

## Особенности

- Высокопроизводительные, большой емкости, программируемые пользователем логические интегральные схемы с архитектурой FPGA (Field Programmable Gate Arrays)
  - Ёмкость от 50К до 1М системных вентиляей
  - Системная производительность до 200 МГц
  - Совместимы с шиной PCI 66 МГц
  - Поддерживают функцию Hot-swap для Compact PCI
- Поддержка большинства стандартов ввода-вывода (технология SelectIO™)
  - 16 высокопроизводительных стандартов ввода - вывода
  - Прямое подключение к ZBTRAM устройствам
- Встроенные цепи управления тактированием
  - Четыре встроенных модуля автоподстройки задержек (DLL - delay-locked loop) для расширенного управления тактовыми сигналами как внутри кристалла, так и всего устройства
  - Четыре глобальные сети распределения тактовых сигналов с малыми разбегами фронтов, плюс 24 локальные тактовые сети
- Иерархическая система элементов памяти
  - На базе 4-х входных таблиц преобразования (4-LUT – Look-Up Table) конфигурируемых либо как 16-ти битовая RAM (Random Access Memory), либо как 16-ти битовая двухпортовая RAM, либо как 16-ти битовый сдвиговый регистр
  - Встроенная блочная память, каждый блок конфигурируется как синхронная двухпортовая RAM ёмкостью 4 Кбит
  - Быстрые интерфейсы к внешней высокопроизводительной RAM
- Гибкая архитектура с балансом быстродействия и плотности упаковки логики
  - Специальная логика ускоренного переноса для высокоскоростных арифметических операций
  - Специальная поддержка умножителей
  - Каскадируемые цепочки для функций с большим количеством входов
  - Многочисленные регистры/защелки с разрешением тактирования и синхронные/асинхронные цепи установки и сброса
  - Внутренние шины с тремя состояниями

- Логика периферийного сканирования в соответствии со стандартом IEEE1149.1
- Датчик температуры кристалла
- Проектирование осуществляется пакетами программного обеспечения Foundation™ и Alliance Series, работающими на ПК или рабочей станции
- Конфигурация кристалла хранится во внешнем ПЗУ, и загружается в кристалл после включения питания автоматически или принудительно
  - Неограниченное число циклов загрузки
  - Четыре режима загрузки
- Производятся по технологии 0.22 мкм к-МОП с 5-ти слойной металлизацией на основе статического ОЗУ
- 100% фабричное тестирование

## Описание

Семейство FPGA Virtex™ позволяет реализовать высокопроизводительные, большой ёмкости, цифровые устройства на одном кристалле. Резкое увеличение эффективности реализаций достигнуто благодаря новой архитектуре, более эффективной для размещения и трассировки элементов, а также производству кристаллов на основе 0.22 микронного процесса с 5-ю слоями металлизации. Всё это позволяет использовать кристаллы Virtex как альтернативу масочно-программируемым вентиляльным матрицам. В состав семейства Virtex входят девять микросхем, отличающихся логической ёмкостью (Табл. 1).

Созданное на основе опыта, приобретённого при разработках предыдущих серий FPGA, семейство Virtex является революционным шагом вперед, определяющим новые стандарты в производстве программируемой логики. Сочетая большое разнообразие новых системных свойств, иерархию высокоскоростных и гибких трассировочных ресурсов с передовой технологией изготовления «кремния», семейство Virtex предоставляет разработчику широкие возможности реализации быстродействующих, большой логической ёмкости цифровых устройств, при значительном снижении времени разработки.

Табл. 1. Основные характеристики семейства Virtex

Тип МС	Системные вентили	Матрица КЛБ	Логические ячейки	Число доступных входов-выходов, макс.	Блочная память, Бит	Память на базе LUT, Бит
XCV50	57 906	16x24	1 728	180	32 768	24 576
XCV100	108 904	20x30	2 700	180	40 960	38 400
XCV150	164 674	24x36	3 888	260	49 152	55 296
XCV200	236 666	28x42	5 292	284	57 344	75 264
XCV300	322 970	32x48	6 912	316	65 536	98 304
XCV400	468 252	40x60	10 800	404	81 920	153 600
XCV600	661 111	48x72	15 552	512	98 304	221 184
XCV800	888 439	56x84	21 168	512	114 688	301 056
XCV1000	1 124 022	64x96	27 648	512	131 072	393 216

## Архитектура Virtex

Основными особенностями архитектуры кристаллов семейства Virtex являются гибкость и регулярность. Кристаллы состоят из матрицы КЛБ (Конфигурируемый Логический Блок), которая окружена программируемыми блоками ввода-вывода (БВВ). Все соединения между основными элементами (КЛБ, БВВ) осуществляются с помощью набора иерархических высокоскоростных программируемых трассировочных ресурсов. Изобилие таких ресурсов позволяет реализовывать на кристалле семейства Virtex даже самые громоздкие и сложные проекты.

Кристаллы семейства Virtex производятся на основе статического ОЗУ (Static Random Access Memory – SRAM), поэтому функционирование кристаллов определяется загружаемыми во внутренние ячейки памяти конфигурационными данными. Конфигурационные данные могут загружаться в кристалл несколькими способами. В ведущем последовательном режиме (Master Serial) загрузка осуществляется из внешнего ПЗУ и полностью управляется самой FPGA Virtex. В других режимах управление загрузкой осуществляется внешними устройствами (режимы Select-MAP™, подчиненный последовательный (Slave Serial) и JTAG).

Конфигурационные данные создаются пользователем при помощи программного обеспечения проектирования Xilinx Foundation™ и Alliance Series. Программное обеспечение включает в себя схемный и текстовый ввод, моделирование, автоматическое и ручное размещение и трассировку, создание, загрузку и верификацию загрузочных данных.

## Быстродействие

Кристаллы Virtex обеспечивают более высокую производительность, чем предыдущие поколения FPGA. Проекты могут работать на системных частотах до 200 МГц, включая блоки ввода-вывода. Блоки ввода-вывода Virtex полностью соответствуют спецификациям PCI шины, поэтому кристалл позволяют реализовывать интерфейсные схемы, работающие на частоте 33 МГц или 66 МГц. В дополнение к этому кристаллы Virtex удовлетворяют требованию hot-swap для Compact PCI.

К настоящему времени кристаллы полностью протестированы на «эталонных» схемах. На основе тестов выявлено, что хотя производительность сильно зависит от конкретного проекта, большинство проектов работают на частотах превышающих 100 МГц и могут достигать системных частот до

200 МГц. В Табл. 2 представлены производительности некоторых стандартных функций, реализованных на кристаллах с градацией быстродействия '6'.

В отличие от предыдущих семейств ПЛИС Xilinx, в сериях Virtex™ и Spartan™ градация по быстродействию обозначается классом, а не задержкой на логическую ячейку. Соответственно, в семействах Virtex™ и Spartan™ чем больше класс, тем выше быстродействие.

Табл. 2. Производительность стандартных функций.

Функция	Разрядность, [Бит]	Virtex-6
<b>Внутрикристалльная производительность</b>		
Сумматор	16	5.0 нс.
	64	7.2 нс.
Конвейерный умножитель	8x8	5.1 нс.
	16x16	6.0 нс.
Декодер адреса	16	4.4 нс.
	64	6.4 нс.
Мультиплексор	16:1	5.4 нс.
Схема контроля по четности	9	4.1 нс.
	18	5.0 нс.
	36	6.9 нс.
<b>Системная производительность</b>		
Стандарт HSTL Class IV		200 МГц
Стандарт LVTTTL		180 МГц

## Описание архитектуры

### Матрица Virtex

Программируемая пользователем вентиляционная матрица серии Virtex показана на Рис.1. Соединение между КЛБ осуществляется с помощью главных трассировочных матриц - ГТМ. ГТМ - это матрица программируемых транзисторных двунаправленных переключателей, расположенных на пересечении горизонтальных и вертикальных линий связи. Каждый КЛБ окружен локальными линиями связи (VersaBlock™), которые позволяют осуществить соединения с матрицей ГТМ.

Интерфейс ввода-вывода VersaRing создает дополнительные трассировочные ресурсы по пери-

ферии кристалла. Эти трассы улучшают общую "трассируемость" устройства и возможности трассировки после закрепления электрических цепей к конкретным контактам.

Архитектура Virtex также включает следующие элементы, которые соединяются с матрицей GRM:

- Специальные блоки памяти (BRAMs) размером 4096 бит каждый.
- Четыре модуля автоподстройки задержек (DLL), предназначенных для компенсации задержек тактовых сигналов, а также деления, умножения и сдвига фазы тактовых частот.
- Буферы с тремя состояниями (BUFT), которые расположены вблизи каждого КЛБ и управляют горизонтальными сегментированными трассами.

Коды, записанные в ячейки статической памяти, управляют настройкой логических элементов и коммутаторами трасс, осуществляющих межсоединения в схеме. Эти коды загружаются в ячейки после включения питания и могут перезагружаться в процессе работы, если необходимо изменить реализуемые микросхемой функции.

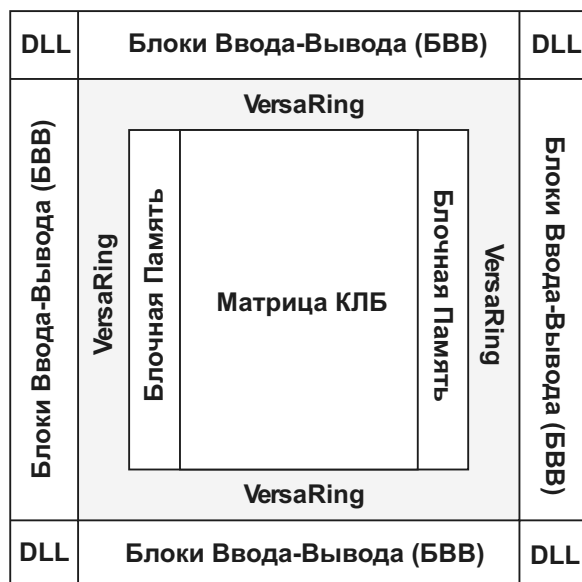


Рис. 1. Структура архитектуры Virtex

### Блок ввода-вывода

Основным отличительным свойством БВВ семейства Virtex является поддержка широкого спектра стандартов сигналов ввода-вывода. На Рис.2 представлена структурная схема БВВ. В Табл. 3 перечислены поддерживаемые стандарты.

БВВ содержит три запоминающих элемента, функционирующих либо как D-триггеры, либо как триггеры-защелки. Каждый БВВ имеет входной сигнал синхронизации (CLK) распределенный на три триггера и независимые для каждого триггера сигналы разрешения тактирования (Clock Enable - CE).

Кроме того, на все триггеры заведен сигнал Сброса/Установки (Set/Reset-SR). Для каждого триггера этот сигнал может быть сконфигурирован независимо как синхронная установка (Set), синхронный сброс (Reset), асинхронная предустановка (Preset) или асинхронный сброс (Clear).

Входные и выходные буферы, а также все управляющие сигналы в БВВ допускают независимый выбор полярности. Данное свойство не отображено на блок схеме БВВ, но контролируется программой проектирования.

Все контакты защищены от повреждения электростатическим разрядом и от всплесков перенапряжения. Реализованы две формы защиты от перенапряжения, одна допускает 5 В совместимость, а другая нет. Для случая 5 В совместимости, структура, подобная диоду Зенера, закорачивает на землю контакт, когда напряжение на нем возрастает приблизительно до 6.5 В. В случае, когда требуется 3.3 В PCI совместимость, обычные диоды ограничения могут подсоединяться к источнику питания выходных каскадов,  $V_{CCO}$ . Тип защиты от перенапряжения может выбираться независимо для каждого контакта. По выбору, к каждому контакту может подключаться:

1. Резистор, соединенный с земляной шиной (pull-down)
2. Резистор, соединенный с шиной питания (pull-up)
3. Маломощная схема удержания последнего состояния (week-keeper).

До начала процесса конфигурирования микросхемы все выводы, не задействованные в этом процессе, принудительно переводятся в состояние высокого импеданса. Pull-down резисторы и элементы week-keeper неактивны, а pull-up резисторы можно активировать.

Активация pull-up резисторов перед конфигурацией управляется внутренними глобальными линиями через управляющие режимные контакты. Если pull-up резисторы не активны, то выводы находятся в состоянии неопределенного потенциала. Если в проекте необходимо иметь определенные логические уровни до начала процесса конфигурирования нужно использовать внешние резисторы.

Все БВВ микросхемы Virtex совместимы со стандартом IEEE 1149.1 периферийного сканирования.

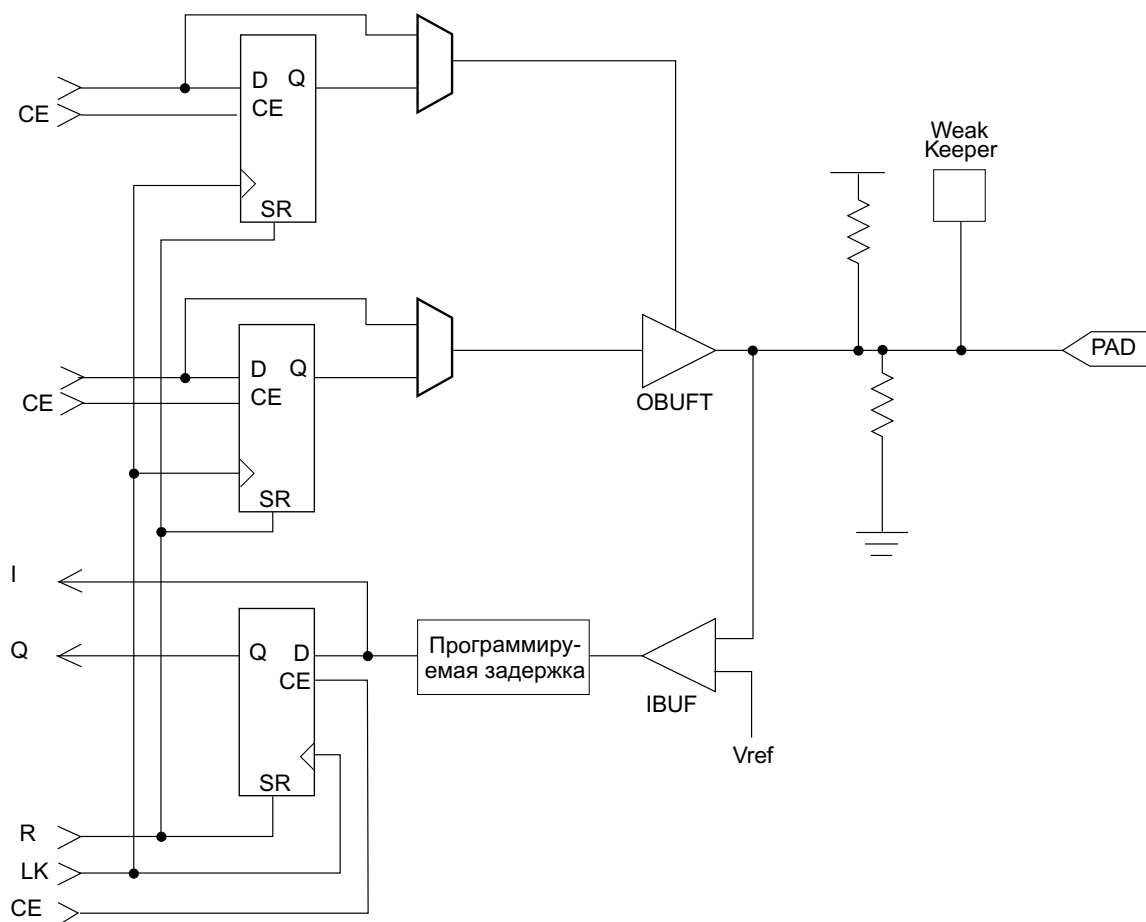


Рис. 2. БВВ Virtex

Табл. 3: Поддерживаемые стандарты ввода-вывода

Стандарт В/В	Напряжение порогового уровня входных каскадов ( $V_{REF}$ )	Напряжение питания выходных каскадов ( $V_{CCO}$ )	Напряжение согласования с платой ( $V_{TT}$ )	5 В совместимость
LVTTL	нет	3.3	нет	да
LVCMOS2	нет	2.5	нет	да
PCI, 5 В	нет	3.3	нет	да
PCI, 3.3 В	нет	3.3	нет	нет
GTL	0.8	нет	1.2	нет
GTL+	1.0	нет	1.5	нет
HSTL Class I	0.75	1.5	0.75	нет
HSTL Class III	0.9	1.5	1.5	нет
HSTL Class IV	0.9	1.5	1.5	нет
SSTL3 Class I & II	1.5	3.3	1.5	нет
SSTL2 Class I & II	1.25	2.5	1.25	нет
CTT	1.5	3.3	1.5	нет
AGP	1.32	3.3	нет	нет

**Ввод сигнала**

Входной сигнал БВВ может быть протрассирован либо непосредственно к блокам внутренней логики, либо через входной триггер.

Кроме того, между выходом буфера и D-входом триггера может быть подключен элемент задержки, исключающий время удержания для случая кон-

такт-контакт. Данная задержка согласована с внутренней задержкой распределения сигнала тактирования FPGA, что гарантирует нулевое время удержания для распределения сигналов контакт-контакт.

Каждый входной буфер может быть сконфигурирован таким образом, чтобы удовлетворять одному

из низковольтных сигнальных стандартов, поддерживаемых устройством. В некоторых из этих стандартов входной буфер использует напряжение порогового уровня ( $V_{REF}$ ), формируемое пользователем. Использование напряжений  $V_{REF}$  позволяет ввести в устройство принудительные опорные величины для различных, близких по используемым логическим уровням стандартов (см. также "Банки ввода-вывода").

К каждому входу после окончания процесса конфигурирования могут быть, по выбору, подключены внутренние резисторы (либо pull-up, либо pull-down). Величина этих резисторов лежит в пределах 50-150 КОм.

### Вывод сигнала

Выходной сигнал проходит через буфер с тремя состояниями, выход которого соединен непосредственно с контактом. Сигнал может быть протраасирован на вход буфера с тремя состояниями, либо непосредственно от внутренней логической структуры, либо через выходной триггер блока ввода-вывода

Управление буфером с тремя состояниями также может осуществляться либо непосредственно от внутренней логической структуры, либо через специальный триггер БВВ, который позволяет создать синхронное управление сигналом разрешения и запрещения для буфера с тремя состояниями. Каждый такой выходной каскад рассчитан на вытекающий ток до 48 мА и вытекающий ток до 24 мА. Программирование мощности и скорости нарастания сигнала выходного каскада позволяет минимизировать переходные процессы в шинах.

Для большинства сигнальных стандартов выходной уровень логической единицы зависит от приложенного извне напряжения  $V_{CCO}$ . Использование напряжения  $V_{CCO}$  позволяет ввести в устройство принудительные опорные величины для различных, близких по используемым логическим уровням стандартов (см. также "Банки ввода-вывода").

По выбору, к каждому выходу может быть подключена схема week-кеерег. Если данная цепь активирована (пользователем на этапе создания схемы), то она следит за напряжением на контакте микросхемы и создает слабую нагрузку для входного сигнала, подключенную либо к "земле" (если на входе уровень логического нуля), либо к источнику питания (если на входе уровень логической единицы). Если контакт подключен к нескольким источникам сигнала, эта цепь удерживает уровень входного сигнала в его последнем состоянии, при условии, что все источники были переведены в состояние с высоким импедансом. Поддержание таким путем одного из допустимых логических уровней позволяет ликвидировать неопределенность уровня шины.

Так как схема week-кеерег использует входной буфер для слежения за входным уровнем, то необходимо использовать подходящее значение напряжения  $V_{REF}$ , если выбранный сигнальный стандарт требует этого. Подключение данного напряжения должно удовлетворять требованиям правил разбегания на банки.

### Банки ввода-вывода

Некоторые из описанных выше стандартов требуют подключения напряжения  $V_{CCO}$  и/или  $V_{REF}$ . Эти внешние напряжения подключаются к контактам микросхемы, которые функционируют группами, называемыми банками.

Как показано на рис.3, каждая кромка микросхемы разделена на два банка. Каждый банк имеет несколько контактов  $V_{CCO}$ , но все они должны быть подключены к одному и тому же напряжению. Это напряжение определяется выбранным для данного банка стандартом выходных сигналов.

Стандарты для выходных сигналов конкретного банка могут быть различными только в том случае, если они используют одинаковое значение напряжения  $V_{CCO}$ . Совместимые стандарты показаны в Табл. 4. GTL и GTL+ присутствуют везде, поскольку их выходы с открытым стоком не зависят от значения  $V_{CCO}$ .

Некоторые сигнальные стандарты требуют подачи соответствующих пороговых напряжений  $V_{REF}$  на входные каскады. При этом определенные БВВ автоматически конфигурируются как входы, соответствующие напряжению  $V_{REF}$ . Приблизительно один контакт из шести в каждом банке может выполнять эту роль.

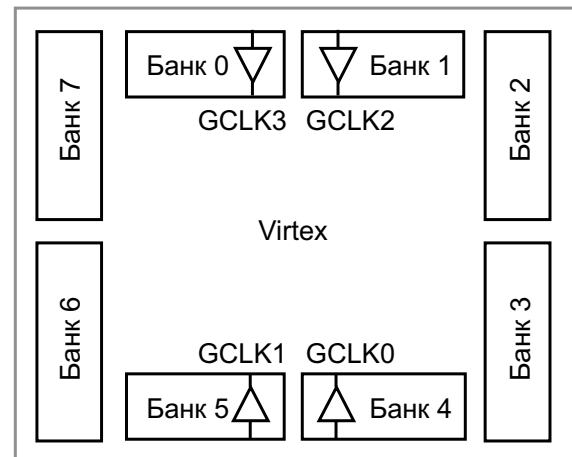


Рис. 3. Банки ввода-вывода Virtex

Табл. 4. Выходные совместимые стандарты

$V_{CCO}$	Совместимые стандарты
3.3 В	PCI, LVTTTL, SSTL3 I, SSTL3 II, CTT, AGP, GTL, GTL+
2.5 В	SSTL2 I, SSTL2 II, LVCMOS2, GTL, GTL+
1.5 В	HSTL I, HSTL III, HSTL IV, GTL, GTL+

Контакты  $V_{REF}$  в пределах одного банка внутренне между собой соединены, следовательно, только одно значение напряжения  $V_{REF}$  может быть использовано в рамках одного банка. Для правильной работы все контакты  $V_{REF}$  одного банка, должны быть подсоединены к внешнему источнику напряжения.

В пределах одного банка можно одновременно использовать входы, которые требуют напряжения  $V_{REF}$  и входы, которые этого не требуют. В то же

время, только одно значение напряжения  $V_{REF}$  может быть использовано в рамках одного банка. Входные буферы, которые используют  $V_{REF}$ , не совместимы с сигналами 5 В стандартов.

Контакты  $V_{CCO}$  и  $V_{REF}$  для каждого банка приведены в таблицах и диаграммах под конкретный корпус и кристалл. На диаграммах также показано к какому банку относится конкретный контакт ввода-вывода.

В рамках конкретного типа корпуса микросхемы число контактов  $V_{CCO}$  и  $V_{REF}$  может меняться в зависимости от ёмкости кристалла. Чем больше кристалл по логической ёмкости, тем большее число контактов ввода-вывода преобразовано в контакты типа  $V_{REF}$ . Поскольку существует максимальный набор контактов  $V_{REF}$  для меньших кристаллов, имеется возможность проектирования печатной платы, позволяющей также использовать на ней и большие кристаллы с таким же типом корпуса. Все контакты  $V_{REF}$ , предполагаемые к использованию для больших кристаллов, при этом, должны быть подсоединены к напряжению  $V_{REF}$  и не должны использоваться как контакты ввода-вывода.

В меньших кристаллах некоторые из контактов  $V_{CCO}$ , используемые в больших кристаллах не соединены внутри корпуса. Эти не присоединенные контакты могут быть оставлены не присоединенными вне микросхемы или быть подключены к напряжению  $V_{CCO}$  при необходимости обеспечения совместимости разрабатываемой печатной платы с большими кристаллами.

В корпусах типа TQ-144 и PQ-240/HQ-240 все кон-

такты  $V_{CCO}$  соединены вместе внутри микросхемы и, следовательно, ко всем из них должно быть подключено одно и тоже напряжение  $V_{CCO}$ . В корпусе CS-144 пары банков, расположенные на одной стороне, внутренне соединены, обеспечивая, таким образом, возможность выбора только четырех возможных значений напряжения для  $V_{CCO}$ . Контакты  $V_{REF}$  остаются внутренне соединенными в рамках каждого из восьми банков и могут использоваться, как было описано выше.

## Конфигурируемый логический блок - КЛБ

Базовым элементом КЛБ является логическая ячейка - ЛЯ (Logic Cell - LC). ЛЯ состоит из 4-х входного функционального генератора, логики ускоренного переноса и запоминающего элемента. Выход каждого функционального генератора каждой логической ячейки подсоединен к выходу КЛБ и к D-входу триггера. Каждый КЛБ серии Virtex содержит четыре логические ячейки, организованные в виде двух одинаковых секций, Рис. 4. На Рис. 5 представлено детальное изображение одной секции.

В дополнение к четырем базовым логическим ячейкам, КЛБ серии Virtex содержит логику, которая позволяет комбинировать ресурсы функциональных генераторов для реализации функций от пяти или шести переменных. Таким образом, при оценке числа эквивалентных системных вентилях для микросхем семейства Virtex, каждый КЛБ приравнивается к 4.5 ЛЯ.

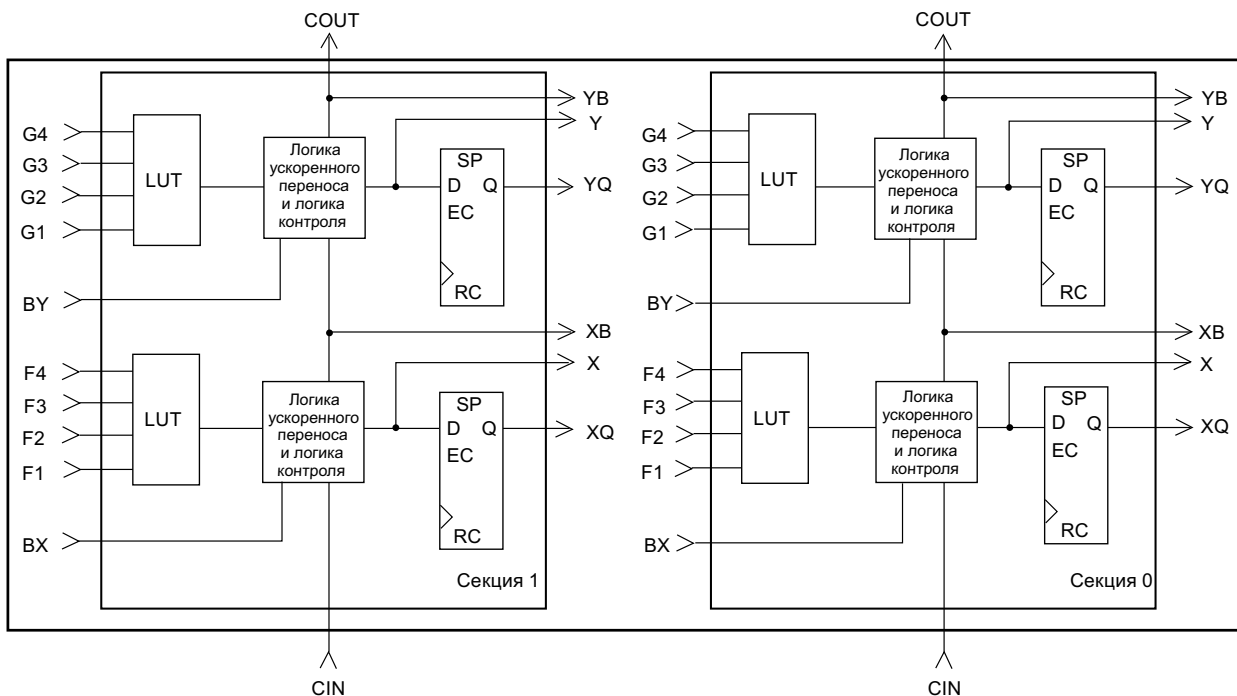


Рис. 4: 2-х секционное КЛБ Virtex

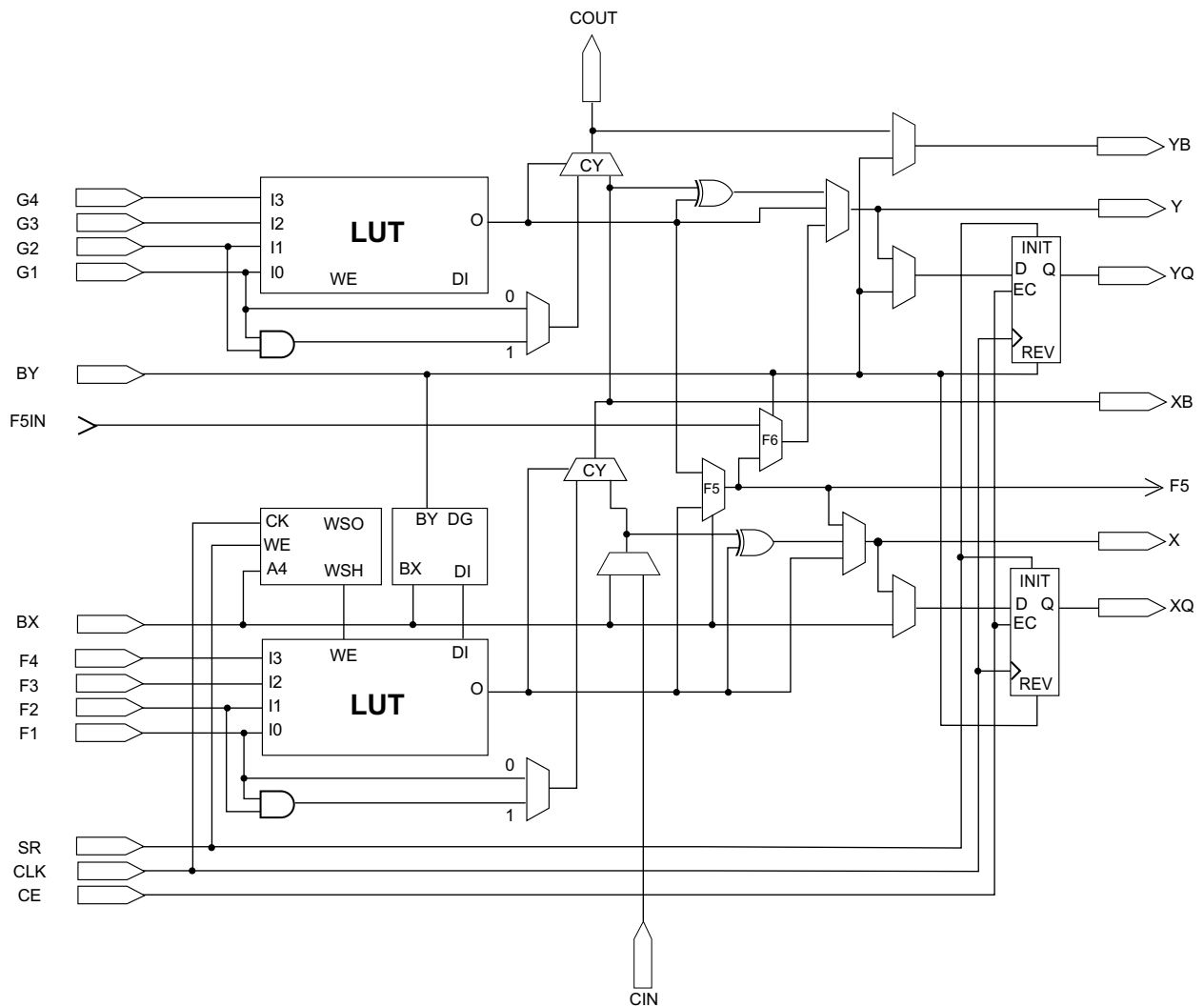


Рис. 5: Детальный вид секции Virtex

**Таблица Преобразования**

Функциональные генераторы реализованы в виде 4-х входных таблиц преобразования (Look-Up Table - LUT). Кроме использования в качестве функциональных генераторов, каждый LUT-элемент может быть также использован как синхронная память типа RAM размерностью 16x1 бит. Более того, из двух LUT-элементов в рамках одной секции можно реализовать синхронную RAM-память размерностью 16x2 бита или 32x1 бит, либо двухпортовую синхронную RAM-память размерностью 16x1 бит.

На LUT-элементе микросхемы Virtex может быть реализован 16-ти битовый сдвиговый регистр, который идеально подходит для захвата высокоскоростных или пакетных потоков данных. Этот режим может также использоваться для запоминания данных в приложениях цифровой обработки сигналов.

**Запоминающие элементы**

Запоминающие элементы в каждой секции КЛБ Virtex могут конфигурироваться как динамические триггеры (чувствительные к фронту сигнала) D-типа, либо как триггеры-защелки, чувствительные к

уровню сигнала. D-вход триггера может управляться либо от функционального генератора в рамках той же секции КЛБ, либо непосредственно от входов данной секции КЛБ, минуя функциональные генераторы.

Кроме сигналов синхронизации (Clock) и разрешения синхронизации (Clock Enable - CE) в каждой секции КЛБ есть сигналы синхронной установки (Set) и сброса (Reset). Обозначение этих сигналов – SR и BY соответственно. Сигнал SR переводит запоминающий элемент в состояние, определенное для него в конфигурационных данных, а сигнал BY – в противоположное состояние. Эти же сигналы могут быть использованы также в качестве асинхронной предустановки (Preset) и очистки (Clear). Все сигналы управления могут быть независимо проинвертированы. Они заведены на оба триггера в рамках конкретной секции КЛБ.

**Дополнительная логика**

Дополнительная логика, входящая в каждый КЛБ, представлена двумя мультиплексорами: F5 и F6.

На вход мультиплексора F5 заведены сигналы с выходов функциональных генераторов данной секции КЛБ. Этот узел может работать как функцио-

нальный генератор, реализующий любую 5-ти входную функцию, либо как мультиплексор 4:1, либо как некоторая функция от девяти входных переменных.

Аналогично, мультиплексор F6 объединяет выходы всех 4-х функциональных генераторов КЛБ, используя один из выходов мультиплексора F5. Это позволяет реализовать либо любую 6-ти входную функцию, либо мультиплексор 8:1, либо некоторую функцию до 19-ти переменных.

Каждый КЛБ имеет четыре сквозных линии – по одной на каждую логическую ячейку. Эти линии используются как дополнительные входы данных, либо как дополнительные трассировочные ресурсы, не расходующие логические ресурсы.

### Арифметическая логика

Каждая ЛЯ содержит специальную логику ускоренного переноса, которая обеспечивает наилучшую реализацию на ПЛИС различных арифметических функций. КЛБ содержит две отдельные цепи переноса - по одной на каждую секцию. Размерность цепи переноса - два бита на КЛБ

Арифметическая логика включает в себя элемент, реализующий функцию исключающего ИЛИ, который позволяет реализовать однобитный сумматор в одной логической ячейке.

В каждой логической ячейке имеется элемент, реализующий функцию И (AND), который предназначен для построения быстродействующих умножителей.

Специальные трассы логики ускоренного переноса могут также использоваться для каскадного включения функциональных генераторов при необходимости создания функций с большим количеством входных переменных.

### Буферы с тремя состояниями

Каждый КЛБ Virtex содержит два буфера с тремя состояниями, которые нагружены на внутренние шины (см. также “Специальные трассы”). Каждый буфер BUFT имеет независимый вход управления третьим состоянием и независимый входной контакт.

### Блочная память (Block RAM)

В FPGA Virtex встроена особая блочная память (Block SelectRAM) большой ёмкости. Она создана в дополнение к распределенной памяти небольшой ёмкости (Select RAM), реализованной на таблицах преобразования (Look Up Table RAM – LUTRAM).

Блоки памяти Block Select RAM+ организованы в виде столбцов. Все устройства Virtex содержат два таких столбца, по одному вдоль каждой вертикальной кромки. Эти колонки увеличивают полный размер кристалла. Каждый блок памяти равен по высоте четырем КЛБ, таким образом, микросхема Virtex, имеющая 64 КЛБ по высоте, содержит 16 блоков памяти на колонку и 32 блока памяти в целом.

В Табл. 5 приводятся ёмкости блочной памяти для различных кристаллов Virtex.

Каждый блок памяти, как показано на Рис. 6 это полностью синхронная двухпортовая RAM с неза-

висимым управлением для каждого порта. Размерность шины данных для обеих портов может быть сконфигурирована независимо, что позволяет создавать преобразователи размерности шины.

В Табл. 6 показаны возможные соотношения размерностей шин данных и адреса.

Табл. 5. Ёмкость блочной памяти

Кристалл Virtex	Число блоков	Общий объём блочной памяти [Бит]
XCV50	8	32 768
XCV100	10	40 960
XCV150	12	49 152
XCV200	14	57 344
XCV300	16	65 536
XCV400	20	81 920
XCV600	24	98 304
XCV800	28	114 688
XCV1000	32	131 072

Табл. 6. Соотношение шин адреса и данных

Разрядность	Глубина	Шина адреса	Шина данных
1	4096	ADDR<11:0>	DATA<0>
2	2048	ADDR<10:0>	DATA<1:0>
4	1024	ADDR<9:0>	DATA<3:0>
8	512	ADDR<8:0>	DATA<7:0>
16	256	ADDR<7:0>	DATA<15:0>

В кристаллах Virtex созданы специальные трассировочные ресурсы для связи блочной памяти с блоками CLB и другими блоками памяти.

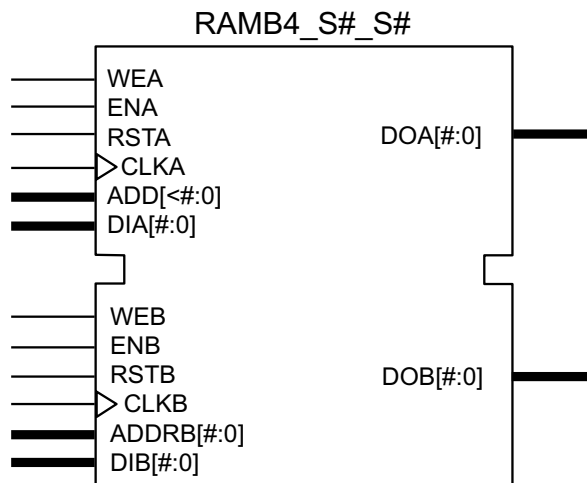


Рис. 6. Блок памяти

## Программируемая трассировочная матрица

Быстродействие проекта, рассчитанного для наилучшего случая, ограничивает величина задержки для наиболее длинной трассы. Поэтому архитектура трассировочных ресурсов и программы размещения и трассировки создавались с учетом использования их в едином процессе оптимизации. Этот совместный процесс оптимизации минимизирует наиболее длинные пути и, таким образом,



создает проект с наилучшей системной производительностью.

Кроме того, совместная оптимизация сокращает время компиляции, так как программное обеспечение и архитектура микросхемы создавались с учетом наилучшего взаимодействия. Циклы проектирования, таким образом, сократились благодаря более коротким временам каждой из итераций всего процесса.

### Локальные связи

Как показано на Рис. 7 в кристалле Virtex созданы локальные трассировочные ресурсы, называемые VersaBlock. Они позволяют реализовать три типа

соединений:

1. Связи между таблицами преобразования (LUT), триггерами и главной трассировочной матрицей (ГТМ)
2. Внутренние обратные связи КЛБ которые создают высокоскоростные связи с таблицами преобразования в рамках одного КЛБ, и позволяют соединять их в виде цепочек с минимальными задержками распространения сигналов
3. Прямые трассы, которые создают высокоскоростные соединения с соседними по горизонтали КЛБ, избегая при этом больших задержек, присущих трассам ГТМ



**Рис. 7. Локальные связи в кристалле Virtex**

### Трассировочные ресурсы общего назначения

Большинство связей в кристаллах Virtex реализуются с помощью трассировочных ресурсов общего назначения и, следовательно, большая часть ресурсов межсоединений связана с этим типом трассировочной иерархии. Трассировочные ресурсы общего назначения расположены в виде горизонтальных и вертикальных трассировочных каналов и размещены в непосредственной близости от строк и столбцов матрицы, образованной блоками КЛБ. Ниже перечислены эти ресурсы:

- Примаыкающая к каждому КЛБ главная трассировочная матрица – ГТМ. ГТМ– это матрица переключателей, с помощью которых коммутируются горизонтальные и вертикальные трассы и посредством которых блоки КЛБ получают доступ к трассировочным ресурсам общего назначения
- ГТМ связана в каждом из четырех направлений с соседней ГТМ посредством 24-х трасс одинарной длины
- 96 буферизованных HEX-линий трассируют ГТМ сигналы к шести другим ГТМ в каждом из четырех направлений. HEX-линии организованы в виде зигзагообразных линий. HEX-линии могут подключаться к источникам сигнала только в своих конечных точках или сере-

динных (три блока от источника). Одна третья часть HEX-линий является двунаправленными, в то время как остальные – однонаправленные.

- 12 длинных линий являются буферизованными, двунаправленными линиями, распространяющими сигналы в микросхеме быстро и эффективно. Вертикальные длинные линии имеют протяженность равную полной высоте кристалла, а горизонтальные длинные линии – полной ширине.

### Трассировочные ресурсы для блоков ввода-вывода

Кристалл Virtex имеет дополнительные трассировочные ресурсы, расположенные по периферии всей микросхемы. Эти трассировочные ресурсы формируют добавочный интерфейс между КЛБ и БВВ. Эти дополнительные ресурсы, называемые VersaRing, улучшают возможности закрепления сигналов за контактами и переадресации уже сделанного закрепления, если это требование накладывается расположением сигналов на печатной плате. При этом сокращается время изготовления всего проекта, т. к. изготовление и проектирование печатной платы можно выполнять одновременно с проектированием FPGA.

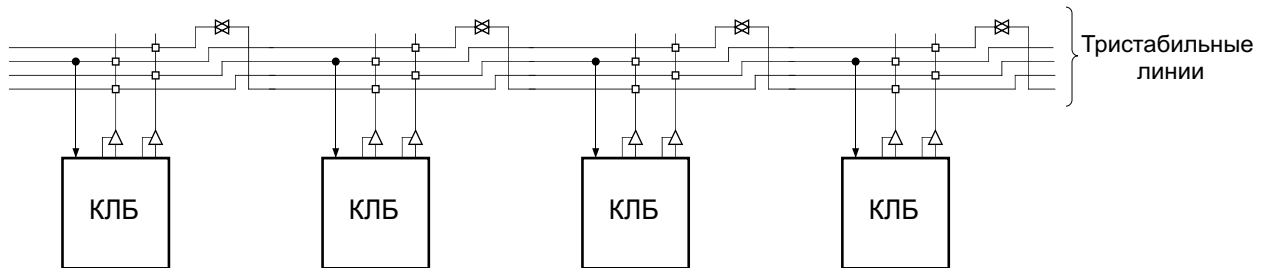
**Специальные трассировочные ресурсы.**

Некоторые классы сигналов требуют наличия специальных трассировочных ресурсов для получения максимального быстродействия. В устройстве Virtex специальные трассировочные ресурсы создавались для двух классов сигналов:

- Горизонтальные трассировочные ресурсы создавались для реализации в микросхеме шин с тремя состояниями. Четыре разделенные ли-

нии шин реализованы для каждой строки КЛБ, позволяя организовывать сразу несколько шин в пределах одной строки. (см. Рис. 8)

- Две специальные линии для распространения сигналов быстрого переноса к прилегающему КЛБ в вертикальном направлении



**Рис. 8. Подключение трехстабильных буферов к горизонтальным линиям**

**Глобальные трассировочные ресурсы**

Глобальные трассировочные ресурсы распределяют тактовые сигналы и другие сигналы с большим коэффициентом разветвления по выходу на всем пространстве кристалла. Кристалл Virtex имеет два типа глобальных трассировочных ресурсов, называемых соответственно первичными и вторичными.

- Первичные глобальные трассировочные ресурсы представляют собой четыре специальные глобальные сети со специально выделенными входными контактами и связанными с ними глобальными буферами, спроектированными для распределения сигналов синхронизации с высоким коэффициентом разветвления и с минимальными разбегами фронтов. Каждая такая сеть может быть нагружена на входы синхронизации всех КЛБ, БВВ и Block RAM – блоков микросхемы. Источниками сигналов для этих сетей могут быть только глобальные буферы. Всего имеется четыре глобальных буфера – по одному для каждой глобальной сети.
- Вторичные глобальные трассировочные ресурсы состоят из 24 магистральных линий, 12 вдоль верхней кромки кристалла и 12 вдоль нижней. По этим связям может быть распро-

странено до 12-ти уникальных сигналов на колонку по 12 длинным линиям данной колонки. Вторичные ресурсы являются более гибкими, чем первичные, т.к. эти сигналы, в отличие от первичных, могут трассироваться не только до входов синхронизации.

**Распределение сигналов синхронизации**

Как было описано выше, Virtex имеет высокоскоростные, с малыми искажениями трассировочные ресурсы для распределения сигналов синхронизации на всем пространстве микросхемы. Типичное распределение цепей синхронизации показано на Рис. 9.

В микросхему встроено четыре глобальных буфера, два - в середине верхней части микросхемы, два - в середине нижней части. Эти буферы через первичные глобальные сети могут подводить сигналы синхронизации на любой тактовый вход.

Для каждого глобального буфера имеется соответствующий, примыкающий к нему контакт микросхемы. Сигнал на вход глобального буфера может подаваться как с этих контактов, так и от сигналов, трассируемых ресурсами общего назначения.

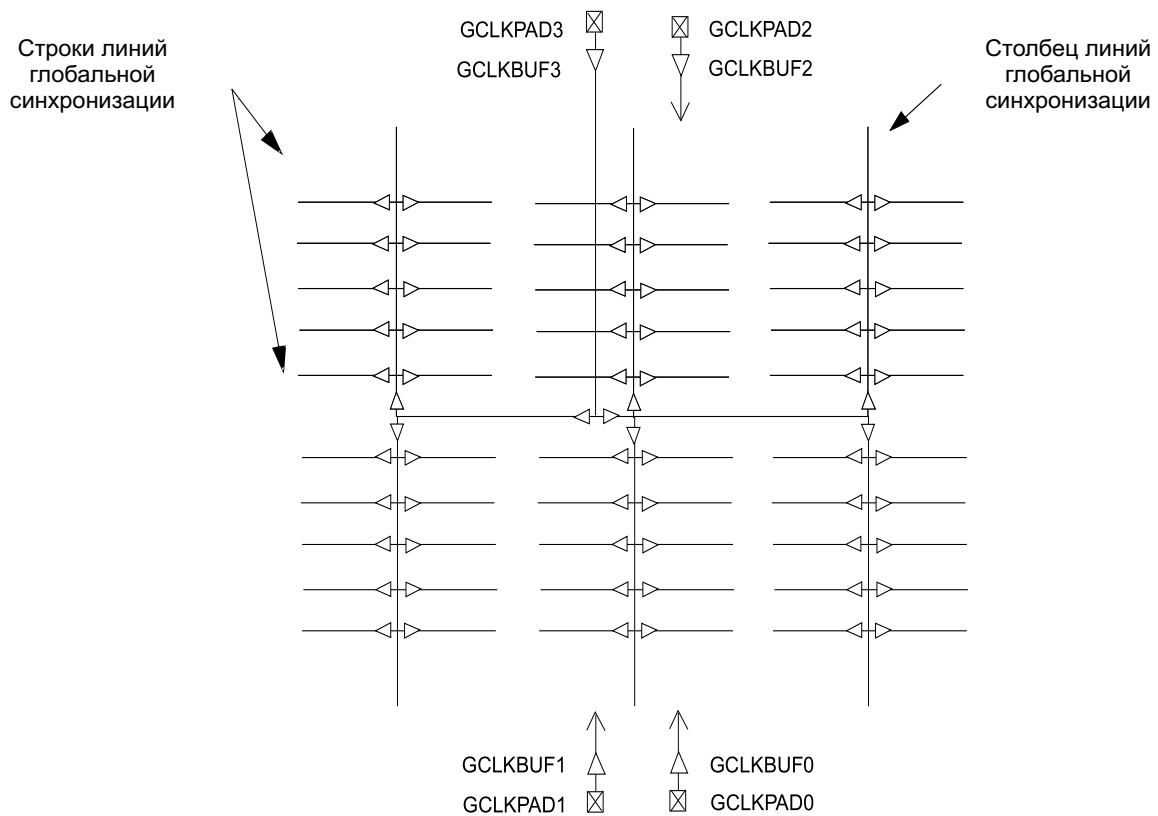


Рис. 9. Глобальные цепи синхронизации

#### Модули автоподстройки задержки (DLL)

Полностью цифровая автоподстройка задержки (DLL), связанная с каждым глобальным буфером может устранять перекося задержек между синхросигналом на входном контакте микросхемы и сигналами на тактовых входах внутренних схем устройства. Каждая DLL может быть нагружена на две глобальные цепи синхронизации. Схема DLL отслеживает сигнал синхронизации на входном контакте микросхемы и тактовый сигнал, распределяемый внутри кристалла, затем автоматически устанавливает необходимую задержку. Дополнительная задержка вводится таким образом, что фронты сигналов синхронизации достигают внутренних триггеров в точности на один период синхронизации позже их прихода на входной контакт. Эта система с обратной связью эффективно устраняет задержку распределения сигналов синхронизации, гарантируя, что фронты синхросигналов на входе микросхемы и на внутренних тактовых входах с большой точностью синхронны.

Вдобавок, для устранения задержек, возникающих при распределении тактовых сигналов, DLL создает новые возможности управления функциями синхронизации. Модуль DLL может создавать четыре квадратурные фазы из исходного источника синхросигнала; удваивать частоту синхросигнала или делить эту частоту на 1.5, 2, 2.5, 3, 4, 5, 8 или 16.

Модуль DLL также функционирует как тактовое зеркало. Путем вывода из микросхемы сигнала с выхода DLL и последующего ввода этого сигнала снова внутрь кристалла, схема DLL может устранить разбег фаз для тактовых сигналов на уровне

печатной платы, при работе с несколькими устройствами Virtex.

Чтобы гарантировать, что системная синхронизация будет нормально функционировать до момента окончания конфигурирования системы и начала штатной работы, схема DLL имеет возможность задерживать процесс конфигурирования до нормальной синхронизации с системой.

#### Периферийное сканирование (ПС)

MC Virtex поддерживают команды периферийного сканирования, приведенные в спецификации стандарта IEEE 1149.1. Порт Test Access Port (TAP) и регистры реализованы для выполнения команд EXTTEST, INTEST, SAMPLE/PRELOAD, BYPASS, IDCODE, USERCODE и HIGHZ. Кроме того, порт TAP поддерживает две внутренние сканирующие цепочки и позволяет загрузить/считать конфигурацию кристалла.

TAP использует predetermined контакты микросхемы и LVTTTL стандарт сигналов. Для того чтобы выход TDO выдавал сигналы в стандарте LVTTTL, на контакт  $V_{CC0}$  второго банка должно быть подано напряжение 3.3 В. В противном случае напряжение на выходе TDO будет меняться в пределах от нуля до  $V_{CC0}$ .

Операции периферийного сканирования не зависят от конкретных конфигураций блоков ввода/вывода и типа корпуса. Все блоки ввода/вывода, включая неподключенные к контактам, рассматриваются как независимые двунаправленные контакты с тремя состояниями, в единой цепочке сканирования. Со-

хранение возможности осуществлять двунаправленное тестирование после конфигурирования, облегчает тестирование внешних межсоединений.

В Табл. 7 приведены команды периферийного сканирования, поддерживаемые кристаллами Virtex. Внутренние сигналы могут быть проанализированы

в процессе выполнения команды EXTEST посредством подключения их к неиспользуемым выходам блоков ввода/вывода, либо к блокам ввода/вывода, не присоединенным к контактам. Они могут быть также подсоединены к неиспользуемым выходам блоков ввода/вывода, которые определены как однопользовательные входные контакты.

Табл. 7. Инструкции периферийного сканирования

Команда	Двоичный код(4:0)	Описание
EXTEST	00000	Разрешает операцию периферийного сканирования EXTEST
SAMPLE/PRELOAD	00001	Разрешает операцию периферийного сканирования SAMPLE/PRELOAD
USER1	00010	Доступ к определенному пользователем регистру 1
USER2	00011	Доступ к определенному пользователем регистру 2
CFG_OUT	00100	Доступ к конфигурационной шине для операций считывания
CFG_IN	00101	Доступ к конфигурационной шине для операций записи
INTEST	00111	Разрешает операцию периферийного сканирования INTEST
USERCODE	01000	Разрешает считывание пользовательского кода
IDCODE	01001	Разрешает считывание ID кода
HIGHZ	01010	Переводит выходы в третье состояние во время операции BYPASS
JSTART	01100	Активизирует вход TCK порта TAP
BYPASS	11111	Разрешает BYPASS
RESERVED	Любой другой	Зарезервированные инструкции

До конфигурации кристалла доступны все команды кроме USER1 и USER2. После конфигурации кристалла доступны все команды без исключения. Во время конфигурации не рекомендуется использовать команды EXTEST, INTEST и SAMPLE/PRELOAD.

В дополнение к описанным выше тестовым командам поддерживаются команды, позволяющие загрузить/считать конфигурацию кристалла.

На Рис. 10 показана логика периферийного сканирования кристаллов серии Virtex. Логика периферийного сканирования состоит из 3-х разрядного регистра данных на один БВВ, контроллера порта TAP и регистра команд с декодированием.

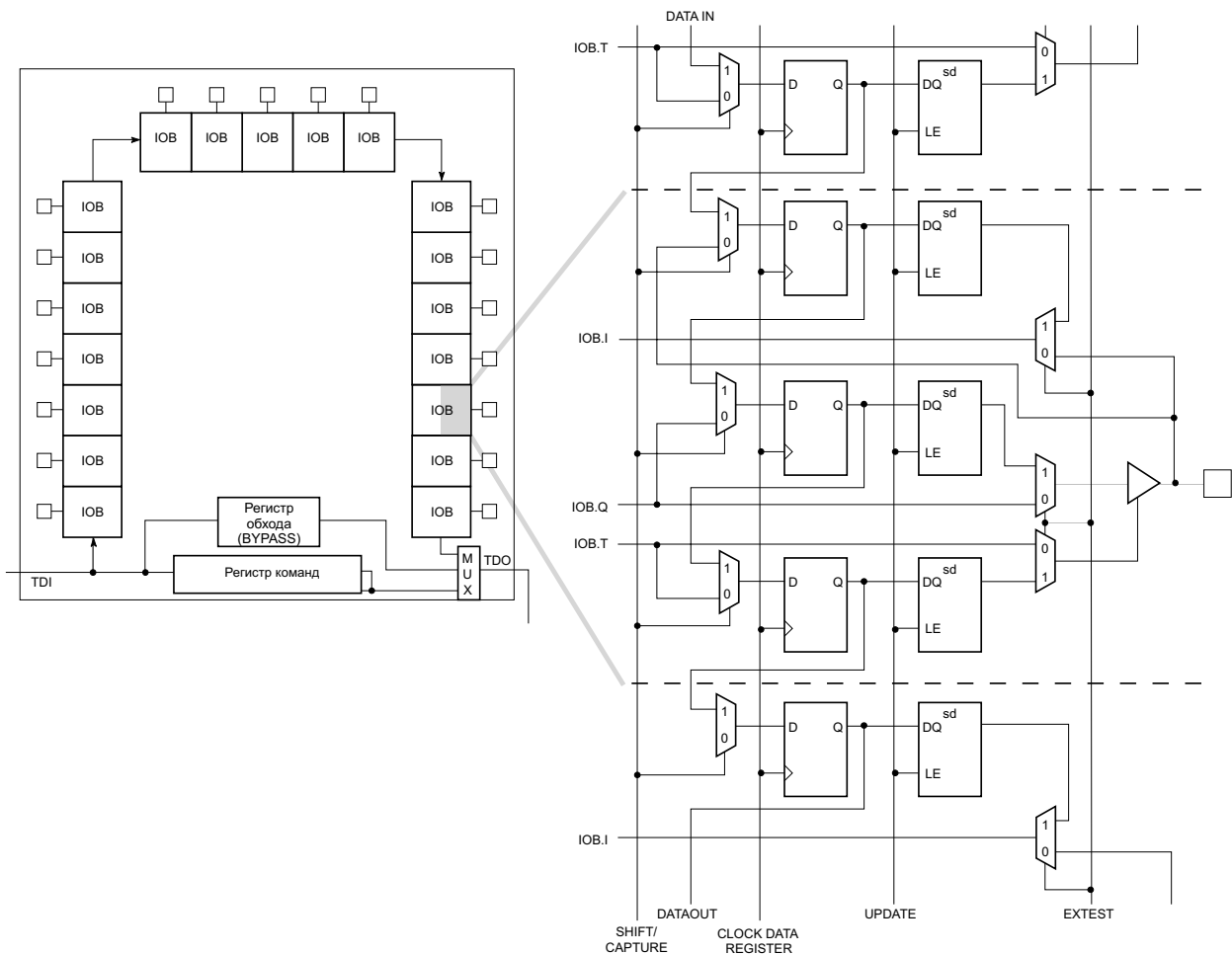


Рис. 10. Логика периферийного сканирования серии Virtex

**Регистры данных**

Первичный регистр данных является регистром периферийного сканирования. Для каждого вывода микросхемы, связанного с программируемым БВВ, регистр данных ПС содержит три разряда сдвигового регистра и три разряда регистра-защелки (для входа, выхода и управления третьим состоянием). Выводы, не доступные для программирования пользователем, имеют только по одному разряду в регистре данных ПС (для входа или выхода).

Другим регистром данных является регистр BYPASS. Данный регистр осуществляет синхронизацию данных проходящих через кристалл в следующее устройство с периферийным сканированием. В кристалле имеется только один такой регистр.

Кристалл семейства Virtex содержит две дополнительные внутренние цепи использования, которые могут быть задействованы использованием в проекте макро модуля BSCAN. Выводы SEL1 и SEL2

макро модуля BSCAN переводятся в логическую единицу при командах USER1 и USER2 соответственно, задействуя эти цепи. Данные с выхода TDO считываются входами TDO1 или TDO2 макро модуля BSCAN. Также, макро модуль BSCAN имеет отдельные тактовые входы DRCK1 и DRCK2 для каждого пользовательского регистра ПС, общий вход TDI и общие выходы RESET, SHIFT и UPDATE, отражающие состояние контроллера TAP порта.

**Порядок бит регистра данных ПС**

Порядок в каждом БВВ: Вход, Выход, Высокий импеданс. Только входные контакты представлены одним битом, а только выходные всеми тремя.

Если смотреть на кристалл, как он представлен в программном обеспечении проектирования (модуль FPGA EDITOR), то последовательность бит в регистре данных ПС будет определяться как на Рис. 11.



Рис. 11. Последовательность бит ПС

**Идентификационные регистры**

Имеются два идентификационных регистра: IDCODE-регистр и USERCODE-регистр. IDCODE позволяет определить микросхему подсоединенную к JTAG-порту.

IDCODE имеет следующий двоичный формат:

vvvv : ffff : fffa : aaaa : aaaa : cccc : cccc : ccc1

где

v - код корпуса

f - код семейства кристаллов (03h для семейства Virtex)

a - число строк матрицы КЛБ (от 010h для XCV50 до 040h для XCV1000)

c - код компании производителя (49h для Xilinx)

В Табл. 8 приведены идентификационные коды (IDCODEs) присвоенные кристаллам серии Virtex.

Используя USERCODE пользователь может записать и считать свой идентификационный номер для данного проекта. Пользовательский идентификационный код включается в конфигурационный файл во время его создания. USERCODE может быть считан только после конфигурации кристалла.

**Включение ПС в проект**

Так как все контакты необходимые для ПС предопределены в каждом кристалле, то не нужно включать в проект дополнительные элементы, если не будут использоваться пользовательские регистры (USER1 и USER2). Для задействования этих реги-

стров в проект необходимо включить элемент BSCAN и соединить соответствующие выводы.

Табл. 8. Идентификационные коды (IDCODEs) присвоенные кристаллам серии Virtex

Кристалл	IDCODE
XCV50	v0610093h
XCV100	v0614093h
XCV150	v0618093h
XCV200	v061C093h
XCV300	v0620093h
XCV400	v0628093h
XCV600	v0630093h
XCV800	v0638093h
XCV1000	v0640093h

**Система проектирования**

Разработка кристаллов Virtex осуществляется программным обеспечением проектирования Xilinx Foundation и/или Xilinx Alliance. Процесс проектирования включает: ввод проекта, размещение в кристалл и верификацию. Для ввода проекта могут применяться стандартные электронные САПР, таких фирм как Aldec, Cadence, Exemplar, Simplicity, Mentor Graphics или Synopsys. Для размещения в кристалл и верификации используются специализированные под архитектуру САПР, выпускаемые только Xilinx.

Система проектирования фирмы Xilinx интегрирована в управляющую программу называемую Xilinx Design Manager (XDM), которая обеспечивает доступ к общему пользовательскому интерфейсу, не-

зависимо от выбора вида программы ввода или верификации. Программа XDM упрощает выбор настроек, необходимых для выполнения проекта, благодаря наличию разветвленного меню и ) и легко доступной справочной системе(on-line help).

Прикладные программы, начиная от создания схемы (schematic capture), до размещения и трассировки (Placement and Routing - PAR), доступны из программы XDM. Цепочка команд, определяющих последовательность обрабатываемых процессов, генерируется до начала их исполнения и запоминается для последующего документирования.

Несколько расширенных свойств программного обеспечения облегчает проектирование микросхем VIRTEX. Например, схемные относительно расположенные макросы (Relationally Placed Macros – RPMs), в которых содержится информация о принудительной взаимной ориентации составных частей элементов проекта, дают необходимую информацию для их реального размещения на кристалле. Они помогают обеспечить оптимальное выполнение стандартных логических функций.

Для ввода проектов с помощью языков описания аппаратных средств (Hardware Description Language – HDL), система проектирования Xilinx Foundation предоставляет интерфейсы к синтезаторам следующих фирм:

- Synopsis (FPGA Compiler, FPGA Express);
- Exemplar (Spectrum);
- Symplicity (Simplify);

Для схемного ввода проектов, системы проектирования Xilinx Foundation и Alliance предоставляют интерфейсы к следующим системам создания схем:

- Mentor Graphics V8 (Design Architect Quick Sim II);
- Innoveda (Viewdraw).

Существует множество других производителей, которые предлагают аналогичные по функциям системы ввода проекта.

Для упрощения взаимодействия различных САПР существует стандартный формат файлов (EDIF), который поддерживается всеми производителями САПР.

САПРы для Virtex включают унифицированную библиотеку стандартных функций. Эта библиотека содержит свыше 400 примитивов и макросов, от двухходовых вентилях И, до 16 битовых аккумуляторов и включает арифметические функции, компараторы, счетчики, регистры данных, дешифраторы, шифраторы, функции ввода/вывода, защелки, булевы функции, мультиплексоры и сдвигающие регистры.

Часть библиотеки, содержащей детальные описания общих логических функций, реализованных в виде «нежестких» макросов (soft macro), в тоже время, не содержит никакой информации о разбиении этих функций на реальные физические блоки и об их размещении в кристалле. Быстродействие данных макросов зависит, таким образом, от этих двух процедур, которые реализуются на этапе раз-

мещения проекта в кристалл. В тоже время, относительно расположенные макросы (RPMs) содержат в себе предварительно определенную информацию о разбиении на физические блоки и о размещении, которая дает возможность для оптимального выполнения этих функций. Пользователи могут создать свою собственную библиотеку «нежестких» макросов и RPM из примитивов и макросов стандартной библиотеки.

Среда проектирования поддерживает ввод иерархических проектов, в которых схемы верхнего уровня содержат основные функциональные блоки, в то время, как системы нижнего уровня определяют логические функции этих блоков. Данные элементы иерархического проекта автоматически объединяются соответствующими средствами на этапе размещения в кристалл. При иерархической реализации могут объединяться различные средства ввода проекта, давая возможность каждую из частей вводить наиболее подходящим для нее методом.

## Размещение проекта в кристалл

Программное средство размещения и трассировки (place – and – route, PAR) обеспечивает автоматическое протекание процесса размещения проекта в кристалл, которое описывается ниже. Процедура разбиения на физические блоки получает исходную информацию о проекте в виде перечня связей формата EDIF и осуществляет привязку абстрактных логических элементов к реальным физическим ресурсам архитектуры FPGA (БВВ, КЛБ). Затем процедура размещения определяет наилучшее место для их размещения, руководствуясь информацией о межсоединениях и желаемом быстродействии. В завершении, процедура трассировки выполняет соединения между блоками.

Алгоритмы программы PAR поддерживают автоматическое выполнение большинства проектов. Тем не менее, в некоторых приложениях пользователь, при необходимости, может осуществлять контроль и управление процессом. На этапе ввода проекта пользователь может задавать свою информацию для разбиения, размещения и трассировки.

В программное обеспечение встроено средство Timing Wizard, управляющее процессом размещения и трассировки с учетом требований к временам распространения сигналов. При вводе проекта пользователь задает эту информацию в виде временных ограничений для определенных цепей. Процедуры анализа временных параметров связей анализируют эти, заданные пользователем, требования и пытаются удовлетворить им.

Временные требования вводятся в схему в виде непосредственных системных ограничений, таких как минимально допустимая частота синхронизации, или максимально допустимая задержка между двумя регистрами. При таком подходе результирующее быстродействие системы с учетом суммарной протяженности путей автоматически подгоняется под требования пользователя. Таким образом, задание временных ограничений для отдельных цепей становится не нужным.

## Верификация проекта

В дополнение к обычному программному моделированию FPGA, пользователь может использовать метод непосредственной отладки реальных цепей. Благодаря неограниченному количеству циклов перепрограммирования кристаллов FPGA, работоспособность проектов можно проверить в реальном масштабе времени, вместо того, чтобы использовать большой набор тестовых векторов, необходимых при программном моделировании.

Система проектирования устройств Virtex поддерживает и программное моделирование и метод отладки непосредственно аппаратных цепей. Для выполнения моделирования система извлекает временную информацию, полученную после размещения из базы данных проекта, и вводит ее в сетевой перечень. Пользователь может и сам проверить критичные по времени части проекта, используя статический временной анализатор TRACE.

Для непосредственной отладки цепей к системе проектирования поставляется с кабель для загрузки конфигурационных данных и обратного считывания данных из микросхемы. Этот кабель соединяет персональный компьютер или рабочую станцию с микросхемой FPGA, установленной в законченное устройство. После загрузки проекта в FPGA, пользователь может выполнить один шаг изменения логического состояния схемы, затем выполнить обратное считывание состояния триггеров в компьютер и проанализировать правильность работы схемы. Простейшие модификации проекта, при этом, можно осуществлять в считанные минуты.

## Конфигурирование кристалла в устройстве

MC Virtex конфигурируются путем загрузки конфигурационных данных во внутреннюю конфигурационную память. Часть специальных контактов, которые при этом используются, не могут применяться для других целей, в то же время некоторые из них могут после завершения конфигурирования слу-

жить в качестве контактов ввода-вывода общего назначения.

К специальным контактам конфигурирования относятся следующие:

- контакты режима конфигурирования (M2, M1, M0);
- контакт синхронизации процесса конфигурирования (CCLK);
- контакт  $\overline{\text{INIT}}$  ;
- контакт DONE;
- контакты порта периферийного сканирования (TDI, TDO, TMS, TCK).

В зависимости от выбранного режима конфигурирования, контакт CCLK может быть либо источником сигнала синхронизации, либо наоборот – приемником сигнала от внешнего генератора синхросигналов.

## Режимы конфигурирования

Virtex поддерживает следующие четыре режима конфигурирования:

- подчиненный последовательный режим (Slave-serial)
- ведущий последовательный режим (Master-serial)
- режим SelectMap
- режим периферийного сканирования (Boundary Scan - JTAG)

Комбинация кодов на специальных входных контактах (M2, M1, M0) позволяет выбрать один из режимов конфигурирования, при этом четыре из восьми кодов соответствуют “подтянутому” (pull-up) состоянию входов блоков ввода-вывода до начала процедуры конфигурирования, и еще четыре комбинации состоянию неопределенного потенциала блоков ввода/вывода. Соответствие этих кодов необходимому режиму приведено в Табл. 9.

Табл. 9. Конфигурационные коды

Режим	M2	M1	M0	CCLK	Разрядность данных	Последовательный выход DOUT	Контакты подтянуты
Master-serial	0	0	0	Выход	1	Есть	Нет
Boundary-scan	1	0	1		1	Нет	Нет
SelectMAP	1	1	0	Вход	8	Нет	Нет
Slave-serial	1	1	1	Вход	1	Есть	Нет
Master-serial	1	0	0	Выход	1	Есть	Да
Boundary-scan	0	0	1		1	Нет	Да
SelectMAP	0	1	0	Вход	8	Нет	Да
Slave-serial	0	1	1	Вход	1	Есть	Да



Конфигурирование микросхемы FPGA через порт периферийного сканирования доступно всегда, независимо от значения этого кода. Задание кода отключает другие режимы. Все три контакта режима конфигурирования имеют внутренние pull-up резисторы и по умолчанию задают, таким образом, состояния высокого логического уровня, если отсутствуют внешние подключения.

**Подчиненный последовательный режим**

В этом режиме FPGA принимает конфигурационные данные в последовательной форме от последовательного ПЗУ (Serial Programmable Read Only Memory – SPROM) или от другого источника последовательных конфигурационных данных. Данные последовательного битового потока (bitstream) должны быть установлены на входе DIN незадолго до появления нарастающего фронта сигнала, генерируемого внешним источником и подаваемого на вход CCLK.

Несколько микросхем FPGA могут быть соединены в цепочку для конфигурирования от единого внешнего источника конфигурационных данных. После того, как одна из микросхем сконфигурирована, данные для следующей появляются на выходе DOUT. Изменение данных на выходе DOUT происходит после нарастающего фронта сигнала на входе CCLK.

Процесс стробирования данных, подаваемых на вход DIN по нарастающему фронту CCLK, отлича-

ется от аналогичного процесса в старых семействах микросхем FPGA, но это не приводит к возникновению проблем для смешанных конфигурационных цепочек. Такое изменение сделано для увеличения скоростей последовательного конфигурирования цепочек FPGA, состоящих только из микросхем Virtex.

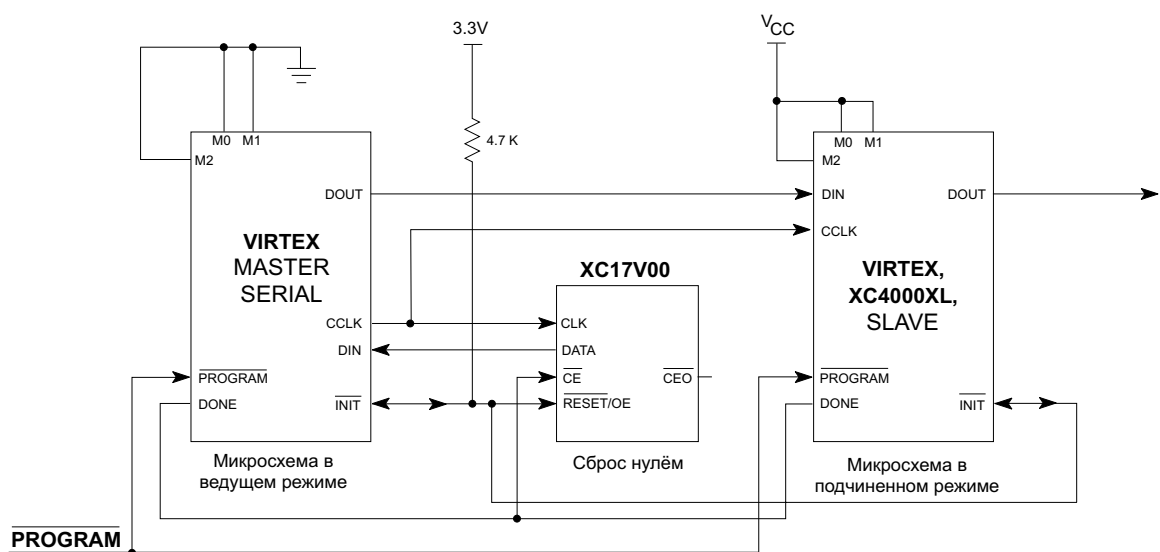
На Рис. 12 изображена полная схема, совмещающая подчиненные режимы и ведущий режим. FPGA Virtex, конфигурируемые в подчиненном режиме, должны быть подключены также, как устройство, изображенное третьим слева.

Подчиненный последовательный режим выбирается заданием кода <111> на входах режима конфигурирования (M2, M1, M0). Внутренние высокоомные резисторы на режимных контактах “подтягивают” данные входы в состояние высокого логического уровня и, таким образом, задают этот режим по умолчанию, если выходы не имеют внешних подключений. На Рис. 13 изображена временная диаграмма для данного режима.

В Табл. 10 содержится более подробная информация для величин, приведенных на Рис. 13. Для FPGA, соединенных в цепочку, процесс конфигурирования должен быть задержан до тех пор, пока на контакте INIT присутствует высокий логический уровень.

**Табл. 10: Параметры сигналов подчиненного и ведущего режимов**

Параметр	Обозначение		Значение	
			Мин.	Макс.
Предустановка/удержание входного сигнала DIN, подчиненный режим	1/2	$T_{DCC}/T_{CCD}$	5.0 нс / 0 нс	
Предустановка/удержание входного сигнала DIN, ведущий режим	1/2	$T_{DSCK}/T_{SCKD}$	5.0 нс / 0 нс	
Задержка сигнала DOUT	3	$T_{CCO}$		12 нс
Длительность высокого уровня	4	$T_{CCH}$	5.0 нс	
Длительность низкого уровня	5	$T_{CCL}$	5.0 нс	
Частота		$F_{CC}$		66 МГц



**Рис. 12. Ведущий/подчиненный режимы конфигурации**

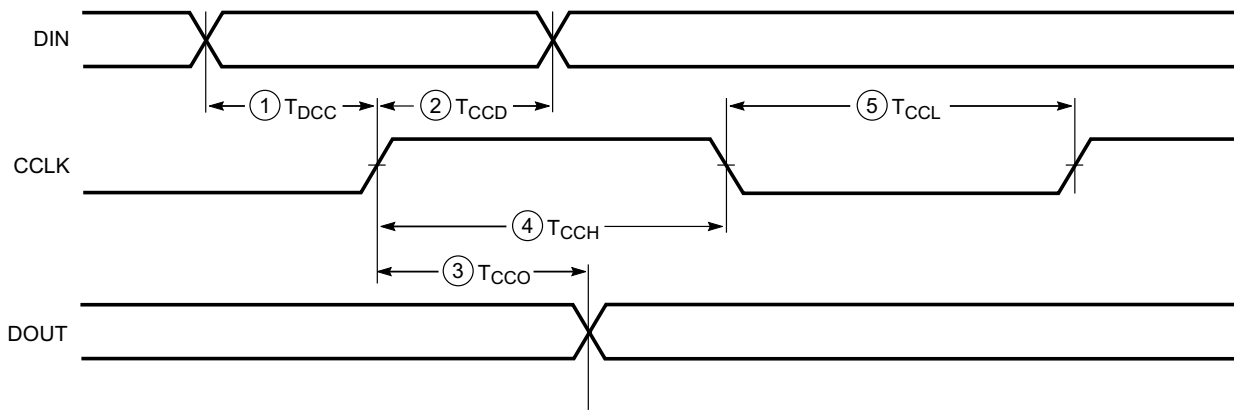


Рис. 13. Временная диаграмма подчиненного режима конфигурации

### Ведущий последовательный режим

В ведущем последовательном режиме с выхода CCLK FPGA сигнал подается на соответствующий вход микросхемы ПЗУ, которая передает данные на DIN-вход той же микросхемы FPGA. Прием данных в FPGA осуществляется по каждому нарастающему фронту сигнала CCLK. После полного конфигурирования микросхемы, данные для следующих устройств, соединенных цепочкой, появляются на выходе DOUT после каждого нарастающего фронта сигнала CCLK. Данные конфигурирования, поступающие на все MC FPGA, соединенные в цепочку, обязательно начинаются с блока, называемого преамбулой.

Интерфейс, поддерживающий этот режим, идентичен интерфейсу подчиненного режима, за исключением того, что для генерации синхросигнала конфигурирования используется внутренний осциллятор FPGA. Частота для этого синхросигнала может быть выбрана из широкого диапазона значений, но по умолчанию всегда используется низкая частота. Переключение на более высокую частоту происходит данными, которые распознаются микросхемой в самом конфигурационном потоке, после чего оставшаяся часть потока загружается уже с новой скоростью. Переключение снова на более низкую частоту запрещается. Частота синхронизации CCLK устанавливается выбором ConfigRate в программе генерации конфигурационного потока. Максимальная частота CCLK, которая может быть выбрана – 60 МГц. Выбирая конкретную частоту CCLK, необходимо убедиться, что используемые ПЗУ и все соединенные в цепочку MC FPGA рассчитаны на конфигурирование в таком темпе.

После включения питания, частота CCLK равна 2.5 МГц. Эта частота используется до момента загруз-

ки битов ConfigRate, после чего частота меняется на новое значение, определенное этими битами. Если в проекте не задается другая частота, то используемая по умолчанию частота равна 4 МГц.

На Рис. 12 показана полная система, содержащая кристалл в ведущем и кристалл в подчиненном режимах. В этой схеме крайнее левое устройство работает в ведущем последовательном режиме. Остальные устройства работают в подчиненном последовательном режиме. На вход RESET микросхемы ПЗУ подается сигнал с контактов INIT MC FPGA. Аналогично, на вход CE – с выхода DONE. При этом в зависимости от выбранной стартовой последовательности, существует конфликт потенциалов на контакте DONE.

Для последовательного конфигурирования микросхем FPGA необходимо использовать последовательность, изображенную в виде алгоритма на Рис. 14.

Временная диаграмма для ведущего последовательного режима показана на Рис. 15. Данный режим выбирается заданием кода <000> или <100> на входах M2, M1, M0. Необходимую временную информацию для этого режима содержит Табл. 10.

Время нарастания напряжения питания  $V_{CC}$  от уровня 1 В до минимально допустимого значения  $V_{CC}$  не должно превышать 50 мс, в противном случае необходимо удерживать сигнал PROGRAM в состоянии низкого логического уровня до момента достижения допустимого уровня  $V_{CC}$

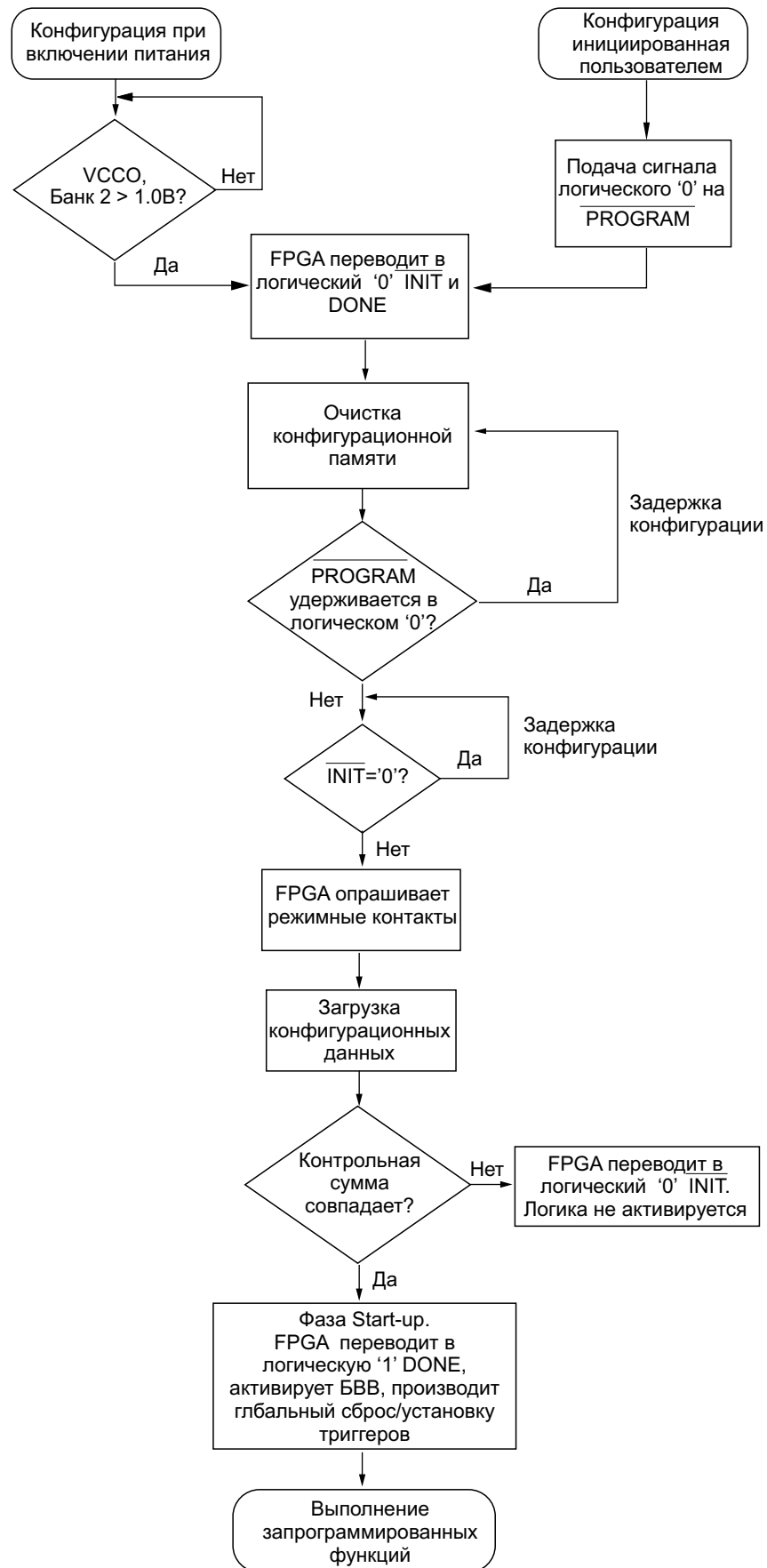


Рис. 14. Алгоритм конфигурирования MC Virtex в последовательных режимах

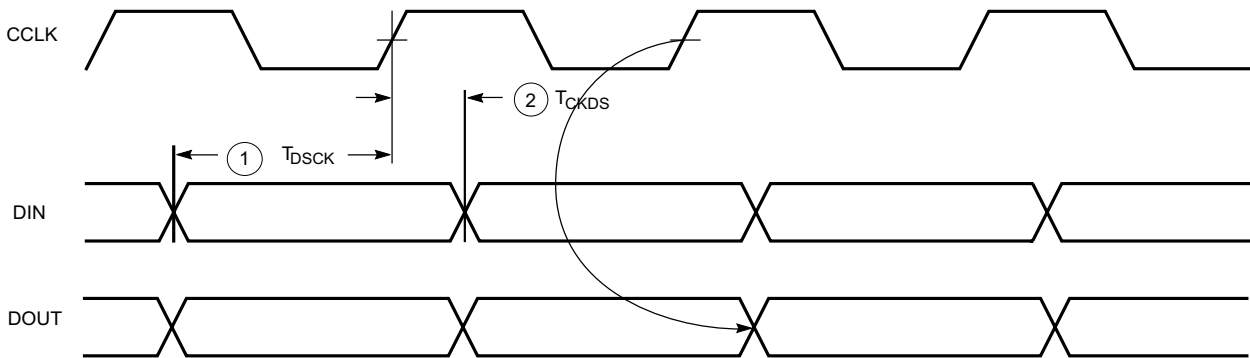


Рис. 15. Временная диаграмма ведущего режима конфигурации

**Режим SelectMAP**

SelectMAP – самый быстрый режим конфигурирования. В этом режиме данные записываются в FPGA побайтно с использованием флага BUSY, управляющего потоком данных.

Внешний источник создает байтовый поток данных и сигналы CCLK, выбор кристалла (Chip Select - CS), запись (WRITE). Если установлен высокий логический уровень сигнала BUSY, данные должны удерживаться до тех пор, пока BUSY не будет переведен в состояние низкого уровня.

Используя этот режим можно считать данные. Если сигнал WRITE не установлен (т.е. находится в состоянии высокого логического уровня), конфигурационные данные читаются обратно из FPGA, как часть операции обратного считывания.

После окончания конфигурирования контакты порта SelectMAP могут использоваться как дополнительные пользовательские входы/выходы. Можно

использовать этот порт для быстрого 8-ми битного обратного считывания конфигурационных данных.

Сохранение такой возможности после конфигурирования реализуется на этапе создания битового потока. Для сохранения такой возможности, необходимо использовать ограничения типа PROHIBIT, предохраняющие контакты порта SelectMAP от использования в качестве пользовательских.

Несколько FPGA Virtex могут конфигурироваться в режиме SelectMAP, и далее одновременно запускаться для штатного функционирования. Для конфигурирования нескольких устройств таким способом, необходимо соединить параллельно индивидуальные сигналы отдельных микросхем CCLK, Data, WRITE и BUSY. Конкретные микросхемы конфигурируются по очереди за счет поочередной подачи активного сигнала на контакт выборки (CS) этой FPGA и записи соответствующих ей данных. В Табл. 11 представлены временные параметры сигналов режима SelectMAP

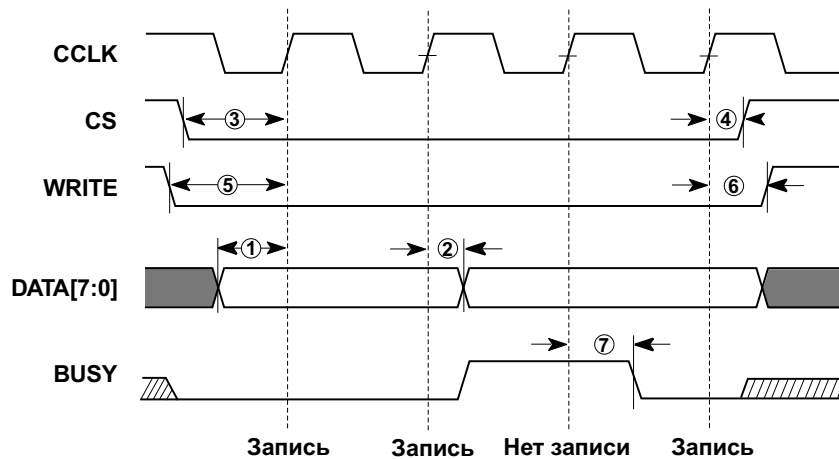
Табл. 11: Параметры сигналов режима SelectMAP

Параметр	Обозначение	Значение	
		Min	Max
Предустановка/удержание входных сигналов D0-D7	1/2 $T_{SMDC}/T_{SMCCD}$	5.0 нс	0 нс
Предустановка/удержание входного сигнала CS	3/4 $T_{SMCSC}/T_{SMCCS}$	7.0 нс	0 нс
Предустановка/удержание входного сигнала WRITE	5/6 $T_{SMCCW}/T_{SMWCC}$	7.0 нс	0 нс
Задержка распространения сигнала BUSY	7 $T_{SMCKBY}$		12.0 нс
Частота	$F_{CC}$		66 МГц
Частота без подтверждения получения данных	$F_{CCNH}$		50 МГц

**Запись**

Процедура записи посылает пакеты конфигурационных данных в FPGA. Необходимо отметить, что конфигурационный пакет можно расщепить на не-

сколько таких последовательностей. Пакет не должен быть закончен за время одной активизации сигнала CS, изображенной на Рис. 16.



**Рис. 16. Операции записи**

Последовательность операций:

1. Установить сигналы  $\overline{\text{WRITE}}$  и  $\overline{\text{CS}}$  в состояние низкого логического уровня. Отметим, что если сигнал  $\overline{\text{CS}}$  активизируется во время уже функционирующего сигнала  $\text{CCLK}$ ,  $\overline{\text{WRITE}}$  должен оставаться неизменным. В противном случае, как описано далее, будет инициировано преждевременное прекращение процедуры.
2. Данные подать на вход  $\text{D}[7:0]$ . Отметим, что для избежания конфликта между данными от различных источников, информация не должна выдаваться во время, когда сигнал  $\overline{\text{CS}}$  имеет значение '0', а  $\overline{\text{WRITE}}$  – значение '1'. Также нельзя активизировать больше одного  $\overline{\text{CS}}$  в то время, когда сигнал  $\overline{\text{WRITE}}$  имеет значение '1'.
3. Данные принимаются по нарастающему фронту  $\text{CCLK}$  при условии, что сигнал  $\text{BUSY}$ , при

этом, имеет значение '0'. В случае если сигнал  $\text{BUSY}$  имел значение '1' от предыдущей записи, данные не принимаются. Данные снова будут приниматься по первому же нарастающему фронту  $\text{CCLK}$  после перехода  $\text{BUSY}$  в состояние '0'; при этом данные должны удерживаться до этого события.

4. Повторять шаги 2 и 3 до тех пор, пока будут переданы все данные.
5. Перевести сигналы  $\overline{\text{CS}}$  и  $\overline{\text{WRITE}}$  в неактивное состояние.

Алгоритм процедуры записи показан на Рис. 17. Отметим, что если сигнал  $\text{CCLK}$  медленнее, чем  $F_{\text{CCNH}}$ , FPGA не будет выставлять сигнал  $\text{BUSY}$ , в этом случае обмен подтверждениями готовности после реального приема данных не нужен, и данные могут просто вводиться в FPGA по каждому циклу сигнала  $\text{CCLK}$ .

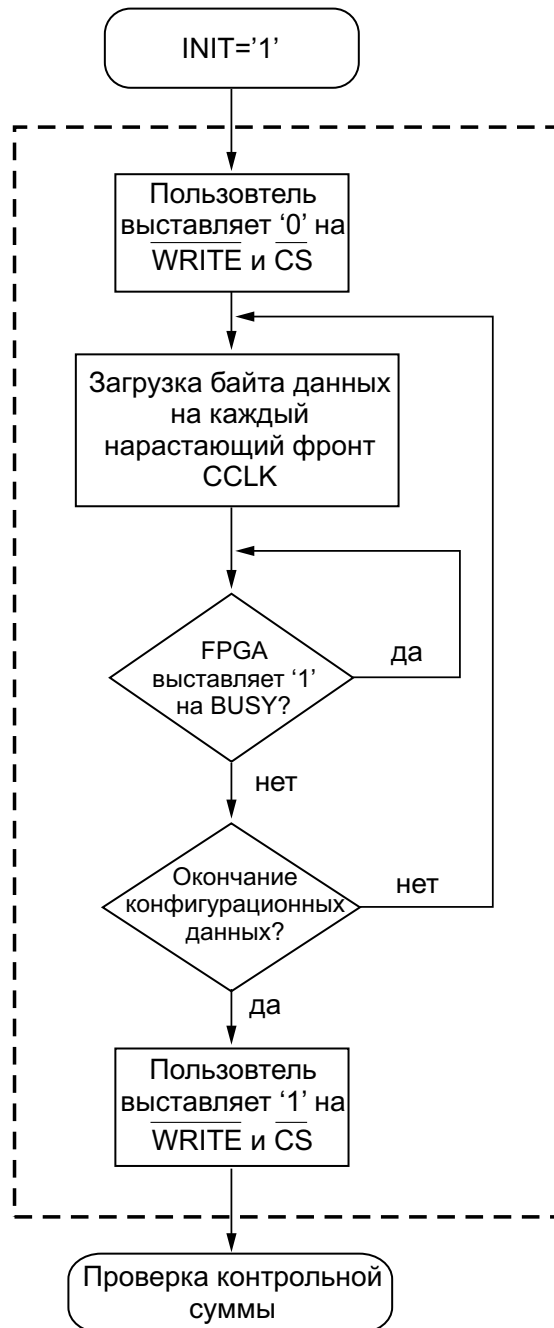


Рис. 17. Алгоритм процедуры записи в режиме SelectMAP

#### Преждевременное прекращение процедуры

После установки активного уровня сигнала CS, пользователь не может переключаться с записи на чтение или наоборот. В противном случае такое действие приведет к преждевременному прекращению текущей пакетной команды. Устройство будет оставаться в состоянии BUSY (ЗАНЯТО) до тех пор, пока прерванная процедура будет завершена. После прекращения процедуры, для продолжения

приема пакета, необходимо повторить передачу того слова, которое было прервано не на границе слова.

Для того, чтобы инициировать преждевременное прекращение процедуры записи, необходимо перевести сигнал WRITE в неактивное состояние. Как показано на Рис. 18, прекращение процедуры начнется с приходом нарастающего фронта CCLK

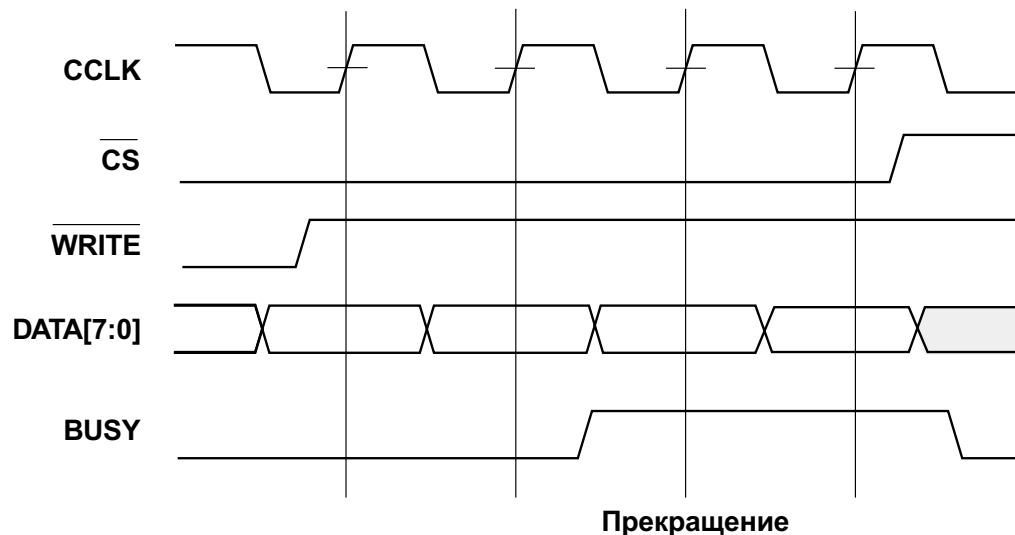


Рис. 18. Преждевременное прекращение процедуры записи при режиме SelectMAP

#### Использование режима периферийного сканирования для конфигурирования Virtex

Для конфигурирования в режиме периферийного сканирования используются только специальные контакты порта тестового доступа (Test Access Port - TAP) в соответствии со стандартом IEEE1149.1.

Конфигурирование через порт TAP выполняется с помощью специальной команды CFG\_IN. Эта команда позволяет преобразовать входные данные, поступающие на вход TDI в пакет данных для внутренней шины конфигурирования.

Для конфигурирования FPGA через порт периферийного сканирования необходимо выполнить следующие действия:

1. Загрузить команду CFG\_IN во внутренний регистр команд (instruction register -IR).
2. Ввести состояние Shift-DR (SDR).
3. Выдать стандартный конфигурационный bit-stream на TDI.
4. Возвратиться к состоянию Run-Test-Idle (RTI).
5. Загрузить в регистр IR команду JSTART.
6. Ввести состояние SDR.
7. Выдать TCK для длины последовательности (длина – программируемая).
8. Возвратиться к состоянию RTI.

Как отмечалось ранее, конфигурирование и обратное считывание всегда доступно в режиме периферийного сканирования. Для выборки режима необходимо подать код <101> или <001> на контакты M2, M1, M0.

#### Последовательность конфигурации

Конфигурирование устройств Virtex – процесс, состоящий из трех фаз. В первой фазе конфигурирования очищается память. Следующая фаза – загрузка данных в конфигурационную память. Наконец, активизируется логика (фаза Start-Up).

Обычно процесс конфигурирования запускается автоматически после подачи напряжения питания, однако, как будет описано далее, он может быть задержан пользователем. Конфигурационный процесс может также быть инициирован установкой активного уровня сигнала PROGRAM. Переход

сигнала INIT в состояние '1' означает окончание фазы очистки памяти, а установка активного уровня сигнала DONE ('1') означает окончание процесса в целом.

Временная диаграмма для конфигурационных сигналов после подачи напряжения питания показана на Рис. 19, а соответствующие временные характеристики – в Табл. 12.

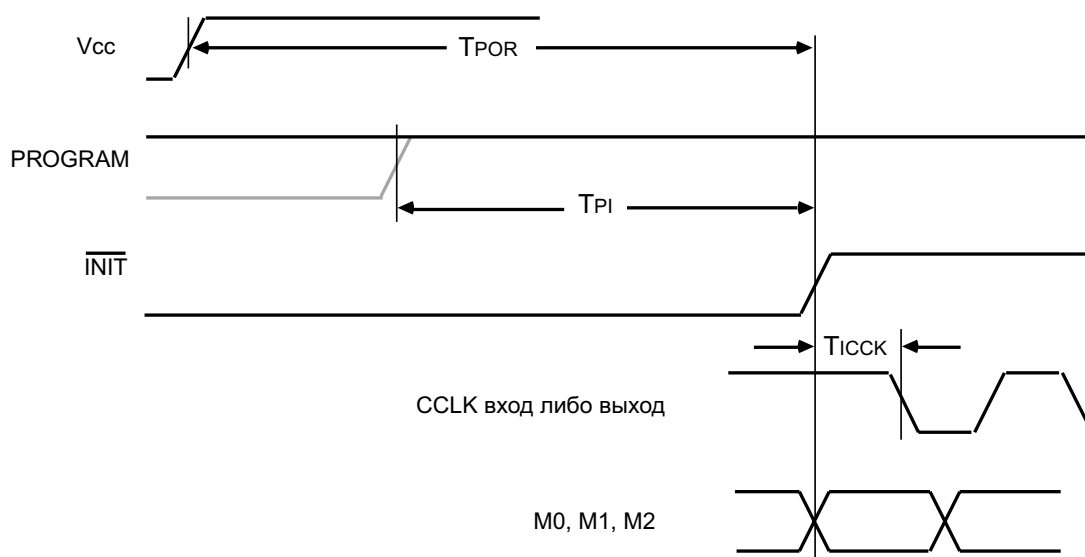


Рис. 19. Временные диаграммы при подаче питания

Табл. 12. Значения временных параметров при подаче питания

Параметр	Значение	
	Мин.	Макс.
T <sub>POR</sub>		2.0 мс
T <sub>PL</sub>		100 мкс
T <sub>ICCK</sub>	0.5 мкс	4.0 мкс
T <sub>PROGRAMM</sub>	300 нс	

### Задержка конфигурирования

Конфигурирование FPGA может быть задержано удержанием сигнала на контакте PROGRAM в состоянии низкого логического уровня до момента готовности системы к конфигурированию. На протяжении фазы очистки конфигурационной памяти последовательность операций состоит из повторения цикла очистки памяти по всем адресам. Эти операции продолжаются до окончания одного полного цикла очистки памяти по всем адресам после установки сигнала на входе PROGRAM в состояние '1'. Таким образом, задержка процесса конфигурирования равнозначна продолжению фазы очистки памяти.

Другой вариант – подача от источника с открытым стоком сигнала низкого уровня на вход INIT. Источник сигнала с открытым стоком необходим потому, что контакт INIT – двунаправленный и работает как выход, имеющий низкий логический уровень во время фазы очистки памяти. Увеличение времени удержания низкого логического уровня на этом контакте приводит к тому, что конфигурационный автомат продолжает выполнять фазу очистки памяти. Таким образом, процесс конфигурирования задерживается, не входя в фазу загрузки данных.

### Последовательность вхождения в штатный режим работы

При выполнении вхождения в штатный режим работы по умолчанию, глобальный сигнал управления третьим состоянием (global tristate – GTS) активизируется через один цикл CCLK после перехода

сигнала DONE в состояние '1'. Это позволяет выходам FPGA включиться надлежащим образом.

Одним циклом CCLK позже, активизируются сигнал глобальной установки/сброса (Global Set/Reset – GSR) и глобального разрешения записи (Global Write Enable – GWE). Это создает условия для начала нормальной работы внутренних запоминающих элементов.

Временная диаграмма для этих событий может быть изменена. Кроме того, события GTS, GSR и GWE могут активизироваться после перехода всех выходов DONE в состояние High при конфигурировании множественных устройств FPGA, что позволяет начинать их работу в штатном режиме синхронно. Во время выполнения последовательности допускается включение на любой фазе паузы до момента нормального захвата следящей системы схемы автоподстройки задержки (DLL).

### Формат потока конфигурационных данных

Кристаллы Virtex конфигурируются последовательной загрузкой в них фреймов данных, которые объединены в двоичный поток (bitstream). В Табл. 13 представлены объемы конфигурационной последовательности для кристаллов Virtex.



**Табл. 13. Размер конфигурационной последовательности для различных МС семейства Virtex**

Кристалл	Конфигурационных Бит
XCV50	559,200
XCV100	781,216
XCV150	1,040,096
XCV200	1,335,840
XCV300	1,751,808
XCV400	2,546,048
XCV600	3,607,968
XCV800	4,715,616
XCV1000	6,127,744

## Обратное считывание

Конфигурационные данные, записанные в конфигурационной памяти FPGA, могут быть считаны обратно для выполнения верификации. Наряду с этими данными возможно обратное считывание содержимого всех триггеров/защелок, LUTRAMs, BlockRAMs. Эта возможность используется для выполнения отладки проектов в реальном масштабе времени.

## Характеристики семейства Virtex по постоянному току

В Табл. 14 приведены максимально допустимые значения параметров МС семейства Virtex по постоянному току

**Внимание: Превышение максимальных значений ведет к повреждению кристалла.**

**Табл. 14. Диапазон максимально допустимых значений МС семейства Virtex**

Обозначение	Описание	Значения	Единицы измерения	
$V_{CCINT}$	Напряжение питания ядра относительно GND	от -0.5 до 3.0	В	
$V_{CCO}$	Напряжение питания выходных каскадов относительно GND	от -0.5 до 4.0	В	
$V_{REF}$	Входное опорное напряжение	от -0.5 до 3.6	В	
$V_{IN}$	Напряжение входного сигнала относительно GND	Используя $V_{REF}$	от -0.5 до 3.6	В
		Не используя $V_{REF}$	от -0.5 до 5.5	В
$V_{TS}$	Напряжение, прикладываемое к 3-х стабильному выходу	от -0.5 до 5.5	В	
$V_{CC}$	Максимальное время нарастания напряжения питания от 1В до 2.375В	50	мс	
$T_{STG}$	Температура хранения (окружающей среды)	от -65 до +150	°C	
$T_{SOL}$	Максимальная температура припоя	+260	°C	
$T_J$	Максимальная рабочая температура контактов	+125	°C	

В Табл. 15 приведены рекомендуемые значения параметров МС семейства Virtex по постоянному току

**Табл. 15. Рекомендуемые значения**

Обозначение	Описание	Мин.	Макс.	Единицы измерения
$V_{CCINT}$	Напряжение питания ядра при $T_J =$ от 0°C до +85°C (Коммерческое исполнение)	2.5-5%	2.5+5%	В
	Напряжение питания ядра при $T_J =$ от 0°C до +85°C (Промышленное исполнение)	2.5-5%	2.5+5%	В
$V_{CCO}$	Напряжение питания выходных каскадов при $T_J =$ от 0°C до +85°C (Коммерческое исполнение)	1.2	3.6	В
	Напряжение питания выходных каскадов при $T_J =$ от 0°C до +85°C (Промышленное исполнение)	1.2	3.6	В
$T_{IN}$	Время передачи входного сигнала		250	нс

## Корпуса

В Табл. 16 приведены комбинации: кристалл Virtex – корпус, и число пользовательских выводов для каждой комбинации.

Табл. 16. Корпуса

Корпус	Максимальное число пользовательских контактов								
	XCV50	XCV100	XCV150	XCV200	XCV300	XCV400	XCV600	XCV800	XCV1000
CS-144	94	94							
TQ-144	98	98							
PQ-240	166	166	166	166	166				
HQ-240						166	166	166	
BG-256	180	180	180	180					
BG-352			260	260	260				
BG-432					316	316	316	316	
BG-560						404	404	404	404
FG-256	176	176	176	176					
FG-456			260	284	312				
FG-676						404	444	444	
FG-680							512	512	512

## Обозначение MC семейства Virtex

Способ обозначения MC семейства Virtex показан на Рис. 20.

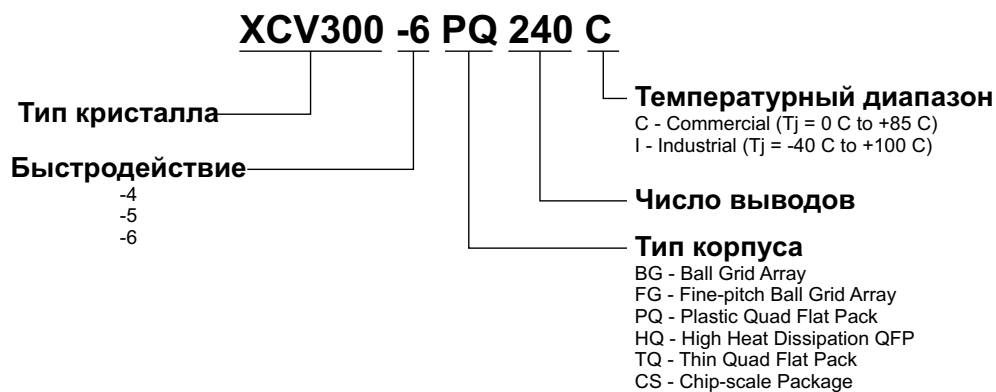


Рис. 20. Обозначение MC семейства Virtex

## Содержание

<b>ОСОБЕННОСТИ</b> .....	<b>1</b>
<b>ОПИСАНИЕ</b> .....	<b>1</b>
<b>АРХИТЕКТУРА VIRTEX</b> .....	<b>2</b>
Быстродействие .....	2
<b>ОПИСАНИЕ АРХИТЕКТУРЫ</b> .....	<b>2</b>
Матрица Virtex .....	2
<b>Блок ввода-вывода</b> .....	<b>3</b>
Ввод сигнала .....	4
Вывод сигнала .....	5
Банки ввода-вывода .....	5
<b>Конфигурируемый логический блок - КЛБ</b> .....	<b>6</b>
Таблица Преобразования .....	7
Запоминающие элементы .....	7
Дополнительная логика .....	7
Арифметическая логика .....	8
Буферы с тремя состояниями .....	8
Блочная память (Block RAM) .....	8
<b>Программируемая трассировочная матрица</b> .....	<b>8</b>
Локальные связи .....	9
Трассировочные ресурсы общего назначения .....	9
Трассировочные ресурсы для блоков ввода-вывода .....	9
Специальные трассировочные ресурсы .....	10
Глобальные трассировочные ресурсы .....	10
<b>Распределение сигналов синхронизации</b> .....	<b>10</b>
Модули автоподстройки задержки (DLL) .....	11
<b>Периферийное сканирование (ПС)</b> .....	<b>11</b>
Регистры данных .....	13
Порядок бит регистра данных ПС .....	13
Идентификационные регистры .....	14
Включение ПС в проект .....	14
<b>СИСТЕМА ПРОЕКТИРОВАНИЯ</b> .....	<b>14</b>
Размещение проекта в кристалл .....	15
Верификация проекта .....	16
<b>КОНФИГУРИРОВАНИЕ КРИСТАЛЛА В УСТРОЙСТВЕ</b> .....	<b>16</b>
Режимы конфигурирования .....	16
Подчиненный последовательный режим .....	17
Ведущий последовательный режим .....	18
Режим SelectMAP .....	20
Использование режима периферийного сканирования для конфигурирования Virtex .....	23
<b>Последовательность конфигурации</b> .....	<b>23</b>
Задержка конфигурирования .....	24
Последовательность вхождения в штатный режим работы .....	24
<b>Формат потока конфигурационных данных</b> .....	<b>24</b>
<b>ОБРАТНОЕ СЧИТЫВАНИЕ</b> .....	<b>25</b>
<b>ХАРАКТЕРИСТИКИ СЕМЕЙСТВА VIRTEX ПО ПОСТОЯННОМУ ТОКУ</b> .....	<b>25</b>
<b>КОРПУСА</b> .....	<b>25</b>
<b>ОБОЗНАЧЕНИЕ MC СЕМЕЙСТВА VIRTEX</b> .....	<b>26</b>
<b>СОДЕРЖАНИЕ</b> .....	<b>27</b>