

Министерство образования Российской Федерации  
Владимирский государственный университет

В.Н. ЛАНЦОВ

ПРОЕКТИРОВАНИЕ ПЛИС НА VHDL

Учебное пособие

Владимир 2000

УДК 681.31.001.8 (075.8)

Л22

Рецензенты:

Доктор технических наук, профессор, зав. кафедрой автоматизации проектирования РЭУ Московского авиационного института

*В.Н. Ильин*

Доктор технических наук, профессор, зав. кафедрой "Менеджмент" Владимирского государственного педагогического университета

*Н.Г. Наянзин*

Кандидат технических наук, нач. отдела "Информационные технологии" ТТС ОАО "Электросвязь", г. Владимир

*Ю.А. Илларионов*

Печатается по решению редакционно-издательского совета Владимирского государственного университета

**Ланцов В.Н.**

Л22 Проектирование ПЛИС на VHDL: Учеб. пособие / Владим. гос. ун-т. Владимир, 2000. 121 с.  
ISBN 5-89368-177-0

Пособие посвящено одному из быстро развивающихся направлений в области проектирования электронно-вычислительной аппаратуры - проектированию программируемых логических интегральных схем (ПЛИС) средствами языков описания цифровой аппаратуры VHDL. Представлены общие сведения и базовые архитектуры ПЛИС, основные материалы и приемы описания на языке VHDL, маршрут проектирования ПЛИС в средствах САПР.

Предназначено для студентов специальности 2201 по дисциплинам специализации САПР средств ВТ "Автоматизация функционально-логического проектирования" и "Интегрированные САПР", а также для магистров по направлению 552810. Может быть использовано при изучении дисциплин "Основы автоматизации проектирования" и "Основы теории и проектирование ЭВМ", "Моделирование", а также при подготовке студентов по специальностям 2203, 2007, 2008, 2205.

Ил. 76. Табл. 6. Библиогр.: 54 назв.

УДК 681.31.001.8 (075.8)

ISBN 5-89368-177-0

© Владимирский государственный университет, 2000

© Ланцов В.Н., 2000

## ВВЕДЕНИЕ

Повышение эффективности внедрения новой техники в различных отраслях народного хозяйства требует широкого использования средств вычислительной техники и новых методологий проектирования. Требования к подготовке инженеров-системотехников в этой области постоянно повышаются. Знание современных методологий проектирования вычислительных систем и узлов электронной и вычислительной аппаратуры (ЭВА) на базе новейших научных достижений и прикладных программных систем является определяющим при создании современной аппаратуры.

С середины 90-х годов в электронной индустрии особенно бурно развиваются персональные компьютеры, сотовая и мобильная связь, высокоскоростные устройства передачи данных. По велению рынка разработчики стремятся производить продукцию на основе увеличения функциональных возможностей, повышения технических характеристик, снижения стоимости, потребляемой мощности и меньших размеров. Для реализации этого создаются высокоинтегрированные, сложные системы с небольшим числом интегральных микросхем (ИМС), малой площадью печатных плат и в кратчайшие сроки. Современные субмикронные технологические процессы, производство печатных плат и новые технологии при изготовлении корпусов ИМС позволяют решать задачу повышения степени интеграции и минимизации размеров электронных средств. Но для производителей электронной техники часто быстрый выход на рынок с готовым изделием при высоких технических характеристиках является более важным, чем все остальные факторы. Современные средства систем автоматизированного проектирования (САПР) позволяют решать и эту проблему, а программируемые логические интегральные схемы (ПЛИС) и стандартные языки описания аппаратуры (Hardware Description Languages – HDL) становятся ключевыми элементами в новых методологиях проектирования [1].

Полузаказные и заказные специализированные ИС (Application Specific Integrated Circuits - ASIC) также используются для обеспечения решаемых проблем, но ПЛИС обеспечивают дополнительную гибкость при проектировании и модификации аппаратуры, обеспечивают более быстрый выход на рынок и более низкие затраты [1, 2].

Стандартные языки описания аппаратуры (VHDL и Verilog) очень удобны для проектирования электронных систем и устройств на основе ПЛИС и растут в популярности. При проектировании сложных (более тысячи затворов) ПЛИС инженеры уже не могут более использовать булевы уравнения или описания в виде логических схем, чтобы обеспечить быстрое и эффективное полное проектирование. Языки описания аппаратуры типа

VHDL (Very high speed integrated circuit Hardware Description Language) обеспечивают высокоуровневое описание сложнейших устройств, позволяют выполнить проектирование значительно быстрее, поддерживают создание библиотек проектирования, в которых компоненты могут быть многократно использованы в последующих проектах. Кроме того, так как это стандартные языки, то они обеспечивают совместимость кодов между средствами синтеза и моделирования, а также независимое от конкретного кристалла проектирование, с помощью их обеспечивается достаточно простое конвертирование проекта с ПЛИС в реализацию в виде полузаказной схемы или базового матричного кристалла (БМК).

Простейшие ПЛИС (до 500 затворов) уже более десяти лет применялись в промышленности. Проектирование таких ПЛИС было возможно с применением традиционных методологий проектирования на основе логического описания, синтеза на основе карт Карно и т.д. Но для сложных проектов на основе ПЛИС или ASIC традиционные методологии проектирования уже не применимы, так как требуют значительных затрат времени и сопровождаются большим числом ошибок. Схемные редакторы, используемые ранее и в настоящее время большинством проектировщиков, имеют много достоинств, например, они обеспечивают графический просмотр проекта, а если есть средства иерархического представления, то и модульность проекта. Но для больших проектов они имеют много недостатков: управляющая логика функционирования схемы представлена с использованием традиционных подходов, схемное представление трудно хранить в формализованном виде и обслуживать, схемы сопровождаются описательной документацией на разных языках, трудно воспользоваться старыми проектами в новых разработках, так как схемные модели не предназначены для многократного использования, сложна стыковка с другими системами САПР в процессе проектирования и верификации. Всегда считается, что более удобной методологией проектирования является та, которая увеличивает эффективность работы проектировщиков, имеет множество уровней представления проекта, наглядных и легко обслуживаемых, понятных или стандартных для промышленности и разных программных средств САПР, где модели могут быть многократно использованы и поддерживать иерархическое представление от схемного до системного уровня, могут быть использованы как для описания, так и для синтеза и моделирования. В настоящее время два языка удовлетворяют этим требованиям: VHDL и Verilog.

# Глава 1. ОБЩИЕ СВЕДЕНИЯ О ПЛИС

## 1.1. Технологии проектирования и изготовления специализированных ИМС

Специализированные ИМС (ASIC) делят (рис. 1):

- на заказные;
- полузаказные;
- программируемые логические интегральные схемы (ПЛИС).

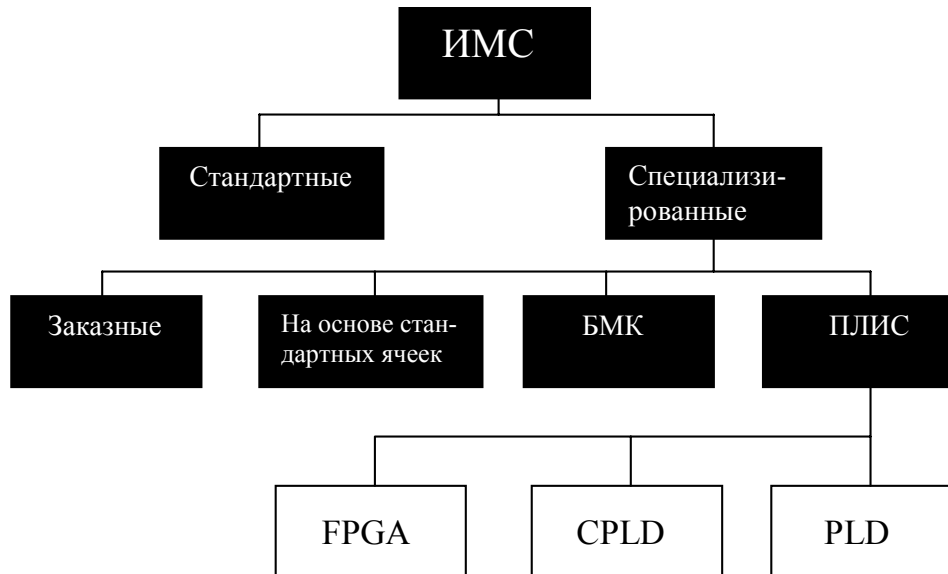


Рис. 1. Классификация ИМС по методологии (технологии) проектирования и изготовления

**Заказные** (полностью заказные, full custom) ИМС проектируются и изготавливаются по технологии *стандартных* ИМС, но в отличие от стандартных в них:

- ИМС выполняют специальные функции;
- серия выпускаемых микросхем обычно невелика.

Процесс проектирования и изготовления заказных ИМС отличается значительными затратами, большими сроками разработки (5 - 9 месяцев), но ИМС полностью удовлетворяют требованиям заказчика, обладают наилучшими техническими характеристиками, минимальными размерами кристалла. Здесь каждый прибор и схемный элемент разрабатывается под конкретный кристалл, эта технология выбирается, когда необходимо минимизировать размер кристалла или реализовать функцию, которая невыполнима или неоптимальна в полузаказных, или когда нет подобных стандартных ИМС.

К *полузаказным* (semi-custom) ИМС относят микросхемы, проектирование и изготовление которых выполняется на основе методов стандартных ячеек и базовых матричных кристаллов.

Технология на основе метода *стандартных ячеек* (*standard cells*) получила наибольшее распространение в мире и имеет следующие особенности:

- основана на применении при проектировании заранее разработанных компонентов (ячеек);
- в качестве ячеек могут быть отдельные транзисторы, логические элементы, триггеры, регистры, АЛУ, ОЗУ и т.п.;
- для ячеек имеется полная технологическая, схемная и конструкторско-топологическая информация, которая хранится в библиотеках САПР;
- проектирование заключается в покрытии исходной (проектируемой) схемы набором ячеек, размещении отобранных ячеек на кристалле и трассировки межсоединений;
- нет сокращения числа фотошаблонов по сравнению с заказными;
- сокращение процесса проектирования за счет использования готовых библиотек стандартных ячеек, которые уже полностью известны и являются, по сути, заказными;
- ИМС полностью удовлетворяют требованиям заказчика, обладают практически наивысшими техническими характеристиками и минимальной площадью кристалла;
- технология характеризуется более низкой стоимостью проектирования и готовых кристаллов (по сравнению с заказной), малыми сроками проектирования (2 - 3 месяца) и изготовления (2 - 3 недели).

Технология на основе метода *базовых матричных кристаллов* (БМК, БИС на БМК, МаБИС, gate arrays) имеет следующие отличительные особенности:

- проектирование основано на использовании заранее подготовленных кристаллов (по заказной технологии) с уже имеющимися элементами чаще всего в виде матрицы одинаковых элементов, а также ряда элементов коммутации и ввода/вывода и т.п.;
- все схемы и элементы БМК прошли полную аттестацию (по электрическим и топологическим параметрам), нет только соединений между элементами, необходимы одна-две операции по выполнению соединений с помощью одного-двух слоев металлизации;
- проектирование схем на БМК можно выполнить или в фирме-разработчике БМК по запросу заказчика, или непосредственно пользователем под конкретную схему;

- БМК поставляются с библиотекой параметров элементов матриц и коммутаций;
- БМК могут быть на основе вентильных матриц для цифровых ИМС или на основе аналоговых матриц кристаллов и функциональных элементов для аналоговых ИМС;
- достоинствами технологии являются низкая стоимость проектирования и изготовления изделий на БМК, малые сроки проектирования и изготовления (1 - 2 месяца);
- недостатки: несколько большая площадь кристалла (по сравнению с заказными и на основе стандартных ячеек), неполное использование кристалла (лишние элементы, избыточность БМК), несколько худшие технические характеристики (по сравнению с заказными).

## **1.2. Простейшие программируемые интегральные схемы**

До недавних пор стандартные цифровые микросхемы (ТТЛ-логика, у нас 133-я и 155-я серии, а за рубежом 54xx и 74xx серии) были основным инструментом проектировщика цифровой аппаратуры. Данные микросхемы реализовали простейшие логические функции И/ИЛИ-НЕ и чуть более сложные, использовались для построения устройств комбинационной и последовательностной логики: мультиплексоров, кодеров и декодеров, регистров, счетчиков, конечных автоматов и др.

Процесс проектирования цифровых устройств на основе стандартных микросхем (“рассыпной логики”) включал следующие шаги. Вначале по описанию (спецификации) проекта строилась таблица истинности, затем при использовании булевой алгебры и карт Карно выполнялась минимизация логических выражений. Для сложных логических выражений на данном этапе использовались программы логического моделирования, которые позволяли разработчику убедиться в правильности функционирования логики. Далее в зависимости от требований проектирования (минимизации числа ИМС, уровней логики, суммарной задержки, стоимости или их комбинация) выбирались ИМС и выполнялась реализация логических выражений в выбранном базисе стандартных микросхем. Затем выполнялось временное логическое моделирование с учетом задержек логических элементов выбранного базиса микросхем. На последней стадии выполнялось размещение микросхем на плате и трассировка межсоединений, физическая реализация устройства на плате и отладка работы устройства при различных тестовых сигналах на входе. Необходимо отметить, что на каждом из этих этапов при обнаружении некорректных решений выполнялся возврат на один или несколько шагов назад и процесс проектирования повторялся.

Недостатками данного подхода являются:

1. Не все этапы процесса проектирования автоматизированы, а если и автоматизированы, то используются примитивные средства.
2. Длительный и дорогостоящий процесс проектирования, изготовления и отладки.
3. Выполнение физической реализации устройства на плате (платах) требует значительных временных ресурсов, часто является источником многих ошибок и требует кропотливой отладки работы устройства.
4. Использование ранее разработанных устройств (или их частей) практически невозможно при реализации новых проектов.

Поэтому в современных условиях разработчики цифровой аппаратуры используют новые методологии и современные САПР на основе ПЛИС и HDL. В этом случае процесс проектирования будет сводиться к описанию схемы на языке описания аппаратуры (поведенческое описание спецификации), использованию средств САПР, где выполняется синтез схемы по исходному описанию, моделированию и реализации схемы в выбранном кристалле, затем формируется файл загрузки (конфигурации). Запрограммированная ИМС может далее быть помещена на плату или использоваться в более крупных проектах. Недорогие и практически полностью автоматизированные средства САПР и методология проектирования ПЛИС позволяют разработчику концентрироваться больше на функциональных вопросах, чем на реализации проекта в кристалле.

*Простейшие ПЛИС* (PLD - Programmable Logic Devices, программируемые логические приборы, или ПЛМ - программируемые логические матрицы) состоят из двух логических матриц [3, 4]: матрицы элементов И, на которую подаются прямые и инверсные значения входных сигналов, и матрицы ИЛИ, на которую подаются сформированные в матрице И логические произведения входных сигналов (рис. 2). Таким образом, PLD реализует логическую сумму произведений.

Эти устройства, в свою очередь, делятся на PROM, PLA и PAL в зависимости от того, структура какой из логических матриц является фиксированной, а какой - программируемой. В устройствах PROM (Programmable Read Only Memory) логическая матрица И является фиксированной, а матрица ИЛИ - программируемой. В устройствах PLA (Programmable Logic Array) обе логические матрицы являются программируемыми, что делает их наиболее гибкими. Расплатой за это являются снижение быстродействия и повышение цены. В устройствах PAL (Programmable Array Logic) программируемой является логическая матрица И, а матрица ИЛИ является фиксированной. Эти устройства сочетают гибкость, свойственную PLA, с быстродействием, свойственным PROM.



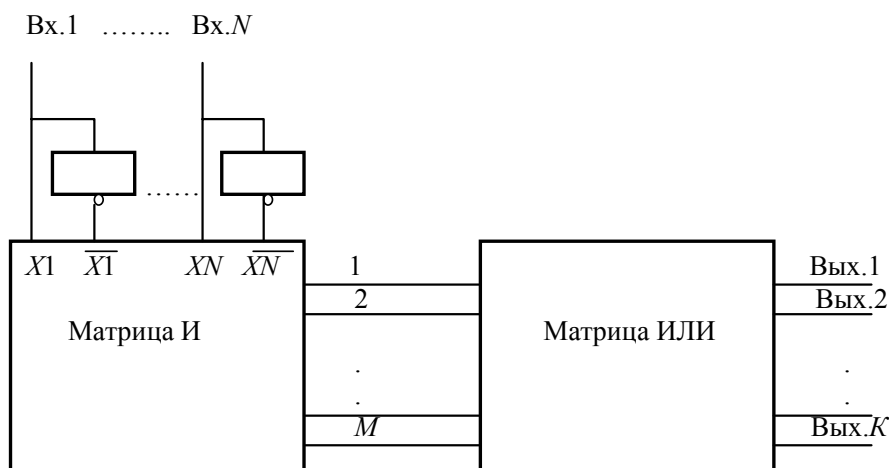


Рис. 2. Структура простейших программируемых приборов

Прежде чем приступить к описанию структур и принципов работы PLD (ПЛИМ), рассмотрим систему принятых обозначений при описании элементов программируемых схем. На рис. 3 приведен пример трехвходового элемента И и его эквивалентного представления в ПЛИМ.

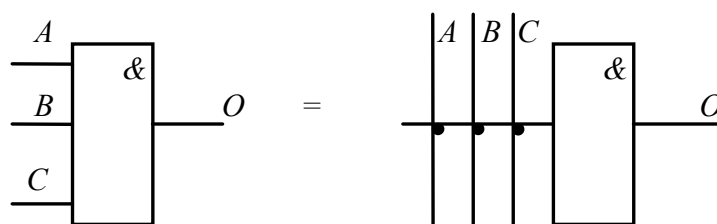


Рис. 3. Стандартный элемент И и его эквивалент в PLD

Одиночная линия на входе И используется для представления нескольких входов. Вертикальные линии представляют сигналы  $A$ ,  $B$  и  $C$ . Точка на пересечении линий обозначает программируемое соединение между входными сигналами  $A$ ,  $B$  и  $C$  и входом элемента И. Программирование этого соединения выполняется с помощью EEPROM или иной технологии, которые мы рассмотрим позже. Еще один пример (рис. 4) иллюстрирует реализацию логической функции

$$F = A\bar{B} + \bar{B}C + \bar{A}B\bar{C}.$$

В этом примере каждая вертикальная линия (шина) соединена с входными сигналами или их инверсными значениями.

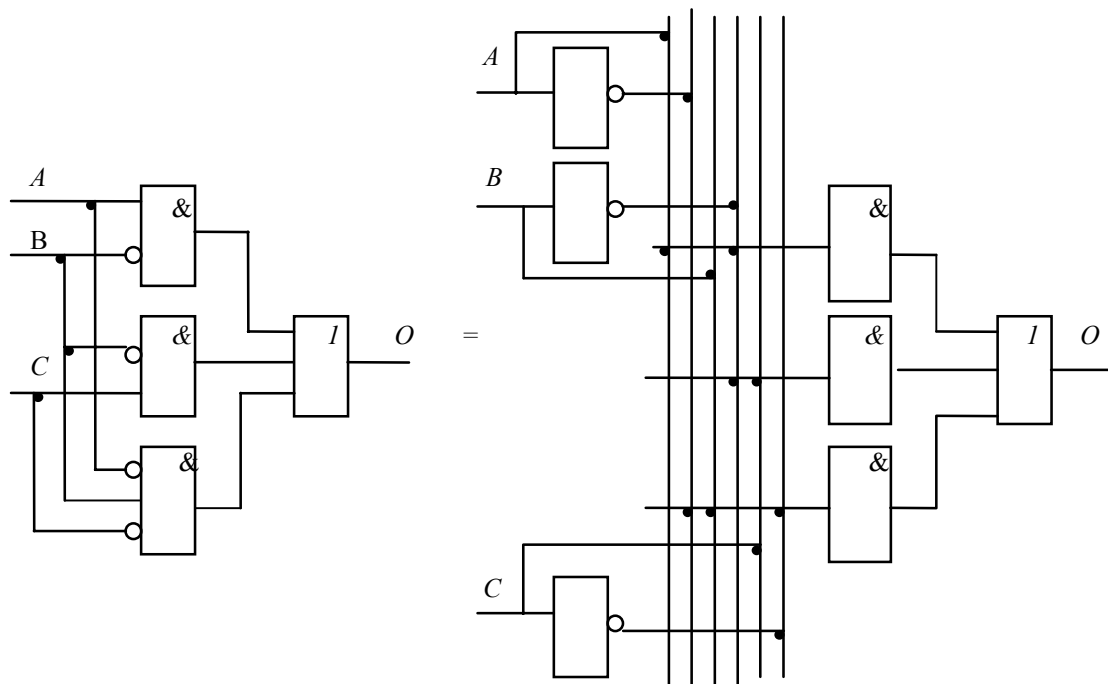


Рис. 4. Пример реализации логической функции в PLD

На рис. 5 представлена структура PLA [4] в принятой системе обозначений.

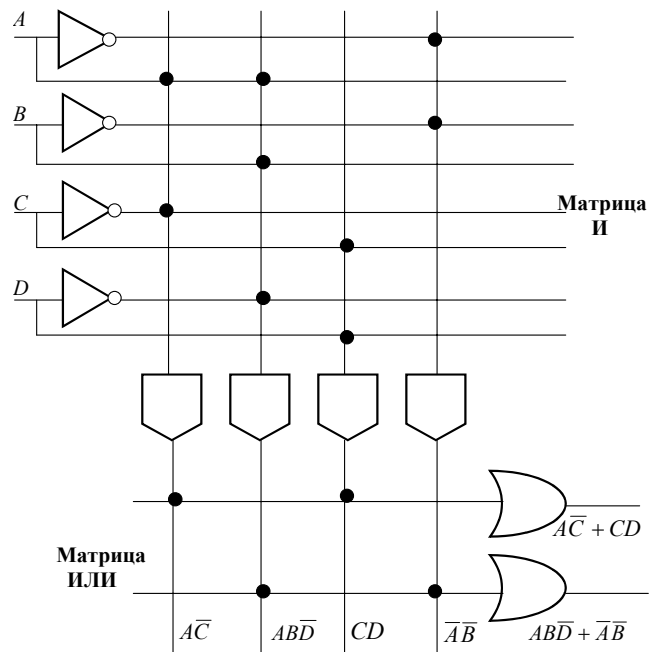


Рис. 5. Структура PLA<sup>1</sup>

Микросхемы со структурой PLA были одними из первых программируемых приборов и появились в начале 70-х годов. Все элементы соединя-

<sup>1</sup> Здесь и далее на рисунках приведены символы логических элементов, принятые в зарубежных стандартах и системах САПР, используемых в данном пособии.

ются между собой матрицами программируемых переключателей. В PLA каждый входной сигнал и его инверсное значение поступают на И-матрицу по горизонтальным линиям. Вертикальные линии матрицы И, которые являются входами элементов И, принимают только тот сигнал, где есть пересечение горизонтальных и вертикальных линий (на рис. 5 обозначены черной точкой в матрице И). Например, первый слева элемент И выполняет логическую операцию И над сигналом *A* и инверсным значением сигнала *C*. Аналогичным образом работает матрица ИЛИ. Таким образом, PLA выполняет функцию суммы произведений. Гибкость, предоставляемая структурой PLA, когда обе матрицы являются программируемыми, не всегда используется. Поэтому инженеры часто используют более простую структуру PAL, показанную на рис. 6.

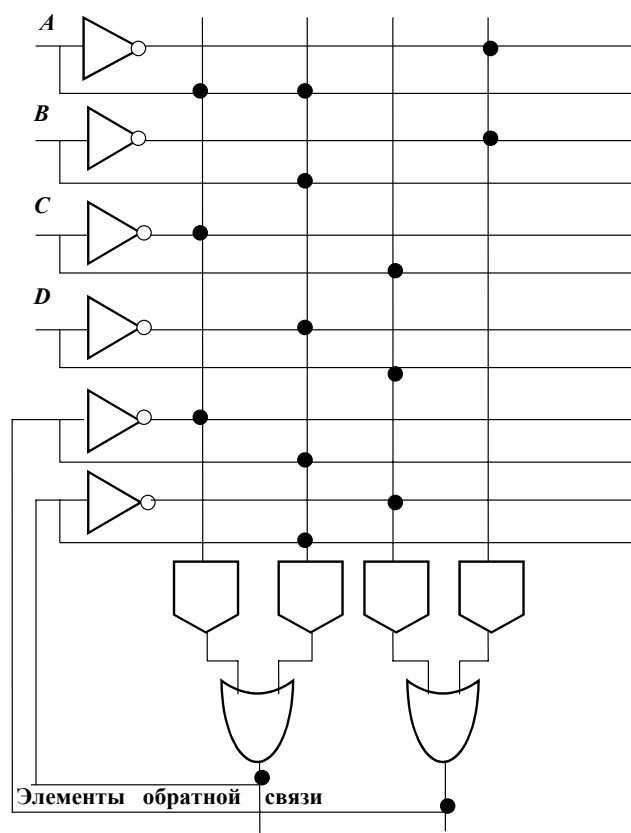


Рис. 6. Структура PAL

PAL-приборы очень удобны для проектирования комбинационной логики. Но они не могут быть использованы для проектирования последовательностных схем без применения внешних триггеров. Поэтому в приборах PLD (простые PLD – simple PLD) добавлены триггеры, как показано на рис. 7. С помощью мультиплексов, добавляемых на каждый выход, осуществляется выбор выхода (комбинационный или последовательностный).

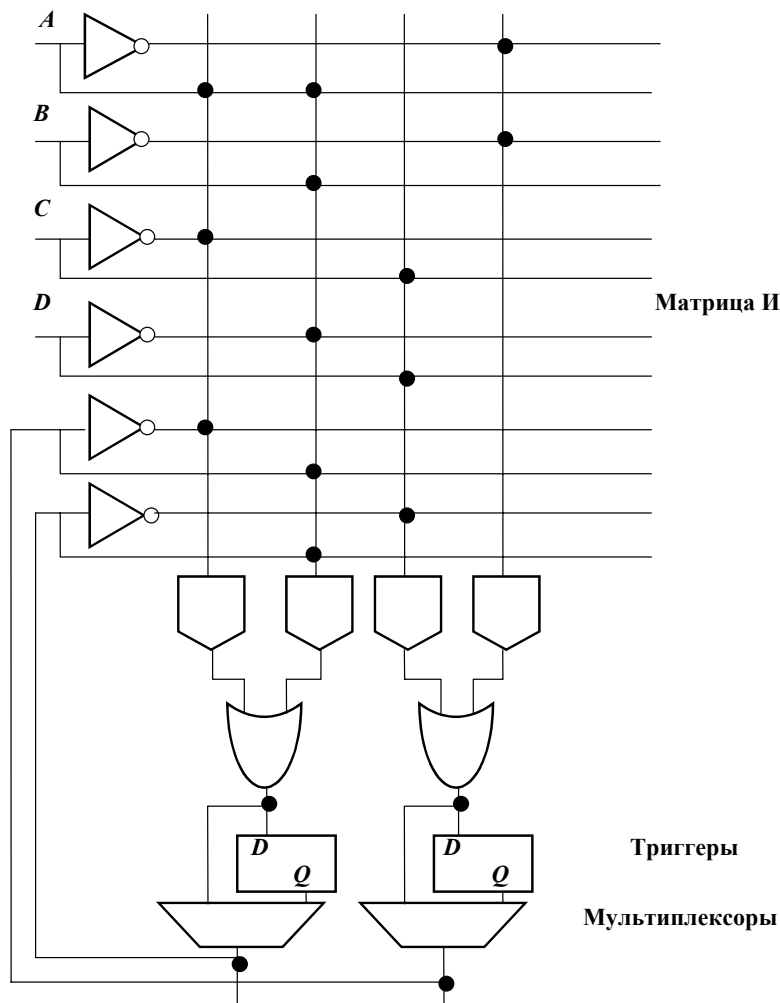


Рис. 7. Структура PLD (SPLD)

### 1.3. Примеры простейших программируемых приборов

Далее для разбора PLD-архитектур рассмотрим микросхему PAL 16L8 (рис. 8), которая поставлялась большинством фирм, включая “AMD”, “Cypress”, “Lattice”, “National Semiconductor”, “Texas Instruments” [1]. Название 16L8 обозначает, что имеется 16 входов на матрицу И и 8 выходов, L - логическая матрица. Восемь контактов ( $I_1 - I_8$ ) определяют входы микросхемы, а остальные восемь контактов ( $O_1 - O_8$ ) являются выходами. Часть выходных контактов являются входами/выходами (I/O pin). Если тристабильные инверторы, связанные с I/O pin, будут установлены (включены), то тогда активизируется обратная связь к логической матрице, которая инвертирована, так как тристабильный буфер является инвертором. В этом случае источник сигнала будет поступать не из I/O-элементов, а из инверсного буфера, который управляется элементом ИЛИ. Если тристабильный инвертор не установлен (не включен), то I/O-элемент работает как вход, и тогда все

контакты будут входными.

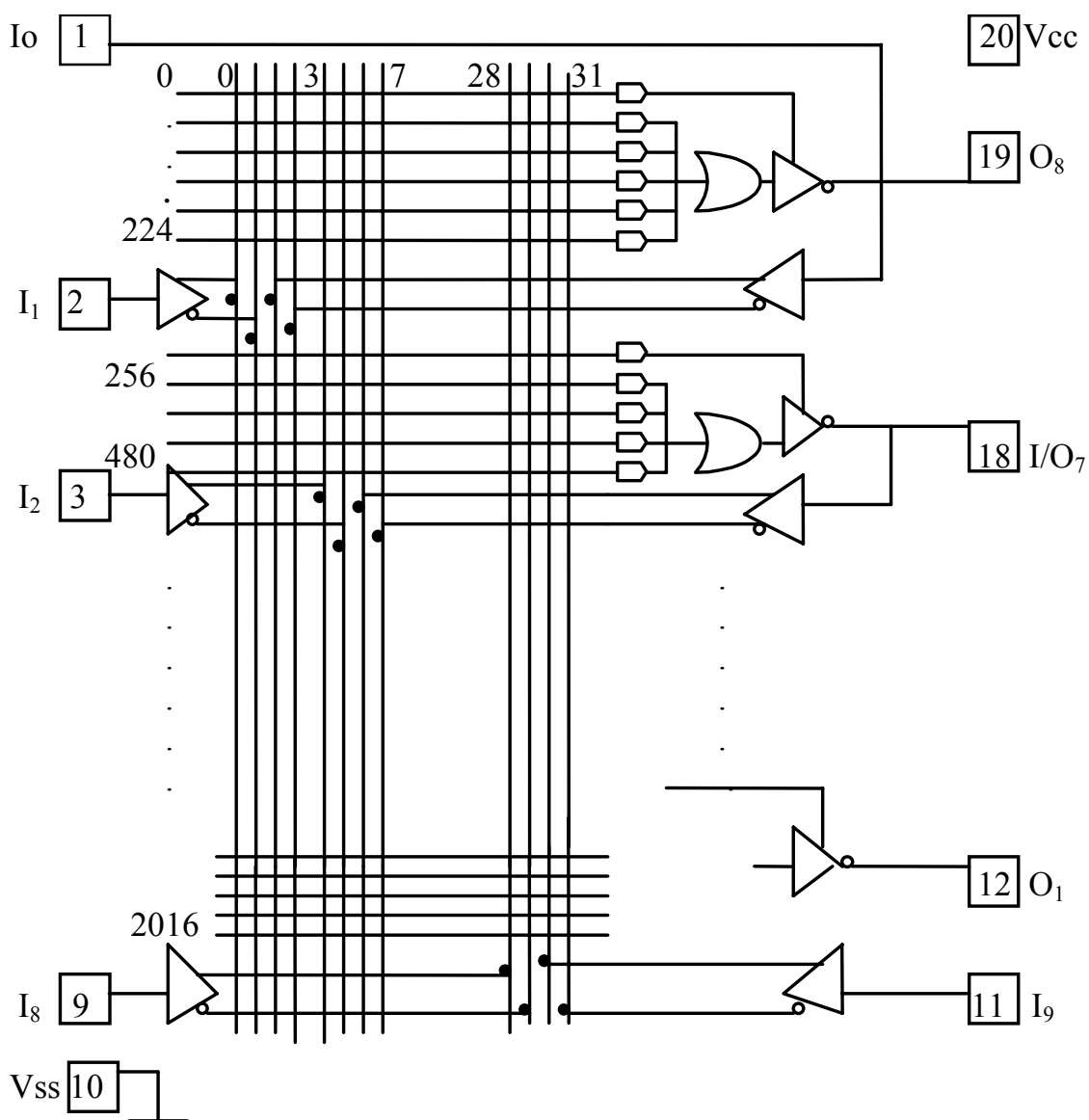


Рис. 8. Архитектура микросхемы PAL16L8

Программируемая матрица И состоит из 64 И-элементов, каждый из которых может быть использован для выполнения произведения с любыми из 16 входов или их инверсными значениями. Матрица ИЛИ фиксирована, каждый из 8 ее элементов суммирует семь произведений. Восьмой элемент используется для управления тристабильным буфером. Этот элемент может быть запрограммирован как функция входов и их инверсных значений или как режим постоянного включения и выключения тристабильного буфера. Если он всегда включен, то соответствующие I/O-элементы функционируют только как выходы. Если он всегда выключен, то соответствующие I/O-элементы функционируют только как входы. Если эти I/O управляются ло-

гической матрицей (суммой произведений), то они могут быть использованы для управления внешней шины или для двунаправленного прохождения сигналов.

В микросхеме PAL 16R8 (рис. 9) имеется 16 входов, логическая матрица также содержит 64 элемента, но в этой архитектуре 8 входов логической матрицы поступают из входных контактов схемы, а остальные 8 - из регистров в обратной связи [1]. Эти регистры являются очень важными при построении схем с дополнительными состояниями, таких как триггеры, счетчики, сдвиговые регистры и т.д. Символ R в названии микросхемы обозначает присутствие регистра. Сигнал синхронизации для триггеров поступает с отдельного входного контакта. Здесь выходы не могут быть сконфигурированы как входы, но могут быть запрограммированы как элементы с тремя состояниями для использования внешней шины.

Имеется несколько популярных промышленных стандартов для простых PLD, с которыми обычно сравнивают другие ПЛИС, это 16V8, 20G10 и 22V10. Рассмотрим наиболее простую из них - 22V10 (рис. 10). В отличие от PLA данная архитектура включает *программируемые макроячейки* (macrocells) и переменную структуру формирования логических произведений. Каждая макроячейка (рис. 11) может быть индивидуально конфигурирована (табл. 1) путем программирования состояния битов конфигурации ( $C_0$ ,  $C_1$ ).

Т а б л и ц а 1

*Варианты конфигурирования макроячейки*

$C_1$	$C_0$	Описание
0	0	регистр/низкий уровень
0	1	регистр/высокий уровень
1	0	комбинационный/низкий
1	1	комбинационный/высокий

Биты конфигурации (входы макроячейки) являются суммой произведений и могут поступать на регистры или передаваться прямо на выходной буфер. Полярность выхода макроячейки также конфигурируется с помощью этих битов.

*Управление полярностью выхода* позволяет использовать либо прямую, либо инверсную логику при реализации логических выражений. Это упрощает выбор логического выражения с меньшим числом произведений, например, инверсия большого числа суммы отдельных элементов

$$X = A + B + C + D + E + F$$

может быть выражена как один элемент произведения

$$\bar{X} = \bar{A} \cdot \bar{B} \cdot \bar{C} \cdot \bar{D} \cdot \bar{D} \cdot \bar{E} \cdot \bar{F}.$$

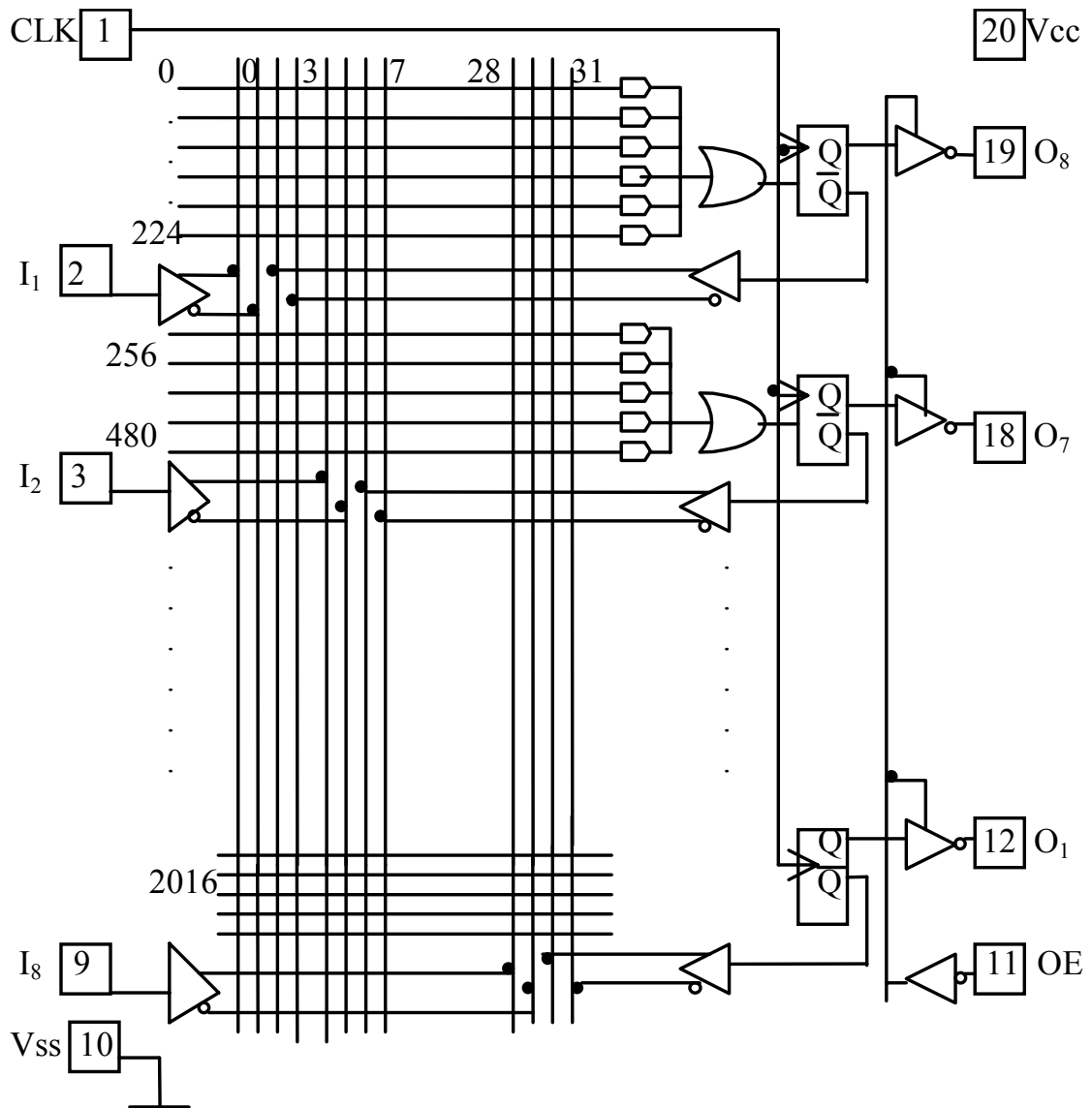


Рис. 9. Архитектура микросхемы PAL16R8

Так как для реализации И-элемента используется больше приборных ресурсов, чем для ИЛИ, то более эффективно применять  $\bar{X}$ . Инверсия от  $X$ , т.е.  $\bar{X}$ , может затем быть получена инвертированием в выходном буфере  $X = \overline{(\bar{X})}$ .

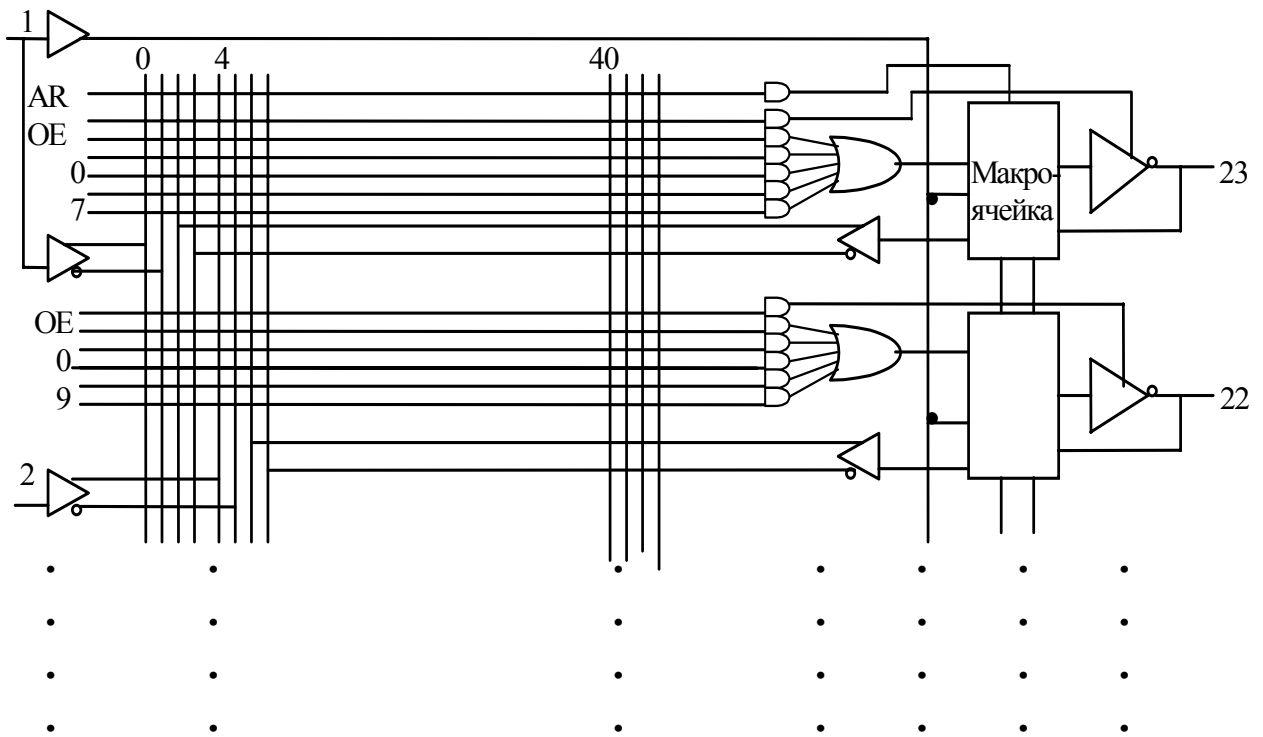


Рис. 10. Архитектура микросхемы PLD 22V10

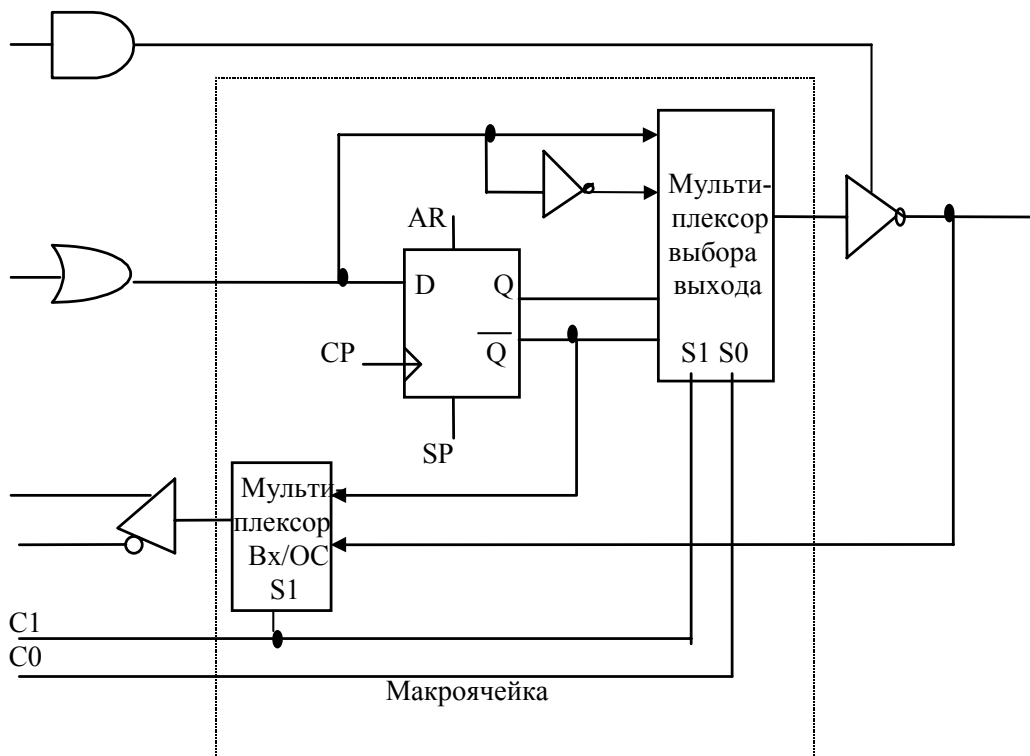


Рис. 11. Архитектура макроячейки PLD 22V10

Кроме того, возможность выбора полярности выхода макроячейки позволяет программному обеспечению легче выполнять логическую оптими-



зацию при синтезе. Программируемая макроячейка имеет также обратную связь, используемую как вход для логической матрицы. Обратная связь может быть направлена от регистра или I/O-буфера, в зависимости от того, был или не был сигнал на регистре. Другим новшеством в 22V10 является то, что число шин произведений может быть переменным (символ V – variable, в названии 22V10). В данном примере число произведений (элементов И) на каждый ИЛИ-элемент варьируется от 8 до 16. Число 22 означает двадцать два входа для логической матрицы, а 10 - число выходов.

#### 1.4. Сравнение технологии ПЛИС с другими технологиями

Технология ПЛИС бурно прогрессирует. В настоящее время ПЛИС характеризуются самыми высокими темпами роста применения в области ИМС (рис. 12), позволяют в сжатые сроки создавать специализированные СБИС до сотен тысяч логических вентилях, позволяют разработчику отказаться от десятков корпусов стандартной логики. Использование стандартных микросхем ТТЛ и КМОП серий имеет следующие недостатки: большой размер плат, низкую надежность из-за значительного числа корпусов и соединений между ними, сложность отладки и настройки устройств на их основе.

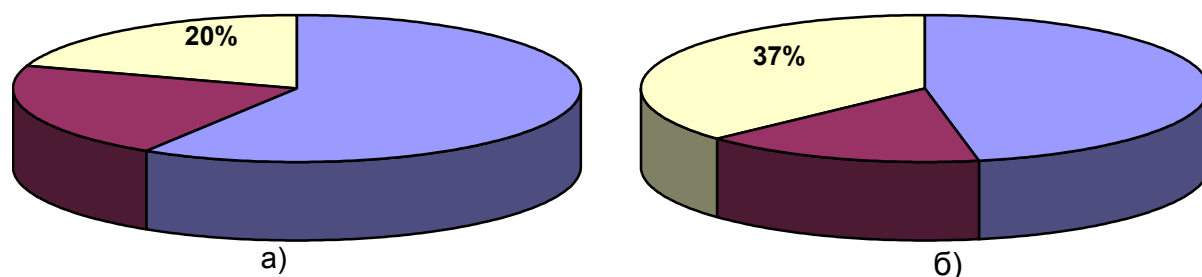


Рис. 12. Сравнение ПЛИС с другими технологиями:  
а – суммарный рынок ИМС в 1996 г. – 9,5 млрд долл.: ■ - ПЛИС – 1,9 млрд долл.; ■ - стандартные ИМС – 2,0 млрд долл.; ■ - ASIC – 5,6 млрд долл.; б – суммарный рынок ИМС в 2001 г. – 15,8 млрд долл.: ■ - ПЛИС – 5,8 млрд долл.; ■ - стандартные ИМС – 2,6 млрд долл.; ■ - ASIC – 7,4 млрд долл.

Основным отличием ПЛИС от БМК и других типов специализированных ИМС является возможность проектирования, отладки и тиражирования ИМС самим разработчиком непосредственно на своем рабочем месте [5, 6]. Это обусловлено как технологией изготовления ПЛИС, так и наличием развитых средств САПР. ПЛИС позволяют значительно уменьшить га-

бариты аппаратуры, при этом достигается высокое быстродействие, низкое энергопотребление, значительно меньшие сроки проектирования (несколько дней) и изготовления (минуты), легкость тестирования и высокую рабочую надежность; особенно это актуально для российских условий.

Выделим следующие достоинства ПЛИС:

- резкое сокращение сроков и затрат на проектирование, возможность модификации и отладки аппаратуры, а также эмуляции схем, подлежащих последующей реализации на другой элементной базе, например на БМК;
- снижение стоимости первоначальной разработки;
- минимизация ручного труда инженера;
- домашнее проектирование (in house) - весь цикл проектирования и конфигурирования (программирования) выполняется на одном рабочем месте;
- возможность перепрограммирования, применение перепрограммируемых ПЛИС позволяет проводить эффективную отладку и реконфигурирование;
- высокая надежность;
- возможность сохранять проект и использовать его как часть в других проектах;
- возможность выполнять отладку и эксперименты с множеством типов кристаллов без физического исполнения устройства;
- сохранение интеллектуальных свойств проекта (проект не передается для изготовления в другую фирму).

К недостаткам ПЛИС следует отнести:

- невысокую рабочую скорость (тактовые частоты меньше, чем в других технологиях);
- невысокую плотность (размеры кристалла превышают размеры по другим технологиям).

ПЛИС используют [7]:

- для изготовления прототипов, когда нужно быстро проверить спецификацию проекта;
- когда нужно малое количество изделий;
- когда ожидается модификация спецификаций.

Применяют для схем:

- цифровой электроники: счетчики, регистры, дешифраторы, мультиплексоры, шины, контроллеры, адресные дешифраторы, логика обрaмления микропроцессоров, формирователи управляющих сигналов (например для микросхем памяти) и др;
- телекоммуникации, передачи данных, компьютерных и промышленных применений;

- электронных ключей для защиты аппаратных и программных средств от несанкционированного доступа и копирования;

- бытового применения от узлов детектора валют, пэйджеров, мультипортовой платы, контроллера ЖКИ до сложных систем обработки изображений, аппаратуры телефонных станций, банкоматов и т.д.

На мировой рынок лидеры производства ПЛИС вышли в 80-е годы ("Altera" - 1984, "Xilinx" - 1985, "Actel" – 1988 г.). В 1997 году общий объем продаж ПЛИС (CPLD и FPGA) составил 2,252 млрд долл. (1990 - 100 млн, 1991 - 270 млн). Лидерами являются:

1. "Xilinx" - 566 млн (1990 - 75 млн, 1991 – 170 млн)
2. "Altera" - 497 млн
3. "AMD" - 320 млн
4. "Lattice" - 240 млн
5. "Actel" - 149 млн
6. "Lucent" - 100 млн

За пять лет объем продаж [8] вырос в 12 раз (25-30% в среднем в год, самые высокие темпы роста в области ИМС).

На рынке FPGA явный лидер – фирма "Xilinx" (рис. 13).

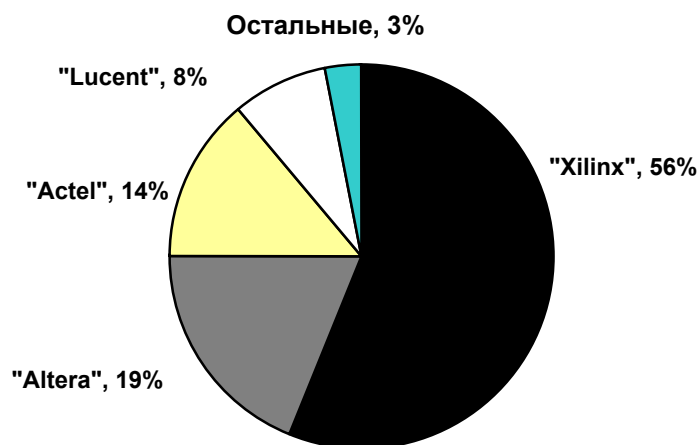


Рис. 13. Рынок FPGA в 1997 году

### 1.5. Технологии программирования кристаллов

Для программирования (создания соединений) в PLD (а также и CPLD) используются технологии EPROM, EEPROM и FLASH.

**EPROM.** На рис. 14 показано, как ячейки EPROM используются для создания функции И-соединения [1].

В действительности это ИЛИ-НЕ-функция, но если инверсные сигналы доступны (как это возможно в PLD), то эквивалент функции И может

быть легко получен, например:

$$\overline{(a + b)} = \bar{a} \cdot \bar{b}.$$

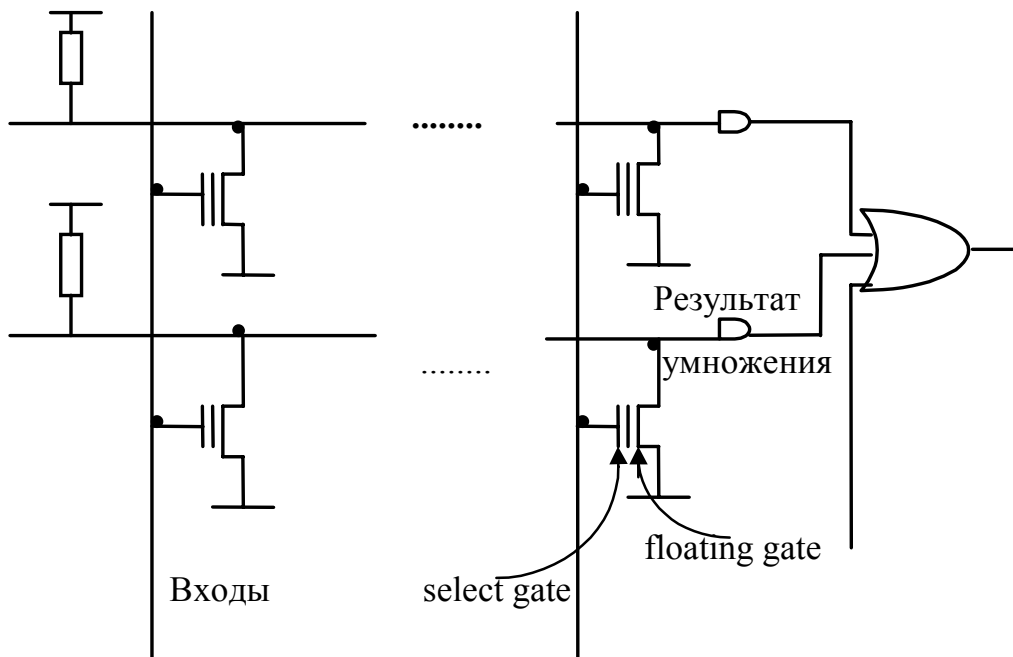


Рис. 14. Ячейка EPROM для создания функции И

Линия (шина) результата (функции умножения) первоначально устанавливается в высокий уровень (“1”) с помощью специального механизма. Если любой из транзисторов будет проводить сигнал (открыт), то шина результата будет иметь низкий уровень (“0”). Транзистор, используемый в EPROM, имеет два затвора: select (access) gate (затвор “выбора/доступа”) и floating gate (“плавающий” затвор) (рис. 15), где плавающий затвор полностью изолирован.

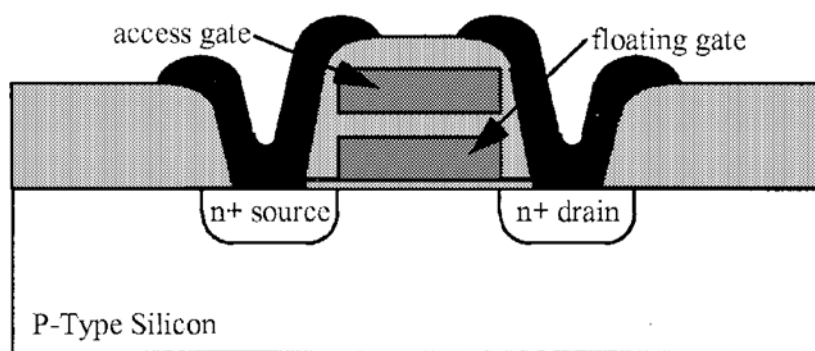


Рис. 15. Структура EPROM/EEPROM-транзистора

Механизм программирования транзистора осуществляется с помощью так называемых “горячих” электронов. Этот механизм запускается подачей программирующих напряжений на сток и затвор выбора (select gate) транзи-

стора. Большое электрическое поле от стока к истоку дает электронам дополнительную энергию, часть горячих электронов будет иметь достаточную энергию для преодоления энергетического барьера и будет попадать в плавающий затвор (floating gate). Эти электроны или их заряд имеют эффект увеличения порогового напряжения транзистора. Таким образом, в нормальном рабочем режиме, даже если на входной шине имеется высокий уровень, который прикладывается к затвору выбора запрограммированного транзистора, транзистор не будет пропускать сигнал (закрыт). Без дополнительного заряда на плавающем затворе транзистор работает как обычный *n*-канальный МОП-транзистор, когда при подаче напряжения на затвор транзистор открывается.

EPROM-ячейки обеспечивают механизм программирования, который используется в PLD и CPLD для создания или несоздания соединения. Для стирания ячейки (удаления заряда из плавающего затвора) микросхема обрабатывается ультрафиолетовым (УФ) облучением. Электроны преодолевают энергетический барьер в обратном направлении после подачи достаточной энергии от УФ-облучения. Типичное время стирания около 35 минут при высокоинтенсивном УФ-облучении.

**EEPROM.** Следующий рис. 16 иллюстрирует, как EEPROM-ячейки используются для создания функции И. Здесь каждая ячейка содержит два транзистора: транзистор “выбора” (select) и транзистор “памяти” (storage). Как и в предыдущем случае шина результата логического умножения (функция И) вначале устанавливается в высокий уровень с помощью специального механизма. Если какая-либо ячейка будет проводить сигнал (транзистор открыт), то шина результата будет равна низкому уровню. Как и в EPROM-ячейке storage-транзистор имеет два затвора: select и floating. Этот транзистор очень похож на транзистор в EPROM, отличие только в том, что пленка оксида здесь намного тоньше - 100 ангстрем (там было 200). В EEPROM механизм накопления заряда в floating gate основывается на туннельном эффекте. Прикладывая напряжение программирования к select gate и нуль к стоку, туннельный ток через пленку оксида передает заряд в floating gate. Важно отметить, что этот процесс может быть легко изменен в обратную сторону путем прикладывания значительно большего напряжения к стоку и нулевого к select gate. При данном способе требуется select-транзистор, так как, когда floating gate не имеет заряда, пороговое напряжение транзистора отрицательное. Таким образом, незапрограммированный транзистор смог бы проводить ток между истоком и стоком, если входная шина была бы соединена с select gate, и потенциал входной шины был нулевой. Select-транзистор добавлен потому, что нам необходимо, чтобы он пропускал ток только когда входная шина имела высокий уровень и floating gate не имел бы заряда. Рис. 17 иллюстрирует четыре возможных состояния

двух транзисторов.

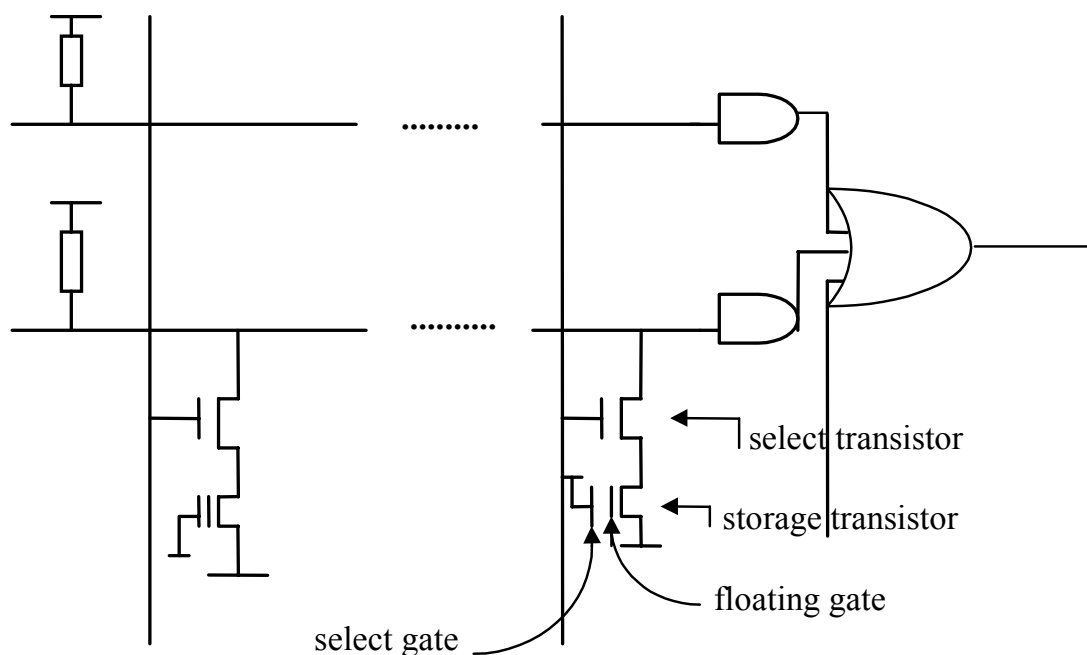


Рис. 16. Ячейка EEPROM для создания функции И

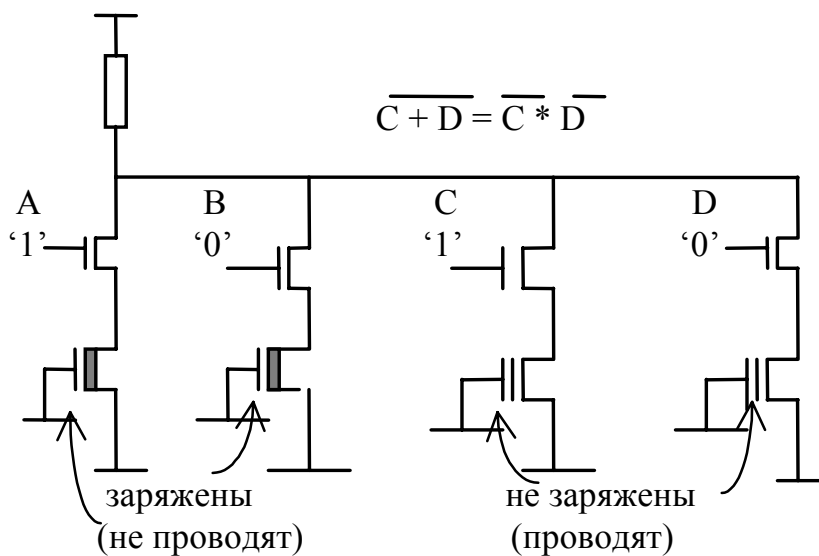


Рис. 17. Возможные состояния ячейки EEPROM

**FLASH.** Как и EEPROM-ячейки, FLASH-ячейки также содержат два транзистора - select и storage, и они создают функцию И на шине похожим способом. Но storage-транзистор здесь такой же, как и в EPROM, т.е. программирование осуществляется за счет горячих электронов, однако floating gate управляется стирающим транзистором так, что заряд уходит с помо-

щью туннельного эффекта. Основное отличие между этими технологиями для конечного пользователя - это то, что EEPROM и FLASH используют электрическое стирание, в то время как EPROM требует УФ-облучения, которое требует большего времени и дороже. Поэтому в современных приборах используют только EEPROM- и FLASH-технологии, которые являются прекрасными средствами для перепрограммирования и программирования в системе (ISP). Важным отличием технологий для CPLD по сравнению с далее рассматриваемой технологией FPGA является то, что исходные соединения для всех элементов уже существуют, при программировании решается задача разрушения этих соединений. После окончания процесса программирования только необходимые соединения будут присутствовать в реализуемом приборе. Состояния этих элементов будут сохраняться даже при отключении питания. Но для программирования обычно требуется довольно высокое напряжение, и поэтому программирование кристалла непосредственно в системе (ISP) не рекомендуется, хотя и возможно.

### 1.6. Временные (timing) параметры PLD

Имеется множество временных параметров и характеристик для оценки работы цифровых устройств, например, в некоторых приложениях используются такие параметры, как минимальная длительность синхросигнала (clock width), допустимая задержка от входа к выходу, время асинхронной переустановки. Но основными временными параметрами, которые наиболее часто используются в литературе и технической документации, являются [1]:

- задержка распространения ( $t_{PD}$  - propagation delay);
- время установки ( $t_S$  - setup time);
- время задерживания ( $t_H$  - hold time);
- задержка от синхросигнала до выхода ( $t_{CO}$  - clock-to-output delay);
- задержка от синхросигнала до выхода через логическую матрицу ( $t_{CO2}$  - clock-to-output delay through the logic array);
- время от системы к системе ( $t_{SCS}$  - system clock to system clock time), которое используется для определения максимальной частоты выполнения операций.

Задержка распространения - это время прохождения сигнала через комбинационную логику от входа до выхода. Время установки - это величина времени, необходимого для установки входов триггеров, при котором можно подать синхросигнал. Время задерживания - промежуток времени, когда входы триггеров станут стабильными после подачи синхросигнала. Задержка от синхросигнала до выхода - это величина времени после подачи синхросигнала на вход и оценки отклика на выходном контакте. Эти вре-

менные параметры иллюстрирует рис. 18.

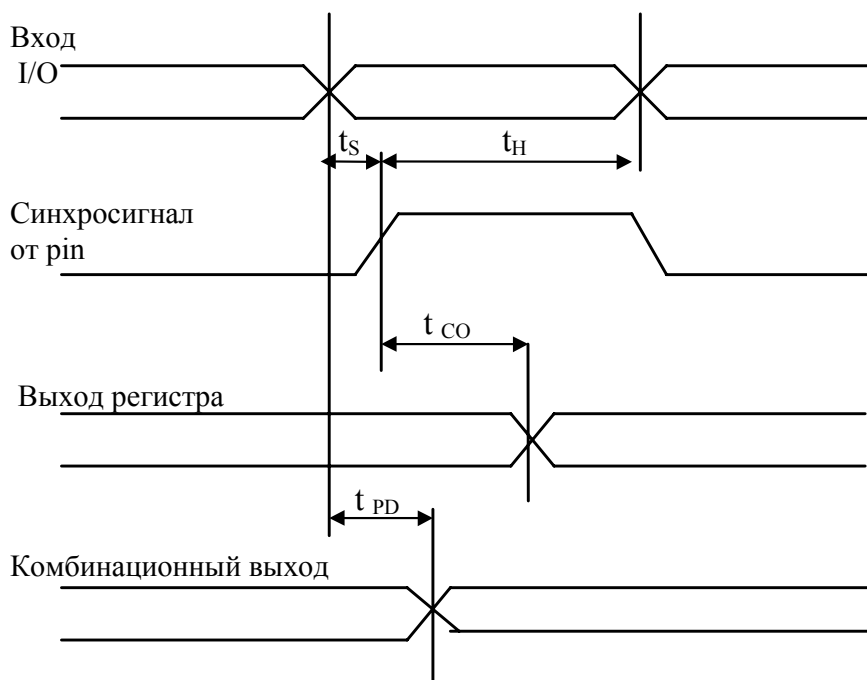


Рис. 18. Временные параметры

Параметр  $t_{CO2}$  представляет задержку от входа синхросигнала до выхода для сигнала, прошедшего через элементы обратной связи (если оценка выполняется для выхода, который соединен с регистром, - это  $t_{CO}$ ). Таким образом, параметр  $t_{CO2}$  представляет задержку между синхросигналом и выходом для сигнала через обратную связь, через регистр и логическую матрицу, макроячейку, сконфигурированную для комбинационного режима и далее к контакту прибора (рис. 19).

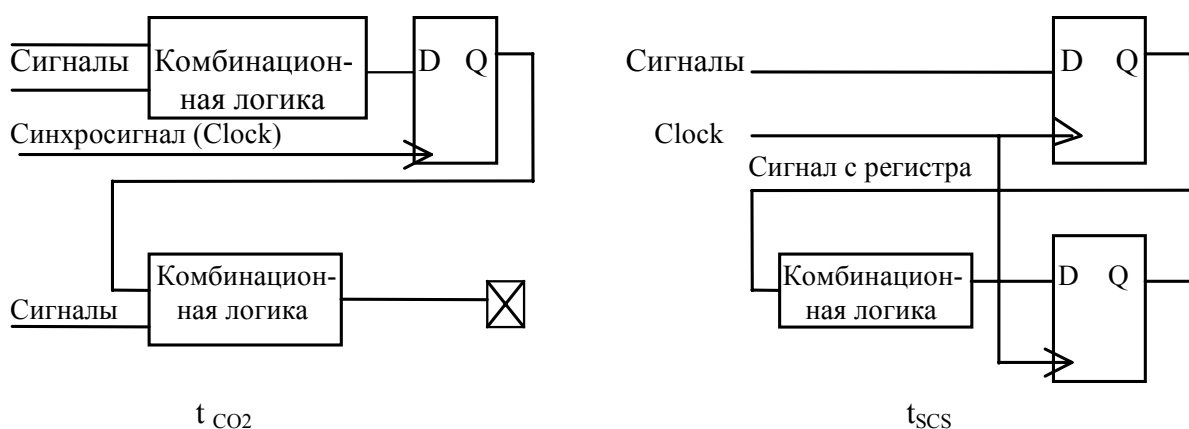


Рис. 19. Оценка временных параметров  $t_{CO2}$  и  $t_{SCS}$

На этом же рисунке представлена схема, когда в элементе обратной связи используется второй регистр, который может быть синхронизирован



только через промежуток времени  $t_{SCS}$  после первого. Параметр  $t_{SCS}$  указывает минимальный период синхросигналов для операций типа "регистр-регистр" и оценивается величиной времени, необходимого для установившегося режима в первом регистре и подаче на второй регистр. Этот параметр используется для вычисления максимальной частоты работы (выполнения операций):  $f_{max} = 1 / t_{SCS}$ .

Микросхема 22V10 при  $t_{PD} = 4$  нс имеет следующие параметры (табл. 2):

Т а б л и ц а 2

Параметр	Значения, нс	
	минимальное	максимальное
$t_{PD}$	-	4
$t_S$	2.5	-
$t_H$	0	-
$t_{CO}$	-	3.5
$t_{CO2}$	-	7
$t_{SCS}$	-	5.5

### Темы для самоконтроля

1. Классификация технологий проектирования и изготовления ИМС (сравнение по техническим характеристикам, срокам проектирования и изготовления).
2. Процесс проектирования и недостатки цифровых устройств на основе стандартных ИМС средней степени интеграции ("рассыпной логики").
3. Классификация простейших программируемых ИМС (PROM, PLA, PAL, PLD, SPLD) и их отличительные особенности.
4. Архитектуры простейших ПЛИС.
5. Назначение основных блоков и особенности микросхем PAL16L8 и PAL16R8.
6. Архитектура ПЛИС 22V10, особенности макроячеек данной ПЛИС.
7. Достоинства, недостатки и области применения ПЛИС.
8. Технологии программирования PLD и CPLD.
9. Временные параметры PLD и способы их оценки.

### Глава 2. СЛОЖНЫЕ PLD (ПЛИС типа CPLD)

## 2.1. Архитектура CPLD

Комплексные (сложные) PLD (CPLD - complex PLD) расширяют концепцию PLD на более высокий уровень интеграции. Здесь уже содержится множество логических блоков, каждый из которых похож на простые PLD, типа 22V10. Логические блоки связываются друг с другом с помощью так же программируемых межсоединений (рис. 20). Такая архитектура более эффективна для использования площади кристалла, приводит к лучшим характеристикам и сокращает стоимость. CPLD позволяют реализовать значительно больше логики, чем PLD, которые ориентированы на двухуровневую логику, в то время как архитектура CPLD ориентирована на реализацию многоуровневой логики.

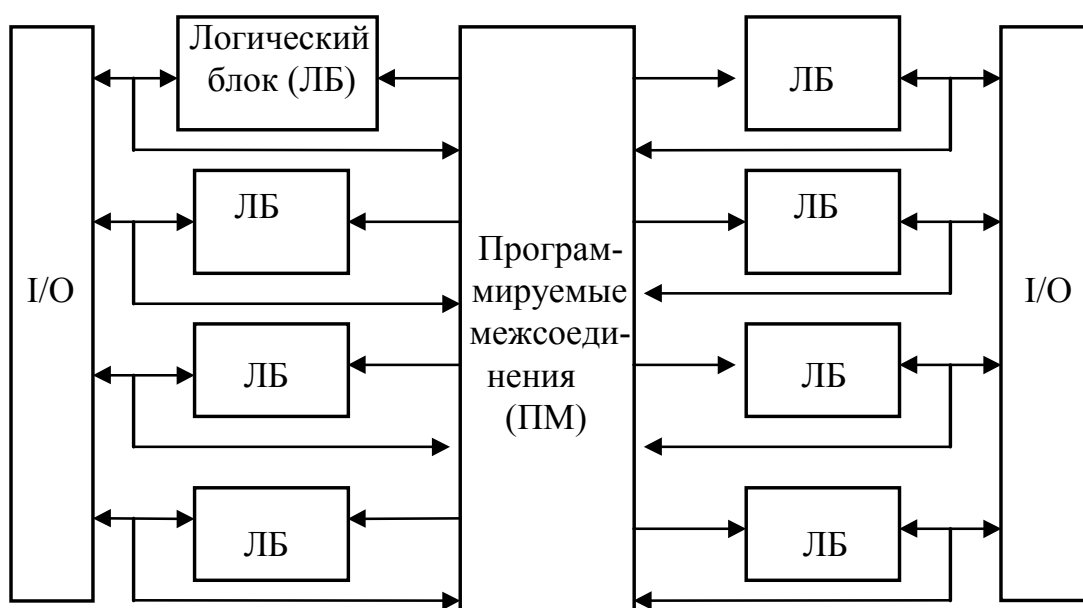


Рис. 20. Обобщенная архитектура CPLD

Чтобы разобраться далее в более сложных архитектурах CPLD, рассмотрим еще раз PLD 22V10 (см. рис. 10) с несколько иных позиций назначения отдельных блоков и введем некоторые определения [1].

Под термином “*входы логической матрицы*” (logic array inputs), или просто входы, будем понимать все сигналы, которые входят в логическую программируемую матрицу. Эти входы включают входы микросхемы, входы с обратных связей от макроячеек и входы от I/O, которые конфигурируются как входы микросхемы. Термин “*матрица произведений*” (product-term array), или выходы логической матрицы, представляет программируемые функции И. Термин “*схема распределения*” (product-term distribution scheme) будет обозначать механизм распределения шин результатов выполнения логического умножения (функция И) по макроячейкам. *Макроячейки*

(macrocells) обычно содержат регистровую (триггер) и комбинационную части с управлением полярности, а также один или более элементов обратной связи. Термин “ячейки ввода/вывода” (I/O cells) используется для описания структуры I/O-буферов. В этом случае, структура логического блока CPLD в определенных нами терминах представлена на рис. 21.

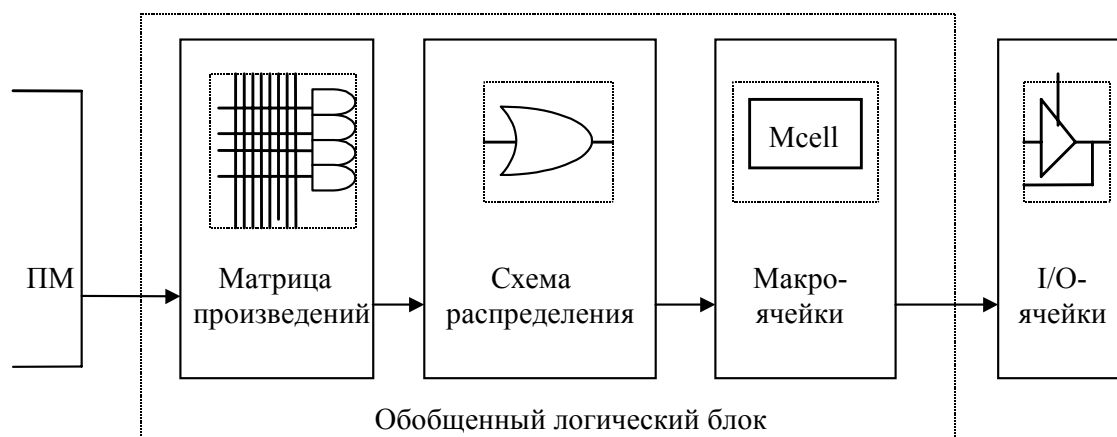


Рис. 21. Структура логического блока CPLD

Рассмотрим кратко некоторые особенности архитектур CPLD, выпускаемых разными производителями ПЛИС.

## 2.2. Программируемые межсоединения

Программируемые межсоединения (ПМ) выполняют соединения между входными сигналами от I/O-ячеек и входами логических блоков или между выходами логических блоков (выходы макроячеек) и входами тех же самых или других логических блоков (некоторые логические блоки имеют местную обратную связь, так что выходы макроячейки, использующиеся в тех же самых логических блоках, не соединятся через глобальные программируемые межсоединения). Так же как и в PLD (на примере 22V10), где было фиксированное значение входов логической матрицы, логический блок имеет фиксированное число входов.

Большинство CPLD используют одну из двух реализаций для программируемых межсоединений: *межсоединения на основе матрицы* (array-based interconnect) и *межсоединения на основе мультиплектора* (multiplexer-based interconnect) [1]. Межсоединения на основе матрицы позволяют любому сигналу в ПМ соединиться с любым логическим блоком (рис. 22). Каждый элемент в ПМ представляется вертикальной шиной и является входом (через высокочувствительный усилитель sens amp) для выбранного логического блока, так что имеется один элемент ПМ для каждого входа логического блока.

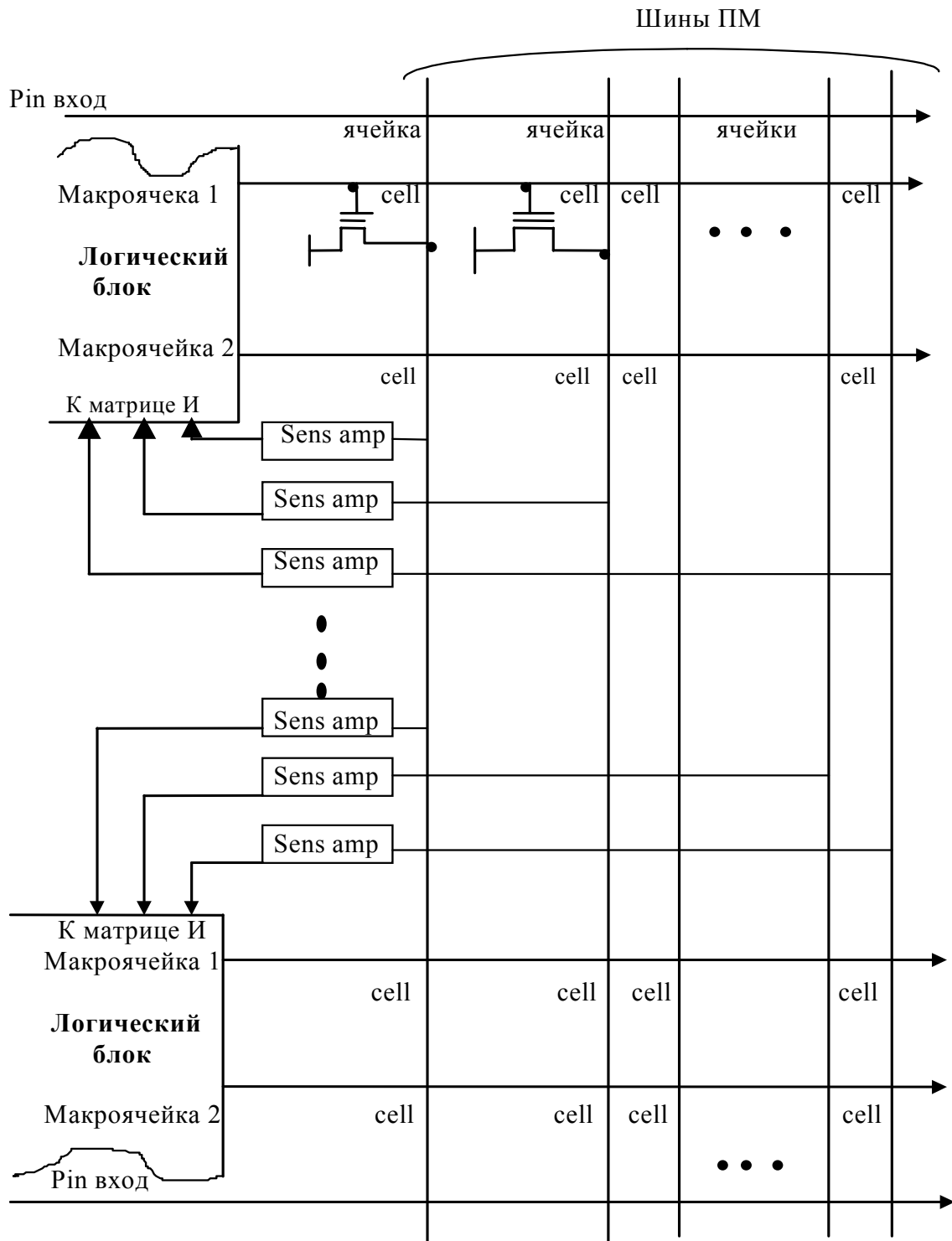


Рис. 22. Межсоединения на основе матрицы

Выход логического блока может быть присоединен к одному из элементов ПМ через элемент памяти (такой же, как EPROM-ячейка). Входы микросхемы могут соединяться с элементами ПМ аналогичным образом.

Такая схема межсоединений реализует полный набор переключений, т.е. любой вход в ПМ может быть соединен с любым логическим блоком. При очень высокой гибкости этого способа полный набор переключений требует большей площади кристалла и снижает технические характеристики ПЛИС.

При межсоединениях на основе мультиплексора (рис. 23), имеется один мультиплексор для каждого входа в логический блок. Сигналы ПМ соединяются с входами разных мультиплексоров для каждого логического блока.

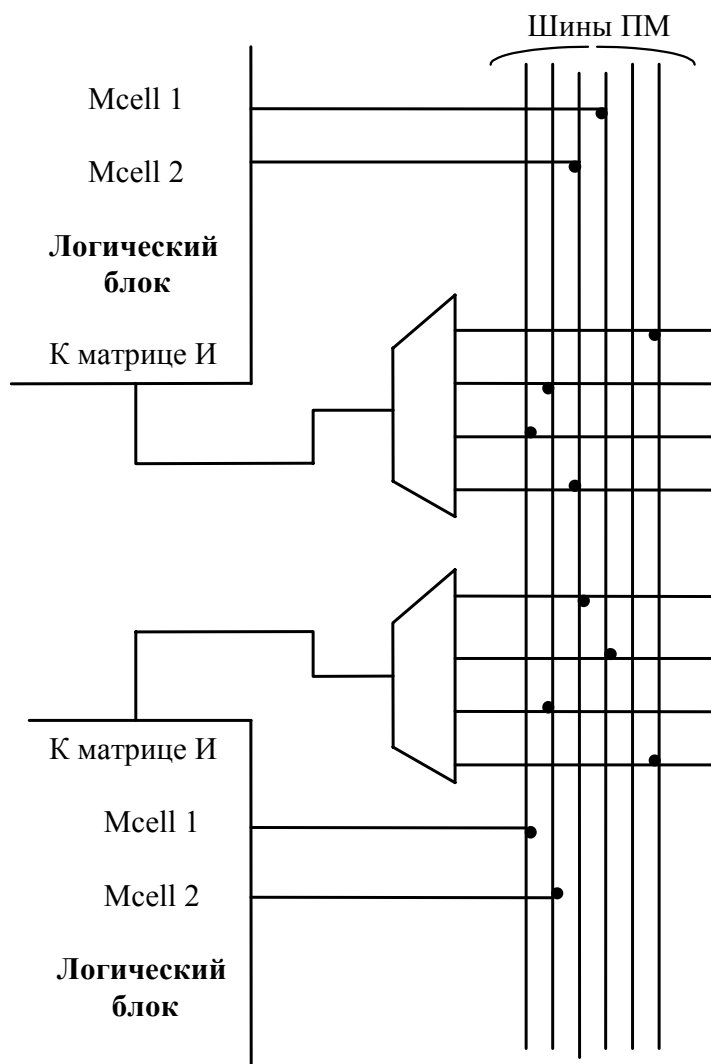


Рис. 23. Межсоединения на основе мультиплексора

Шины выбора таких мультиплексоров программируются (например, с помощью FLASH ячеек), так чтобы обеспечивался один вход для каждого мультиплексора для передачи сигнала в логический блок. Возможности более гибкого и более полного переключения могут быть увеличены при использовании несколько большего числа мультиплексоров (wider multiplexers). В этом случае, каждый сигнал ПМ соединяется с входами не-

скольких мультиплексоров для каждого логического блока. Но увеличение числа мультиплексоров на каждый логический блок увеличивает площадь (могут быть ухудшены и технические характеристики) кристалла.

### 2.3. Логические блоки

Логические блоки CPLD очень похожи на PLD типа 22V10: каждый имеет матрицу произведений, схему распределения и макроячейки. I/O-ячейки иногда рассматриваются как часть логического блока, но мы будем рассматривать их отдельно, как было изображено на рис. 21.

Размер логического блока измеряется его емкостью - т.е. количеством логических элементов, которые могут быть в нем реализованы. Обычно это выражается числом макроячеек, но важными также являются число входов в логический блок, число элементов произведений и схема распределения. Логические блоки обычно бывают в диапазоне от 4 до 20 макроячеек. Шестнадцать или более макроячеек обеспечивают 16-разрядные функции при реализации в одном логическом блоке при условии, что достаточное число входов от ПМ к логическому блоку существует.

**Матрицы произведений.** Имеется незначительная разница в матрицах произведений в различных CPLD. Важнейшей характеристикой является размер матрицы, так как он определяет среднее число элементов произведений на макроячейку и максимальное число элементов произведений на логический блок.

**Схемы распределения.** Одним из главных отличий схем CPLD для разных разработчиков и ПЛИС разных семейств является наличие различных схем распределения. Семейство MAX (рис. 24), разработанное совместно корпорациями "Altera" и "Cypress", было первым семейством CPLD на рынке (фирма "Altera" называет его как MAX5000, "Cypress" – как MAX340). Два из возможных вариантов схем распределения мы рассмотрели в примерах приборов PAL 16L8 и PLD 22V10. В первом варианте была фиксированная схема распределения, когда каждые восемь элементов произведения (элементов И) были фиксировано связаны с одним элементом ИЛИ. Во втором - применялась переменнo-фиксированная структура, где использовалось фиксированное число элементов И, но оно было разным (переменным) для разных элементов ИЛИ. В семействе MAX имеется 4 элемента произведений на одну макроячейку, но дополнительно имеется несколько элементов произведений (expander product terms), которые индивидуально распределяются для любой макроячейки или макроячеек. Эти дополнительные элементы назначаются только тем макроячейкам, которые необходимы при реализации.

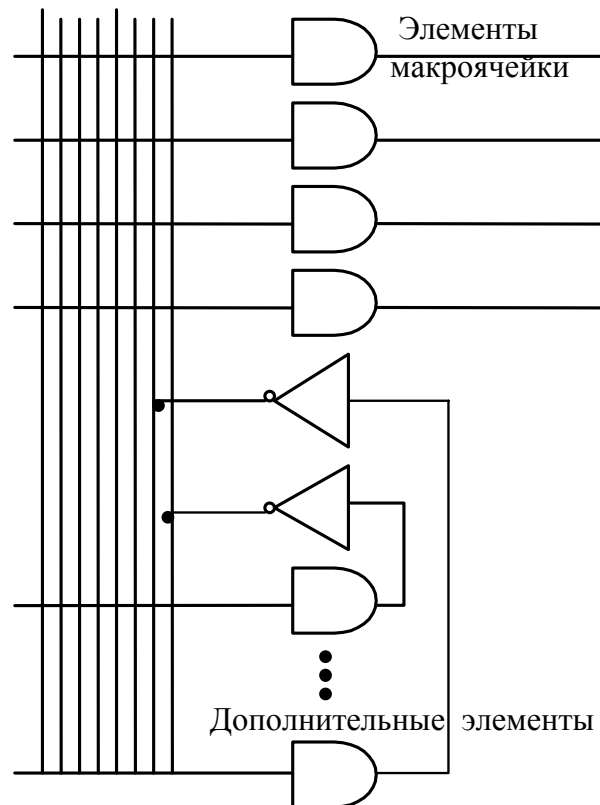


Рис. 24. Схема распределения в семействе MAX5000

Концепция, в которой элементы могут использоваться конкретной макроячейкой, называется "управление произведениями" (product-term steering), а концепция, в которой те же самые элементы могут использоваться множеством макроячеек, называется "дележ произведений" (product-term sharing). Но для данного варианта схемы распределения появляется дополнительная задержка для сигналов, которые должны передаваться через логическую матрицу перед распространением в макроячейку.

Семейство MAX7000 (рис. 25) фирмы "Altera" развивает схему распределения семейства MAX5000. Кроме дополнительных элементов, которые могут индивидуально управляться или делиться, в новую архитектуру включен механизм управления произведениями. В данном случае пять элементов могут управляться соседней макроячейкой. Это добавляет значительно меньшую задержку, так как сигнал не проходит через матрицу произведений, как это было в MAX5000.

Семейство Flash370 фирмы "Cypress" содержит еще одну, отличную от предыдущих схему распределения (рис. 26). Здесь каждая макроячейка связывается с элементами (от 0 до 16) произведений в зависимости от требований логического выражения, используемого в данной макроячейке.

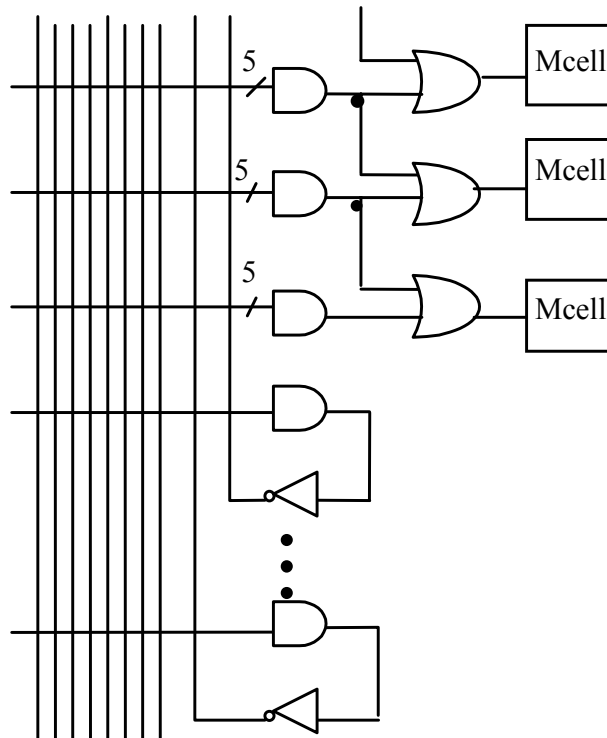


Рис. 25. Схема распределения в семействе MAX7000

Каждый элемент может индивидуально управляться. Такие схемы распределения обеспечивают гибкость для проектировщиков, но более важно то, что они обеспечивают гибкость для программного обеспечения САПР при выборе ресурсов при реализации.

**Макроячейки.** Как и в простых PLD, в сложных PLD используются макроячейки, имеющие триггерные элементы и элементы управления полярностью сигналов. Управление полярностью помогает реализации различных схем в ПЛИС и особенно синтезу логики по исходному выражению. Основное отличие макроячеек CPLD в том, что они имеют больше возможностей по конфигурированию, чем в устройствах типа 22V10. Например, многие включают триггеры, которые могут быть сконфигурированы как D- или T-тип с целью более эффективной реализации логических выражений. Дополнительно многие CPLD имеют несколько типов макроячеек: I/O-макроячейки, входные и внутренние (закрытые - buried) макроячейки. PLD типа 22V10 имела только I/O-макроячейку (т.е. макроячейку, связанную с I/O). Входная макроячейка, как можно догадаться, связана с входными портами (pin). Внутренние макроячейки обычно очень похожи на I/O-макроячейки за исключением того, что их выход не может распространяться непосредственно в I/O и обычно является обратной связью в ПМ. Некоторые варианты макроячеек мы рассмотрим позже, при изучении CPLD фирмы "Xilinx".



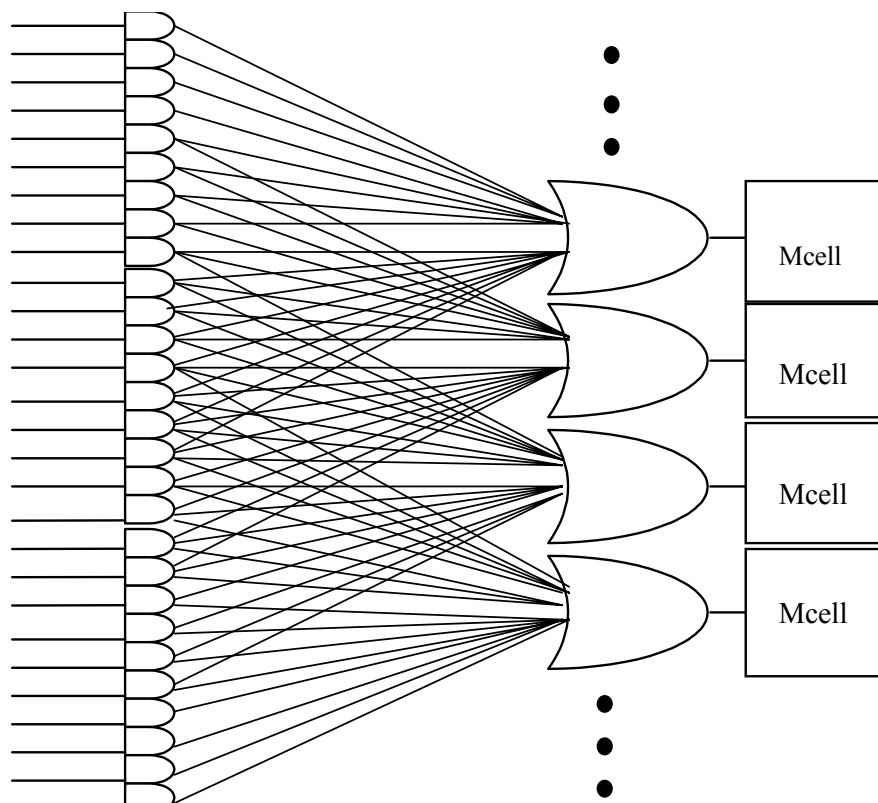


Рис. 26. Схема распределения в семействе Flash370

## 2.4. Временные параметры CPLD

Временные параметры в CPLD такие же, как и в PLD: задержка распространения, время установки, задержка от синхросигнала до выхода и от регистра до регистра. Временные параметры после реализации в CPLD обычно являются более предсказуемыми, чем для FPGA (причины, по которым делаются такие выводы, будут пояснены позже). Для некоторых проектировщиков это свойство CPLD является главным достоинством по сравнению с FPGA, так как перед началом проектирования характеристики проекта могут быть предсказаны с достаточной точностью. Поэтому конкретный прибор можно выбрать задолго до его реализации, зная, что в конце проектирования инженер получит почти желаемые параметры.

Например, для прибора 22V10 характеристики проектирования могут быть оценены до проектирования, если точно разобраться, сколько раз потребуется пропустить (прогнать) требуемое логическое выражение через логическую матрицу. Любое логическое выражение, имеющее больше чем 16 элементов (произведений), будет требовать дополнительный прогон. Выражение до 112 элементов ( $16+16+14+14+12+12+10+10+8=112$ ) может быть реализовано в два прогона. Таким образом, если при проектировании

число прогонов может быть определено, то и временные параметры могут быть легко оценены.

## 2.5. Дополнительные возможности CPLD

Кроме рассмотренных ранее возможностей и характеристик в CPLD имеются еще ряд дополнительных возможностей: программирование в системе (in-system programmability - ISP), перепрограммирование в системе (in-system reprogrammability - ISR), работа в режиме 5V/3.3V, наличие портов доступа для тестирования и пограничного сканирования (boundary scan), поддерживающего стандарт IEEE 1149.1 (JTAG - Joint Test Action Group), входные и выходные буферы для поддержки стандарта PCI (Peripheral Component Interconnect). Кроме того, приборы могут поставляться в различных корпусах.

*Программирование в системе* - это возможность программирования прибора прямо на плате. Это сокращает маршрут изготовления готовой системы и экономит время, так как нет необходимости в операции программирования. Это, в частности, важно для современных корпусов и, кроме того, позволяет сразу помещать все программируемые и непрограммируемые части на плату, затем программирование может быть выполнено с помощью дополнительной карты или через автоматическое тестовое оборудование (automatic test equipment - АТЕ).

*Перепрограммирование в системе* - это возможность перепрограммирования прибора, когда он уже находится в схеме. Это особенно важно при изготовлении прототипов, изменении напряжения питания, изменении функций схемы во время работы в системе.

*Спецификация JTAG* определяет метод тестирования функционирования прибора и соединений с другими приборами на плате через специальный порт тестирования и *пограничного сканирования*. При передвижении данных с помощью пограничного сканирования межсоединения с другими приборами могут быть верифицированы, а тестовый вектор - применен для проверки внутренней логики прибора. JTAG - это методология, которая может быть использована для тестирования и оценки качества разделения схемы на подсистемы. Для ограничения размерности вектора, который используется для сканирования и должен быть синхронизирован, JTAG также определяет режим встроенного тестирования (BIST - built-in self-test). Прибор в этом режиме генерирует псевдослучайные тестовые векторы в качестве стимулов, сравнивает внутренние выходы с имеющимися (предсказуемыми) результатами и определяет ошибку в работе схемы или ее отсутствие.

*PCI-шина* - это стандарт на местную шину, который поддерживает

сверхширокополосные приложения, такие как мультимедиа и видео, сети, хранение данных и режим “plug-and-play”. Так как стандарт PCI очень быстро стал популярным, то многие разработчики ПЛИС предоставляют возможность совместимости приборов с PCI.

Другой очень важной характеристикой ПЛИС является возможность реализации ИМС в разнообразных корпусах. Наиболее распространены DIP (dual in-line packages) и PGA (pin-grid arrays) корпуса. Они удовлетворяют многих пользователей, но в последние годы особой популярностью стали пользоваться корпуса на основе “surface-mount”, такие как SOIC (small outline IC), PLCC (plastic leadless chip carrier), LCC (J-leaded chip carriers), PQFP (plastic-quad flatpack), CQFP (ceramic-quad flatpack), BGA (ball-grid array) с числом выводов от 20 до 712 ножек.

### **Темы для самоконтроля**

1. Обобщенная архитектура CPLD.
2. Разновидности программируемых межсоединений.
3. Характеристики логических блоков CPLD.
4. Характеристики матриц произведений.
5. Особенности схем распределений у разработчиков разных фирм.
6. Характеристики макроячеек CPLD.
7. Временные параметры CPLD.
8. Дополнительные возможности CPLD (программирование и перепрограммирование в системе, пограничное сканирование, согласование с PCI-шиной).
9. Области применения CPLD.