

21 ноября 2000 г. Краткое техническое описание

Семейство Spartan™-II – второе поколение ПЛИС (Программируемые Логические Интегральные Схемы), предназначенных для использования в крупносерийных проектах. Архитектура семейства Spartan™-II основана на архитектуре популярного семейства Virtex™. ПЛИС семейства Spartan™-II могут применяться в проектах как альтернатива специализированным интегральным схемам ёмкостью до 200 000 вентилях и системным быстродействием до 200 МГц. Напряжение питания ядра кристалла семейства Spartan™-II составляет 2,5 В.

Стоимость ПЛИС Spartan™-II эквивалентна стоимости заказных специализированных интегральных

схем в партиях от 100 000 штук и в 3-4 раза меньше, чем стоимость ПЛИС семейства Virtex™ при розничных объёмах. Такое снижение цены в сравнении с семейством Virtex™ достигнуто благодаря использованию новой технологии производства кремния и уменьшению номенклатуры корпусов. Набор выполняемых функций аналогичен семейству Virtex™, но отсутствует термодиод.

Семейство состоит из 6-ти кристаллов, отличающихся логической ёмкостью. Сравнительные параметры приведены в таблице 1.

Таблица 1: Основные характеристики семейства Spartan™-II.

Кристалл	Логические Ячейки	Системные вентили	Матрица КЛБ	КЛБ	Блочная ОЗУ, Бит	Пользовательские блоки ввода-вывода, max
XC2S15	432	15 000	8x12	96	16 384	86
XC2S30	972	30 000	12x18	216	24 576	132
XC2S50	1 728	50 000	16x24	384	32 768	176
XC2S100	2 700	100 000	20x30	600	40 960	196
XC2S150	3 888	150 000	24x36	864	49 152	260
XC2S200	5 292	200 000	28x42	1 176	57 344	284

## Особенности

- Высокопроизводительные, программируемые пользователем логические интегральные схемы с архитектурой FPGA (Field Programmable Gate Arrays)
  - Ёмкость от 15 000 до 200 000 системных вентилях
  - Системная производительность до 200 МГц
  - Совместимость с шиной PCI 66 МГц
  - Поддержка функции Hot-swap для Compact PCI
  - Поддержка большинства стандартов ввода-вывода (технология SelectIO™)
  - 16 высокопроизводительных стандартов ввода-вывода
  - Прямое подключение к ZBTRAM устройствам
  - Корпусировка в наиболее дешёвые корпуса
  - Совместимость по выводам кристаллов разной ёмкости в одинаковых корпусах
- Встроенные цепи управления тактированием
  - Четыре встроенных модуля автоподстройки задержек (DLL - delay-locked loop) для расширенного управления тактовыми сигналами как внутри ПЛИС, 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
- Проектирование осуществляется пакетами программного обеспечения Foundation™ и Alliance Series, работающими на ПК или рабочей станции
- Конфигурация кристалла хранится во внешнем ПЗУ и загружается в ПЛИС после включения питания автоматически или принудительно
  - Неограниченное число циклов загрузки
  - Четыре режима загрузки
- Производятся по гибридной технологии 0.18мкм/0.22мкм к-МОП с 6-ти слойной металлизацией на основе статического ОЗУ
- 100% фабричное тестирование

## Архитектура Spartan-II

Основными особенностями архитектуры кристаллов семейства Spartan™-II 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 ПЛИС семейства Spartan™-II даже самые насыщенные и сложные проекты.

Кристаллы семейства Spartan-II 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 Spartan™-II. В других режимах управление загрузкой осуществляется внешними устройствами (подчиненный параллельный режим (Slave Parallel), 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 модули схемного и текстового ввода, моделирования, автоматического и ручного размещения и трассировки, создания, загрузки и верификации конфигурационных данных.

### Быстродействие

Кристаллы Spartan-II обеспечивают более высокую производительность, чем предыдущие поколения FPGA. Проекты могут работать на системных частотах до 200 МГц и частотах внутри кристалла, превышающих 350 МГц. Блоки ввода-вывода Spartan-II полностью соответствуют спецификациям PCI шины, поэтому микросхемы позволяют реализовывать интерфейсные схемы, работающие на частоте 33 МГц или 66 МГц. В дополнение к этому кристаллы Spartan-II 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.

## Описание архитектуры

### Матрица Spartan-II

Программируемая пользователем вентиляемая матрица кристалла серии Spartan-II показана на рис.1.

Основными программируемыми элементами матрицы являются:

- Конфигурируемый Логический Блок - КЛБ (в английском варианте Configurable Logic Block – CLB). КЛБ являются основными элементами, на основе которых реализуется вся логика
- Блок Ввода-Вывода - БВВ (в английском варианте Input/Output Blocks - IOB). БВВ осуществляют интерфейс между контактами микросхемы и КЛБ.

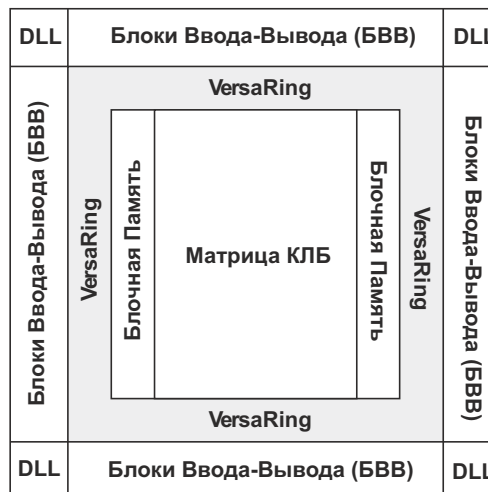


Рис.1: Структура архитектуры Spartan-II

Соединение между КЛБ осуществляется с помощью главных трассировочных матриц - ГТМ (General Routing Matrix - GRM). 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 периферии кристалла. Эти трассы улучшают общую «трассируемость» устройства и возможности трассировки после закрепления электрических цепей к конкретным контактам.

Архитектура Spartan-II 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.

## Блок ввода-вывода

Основным отличительным свойством БВВ семейства Spartan-II является поддержка широкого спектра стандартов сигналов ввода-вывода, что позволяет сопрягать Spartan-II с большинством быстродействующих элементов памяти и шинных интерфейсов. На рис.2 представлена структурная схема БВВ. В таблице 2. 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 отражено 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 при-

близительно 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.

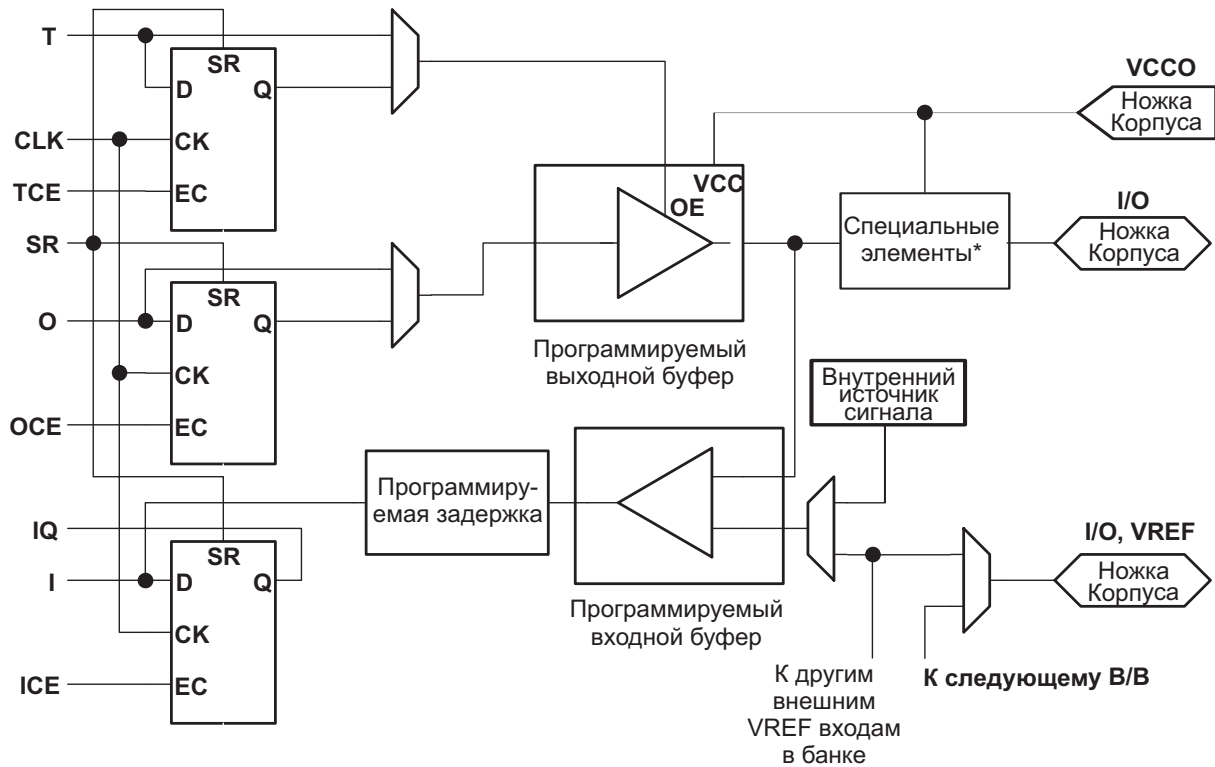
По выбору, к каждому контакту может подключаться:

- Резистор, соединенный с земляной шиной (pull-down)
- Резистор, соединенный с шиной питания (pull-up)
- 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, то нужно использовать внешние резисторы.

Все БВВ микросхемы Spartan-II совместимы со стандартом периферийного сканирования IEEE 1149.1



\*Специальные элементы включают:

- Программируемый резистор, соединенный с земляной шиной (pull-down)
- Программируемый резистор, соединенный с шиной питания (pull-up)
- Маломощная схема удержания последнего состояния (week - keeper)
- Цепи защиты от перенапряжения и электростатического разряда

Рис. 2: БВВ Spartan-II

Таблица 2: Поддерживаемые стандарты В/В

Стандарт В/В	Напряжение порогового уровня входных каскадов ( $V_{REF}$ )	Напряжение питания выходных каскадов ( $V_{CCO}$ )	Напряжение согласования с платой ( $V_{TT}$ )
LVTTL 2–24 мА	нет	3.3	нет
LVCMS2	нет	2.5	нет
PCI	нет	3.3	нет
GTL	0.8	нет	1.2
GTL+	1.0	нет	1.5
HSTL Class I	0.75	1.5	1.5
HSTL Class III	0.75	1.5	1.5
HSTL Class IV	0.75	1.5	1.5
SSTL3 Class I & II	1.5	3.3	1.5
SSTL2 Class I & II	1.25	2.5	1.25
CTT	1.5	3.3	1.5
AGP	1.32	3.3	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 контакт-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 стандартов ввода-вывода, 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). Номинал 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 протрассирован 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 стандартов ввода-вывода 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 (задаётся 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 неопределённость 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 разбивки на банки.

### Банки ввода-вывода

Некоторые из указанных 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 не-

сколько 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 таблице 3. Стандарты GTL и GTL+ присутствуют во всех вариантах, поскольку их выходы с открытым стоком не зависят от значения  $V_{CCO}$ .

Некоторые 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.

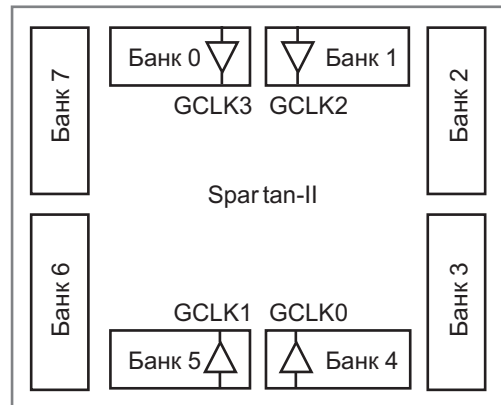


Рис. 3: Банки ввода-вывода Spartan-II

Таблица 3: Совместимые стандарты ввода-вывода

$V_{CCO}$	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. Входные буферы, которые используют  $V_{REF}$ , не совместимы с сигналами 5В стандартов. БВВ, запрограммированные на стандарты LVTTTL, LVCMOS2 и PCI, совместимы с 5В-стандартами.

Номера контактов  $V_{CCO}$  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 для меньших кристаллов существует максимальный набор контактов  $V_{REF}$ , 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  $V_{CC0}$ , используемые в больших кристаллах, не соединены внутри корпуса. Эти не присоединенные контакты могут быть оставлены не присоединенными вне микросхемы или быть подключены к напряжению  $V_{CC0}$  при необходимости обеспечения совместимости разрабатываемой печатной платы с большими кристаллами.

В корпусах TQ144 и PQ208 все контакты  $V_{CC0}$  соединены вместе внутри микросхемы и, следовательно, ко всем из них должно быть подключено одно и тоже напряжение  $V_{CC0}$ . В корпусе CS144 пары банков, расположенные на одной стороне, внутренне соединены, обеспечивая, таким образом, возможность выбора только четырех возможных значений напряжения для  $V_{CC0}$ . Контакты  $V_{REF}$  ос-

таются внутренне соединенными в рамках каждого из восьми банков и могут использоваться, как было описано выше.

## Конфигурируемый логический блок

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 Spartan-II содержит четыре логические ячейки, организованные в виде двух одинаковых секций (Slice). Одна секция показана на рис. 4.

В дополнение к четырем базовым логическим ячейкам, КЛБ серии Spartan-II 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 Spartan-II, каждый КЛБ приравнивается к 4,5 ЛЯ.

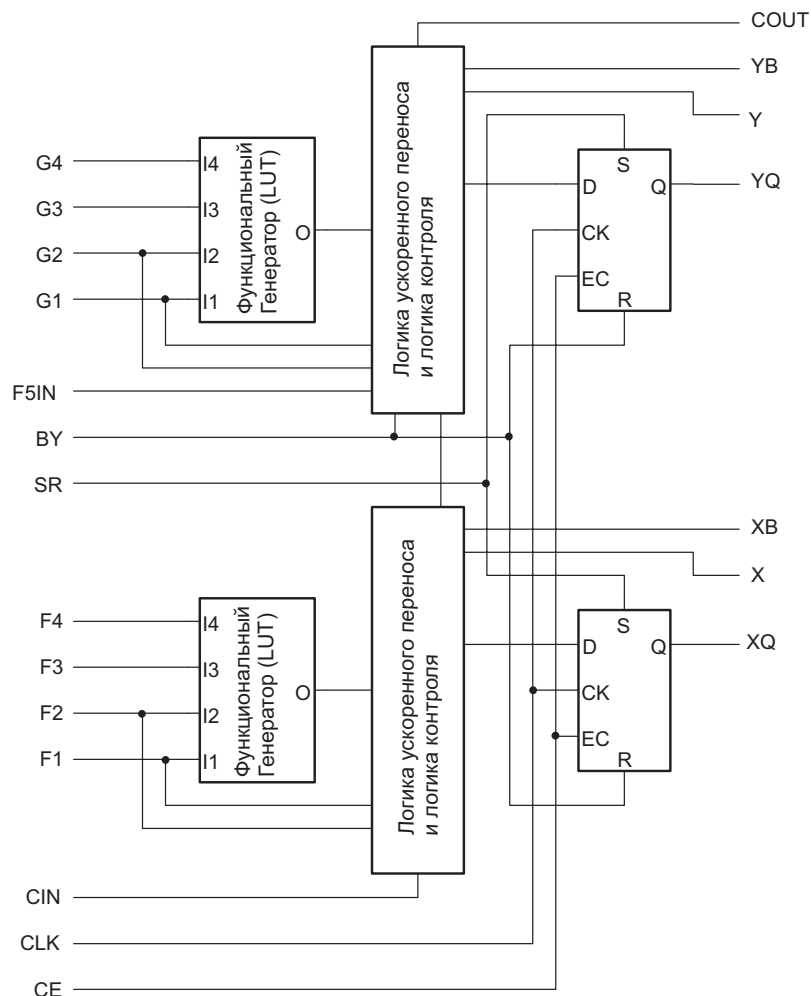


Рис. 4: Секция КЛБ семейства Spartan-II

### 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 Spartan-II 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 Spartan-II 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, определённое 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 функциональный 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 (XOR), который позволяет реализовать однобитный сумматор в одной логической ячейке.

В каждой логической ячейке имеется элемент, реализующий функцию И, 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 Spartan-II содержит два буфера с тремя состояниями, которые нагружены на внутренние шины (см. также “Специальные трассировочные ресурсы”). Каждый буфер BUFT имеет независимый вход управления третьим состоянием и независимый входной контакт.

### Блочная память (Block RAM)

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

Блоки памяти Block Select RAM организованы в виде столбцов. Все кристаллы Spartan-II 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 Spartan-II, имеющая 8 КЛБ по высоте, содержит 2 блока памяти на колонку и 4 блока памяти в целом.

В таблице 4 приводятся ёмкости блочной памяти для различных кристаллов Spartan-II.

Каждый блок памяти, как показано на рис. 5, это полностью синхронная двухпортовая RAM с независимым управлением для каждого порта. Размерность шины данных для обеих портов может быть сконфигурирована независимо, что позволяет создавать преобразователи размерности шины.

В таблице 5 показаны возможные соотношения размерностей шин данных и адреса.

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

Таблица 4: Ёмкость блочной памяти

Кристалл Spartan-II	Число блоков	Общий объём блочной памяти, Бит
XC2S15	4	16 384
XC2S30	6	24 576
XC2S50	8	32 768
XC2S100	10	40 960
XC2S150	12	49 152
XC2S200	14	57 344

Таблица 5: 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	216	ADDR<7:0>	DATA<15:0>

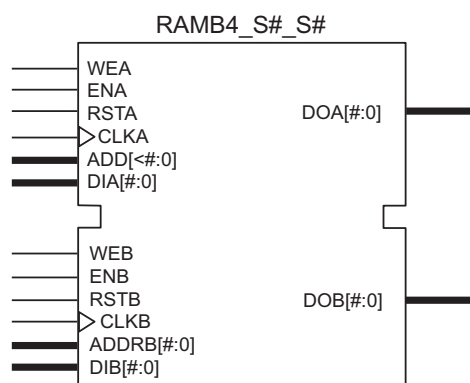


Рис. 5: Блок памяти

## 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. 6, в кристалле Spartan-II созданы локальные трассировочные ресурсы, называемые VersaBlock. Они позволяют реализовать три типа соединений:

1. Связи между таблицами преобразования (LUT), триггерами и главной трассировочной матрицей – ГТМ (GRM)
2. Внутренние обратные связи КЛБ (CLB), которые создают высокоскоростные связи с таблицами преобразования (LUT) в рамках одного КЛБ (CLB), 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 главной трассировочной матрицы



Рис. 6: Локальные связи в кристалле Spartan-II



**Трассировочные ресурсы общего назначения**

Большинство связей в кристаллах Spartan-II 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:

- Примыкающая к каждому КЛБ главная трассировочная матрица – ГТМ (General Routing Matrix – GRM). 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 серединных (три блока от источника). Одна третья часть HEX-линий является двунаправленными, в то время как остальные – однонаправленные.
- 12 длинных линий (Long lines) 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.

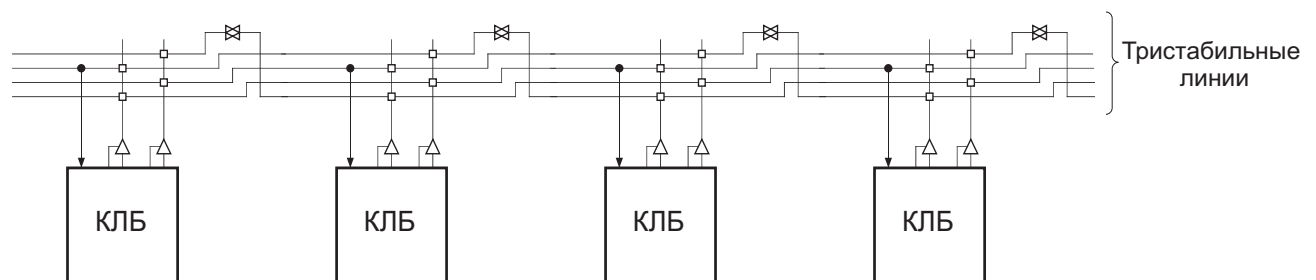
**Трассировочные ресурсы для блоков ввода-вывода**

Кристалл Spartan-II имеет дополнительные трассировочные ресурсы, расположенные по периферии всей микросхемы. Эти трассировочные ресурсы формируют добавочный интерфейс между блоками КЛБ и блоками БВВ. Эти дополнительные ресурсы, называемые VersaRing, улучшают возможности закрепления сигналов за контактами и переназначения уже сделанного закрепления, если это требование накладывается расположением проводников 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 максимизации быстродействия. В устройстве Spartan-II специальные трассировочные ресурсы создавались для двух классов сигналов:

- Горизонтальные трассировочные ресурсы создавались для реализации в микросхеме шин с тремя состояниями. Четыре разделенные линии шин реализованы для каждой строки КЛБ, позволяя организовывать сразу несколько шин в пределах одной строки. (см. рис. 7)
- Две специальные линии для распространения сигналов быстрого переноса к прилегающему КЛБ в вертикальном направлении.



**Рис. 7: 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 Spartan-II 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 BlockRAM – 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 распространено 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, Spartan-II имеет высокоскоростные, с малыми искажениями трассировочные ресурсы для распределения сигналов синхронизации на всем пространстве микросхемы. Типичное распределение цепей синхронизации показано на рис. 8

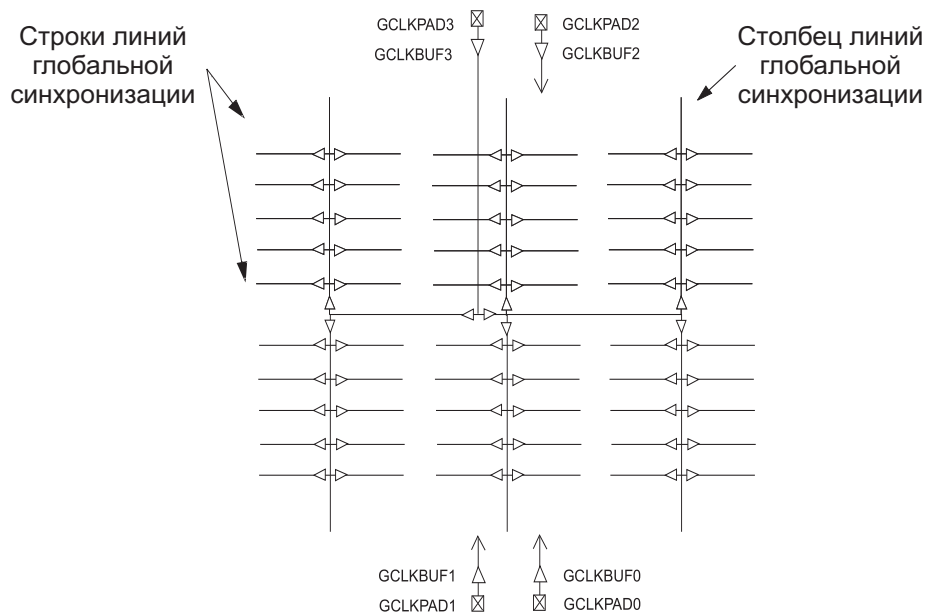


Рис. 8: Глобальные цепи синхронизации

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.

### 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 микросхемами Spartan-II.

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 (PC)

Кристаллы Spartan-II поддерживают команды периферийного сканирования, приведенные в спецификации стандарта IEEE 1149.1. Порт Test Access Port (TAP) и регистры реализованы для выполнения команд EXTEST, 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<sub>CC0</sub> второго банка должно быть подано напряжение 3.3 В. В противном случае напряжение на выходе TDO будет меняться в пределах от нуля до V<sub>CC0</sub>.

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

В таблице 6 приведены команды периферийного сканирования, поддерживаемые кристаллами Spartan-II. 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 однонаправленные входные контакты.

**Таблица 6: Команды 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, SAMPLE / PRELOAD и INTEST.

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

На рис. 9 показана логика периферийного сканирования кристаллов серии Spartan-II. Логика периферийного сканирования состоит из 3-х разрядного регистра данных на один БВВ, контроллера порта TAP и регистра команд с декодированием.

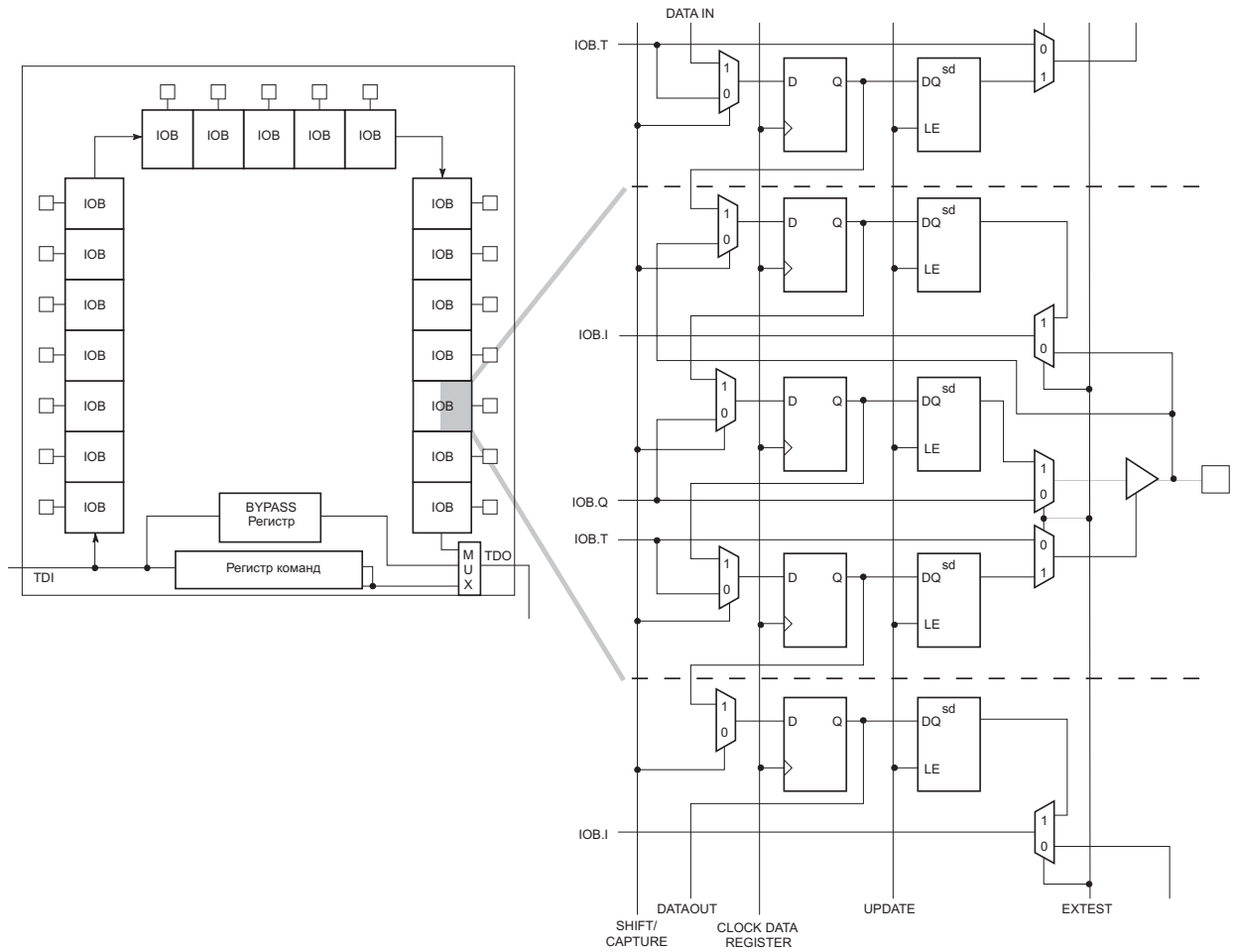


Рис. 9: Логика периферийного сканирования серии Spartan-II

**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 Spartan-II содержит две дополнительные внутренние цепи сканирования, 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. 10.

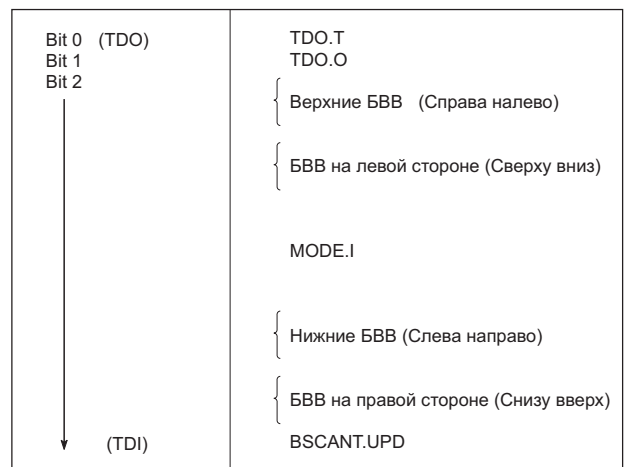


Рис. 10: Последовательность бит ПС

### 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 регистров в проект необходимо включить элемент BSCAN и соединить соответствующие выводы.

## Система проектирования

Разработка кристаллов Spartan-II 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, независимо от выбора вида программы ввода или верификации. Программа XDM упрощает выбор настроек, необходимых для выполнения проекта, благодаря наличию разветвленного меню 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 Spartan-II. 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:

- Synopsys (FPGA Compiler, FPGA Express);
- Exemplar (Spectrum);
- Simplicity (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);
- Viewlogic System (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 Spartan-II включают унифицированную библиотеку (Unified library) стандартных функций. Эта библиотека содержит свыше 400 примитивов и макросов, от двухходовых вентилей 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 размещения 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 Spartan-II 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 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

Конфигурирование – это процесс загрузки битовой последовательности, полученной с помощью программного обеспечения проектирования, во внутреннюю конфигурационную память кристаллов FPGA. Spartan-II может загружаться как побитно (ведущий/подчиненный последовательные режимы и JTAG), так и побайтно (подчиненный параллельный режим).

Конфигурационные данные при выключенном питании должны храниться во внешнем устройстве статической памяти. Обычно для этого применяются ПЗУ Xilinx серии XC1700 или XC1800. В таблице 7 представлены объемы конфигурационной последовательности для кристаллов Spartan-II.

Таблица 7. Размер конфигурационной последовательности в Spartan-II

Кристалл	Объем конфигурационной последовательности, Бит
XC2S15	197 728
XC2S30	336 800
XC2S50	559 232
XC2S100	781 248
XC2S150	1 040 128
XC2S200	1 335 840

## Режимы конфигурирования

Spartan-II поддерживает следующие четыре режима конфигурирования:

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

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 таблице 8.

Таблица 8: 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
Slave Parallel	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
Slave Parallel	0	1	0	Вход	8	Нет	Да
Slave-serial	0	1	1	Вход	1	Есть	Да

Конфигурирование через порт периферийного сканирования (режим Boundary-scan) может осуществляться при любом значении M2, M1, M0. Установка значений, перечисленных в таблице 8 отключает все остальные режимы. Контакты M2, M1, M0 внутри кристалла подключены к pull-up резисторам, поэтому если на плате они ни куда не подключены, то на них присутствует логическая '1'.

### Сигналы конфигурации

Микросхемы Spartan-II 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 служить 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  $\overline{\text{PROGRAM}}$ ;
- контакт DONE;
- контакты 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

### Последовательность конфигурации

Конфигурирование устройств Spartan-II – процесс, состоящий из трех фаз. В первой фазе конфигурирования очищается конфигурационная память. Следующая фаза – загрузка данных в конфигурационную память. Наконец, активизируется логика (фаза Start-Up). Алгоритм выполняемых операций при этом процессе показан на рис. 11.

Первая и последняя фазы одинаковы для всех режимов конфигурирования, а фаза загрузки данных в конфигурационную память различаются и будут

описаны ниже, в разделах посвященных конкретному режиму конфигурирования.

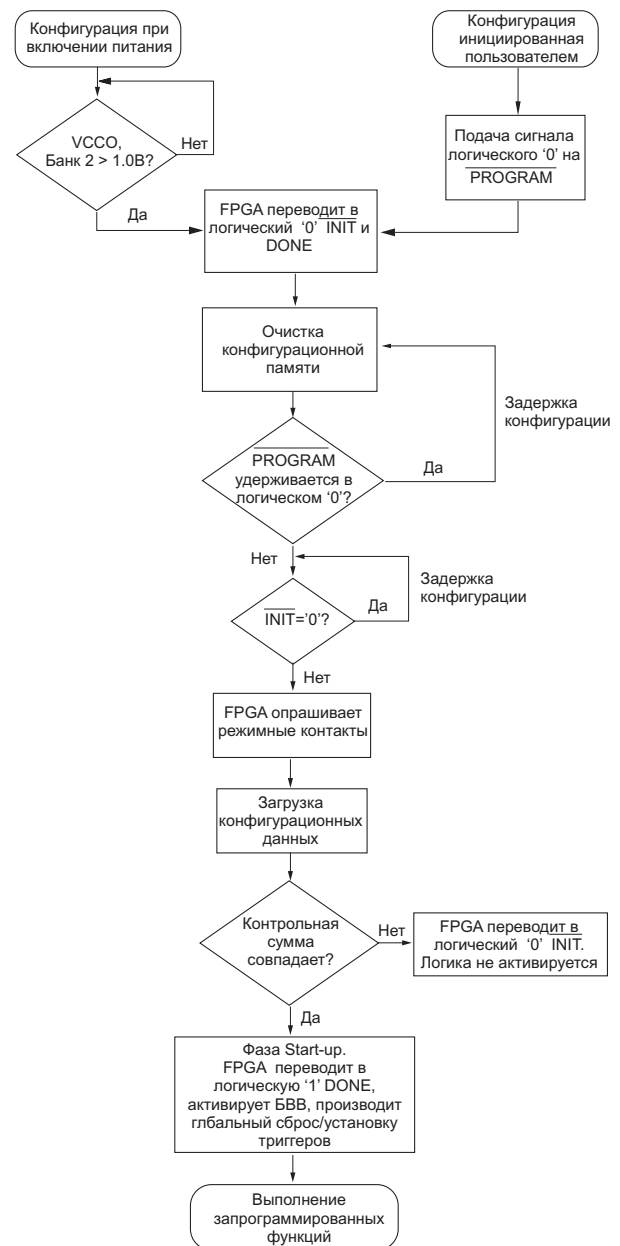


Рисунок 11. Алгоритм конфигурирования

#### Инициализация конфигурирования

Обычно процесс конфигурирования запускается автоматически после подачи напряжения питания, од-

нако, как будет описано далее, он может быть задержан пользователем. Конфигурационный процесс может также быть инициирован установкой активного уровня сигнала PROGRAM. Переход сигнала  $\overline{\text{INIT}}$  в состояние логической '1' означает окончание фазы очистки памяти, а установка активного уровня сигнала DONE ('1') означает окончание процесса в целом.

Для того, чтобы при включении питания начался процесс конфигурирования, необходимо, чтобы напряжение на контактах  $V_{\text{CCO}}$  банка 2 превышало 1,0 В, а на все  $V_{\text{CCINT}}$  было подано напряжение 2,5 В.

Временная диаграмма для конфигурационных сигналов после подачи напряжения питания показана на рис. 12, а соответствующие временные характеристики – в таблице 9.

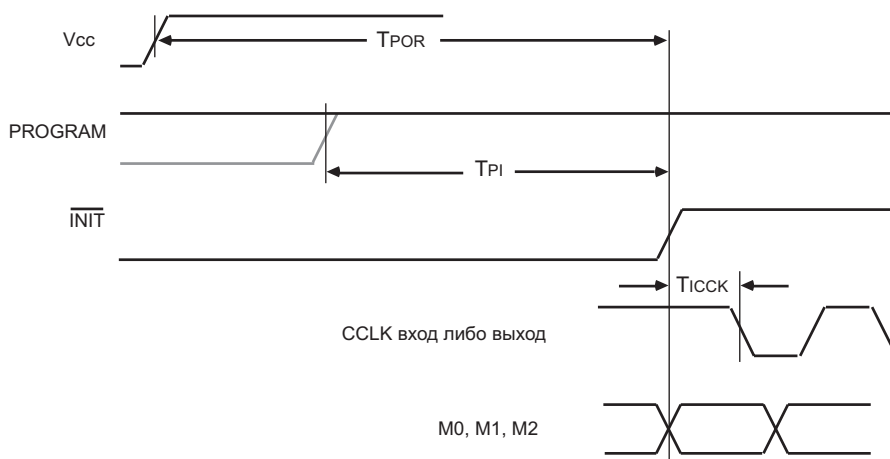


Рис. 12: Временные диаграммы конфигурирования при подаче питания

Таблица 9: Значения временных параметров конфигурирования при подаче питания

Параметр	Значение	
	Min	Max
$T_{\text{POR}}$		2.0 мс
$T_{\text{PL}}$		100 мкс
$T_{\text{ICCK}}$	0.5 мкс	4.0 мкс
$T_{\text{PROGRAMM}}$	300 нс.	

### Очистка конфигурационной памяти

Процесс очистки конфигурационной памяти обозначается выставлением микросхемой Spartan-II логического нуля на контакте  $\overline{\text{INIT}}$ . В это время 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 задержки конфигурирования – подача от источника с открытым стоком сигнала низкого уровня на вход  $\overline{\text{INIT}}$ . Источник сигнала с открытым стоком необходим потому, что контакт  $\overline{\text{INIT}}$  – 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.

Окончание фазы очистки памяти обозначается выставлением микросхемой Spartan-II логической единицы на контакте  $\overline{\text{INIT}}$ .

### Загрузка конфигурационных данных

После перехода  $\overline{\text{INIT}}$  в '1' можно начинать загрузку конфигурационных данных. Процесс загрузки для каждого режима специфичен и будет описан далее, в разделах, посвященных конкретному режиму.

### Проверка контрольной суммы

Во время загрузки конфигурационных данных контрольная сумма, записанная в загружаемом файле, сравнивается с контрольной суммой данных в конфигурационной памяти микросхемы Spartan-II. В случае их несовпадения микросхема Spartan-II переводит  $\overline{\text{INIT}}$  в ноль и прекращает процесс загрузки.

Для реконфигурирования нужно подать логический '0' на контакт PROGRAM на время не меньше 300 нс.

**Последовательность вхождения в штатный режим работы**

При выполнении вхождения в штатный режим работы выполняются четыре операции:

1. Сигнал DONE переводится в '1'. Если этого не произошло, то процедура загрузки конфигурационных данных прошла с ошибками
2. Проходит глобальный сигнал управления третьим состоянием (global tristate – GTS). Это позволяет выходам FPGA включиться надлежащим образом.
3. Проходит сигнал глобальной установки/сброса (Global Set/Reset – GSR).
4. Проходит сигнал глобального разрешения записи (Global Write Enable – GWE). Это создает условия для начала нормальной работы внутренних запоминающих элементов.

По умолчанию процедура вхождения в штатный режим работы синхронизована с сигналом CCLK. Вся процедура занимает 8 тактов CCLK, обозначаемых как C0-C7, после чего загруженный проект может функционировать в штатном режиме. Временная диаграмма представлена на рис.13а. Четыре операции могут выполняться на любом такте CCLK с C1 по C6. Такт выполнения операции программируется программным обеспечением проектирования. На рисунке жирной линией показана временная диаграмма, заданная по умолчанию.

На рис.13б показана временная диаграмма другой часто используемой версии процедуры вхождения в штатный режим. В этой версии сигналы GSR, GTS, GWE проходят одновременно, после перехода DONE в единицу. Данный метод применяется при конфигурации в последовательном режиме цепочки из нескольких микросхем FPGA. Данный метод реализуется путем выставления в ПО GSR, GTS, GWE через такт после DONE.

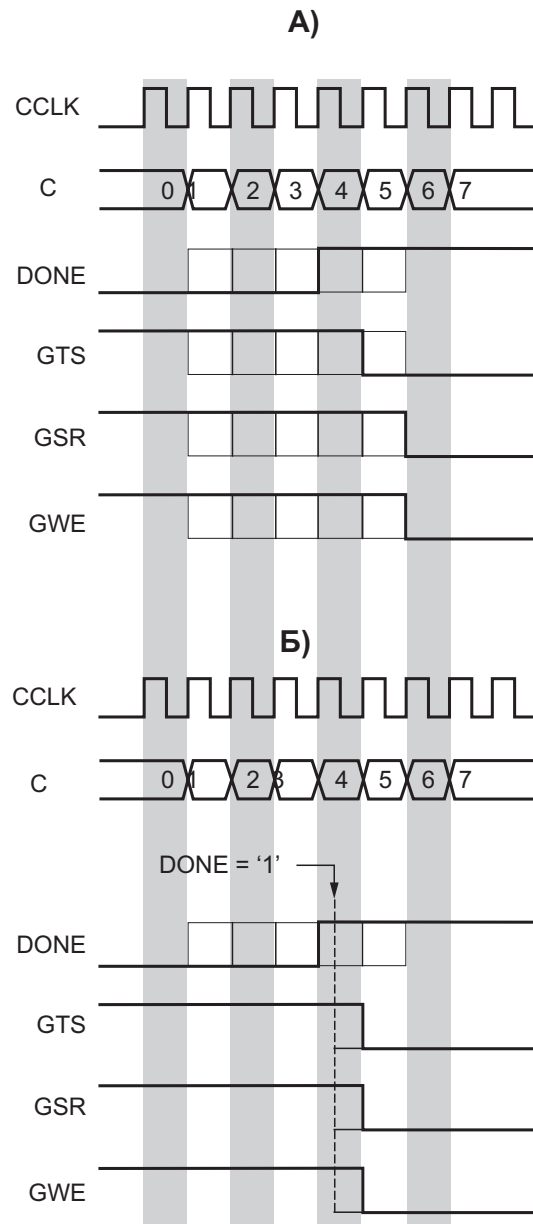


Рисунок 13. Временные диаграммы фазы Start-up

**Последовательные режимы конфигурации**

Существует два вида последовательного режима конфигурирования:

- Ведущий последовательный, при котором загрузкой управляет микросхема FPGA и её контакт CCLK является выходом, то есть источником синхросигнала для внешней памяти, где хранится конфигурационный файл
- Подчиненный последовательный, при котором загрузкой управляет внешнее устройство (Микропроцессор, CPLD, другая FPGA и т.д.). Контакт CCLK при этом является входом.

В обоих случаях за один такт CCLK загружается 1 бит конфигурационных данных. Первым загружается старший бит конфигурационного байта данных.

Последовательность операций изображена на рис.14. Данная схема является более полным представлением блока «Загрузка конфигурационных данных» на рис.11

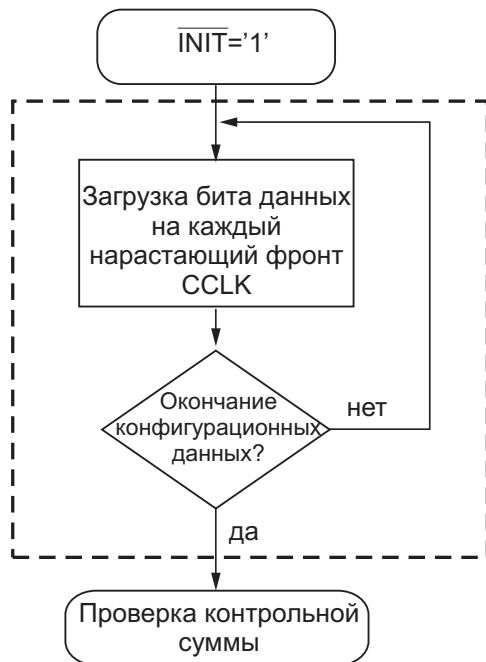


Рисунок 14. Загрузка конфигурационных данных в последовательных режимах

#### Подчиненный последовательный режим (Slave Serial)

В этом режиме FPGA принимает конфигурационные данные в последовательной форме от последовательного ПЗУ (Serial Programmable Read Only Mem-

ory – 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, отличается 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 Spartan-II.

На рис. 15 изображена полная схема, совмещающая подчиненный режим и ведущий режим. FPGA Spartan-II, конфигурируемые в подчиненном режиме, должны быть подключены так же, как устройство, изображенное третьим слева.

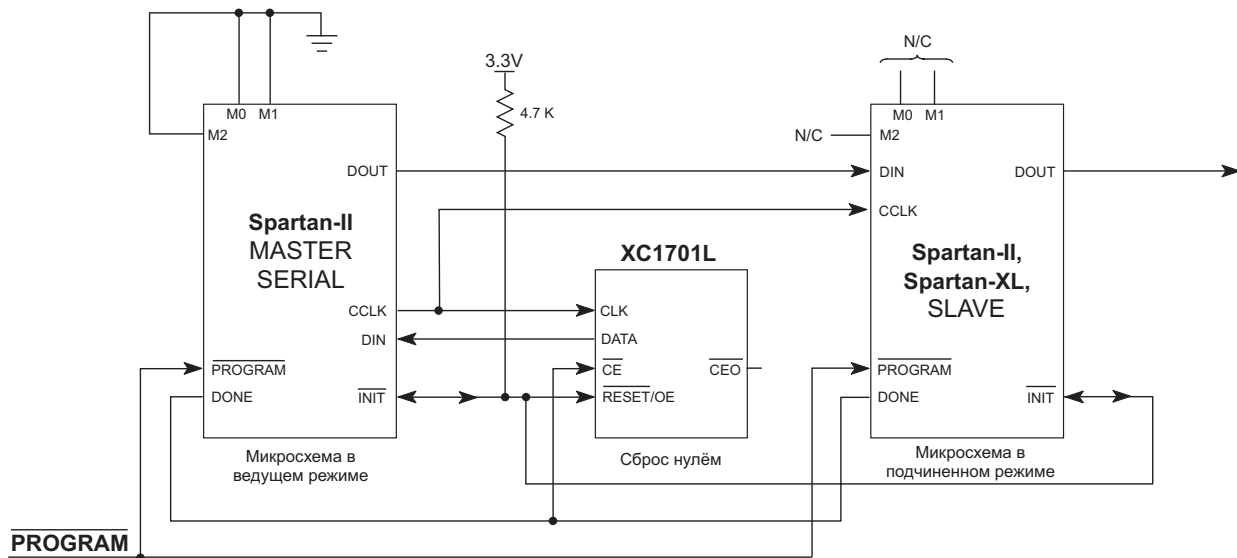


Рис. 15: Ведущий/подчиненный режимы конфигурации

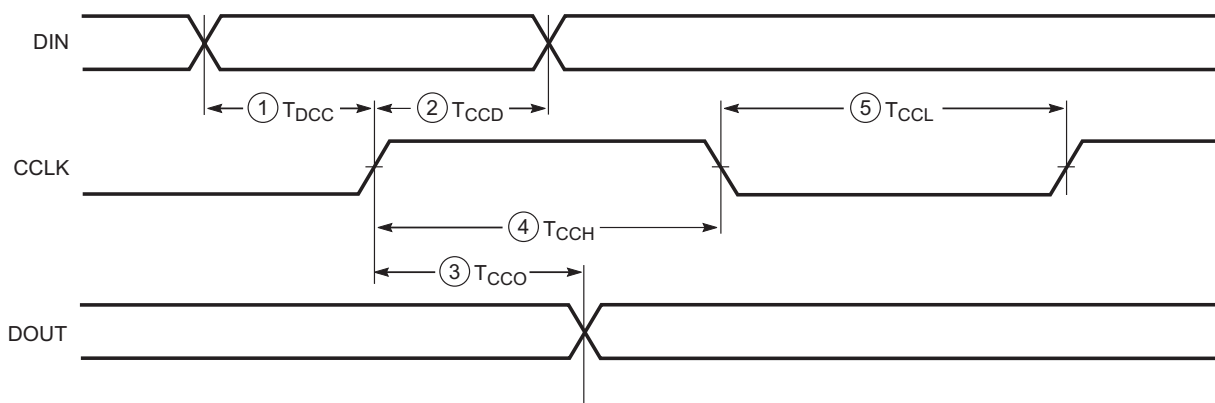
Подчиненный последовательный режим выбирается заданием кода <111> на входах режима конфигурирования (M2, M1, M0). Внутренние высокоомные резисторы на режимных контактах «подтягивают» данные входы в состояние высокого логического уровня и, таким образом, задают этот режим по

умолчанию, если выходы не имеют внешних подключений. На Рисунке 16 изображена временная диаграмма для данного режима.

В таблице 10 содержится более подробная информация для величин, приведенных на рис. 16. Для



FPGA, соединенных в цепочку, процесс конфигурирования должен быть задержан до тех пор, пока на контакте  $\overline{\text{INIT}}$  присутствует высокий логический уровень.



**Рис. 16:** Временная диаграмма подчиненного режима конфигурации

**Таблица 10:** Параметры сигналов подчиненного режима

Параметр	Обозначение		Значение	
			Min	Max
Предустановка входного сигнала DIN	1	$T_{DCC}/T_{CCD}$	5.0нс	
Удержание входного сигнала DIN	2	$T_{DSCCK}/T_{SCKD}$	0нс	
Задержка сигнала DOUT	3	$T_{CCO}$		12нс
Длительность высокого уровня	4	$T_{CCH}$	5.0нс	
Длительность низкого уровня	5	$T_{CCL}$	5.0нс	
Частота		$F_{CC}$		66МГц

**Ведущий последовательный режим (Master Serial)**

В ведущем последовательном режиме с выхода CCLK FPGA сигнал подается на соответствующий вход микросхемы SPROM, 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 FPGA, соединенные в цепочку, обязательно начинаются с блока, называемого преамбулой.

Интерфейс, поддерживающий этот режим, идентичен интерфейсу подчиненного режима (Slave), 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 некоторую частоту CCLK, необходимо убедиться, что используемые SPROM и все соединенные в цепочку FPGA рассчитаны на конфигурирование в таком темпе.

После включения питания, частота CCLK равна 4 МГц. Эта частота используется до момента загрузки 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. 15 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  $\overline{\text{RESET}}$  микросхемы SPROM подается сигнал с контактов  $\overline{\text{INIT}}$  устройства FPGA. Аналогично, на вход CE – с выхода DONE. При этом, в зависимости от выбранной стартовой последовательности, существует конфликт потенциалов на контакте DONE.

Временная диаграмма для ведущего последовательного режима показана на рис. 17. Данный режим выбирается заданием кода 000 или 100 на входах M2, M1, M0. Необходимую временную информацию для этого режима содержит таблица 11.

Время нарастания напряжения питания  $V_{CC}$  от уровня 1 В до минимально допустимого значения  $V_{CC}$  не должно превышать 50 мс, в противном слу-

чае необходимо удерживать сигнал PROGRAM в состоянии низкого логического уровня до момента достижения допустимого уровня  $V_{CC}$ .

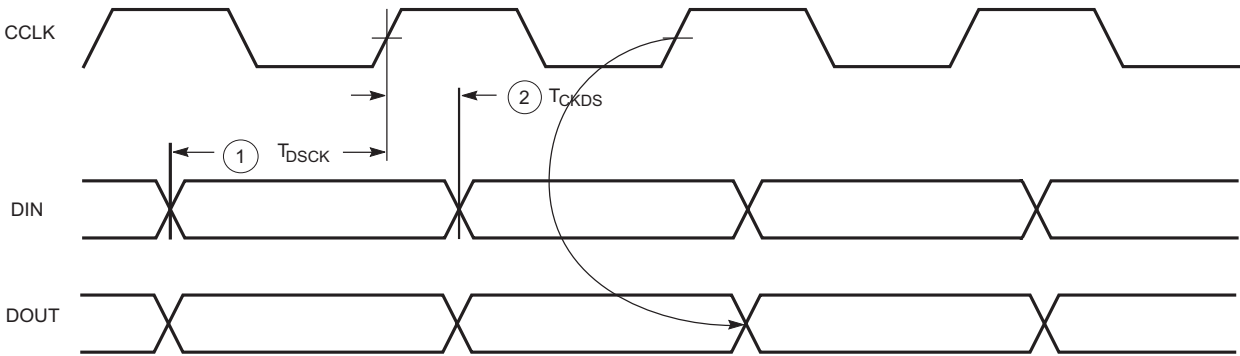


Рис. 17: Временная диаграмма ведущего режима конфигурации

Таблица 11: Параметры сигналов ведущего режима

Параметр	Обозначение		Значение	
			Min	Max
Предустановка входного сигнала DIN	1	$T_{DCC}/T_{CCD}$	5.0нс	
Удержание входного сигнала DIN	2	$T_{DSCK}/T_{SCKD}$	0нс	

### Подчинённый параллельный режим (Slave Parallel)

Подчинённый параллельный режим – самый быстрый режим конфигурирования. В этом режиме данные записываются в FPGA побайтно с использованием флага BUSY, управляющего потоком данных на системных частотах, превышающих 50 МГц.

На рис.18 показаны две микросхемы Spartan-II, использующие подчинённый параллельный режим для конфигурирования. Режим выбирается подачей на M0, M1, M2 либо <011>, либо <010>.

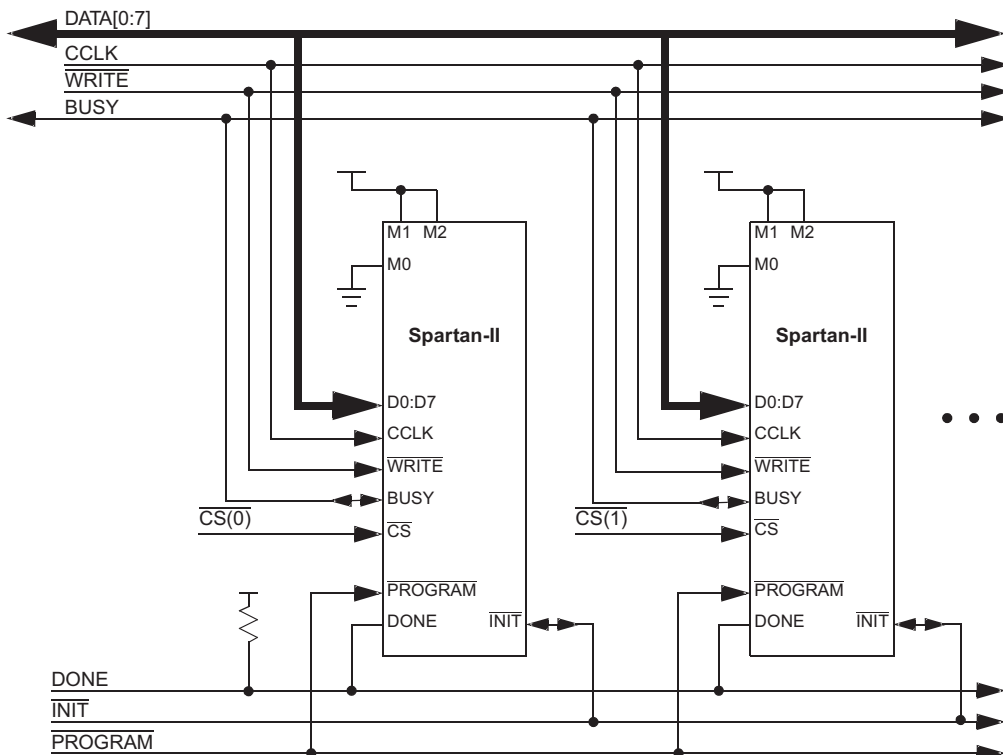


Рисунок 18. Подчинённый параллельный режим

Внешний источник создает байтовый поток данных и сигналы CCLK, выбор кристалла (Chip Select -  $\overline{CS}$ ), запись ( $\overline{WRITE}$ ). 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  $\overline{WRITE}$  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 могут использоваться как дополнительные пользовательские входы/выходы. Можно использовать этот порт для быстрого 8-ми битного обратного считывания конфигурационных данных.

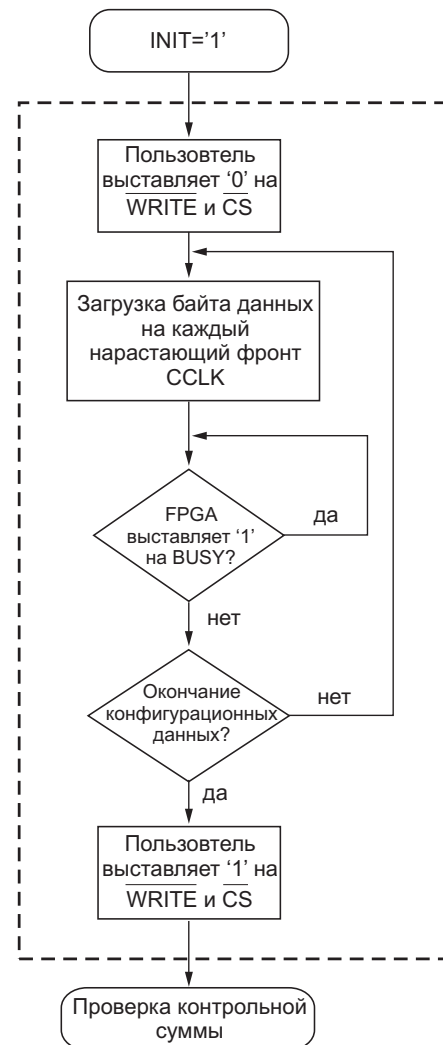
Сохранение такой возможности после конфигурирования реализуется на этапе создания битового потока. Для сохранения такой возможности, необходимо использовать ограничения типа PROHIBIT, предохраняющие контакты порта от использования в качестве пользовательских.

Несколько FPGA Spartan-II могут конфигурироваться в подчиненном параллельном режиме, и далее одновременно запускаться для штатного функционирования. Для конфигурирования нескольких устройств таким способом, необходимо соединить параллельно индивидуальные сигналы отдельных микросхем CCLK, Data,  $\overline{WRITE}$  и BUSY. Конкретные микросхемы конфигурируются по очереди за счет поочередной подачи активного сигнала на контакт выборки ( $\overline{CS}$ ) этой FPGA и записи соответствующих ей данных.

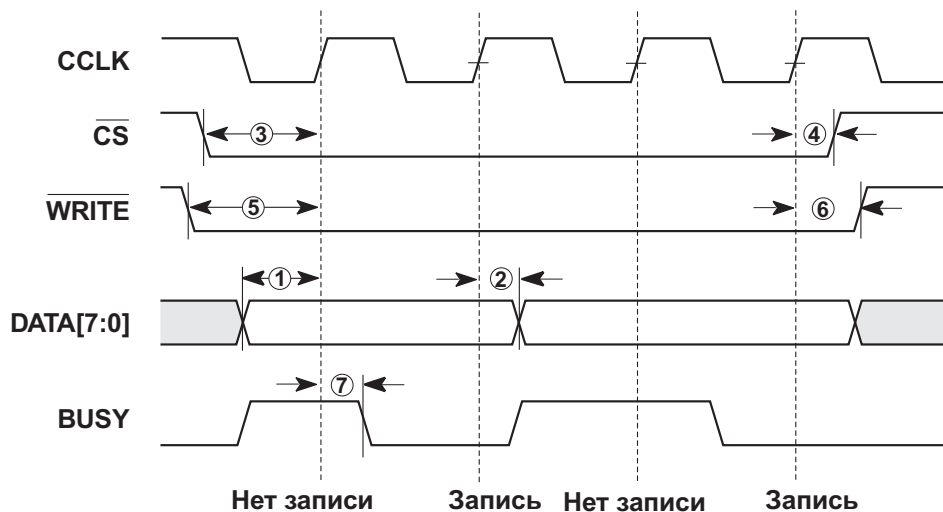
**Запись**

Процедура записи посылает байтовые пакеты конфигурационных данных в FPGA. Последовательность операций для процедуры записи показана на рис. 19. Данная схема является более полным представлением блока «Загрузка конфигурационных данных» на рис.11. Временная диаграмма операций

раций представлена на рис.20, а в таблице 12 представлены её параметры.



**Рисунок 19. Загрузка конфигурационных данных в подчиненном параллельном режиме загрузки.**



**Рисунок 10. Временная диаграмма операций записи подчиненного параллельного режима**

Таблица 12: Параметры сигналов подчинённого параллельного режима

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

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 CCLK,  $\overline{\text{WRITE}}$  должен оставаться неизменным. В противном случае, как описано далее, будет инициировано преждевременное прекращение процедуры.
2. Данные подать на вход 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 CCLK при условии, что сигнал BUSY при этом имеет значение '0'. В случае если сигнал BUSY имел значение '1' от предыдущей записи, данные не принимаются. Данные снова будут приниматься по первому же нарастающему фронту CCLK после перехода 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.

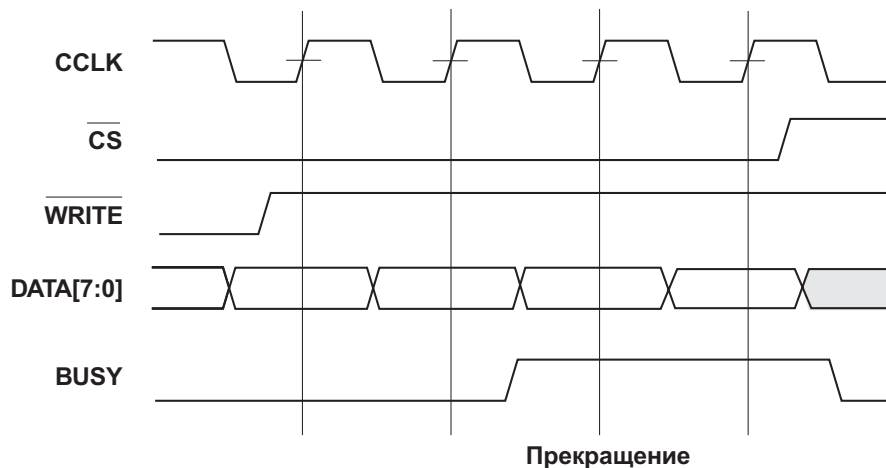
Если сигнал CCLK медленнее, чем  $F_{\text{CCNH}}$ , FPGA не будет выставлять сигнал BUSY, в этом случае обмен подтверждениями готовности после реального приема данных не нужен, и данные могут просто вводиться в FPGA по каждому циклу сигнала CCLK.

Необходимо отметить, что конфигурационный пакет можно расцепить на несколько таких последовательностей. Пакет не должен быть закончен за время одной активизации сигнала  $\overline{\text{CS}}$ .

#### Преждевременное прекращение процедуры

После установки активного уровня сигнала  $\overline{\text{CS}}$ , 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  $\overline{\text{WRITE}}$  в неактивное состояние. Как показано на рис. 21, прекращение процедуры начнется с приходом нарастающего фронта CCLK.



**Рис. 21:** Преждевременное прекращение процедуры записи в подчиненном параллельном режиме загрузки

### Использование режима периферийного сканирования для конфигурирования Spartan-II

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.

### Обратное считывание

Конфигурационные данные, запомненные 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 Spartan-II по постоянному току

### Диапазон максимально допустимым значений

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
		Не используя $V_{REF}$	от -0.5 до 5.5
$V_{TS}$	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

Превышение максимальных значений ведет к повреждению кристалла.

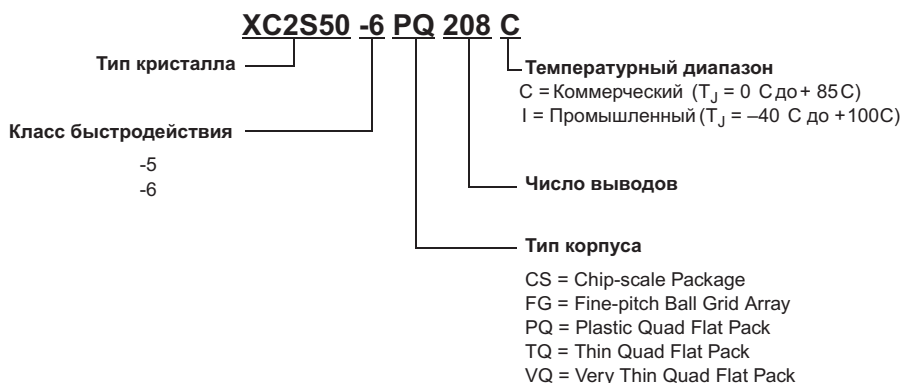
### Рекомендуемые значения

Обозначение	Описание	Min	Max	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

Корпус	Максимальное число пользовательских контактов					
	XCS15	XCS30	XCS50	XCS100	XCS150	XCS200
VQ100	60	60	-	-	-	-
TQ144	86	92	92	92	-	-
CS144	86	92	-	-	-	-
PQ208	-	132	140	140	140	140
FG256			176	176	176	176
FG456				196	260	284

## Маркировка



## Содержание

<b>ОСОБЕННОСТИ</b> .....	<b>1</b>
<b>АРХИТЕКТУРА SPARTAN-II</b> .....	<b>2</b>
Быстродействие .....	2
<b>ОПИСАНИЕ АРХИТЕКТУРЫ</b> .....	<b>2</b>
Матрица Spartan-II.....	2
<b>Блок ввода-вывода</b> .....	<b>3</b>
Ввод сигнала .....	4
Вывод сигнала.....	5
Банки ввода-вывода.....	5
<b>Конфигурируемый логический блок</b> .....	<b>6</b>
Таблица Преобразования .....	7
Запоминающие элементы.....	7
Дополнительная логика.....	7
Арифметическая логика.....	7
Буферы с тремя состояниями.....	7
Блочная память (Block RAM).....	7
<b>Программируемая трассировочная матрица</b> .....	<b>8</b>
Локальные связи .....	8
Трассировочные ресурсы общего назначения.....	9
Трассировочные ресурсы для блоков ввода-вывода.....	9
Специальные трассировочные ресурсы.....	9
Глобальные трассировочные ресурсы .....	9
<b>Распределение сигналов синхронизации</b> .....	<b>10</b>
Модули автоподстройки задержки (DLL) .....	10
<b>Периферийное сканирование (ПС)</b> .....	<b>10</b>
Регистры данных.....	12
Порядок бит регистра данных ПС.....	12
Включение ПС в проект.....	13
<b>СИСТЕМА ПРОЕКТИРОВАНИЯ</b> .....	<b>13</b>
Размещение проекта в кристалл.....	13
Верификация проекта.....	14
<b>КОНФИГУРИРОВАНИЕ КРИСТАЛЛА В УСТРОЙСТВЕ</b> .....	<b>14</b>
Режимы конфигурирования .....	14
Сигналы конфигурации .....	15
Последовательность конфигурации.....	15
Инициализация конфигурирования .....	15
Очистка конфигурационной памяти .....	16
Загрузка конфигурационных данных .....	16
Проверка контрольной суммы.....	16
Последовательность вхождения в штатный режим работы.....	17
Последовательные режимы конфигурации.....	17
Подчиненный последовательный режим (Slave Serial) .....	18
Ведущий последовательный режим (Master Serial) .....	19
Подчинённый параллельный режим (Slave Parallel) .....	20
Запись .....	21
Преждевременное прекращение процедуры .....	22
Использование режима периферийного сканирования для конфигурирования Spartan-II.....	23
<b>ОБРАТНОЕ СЧИТЫВАНИЕ</b> .....	<b>23</b>
<b>ХАРАКТЕРИСТИКИ СЕМЕЙСТВА SPARTAN-II ПО ПОСТОЯННОМУ ТОКУ</b> .....	<b>24</b>
Диапазон максимально допустимым значений .....	24
Рекомендуемые значения .....	24
<b>КОРПУСА</b> .....	<b>24</b>
<b>МАРКИРОВКА</b> .....	<b>24</b>
<b>СОДЕРЖАНИЕ</b> .....	<b>25</b>