

ПРОЕКТИРОВАНИЕ ЦИФРОВЫХ СИСТЕМ НА СБИС ПРОГРАММИРУЕМОЙ ЛОГИКИ. I

(аналитический обзор)

*РУСТИНОВ В.А., ХАХАНОВ В.И.,
БАРАБАШ А.А., ГЕРАСИМОВ М.А.*

Данная публикация открывает цикл статей, посвященных технологиям проектирования цифровых систем на основе устройств программируемой логики. Авторы предполагают рассмотреть вопросы, касающиеся элементной базы, систем автоматизированного проектирования, языков описания аппаратуры, технологий проектирования, верификации, отладки и диагностики таких устройств.

1. Введение

Основой для написания данной статьи является постоянно возрастающий интерес разработчиков электронной аппаратуры к использованию сверх больших интегральных схем (СБИС) программируемой логики. К сожалению в Украине до сих пор не издана ни одна книга, посвященная данной тематике. Большую помощь в освоении технологий проектирования на СБИС программируемой логики оказывает сеть Internet, предоставляющая доступ к сайтам крупнейших производителей интегральных микросхем и профессиональных пакетов автоматизированного проектирования. Однако вся предоставляемая таким образом информация поступает лишь на английском языке. Следует также отметить, что большой объем информации, которая зачастую носит рекламный характер, затрудняет освоение данных технологий.

Проектирование цифровых устройств (ЦУ) на основе программируемой логики предоставляет целый ряд новых возможностей не только разработчикам, но и студентам соответствующих вузов, а также всем тем, кто так или иначе связан с разработкой современных цифровых систем обработки информации. Ведь главной отличительной особенностью программируемой логики является возможность их гибкой настройки на выполнение заданных функций самим пользователем. Микросхемы программируемой логики удачно сочетают в себе гибкость заказных БИС и доступность микросхем стандартной логики. Современные СБИС программируемой логики имеют очень низкую стоимость (стоимость простейших чипов составляет \$1-2), высокое быстродействие (профессиональные серии микросхем работают на частоте до 450 МГц), значительные функциональные возможности (один программируемый чип может заменить несколько сот корпусов традиционной "жесткой" логики), возможность многократного перепрограммирования, низкую потребляемую мощность и ряд других достоинств.

Процесс проектирования ЦУ на основе СБИС программируемой логики заключается в описании его

функционирования на входном языке используемого программного средства, выполнении автоматизированного синтеза, проведении моделирования, верификации и настройки архитектуры СБИС программируемой логики. При этом время разработки даже достаточно сложных проектов может составлять всего несколько часов.

2. Историческая справка

История развития программируемой логики началась с появления программируемых постоянных запоминающих устройств (ППЗУ - Programmable Read Only Memory - PROM) в начале 70-х годов. Первое время PROM использовались исключительно для хранения данных, позже их стали применять для реализации логических функций. Неудобство использования PROM в качестве логических преобразователей заключается в том, что логические функции перед записью в PROM необходимо приводить к совершенной дизъюнктивной нормальной форме, кроме того, емкость PROM не позволяла реализовать функции большого числа переменных.

Специально для реализации систем булевых функций большого числа переменных были разработаны программируемые логические матрицы (Programmable Logic Arrays - PLA). Именно PLA можно считать первыми программируемыми логическими устройствами (Programmable Logic Devices - PLD).

Совершенствование архитектуры PLA привело к появлению программируемых матриц логики (Programmable Array Logics - PAL), которые на долгие годы определили наиболее популярную архитектуру PLD. С момента своего появления PAL стали успешно конкурировать с PLA и благодаря ряду присущих им положительных свойств практически полностью вытеснили программируемые пользователем PLA.

Дальнейшее совершенствование технологии производства интегральных схем, повышение степени интеграции, успехи в создании корпусов с большим числом внешних выводов в начале 90-х годов привели к возможности реализации на одном кристалле нескольких PAL, объединяемых программируемыми соединениями. Подобные архитектуры получили название сложных PLD (Complex PLD - CPLD), соответственно все разработанные ранее PLD стали называть стандартными PLD (Standart PLD - SPLD) или классическими PLD (Classic PLD)[1].

Основу всех рассмотренных выше устройств составляют программируемые матрицы. Поэтому их еще называют программируемыми логическими устройствами, имеющими матричную структуру.

Параллельно с PLD также развивались архитектуры вентилярных матриц (Gate Array - GA), или матриц логических ячеек (Logic Cell Array - LCA), в русскоязычной литературе получившие название базовых матричных кристаллов. Первые вентилярные матрицы были полузаказными, т.е. программировались во время изготовления, что сдерживало их широкое практическое использование. Однако в 1985 г. фирма Xilinx выпустила программируемые пользователем вентилярные матрицы (Field Programmable Gate

Аrray - FPGA), которые стали успешно конкурировать с CPLD[3].

В настоящее время наблюдается бурное развитие архитектур CPLD и FPGA (каждый год появляются новые поколения этих устройств), их сближение, приобретение ряда общих свойств, снижение стоимости, повышение быстродействия и функциональной мощности (таблица).

Таблица 1
Тенденции развития устройств программируемой логики

Параметры	Год			
	1985	1990	1995	2000
Стоимость	-	>\$500	<\$50	<\$5
Число вентиляей	800	5.000	100.000	2.000.000
Число выводов	64	256	500	1000
Число транзисторов	85K	2M	6M	500M
Миним. задержка	40ns	15ns	3,5ns	1ns

Анализ тенденции развития архитектур программируемой логики позволяет предположить, что в ближайшее время основу элементной базы цифровых систем будут составлять CPLD и FPGA. Преимущество использования программируемой логики в цифровых системах становится особенно очевидным, когда необходимо быстро разработать опытный образец изделия, предполагаются частые корректировки проекта в процессе его разработки или реконфигурация структуры устройства при его функционировании [5].

3. Основные свойства устройств программируемой логики

Практически всем современным PLD присущи следующие основные свойства [5].

1. Низкая стоимость, которая вплотную приближается к стоимости стандартных ИС.
2. Высокое быстродействие. Применение передовых технологий привело к снижению задержки прохождения сигнала через PLD до 1 нс на один логический элемент, что позволяет достигать частоты функционирования устройств до 500 МГц.
3. Высокая степень интеграции. PLD имеют достаточно регулярную структуру, что позволяет применять 0,22-микронную технологию и достигать свыше 1 000 000 вентиляей на кристалл.
4. Функциональная мощность, благодаря которой PLD могут заменять сотни корпусов “жесткой” логики, а сложные PLD - тысячи корпусов, что позволяет реализовать на одном чипе всю или некоторую часть цифровой системы, например сопроцессор.
5. Универсальность. На одном и том же PLD можно строить различные функциональные узлы: комбинационные, регистровые, синхронные, асинхронные, арифметические и др.

6. Многократность программирования. В зависимости от технологии производства и типа настраиваемого элемента число перепрограммирований PLD может не ограничиваться или достигать нескольких тысяч.

7. Высокий процент выхода годных изделий. Благодаря передовым технологиям и наличию встроенных средств тестопригодного проектирования, PLD отличаются исключительной надежностью, что выражается в 100%-ном выходе годных изделий после программирования.

8. Защита информации от считывания. Все современные PLD имеют бит защиты, установка которого препятствует считыванию настройки PLD. Бит защиты сбрасывается только при перезаписи всей информации о настройке PLD.

9. Сброс регистров при включении питания. Это свойство позволяет однозначно определять начальное состояние скрытых элементов памяти при синтезе последовательностных устройств и отказаться от процедуры инициализации в начале работы устройства.

10. Предзагрузка регистров. В дополнение к свойству 9 PLD допускают предварительную загрузку внутренних регистров определенными значениями. Это позволяет осуществить процедуру инициализации последовательностных устройств путем перевода их в некоторое состояние. Однако на практике данное свойство чаще используется для тестирования PLD и проверки поведения устройства в запрещенных (illegal) состояниях.

11. Совместимость по входам и выходам с TTL-логикой. Большинство PLD, выполненных по технологиям, отличным от TTL, остаются совместимыми с TTL-логикой по коэффициентам расширения входов и выходов, уровням напряжений и др.

12. Разнообразие корпусов PLD. Одна и та же микросхема PLD может быть упакована в различные типы корпусов: штыревые - DIP, планарные - SOIC, PGA, PQFP, TQFP, J-lead и др.

13. Возможность реконфигурирования внутренней структуры PLD в процессе функционирования для достижения высокой скорости решения сложных задач обработки данных [6].

4. Классификация PLD

Наибольшей оперативностью и гибкостью использования характеризуются ИС, программируемые пользователем. С их помощью разработчики могут значительно сократить цикл производства и улучшить качество изделий, благодаря проверке функциональных возможностей, ещё на этапе проектирования, когда внесение изменений в системы легко и относительно дешево. Эти преимущества и заставили разработчиков специализированных микросхем (ASIC) обратиться к использованию PLD. Конечно, при данном подходе имеются некоторые недостатки. К ним следует отнести прежде всего не всегда рациональное использование площади кристалла, отсутствие эффективных методов проектирования

сложных устройств. Кроме того, характеристики аналогичных устройств, выполненных на базе ASIC, обычно более предпочтительны. Однако эти недостатки не столь ощутимы по сравнению с выигрышем, получаемым за счет сокращения времени разработки цифровых устройств на базе PLD. Поэтому, несмотря на указанные недостатки, в настоящее время программируемые пользователем ИС считаются наиболее перспективной элементной базой. Это подтверждается многочисленными статистическими исследованиями и неизменным увеличением объема их производства.

В свою очередь программируемые пользователем ИС можно разделить на микропрограммные и ИС программируемой логики в соответствии с двумя основными подходами к проектированию ЦУ: микропрограммным и аппаратным. Первый подход предполагает построение ЦУ на базе некоторого универсального элемента (микропроцессора, микрокомпьютера, микроконтроллера и др.), который специализируется загружаемой в оперативное запоминающее устройство (ОЗУ) или зашиваемой в перепрограммируемое постоянное запоминающее устройство (ППЗУ) программой (микропрограммой). Недостатком такого подхода является невысокая скорость работы устройства, однако последнее может быть легко перенастроено на другой алгоритм работы путем замены программы в ОЗУ или в ППЗУ.

Характерной особенностью PLD является возможность их настройки на заданный алгоритм функционирования путем изменения своей внутренней структуры. Проектирование ЦУ на основе PLD заключается в определении настройки по заданным спецификациям и программировании микросхемы на специальном оборудовании, называемом программатором (programmer). Построенные таким образом ЦУ отличаются прежде всего высокой скоростью работы, низкая стоимость и малые сроки проектирования. Первые промышленные PLD характеризовались однократностью настройки, которая осуществлялась путем пережигания плавких перемычек (fuse-link). Однако с появлением перепрограммируемых PLD с электрическим и ультрафиолетовым стиранием этот недостаток устраняется и по способу своего использования они ничем не уступают микропрограммным ИС.

В настоящее время, согласно архитектурным особенностям, PLD принято делить на три больших класса [5]: стандартные PLD (SPLD), сложные PLD (CPLD) и программируемые пользователем вентильные матрицы (FPGA), что продемонстрировано на рис. 1.

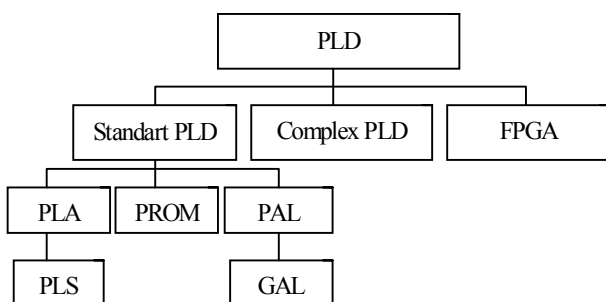


Рис. 1. Классификация PLD по структурной организации

Все PLD можно также разделить по типу настраиваемого элемента [3] (рис.2):

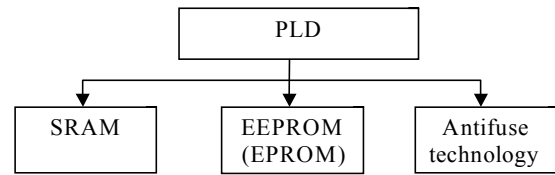


Рис.2. Классификация программируемой логики по типу настраиваемого элемента

- 1) PLD, программируемые с помощью статического ОЗУ (SRAM);
- 2) PLD, программируемые с помощью электрически стираемого постоянного запоминающего устройства (ПЗУ, EEPROM), либо ПЗУ с ультрафиолетовым стиранием (EPROM);
- 3) PLD, программируемые с помощью так называемой Antifuse-технологии, интересной тем, что при программировании между коммутационными слоями образуются проводящие каналы.

Число настроек элементов первого класса обычно не ограничивается. Для второго класса число перезаписей данных настройки в настоящее время достигает 10000. Представители устройств третьего класса программируются однократно, что можно отнести к их недостаткам. Достоинство таких PLD заключается в отсутствии какого-либо вспомогательного оборудования на стадии эксплуатации готового устройства.

По количеству перепрограммирований PLD можно также разделить на многократно и однократно программируемые (рис.3).

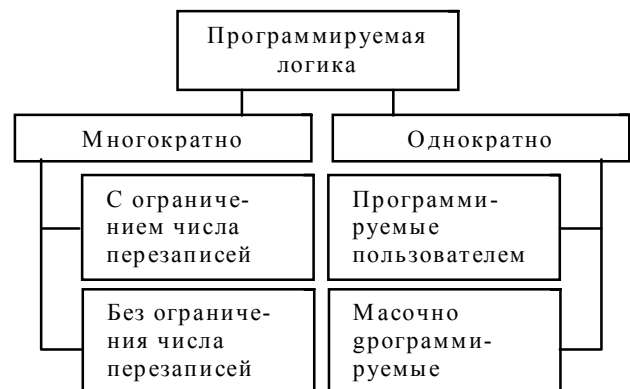


Рис.3. Классификация программируемой логики по способу программирования

Большинство PLD (как однократно, так и многократно настраиваемых) программируется самим пользователем во время эксплуатации (field). Однако если проект тщательно отлажен и изделие производится массовыми тиражами, в нем могут применяться масочно программируемые PLD, настройка которых выполняется при их изготовлении.

Еще одним важным критерием классификации PLD является предсказуемость задержки прохождения сигнала со входа на выход устройства [5] (рис.4).

Для всех SPLD задержка прохождения сигнала с любого входа на любой выход всегда постоянна. Поэтому при проектировании на этих устройствах



Рис.4. Классификация программируемой логики по времени прохождения сигнала (задержки) с любого входа на любой выход

можно не выполнять временное моделирование сигналов. Для представителей класса CPLD задержка переменная, но легко вычисляемая. В FPGA задержка полностью зависит от пути, по которому проходит сигнал со входа на выход. При изменении трассировки межсоединений изменяются и задержки сигналов.

В данной статье были рассмотрены только основные способы классификации программируемой логики. Этот процесс можно было бы продолжить дальше, разделяя PLD на классы в зависимости от тех или иных свойств. Но такая классификация не даст положительных результатов, поскольку свойств PLD достаточно много, и часто микросхемы в пределах одного поколения и даже одной серии имеют различные свойства. Поэтому ограничимся рассмотрением наиболее общих свойств, которые присущи различным PLD.

5. Основы архитектуры устройств программируемой логики

PLD - многоцелевой чип для реализации цифровых схем. Он содержит определенную совокупность логических элементов, которые могут быть соединены между собой различными способами. В общем случае, PLD может быть представлено в виде "черного ящика", который содержит логические элементы и программируемые ключи [4]. Последние позволяют программировать соединения логических элементов друг с другом и с элементами, находящимися в других микросхемах, для реализации необходимых логических схем.

5.1. PLA-программируемые логические матрицы

Первым представителем большого класса программируемых логических устройств можно считать программируемые логические матрицы. Их общая структура изображена на рис.5.

Основная идея заключается в реализации логических функций, представленных в ДНФ.

Для этого PLA содержит совокупность элементов И, выходы которых подаются на входы набора элементов ИЛИ. Как показано на рис.5, входы PLA $x_1, \dots, x_n$ заведены через набор буферов (Input buffers and inverters), которые обеспечивают подачу как прямых, так и инверсных входных сигналов на блок схемы, называемый матрицей И (AND plane). На выходе матрицы И формируются термы $P_1, \dots, P_k$, каждый из которых может быть конфигурирован для осуществления конъюнкции входных переменных $x_1, \dots, x_n$. Полученные термы служат входными сигналами для

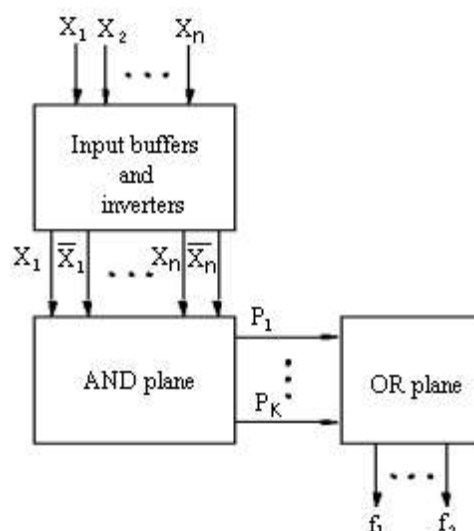


Рис. 5. Общая структура PLA

матрицы ИЛИ (OR plane), на выходе которой формируются значения функций $f_1, \dots, f_m$. Каждая такая функция реализует любую дизъюнкцию термов $P_1, \dots, P_k$. Таким образом, с помощью PLA можно реализовывать логические функции, записанные в ДНФ.

Внутренняя структура небольшой PLA приведена на рис.6.

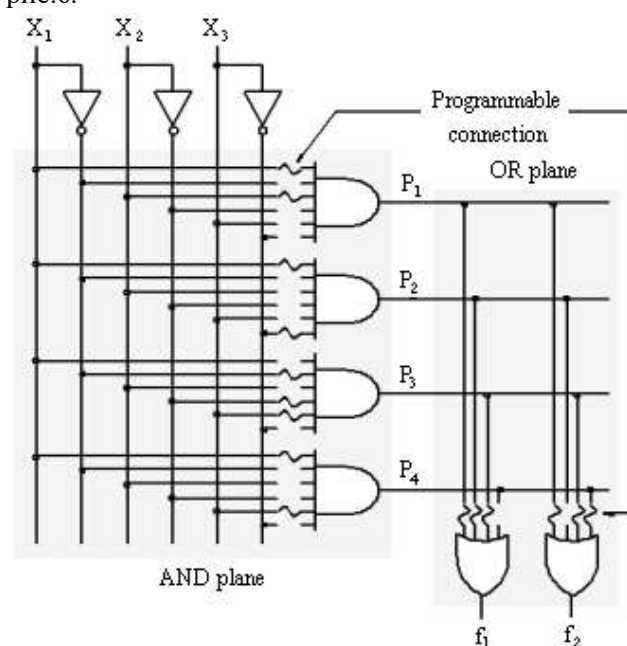


Рис. 6. Структура PLA

PLA имеет три входа, четыре внутренних терма и два выхода. Каждый конъюнктор матрицы И имеет шесть входов, соответствующих прямым и инверсным значениям трех входных сигналов. Каждое соединение к матрице И программируемо (programmable connection). Любые не подключенные к матрице И входы не влияют на значения выходного терма соответствующего конъюнктора.

На рис.6. матрица И формирует терм P_1 , как конъюнкцию входных переменных x_1 и x_2 . Следовательно, $P_1 = x_1 x_2$. Аналогично $P_2 = x_1 \bar{x}_3$, $P_3 = \bar{x}_1 \bar{x}_2 x_3$, и $P_4 = x_1 x_3$. Программируемые соединения существуют

также и для матрицы ИЛИ. Выход f_1 связан с термами P_1 , P_2 и P_3 . Поэтому на первом выходе получим реализацию функции $f_1 = x_1x_2 + x_1\bar{x}_3 + \bar{x}_1\bar{x}_2x_3$. Аналогично $f_2 = x_1x_2 + \bar{x}_1\bar{x}_2x_3 + x_1x_3$.

Таким образом, на рис.6. изображена PLA, запрограммированная для реализации описанных выше функций. Однако матрицы И и ИЛИ можно запрограммировать и иначе для реализации других функций входных сигналов x_1 , x_2 и x_3 . Единственное ограничение, накладываемое на такие функции, это разрядность матрицы И (в данном примере на выходе матрицы И можно получить только 4 терма). Коммерческие PLA имеют гораздо более высокую разрядность. Типичные PLA включают 16 входов, 32 промежуточные шины (термы) и 8 выходов.

Отметим, что при рассмотрении структуры большинства программируемых кристаллов используют другой стиль представления схемы. Примером данного стиля изображения рассматриваемой выше схемы является рис.7.

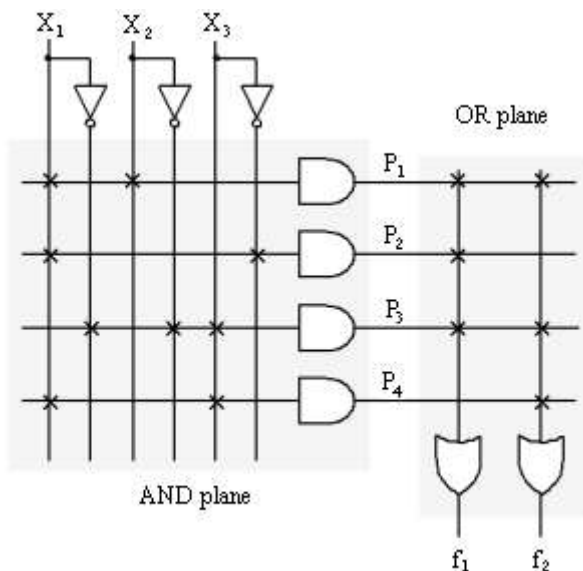


Рис.7. Общепринятое представление структуры схем PLA

На нем показаны программируемые соединения при реализации функций f_1 и f_2 , изображенных на рис. 6.

PLA, программируемые однократно, применяются для замены микросхем стандартной логики. PLA с соединениями, программируемыми многократно, часто включаются в состав СБИС типа микропроцессоров. В этом случае значения логических функций могут быть получены микропроцессором гораздо быстрее, чем при программной реализации данных функций.

5.2 PAL - программируемые матрицы логики

В PLA обе матрицы И и ИЛИ программируемы. PLD, в которых матрица И является программируемой, а матрица ИЛИ фиксированной, получили название программируемых матриц логики (PAL). Поскольку изготовить такие устройства проще, чем PLA, они имеют меньшую стоимость и улучшенные

характеристики, что способствовало их высокой популярности.

Пример PAL с тремя входами, четырьмя термами и двумя выходами представлен на рис.8.

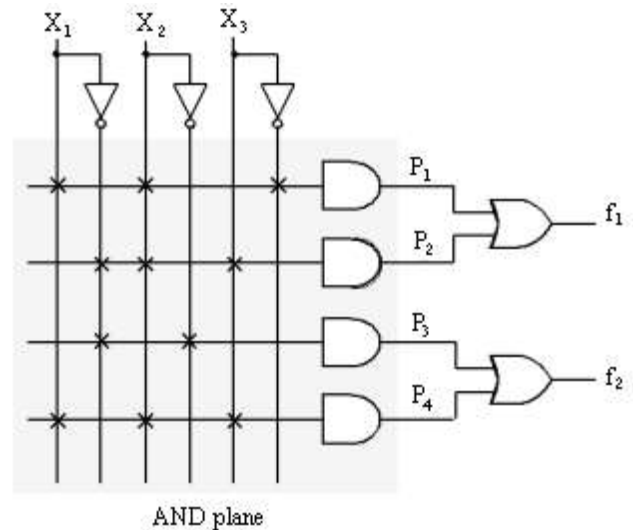


Рис. 8. Пример типичной структуры PAL

Термы P_1 и P_2 аппаратно соединены с первым элементом ИЛИ, P_3 и P_4 - со вторым элементом. Эти соединения не могут программироваться. PAL реализует две логические функции $f_1 = x_1x_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3$ и

$f_2 = \bar{x}_1\bar{x}_2 + x_1x_2x_3$. По сравнению с PLA (рис. 7), PAL обладает меньшей гибкостью программирования. PLA позволяет использовать четыре терма в качестве входов для элементов ИЛИ, а элементы ИЛИ в PAL имеют только два фиксированных входа. Для компенсации этого недостатка производители PAL производят чипы разного размера, с различным числом входов и выходов, а также входов к вентилям ИЛИ.

Выходы всех ранее рассмотренных нами PLD соединялись непосредственно с внешними выводами микросхемы. В современных чипах эти выходы поступают на внешние выводы через так называемые макроячейки (МЯ), в качестве которых могут использоваться триггеры различных типов, тристабильные буферы, различные логические элементы и ряд других. Пример подобной МЯ приведен на рис.9 [4].

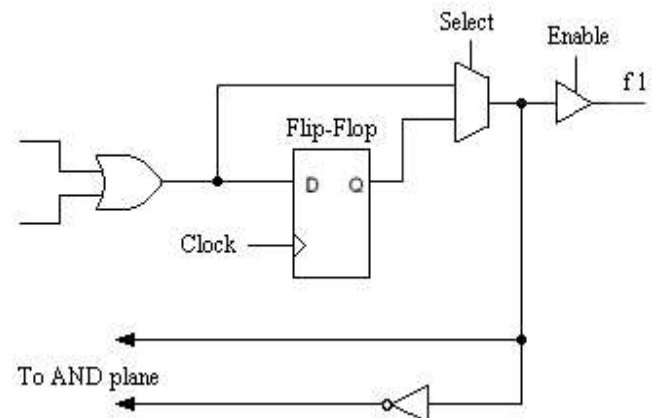


Рис. 9. Пример макроячейки микросхемы PAL

В ней используется D-триггер, мультиплексор 2@1 и тристабильный буфер. D-триггер служит в качестве элемента памяти и позволяет сохранить значение функции, полученной на выходе PAL. Триггер стробируется сигналом Clock, который может быть как внешним, так и внутренним. Мультиплексор позволяет по сигналу Select либо подавать на вход буфера сигнал прямо с выхода элемента ИЛИ, что характерно при построении схем комбинационной логики, либо использовать значение, хранимое D-триггером. Такая ситуация характерна при построении различных регистровых схем и автоматов с памятью. Кроме того, сигнал с выхода мультиплексора может быть подан обратно на матрицу И, что позволяет реализовывать с помощью PAL более сложные логические функции. Помимо этого, выходы микросхемы могут быть переведены в высокоимпедансное состояние по сигналу Enable, который управляет выходными буферами.

Многие компании объединяют PLA, PAL и некоторые другие сходные с ними структуры в отдельный класс Simple PLD.

5.3 CPLD — комплексные программируемые логические устройства

При реализации сложных проектов обычно применяются так называемые комплексные программируемые логические устройства (CPLD).

CPLD состоит из блоков, расположенных на одном кристалле. Каждый блок соединен с блоками ввода-вывода, осуществляющими высокоскоростной обмен с другими микросхемами. Кроме того, все блоки связаны между собой внутренними параллельными шинами. Отдельно взятый блок подобен микросхеме либо PLA, либо PAL (PAL-like block). Пример CPLD приведен на рис. 10.

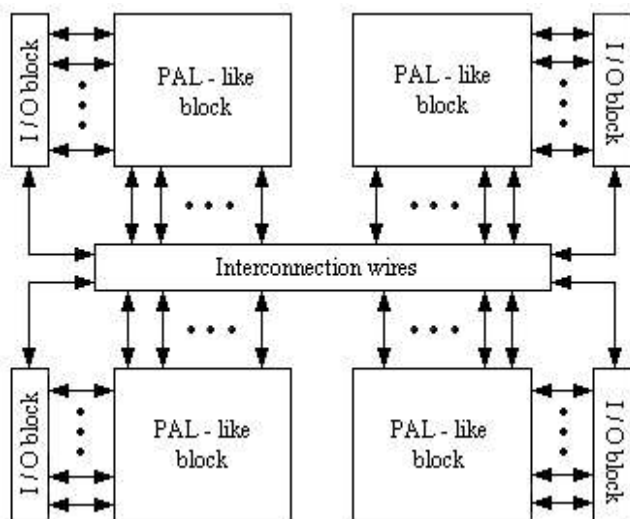


Рис. 10. Структура микросхемы CPLD

Чип включает четыре таких блока, которые связаны между собой внутренними шинами и также соединяются с блоками ввода-вывода.

На рис. 11. приведен пример внутренней структуры PAL-like блоков и их соединения в CPLD.

Каждый блок включает 3 МЯ. В состав МЯ входит 4-входовой элемент ИЛИ. Выход этого элемента подается на вход логического элемента ИСКЛЮЧАЮЩЕЕ ИЛИ. Другой вход вентиль ИКЛЮЧАЮЩЕЕ ИЛИ может программироваться на подачу нуля или единицы. Если вход запрограммирован на подачу логической единицы, то данный вентиль будет инвертировать значение, снимаемое с выхода дизъюнктора. Если вход настроен на подачу 0, то вентиль будет передавать значение выходного сигнала без изменения. МЯ также включает триггер, мультиплексор и буфер с тремя состояниями. Буфер позволяет настраивать выводы чипа как на ввод, так и на вывод сигналов. Чтобы вывод микросхемы мог стать входным, достаточно перевести соответствующий буфер в высокоимпедансное состояние. Дальнейший путь прохождения сигнала может быть запрограммирован. На схеме это указывается символом X, стоящим на пересечении вертикальных и горизонтальных линий. Необходимо заметить, что если вывод запрограммирован на прием информации, то он, естественно, не может быть использован как выходной, и соответственно вся МЯ, стоящая перед ним окажется незадействованной. Обычно фирмы-изготовители указывают максимально возможное число входных и выходных линий. Оптимальное распределение между ними ложится на плечи пакетов проектирования или на самого разработчика.

Некоторые CPLD включают в свой состав дополнительные выводы, которые позволяют не тратить понапрасну целые МЯ.

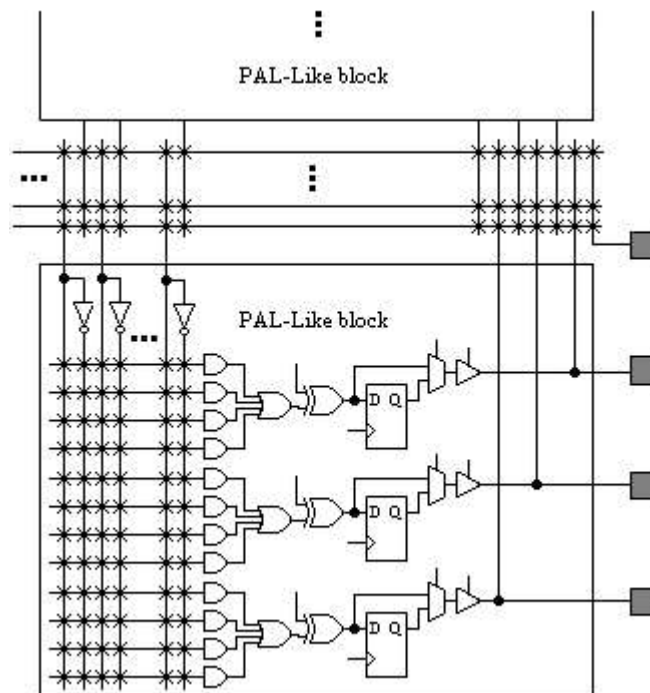


Рис. 11. Внутренняя организация блока микросхемы CPLD

5.4. FPGA - программируемые пользователем вентильные матрицы

Как уже говорилось ранее, микросхемы типа CPLD применяются в проектах с большим количеством вентиляей. Но по современным меркам схема даже с 20000 вентилями не является большой [2]. Для реализации крупных проектов удобно использовать чипы, обладающие еще большей логической емкостью. Такими чипами являются программируемые пользователем вентильные матрицы. По своей структуре FPGA отличаются от SPLD и CPLD, потому что не содержат матриц И и ИЛИ [4]. Вместо этого FPGA включают в себя специальные логические блоки для реализации требуемых функций. Общая структура FPGA изображена на рис.12.

Типичный чип FPGA содержит три типа элементов: логические блоки, блоки ввода-вывода для соединения с внешними выводами чипа и программируемые ключи для создания соединений между блоками. Логические блоки размещаются в узлах решетки вертикальных и горизонтальных проводников, с помощью которых можно создавать самые различные соединения между блоками. Соединения являются программируемыми и осуществляются путем настройки внутренних соединительных ключей (interconnection switches).

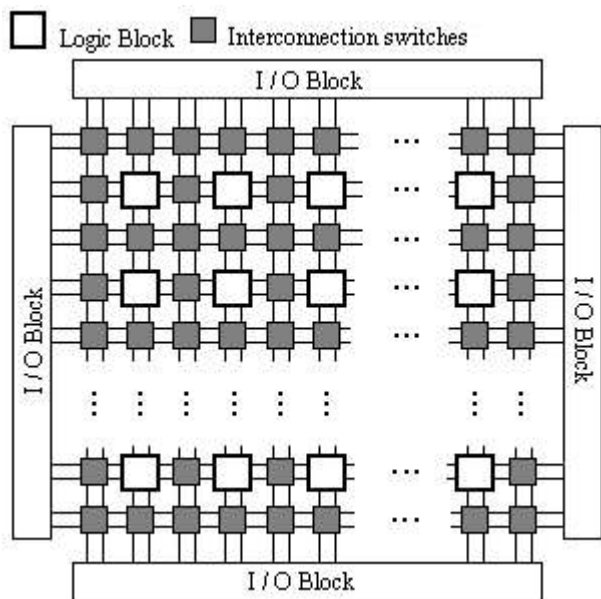


Рис.12. Общая структура чипа FPGA

Соединения с блоками ввода-вывода также программируются. У различных серий FPGA реальное число программируемых ключей значительно отличается, что необходимо учитывать при выборе фирмы-изготовителя и конкретного семейства микросхем.

Каждый логический блок в FPGA обычно имеет небольшое число входов и один выход. Типичный логический блок является функциональным генератором, называемым LUT (Look Up Table). Каждый LUT содержит запоминающие ячейки, которые используются при реализации небольшой логической функции. Единичная ячейка способна хранить значение одной логической переменной, т.е. 0 или 1. Размер LUT определяется числом входов, которое зависит от типа микросхемы.

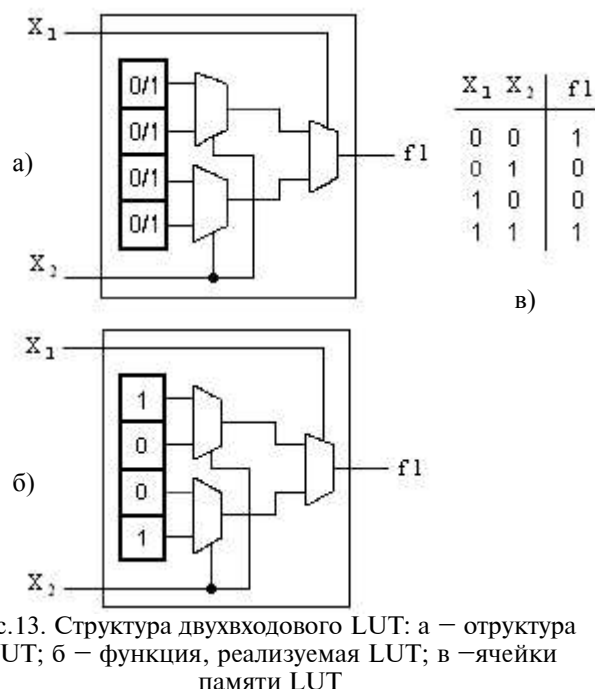


Рис.13. Структура двухвходового LUT: а – отруктура LUT; б – функция, реализуемая LUT; в –ячейки памяти LUT

LUT, изображенный на рис.13, способен реализовать любую логическую функцию двух переменных. Поскольку таблица истинности функций двух переменных имеет четыре строки, этот LUT имеет четыре запоминающих ячейки. Одна ячейка соответствует значению выхода в каждой строке таблицы истинности. Входные переменные x_1 и x_2 используются как адресные входы трех мультиплексоров, которые в зависимости от значений x_1 и x_2 передают содержание одной из этих четырех запоминающих ячеек на выход LUT.

Как видно из рис.13, для программирования LUT необходимо лишь сохранить таблицу истинности реализуемой функции в ячейках памяти блока. Тогда при $x_1 = x_2 = 0$ на выход LUT будет подано значение самой верхней запоминающей ячейки, которая хранит значение функции для набора $x_1 x_2 = 00$. Точно так же для всех наборов x_1 и x_2 значение, сохраненное в запоминающей ячейке, соответствующей положению в таблице истинности, подается на выход LUT.

На рис.14 показан трехвходовой LUT.

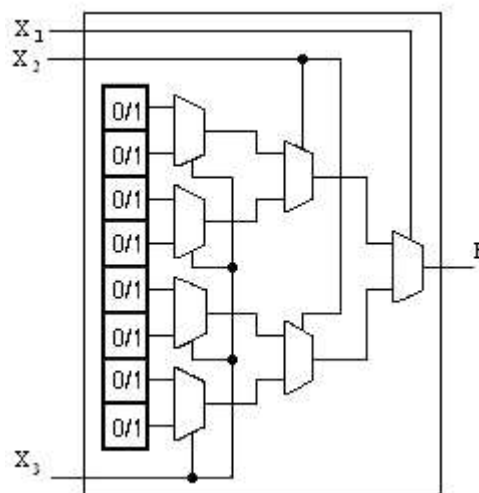


Рис14. Трехвходовой LUT

Он имеет восемь запоминающих ячеек и может реализовать любую функцию трех переменных. LUT современных чипов FPGA обычно имеет или четыре или пять входов, которые требуют 16 и 32 запоминающие ячейки соответственно. Как и в случае с PAL-структурой, помимо схем реализации булевых функций каждый LUT дополняется элементами, которыми оснащена МЯ PAL. На рис.15 показано подключение дополнительного D-триггера, который может хранить значение функции, реализуемой LUT.

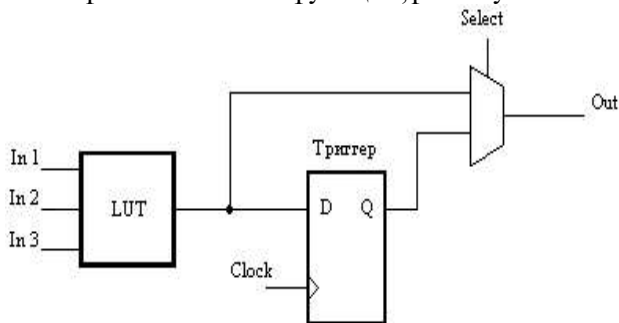


Рис.15. Подключение триггера в логическом блоке FPGA

Триггер управляется сигналом Clock, который может быть как внутренним, так и внешним. Мультиплексор позволяет выбирать задействие D-триггера при построении регистровых схем или отключение его от схемы при проектировании комбинационной логики.

FPGA можно использовать тогда, когда логические блоки запрограммированы для реализации необходимых функций, а линии соединений – для реализации требуемых взаимосвязей между логическими блоками и блоками ввода-вывода. Запоминающие ячейки LUT в FPGA энергозависимы. Это означает, что они теряют свое содержимое всякий раз при отключении питания. Следовательно, чип FPGA должен программироваться каждый раз при включении питания. Обычно данные, необходимые для программирования, хранятся в микросхеме ППЗУ, расположенной на той же плате, что и чип FPGA. Данные, хранимые в ППЗУ, автоматически переписываются в ячейки памяти FPGA при включении питания.

Фрагмент структуры чипа FPGA, который запрограммирован для реализации необходимой логической функции, изображен на рис.16.

FPGA имеет блоки LUT с двумя входами и четыре программируемых линии соединений. Каждый активный ключ соединяет вертикальную и горизонтальную соединительные линии и показан символом •. Первые два блока LUT запрограммированы на реализацию функций $f_1 = x_1 x_2$ и $f_2 = \bar{x}_2 x_3$. Третий LUT реализует функцию $f = f_1 + f_2$.

6. Заключение

В первой части цикла статей представлены основные положения технологии проектирования цифровых схем на основе устройств программируемой логики. При этом проведен обзор, касающийся вопросов классификации устройств программируемой логики, их основных свойств и внутренней организации. Отмечено, что самыми современными классами подобных устройств являются CPLD и FPGA. Уже сейчас они составляют конкуренцию технологии проектирования заказных СБИС, благодаря ряду достоинств:

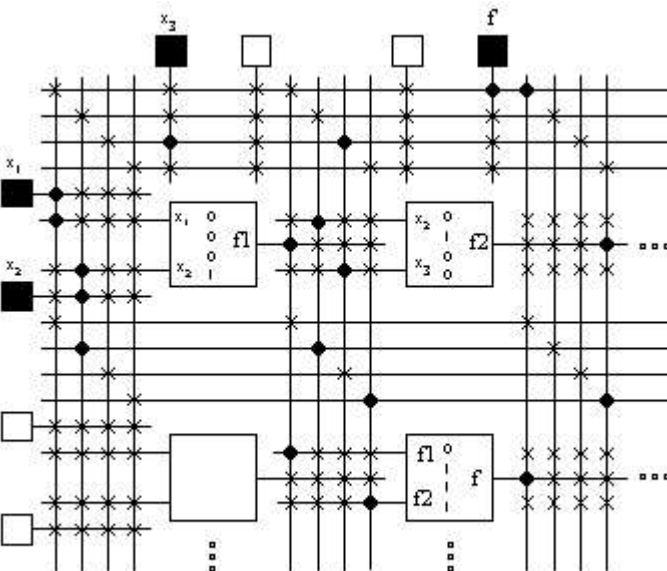


Рис.16. Пример структуры запрограммированного чипа FPGA

- высокой скорости проектирования;
- низкой стоимости разрабатываемых устройств;
- высокому быстродействию.

Отметим, что применение CPLD особенно эффективно при реализации управляющих и интерфейсных схем, а FPGA – при реализации алгоритмов цифровой обработки сигналов. Вопросы выбора конкретного класса PLD при реализации цифровых проектов, а также обзор современных семейств PLD и систем автоматизированного проектирования на их основе будут рассмотрены в следующих частях цикла.

Литература: 1. Data Book. Altera, 1999. 2. The Programmable Logic Data Book. Xilinx, 1999. 3. Programmable Logic Data Book and Design Guide. Actel, 1999. 4. Brown S., Zvonko Vranesic. Fundamentals of Digital Logic with VHDL Design. The McGraw Hill, 2000. 530 p. 5. Соловьев В.В., Васильев А.Г. Программируемые логические интегральные схемы и их применение. Мн.: “Беларуская навука”, 1998. 266 с. 6. Стещенко В. Школа разработки аппаратуры цифровой обработки сигналов на ПЛИС // Chip News.- 1999. N8. С. 2-6.

Поступила в редколлегию 18.03.2000

Рецензент: д-р техн. наук Кривуля Г.Ф.

Рустинов Владимир Алексеевич, канд. техн. наук, зам. директора ХАЭР. Научные интересы: автоматизация проектирования и диагностика вычислительных систем. Увлечения: теннис, горные лыжи, плавание. Адрес: Украина, 61057, Харьков, пер. Театральный, 11/13, тел. 43-11-83.

Хаханов Владимир Иванович, д-р. техн. наук, профессор кафедры автоматизации проектирования вычислительной техники ХТУРЭ. Научные интересы: техническая диагностика вычислительных устройств. Увлечения: баскетбол, горные лыжи, футбол. Адрес: Украина, 310726, Харьков, пр. Ленина, 14, тел. 40-93-26.

Барабаш Александр Алексеевич, студент кафедры ЭВМ ХТУРЭ. Научные интересы: разработка цифровых устройств на базе новых методов проектирования. Увлечения: шахматы. Адрес: Украина, 61057, Харьков, пер. Театральный, 11/13, тел. 43-11-83.

Герасимов Максим Александрович, студент кафедры ЭВМ ХТУРЭ. Научные интересы: разработка цифровых устройств на базе микросхем программируемой логики. Увлечения: программирование. Адрес: Украина, 61057, Харьков, пер. Театральный, 11/13, тел. 43-11-83.