

Владимирский государственный университет

С. А. САМОЙЛОВ

**ПРОГРАММИРОВАНИЕ
ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ
СХЕМ В РАДИОТЕХНИКЕ**

Учебное пособие

Владимир 2024

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

С. А. САМОЙЛОВ

ПРОГРАММИРОВАНИЕ
ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМ
В РАДИОТЕХНИКЕ

Учебное пособие

Электронное издание



Владимир 2024

ISBN 978-5-9984-1817-4

© ВлГУ, 2024

УДК 621.396

ББК 32.844

Рецензенты:

Доктор технических наук, профессор
профессор кафедры радиоволновых процессов и технологий
Российского технологического университета МИРЭА

В. К. Битюков

Доктор технических наук, профессор
профессор кафедры радиотехники и радиосистем
Владимирского государственного университета
имени Александра Григорьевича и Николая Григорьевича Столетовых

П. А. Полушин

Издается по решению редакционно-издательского совета ВлГУ

Самойлов, С. А. Программирование логических интегральных схем в радиотехнике [Электронный ресурс] : учеб. пособие / С. А. Самойлов ; Владим. гос. ун-т им. А. Г. и Н. Г. Столетовых. – Владимир : Изд-во ВлГУ, 2024. – 148 с. – ISBN 978-5-9984-1817-4. – Электрон. дан. (4,3 Мб). – 1 электрон. опт. диск (CD-ROM). – Систем. требования: Intel от 1,3 ГГц ; Windows XP/7/8/10 ; Adobe Reader ; дисковод CD-ROM. – Загл. с титул. экрана.

Рассмотрены современные программируемые логические интегральные схемы, их архитектура и устройства ввода/вывода. Описаны основные этапы программирования цифровых устройств на базе микросхем фирмы Xilinx. Приведено программирование помехоустойчивых кодеков. Представлены основные элементы схемотехнического редактора и их таблицы истинности.

Предназначено для студентов очной и заочной форм обучения направлений подготовки 11.03.01, 11.04.01 «Радиотехника» и 11.03.02 «Инфокоммуникационные технологии и системы связи», будет полезно специалистам в области проектирования радиотехнических устройств и систем.

Рекомендовано для формирования профессиональных компетенций в соответствии с ФГОС ВО.

Табл. 1. Ил. 44. Библиогр.: 26 назв.

ISBN 978-5-9984-1817-4

© ВлГУ, 2024

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ	5
Глава 1. ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ ИНТЕГРАЛЬНЫЕ СХЕМЫ	8
1.1. История развития и области применения ПЛИС	9
1.2. Методы проектирования на ПЛИС	14
1.3. Архитектура CPLD	18
1.4. Архитектура FPGA	24
1.5. Основные средства проектирования фирмы Xilinx	38
Контрольные вопросы.....	43
Глава 2. ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ НА БАЗЕ ПЛИС	44
2.1. Особенности проектирования цифровых устройств на базе ПЛИС	44
2.2. Пакет программ WebPack ISE	46
2.3. Основные характеристики пакета WebPACK ISE	47
2.4. Пользовательский интерфейс пакета WebPACK ISE	49
2.5. Этапы проектирования цифровых устройств на базе ПЛИС Xilinx	50
2.6. Методика работы с проектом в среде пакета WebPACK ISE	52
Контрольные вопросы.....	58
Глава 3. ПРИМЕРЫ РЕАЛИЗАЦИИ РАЗЛИЧНЫХ ЦИФРОВЫХ УСТРОЙСТВ НА ПЛИС	59
3.1. Преобразование последовательного кода в параллельный	59
3.2. Преобразование параллельного кода в последовательный код ...	60
3.3. Сверточный кодер	62
3.4. Генератор псевдослучайной последовательности	64
3.5. Кодер Хемминга	65
3.6. Декодер Хемминга	65
3.7. Особенности реализации радиотехнических задач на ПЛИС	66

3.8. Разработка лабораторного макета на ПЛИС	69
Контрольные вопросы.....	72

Глава 4. УНИФИЦИРОВАННЫЕ БИБЛИОТЕКИ КОМПОНЕНТОВ СХЕМОТЕХНИЧЕСКОГО РЕДАКТОРА ECS.....	73
4.1. Триггеры с динамическим управлением.....	73
4.2. Триггеры с потенциальным (статическим) управлением (защелки)	76
4.3. Входные и выходные триггеры с динамическим управлением ...	85
4.4. Входные триггеры с потенциальным (статическим) управлением (защелкой)	90
4.5. Сдвиговые регистры.....	94
4.6. Счетчики.....	98
4.7. Компараторы.....	104
4.8. Дешифраторы.....	109
4.9. Мультиплексоры.....	112
4.10. Устройства циклического сдвига	128
4.11. Арифметико-логические устройства.....	131
Контрольные вопросы.....	143
ЗАКЛЮЧЕНИЕ	144
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	145

ПРЕДИСЛОВИЕ

Появление в середине прошлого века программируемых электронных вычислительных систем привело к бурному развитию вычислительной техники. Изобретение микропроцессора, а также интегрированных микросхем оперативной памяти обусловило зарождение высокоуровневых языков программирования. Разработка программируемых логических интегральных схем (ПЛИС), где в одном корпусе могут присутствовать до миллиона логических вентилях, современные устройства ввода/вывода и микропроцессорные ядра, позволяет решать множество задач по обработке цифровой информации, управлению радиотехническими устройствами и организации беспроводных каналов передачи информации. Факторы доступности, низкой стоимости и относительной простоты программирования обусловили широкое применение программируемых логических интегральных схем в современной науке и технике.

Одна из наиболее важных черт ПЛИС – возможность перепрограммирования, позволяющая значительно уменьшить время разработки электронного устройства и снизить негативное влияние от возможных ошибок при разработке. Кроме того, перепрограммирование ПЛИС позволяет изменить алгоритм работы электронного устройства без изменения его аппаратной части. Таким образом, можно создавать электронные устройства с неизменными аппаратными средствами, но различными алгоритмами функционирования в зависимости от решаемой задачи.

Наряду с традиционными разделами курса по изучению программируемых логических устройств, такими как архитектура ПЛИС, устройства ввода/вывода, интерфейсы, средства программирования, логические функции, в пособие включены разделы по организации генераторов псевдослучайных последовательностей и помехоустойчи-

вых кодеков. Многообразие существующих программируемых логических интегральных схем, их архитектуры, устройств ввода/вывода, а также средств и языков программирования заставило авторов ограничиться рассмотрением основных и практически значимых устройств и разделов, отсылая читателей по вспомогательным вопросам к известной литературе.

В первой главе изложена краткая история развития программируемых логических устройств, приведены наиболее популярные на сегодняшний день ПЛИС различных фирм-производителей, осуществлен обзор методов проектирования цифровых устройств на ПЛИС, рассмотрены варианты архитектуры программируемых логических устройств, такие как CPLD и FPGA, подробно изложены основные средства проектирования фирмы Xilinx.

Во второй главе рассмотрены особенности проектирования цифровых устройств на базе ПЛИС, приведены этапы проектирования и проведен обзор средств проектирования на базе микросхем фирмы Xilinx, подробно описан пакет программ для проектирования ПЛИС WebPACK ISE, представлен пользовательский интерфейс пакета WebPACK ISE и показаны этапы проектирования и методика работы с проектом в среде WebPACK ISE.

В третьей главе приведены примеры реализации простейших устройств на микросхемах ПЛИС фирмы Xilinx, предложены принципиальные схемы и приведены временные диаграммы работы, рассмотрены варианты преобразований последовательного цифрового потока в параллельный и обратно, показана возможность реализации генератора псевдослучайной последовательности для синтеза шумоподобных сигналов, приведены примеры сверточного помехоустойчивого кодера, а также кодера и декодера Хемминга на ПЛИС.

В четвертой главе представлены основные компоненты унифицированной библиотеки схемотехнического редактора ECS, такие как триггеры со статическим и динамическим управлением, сдвиговые регистры, счетчики, компараторы, шифраторы и мультиплексоры, устройства циклического сдвига и арифметико-логические устройства,

показаны таблицы истинности рассмотренных элементов и семейства ПЛИС фирмы Xilinx, где они могут быть использованы.

Пособие обобщает материал лекционных курсов, читавшихся авторами более 20 лет во Владимирском государственном университете имени Александра Григорьевича и Николая Григорьевича Столетовых, а также результаты их научно-исследовательских работ по проектированию мощных генераторов сигналов, усилителей мощности и устройств помехоустойчивого кодирования – кодеров и декодеров.

Авторы выражают глубокую благодарность доктору технических наук, профессору кафедры «Радиотехника и радиосистемы» Владимирского государственного университета имени Александра Григорьевича и Николая Григорьевича Столетовых Полушину Петру Алексеевичу и доктору технических наук, профессору кафедры «Радиоволновые процессы и технологии» Российского технологического университета МИРЭА Битюкову Владимиру Ксенофонтовичу за обсуждение материала и ценные советы по улучшению содержания учебного пособия.

Глава 1. ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ ИНТЕГРАЛЬНЫЕ СХЕМЫ

1.1. История развития и области применения ПЛИС

Цифровая интегральная схема с высокой степенью интеграции, у которой логическая функция программируется после изготовления, называется программируемой логической интегральной схемой (ПЛИС). Возможность задания различных соединений между ее элементами или, как принято их называть, вентилями, позволяет осуществлять программирование ПЛИС.

Вентиль – это элементарное цифровое устройство, функция которого состоит в том, чтобы пропускать или не пропускать цифровой поток информации. В общем случае вентиль содержит несколько входов, а его выходной сигнал в каждый момент времени зависит от сигналов на входах.

В 70-х годах прошлого века появились программируемые логические матрицы (ПЛМ), которые стали первыми программируемыми логическими устройствами. ПЛМ – это двухблочная структура, содержащая вентили И и ИЛИ, с возможностью программирования их соединений. Программирование соединений осуществлялось путем пережигания их перемычек. Пережигание перемычек было возможно за счет подачи высоковольтных импульсов (10 – 30 В) на специальный вход устройства и выбора необходимого адреса перемычки на адресных входах ПЛМ. На рис. 1.1. приведен пример программируемой логической матрицы 4×3 .

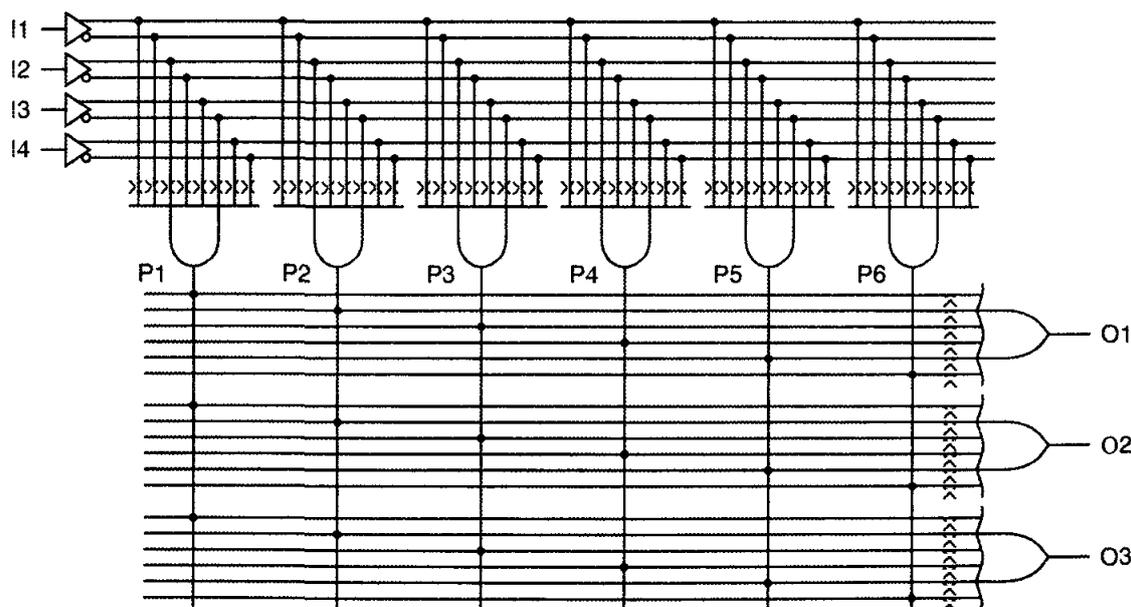


Рис. 1.1. Программируемая логическая матрица 4×3

Со временем для процесса программирования ПЛМ стали использовать электрическое или ультрафиолетовое стирание записанных ранее логических функций (ЭСПЛИС – EEPROM или СПЛИС – EPROM) с применением технологии перепрограммируемых постоянных запоминающих устройств (ППЗУ), спроектированных на транзисторах с плавающим затвором (МДП-транзисторах) или на МНОП-транзисторах (металл – нитрид – окисел – полупроводник).

Стираемые программируемые ПЗУ (erasable programmable read-only memory EPROM) похожи на обычные ППЗУ, но имеют возможность стереть свое содержимое с помощью ультрафиолетового света или электрического импульса. EPROM использует технологию МОП-транзисторов с плавающим затвором.

В месте хранения каждого бита EPROM используется МОП-транзистор с плавающим затвором (рис. 1.2). У каждого такого транзистора существуют два затвора. Плавающий затвор окружен изоляционным материалом с крайне низкой проводимостью и изначально никуда не подключен. Для программирования бита, значение которого должно быть равно логическому нулю, подается высокое напряжение, благодаря чему происходит электрический пробой изоляционного слоя и как следствие накопление отрицательного заряда на затворе. Далее высокое напряжение снимается, а накопленный отрицательный заряд остается на затворе. В дальнейшем при чтении из ППЗУ отрицательный заряд на затворе МОП-транзистора не дает ему открыться, что и позволяет считывать логический ноль на выходе такого транзистора.

Однако запрограммированное содержимое памяти EPROM можно стереть. При облучении ультрафиолетовым светом (определенной длины волны) изоляционного материала плавающего затвора в течение 10 – 20 минут изоляционный материал становится проводящим. Накоп-

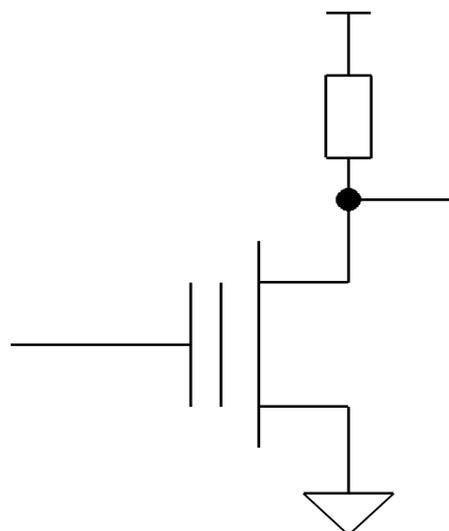


Рис. 1.2. МОП-транзистор с плавающим затвором

ленные отрицательные заряды рассасываются и память EPROM возвращается в первоначальное состояние. Микросхемы памяти EPROM содержали кварцевое окошко, через которое можно было засветить кристалл микросхемы и стереть всю записанную информацию.

Современная память типа EPROM в отличие от ультрафиолетового стирания использует другую технологию стирания – с помощью электрического сигнала (*electrically erasable programmable read-only memory, EEPROM*). В современных EPROM плавающие затворы окружены гораздо более тонким изолирующим слоем, что позволяет подавать к неплавающему затвору напряжение противоположной полярности, чем полярность заряда, который был накоплен на плавающем затворе.

EEPROM большого размера (более 1 Мбит) имеют возможность стирать информацию только блоками фиксированного размера. Поскольку процесс стирания блоками происходит практически мгновенно (*in a flash*), память подобного типа стали называть флеш-памятью (*flash memory, flash EPROM*).

Появившиеся программируемые логические устройства (ПЛУ) с решетчатой структурой (*programmable logic devices, PLDs*) стали прототипами современных ПЛИС с архитектурой CPLD. ПЛУ в отличие от ПЛМ были усовершенствованы с точки зрения технологии EPROM, было повышено количество логических ресурсов (число входов и выходов вентиля), а также добавлены элементы с памятью, триггеры, которые были объединены с выходным ИЛИ в макроячейку

На рынке первые ПЛИС появились в 80-х годах прошлого века. Основной областью их применения являлись задачи вспомогательной логики и соединительных цепей. Также ПЛИС использовались в простых задачах обработки информации и для реализации систем конечных автоматов средней сложности. С ростом увеличения размеров и структуры ПЛИС программируемые логические интегральные микросхемы начинают пользоваться все большим спросом.

С резким ростом инфокоммуникационных сетей в 90-х годах прошлого века появляются задачи обработки и передачи больших потоков информации. Это приводит к еще большей востребованности микросхем ПЛИС, которые уже используются для создания специализированных микросхем и различных испытательных систем для отработки новых алгоритмов обработки цифровой информации.

В настоящее время на рынке присутствуют программируемые логические интегральные схемы, имеющие в своем составе до нескольких десятков миллионов вентиляей. Стоит отметить такие ПЛИС, которые содержат микропроцессорные ядра и современные интегрированные устройства ввода/вывода.

На сегодняшний день ПЛИС заполняют несколько крупных сегментов рынка: заказные интегральные схемы, цифровая обработка сигналов, системы на основе встраиваемых микроконтроллеров. Кроме того, с появлением ПЛИС возник новый сектор рынка – *системы с перестраиваемой архитектурой*, или *reconfigurable computing (RC)* [1 – 8].

В последнее время цены на ПЛИС постоянно снижаются, что обуславливает применение ПЛИС не только в качестве заказных микросхем, но и в качестве их альтернативы, завоевывая новые сегменты рынка. Цены на ПЛИС неуклонно падают и потому они все чаще используются не только для создания прототипов заказных интегральных схем (ИС), но и как их альтернатива, вытесняя таким образом заказные ИС с традиционных секторов рынка.

Ранее цифровая обработка сигналов осуществлялась с помощью специализированных микропроцессоров, или цифровых сигнальных процессоров (ЦСП) (*digital signal processors (DSP)*). Но современные ПЛИС, которые могут содержать встроенные арифметико-логические устройства, а также иметь большой объем оперативной высокоскоростной памяти внутри кристалла, с учетом параллельной обработки цифровой информации обеспечивают выигрыш программируемых логических интегральных схем по сравнению с сигнальными процессорами более чем в 500 раз.

Постепенно происходит вытеснение микроконтроллеров, которые, будучи недорогими цифровыми устройствами, выполняли ряд несложных задач. Микроконтроллеры содержат встроенные таймеры, интерфейсы ввода/вывода, память команд и память данных. Все эти вспомогательные устройства располагаются в одном корпусе и в одном кристалле вместе с микропроцессорным ядром. Но снижение себестоимости ПЛИС позволило использовать самые простые ПЛИС в качестве альтернативы и микроконтроллерам и выполнять функции последних с более высокой скоростью.

Одними из самых современных цифровых систем являются системы с перестраиваемой архитектурой. Такие системы представляют

собой динамически перепрограммируемые ПЛИС, которые могут, изменив свою внутреннюю структуру, оптимально перестроиться для решения конкретной задачи в виде логических или арифметических операций. В текущий момент ведется разработка больших перестраиваемых вычислительных машин на ПЛИС. Такие цифровые системы могут использоваться для задач моделирования аппаратуры, прогнозирования, сложного многофакторного анализа, тем самым приближаясь к системам с искусственным интеллектом.

Например, компания National Instruments предлагает технологию реконфигурируемого ввода/вывода National Instruments (reconfigurable input/output – RIO). NI RIO предоставляет разработчикам приложений в LabVIEW возможность разработки своих собственных аппаратных контрольно-измерительных схем, используя программируемые логические интегральные схемы и графическую среду программирования LabVIEW. Реконфигурируемые устройства National Instruments содержат в своем составе ПЛИС, логика работы которой определяется на аппаратном уровне с помощью программного модуля LabVIEW FPGA Module [10; 11].

Интегральные схемы с архитектурой CPLD (рис. 1.3, а) представляют собой совокупность нескольких программируемых логических устройств, выполненных на одном кристалле. Структура внутренних соединений, как и сами ПЛУ, является программируемой. Увеличение размера интегральных схем возможно за счет увеличения числа как самих ПЛУ, так и развития и повышения количества программируемых соединений между ними.

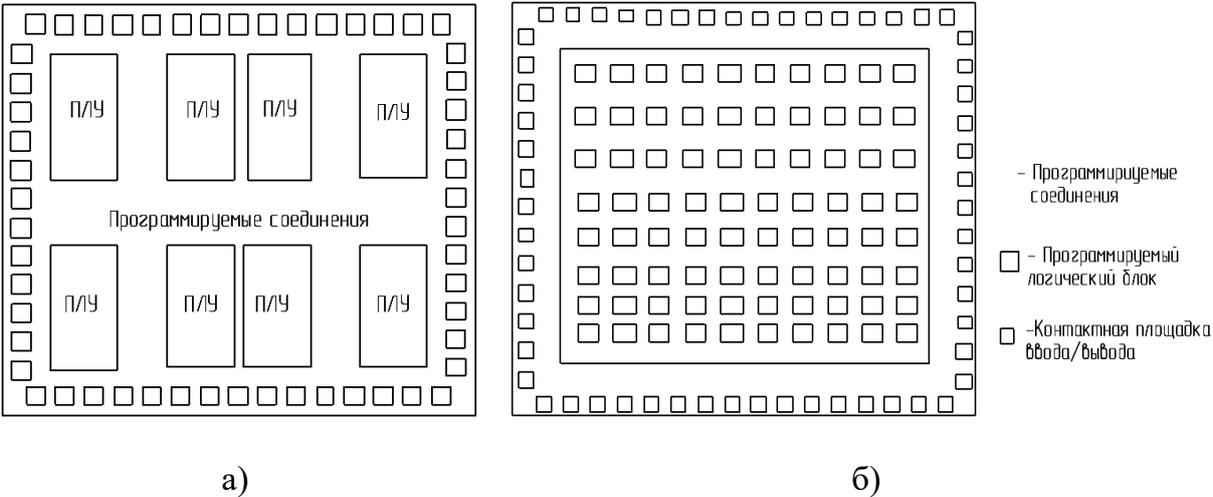


Рис. 1.3. ПЛИС архитектуры CPLD (а) и FPGA(б)

Еще одним видом архитектуры ПЛИС, наряду с CPLD, считается архитектура FPGA (field-programmable gates arrays), представляющая собой решетки вентилях, программируемые в процессе эксплуатации (рис. 1.3, б). Архитектура FPGA содержит меньшие по размерам, но большие по количеству логические вентиля и имеет более распределенную и развитую внутреннюю структуру соединений, которая занимает большую часть кристалла ПЛИС. На рис. 1.3 показано различие между этими двумя конструкциями ПЛИС.

Существует несколько крупнейших производителей ПЛИС: Xilinx, Altera, Actel, Lattice, Atmel и др. Фирма Xilinx выпускает семейства ПЛИС XC9500 и CoolRunner по архитектуре CPLD и семейства ПЛИС Virtex и Spartan по архитектуре FPGA [4].

Отдельно стоит упомянуть такое направление производства цифровых программируемых микросхем, как заказные микросхемы (ASIC – application specific integrated circuit) и полузаказные микросхемы.

Заказная микросхема разрабатывается по функциональной схеме заказчика на основе стандартных и (или) специально созданных элементов и узлов.

Полузаказная микросхема – микросхема, разработанная на основе базовых кристаллов (в том числе матричных).

Базовые матричные кристаллы (БМК) появились в середине 70-х годов XX в. как альтернатива полностью заказным ИС и ознаменовали собой новый подход к проектированию специализированных больших интегральных схем (БИС) и СБИС. Их стали называть полузаказными БИС и СБИС [8; 9].

БМК (англоязычные термины gate array и ULA, Uncommitted Logic Array) – это набор регулярно расположенных элементов (базовых ячеек), которые могут соединяться между собой для образования различных электронных схем [3; 8].

БМК в отличие от ПЛИС программируются путем нанесения маски соединений последнего слоя металлизации. Обычно БМК изготавливались небольшими сериями. Достоинство базовых матричных кристаллов заключается в следующем. При решении оригинальных задач возможна ситуация, когда существующие большие интегральные схемы не подходят. Разработка новых микросхем в этом случае зачастую бывает неэффективной и дорогой. В этом случае задача решается с помощью базовых матричных кристаллов.

По своей структуре базовый матричный кристалл в чем-то напоминает языки программирования с библиотекой готовых подпрограмм. На нем присутствуют элементарные логические цепи, не соединенные между собой. Выбирая с помощью маски необходимые соединения, можно как и при программировании добиваться выполнения необходимых функций. Маска последнего слоя металлизации наносится на поверхность кристалла и элементарные схемы, а также отдельные функции складываются в одну большую схему. В итоге получается интегральная микросхема, по себестоимости почти не отличающаяся от БМК. Основное применение больших матричных кристаллов – это области вычислительной техники, системы автоматической регулировки технологическими процессами и т. п.

В настоящее время БМК, несмотря на ряд несомненных достоинств, почти полностью заменен ПЛИС. Преимущество последних заключается в том, что при их программировании не требуется сложного технологического оборудования, а также существует возможность их многократного перепрограммирования. В России базовые матричные кристаллы производятся ОАО «Ангстрем» [2; 6; 7].

1.2. Методы проектирования на ПЛИС

Существуют три основных способа описания цифровых устройств на ПЛИС, реализуемых в том числе в САПР Xilinx Foundation ISE[2;6].

1. Классический способ проектирования – описание проектируемого цифрового устройства в виде принципиальной схемы. САПР Xilinx Foundation ISE содержит библиотеку цифровых устройств основные элементы которой приведены в гл. 4. Основные базовые элементы – простейшие логические функции, триггеры, счетчики, регистры сдвига, мультиплексоры и демультимплексоры, а также элементы с более сложным функционалом, сумматоры, компараторы, умножители и т. п. Базовые элементы имеют традиционное представление в виде условных графических обозначений.

2. Описание цифрового устройства на языках описания цифровых устройств (HDL, Hardware Description Language). Самым распространенным языком программирования на сегодняшний день считается язык VHDL. Еще одним из популярных языков программирования является язык программирования Verilog. Существуют и другие языки

программирования ПЛИС, такие как ABEL, или AHDL, предложенный фирмой Altera.

3. Синтезирование цифрового устройства на основе представленной пользователем диаграммы состояний цифрового автомата (что доступно для ПЛИС фирмы Xilinx) или введение временной диаграммы, что поддерживается средствами фирмы Altera. Результатом синтеза будет описание цифрового устройства на одном из языков программирования ПЛИС.

Схемотехническое описание проектируемого цифрового устройства имеет ряд преимуществ, которые заключаются в наглядности и отсутствии необходимости ознакомления с языками VHDL или Verilog, а также в сокращении времени разработки. В свою очередь, описание цифрового устройства средствами языка VHDL дает возможность быстрого переноса проекта на другую элементную базу, на ПЛИС других производителей или на иные программируемые устройства, такие как, например, базовые матричные кристаллы или заказные микросхемы.

Этапы проектирования на ПЛИС

В процессе проектирования цифровых устройств на базе ПЛИС фирмы Xilinx можно выделить следующие этапы:

– на первом этапе производится выбор кристалла и семейства микросхем для реализации разрабатываемого устройства. Прежде чем начинать проектирование нового устройства, следует определиться с методом его описания и средств синтеза. Разрабатываемое устройство может быть представлено в виде принципиальных схем, программ на языках HDL или диаграмм состояний. Нередко используется и смешанный метод, когда отдельные функциональные блоки разрабатываемого устройства описываются на языках HDL, а соединение их вместе представляется в виде принципиальной схемы;

– на втором этапе создается новый проект в среде САПР Xilinx ISE или Xilinx Foundation. При этом указывается информация о выбранном типе ПЛИС, ее семейства, средствах синтеза и моделирования разрабатываемого устройства;

– третий этап проектирования используется для подготовки файлов исходного описания проектируемого устройства в схемотехнической, алгоритмической или текстовой форме. Кроме того, на этом этапе

устанавливаются топологические или временные ограничения, которые будут учитываться при синтезе, размещении и трассировке проекта в кристалле выбранной микросхемы;

– на четвертом этапе осуществляется синтез разрабатываемого устройства. В процессе синтеза формируется список соединений, содержащий набор примитивов, или компонентов, который может быть реализован на основе элементов выбранного кристалла ПЛИС. Далее результаты синтеза используются в качестве исходных данных средствами размещения и трассировки;

– на пятом этапе используется метод функционального моделирования для верификации исходных описаний разрабатываемого цифрового устройства. На этом этапе реальные временные задержки не учитываются, а цель верификации заключается в проверке соответствия выходных сигналов предполагаемому алгоритму работы;

– шестой этап посвящен трассировке и размещению разработанного проекта в кристалле. На этапе размещения и трассировки проекта в кристалле производится распределение выполняемых функций в конфигурируемые логические блоки CLB (Configurable Logic Block) или макроячейки (Macrocell) в зависимости от используемого семейства ПЛИС и формирование необходимых связей;

– на седьмом этапе в процессе трассировки и размещения происходит расчет значений реальных временных задержек и производится полное, в том числе и по времени, моделирование разработанного проекта. Этап моделирования выполняется с учетом всех временных задержек распространения сигналов внутри микросхемы ПЛИС;

– восьмой этап посвящен формированию последовательности конфигурации ПЛИС. На этом этапе происходит формирование файла, в котором размещена информация о конфигурации ПЛИС, реализующей разрабатываемое устройство;

– последний этап заключается в прошивке спроектированного устройства в кристалл ПЛИС или в конфигурационное ПЗУ/ППЗУ. Итогом процесса загрузки конфигурационного файла является программирование выбранной микросхемы ПЛИС с помощью соответствующих средств САПР серии Xilinx ISE и загрузочного кабеля.

Типовой маршрут проектирования цифровых устройств, выполняемых на основе ПЛИС фирмы Xilinx, показан на рис. 1.4.

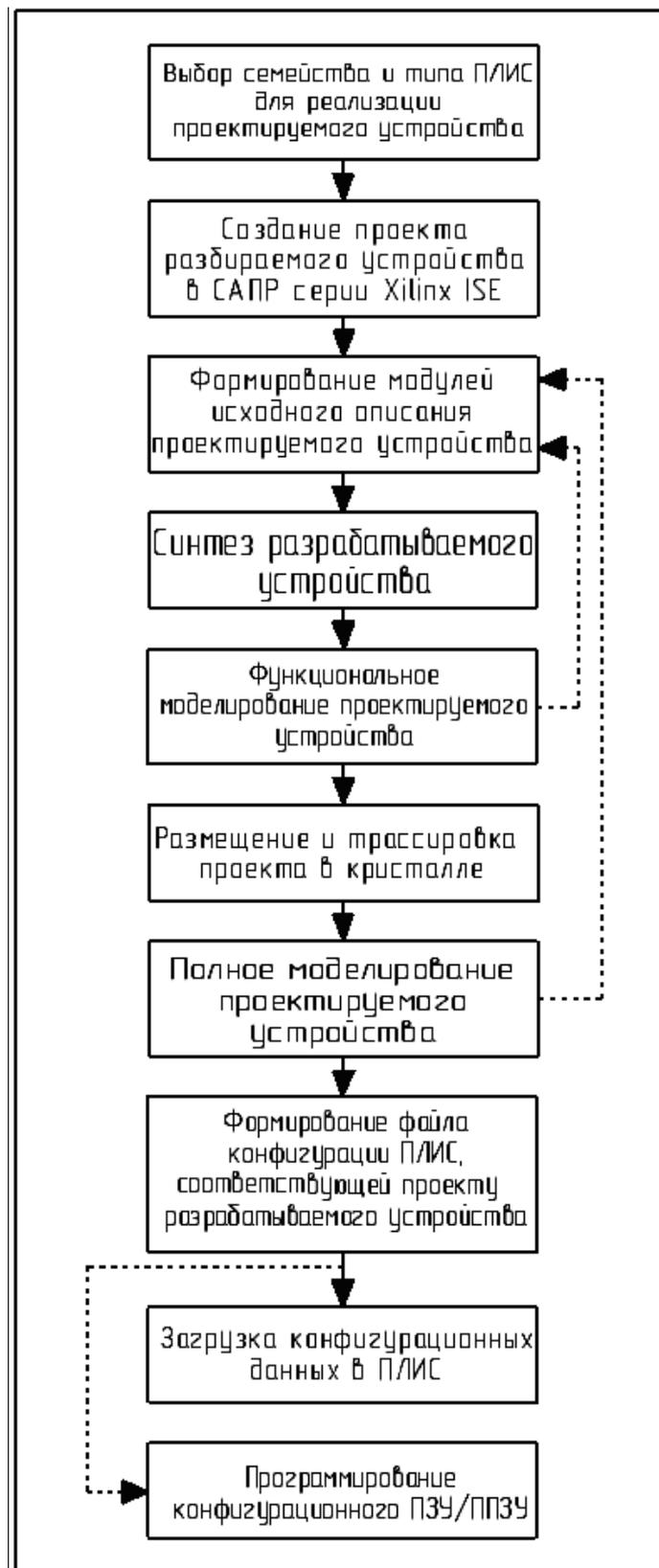


Рис. 1.4. Этапы проектирования цифровых устройств на ПЛИС Xilinx

Следует обратить внимание на то, что этапы функционального и временного моделирования не являются обязательными. Тем не менее использование эффективных средств моделирования, включаемых в состав пакетов САПР серии Xilinx ISE, позволяет обнаружить большинство возможных ошибок и тем самым значительно сократить общее время разработки устройства. При обнаружении ошибок на любом из этих этапов моделирования, например, логических ошибок на этапе функционального моделирования или при получении неудовлетворительных результатов временного моделирования, следует вернуться на стадию разработки исходных описаний проекта, внести необходимые изменения и повторить соответствующие этапы [5].

Выполнение этапов создания нового проекта и подготовки исходных описаний проектируемого устройства не зависит от выбранного типа архитектуры ПЛИС (CPLD или FPGA), используемого для его реализации. Содержание последующих этапов (синтеза, функционального и временного моделирования, размещения, трассировки и загрузки проекта в кристалл) зависит от семейства ПЛИС – CPLD или FPGA [5;8].

1.3. Архитектура CPLD

ПЛИС архитектуры CPLD произошли от ПЛУ (PLD), построенных по принципу «сумма произведений», имеющих элементы с памятью и возможность перепрограммирования. Развитие программируемых логических устройств со временем привело к пониманию того, что увеличение вычислительных ресурсов неэффективно из-за снижения быстродействия и уменьшения помехозащищенности, а также из-за неэффективного использования площади кристалла, потому что большую часть кристалла занимают линии трассировки.

Произошел переход к составным ПЛУ, или CPLD, состоящим из нескольких ПЛУ, помещенных в один корпус и соединяющихся с помощью переключающей матрицы.

CPLD состоит:

- из функциональных блоков (ФБ, ПЛУ, PLD);
- блоков ввода-вывода (БВВ, IOB);
- переключающей матрицы (ПМ, FastCONNECT matrix).

Рассмотрим архитектуру CPLD на примере семейства XC9500. Микросхемы XC9500 фирмы Xilinx представляют собой семейство ИС типа CPLD одинаковой архитектуры, но с различным числом внешних I/O-выводов и с разным числом *функциональных блоков*. Маркировка микросхем определяется числом имеющихся в них макроячеек [1; 3 – 6].

Макроячейка – элемент ПЛУ или функционального блока ПЛИС – содержит в себе элементы комбинационной логики и триггер.

Самый маленький представитель семейства содержит два функциональных блока с 36 макроячейками, а самый большой – 16 функциональных блоков с 288 макроячейками.

На рис. 1.5 показана архитектура CPLD так, как она приведена в фирменной документации Xilinx (см. сайт Xilinx.com, DS063.pdf) на семейство XC9500.

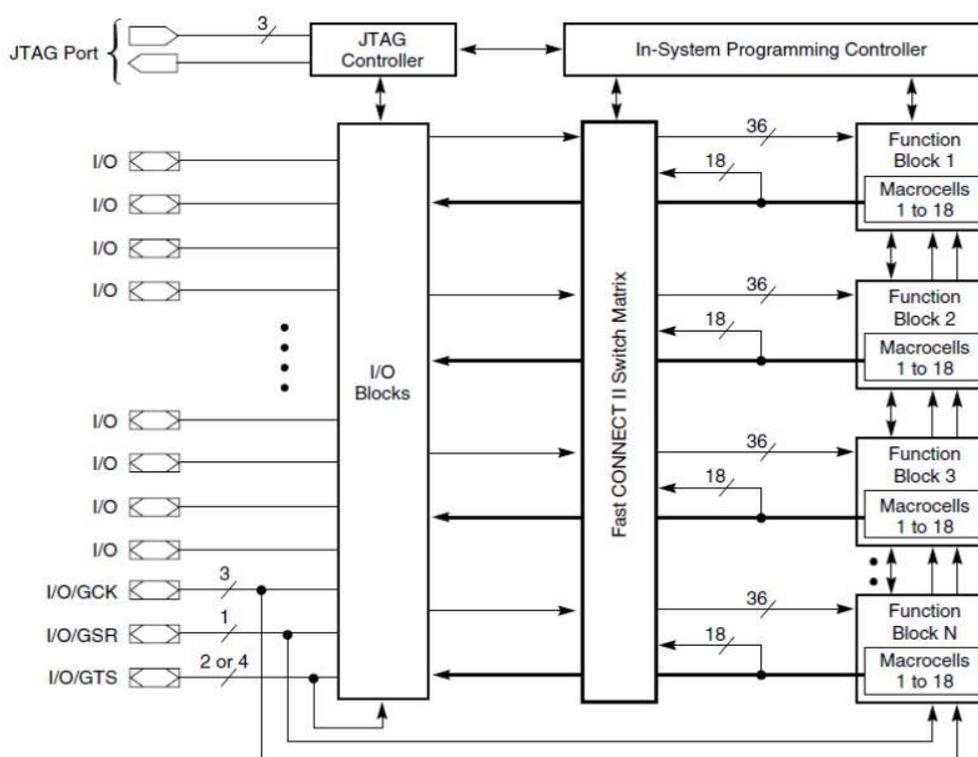


Рис. 1.5. Архитектура ПЛИС CPLD XC9500

Каждый внешний I/O-вывод можно использовать в качестве входа, выхода или двунаправленного вывода в соответствии с тем, как запрограммировано устройство. Выводы, расположенные в нижней части

рисунка, можно использовать также для тех или иных специальных целей. На любой из трех выводов **GCK** можно подавать «общие тактовые сигналы»; каждую макроячейку можно запрограммировать так, чтобы на нее поступал тактовый сигнал с выбранного входа. Вывод **GSR** можно использовать для подачи сигнала «общая установка/сброс»; каждую макроячейку можно запрограммировать так, чтобы с помощью этого сигнала производилась асинхронная предварительная установка или сброс. Наконец, на любой из двух или из четырех выводов **GTS** (в зависимости от типа устройства) можно подавать сигнал, осуществляющий «общее управление третьим состоянием»; в каждой макроячейке можно выбрать один из этих сигналов для отпириания или запириания соответствующего выхода, когда выход макроячейки подключен к внешнему I/O-выводу[4;5].

Кроме того, CPLD снабжены JTAG-контроллером, позволяющим производить периферийное сканирование и программирование ПЛИС. На входы каждого функционального блока путем программирования переключающей матрицы подаются 36 сигналов. На входы переключающей матрицы поступают сигналы с 18 выходов макроячеек от каждого функционального блока и внешние входные сигналы с I/O-выводов.

Кроме того, у каждого функционального блока есть 18 выходов, сигналы на которых проходят «мимо» переключающей матрицы и поступают на блоки ввода/вывода (линия без стрелки). Это сигналы разрешения выхода для выходных каскадов блока ввода/вывода; эти сигналы действуют в том случае, когда выход макроячейки данного функционального блока подключен к внешнему I/O-выводу.

Рассмотрим подробнее составные элементы ПЛИС CPLD.

Функциональный блок состоит:

- из макроячеек;
- массива элементов «И» (90 штук);
- распределителя термов.

В каждом функциональном блоке присутствует 18 независимых макроячеек. Любая макроячейка может быть использована для регистровой или какой-либо комбинационной функции.

Функциональный блок (ФБ) позволяет реализовать устройства с 36 входами и 18 выходами. К ФБ также подключены глобальные сигналы – тактовый, разрешения выхода и сброса/установки. Функциональный блок формирует (имеет) 18 выходов, которые подключены к

переключающей матрице (FastCONNECT switch matrix). Эти 18 сигналов, а также соответствующие им 18 сигналов разрешения выхода (output enable) подключаются также и к блокам ввода-вывода непосредственно.

Логические ячейки внутри ФБ могут быть представлены как «сумма произведений». 36 входов позволяют сформировать 36 элементарных пар сигналов на входах массива из 90 вентилях «И». Любое количество вентилях из 90 доступных может быть подключено к каждой из макроячеек функционального блока. Это достигается с помощью «распределителя термов».

Каждый функциональный блок (за исключением ПЛИС XC9536) поддерживает «локальную обратную связь», что позволяет выходы ФБ подавать на входы вентилях «И» этого же ФБ, не выходя за пределы ФБ. Эти дорожки используются для построения быстродействующих счетчиков и цифровых автоматов, в которых все регистры находятся внутри одного функционального блока. Архитектура функционального блока представлена на рис. 1.6.

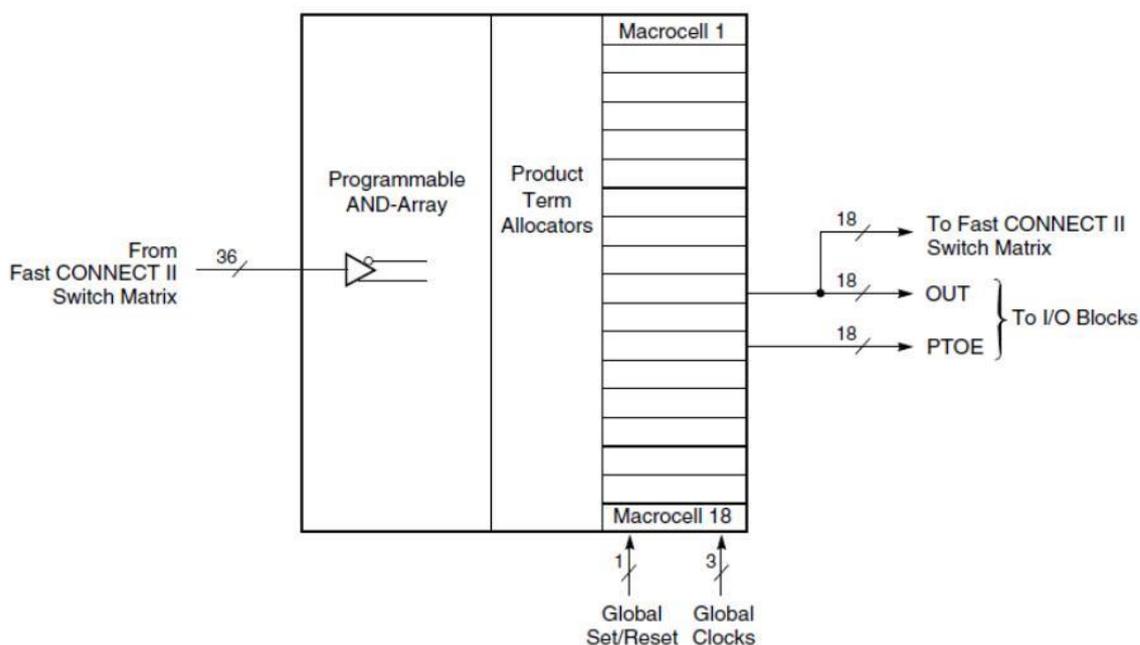


Рис. 1.6. Архитектура функционального блока

Макроячейка

В каждой макроячейке присутствует пять вентилях с логической функцией «И» и каждая макроячейка может выполнять регистровую

или комбинационную функцию. Внутри макроячейки присутствует логический элемент «ИЛИ» с шестью входами, который в совокупности с множеством элементов «И» реализует функцию «сумма произведений». С помощью совокупности таких элементов могут быть реализованы комбинационная логика или коммутация сигналов, подведенных к макроячейке.

Регистр макроячейки может быть использован как D- или T-триггер; регистры поддерживают асинхронный сброс и установку. Пользователь может задать состояние регистра при включении питания (следует помнить, что по умолчанию это логический ноль).

Все управляющие сигналы доступны для каждой из макроячеек. В частности, тактовый сигнал регистров макроячеек может быть одним из глобальных тактовых сигналов или может быть подан через вентили «И».

Распределитель термов (элементов «И»)

Используется для распределения логических вентилях «И» между макроячейками. Варианты использования:

- все пять вентилях подключены к элементу «ИЛИ» макроячейки;
- логическая емкость одной из макроячеек наращивается за счет всех или нескольких вентилях «И» других макроячеек (при этом вносятся дополнительная задержка распространения сигнала).

Блоки ввода-вывода

Структура БВВ в ИС семейства XC9500 показана на рис. 1.7. Имеются семь вариантов выбора сигнала разрешения выхода для выходного буфера с тремя состояниями. Буфер может быть всегда открытым, всегда запертым, его состояние может определяться термом-произведением РТОЕ, поступающим от соответствующей макроячейки, или любым из четырех сигналов общего разрешения выхода. Сигналы общего разрешения выхода могут иметь как высокий активный уровень, так и низкий в зависимости от сигналов на внешних выводах GTS.

ИС семейства XC9500 совместимы с внешними устройствами с напряжением питания 5 и 3,3 В. Входной буфер и внутренняя логика работают от источника питания с напряжением V_{CCINT} , равным 5 В. В зависимости от напряжения питания внешних устройств в выходном каскаде используется напряжение питания V_{CCO} , равное 5 или 3,3 В. Включение резистора между выходом и шиной питания подтягивает

напряжение на выходе до напряжения питания блока I/O, т. е. до напряжения V_{CCO} . Номинал подтягивающего резистора обычно 10 кОм. Он служит для предотвращения дрейфа при включении питания и программировании. В рабочем режиме подтягивающий резистор отключается.

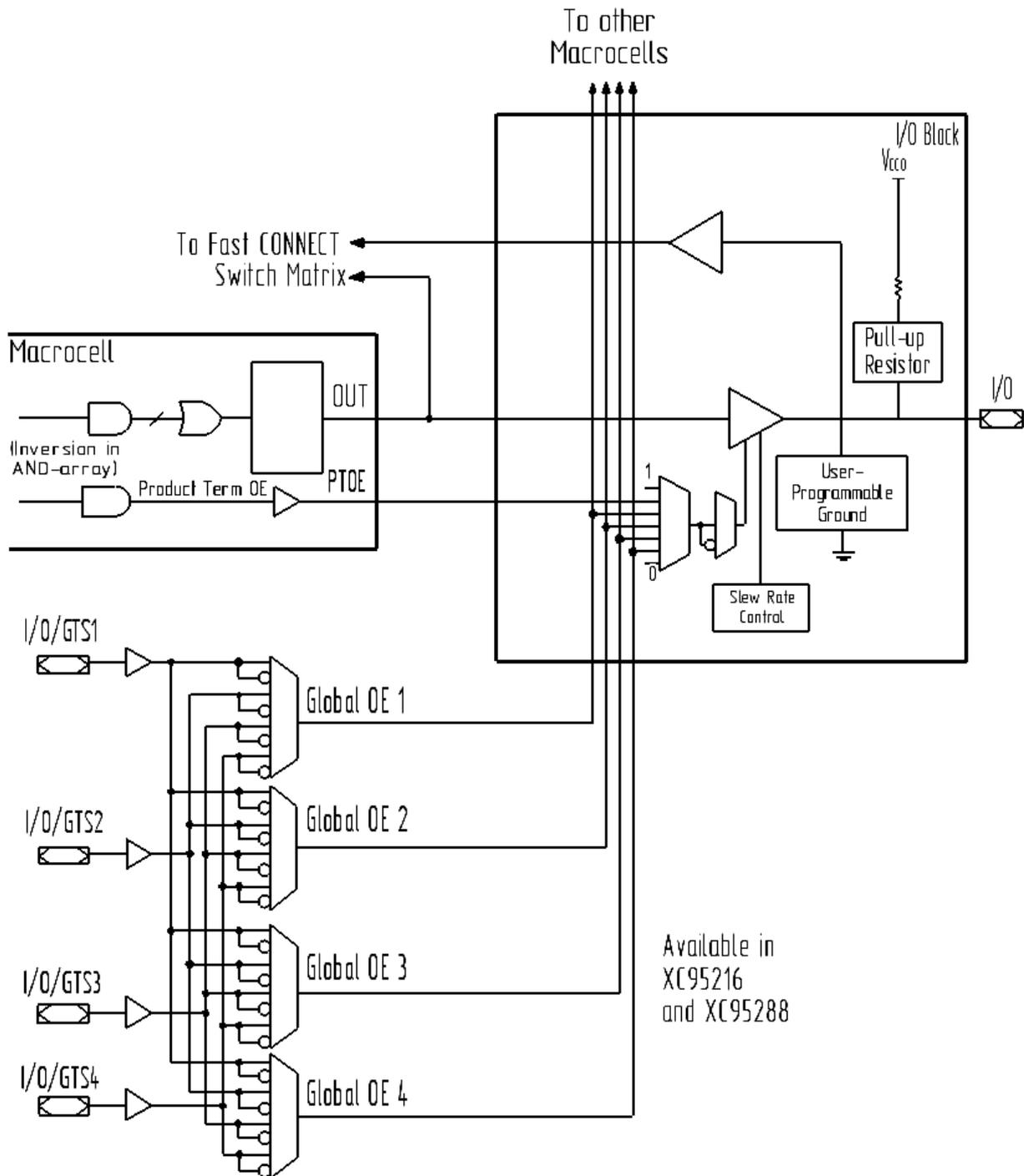


Рис. 1.7. Архитектура блока ввода-вывода

Переключающая матрица (Fast CONNECT Switch matrix)

Переключающая матрица (ПМ) используется для соединений входных сигналов со входами функциональных блоков. Все выходы функциональных блоков и блоки ввода/вывода подключены к переключающей матрице. Для всех упомянутых сигналов, как сигналов из ФБ, так и сигналов ввода/вывода, присутствует возможность установки задержки распространения, которая будет одинаковой для всех функциональных блоков. Переключающая матрица фактически представляет собой логическую схему «монтажное И».

Монтажное И – способ объединения выходов элементов с открытым коллектором (стоком), при котором все выходы присоединяются к одному подтягивающему резистору [9].

Это дает дополнительную логическую емкость и нагрузочную способность (коэффициент объединения) ФБ по входу без дополнительной временной задержки. Эта возможность доступна только для внутренних соединений выходов ФБ. Она включается автоматически САПРОМ.

1.4. Архитектура FPGA

FPGA расшифровывается как field programmable gate array, что в переводе означает «массив программируемых логических вентиляей». FPGA в какой-то мере подобна CPLD, вывернутой изнутри наружу. Как показано на рис. 1.8, на кристалле расположено большое число программируемых логических блоков, каждый из которых меньше, чем ПЛУ. Они распределены по всему кристаллу среди программируемых соединений, а вся матрица окружена программируемыми блоками ввода/вывода. Программируемый логический блок ИС типа FPGA обладает меньшими возможностями, чем типичное ПЛУ, но одна микросхема типа FPGA содержит гораздо больше логических блоков, чем ИС типа CPLD при том же самом размере кристалла [12].

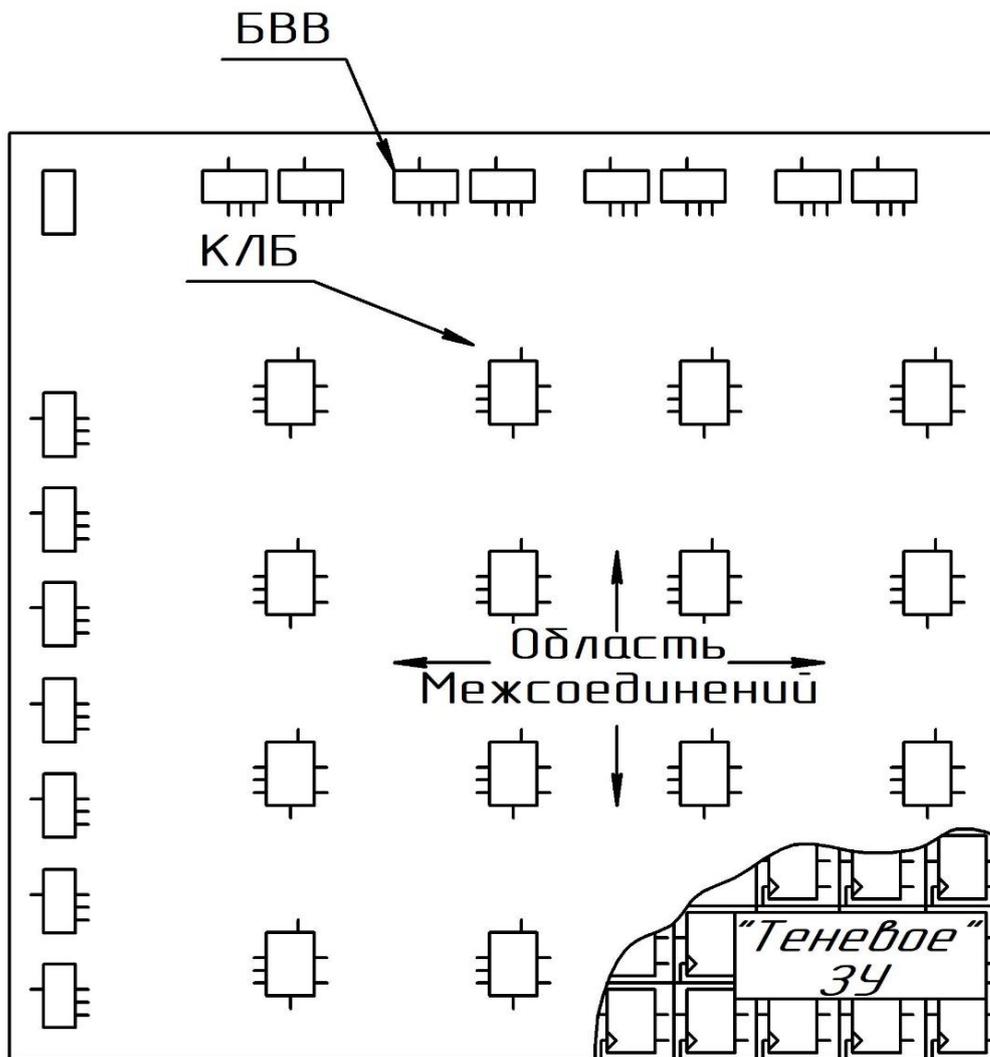


Рис. 1.8. Архитектура кристалла ПЛИС FPGA XC2000

Для иллюстрации архитектуры ИС типа FPGA мы воспользуемся простейшим семейством FPGA фирмы Xilinx семейством XC2000. Микросхема ПЛИС FPGA – двухслойная (см. рис. 1.8). Верхний слой содержит пользовательскую часть, т. е. логические элементы, из которых собирается схема, и линии межсоединений, которые соединяют логические элементы. Нижний слой содержит служебную часть, управляющую соединениями логических элементов. Служебная часть представляет собой конфигурируемое запоминающее устройство – СРАМ, ячейки которого имеют выход в верхний слой для управления всеми блоками ПЛИС.

Конфигурируемый логический блок (КЛБ)

Простейший КЛБ ПЛИС семейства XC2000 представлен на рис. 1.9.

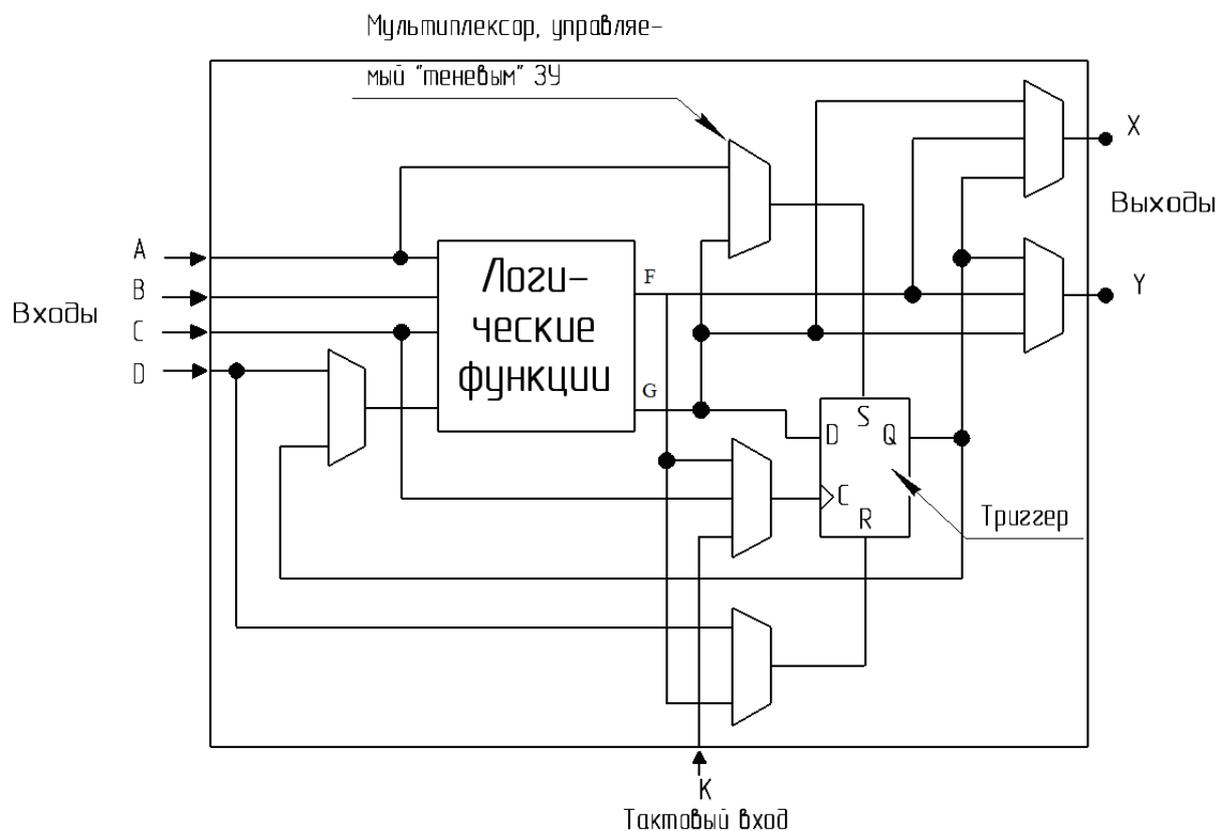


Рис. 1.9. КЛБ ПЛИС семейства XC2000

В КЛБ можно выделить три части:

1. Набор мультиплексоров, предназначенный для внутренних соединений и управления. Мультиплексоры служат для расширения возможностей блока АЛУ, управления подачей управляющих сигналов на триггер и формирования выходов КЛБ. На набор мультиплексоров поступают входы A, C, D, тактовый вход K, выходы табличного преобразователя G и F, выход триггера Q. Блок мультиплексоров запитывает тактовый и управляющие входы триггера, выходы КЛБ X, Y.

2. Запоминающий элемент (триггер) для хранения значения одной из логических функций. В соответствии с требованиями пользователя могут быть заданы фронт срабатывания триггера, уровень на выходе триггера при включении питания, источники управляющих сигналов (все это с помощью мультиплексоров). На вход D триггера подается выход F табличного преобразователя.

3. Табличный преобразователь – универсальный комбинационный узел, служащий для выработки любых логических функций четырех переменных. При решении подобной задачи следует учитывать, что если ее решать на уровне вентилях, то такая схема оказывается очень сложной и запутанной. Но можно применить иной подход и решить задачу функции четырех переменных как набор из 16 различных вариантов, сведенных в таблицу. При таком подходе решение подобной задачи значительно упрощается.

Предположим, что мы храним таблицу истинности в 1-разрядной памяти на 16 слов. Подавая на адресные входы памяти четыре входных бита, мы получаем на выходе значение функции для этой комбинации значений переменных. Фактически решение такой задачи сводится к блоку памяти с четырьмя адресными входами и хранением 16 бит информации.

Именно такой подход был принят разработчиками ИС типа FPGA фирмы Xilinx. LUT, вырабатывающие значения логических функций, фактически являются всего лишь очень компактными и быстрыми статическими оперативными запоминающими устройствами (ОЗУ). Их называют «табличными преобразователями», LUT, от английского Look-Up Table [8;9].

Кроме явного удобства при программировании логических функций, применение памяти имеет очень важное значение, поскольку подобную память можно сконфигурировать таким образом, чтобы в процессе работы ПЛИС использовать ее не как логику, а как оперативную память. Наличие внутренней оперативной памяти в ПЛИС резко расширяет возможности при решении ряда практических задач, таких как генерация шумоподобных сигналов, помехоустойчивое кодирование, шифрование информации и псевдослучайная перестройка радиочастоты (ППРЧ). Возможность подобной конфигурации присутствует в микросхеме XC4000, которая при запуске позволяет использовать табличные преобразователи в качестве оперативной памяти.

В более современных ПЛИС КЛБ содержат несколько 4- и 3-входовых LUT. При этом открываются новые возможности – построение ОЗУ 32×1 , двухпортовых ОЗУ.

Блоки ввода/вывода (БВВ)

Рассмотрим ПЛИС FPGA на примере простейших представителей – XC2000. БВВ служат для выбора пользователем конфигурации сопряжения внешних выводов корпуса ПЛИС с ее внутренней частью. Большинство выводов ПЛИС связано с конфигурируемыми блоками ввода-вывода, за исключением выводов питания «земли» и служебных выводов. Располагаются БВВ по всем четырем сторонам кристалла. На рис. 1.10 показана общая структура БВВ ПЛИС серии XC2000.

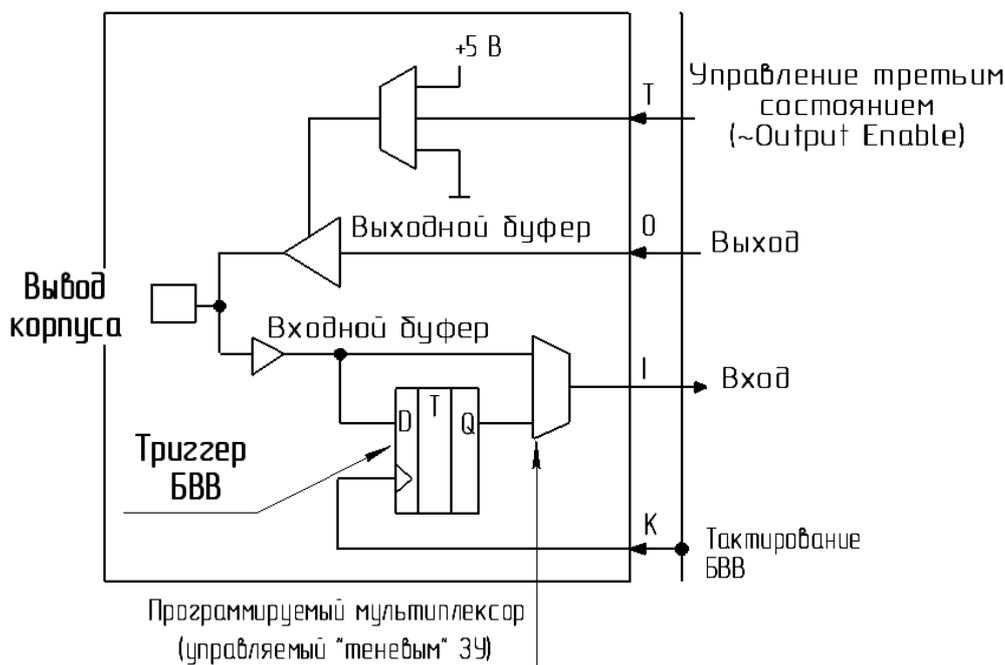


Рис. 1.10. Блок ввода/вывода ПЛИС FPGA XC2000

Каждый БВВ включает программируемый входной канал (через вывод I) и программируемый выходной буфер. Входной буфер обеспечивает согласование внешних сигналов, поступающих на выводы корпуса ПЛИС, и внутренних логических сигналов.

Порог входного буфера можно запрограммировать на совместимость с уровнями либо ТТЛ-схем (пороговое напряжение 1,4 В), либо схем на КМОП-структурах (пороговое напряжение 2,2 В). Буферизованный входной сигнал поступает на вход данных триггера и на один из входов программируемого мультиплексора. Выходной сигнал триг-

гера поступает на другой вход мультиплексора. Наличие программируемого мультиплексора дает возможность пользователю выбрать либо прямой ввод сигнала, либо ввод с запоминанием на триггере. Расположенные на одной стороне кристалла БВВ используют общие синхронизирующие импульсы, поступающие на вход К. Триггеры сбрасываются в 0 в процессе установления конфигурации, а также входом RESET, имеющим активный нижний уровень (при переходе этого сигнала в логический 0).

Особенности архитектуры XC4000/

КЛБ: два 4-входовых и один 3-входовой табличный преобразователь, два триггера, 17 мультиплексоров. КЛБ имеет 13 входов и 4 выхода.

БВВ: выходной сигнал тоже можно провести через триггер.

Назначение контактов ПЛИС. Программирование контактов

Все контакты ПЛИС можно разделить на несколько категорий: пользовательские (в документации обозначаются I/O), специального назначения (тактирование, загрузка, выбор режима), питания (обозначаются VCC) и общего проводника (GND).

Пользовательские контакты и иногда контакты специального назначения можно использовать для назначения входов и выходов проектируемого цифрового устройства. Для того чтобы закрепить за контактом ПЛИС какой-либо вход или выход своей схемы, проектировщик пользуется файлом ограничений (расширение *.ucf). Формат назначения контакта имеет вид:

NET <имя цепи в проекте> LOC = P_i, где P_i – номер контакта ПЛИС.

Тактовую частоту, которая будет использоваться для синхронизации всего устройства, подают на специально отведенные для этого контакты – GCLK, которых в ПЛИС может быть от одного до четырех.

Если загрузка ПЛИС осуществляется в последовательном режиме, то контакты для параллельной загрузки можно использовать в проекте наряду с контактами общего назначения.

Программируемые межсоединения в FPGA позволяют объединять входы и выходы любых БВВ и КЛБ. Все межсоединения представляют собой сетку вертикальных и горизонтальных металлических сегмен-

тов, в месте пересечения которых расположены транзисторы, выполняющие роль программируемых связных точек (англ. PIP's – Physical Interconnect Points – точки физического межсоединения), что дает возможность реализовать требуемый маршрут.

Программируемые межсоединения в ранних семействах ПЛИС FPGA (XC2000, XC3000) подразделяются на межсоединения общего назначения, прямые соединения и длинные линии.

Межсоединения общего назначения (англ. – General Purpose Interconnect) (рис. 1.11) используются для передачи сигналов между любыми блоками. Они содержат переключающие матрицы (ПМ – англ. Swith Matrix) для изменения направления и разветвления.

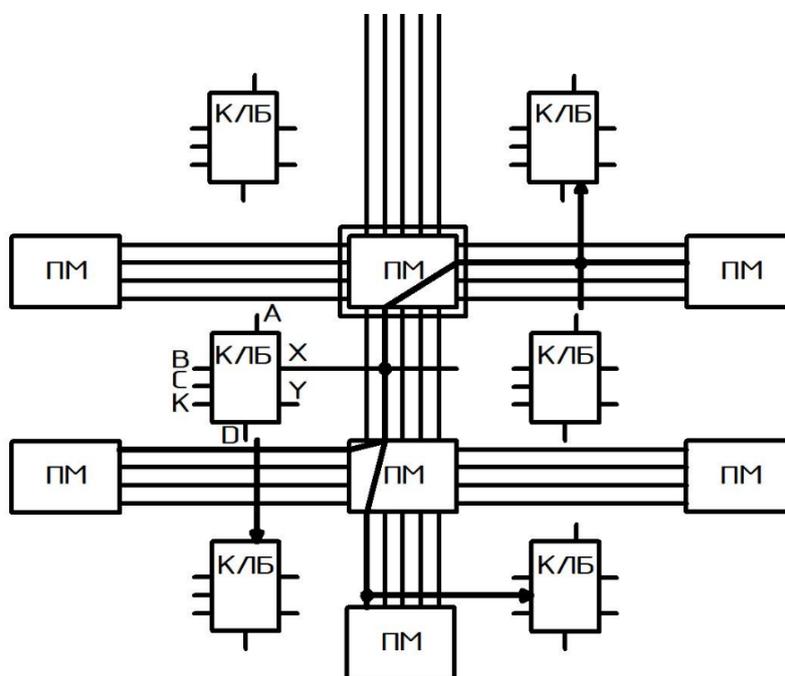


Рис. 1.11. Межсоединения общего назначения с переключающими матрицами в ПЛИС XC2000

Следует помнить, что программируемые матрицы вносят достаточно большую задержку в распространение сигнала, которая, кроме того, зависит от длины проводника. Поэтому какой-либо управляющий сигнал или сигнал с данными может приходиться к разным блокам ПЛИС с разной задержкой. Это создает предпосылки для различного рода сбоев при работе цифрового устройства на ПЛИС. Такой паразитный эффект получил в литературе название «гонки сигналов».

Прямые соединения (Direct Interconnect) обеспечивают практически нулевую задержку сигналов между соседними логическими блоками, а также между БВВ и близлежащими к ним КЛБ.

Длинные линии (ДЛ – от англ. Long Line), проходящие мимо ПМ (рис. 1.12), также позволяют подводить ко всем логическим блокам сигналы с практически нулевой задержкой.

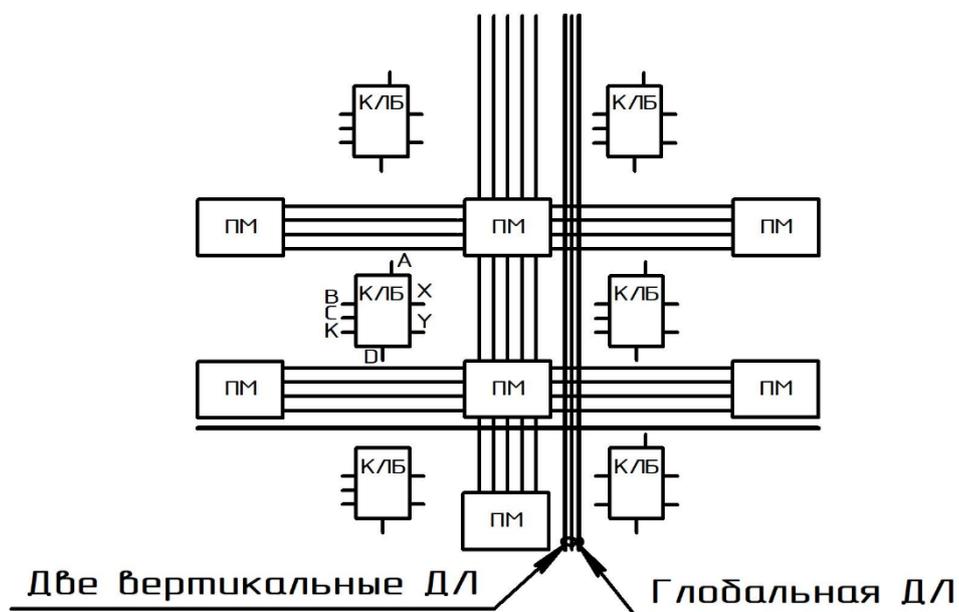


Рис. 1.12. Длинные линии в ПЛИС XC2000

В ПЛИС серии XC2000 имеются три вида длинных линий:

- горизонтальные (по одной на каждую строку КЛБ);
- вертикальные (по две на каждый столбец КЛБ);
- глобальная (одна, проходящая возле каждого столбца КЛБ).

В ПЛИС серии XC3000 имеются пять видов длинных линий:

- горизонтальные (по две на каждую строку КЛБ);
- вертикальные (по три на каждый столбец КЛБ);
- глобальная (одна, проходящая возле каждого столбца КЛБ);
- ДЛ тактирования БВВ;
- дополнительные ДЛ половинной длины (англ. – Half-length).

Длинные линии по отдельности могут подключаться к выходам КЛБ, позволяя создавать быстродействующие локальные тактовые цепи, проходящие вдоль одного столбца или строки КЛБ. К горизонтальным длинным линиям сигналы могут подаваться через трехстабильные буферы (буферы с тремя состояниями).

Программируемые межсоединения в микросхеме (МС) серии ХС4000 имеют более сложную структуру, чем в серии ХС3000. Они подразделяются на межсоединения общего назначения (линии одинарной длины, линии двойной длины) и ДЛ (прямых соединений нет).

Линии одинарной длины соединяют соседние переключающие матрицы (ПМ) в вертикальном и горизонтальном направлениях и таким образом позволяют подводить сигналы к разным сторонам логического блока (рис. 1.13).

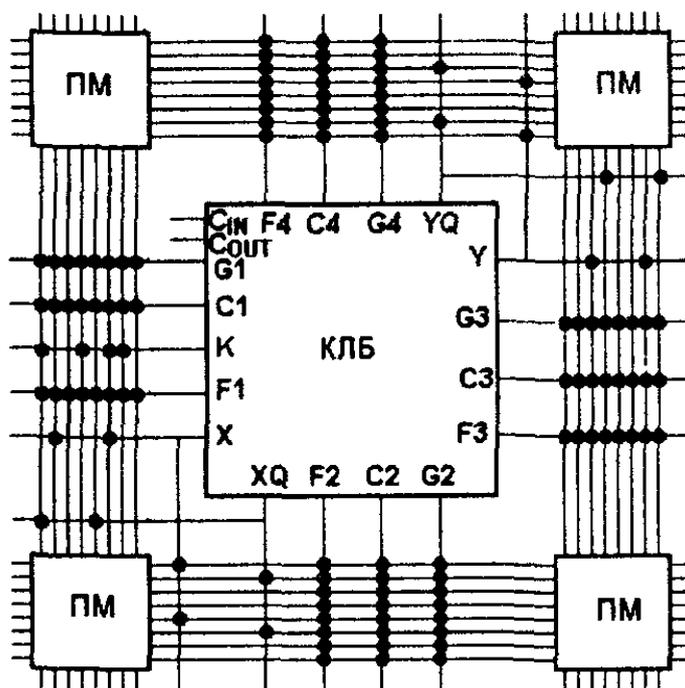


Рис. 1.13. Межсоединения линиями одинарной длины в ПЛИС ХС4000

Переключающие матрицы (рис.1.14) имеют регулярную структуру и позволяют передавать сигналы влево/вправо или вверх/вниз между смежными одинарными линиями, а также изменять направление и разветвлять сигнал.

Для уменьшения времени распространения сигнала в микросхеме серии ХС4000 имеются еще и так называемые **линии двойной длины**, которые соединяют не соседние программируемые микросхемы (ПМ), а через одну (рис. 1.15).

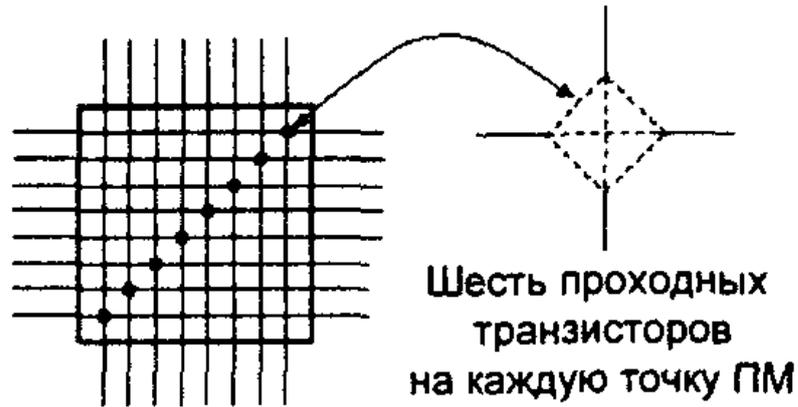


Рис. 1.14. Переключающая матрица XC4000

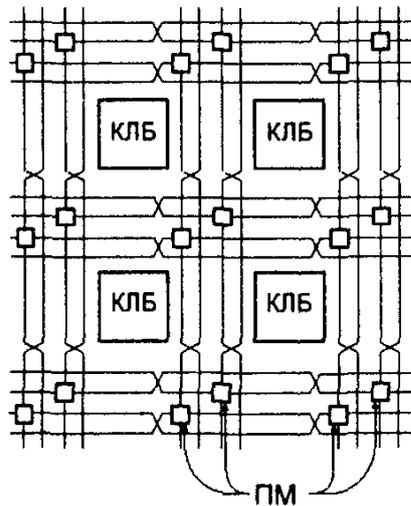


Рис. 1.15. Линии двойной длины в ПЛИС XC4000

Длинные линии, проходящие мимо ПМ, позволяют подводить ко всем КЛБ сигналы практически с нулевой задержкой.

В ПЛИС серии XC4000 различают четыре вида длинных линий:

- горизонтальные и вертикальные ДЛ (в зависимости от семейства несколько линий на каждую строку и столбец КЛБ), обеспечивающие подвод сигнала к строго определенным выводам блока;

- длинные линии тактирования БВВ (по две линии на каждой стороне кристалла, проходящие вдоль БВВ);

- глобальные ДЛ, имеющие выход на определенные БВВ;

- длинные линии распределенного дешифратора.

Каждая длинная линия в середине имеет переключающий транзистор, управляемый «теневым» ЗУ конфигурации, который может разделить эту линию на две независимые линии половинной длины.

Для распространения тактового сигнала в ПЛИС XC2000, XC3000, XC4000 используются глобальные длинные линии.

В более современных FPGA тактовые сигналы, как и прежде, распространяются по специально отведенным глобальным линиям. Трассировочные ресурсы общего назначения подразделяются на большее количество видов. Например, ПЛИС серии Spartan 3 и Virtex II¹ имеют четыре вида трассировочных ресурсов:

- длинные линии (long line), идущие вдоль ряда логических ячеек, и подключаемые к каждой шестой ячейке в ряду;
- hex line, подключаемые к каждой третьей ячейке;
- double line, подключаемые «через одну» ячейку;
- прямые соединения (direct line), подключающие ячейку к каждой из 8 соседних.

Виды трассировочных ресурсов, перечисленных в этом списке, представлены на рис. 1.16.

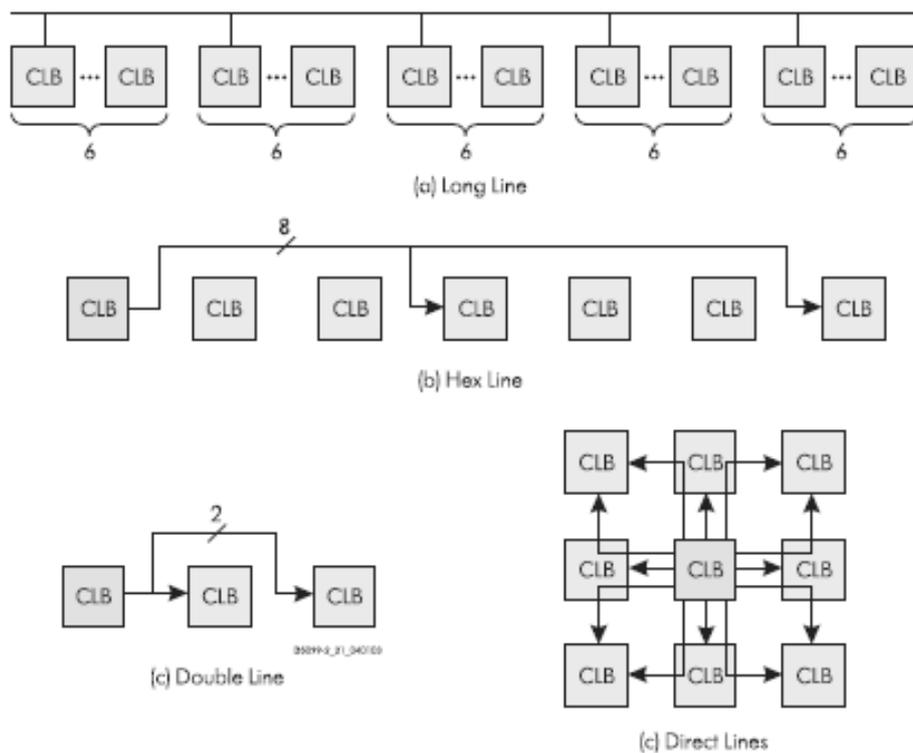


Рис. 1.16. Трассировочные ресурсы FPGA Spartan 3 и Virtex II

При высоких коэффициентах заполнения кристалла происходит резкое увеличение задержек распространения (трассировка насыщен-

ного проекта затруднена для систем автоматизированного проектирования (САПР). Особенно проблематичны ячейки, которые реализуют сквозное соединение. Если трассировочных линий недостаточно и нельзя обойти САПР, иногда можно использовать внутренние ресурсы ячейки таким образом, чтобы они передавали копию входного сигнала на выход. Однако это может привести к увеличению задержек сигнала, проходящего через комбинационную логику ячейки, и, следовательно, таких ситуаций следует избегать. Недостаток трассировочных ресурсов может привести к негативным последствиям, которых также следует избегать. Однако использование ячеек в качестве трассировочного ресурса не является неэффективным решением, а скорее свидетельствует о гибкости САПР. В противном случае некоторые проекты не могли бы быть транслированы при относительно низком коэффициенте использования кристалла.

Блоки управления тактовым сигналом (DLL, DCM)

Синхронный стиль проектирования в современных условиях наиболее предпочтителен, поэтому эффективность распределения по кристаллу тактового сигнала является в некотором роде определяющей для успешной реализации проекта, работающего на большой частоте. Эту задачу несколько облегчает тот факт, что тактовый сигнал подается на вполне определенный (тактовый) вход триггеров логических ячеек и тактовые входы выделенных устройств с синхронным интерфейсом (например, блочной памяти). Поэтому *тактовые линии не перегружены коммутационными устройствами и могут быть более или менее равномерно распределены по кристаллу, обеспечивая минимальные относительные задержки распространения*. Эффективность работы этих линий дополнительно обеспечивается блоками автоподстройки задержки (Delay Locked Loop, DLL), а в более поздних ПЛИС – блоками управления тактовым сигналом (Digital Clock Management, DCM). DLL и DCM расположены рядом с тактовыми входами ПЛИС и доступны только при использовании ими [2;4].

Основное назначение:

- фазовая подстройка тактовой частоты;
- разделение ПЛИС на несколько областей (по количеству специализированных тактовых входов), управляемых собственными тактовыми частотами.

Блочное ОЗУ в ПЛИС FPGA

Блочное ОЗУ (Block SelectRAM) в ПЛИС появляется, начиная с Virtex и Spartan-II. Оно создано в дополнение к распределенной памяти небольшой емкости, реализованной на таблицах преобразования (LUTRAM). Блоки памяти Block Select RAM организованы в виде столбцов. Все кристаллы Spartan-II содержат два таких столбца – по одному вдоль каждой вертикальной кромки (рис. 1.17). Эти колонки увеличивают полный размер кристалла. Каждый блок памяти равен по высоте четырем КЛБ, таким образом, микросхема Spartan-II, имеющая 8 КЛБ по высоте, содержит два блока памяти на колонку и четыре блока памяти в целом.

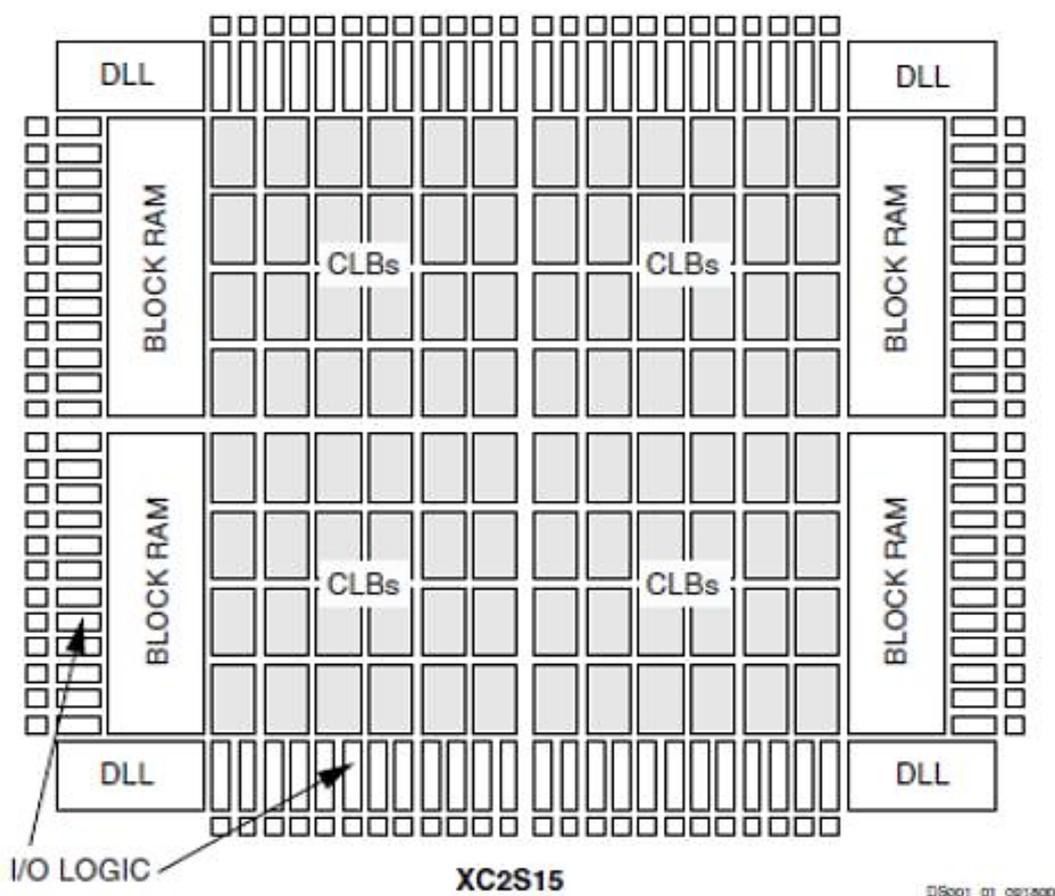


Рис. 1.17. Расположение блочного ОЗУ в ПЛИС Spartan II (на примере XC2S15)

Следует отметить, что при переходе к семействам Virtex-II и Spartan-3 блоки памяти объемом 4 кбит были заменены на 18-килобитные блоки, ассоциированные с выделенными блоками умножения

(т. е. выход каждого блока памяти подключается к одному из входов блока умножения, что облегчает реализацию таких задач, как перемножение входного потока данных с таблицей, хранящейся в блочной памяти).

В таблице приводятся емкости блочной памяти для различных кристаллов Spartan-II.

Емкости блочной памяти для различных кристаллов Spartan-II

Кристалл Spartan-II	Число блоков	Общий объём блочной памяти, бит
XC2S15	4	16384
XC2S30	6	24576
XC2S50	8	32768
XC2S100	10	40960
XC2S150	12	49152
XC2S200	14	57344

Каждый блок памяти – это синхронная двухпортовая RAM с независимым управлением для каждого порта (рис. 1.18). Размерность шины данных для обоих портов может быть сконфигурирована независимо, что позволяет создавать преобразователи размерности шины. В кристаллах Spartan-II созданы специальные трассировочные ресурсы для связи блочной памяти с КЛБ и другими блоками блочной памяти.

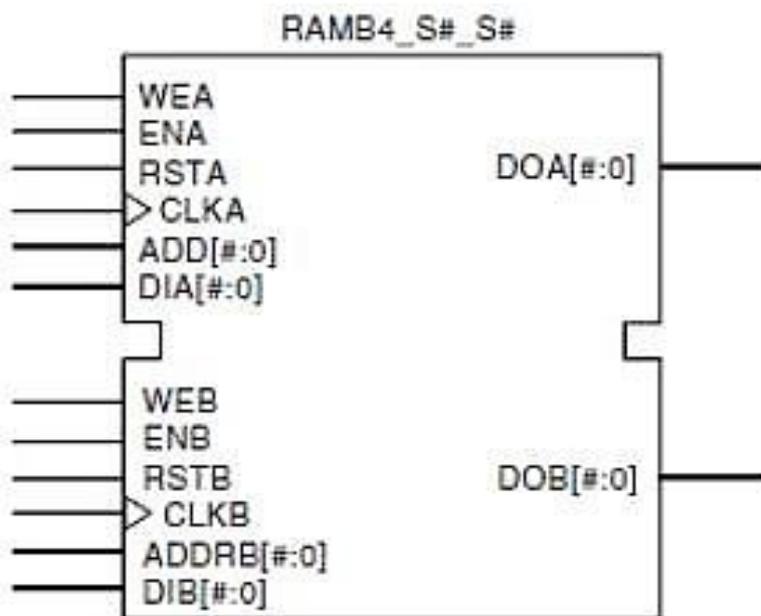


Рис. 1.18. Двухпортовое блочное ОЗУ в ПЛИС Spartan II

Назначение входов и выходов

Входы enable (EN) управляют чтением, записью и сбросом. Если $EN = 0$, данные в ОЗУ не записываются, а на выходах сохраняется прежнее состояние. Если EN и RST равны 1, выходы ОЗУ очищаются по фронту тактового импульса CLK.

Если на входе $WE = 1$, происходит запись в ОЗУ по указанному адресу. Если $EN = 1$, $WE = 0$, по фронту тактового импульса данные считываются из ячейки по указанному адресу. Если $EN = 1$, $WE = 1$, происходит запись по указанному адресу по фронту тактового импульса

if write enable (WE) is High, the memory contents reflect the data at DI.

Области применения блочного ОЗУ:

– при реализации ОЗУ в рамках разрабатываемого проекта (хранение данных);

– при реализации процессора на ПЛИС для хранения программ и данных. Особенности процессоров на ПЛИС: алгоритмы зачастую сводятся к однообразным операциям с большими входными потоками;

– при реализации аппаратных умножителей. Начиная с Virtex-II, где появились полнофункциональные устройства умножения независимых операндов, блочная память организована таким образом, чтобы ее выход мог быть подан непосредственно на один из входов блока умножения (что требуется для целого ряда DSP-задач). В семействе Virtex-4 сделан еще один шаг – добавлен 48-разрядный аккумулятор, работающий на той же частоте, что и блочная память и блоки умножения.

ПЛИС семейства Virtex II имеют дополнительные трассировочные ресурсы типа «fast connect» внутри КЛБ, соединяющие входы и выходы LUT.

1.5. Основные средства проектирования фирмы Xilinx

Для фирмы Xilinx программные средства являются ключевой частью решений программируемой логики. С момента появления на рынке Xilinx продал более 150 тыс. лицензий своей САПР ISE по всему

миру. Фирма Xilinx также предлагает для разработчиков многочислен-ные программные средства, доступные через Интернет. Эти програм-мные средства позволяют специалистам получать мгновенный прямой доступ к технической поддержке на сайте Xilinx. Предлагаемые Xilinx средства проектирования являются наиболее быстрыми по отрасли и обеспечивают уровень производительности, до 70 % превышающий про-изводительность конкурирующих средств проектирования для FPGA.

Программное обеспечение ISE фирмы Xilinx давно известно как средство, обеспечивающее высокую производительность разрабаты-ваемых устройств с программируемой логикой, при этом ISE является простым в освоении и недорогим продуктом. В версии программного обеспечения ISE 8.1i Xilinx представляет новую технологию ISE Fmax, которая, как следует из ее названия, разработана для повышения про-изводительности создаваемых устройств на базе ПЛИС и устранения сложностей в проектах. Использование разработчиками более разви-тых средств, таких как программное обеспечение (ПО) для разработки и анализа проектов PlanAhead, системы аппаратной отладки в реаль-ном времени ChipScope Pro, САПР ISE, позволяет значительно сокра-тить сроки реализации проектов.

Не будет преувеличением сказать, что для разработчиков, исполь-зующих ПЛИС FPGA или CPLD, наступили хорошие времена. Для проектов с большим логическим объемом и высокой производи-тельностью фирма Xilinx предлагает семейство ПЛИС Virtex-4 – это ап-паратная платформа для реализации наиболее сложных проектов. Для крупносерийных проектов предлагаются семейство Spartan-3 и ПЛИС CPLD, которые предназначены для разработки изделий, рас-считанных на большие тиражи при низкой стоимости. Одновременно Xilinx предоставляет разработчикам полноценный набор средств проектирования, позволяющих раскрыть всю мощь аппаратных решений.

Разработчикам, желающим освоить ISE, достаточно обратиться к пакету ISE WebPACK. Например, ISE WebPACK 8.1i является бесплат-ным, свободно загружаемым через Интернет средством проектирова-ния, которое обеспечивает сквозной маршрут проектирования для ПЛИС FPGA и CPLD. Эта САПР включает синтез и моделирование на HDL, размещение и трассировку проектов, а также программирование

кристаллов по JTAG-интерфейсу. ISE WebPACK обеспечивает простую в освоении и полнофункциональную среду проектирования, как и отмеченная призами САПР Xilinx ISE Foundation, только бесплатно. Таким образом, компанией Xilinx разработано удобное для работы средство проектирования, его легко актуализировать путем загрузки обновлений или полной инсталляции из одного файла.

ISE Foundation – полнофункциональная версия программного обеспечения ISE, сочетающая поддержку всех ведущих семейств ПЛИС FPGA и CPLD фирмы Xilinx. Это наиболее полное средство проектирования для устройств с программируемой логикой, обеспечивающее оптимальную производительность, управление потребляемой мощностью, снижение стоимости изделия, а также имеющее лучшую по отрасли службу технической поддержки Xilinx. ISE Foundation также поддерживает методологию частичного реконфигурирования, что позволяет уменьшить размер проектируемого устройства, его массу, потребляемую мощность и стоимость. ISE Foundation и ISE WebPACK в настоящее время поддерживают технологию Fmax, которая обеспечивает повышение производительности до 70 % для ПЛИС Virtex-4 по сравнению с аналогичными решениями других фирм.

Особенности Virtex-4 заключаются в следующем:

- алгоритмы синтеза нового поколения предусматривают технологию, определяемую термином *retiming*; отображение проекта на физические ресурсы с учетом заданных временных параметров; глобальную оптимизацию и оптимизацию по итогам размещения;

- повышение производительности на 37 % при использовании новой версии системы автоматизированного проектирования в режиме «по умолчанию» (при незадаваемых физических ограничениях) по сравнению с предыдущими версиями;

- утилита Xplorer – бесплатное, простое в использовании программное средство, обеспечивающее 10%-ный прирост производительности для проектов с временными ограничениями.

Помимо ISE Foundation и WebPACK компания Xilinx предлагает пакет дополнительных средств разработки, предназначенный для проектирования на системном уровне, достижения максимальной производительности, а также отладки и анализа проектов.

Программное обеспечение PlanAhead позволяет разработчикам анализировать топологию кристалла и улучшать производительность проектов. PlanAhead упрощает переход между синтезом и трассировкой проектов, обеспечивая существенное уменьшение количества и длительности итераций повторного проектирования. PlanAhead предоставляет возможность взглянуть изнутри на процесс размещения и трассировки и быстро проверить «а что будет, если...», предлагая таким образом быстро идентифицировать и устранить проблемы. В сочетании с ISE 8.1i разработчики могут получить до 30 % прироста производительности по сравнению с аналогичными решениями, в то время как для сложных проектов с несколькими тактовыми доменами можно ожидать до 56 % прироста [5;6;8].

ChipScope Pro – лучшая на рынке ПЛИС FPGA система внутрисхемной отладки. Она работает посредством внедрения в проект IP-ядер логического анализатора, шинного анализатора и виртуального ввода/вывода, позволяя наблюдать за любым заданным внутренним сигналом или узлом, включая встроенные аппаратные или софт-процессоры. Сигналы захватываются со скоростью, допустимой хост-компьютером, и передаются через интерфейс JTAG, освобождая таким образом программируемые выходы ПЛИС для использования разработчиком. Можно отметить, что по интерфейсу JTAG производится и программирование ПЛИС, соответственно для отладки не требуется какого-либо дополнительного оборудования [8].

Захваченные сигналы могут быть проанализированы с помощью логического анализатора, входящего в состав ChipScope Pro. Также ChipScope Pro совместима по интерфейсу с тестовым оборудованием Agilent, что позволяет обеспечить больше памяти для хранения данных тестирования, более высокие тактовые частоты и широкие дополнительные возможности, а кроме того, использует всего несколько выходов FPGA.

Xilinx System Generator for DSP быстро стал весьма популярным инструментом для разработки и отладки высокопроизводительных систем цифровой обработки сигналов. Он позволяет использовать наиболее производительные из имеющихся на рынке FPGA фирмы Xilinx.

Этот инструмент обеспечивает высокоуровневое представление проекта, абстрагированное от конкретной аппаратной платформы, которое автоматически компилируется в FPGA «одним нажатием кнопки» без потерь в производительности по сравнению с проектами, реализованными на языках более низкого уровня, таких как VHDL. System Generator является частью технологии XtremeDSP фирмы Xilinx, которая объединяет самые современные FPGA, средства разработки, IP-ядра, а также обеспечивается поддержкой центров разработки проектов и центров обучения Xilinx [1;5].

Xilinx Embedded Development Kit (EDK) – интегрированный программный пакет для сквозной разработки встраиваемых программируемых процессорных систем. Этот пакет включает программное средство Platform Studio, всю необходимую документацию и IP-ядра, которые могут потребоваться для разработки встраиваемых систем на основе FPGA фирмы Xilinx с встроенными аппаратными ядрами процессора PowerPC и/или софт-процессорами MicroBlaze.

В дополнение к вышеперечисленным средствам проектирования компания Xilinx предлагает ModelSim Xilinx Edition-III, сокращенную версию очень популярного в отрасли средства моделирования, и интерфейсы ко всем САПР других известных фирм. На примере столь исчерпывающего набора средств разработки становится понятно, почему по оценкам независимых экспертов программное обеспечение Xilinx регулярно занимает первые места среди средств проектирования; теперь у разработчиков появилась возможность бесплатно освоить и оценить полную мощь средств проектирования Xilinx.

В маршрут проектирования САПР ISE интегрируются продукты сторонних производителей, поэтому можно использовать передовые средства синтеза логических схем таких фирм, как Mentor Graphics, Cadence, Synplicity. Ввод проекта осуществляется самыми разными способами – начиная от графического ввода схем и заканчивая современными языками описания аппаратуры (VHDL, Verilog). Можно использовать и высокоуровневые языки описания аппаратуры (HLL, High Level Languages), такие как System C, Handel C и им подобные.

Контрольные вопросы

1. В чем заключается принцип работы перепрограммируемых ПЗУ?
2. Какие транзисторы используются в технологии EPROM?
3. Какие основные преимущества современных ПЛИС по сравнению с другими программируемыми микросхемами вы можете назвать?
4. Охарактеризуйте основные методы проектирования ПЛИС. В чем их плюсы и минусы?
5. Назовите основные этапы проектирования цифровых устройств на ПЛИС.
6. В чем заключается основные особенности архитектуры CPLD?
7. Назовите основные особенности архитектуры FPGA?
8. Перечислите основные составные элементы ПЛИС CPLD.
9. Что такое макроячейка?
10. Что такое переключаящая матрица?
11. Перечислите основные составные элементы ПЛИС FPGA.
12. Назовите основные типы соединений в ПЛИС FPGA.
13. Поясните принцип работы блоков управления тактовыми сигналами в ПЛИС FPGA.
14. Объясните принцип устройства блочного ОЗУ в ПЛИС FPGA.
15. Каковы основные части программного обеспечения фирмы Xilinx для проектирования цифровых устройств на ПЛИС?
16. Какие основные производители программируемых логических интегральных схем вам известны?
17. В чем особенности ПЛИС семейства Vertex?
18. Каковы особенности ПЛИС семейства Spartan?
19. Назовите основные варианты программаторов для ПЛИС.
20. Перечислите основные программные продукты для моделирования разрабатываемых цифровых устройств на ПЛИС.

Глава 2. ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ УСТРОЙСТВ НА БАЗЕ ПЛИС

2.1. Особенности проектирования цифровых устройств на базе ПЛИС

Создание цифровых устройств традиционным способом, используя стандартные интегральные схемы комбинационного и последовательностного типов, является специализированным процессом с определенными индивидуальными функциями для каждого специалиста. Этот процесс включает различные стадии: проектирование, подготовку производства и отработку технологии. Создание цифровых устройств на основе специализированных интегральных схем высокой степени интеграции также требует сложного процесса. Однако использование программируемых логических интегральных схем позволяет значительно сократить время их создания, ограничиваясь только этапом проектирования с компьютером.

Главным преимуществом ПЛИС является их универсальность и возможность быстрого программирования для выполнения функций практически любого цифрового устройства. ПЛИС служит полумонтажом, который может программироваться разработчиком с помощью персонального компьютера, специального аппаратного оборудования и программного обеспечения, такого как система автоматизированного проектирования (САПР). Это позволяет проектировщику цифровых устройств использовать ПЛИС для создания устройств в кратчайшие сроки без необходимости привлечения дополнительных специалистов.

Современные САПР для ПЛИС работают под управлением операционных систем Windows и воспроизводят все преимущества их графического интерфейса. Производители ПЛИС обычно предоставляют программное обеспечение для поддержки своих продуктов. Процесс проектирования ПЛИС с использованием САПР включает следующие этапы:

1. Разработка структурной формулы проектируемого цифрового устройства, ее минимизация и адаптация под тип и возможности используемой ПЛИС (для этих целей используются методы, рассмотренные в предыдущей главе).
2. Создание нового файла проекта, назначение семейства ПЛИС для проекта.

3. Ввод структурной формулы или схемы проектируемого устройства с помощью специальных программ – редакторов.

4. Компиляция введенной структурной формулы или схемы, т. е. получение битовой последовательности загружаемой в ПЛИС программы.

5. Временной анализ задержек в отдельных элементах и проверка работоспособности всей структуры цифрового устройства путем моделирования с помощью специальных программ – симуляторов.

6. Программирование микросхемы ПЛИС с помощью специальных аппаратных средств – программаторов.

Проектировщикам, использующим систему автоматизированного проектирования (САПР), доступны обширные программные библиотеки с функциями стандартных интегральных схем ТТЛ и КМОП. Это позволяет синтезировать схему цифрового устройства не только из элементарных функций И, ИЛИ и НЕ, но и с использованием готовых программных аналогов реальных микросхем. Широкие возможности предоставляет также глобальная сеть Интернет, через которую распространяются специально разработанные библиотеки компонентов, которых нет в стандартных версиях САПР. Для создания схемы цифрового устройства на основе ПЛИС часто используются языки описания аппаратных средств (HDL), которые позволяют задать типы устройств, определить входные воздействия и связи между ними, а также задать константы и переменные в проекте.

Среди лидеров в производстве ПЛИС и САПР сегодня можно выделить компании Altera Corporation, Xilinx и Actel Corporation. Они занимают до 80 % рынка ПЛИС и предлагают устройства с частотой работы до нескольких сотен мегагерц, емкостью до нескольких миллионов элементов и стоимостью в десятки долларов США. Благодаря кратчайшим срокам проектирования и высокой надежности, обусловленной высокой степенью интеграции элементов на кристалле полупроводника и минимальным количеством внешних межсоединений, ПЛИС являются предпочтительным выбором по сравнению со стандартными логическими микросхемами комбинационного и последовательностного типов.

2.2. Пакет программ WebPack ISE

Программируемые логические интегральные схемы все более широко используются для создания цифровых систем различного назначения. Фирма Xilinx, являясь ведущим мировым производителем ПЛИС, предоставляет разработчикам широкий спектр кристаллов с различной технологией производства, степенью интеграции, архитектурой, быстродействием, потребляемой мощностью и напряжением питания, выпускаемых в различных типах корпусов и в нескольких вариантах исполнения, включая промышленное, военное и радиационно-стойкое [1 – 5].

Кристаллы, выпускаемые фирмой Xilinx, в полной мере реализуют преимущества ПЛИС по сравнению с «жесткой логикой»:

- высокое быстродействие;
- возможность перепрограммирования непосредственно в системе;
- высокая степень интеграции, позволяющая разместить цифровое устройство в одном кристалле и тем самым снизить время и затраты на трассировку и производство печатных плат;
- сокращение времени цикла разработки и производства устройства;
- наличие мощных инструментов САПР, позволяющих устранить возможные ошибки в процессе проектирования устройства;
- сравнительно низкая стоимость (в пересчете на один логический вентиль);
- возможность последующей реализации проектов ПЛИС для серийного производства в виде заказных СБИС, что позволяет значительно снизить их себестоимость.

До недавнего времени, несмотря на все достоинства ПЛИС Xilinx, существовало обстоятельство, сдерживающее их применение (особенно недорогих кристаллов при разработке несерийных устройств) – необходимость дополнительных затрат на приобретение пакета программных средств проектирования и программирования. Чтобы устранить это препятствие, фирма Xilinx предоставила разработчикам возможность использовать бесплатное программное обеспечение – пакет WebPACK™ ISE™ (Integrated Synthesis Environment). Цель настоящей

публикации – познакомить разработчиков цифровых устройств с возможностями САПР WebPACK ISE и основами методики выполнения проектов в среде данного пакета.

2.3. Основные характеристики пакета WebPACK ISE

Программные средства WebPACK ISE представляют собой систему сквозного проектирования, которая реализует все этапы создания цифрового устройства на базе ПЛИС, включая программирование кристалла: разработка проекта, синтез, моделирование, трассировка и загрузка в кристалл. Версия 3.3WP8.0 САПР WebPACK ISE предназначена для проектирования цифровых устройств на базе ПЛИС производства фирмы Xilinx, относящихся как к семействам CPLD: XC9500, XC9500XL, XC9500XV, XCR22V10, XCR3000 (XPLA1_3, XPLA2), XCR3000XL (XPLA3), XCR5000 (XPLA1_5), так и FPGA: Spartan™-II, Virtex™-E (только кристалл XCV300E), Virtex-II (кристаллы 2V40, 2V80 и 2V250).

Отличительные особенности пакета:

- поддержка различных методов описания проектируемых устройств (графических и текстовых);
- возможность использования проектов, подготовленных в других системах проектирования, в том числе в среде пакета Altera MAX+PlusII™;
- наличие схемотехнического редактора, укомплектованного набором обширных библиотек;
- интеллектуальные средства создания HDL (Hardware Description Language)-описаний, формирующие шаблоны на основании информации, предоставляемой пользователем, для языков описания аппаратуры VHDL, Verilog™ и ABEL™ HDL;
- высокоэффективные средства синтеза HDL-проектов, поддерживающие языки VHDL, Verilog и ABEL HDL, с возможностью оптимизации;
- развитые средства верификации проекта, позволяющие сократить полное время разработки устройства за счет обнаружения возможных ошибок на более ранних стадиях проектирования и сокращения длительности и количества возможных итераций;

– автоматические средства трассировки проекта в кристаллы различных семейств ПЛИС Xilinx с учетом оптимизации проекта по различным параметрам;

– средства программирования кристаллов семейств ПЛИС Xilinx, выполненных по различной технологии (CPLD и FPGA), поддерживающие несколько типов загрузочных кабелей JTAG-интерфейса;

– удобный для разработчика пользовательский интерфейс и наличие в каждом модуле пакета справочной системы, сокращающие время освоения САПР;

– наличие интегрированного с пакетом САПР набора инструментов и утилит других фирм, предоставляющих дополнительные удобства в процессе проектирования, включающего утилиту генерации тестовых сигналов HDL Bencher™, программу моделирования ModelSim XE Starter™ и редактор диаграмм состояний StateCAD™.

Для получения программного обеспечения WebPACK ISE необходимо зарегистрироваться на web-сайте www.xilinx.com. После регистрации следует выполнить процедуру копирования модулей пакета на ПК, который будет использован для развертывания САПР. Пакет работает под управлением операционных систем Windows 98, Windows NT 4.0, Windows 2000. Требования, предъявляемые к аппаратным ресурсам ПК, следующие: процессор – не ниже Pentium 75 МГц; ОЗУ – не менее 16 Мбайт; размер свободного пространства на жестком диске для установки WebPACK ISE в полном объеме должен составлять не менее 350 Мбайт; для последующей работы с пакетом необходимо не менее 60 Мбайт [1; 4 – 6].

Дистрибутив пакета выполнен в виде набора модулей, каждый из которых представляет собой самораспаковывающийся архив. После копирования следует поочередно запустить на выполнение каждый из полученных файлов. По окончании распаковки каждого архива автоматически запускается программа установки соответствующих модулей. Пользователь должен последовательно выполнить все инструкции каждой программы инсталляции. Следует обратить внимание на то, что после установки программы моделирования ModelSim XE Starter следует получить файл лицензии. Для этого требуется запустить программу Licensing Wizard, которая соберет необходимую для получения лицензионного кода информацию об используемом ПК. Файл с этой информацией должен быть отправлен по электронной почте. Если ПК разработчика подключен к Интернету, то возможна регистрация в режиме online [9].

2.4. Пользовательский интерфейс пакета WebPACK ISE

Управляющая оболочка пакета WebPACK ISE Навигатор проекта (Project Navigator™) предоставляет пользователю удобный интерфейс для работы с проектом и управления всеми процессами проектирования и программирования ПЛИС. Запуск всех необходимых программных модулей пакета осуществляется непосредственно в среде Навигатора проекта. Основное окно Навигатора проекта (рис. 2.1) помимо стандартных элементов управления (основного меню и оперативной панели управления) содержит четыре встроенных окна:

- окно исходных модулей (файлов) проекта (Sources in Project);
- окно необходимых процедур (процессов) для выбранного исходного модуля проекта (Processes for Current Source);
- окно консольных сообщений программных модулей (Console);
- окно редактора текстовых HDL-описаний проекта.

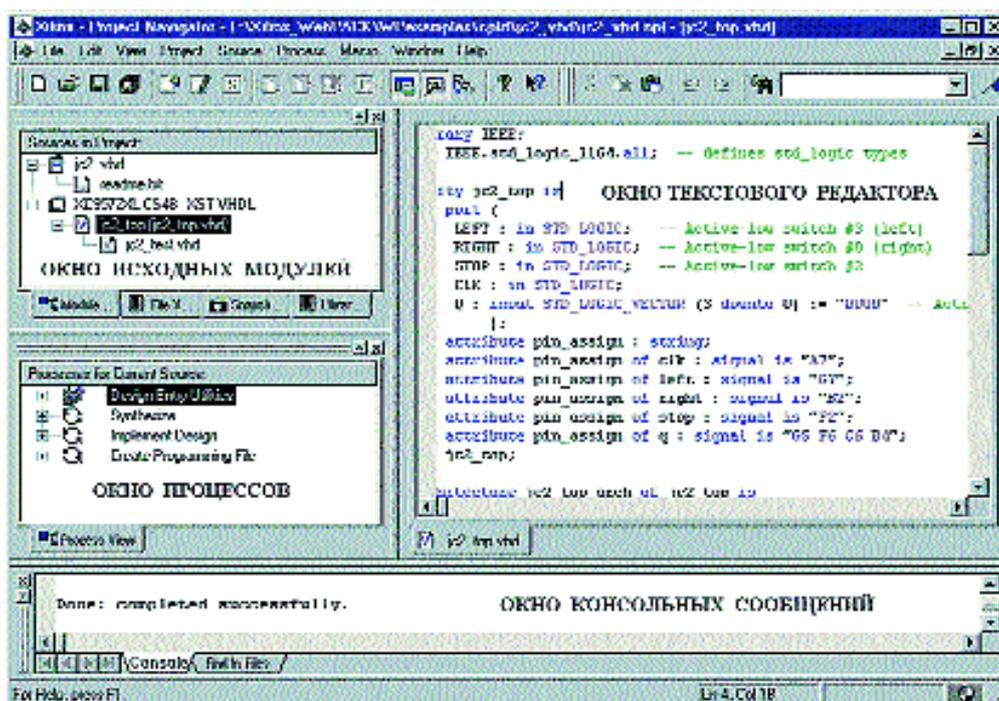


Рис. 2.1. Основное окно Навигатора проекта пакета WebPACK ISE

В окне исходных модулей (файлов) проекта отображается иерархическая структура, состоящая из модулей (файлов), в которых содержится описание проектируемого устройства в графической или текстовой форме, а также описание тестовых воздействий, используемых в

процессе моделирования. Каждый тип модуля имеет соответствующее графическое обозначение – пиктограмму.

Окно необходимых процедур (процессов) показывает маршрут обработки выделенного исходного модуля в процессе проектирования устройства. Таким образом, в данном окне подробно отображаются все этапы процесса разработки и программирования ПЛИС, делая последний «прозрачным» для пользователя САПР. Последовательность и содержание этапов определяется типом исходного модуля и семейством ПЛИС. Навигатор проекта автоматически показывает в окне процедур (процессов) структуру процесса проектирования, соответствующую выбранному семейству ПЛИС, исключая тем самым возможные ошибки в последовательности действий разработчика. В этом же окне указывается информация о дополнительных инструментах, которые могут быть использованы на каждом этапе.

Окно консольных сообщений предназначено для вывода информации программных модулей пакета, работающих в консольном режиме. Ряд программных модулей пакета WebPACK ISE, как, например, программы трансляции, синтеза, автоматической трассировки, являются консольными приложениями, т. е. не создают собственных окон. Чтобы информация о ходе выполнения этих программ была доступна разработчику непосредственно в процессе работы с проектом, она отображается в окне консольных сообщений Навигатора проекта.

Окно интегрированного текстового редактора становится активным, если для проектируемого устройства или используемых библиотек выбран способ описания на языке HDL.

2.5. Этапы проектирования цифровых устройств на базе ПЛИС Xilinx

В процессе создания цифровых устройств на базе ПЛИС Xilinx можно выделить следующие этапы:

- создание нового проекта (выбор семейства и типа ПЛИС, а также средств синтеза);
- подготовка описания проектируемого устройства в схемотехнической, алгоритмической или текстовой форме;
- синтез устройства;

- функциональное моделирование;
- трассировка проекта в кристалл;
- временное моделирование;
- программирование ПЛИС (загрузка проекта в кристалл).

Исходная информация о проектируемом устройстве может быть представлена в виде принципиальных схем, описаний на языке HDL, диаграмм состояний и библиотек пользователя. В процессе синтеза на основании исходных модулей проекта формируется список цепей, который далее используется средствами трассировки в качестве исходных данных. Функциональное моделирование устройства производится без учета реальных значений задержек прохождения сигналов и позволяет проконтролировать соответствие выходных сигналов алгоритмам работы проектируемого устройства.

На этапе трассировки проекта в кристалл производятся распределение выполняемых функций в конфигурируемые логические блоки CLB (Configurable Logic Block) или макроячейки Macrocell в зависимости от используемого семейства ПЛИС и формирование необходимых связей в кристалле. В процессе трассировки проекта в кристалл также определяются реальные значения задержек распространения сигналов, которые необходимы для полного (временного) моделирования устройства. Основным результатом этапа трассировки – формирование файла, в котором содержится информация о конфигурации ПЛИС, реализующей проектируемое устройство. Завершением процесса разработки цифрового устройства является загрузка конфигурационных данных в кристалл с помощью соответствующих программ и загрузочного кабеля.

Следует обратить внимание на то, что этапы функционального и временного моделирования не являются обязательными. Пренебрегать этими этапами, однако, не рекомендуется, так как высокоэффективные средства моделирования пакетов САПР Xilinx позволяют обнаружить большинство возможных ошибок и тем самым значительно сократить общее время разработки устройства. При обнаружении ошибок на любом из этапов (например, логических ошибок на этапе функционального моделирования или при получении неудовлетворительных результатов временного моделирования) следует вернуться на стадию

разработки исходных описаний проекта, внести необходимые изменения и повторить последующие этапы.

Далее кратко рассматривается поэтапный процесс создания цифрового устройства в среде пакета WebPACK ISE.

2.6. Методика работы с проектом в среде пакета WebPACK ISE

Создание нового проекта инициируется командой FILE/New Project основного меню. При выборе этого пункта меню открывается диалоговая панель (рис. 2.2), в которой разработчик должен указать имя и расположение проекта на жестком диске, а также выбрать семейство ПЛИС, тип кристалла и средства синтеза устройства. После ввода указанных данных в окне исходных модулей проекта появится пиктограмма основного модуля с указанием типа кристалла и инструментов синтеза.

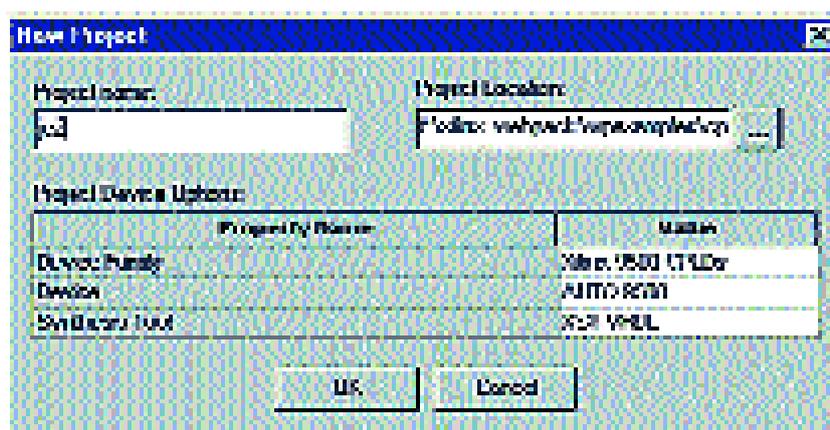


Рис. 2.2. Диалоговая панель параметров нового проекта

Для ввода описания проекта следует выбрать пункт New Source основного меню или нажать соответствующую кнопку на панели инструментов. Далее открывается список возможных типов исходных модулей: схемотехническое представление, описание модулей, библиотек и тестовых воздействий на языке HDL, диаграмма состояний, модули документации. Разработчик должен выбрать тип нового исходного модуля и указать имя файла для его последующего сохранения. В зависимости от типа создаваемого исходного модуля открывается окно схе-

мотехнического редактора Schematic Editor (рис. 2.3), редактора диаграмм состояний StateCad (рис. 2.4), генератора тестов HDL Bencher (рис. 2.5) или активизируется окно текстового редактора.

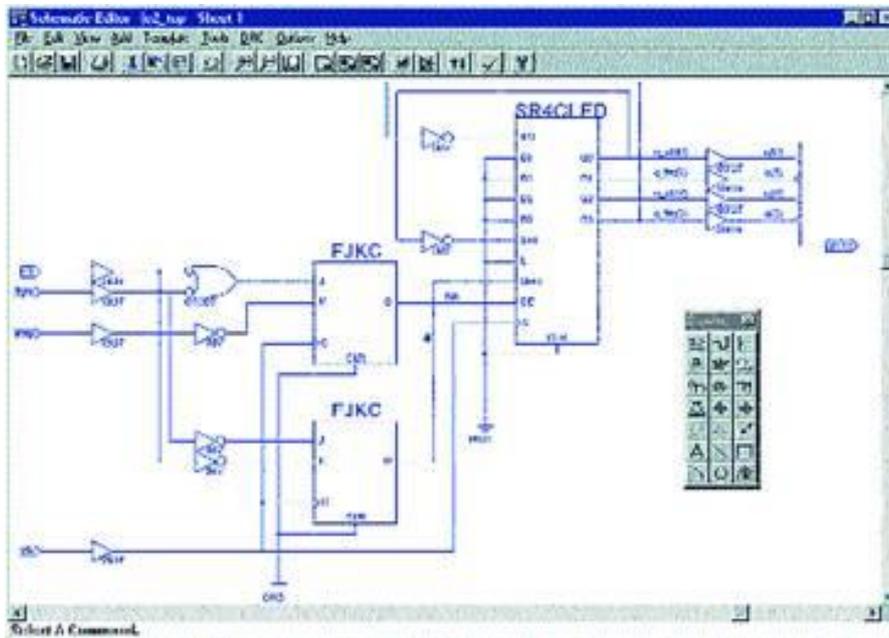


Рис. 2.3. Рабочее окно схмотехнического редактора Schematic Editor пакета WebPACK ISE

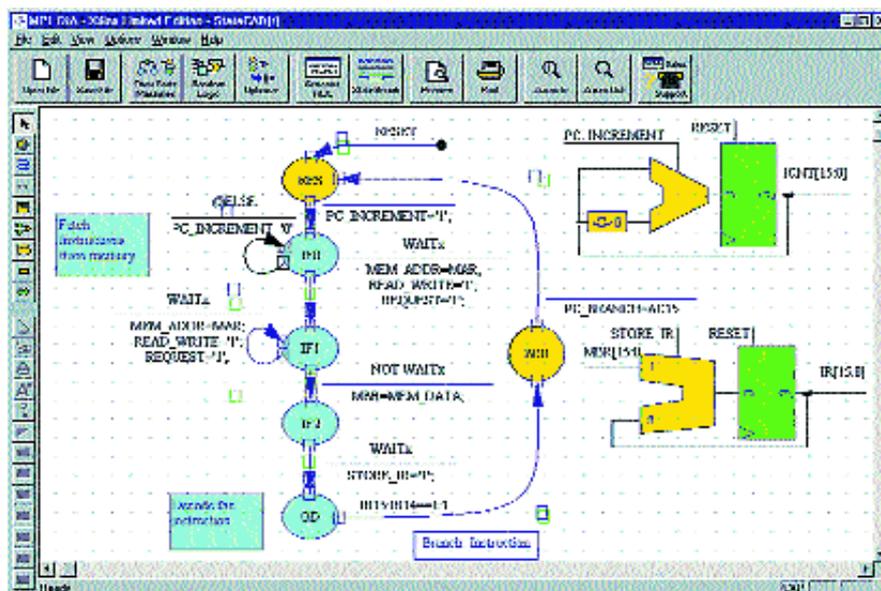


Рис. 2.4. Окно редактора диаграмм состояний StateCad

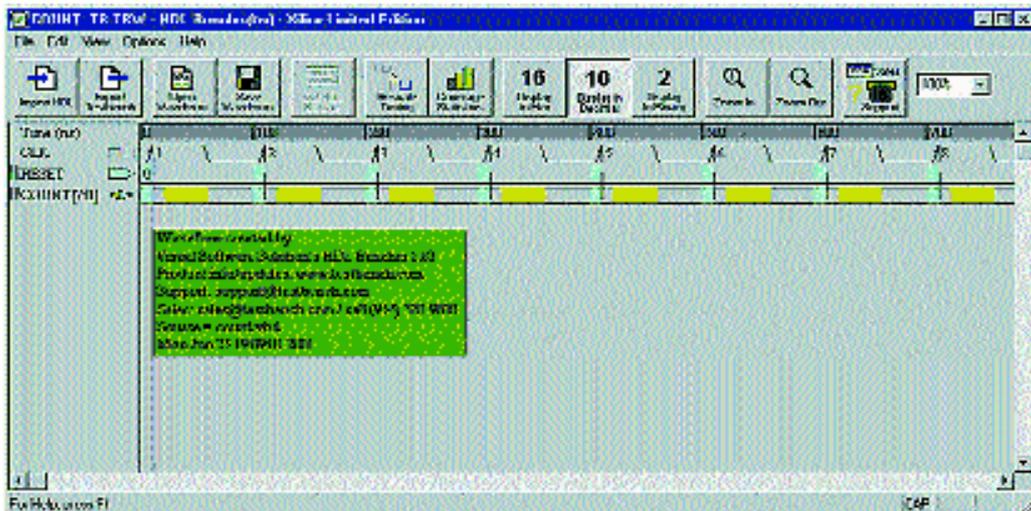


Рис. 2.5. Внешний вид окна генератора тестов HDL BenchTools

В случае успешного завершения создания исходного модуля он автоматически добавляется к проекту и отображается в окне исходных модулей в виде соответствующей пиктограммы. Разработчик должен поочередно создать все необходимые модули описания устройства, после чего перейти к следующему этапу проектирования.

При выборе способа описания проектируемого устройства рекомендуется использовать язык описания VHDL как наиболее эффективный и перспективный метод. Для разработчиков, использующих САПР других фирм, предоставлена возможность ввода исходных данных проекта в виде списка соединений Netlist.

Выделив подготовленный модуль описания проекта в окне исходных модулей, разработчик получает в окне процессов поэтапную структуру последующих процедур проектирования. Если перед названием этапа указан знак «+», то этот этап включает в себя несколько процедур. Чтобы увидеть структуру такого этапа в развернутом виде, пользователь должен поместить курсор мыши на изображение значка «+» и щелкнуть левой кнопкой.

Прежде чем перейти непосредственно к этапу синтеза, разработчик при необходимости может воспользоваться утилитами ввода проекта (Design Entry Utilities), например, редактором временных и топологических ограничений (Constraints Editor), инструментами подготовки тестов для моделирования HDL-проектов (HDL BenchTools). Перед активизацией того или иного процесса следует указать его параметры. Для этого необходимо выделить строку с названием процесса и

выбрать в основном меню команду редактирования свойств процесса Process/Properties или нажать соответствующую кнопку на панели инструментов. На экран выводится диалоговая панель, содержащая список параметров, доступных пользователю. Опции процессов могут быть разбиты на группы, которые представлены на отдельных вкладках диалоговой панели. Содержание списка параметров определяется выполняемым процессом и семейством ПЛИС, на базе которого реализуется проект. Так, например, на рис. 2.6 показана диалоговая панель свойств для этапа трассировки проекта в кристалл семейства CPLD XC9500 [1 – 3]. Разработчик может оставить значения параметров, предлагаемые по умолчанию, или при необходимости установить требуемые значения.

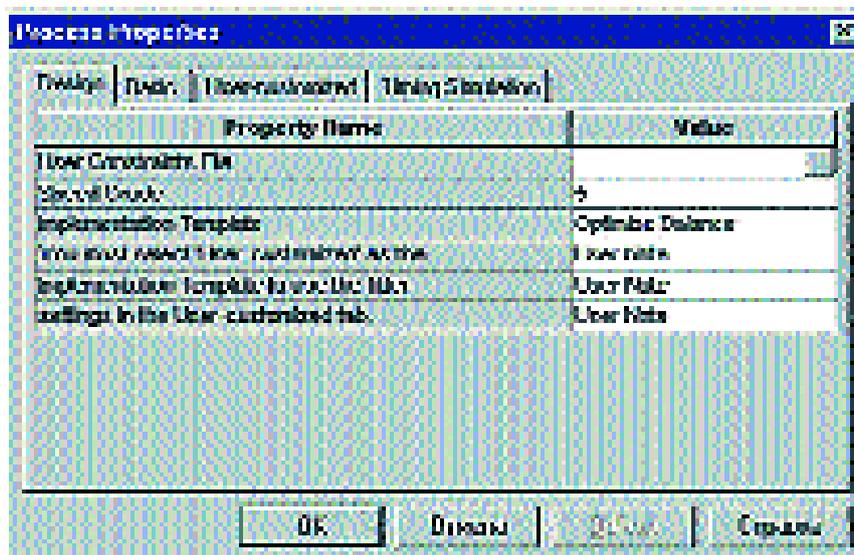


Рис. 2.6. Диалоговая панель параметров процесса трассировки проекта в кристалл семейства CPLD XC9500

Чтобы выполнить процедуру (активизировать процесс), следует выбрать команду Process/Run основного меню или просто дважды щелкнуть левой кнопкой мыши на названии соответствующей процедуры в окне процессов. В случае успешного выполнения процедуры в окне консольных сообщений после названия процесса отображается строка:

Done: completed successfully.

Кроме того, в окне процессов перед названием выполненной процедуры появляется пиктограмма в виде символа «V», соответствующая

успешному завершению процесса. При обнаружении ошибок в окне консольных сообщений выводится строка с указанием кода ошибки, модуля и строки в модуле. После сообщений об ошибках отображается строка, указывающая на неудачное завершение процесса, и соответствующий код:

Done: failed with exit code: 0001.

В окне процессов неудачное завершение процедуры обозначается пиктограммой в виде символа «X» красного цвета.

Кроме консольных сообщений и пиктограмм в окне процессов после выполнения процедур на каждом этапе создается отчет (Report), который содержит подробную информацию о ходе и результатах выполнения процесса. Рекомендуется анализировать отчеты для каждого этапа проектирования не только в случае обнаружения ошибок, но и при успешном выполнении процедур.

Этапы синтеза и трассировки выполняются в пакете WebPACK ISE автоматически. Разработчику необходимо только определить параметры этих процессов. Моделирование устройства осуществляется в среде программы ModelSim XE Starter с использованием тестов, сформированных с помощью программы HDL Bencher. Управление процессом моделирования может осуществляться с помощью как элементов управления ModelSim XE Starter (основного меню и кнопок быстрого доступа), так и командного файла, подготовленного ранее.

После успешного завершения этапа временного моделирования можно приступить непосредственно к программированию кристалла. Загрузочный кабель рекомендуется заранее подключить к используемому порту ПК. Навигатор проекта автоматически выберет средства программирования для используемого семейства ПЛИС. В программаторе следует определить порт ПК, зарезервированный для подключения кабеля, используя для этого команду автоматического обнаружения кабеля. Далее следует выбрать кристалл, указать файл конфигурации и запустить команду загрузки конфигурации.

Если в процессе проектирования цифрового устройства разработчику стали необходимы кристаллы ПЛИС с более высокой степенью интеграции или семейств, которые не поддерживаются используемой версией пакета WebPACK ISE, рекомендуется прежде всего выяснить

возможность получения новой версии САПР. Фирма Xilinx периодически обновляет версии предлагаемого программного обеспечения для проектирования ПЛИС. Это относится не только к коммерческим, но и свободно распространяемым программным продуктам. Новые версии САПР, как правило, поддерживают большее количество кристаллов, а также включают более эффективные средства синтеза и трассировки проектов. Если же и последняя версия пакета WebPACK ISE не поддерживает требуемый тип кристаллов, то необходимо перейти к использованию одного из следующих пакетов САПР: Foundation Series™ или Foundation ISE™. Для разработчиков, использующих САПР других производителей, предназначены программные средства Alliance Series™.

Фирма Xilinx – разработчик ПЛИС, поставляет также САПР для разработки устройств на FPGA и CPLD. Пожалуй, из всех производителей ПЛИС фирма Xilinx может считаться лидером по номенклатуре серий ПЛИС и программного обеспечения (ПО). При этом новые версии ПО поддерживают старые серии ПЛИС, позволяя разработчику производить плавную миграцию проектов на новые серии. Первоначально фирма ориентировалась на поддержку своих САПР дополнительными программными средствами других компаний, интегрируя их под заказчика.

Сегодня среди программных продуктов Xilinx имеются как относительно простые пакеты проектирования, так и мощные, интегрированные программные решения, позволяющие разрабатывать ПЛИС эквивалентной емкостью более 1000000 вентилей. Среди облегченных систем проектирования Xilinx следует выделить систему WebFITTER, особенностью которой является возможность использования в ходе проектирования web-ресурсов Internet. Альтернативой использованию WebFITTER считается применение пакета WebPack ISE, ориентированного на проектирование CPLD схем. В последней системе ввод описания проекта возможен как с помощью схемного редактора, так и с использованием языков описания аппаратуры ABEL и VHDL. Возможно программирование устройств непосредственно в системе с применением аппаратного загрузчика XChecker.

Контрольные вопросы

1. Какие основные этапы проектирования ПЛИС с использованием САПР вы знаете?
2. Назовите основных производителей современных программируемых логических интегральных схем.
3. В чем заключаются основные преимущества ПЛИС перед цифровыми микросхемами с «жесткой» логикой?
4. Каковы отличительные особенности пакета WebPACK ISE?
5. Поясните содержимое окна Навигатора проекта пакета WebPACK ISE.
6. Что такое синтез устройства на этапе проектирования ПЛИС?
7. Что такое функциональное моделирование на этапе проектирования ПЛИС?
8. После какого этапа можно приступать к трассировке разрабатываемого проекта в кристалл?
9. Что такое временное моделирование?
10. Какой этап проектирования ПЛИС является заключительным?
11. Что является основным результатом этапа трассировки?
12. Являются ли обязательными этапы функционального и временного моделирования?
13. Какой язык является наиболее эффективным при разработке цифровых устройств на ПЛИС?
14. Что такое Schematic Editor?
15. Как называется редактор диаграмм состояний?
16. Расскажите о генераторе тестов HDL Bencher.
17. Что такое Constraints Editor?
18. В каком режиме осуществляются этапы синтеза и трассировки в пакете WebPACK ISE?
19. Как подключить загрузочный кабель для осуществления процесса программирования ПЛИС?
20. В какой среде осуществляется моделирование разрабатываемого цифрового устройства?

Глава 3. ПРИМЕРЫ РЕАЛИЗАЦИИ РАЗЛИЧНЫХ ЦИФРОВЫХ УСТРОЙСТВ НА ПЛИС

3.1. Преобразование последовательного кода в параллельный

На рис. 3.1 представлена схема преобразования последовательного кода в параллельный.

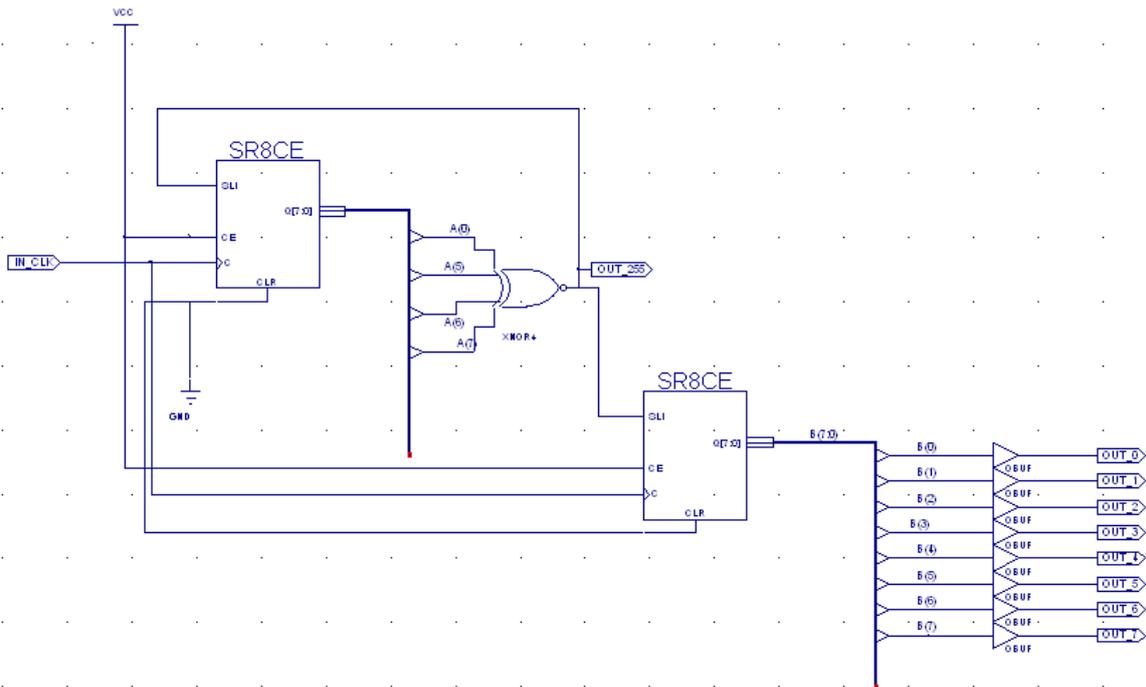


Рис. 3.1. Преобразователь последовательного кода в параллельный

Преобразователь выполнен на восьмиразрядном сдвиговом регистре, что позволяет получать восьмиразрядный параллельный код. На вход сдвигового регистра поступает псевдослучайная последовательность (рис. 3.2), которая преобразуется в последовательность байтов. Временные диаграммы приведены на рис. 3.3.

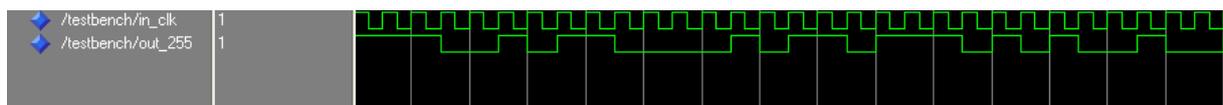


Рис. 3.2. Входная псевдослучайная последовательность на входе преобразователя

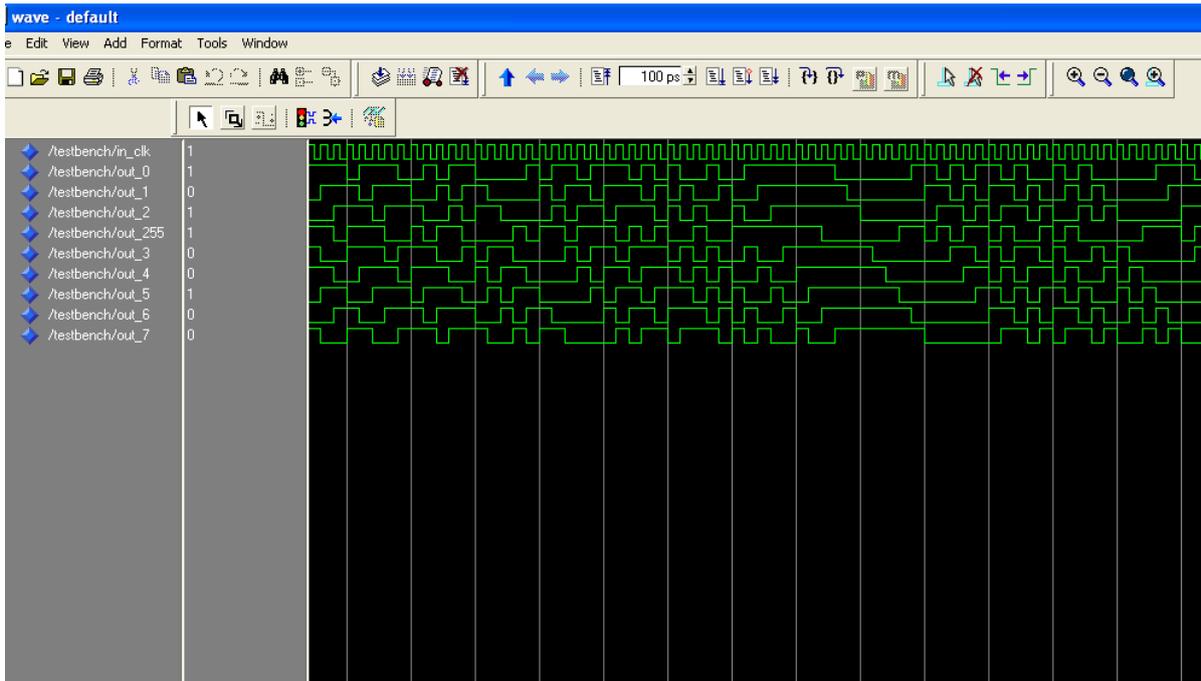


Рис. 3.3. Временные диаграммы на выходе последовательно-параллельного преобразователя

На рис. 3.4 изображено внутреннее устройство сдвигового регистра на D-триггерах.

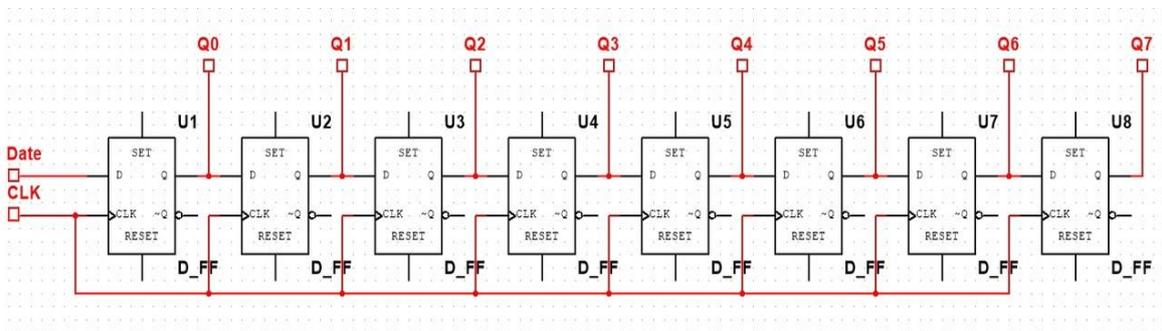


Рис. 3.4. Восьмиразрядный сдвиговый регистр на D-триггерах

3.2. Преобразование параллельного кода в последовательный код

На рис. 3.5 приведена схема преобразования из последовательного кода в параллельный и обратно в последовательный. На входе расположен представленный выше последовательно-параллельный преобразователь. Данные с входного преобразователя записываются в буфер, схема которого на D-триггерах изображена на рис. 3.6, и хранятся в нем

в течение 8 тактов. На выходе преобразователь состоит из восьмиразрядного мультиплексора и трехразрядного счетчика. Двоичные сигналы счетчика поступают на адресные входы мультиплексора, за счет чего осуществляется преобразование.

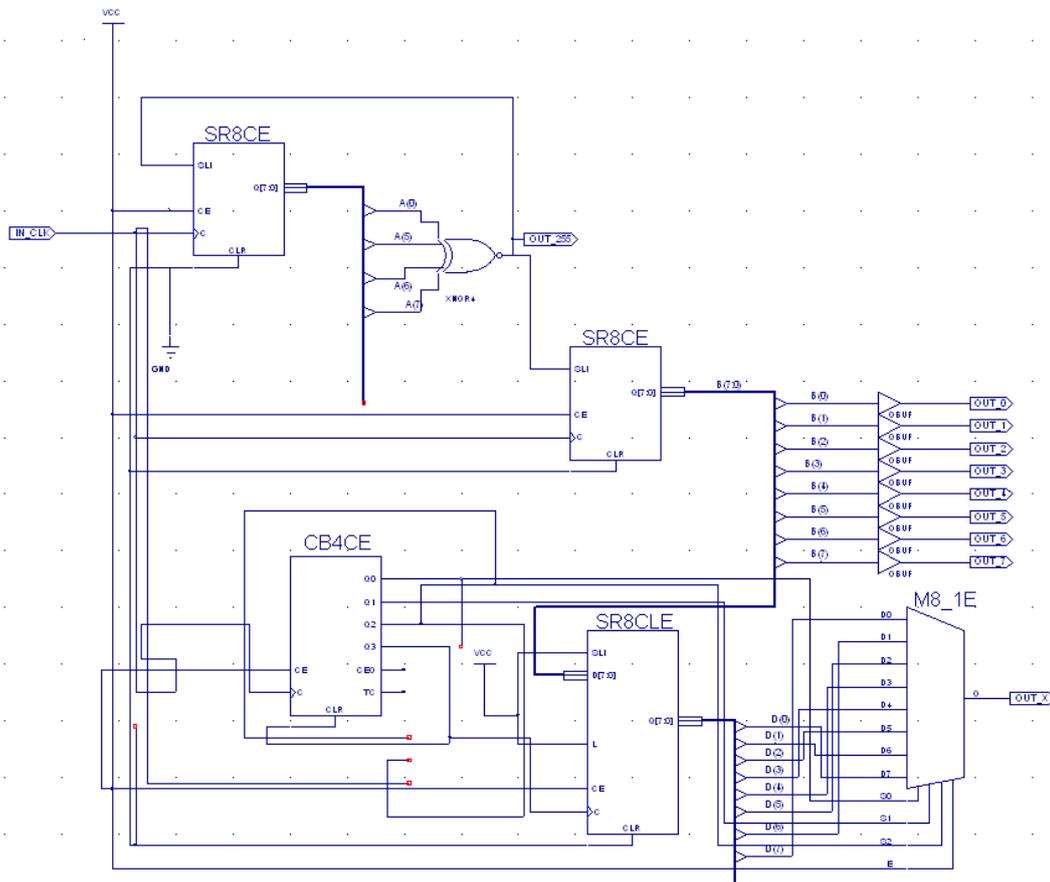


Рис. 3.5. Преобразователь параллельного кода в последовательный

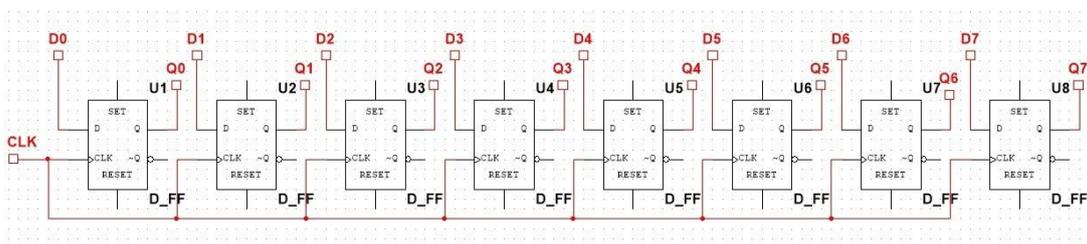


Рис. 3.6. Восмиразрядный буфер

На рис. 3.7 изображены временные диаграммы входной и выходной последовательностей. Можно наблюдать некоторую задержку во время двойного преобразования, что обусловлено наличием задержки в 8 тактов во время записи в сдвиговый регистр на входе.

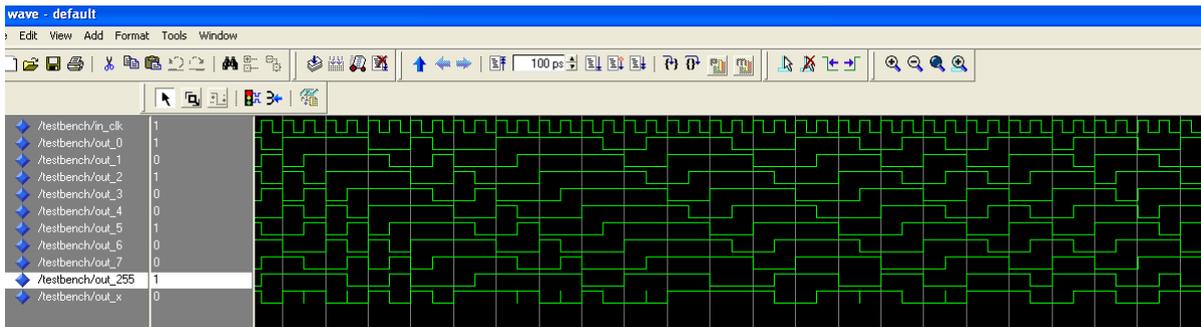


Рис. 3.7. Входные и выходные диаграммы двойного преобразователя

3.3. Сверточный кодер

На рис. 3.8 приведена схема сверточного кодера 5/7. Сверточный кодер представляет собой трехразрядный сдвиговый регистр и два устройства суммирования по модулю 2. На выходе кодера присутствуют два потока данных. Временные диаграммы закодированных потоков приведены на рис. 3.9.

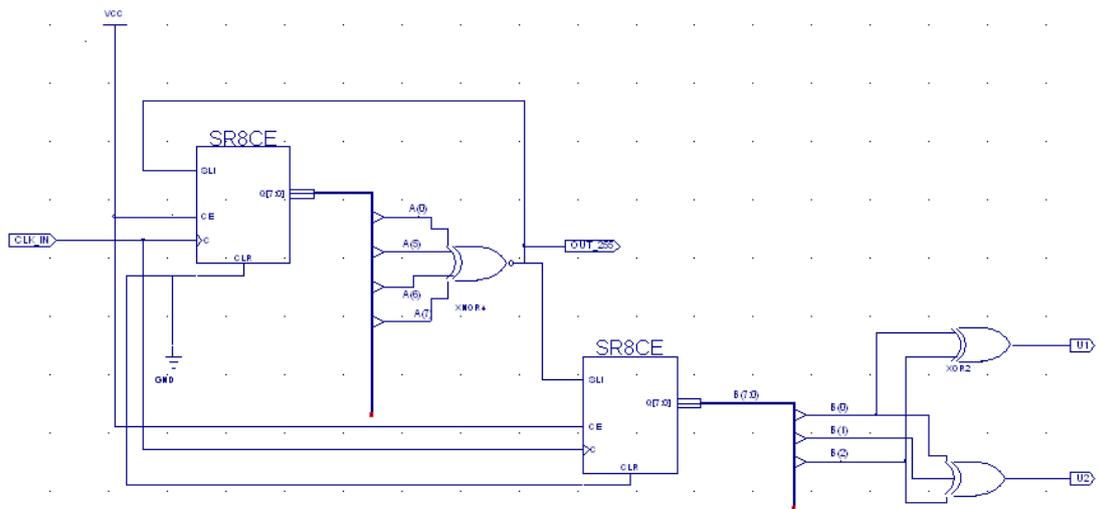


Рис. 3.8. Сверточный кодер

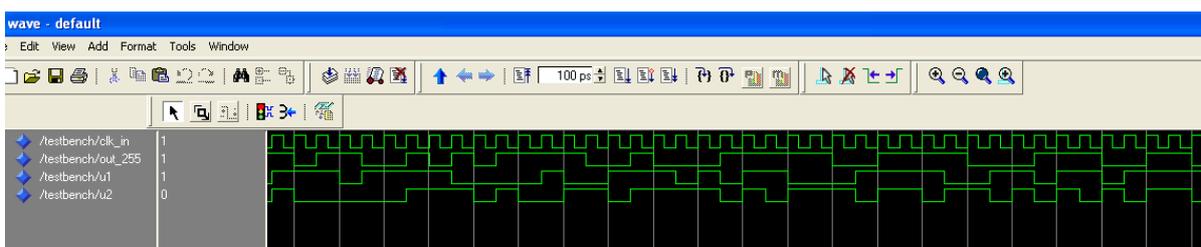


Рис. 3.9. Временные диаграммы закодированных потоков

На рис. 3.10 дана схема полного преобразования двоичных данных, полученных с помощью генератора псевдослучайной последовательности (ГПСП), в последовательный закодированный поток. Особенностью схемы является объединение двух закодированных потоков в один, выполненное на двухразрядном демультиплексоре. Таким образом, частота сигнала на выходе вдвое выше входной. На рис. 3.11 приведен закодированный выходной поток данных.

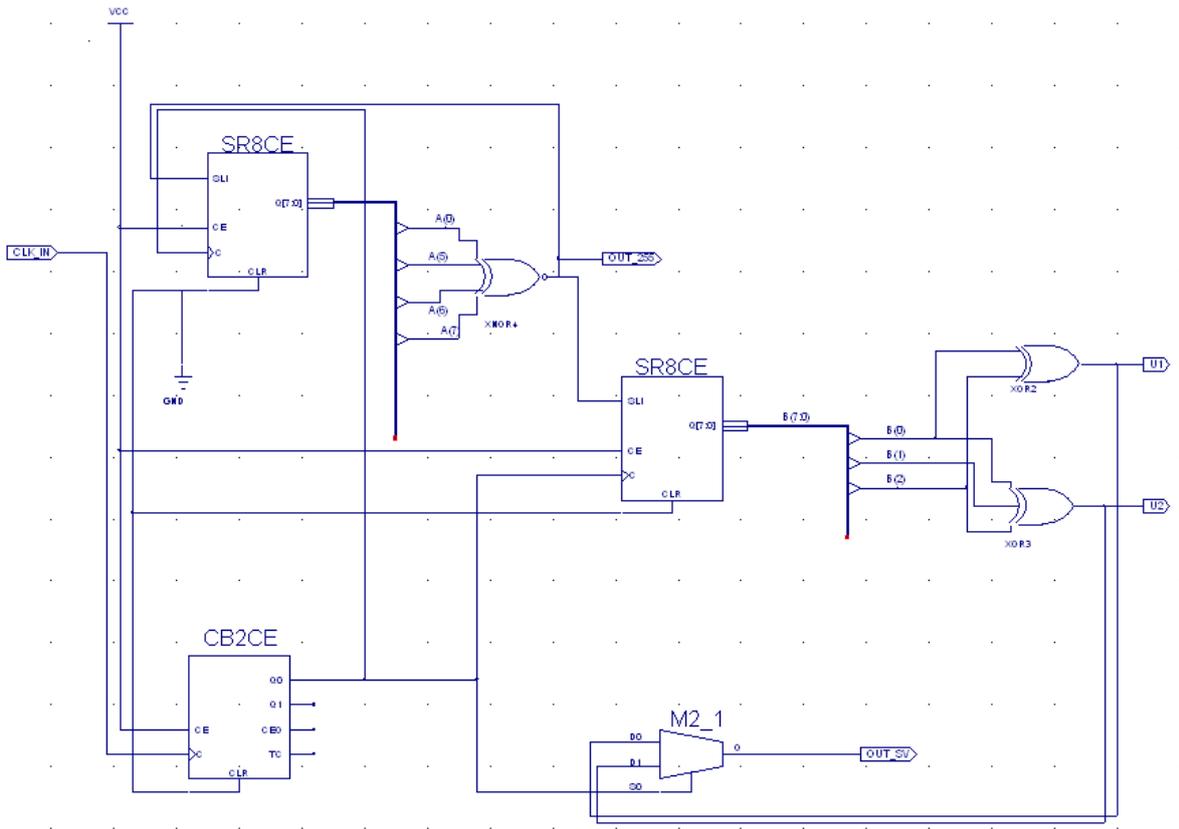


Рис. 3.10. Схема кодирования псевдослучайного потока

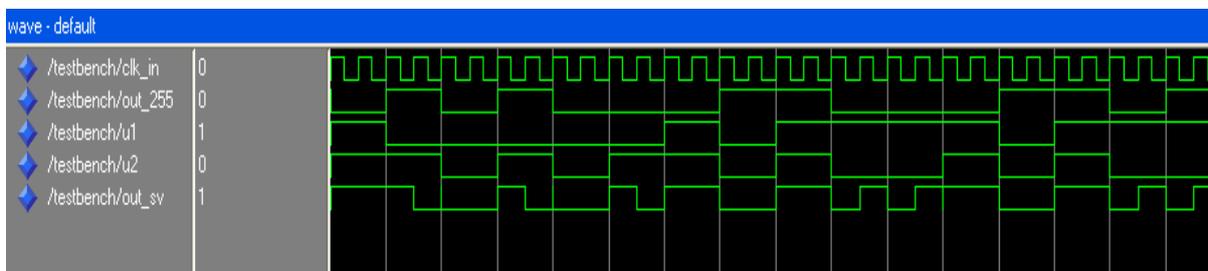


Рис. 3.11. Выходная диаграмма закодированного псевдослучайного потока данных

3.4. Генератор псевдослучайной последовательности

На рис. 3.12 приведена схема генератора псевдослучайной последовательности (ГПСП). Схема построена на восьмиразрядном циклическом сдвиговом регистре, построенном на D-триггерах, что позволяет получать псевдослучайную последовательность с периодом повторения 255 бит. Алгоритм генерации заключается в выборе некоторых строго определенных отводов сдвигового регистра и суммировании их сигналов по модулю два. Поскольку в программируемых логических схемах по умолчанию значения D-триггеров установлены в ноль, а подобного рода генераторы случайных последовательностей имеют нулевое запрещенное состояние во всех ячейках одновременно, в цепи обратной связи включен инвертор. Временная диаграмма полученной псевдослучайной последовательности относительно тактовых сигналов представлена на рис. 3.13.

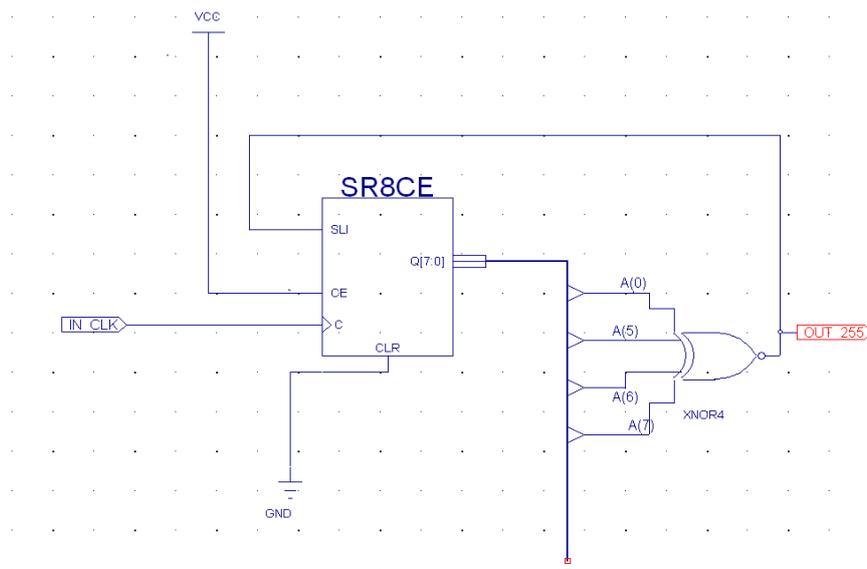


Рис. 3.12. Схема восьмиразрядного генератора псевдослучайной последовательности

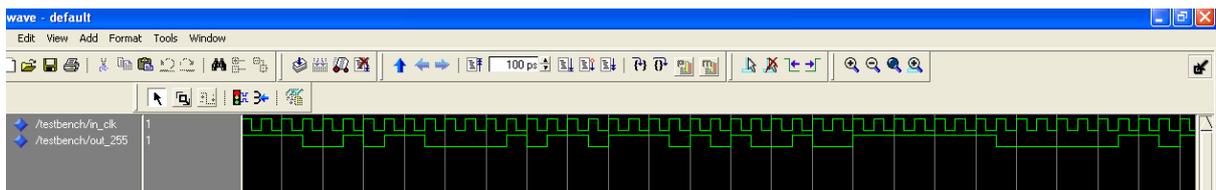


Рис. 3.13. Временные диаграммы генератора псевдослучайной последовательности

3.5. Кодер Хемминга

На рис. 3.14 изображена принципиальная схема кодера Хемминга с параметрами 4/7. Кодирование заключается в вычислении трех избыточных бит в зависимости от четырех информационных бит. Следует отметить, что подобный кодер Хемминга может функционировать в асинхронном режиме.

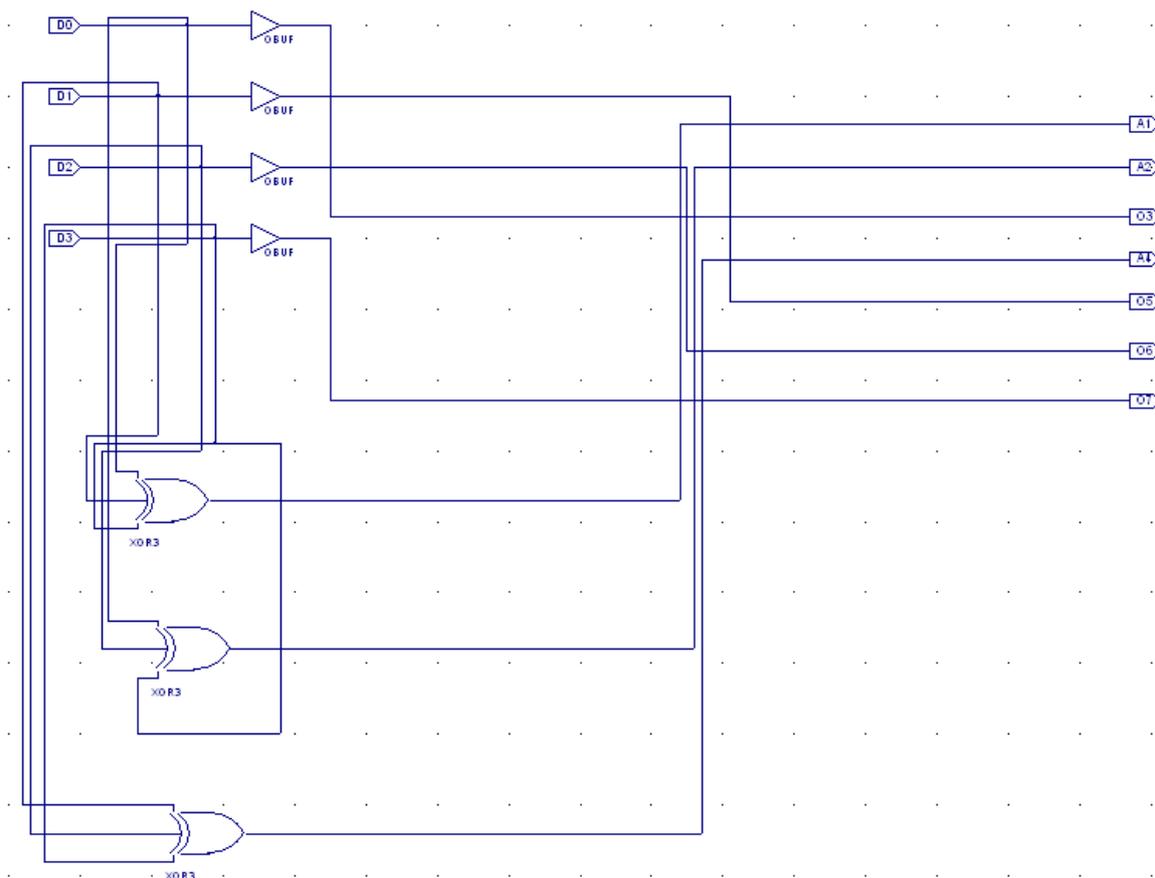


Рис. 3.14. Кодер Хемминга

3.6. Декодер Хемминга

На рис. 3.15 приведена принципиальная схема декодера Хемминга. Алгоритм декодирования заключается в вычислении синдромов и исправлении поврежденного бита. В случае равенства нулю всех синдромов коррекция поврежденного бита не производится. Коррекция по-

врежденного бита осуществляется с помощью восьмиразрядного мультиплексора и управляемых инверторов на логике «Исключающее ИЛИ». Следует отметить возможность асинхронной работы декодера.

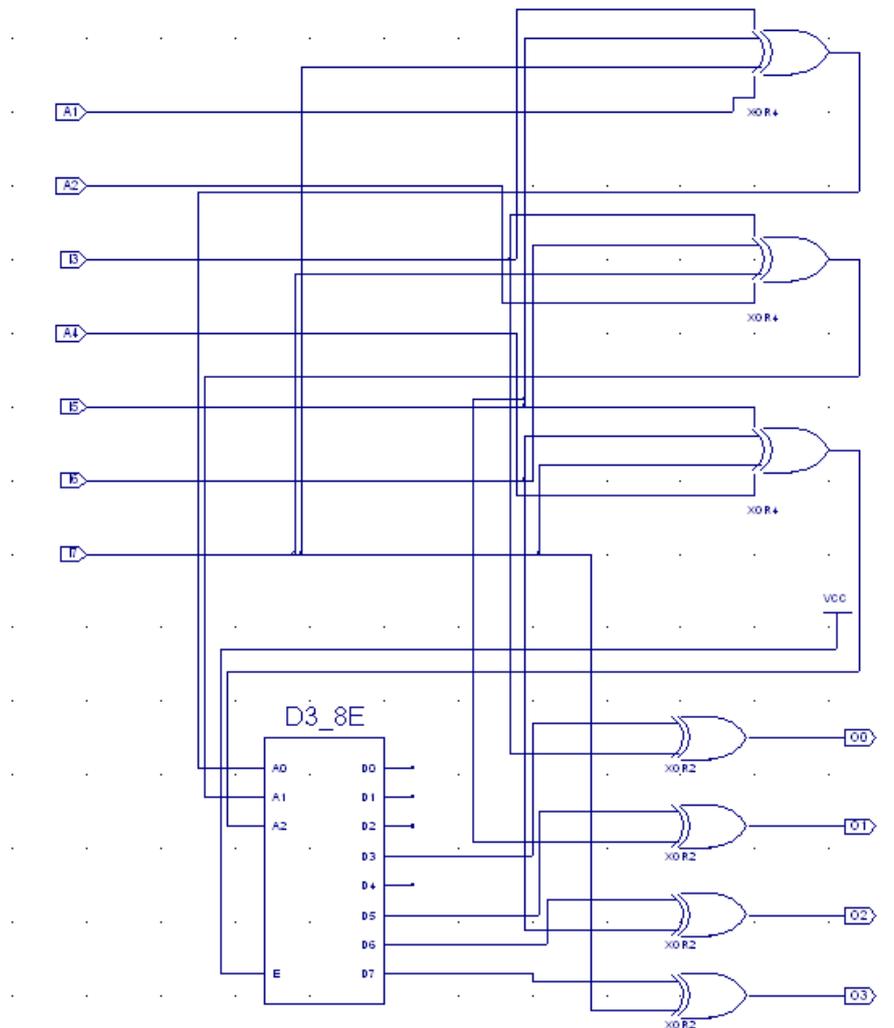


Рис. 3.15. Декодер Хемминга

3.7. Особенности реализации радиотехнических задач на ПЛИС

В настоящее время в области радиотехники и радиосвязи актуальными считаются задачи генерации шумоподобных сигналов, шифрования данных, помехоустойчивого кодирования и декодирования, создания систем с псевдослучайной перестройкой радиочастоты. Подобные задачи можно решить с помощью современных программируемых цифровых устройств, например, с помощью микроконтроллеров и программируемых логических интегральных схем.

Выбор типа программируемого цифрового устройства для решения той или иной задачи обусловлен множеством факторов. В первую очередь это требования к быстродействию обработки цифровой информации и наличию необходимой оперативной памяти для хранения обрабатываемых данных. Современные ПЛИС, способные работать на частотах свыше 300 МГц, предпочтительнее в задачах, требующих высокого быстродействия. Однако зачастую в задачах, требующих большого объема оперативной памяти, для хранения и обработки информации основную роль выполняют микроконтроллеры.

Еще одним фактором при выборе типа программируемого цифрового устройства следует назвать сложность и разветвленность алгоритма реализации разрабатываемого цифрового устройства. Так, при большом числе ветвлений и рекурсий предпочтительнее выглядит выбор микроконтроллера, содержащего микропроцессорное ядро и программируемого на ассемблере, C++ или других языках программирования. Однако в задачах потоковой обработки информации при малом числе ветвлений и рекурсий программируемые логические интегральные схемы выигрывают по сравнению даже с самыми современными микроконтроллерами.

Кроме того, современные ПЛИС по сравнению с микроконтроллерами способны к параллельной обработке информации и даже могут решать несколько не связанных между собой задач внутри одной микросхемы. Это особенно актуально при разработке некоторых помехоустойчивых декодеров, например, сверточного декодера с алгоритмом Витерби, или помехоустойчивых декодеров с «мягкими» алгоритмами декодирования, алгоритмы работы которых можно распараллелить.

Еще одним примером может служить приемник шумоподобных цифровых последовательностей с кодом Голда. Такой приемник должен произвести множество расчетов корреляционных функций с принятой цифровой последовательностью. Применение программируемых логических интегральных схем в этом случае гораздо актуальнее из-за возможностей параллельных вычислений.

В общем случае выбор программируемого устройства для решения той или иной радиотехнической задачи – это творческий процесс, требующий от разработчика обширных и специальных знаний. Кроме

того, необходимо ясно представлять себе алгоритм решения поставленной задачи и подходить к выбору элементной базы с учетом всех возможных факторов.

Пример реализации адаптивного кодека Рида-Соломона на ПЛИС фирмы Xilinx семейства XC9500 приведен на рис. 3.16. На рис. 3.17 показана его структурная схема. Адаптация производится к длине информационного сообщения выбором оптимальной избыточности и последующим кодированием информации.

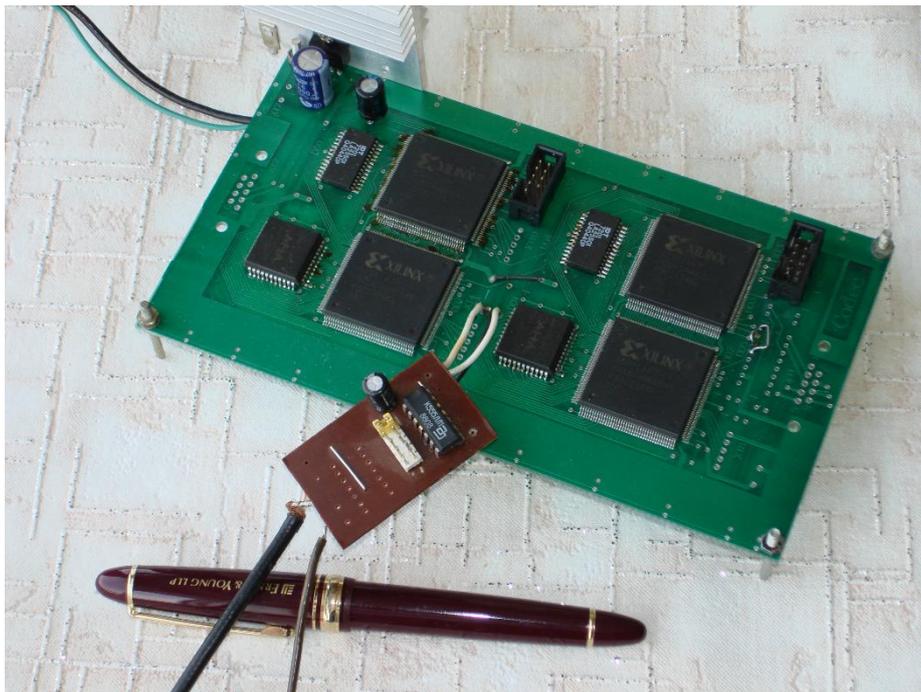


Рис. 3.16. Адаптивный кодек Рида-Соломона на ПЛИС

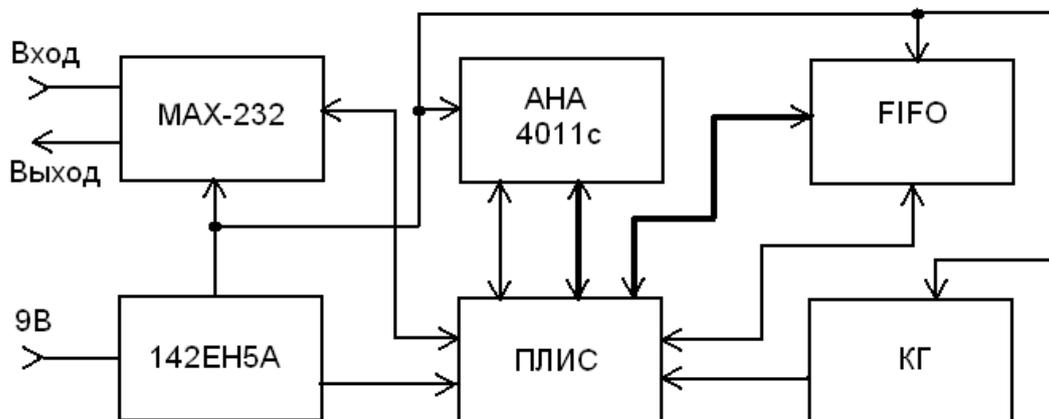


Рис. 3.17. Структурная схема адаптивного кодека Рида-Соломона на ПЛИС

Как видно из структурной схемы, кроме ПЛИС в схеме присутствуют еще микросхемы оперативной памяти (FIFO), специализированные микросхемы кодирования/декодирования по коду Рида-Соломона (АНА4011с), тактовый генератор (КГ) и другие вспомогательные устройства. Такое техническое решение вызвано тем, что алгоритм декодирования Рида-Соломона достаточно сложный, требующий множества итераций и наличия в оперативной памяти таблиц полей Галуа и блока поступившей информации объемом до 255 байт. С другой стороны, подобный кодек имеет возможность обрабатывать непрерывный цифровой поток (блоки информации) с тактовой частотой до нескольких десятков мегагерц. Реализация подобного устройства на микроконтроллере снизила бы число микросхем, в том числе и специализированных, но увеличила бы время адаптивного кодирования/декодирования и снизила бы требования к тактовым частотам обрабатываемого цифрового потока.

3.8. Разработка лабораторного макета на ПЛИС

На сегодняшний день на рынке присутствует множество отладочных плат для ПЛИС, которые могут быть использованы для проведения лабораторных работ и обучения студентов высших и средних учебных заведений. Однако такие платы имеют достаточно высокую стоимость, а с учетом того, что современная цифровая программируемая логика бурно развивается и приобретение дорогостоящего оборудования раз в несколько лет нецелесообразно, актуальной является задача разработки лабораторного макета на ПЛИС в рамках учебно-исследовательской работы студентов.

Следует учитывать еще и фактор различных аварийных ситуаций во время лабораторных работ, что зачастую выводит из строя весь лабораторный комплекс, поскольку в большинстве из них отсутствует возможность замены микросхемы ПЛИС. Исходя из этих соображений можно сформулировать задачу разработки лабораторного комплекса по изучению программирования на ПЛИС следующим образом: заменяемость микросхем ПЛИС и возможность подключения нескольких микросхем ПЛИС, наличие блока оперативной памяти, высокочастотного генератора прямоугольных импульсов и цифровых интерфейсов связи.

Пример структурной схемы подобного отладочного и лабораторного комплекса приведен на рис. 3.18.

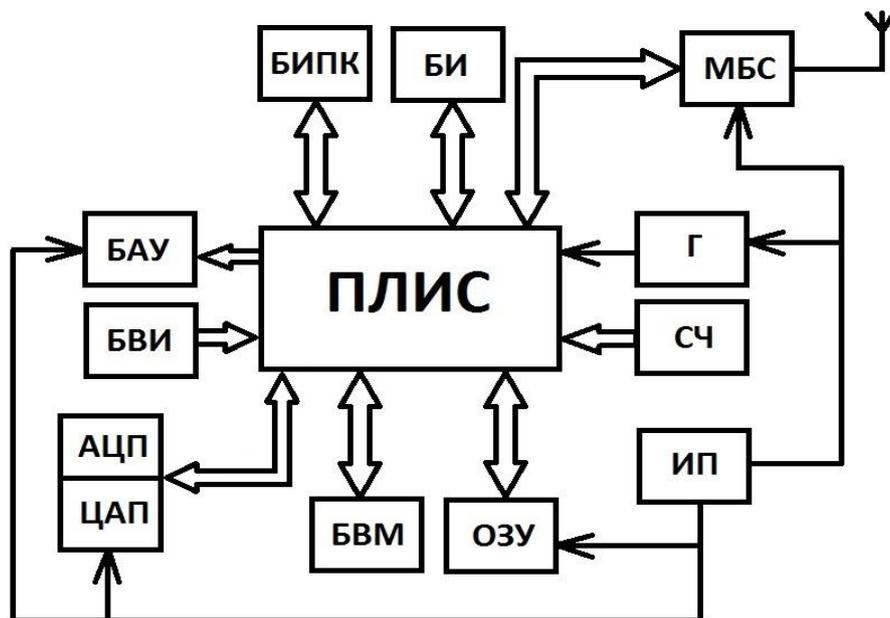


Рис. 3.18. Структурная схема лабораторного комплекса на ПЛИС

Комплекс содержит одну или несколько микросхем ПЛИС с возможностью их замены, высокочастотный генератор прямоугольных импульсов (Г), адресуемое или неадресуемое оперативное запоминающее устройство (ОЗУ), блок подключения внешних модулей в виде разъемов (БВМ), преобразователи аналоговых сигналов в цифровые и обратно (АЦП и ЦАП), блок аналоговых усилителей (БАУ), блок ввода информации в виде переключателей (БВИ), блок индикации в виде линейки светодиодов, в том числе и разноцветных (БИ), блок связи с персональным компьютером (БИПК), модуль беспроводной связи, или трансивер (МБС), и источник стабилизированного питания (ИП).

На рис. 3.19 и 3.20 приведены примеры реализации лабораторного комплекса по изучению программирования ПЛИС, разработанного студентами специальности «Радиотехника» А. М. Даниленко, Е. С. Коных, Н. Н. Колесником под руководством доцента кафедры «Радиотехника и радиосистемы» С. А. Самойлова в рамках учебно-исследовательской работы. Подобные разработки относительно недороги, доступны студентам радиотехнических специальностей и позволяют не только овладеть навыками программирования ПЛИС, но навыками разработки многофункциональных цифровых устройств.

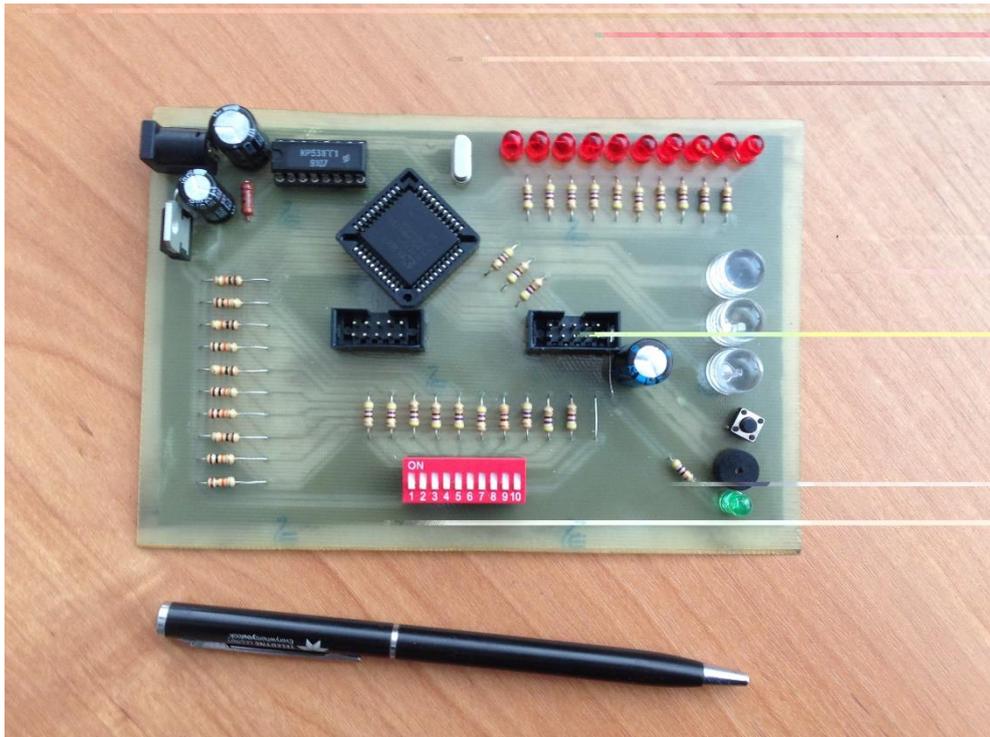


Рис. 3.19. Лабораторный комплекс по изучению программирования ПЛИС

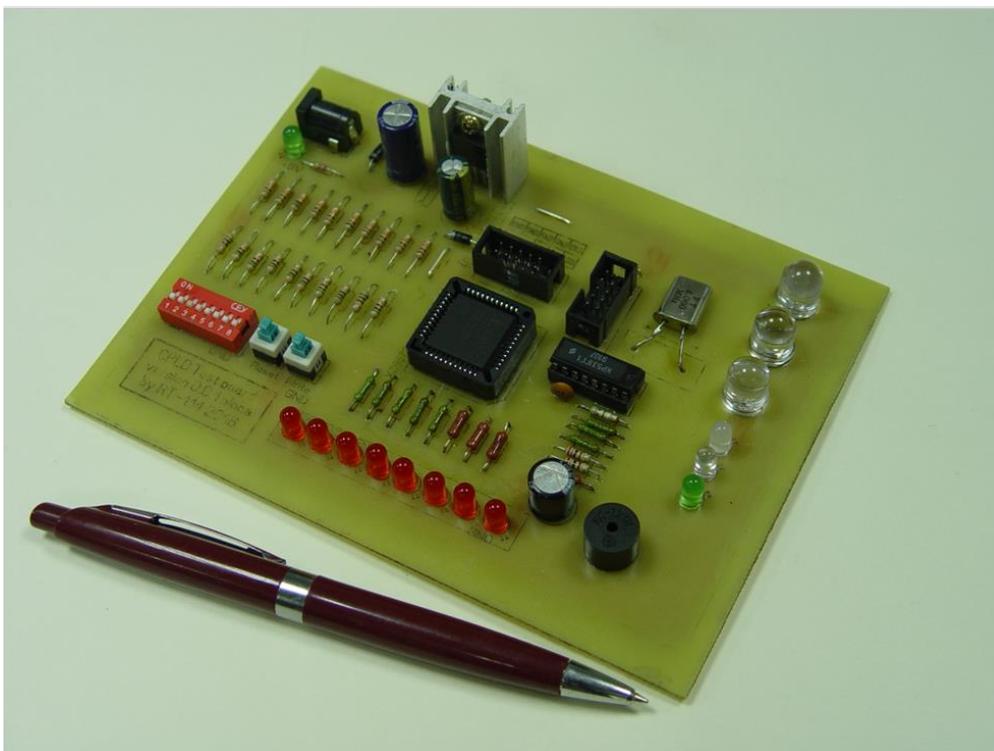


Рис. 3.20. Лабораторный комплекс на ПЛИС

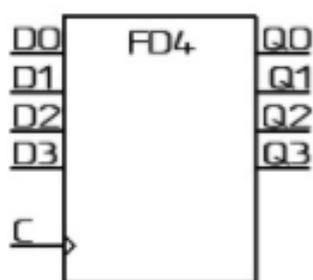
Контрольные вопросы

1. В чем заключается алгоритм преобразования последовательного кода в параллельный?
2. В чем преимущества последовательной передачи данных?
3. Нарисуйте схему сдвигового регистра.
4. Что такое D-триггер?
5. В чем заключается преобразование параллельного кода в последовательный?
6. Нарисуйте схему буфера.
7. Что такое шина данных?
8. Какими параметрами характеризуются шины данных?
9. На чем основан алгоритм сверточного кодирования?
10. Какой разрядности сдвиговый регистр в схеме сверточного кодера?
11. Нарисуйте структурную схему генератора псевдослучайной последовательности?
12. Какой максимальный период повторения псевдослучайной последовательности?
13. Где применяются генераторы псевдослучайных последовательностей?
14. Напишите таблицу истинности четырехходового логического элемента «Исключающее ИЛИ».
15. В чем заключается алгоритм кодирования по Хеммингу?
16. Чем отличаются синхронные и асинхронные схемы?
17. В чем заключается алгоритм декодирования по Хеммингу?
18. Каковы преимущества ПЛИС по сравнению с микроконтроллерами?
19. Объясните, что такое оперативная память FIFO.
20. Расскажите, что такое оперативная память LIFO.
21. Чем отличаются ОЗУ типа FIFO/LIFO от адресуемой оперативной памяти?

Глава 4. УНИФИЦИРОВАННЫЕ БИБЛИОТЕКИ КОМПОНЕНТОВ СХЕМОТЕХНИЧЕСКОГО РЕДАКТОРА ECS

4.1. Триггеры с динамическим управлением

FDD4 – четыре D-триггера, тактируемых фронтом и спадом импульса синхронизации, с общим входом синхронизации (четырёхразрядный регистр)



Назначение выводов:

D0 – D3 – информационные входы D соответствующих триггеров

C – вход синхронизации

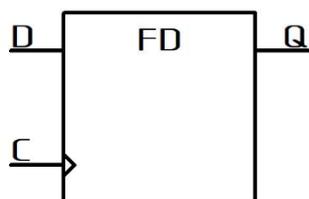
Q0 – Q3 – выходы соответствующих триггеров

Таблица истинности элемента FDD4

Вход C	Вход Di	Выход Qi	Режим работы
/	0	0	Загрузка 0 (сброс) i-го триггера
/	1	1	Загрузка 1 (установка) i-го триггера
\	0	0	Загрузка 0 (сброс) i-го триггера
\	1	1	Загрузка 0 (Установка) i-го триггера
0	X	Qni	Хранение
1	X	Qni	Хранение

Поддерживаемые семейства ПЛИС: XC9500/XV/XL, CoolRunner XPLA3,

CoolRunner-II. Тип элемента – макрос.



FD – D- триггер

Назначение выводов:

D – информационные входы D

C – вход синхронизации

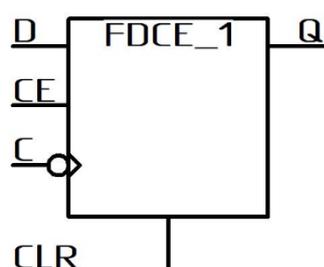
Q – выход

Таблица истинности элемента FD

Вход С	Вход D	Выход Q	Режим работы
/	0	0	Загрузка 0 (сброс)
/	1	1	Загрузка 1 (установка)
0	X	Q _n	Хранение
1	X	Q _n	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – для Spartan-II, Spartan-III, Virtex-E, Virtex, Virtex-II Virtex-II Pro-примитив, для XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II- макрос.

FDC_1 – D-триггер с асинхронным сбросом, тактируемый спадом импульса синхронизации



Назначение выводов:

D – информационный вход D

C – вход синхронизации

CLR – вход сброса

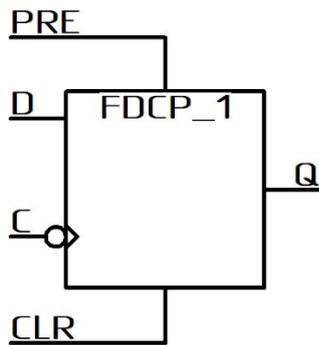
Q – выход

Таблица истинности элемента FDC–1

Вход CLR	Вход С	Вход D	Выход Q	Режим работы
0	\	0	0	Загрузка 0 (сброс)
0	\	1	1	Загрузка 1 (установка)
0	0	X	Q _n	Хранение
0	1	X	Q _n	Хранение
1	X	X	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС Spartan-II, Spartan-III, Virtex-E, Virtex, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

FDCP_1 – D-триггер с асинхронным сбросом и асинхронной установкой, тактируемый спадом импульса синхронизации



Назначение выводов:
 D – информационный вход D
 C – вход синхронизации
 CLR – вход сброса
 PRE – вход установки
 Q – выход

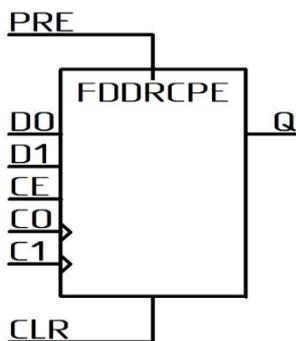
Таблица истинности элемента FDCP_1

Вход CLR	Вход PRE	Вход C	Вход D	Выход Q	Режим работы
0	0	\	0	0	Загрузка 0 (сброс)
0	0	\	1	1	Загрузка 1 (установка)
0	0	0	Qn	Qn	Хранение
0	0	1	Qn	Qn	Хранение
0	1	X	1	1	Асинхронная установка
1	X	X	0	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

FDDRCP_E – D-триггер с асинхронным сбросом, асинхронной установкой, входом разрешения тактового сигнала и двумя входами данных и синхронизации

Триггер FDDRCP_E тактируется двумя противофазными сигналами синхронизации.



Назначение выводов:
 D0, D1 – информационные входы
 C0, C1 – входы синхронизации
 CE – вход разрешения тактового сигнала
 CLR – вход сброса
 PRE – вход установки
 Q – выход

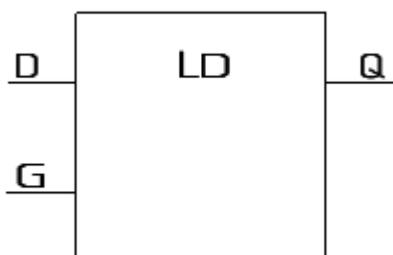
Таблица истинности элемента FDDRCPЕ

Вход CLR	Вход PRE	Вход CE	Вход C0	Вход C1	Вход D0	Вход D1	Выход Q	Режим работы
0	0	1	/	X	0	X	0	Загрузка 0 со входа D0
0	0	1	/	X	1	X	1	Загрузка 1 со входа D0
0	0	1	X	/	X	0	0	Загрузка 0 со входа D1
0	0	1	X	/	X	1	1	Загрузка 1 со входа D1
0	0	1	0	0	X	X	Qn	Хранение
0	0	1	1	0	X	X	Qn	Хранение
0	0	1	0	1	X	X	Qn	Хранение
0	0	1	1	1	X	X	Qn	Хранение
0	0	0	X	X	X	X	Qn	Хранение
0	1	X	X	X	X	X	1	Асинхронная установка
1	X	X	X	X	X	X	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Virtex-II, Virtex-II Pro. Тип элемента – примитив.

4.2. Триггеры с потенциальным (статическим) управлением (защелки)

LD – D-триггер с потенциальным управлением (D-защелка)



Назначение выводов:

D – информационный вход D

G – вход синхронизации

Q – выход

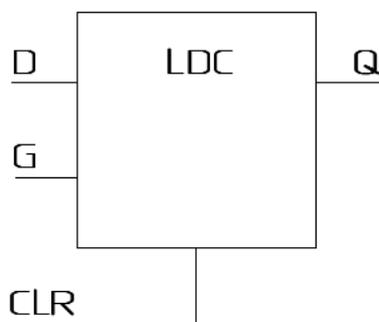
Таблица истинности элемента LD

Вход G	Вход D	Выход Q	Режим работы
1	0	0	Передача данных со входа D (0) на выход триггера
1	1	1	Передача данных со входа D (1) на выход триггера
\	0	0	Фиксация входных данных (0) в триггере
\	1	1	Фиксация входных данных (1) в триггере
0	X	Qn	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента: для Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, CoolRunner XPLA3, CoolRunner-II – примитив; для XC9500/XV/XL – макрос.

LDC – D-триггер с потенциальным управлением (D-защелка) и асинхронным сбросом



Назначение выводов:

D – информационный вход D

G – вход синхронизации

CLR – вход сброса

Q – выход

Таблица истинности элемента LDC

Вход CLR	Вход G	Вход D	Выход Q	Режим работы
0	1	0	0	Передача данных со входа D (0) на выход триггера
0	1	1	1	Передача данных со входа D (1) на выход триггера
0	\	0	0	Фиксация входных данных (0) в триггере

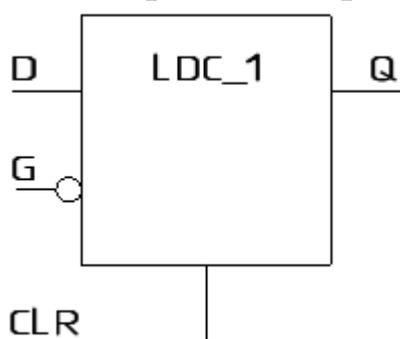
Окончание таблицы

Вход CLR	Вход G	Вход D	Выход Q	Режим работы
0	\	1	1	Фиксация входных данных (1) в триггере
0	0	X	Q _n	Хранение
1	X	X	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента: для Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, CoolRunner XPLA3, CoolRunner-II – примитив; для XC9500/XV/XL – макрос.

LDC_1 – D-триггер с потенциальным управлением (D-защелка), асинхронным сбросом и инверсией по входу синхронизации



Назначение выводов:

D – информационный вход D

G – вход синхронизации

CLR – вход сброса

Q – выход

Таблица истинности элемента LDC_1

Вход CLR	Вход G	Вход D	Выход Q	Режим работы
0	0	0	0	Передача данных со входа D (0) на выход триггера
0	0	1	1	Передача данных со входа D (1) на выход триггера
0	/	0	0	Фиксация входных данных (0) в триггере
0	/	1	1	Фиксация входных данных (1) в триггере
0	1	X	Q _n	Хранение
1	X	X	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

LDE – D-триггер с потенциальным управлением (D-защелка) и входом разрешения тактового сигнала

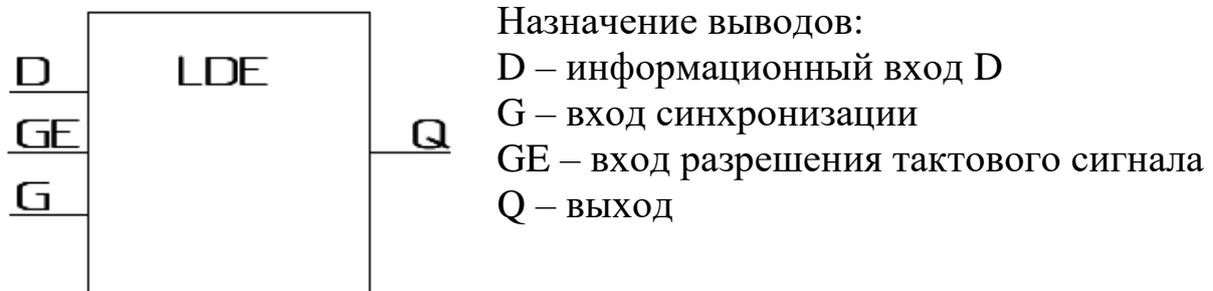


Таблица истинности элемента LDE

Вход GE	Вход G	Вход D	Выход Q	Режим работы
1	1	0	0	Передача данных со входа D (0) на выход триггера
1	1	1	1	Передача данных со входа D (1) на выход триггера
1	\	0	0	Фиксация входных данных (0) в триггере
1	\	1	1	Фиксация входных данных (1) в триггере
1	0	X	Qn	Хранение
0	X	X	Qn	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

LDCP – D-триггер с потенциальным управлением (D-защелка), асинхронным сбросом и асинхронной установкой

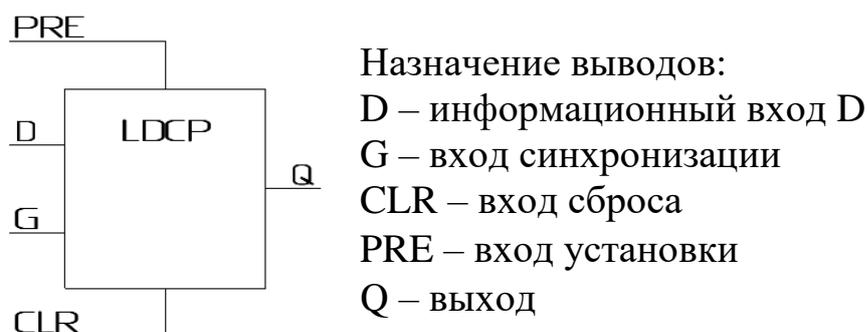


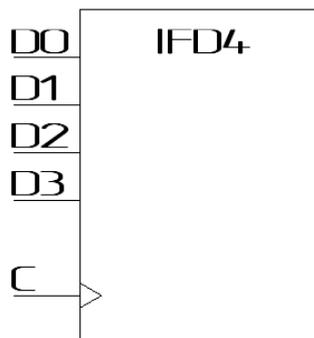
Таблица истинности элемента LDСР

Вход CLR	Вход PRE	Вход G	Вход D	Выход Q	Режим работы
0	0	1	0	0	Передача данных со входа D (0) на выход триггера
0	0	1	1	1	Передача данных со входа D (1) на выход триггера
0	0	\	0	0	Фиксация входных данных (0) в триггере
0	0	\	1	1	Фиксация входных данных (1) в триггере
0	0	0	X	Qn	Хранение
0	1	X	X	1	Асинхронная установка
1	X	X	X	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента: для Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, CoolRunner XPLA3, CoolRunner-II – примитив; для XC9500/XV/XL – макрос.

IFD4 – четыре входных D-триггера с общим входом синхронизации



Назначение выводов:

D0 – D3 – информационные входы D соответствующих триггеров

C – вход синхронизации

Q0 – Q3 – выходы соответствующих триггеров

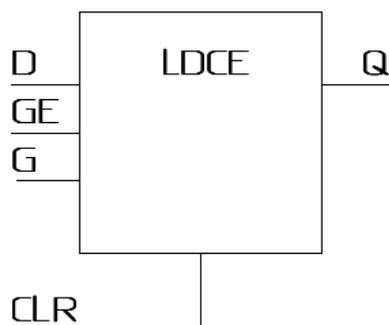
Таблица истинности элемента IFD4

Вход C	Вход Di	Выход Qi	Режим работы
/	0	0	Загрузка 0 (сброс) i-го триггера
/	1	1	Загрузка 1 (сброс) i-го триггера
0	X	Qni	Хранение
1	X	Qni	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

LDCE – D-триггер с потенциальным управлением (D-защелка), входом разрешения тактового сигнала и асинхронным сбросом



Назначение выводов:

D – информационный вход D

G – вход синхронизации

CLR – выход сброса

GE – вход разрешения тактового сигнала

Q – выход

Таблица истинности элемента LDCE

Вход CLR	Вход PRE	Вход G	Вход D	Выход Q	Режим работы
0	1	1	0	0	Передача данных со входа D (0) на выход триггера
0	1	1	1	1	Передача данных со входа D (1) на выход триггера
0	1	\	0	0	Фиксация входных данных (0) в триггере
0	1	\	1	1	Фиксация входных данных (1) в триггере
0	1	0	X	Qn	Хранение
0	0	X	X	1	Асинхронная установка
1	X	X	X	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

LD4CE – четыре входных D-триггера с потенциальным управлением (D-защёлка), асинхронным сбросом, общими входами синхронизации, разрешения тактового сигнала и сброса (четырёхразрядный регистр)

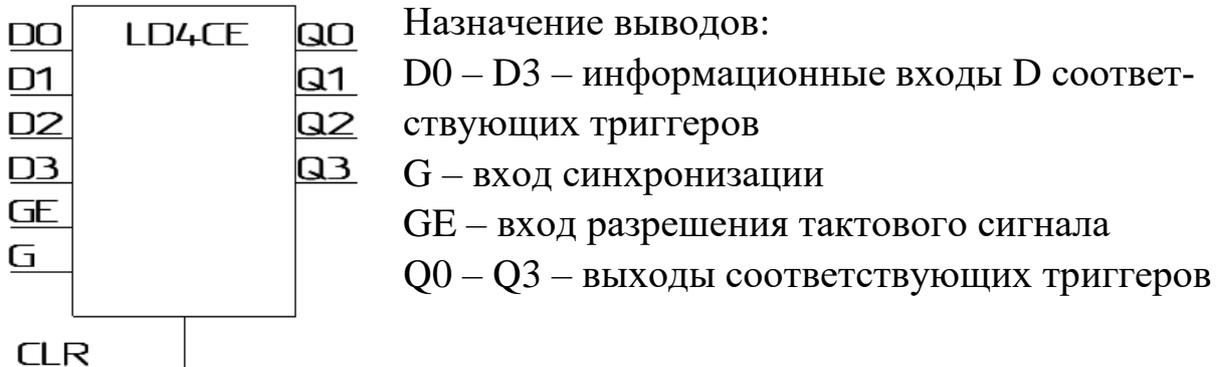
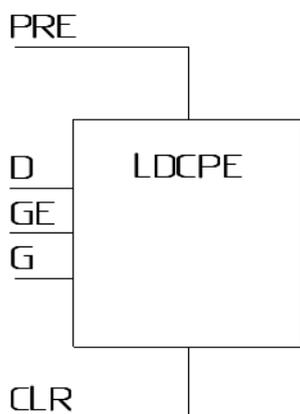


Таблица истинности элемента LD4CE

Вход CLR	Вход GE	Вход G	Вход Di	Вход Qi	Режим работы
0	1	1	0	0	Передача данных со входа Di (0) на выход i-го триггера
0	1	1	1	1	Передача данных со входа Di (1) на выход i-го триггера
0	1	\	0	0	Фиксация входных данных (0) в i-м триггере
0	1	\	1	1	Фиксация входных данных (1) в i-м триггере
0	1	0	X	Qni	Хранение
0	0	X	X	Qni	Хранение
1	X	X	X	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

**LDCPE – D-триггер с потенциальным управлением (D-защелка),
асинхронным сбросом, асинхронной установкой и входом
разрешения тактового сигнала**



Назначение выводов:

D – информационный вход D

G – вход синхронизации

CLR – выход сброса

GE – вход разрешения тактового сигнала

PRE – вход установки

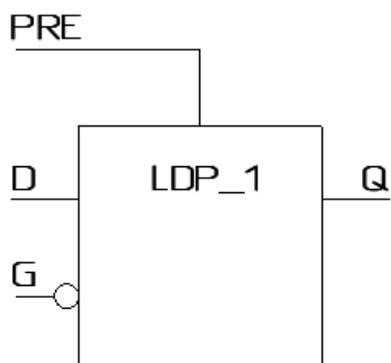
Q – выход

Таблица истинности элемента LDCPE

Вход CLR	Вход PRE	Вход GE	Вход G	Вход D	Выход Q	Режим работы
0	0	1	1	0	0	Передача данных со входа D (0) на выход триггера
0	0	1	1	1	1	Передача данных со входа D (1) на выход триггера
0	0	1	\	0	0	Фиксация входных данных (0) в триггере
Вход CLR	Вход PRE	Вход GE	Вход G	Вход D	Выход Q	Режим работы
0	0	1	\	1	1	Фиксация входных данных (1) в триггере
0	0	1	0	X	Qn	Хранение
0	0	0	X	X	Qn	Хранение
0	1	X	X	X	1	Асинхронная установка
1	X	X	X	X	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

LDP_1 – D-триггер с потенциальным управлением (D-защелка), асинхронной установкой и инверсией по входу синхронизации



Назначение выводов:
 D – информационный вход D
 G – вход синхронизации
 CLR – вход сброса
 PRE – вход установки
 Q – выход

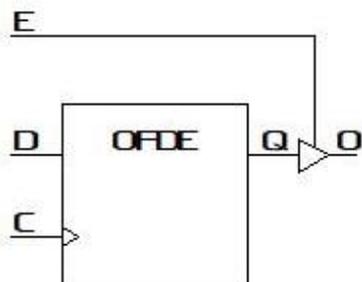
Таблица истинности элемента LDP_1

Вход PRE	Вход G	Вход D	Выход Q	Режим работы
0	0	0	0	Передача данных со входа D (0) на выход триггера
0	0	1	1	Передача данных со входа D (1) на выход триггера
0	/	0	0	Фиксация входных данных (0) в триггере
0	/	1	1	Фиксация входных данных (1) в триггере
0	1	X	Qn	Хранение
1	X	X	1	Асинхронная установка

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

4.3. Входные и выходные триггеры с динамическим управлением

OFDE – выходной D-триггер с тристабильным выходом, управляемым высоким логическим уровнем сигнала разрешения



Назначение выводов:

D – информационный вход D

C – вход синхронизации

E – вход разрешения (переключение выхода в состояние высокого импеданса)

O – выход

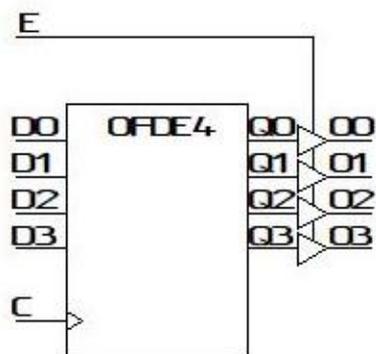
Таблица истинности элемента OFDE

Вход E	Вход C	Вход D	Выход O	Режим работы
1	/	0	0	Загрузка 0 (сброс) и разрешение выхода
1	/	1	1	Загрузка 1 (установка) и разрешение выхода
1	0	X	Qn	Хранение и разрешение выхода
1	1	X	Qn	Хранение и разрешение выхода
0	/	0	Z	Загрузка 0 (сброс) и запрет выхода (состояние «выключено»)
0	/	1	Z	Загрузка 1 (установка) и запрет выхода (состояние «выключено»)
0	0	X	Z	Хранение и запрет выхода (состояние «выключено»)
0	1	X	Z	Хранение и запрет выхода (состояние «выключено»)

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

OFDE4 – четыре выходных D-триггера с тристабильным выходом, управляемым высоким логическим уровнем сигнала разрешения, и общими входами синхронизации и управления



Назначение выводов:

D0 – D3 – информационные входы D соответствующих триггеров

C – вход синхронизации

E – вход разрешения (переключения выхода в состояние высокого импеданса)

O0 – O3 – выходы соответствующих триггеров

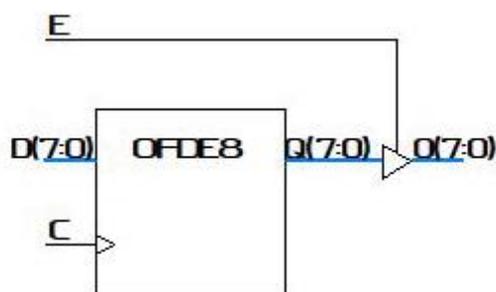
Таблица истинности элемента OFDE4

Вход E	Вход C	Вход Di	Выход Oi	Режим работы
1	/	0	0	Загрузка 0 (сброс) i-го триггера и разрешение выхода
1	/	1	1	Загрузка 1 (установка) i-го триггера и разрешение выхода
1	0	X	Qni	Хранение i-го триггера и разрешение выхода
1	1	X	Qni	Хранение и разрешение выхода
0	/	0	Z	Загрузка 0 (сброс) i-го триггера и запрет выхода (состояние «выключено»)
0	/	1	Z	Загрузка 1 (установка) i-го триггера и запрет выхода (состояние «выключено»)
0	0	X	Z	Хранение i-го триггера и запрет выхода (состояние «выключено»)
0	1	X	Z	Хранение i-го триггера и запрет выхода (состояние «выключено»)

Поддерживаемые семейства ПЛИС: Spartan-11, Spartan-IIЕ, Virtex. Virtex-I, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

OFDE8 – восемь выходных D-триггеров с тристабильным выходом, управляемым высоким логическим уровнем сигнала разрешения, и общими входами синхронизации и управления



Назначение выводов:

D0 – D7 – информационные входы D соответствующих триггеров, объединенные в шину D[7:0]

C – вход синхронизации

E – вход разрешения (переключение выхода в состояние высокого импеданса)

O0 – O7 – выходы соответствующих триггеров, объединенные в шину O[7:0]

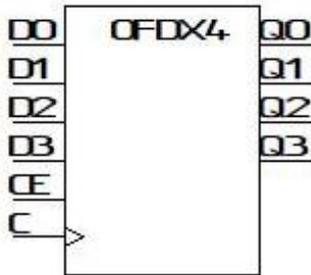
Таблица истинности для элемента OFDE8

Вход E	Вход C	Вход Di	Выход Oi	Режим работы
1	/	0	0	Загрузка 0 (сброс) i-го триггера и разрешение выхода
1	/	1	1	Загрузка 1 (установка) i-го триггера и разрешение выхода
1	0	X	Qni	Хранение i-го триггера и разрешение выхода
1	1	X	Qni	Хранение и разрешение выхода
0	/	0	Z	Загрузка 0 (сброс) i-го триггера и запрет выхода (состояние «выключено»)
0	/	1	Z	Загрузка 1 (установка) i-го триггера и запрет выхода (состояние «выключено»)
0	0	X	Z	Хранение i-го триггера и запрет выхода (состояние «выключено»)
0	1	X	Z	Хранение i-го триггера и запрет выхода (состояние «выключено»)

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

OFDX4 – четыре выходных D-триггера с общими входами синхронизации и разрешения тактового сигнала



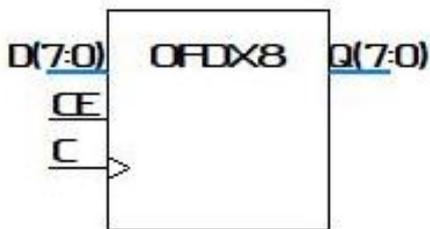
Назначение выводов:
 D0 – D3 – информационные входы D соответствующих триггеров
 C – вход синхронизации
 CE – вход разрешения тактового сигнала
 Q0 – Q3 – выходы соответствующих триггеров

Таблица истинности элемента OFDX4

Вход CE	Вход C	Вход Di	Выход Qi	Режим работы
1	/	0	0	Загрузка 0 (сброс) i-го триггера
1	/	1	1	Загрузка 1 (установка) i-го триггера
1	0	X	Qni	Хранение
1	1	X	Qni	Хранение
0	X	X	Qni	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

OFDX8 – восемь выходных D-триггеров с общими входами синхронизации и разрешения тактового сигнала



Назначение выводов:
 D0 – D7 – информационные входы D соответствующих триггеров, объединенные в шину D[7:0]
 C – вход синхронизации
 CE – вход разрешения тактового сигнала

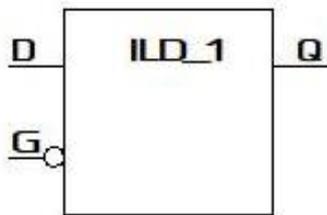
Q0 – Q7 – выходы соответствующих триггеров, объединенные в шину Q[7:0]

Таблица истинности элемента OFDX8

Вход SE	Вход C	Вход Di	Выход Qi	Режим работы
1	/	0	0	Загрузка 0 (сброс) i-го триггера
1	/	1	1	Загрузка 1 (установка) i-го триггера
1	0	X	Qni	Хранение
1	1	X	Qni	Хранение
0	X	X	Qni	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

ILD_1 – входной D-триггер с потенциальным управлением (D-защелка)



Назначение выводов:

D – информационный вход

D, G – вход синхронизации

Q – выход

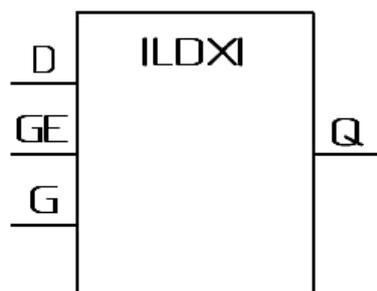
Таблица истинности элемента ILD_1

Вход G	Вход D	Вход Q	Режим работы
0	0	0	Передача данных со входа D (0) на выход триггера
0	1	1	Передача данных со входа D (1) на выход триггера
/	0	0	Фиксация входных данных (0) в триггере
/	1	1	Фиксация входных данных (1) в триггере
1	X	Qn	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

4.4. Входные триггеры с потенциальным (статическим) управлением (защелкой)

ILDXI – входной D-триггер с потенциальным управлением (D-защелка), входом разрешения тактового сигнала и асинхронной установкой при подаче напряжения питания



Назначение выводов:

D – информационный вход D

G – вход синхронизации

GE – вход разрешения тактового сигнала

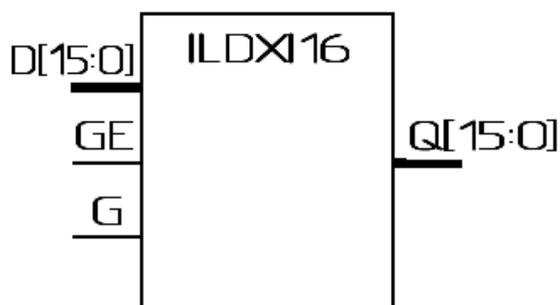
Q – выход

Таблица истинности элементов ILDXI

Вход GE	Вход G	Вход D	Выход Q	Режим работы
1	1	0	0	Передача данных со входа D (0) на выход триггера
1	1	1	1	Передача данных со входа D (1) на выход триггера
1	\	0	0	Фиксация входных данных (0) в триггере
1	\	1	1	Фиксация входных данных (1) в триггере
1	0	X	Qn	Хранение
0	X	X	Qn	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

ILDХ16 – шестнадцать выходных D-триггеров с потенциальным управлением (D-защелок) с общими входами синхронизации и разрешения тактового сигнала



Назначение выводов:

D0 – D15 – информационные входы D соответствующих триггеров, объединенные в шину D[15:0]

G – вход синхронизации

GE – вход разрешения тактового сигнала

Q0 – Q13 – выходы соответствующих триггеров, объединенные в шину Q[15:0]

Таблица истинности элемента ILDXI_1

Вход GE	Вход G	Вход Di	Выход Qi	Режим работы
1	1	0	0	Передача данных со входа Di (0) на выход i-го триггера
1	1	1	1	Передача данных со входа Di (1) на выход i-го триггера
1	/	0	0	Фиксация входных данных (0) в i-м триггере
1	/	1	1	Фиксация входных данных (1) в i-м триггере
1	0	X	Qni	Хранение
0	X	X	Qni	Хранение

Поддержка семейства ПЛИС: Spdrtan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

SR4CE – четырехразрядный сдвиговый регистр со входом разрешения тактового сигнала, асинхронным сбросом, последовательным входом данных и параллельными выходами

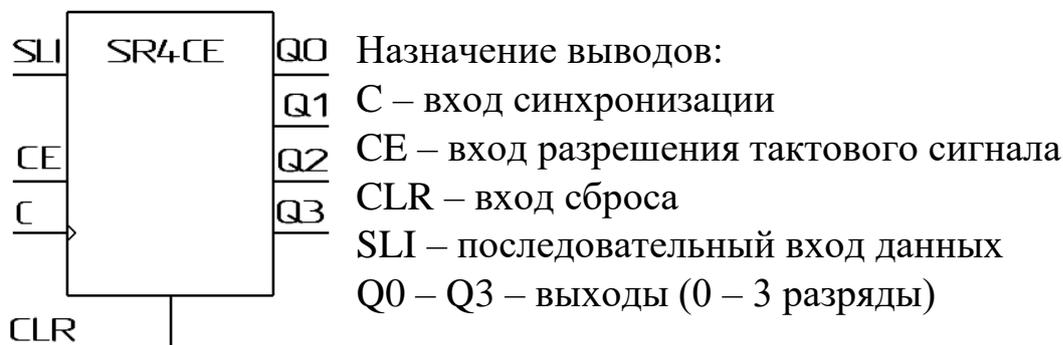


Таблица истинности элементов SR4CE

C	CLR	CE	SLI	Q0	Q1	Q2	Q3	Режим работы
/	0	1	0	0	Q0n	Q1n	Q2n	Сдвиг и запись 0 в разряды Q0
/	0	1	1	1	Q0n	Q1n	Q2n	Сдвиг и запись 0 в разряды Q0
X	0	0	X	Q0n	Q1n	Q2n	Q3n	Хранение
X	1	X	X	0	0	0	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA, CoolRunner-II.

Тип элемента – макрос.

ILD XI_1 – входной D-триггер с потенциальным управлением (D-защелка), входом разрешения тактового сигнала, инверсией по входу синхронизации и асинхронной установкой при подаче напряжения питания

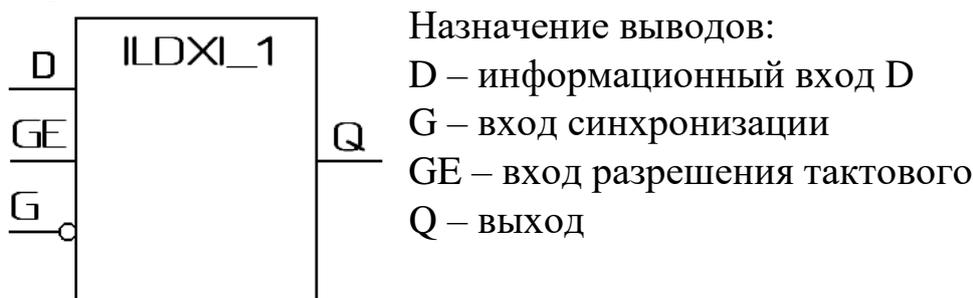
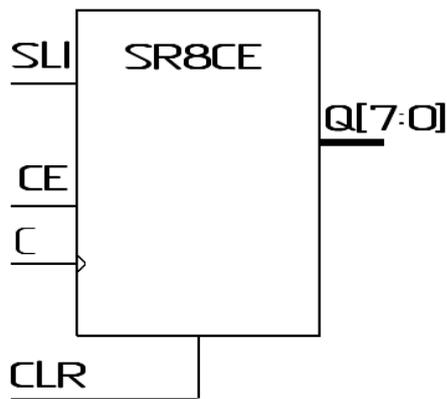


Таблица истинности элемента ILDXI_1

Вход GE	Вход G	Вход D	Выход Q	Режим работы
1	0	0	0	Передача данных со входа D (0) на выход триггера
1	0	1	1	Передача данных со входа D (1) на выход триггера
1	/	0	0	Фиксация входных данных (0) в триггере
1	/	1	1	Фиксация входных данных (1) в триггере
1	1	X	Qn	Хранение
0	X	X	Qn	Хранение

Поддержка семейства ПЛИС: Spdrtan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

SR8CE – восьмиразрядный сдвиговый регистр со входом разрешения тактового сигнала, асинхронным сбросом, последовательным входом данных и параллельными выходами



Назначение выводов:

C – вход синхронизации

CE – вход разрешения тактового сигнала

CLR – вход сброса

SLI – последовательный вход данных

Q0 – Q7 – выходы (0 – 7), объединенные в шину Q[7:0]

Таблица истинности элементов SR8CE

C	CLR	CE	SLI	Q0	Q1	...	Qi	...	Q6	Q7	Режим работы
/	0	1	0	0	Q0n		Q(i-1)n		Q5n	Q6n	Сдвиг и запись 0 в разряд Q0

Окончание таблицы

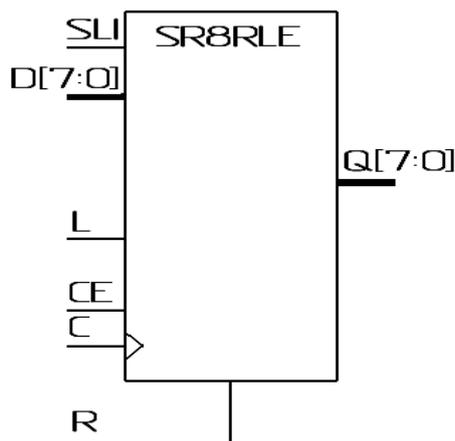
C	CLR	CE	SLI	Q0	Q1	...	Qi	...	Q6	Q7	Режим работы
/	0	1	1	1	Q0n		Q(i-1)n		Q5n	Q6n	Сдвиг и запись 1 в разряд Q0
X	0	0	X	Q0n	Q1n		Qin		Q6n	Q7n	Хранение
X	1	X	X	0	0		0		0	0	Асинхронный сброс

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

4.5. Сдвиговые регистры

SR8RLE – восьмиразрядный сдвиговый регистр со входом разрешения тактового сигнала, синхронным сбросом, последовательным и параллельными входами данных и параллельными выходами



Назначение выводов:

C – вход синхронизации

CE – вход разрешения тактового сигнала

R – вход сброса

SLI – последовательный вход данных

L – вход разрешения параллельной загрузки.

D0 – D7 – параллельные входы данных (0 – 7 разряды), объединенные в шину D[0:7]

Q0 – Q7 – выходы (0 – 7 разряды), объединенные в шину Q[0:7]

Таблица истинности элементов SR8RLE

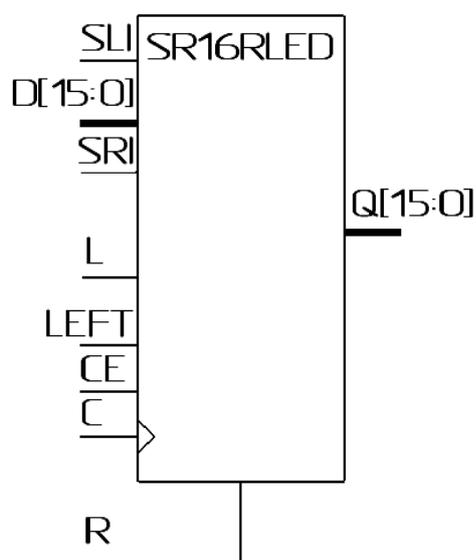
C	R	L	CE	SLI	Q0	Q1	...	Qi	...	Q6	Q7	Режим работы
/	0	0	1	0	0	Q0n		Q(i-1)n		Q5n	Q6n	Сдвиг и запись 0 в разряд Q0
/	0	0	1	1	1	Q0n		Q(i-1)n		Q5n	Q6n	Сдвиг и запись 0 в разряд Q0
/	0	1	X	X	ID0	ID1		Idi		ID6	ID7	Параллельная загрузка
/	1	X	X	X	0	0		0		0	0	Синхронный сброс
X	0	0	0	X	Q0n	Q1n		Qin		Q6n	Q7n	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

SR16RLED – шестнадцатиразрядный реверсный сдвиговый регистр со входом разрешения тактового сигнала, синхронным сбросом, последовательным и параллельными входами данных и параллельными выходами

Назначение выводов:



C – вход синхронизации

CE – вход разрешения тактового сигнала

R – вход сброса

SLI – последовательный вход данных при сдвиге вправо

SRI – последовательный вход данных при сдвиге влево

L – вход разрешения параллельной загрузки

LEFT – вход сигнала направления сдвига

D0 – D15 – параллельные входы данных (0 – 15 разряды), объединенные в шину D[15:0]

Q0 – Q15 – выходы (0 – 15 разряды), объединенные в шину Q[15:0]

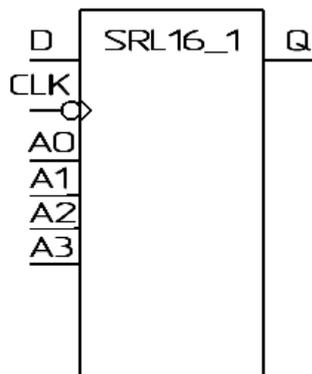
Таблица истинности элементов SR16RLED

C	R	L	LEFT	CE	Q0	Q1	...	Qi	...	Q14	Q15	Режим работы
/	0	0	1	1	SL	Q0n		Q(i-1)n		Q13n	Q14n	Сдвиг вправо и запись в разряд Q0
/	0	0	0	1	Q1n	Q2n		Q(i+1)n		Q15n	SR	Сдвиг влево и запись в разряд Q15
/	0	1	X	X	ID0	ID1		Idi		ID14	ID15	Параллельная загрузка
/	1	X	X	X	0	0		0		0	0	Синхронный сброс
X	0	0	X	0	Q0n	Q1n		Qin		Q14n	Q15n	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

SRL16_1 – шестнадцатиразрядный сдвиговый регистр, тактируемый спадом импульса синхронизации, с последовательным входом и выходом данных, реализуемый на основе таблиц преобразования (LUT) ПЛИС



Назначение выводов:

CLK – вход синхронизации

D – вход данных

Q – выход

A0 – A1 – входы выборы разрядности сдвигового регистра (номера разряда, с которого данные поступают на выход)

Таблица истинности элемента SRL16_1

CLK	A0 – A3	D	Q	Режим работы
\	An	ID	Q(An-1)	Сдвиг вправо и запись данных со входа D в первый разряд регистра (Q0)
0	X	X	QAn	Хранение
1	X	X	QAn	Хранение

ID – значение сигнала на входе данных D.

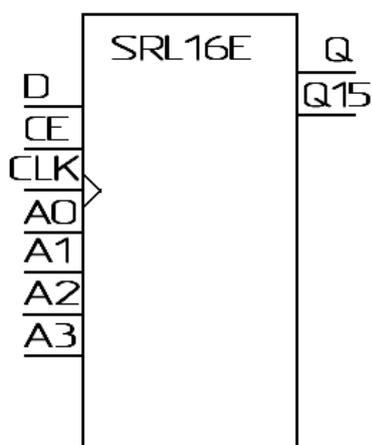
An – значение, соответствующее сочетанию сигналов на входах A0 – A3.

QAn – состояние An-го разряда регистра на предыдущем шаге.

Q(An-1) – состояние разряда (An-1)-го регистра на предыдущем шаге.

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

SRL16E – шестнадцатиразрядный каскадируемый сдвиговый регистр, тактируемый фронтом импульса синхронизации, со входом разрешения тактового сигнала, последовательным входом и выходом данных, реализуемый на основе таблиц преобразования (LUT) ПЛИС



Назначение выводов:

CLK – вход синхронизации

CE – вход разрешения тактового сигнала

D – вход данных

Q – выход

Q15 – выход, предназначенный для каскадного сдвигового регистра

A0 – A3 – выходы выбора разрядности сдвигового регистра (номера разряда, с которого данные поступают на выход)

Таблица истинности элемента SRL16E

CLK	CE	A0 – A3	D	Q	Q15	Режим работы
/	1	An	ID	Q(An-1)	Q14n	Сдвиг вправо и запись данных со входа D в первый разряд регистра (Q0)
0	1	X	X	QAn	Q15n	Хранение
1	1	X	X	QAn	Q15n	Хранение
X	0	X	X	QAn	Q15n	Хранение

ID – значение сигнала на входе данных D.

An – значение, соответствующее сочетанию сигналов на входах A0 – A3.

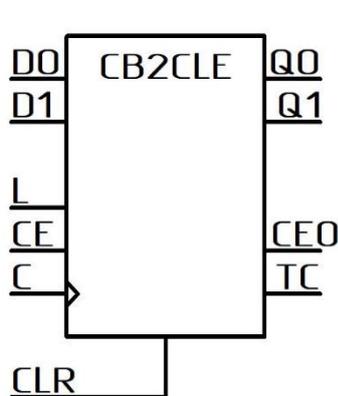
QAn – состояние An-го разряда регистра на предыдущем шаге.

Q(An-1) – состояние разряда (An-1)-го регистра на предыдущем шаге.

Поддерживаемые семейства ПЛИС: Virtex-II, Virtex-II Pro. Тип элемента – примитив.

4.6. Счетчики

CB2CLE – двухразрядный двоичный счетчик с параллельной синхронной загрузкой, асинхронным сбросом и входом разрешения счетчика



Назначение выводов:

C – тактовый вход

CE – вход разрешения счета

L – вход разрешения параллельной загрузки

CLR – вход сброса

D0 – D1 – параллельные входы данных (0 – 1 разряды)

Q0 – Q1 – выходы (0 – 1 разряды)

TC – выход переноса (выход завершения счета)

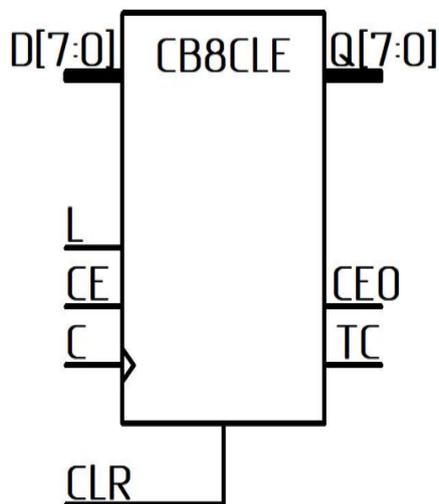
CEO – выходы разрешения счета (CEO = TC & CE)

Таблица истинности элемента CB2CLE

С	CLR	L	CE	QI	TC	CEO	Режим работы
X	1	X	X	0	0	0	Асинхронный сброс
/	0	1	X	DI	0	0	Параллельная загрузка (D[1:0] # 11)
/	0	1	X	DI	1	ICE	Параллельная загрузка (D[1:0] = 11)
/	0	0	1	Счет	0	0	Счет (Q[1:0] # 11)
/	0	0	1	Счет	1	1	Счет (Q[1:0] # 11)
X	0	0	0	QIn	TCn	0	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

CB8CLE – восьмиразрядный двоичный счетчик с параллельной синхронной загрузкой, асинхронным сбросом и входом разрешения счетчика



Назначение выводов:

- С – тактовый вход
- CE – вход разрешения счета
- L – вход разрешения параллельной загрузки
- CLR – вход сброса
- D0 – D7 – параллельные входы данных (0 – 7 разряды), объединенные в шину D[7:0]
- Q0 – Q7 – выходы (0 – 7 разряды), объединенные в шину Q[7:0]

TC – выход переноса (выход завершения счета)

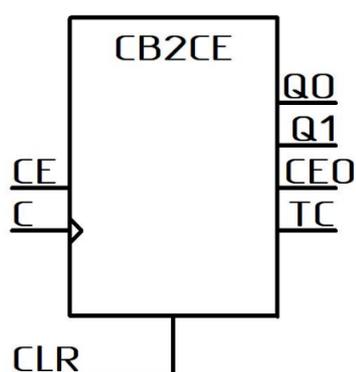
CEO – выходы разрешения счета (CEO = TC & CE)

Таблица истинности элемента CB8CLE

C	CLR	L	CE	QI	TC	CEO	Режим работы
X	1	X	X	0	0	0	Асинхронный сброс
/	0	1	X	DI	0	0	Параллельная загрузка (D[7:0] # FF)
/	0	1	X	DI	1	ICE	Параллельная загрузка (D[7:0] = FF)
/	0	0	1	Счет	0	0	Счет (Q[7:0] # FF)
/	0	0	1	Счет	1	1	Счет (Q[7:0] = FF)
X	0	0	0	QIn	TCn	0	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan- IIE, Virtex, Virtex-E, Virtex- II, Virtex- II Pro XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

CB2CE – двухразрядный двоичный счетчик с асинхронным сбросом и входом разрешения счетчика



Назначение выводов:

C – тактовый вход

CE – вход разрешения счета

CLR – вход сброса

Q0 – Q1 – выходы (0 – 1 разряды)

TC – выход переноса (выход завершения счета)

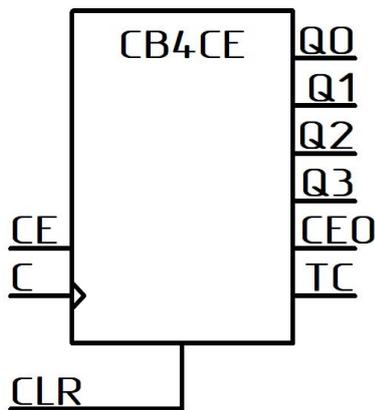
CEO – выходы разрешения счета (CEO = TC & CE)

Таблица истинности элемента CB2CE

C	CLR	CE	QI	TC	CEO	Режим работы
X	1	X	0	0	0	Асинхронный сброс
/	0	1	Счет	0	0	Счет (Q[1:0] # 11)
/	0	1	Счет	1	1	Счет (Q[1:0] = 11)
X	0	0	QIn	TCn	0	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIE, Virtex, Virtex-E, Virtex-II, Virtex-II Pro XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

CB4CE – четырехразрядный двоичный счетчик с асинхронным сбросом и входом разрешения счетчика



Назначение выводов:

C – тактовый вход

CE – вход разрешения счета

CLR – вход сброса

Q0 – Q3 – выходы (0-3 разряды)

TC – выход переноса (выход завершения счета)

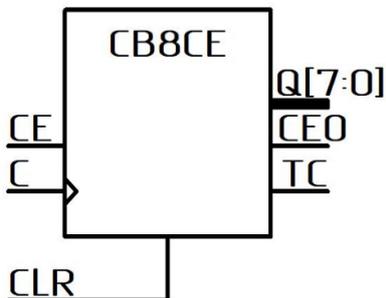
CEO – выходы разрешения счета (CEO = TC & CE)

Таблица истинности элемента CB4CE

C	CLR	CE	QI	TC	CEO	Режим работы
X	1	X	0	0	0	Асинхронный сброс
/	0	1	Счет	0	0	Счет (Q[3:0] # 1111)
/	0	1	Счет	1	1	Счет (Q[3:0] = 1111)
X	0	0	QIn	TCn	0	Хранение

Поддерживаемые семейства ПЛИС: Spartan- II, Spartan- IIE, Virtex, Virtex-E, Virtex- II, Virtex- II Pro XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

CB8CE – восьмиразрядный двоичный счетчик с асинхронным сбросом и входом разрешения счетчика



Назначение выводов:

C – тактовый вход

CE – вход разрешения счета

CLR – вход сброса

Q0 – Q7 – выходы (0 – 7 разряды), объединенные в шину Q[7:0]

TC – выход переноса (выход завершения счета)

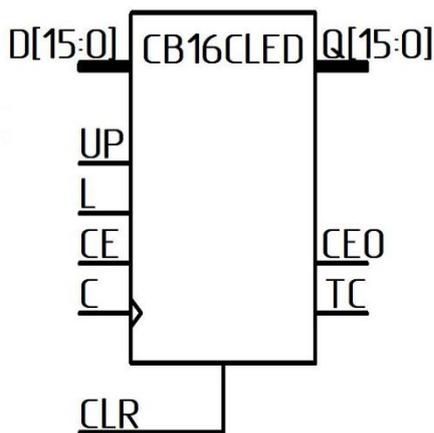
CEO – выходы разрешения счета (CEO = TC & CE)

Таблица истинности элемента CB8CE

C	CLR	CE	QI	TC	CEO	Режим работы
X	1	X	0	0	0	Асинхронный сброс
/	0	1	Счет	0	0	Счет (Q[7:0] # FF)
/	0	1	Счет	1	1	Счет (Q[7:0] = FF)
X	0	0	QIn	TCn	0	Хранение

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

CB16CLED – шестнадцатиразрядный двоичный реверсивный счетчик с параллельной синхронной загрузкой, асинхронным сбросом и входом разрешения счетчика



Назначение выводов:

C – тактовый вход

CE – вход разрешения счета

L – вход разрешения параллельной загрузки

CLR – вход сброса

UP – вход изменения направления счета

D0 – D15 – параллельные входы данных (0 – 15 разряды), объединенные в шину D[15:0].

Q0 – Q15 – выходы (0 – 15 разряды), объединенные в шину Q[15:0]

TC – выход переноса (выход завершения счета)

CEO – выходы разрешения счета (CEO = TC & CE)

Таблица истинности элемента CB16CLED

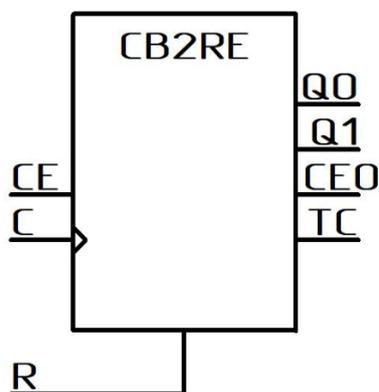
C	CLR	L	CE	UP	QI	TC	CEO	Режим работы
X	1	X	X	X	0	0	0	Асинхронный сброс
/	0	1	X	1	DI	0	0	Параллельная загрузка (D[15:0] # FFFF)
/	0	1	X	1	DI	1	ICE	Параллельная загрузка (D[15:0] = FFFF)

Окончание таблицы

C	CLR	L	CE	UP	QI	TC	CEO	Режим работы
/	0	1	X	0	DI	0	0	Параллельная загрузка (D[15:0] # 0000)
/	0	1	X	0	DI	1	ICE	Параллельная загрузка (D[15:0] = 0000)
/	0	0	1	1	Счет	0	0	Счет на увеличение (Q [15:0] # FFFF)
/	0	0	1	1	Счет	1	1	Счет на увеличение (Q [15:0] = = FFFF)
C	CLR	L	CE	UP	QI	TC	CEO	Режим работы
/	0	0	1	0	Счет	0	0	Счет на увеличение (Q [15:0] # 0000)
/	0	0	1	0	Счет	1	1	Счет на увеличение (Q [15:0] = = 0000)
X	0	0	0	0	Счет	TCn	0	Хранение

Поддерживаемые семейства ПЛИС: Spartan- II, Spartan- IIE, Virtex, Virtex-E, Virtex- II, Virtex- II Pro XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

CB2RE – двухразрядный двоичный счетчик с синхронным сбросом и входом разрешения счетчика



Назначение выводов:

C – тактовый вход

CE – вход разрешения счета

R – вход сброса

Q0 – Q1 – выходы (0 – 1 разряды)

TC – выход переноса (выход завершения счета)

CEO – выходы разрешения счета (CEO = TC & CE)

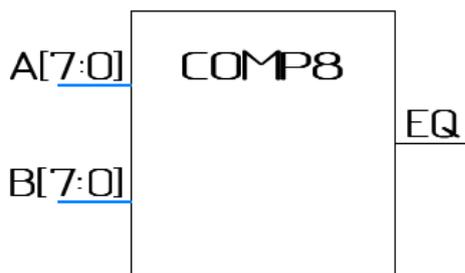
Таблица истинности элемента СВ2RE

C	R	CE	QI	TC	CEO	Режим работы
/	1	X	0	0	0	Синхронный сброс
/	0	1	Счет	0	0	Счет (Q[1:0] # 11)
/	0	1	Счет	1	1	Счет (Q[1:0] = 11)
X	0	0	QIn	TCn	0	Хранение

Поддерживаемые семейства ПЛИС: Spartan- II, Spartan- IIE, Virtex, Virtex-E, Virtex-II, Virtex-II Pro XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

4.7. Компараторы

COMP8 – цифровой восьмиразрядный компаратор



Назначение выводов:

A0 – A7 – входы первого операнда (разряды 0 – 7), объединённые в шину A[7:0]

B0 – B7 – входы второго операнда (разряды 0 – 7), объединённые в шину B[7:0]

EQ – выход A = B

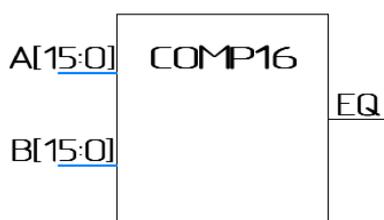
Таблица истинности элемента COMP8

Соотношение значений сигналов на входах A0 – A7, B0 – B7	Выход EQ
IA0 = IB0 и IA2 = IB2 и IA3 = IB3 и IA4 = IB4 и IA5 = IB5 и IA6 = IB6 и IA7 = IB7	1
IA0 # IB0 или IA2 # IB2 или IA3 # IB3 или IA4 # IB4 или IA5 # IB5 или IA6 # IB6 или IA7 # IB7	0

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIE, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

COMP16 – цифровой шестнадцатиразрядный компаратор



Назначение выводов:

A0 – A15 – входы первого операнда (разряды 0 – 15), объединённые в шину A[15:0]

B0 – B15 – входы второго операнда (разряды 0 – 15), объединённые в шину B[15:0]

EQ – выход A = B

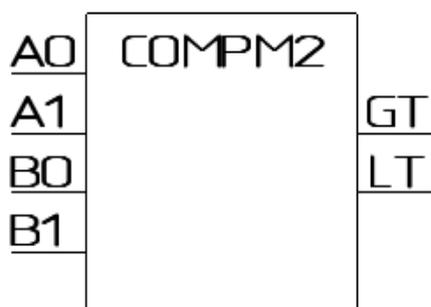
Таблица истинности элемента COMP16

Соотношение значений сигналов на входах A0 – A7, B0 – B7	Выход EQ
IA0 = IB0 и IA2 = IB2 и IA3 = IB3 и IA4 = IB4 и IA5 = IB5 и IA6 = IB6 и IA7 = IB7 и IA8 = IB8 и IA9 = IB9 и IA10 = IB10 и IA11 = IB11 и IA12 = IB12 и IA13 = IB13 и IA14 = IB14 и IA15 = IB15	1
IA0 # IB0 или IA2 # IB2 или IA3 # IB3 или IA4 # IB4 или IA5 # IB5 или IA6 # IB6 или IA7 # IB7 или IA8 # IB8 или IA9 # IB9 или IA10 # IB10 или IA11 # IB11 или IA12 # IB12 или IA13 # IB13 или IA14 # IB14 или IA15 # IB15	0

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

COMP2 – цифровая двухразрядная схема сравнения двух операндов



Назначение выводов:

A0 – A1 – входы первого операнда (разряды 0 и 1)

B0 – B1 – входы второго операнда (разряды 0 и 1)

LT – выход A < B

GT – выход A > B

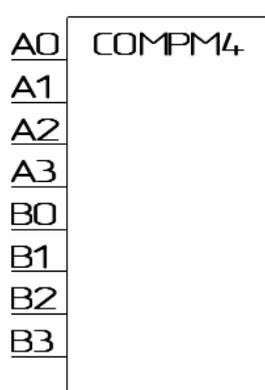
Таблица истинности элемента COMPM2

Соотношение значений сигналов на входах A0 – A1, B0 – B1	Выход LT	Выход GT
$IA0 = IB0$ и $IA1 = IB1$	0	0
$IA0 > IB0$ и $IA1 = IB1$	0	1
$IA0 < IB0$ и $IA1 = IB1$	1	0
$IA1 < IB1$	1	0
$IA1 > IB1$	0	1

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

COMPM4 – цифровая четырехразрядная схема сравнения двух операндов



Назначение выводов:

A0 – A3 – входы первого операнда (разряды 0 и 3)

B0 – B3 – входы второго операнда (разряды 0 и 3)

GT LT – выход $A < B$

LT GT – выход $A > B$

Таблица истинности элемента COMPM4

Соотношение значений сигналов на входах A0 – A3, B0 – B3	Выход LT	Выход GT
$IA0 = IB0; \dots; IA_i = IB_i; \dots; IA3 = IB3$	0	0
$IA0 > IB0; IA1 = IB1 \dots; IA_i = IB_i; \dots; IA3 = IB3$	0	1
$IA0 < IB0; IA1 = IB1 \dots; IA_i = IB_i; \dots; IA3 = IB3$	1	0
...		
$IA_i > IB_i; IA_{i+1} = IB_{i+1}; \dots; IA3 = IB3$	0	1
$IA_i < IB_i; IA_{i+1} = IB_{i+1}; \dots; IA3 = IB3$	1	0
...		
$IA3 > IB3$	0	1
$IA3 < IB3$	1	0

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

COMP8 – цифровая восьмиразрядная схема сравнения двух операндов



Назначение выводов:

A0 – A7 – входы первого операнда (разряды 0 и 7)

B0 – B7 – входы второго операнда (разряды 0 и 7)

LT – выход A < B

GT – выход A > B

Таблица истинности элемента COMP8

Соотношение значений сигналов на входах A0 – A7, B0 – B7	Выход LT	Выход GT
$IA_0 = IB_0; \dots; IA_i = IB_i; \dots; IA_7 = IB_7$	0	0
$IA_0 > IB_0; IA_1 = IB_1; \dots; IA_i = IB_i; \dots; IA_7 = IB_7$	0	1
$IA_0 < IB_0; IA_1 = IB_1; \dots; IA_i = IB_i; \dots; IA_7 = IB_7$	1	0
...		
$IA_i > IB_i; IA_{i+1} = IB_{i+1}; \dots; IA_7 = IB_7$	0	1
$IA_i < IB_i; IA_{i+1} = IB_{i+1}; \dots; IA_7 = IB_7$	1	0
...		
$IA_7 > IB_7$	0	1
$IA_7 < IB_7$	1	0

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

COMPM16 – цифровая шестнадцатиразрядная схема сравнения двух операндов



Назначение выводов:

A0 – A15 – входы первого операнда (разряды 0 и 15)

B0 – B15 – входы второго операнда (разряды 0 и 15).

LT – выход A < B

GT – выход A > B

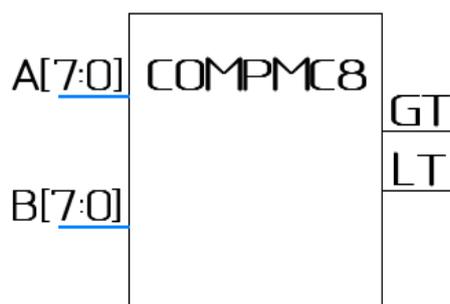
Таблица истинности элемента COMPM16

Соотношение значений сигналов на входах A0 – A15, B0 – B15	Выход LT	Выход GT
$IA_0 = IB_0; \dots; IA_i = IB_i; \dots; IA_{15} = IB_{15}$	0	0
$IA_0 > IB_0; IA_1 = IB_1; \dots; IA_i = IB_i; \dots; IA_{15} = IB_{15}$	0	1
$IA_0 < IB_0; IA_1 = IB_1; \dots; IA_i = IB_i; \dots; IA_{15} = IB_{15}$	1	0
...		
$IA_i > IB_i; IA_{i+1} = IB_{i+1}; \dots; IA_{15} = IB_{15}$	0	1
$IA_i < IB_i; IA_{i+1} = IB_{i+1}; \dots; IA_{15} = IB_{15}$	1	0
...		
$IA_{15} > IB_{15}$	0	1
$IA_{15} < IB_{15}$	1	0

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

COMPM8 – цифровая восьмиразрядная схема сравнения двух операндов, выполненная с использованием логики ускоренного переноса



Назначение выводов:

A0 – A7 – входы первого операнда (разряды 0 и 7), объединенные в шину A[7:0]

B0 – B7 – входы второго операнда (разряды 0 и 7), объединенные в шину B[7:0]

LT – выход A < B

GT – выход A > B

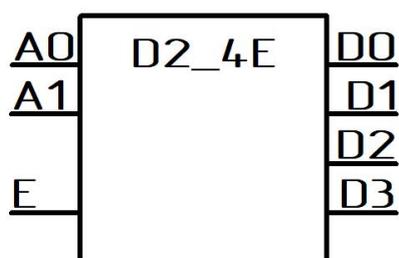
Таблица истинности элемента COMPM8

Соотношение значений сигналов на входах A0 – A7, B0 – B7	Выход LT	Выход GT
IA0 = IB0;...; IAi=IBi;...; IA7 = IB7	0	0
IA0 > IB0; IA1 = IB1...; IAi = IBi;...; IA7 = IB7	0	1
IA0 < IB0; IA1 = IB1...; IAi = IBi;...; IA7 = IB7	1	0
...		
IAi > IBi; IAi+1 = IBi+1;...; IA7 = IB7	0	1
IAi < IBi; IAi+1 = IBi+1;...; IA7 = IB7	1	0
...		
IA7 > IB7	0	1
IA7 < IB7	1	0

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – макрос.

4.8. Дешифраторы

D2_4E – дешифратор с 2 по 4 со входом разрешения, преобразующий двухразрядный двоичный код, поступающий на входы A0 A1, в напряжение высокого уровня, появляющееся на одном из четырех входов D0 – D3, при наличии сигнала высокого уровня на входе E



Назначение выводов:

E – вход разрешения

A0 – A1 – входы (разряды 0 и 1)

D0 – D3 – выходы (0 – 3 разряды)

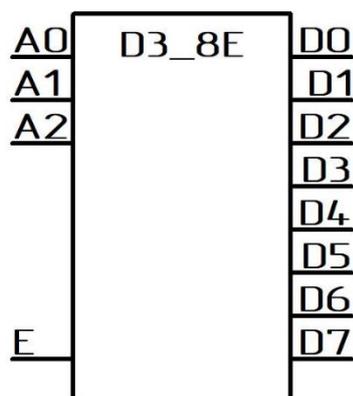
Таблица истинности элемента D2_4E

E	A0	A1	D0	D1	D2	D3	Режим работы
1	0	0	1	0	0	0	Преобразование двоичного кода в напряжение высокого уровня на соответствующем уровне
1	1	0	0	1	0	0	
1	0	1	0	0	1	0	
1	1	1	0	0	0	1	
0	X	X	0	0	0	0	Стробирование

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

D3_8E – дешифратор с 3 по 8 со входом разрешения, преобразующий трехразрядный двоичный код, поступающий на входы A0 – A2, в напряжение высокого уровня, появляющееся на одном из восьми входов D0 – D7, при наличии сигнала высокого уровня на входе E



Назначение выводов:

E – вход разрешения

A0 – A2 – входы (разряды 0 – 2)

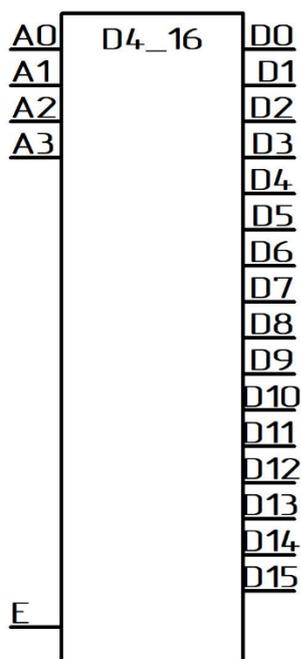
D0 – D7 – выходы (0 – 7 разряды)

Таблица истинности элемента D3_8E

E	A0	A1	A2	D0	D1	D2	D3	D4	D5	D6	D7	Режим работы
1	0	0	0	1	0	0	0	0	0	0	0	Преобразование двоичного кода в напряжение высокого уровня на соответствующем уровне
1	1	0	0	0	1	0	0	0	0	0	0	
1	0	1	0	0	0	1	0	0	0	0	0	
1	1	1	0	0	0	0	1	0	0	0	0	
1	0	0	1	0	0	0	0	1	0	0	0	
1	1	0	1	0	0	0	0	0	1	0	0	
1	0	1	1	0	0	0	0	0	0	1	0	
1	1	1	1	0	0	0	0	0	0	0	1	
0	X	X	X	0	0	0	0	0	0	0	0	Стробирование

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.



D4_16E – дешифратор с 4 по 16 со входом разрешения, преобразующий четырехразрядный двоичный код, поступающий на входы **A0 – A3**, в напряжение высокого уровня, появляющееся на одном из восьми входов **D0 – D15**, при наличии сигнала высокого уровня на входе **E**

Назначение выводов:

E – вход разрешения

A0 – A3 – входы (разряды 0 – 3)

D0 – D15 – выходы (0 – 15 разряды)

Таблица истинности элемента D4_16E

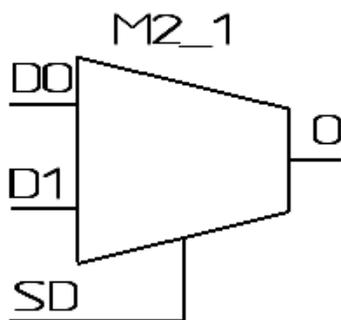
E	A0	A1	A2	A3	D0	D1	D2	...	D12	D13	D14	D15	Режим работы
1	0	0	0	0	1	0	0		0	0	0	0	Преобразование двоичного кода в напряжение высокого уровня на соответствующем уровне
1	1	0	0	0	0	1	0		0	0	0	0	
1	0	1	0	0	0	0	1		0	0	0	0	
1	1	1	0	0	0	0	0		0	0	0	0	
1	0	0	1	0	0	0	0		0	0	0	0	
1	1	0	1	0	0	0	0		0	0	0	0	
1	0	1	1	0	0	0	0		0	0	0	0	
1	1	1	1	0	0	0	0		0	0	0	0	
1	0	0	0	1	0	0	0		0	0	0	0	
1	1	0	0	1	0	0	0		0	0	0	0	
1	0	1	0	1	0	0	0		0	0	0	0	
1	1	1	0	1	0	0	0		0	0	0	0	
1	0	0	1	1	0	0	0		1	0	0	0	
1	1	0	1	1	0	0	0		0	1	0	0	
1	0	1	1	1	0	0	0		0	0	1	0	
1	1	1	1	1	0	0	0		0	0	0	1	
0	X	X	X	X	0	0	0		0	0	0	0	Стробирование

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

4.9. Мультиплексоры

M2_1 двухвходовый мультиплексор (2 в 1)



Назначение выводов:

D0, D1 – информационный входы

S0 – вход выбора

O – выход

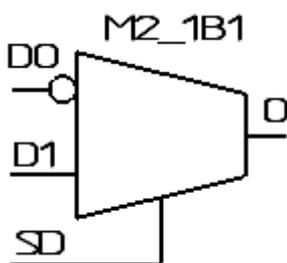
Таблица истинности элемента M2_1

S0	O	Режим работы
0	ID0	Передача данных со входа D0 на выход
1	ID1	Передача данных со входа D1 на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

M2_1B1 – двухвходовый мультиплексор (2 в 1) с инверсией по одному входу



Назначение выводов:

D0, D1 – информационные входы

S0 – вход выбора

O – выход

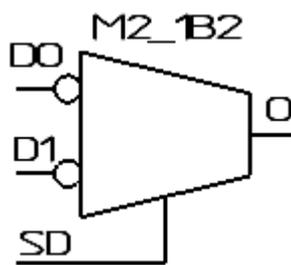
Таблица истинности элемента M2_IB1

S0	O	Режим работы
0	$\sim ID0$	Передача данных со входа D0 на выход
1	ID1	Передача данных со входа D1 на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

M2_IB2 – двухвходовый мультиплексор (2 в 1) с инверсией по входам



Назначение выводов:

- D0, D1 – информационные входы
- S0 – вход выбора
- O – выход

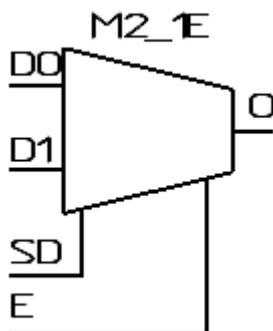
Таблица истинности элемента M2_IB2

S0	O	Режим работы
0	$\sim ID0$	Передача данных со входа D0 на выход
1	$\sim ID1$	Передача данных со входа D1 на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

M2_1E – двухвходовый мультиплексор (2 в 1) со стробированием



Назначение выводов:

- D0, D1 – информационные входы
- S0 – вход выбора
- E – вход разрешения
- O – выход

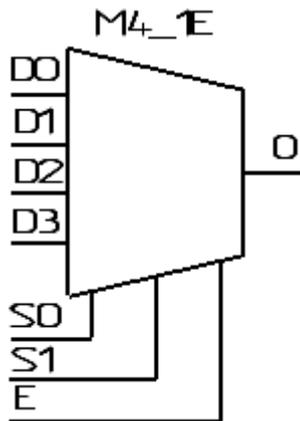
Таблица истинности элемента M2_1E

E	S0	O	Режим работы
1	0	ID0	Передача данных со входа D0 на выход
1	1	ID1	Передача данных со входа D1 на выход
0	X	0	Стробирование

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

**M4_1E – четырехходовый мультиплексор (4 в 1)
со стробированием**



Назначение выводов:

- D0 – D3 – информационные входы
- S0, S1 – входы выбора
- E – вход разрешения
- O – выход

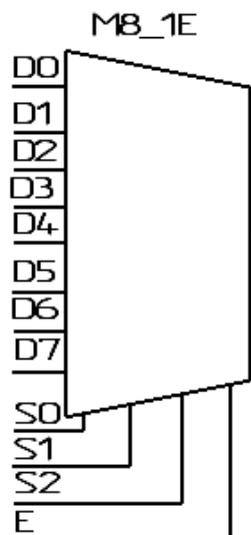
Таблица истинности элемента M4_1E

E	S0	S1	O	Режим работы
1	0	0	ID0	Передача данных со входа D0 на выход
1	1	0	ID1	Передача данных со входа D1 на выход
1	0	1	ID2	Передача данных со входа D2 на выход
1	1	1	ID3	Передача данных со входа D3 на выход
0	X	X	0	Стробирование

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

M8_1E – восьмивходовый мультиплексор (8 в 1) со стробированием



Назначение выводов:

D0 – D7 – информационные входы

S0 – S2 – входы выбора

E – вход разрешения

O – выход

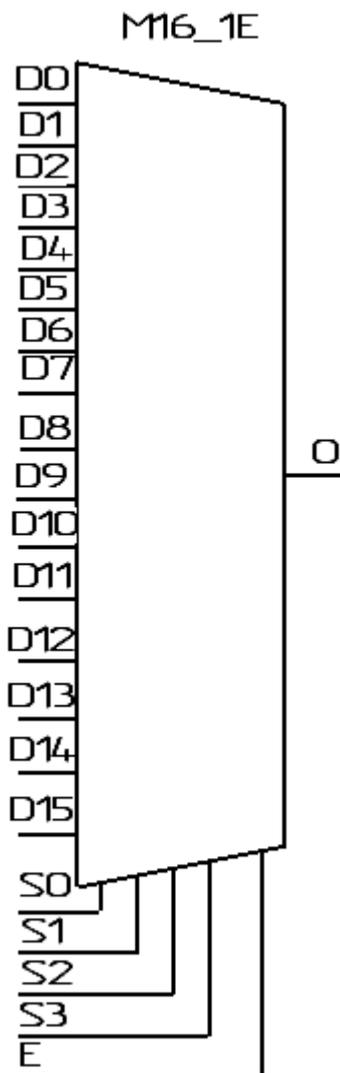
Таблица истинности элемента M8_1E

E	S0	S1	S2	O	Режим работы
1	0	0	0	ID0	Передача данных со входа D0 на выход
1	1	0	0	ID1	Передача данных со входа D1 на выход
1	0	1	0	ID2	Передача данных со входа D2 на выход
1	1	1	0	ID3	Передача данных со входа D3 на выход
1	0	0	1	ID4	Передача данных со входа D4 на выход
1	1	0	1	ID5	Передача данных со входа D5 на выход
1	0	1	1	ID6	Передача данных со входа D6 на выход
1	1	1	1	ID7	Передача данных со входа D7 на выход
0	X	X	X	0	Стробирование

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

**M16_1E – шестнадцативходовый мультиплексор (16 в 1)
со стробированием**



Назначение выводов:

D0 – D15 – информационные входы

S0 – S3 – входы выбора

E – вход разрешения

O – выход

Таблица истинности элемента M16_1E

E	S0	S1	S2	S3	O	Режим работы
1	0	0	0	0	ID0	Передача данных со входа D0 на выход
1	1	0	0	0	ID1	Передача данных со входа D1 на выход
1	0	1	0	0	ID2	Передача данных со входа D2 на выход
1	0	0	1	0	ID3	Передача данных со входа D3 на выход

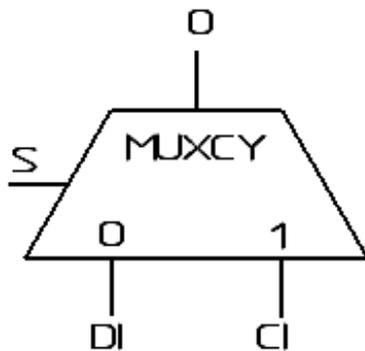
Окончание таблицы

E	S0	S1	S2	S3	O	Режим работы
1	1	0	1	0	ID4	Передача данных со входа D4 на выход
1	0	1	1	0	ID5	Передача данных со входа D5 на выход
1	1	1	1	0	ID6	Передача данных со входа D6 на выход
1	0	0	0	1	ID7	Передача данных со входа D7 на выход
1	1	0	0	1	ID8	Передача данных со входа D8 на выход
1	0	1	0	1	ID9	Передача данных со входа D9 на выход
1	1	1	0	1	ID10	Передача данных со входа D10 на выход
1	0	0	1	1	ID11	Передача данных со входа D11 на выход
1	1	0	1	1	ID12	Передача данных со входа D12 на выход
1	1	0	1	1	ID13	Передача данных со входа D13 на выход
1	0	1	1	1	ID14	Передача данных со входа D14 на выход
1	1	1	1	1	ID15	Передача данных со входа D15 на выход
0	X	X	X	X	0	Стробирование

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макрос.

MUXCY – двухвходовый мультиплексор (2 в 1), реализуемый на базе логики ускоренного переноса



Назначение выводов:

DI, CI – информационные входы

S – вход выбора

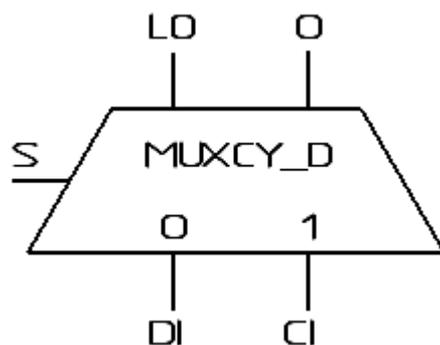
O – основной выход

Таблица истинности элемента MUXCY

S	DI	CI	O	Режим работы
0	0	X	0	Передача данных (0) со входа DI на выход
0	1	X	1	Передача данных (1) со входа DI на выход
1	X	0	0	Передача данных (0) со входа CI на выход
1	X	1	1	Передача данных (1) со входа CI на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

MUXCY_D – двухвходовый мультиплексор (2 в 1), реализуемый на базе логики ускоренного переноса, с локальным выходом



Назначение выводов:

DI, CI – информационные входы

S – вход выбора

O – основной выход

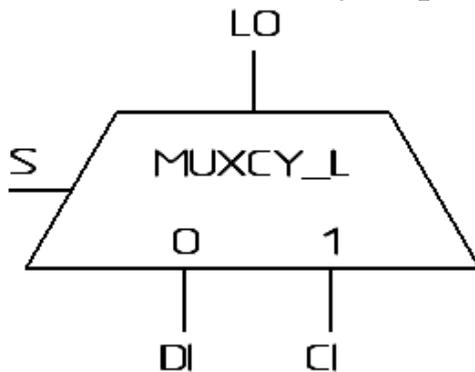
LO – дополнительный (локальный) выход

Таблица истинности элемента MUXCY_D

S	DI	CI	O	LO	Режим работы
0	0	X	0	0	Передача данных (0) со входа DI на выход
0	1	X	1	1	Передача данных (1) со входа DI на выход
1	X	0	0	0	Передача данных (0) со входа CI на выход
1	X	1	1	1	Передача данных (1) со входа CI на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

MUXCY_L – двухвходовый мультиплексор (2 в 1), реализуемый на базе логики ускоренного переноса, с локальным выходом



Назначение выводов:

DI, CI – информационные входы

S – вход выбора

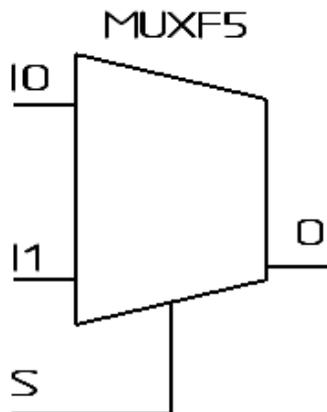
LO – выход

Таблица истинности элемента MUXCY_L

S	DI	CI	LO	Режим работы
0	0	X	0	Передача данных (0) со входа DI на выход
0	1	X	1	Передача данных (1) со входа DI на выход
1	X	0	0	Передача данных (0) со входа CI на выход
1	X	1	1	Передача данных (1) со входа CI на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

MUXF5 – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F5 конфигурируемого логического блока



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

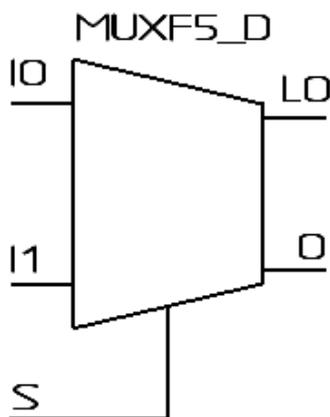
O – выход

Таблица истинности элемента MUXF5

S	I0	S	I0	I1	O	Режим работы
0	0	0	0	X	0	Передача данных (0) со входа I0 на выход
0	1	0	1	X	1	Передача данных (1) со входа I0 на выход
1	X	1	X	0	0	Передача данных (0) со входа I1 на выход
1	X	1	X	1	1	Передача данных (1) со входа I1 на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

MUXF5_D – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F5 дополнительной логики конфигурируемого логического блока, с двумя выходами



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

O – основной выход

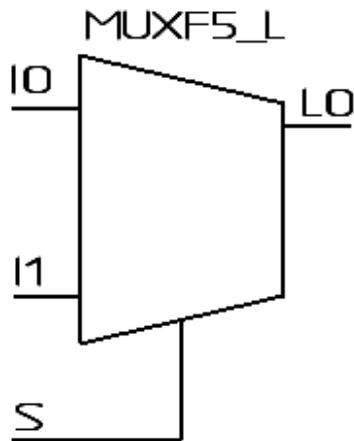
LO – дополнительный (локальный) выход

Таблица истинности элемента MUXF5_D

S	I0	S	I0	I1	O	LO	Режим работы
0	0	0	0	X	0	0	Передача данных (0) со входа I0 на выходы
0	1	0	1	X	1	1	Передача данных (1) со входа I0 на выходы
1	X	1	X	0	0	0	Передача данных (0) со входа I1 на выходы
1	X	1	X	1	1	1	Передача данных (1) со входа I1 на выходы

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

MUXF5_L – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F5 дополнительной логики конфигурируемого логического блока, с локальным выходом



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

O – основной выход

LO – локальный выход

Таблица истинности элемента MUXF5_L

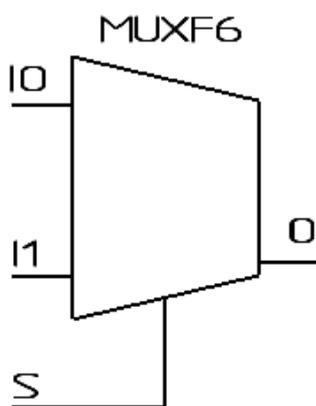
S	I0	S	I0	I1	LO	Режим работы
0	0	0	0	X	0	Передача данных (0) со входа I0 на выход
0	1	0	1	X	1	Передача данных (1) со входа I0 на выход

Окончание таблицы

S	I0	S	I0	I1	LO	Режим работы
1	X	1	X	0	0	Передача данных (0) со входа I1 на выход
1	X	1	X	1	1	Передача данных (1) со входа I1 на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IE, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

MUXF6 – двухходовый мультиплексор (2 в 1), реализуемый на базе элемента F6 дополнительной логики конфигурируемого логического блока



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

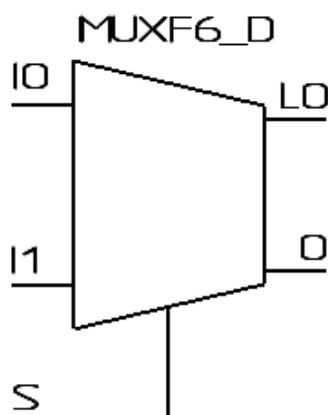
O – выход

Таблица истинности элемента MUXF6

S	I0	S	I0	I1	O	Режим работы
0	0	0	0	X	0	Передача данных (0) со входа I0 на выход
0	1	0	1	X	1	Передача данных (1) со входа I0 на выход
1	X	1	X	0	0	Передача данных (0) со входа I1 на выход
1	X	1	X	1	1	Передача данных (1) со входа I1 на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-IE, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

MUXF6_D – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F6 дополнительной логики конфигурируемого логического блока, с двумя выходами



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

O – основной выход

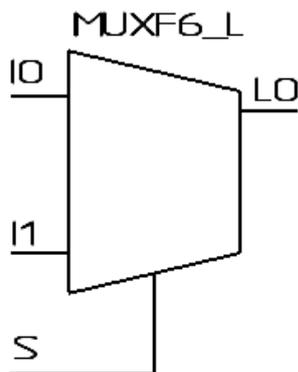
LO – дополнительный (локальный) выход

Таблица истинности элемента MUXF6_D

S	I0	S	I0	I1	O	LO	Режим работы
0	0	0	0	X	0	0	Передача данных (0) со входа I0 на выходы
0	1	0	1	X	1	1	Передача данных (1) со входа I0 на выходы
1	X	1	X	0	0	0	Передача данных (0) со входа I1 на выходы
1	X	1	X	1	1	1	Передача данных (1) со входа I1 на выходы

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro. Тип элемента – примитив.

MUXF6_L – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F6 дополнительной логики конфигурируемого логического блока, с локальным выходом



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

O – основной выход

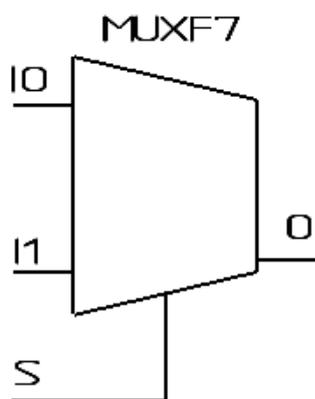
LO – локальный выход

Таблица истинности элемента MUXF6_L

S	I0	S	I0	I1	LO	Режим работы
0	0	0	0	X	0	Передача данных (0) со входа I0 на выход
0	1	0	1	X	1	Передача данных (1) со входа I0 на выход
1	X	1	X	0	0	Передача данных (0) со входа I1 на выход
1	X	1	X	1	1	Передача данных (1) со входа I1 на выход

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II. Virtex-II Pro. Тип элемента – примитив.

MUXF7 – двухходовый мультиплексор (2 в 1), реализуемый на базе элемента F7 дополнительной логики конфигурируемого логического блока



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

O – выход

Таблица истинности элемента MUXF7

S	I0	S	I0	I1	O	Режим работы
0	0	0	0	X	0	Передача данных (0) со входа I0 на выход
0	1	0	1	X	1	Передача данных (1) со входа I0 на выход
1	X	1	X	0	0	Передача данных (0) со входа I1 на выход
1	X	1	X	1	1	Передача данных (1) со входа I1 на выход

Поддерживаемые семейства ПЛИС: Virtex-II, Virtex-II Pro.

Тип элемента – примитив.

MUXF7_D – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F7 дополнительной логики конфигурируемого логического блока, с двумя выходами

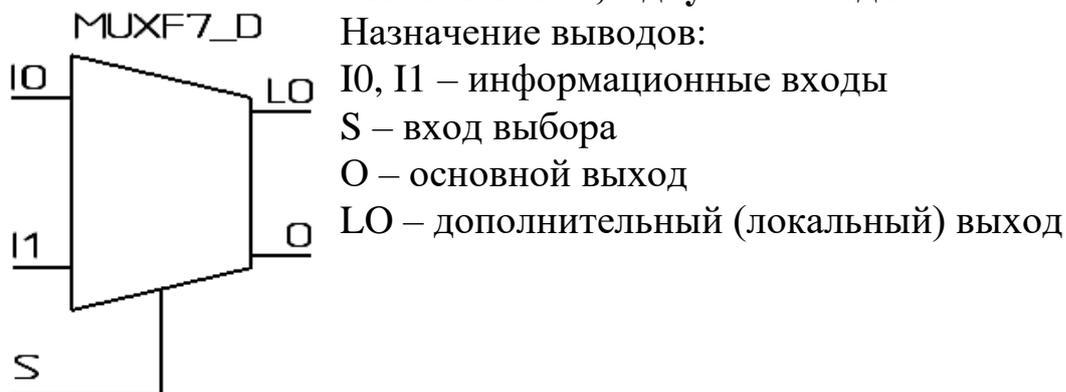


Таблица истинности элемента MUXF7_D

S	I0	I1	LO	Режим работы
0	0	X	0	Передача данных (0) со входа I0 на выходы
0	1	X	1	Передача данных (1) со входа I0 на выходы
1	X	0	0	Передача данных (0) со входа I1 на выходы
1	X	1	1	Передача данных (1) со входа I1 на выходы

Поддерживаемые семейства ПЛИС: Virtex-II, Virtex-II P.

Тип элемента – примитив.

MUXF7_L – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F7 дополнительной логики конфигурируемого логического блока, с локальным выходом

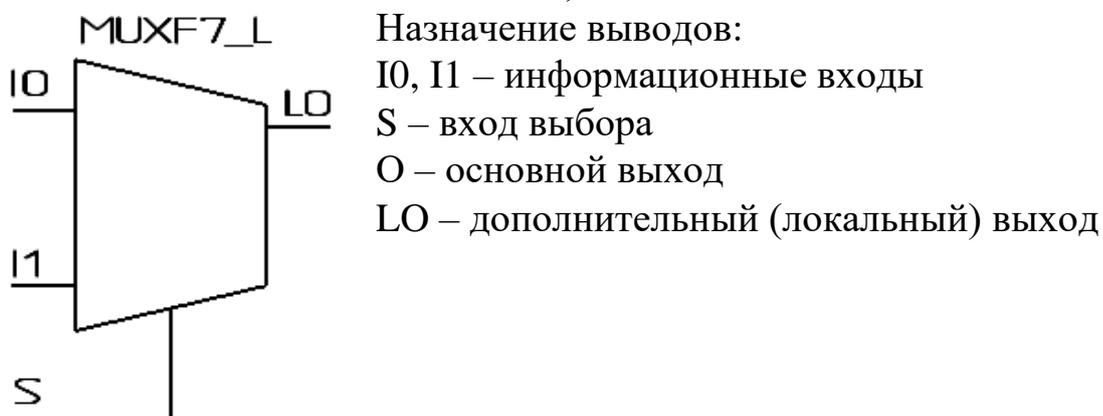


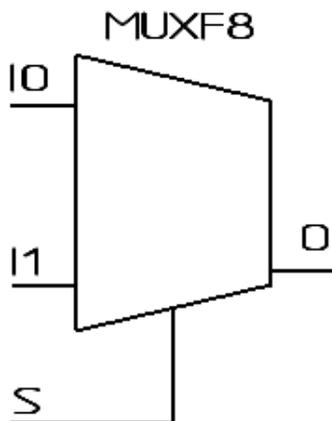
Таблица истинности элемента MUXF7_L

S	I0	I1	LO	Режим работы
0	0	X	0	Передача данных (0) со входа I0 на выход
0	1	X	1	Передача данных (1) со входа I0 на выход
1	X	0	0	Передача данных (0) со входа I1 на выход
1	X	1	1	Передача данных (1) со входа I1 на выход

Поддерживаемые семейства ПЛИС: Virtex-II, Virtex-II Pro.

Тип элемента – примитив.

MUXF8 – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F8 дополнительной логики конфигурируемого логического блока



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

O – выход

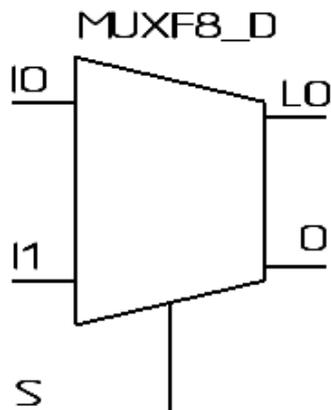
Таблица истинности элемента MUXF8

S	I0	I1	LO	Режим работы
0	0	X	0	Передача данных (0) со входа I0 на выход
0	1	X	1	Передача данных (1) со входа I0 на выход
1	X	0	0	Передача данных (0) со входа I1 на выход
1	X	1	1	Передача данных (1) со входа I1 на выход

Поддерживаемые семейства ПЛИС: Virtex-II, Virtex-II Pro.

Тип элемента – примитив.

MUXF8_D – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F8 дополнительной логики конфигурируемого логического блока, с двумя выходами



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

O – основной выход

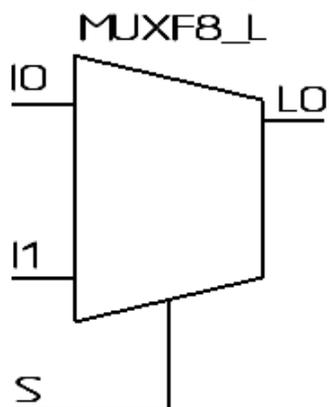
LO – дополнительный (локальный) выход

Таблица истинности элемента MUXF8_D

S	I0	S	I0	I1	O	LO	Режим работы
0	0	0	0	X	0	0	Передача данных (0) со входа I0 на выход
0	1	0	1	X	1	1	Передача данных (1) со входа I0 на выход
1	X	1	X	0	0	0	Передача данных (0) со входа I1 на выход
1	X	1	X	1	1	1	Передача данных (1) со входа I1 на выход

Поддерживаемые семейства ПЛИС: Virtex-II, Virtex-II Pro. Тип элемента – примитив.

MUXF8_L – двухвходовый мультиплексор (2 в 1), реализуемый на базе элемента F8 дополнительной логики конфигурируемого логического блока, с локальным выходом



Назначение выводов:

I0, I1 – информационные входы

S – вход выбора

O – основной выход

LO – дополнительный (локальный) выход

Таблица истинности элемента MUXF8_L

S	I0	I1	LO	Режим работы
0	0	X	0	Передача данных (0) со входа I0 на выход
0	1	X	1	Передача данных (1) со входа I0 на выход
1	X	0	0	Передача данных (0) со входа I1 на выход
1	X	1	1	Передача данных (1) со входа I1 на выход

Поддерживаемые семейства ПЛИС: Virtex-II, Virtex-II Pro.

Тип элемента – примитив.

4.10. Устройства циклического сдвига

BRLSHFT4 – четырехразрядное устройство циклического сдвига данных, поступающих со входа I0 – I3

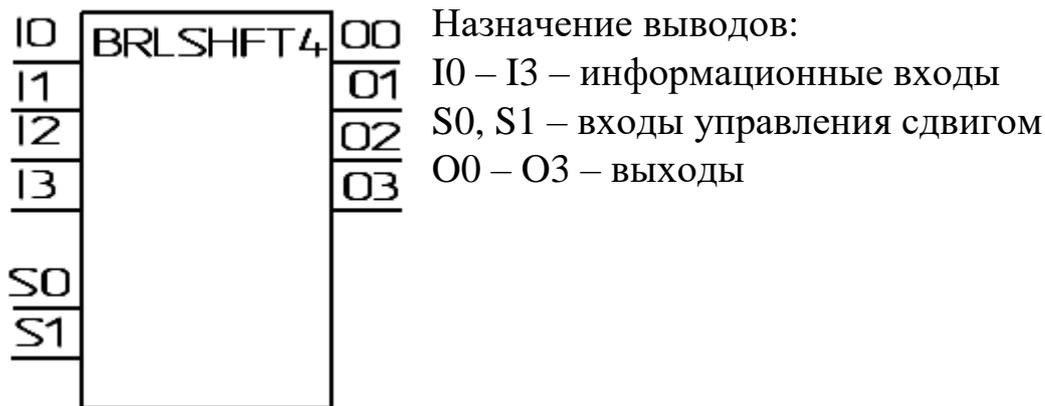


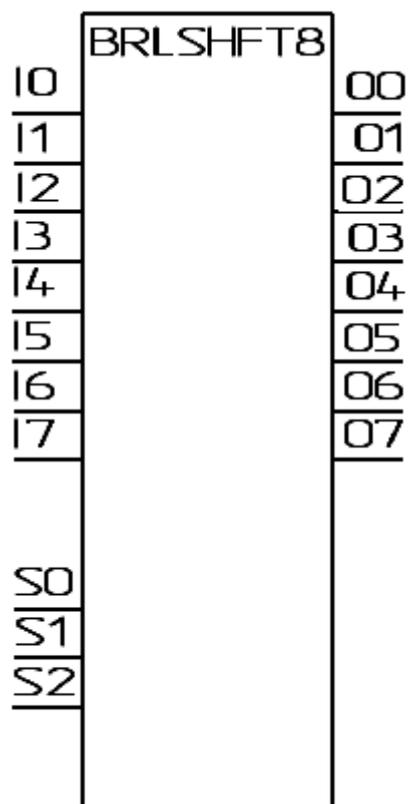
Таблица истинности элемента BRLSHFT4

S0	S1	O0	O1	O2	O3	Режим работы
0	0	DI0	DI1	DI2	DI3	Передача данных на выходы без сдвига
1	0	DI1	DI2	DI3	DI0	Циклический сдвиг влево на один разряд
0	1	DI2	DI3	DI0	DI1	Циклический сдвиг влево на два разряда
1	1	DI3	DI0	DI1	DI2	Циклический сдвиг влево на три разряда

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макросы.

BRLSHFT8 – восьмиразрядное устройство циклического сдвига данных, поступающих со входа I0 – I7



Назначение выводов:

I0 – I7 – информационные входы

S0 – S2 – входы управления сдвигом

O0 – O7 – выходы

Таблица истинности элемента BRLSHFT8

S0	S1	S2	O0	O1	O2	O3	O4	O5	O6	O7	Режим работы
0	0	0	DI0	DI1	DI2	DI3	DI4	DI5	DI6	DI7	Передача данных на выходы без сдвига
1	0	0	DI1	DI2	DI3	DI4	DI5	DI6	DI7	DI0	Циклический сдвиг влево на один разряд

Окончание таблицы

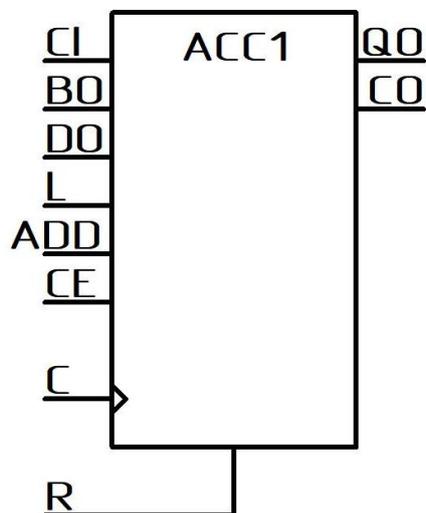
S0	S1	S2	O0	O1	O2	O3	O4	O5	O6	O7	Режим работы
0	1	0	DI2	DI3	DI4	DI5	DI6	DI7	DI0	DI1	Циклический сдвиг влево на два разряда
1	1	0	DI3	DI4	DI5	DI6	DI7	DI0	DI1	DI2	Циклический сдвиг влево на три разряда
0	0	1	DI4	DI5	DI6	DI7	DI0	DI1	DI2	DI3	Циклический сдвиг влево на четыре разряда
1	0	1	DI5	DI6	DI7	DI0	DI1	DI2	DI3	DI4	Циклический сдвиг влево на пять разрядов
0	1	1	DI6	DI7	DI0	DI1	DI2	DI3	DI4	DI5	Циклический сдвиг влево на шесть разрядов
1	1	1	DI7	DI0	DI1	DI2	DI3	DI4	DI5	DI6	Циклический сдвиг влево на семь разрядов

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II.

Тип элемента – макросы.

4.11. Арифметико-логические устройства

АСС1 – одноразрядный аккумулятор с синхронным сбросом, синхронной загрузкой и входом разрешения синхронизации



Назначение выводов:

B0 – вход первого операнда

D0 – вход второго операнда

CI – вход переноса

R – вход сброса

L – вход загрузки

CE – вход разрешения загрузки

C – вход синхронизации

ADD – вход управления режимами суммирования / вычитания

Q0 – выход суммы/разности

CO – выход переноса

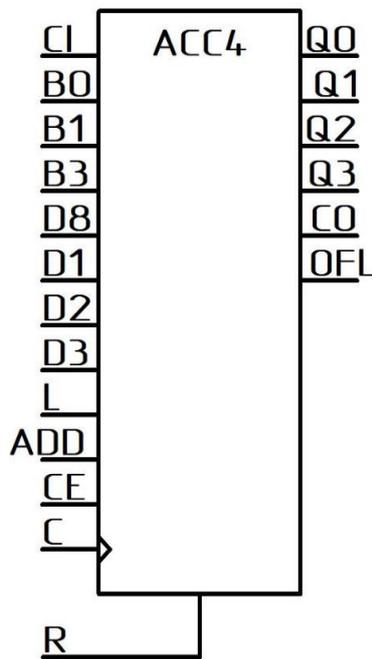
Таблица истинности элемента АСС1

R	L	ADD	CE	C	Q0	Режим работы
1	X	X	X	/	0	Синхронный сброс
0	1	X	X	/	D0	Синхронная загрузка данных со входа D0
0	0	X	0	/	Q0n	Хранение
0	0	1	1	/	$Q0n + B0 + C $	Суммирование содержимого внутреннего регистра с данными со входом B0 с учетом сигнала входного переноса CI
0	0	0	1	/	$Q0n - B0 - C $	Вычитание данных со входа B0 из содержимого внутреннего регистра с учетом сигнала входного переноса (заема) CI

$$CO = (Q0 \& \sim(B0 \wedge ADD)) \vee (Q0 \& CI) \vee (\sim(B0 \wedge ADD) \& CI).$$

Поддерживаемые семейства ПЛИС : XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

АСС4 – четырехразрядный аккумулятор с синхронным сбросом, синхронной загрузкой и входом разрешения синхронизации



Назначение выводов:

- B0 – B3 – входы первого операнда (разряды 0 – 3)
- D0 – D3 – входы второго операнда (разряды 0 – 3)
- CI – вход переноса
- R – вход сброса
- L – вход загрузки
- CE – вход разрешения синхронизации
- C – вход синхронизации
- ADD – вход управления режимами суммирования/ вычитания
- Q0 – Q3 – выход суммы/разности (разряды 0 – 3)
- CO – выход переноса (для операндов без знака)
- OFL – выход переполнения (для операндов со знаком, представленных в дополнительном коде)

Таблица истинности элемента АСС4

R	L	ADD	CE	C	Q[3:0]	Режим работы
1	X	X	X	/	0	Синхронный сброс
0	1	X	X	/	ID[3:0]	Синхронная загрузка данных со входов D0 – D3
0	0	X	0	/	Qn[3:0]	Хранение
0	0	1	1	/	$Qn[3:0] + B[3:0] + C $	Суммирование содержимого внутреннего регистра с данными со входов B0 – B3 с учетом сигнала входного переноса CI
0	0	0	1	/	$Qn[3:0] - B[3:0] - C $	Вычитание данных со входов B0 – B3 из содержимого внутреннего регистра с учетом сигнала входного переноса (заема) CI

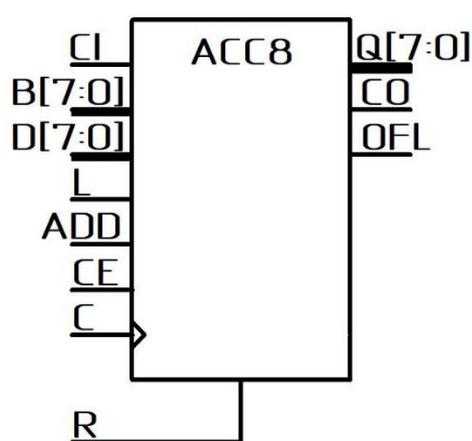
$$CO = (Q3 \& \sim(B3 \wedge ADD)) \vee (Q3 \& C2) \vee (\sim(B3 \wedge ADD) \& C2)$$

$$OFL = (Q3 \& \sim(B3 \wedge ADD)) \vee (Q3 \& C2) \vee (\sim(B3 \wedge ADD) \& C2) \wedge C2$$

C2 – внутренний сигнал сумматора (сигнал переноса из соответствующего разряда).

Поддерживаемые семейства ПЛИС : Spartan-II, Spartan-IIЕ, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

ACC8 – восьмиразрядный аккумулятор с синхронным сбросом, синхронной загрузкой и входом разрешения синхронизации



Назначение выводов:

B0 – B7 – входы первого операнда (разряды 0 – 7), объединенные в шину B[7:0]

D0 – D7 – входы второго операнда (разряды 0 – 7), объединенные в шину D [7:0]

CI – вход переноса

R – вход сброса

L – вход загрузки

CE – вход разрешения синхронизации

C – вход синхронизации

ADD – вход управления режимами суммирования/вычитания

Q0 – Q7 – выход суммы/разности (разряды 0 – 7), объединенные в шину Q [7:0]

CO – выход переноса (для операндов без знака)

OFL – выход переполнения (для операндов со знаком, представленных в дополнительном коде)

Таблица истинности элемента ACC8

R	L	ADD	CE	C	Q[7:0]	Режим работы
1	X	X	X	/	0	Синхронный сброс
						Синхронная загрузка данных со входов D0 – D7
0	1	X	X	/	ID[7:0]	
0	0	X	0	/	Qn[7:0]	Хранение

Окончание таблицы

R	L	ADD	CE	C	Q[7:0]	Режим работы
0	0	1	1	/	$Q_n[7:0] + B[7:0] + C $	Суммирование содержимого внутреннего регистра с данными со входов В0 – В7 с учетом сигнала входного переноса СI
0	0	0	1	/	$Q_n[3:0] - B[3:0] - C $	Вычитание данных со входов В0 – В7 из содержимого внутреннего регистра с учетом сигнала входного переноса (заема) СI

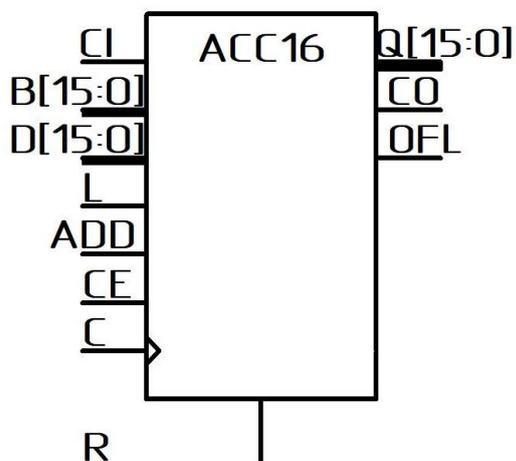
$$CO = (Q7 \& \sim(B7 \wedge ADD)) \vee (Q7 \& C6) \vee (\sim(B3 \wedge ADD) \& C6)$$

$$OFL = ((Q7 \& \sim(B7 \wedge ADD)) \vee (Q7 \& C6) \vee (\sim(B3 \wedge ADD) \& C6)) \wedge C6$$

С6 – внутренний сигнал сумматора (сигнал переноса из соответствующего разряда).

Поддерживаемые семейства ПЛИС : Spartan- II, Spartan- IIE, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

ACC16 – шестнадцатиразрядный аккумулятор с синхронным сбросом, синхронной загрузкой и входом разрешения синхронизации



Назначение выводов:

В0 – В15 – входы первого операнда (разряды 0 – 7), объединенные в шину В[15:0]

D0 – D15 – входы второго операнда (разряды 0 – 7), объединенные в шину D [15:0]

СI – вход переноса

R – вход сброса

L – вход загрузки

CE – вход разрешения синхронизации

C – вход синхронизации

ADD – вход управления режимами суммирования/ вычитания

Q0 – Q15 – выход суммы/разности (разряды 0 – 7), объединенные в шину Q [15:0]

CO – выход переноса (для операндов без знака)

OFL – выход переполнения (для операндов со знаком, представленных в дополнительном коде)

Таблица истинности элемента ACC8

R	L	ADD	CE	C	Q[15:0]	Режим работы
1	X	X	X	/	0	Синхронный сброс
0	1	X	X	/	ID[15:0]	Синхронная загрузка данных со входов D0 – D7
0	0	X	0	/	Qn[7:0]	Хранение
0	0	1	1	/	$Qn[15:0] + B[15:0] + C $	Суммирование содержимого внутреннего регистра с данными со входов B0 – B7 с учетом сигнала входного переноса CI
0	0	0	1	/	$Qn[15:0] - B[15:0] - C $	Вычитание данных со входов B0 – B7 из содержимого внутреннего регистра с учетом сигнала входного переноса (заема) CI

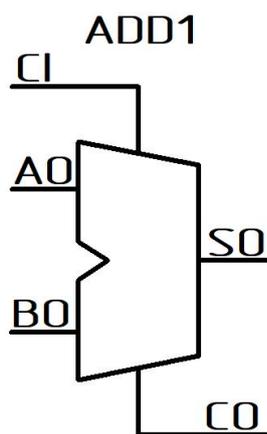
$$CO = (Q15 \& \sim(B15 \wedge ADD)) | (Q15 \& C14) | (\sim(B15 \wedge ADD) \& C14)$$

$$OFL = ((Q15 \& \sim(B15 \wedge ADD)) | (Q15 \& C14) | (\sim(B15 \wedge ADD) \& C14)) \wedge C6$$

C14 – внутренний сигнал сумматора (сигнал переноса из соответствующего разряда).

Поддерживаемые семейства ПЛИС: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

ADD1 – одноразрядный полный сумматор



Назначение выводов:

A0 – вход первого слагаемого

B0 – вход второго слагаемого

CI – вход переноса

S0 – выход суммы

CO – выход переноса

Таблица истинности элемента ADD1

A0	B0	CI	S0	CO	Режим работы
0	0	0	0	0	Суммирование двух операндов с учетом входного переноса (результат меньше 2)
1	0	0	1	0	Суммирование двух операндов с учетом входного переноса (результат меньше 2)
0	1	0	1	0	Суммирование двух операндов с учетом входного переноса (результат меньше 2)
1	1	0	0	1	Суммирование двух операндов с учетом входного переноса (результат равен 2)
0	0	1	1	0	Суммирование двух операндов с учетом входного переноса (результат меньше 2)

Окончание таблицы

A0	B0	CI	S0	CO	Режим работы
1	0	1	0	1	Суммирование двух операндов с учетом входного переноса (результат равен 2)
0	1	1	0	1	Суммирование двух операндов с учетом входного переноса (результат равен 2)
1	1	1	1	1	Суммирование двух операндов с учетом входного переноса (результат больше 2)

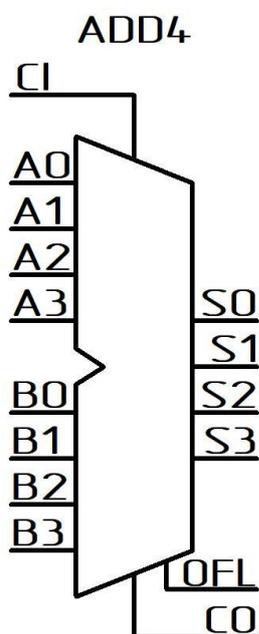
Формирование выходных сигналов:

$$S0 = A0 \wedge B0 \wedge CI$$

$$CO = (A0 \& B0) \vee (A0 \& CI) \vee (B0 \& CI).$$

Поддерживаемые семейства ПЛИС: XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

ADD4 – четырехразрядный полный сумматор



Назначение выводов:

A0 – A3 – входы первого слагаемого (разряды 0 – 3)

B0 – B3 – входы второго слагаемого (разряды 0 – 3)

CI – вход переноса

S0 – S3 – выходы суммы (разряды 0 – 3)

S0 – выход суммы

CO – выход переноса (для операндов без знака)

OFL – выход переполнения (для операндов со знаком, представленных в дополнительном коде)

Таблица истинности элемента ADD4

S	CO	OFL	Режим работы
$ A[3:0]+ B[3:0]+ C $	0	X	Суммирование двух операндов без знака (результат меньше 16)
$ A[3:0]+ B[3:0]+ C $	1	X	Суммирование двух операндов без знака (результат больше или равен 16)
$ A[3:0]+ B[3:0]+ C $	X	0	Суммирование двух операндов со знаком, представленных в дополнительном коде (результат больше – 9 и меньше 8)
$ A[3:0]+ B[3:0]+ C $	X	1	Суммирование двух операндов со знаком, представленных в дополнительном коде (результат больше 7 или меньше – 8)

Формирование выходных сигналов:

$$S0 = A0 \wedge B0 \wedge C1 ; CO = (A0 \& B0)|(A0 \& C1)|(B0 \& C1)$$

$$S1 = A1 \wedge B1 \wedge C0 ; CO = (A1 \& B1)|(A1 \& C0)|(B1 \& C0)$$

$$S2 = A2 \wedge B2 \wedge C1 ; CO = (A2 \& B2)|(A2 \& C1)|(B2 \& C1)$$

$$S3 = A3 \wedge B3 \wedge C2 ; CO = (A3 \& B3)|(A3 \& C2)|(B3 \& C2)$$

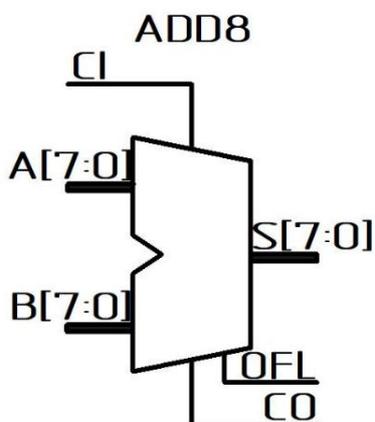
$$OFL = ((A3 \& B3)|(A3 \& C2)|(B3 \& C2)) \wedge C2$$

$$CO = (Q3 \& \sim(B3 \wedge ADD))|(Q3 \& C2)|((B3 \wedge ADD) \& C2)$$

CO – C2 – внутренние сигналы сумматора

Поддерживаемые семейства: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

ADD8 – восьмиразрядный полный сумматор



Назначение выводов:

A0 – A7 – входы первого слагаемого (разряды 0 – 7), объединенные в шину A[7:0]

B0 – B7 – входы второго слагаемого (разряды 0 – 7), объединенные в шину B[7:0]

CI – вход переноса

S0 – S7 – выходы суммы (разряды 0 – 7), объединенные в шину S[7:0]

S0 – выход суммы

CO – выход переноса (для операндов без знака)

OFL – выход переполнения (для операндов со знаком, представленных в дополнительном коде)

Таблица истинности элемента ADD8

S	CO	OFL	Режим работы
$ A[7:0]+ B[7:0]+ C $	0	X	Суммирование двух операндов без знака (результат меньше 256)
$ A[7:0]+ B[7:0]+ C $	1	X	Суммирование двух операндов без знака (результат больше или равен 256)
$ A[7:0]+ B[7:0]+ C $	X	0	Суммирование двух операндов со знаком, представленных в дополнительном коде (результат больше – 129 и меньше 128)
$ A[7:0]+ B[7:0]+ C $	X	1	Суммирование двух операндов со знаком, представленных в дополнительном коде (результат больше 127 или меньше – 128)

Формирование выходных сигналов:

$$S_0 = A_0 \oplus B_0 \oplus C_0 ; CO = (A_0 \& B_0) \vee (A_0 \& C_0) \vee (B_0 \& C_0)$$

$$S_1 = A_1 \oplus B_1 \oplus C_0 ; CO = (A_1 \& B_1) \vee (A_1 \& C_0) \vee (B_1 \& C_0)$$

$$S_i = A_i \oplus B_i \oplus C_{(i-1)} ; C_i = (A_i \& B_i) \vee (A_i \& C_{(i-1)}) \vee (B_i \& C_{(i-2)})$$

$$S_6 = A_6 \oplus B_6 \oplus C_5 ; C_6 = (A_6 \& B_6) \vee (A_6 \& C_5) \vee (B_6 \& C_5)$$

$$S_7 = A_7 \oplus B_7 \oplus C_6 ; CO = (A_7 \& B_7) \vee (A_7 \& C_6) \vee (B_7 \& C_6)$$

$$OFL = ((A_3 \& B_3) \vee (A_3 \& C_2) \vee (B_3 \& C_2)) \wedge C_2$$

C0 – C6 – внутренние сигналы сумматора.

S_i – значение i-го разряда суммы.

A_i – значение i-го разряда первого слагаемого.

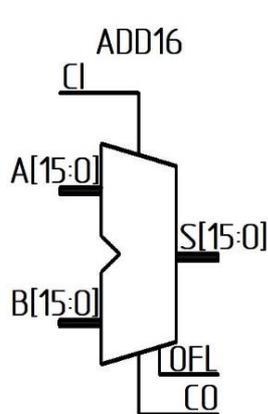
B_i – значение i-го разряда второго слагаемого.

C_i – значение сигнала переноса из i-го разряда.

C_(i – 1) – значение сигнала переноса из (i – 1)-го разряда.

Поддерживаемые семейства: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

ADD16 – шестнадцатиразрядный полный сумматор



Назначение выводов:

A0 – A15 – входы первого слагаемого (разряды 0 – 15), объединенные в шину A[15:0]

B0 – B7 – входы второго слагаемого (разряды 0 – 15), объединенные в шину B[15:0]

CI – вход переноса

S0 – S7 – выходы суммы (разряды 0 – 15), объединенные в шину S[15:0]

CO – выход суммы

CO – выход переноса (для операндов без знака)

OFL – выход переполнения (для операндов со знаком, представленных в дополнительном коде)

Таблица истинности элемента ADD4

S	CO	OFL	Режим работы
$ A[15:0]+ B[15:0]+ C $	0	X	Суммирование двух операндов без знака (результат меньше 65536)
$ A[15:0]+ B[15:0]+ C $	1	X	Суммирование двух операндов без знака (результат больше или равен 65536)
$ A[15:0]+ B[15:0]+ C $	X	0	Суммирование двух операндов со знаком, представленных в дополнительном коде (результат больше – 32769 и меньше 32768)
$ A[15:0]+ B[15:0]+ C $	X	1	Суммирование двух операндов со знаком, представленных в дополнительном коде (результат больше 32767 или меньше – 32768)

Формирование выходных сигналов:

$$S0 = A0 \wedge B0 \wedge C0 ; CO = (A0 \& B0)|(A0 \& C0)|(B0 \& C0)$$

$$S1 = A1 \wedge B1 \wedge C0 ; CO = (A1 \& B1)|(A1 \& C0)|(B1 \& C0)$$

$$Si = Ai \wedge Bi \wedge C(i-1) ; Ci = (Ai \& Bi)|(Ai \& C(i-1)|(Bi \& C(i-2))$$

$$S14 = A14 \wedge B14 \wedge C14 ; C14 = (A14 \& B14)|(A14 \& C13)|(B14 \& C13)$$

$$S15 = A15 \wedge B15 \wedge C15 ; CO = (A15 \& B15)|(A15 \& C14)|(B15 \& C14)$$

$$OFL = ((A15 \& B15)|(A15 \& C14)|(B15 \& C14)) \wedge C14$$

C0 – C14 – внутренние сигналы сумматора.

Si – значение i-го разряда суммы.

Ai – значение i-го разряда первого слагаемого.

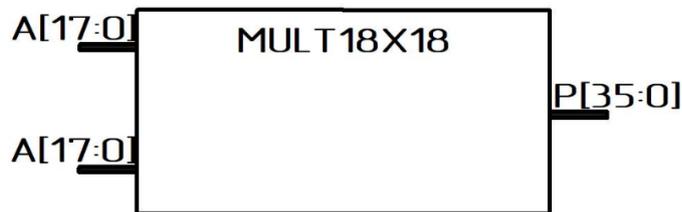
Bi – значение i-го разряда второго слагаемого.

Ci – значение сигнала переноса из i-го разряда.

C(i-1) – значение сигнала переноса из(i – 1)-го разряда.

Поддерживаемые семейства: Spartan-II, Spartan-III, Virtex, Virtex-E, Virtex-II, Virtex-II Pro, XC9500/XV/XL, CoolRunner XPLA3, CoolRunner-II. Тип элемента – макрос.

MULT18X18 – восемнадцатиразрядный умножитель



Назначение выводов:

A0 – A17 – входы первого слагаемого (разряды 0 – 17), объединенные в шину A[17:0]

B0 – B17 – входы второго слагаемого (разряды 0 – 17), объединенные в шину B[17:0]

P0 – P35 – выходы суммы (разряды 0 – 35), объединенные в шину P[35:0]

Таблица истинности элемента MULT18X18

A[17:0]	B[17:0]	P[35:0]	Режим работы
A[17:0]	B[17:0]	A[17:0]* B[17:0]	Умножение двух операндов

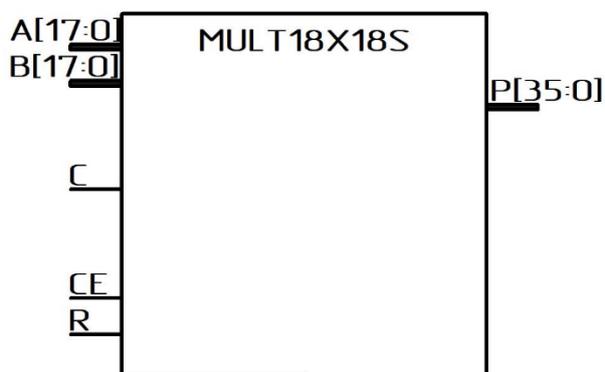
IA[17:0] – значение сигналов на входах A[17:0]

IB[17:0] – значение сигналов на входах B[17:0]

Поддерживаемые семейства: Virtex-II, Virtex-II Pro.

Тип элемента – примитив.

MULT18X18S – восемнадцатиразрядный умножитель с синхронным сбросом и входом разрешения синхронизации



Назначение выводов:

A0 – A17 – входы первого слагаемого (разряды 0 – 17), объединенные в шину A[17:0]

B0 – B17 – входы второго слагаемого (разряды 0 – 15), объединенные в шину B[17:0]

R – вход сброса

CE – вход разрешения синхронизации

S0 – S15 – выходы суммы (разряды 0 – 15), объединенные в шину S[15:0]

CO – выход переноса (для операндов без знака)

OFL – выход переполнения (для операндов со знаком, представленных в дополнительном коде)

Таблица истинности элемента MULT18X18

C	CE	R	A[17:0]	B[17:0]	P[35:0]	Режим работы
/	X	1	X	X	0	Синхронный сброс
/	0	0	X	X	Pn[35:0]	Хранение
/	1	0	A[17:0]	B[17:0]	A[17:0]* B[17:0]	Умножение двух операндов

Pn[35:0] – состояние выходов P[35:0] на предыдущем шаге

IA[17:0] – значение сигналов на входах A[17:0]

IB[17:0] – значение сигналов на входах B[17:0]

Поддерживаемые семейства: Virtex-II, Virtex-II Pro.

Тип элемента – примитив.

Контрольные вопросы

1. Объясните принцип работы D-триггера.
2. В чем заключается принцип работы триггеров с динамическим управлением?
3. Каков принцип работы синхронных триггеров?
4. В чем отличие асинхронного и синхронного управления триггерами?
5. Приведите пример триггера с асинхронным сбросом и установкой.
6. Что такое асинхронный сброс триггера?
7. Назовите входные и выходные триггеры. Опишите их работу.
8. Что такое Z-состояние?
9. Что такое сдвиговый регистр?
10. Приведите схему сдвигового регистра на D-триггерах.
11. Охарактеризуйте регистр с параллельной загрузкой данных?
12. Приведите схему регистра с параллельной загрузкой данных.
13. Что такое счетчик?
14. Приведите схему четырехразрядного счетчика на D-триггерах.
15. В чем разница прямых и реверсивных счетчиков?
16. Дайте схему делителя частоты на 10.
17. Объясните принцип работы компаратора.
18. Что такое дешифратор?
19. Приведите схему дешифратора на логических элементах.
20. Что такое мультиплексор?
21. Изобразите схему мультиплексора на логических элементах.
22. Сколько адресных входов содержит 128-входовый мультиплексор?
23. Какие примеры арифметико-логических элементов вы можете привести?
24. Что такое аккумулятор?
25. Приведите схему 4-разрядного вычитания на базе 4-разрядного сумматора.
26. Сколько выходных разрядов имеет восемнадцатиразрядный умножитель?

ЗАКЛЮЧЕНИЕ

В учебном пособии приведен достаточно полный обзор современных программируемых логических интегральных схем, их архитектуры и устройств ввода/вывода. Описаны основные этапы проектирования цифровых устройств на ПЛИС.

Изложены основы программирования логических интегральных схем на базе ПЛИС фирмы Xilinx как одних из наиболее популярных на сегодняшний день и позволяющих достаточно быстро овладеть общими навыками программирования. Приведены описание этапов проектирования цифровых устройств и программирование ПЛИС фирмы Xilinx. Рассмотрены основные семейства схемотехнических элементов, представлены таблицы истинности и поддерживаемые серии микросхем ПЛИС.

Особое внимание уделено примерам программирования ПЛИС. Изложены принципы программирования помехоустойчивых кодеков и генераторов псевдослучайных последовательностей. Приведены принципиальные схемы и временные диаграммы кодека Хемминга и генератора псевдослучайной последовательности на восьмиразрядном регистре.

Рассмотренный в книге материал даст возможность студентам радиотехнических специальностей достаточно полно изучить принципы функционирования программируемых логических интегральных схем, правила и этапы их программирования, понять особенности разработки цифровых устройств на базе ПЛИС. Авторы полагают, что приведенные в пособии сведения будут полезны при разработке и эксплуатации беспроводных систем передачи информации и других управляющих устройств на базе программируемых логических интегральных схем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кузелин, М. О. Современные семейства ПЛИС фирмы Xilinx : справочное пособие / М. О. Кузелин, Д. А. Кнышев, В. Ю. Зотов. – М. : Горячая линия-Телеком, 2004. – 440с. – ISBN 5-93517-189-9.
2. Стешенко, В. Б. Плис фирмы «ALTERA». Элементная база, система проектирования и языки описания аппаратуры / В. Б. Стешенко. – М. : Додэка, 2002. – 576 с. – ISBN 978-5-97060-342-0.
3. Угрюмов, Е. П. Цифровая схемотехника : учеб. пособие для вузов. / Е. П. Угрюмов. 2-е изд., стер. – СПб. : БХВ-Петербург, 2004. – 357с. – ISBN 978-5-9775-0162-0.
4. Тарасов, И. Е. Разработка цифровых устройств на основе ПЛИС Xilinx с применением языка VHDL / И. Е. Тарасов. – М. : Горячая линия-Телеком, 2005. – 256 с. – ISBN 5-93517-242-9.
5. Зотов, В. Ю. Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx / В. Ю. Зотов. – М. : Горячая линия-Телеком, 2006. – 388 с. – ISBN 5-93517-165-1.
6. Зотов, В. Ю. Новый инструментальный комплект Spartan-3 Starter Kit / В. Ю. Зотов // Компоненты и технологии. – 2005. – № 7. – С. 200 – 204.
7. Зотов, В. Ю. MicroBlaze – семейство тридцатидвухразрядных микропроцессорных ядер // Компоненты и технологии. – 2003. – № 9. – С. 146 – 151.
8. Зотов, В. Ю. Проектирование цифровых устройств на основе ПЛИС фирмы Xilinx в САПР WebPack ISE / В. Ю. Зотов. – М. : Горячая линия-Телеком, 2003. – 624 с. – ISBN 5-93517-136-8.
9. Кнышев, Д. А. ПЛИС фирмы «XILINX». Описание структуры основных семейств / Д. А. Кнышев, М. О. Кузелин. – М. : ДМК-Пресс, 2017. – 238 с. – ISBN 978-5-97060-546-2.
10. Программируемые логические интегральные схемы фирмы Xilinx. Серия FAST FLASH CPLD. Краткое техническое описание семейств XC9500 и XC9500XL. – М. : SCAN, 2000. – 108 с.
11. Программируемые логические интегральные схемы фирмы Xilinx. Серия VIRTEX. Краткое техническое описание. – М. : SCAN, 2000. – 204 с.

12. Скляр, Б. Цифровая связь. Теоретические основы и практическое применение / Б. Скляр. – М. : ИД Вильямс, 2003. – 1104 с. – ISBN 978-5-8459-2071-3.

13. Википедия. JTAG [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/JTAG> (дата обращения: 17.09.2022).

14. Оппенгейм, А. Цифровая обработка сигналов / А. Оппенгейм. – 3-е изд. стер. – М. : Техносфера, 2012. – 1048 с. – ISBN 978-5-94836-329-5.

15. Шустов, М. А. Цифровая схемотехника. Основы построения / М. А. Шустов. – М. : Нит, 2018. – 320 с. – ISBN 978-5-94387-875-6.

16. Радиомодуль NRF24L01+ / PA+LNA 2.4G (Trema-модуль V2.0) [Электронный ресурс]. – URL: <https://wiki.arduino.ru/page/NRF24L01-trema/> (дата обращения: 24.04.2022).

17. Радиомодуль NRF24L01 [Электронный ресурс]. – URL: <https://3d-diy.ru/wiki/arduino-moduli/radio-modul-nrf24l01/> (дата обращения: 16.01.2022).

18. Шмаков, С. Б. Энциклопедия радиолобителя. Современная элементная база / С. Б. Шмаков. – 2-е изд. стер. – СПб. : Наука и Техника, 2012. – 384 с. – ISBN 978-5-94387-859-6.

19. Голиков, А. М. Модуляция, кодирование и моделирование в телекоммуникационных системах. Теория и практика : учеб. пособие / А. М. Голиков. – Томск : Томск. гос. ун-т систем упр. и радиоэлектроники, 2016. – 516 с. – (Учебная литература для вузов). – ISBN 978-5-8114-7828-6.

20. Подключение модуля nRF24L01+ к Arduino – соединяем две arduino по радиоканалу [Электронный ресурс]. – URL: <https://micro-ri.ru/подключение-модуля-nrf24l01-к-arduino/> (дата обращения: 4.10.2023).

21. Левашов, Ю. А. Прием и обработка сигналов : учеб. пособие / Ю. А. Левашов. – Владивосток : Изд-во ВГУЭС, 2004. – 108 с. – ISBN 978-5-7996-1497-3.

22. Подключение модулей связи 2,4 ГГц на базе чипов nRF24L01+ к микроконтроллеру [Электронный ресурс]. – URL: https://aterlux.ru/article/nrf24l01p#_h3_38 (дата обращения: 11.05.2023).

23. Колосовский, Е. А. Устройства приема и обработки сигналов / Е. А. Колосовский. – 2-е изд. стер. – М. : Горячая линия-Телеком, 2012. – 457 с. – ISBN 978-5-9912-0265-7.

24. Коберниченко, В. Г. Основы цифровой обработки сигналов : учеб. пособие / В. Г. Коберниченко. – Екатеринбург : Изд-во Урал. ун-та, 2018. – 150 с. – ISBN 978-5-7996-2464-4.

25. Владимиров, С. С. Математические основы теории помехоустойчивого кодирования : учеб. пособие / С. С. Владимиров. – СПб. : СПбГУТ, 2016. – 96 с. – ISBN 978-5-89160-131-4.

26. Кудряшов, Б. Д. Основы теории кодирования : учеб. пособие / Б. Д. Кудряшов. – СПб. : БХВ-Петербург, 2016. – 400 с. – (Учебная литература для вузов). – ISBN 978-5-9775-3527-4.

Учебное электронное издание

САМОЙЛОВ Сергей Александрович

ПРОГРАММИРОВАНИЕ ЛОГИЧЕСКИХ ИНТЕГРАЛЬНЫХ СХЕМ
В РАДИОТЕХНИКЕ

Учебное пособие

Редактор А. П. Володина

Технический редактор Ш. Ш. Амирсейидов

Компьютерная верстка Е. А. Кузьминой, А. Н. Герасина

Корректор О. В. Балашова

Выпускающий редактор А. А. Амирсейидова

Системные требования: Intel от 1,3 ГГц; Windows XP/7/8/10; Adobe Reader;
дисковод CD-ROM.

Тираж 9 экз.

Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых
Изд-во ВлГУ
rio.vlgu@yandex.ru

Институт информационных технологий и радиоэлектроники
кафедра радиотехники и радиосистем
samoylow@rambler.ru