетчика.

## 7. Тестирование под управлением сценария

Если учащиеся должны выполнять упражнения в цифровом формате, может быть полезно, если схемы, представленные студентам могут быть проверены в автоматическом режиме. Для выполнения этой проверки можно запустить Digital из командной строки. Вызов осуществляется следующим образом:

```
java -cp Digital.jar CLI test [file to test] [-tests [optional file with test cases]]
```

Если указан только проверяемый файл, выполняются тестовые случаи в этом файле. Таким образом, могут быть выполнены тестовые примеры, созданные самими студентами.

Если указано второе имя файла, тестовые примеры берутся из второго файла и первой схемы. проверяется с помощью этих тестовых примеров. Поэтому второй файл обычно будет содержать образец решения. чьи тестовые примеры полны и верны. Схема, содержащаяся во втором файле, игнорируется. Из него взяты только тестовые примеры.

Чтобы протестировать представленную схему на примере решения, имена сигналов входов и выходы в обеих цепях должны совпадать.

## 8. Часто задаваемые вопросы

### Как переместить провод?

Выберите одну из конечных точек с помощью прямоугольного выделения. Затем переместите эту точку с помощью мыши. Вы также можете выбрать провод с помощью CTRL + кнопка мыши.

### Как удалить провод?

Выберите одну из конечных точек и нажмите *DEL* или щелкните корзину. Вы также можете выбрать провод с помощью CTRL + кнопка мыши.

### Как переместить компонент, включая все подключенные провода?

Выделите компонент прямоугольным выделением. Выборка должна включать все составная часть. Затем переместите компонент, включая провода, с помощью мыши.

### Компонент не подключен к проводу, хотя контакты находятся на проводе.

Вывод подключается к проводу только в том случае, если у провода есть конечная точка на выводе.

### Если имена выводов в цепи длинные, они становятся нечитаемыми, когда схема встроена. Что я могу сделать?

Ширину блока можно увеличить с помощью пункта меню *Изменить* → *настройки контура*.

### Неоптимальный порядок выводов встроеной схемы. Как это можно изменить?

Последовательность можно изменить с помощью пункта меню *Редактировать* → *Порядок входов* или *Изменить* → *порядок выводов*.

### Когда начинается симуляция, провод становится серым. Что это значит?

Светло-зеленый и темно-зеленый цвета используются для обозначения высокого и низкого состояния. Серый цвет означает, что провод находится в состоянии высокого Z.

**У меня есть таблица истинности. Как вычислить минимизированные логические уравнения?**

В меню *Анализ* выберите запись *Синтез*. Затем войдите в таблицу истинности. Внизу окна вы можете найти соответствующее логическое уравнение. Если вы введете более одного зависимой переменной открывается новое окно, в котором показаны все логические уравнения.

**Я ввел таблицу истинности, но здесь показано более одного логического уравнения. Какой из них правильный?**

Минимизация логического уравнения может привести к множеству уравнений, описывающих одну и ту же функцию. Цифровой показывает их все, и все они составляют одну и ту же таблицу истинности. Могут быть различия в зависимости от «безразлично» в таблице истинности.

**У меня есть таблица истинности. Как создать схему, представляющую таблицу истинности?**

В меню *Анализ* выберите запись *Синтез*. Затем войдите в таблицу истинности. Таблицу можно редактировать с помощью меню *Создать* или *Изменить*. В меню *Create* вы можете создать схему с помощью элемента *Circuit*.

**Как отредактировать имя сигнала в таблице истинности?**

Щелкните правой кнопкой мыши имя в заголовке таблицы, чтобы изменить имя.

**У меня есть логическое уравнение. Как создать схему?**

В меню *Анализ* выберите запись *Выражение*. Затем введите уравнение.

**Как создать таблицу истинности из логического уравнения?**

В меню *Анализ* выберите запись *Выражение*. Затем введите выражение. Затем создайте схему и в меню *Анализ* используйте запись *Анализ*, чтобы создать истину. Таблица.

**Как создать файл JEDEC из заданной цепи?**

В меню *Анализ* выберите запись *Анализ*. Затем в меню *Создать*. в новом окне выберите нужное устройство в подменю *Устройство*.

**При создании файла JEDEC: как назначить пин-код определенному сигналу?**

На соответствующих входах и выходах вы можете ввести количество выводов в диалог настройки вывод.

**Я создал файл JEDEC. Как запрограммировать его на GAL16v8 или GAL22v10?**

Для программирования такой микросхемы необходимо специальное программаторное оборудование.

**Я создал схему, которую хочу использовать во многих других схемах. Как я могу сделать это, не копируя файл снова и снова в соответствующие папки?**

Схему можно сохранить в папке "lib". Тогда он доступен во всех остальных схемах.

## 9. Сочетания клавиш

<b>Space</b>	Запускает или останавливает моделирование.
<b>F6</b>	Открывает диалоговое окно таблицы измерений.
<b>F7</b>	Бежать до конца.
<b>F8</b>	Выполнить тестовые примеры.
<b>C</b>	Один тактовый шаг (работает только при работающем моделировании и только при наличии одного часовая составляющая).
<b>V</b>	Выполнить один шаг гейта.
<b>B</b>	Выполнять все шаги с одним вентиляем, пока схема не стабилизируется или, если присутствует разрывной компонент, до разрыва.
<b>F9</b>	Анализ схемы.
<b>CTRL-A</b>	Выбрать все.
<b>CTRL-X</b>	Вырезает выбранные компоненты в буфер обмена.
<b>CTRL-C</b>	Копирует выбранные компоненты в буфер обмена.
<b>CTRL-V</b>	Вставляет компоненты из буфера обмена.
<b>CTRL-D</b>	Дублировать текущий выбор без изменения буфера обмена.
<b>R</b>	При вставке компоненты поворачиваются.
<b>L</b>	Снова вставляет последний вставленный компонент.
<b>T</b>	Вставляет новый туннель.
<b>CTRL-N</b>	Новая цепь.
<b>CTRL-O</b>	Обрыв цепи.
<b>CTRL-S</b>	Сохраните схему.
<b>CTRL-Z</b>	Отменить последнее изменение.
<b>CTRL-Y</b>	Вернуть последнее отмененное изменение.
<b>P</b>	Программирует диод или FGFET.
<b>D</b>	При рисовании провода переключается в диагональный режим.
<b>F</b>	При рисовании линии меняет ориентацию.
<b>S</b>	Разделяет один провод на два провода.
<b>ESC</b>	Прервать текущее действие.
<b>Del</b>	Удаляет выбранные компоненты.
<b>Backspace</b>	Удаляет выбранные компоненты.
<b>+</b>	Увеличивает количество входов в компоненте, на который указывает мышь. Если он используется с константы, значение увеличивается.
<b>-</b>	Уменьшает количество входов в компоненте, на который указывает мышь. Если он используется с константы, значение уменьшается.



<b>CTRL +</b>	Увеличить масштаб.
<b>CTRL -</b>	Уменьшить.
<b>F1</b>	По размеру.
<b>F5</b>	Показать или скрыть древовидное представление компонентов.

## В Настройки

Ниже описаны доступные настройки симулятора.

### Настройки

В глобальных настройках симулятора указываются, среди прочего, вещи, язык, форма символа или пути внешних инструментов.

#### Атрибуты

Используйте формы IEEE 91-1984

Используйте формы IEEE 91-1984 вместо прямоугольных

Язык

Язык графического интерфейса. Вступит в силу только после перезапуска.

Формат

Экранный формат выражений.

Цветовая схема

Цветовая схема

Цвета, определяемые пользователем

Цвета, определяемые пользователем

Древовидное представление компонентов отображается при запуске.

Если установлено, представление дерева компонентов включается при запуске.

Показать сетку

Показывает сетку в главном окне.

Показать количество проводов на шине.

**ВНИМАНИЕ:** значение обновляется только при запуске моделирования.

На главной панели нет подсказок для компонентов.

Если установлено, подсказки для компонентов на главной панели не отображаются. Эти подсказки могут сильно раздражать, особенно в презентации.

Подсказки по инструментам для проводов

Если установлено, линии выделяются при наведении на них указателя мыши.

Библиотека

Папка, которая содержит библиотеку с предопределенными подсхемами. Содержит, например, компоненты серии 74xx. Вы также можете добавить свои собственные схемы, сохранив их в этом месте. Необходимо убедиться, что имена всех файлов в этой папке и во всех подпапках уникальны.

Библиотека Java

Файл jar, содержащий дополнительные компоненты, реализованные на Java.

Установщик ATF15xx

Путь к установщику ATF15xx. Войдите в каталог, содержащий файлы fit15xx.exe, предоставленные Microchip (ранее ATMEL).

ATMISP

Путь к исполняемому файлу ATMISP.exe. Если установлено, программное обеспечение ATMISP может запускаться автоматически!

GNDL

Путь к исполняемому файлу ghdl. Требуется только в том случае, если вы хотите использовать ghdl для моделирования компоненты, определенные с помощью VHDL.

## IVerilog

Путь к папке установки Icarus Verilog. Требуется только в том случае, если вы хотите использовать iverilog для моделирования компоненты, определенные с помощью Verilog.

### Конфигурация цепочки инструментов

Используется для настройки интеграции цепочки инструментов. Позволяет запускать внешние инструменты, например для программирования FPGA или подобного.

### Размер шрифта меню [%]

Размер шрифтов, используемых в меню, в процентах от размера по умолчанию.

### Используйте щелчки мышью в macOS.

Использует щелчок с нажатой клавишей CTRL вместо щелчка правой кнопкой мыши.

### Используйте Equals-Key

Используйте клавишу равенства вместо клавиши плюса. Это всегда полезно, если знак плюса не является первичным ключом, а является вторым назначением ключа. равно символ, например для американской или французской раскладки клавиатуры.

### Показать диалог для автоматического переименования туннелей.

Если установлено, диалоговое окно для автоматического переименования всех туннелей с тем же именем отображается после туннель был переименован.

## Настройки контура

Параметры конкретной схемы влияют на поведение в настоящее время разомкнутая цепь. Например, форма, которая представляет схему, когда она встроена в другие схемы. Эти настройки сохраняются вместе со схемой.

### Атрибуты

#### Метка

Имя этого элемента.

#### Ширина

Ширина символа, если эта цепь используется в качестве компонента в другой цепи.

#### Цвет фона

Цвет фона схемы, когда она встроена в другую схему. Не является используется для пакетов DIL.

#### Описание

Краткое описание этого элемента и его использования.

#### Модификация заблокирована

Схема заблокирована. Возможна настройка диодов и FGF-FET.

#### Форма

Форма, которая будет использоваться для представления схемы в схеме внедрения. В «Простом» режиме входные данные отображаются слева, а выходы - справа. прямоугольник. С помощью «Layout» положение входов и выходов и их ориентация в схеме определяет положение штифтов. Здесь могут быть булавки вверху или внизу. При выборе «DIL-Chip» для отображения схемы используется корпус DIL. Номера контактов входов и выходы в этом случае определите положение штифтов.

#### Пользовательская форма

Импорт файла SVG

#### Высота

Высота символа, если эта цепь используется в качестве компонента в другой цепи.

#### Количество контактов DIL

Количество контактов. Ноль означает, что количество выводов определяется автоматически.

## Содержимое ПЗУ

Содержимое всех используемых ПЗУ

Показывать значения измерений при запуске моделирования

При запуске моделирования отображается таблица с измеренными значениями.

Показать график измерений при запуске моделирования

При запуске моделирования отображается график с измеренными значениями.

Показывать график измерений в пошаговом режиме с одним гейтом при запуске моделирования

При запуске моделирования график с измеренными значениями в Показан пошаговый режим гейта. Все изменения ворот отражены в графике.

Предварительная загрузка памяти программы при запуске.

При моделировании процессора, который использует устройство RAM в качестве памяти программ, этот процессор сложно запустить, потому что содержимое RAM всегда инициализируется нулями в начале моделирования. Этот параметр позволяет загружать данные в память программы при запуске. Программная память в моделировании должна быть отмечена как таковая.

Программный файл

Файл, который должен быть загружен в память программы в начале моделирование.

Схема является общей

Позволяет создать общую схему.

## С Интерфейс командной строки

```
java -cp Digital.jar CLI
```

```
test -circ [String] [-tests [String]] [-allowMissingInputs]:
```

Первое имя файла определяет схему, которую нужно протестировать. Если указано второе имя файла, тестовые примеры выполняются из этого файла. Если второе имя файла не указано, тесты выполняются из первого файла.

Параметры:

```
-circ [String(def: )]
```

Имя файла для тестирования.

```
[-tests [String(def: )]]
```

Имя файла с тестовыми примерами.

```
[-allowMissingInputs(def: false)]
```

Разрешает отсутствие входов в цепи, которые определено в тестовом примере.

Это может быть полезно, если есть несколько возможных решений, которые могут зависеть от разных входов.

```
svg -dig [String] [-svg [String]] [-ieeee] [-LaTeX] [-pinsInMathMode] [-hideTest] [-noShapeFilling] [-smallIO] [-noPinMarker] [-thinnerLines] [-highContrast] [-monochrome]:
```

Может использоваться для создания файла SVG из схемы.

Параметры:

```
-dig [String(def: )]
```

Имя файла схемы.

```
[-svg [String(def: )]]
```

Имя записываемого файла SVG.

```
[-ieeee(def: false)]
```

Используйте символы IEEE.

```
[-LaTeX(def: false)]
```

Текст вставлен в нотации LaTeX. Для дальнейшей обработки требуется Inkscape.

```
[-pinsInMathMode(def: false)]
```

Для меток контактов используйте математический режим, даже если индексы отсутствуют.

```
[-hideTest(def: false)]
```

Скрыть тестовые наборы

```
[-noShapeFilling(def: false)]
```

Полигоны не заполнены.

```
[-smallIO(def: false)]
```

Входы и выходы представлены в виде маленьких кружков.

`[-noPinMarker(def: false)]`

Синие и красные маркеры булавки на символах опущены.

`[-thinnerLines(def: false)]`

Если установлено, линии рисуются немного тоньше.

`[-highContrast(def: false)]`

Провода и текст контактов отображаются черным цветом.

`[-monochrome(def: false)]`

Используются только серые цвета.

`stats -dig [String] [-csv [String]]:`

Создает файл CSV, содержащий статистику схемы. Все используемые компоненты перечислены в файле CSV.

Параметры:

`-dig [String(def: )]`

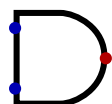
Имя файла схемы.

`[-csv [String(def: )]]`

Имя создаваемого файла CSV. Если этот параметр отсутствует, таблица записывается в стандартный вывод.

## D Компоненты

### 1. Логика



#### 1.1. И

Двоичный логический элемент И. Возвращает высокий уровень, только если все входы также установлены в высокий уровень. Также можно использовать шины с несколькими битами в качестве входов и выходов. В этом случае выполняется поразрядное И. Это означает, что младшие биты всех входов связаны логическим оператором И и выводятся как младшие биты на выходе. То же самое происходит с битом 1, битом 2 и так далее. Возможность экспорта в VHDL / Verilog.

Входные данные

In\_1

1. входное значение для логической операции.

In\_2

2. входное значение для логической операции.

Выводы

out

Возвращает результат логической операции.

Атрибуты

Биты данных

Количество используемых битов данных.

Количество входов

Количество используемых входов. Каждый вход должен быть подключен.

Инвертированные входы

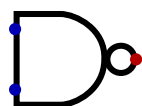
Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

Широкая форма

Использует более широкую форму для визуализации ворот.



#### 1.2. И-НЕ

Комбинация AND и NOT. Возвращает 0, только если для всех входов установлено значение 1. Если для одного из входов установлено значение 0, для выхода устанавливается значение 1. Также можно использовать шины с несколькими битами на вход. В этом случае операция применяется к каждому бит входов. Возможность экспорта в VHDL / Verilog.

#### Входные данные

In\_1

1. входное значение для логической операции.

In\_2

2. входное значение для логической операции.

#### Выводы

out

Возвращает результат логической операции.

#### Атрибуты

Биты данных

Количество используемых битов данных.

Количество входов

Количество используемых входов. Каждый вход должен быть подключен.

Инвертированные входы

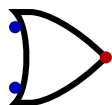
Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

Широкая форма

Использует более широкую форму для визуализации ворот.



### 1.3. ИЛИ

Логический элемент логического ИЛИ. Возвращает 1, если для одного из входов установлено значение 1. Если все входы установлены на 0, выход также устанавливается на 0. Также можно использовать шины с несколькими битами в качестве входов и выходов. В этом случае выполняется поразрядное ИЛИ. Это означает, что младшие биты всех входов связаны с помощью ИЛИ и выводятся как младшие биты на выходе. То же самое происходит с битом 1, битом 2 и так далее. Возможность экспорта в VHDL / Verilog.

#### Входные данные

In\_1

1. входное значение для логической операции.

In\_2

2. входное значение для логической операции.

#### Выводы

out

Возвращает результат логической операции.

#### Атрибуты

Биты данных

Количество используемых битов данных.

Количество входов

Количество используемых входов. Каждый вход должен быть подключен.



Инвертированные входы

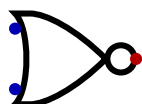
Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

Широкая форма

Использует более широкую форму для визуализации ворот.



#### 1.4. ИЛИ-НЕ

Комбинация ИЛИ и НЕ. Возвращает 0, если для одного из входов установлено значение 1. Если все входы установлены на 0, выход также устанавливается на 1. Также можно использовать шины с несколькими битами на вход. В этом случае операция применяется к каждому бит входов. Возможность экспорта в VHDL / Verilog.

Входные данные

In\_1

1. входное значение для логической операции.

In\_2

2. входное значение для логической операции.

Выводы

out

Возвращает результат логической операции.

Атрибуты

Биты данных

Количество используемых битов данных.

Количество входов

Количество используемых входов. Каждый вход должен быть подключен.

Инвертированные входы

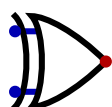
Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

Широкая форма

Использует более широкую форму для визуализации ворот.



#### 1.5. XOR

Если используются два входа, выход равен 0, если оба входных бита равны. В противном случае на выходе будет установлено значение 1. Если используется более двух входов, он ведет себя как каскадные вентили XOR ( $A \text{ XOR } B \text{ XOR } C = (A \text{ XOR } B) \text{ XOR } C$ ). Также можно

использовать шины с несколькими битами на вход. В этом случае операция применяется к каждому бит входов. Возможность экспорта в VHDL / Verilog.

Входные данные

In\_1

1. входное значение для логической операции.

In\_2

2. входное значение для логической операции.

Выводы

out

Возвращает результат логической операции.

Атрибуты

Биты данных

Количество используемых битов данных.

Количество входов

Количество используемых входов. Каждый вход должен быть подключен.

Инвертированные входы

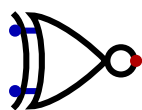
Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

Широкая форма

Использует более широкую форму для визуализации ворот.



## 1.6. XNOr

Комбинация XOR и NOT. Входы комбинируются с операцией XOR. Результат этой операции инвертируется. Также можно использовать шины с несколькими битами на вход. В этом случае операция применяется к каждому бит входов. Возможность экспорта в VHDL / Verilog.

Входные данные

In\_1

1. входное значение для логической операции.

In\_2

2. входное значение для логической операции.

Выводы

out

Возвращает результат логической операции.

Атрибуты

Биты данных

Количество используемых битов данных.

Количество входов

Количество используемых входов. Каждый вход должен быть подключен.

Инвертированные входы

Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

Широкая форма

Использует более широкую форму для визуализации ворот.



## 1.7. НЕ

Инвертирует входное значение. 1 становится 0, а 0 становится 1. Также можно использовать шину с несколькими битами на вход. В этом случае операция применяется к каждому бит входов. Возможность экспорта в VHDL / Verilog.

Входные данные

in

Вход элемента НЕ.

Выводы

out

Инвертированное входное значение.

Атрибуты

Биты данных

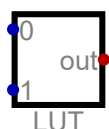
Количество используемых битов данных.

Вращение

Ориентация элемента в цепи.

Широкая форма

Использует более широкую форму для визуализации ворот.



## 1.8. Таблица поиска

Получает выходное значение из сохраненной таблицы. Таким образом, этот вентиль может имитировать любой комбинаторный вентиль. Возможность экспорта в VHDL / Verilog.

Входные данные

0

Ввод 0. Этот вход в сочетании со всеми другими входами определяет адрес возвращаемого сохраненного значения.

1

Ввод 1. Этот вход в сочетании со всеми другими входами определяет адрес возвращаемого сохраненного значения.

## Выводы

out

Возвращает сохраненное значение по адресу, заданному через входы.

## Атрибуты

Биты данных

Количество используемых битов данных.

Количество входов

Количество используемых входов. Каждый вход должен быть подключен.

Метка

Имя этого элемента.

Данные

Значения, хранящиеся в этом элементе.

Вращение

Ориентация элемента в цепи.

## 2. ВВОД-ВЫВОД



### 2.1. Вывод

Может использоваться для отображения выходного сигнала в цепи. Этот элемент также используется для подключения схемы к схеме внедрения. В этом случае соединение будет двунаправленным. Также используется для присвоения номера контакта, если генерируется код для CPLD или FPGA. Возможность экспорта в VHDL / Verilog.

#### Входные данные

in

Это значение используется для выходного соединения.

#### Атрибуты

Биты данных

Количество используемых битов данных.

Метка

Имя этого элемента.

Описание

Краткое описание этого элемента и его использования.

Вращение

Ориентация элемента в цепи.

Формат числа

Формат, используемый для отображения чисел.

цифры с фиксированной запятой

Количество дробных двоичных цифр

Номер контакта

Номер этого пина. Используется для представления схемы в виде пакета DIL и назначение контактов при программировании CPLD. Если битов несколько, все номера контактов можно указать в виде списка, разделенного запятыми.

Показать на графике измерений  
Показывает значение на графике измерения.



## 2.2. светодиод

Светодиод может использоваться для визуализации выходного значения. Принимает один бит. Загорается, если вход установлен на 1.

Входные данные

in  
Светодиодный вход. Светодиод загорается, если вход установлен на 1.

Атрибуты

Метка  
Имя этого элемента.  
Цвет  
Цвет элемента.  
Вращение  
Ориентация элемента в цепи.  
Размер  
Размер фигуры в цепи.



## 2.3. Ввод

Может использоваться для интерактивного управления входным сигналом в цепи с помощью мышь. Этот элемент также используется для подключения схемы к схеме внедрения. В этом случае соединение будет двунаправленным. Также используется для присвоения номера контакта, если генерируется код для CPLD или FPGA. Возможность экспорта в VHDL / Verilog.

Выводы

out  
Дает значение, которое связано с этим входом.

Атрибуты

Биты данных  
Количество используемых битов данных.  
Метка  
Имя этого элемента.  
Описание  
Краткое описание этого элемента и его использования.  
Вращение  
Ориентация элемента в цепи.  
По умолчанию  
Это значение устанавливается, если цепь запущена. "Z" означает высокий уровень z.

**Вход с тремя состояниями**

Если установлено, вход может находиться в состоянии высокого *z*. На входном компоненте это также разрешено, если в качестве значения по умолчанию установлено high-*z* ("Z").

**Нет нулевого вывода.**

Избегает нулевого вывода. Это особенно полезно при настройке цепей реле. Может быть активирован, только если разрешен вывод с высоким *z*.

**Формат числа**

Формат, используемый для отображения чисел.

**цифры с фиксированной запятой**

Количество дробных двоичных цифр

**Номер контакта**

Номер этого пина. Используется для представления схемы в виде пакета DIL и назначение контактов при программировании CPLD. Если битов несколько, все номера контактов можно указать в виде списка, разделенного запятыми.

**Показать на графике измерений**

Показывает значение на графике измерения.



## 2.4. Вход часов

Тактовый сигнал. Его можно контролировать по часам реального времени. В зависимости от сложности схемы достигнутая тактовая частота может быть меньше выбранного значения. Если частота больше 50 Гц, графическое представление схемы больше не будет обновляться при каждом такте, чтобы цвета проводов больше не обновлялись. Если часы реального времени не активированы, ими можно управлять щелчком мыши. Также используется для присвоения номера контакта, если генерируется код для CPLD или FPGA. Возможность экспорта в VHDL / Verilog.

**Выводы****C**

Переключение между 0 и 1 с выбранной тактовой частотой.

**Атрибуты****Метка**

Имя этого элемента.

**Запуск часов реального времени**

Если включено, часы времени выполнения запускаются, когда цепь началось

**Частота/Гц**

Частота реального времени, используемая для часов реального времени

**Вращение**

Ориентация элемента в цепи.

**Номер контакта**

Номер этого пина. Используется для представления схемы в виде пакета DIL и назначение контактов при программировании CPLD. Если битов несколько, все номера контактов можно указать в виде списка, разделенного запятыми.



## 2.5. Кнопка

Простая кнопка, которая возвращается в исходное состояние при отпускании.

Выводы

out

Выходной сигнал кнопки.

Атрибуты

Метка

Имя этого элемента.

Активный минимум

Если выбрано, вывод будет низким, если компонент активен.

Сопоставить клавиатуре

Кнопка привязана к клавиатуре. Чтобы использовать клавиши курсора, используйте кнопки ВВЕРХ, ВНИЗ, ВЛЕВО или ВПРАВО в качестве метки.

Вращение

Ориентация элемента в цепи.

Показать на графике измерений

Показывает значение на графике измерения.



## 2.6. DIP-переключатель

Простой DIP-переключатель, который может выводить как высокий, так и низкий уровень.

Выводы

out

Выходное значение переключателя.

Атрибуты

Метка

Имя этого элемента.

Описание

Краткое описание этого элемента и его использования.

Вращение

Ориентация элемента в цепи.

Высокий уровень вывода

Выходное значение DIP-переключателя по умолчанию при запуске моделирования.



## 2.7. Зонд

Значение измерения, которое может отображаться на графике данных или в таблице измерений. Этот компонент можно использовать для простого наблюдения за значениями встроенных схем. Не влияет на симуляцию.

Входные данные

in  
Значение измерения.

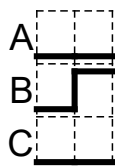
Атрибуты

Метка  
Имя этого элемента.

Вращение  
Ориентация элемента в цепи.

Формат числа  
Формат, используемый для отображения чисел.  
цифры с фиксированной запятой  
Количество дробных двоичных цифр

Показать на графике измерений  
Показывает значение на графике измерения.



## 2.8. График данных

Показывает график данных внутри панели схемы. Вы можете нанести на график полные тактовые циклы или отдельные изменения гейта. Не влияет на симуляцию.

Атрибуты

Показать шаги одиночного шлюза  
Показывает все пошаговые шаги на графике.

Максимальное количество шагов для отображения  
Максимальное количество сохраняемых значений. Если достигается максимальное количество, самые старые значения отбрасываются.

Привязка к сетке  
Если установлено, компонент выравнивается по сетке.





## 2.9. График инициируемых данных

Показывает график измеренных значений, при этом измеренные значения сохраняются только в том случае, если входной сигнал изменится. Сохранение происходит, когда цепь стабилизируется. Триггер не запускает измерения, как в реальном осциллографе, но каждое событие триггера сохраняет одно значение измерения для каждого из показанных сигналов. В качестве прямого входа используется только триггер. Входы и выходы схемы, триггеры, регистры и компонент зонда можно использовать в качестве сигналов. Это можно активировать в соответствующих компонентах.

Входные данные

T

Изменение на этом входе вызывает сохранение измеренных значений.

Атрибуты

Метка

Имя этого элемента.

Триггер

Условие запуска для записи данных.

Максимальное количество шагов для отображения

Максимальное количество сохраняемых значений. Если достигается максимальное количество, самые старые значения отбрасываются.

## 3. ввод-вывод - Индикаторы



### 3.1. RGB-светодиод

Светодиод RGB, цвет которого можно контролировать с помощью трех входов. К каждому из трех входов подключен канал цвета.

Входные данные

R

Канал красного цвета.

G

Канал зеленого цвета.

B

Канал синего цвета.

Атрибуты

Биты данных

Количество используемых битов данных.

- Метка
  - Имя этого элемента.
- Вращение
  - Ориентация элемента в цепи.
- Размер
  - Размер фигуры в цепи.



### 3.2. Светодиод с двумя подключениями.

Светодиод с подключениями для катода и анода. Светодиод загорается, если анод подключен к высокому уровню, а катод подсоединен к низкому уровню. Этот светодиод нельзя использовать в качестве понижающего резистора. Он действует исключительно как элемент отображения. Показанный резистор предназначен только для обозначения необходимого последовательного резистора для ограничения тока.

Входные данные

- A
  - Анодное соединение светодиода.
- C
  - Катодное соединение светодиода.

Атрибуты

- Метка
  - Имя этого элемента.
- Цвет
  - Цвет элемента.
- Вращение
  - Ориентация элемента в цепи.



### 3.3. Кнопка со светодиодом

Простая кнопка, которая возвращается в исходное состояние при отпускании. На кнопке есть светодиод, который можно переключать с помощью входного сигнала.

Входные данные

- in
  - Вход для управления светодиодом.

Выводы

- out
  - Выходной сигнал кнопки.

## Атрибуты

## Метка

Имя этого элемента.

## Активный минимум

Если выбрано, вывод будет низким, если компонент активен.

## Сопоставить клавиатуре

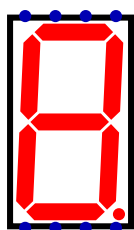
Кнопка привязана к клавиатуре. Чтобы использовать клавиши курсора, используйте кнопки ВВЕРХ, ВНИЗ, ВЛЕВО или ВПРАВО в качестве метки.

## Цвет

Цвет элемента.

## Вращение

Ориентация элемента в цепи.



## 3.4. Семисегментный дисплей

Семисегментный дисплей, каждый сегмент имеет свой собственный вход управления.

## Входные данные

a

Этот ввод управляет верхней горизонтальной линией.

b

Этот ввод управляет верхней, правой вертикальной линией.

c

Этот ввод управляет нижней правой вертикальной линией.

d

Этот ввод контролирует нижнюю горизонтальную линию.

e

Этот ввод управляет нижней левой вертикальной линией.

f

Этот ввод управляет верхней, левой вертикальной линией.

g

Этот ввод управляет средней горизонтальной линией.

dp

Этот ввод управляет десятичной точкой.

## Атрибуты

## Цвет

Цвет элемента.

## Обычное соединение

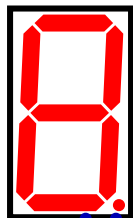
Если этот параметр выбран, также моделируется общий ввод катода или анода.

## Обычный

Тип обычного подключения.

### Избегать мерцания

Невозможно увеличить частоту настолько, чтобы мигание исчезает. Чтобы все же подавить мерцание, можно включить «послесвечение» для светодиодов с помощью этого вариант. Если этот параметр включен, светодиоды остаются включенными, даже если один из выводов переключается на высокий  $z$ . Это имитирует частоту выше критической частоты слияния мерцания.



### 3.5. Семисегментный шестнадцатеричный дисплей

Семисегментный дисплей с 4-битным шестнадцатеричным входом

Входные данные

`d`  
Значение на этом входе отображается на дисплее.

`dp`  
Этот ввод управляет десятичной запятой.

Атрибуты

Цвет  
Цвет элемента.

Размер  
Размер фигуры в цепи.



### 3.6. 16-сегментный дисплей

Вход светодиода имеет 16 бит, которые управляют сегментами. Второй вход управляет десятичной точкой.

Входные данные

`led`  
16-битная шина для управления светодиодами.

`dp`  
Этот ввод управляет десятичной точкой.

Атрибуты

- Цвет
  - Цвет элемента.
- Размер
  - Размер фигуры в цепи.



### 3.7. Лампочка

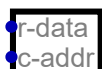
Лампочка с двумя подключениями. Если ток течет, лампочка загорается! Направление тока значения не имеет. Лампа загорается, когда входы имеют разные значения. Лампочка ведет себя аналогично вентилю XOR.

Входные данные

- A
  - Подключение
- B
  - Подключение

Атрибуты

- Метка
  - Имя этого элемента.
- Цвет
  - Цвет элемента.
- Вращение
  - Ориентация элемента в цепи.



Светодиодная матрица

### 3.8. Светодиодная матрица

Матрица светодиодов. Светодиоды отображаются в отдельном окне. Светодиоды столбца дисплея управляются словом данных. На другом входе текущий столбец выбрано. Так реализован мультиплексный дисплей. Во время симуляции светодиоды могут гореть бесконечно, чтобы дисплей не мерцал.

Входные данные

- r-data
  - Состояние строки светодиодных индикаторов столбца. Каждый бит в этом слове данных представляет состояние строки текущего столбца.
- c-addr
  - Номер текущего столбца, состояние которого в настоящее время видно на другом входе.

Атрибуты

- Метка
  - Имя этого элемента.

**Строки**

Задаёт количество строк, указав количество битов в слове-строке.

**Адресные биты столбцов**

Адреса отдельным столбцам. Три бита означают восемь столбцов.

**Цвет**

Цвет элемента.

**Избегать мерцания**

Невозможно увеличить частоту настолько, чтобы мигание исчезает. Чтобы все же подавить мерцание, можно включить «послесвечение» для светодиодов с помощью этого вариант. Если этот параметр включен, светодиоды остаются включенными, даже если один из выводов переключается на высокий z. Это имитирует частоту выше критической частоты слияния мерцания.

**Вращение**

Ориентация элемента в цепи.

## 4. ввод-вывод - Электромеханика



### 4.1. Поворотный энкодер

Поворотная ручка с поворотным энкодером. Используется для обнаружения вращательных движений.

**Выходы**

A

сигнал энкодера A

B

сигнал энкодера B

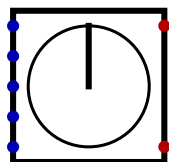
**Атрибуты**

Метка

Имя этого элемента.

Вращение

Ориентация элемента в цепи.



### 4.2. Шаговый двигатель, униполярный

Униполярный шаговый двигатель с двумя концевыми выключателями. Поддерживаются полный шаговый привод, полушаговый привод и волновой привод.

## Входные данные

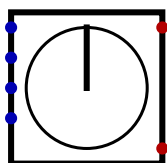
- P0  
Фаза 0
- P1  
Фаза 1
- P2  
Фаза 2
- P3  
Фаза 3
- com  
Подключение катушки с общим центром

## Выводы

- S0  
Переключатель конечного положения 0 становится равным 1, когда угол двигателя равен  $0^\circ$ .
- S1  
Переключатель предельного положения 1 становится равным 1, когда угол двигателя составляет  $180^\circ$ .

## Атрибуты

- Метка  
Имя этого элемента.
- Инвертированный вывод  
Если выбрано, вывод инвертируется.
- Вращение  
Ориентация элемента в цепи.



### 4.3. Шаговый двигатель, биполярный

Биполярный шаговый двигатель с двумя концевыми выключателями. Поддерживаются полный шаговый привод, полушаговый привод и волновой привод.

## Входные данные

- A+  
Катушка A, положительная
- A-  
Катушка A, отрицательная
- B+  
Катушка B, положительная
- B-  
Катушка B, отрицательная

**Выводы**

S0

Переключатель конечного положения 0 становится равным 1, когда угол двигателя равен 0 °.

S1

Переключатель предельного положения 1 становится равным 1, когда угол двигателя составляет 180 °.

**Атрибуты**

Метка

Имя этого элемента.

Инвертированный вывод

Если выбрано, вывод инвертируется.

Вращение

Ориентация элемента в цепи.

**5. ввод-вывод - Периферийные устройства**

Клавиатура

**5.1. Клавиатура**

Клавиатура, которую можно использовать для ввода текста. Этот компонент буферизует ввод, который затем может быть считан. Для ввода текста открывается отдельное окно.

**Входные данные**

C

Часы. По переднему фронту из буфера удаляется самый старый символ.

en

Если высокий, выход D активен и выводится один символ. Он также включает вход часов.

**Выводы**

D

Последний набранный символ или ноль, если символ недоступен. Вывод - это 16-битное значение символа Java.

av

Этот вывод показывает, что символы доступны. Его можно использовать для запуска прерывания.

**Атрибуты**

Метка

Имя этого элемента.

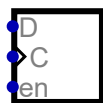
Инвертированные входы

Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.





Терминал

## 5.2. Терминал

Вы можете записывать символы ASCII в этот терминал. Терминал открывает собственное окно для визуализации вывода.

Входные данные

D

Данные для записи в терминал

C

Часы. Нарастающий фронт записывает значение на входе в окно терминала.

en

Высокий уровень на этом входе включает вход часов.

Атрибуты

Символов в строке

Количество символов, отображаемых в одной строке.

Строки

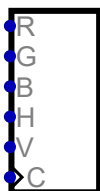
Количество отображаемых строк.

Метка

Имя этого элемента.

Вращение

Ориентация элемента в цепи.



Монитор VGA

## 5.3. Монитор VGA

Анализирует входящие видеосигналы и отображает соответствующую графику. Поскольку моделирование не может выполняться в реальном времени, в дополнение к видеосигналам требуются пиксельные часы.

## Входные данные

R	Компонент красного цвета
G	Компонент зеленого цвета
B	Компонент синего цвета
H	Сигнал горизонтальной синхронизации
V	Сигнал вертикальной синхронизации
C	Пиксельные часы

## Атрибуты

Метка	Имя этого элемента.
Вращение	Ориентация элемента в цепи.

**5.4. MIDI**

Использует систему MIDI для воспроизведения нот.

## Входные данные

N	Примечание
V	Объем
OnOff	Если установлено, это означает нажатие клавиши клавиатуры (событие нажатия клавиши), если не установлен, это означает отпускание клавиши (событие нажатия клавиши).
en	Включает компонент
C	Часы

## Атрибуты

Метка	Имя этого элемента.
MIDI-канал	Выбирает MIDI-канал для использования.

MIDI-инструмент

Используемый MIDI-инструмент.

Разрешить изменение программы

Добавляет новый входной ПК. Если этот вход установлен на высокий, значение на входе N используется для изменения программы (инструмента).

Вращение

Ориентация элемента в цепи.

## 6. Провода



### 6.1. Земля

Подключение к земле. Вывод всегда равен нулю. Возможность экспорта в VHDL / Verilog.

Выводы

out

Вывод всегда возвращает 0.

Атрибуты

Биты данных

Количество используемых битов данных.

Метка

Имя этого элемента.

Вращение

Ориентация элемента в цепи.



### 6.2. Напряжение питания

Подключение к источнику питания. Вывод всегда один. Возможность экспорта в VHDL / Verilog.

Выводы

out

Этот вывод всегда возвращает 1.

Атрибуты

Биты данных

Количество используемых битов данных.

Метка

Имя этого элемента.

Вращение

Ориентация элемента в цепи.

1•

### 6.3. Постоянное значение

Компонент, который возвращает заданное значение как простое постоянное значение. Значение можно установить в диалоговом окне атрибута. Возможность экспорта в VHDL / Verilog.

Выводы

out  
Возвращает заданное значение как константу.

Атрибуты

Биты данных  
Количество используемых битов данных.  
Значение  
Значение константы.  
Вращение  
Ориентация элемента в цепи.  
Формат числа  
Формат, используемый для отображения чисел.  
цифры с фиксированной запятой  
Количество дробных двоичных цифр



### 6.4. Туннель

Соединяет компоненты без проводов. Все элементы туннеля, имеющие одно и то же сетевое имя, связаны вместе. Работает только локально, поэтому невозможно подключить разные цепи. Без названия туннели игнорируются. Возможность экспорта в VHDL / Verilog.

Входные данные

in  
Подключение к туннелю.

Атрибуты

Сетевое имя  
Все сети с одинаковым именем соединяются вместе.  
Вращение  
Ориентация элемента в цепи.



### 6.5. Разделитель / Слияние

Разделяет или создает пучок проводов или шину данных с более чем одним битом. В автобусе это, например, возможно создание 16-битных соединений без необходимости прокладывать 16

отдельных проводов. Все 16 соединений можно объединить в один провод. Разветвитель имеет направление, то есть он может передавать сигналы только в одном направлении. Возможность экспорта в VHDL / Verilog.

#### Входные данные

0-3

Входные биты 0-3.

4-7

Входные биты 4-7.

#### Выводы

0-7

Выходные биты 0-7.

#### Атрибуты

##### Разделение ввода

Если, например, четыре бита, два бита и два дополнительных бита должны использоваться в качестве входов, это можно настроить с помощью «4,2,2». Число указывает количество бит. Для удобства звездочка можно использовать: 16 бит можно настроить с помощью «[Биты] \* [Число]» как «1 \* 16». Также возможно указать биты, которые будут использоваться напрямую и в любом порядке. Например, «4-7,0-3» конфигурирует биты 4-7 и 0-3. Это обозначение допускает любое расположение битов. Входные биты должны быть указаны полностью и однозначно.

##### Разделение вывода

Если, например, четыре бита, два бита и два дополнительных бита должны использоваться в качестве выходов, это можно настроить с помощью «4,2,2». Число указывает количество бит. Для удобства звездочка можно использовать: 16 бит можно настроить с помощью «[Биты] \* [Число]» как «1 \* 16». Также возможно указать биты, которые будут использоваться напрямую и в любом порядке. Например, «4-7,0-3» конфигурирует биты 4-7 и 0-3. Это обозначение допускает любое расположение битов. Выходные биты также могут выводиться несколько раз: «0-7,1-6,4-7».

##### Вращение

Ориентация элемента в цепи.

##### Зеркало

Зеркально отображает компонент в цепи.

##### Распространение

Настраивает разброс входов и выходов в схеме.



## 6.6. Драйвер

Драйвер можно использовать для подключения значения сигнала к другому проводу. Драйвер управляется входом sel. Если на входе sel низкий уровень, значит, на выходе высокий уровень z. Если на входе sel высокий уровень, на выходе устанавливается входное значение. Возможность экспорта в VHDL / Verilog.

#### Входные данные

in

Входное значение драйвера.

sel

Пин для управления драйвером. Если его значение равно 1, вход устанавливается на выход. Если значение равно 0, выход находится в состоянии высокого z.

#### Выводы

out

Если вход sel равен 1, вход передается этому выходу. Если вход sel равен 0, этот выход находится в состоянии высокого z.

#### Атрибуты

Биты данных

Количество используемых битов данных.

Отразить положение селектора

Этот параметр позволяет переместить штифт селектора на противоположную сторону плексера.

Вращение

Ориентация элемента в цепи.



### 6.7. Драйвер, инвертированный выбор

Для подключения слова данных к другой строке можно использовать драйвер. Драйвер управляется входом sel. Если на входе sel высокий уровень, значит, на выходе высокий уровень z. Если на входе sel низкий уровень, на выходе устанавливается входное значение. Возможность экспорта в VHDL / Verilog.

#### Входные данные

in

Входное значение драйвера.

sel

Пин-код для управления драйвером. Если его значение равно 0, вход передается на выход. Если значение равно 1, выход находится в состоянии высокого z.

#### Выводы

out

Если вход sel равен 1, вход передается этому выходу. Если вход sel равен 0, этот выход находится в состоянии высокого z.

#### Атрибуты

Биты данных

Количество используемых битов данных.

Отразить положение селектора

Этот параметр позволяет переместить штифт селектора на противоположную сторону плексера.

Вращение  
Ориентация элемента в цепи.



## 6.8. Задержка

Задерживает сигнал на одно время задержки распространения. Задерживает сигнал на регулируемое количество задержек гейта. Все остальные компоненты в Digital имеют задержку затвора, равную одному времени задержки распространения. Этот компонент может использоваться для реализации любой необходимой задержки распространения.

Входные данные

in  
Ввод сигнала для задержки.

Выводы

out  
Входной сигнал задерживается на время задержки в один строб.

Атрибуты

Биты данных  
Количество используемых битов данных.  
Продолжительность  
Время задержки в единицах общей задержки распространения затвора.  
Вращение  
Ориентация элемента в цепи.



## 6.9. Подтягивающий резистор

Если цепь находится в состоянии HighZ, этот резистор поднимает цепь до высокого уровня. В любом другом случае этот компонент не действует.

Выводы

out  
"слабый максимум".

Атрибуты

Биты данных  
Количество используемых битов данных.  
Вращение  
Ориентация элемента в цепи.



## 6.10. Понижающий резистор

Если цепь находится в состоянии HighZ, этот резистор подтягивает цепь к земле. В любом другом случае этот компонент не действует.

Выводы

out  
"слабый минимум".

Атрибуты

Биты данных  
Количество используемых битов данных.  
Вращение  
Ориентация элемента в цепи.



## 6.11. Не подключен

Этот компонент можно использовать для установки провода на High-Z. Если вход логического элемента установлен на высокий Z, значение чтения не определено. Обратите внимание, что на самом деле во многих случаях чрезмерное потребление тока и даже повреждение могут возникнуть, если для цифрового входа не установлен высокий или низкий уровень, но он остается неподключенным.

Выводы

out  
Этот вывод всегда выводит High-Z.

Атрибуты

Биты данных  
Количество используемых битов данных.

## 7. Плексеры



### 7.1. Мультиплексор

Компонент, который использует значение пина sel, чтобы решить, какое входное значение является установлен на выход. Возможность экспорта в VHDL / Verilog.



### Входные данные

sel

Этот ввод используется для выбора ввода данных, который выводится.

in\_0

0. ввод данных мультиплексора.

in\_1

1. ввод данных мультиплексора.

### Выводы

out

Значение выбранного входа.

### Атрибуты

Биты данных

Количество используемых битов данных.

Количество битов селектора

Количество битов, используемых для ввода селектора.

Отразить положение селектора

Этот параметр позволяет переместить штифт селектора на противоположную сторону плексера.

Вращение

Ориентация элемента в цепи.



## 7.2. Демультимплексор

Компонент, который может выводить входное значение на один из выходов. Для остальных выходов установлено значение по умолчанию. Возможность экспорта в VHDL / Verilog.

### Входные данные

sel

Этот вывод выбирает используемый выход.

in

Значение этого входа передается выбранному выходу данных.

### Выводы

out\_0

Вывод данных 0.

out\_1

Вывод данных 1.

### Атрибуты

Биты данных

Количество используемых битов данных.

Количество битов селектора

Количество битов, используемых для ввода селектора.

Отразить положение селектора

Этот параметр позволяет переместить штифт селектора на противоположную сторону плексера.

Вращение

Ориентация элемента в цепи.

По умолчанию

Это значение устанавливается, если цепь запущена. В демультимплексоре это значение устанавливается для невыбранных выходов.



### 7.3. Декодер

Один выбираемый выходной контакт - 1, все остальные выходы - 0. Возможность экспорта в VHDL / Verilog.

Входные данные

`sel`

Этот вход выбирает разрешенный выход. Выбранный выход установлен на 1. Все остальные выходы установлены на 0.

Выводы

`out_0`

Выход 0. Этот выход равен 1, если выбран входом `sel`.

`out_1`

Выход 1. Этот выход равен 1, если выбран входом `sel`.

Атрибуты

Количество битов селектора

Количество битов, используемых для ввода селектора.

Отразить положение селектора

Этот параметр позволяет переместить штифт селектора на противоположную сторону плексера.

Вращение

Ориентация элемента в цепи.



### 7.4. Селектор бит

Выбирает один бит из шины данных. Возможность экспорта в VHDL / Verilog.

Входные данные

`in`

Входная шина

`sel`

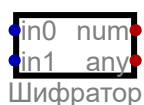
Этот вход выбирает бит

## Выводы

out  
Выбранный бит.

## Атрибуты

Количество битов селектора  
Количество битов, используемых для ввода селектора.  
Отразить положение селектора  
Этот параметр позволяет переместить штифт селектора на противоположную сторону плексера.  
Вращение  
Ориентация элемента в цепи.



## 7.5. Приоритетный шифратор

Если один из входов установлен, выводится его номер. Если одновременно задано несколько входов, выводится наибольшее число. Возможность экспорта в VHDL / Verilog.

### Входные данные

in0  
0. ввод приоритетного шифратора.  
in1  
1. ввод приоритетного шифратора.

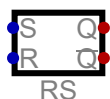
### Выводы

num  
Номер заданного ввода.  
any  
Если этот выход установлен, устанавливается по крайней мере один из входов.

### Атрибуты

Метка  
Имя этого элемента.  
Количество битов селектора  
Количество битов, используемых для ввода селектора.  
Вращение  
Ориентация элемента в цепи.

## 8. Триггеры



### 8.1. RS-триггер

Компонент для хранения одного бита. Предоставляет функции «установить» и «сбросить» для установки или сброса сохраненного бита. Если оба входа переключены на один, оба выхода также выдают один. Если оба входа возвращаются к нулю одновременно, конечное состояние будет случайным.

Входные данные

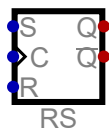
- S  
Установленный ввод.
- R  
Вход сброса.

Выводы

- Q  
Возвращает сохраненное значение.
- $\neg Q$   
Возвращает инвертированное сохраненное значение.

Атрибуты

- Метка  
Имя этого элемента.
- Инвертированные входы  
Вы можете выбрать входы, которые нужно инвертировать.
- Вращение  
Ориентация элемента в цепи.
- Зеркало  
Зеркально отображает компонент в цепи.
- По умолчанию  
Это значение устанавливается, если цепь запущена. В демультимплексоре это значение устанавливается для невыбранных выходов.
- Использовать как значение измерения  
Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.



## 8.2. RS-триггер, синхронизированный

Компонент для хранения одного бита. Предоставляет функции «установить» и «сбросить» для установки или сброса сохраненного бита. Если оба входа (S, R) установлены на переднем фронте тактового сигнала, конечное состояние будет случайным.

Входные данные

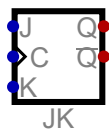
- S  
Установленный ввод.
- C  
Вход часов. Нарастающий фронт инициирует переход состояния.
- R  
Вход сброса.

Выводы

- Q  
Возвращает сохраненное значение.
- $\neg Q$   
Возвращает инвертированное сохраненное значение.

Атрибуты

- Метка  
Имя этого элемента.
- Инвертированные входы  
Вы можете выбрать входы, которые нужно инвертировать.
- Вращение  
Ориентация элемента в цепи.
- Зеркало  
Зеркально отображает компонент в цепи.
- По умолчанию  
Это значение устанавливается, если цепь запущена. В демультимплексоре это значение устанавливается для невыбранных выходов.
- Использовать как значение измерения  
Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.



### 8.3. JK-триггер

Имеет возможность сохранять ( $J = K = 0$ ), устанавливать ( $J = 1, K = 0$ ), сбрасывать ( $J = 0, K = 1$ ) или переключать ( $J = K = 1$ ) сохраненное значение. Изменение состояния происходит только при нарастающем фронте тактового входа  $C$ . Возможность экспорта в VHDL / Verilog.

Входные данные

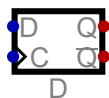
- J  
Установленный вход триггера.
- C  
Вход часов. Нарастающий фронт инициирует изменение состояния.
- K  
Вход сброса триггера.

Выводы

- Q  
Возвращает сохраненное значение.
- $\neg Q$   
Возвращает инвертированное сохраненное значение.

Атрибуты

- Метка  
Имя этого элемента.
- Инвертированные входы  
Вы можете выбрать входы, которые нужно инвертировать.
- Вращение  
Ориентация элемента в цепи.
- Зеркало  
Зеркально отображает компонент в цепи.
- По умолчанию  
Это значение устанавливается, если цепь запущена. В демультимплексоре это значение устанавливается для невыбранных выходов.
- Использовать как значение измерения  
Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.



## 8.4. D-триггер

Компонент, используемый для хранения значения. Значение на выводе D сохраняется на переднем фронте тактового вывода C. Можно выбрать разрядность, что позволяет хранить несколько бит. Возможность экспорта в VHDL / Verilog.

Входные данные

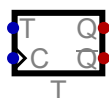
- D  
Ввод бита для сохранения.
- C  
Вывод часов для хранения значения. Значение на входе D сохраняется на переднем фронте этого вывода.

Выводы

- Q  
Возвращает сохраненное значение.
- $\neg Q$   
Возвращает инвертированное сохраненное значение.

Атрибуты

- Биты данных  
Количество используемых битов данных.
- Метка  
Имя этого элемента.
- Инвертированные входы  
Вы можете выбрать входы, которые нужно инвертировать.
- Вращение  
Ориентация элемента в цепи.
- Зеркало  
Зеркально отображает компонент в цепи.
- По умолчанию  
Это значение устанавливается, если цепь запущена. В демультимплексоре это значение устанавливается для невыбранных выходов.
- Использовать как значение измерения  
Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.



## 8.5. T-триггер

Хранит один бит. Переключает состояние по нарастающему фронту на входе C.

## Входные данные

T

Включает функцию переключения.

C

Вход часов. Нарастающий фронт переключает выход, если вход T установлен на 1.

## Выводы

Q

Возвращает сохраненное значение.

 $\neg Q$ 

Возвращает инвертированное сохраненное значение.

## Атрибуты

Метка

Имя этого элемента.

Включить ввод

Если установлено, доступен разрешающий вход (T).

Инвертированные входы

Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

Зеркало

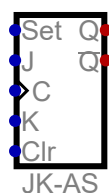
Зеркально отображает компонент в цепи.

По умолчанию

Это значение устанавливается, если цепь запущена. В демультимплекторе это значение устанавливается для невыбранных выходов.

Использовать как значение измерения

Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.



## 8.6. JK-триггер, асинхронный

Имеет возможность сохранять ( $J = K = 0$ ), устанавливать ( $J = 1, K = 0$ ), сбрасывать ( $J = 0, K = 1$ ) или переключать ( $J = K = 1$ ) сохраненное значение. Изменение состояния происходит только при нарастающем фронте тактового входа C. Есть два дополнительных входа, которые устанавливают или сбрасывают состояние немедленно без тактового сигнала. Возможность экспорта в VHDL / Verilog.



## Входные данные

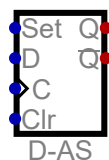
- Set  
асинхронный набор. Высокое значение на этом входе устанавливает триггер.
- J  
Установленный вход триггера.
- C  
Вход часов. Нарастающий фронт инициирует изменение состояния.
- K  
Вход сброса триггера.
- Clr  
асинхронная очистка. Высокое значение на этом входе очищает триггер.

## Выводы

- Q  
Возвращает сохраненное значение.
- $\neg Q$   
Возвращает инвертированное сохраненное значение.

## Атрибуты

- Метка  
Имя этого элемента.
- Инвертированные входы  
Вы можете выбрать входы, которые нужно инвертировать.
- Вращение  
Ориентация элемента в цепи.
- Зеркало  
Зеркально отображает компонент в цепи.
- По умолчанию  
Это значение устанавливается, если цепь запущена. В демультимплексоре это значение устанавливается для невыбранных выходов.
- Использовать как значение измерения  
Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.



## 8.7. D-триггер, асинхронный

Компонент, используемый для хранения значения. Значение на выводе D сохраняется на переднем фронте тактового вывода C. Есть два дополнительных входа, которые устанавливают или сбрасывают состояние немедленно без тактового сигнала. Можно выбрать разрядность, что позволяет хранить несколько бит. Возможность экспорта в VHDL / Verilog.

## Входные данные

Set

асинхронный набор. Если установлено в единицу, все сохраненные биты устанавливаются в единицу.

D

Ввод бита для сохранения.

C

Пин управления для хранения бита. Бит на входе D сохраняется на переднем фронте этого вывода.

Clr

асинхронная очистка. Если установлено в единицу, все сохраненные биты устанавливаются в ноль.

## Выводы

Q

Возвращает сохраненное значение.

¬Q

Возвращает инвертированное сохраненное значение.

## Атрибуты

Биты данных

Количество используемых битов данных.

Метка

Имя этого элемента.

Инвертированные входы

Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

Зеркало

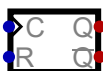
Зеркально отображает компонент в цепи.

По умолчанию

Это значение устанавливается, если цепь запущена. В демультимплексоре это значение устанавливается для невыбранных выходов.

Использовать как значение измерения

Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.



Монохромный

## 8.8. Монофlop

Монофlop устанавливается по переднему фронту на входе часов. По истечении настраиваемого времени задержки монофlop будет очищен автоматически. Монофlop перезапускается. Его можно использовать только в том случае, если в цепи присутствует ровно один тактовый компонент. Этот компонент часов используется в качестве временной базы для измерения временной задержки.

### Входные данные

C

Вход часов. Нарастающий фронт устанавливает монофlop.

R

Сбросить ввод. Высокое значение очищает монофlop.

### Выводы

Q

вывод

$\neg Q$

инвертированный вывод

### Атрибуты

Метка

Имя этого элемента.

Ширина импульса

Ширина импульса измеряется в тактах.

Инвертированные входы

Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

Зеркало

Зеркально отображает компонент в цепи.

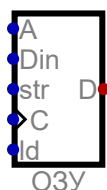
По умолчанию

Это значение устанавливается, если цепь запущена. В демультимплексоре это значение устанавливается для невыбранных выходов.

Использовать как значение измерения

Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.

## 9. Память - ОЗУ



### 9.1. ОЗУ, отдельные порты

Модуль RAM с отдельными входами для хранения и выходом для чтения сохраненных данных. Возможность экспорта в VHDL / Verilog.

## Входные данные

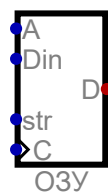
- A  
Адрес для чтения или записи.
- Din  
Данные, которые будут храниться в ОЗУ.
- str  
Если этот вход высокий и когда часы становятся высокими, данные сохраняются.
- C  
Вход часов
- ld  
Если этот вход высокий, выход активируется, и данные видны на выходе.

## Выводы

- D  
Вывод вывода данных

## Атрибуты

- Биты данных  
Количество используемых битов данных.
- Биты адреса  
Количество используемых битов адреса.
- Метка  
Имя этого элемента.
- Вращение  
Ориентация элемента в цепи.
- Формат числа  
Формат, используемый для отображения чисел.
- цифры с фиксированной запятой  
Количество дробных двоичных цифр
- Программная память  
Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.



## 9.2. ОЗУ блока, разделенные порты

Модуль RAM с отдельными входами для хранения и выходом для чтения сохраненные данные. Это ОЗУ обновляет свой вывод только по нарастающему фронту тактового сигнала. Это позволяет использовать блочную ОЗУ на ПЛИС. Возможность экспорта в VHDL / Verilog.

## Входные данные

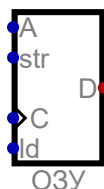
- A  
Адрес для чтения или записи.
- Din  
Данные, которые будут храниться в ОЗУ.
- str  
Если этот вход высокий и когда часы становятся высокими, данные хранятся.
- C  
Вход часов

## Выводы

- D  
Вывод вывода данных

## Атрибуты

- Биты данных  
Количество используемых битов данных.
- Биты адреса  
Количество используемых битов адреса.
- Метка  
Имя этого элемента.
- Вращение  
Ориентация элемента в цепи.
- Программная память  
Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.

**9.3. ОЗУ, двунаправленный порт**

Модуль RAM с двунаправленным выводом для чтения и записи данных.

## Входные данные

- A  
Адрес для чтения и записи.
- str  
Если этот вход высокий, когда часы становятся высокими, данные сохраняются.
- C  
Часы
- ld  
Если этот вход высокий, выход активируется, и данные видны на выходе.

**Выводы**

D

Двунаправленное соединение для передачи данных.

**Атрибуты**

Биты данных

Количество используемых битов данных.

Биты адреса

Количество используемых битов адреса.

Метка

Имя этого элемента.

Вращение

Ориентация элемента в цепи.

Формат числа

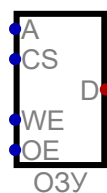
Формат, используемый для отображения чисел.

цифры с фиксированной запятой

Количество дробных двоичных цифр

Программная память

Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.

**9.4. RAM, выбор микросхемы**

Модуль RAM с двунаправленным соединением для чтения и записи данных. Если на входе CS низкий уровень, компонент отключен. Это позволяет создать большую оперативную память из нескольких меньших ОЗУ и декодера адресов. Цикл записи работает следующим образом: при установке CS на высокий уровень выбирается компонент. Передний фронт в WE фиксирует адрес, а следующий за ним задний фронт в WE сохраняет данные.

**Входные данные**

A

Адрес для чтения и записи.

CS

Если этот вход высокий, это ОЗУ включено. В противном случае выход всегда находится в состоянии высокого Z.

WE

Если установлено высокое значение, данные записываются в ОЗУ.

OE

Если этот вход высокий, сохраненное значение выводится.

**Выводы**

D

Двунаправленное соединение для передачи данных.

## Атрибуты

### Биты данных

Количество используемых битов данных.

### Биты адреса

Количество используемых битов адреса.

### Метка

Имя этого элемента.

### Инвертированные входы

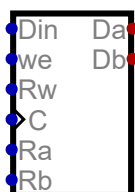
Вы можете выбрать входы, которые нужно инвертировать.

### Вращение

Ориентация элемента в цепи.

### Программная память

Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.



[Зарегистрироваться](#)

## 9.5. Файл регистрации

Память с одним портом для записи и двумя портами для чтения из памяти одновременно. Может использоваться для реализации регистров процессора. Два регистра могут быть прочитаны одновременно, а третий может быть записан. Возможность экспорта в VHDL / Verilog.

### Входные данные

#### Din

Данные, которые будут сохранены в регистре Rw.

#### we

Если этот вход высокий и когда часы становятся высокими, данные сохраняются.

#### Rw

Регистр, в который записываются данные.

#### C

Часы

#### Ra

Регистр, который виден на порту a.

#### Rb

Регистр, который виден на порту b.

### Выводы

#### Da

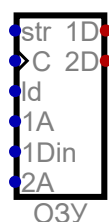
Порт вывода a

#### Db

Выходной порт b

## Атрибуты

- Биты данных  
Количество используемых битов данных.
- Биты адреса  
Количество используемых битов адреса.
- Метка  
Имя этого элемента.
- Вращение  
Ориентация элемента в цепи.



## 9.6. ОЗУ, двойной порт

RAM с одним портом, который позволяет записывать и читать из RAM, а второй порт только для чтения. Этот второй порт можно использовать для предоставления некоторой графической логики доступа к содержимому памяти. Таким образом, процессор может записывать в ОЗУ, а графическая логика может одновременно читать из ОЗУ. Возможность экспорта в VHDL / Verilog.

## Входные данные

- str  
Если этот вход высокий и когда часы становятся высокими, данные хранятся.
- C  
Часы
- ld  
Если этот вход высокий, выход активируется, и данные видны на выходе 1D.
- 1A  
Адрес, по которому выполняется чтение или запись в порт 1.
- 1Din  
Данные, которые будут храниться в ОЗУ.
- 2A  
Адрес, используемый для чтения через порт 2.

## Выводы

- 1D  
Порт вывода 1
- 2D  
Выходной порт 2

## Атрибуты

- Биты данных  
Количество используемых битов данных.



Биты адреса

Количество используемых битов адреса.

Метка

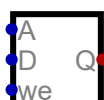
Имя этого элемента.

Вращение

Ориентация элемента в цепи.

Программная память

Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.



ОЗУ, асинхронно.

## 9.7. ОЗУ, асинхронно.

Пока мы настроены, он сохраняется. Соответствует очень простой RAM, где линии адреса и данных напрямую подключены к декодерам ячеек памяти. Возможность экспорта в VHDL / Verilog.

Входные данные

A

Адрес, по которому происходит чтение или запись.

D

Данные для сохранения.

we

Разрешение записи. Пока этот вход установлен на 1, значение, применяемое к D, равно сохраняется по адресу, применяемому к A при изменении A или D.

Выводы

Q

Вывод сохраненных данных.

Атрибуты

Биты данных

Количество используемых битов данных.

Биты адреса

Количество используемых битов адреса.

Инвертированные входы

Вы можете выбрать входы, которые нужно инвертировать.

Метка

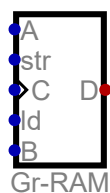
Имя этого элемента.

Вращение

Ориентация элемента в цепи.

Программная память

Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.



## 9.8. Графическая RAM

Используется для отображения растрового изображения. Этот элемент ведет себя как оперативная память. Вдобавок это показывает его содержимое на графическом экране. Каждый пиксель представлен адресом памяти. Сохраненное значение определяет цвет пикселя с использованием фиксированной цветовой палитры. Для поддержки страницы реализованы два экранных буфера. литье. Вход В выбирает отображаемый буфер. Таким образом, общий объем памяти составляет  $dx * dy * 2$  слова. Используемая палитра структурирована следующим образом: индексы 0-9 соответствуют цветам: белый, черный, красный, зеленый, синий, желтый, голубой, пурпурный, оранжевый и розовый. Индексы 32-63 отображают значения серого, а индексы 64-127 представляют 64 значения цвета, каждое с двумя битами на канал цвета. В результате получается простая палитра, к которой можно обращаться только с помощью 7-битного кода. Если архитектура поддерживает 16-битный индекс, начиная с индекса 0x8000, режим высокого цвета с 5 битами на цветовой канал может использоваться, что позволяет использовать 32768 цветов.

### Входные данные

- A  
Адрес для чтения и записи.
- str  
Если этот вход высокий, когда часы становятся высокими, данные сохраняются.
- C  
Часы
- Id  
Если этот вход высокий, выход активируется, и данные видны на выходе.
- B  
Выбирает экранный буфер для отображения.

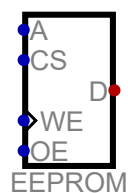
### Выводы

- D  
Двунаправленное соединение для передачи данных.

### Атрибуты

- Биты данных  
Количество используемых битов данных.
- Метка  
Имя этого элемента.
- Ширина в пикселях  
Ширина экрана в пикселях.
- Высота в пикселях  
Высота экрана в пикселях.
- Вращение  
Ориентация элемента в цепи.

## 10. Память - EEPROM



### 10.1. EEPROM

Модуль EEPROM с двунаправленным соединением для чтения и записи данных. Если на входе CS низкий уровень, компонент отключен. Содержимое данных хранится как в ПЗУ. Таким образом, он сохраняется при завершении и перезапуске моделирования. Цикл записи работает следующим образом: при установке CS на высокий уровень выбирается компонент. Передний фронт в WE фиксирует адрес, а следующий за ним задний фронт в WE сохраняет данные.

Входные данные

A

Адрес для чтения и записи.

CS

Если этот вход высокий, эта EEPROM включена. В противном случае выход всегда находится в состоянии высокого Z.

WE

Если установлено высокое значение, данные записываются в EEPROM.

OE

Если этот вход высокий, сохраненное значение выводится.

Выводы

D

Двунаправленное соединение для передачи данных.

Атрибуты

Биты данных

Количество используемых битов данных.

Биты адреса

Количество используемых битов адреса.

Метка

Имя этого элемента.

Инвертированные входы

Вы можете выбрать входы, которые нужно инвертировать.

Данные

Значения, хранящиеся в этом элементе.

Вращение

Ориентация элемента в цепи.

Формат числа

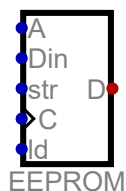
Формат, используемый для отображения чисел.

цифры с фиксированной запятой

Количество дробных двоичных цифр

### Программная память

Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.



## 10.2. EEPROM, разделенные порты

Модуль EEPROM с отдельными входами для хранения и выходом для чтения сохраненных данных.

### Входные данные

- A  
Адрес для чтения или записи.
- Din  
Данные, которые будут сохранены в EEPROM.
- str  
Если этот вход высокий и когда часы становятся высокими, данные хранятся.
- C  
Вход часов
- Id  
Если этот вход высокий, выход активируется, и данные видны на выходе.

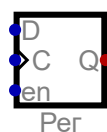
### Выводы

- D  
Вывод вывода данных

### Атрибуты

- Биты данных  
Количество используемых битов данных.
- Биты адреса  
Количество используемых битов адреса.
- Метка  
Имя этого элемента.
- Данные  
Значения, хранящиеся в этом элементе.
- Вращение  
Ориентация элемента в цепи.
- Формат числа  
Формат, используемый для отображения чисел.
- цифры с фиксированной запятой  
Количество дробных двоичных цифр
- Программная память  
Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.

## 11. Память



### 11.1. Зарегистрироваться

Компонент для хранения значений. Можно выбрать разрядность слова данных. В отличие от D-триггера, регистр обеспечивает вход, который включает часы. Возможность экспорта в VHDL / Verilog.

Входные данные

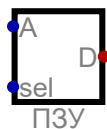
- D  
Входной контакт слова данных для сохранения.
- C  
Вход часов. Нарастающий фронт сохраняет значение на выводе D.
- en  
Включить вывод. Сохранение значения работает, только если этот вывод установлен в высокий уровень.

Выводы

- Q  
Возвращает сохраненное значение.

Атрибуты

- Биты данных  
Количество используемых битов данных.
- Метка  
Имя этого элемента.
- Инвертированные входы  
Вы можете выбрать входы, которые нужно инвертировать.
- Вращение  
Ориентация элемента в цепи.
- Программный счетчик  
Делает этот регистр программным счетчиком. Возвращается значение этого регистра во внешнюю среду разработки ассемблера, чтобы пометить текущую строку кода во время отладки.
- Использовать как значение измерения  
Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.



## 11.2. ПЗУ

Компонент энергонезависимой памяти. Сохраненные данные можно редактировать в диалоговом окне атрибутов. Возможность экспорта в VHDL / Verilog.

Входные данные

A

Этот вывод определяет адрес слова данных для вывода.

sel

Если вход высокий, выход активируется. Если он низкий, выходные данные находятся в состоянии высокого Z.

Выводы

D

Выбранное слово данных, если вход sel высокий.

Атрибуты

Биты данных

Количество используемых битов данных.

Биты адреса

Количество используемых битов адреса.

Метка

Имя этого элемента.

Данные

Значения, хранящиеся в этом элементе.

Вращение

Ориентация элемента в цепи.

Формат числа

Формат, используемый для отображения чисел.

цифры с фиксированной запятой

Количество дробных двоичных цифр

Программная память

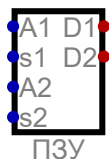
Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.

Обновить при запуске модели

Перезагружает HEX-файл каждый раз при запуске модели.

Файл

Файл для загрузки в ПЗУ.



### 11.3. Двойной порт ПЗУ

Компонент энергонезависимой памяти. Сохраненные данные можно редактировать в диалоговом окне атрибутов.

Входные данные

A1

Этот вывод определяет адрес слова данных, которое будет выводиться на D1.

s1

Если на входе высокий уровень, активируется выход D1. Если он низкий, выходные данные находятся в состоянии высокого Z.

A2

Этот вывод определяет адрес слова данных, которое будет выводиться на D2.

s2

Если на входе высокий уровень, активируется выход D2. Если он низкий, выходные данные находятся в состоянии высокого Z.

Выводы

D1

Выбранное слово данных, если на входе s1 высокий уровень.

D2

Выбранное слово данных, если на входе s2 высокий уровень.

Атрибуты

Биты данных

Количество используемых битов данных.

Биты адреса

Количество используемых битов адреса.

Метка

Имя этого элемента.

Данные

Значения, хранящиеся в этом элементе.

Вращение

Ориентация элемента в цепи.

Формат числа

Формат, используемый для отображения чисел.

цифры с фиксированной запятой

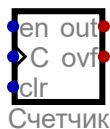
Количество дробных двоичных цифр

Программная память

Делает это ПЗУ в памяти программ. Таким образом, к нему можно получить доступ из внешней IDE.

Обновить при запуске модели

Перезагружает HEX-файл каждый раз при запуске модели.



## 11.4. Счетчик

Простой компонент счетчика. Вход часов увеличивает счетчик. Может быть сброшен обратно в 0 с помощью входа clr. Количество битов можно установить в диалоговом окне атрибута. Возможность экспорта в VHDL / Verilog.

### Входные данные

- en  
Если установлено в 1, счетчик включен!
- C  
Вход часов. Нарастающий фронт увеличивает счетчик.
- clr  
Синхронный сброс счетчика, если он установлен на 1.

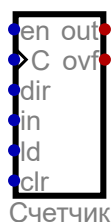
### Выводы

- out  
Возвращает подсчитанное значение.
- ovf  
Вывод переполнения. Этот вывод устанавливается в 1, если счетчик находится на максимальном значении. а вход en установлен на 1.

### Атрибуты

- Биты данных  
Количество используемых битов данных.
- Инвертированные входы  
Вы можете выбрать входы, которые нужно инвертировать.
- Метка  
Имя этого элемента.
- Вращение  
Ориентация элемента в цепи.
- Использовать как значение измерения  
Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.
- Программный счетчик  
Делает этот регистр программным счетчиком. Возвращается значение этого регистра во внешнюю среду разработки ассемблера, чтобы пометить текущую строку кода во время отладки.





### 11.5. Счетчик с предустановкой

Счетчик, значение которого можно установить. Кроме того, максимальное значение и подсчет направление можно указать. Возможность экспорта в VHDL / Verilog.

Входные данные

- en  
Если установлено в 1, счетчик включен!
- C  
Вход часов. Нарастающий фронт увеличивает или уменьшает счетчик.
- dir  
Задаёт направление счёта. 0 означает вверх.
- in  
Это слово данных сохраняется в счетчике, когда установлен ld.
- ld  
Если установлено, значение на входе 'in' сохраняется в счетчике на следующих часах. сигнал.
- clr  
Синхронный сброс счетчика, если установлено значение 1.

Выводы

- out  
Возвращает подсчитанное значение.
- ovf  
Вывод переполнения. Он установлен в 1, если вход 'en' установлен в 1 и если счетчик достигает максимального значения при обратном отсчете или достиг 0 при обратном отсчете.

Атрибуты

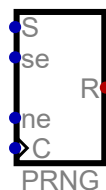
- Биты данных  
Количество используемых битов данных.
- Максимальное значение  
Если вводится ноль, используется максимально возможное значение (все биты равны единице).
- Инвертированные входы  
Вы можете выбрать входы, которые нужно инвертировать.
- Метка  
Имя этого элемента.
- Вращение  
Ориентация элемента в цепи.

Использовать как значение измерения

Если установлено, значение является значением измерения и отображается на графике и в таблице данных. Кроме того, должна быть указана метка, которая может служить идентификатором значения.

Программный счетчик

Делает этот регистр программным счетчиком. Возвращается значение этого регистра во внешнюю среду разработки ассемблера, чтобы пометить текущую строку кода во время отладки.



## 11.6. Генератор случайных чисел

Может использоваться для генерации случайных чисел. Когда моделирование запускается, генератор повторно инициализируется, так что новое псевдослучайное число последовательность генерируется при каждом запуске. Генератор может быть инициализирован в текущем моделировании с определенным начальным значением для генерации определенной последовательности псевдослучайных чисел.

Входные данные

- S  
Новое начальное значение генератора.
- se  
Если установлено, генератор случайных чисел повторно инициализируется новым начальным значением при следующем нарастающем фронте тактового сигнала.
- ne  
Если установлено, новое случайное число выводится при следующем нарастающем фронте тактового сигнала.
- C  
Вход часов.

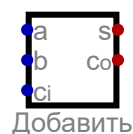
Выводы

- R  
Вывод псевдослучайного числа.

Атрибуты

- Биты данных  
Количество используемых битов данных.
- Метка  
Имя этого элемента.
- Вращение  
Ориентация элемента в цепи.

## 12. Арифметика



### 12.1. Сумматор

Компонент для простых вычислений сложения. Складывает два целых числа из ввода a и ввода b ( $a + b$ ). Результат будет увеличен на единицу, если установлен ввод переноса. Возможность экспорта в VHDL / Verilog.

Входные данные

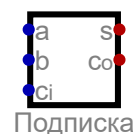
- a  
Первый ввод для добавления.
- b  
Второй ввод для добавления.
- c\_i  
Перенести ввод, если установлено, результат увеличивается на единицу.

Выводы

- s  
Результат добавления
- c\_o  
Перенести вывод. Если установлено, произошло переполнение.

Атрибуты

- Метка  
Имя этого элемента.
- Биты данных  
Количество используемых битов данных.
- Вращение  
Ориентация элемента в цепи.



### 12.2. Вычесть

Компонент для простого вычитания. Вычитает двоичные числа на входе a и входе b ( $a - b$ ). Если вход переноса установлен на 1, результат уменьшается на 1. Возможность экспорта в VHDL / Verilog.

#### Входные данные

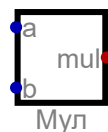
- a Введите a для вычитания.
- b Введите b для вычитания.
- c\_i Перенести ввод, если установлено, результат уменьшается на единицу.

#### Выводы

- s Вывод возвращает результат вычитания.
- c\_o Вывод возвращает 1, если произошло переполнение.

#### Атрибуты

- Метка  
Имя этого элемента.
- Биты данных  
Количество используемых битов данных.
- Вращение  
Ориентация элемента в цепи.



### 12.3. Умножить

Компонент для умножения. Умножает целые числа на входном контакте a и входном контакте b. Возможность экспорта в VHDL / Verilog.

#### Входные данные

- a Введите a для умножения.
- b Введите b для умножения.

#### Выводы

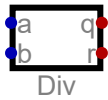
- mul Вывод результата умножения.

#### Атрибуты

- Метка  
Имя этого элемента.
- Подписанная операция  
Если выбрано, операция выполняется со значениями со знаком (2-е дополнение).
- Биты данных  
Количество используемых битов данных.

Вращение

Ориентация элемента в цепи.



## 12.4. Деление

Компонент для разделения. Делит целое число, примененное к входу a, на целое число, примененное к входу b. Если делитель равен нулю, вместо этого он делится на единицу. В знаковом делении остаток всегда положительный.

Входные данные

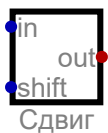
a  
    дивиденд  
b  
    делитель

Выводы

q  
    частное  
r  
    остаток

Атрибуты

Метка  
    Имя этого элемента.  
Биты данных  
    Количество используемых битов данных.  
Подписанная операция  
    Если выбрано, операция выполняется со значениями со знаком (2-е дополнение).  
Остаток всегда положительный  
    Если установлено, остаток от деления со знаком всегда положительный.  
Вращение  
    Ориентация элемента в цепи.



## 12.5. Баррель-шифтер

Компонент для сдвига битов. Сдвигает входное значение на количество бит, заданное входом сдвига.

**Входные данные**

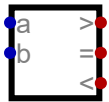
- in  
Ввод с битами, которые нужно сдвинуть.
- shift  
Ввод с шириной сдвига.

**Выводы**

- out  
Вывод со смещенным значением.

**Атрибуты**

- Метка  
Имя этого элемента.
- Биты данных  
Количество используемых битов данных.
- ввод сдвига имеет знак  
входные данные сдвига имеют два дополнительных формата
- Направление  
Установить направление.
- Режим  
Режим сдвига ствола
- Вращение  
Ориентация элемента в цепи.

**12.6. Компаратор**

Компонент для сравнения битовых значений. Сравнивает двоичные числа на входном контакте a и входном контакте b и устанавливает соответствующие выходы. Возможность экспорта в VHDL / Verilog.

**Входные данные**

- a  
Введите a для сравнения.
- b  
Введите b для сравнения.

**Выводы**

- >  
На выходе будет 1, если вход a больше, чем вход b
- =  
Вывод равен 1, если ввод a равен вводу b
- <  
На выходе будет 1, если на входе a меньше, чем на входе b

**Атрибуты**

Метка

Имя этого элемента.

Биты данных

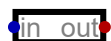
Количество используемых битов данных.

Подписанная операция

Если выбрано, операция выполняется со значениями со знаком (2-е дополнение).

Вращение

Ориентация элемента в цепи.



Отрицательный

## 12.7. Отрицание

Отрицание во 2-м дополнении Возможность экспорта в VHDL / Verilog.

Входные данные

in

Ввод слова данных, которое должно быть инвертировано во 2-м дополнении

Выводы

out

Возвращает результат отрицания во 2-м дополнении.

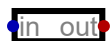
Атрибуты

Биты данных

Количество используемых битов данных.

Вращение

Ориентация элемента в цепи.



SignEx

## 12.8. Расширитель знаков

Увеличивает разрядность значения со знаком, сохраняя знак значения. Если на входе один бит, этот бит будет выводиться на всех выходных битах. Возможность экспорта в VHDL / Verilog.

Входные данные

in

Входное значение. Ширина входного бита должна быть меньше ширины выходного бита!

Выводы

out

Расширенное входное значение. Ширина входного бита должна быть меньше ширины выходного бита!

Атрибуты

Метка

Имя этого элемента.

Входная разрядность

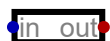
Количество выходных битов должно быть больше количества входных битов.

Разрядность вывода

Количество выходных битов должно быть больше количества входных битов.

Вращение

Ориентация элемента в цепи.



Количество бит

## 12.9. Битовый счетчик

Возвращает количество единиц во входном значении.

Входные данные

in

Вход, для которого подсчитываются 1-биты.

Выводы

out

Выводит количество 1-битов.

Атрибуты

Биты данных

Количество используемых битов данных.

Вращение

Ориентация элемента в цепи.

## 13. Переключатели



### 13.1. Переключить

Простой переключатель. Нет задержки гейта: изменение сигнала распространяется немедленно.

Выводы

A1

Одно из соединений переключателя.

B1

Одно из соединений переключателя.

Атрибуты

Биты данных

Количество используемых битов данных.

Метка

Имя этого элемента.



**Счетчик полюсов**

Количество доступных полюсов.

**Закрыто**

Устанавливает начальное состояние переключателя.

**Вращение**

Ориентация элемента в цепи.

**Зеркало**

Зеркально отображает компонент в цепи.

**Переключатель ведет себя как вход**

Если модель анализируется, переключатель ведет себя как вход, где «open» соответствует «0», а «closed» - «1».

**13.2. Двойной переключатель**

Переключатель двойного действия. Нет задержки гейта: изменение сигнала распространяется немедленно.

**Выводы****A1**

Одно из соединений переключателя.

**B1**

Одно из соединений переключателя.

**C1**

Одно из соединений переключателя.

**Атрибуты****Биты данных**

Количество используемых битов данных.

**Метка**

Имя этого элемента.

**Счетчик полюсов**

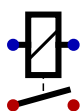
Количество доступных полюсов.

**Вращение**

Ориентация элемента в цепи.

**Зеркало**

Зеркально отображает компонент в цепи.

**13.3. Реле**

Реле - это переключатель, которым можно управлять с помощью катушки. Если через катушку протекает ток, переключатель замыкается или размыкается. Обратного диода нет, поэтому направление тока не имеет значения. Переключатель срабатывает, если входы имеют разные значения. Реле ведет себя аналогично воротам XOr.

#### Входные данные

in1

Один из входов для управления реле.

in2

Один из входов для управления реле.

#### Выводы

A1

Одно из соединений переключателя.

B1

Одно из соединений переключателя.

#### Атрибуты

Биты данных

Количество используемых битов данных.

Метка

Имя этого элемента.

Счетчик полюсов

Количество доступных полюсов.

Реле нормально закрыто.

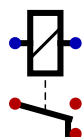
Если установлено, реле закрывается, если на входе низкий уровень.

Вращение

Ориентация элемента в цепи.

Зеркало

Зеркально отображает компонент в цепи.



### 13.4. Реле двойного выброса

Реле - это переключатель, которым можно управлять с помощью катушки. Если через катушку протекает ток, переключатель замыкается или размыкается. Обратного диода нет, поэтому направление тока не имеет значения. Переключатель срабатывает, если входы имеют разные значения. Реле ведет себя аналогично воротам XOR.

#### Входные данные

in1

Один из входов для управления реле.

in2

Один из входов для управления реле.

### Выводы

- A1  
Одно из соединений переключателя.
- B1  
Одно из соединений переключателя.
- C1  
Одно из соединений переключателя.

### Атрибуты

- Биты данных  
Количество используемых битов данных.
- Метка  
Имя этого элемента.
- Счетчик полюсов  
Количество доступных полюсов.
- Вращение  
Ориентация элемента в цепи.
- Зеркало  
Зеркально отображает компонент в цепи.



## 13.5. P-канальный полевой транзистор

P-канальный полевой транзистор. Основная часть подключается к поз. Шина напряжения и транзистор моделируются без основного диода.

### Входные данные

- G  
Ворота

### Выводы

- S  
Источник
- D  
Слив

### Атрибуты

- Биты данных  
Количество используемых битов данных.
- Однонаправленный  
Однонаправленные транзисторы распространяют сигнал только от истока к стоку. Они находятся намного быстрее моделировать, чем двунаправленные транзисторы. Поскольку нет обратной связи от стока к источнику, в этом В режиме транзистор не может закоротить подключенные провода, когда он является проводящим. Таким образом, этот режим необходим для имитировать определенные схемы CMOS.
- Метка  
Имя этого элемента.

Вращение

Ориентация элемента в цепи.

Зеркало

Зеркально отображает компонент в цепи.



### 13.6. N-канальный полевой транзистор

N-канальный полевой транзистор. Основная часть соединена с землей, и транзистор моделируется без внутреннего диода.

Входные данные

G

Ворота

Выводы

D

Слив

S

Источник

Атрибуты

Биты данных

Количество используемых битов данных.

Однонаправленный

Однонаправленные транзисторы распространяют сигнал только от истока к стоку.

Они находятся намного быстрее моделировать, чем двунаправленные транзисторы.

Поскольку нет обратной связи от стока к источнику, в этом В режиме транзистор не может закоротить подключенные провода, когда он является проводящим. Таким образом, этот режим необходим для имитировать определенные схемы CMOS.

Метка

Имя этого элемента.

Вращение

Ориентация элемента в цепи.

Зеркало

Зеркально отображает компонент в цепи.



### 13.7. Предохранитель

Предохранитель, используемый для создания одноразовой программируемой памяти.

Выводы

out1

Одно из соединений переключателя.

out2

Одно из соединений переключателя.

#### Атрибуты

##### Запрограммированный

Если установлено, диод "перегорел" или "запрограммирован". На полевом транзисторе с плавающим затвором заряжается плавающий затвор. Вы можете изменить эту настройку с помощью клавиши [P].

##### Вращение

Ориентация элемента в цепи.



### 13.8. Диод к VDD

Упрощенный однонаправленный диод, используемый для подключения провода к VDD. Он используется для реализации проводного ИЛИ. Поэтому к выходу диодов необходимо подключить понижающий резистор. В моделировании диод ведет себя как активный вентиль с трехвалентной таблицей истинности: Если вход высокий, выход также высокий. Во всех остальных случаях (вход низкий или высокий z) выход имеет высокий z государственный. Таким образом, два антипараллельно соединенных диода могут поддерживать друг друга в высоком состоянии, что невозможно с настоящими диодами. Это идеальный диод: на диоде с прямым смещением нет падения напряжения.

#### Входные данные

in

Если вход высокий, то и выход высокий. Во всех остальных случаях выход находится в состоянии высокого z.

#### Выводы

out

Если вход высокий, то и выход высокий. Во всех остальных случаях выход находится в состоянии высокого z.

#### Атрибуты

##### Запрограммированный

Если установлено, диод "перегорел" или "запрограммирован". На полевом транзисторе с плавающим затвором заряжается плавающий затвор. Вы можете изменить эту настройку с помощью клавиши [P].

##### Вращение

Ориентация элемента в цепи.



### 13.9. Диод - земля

Упрощенный однонаправленный диод, используемый для заземления провода. Он используется для реализовать проводное И. Поэтому к выходу диодов необходимо подключить подтягивающий резистор. Если вход низкий, выход также низкий. В других случаях (вход высокий или высокий z) выход имеет высокий z государственный. Таким образом, два антипараллельно соединенных диода могут удерживать друг друга в низком состоянии, что

невозможно с настоящими диодами. Итак, это идеальный диод: на диоде с прямым смещением нет падения напряжения.

#### Входные данные

in

Если на входе низкий уровень, на выходе также низкий уровень. Во всех остальных случаях выход находится в состоянии высокого z.

#### Выводы

out

Если на входе низкий уровень, на выходе также низкий уровень. Во всех остальных случаях выход находится в состоянии высокого z.

#### Атрибуты

##### Запрограммированный

Если установлено, диод "перегорел" или "запрограммирован". На полевом транзисторе с плавающим затвором заряжается плавающий затвор. Вы можете изменить эту настройку с помощью клавиши [P].

##### Вращение

Ориентация элемента в цепи.



### 13.10. P-канальный полевой транзистор с плавающим затвором

P-канальный полевой транзистор с плавающим затвором. Основная часть соединена с землей, и транзистор моделируется без внутреннего диода. Если в плавающем затворе хранится заряд, он не проводит ток, даже если затвор низкий.

#### Входные данные

G

Ворота

#### Выводы

S

Источник

D

Слив

#### Атрибуты

##### Биты данных

Количество используемых битов данных.

##### Метка

Имя этого элемента.

##### Запрограммированный

Если установлено, диод "перегорел" или "запрограммирован". На полевом транзисторе с плавающим затвором заряжается плавающий затвор. Вы можете изменить эту настройку с помощью клавиши [P].

Вращение

Ориентация элемента в цепи.

Зеркало

Зеркально отображает компонент в цепи.



### 13.11. N-канальный полевой транзистор с плавающим затвором

N-канальный полевой транзистор с плавающим затвором. Основная часть соединена с землей, и транзистор моделируется без внутреннего диода. Если в плавающих затворах хранится заряд, то он не проводит ток, даже если затвор высокий.

Входные данные

G

Ворота

Выводы

D

Слив

S

Источник

Атрибуты

Биты данных

Количество используемых битов данных.

Метка

Имя этого элемента.

Запрограммированный

Если установлено, диод "перегорел" или "запрограммирован". На полевом транзисторе с плавающим затвором заряжается плавающий затвор. Вы можете изменить эту настройку с помощью клавиши [P].

Вращение

Ориентация элемента в цепи.

Зеркало

Зеркально отображает компонент в цепи.



### 13.12. Шлюз передачи

Настоящий передаточный вентиль состоит всего из двух транзисторов. Поэтому его часто используют для экономии транзисторов при реализации на кремнии.

Входные данные

- S  
управляющий ввод.
- $\neg S$   
инвертированный управляющий вход

Выводы

- A  
выведите A
- B  
вход B

Атрибуты

- Биты данных  
Количество используемых битов данных.
- Вращение  
Ориентация элемента в цепи.

## 14. Разное

Test

### 14.1. Тестовый пример

Описывает тестовый пример. В тестовом примере вы можете описать, как должна себя вести схема. Затем можно автоматически проверить, поведение схемы фактически соответствует этому описанию. Если это не так, отображается сообщение об ошибке. В справочном тексте редактора тестовых примеров подробно описано, как можно создать такой тестовый пример. Возможность экспорта в VHDL / Verilog.

Атрибуты

- Метка  
Имя этого элемента.
- Тестовые данные  
Описание тестового примера. Подробности синтаксиса можно найти в диалоговом окне справки редактора тестовых данных.
- Включено  
Включает или отключает этот компонент.

## 15. Разное - Украшение

Текст

### 15.1. Текст

Показывает текст в цепи. Не влияет на симуляцию. Текст можно изменить в диалоговом окне атрибутов.



## Атрибуты

### Описание

Краткое описание этого элемента и его использования.

### Размер шрифта

Устанавливает размер шрифта для этого текста.

### Вращение

Ориентация элемента в цепи.

### Ориентация

Положение координаты относительно текста.

### Привязка к сетке

Если установлено, компонент выравнивается по сетке.

## Текст



## 15.2. Прямоугольник

Показывает прямоугольник в цепи. Не влияет на симуляцию. Если в качестве заголовка используется знак минус, заголовок опускается.

## Атрибуты

### Метка

Имя этого элемента.

### Ширина

Ширина в единицах сетки

### Высота

Высота в единицах сетки

### Размер шрифта

Устанавливает размер шрифта для этого текста.

### Текст внутри

Поместите текст внутри прямоугольника.

### Текст внизу

Поместите текст внизу прямоугольника.

### Текст справа

Поместите текст справа от прямоугольника.

### Привязка к сетке

Если установлено, компонент выравнивается по сетке.

## 16. Разное - Общий

init

### 16.1. Общая инициализация

Код, который выполняется для непосредственного запуска общей схемы. Если общая схема должна быть запущена напрямую, такой компонент должен присутствовать. Возможность экспорта в VHDL / Verilog.

Атрибуты

- Метка
  - Имя этого элемента.
- Включено
  - Включает или отключает этот компонент.
- Общая параметризация
  - Операторы, используемые для создания схемы.

Код

### 16.2. Код

Код, который выполняется при генерации общей схемы. Может использоваться, например, для добавления компонентов или проводов в схему. Возможность экспорта в VHDL / Verilog.

Атрибуты

- Общая параметризация
  - Операторы, используемые для создания схемы.

## 17. Разное - VHDL/Verilog

in out  
Внешний

### 17.1. Внешний

Компонент для выполнения внешнего процесса для вычисления логической функции. Используется для определения поведения компонента с помощью VHDL или Verilog. Фактическое моделирование поведения должно выполняться с помощью внешнего симулятора. В настоящее время поддерживаются только симулятор VHDL ghdl и симулятор Verilog Icarus Verilog. Метка компонента должна соответствовать имени объекта или модуля! Возможность экспорта в VHDL / Verilog.

Входные данные

in

**Выводы**

out

**Атрибуты****Метка**

Имя этого элемента.

**Ширина**

Ширина символа, если эта цепь используется в качестве компонента в другой цепи.

**Входы**

Входные данные внешнего процесса. Это список имен сигналов, разделенных запятыми. Для каждого имени сигнала количество битов, разделенных двоеточием. можно указать. Таким образом, входы 8-битного сумматора можно описать как «a: 8, b: 8, c\_in».

**Выводы**

Выходные данные внешнего процесса. Это список имен сигналов, разделенных запятыми. Для каждого имени сигнала количество битов, разделенных двоеточием. можно указать. Таким образом, выходы 8-битного сумматора можно описать как «s: 8, c\_out».

**Программный код**

Программный код, выполняемый внешним приложением.

**Приложение**

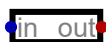
Определяет, какое приложение использовать.

**Параметры GHDL**

Параметры, которые используются GHDL для всех этапов обработки.

**Параметры IVerilog**

Параметры, которые используются IVerilog на всех этапах обработки.



Внешний файл

## 17.2. Внешний файл

Компонент для выполнения внешнего процесса для вычисления логической функции. Используется для определения поведения компонента с помощью VHDL или Verilog. Фактическое моделирование поведения должно выполняться с помощью внешнего симулятора. В настоящее время поддерживаются только симулятор VHDL ghdl и симулятор Verilog Icarus Verilog. Метка компонента должна соответствовать имени объекта или модуля! Возможность экспорта в VHDL / Verilog.

**Входные данные**

in

**Выводы**

out

**Атрибуты****Метка**

Имя этого элемента.

**Ширина**

Ширина символа, если эта цепь используется в качестве компонента в другой цепи.

**Входы**

Входные данные внешнего процесса. Это список имен сигналов, разделенных запятыми. Для каждого имени сигнала количество битов, разделенных двоеточием. можно указать. Таким образом, входы 8-битного сумматора можно описать как «a: 8, b: 8, c\_in».

**Выводы**

Выходные данные внешнего процесса. Это список имен сигналов, разделенных запятыми. Для каждого имени сигнала количество битов, разделенных двоеточием. можно указать. Таким образом, выходы 8-битного сумматора можно описать как «s: 8, c\_out».

**Программный код**

Файл, содержащий программный код, который будет выполняться внешним приложением.

**Приложение**

Определяет, какое приложение использовать.

**Параметры GHDL**

Параметры, которые используются GHDL для всех этапов обработки.

**Параметры IVerilog**

Параметры, которые используются IVerilog на всех этапах обработки.

**17.3. Пин-контроль**

Управляющая логика для двунаправленного вывода. Этот компонент необходим только в контексте генерации VHDL или Verilog, чтобы создать двунаправленный порт HDL! Если вы не хотите использовать двунаправленный порт ввода-вывода на ПЛИС, не используйте этот компонент! Компонент PinControl нельзя использовать во встроенной схеме! Это разрешено только на трассе верхнего уровня! Возможность экспорта в VHDL / Verilog.

**Входные данные**

wr

Данные для вывода.

oe

Активирует вывод.

**Выводы**

rd

Данные для чтения.

pin

Разъем для фактического контакта. Должен быть подключен только один выход здесь.

**Атрибуты**

Биты данных

Количество используемых битов данных.

Вращение

Ориентация элемента в цепи.

Зеркало

Зеркально отображает компонент в цепи.

## 18. Разное



Питание

### 18.1. Питание

Не имеет функции. Убедитесь, что VDD и GND подключены. Может использоваться в схемах 74xx для генерации контактов для источника напряжения, которые проверяются на правильность подключения.

Входные данные

VDD

Должен быть подключен к VDD!

GND

Должен быть подключен к GND!

Атрибуты

Метка

Имя этого элемента.

Вращение

Ориентация элемента в цепи.



### 18.2. Двухнаправленный разделитель

Может использоваться для шин данных и особенно упрощает создание модули памяти в пакете DIP, так как реализация шины данных упрощается.

Входные данные

OE

Если установлено, значение на клемме общих данных D выводится в бит выводит D [i], в противном случае биты D [i] выводятся на общий вывод D.

Выводы

D

Подключение к общим данным.

D0

Бит данных 0 разделителя шины.

Атрибуты

Биты данных

Количество используемых битов данных.

Вращение

Ориентация элемента в цепи.

### Распространение

Настраивает разброс входов и выходов в схеме.



### 18.3. Сброс

Выходной сигнал этого компонента удерживается на высоком уровне во время инициализации схемы. После того, как цепь стабилизируется, на выходе будет низкий уровень. Если выход инвертирован, он ведет себя противоположным образом. Возможность экспорта в VHDL / Verilog.

#### Выводы

Reset

Сбросить вывод.

#### Атрибуты

Метка

Имя этого элемента.

Инвертированный вывод

Если выбрано, вывод инвертируется.

Вращение

Ориентация элемента в цепи.



### 18.4. Разрыв

Если этот компонент используется в схеме, кнопка «Run To Break» между «Start» и "Стоп" включен. Эта кнопка синхронизирует схему до тех пор, пока на входе этого компонента не появится нарастающий фронт. обнаружен. Этот элемент можно использовать для отладки, синхронизируя схему до любой точки останова. Также может быть реализована команда ассемблера BRK. Это позволяет выполнять программу до следующей команды BRK. Эту функцию можно использовать только в том случае, если часы реального времени отключены!

#### Входные данные

brk

Останавливает быструю синхронизацию симуляции при обнаружении нарастающего фронта.

#### Атрибуты

Метка

Имя этого элемента.

Включено

Включает или отключает этот компонент.

Циклы тайм-аута

Если это количество циклов достигается без сигнала прерывания, создается ошибка.

Вращение

Ориентация элемента в цепи.



## 18.5. Остановить

Нарастающий фронт на входе останавливает симуляцию. Имеет тот же эффект, что и нажатие кнопки «Стоп» на панели инструментов.

Входные данные

stop

Нарастающий фронт останавливает моделирование.

Атрибуты

Метка

Имя этого элемента.

Инвертированные входы

Вы можете выбрать входы, которые нужно инвертировать.

Вращение

Ориентация элемента в цепи.

A small icon of a button with the word "Async" written on it, enclosed in a pink rectangular border.

## 18.6. Асинхронное время

Позволяет настроить синхронизацию асинхронной последовательной схемы, такой как Мюллер-трубопровод. Схема должна быть запущена в пошаговом режиме с одним затвором и должна быть в состоянии достичь стабильного состояния. при запуске. Последовательная схема может быть запущена интерактивно или с помощью сброса. В этом режиме нельзя использовать обычный часовой компонент.

Атрибуты

Запуск часов реального времени

Если включено, часы времени выполнения запускаются, когда цепь началось

Частота/Гц

Частота реального времени, используемая для часов реального времени

## Е Библиотека

- 27c801: 8 Mbit (1Mb x 8) UV EPROM
- 28c010: 1-Megabit (128K x 8) Paged Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c16: 16K (2K x 8) Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c64: 64K (8K x 8) Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c256: 256K (32K x 8) Paged Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 28c512: 512K-Bit (64K x 8) CMOS Parallel EEPROM; DATA Polling for End of Write Detection not implemented!
- 7400: quad 2-input NAND gate
- 7401: quad 2-input NAND gate with open-collector outputs
- 7402: quad 2-input NOR gate
- 7403: quad 2-input NAND gate with open-collector outputs, different pinout than 7401
- 7404: hex inverter
- 7405: hex inverter, open-collector output
- 7406: hex inverter buffer, open-collector output
- 7407: hex buffer, open-collector output
- 7408: quad 2-input AND gate
- 7409: quad 2-input AND gate with open-collector outputs
- 7410: triple 3-input NAND gate
- 7411: triple 3-input AND gate
- 7412: triple 3-input NAND gate with open-collector outputs
- 7413: dual 4-input NAND gate, Schmitt trigger
- 7414: hex inverter, Schmitt trigger
- 7415: triple 3-input AND gate with open-collector outputs
- 7416: hex inverter buffer, open-collector output, same as 7406
- 7417: hex buffer, open-collector output, same as 7407
- 7420: dual 4-input NAND gate
- 7421: dual 4-input AND gate
- 7425: dual 4-input NOR gate
- 7427: triple 3-input NOR gate
- 7428: quad 2-input NOR buffer
- 7430: 8-input NAND gate
- 7432: quad 2-input OR gate
- 7440: dual 4-input NAND buffer
- 7442: 4-line BCD to 10-line decimal decoder
- 7447: BCD to 7-segment decoder, active low
- 7448: BCD to 7-segment decoder, active high
- 7451: 2-input/3-input AND-NOR gate
- 7454: 2-3-2-3-line AND NOR gate
- 7455: 2 wide 4-input AND-NOR gate
- 7458: dual AND OR gate
- 7474: dual D-flip-flop
- 7476: dual J-K flip-flops with preset and clear
- 7480: Gated Full Adder with Complementary Inputs and Complementary Sum Outputs
- 7482: 2-bit binary full adder
- 7483: 4-bit binary full adder
- 7483Real: 4-bit binary full adder, real gates



7485: 4-bit comparator  
7486: quad 2-input XOR gate  
7489: 64-bit RAM  
74107: dual J-K flip-flops with clear  
74109: Dual J-NOT-K flip-flop with set and reset; positive-edge-trigger  
74112: Dual J-K negative-edge-triggered flip-flop, clear and preset  
74116: dual 4-bit D-type latches  
74133: 13-input NAND gate  
74138: 3-line to 8-line decoder/demultiplexer, inverted out  
74139: dual 2-line to 4-line decoder/demultiplexer  
74147: 10-line to 4-line priority encoder  
74148: 8-line to 3-Line priority encoder  
74150: 4-line to 16-line data selectors/multiplexers  
74151: 3-line to 8-line data selectors/multiplexers  
74153: dual 4-line to 1-line data selectors/multiplexers  
74154: 4-line to 16-line decoders/demultiplexers  
74157: quad 2-line to 1-line data selectors/multiplexers  
74160: decimal synchronous counter, async clear  
74161: hex synchronous counter, async clear  
74162: decimal synchronous counter  
74162**Real**: decimal synchronous counter, real gates  
74163: hex synchronous counter  
74164: 8-bit parallel-out serial shift register, asynchronous clear  
74165: parallel-load 8-bit shift register  
74166: 8-Bit Parallel-In/Serial-Out Shift Register  
74173: quad 3-state D flip-flop with common clock and reset  
74174: hex D-flip-flop  
74181: 4-bit arithmetic logic unit  
74182: look-ahead carry generator  
74189: 64-Bit Random Access Memory with 3-STATE Outputs  
74190: Presettable synchronous 4-bit bcd up/down counter  
74191: Presettable synchronous 4-bit binary up/down counter  
74193: Synchronous 4-Bit Up/Down Binary Counter with Dual Clock  
74198: 8-bit shift register  
74238: 3-line to 8-line decoder/demultiplexer  
74244: octal 3-state buffer/line driver/line receiver  
74245: octal bus transceivers with 3-state outputs  
74247: BCD to 7-segment decoder, active low, tails on 6 and 9  
74248: BCD to 7-segment decoder, active high, tails on 6 and 9  
74253: dual tri state 4-line to 1-line data selectors/multiplexers  
74260: dual 5-input NOR gate  
74266: quad 2-input XNOR gate  
74273: octal D-type flip-flop with clear  
74280: 9 bit Odd-Even Parity Generator-Checker  
74283: 4-bit binary full adder, alternative pinning  
74299: 8-Input Universal Shift/Storage Register with Common Parallel I/O Pins  
74373: octal transparent latches  
74374: octal positive-edge-triggered flip-flops  
74377: Octal D Flip-Flop with enable  
74382: 4-Bit Arithmetic Logic Unit

- 74540:** octal buffer/line driver, inverted
- 74541:** octal buffer/line driver
- 74573:** octal transparent latches, different pinout compared to 74373
- 74574:** octal positive-edge-triggered flip-flops, different pinout compared to 74374
- 74590:** 8-bit binary counter with tri-state output registers
- 74595:** 8-Bit Shift Registers with 3-State Output Registers
- 74670:** 3-state 4-by-4 Register File
- 74682:** 8-bit digital comparator
- 74688:** 8-bit identity comparator
- 74779:** 8-Bit Bidirectional Binary Counter with 3-STATE Outputs
- 74804:** hex 2-input NAND gate <https://www.ti.com/lit/ds/symlink/sn74as804b.pdf>
- 74805:** hex 2-input NOR gate <http://www.ti.com/lit/ds/symlink/sn54as805b.pdf>
- 74808:** hex 2-input AND gate <http://www.ti.com/lit/ds/symlink/sn54as808b.pdf>
- 74832:** hex 2-input OR gate <http://www.ti.com/lit/ds/symlink/sn54as832b.pdf>
- 744017:** Johnson decade counter with 10 decoded outputs
- 744075:** triple 3-input OR gate
- A623308A:** 8K X 8 BIT CMOS SRAM
- RAM32Bit:** A 32-bit memory that allows byte access and can handle non-aligned memory addresses.