МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Учебное пособие

Методические указания
по выполнению реферата
по единому комплексному заданию по блоку дисциплины

«Системотехника ЭВС, комплексы и сети»

МГТУ имени Н.Э. Баумана

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени Н.Э. БАУМАНА

Методические указания
по выполнению реферата
по единому комплексному заданию по блоку дисциплины

«Системотехника ЭВС, комплексы и сети»

Москва
МГТУ имени Н.Э. Баумана

2012

УДК 681.3.06(075.8)
ББК 32.973-018
И201

Методические указания по выполнению реферата по единому комплексному заданию по блоку дисциплины «Системотехника ЭВС, комплексы и сети» /
Коллектив авторов –
М.: МГТУ им. Н.Э. Баумана, 2012. – XX с.: ил.

В методических указаниях рассмотрены основные этапы, их последовательность и содержание по выполнению реферата по единому комплексному заданию по блоку дисциплины «Системотехника ЭВС, комплексы и сети».

Ил. 39. Табл. 5. Библиогр. 7 назв.

УДК 681.3.06(075.8)

© МГТУ им. Н.Э. Баумана, 2012

АННОТАЦИЯ

В данном реферате рассматривается построение архитектуры для FPGA ПЛИС. В ходе рассмотрения были раскрыты вопросы иерархии структуры, рассмотрены основные составные части. Также рассмотрены основные технические параметры FPGA на примере FPAG фирмы Xilinx серии 7. Рассмотрены особенности конфигурирования FPGA.

ABSTRACT

In this lecture considers the construction of architectures for FPGA. During consideration of the issues were uncovered hierarchy structure, the main component parts. It also describes the main technical parameters of the example FPGA from Xilinx FPAG Series 7. The features of configuring the FPGA.

СОДЕРЖАНИЕ

СПИСОК УСЛОВНЫХ СОКРАЩЕНИИ И ТЕРМИНОВ

AES – Advanced Encryption Standard – расширение системы команд x86 для микропроцессоров

CLB – Configurable logic lock – конфигурируемый логический блок – КЛБ

DSP – digital signal processing – Цифрова́я обрабо́тка сигна́лов

FPGA – Field Programmable gate array – полупроводниковое устройство, которое может быть сконфигурировано производителем или разработчиком после изготовления

JTAG – Joint Test Action Group – специализированным аппаратным интерфейсом на базе стандарта IEEE 1149.1

LAB – Logic array block – блок логических массивов

LUT – Lookup table – таблица соответствия

MAC – Multiple-and-accumulate – умножение с накоплением

MUX – Multiplexer – мультиплексор

ОЗУ – Оперативно-запоминающее устройство

ПЛИС – Программируемая логическая интегральная схема

ПП – Печатная плата

ППЗУ – Программируемое постоянное запоминающее устройство

СППЗУ – Стираемое программируемое постоянное запоминающее устройство

ЦОС – Цифровая обработка сигналов

ЭСППЗУ – Электрически-стираемое программируемое постоянное запоминающее устройство

ВВЕДЕНИЕ

FPGA (field programmable gate array) представляют собой *цифровые интегральные микросхемы*, состоящие из программируемых логических блоков и программируемых соединений между этими блоками. Возможность конфигурировать эти устройства позволяет инженерам-разработчикам решать множество различных задач.

Словосочетание «field programmable», содержащееся в расшифровке аббревиатуры FPGA, означает, что программирование FPGA-устройств выполняется на месте, «в полевых условиях» (в отличие от устройств, внутренняя функциональность которых жестко прописана производителем).

Данный реферат посвящен архитектуре FPGA. При этом архитектура рассматривалась так сказать в «общем виде», т.е. без акцента на конкретное семейство какого-либо производителя. Стоит также отметить, что у разных производителей часто одни и те же «части» ПЛИС зачастую называются по-разному.

1 АРХИТЕКТУРА ПЛИС FPGA

1.1 Мелко-, средне- и крупномодульные архитектуры

В общем случае ПЛИС подразделяются на *мелкомодульные* и *крупномодульные*. Главной особенностью ПЛИС является их внутренняя структура, которая преимущественно состоит из большого количества простых программируемых логических блоков-«островков» в «море» программируемых внутренних связей (Рисунок 1.1).

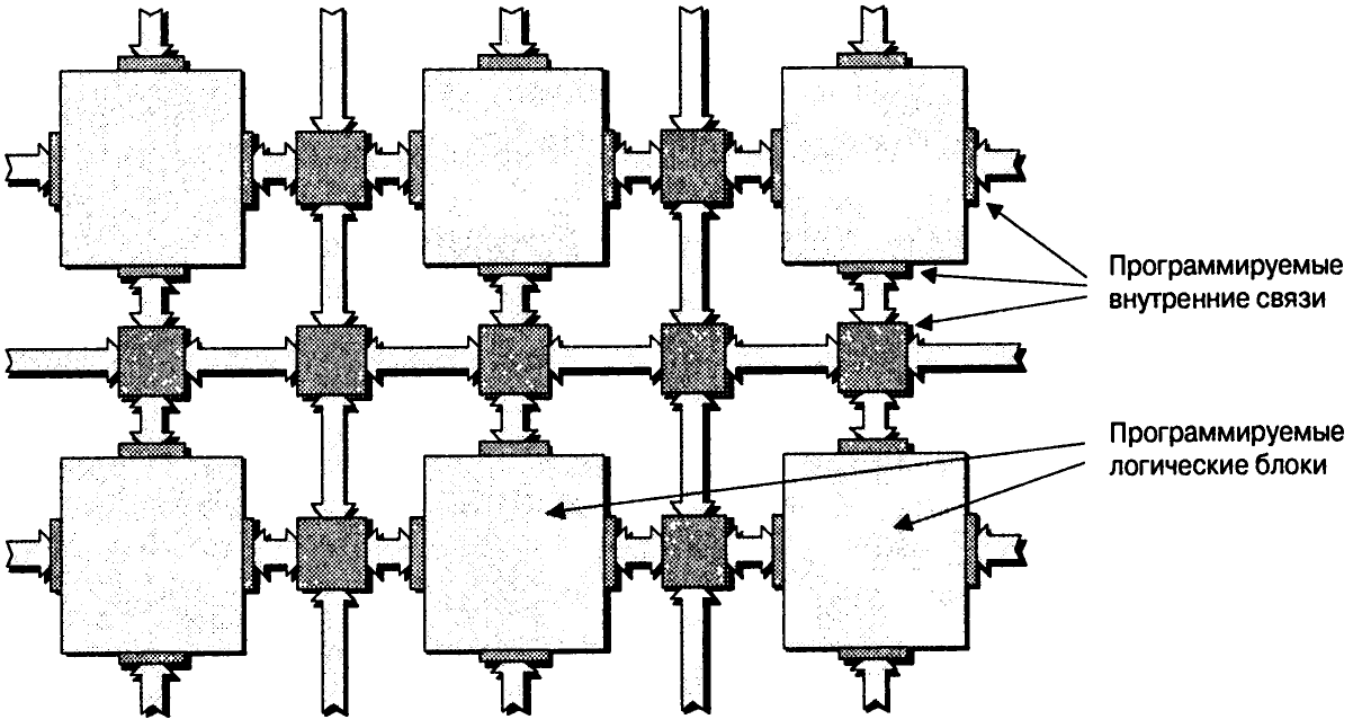

Рисунок 1 – Внутренняя структура ПЛИС

В мелкомодульной архитектуре каждый логический блок может использоваться для реализации только очень простой функции. Например, блок можно сконфигурировать для работы в качестве 3-входового простого логического элемента (И, ИЛИ, И-НЕ и так далее) или элемента памяти (триггер D-типа, защелка D-типа и так далее).

Для мелкомодульных ПЛИС характерно большое количество соединений внутри блоков и между ними. По мере увеличения модульности устройств до среднемодульных и выше количество соединений в блоках уменьшается. Это важное свойство, так как внутренние связи определяют величину подавляющего большинства задержек, связанных с прохождением сигналов через ПЛИС.

Крупномодульные устройства содержат массивы узлов, где каждый узел представляет собой сложный элемент, реализующий алгоритмические функции, например быстрое преобразование Фурье, ил даже ядро микропроцессора общего назначения. Суть заключается в том, что на самом деле эти устройства «мутят воду», поскольку не классифицируются как ПЛИС.

1.2 Логические блоки на мультиплексорах и таблицах соответствия

Существует два основных способа реализации программируемых логических блоков, используемых для формирования среднемодульных устройств на основе мультиплексоров (MUX – от multiplexer) и на основе таблиц соответствия (LUT – от lookup table).

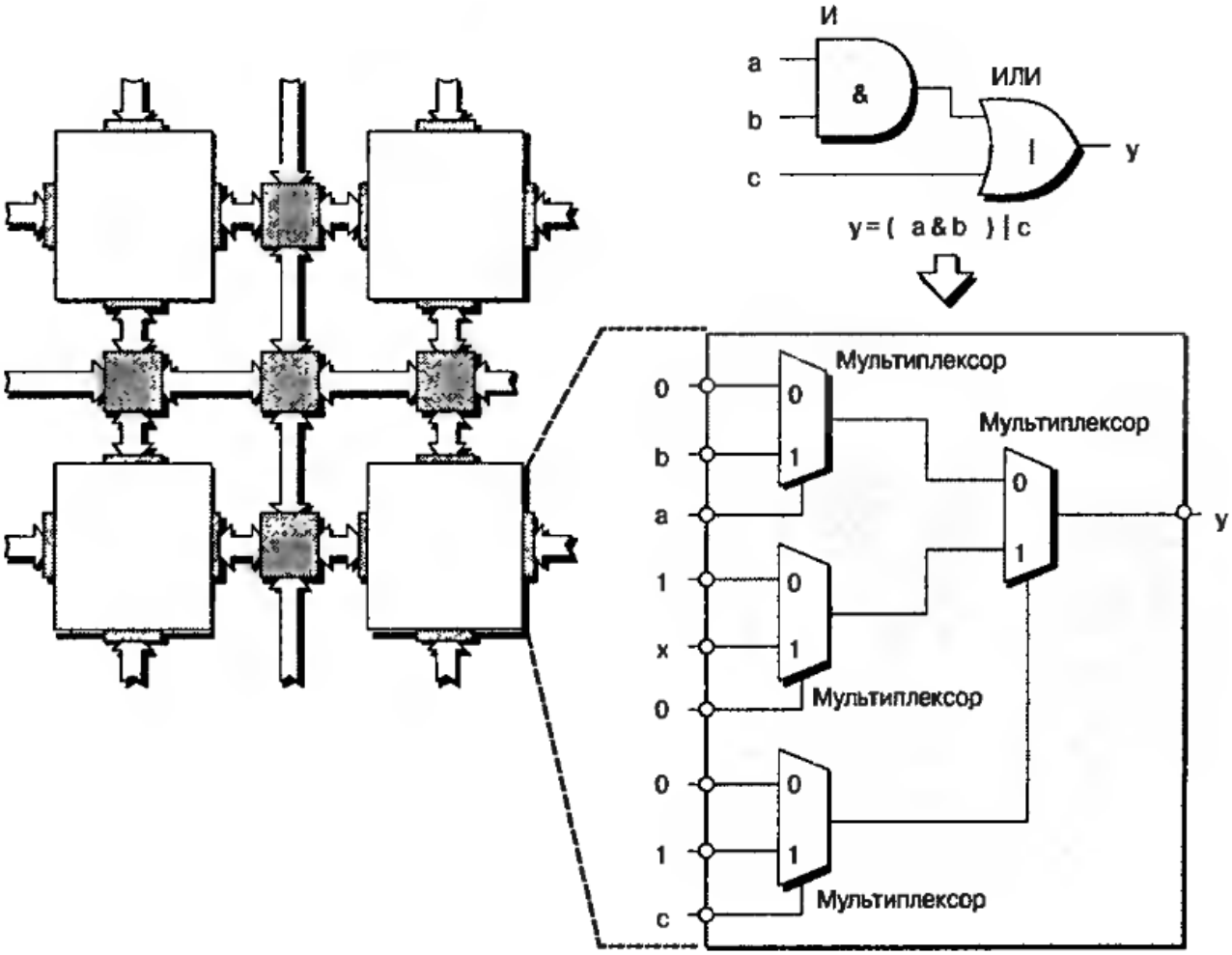

Рисунок 2 – Логический блок на мультиплексорах

На рисунке 2 представлен пример реализации устройств на основе мультиплексоров с 3-входовой функцией *y=(a & b)|c*, реализованного с помощью блока, содержащего только мультиплексоры. Устройство может быть запрограммировано таким образом, что на каждый его вход может подаваться логический 0 либо логическая 1, либо истинное, либо инверсное значение входного сигнала, приходящего с другого блока или с входа микросхемы. Такой подход позволяет для каждого блока создавать огромнейшее количество вариантов конфигурирования для выполнения разнообразнейших функции.

Основная концепция таблиц соответствия относительно проста. В таких микросхемах группа входных сигналов используется в качестве индекса таблицы соответствия. Содержимое этой таблицы организовано таким образом, что ячейки, указываемые каждой входной комбинацией, содержат требуемое выходное значение. Допустим, требуется реализовать функцию *y=(a & b).* Для этого надо загрузить 3-входовую таблицу соответствующими значениями. Допустим, что таблица соответствия формируется из ячеек памяти статического ОЗУ. Она также может быть сформирована наращиваемыми перемычками, ЭСППЗУ- или Flash-ячейками памяти. Для выбора требуемой ячейки ОЗУ с помощью каскада передаточных вентилей используются входные сигналы (рисунок 3). При этом ячейки памяти статического ОЗУ для нагрузки конфигурационных данных должны быть соединены в длинную цепочку (На рисунке не показаны с целью его упрощения). На схеме открытый, ли активный, передаточный вентиль пропускает сигнал с входа на выход. Закрытый вентиль электрически отключает свой выход от проводника, к которому он подсоединен. Вентили с «кружком» активируются при по-

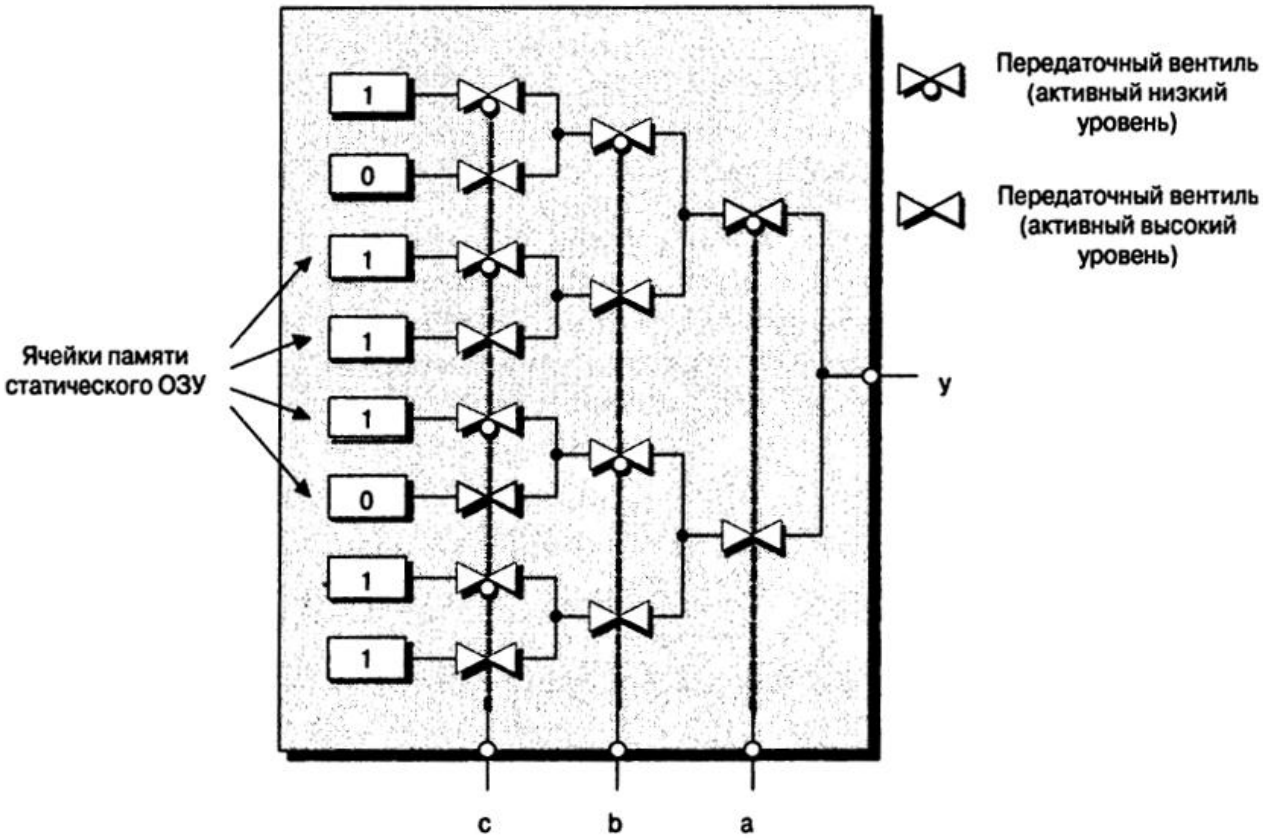

Рисунок 3 – Таблицы соответствия на основе передаточных вентилей

даче на управляющий вход логического 0, соответственно вентили без «кружка» - при подаче логической 1. Основываясь на этом, легко проследить, как различные комбинаций могут использоваться для выбора содержимого требуемой ячейки памяти.

Большинство современных архитектур ПЛИС основывается на таблицах соответствия.

1.3 Конфигурируемые логические блоки, блоки логических массивов, секции

Помимо одной или нескольких таблиц соответствия логический блок может содержать другие элементы, такие как мультиплексоры и регистры.

Логические ячейки фирмы Xilinx

Каждый производитель ПЛИС дает им собственное название. Блоки, из которых состоит современная ПЛИС фирмы Xilinx, называются *логическими ячейками(logic cell).* Логическая ячейка содержит 4-взодовую таблицу соответствия, которая может работать как ОЗУ 16x1 или как 16-битный сдвигающий регистр, а также мультиплексор и регистр (Рисунок 4). Схема, представленная на рисунке 4 сильно упрощена, тем не менее, она удовлетворяет контексту рассматриваемого материала. Кроме таблиц соответствия, мультиплексоров и регистров, логические ячейки содержат небольшое количество других элементов, включая специальную логику быстрого переноса для использования в арифметических действиях.

Логические элементы компании Altera

Блоки, из которых состоят ПЛС компании Altera, называются *логическими элементам (logic element).* Между логическими ячейками Xilinx и логическими элементами Altera существует ряд отличий, но в целом их концепции очень похожи.

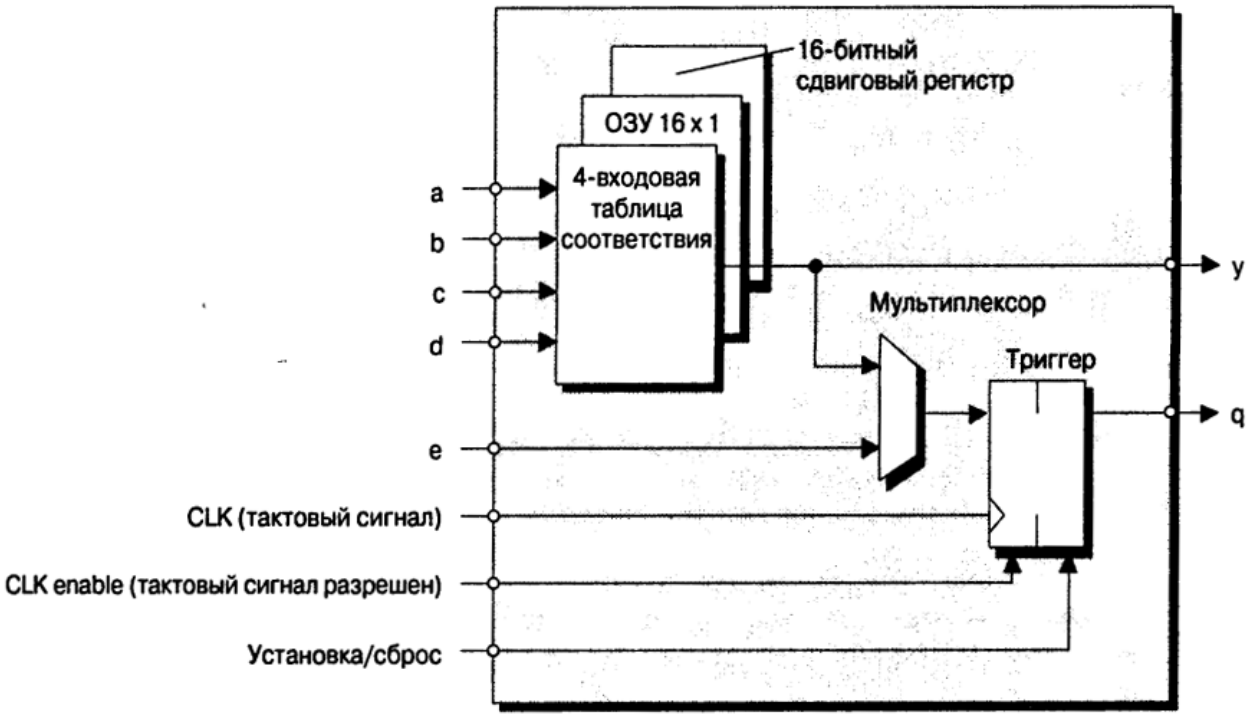

Рисунок 4 – Упрощенный вид логической ячейки Xilinx

Секции и логические ячейки

Следующей ступенью в иерархии построения микросхем программируемой логики является так называемая *секция (slice)*. На рисунке 5 показана секция, содержащая 2 логические ячейки. На рисунке для упрощения не показаны внутренне связи. Секция имеет общие тактовые сигналы, разрешения тактовых сигналов и установки/сброса для обеих логических ячеек.

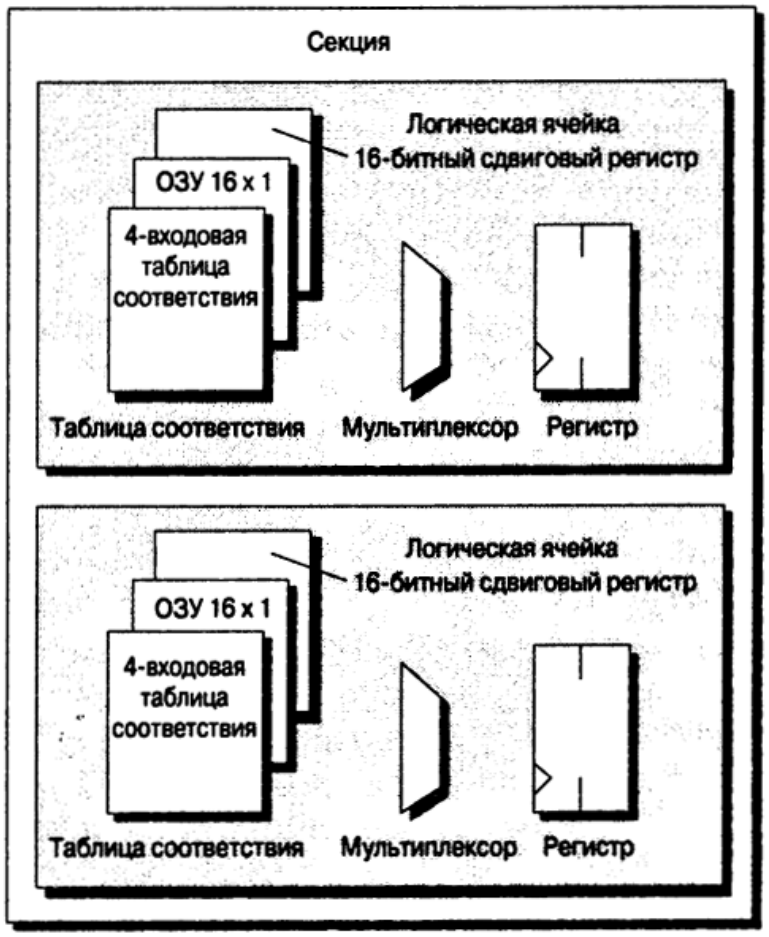

Рисунок 5 – Секция, содержащая две логических ячейки

Конфигурируемые логические блоки CLB и блоки логических массивов LAB

Поднимаясь по иерархической лестнице, достигаем уровня, который компания Xilinx называет *конфигурируемый логический блок КЛБ* или (*CLB – configurable logic block*). Компания Altera, в свою очередь, называет его *блоком логических массивов* или (*LAB – logic array block*). Используя КЛБ в качестве примера, заметим, что некоторые ПЛИС фирмы Xilinx содержат по 2 секции в каждом блоке, другие устройства по 4 секции в каждом блоке (рисунок 6). Внутри логического блока находятся быстрые программируемые внутренние соединения. Эти проводники используются для соединения соседних секции.

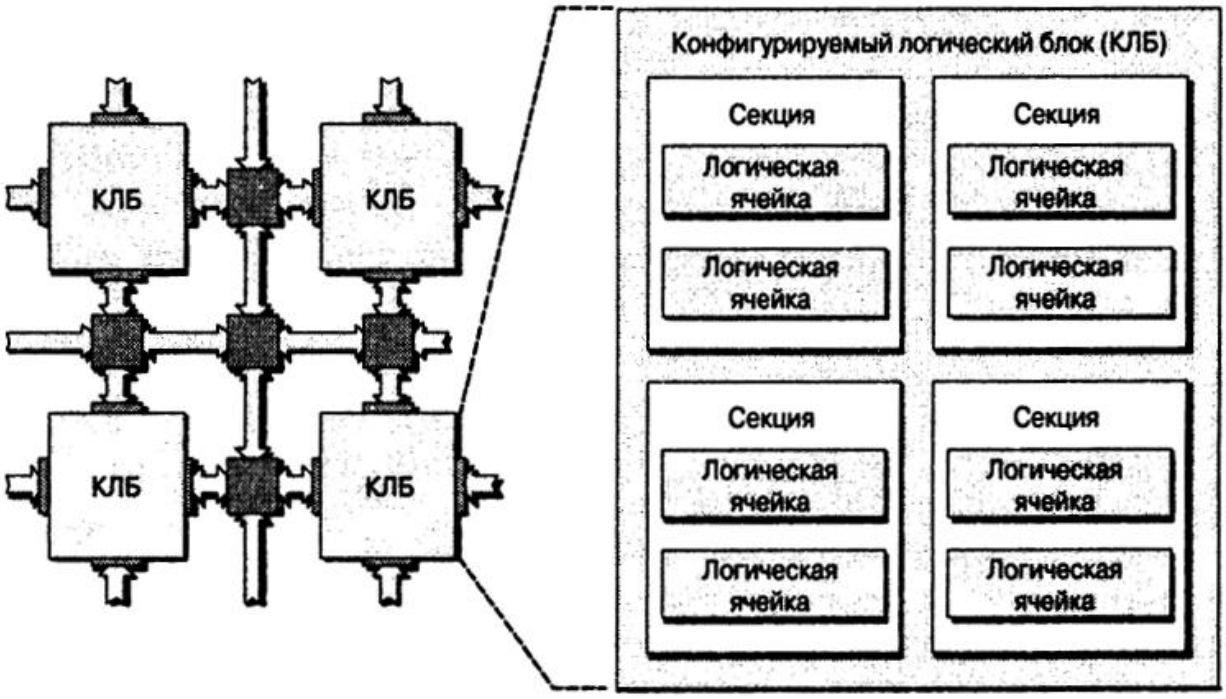

Рисунок 6 – Конфигурируемый логический блок, содержащий четыре секции (количество секции зависит от семейства ПЛИС)

Причина существования такой логико-блочной иерархии заключается в том, что она дополняется эквивалентной иерархией внутренних соединений, т.е. существуют быстрые внутренние соединения между логическими ячейками внутри секции, затем менее быстрые соединения между секциями внутри логического блока и соединения между блоками. Подобная иерархия отражает пошаговое достижение оптимального компромисса между простотой соединений внутренних структур и чрезмерными задержками сигнала на внутренних соединениях.

1.4 Встроенные блоки ОЗУ

В процессе реализации большинства приложении возникает необходимость использовать ячейки памяти, поэтому современные ПЛИС содержат довольно большие блоки встроенной памяти, называемой *блоками встроенного ОЗУ*. Эти блоки могут быть расположены по периметру кристалла, разбросаны по его поверхности и относительно изолированы друг от друга или организованы в столбцы (Рисунок 7). Размер блоков ОЗУ может меняться в зависимости от устройства. Каждый блок ОЗУ может использоваться либо как независимое запоминающее устройство, либо находиться в связке с несколькими блоками для реализации массивов памяти большого объема.

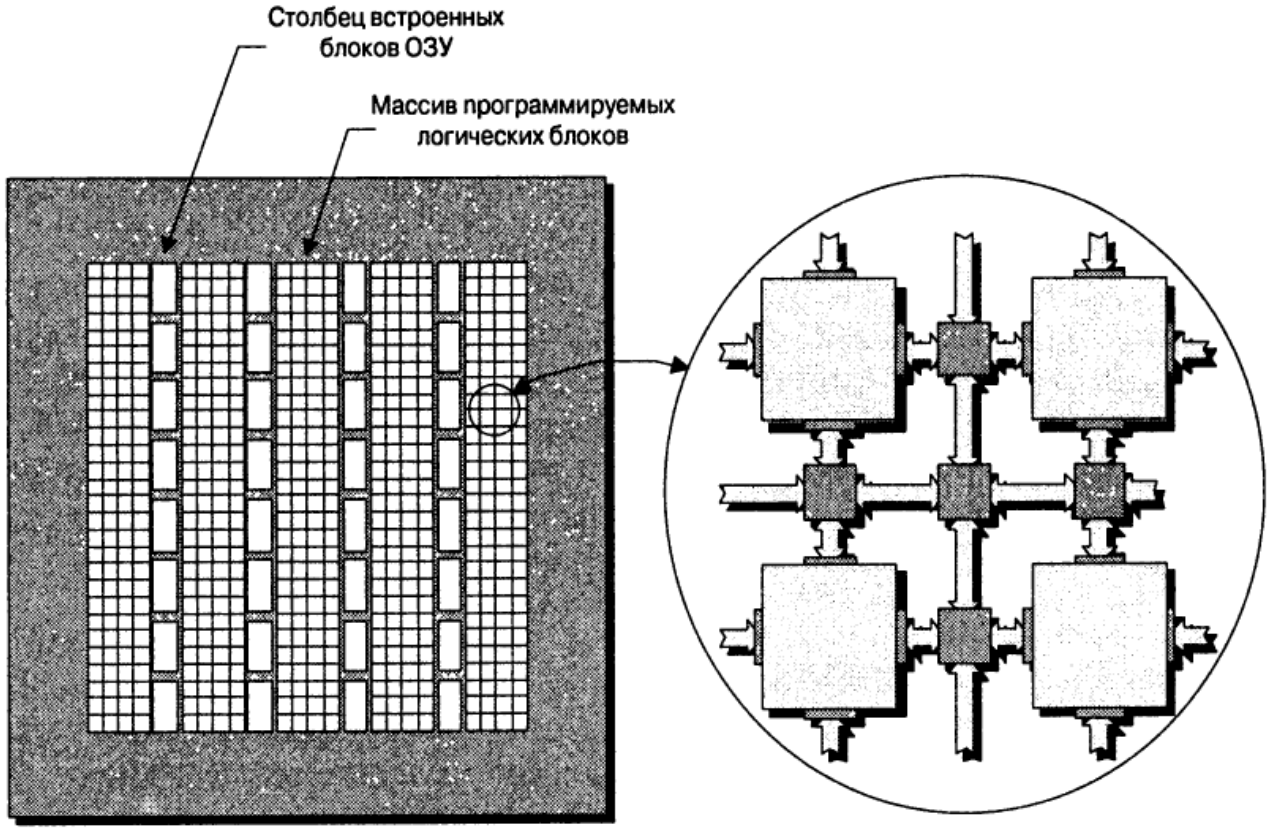

Рисунок 7 – Вид на кристалл со столбцами встроенных блоков ОЗУ

1.5 Встроенные умножители, сумматоры, блоки умножения с накоплением

Некоторые типы функции, например умножители, по своей сути являются довольно медленными, если их реализовывать с помощью большого количества программируемых логических блоков соединенных вместе. Поэтому многие ПЛИС содержат специальные аппаратные блоки умножения. Эти блоки обычно расположены в непосредственной близости от блоков встроенного ОЗУ (рисунок 8).

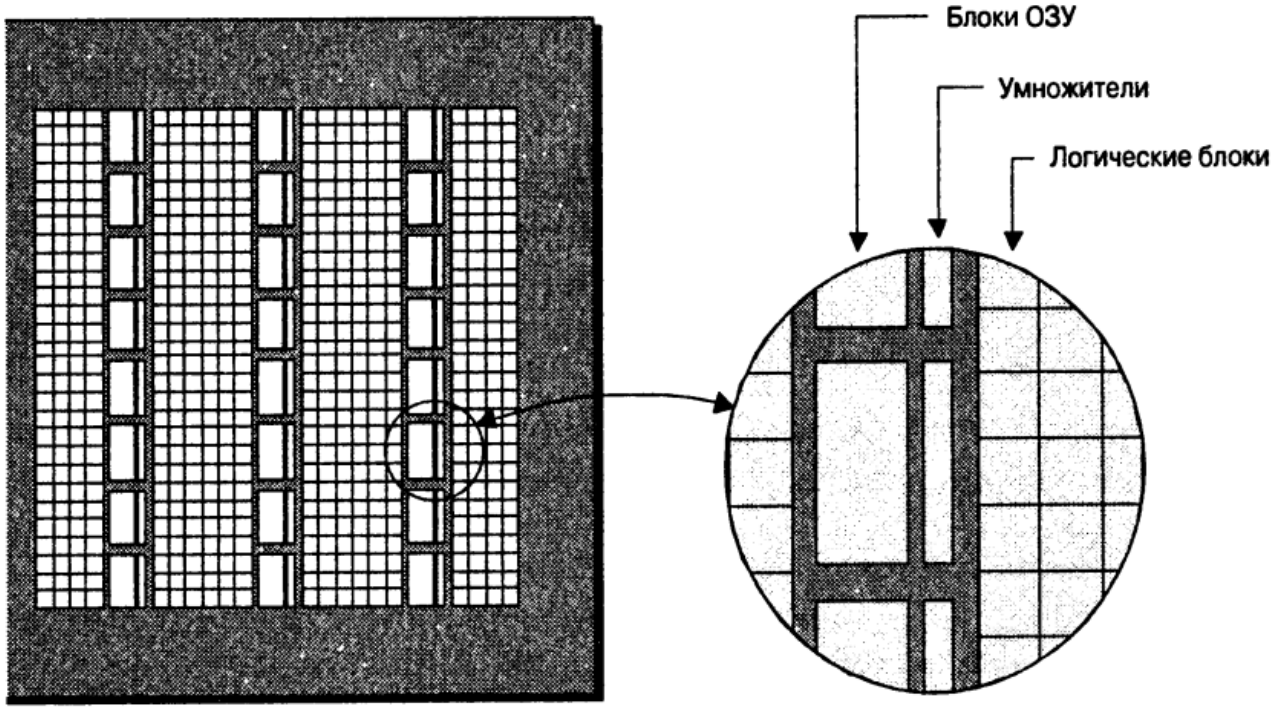

Рисунок 8 – Вид на кристалл со столбцами встроенных умножителей и блоков ОЗУ

Некоторые производители ПЛИС также предлагают выделенные сумматоры. В то же время, одной из самых распространенных операции, применяемых в ЦОС, является *умножение с накоплением* (*multiply-and-accumulate* или *MAC*)(рисунок 9).

Жизнь становится проще, когда ПЛИС содержит *встроенные сумматоры*, а отдельные х виды даже поддерживают *встроенные умножители с накоплением*.

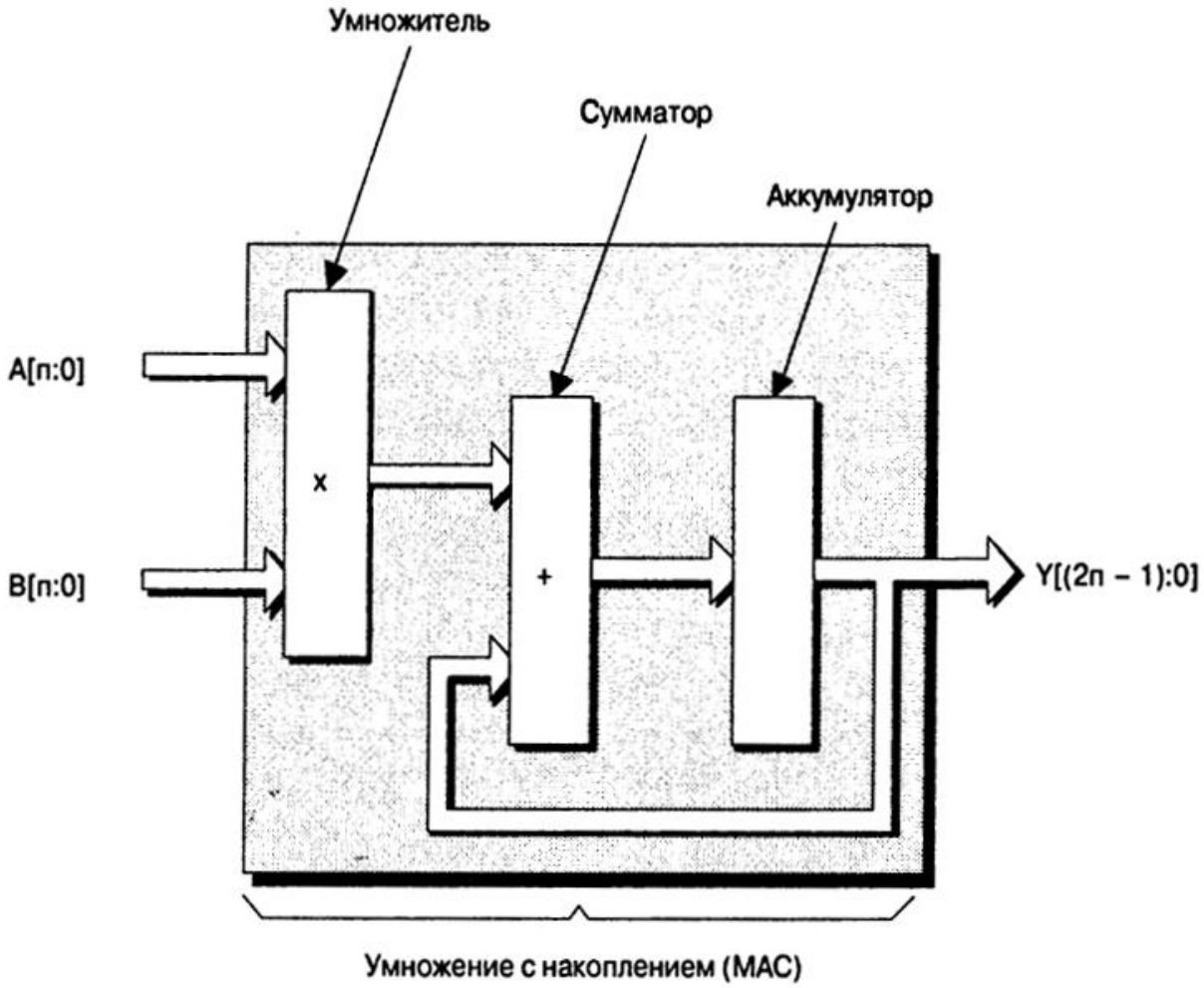

Рисунок 9 – Функции, формирующие операцию умножения с накоплением

1.6 Аппаратные и программные встроенные микропроцессорные ядра

Важной особенностью ПЛИС является то, что в них почти все части электронного устройства могут быть реализованы аппаратно (с использованием логических вентилей, регистров и т.д.) или программно (в виде инструкции микропроцессора).

Аппаратные микропроцессорные ядра изготавливают как отдельные предопределенные блоки. Существует 2 способа интеграции таких ядер в ПЛИС. Первый предусматривает расположение ядра в виде полосы (рисунок 10). При таком подходе все компоненты обычно формируются на одном кремневом кристалле, хотя они могут быть выполнены и на двух кристаллах и размещены в виде *многокристального модуля*. Главная часть ПЛИС также включает встроенные блоки ОЗУ, умножители и другие рассмотренные выше блоки.

Преимущества такой реализации проявляется в главной части ПЛИС, которая получается идентичной для устройств со встроенной и без встроенного микропроцессорного ядра. Другой плюс в том, что разработчики могут связать все дополнительные функции в одну полосу для дополнения микропроцессорного ядра.

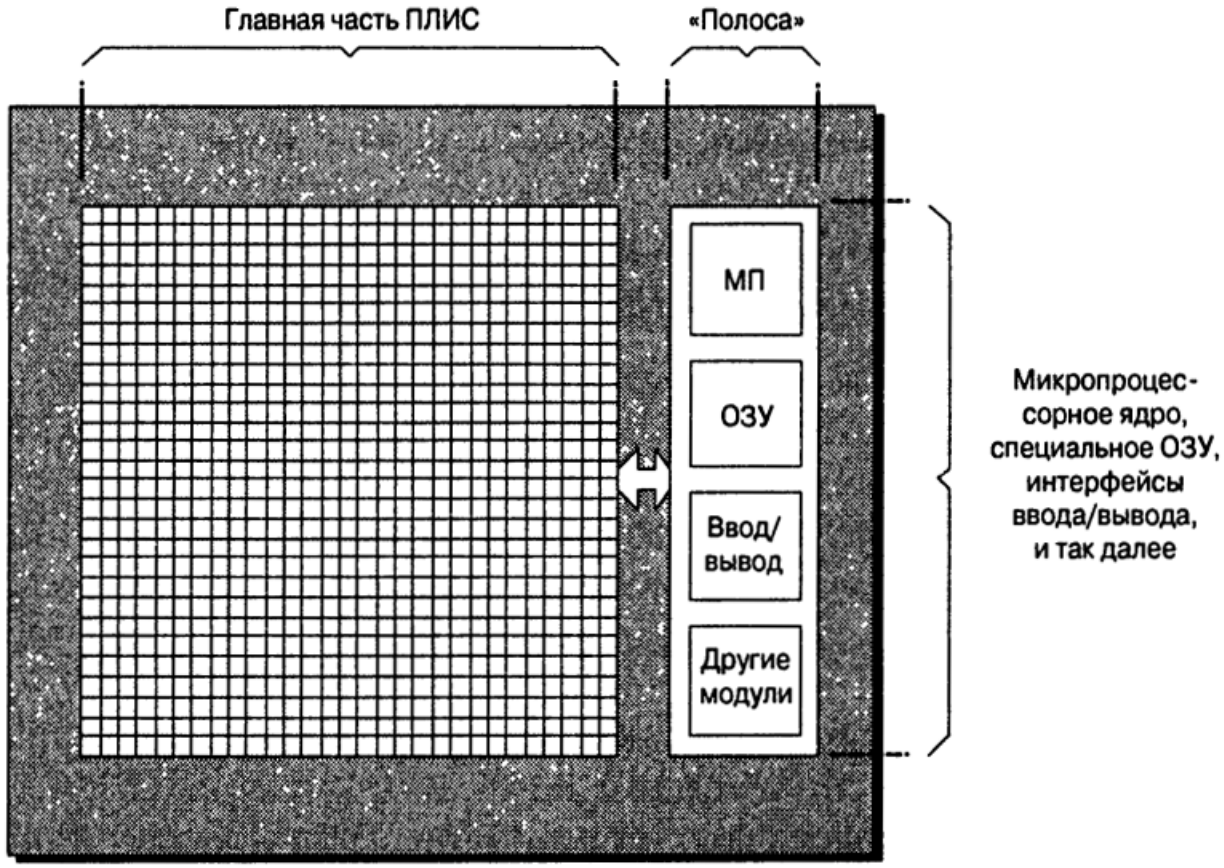

Рисунок 10 – Вид на кристалл со встроенным ядром, находящимся за пределами главной части

Другим из возможных решений является встраивание одного или более микропроцессорных ядер прямо в главную часть ПЛИС (рисунок 11). При использовании этого способа используемые средства проектирования должны учитывать присутствие микропроцессоров в структуре микросхемы.

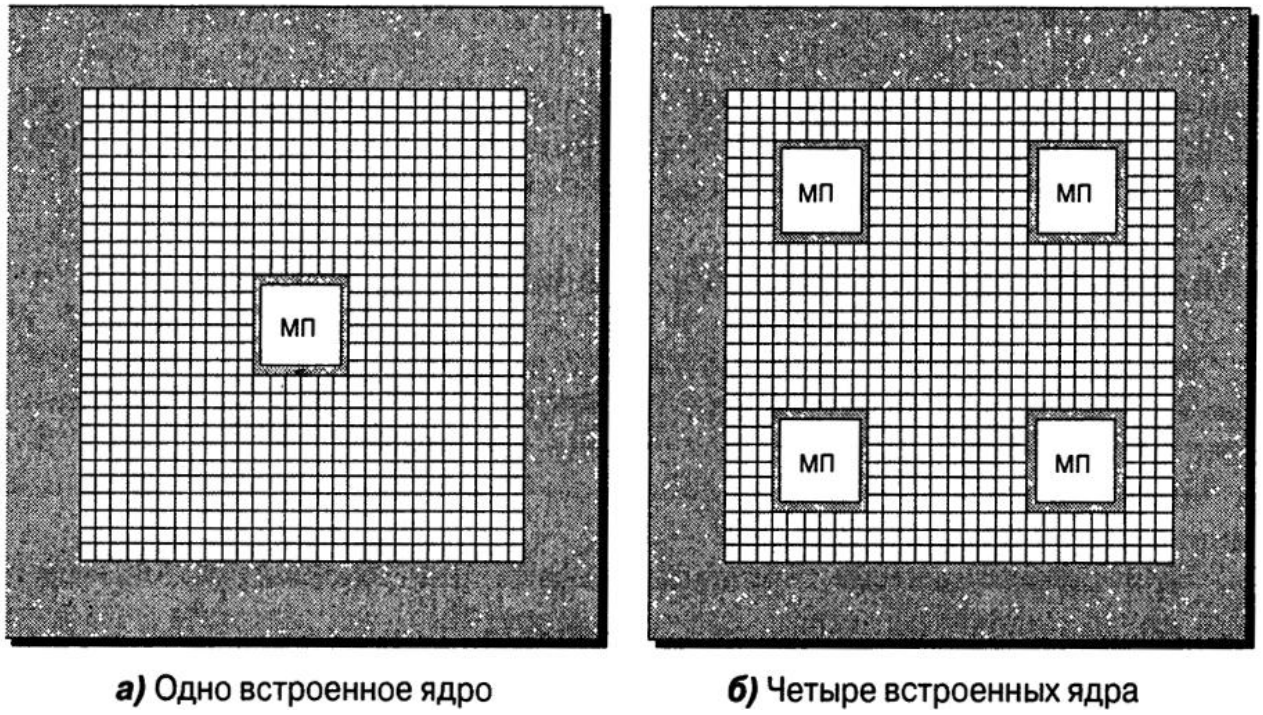

Рисунок 11 – Вид на кристалл с ядрами, встроенными внутрь главной части

Память, используемая ядром, формируется из встроенных блоков ОЗУ, а любые функции сопряжения реализуются с помощью групп программируемых логических блоков общего назначения. Плюсы такой схемы в том, что размещение микропроцессорного ядра в непосредственной близости к главной части ПЛИС обеспечивает ей преимущество в скорости.

Кроме физического встраивания микропроцессора в структуру кристалла, можно сконфигурировать группу программируемых логических блоков для работы в качестве микропроцессора. Такую группу блоков обычно называют *программным ядром*. Программные ядра проще и медленнее, чем их аппаратные аналоги. Однако у них есть одно преимущество – при необходимости можно реализовать ядро ли несколько ядер в том объеме, который можно достичь пока не будут исчерпаны все ресурсы в виде программируемых логических блоков.

1.7 Дерево синхронизации и диспетчеры синхронизации

Все синхронные элементы внутри ПЛИС, например регистры внутри программируемого логического блока, сконфигурированные для работы в виде триггеров, необходимо синхронизировать с помощью тактового сигнала. Тактовые сигналы обычно вырабатываются за пределам микросхемы и поступают в неё через специальные входы синхронизации, а затем распределяются через специальные устройства и подаются на соответствующие регистры.

На рисунке 12 показано упрощенное изображение дерева синхронизации. Название *дерево синхронизации* возникло потому, что главный синхросигнал разветвляется подобно ветвям дерева, при этом триггеры могут рассматриваться как «листья» на концах веток. Такая структура вселяет уверенность в том, что все триггеры увидят свои тактовые сигналы одновременно. Если бы тактовые сигналы распространялись по одному длинному проводнику, синхронизируя все триггеры поочередно, триггер, расположенный ближе к выводу синхронизации микросхемы увидел бы синхроимпульс намного раньше, чем последний триггер в этой цепочке (фазовый сдвиг).

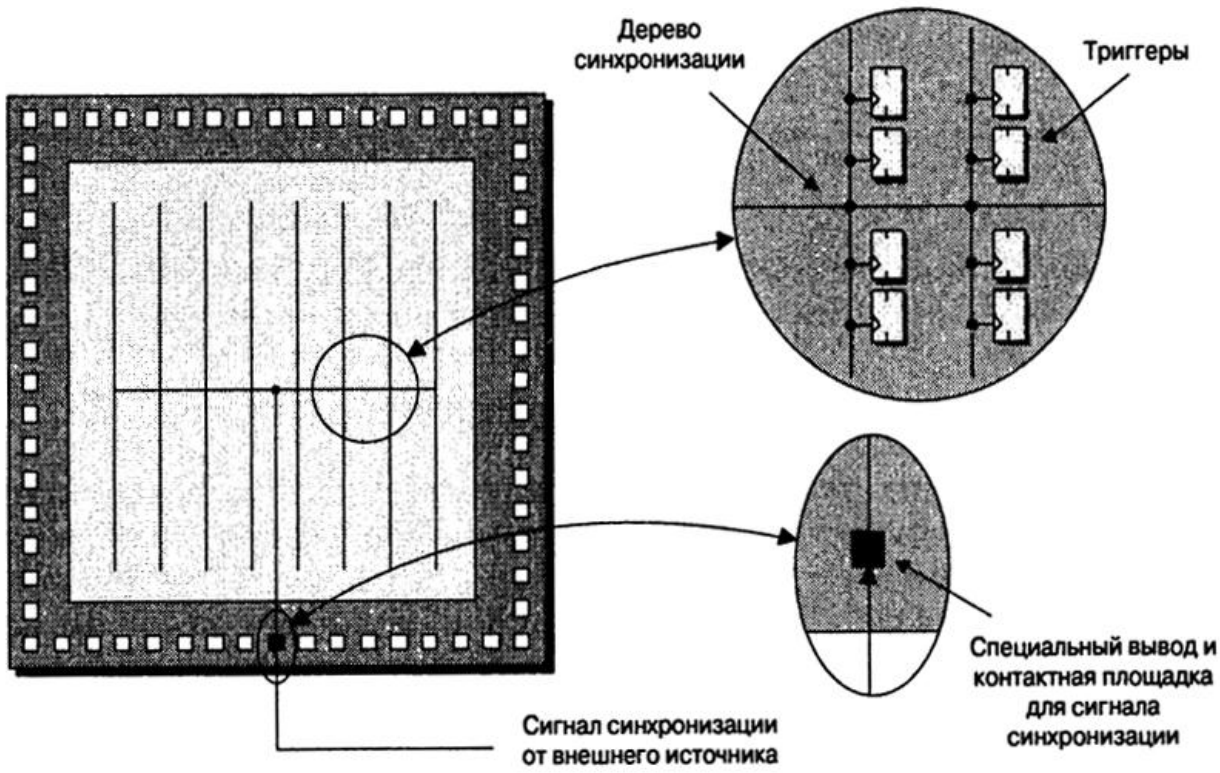

Рисунок 12 – Простое дерево синхронизации

Дерево синхронизации реализуется с помощью специальных проводников, которые отделены от внутренних соединений общего назначения. Принцип действия, рассмотренный выше, на самом деле сильно упрощен. На практике в микросхемах существует множество выводов синхронизации (неиспользуемые выводы синхронизации могут быть использованы как выводы общего назначения), а внутри устройства имеются множественные домены синхронизации (деревья синхронизации).

Вывод синхронизации микросхемы может быть подключен к дереву синхронизации. Однако, как правило, этот вывод подключают не напрямую к дереву синхронизации, в к входу устройства (или блока) управления синхронизацией, называемого *диспетчером синхронизации*, который генерирует *дочерние тактовые сигналы* (рисунок 13). Дочерние тактовые сигналы могут использоваться для управления внутренними деревьями (доменами) синхронизации ли для выдачи сигналов на внешние выводы микросхемы, которые, в свою очередь, можно использовать для синхронизации других устройств, расположенных на печатной плате. Каждое семейство микросхем ПЛИС располагает собственным типом диспетчера синхронизации, и в одном устройстве могут находиться множество модулей диспетчера синхронизации.

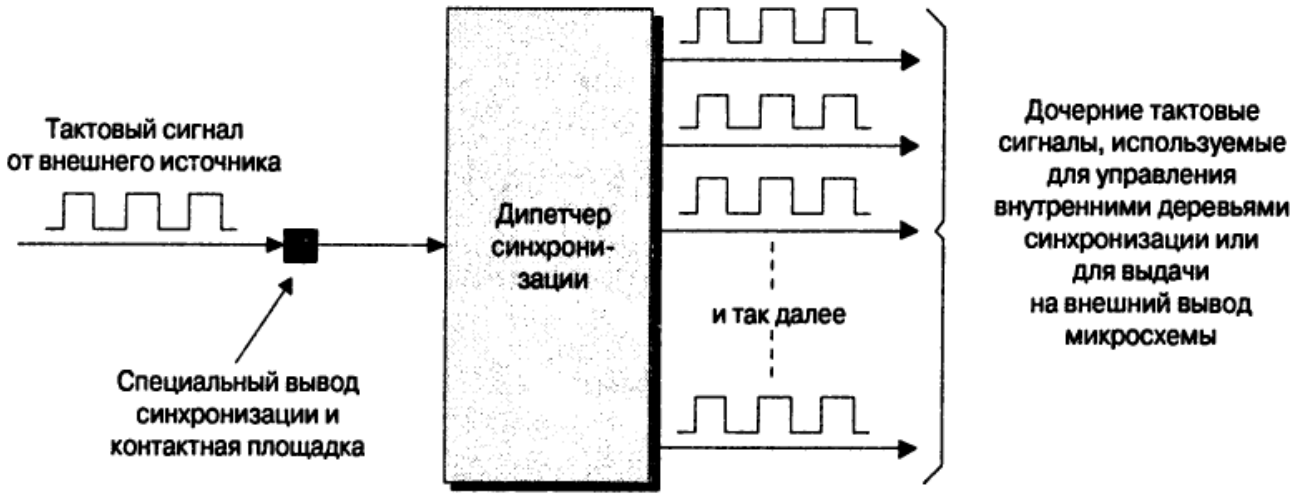

Рисунок 13 – Диспетчер синхронизации, генерирующий дочерние тактовые сигналы

1.8 Ввод/вывод общего назначения

Современные ПЛИС могут содержать 1000 и более выводов, которые располагаются по всему корпусу микросхемы. Аналогично они подходят к кристаллу внутри корпуса микросхемы. Внутрь корпуса кристалла устанавливают в перевернутом виде. Это позволяет подсоединять общий провод, проводники питания, синхронизации и сигналов ввода/вывода к любой точке на поверхности. Однако предположим, что все выводы на кристалле расположены по его контуру, как и было на самом деле много лет назад.

Ввод/вывод общего назначения ПЛИС может быть сконфигурирован для приема и передачи сигналов, соответствующих любому стандарту. Эти сигналы ввода/вывода разделяются на несколько банков. Будем полагать, что существует восемь банков, пронумерованных от 0 до 7 (рисунок 14). Каждый банк может быть индивидуально сконфигурирован для поддержки определенных стандартов ввода/вывода. Таким образом ПЛИС имеет возможность работать с устройствами, используя многочисленные стандарты ввода/вывода, эти микросхемы могут служить интерфейсом между различными стандартами ввода/вывода, а также осуществлять связь между различными протоколами, которые могут основываться на частных электрических стандартах.

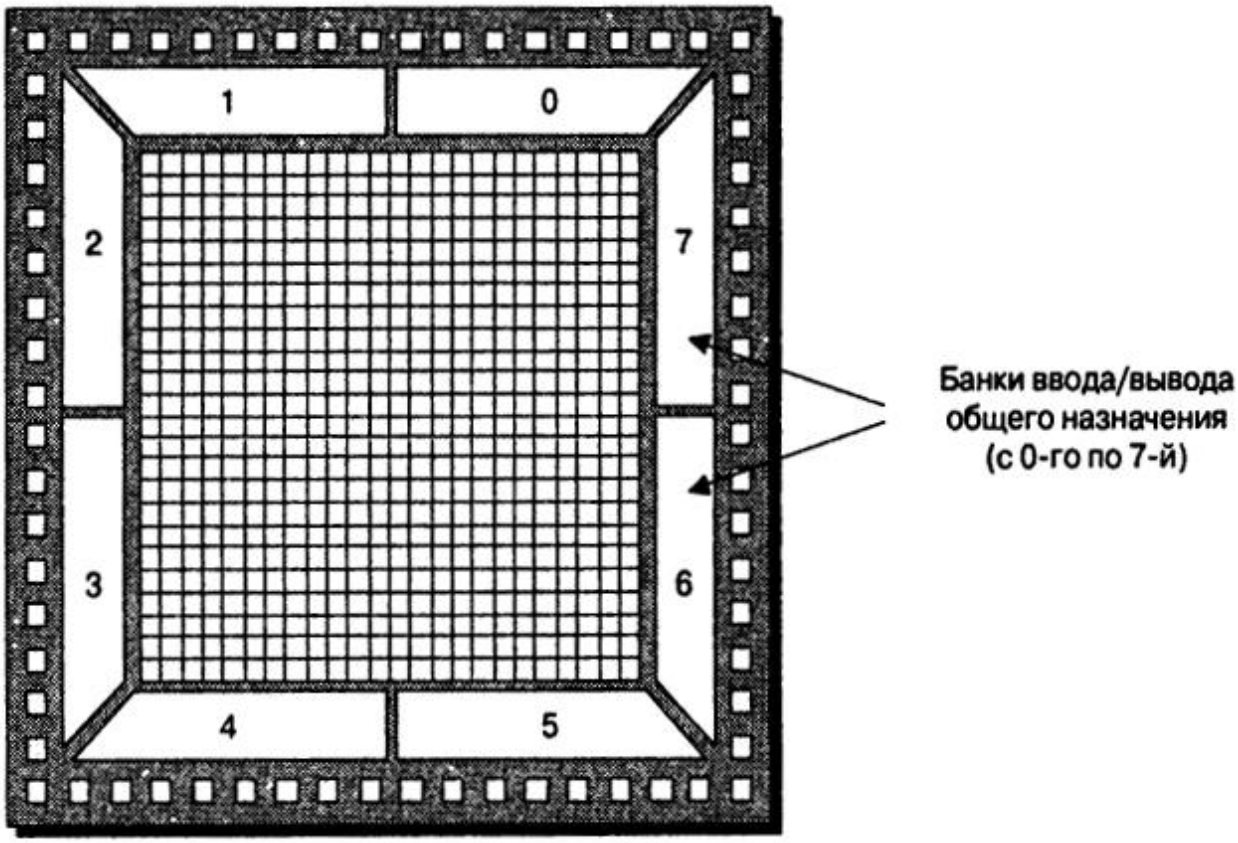

Рисунок 14 – Вид на кристалл с банками ввода/вывода

Сигналы в современных печатных платах часто обладают быстрым *временем переключения*. Имеется в виду время, которое необходимо сигналу для переключения с одного логического уровня на другой. Чтобы предотвратить отражение сигналов, приводящее к возникновению пульсации, к выводам микросхемы необходимо подключать соответствующие согласующие резисторы, т.е. терминаторы. В прошлом эти резисторы применялись как отдельные компоненты, которые размещались на ПП за пределами ПЛИС. Но поскольку количество выводов у микросхемы увеличивалось, а шаг уменьшался, то поэтому современные ПЛИС позволяют использовать внутренние согласующие резисторы, или внутренние терминаторы, сопротивление которых может задаваться пользователем.

1.9 Основные производители

К основам поставщикам ПЛИС, в частности ПЛИС FPGA, относятся: Actel, Altera, Anadigm, Atmel, Lattice Semiconductors, Xilinx.

2 ТЕХНИЧЕСКИЕ ПАРАМЕТРЫ

Рассмотрим технические характеристики FPGA ПЛИС на примере ПЛИС фирмы Xilinx, а именно характеристики FPGA 7-серии, выполненных по технологическим нормам 28 нм.

Ведущие производители ПЛИС, как правило, своевременно переходят к новым технологическим процессам, поэтому новые семейства FPGA появляются достаточно регулярно. Современная технология позволяет расширить функциональные возможности, увеличить логическую емкость и тактовую частоту. Соответственно, от каждого нового семейства FPGA разработчики ожидают улучшений по всем параметрам, включая появление новых возможностей, увеличение максимального логического объема, тактовой частоты, снижение потребляемой мощности и цены.

Для новой 28нм технологии Xilinx представляет три новых семейства FPGA. Из них только Virtex-7 является продолжением существующей линейки высокопроизводительных ПЛИС, а два других пришли на замену серии Spartan. Это семейства Artix-7 и Kintex-7, причем первое из них предназначено для приложений с высоким тиражом и отличается малым энергопотреблением и невысокой стоимостью, а Kintex представляет собой семейство с уклоном в сторону цифровой обработки сигналов. В качестве ключевого свойства нового поколения FPGA отмечается унификация программируемых ресурсов новых семейств. Xilinx предполагает, что для нового поколения FPGA станет возможной быстрая миграция между семействами Virtex/Kintex/Artix. При сохранении архитектуры проекта разработчик сможет выбирать наиболее подходящее сочетание производительности и потребляемой мощности (рисунок 15).

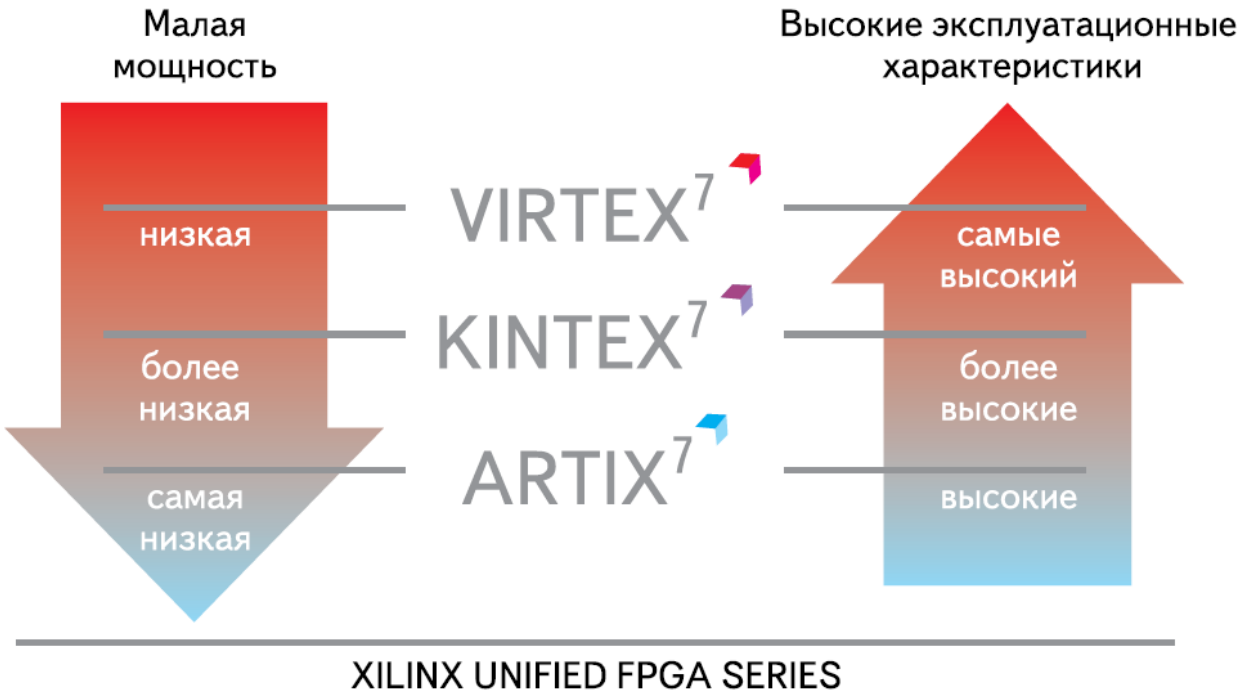

Рисунок 15 – Характеристики производительности и потребляемой мощности семейств серии 7

На основании спецификации, опубликованной в документе [4], можно показать характеристики и состав семейств. В таблице 1 приведены сводные характеристики трех семейств.

Таблица 1 – Сводные характеристики семейств FPGA серии 7

Максимальное значение параметра	Artix-7	Kintex-7	Virtex-7
Логические ячейки, тыс.	352	407	1955
Блочная память, Мбит	12	29	65
Секций DSP	700	1540	3960
Пиковая производительность ЦОС для фильтров с симметричными коэффициентами, ГМАС/с	504	1848	4752
Приемопередатчики	4	16	80
Пиковая скорость передачи, Гбит/с	30	330	1886
Интерфейсы PCI Express	Gen1x4	Gen2x8	Gen3x8
Скорость обмена по интерфейсам памяти, Мбит/с	800	2133	2133
Внешние вывод	450	500	1200

Конфигурирование. ПЛИС серии 7 поддерживают различные режимы загрузки конфигурации — SPI x1, x2 и x4, BPI x8 и x16. С учетом большого объема данных для конфигурирования (от 0,6 до 54 Мбайт) может оказаться удобным загружать ПЛИС с помощью внешнего процессора, имеющего доступ к устройству хранения данных большого объема. ПЛИС семейства Artix с объемом более 100 000 ячеек, а также все ПЛИС Kintex и Virtex поддерживают шифрование конфигурационной последовательности по алгоритму AES с 256-битным ключом. ПЛИС семейства Virtex-7 поддерживают частичную реконфигурацию.

3 ОСОБЕННОСТИ КОНФИГУРАЦИИ

Конфигурационный, или *битовый, файл* – это файл, который содержит информацию, предназначенную для загрузки в микросхему, т.е. для её программирования с целью выполнения определенных функции.

Все схемы, все содержимое FPGA хранится в стоящей рядом крошечной по своим размерам микросхеме последовательной памяти, которая каждый раз по включению питания загружает прообраз схемы внутрь FPGA (или, как принято говорить, конфигурирует FPGA), обмениваясь с ней тактовыми и статусными сигналами.

В свое время первые ПЛИС использовали инструмент, называемый *конфигурационным портом*. Даже сегодня, когда доступны более сложные методы, подобные JTAG-интерфейсу, конфигурационный порт все ещё находит широкое применение.

Начнем рассмотрение с небольшой группы специальных контактов, или выводов, режима конфигурации, которые используются для установки режима конфигурации устройства. В первых ПЛИС для этих целей использовались только 2 вывода, которые позволяли задавать 4 режима конфигурации (таблица 2).

Таблица 2 – Четыре первичных режима конфигурации

Выводы установки режима	Наименование режима
00	Последовательная загрузка, ПЛИС в режиме ведущий
01	Последовательная загрузка, ПЛИС в режиме ведомый
10	Параллельная загрузка, ПЛИС в режиме ведущий
11	Параллельная загрузка, ПЛИС в режиме ведомый

Следует иметь в виду, название режимов, а также соответствие кода на выводах установки режима и названием режима приведены только в качестве примера. Настоящие коды и названия режимов у каждого поставщика свои.

Последовательная загрузка, ПЛИС в режиме ведущий. Вероятно, этот режим самый простой з всех режимов программирования. Раньше для работы в этом режиме использовались внешние микросхемы ППЗУ, впоследствии их заменили на СППЗУ, затем ЭСППЗУ и теперь, как правило, используются устройства на ячейках flash-памяти. Эти компоненты специальной памяти содержат один вывод для выхода данных, который подсоединяется к выводу ввода конфигурационных данных ПЛИС (рисунок 16). ПЛИС также использует несколько сигналов для управления внешней памятью. К ним относятся сигнал сброса, который выдает ПЛИС при готовности ею начать чтение данных, а также синхросигнал, предназначенный для синхронизации конфигурационных данных.

Параллельная загрузка, ПЛИС в режиме ведущий. Во многих отношениях этот режим подобен предыдущему, но при этом данные читаются 8-битными кусками памяти, которая содержит также восемь выводов данных. Кроме обеспечения управляющих сигналов, ПЛИС формирует для внешней памяти сигналы адреса, которые используются для указания байта конфигурационных данных, который будет загружаться на следующем такте (рисунок 17). Подобная схема работает при условии, что ПЛИС содержит внутренний счетчик памяти, который используется для генерации адреса для внешней памяти.

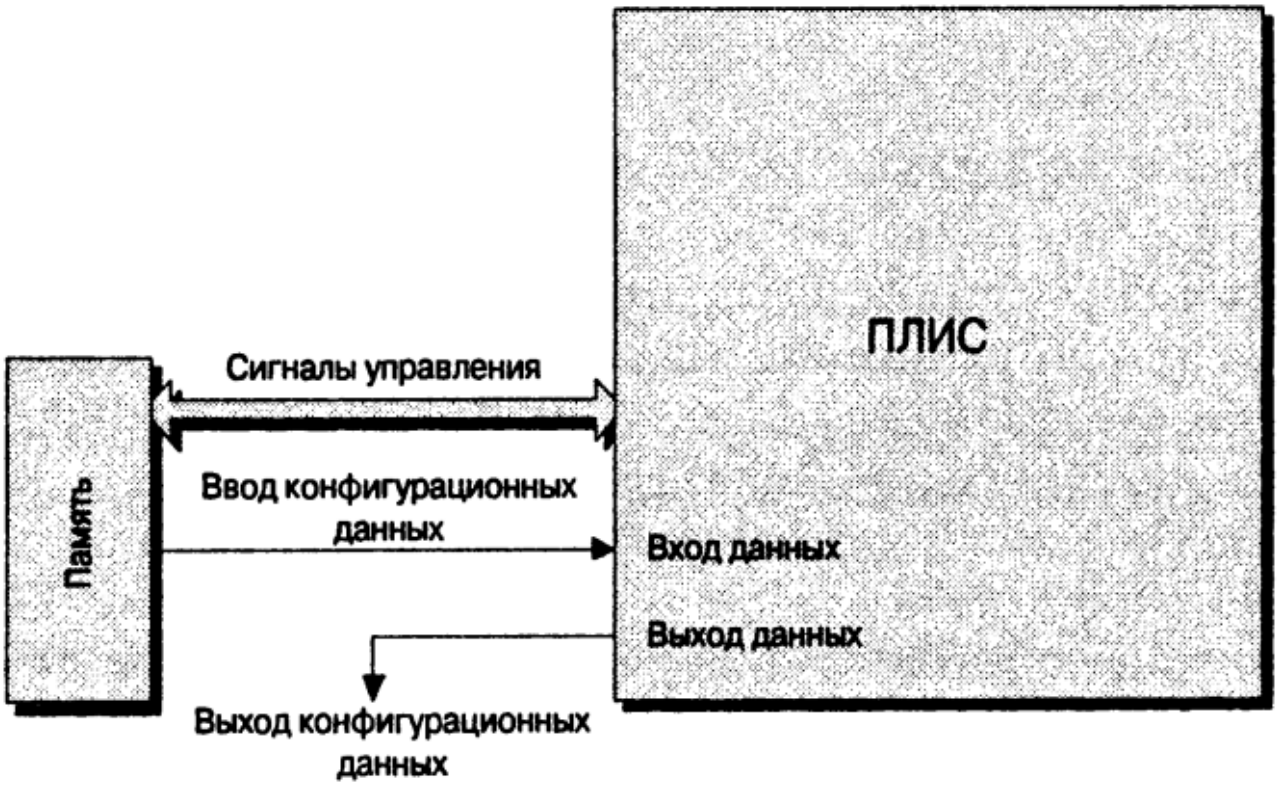

Рисунок 16 – Последовательная загрузка, ПЛИС в режиме ведущий

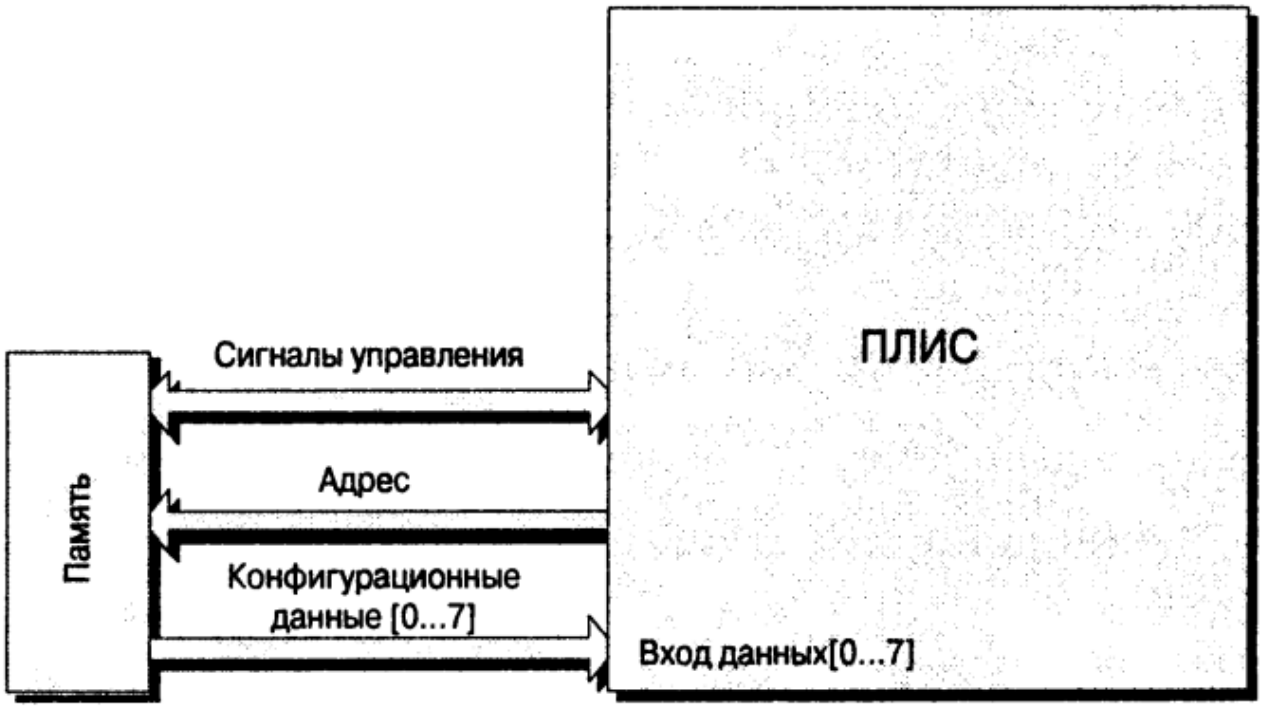

Рисунок 17 – Параллельная загрузка, ПЛИС в режиме ведущий

Параллельная загрузка, ПЛИС в режиме ведомый. Рассмотренные выше режимы конфигурирования, в которых ПЛИС выступала в роли ведущей довольно привлекательны благодаря своей простоте и непритязательности, так как работы требуются только ПЛИС и одна микросхема внешней памяти. Однако во многих ПП содержатся микропроцессоры, которые обычно используются для выполнения разнообразных внутренних задач. В этом случае разработчики могут решить использовать микропроцессоры для загрузки конфигурационных данных в ПЛИС (рисунок 18). Идея заключается в том, что микропроцессор является

управляющим устройством. В этом режиме микропроцессор сообщает ПЛИС, что он готов начать процесс конфигурирования, после этого производит чтение байта конфигурационных данных с определённого устройства памяти или с внешнего устройства, ли ещё откуда-то, записывает эти данные в ПЛИС, считывает следующий байт данных из памяти, записывает его в ПЛИС и так до окончания процесса конфигурирования.

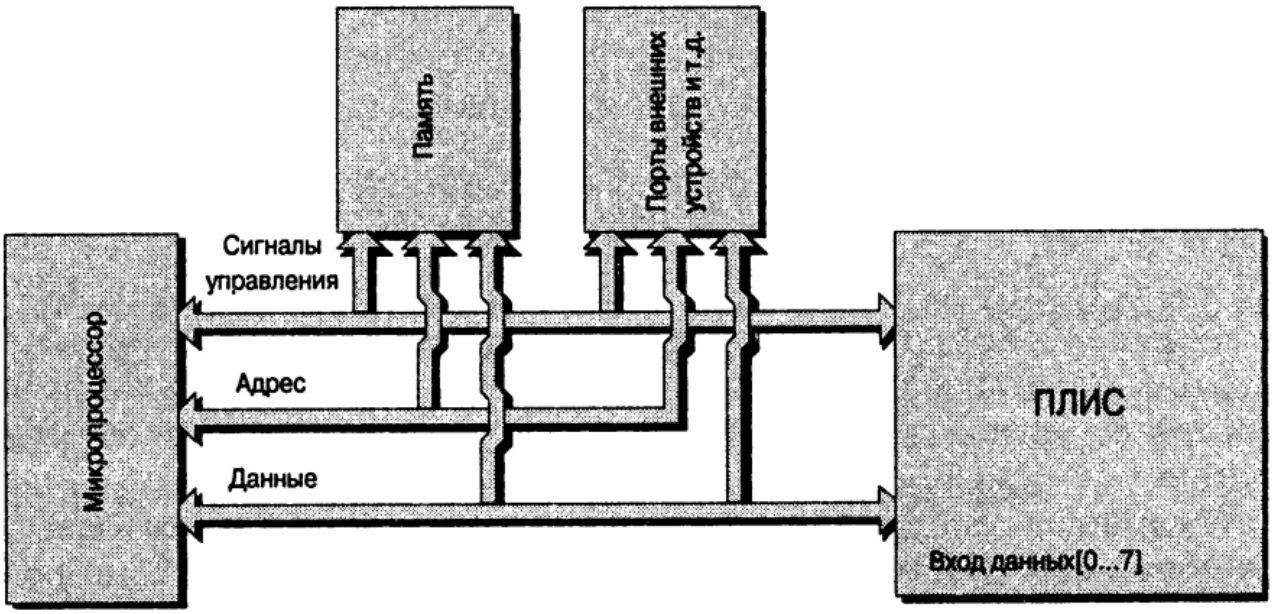

Рисунок 18 – Параллельная загрузка, ПЛИС в режиме ведомый

Последовательная загрузка, ПЛИС в режиме ведомый. Этот режим аналогичен своему «параллельному родственнику», но при этом для загрузки данных в ПЛИС используется только один бит. Микропроцессор по-прежнему производит чтение одного байта данных из устройства памяти и затем преобразует этот байт в последовательность битов для записи в ПЛИС.

ЗАКЛЮЧЕНИЕ

В данном реферате описана архитектура FPGA. Были рассмотрены мелко-, средне- и крупномодульные архитектуры (стр.6), показано два основных способа реализации программируемых логических блоков, используемых для формирования среднемодульных устройств на основе мультиплексоров и на основе таблиц соответствия (стр.6), показано, что одни и те же (почти одни и те же) элементы ПЛИС у разных производителей могут называться по-разному (логические ячейки фирмы Xilinx и логические элементы компании Altera) (стр.8), приведена «иерархия» ПЛИС (логические ячейки → секции → конфигурируемый логический блок)(стр.8), показано, что в современных ПЛИС есть блоки встроенного ОЗУ (стр.10), встроенные умножители, сумматоры, блоки умножения с накоплением (стр.11), аппаратные и программные встроенные микропроцессорные ядра (стр.12), дерево синхронизации и диспетчер синхронизации (стр.14), ввод/вывод общего назначения (стр.15). Также приведены основные поставщики ПЛИС (стр.16). Стоит отметить, что рассмотрение архитектуры приводилось так сказать в общем виде, без привязки к какой-либо серии ПЛИС какого-либо производителя.

Также в данном реферате были приведены основные параметры ПЛИС FPGA на примере ПЛИС фирмы Xilinx 7-ой серии (стр.17).

Рассмотрены особенности конфигурирования FPGA.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Максфилд К., Проектирование на ПЛИС. Курс молодого бойца. – М.: Издательский дом «Додэка-XXI», 2007 – 408 с: ил.
2. Steve Kilts, Advanced FPGA Design – Architecture, Implementation and Optimization, 2007
3. Зотов В.Ю., Проектирование встраиваемых микропроцессорных систем на основе ПЛИС фирмы Xilinx. – М.: Горячая линия – Телеком, 2006 – 520 с., ил.
4. http://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf