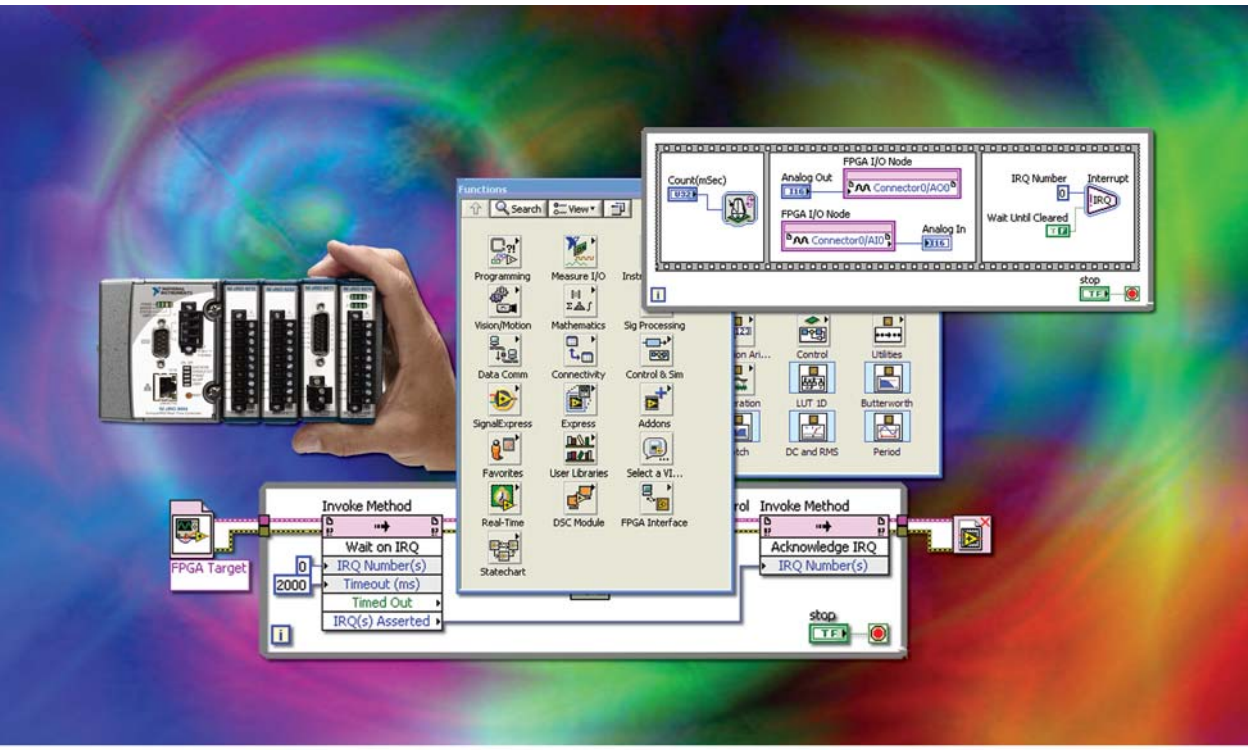
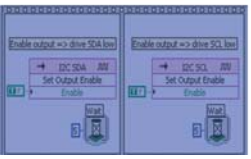
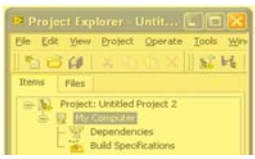
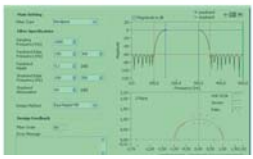
LabVIEW FPGA

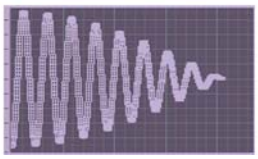
РЕКОНФИГУРИРУЕМЫЕ ИЗМЕРИТЕЛЬНЫЕ И УПРАВЛЯЮЩИЕ СИСТЕМЫ











Баран Е. Д.

ДМК ПРЕСС
издательство

УДК 621.38
ББК 32.973.26-108.2
Б24

Баран Е. Д.
Б24 LabVIEW FPGA. Реконфигурируемые измерительные и управляющие системы. – М.: ДМК Пресс, 2009. – 448 с.

ISBN 978-5-94074-494-8

В книге представлено описание нового модуля графической среды проектирования LabVIEW. С помощью этого модуля, расширяющего концепцию виртуальных инструментов в область разработки аппаратных средств, можно создавать собственные каналы ввода-вывода и устройства обработки данных, функциональность и характеристики которых определяются не на заводе изготовителе, а инженером-разработчиком прикладных систем автоматизации экспериментальных исследований, испытаний и управления. Рассмотрены архитектурные особенности реконфигурируемых систем, изложен порядок и основные приемы их проектирования, приведены описания и характеристики технических компонентов, а также некоторые примеры практической реализации технологии реконфигурированного ввода-вывода.

Издание предназначено для специалистов в области разработки информационно-измерительных и управляющих систем, может использоваться в процессе обучения студентов соответствующих специальностей.

УДК 621.38
ББК 32.973.26-108.2

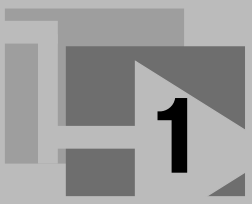
ISBN 978-5-94074-494-8

СОДЕРЖАНИЕ

▼ 10

▼ 11

Программируемые логические интегральные схемы



По мере развития микроэлектронных технологий и возрастания степени интеграции элементов на кристалле все очевиднее становилось противоречие между потребностями создавать компактные специализированные устройства из наименьшего количества компонентов и нерентабельностью расширения номенклатуры выпускаемых массовым тиражом типовых логических элементов и блоков. Действительно, уже с конца 60-х гг. прошлого века стало возможным на одном кристалле размещать десятки и сотни логических вентилей, но чем насыщеннее становился каждый новый кристалл, тем больше возрастала его специализация и тем меньше становилась относительная доля объема его выпуска.

Появление универсальных логических устройств, микропроцессоров, конечная функция которых определялась загружаемой в них программой, казалось, сблизило возможности изготовителей микросхем, владеющих высокотехнологичными средствами производства, с одной стороны, потребности и возможности разработчиков прикладных устройств и систем – с другой. Однако инженеры попрежнему вынуждены были для решения своих задач использовать компоненты высокой (по меркам того времени) степени интеграции – в качестве основного функционального блока (например, процессора) и десятки логических элементов малой степени интеграции – для того, чтобы состыковать между собой микросхемы разных классов и типов, реализовать на аппаратном уровне специфические для каждого конкретного случая узлы.

Например, контроллер выполнялся на основе 4- или 8-разрядного микропроцессора и нескольких микросхем памяти, содержащих тысячи логических вентилей, и до десятка и более интегральных схем низкой степени интеграции – дешифраторов, регистров, простейших логических элементов, с помощью которых реализовывалась системная шина, каналы ввода-вывода и т. п. При этом для изготовления такого контроллера требовалась печатная плата с площадью, достаточной для установки нескольких микросхем повышенной степени интеграции (микропроцессор, память) и десятка, а то и более, микросхем малой степени интеграции, размещения всех необходимых печатных проводников. Сборка относительно несложных логических узлов контроллера проводилась с помощью паяльника, в то время как существенно более сложные микропроцессор и блоки памяти

поступали «в собранном виде», готовые. Очевидно, что и стоимость монтажа оказывалась непропорциональной сложности узлов, а надежность устройства в целом определялась надежностью элементов малой степени интеграции, количеством их выводов и соединительных проводников на печатной плате.

Таким образом, разработчики цифровой аппаратуры, получив возможность использовать достаточно мощные и универсальные процессорные компоненты, вынуждены были по-прежнему применять простые и тоже универсальные элементарные логические схемы, собирая их в специализированные блоки, без которых невозможно было спроектировать и изготовить устройство, решающее специальную задачу измерений и обработки информации, управления или тестирования.

Назревала необходимость усовершенствования технологии проектирования и производства конечных изделий, приближения ее к технологии проектирования и производства микроэлектронных компонентов. Безусловно, передовой являлось массовое производство микросхем, где связи между элементами в кристалле создавались с помощью масок. Однако использовать эту технологию в многочисленных лабораториях разработчиков цифровых устройств невозможно, так как изготовление масок – весьма трудоемкий и дорогой процесс. Необходимо было придумать более доступный механизм произвольного соединения элементов кристалла между собой и с внешними выводами микросхемы, а разместить на кристалле необходимый набор типовых логических элементов (десятки или сотни) к тому времени проблемой уже не являлось.

Другими словами, разработчику надо было предоставить возможность более простыми и в то же время более эффективными средствами **создавать собственную микросхему**, которая могла бы заменить горсть универсальных микросхем малой степени интеграции, объединяемых в устройстве в специализированный узел. Такая возможность была реализована в программируемых логических интегральных структурах – ПЛИС (PLD – Programmable Logic Device). На кристалле ПЛИС размещались простые логические вентили различных типов и объединяющие их регулярные шины проводников. Подобная микросхема представляла собой некоторую универсальную заготовку, в которой разработчик доступными средствами может удалить ненужные связи между вентилями, изменяя тем самым логическую функцию ПЛИС и, по существу, создавая уникальное, необходимое только ему логическое устройство, сохраняя почти все качества интегрального исполнения.

1.1. Простые программируемые логические устройства – SPLD

Первые кристаллы, структуру и функции которых мог определять пользователь, – программируемые постоянные запоминающие устройства ППЗУ (PROM – Programmable Read Only Memory). ППЗУ состоит из двух матриц логических элементов – матрицы элементов «И» и матрицы элементов «ИЛИ». Входы элементов «И» соединяются с внешними входами микросхемы через повторитель, или ин-

вертор, а входы элементов «ИЛИ» – с выходами элементов «И» [1]. Таким образом, на выходах элементов «ИЛИ», являющихся внешними выходами микросхемы, формируется логическая сумма произведений входных переменных.

В показанной на рис. 1-1 схеме переменные, подаваемые на входы адреса, в дешифраторах строк образуют все возможные конъюнкции – функции «И» от всех входных переменных. Объединением выходов конъюнкторов в столбцах матрицы элементов памяти реализуются логические функции «ИЛИ». Именно отключением определенных строк от столбцов определяются конечные логические функции ППЗУ.

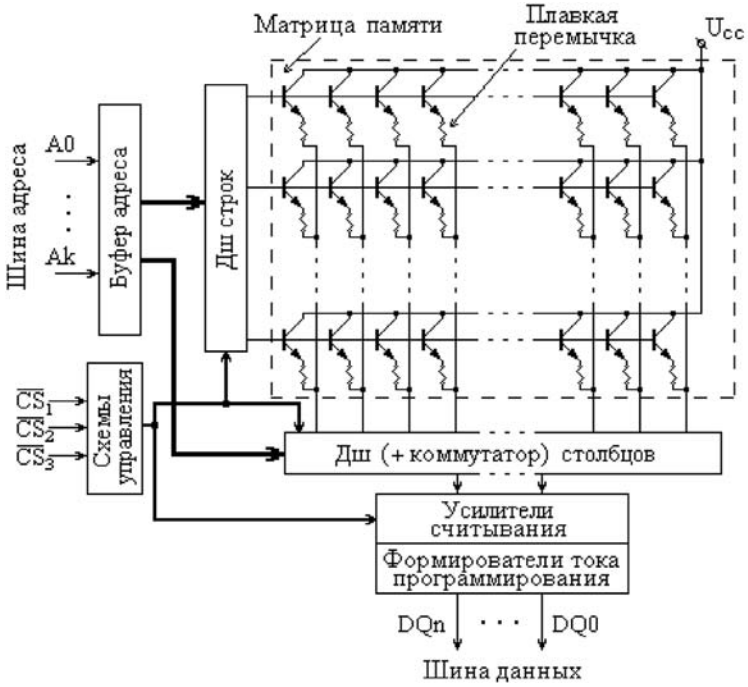


Рис. 1-1. Структура ППЗУ

От изготовителя ППЗУ поступало «чистым» – подключены все входы всех элементов «И» и «ИЛИ», однако пользователь мог удалять ненужные соединения, адресно разрушая их в специальном режиме. В качестве разрушаемых соединений (перемычек) использовались выжигаемые током проводники или пробиваемые p-n-переходы (диоды), сохраняемые же связи и являлись собственно запоминающими элементами и определяли логическую функцию для каждого выхода ППЗУ. Следует отметить, что использовались и иные способы программирования ППЗУ – не удалением ненужных из предварительно созданных межсоединений, а наоборот – созданием нужных соединений. Однако подобные технологические нюансы для разработчика прикладных устройств непринципиальны.

Различают ППЗУ двух основных типов:

- в микросхемах, получивших название PROM, изменять можно только подключения входов элементов «ИЛИ», матрица связей элементов «И» фиксирована (пример на рис. 1-1);
- в микросхемах типа PAL/GAL (Programmable/Generic Array Logic) – программируются связи элементов «И», а соединения элементов «ИЛИ» изменению не подлежат.

Набор из N произвольных логических функций от k входных переменных может быть реализован в одной микросхеме ППЗУ с N выходами емкостью 2^k вентилей, достаточно записать эти функции в совершенной дизъюнктивной нормальной форме и ввести их таблицы истинности в программатор.

Очевидно, что в программируемых ПЗУ площадь кристалла расходуется неэкономно – для всех возможных конъюнкций и дизъюнкций входных и промежуточных переменных должны быть зарезервированы входы элементов «И» и «ИЛИ», значительная часть которых при программировании отключается.

Более рационально логические функции реализуются на базе ПЛМ – программируемых логических матриц (PLA – Programmable Logic Array) [2, 3]. В ПЛМ могут быть запрограммированы обе матрицы логических элементов – «И» и «ИЛИ», при этом количество входов конъюнкторов и дизъюнкторов может быть уменьшено без существенных потерь сложности реализуемых логических функций (рис. 1-2).

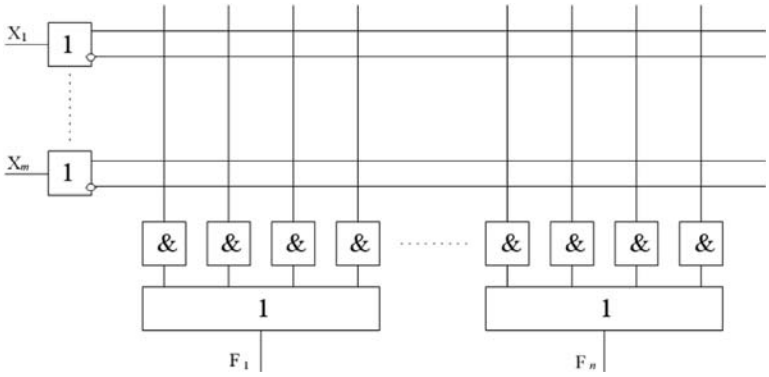


Рис. 1-2. Структура ПЛМ

ПЛИС, рассмотренные выше, – PROM, PAL, GAL, PLA – предоставили разработчикам определенную свободу в реализации проектов, обеспечили возможность уменьшения количества компонентов, повышения надежности изделий и снижения их стоимости. Стало возможным создавать вычислительные устройства не только на основе микропроцессоров с жесткой системой команд, но и с использованием микропроцессорных секций с микропрограммным управлением, причем набор команд уже определялся самим разработчиком, который запи-

сывал микропрограммы в ПЛИС. Таким же образом проектировались специализированные комбинационные устройства – преобразователи кодов, шифраторы и дешифраторы, а применение ПЛИС совместно с элементами памяти – триггерами, регистрами, счетчиками – позволяло разрабатывать быстродействующие устройства микропрограммного управления различными объектами.

Дальнейшее развитие технологии ПЛИС позволило включить в них и элементы памяти, так что разработчик смог перейти на качественно новый уровень, создавая не простые комбинационные устройства, а законченный последовательностный автомат, полностью реализующий требуемую диаграмму состояний-переходов без использования дополнительных элементов малой и средней степени интеграции. Достаточно наглядное представление о возможностях, предоставляемых ПЛИС с памятью, дает приведенная на рис. 1-3 схема популярной микросхемы PAL22V10, клоны которой выпускали и выпускают многие компании [4, 5].

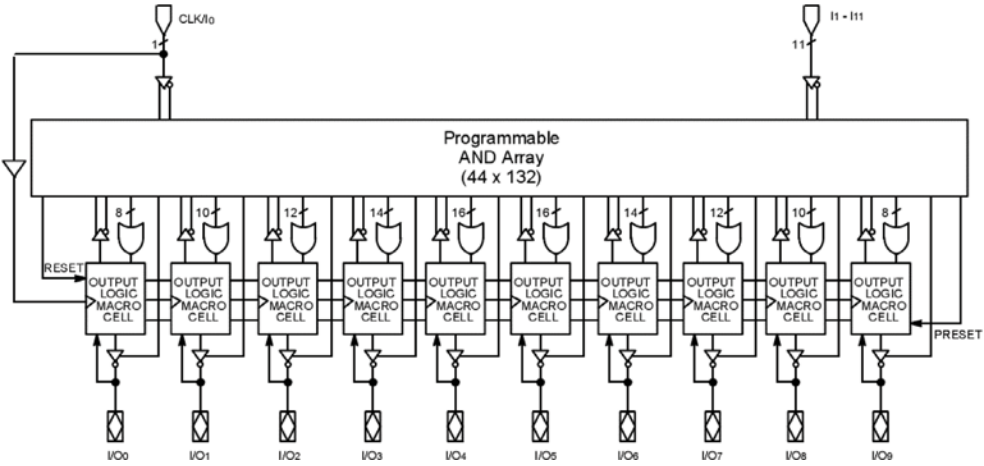
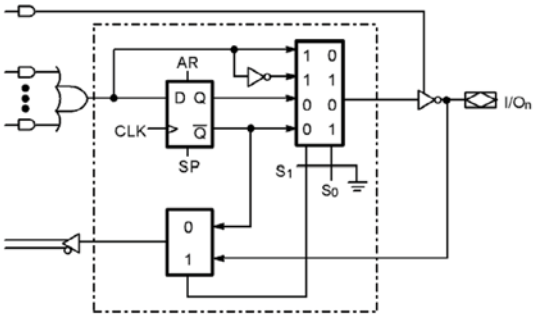


Рис. 1-3. Структура ПЛИС с памятью

Эта микросхема, относящаяся к классу PAL, содержит программируемую матрицу элементов «И» (Programmable AND Array), на входы которых I_1–I_{11} поданы 11 переменных (или их инверсии) с внешних входов ПЛИС и 10 внутренних переменных (или их инверсии), ассоциируемых с внешними входами/выходами ПЛИС. Выходы элементов «И» объединяются в линейке из 10 непрограммируемых элементов «ИЛИ» с фиксированным количеством входов – от 8 до 16. Сигналами с выходов элементов «ИЛИ» управляются выходные макроячейки (OUTPUT LOGIC MACRO CELL), каждая из которых содержит тактируемый D-триггер с входами сброса и установки.

Каждая из 10 макроячеек может быть сконфигурирована для работы в одном из 4 режимов, выбираемых с помощью программируемых перемычек S0 и S1 (рис. 1-4).



S0	S1	Выходной сигнал
0	0	Триггер, инверсный
0	1	Триггер, прямой
1	0	Комбинационный, инверсный
1	1	Комбинационный, прямой

Рис. 1-4. Макроячейка PAL22V10

Этими перемычками основной мультиплексор ячейки настраивается таким образом, что на выходной контакт ПЛИС поступает сигнал, реализуемый комбинационной логикой (матрицей «И» и «ИЛИ»), или тот же сигнал, зафиксированный D-триггером по приходу синхроимпульса CLK. Выходной сигнал (I/O_n) с помощью дополнительного мультиплексора может быть возвращен во входную комбинационную схему в качестве сигнала обратной связи для текущей макроячейки или в качестве дополнительной логической переменной – для остальных макроячеек.

Любой выход может быть переведен в высокоимпедансное состояние и использован как вход или двунаправленный вход/выход ПЛИС. Перечень полезных свойств этой микросхемы расширяют возможности сброса элементов памяти в исходное состояние по включению питания, загрузки в регистр произвольного кода через выходные контакты (что улучшает тестопригодность создаваемых на основе этой ПЛИС устройств), а также возможность установки защиты от несанкционированного копирования внутренней структуры ПЛИС – интеллектуальной собственности (Intellectual Property) разработчика.

Микросхемы типа PAL с элементами памяти вместе с ПЛИС комбинационного типа PROM, PAL, GAL в последующем были отнесены к классу простых программируемых логических устройств – SPLD (Simple Programmable Logic Device).

Отличительные признаки SPLD:

- каждая макроячейка имеет свой внешний выход и свой собственный триггер;
- в микросхеме SPLD реализовано не менее двух макроячеек;
- обычно все макроячейки выполнены одинаковыми;
- логическая функция макроячейки описывается одним логическим термом;
- логическая функция макроячейки реализуется матрицами элементов «И» и «ИЛИ».

К достоинствам SPLD обычно относят простоту проектирования специализированных устройств, постоянное и, как правило, одинаковое время прохождения сигналов со входов на выходы, возможность замены одной или несколькими микросхемами SPLD достаточно большого количества типовых микросхем малой и средней степени интеграции.

Основные недостатки SPLD – неэффективное использование ресурсов (логических вентилей) и, как следствие, проблематичность создания на их основе сложных цифровых устройств.

1.2. Технологии программирования ПЛИС

Последовательное увеличение степени интеграции микросхем и функциональной насыщенности реализуемых в кристалле ячеек, которые мог сконфигурировать разработчик, сопровождалось совершенствованием технологии программирования ПЛИС. И прежде чем продолжить обзор архитектур ПЛИС, целесообразно обсудить параллельно развивающиеся технологии программирования.

В первых семействах SPLD (PROM, PAL/GAL, PLA), выпускаемых на основе биполярных полупроводниковых элементов, однажды реализованная пользователем функция не могла быть изменена, структура соединений логических вентилей в ПЛИС записывалась однократно [5, 6]. Из-за этого если в процессе отладки вновь разработанного устройства обнаруживались ошибки, то для их исправления необходимо было извлечь из печатной платы ПЛИС, «прошитую» с ошибками, и заменить на исправленную. То есть по-прежнему разработчик был вынужден использовать паяльник (хоть и в меньшей степени, чем раньше) или, если это допускалось с точки зрения надежности изделия, монтировать на печатной плате гнезда (сокеты), позволяющие оперативно заменять микросхемы ПЛИС.

Совершенно новое качество было достигнуто после перехода на кристаллы с униполярными полупроводниковыми компонентами, в которых вместо однократно разрушаемых (создаваемых) связей-перемычек применялись запоминающие элементы на МОП-транзисторах с изолированным затвором. Заряд, создаваемый на затворе МОП-транзистора при программировании таких структур, сохранялся годами и не исчезал при отключении питания. Заряд этот можно снять, восстановив тем самым исходное состояние ячейки памяти, а при необходимости вновь записать – то есть появилась возможность **неоднократного** перепрограммирования ПЛИС. Технологии производства таких изделий разработаны в 1971 г. фирмой Intel, а в 1974 г. – фирмой Toshiba и реализованы в репрограммируемых ПЗУ (РППЗУ). Запись информации в РППЗУ производится в специальном режиме импульсами напряжения повышенной амплитуды, а стирание осуществляется двумя способами:

- ультрафиолетовым излучением – такие микросхемы назывались UV-EPROM (Ultraviolet Erasable PROM), УФ РППЗУ;
- электрическим напряжением противоположной полярности – микросхемы EEPROM (Electrically Erasable ROM), в отечественной литературе их обычно называют ЭСППЗУ – электрически стираемые программируемые ПЗУ.

Для стирания информации ультрафиолетовым излучением в корпусах микросхем РППЗУ создается закрытое кварцевым стеклом окошко, через которое облучается кристалл.

Теперь инженеру не нужно было иметь запас однократно программируемых устройств, чтобы заменить неправильно спроектированную микросхему на ис-

правленную. А вновь разработанное прикладное изделие можно было запускать в производство, комплектуя его набором универсальных типовых элементов и достаточно дешевыми микросхемами «собственного изготовления» – специфическими только для данного изделия и запрограммированными самим разработчиком логическими устройствами.

Однако по-прежнему необходимо было предусматривать возможность извлечения микросхемы РППЗУ из печатной платы для стирания и последующего перепрограммирования, а самое главное – надо было оснастить рабочее место специальным программатором и устройством для стирания ультрафиолетовым излучением.

Следующий шаг в развитии программируемых пользователем компонентов был сделан, когда технология позволила реализовать стирание/перезапись информации внутри кристалла, без использования внешнего специального оборудования. Подобные микросхемы – Electronically Erasable PROM – разработаны фирмой Intel в 1979 г. и, по существу, исключили из процесса разработки, изготовления и отладки макетных образцов аппаратуры демонтаж исправных, но неправильно запрограммированных инженером компонентов.

Наилучшими качествами в семействе микросхем класса Electronically Erasable PROM обладает энергонезависимая память типа Flash, предложенная в 1984 г. фирмой Toshiba (с 1988 г. развиваемая фирмой Intel и рядом других фирм), которая широко используется сейчас в твердотельных накопителях информации большой емкости. Flash-память характеризуется большим количеством циклов перезаписи (до 10^6 и более), высоким быстродействием, некоторые разновидности допускают возможность стирания/модификации содержимого произвольной ячейки. Благодаря этим качествам, высокой технологичности в производстве, а следовательно, и более низкой стоимости Flash-память практически вытеснила остальные разновидности программируемой энергонезависимой памяти из набора комплектующих, используемых при разработке цифровых устройств и систем.

И еще о некоторых задачах, которые приходилось решать в процессе развития микроэлектроники и системотехники.

С увеличением степени интеграции ПЛИС, как, собственно, и любых других интегральных схем, обострялась и становилась все более важной проблема тестирования компонентов и систем, создаваемых на их основе. Являясь составной частью общей проблемы проектирования сложных цифровых автоматов, проблема их отладки и верификации оказалась принципиальной при массовом использовании ПЛИС в новых изделиях и массовом производстве систем, содержащих микросхемы высокой степени интеграции. Тестирование подобных компонентов и устройств требовало значительных ресурсов времени как на подготовку тестов, так и на их прогон, специализированного дорогостоящего оборудования и программного обеспечения, сопоставимого по сложности и трудоемкости разработки с прикладным (целевым) программным обеспечением.

Радикальным и эффективным решением проблемы тестирования стала взятая на вооружение разработчиками концепция проектирования тестопригодных или

легко тестируемых компонентов и систем (Design For Testability – DFT). Из нескольких опробованных на практике подходов к созданию тестопригодных устройств наиболее экономным и универсальным был признан подход, основанный на методе тестирования, получившем название метода пограничного сканирования (Boundary-Scan). Суть метода заключается в том, что в режиме тестирования обеспечиваются возможности:

- изоляции любого компонента от всех остальных компонентов системы;
- доступа к любому изолированному компоненту для его автономного тестирования;
- выполнения отдельной процедуры проверки межсоединений компонентов.

При этом в основном режиме работы и компоненты, и система в целом попрежнему функционируют по прямому назначению.

Концепция пограничного сканирования в 1990 г. была закреплена международным стандартом IEEE-1149.1 – IEEE Standard Test Access Port and Boundary-Scan Architecture, определяющим архитектуру устройств с пограничным сканированием, а также структуру и функционирование специального тестового порта (Test Access Port – TAP-порт) [7]. Этот порт часто называют JTAG-интерфейсом по наименованию объединенной рабочей группы по тестированию Joint Test Action Group. Стандарт стал руководством к действию подавляющего большинства производителей электронных компонентов, разработчиков и изготовителей цифровых систем.

Теперь в каждую микросхему, выполненную в соответствии со стандартом IEEE-1149.1, введены TAP-порт и совокупность JTAG-ячеек, по одной на каждый из выводов микросхемы, которые образуют регистр пограничного сканирования. JTAG-ячейка позволяет:

- подключить выводы ядра микросхемы к внешним выводам для основного режима работы микросхемы в составе устройства;
- подключить к внешним выводам микросхемы регистр пограничного сканирования и отключить от них ядро микросхемы для тестирования соединений между микросхемами с помощью внешнего тестера;
- отключить внешние выводы микросхемы от ядра и подключить к ядру регистр пограничного сканирования для тестирования ядра.

Таким образом, регистр пограничного сканирования может работать в нескольких режимах. В основном режиме работы микросхемы регистр «прозрачен» для внешних сигналов, а в тестовом – ядро микросхемы изолировано от внешней среды. При этом на входы ядра тестовые сигналы поступают с выходов регистра пограничного сканирования, а выходные сигналы ядра могут быть записаны в этот же регистр. В тестовом режиме загрузка и считывание данных в регистр осуществляются последовательно через TAP-порт.

TAP-порт содержит четыре обязательных и одну дополнительную линию для передачи сигналов:

- **TDI** – *Test Data Input* – вход последовательного потока данных. В зависимости от состояния интерфейса в этот порт передаются либо команды для управления JTAG-ячейками, либо тестовые данные;

- **TDO** – *Test Data Output* – выход последовательного потока данных. Через эту линию в тестер передаются результаты исполнения команд и состояния JTAG-ячеек;
- **TCK** – *Test Port Clock* – последовательность тактовых импульсов для синхронизации последовательного обмена. Этот сигнал вырабатывается тестером и необязательно должен иметь фиксированную частоту. Максимальная частота следования импульсов TCK для многих микросхем находится в пределах 5–10 МГц. Стандарт не ограничивает частоту тактирования снизу;
- **TMS** – *Test Mode Select* – этот сигнал управляет режимом JTAG интерфейса;
- **TRST*** – *Test Port Reset* – сброс узлов тестовой логики (сигнал необязателен). Этот сигнал не заменяет сигнал сброса ядра микросхемы, поскольку JTAG-логика управляет только JTAG-ячейками и не влияет на работу ядра.

Регистры пограничного сканирования всех микросхем могут объединяться в один общий последовательный регистр, чтобы обеспечить доступ к любой микросхеме устройства. Для этого в каждой микросхеме предусмотрен регистр обхода (Bypass) из одного бита. Кроме того, TAP-порт содержит 4-битовый регистр команд для управления режимами работы.

Стандарт IEEE-1149.1 устанавливает только 4 обязательные команды, остальные 12 возможных кодов команд зарезервированы для дальнейших расширений, и их действие определяется изготовителем микросхем.

В целом реализация метода пограничного сканирования предельно проста как с точки зрения необходимых аппаратных ресурсов, так и с точки зрения программирования. В каждой микросхеме может потребоваться не более 4–5 дополнительных выводов для TAP-порта и от нескольких десятков до нескольких тысяч дополнительных логических элементов и триггеров, что обычно составляет не более единиц процентов от общего количества логических вентилей на кристалле.

Исключительно важно, что тестирование методом пограничного сканирования не требует сложного и дорогого внешнего тестера [8] – его функции может выполнять обычный персональный компьютер с простейшим адаптером, позволяющим использовать типовой последовательный или параллельный порт компьютера в качестве TAP-порта (рис. 1-5).

Благодаря стандарту IEEE-1149.1 качественное тестирование стало возможным не только в крупных компаниях, которые смогли выложить миллионы долларов за современный тестер, но и в любой лаборатории, где тестером становится персональный компьютер со свободным портом принтера.

Одновременно существенно упрощается и разработка программ тестирования вследствие регулярности структуры ПЛИС и доступности большого количества внутренних узлов синтезируемых в ПЛИС автоматов. С целью облегчения проектирования тестов и унификации тестового обеспечения созданы специальные языки программирования высокого уровня – **Boundary Scan Description Language** (BSDL) – язык описания устройств с пограничным сканированием, **Jam** и др.

Для нас несомненный интерес представляет то обстоятельство, что JTAG-интерфейс и созданные для работы с TAP-портом языки могут быть использованы не только для тестирования ПЛИС, но и для их конфигурирования [9]. Собрав схему, подобную изображенной на рис. 1-5, разработчик в считанные минуты мо-

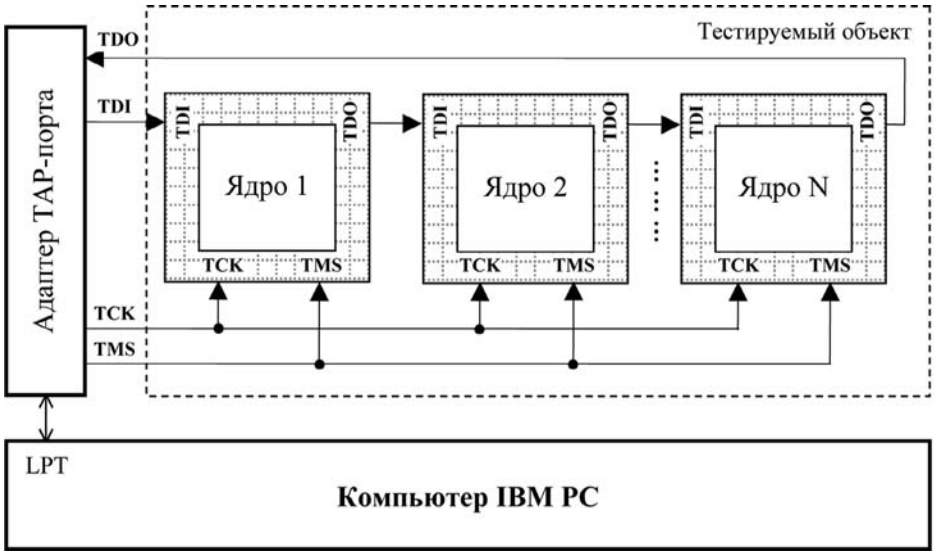


Рис. 1-5. Тестирование методом пограничного сканирования

жет «загрузить» в чистые ПЛИС только что спроектированное устройство, протестировать его, а в дальнейшем, при необходимости, оперативно изменять структуру ПЛИС, находясь в одной и той же среде проектирования. Аналогично, если на печатной плате вместе с ПЛИС смонтированы другие микросхемы, поддерживающие стандарт IEEE-1459.1, например микропроцессоры или запоминающие устройства, то и в них можно записать программу, воспользовавшись TAP-портом, протестировать отдельные компоненты или систему в целом.

Таким образом, концепция пограничного сканирования дала в руки инженеру универсальное средство реконфигурирования аппаратных компонентов, программирования микропроцессорных и иных устройств, а также тестирования. При использовании этих средств стала доступной **технология программирования в системе** (In-Circuit Programming) – без паяльника и гнезд под микросхемы, благодаря которой не только упростился процесс проектирования и производства, но упростилось и необходимое специальное оборудование – отпала необходимость в отдельных устройствах – программаторах, устройствах стирания информации в ПЛИС, сложном и дорогостоящем тестовом оборудовании.

1.3. Сложные программируемые логические устройства – CPLD

Вернемся к обзору архитектур программируемых логических схем. Вслед за простыми – SPLD – появился класс сложных программируемых логических устройств – Complex Programmable Logic Device (CPLD). Одной из первых по-

добные изделия выпустила на рынок компания Altera под названием Classic Programmable Logic Device (аббревиатура та же). В качестве примера на рис. 1-6 показана структурная схема простейшей микросхемы из семейства CPLD EP610 [10, 11].

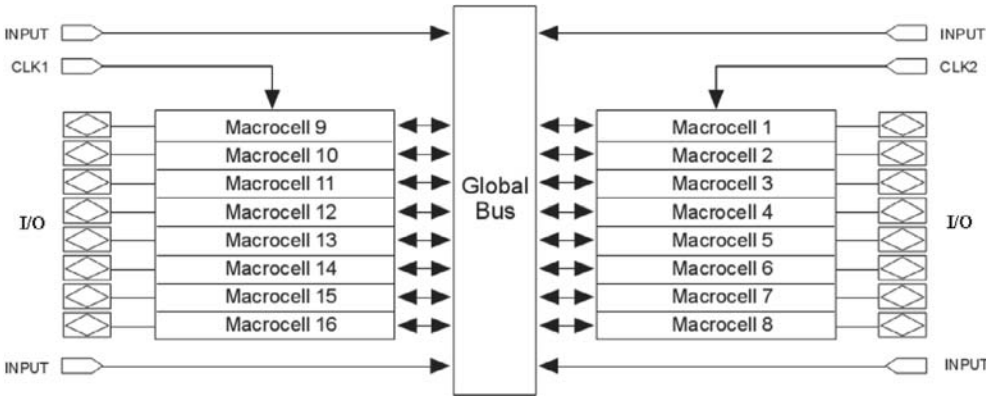


Рис. 1-6. Структурная схема Classic PLD Altera EP610

Чем же отличаются сложные ПЛИС от простых?

В SPLD логические макроячейки, содержащие триггер, соединялись преимущественно с выходными контактами ПЛИС, причем количество макроячеек было невелико. Сложные ПЛИС при бóльшем количестве макроячеек – в простейшей CPLD микросхеме типа EP610 их уже 16, обеспечивают возможность практически произвольного соединения макроячеек (Macrocell) друг с другом и с внешними выводами микросхемы (I/O) и позволяют создавать существенно более сложные последовательностные устройства. Для организации связей внутри ПЛИС используется программируемая матрица (шина) глобальных проводников (Global Bus). В ПЛИС EP610, кроме 16 двунаправленных выводов (I/O), имеются 4 дополнительных входа (Input) и 2 глобальных входа синхронизации (Clk1 и Clk2).

Возможности реализации логических функций в CPLD, в отличие от SPLD, определяются не только связями между макроячейками через разветвленную общую шину коммутации Global Bus, но и связями, создаваемыми внутри каждой макроячейки (рис. 1-7). Элемент памяти макроячейки (Programmable Register) может быть сконфигурирован как D-, T-, SR- или JK-триггер с индивидуальным управлением установки в «0» или «1».

Комбинаторная логическая функция макроячейки образуется программированием линейки элементов «И» на первом уровне при фиксированных соединениях с 8-входовым элементом «ИЛИ» на втором уровне. Возможно программирование цепей синхронизации, входов/выходов, использование сигналов обратной связи и т. п.

В этом первом семействе Classic PLD пока еще отсутствует TAP-порт и внутрисхемное программирование невозможно.

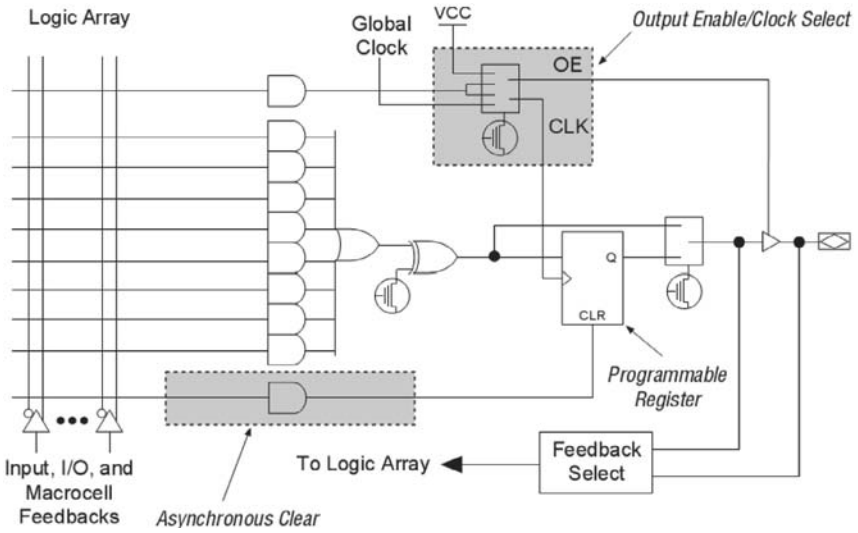


Рис. 1-7. Макроячейка EP610

Однако уже в следующих семействах – MAX 3000 (5000, 7000 и 9000) компании Altera, «настоящих» сложных программируемых устройствах (Complex PLD) – концепция пограничного сканирования поддерживается, и тем самым обеспечивается достижение всех связанных с этой концепцией преимуществ – простоты тестирования и программирования структуры ПЛИС.

Общее представление об особенностях CPLD этих семейств дает рис. 1-8, на котором представлена архитектура самой простой микросхемы MAX 3000 [11, 12].

Основным крупным компонентом MAX 3000 является блок логических матриц – Logical Array Blocks (LAB), каждый из которых содержит по 16 макроячеек (Macrocell), подобных рассмотренным выше. Макроячейки и логические блоки соединяются между собой, с внешними программируемыми контактами ввода-вывода (I/O), а также с внешними общими линиями управления и синхронизации (Input/GClkX, Input/OEX, Input/GClrn) с помощью глобальной программируемой матрицы межсоединений – Programmable interconnect array (PIA).

Эта матрица (рис. 1-9) представляет собой набор переключателей, управляемых ячейкой памяти с электрическим стиранием (EEPROM), которые позволяют подсоединить любой из 36 входов логического блока (LAB) к источнику любого из сигналов – внешнего, внутреннего обратной связи и т. д., подключенного к вертикальным линиям матрицы.

Для увеличения гибкости и сложности реализуемых в рассматриваемых CPLD структур предусмотрены возможности расширения комбинаторной логики макроячеек. В каждой макроячейке некоторый промежуточный терм логической функции может быть через расширитель общих ресурсов (Shareable Expander) подключен к линии матрицы межсоединений для использования в других логических блоках. Соответственно в логическую функцию «ИЛИ» в каждой

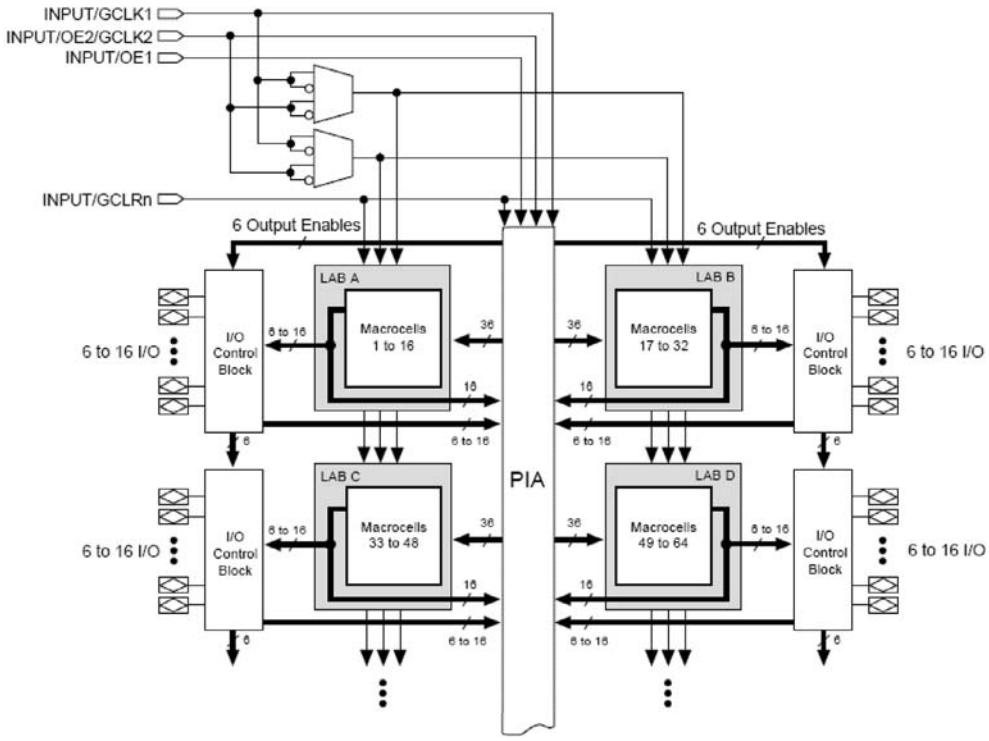


Рис. 1-8. Архитектура CPLD MAX 3000

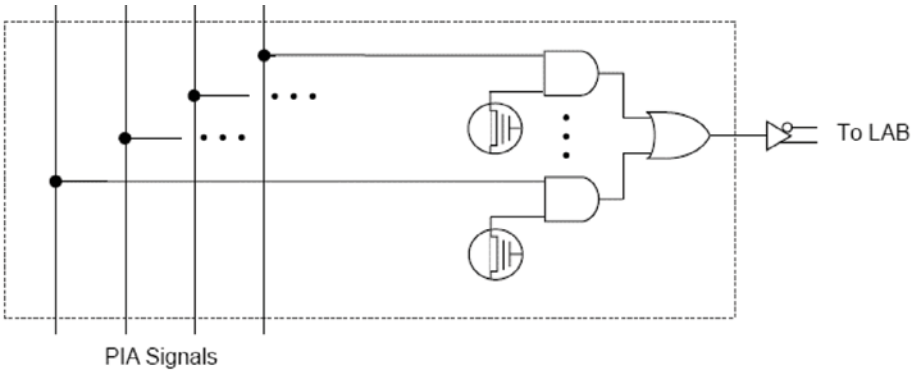


Рис. 1-9. PIA – программируемая матрица межсоединений CPLD MAX 3000

макроячейке с помощью схемы расширения может быть добавлен терм, созданный в других макроячейках. Тем самым практически снимаются ограничения на сложность логики проектируемых устройств при одновременной минимизации расходуемых ресурсов (логических вентилей).

Не останавливаясь на схемотехнических особенностях макроячейки, коротко рассмотрим, как выполнен блок управления внешними выводами MAX 3000A (I/O Control Block) – рис. 1-10.

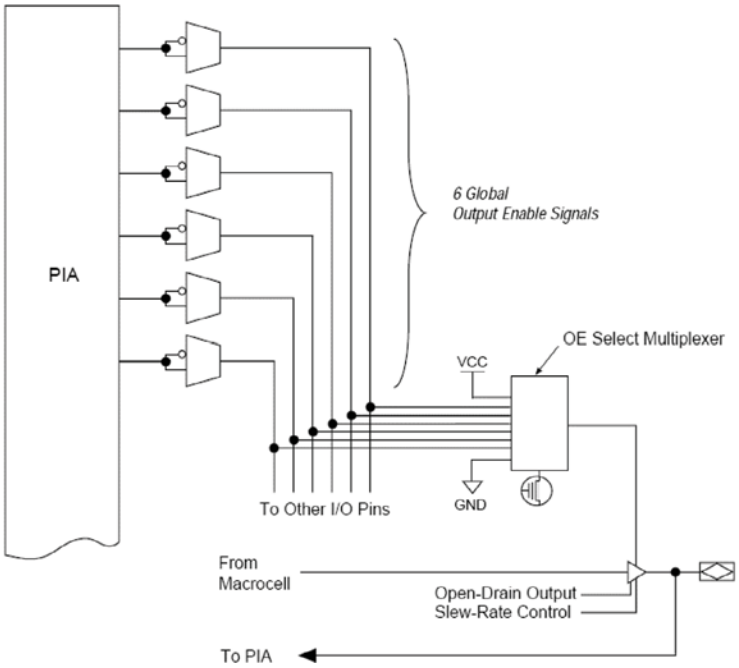


Рис. 1-10. Блок управления внешними выводами CPLD MAX 3000

Каждый контакт ввода-вывода микросхемы (I/O pin) конфигурируется как входной, выходной или двунаправленный, может устанавливаться в высокоимпедансное состояние. Индивидуальное управление логикой контакта осуществляется с помощью мультиплексора выбора режима (OE Select Multiplexer) по 6 глобальным сигналам разрешения (Global Output Signals), поступающим из матрицы межсоединений (PIA), и константным уровням «0» и «1». Дополнительно функциональность контакта расширяется сигналом перевода выхода в режим с открытым стоком (Open-Drain Output) и сигналом управления скоростью изменения выходного напряжения (Slew-Rate Control).

Дальнейшее развитие архитектуры программируемых логических устройств класса Complex PLD идет в направлении усложнения логики макроячеек, расширения возможностей конфигурирования внутренних связей, повышения быстродействия и снижения потребляемой мощности, реализации дополнительных пользовательских функций и т. п. Так, например, компания Altera выпускает CPLD с числом эквивалентных макроячеек до 2210, количеством каналов ввода-вывода до 272, возможностью выбора уровней и типа сигналов TTL-логики, бло-