

Тема 1. История развития интегральных схем (от ASIC до FPGA). Примеры современного использования FPGA.

Что такое ПЛИС?

Чтобы ответить на этот вопрос в самом простом понимании необходимо хотя бы поверхностно понять, как функционируют цифровые устройства. Начнем обзор цифровых устройств.

Транзистор

Рассмотрим элементарный компонент цифровой схемы транзистор. Транзистор – это полупроводниковый компонент, использующийся для усиления или переключения сигнала. Был изобретен в 1947 году. Состоит из полупроводниковых областей с разной проводимостью и подключенным к ним металлическим выводам. Современные цифровые устройства (КМОП или CMOS логика [1]) в большинстве своем строятся на полевых транзисторах, впервые увидевших свет в 1950 году. Отличная справка по этому устройству есть на Википедии [2]. Вкратце, полевые транзисторы бывают двух типов: PMOS и NMOS (рис 1).

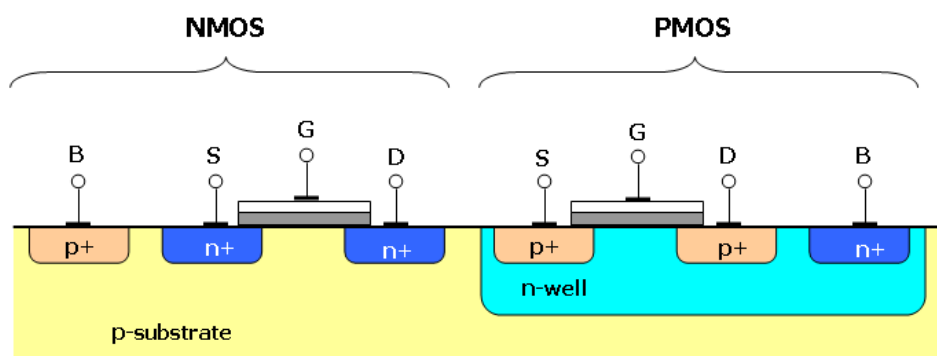


Рис 1. PMOS и NMOS транзисторы в разрезе. S – исток, D – сток, G – затвор. Вывод B соединяется с S для NMOS и с D для PMOS.

В обоих типах транзистора выделяют следующие части: исток (source – S), сток (drain – D), затвор (gate – G), подложку p-substrate, области с различными проводимостями n+ и p+ и колодец с проводимостью n (для PMOS транзистора). Source или исток является источником основным носителей заряда, drain или сток – это приемник основных носителей заряда. Для обоих типов затвор G электрически изолирован от подложки. Рассмотрим NMOS транзистор. S и D соединены с областями полупроводника с проводимостью n+. На подложку подается напряжение истока S. Если к затвору не прикладывать положительного потенциала, то проводимость между стоком и истоком очень мала. Однако при прикладывании положительного потенциала (больше потенциала истока) к затвору, между областями n+ образуется n-канал, по которому могут перемещаться основные носители заряда и проводимость возрастает. Аналогично для PMOS транзистора, S и D соединены с областями полупроводника с проводимостью p+. На колодец p-well подается напряжение стока D. Теперь, в отличие NMOS транзистора, если к затвору прикладывать потенциал равный потенциалу стока, то его проводимость будет мала. Если же приложить потенциал меньший чем потенциал стока, то его проводимость возрастет.

Лучше всего понять это на примере инвертора – простейшего элемента НЕ (рис 2).

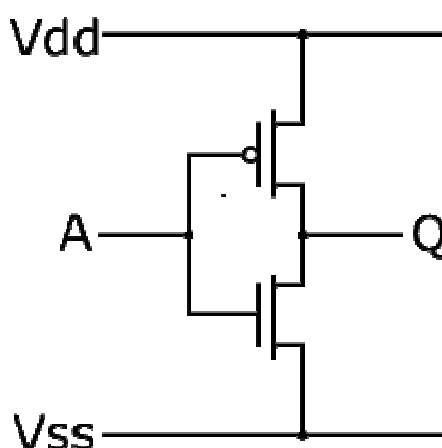


Рис 2. Инвертор в CMOS логике

Пусть V_{ss} – напряжение логического нуля, V_{dd} – напряжение логической единицы, A – входное напряжение, Q – выходное. Нижний транзистор – это NMOS транзистор, верхний (с кружком на затворе) – PMOS транзистор. Исток NMOS соединен с V_{ss} , а сток PMOS соединен с V_{dd} . Если $A = '0'$, то нижний (NMOS) транзистор закрыт, а верхний (PMOS) открыт, и в точке Q устанавливается потенциал равный V_{dd} ('1'). Если $A = '1'$, то наоборот верхний транзистор закрыт, а нижний открыт, и в точке Q устанавливается потенциал V_{ss} ('0'). Таким образом наше устройство ведет себя как инвертор.

На базе КМОП транзисторов можно построить любой логический элемент. Например элемент И-НЕ (рис 3)

A	B	Q
0	0	1
0	1	1
1	0	1
1	1	0

Рис 3. Таблица истинности элемента И-НЕ

можно создать, соединив 4 транзистора следующим образом (рис 4):

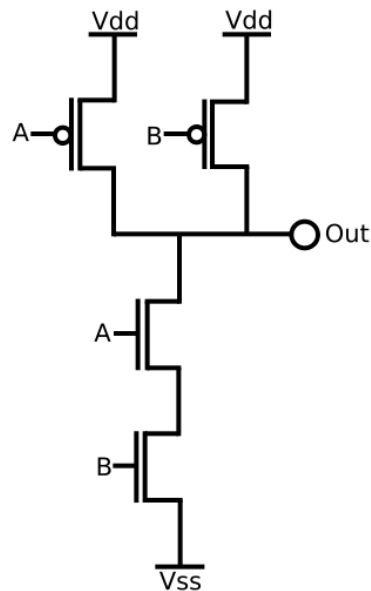


Рис 4. Элемент И-НЕ в CMOS логике

По рисунку видно, что если хотя бы один из входов A или B равен '0', то один из верхних транзисторов PMOS открываются, и на выход подается напряжение $V_{dd} = '1'$. Если же они оба равны '1', то верхняя часть схемы закрыта, зато открывается нижняя, и на выход подается $V_{ss} = '0'$.

Микросхемы

Соединив много транзисторов, можно создать цифровую схему. Получается правда очень громоздко. Тогда родилась идея объединения нескольких транзисторов на одном куске полупроводника (кремния). Так была создана первая интегральная схема (ИС) 1958 г. Первые ИС

включали в себя совсем мало транзисторов (десятки), но потом их количество стало расти экспоненциально. Закон Мура говорит об удвоении количества транзисторов на единицу площади кристалла каждые 18-24 месяца. Исторически можно выделить этапы роста плотности транзисторов на интегральной схеме (ИС):

Малая ИС (МИС) — до 100 элементов в кристалле.

Средняя ИС (СИС) — до 1000 элементов в кристалле.

Большая ИС (БИС) — до 10000 элементов в кристалле.

Сверхбольшая ИС (СБИС) — до 1 миллиона элементов в кристалле.

Ультрбольшая ИС (УБИС) — до 1 миллиарда элементов в кристалле.

Процессор Pentium 4 содержит несколько сотен миллионов транзисторов

Технология изготовления современных ИС очень сложна. На разработку проекта нового чипа уходит около полутора лет, при этом затраты могут достигать до миллиарда долларов США. Сначала необходимо разработать проект микросхемы в САПР (ПО система автоматического проектирования). Раньше необходимо было задавать расположение каждого транзистора на куске кремния. Теперь, когда типовые линейные размеры транзисторов приближаются к 100 нм (для 20 нм тех процесса), а число транзисторов на чипе перевалило за миллиард, задавать руками расположение отдельного транзистора не представляется возможным. В современном процессе проектирования используются библиотеки элементов – уже соединенные транзисторы вместе. Транзисторы объединяются в логические элементы и функции, логические элементы в функциональные узлы, узлы в блоки и интерфейсы. Разработчик может пользоваться уже готовыми элементами при проектировании, и это существенно сокращается время разработки.

Непосредственно сам процесс изготовления микросхемы занимает до 8ми недель и происходит на специализированной фабрике. Сама ИС – это слоистая структура (10-15 слоев), слои бывают полупроводниковыми и металлическими. На полупроводниковых слоях изготавливаются транзисторы, на слоях металлизации – соединения между ними и внешние выводы. На полупроводниковых слоях необходимо определить месторасположение каждого транзистора, т.е. указать, где будут области с положительной проводимостью, а где – с отрицательной. Структура слоев задается процессом, называемым фотолитография. Рассмотрение фотолитографии не входит в данный курс.

Классификация микросхем.

Теперь, когда мы поняли, как примерно делают микросхемы, мы можем рассмотреть их классификацию. Это приблизит нас к ответу на вопрос «Что такое ПЛИС?». Глобально современные микросхемы можно разделить на два класса: с жестко заданной структурой (заказные микросхемы или ASIC – Application Specific Integrated Circuit) и структуру которых можно менять или перепрограммировать. Рассмотрим ASIC. Сейчас различают 4 класса технологий, по которым делают микросхемы: вентильные матрицы, структурированные ASIC, схемы на стандартных элементах и полностью заказные микросхемы.

Полностью заказные микросхемы

Проектировщики микросхемы по этой технологии могут создавать все слои (полупроводниковые и металлизации) так как им угодно, т.е. разработчик может управлять расположением каждого транзистора в микросхеме. Это очень трудоемкий долгий и дорогой процесс. Но при этом получаются наиболее эффективные схемы. Тем не менее, сегодня редко можно встретить микросхему, выполненную по этой технологии. Этот подход чаще используется для проектирования малых критически важных частей будущей микросхемы или при создании базовых логических элементов других технологий.

Схемы на стандартных элементах

Устройство создается из набора заранее определенных компонентов, выполняющих простые логические функции и выполненных по технологии полностью заказных микросхем. Эти компоненты образуют так называемую библиотеку элементов. Библиотека элементов создается производителем микросхемы (заводом), а заказчик или разработчик микросхемы по этой технологии вправе брать любые элементы из этой библиотеки и размещать их в любых частях кристалла. При этом разработка микросхемы становится легче, т.к. проектировщик не управляет расположениями отдельных транзисторов, а оперирует уж готовыми логическими элементами. Хотя при этом подходе все равно необходимо изготавливать все слои. По этой технологии сейчас выполнено большинство высокотехнологичных микросхем.

Вентильные матрицы

Устройство состоит из готовых ячеек, находящихся в определенных местах и выполняющих определенные логические функции. Разработчик вправе только соединять эти элементы произвольным образом для достижения нужной ему функциональности. Этот подход значительно упрощает создание схемы, так как необходимо создать только нужную схему соединений элементов с помощью слоев металлизации (полупроводниковые слои уже созданы заранее). В итоге уменьшается время создания схемы и ее стоимость, однако часть ресурсов изначальной заготовки может оказаться не востребованной, и многие параметры схемы (максимальная

таковая частота, энергопотребление...) будут хуже по сравнению с той же схемой, но выполненной по технологии, например, схем на стандартных элементах.

Структурированные ASIC

Читателю предлагается самостоятельно прочитать главу 3 книги Курс молодого бойца Максфилда и разобрать раздел о структурированных микросхемах. Объяснить отличие от вентиляных матриц. Также предлагается познакомиться с продукцией фирмы EAsic [3], которая предлагает смешанный подход к проектированию заказных микросхем.

Программируемые логические интегральные схемы (ПЛИС)

ПЛИС являются ярким представителем микросхем, структуру которых можно менять. Состоят они из набора так называемых логических блоков и матрицы программируемых соединений между ними.

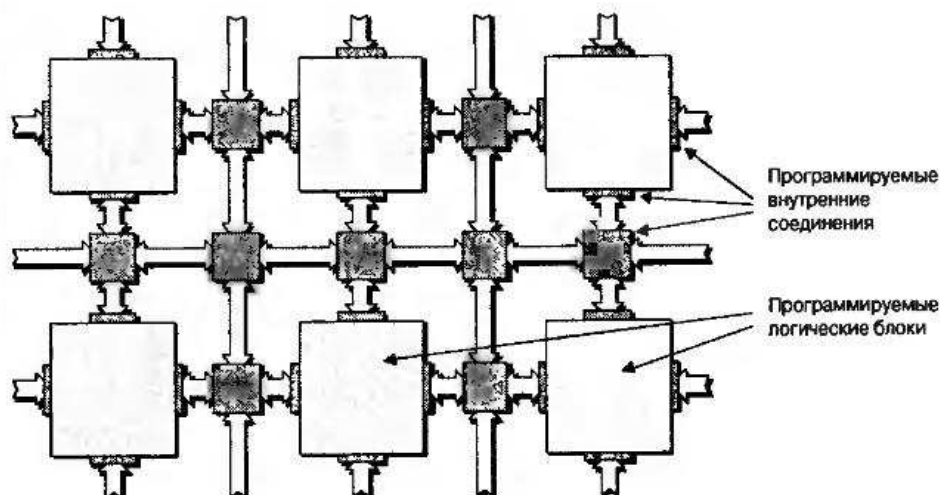


Рис 5. Структура ПЛИС

В отличие от ячеек, используемых в технологиях схем на стандартных элементах или вентиляных матриц, логические блоки ПЛИС, во-первых, все одинаковые (т.е. нету никаких библиотек элементов), во-вторых, имеют значительно больший размер, и наконец, могут быть «запрограммированы» на выполнение разных функций. Как правило, логический блок ПЛИС может быть настроен на реализацию любой логической функции бти элементов + для хранения одного бита данных в триггере. На рис.6 представлена упрощенная структура логического блока ПЛИС, включающая в себя таблицу соответствия LUT на 3 входа (для простоты объяснения), триггер и мультиплексор.

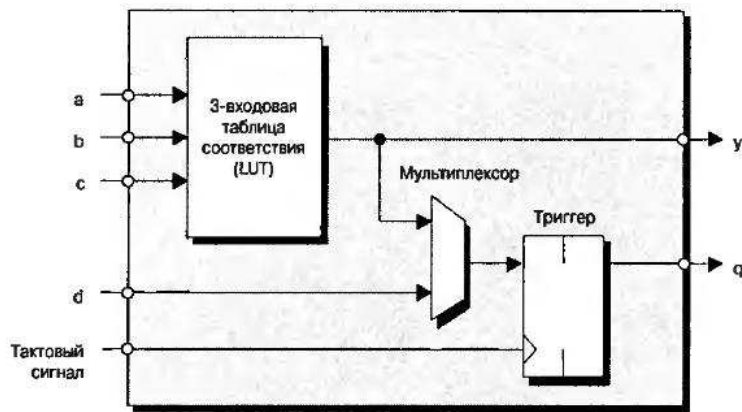


Рис 6. Упрощенная структура логического блока ПЛИС

Любая логическая функция может быть представлена своей таблицей истинности. Таблица истинности может быть представлена как массив памяти, адресами которой являются аргументы логической функции, а значения, записанные в ячейки этого массива памяти, будут являться значениями функции. Массивы памяти, представляющие каждую таблицу истинности в каждом логическом блоке, включены в конфигурационную память (см рис 7).

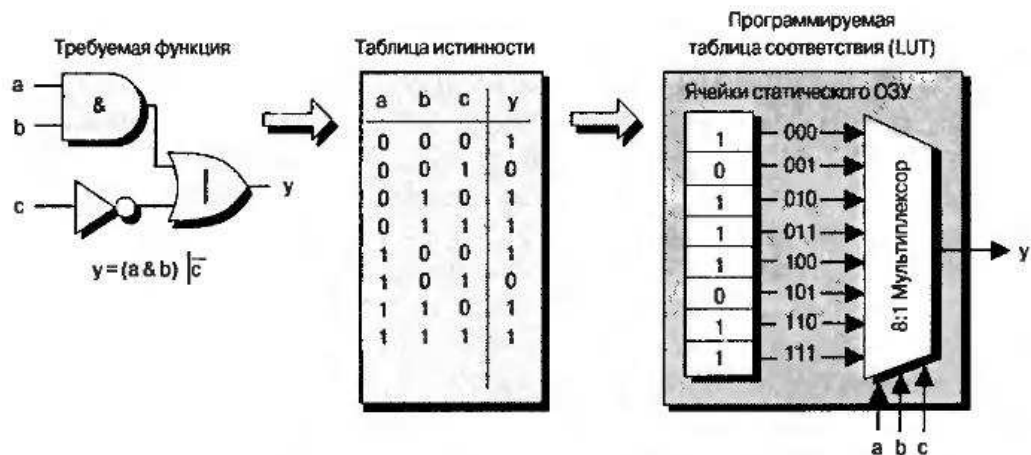


Рис 7. Соответствие между LUT и логической функцией.

Матрица соединений также может быть запрограммирована и позволяет соединять эти блоки в разные конфигурации. Программирование схемы достигается следующим образом. Вся схема ПЛИС «пронизана» ячейками конфигурационного ОЗУ. Каждая ячейка этого ОЗУ – это элемент, который может содержать либо '0' либо '1'. Далее рассмотрим соединение блоков. Блоки соединяются проводниками через ключи, к каждому из которых подключена одна ячейка конфигурационного ОЗУ. Если ячейка содержит 1, то ключ замкнут, и соединение есть, если 0, то ключ разомкнут, и соединения нет. Программирование логического блока на выполнение какой-либо функции производится также с помощью этого конфигурационного ОЗУ.

Программирование ПЛИС заключается в процессе записи соответствующих значений в конфигурационную память кристалла. Оно может быть осуществлено просто загрузкой значений с компьютера через соответствующий разъем на плате.

В итоге с использованием ПЛИС можно создавать практически любые цифровые схемы. Конечно, их параметры будут далеки от наилучших сравнительно со схемами, выполненными по технологии схем на стандартных элементах (будет не такая высокая максимальная тактовая частота, больше энергопотребление, часть схемы использоваться не будет, т.к. использовать ресурсы ПЛИС на 100% невозможно). Но при этом цена разработки схемы будет в тысячи раз меньше, и становится возможным разрабатывать свои схемы в учебных и научных целях.

Применения ПЛИС

Альтернатива «рассыпной логике»

До появления ПЛИС, когда перед конструкторами цифровых устройств вставала необходимость в создании некоторой цифровой схемы, у них было два варианта: использовать заказную микросхему ASIC, либо собирать эту схему из множества ИС малой интеграции, выполняющих простые функции – микросхем так называемой рассыпной логике. Разработка заказной микросхемы - очень дорогой процесс, и он может быть окуплен только при массовом серийном выпуске продукта. Собирать схему из рассыпной логике связано с большими трудностями по соединению этих элементов на интегральной схеме. Таким способом собрать большую схему в разумных (на современном уровне) габаритах невозможно, либо она будет занимать несколько шкафов и потреблять киловатты энергии. С появлением ПЛИС жизнь сильно упростилась, так как в одной микросхеме теперь можно было реализовать сложную схему, непосредственно по проекту разработчика.

Прототипирование микросхем

Другое применение ПЛИС заключается в том, что в одной или нескольких ПЛИС можно реализовать прототип будущего проекта заказной микросхемы и проверить его функциональность перед тем, как начинать долгий и дорогостоящий процесс ее изготовления.

Высокоскоростная обработка сигналов (радио, звук, видео)

В ПЛИС можно реализовать многие алгоритмы обработки потока данных (одномерных – звук, радио сигналы; двумерных – видео) на аппаратном уровне. Такая схема по сути будет являться специализированным вычислителем, способная обрабатывать данные только по одному алгоритму (в отличие от процессора общего назначения, на котором можно считать по любому алгоритму). Но за счет того, что она будет создана для реализации одного и только одного

алгоритма, она будет работать максимально эффективно. В итоге быстродействие алгоритма будет намного выше, чем, например, при реализации на сигнальном или универсальном процессоре.

В таком ключе ПЛИСы применяются при кодировании/декодировании сигналов радио сигналов сотовой связи или 4G WiMAX и LTE стандартов [4] – там где нужна высокоскоростная обработка в реальном времени. Также часто в реальном времени по фиксированным алгоритмам требуется обработка видео сигнала при кодировании/декодировании и при приеме/передаче. В системах видео наблюдения (отслеживание определенных событий при анализе экспериментальных данных в различных областях науки, или, например, в системах помощи водителю автомобиля) также могут использоваться ПЛИС. Множество статей о применении FPGA в области потоковой обработки сигналов можно найти на страницах онлайн журнала DSP-FPGA [5].

Высокопроизводительные реконфигурируемые вычисления

Эта тема весьма актуальна на данный момент. ПЛИСы могут использоваться как со-процессоры (ускорители) для универсальных процессоров, берущие на себя вычисления наиболее математически интенсивных участков алгоритма. Как было сказано в предыдущем параграфе, в ПЛИСе можно реализовать специализированный вычислитель со структурой, соответствующей выполняемому участку алгоритма. Тогда этот вычислитель будет наиболее эффективно использовать аппаратные ресурсы и решать задачу быстрее чем универсальный процессор. Дело в том, что процессор по сути последовательный вычислитель по программе, заложенной в памяти. Непосредственно полезными вычислениями он занимается довольно редко, а большую часть тактов выполняется другие задачи: выборка команды и данных из памяти, пересылка данных, ожидание готовности данных, обслуживание операционной системы.

ПЛИСы в этом отношении могут быть более эффективны. В них можно организовать параллельные потоки вычислений, каждый такт которых будет использован для выполнения какой-либо математической операции: сложение, умножением и т.д. Эффективная специализированная схема может обогнать процессор потенциально в 100 раз при реализации одного и того же логического алгоритма.

ПЛИСы используются как сопроцессоры во многих суперкомпьютерах:

- Maxwell <http://www.fhpca.org/index.html>
- Cray — XD1, XT5, <http://www.cray.com/>
- Применения ПЛИС для ускорения финансовых расчетов <http://www.computerworlduk.com/news/it-business/3290494/jp-morgan-supercomputer-offers-risk-analysis-in-near-real-time/>

<http://www.maxeler.com/content/frontpage/>

- Российско-Белорусский проект СКИФ <http://www.rsc-skif.ru/>
- НИИ «Квант» <http://www.rdi-kvant.ru/>

Вычисления на ПЛИС как ускорителях ведутся также и в Институте Прикладной Математики РАН <http://www.kiam.ru/MVS/contacts/>.

Отдельной областью применения FPGA является построение суперкомпьютера полностью на FPGA. Т.е. тут применяется другой подход, когда ПЛИС используется не как ускоритель некоторых частей алгоритма, исполняемого на CPU, а когда вся задача (весь ее алгоритм) реализован в поле (множестве) ПЛИС, соединенных друг с другом в 2D или 3D решетку. Примером такой работы является реконфигурируемый суперкомпьютер PBC-5, установленный в НИВЦ МГУ [6], который разработан и произведен в Южном Федеральном Университете г. Таганрог [7]. Ссылки на публикации по этой теме можно найти тут [8].

Вообще множество примеров применения ПЛИС вы найдете на страницах бесплатного журнала XCell, который выпускает Xilinx, его онлайн версию можно найти тут [9].

Классификация микросхем фирмы Xilinx

Фирма Xilinx является наиболее крупной компанией на рынке FPGA. Ей принадлежит 51% рынка. Ближайший ее конкурент с около 40-45% рынка – это компания Altera. Остальные компании (Actel, Lattice...) занимают нишевые позиции на рынке и делят остальные 5% рынка.

Продукция Xilinx и продукция Altera функционально очень схожи, и использование того или производителя обуславливается просто привычкой и более реже значимыми различиями в технологиях. Наш практический курс использует продукцию Xilinx, ее микросхемы мы и будем рассматривать.

Xilinx выпускает FPGA с различным количеством логических ресурсов. Количество логических ресурсов фирма Xilinx оценивает в так называемых Logic Cells – логических ячеек. Логическая ячейка в современных ПЛИС – это совокупность таблицы соответствия LUT и двух триггеров (такова архитектура современных ПЛИС). Т.е. одна LUT и два триггера – это как бы логическая ячейка. Я говорю как бы, потому что, чтобы получить количество Logic Cells, которое можно увидеть в документации Xilinx, нужно это число умножить на 1.6. Получается запутанно немного, но это нормально. В индустрии в этом отношении имеется путаница, и все это признают.

Просто надо запомнить правило, что количество Logic Cells = количество LUT x 1.6. Вооружившись этими знаниями можно смотреть таблицы.

Итак, старые семейства FPGA фирмы Xilinx – это Spartan, Spartan 2, Spartan 3, Virtex, Virtex-2, Virtex-4, Virtex-5. Семейство Spartan – это довольно дешевые маленькие ПЛИСы, Virtex – уже намного дороже и больше. На сегодняшний день широко используется 6 серия - это Spartan 6 и Virtex 6. Уже существует 7 серия, но она еще не очень распространена. На нашей отладочной плате Atlys стоит Spartan 6. Это семейство и рассмотрим.

Table 3: Spartan-6 FPGA Logic Resources

Device	Logic Cells	Total Slices	SLICEMs	SLICELs	SLICEXs	Number of 6-Input LUTs	Maximum Distributed RAM (Kb)	Shift Registers (Kb)	Number of Flip-Flops
XC6SLX4	3,840	600	300	0	300	2,400	75	38	4,800
XC6SLX9	9,152	1,430	360	355	715	5,720	90	45	11,440
XC6SLX16	14,579	2,278	544	595	1,139	9,112	136	68	18,224
XC6SLX25	24,051	3,758	916	963	1,879	15,032	229	115	30,064
XC6SLX45	43,661	6,822	1,602	1,809	3,411	27,288	401	200	54,576
XC6SLX75	74,637	11,662	2,768	3,063	5,831	46,648	692	346	93,296
XC6SLX100	101,261	15,822	3,904	4,007	7,911	63,288	976	488	126,576
XC6SLX150	147,443	23,038	5,420	6,099	11,519	92,152	1,355	678	184,304
XC6SLX25T	24,051	3,758	916	963	1,879	15,032	229	115	30,064
XC6SLX45T	43,661	6,822	1,602	1,809	3,411	27,288	401	200	54,576
XC6SLX75T	74,637	11,662	2,768	3,063	5,831	46,648	692	346	93,296
XC6SLX100T	101,261	15,822	3,904	4,007	7,911	63,288	976	488	126,576
XC6SLX150T	147,443	23,038	5,420	6,099	11,519	92,152	1,355	678	184,304

Кодовое название всех FPGA серии Spartan 6 – это XC6SLXYYY, где YYY – это примерное количество тысяч Logic Cells. Взять первую строчку таблицы XC6SLX4: это значит примерно 4 тысячи Logic Cells (точно 3840). Если это число поделить на 1.6, то получим 2400 – количество таблиц LUT. На каждую LUT приходится 2 триггера – 4800, что и видим из таблицы. Мы будем работать с кристаллом XC6SLX45, у него будет 27288 LUTs и 54576 триггеров. Максимальное размер кристалла семейства Spartan 6 – это 150 тысяч логических ячеек.

Контрольные вопросы

1. Принцип работы полевого транзистора, основы CMOS логики.
2. Классификация микросхем по типу соединения элементов.
3. Структура ПЛИС.

4. Применения ПЛИС

Источники

1. <http://en.wikipedia.org/wiki/CMOS>
2. [http://ru.wikipedia.org/wiki/Полевой транзистор](http://ru.wikipedia.org/wiki/Полевой_транзистор)
3. <http://www.easic.com/>
4. <http://www.eetimes.com/design/programmable-logic/4017742/Implementing-LTE-on-FPGAs>
5. <http://dsp-fpga.com/magazine/articles>
6. <http://www.srcc.msu.su/>
7. <http://mvs.sfedu.ru/>
8. <http://fpga.parallel.ru/publications.html>
9. <http://www.xilinx.com/publications/xcellonline/>