ГЛАВА 3

ИСТОРИЯ РАЗВИТИЯ ПЛИС

Родственные технологии

Чтобы проследить историю развития ПЛИС и выявить причины, по которым они заняли лидирующее положение в этой области, рассмотрим родственные ей технологии (**Рис. 3.1**).

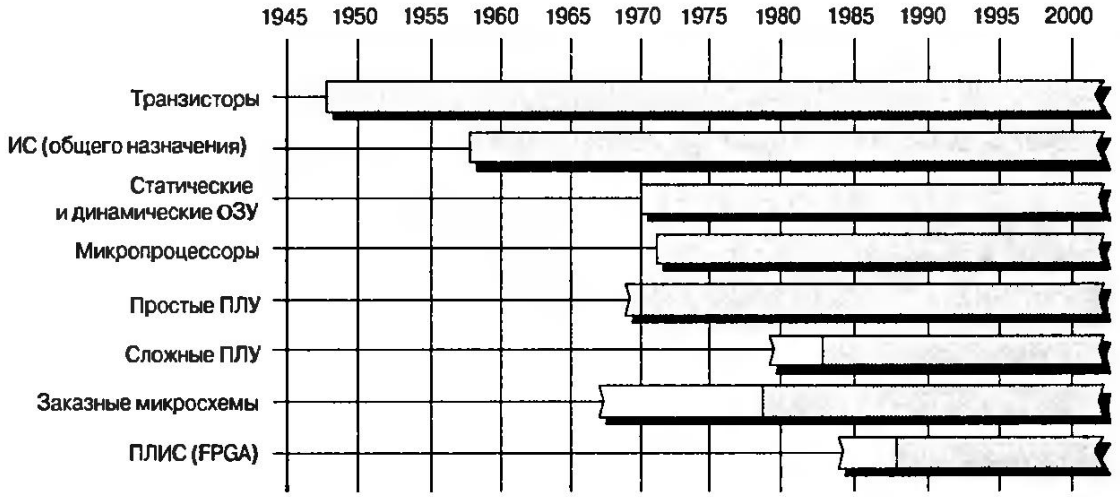


Рис. 3.1. Диаграмма развития технологий (даты указаны приблизительно)

Белые области на полосах диаграммы свидетельствуют о том, что, не смотря на возможность более раннего воплощения в жизнь, эти технологии по той или иной причине в то время не вызвали большого восторга у инженеров. Так, фирма Xilinx еще в 1984 году продемонстрировала миру первую микросхему ПЛИС. Однако эти маленькие штучки не произвели должного впечатления на инженеров и не пользовались особой популярностью вплоть до начала 90-х прошлого века.

Транзисторы

23 декабря 1947 года физики Вильям Шокли (William Shockley), Вальтер Брэттин (Walter Brattain) и Джон Бардин (John Bardeen) (Исследовательский центр компании Bell Laboratories, США), создали *первый транзистор* — точечное устройство на основе германия (символ химического элемента — Ge).

1950-й год стал свидетелем появления более сложного устройства — *биполярного транзистора*. Производство этих транзисторов было достаточно простым и дешевым и, кроме того, налицо было неоспоримое преимущество — более высокая надежность. Однако к концу 50-х германий, несмотря на определенные преимущества в части электрических характеристик, уступил место кремнию (символ химического элемента — Si). Низкая стоимость кремния и легкость его обработки привели к тому, что транзисторы стали производить чаще из кремния, чем из германия.

Если соединить биполярные транзисторы определенным образом, получим базовый логический элемент или логический вентиль *транзисторно-транзисторной логики* (ТТЛ). Соединив те же транзисторы по другому, получим логический вентиль *эмиттерно-связанной логики* (ЭСЛ). Логические вентили ТТЛ обладают высокой скоростью переключения, но при этом потребляют относительно большой ток. Логические вентили ЭСЛ потенциально быстрее ТТЛ-вентилей, но потребляют еще больший ток.

В 1962 году Стивен Хофстайн (Steven Hofstein) и Фридрих Хейман (Fredric Heiman) из Исследовательской лаборатории RCA, Прингстон, шт. Нью-Джерси разработали новое семейство устройств — *металл-оксид-полупроводниковые полевые транзисторы (metal-oxide semiconductor field-effect transistors — MOSFET)*. Для краткости их часто называют просто *полевыми транзисторами* или *МОП-транзисторами*. В целом полевые транзисторы несколько медленнее, чем их биполярные родственники, но они дешевле, меньшего размера и потребляют существенно меньше энергии.

Существуют два основных типа полевых транзисторов, а именно *n*-типа, или *n*-МОП, и *p*-типа, или *p*-МОП. Логические элементы, построенные из соединенных вместе *n*-МОП и *p*-МОП транзисторов, называются *комплементарным металл-оксид полупроводниковым элементом* (КМОП). КМОП-вентили, если и были несколько медленнее своих ТТЛ родственников, в настоящее время практически догнали их по скоростным характеристикам. Однако преимущество КМОП вентилей состоит в том, что в статическом (некоммутируемом) режиме их мощность потребления чрезвычайно мала.

Интегральные микросхемы

Первые транзисторы поставлялись в виде отдельных компонентов, заключенных в небольшие металлические корпуса. Спустя некоторое время возникла идея выполнить всю электрическую схему целиком на одном куске полупроводника. Впервые эта идея была «озвучена» в статье, опубликованной в 1952 году, автором который был британский эксперт в области радиолокации Дж. Даммер (G.W.A. Dummer). Обсуждение этой идеи и возможности ее реализации продолжались до тех пор, пока летом 1958 года Джеку Килби (Jack Kilby) из компании Texas Instruments не удалось создать генератор с фазовым сдвигом, состоящего из пяти компонентов, расположенных на одном куске полупроводника.

Почти в это же время, пока Д. Килби работал над своим генератором, два основателя компании Fairchild Semiconductor, швейцарский физик Джин Хорни (Jean Hoerni) и американский физик Роберт Нойс (Robert Noyce), разработали основы технологии оптической литографии, которая в настоящее время используется для создания транзисторов, изолирующих слоёв и межэлементных соединений в современных микросхемах.

В середине 60-х Texas Instruments представила большое семейство базовых микросхем 54-й и 74-й серий, которые предназначались соответственно для военных и коммерческих целей. Эти устройства-«конфетки», размером 3/4 дюйма (1.89 см) в длину и 3/8 дюйма (0.95 см) в ширину, имели 14 или 16 выводов и содержали небольшой набор простой логики (для особо педантичных читателей следует заметить, что некоторые из них были длиннее, шире и имели большее количество выводов). Например, микросхема 7400 содержала четыре элемента 2-входового И-НЕ, микросхема 7402 —

четыре элемента 2-входового ИЛИ-НЕ, микросхема 7404 состояла их шести инвертирующих (НЕ) логических элементов.

54-я и 74-я серии микросхем компании Texas Instruments изготавливались по технологии ТТЛ. Но уже в 1968 году компания RCA анонсировала практически эквивалентный набор КМОП микросхем, названный серией 4000.

Статическое и динамическое ОЗУ, микропроцессоры

Конец 60-х и начало 70-х оказались урожайными на разработки в области интегральных микросхем. Так, в 1970 году компания Intel анонсирует первую 1024-битную микросхему динамического ОЗУ (модель 1103), а компания Fairchild представляет первую 256-битную микросхему статического ОЗУ (модель 4100).

Год спустя, т. е. в 1971, Intel демонстрирует первый в мире *микропроцессор* (*МП*), так называемую модель 4004, создателями которой были Марсиан Хофф, известный как Тэд (Marcian «Ted» Hoff), Стэн Мазор (Stan Mazor) и Федерико Фэггин (Federico Faggin). Микропроцессор 4004 представлял собой «компьютер-на-кристалле», содержал всего лишь около 2300 транзисторов и мог выполнять 60000 операций в секунду.

Несмотря на то что 4004-й часто упоминается как первый микропроцессор, на это звание претендовали многие. Так, в феврале 1968 года компания International Research Corporation представила разработку устройства, относящуюся к классу «компьютер-на-кристалле». В 1970-м, за год до того, как микропроцессор 4004 увидел свет, некий Гилберт Хайатт (Gilbert Hyatt) подал заявку на патент под названием «Принцип построения однокристального компьютера» (споры относительно этого патента продолжаются и по сей день). Но что никем и никогда не оспаривалось и не оспаривается, так это то, что микропроцессор 4004 был первым физически исполненным, серийно выпускаемым микропроцессором, который действительно был способен выполнять некие полезные функции.

Микропроцессоры и микросхемы статического ОЗУ заслуживают особого внимания, поскольку большинство современных ПЛИС основаны на ячейках статического ОЗУ, а некоторые их современные высокотехнологичные представители содержат встроенные микропроцессорные ядра (*см.* гл. 4).

Простые и сложные ПЛУ

Первые программируемые интегральные микросхемы можно отнести к *программируемым логическим устройствам* (ПЛУ). Они появились в 70-м в виде микросхем ППЗУ и были довольно примитивными, но из вежливости к их авторам все предпочитали об этом помалкивать. Только к концу 70-х появились более сложные версии этих устройств. Чтобы отличить их от менее сложных предшественников, которые все еще находят применение в наши дни, эти устройства стали называть *сложными ПЛУ* (CPLD — *complex programmable logic devices*). Естественно, что впоследствии менее сложные устройства стали называться *простыми ПЛУ* (SPLD — *simple programmable logic devices*).

В результате возникла некоторая путаница, причем одни воспринимают понятия *простые и сложные ПЛУ* как синонимы, в то время как другие считают, что термин *ПЛУ* включает в себя как простые, так и сложные устройства.

1500 г. *Италия. Леонардо да Винчи нарисовал детали простого механического калькулятора.*

Ситуация осложняется еще и тем, что инженеры часто используют один акроним (аббревиатуру, сокращение) для обозначения разных понятий, либо разные акронимы для обозначения одного и того же понятия (самые стойкие могут впасть в истерику, слушая гогот инженеров, которые потчуют друг друга изысками во время своих бесед). Например, в случае простых ПЛУ существует множество производных архитектур, многие из которых обозначаются аббревиатурами из различных комбинаций тех же трех или четырех букв (**Рис. 3.2**).

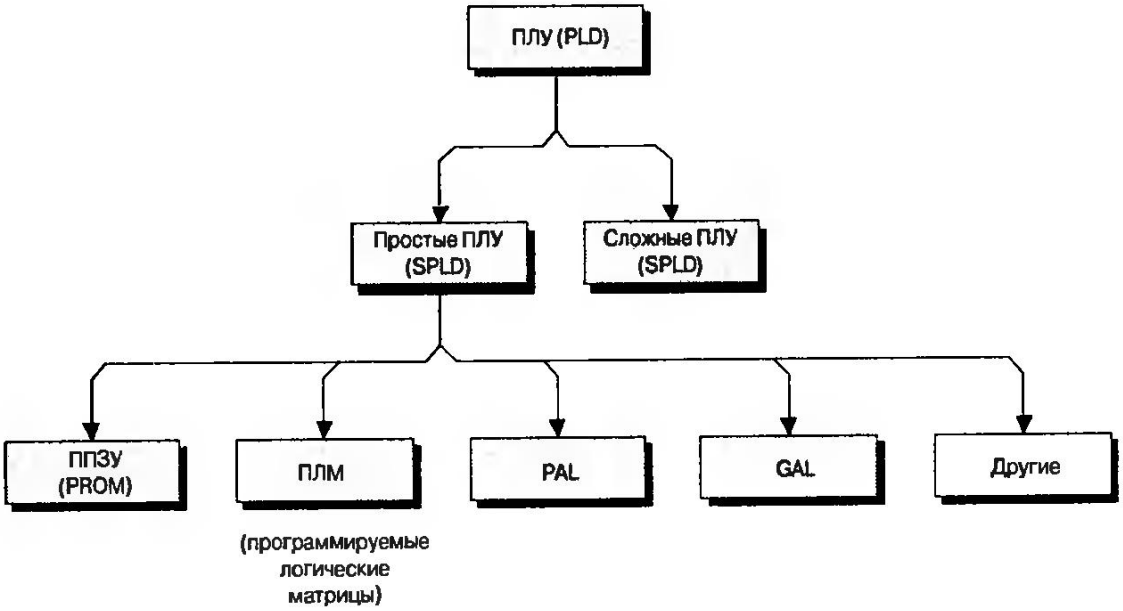


Рис. 3.2. Изобилие архитектур ПЛУ

Конечно, существуют и стираемые ПЛУ, и электрически стираемые ПЛУ, и Flash-версии различных устройств, например СППЗУ и ЭСППЗУ (*см.* гл. 2). На **Рис. 3.2** они отсутствуют с целью его упрощения.

ППЗУ

Первыми представителями простых ПЛУ были ППЗУ-микросхемы, которые появились в 70-м. Чтобы понять какие чудеса происходят внутри этих устройств, лучше всего представить их состоящими из фиксированного массива логических функций И, подсоединенного к программируемому массиву логических функций ИЛИ.

Для примера рассмотрим работу ППЗУ с тремя входами и тремя выходами (**Рис. 3.3**).

В качестве программируемых связей в массиве элементов ИЛИ могут применяться плавкие перемычки, либо СППЗУ-транзисторы или ячейки ЭСППЗУ, соответственно для микросхем СППЗУ или ЭСППЗУ. **Рис. 3.3** дает лишь общее представление о принципе действия рассматриваемого устройства и не отображает реальную диаграмму соединений. В действительности каждая функция И предопределенного массива имеет три входа, которые подсоединены к истинным или инвертированным входам *a*, *b* или *c* устройства. Аналогично каждая функция ИЛИ программируемого массива имеет восемь входов, которые подсоединены к выходам массива функций И.

Как уже отмечалось, микросхемы ППЗУ изначально предназначались для хранения программных инструкций и значений констант, т. е. для выполнения функций компьютерной памяти. Однако разработчики используют их также для реализации простых логических функций, таких как таблицы соответствия или конечные автоматы. В действи-

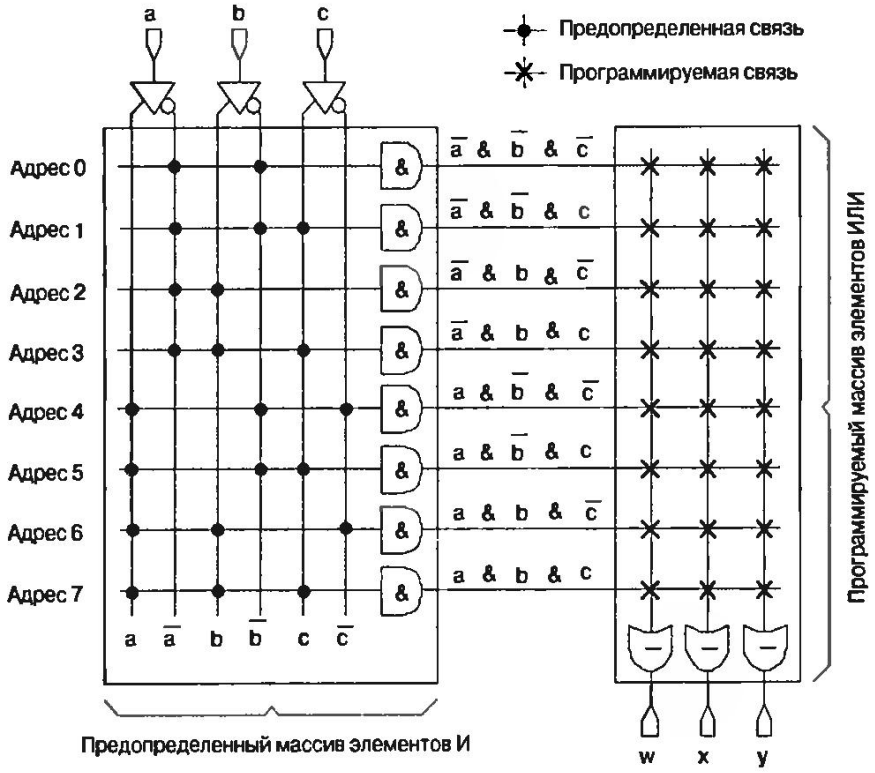
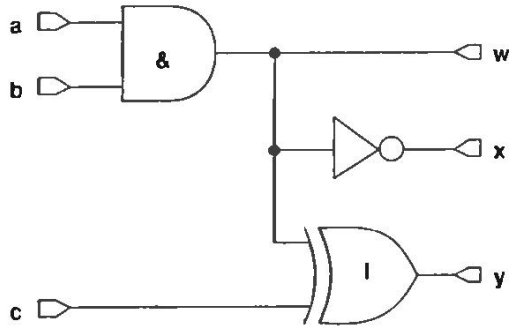


Рис. 3.3. Незапрограммированная микросхема ППЗУ

тельности микросхемы **ППЗУ** могут использоваться для реализации любого блока комбинационной логики[1], или комбинационного устройства, при условии, что он имеет небольшое количество входов и выходов. Например, простое **ППЗУ** с тремя входами и тремя выходами, показанное на **Рис. 3.3**, может использоваться для синтеза любой комбинационной функции с не более чем тремя входными и тремя выходными параметрами. Чтобы понять, как оно работает, рассмотрим небольшой логический блок, показанный на **Рис. 3.4.** Следует иметь в виду, что эта схема имеет смысл только для данного примера.



a	b	c	w	x	y
0	0	0	0	1	0
0	0	1	0	1	1
0	1	0	0	1	0
0	1	1	0	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	1
1	1	1	1	0	0

Рис. 3.4. Небольшой блок комбинационной логики

Логический блок (**Рис. 3.4**) может быть заменен микросхемой ППЗУ с тремя входами и тремя выходами. Для этого надо всего лишь запрограммировать соответствующие связи в массиве логических элементов ИЛИ (**Рис. 3.5**).

В выражениях на этом рисунке использованы следующие обозначения: «&» — операция И (AND), «|» — операция ИЛИ (OR), «^» — операция ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR), « ¯ » — операция НЕ (NOT).

---

[1] Комбинационную логику часто называют *комбинаторной логикой*.

На ранних этапах развития ПЛУ имел широкое хождение другой синтаксис: вместо символа « ¯ » использовался символ «!», так как с его помощью можно было легко и сжато записать логическое выражение в текстовый файл, используя стандартные символы клавиатуры.

Рассмотренная выше микросхема ППЗУ является очень простой. В действительности микросхемы ППЗУ имеют значительно больше входов и выходов и могут использоваться для реализации больших блоков комбинационной логики. С середины 60-х и до середины 80-х (или даже позже) комбинационная логика в общем случае реализовывалась с помощью микросхем-«конфеток», таких как 74-ая серия компании Texas Instruments.

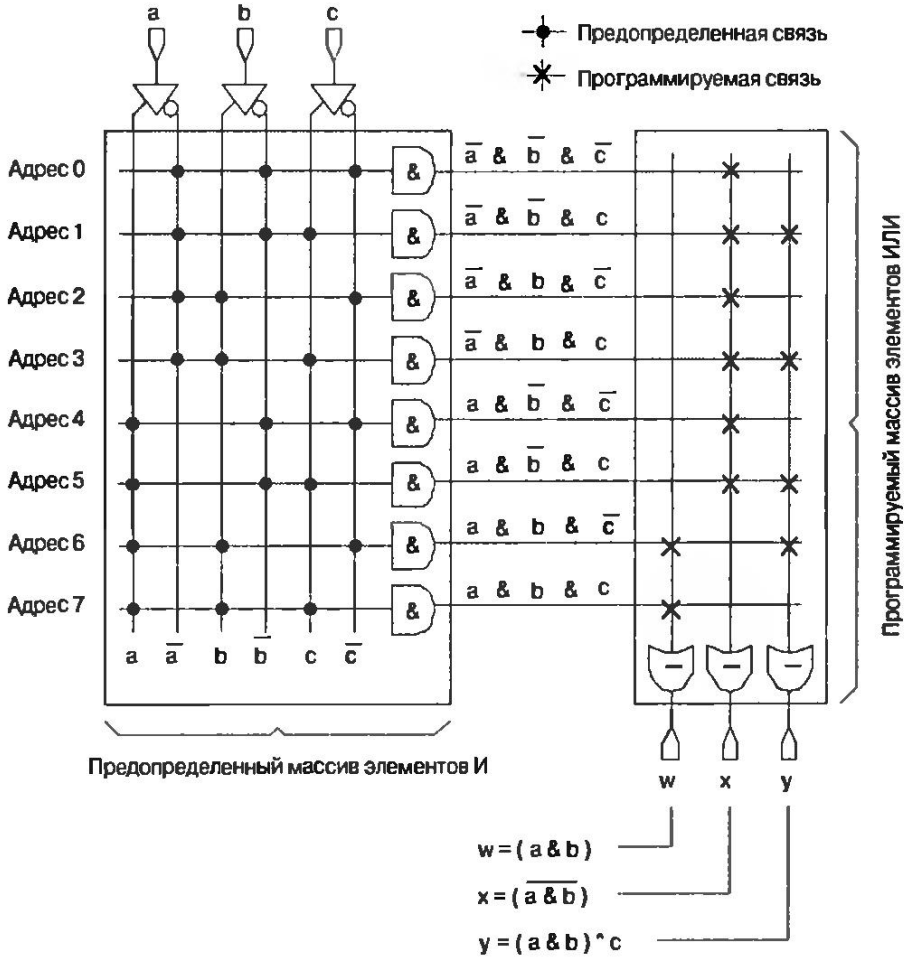


Рис. 3.5. Запрограммированное ППЗУ

Тот факт, что довольно большое число таких микросхем можно заменить одной микросхемой ППЗУ, означает, что монтажную плату можно сделать меньше, проще, дешевле и менее подверженной сбоям (каждое паяное соединение на монтажной плате — потенциально сбойный узел). Кроме того, если в этой части системы обнаружится логическая ошибка, например разработчик ошибочно использовал функцию И вместо И-НЕ, она может быть легко исправлена путем прошивки новой ППЗУ-микросхемы или посредством очистки и перепрограммирования СППЗУ либо ЭСППЗУ-микросхем. Этот способ предпочтительнее, чем обнаружение ошибок на печатной плате, выполненной на основе отдельных микросхем. В подобной ситуации на печатную плату приходится добавлять новые микросхемы, перерезать скальпелем существующие дорожки и с помощью дополнительных проводов вручную подсоединять новые микросхемы к требуемому месту на плате.

В логических выражениях оператор И («&») называется логическим умножением или произведением, а оператор ИЛИ («|») логическим сложением или суммой. Тем не менее, когда речь идет о логическом выражении вида

$$y = (a \,\&\, \bar{b} \,\&\, c) \,|\, (\bar{a} \,\&\, b \,\&\, c) \,|\, (a \,\&\, \bar{b} \,\&\, \bar{c}) \,|\, (a \,\&\, \bar{b} \,\&\, c),$$

понятие *литерал* означает либо истинную, либо инвертированную переменную (a, $\bar{a}$, b, $\bar{b}$ и т. д.), а группа литералов, связанная оператором «&», называется *произведением*. Таким образом, произведение (a & $\bar{b}$ & c) состоит из трех литералов, а именно a, $\bar{b}$ и c, и приведенное выше уравнение будет *суммой произведений*.

Суть заключается в том, что когда эти выражения используются для реализации комбинационной логики (**Рис. 3.4** и **3.5**), ППЗУ удобно использовать для выражений с большим количеством произведений, но небольшим количеством входов, так как все входные комбинации жестко зашиты в матрице **И** и всегда декодируются.

1600 г. *Джон Непер (John Napier) разработал простую таблицу умножения, которую назвали «Кости Непера» (Napier's Bones).*

1614 г. *Джон Непер (John Napier) изобрёл логарифмы.*

Программируемые логические матрицы (ПЛМ)

Следующая ступень развития ПЛУ — решение проблем, связанных с ограничениями, свойственными архитектуре ППЗУ.

Первые *программируемые логические матрицы* (ПЛМ) появились примерно в 1975 году. Большинство пользователей использовали их как простые ПЛУ, так как оба массива, т. е. и массив функций И, и массив функций ИЛИ, были программируемыми.

Если взять простую ПЛМ с тремя входами и тремя выходами в незапрограммированном состоянии (**Рис. 3.6**), в отличие от ППЗУ количество функций И в одноименном массиве не зависит от количества входов матрицы. При этом дополнительные функции И могут быть сформированы путем простого добавления в массиве строк.

Аналогично количество функций ИЛИ в одноименном массиве не зависит от количества входов матрицы и от размера массива функций И. Дополнительные функции ИЛИ могут быть сформированы путем простого добавления в этом массиве столбцов.

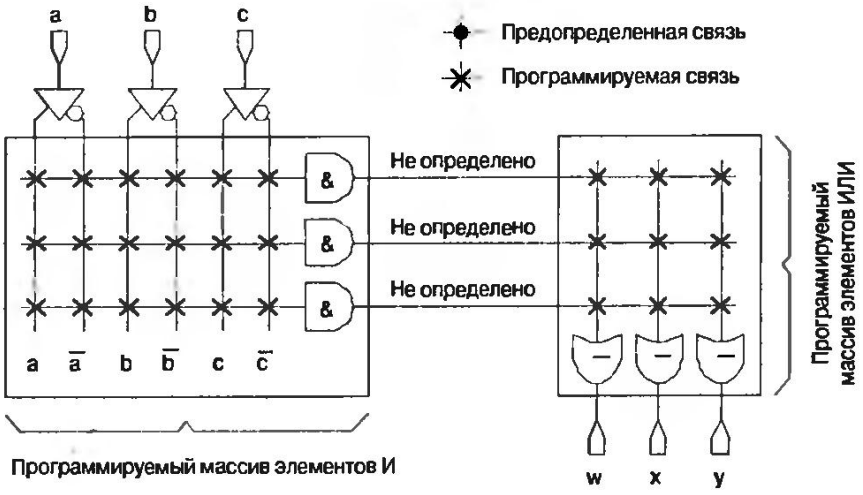


Рис. 3.6. Незапрограммированная ПЛМ

Предположим, что ПЛМ должна реализовать следующие три уравнения:

$$w = (a \,\&\, c) \,|\, (\bar{b} \,\&\, \bar{c}),\ x = (a \,\&\, b \,\&\, c) \,|\, (\bar{b} \,\&\, \bar{c}) \text{ и } y = (a \,\&\, b \,\&\, c).$$

Решить эту задачу можно путем программирования соответствующих связей, как показано на **Рис. 3.7.**

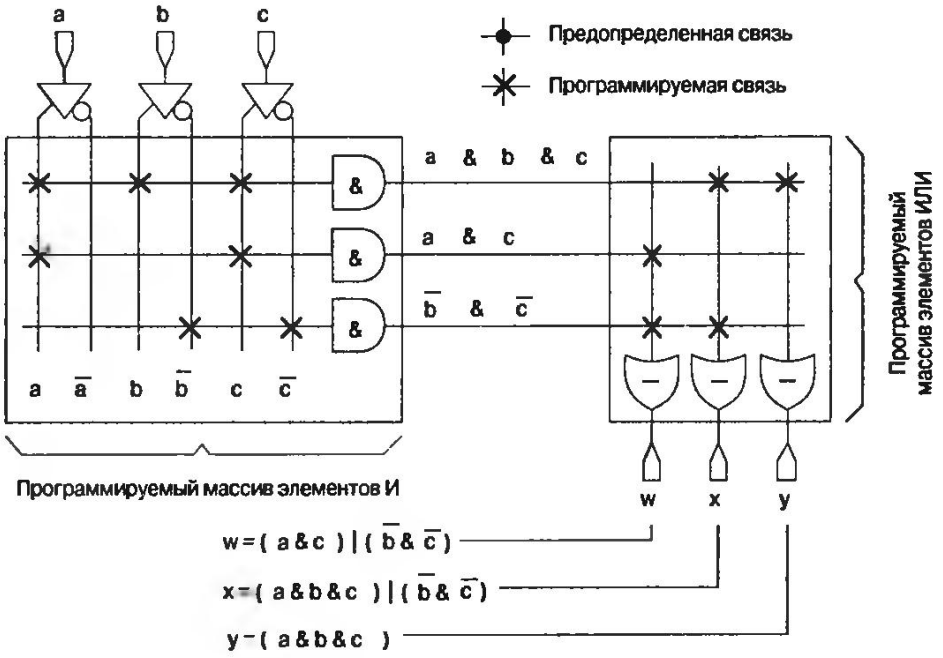


Рис. 3.7. Запрограммированная ПЛМ

Судьба распорядилась таким образом, что ПЛМ никогда не занимали существенной доли рынка, но при этом время от времени некоторым поставщикам удавалось поразить рынок какими-нибудь изюминками своих устройств. Так, в ПЛМ нет жесткой необходимости подключать массив функций ИЛИ к выходу массива функций И. Поэтому некоторые альтернативные конфигурации, например массив функций И, подключенный к входу массива ИЛИ-НЕ, иногда показывал неплохие результаты. Однако, несмотря на теоретическую возможность реализации на практике, функции ИЛИ-И, И-НЕ-ИЛИ, И-НЕ-ИЛИ-НЕ относительно редко встречались или просто не существовали. Одной из причин, по которой ПЛМ продолжают работать с архитектурой И-ИЛИ[1] (и И-ИЛИ-НЕ), стало то, что выражение *сумма-произведений* наиболее часто используется в логических уравнениях. Другие типы уравнений, такие как *произведение-сумм*, приводятся к ним с помощью стандартных алгебраических методов. Такое преобразование обычно выполняется с помощью программ, которые, условно говоря, могут сделать эту работу и со связанными руками.

Микросхемы ПЛМ рекламировались как весьма полезные устройства для больших схем, логические уравнения которых характеризовались большим количеством одинаковых произведений. Например, произведение вида ($\bar{b}$ & $\bar{c}$) используется дважды: для вычисления выхода *w* и для вычисления входа *x* (**Рис. 3.7**). Это свойство называется *распределением произведений*.

Вместе с тем для прохождения через программируемые связи сигналам требуется намного больше времени, чем при прохождении через их предопределенные аналоги. Поэтому ПЛМ работает медленнее чем ППЗУ, так как оба массива и функций И, и функций ИЛИ являются программируемыми.

---

[1] Действительно, как-то один разработчик сказал мне, что его команда создала архитектуру НЕ-ИЛИ-НЕ-ИЛИ-НЕ-НЕ, которая имеет незначительное преимущество в скорости, но своих покупателей они заверили в том, что это архитектура И-ИЛИ, так как «именно этого желали покупатели». И в наше время, если производители говорят, что они что-то сделали, это отнюдь не значит, что они сделали именно то, о чем говорят.

Программируемые матрицы PAL и GAL

Для того чтобы решить проблему скорости, свойственной программируемым логическим матрицам, в конце 1970-х появился новый класс устройств, называемый *программируемый массив логики (ПМЛ* или *PAL — Programmable Array Logic)*. На понятийном уровне устройства PAL почти полностью противоположны микросхемам ППЗУ, так как состоят из программируемого массива логических функций И и предопределенного массива функций ИЛИ. Устройства *GAL (Generic Array Logic — изменяемый массив логики)*, разработанные в 1983 году компанией Lattice Semiconductor Corporation, представляют собой более сложные электрически стираемые КМОП-разновидности идеологии PAL.

PAL — зарегистрированная торговая марка компании Monolithic Memories, Inc.

В качестве примера рассмотрим простое PAL-устройство с тремя входами и тремя выходами (**Рис. 3.8**). Преимуществом микросхем PAL (по сравнению с программируемыми логическими матрицами) является более высокая скорость, так как из двух массивов у них только один программируемый.

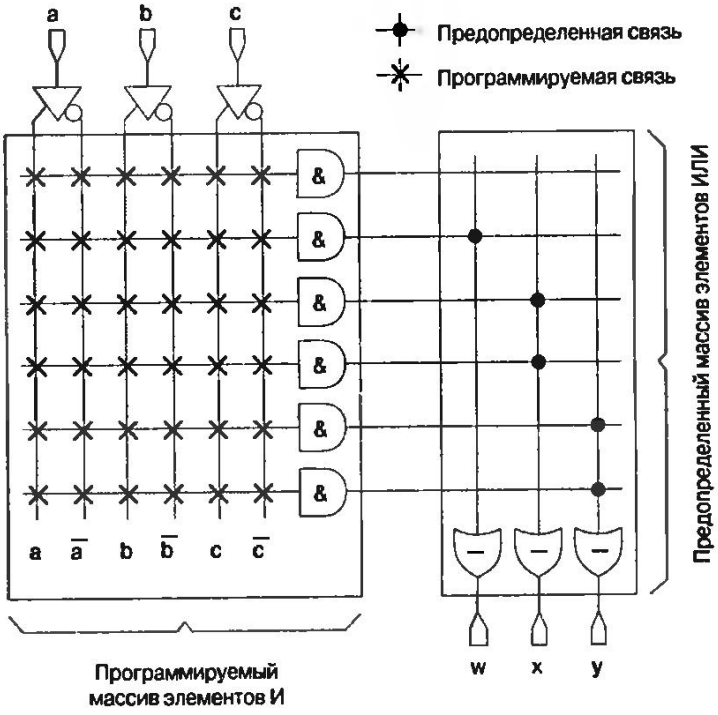


Рис. 3.8. Незапрограммированное устройство PAL

Вместе с тем программируемый массив логики более ограничен по функциональности, так как он позволял складывать только ограниченное количество произведений. Но инженеры люди изобретательные: у них всегда имеется в запасе много разных уловок, помогающих обойти все эти трудности.

Дополнительные программируемые опции

Рассмотренные выше в качестве примеров ПЛМ и PAL на самом деле очень большие, с множеством входов, выходов и с внутренними сигналами. В них могут быть предусмотрены различные дополнительные программируемые опции, такие как возможность инвертировать выходы, либо иметь выходы с тремя состояниями, либо и то, и другое. Кроме того, некоторые из них поддерживают регистровые выходы, т. е. выходы с защелкой, аналогично программируемым мультиплексо-

1621 г. *Вильям Отред (William Oughtred) на основе логарифмов Джона Непера разработал логарифмическую линейку.*

рам, которые позволяют пользователю устанавливать поочередно для каждого вывода, какую версию выхода использовать — регистровую или нерегистровую. Некоторые устройства позволяют конфигурировать определенные выводы в качестве либо выходов, либо дополнительных входов. И это далеко не все опции, и список можно продолжать и продолжать.

Проблема заключается в том, что различные устройства могут обеспечивать разные наборы опций, что затрудняет выбор оптимального устройства для конкретного приложения. В подобных случаях инженеры либо ограничивают себя в выборе устройств и подгоняют свою конструкцию под эти устройства, либо используют компьютерные программы, с помощью которых выбирают устройство максимально удовлетворяющее их требованиям.

Сложные ПЛУ

Типичное для электроники явление — вечный поиск способа, как сделать устройство больше, в смысле функциональности, меньше, в смысле физических размеров, быстрее, производительнее и дешевле. Я бы сказал, довольно скромненькие требования. Не правда ли? В связи с этим в конце 70-х — начале 80-х появились более сложные программируемые логические устройства, так называемые *сложные ПЛУ (CPLD — complex PLD)*.

Лидерами были изобретатели оригинальных устройств PAL — разработчики компании Monolithic Memories (MMI), которые представили компонент под названием Мега-PAL (Mega-PAL). Это устройство с 84 выводами по существу включало в себя 4 PAL-устройства и соединения между ними. К сожалению, Мега-PAL потребляла непропорциональное много энергии, и давала небольшое преимущество по сравнению с использованием четырех отдельных устройств.

Существенный прорыв пришелся на 1984 год, когда только что образованная компания Altera представила сложное ПЛУ, основанное на сочетании КМОП- и СППЗУ-технологий. Использование КМОП позволило компании Altera достичь невероятной функциональной плотности и сложности при сравнительно небольшом потреблении энергии. Основой для программирования этих устройств были ячейки СППЗУ, и именно это сделало их идеальными для использования при разработке и создании прототипов оборудования.

Компания Altera заслужила свою славу не только благодаря комбинации технологий КМОП и СППЗУ. При наращивании архитектуры простых ПЛУ до уровня больших устройств, подобных Мега-PAL, инженеры полагали, что центральный коммутационный массив, известный так же как *программируемая коммутационная матрица*, и связывающий индивидуальные блоки простых ПЛУ, должен на 100% соединяться со всем входам и выходам блоков. В результате это приводило к существенному снижению скорости, повышению рассеиваемой мощности и увеличению стоимости компонентов.

Понятие «software» (в переводе с англ. — *программное обеспечение*) также «изобрел» Джон Таки (John Wilder Tukey).

Altera удалось совершить технологический прорыв, используя центральный коммутационный массив с количеством соединений с входами/выходами блоков менее 100% (подробнее на **Рис. 3.10**). Это усложнило *программное обеспечение* для проектирования ПЛУ, но скорость, потребляемая мощность и стоимость этих устройств были вполне приемлемыми.

Несмотря на то что каждый производитель сложных ПЛУ реализовывал свои уникальные технологии, в общем случае устройство состо-

яло из нескольких блоков простых ПЛУ, обычно PAL, объединенных общей программируемой коммутационной матрицей (**Рис. 3.9**). Помимо отдельных блоков простых ПЛУ можно было также запрограммировать соединения между ними с помощью программируемой коммутационной матрицы.

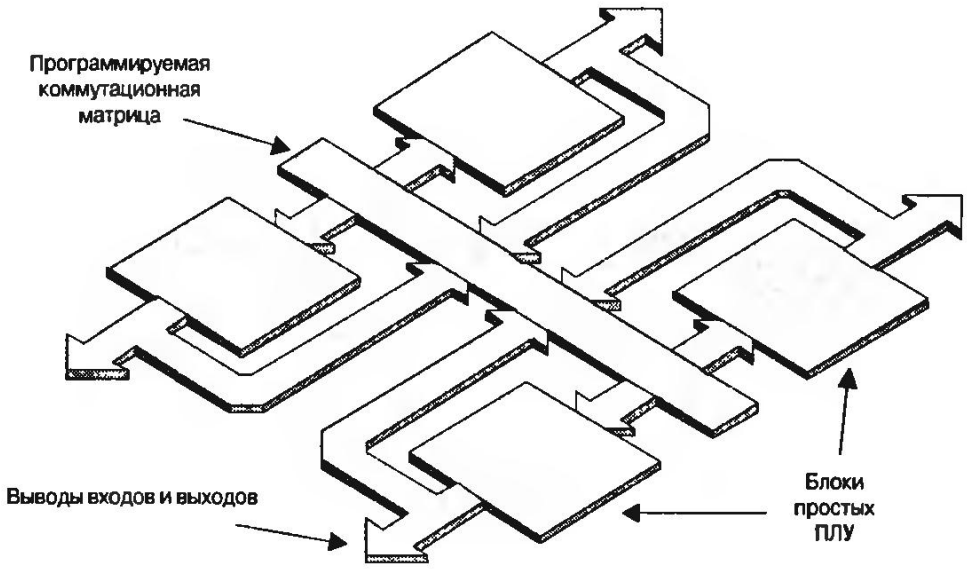


Рис. 3.9. Общая структура сложного ПЛУ

На **Рис. 3.9** не показаны различные дополнительные компоненты, и он дает лишь поверхностное представление о работе сложного ПЛУ. В действительности все структуры устройства сформированы на одном куске кремния. Например, программируемая коммутационная матрица может содержать большое количество проводников, скажем 100. Но это больше, чем может быть подключено к блоку простого ПЛУ, который способен работать только с ограниченным количеством сигналов, скажем 30. Блоки простых ПЛУ связаны с коммутационной матрицей своего рода программируемыми мультиплексорами (**Рис. 3.10**).

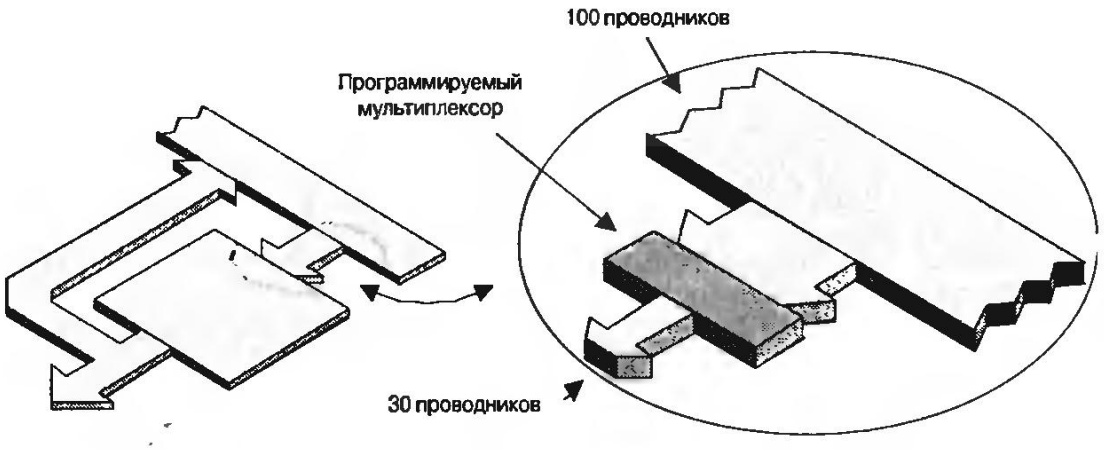


Рис. 3.10. Использование программируемого мультиплексора

В зависимости от производителя и от типа устройства программируемые переключатели сложных ПЛУ могут быть выполнены на ячейках памяти типа СППЗУ, ЭСППЗУ, Flash или на статическом ОЗУ. При использовании статического ОЗУ появляется возможность увеличить универсальность этой памяти, используя её в качестве программируемых переключателей и в качестве фактической оперативной памяти.

Программные средства ABEL, CUPL, PALASM, JEDEC и другие

Средневековье относится к периоду истории от классической античности до итальянского Ренессанса, эпоха Возрождения. Различные источники определяют дату начала средневековья с точностью до нескольких сотен лет.

Для инженеров первые дни существования микросхем ПЛУ во многом напоминали времена средневековья.

Спецификация на новые устройства обычно представляла собой принципиальную схему или схему конечного автомата. Эти схемы создавались с помощью бумаги и карандаша, так как в то время не существовало автоматизированных систем описания принципиальных схем в том виде, в котором они известны нам.

Однажды конструкция устройства, представленная в графическом виде, была вручную преобразована в табличный эквивалент и затем введена в текстовый файл. Этот текстовый файл, кроме всего прочего, определял, какие плавкие перемычки должны быть подвержены пережиганию или какие наращиваемые перемычки должны быть созданы. В те незапамятные дни текстовый файл напрямую вводился в специальное устройство, называемое *программатором*, который использовался для программирования микросхем. Однако со временем файл начали создавать на компьютере, с помощью которого производилась загрузка и управление программатором (**Рис. 3.11**).

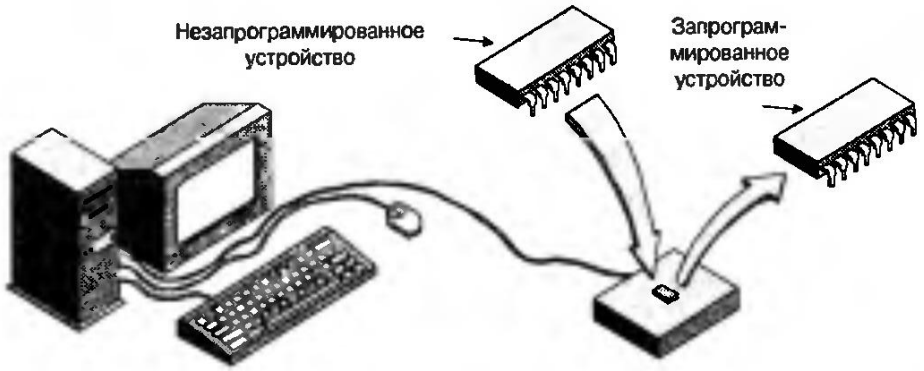


Рис. 3.11. Программирование микросхем ПЛУ

При создании файла от инженера требовались знание внутренних связей устройства и формата файла, записываемого в программатор. Исключительно ради развлечения каждый производитель ПЛУ развивал собственные форматы файлов, которые обычно работали со своими типами устройств. Всем заинтересованным было очевидно, что такой порядок проектирования был трудоемким, чреват ошибками и, к тому же, затруднял их поиск и исправление.

В 1980 году комиссия Объединенного инженерного совета по электронным устройствам (JEDEC), подразделение Ассоциации электронной промышленности США, предложила стандарт форматов текстовых файлов для программирования устройств ПЛУ. Это случилось незадолго до того, как программисты электронных устройств уже были готовы принять этот формат.

Примерно в это же время Джон Беркнер (John Birkner), человек который разработал первые устройства PAL и стоял во главе их дальнейшего развития, создал *PAL Ассемблер* (*PALASM*). PAL Ассемблер сочетает в себе язык описания аппаратных средств (*HDL — Hardware Description Language*) и прикладное программное обеспечение. Как язык описания аппаратных средств PAL Ассемблер позволяет инженерам-разработчикам точно описать функции принципиальной схемы в

текстовом файле с исходными кодами. Файл состоял из булевых уравнений, записанных в формате «сумма-произведений». Как прикладная программа, которая на языке программирования Фортран[1] занимала всего лишь 6 страниц (сегодня мы назвали бы ее САПР электронных устройств) PALASM читал текстовый файл с исходными кодами и автоматически генерировал текстовый программный файл, пригодный для программирования устройства.

Появление PALASM, несомненно, было эпохальным событием для того времени, но его первоначальные версии поддерживали только устройства компании MMI (Monolithic Memories Inc), и не поддерживали какие-либо виды минимизации или оптимизации. Для решения этих проблем в 1983 году компания Data I/O представила язык *ABEL (Advanced Boolean Expression Language — расширенный язык логических выражений)*. Примерно в это же время компания Assisted Technology обнародовала пакет CUPL *(Common Universal tool for Programmable Logic — Общие универсальные средства проектирования для программируемой логики)*. Оба продукта сочетали в себе язык HDL и программные приложения. Кроме того, что они поддерживали конструкции конечных автоматов и алгоритмы автоматической минимизации логики, они позволяли работать с многочисленными типами ПЛУ устройств разных производителей.

Кроме PALASM, ABEL и CUPL, которые были, несомненно, лучшими из ранних версий языка HDL, существовало много других версий, таких как *AMAZE (Automated Map and Zap of Equation — автоматическое преобразование и исправление выражений)* компании Signetics. Эти простые языки и сопутствующие им средства проектирования прокладывали путь для высокоуровневых версий языка HDL, таких как Verilog или VHDL, и средств проектирования, таких как логический синтез, которые в настоящее время используются для проектирования современных заказных микросхем и устройств с использованием ПЛИС.

Заказные интегральные схемы — ASIC

Во время написания этой книги существовало четыре основных класса *специализированных заказных интегральных микросхем (ASIC — application specific integrated circuit)*, заслуживающих внимания. В порядке увеличения сложности такими устройствами были *вентильные матрицы, структурированные ASIC, схемы на стандартных элементах* и *полностью заказные микросхемы* (**Рис. 3.12**).

Типы специализированных микросхем можно рассмотреть и в порядке возрастания их сложности, как показано на **Рис. 3.12**. Однако имеет смысл описать их в той последовательности, в какой они появились на свет, т. е. заказные микросхемы одновременно с вентильными матрицами[2], затем микросхемы на стандартных элементах и, наконец, структурированные специализированные интегральные схемы, или структурированные специализированные ИС. Вопрос, какое из уст-

---

[1] Разработанный компанией IBM в 1950 году язык Фортран (FORTRAN — FORmula TRANslator — транслятор формул) стал первым компьютерным языком программирования, уровень которого был выше, чем уровень Ассемблера.

[2] Для вентильной матрицы часто можно встретить другой термин — *базовый матричный кристалл* (БМК). — *Прим. ред.*

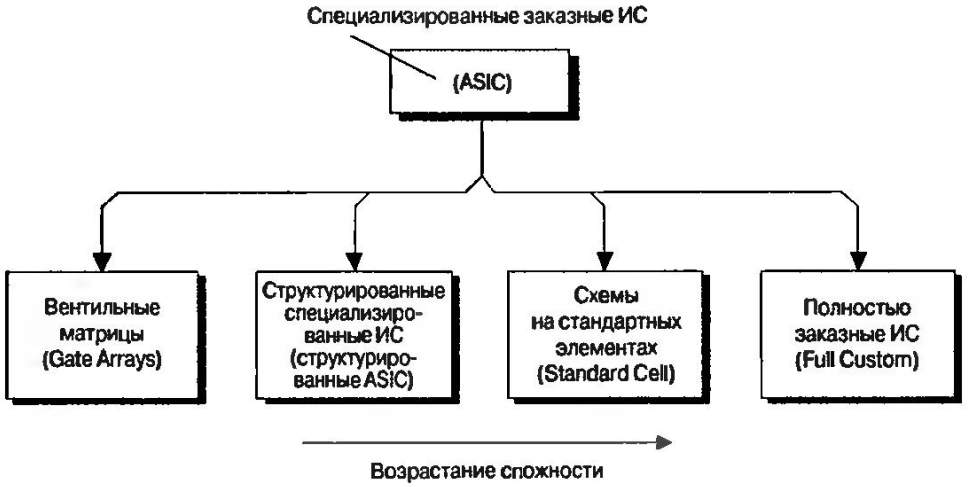


Рис. 3.12. Различные типы специализированных заказных ИС

ройств, структурированная специализированная ИС или вентильная матрица, является более сложным, довольно спорный[1)].

Заказные интегральные микросхемы

На заре появления цифровых микросхем существовало только два типа устройств, кроме, естественно, микросхем памяти. К первому типу относились относительно простые блоки, которые производились такими компаниями, как Texas Instruments или Fairchild, и продавались как стандартные компоненты для всех желающих. Ко второму типу принадлежали заказные специализированные микросхемы, такие как микропроцессоры, которые разрабатывались и производились по заказу.

При создании заказных устройств разработка маски для каждого слоя микросхемы была прерогативой инженеров — разработчиков конечного устройства. Поставщики специализированных микросхем предварительно не создавали никаких компонентов на кремниевом кристалле заводским способом и не поставляли библиотек предопределенных логических вентилей или функций.

Используя соответствующие средства проектирования инженеры задавали размеры отдельных транзисторов и на их основе создавали высокоуровневые функции. Например, при необходимости слегка увеличить скорость логического вентиля они могли изменить размеры транзисторов, используемых для построения вентилей. Средства проектирования, используемые для разработки заказных устройств, часто создавались инженерами фирм-разработчиков.

Разработка заказных устройств является очень сложным и трудоемким процессом, но созданные таким способом микросхемы содержат необходимое количество вентилей с минимальными потерями свободного места на кремниевом кристалле.

[1)] Наиболее въедливые читатели могут предложить альтернативную терминологию. Например, специализированные микросхемы (ASIC) логичнее назвать *полузаказными*, а специализированные схемы (Full Custom) — уже по-настоящему *заказными* микросхемами. Пожалуй, они тоже правы... — *Прим. ред.*

Микроматрица и микромозаика

1642 г. *Блез Паскаль (Blaise Pascal) построил первый механический калькулятор, так называемую* ***арифметическую машину.***

В середине 60-х компания Fairchild Semiconductor представила новое устройство под названием *микроматрица* (*Micromatrix*). Это устройство состояло из небольшого количества, около 100, ни к чему не подключенных транзисторов. Для того чтобы это устройство могло выполнять полезные функции, разработчики вручную вычерчивали на двух пластиковых листах слои металлизации для соединения транзисторов.

На первом листе с помощью зеленого карандаша изображались проводники по оси **Y**, т. е. сверху вниз, с помощью которых изготавливался первый слой металлизации. На втором листе, но уже с помощью красного карандаша, изображались проводники по оси **X**, т. е. слева направо, для изготовления второго слоя металлизации. Кроме того, использовались дополнительные листы для изображения переходных отверстий, соединяющих первый слой металлизации с транзисторами и для переходных отверстий, соединяющих первый и второй слои металлизации.

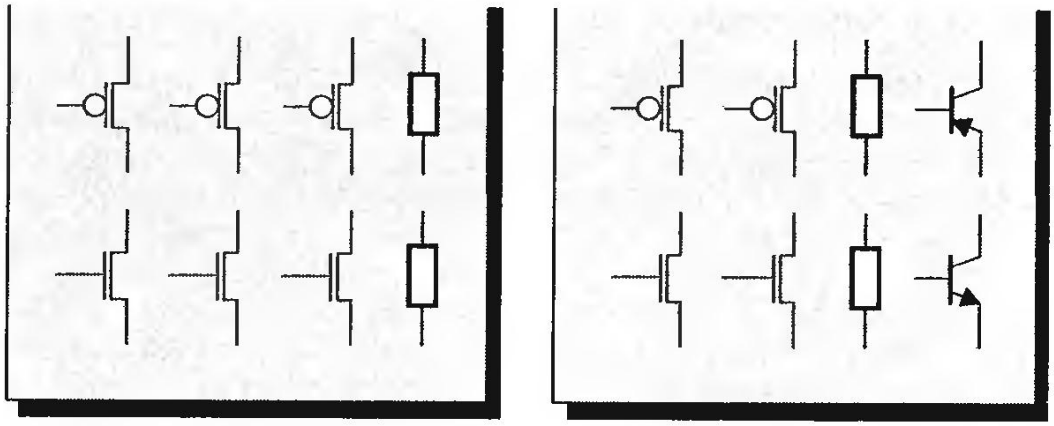
Изготовление устройства таким способом было крайне трудоемким занятием и способствовало возникновению ошибок, но, так или иначе, дорогостоящая и по настоящему трудоемкая работа по созданию транзисторов выполнялась. Другими словами, микроматрица позволяла разработчикам создавать заказные устройства, приемлемые по стоимости (хотя все же дорогие), и срокам изготовления (хотя и не очень быстро).

По прошествии нескольких лет, а точнее в 1967 году, компания Fairchild представила новое устройство, получившее название *микромозаики* (*Micromosaic*). На этот раз устройство содержало несколько сотен отдельных транзисторов, которые могли соединяться вместе, образуя около 150 логических элементов И, ИЛИ и НЕ. Уникальность микромозаики заключалась в том, что разработчики могли задавать функции заказного устройства, используя логические выражения в виде текстового файла, с помощью которого компьютерная программа определяла необходимые межтранзисторные соединения и изготавливала фотошаблон, по которому уже создавалось устройство. Такой подход был настоящей революцией для того времени, и сегодня микромозаика считается предтечей современных *вентильных матриц* как разновидности специализированных заказных микросхем и первым настоящим приложением *средств автоматизированного проектирования* (*САПР*).

Вентильные матрицы

Идея создания матриц логических элементов, или *вентильных матриц*, витала еще в конце 60-х в компаниях IBM, Fujitsu и др. Однако первые устройства использовались только для внутреннего потребления, и только в середине 70-х вентильные матрицы, изготовленные по технологии КМОП, вышли на широкую дорогу и стали доступны широкому кругу потребителей. Первые вентильные матрицы назывались *нескоммутированная логическая матрица (ULA — uncommitted logic array)*. Со временем этот термин перестал употребляться.

Основу вентильных матриц составляет *базисная ячейка*, состоящая из набора несоединённых транзисторов и резисторов. Каждый поставщик заказных микросхем самостоятельно определяет оптимальный набор компонентов, входящих в состав отдельной базисной ячейки (**Рис. 3.13**).

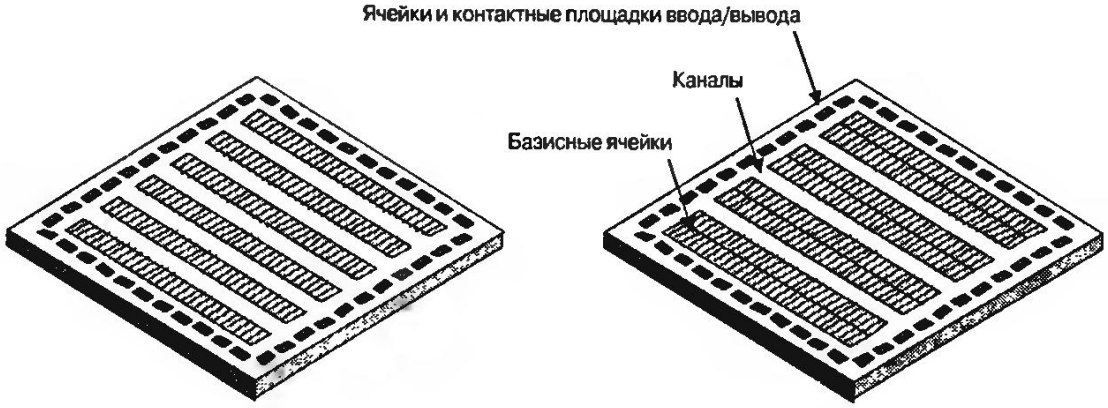


а) Однородная КМОП базисная ячейка

б) Смешанная биполярная — КМОП базисная ячейка

Рис. 3.13. Примеры простых базисных ячеек вентильной матрицы

Поставщики заказных микросхем начинали свою работу с изготовления заводским способом кремниевого кристалла, содержащего массив базисных ячеек. В случае *канальной* матрицы базисные ячейки, как правило, располагались в виде одностолбцовых или двухстолбцовых массивов, причем между столбцами имелись свободные области, так называемые *каналы* (**Рис. 3.14**).



а) Одностолбцовые массивы

б) Двухстолбцовые массивы

Рис. 3.14. Архитектура канальных логических матриц

При отсутствии каналов на матрице ячейки располагались в виде одного большого массива. Поверхность такого устройства была покрыта большим количеством ячеек и не имела выделенных каналов для внутренних соединений. Такие матрицы назывались *бесканальными*.

Разработчики могут использовать определяемый поставщиком микросхем набор логических функций, таких как простые логические вентили, мультиплексоры и регистры. Каждая из этих функций-блоков рассматривается как отдельный элемент или ячейка. Не путать с базисной ячейкой! Набор таких функций, поддерживаемый поставщиком, называется *библиотекой элементов*.

Средства, с помощью которых создаются заказные микросхемы, здесь не рассматриваются. Достаточно сказать, что разработчики в конечном итоге получают таблицу соединений на уровне вентилей, которая описывает, какие вентили будут использоваться, и какие между ними будут соединения. Чтобы установить соответствие логических вентилей и базисных ячеек и определить, как эти ячейки будут связываться между собой, используются специальные программы определения соответствий, программы расстановки, трассировки и другие.

На основе полученных результатов изготовляются фотошаблоны, с помощью которых создаются слои металлизации. Эти слои будут связывать компоненты внутри базисной ячейки, а также ячейки между собой, с входами и выходами устройства.

Вентильные матрицы имеют разумную цену, так как транзисторы и другие компоненты производятся заводским способом, а по заказу изготавливаются только слои металлизации. Большинство таких устройств не использует всех внутренних ресурсов. Кроме того, расположение вентилей жестко определено, а трассировка внутренних соединений далека от оптимальной. Все эти недостатки отрицательно сказались на производительности и потребляемой мощности этих устройств.

1671 г. *Барон Готфрид фон Лейбниц (Gottfried von Leibniz) построил первый механический калькулятор.*

Схемы на стандартных элементах

Решение проблем, свойственных вентильным матрицам стало возможным с появлением в начале 90-х *схем на стандартных элементах*. Эти компоненты имели много общего с вентильными матрицами.

Итак, еще раз: поставщик специализированной заказной (ASIC) микросхемы составляет библиотеку элементов, и этой библиотекой могут пользоваться разработчики. Поставщик также поддерживает библиотеки аппаратных и программных макроопределений, которые включают такие элементы, как процессоры, функции связи, ОЗУ или ПЗУ. Наконец последнее, но не менее важное, разработчики могут использовать ранее созданные функции или приобрести *блоки интеллектуальной собственности*.

Когда команда инженеров конструирует сложную микросхему, дабы не изобретать велосипеда, она может принять решение о покупке уже готового, т. е. кем-то разработанного, проекта одного или нескольких функциональных блоков. Проект таких функциональных блоков называется *интеллектуальной собственностью* или *IP (intellectual property)*. Средства интеллектуальной собственности могут охватывать любые функциональные блоки, в том числе функции связи и микропроцессоры. Наиболее сложные функции, такие как микропроцессоры, могут называться *ядрами*.

Тем или иным способом, в наши дни с помощью сложных систем автоматизированного проектирования, разработчики завершают свою работу созданием таблицы соединений на уровне вентилей. Такая таблица описывает, какие вентили используются и как они соединены между собой.

В отличие от вентильных матриц схемы на стандартных элементах не используют концепцию базисных ячеек, и компоненты на кремниевом кристалле не изготовляются заводским способом. Для индивидуального расположения каждого вентиля в таблице соединений и для оптимальной разводки соединений между ними используются специальные средства автоматизированного проектирования. Полученные результаты используются для создания заказных фотошаблонов для каждого слоя микросхемы.

Концепция схем на стандартных элементах позволяет создавать логические функции, используя минимальное количество транзисторов без какой-либо избыточности компонентов; сами функции могут реализовываться таким образом, чтобы сделать менее трудоемким создание внутренних связей между ними. Именно поэтому схемы на стандартных элементах обеспечивают почти оптимальное использование поверхности кремниевого кристалла по сравнению с вентильными матрицами.

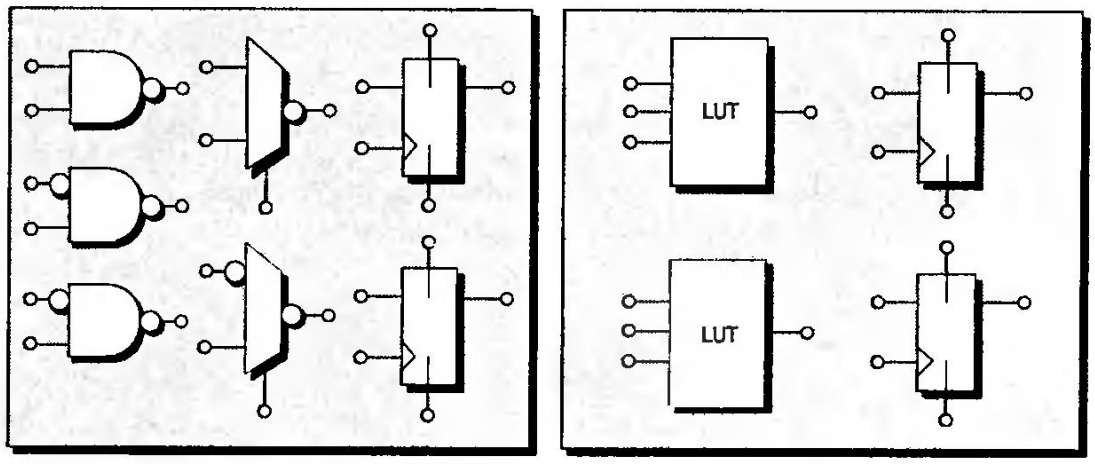
1746 г. *Голландия. В Лейденском университете изобретена* ***лейденская банка.***

Структурированные специализированные микросхемы и структурированные ASIC

С появлением схем на стандартных элементах эксперты предсказывали конец вентильным матрицам. Но, несмотря на то что, согласно пословице «ничто не вечно под луной», они продолжают занимать свою нишу на рынке, а несколько лет назад даже пережили свое второе рождение.

Структурированные специализированные микросхемы, кстати вначале они назывались иначе, появились на свет примерно в начале 90-х и прослонявшись без дела некоторое время, вернулись в преисподнюю, откуда собственно и пришли. Спустя 10 лет, примерно где-то в 2001-2002 году, некоторые производители приступили к изучению способов снижения затрат на проектирование специализированных микросхем и сокращения продолжительности их разработки. Всех пугал тот факт, что новые устройства ассоциировались с традиционными вентильными матрицами. Поэтому, когда примерно в 2003 году появилось название *структурированные специализированные микросхемы (structured ASIC)*, оно сразу пришлось всем по душе.

Конечно же, как обычно, у каждого поставщика собственная архитектура устройства. В связи с этим будут рассмотрены только общие черты этих устройств. Каждое устройство начинается с фундаментального элемента, который одни называют *модуль*, а другие *элемент мозаики*. Такой элемент может содержать изготовленный заводским способом набор общей логики, выполненной в виде логических вентилей, мультиплексоров или таблиц соответствия, одного или нескольких регистров, и, возможно, небольшого локального ОЗУ (**Рис. 3.15**).



а) Модуль содержит вентили, мультиплексоры и триггеры

б) Модуль содержит таблицы соответствия (LUT) и триггеры

Рис. 3.15. Примеры модулей структурированных специализированных ИС

Массив таких элементов изготавливается заводским способом по всей поверхности кристалла. Некоторые альтернативные архитектуры начинаются с *базисной ячейки* или *базисного модуля*, или *базисного элемента мозаики*, или др., в состав которой входят только элементы общей логики в форме изготовленных заводским способом вентилей, мультиплексоров или таблиц соответствия. Массив таких базисных единиц (говорят 4×4, 8×8 или 16×16) в соединении с некоторыми специальными модулями, содержащими регистры, небольшие элементы памяти и другую логику, образует *базовую ячейку* или *базовый модуль*, или *базовый элемент мозаики*, или др. Этот массив изготавливается заводским способом по всей поверхности кристалла.

Заводским способом также изготавливаются, обычно по контуру кристалла, некоторые дополнительные функции, например блоки ОЗУ, тактовые генераторы, логика периферийного сканирования и другие функции (**Рис. 3.16**).

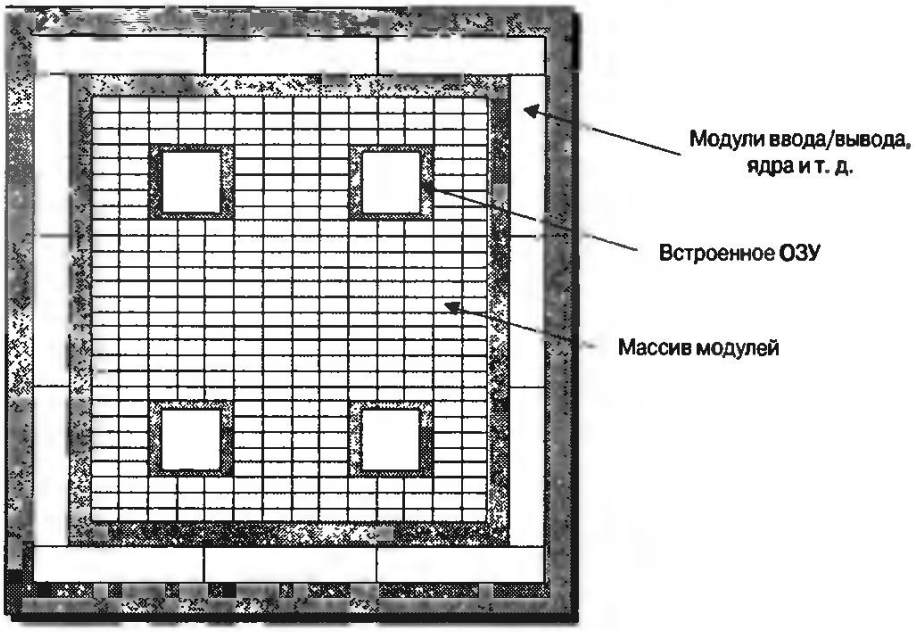


Рис. 3.16. Типовая структурированная специализированная ИС

Дело в том, что при выполнении устройства по заказу изготавливаются только слои металлизации, как и в случае стандартных вентильных матриц. Отличие кроется в том, что вследствие большей сложности модулей структурированных специализированных ИС большинство слоев металлизации уже предопределено. Таким образом, при изготовлении многих структурированных микросхем по заказу их архитектура нуждается всего лишь в двух или трех слоях металлизации, а в некоторых случаях требуется изготовить только один слой с переходными отверстиями. В результате значительно снижается стоимость и сокращается время изготовления оставшихся фотошаблонов, необходимых для создания устройства.

Определить точное значение параметров довольно сложно, однако предопределенная и изготовленная заводским способом логика структурированных микросхем, по сравнению со схемами на стандартных элементах, потребляет большую мощность, обладает большей производительностью и занимает больше места на кристалле. Для выполнения одних и тех же функций структурированные микросхемы в среднем требуют в три раза больше места на кристалле и потребляют в два-три раза больше энергии, чем схемы на стандартных элементах. На практике эти параметры существенно зависят от типа устройства и его архитектуры. К сожалению, в то время, когда писалась эта книга, не существовало каких-либо оценок параметров, основанных на промышленных стандартах.

ПЛИС

В начале 80-х стало ясно, что в сфере цифровых микросхем образовался пробел. С одной стороны, существовали программируемые устройства, подобные простым и сложным ПЛУ, которые отличались высокой конфигурируемостью и малым временем изготовления и модификации. Но эти устройства не были способны поддерживать большие и сложные функции.

1752 г. *Америка. Бенджамин Франклин, проведя опыт с воздушным змеем, доказал, что молния представляет из себя электрический разряд.*

1775 г. *Италия. Алессандро Вольта изобрел генератор статического электричества, так называемую* ***электростатическую индукционную машину.***

С другой стороны, существовали заказные специализированные интегральные микросхемы. Они поддерживали чрезвычайно большие и сложные функции, но также были чрезвычайно дорогими, и для их изготовления требовалось значительное время. Кроме того, окончательный вариант заказной микросхемы «замораживался в кремнии» **(Рис. 3.17)**.

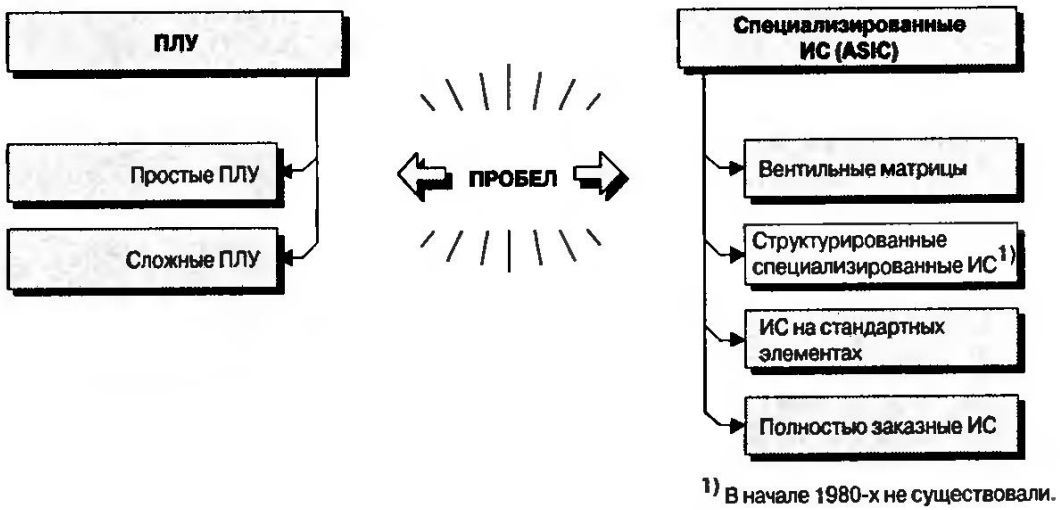


Рис. 3.17. Пробел между ПЛУ и заказными ИС

Чтобы заполнить этот пробел, фирма Xilinx разработала новый класс микросхем FPGA (Field programmable gate arrays), или ПЛИС (программируемые логические интегральные схемы), которые появились в продаже в 1984 году. Различные типы существующих ПЛИС более подробно рассматриваются в гл. 4, а здесь необходимо только уточнить, что ПЛИС изготавливались по КМОП-технологии и для хранения конфигурации использовали статическое ОЗУ. Несмотря на то что первые версии этих устройств были сравнительно простыми и содержали, по сегодняшним меркам, относительно мало вентилей или их эквивалентов, многие представления, лежащие в основе их архитектуры, используются и в наши дни.

Основу первых ПЛИС составляла концепция программируемых логических блоков, которые включали в себя 3-х входовую таблицу соответствия (*LUT — lookup table*), регистр, выполняющий функцию триггера или защелки, мультиплексор, а также некоторые другие элементы, не представляющие в данном контексте особого интереса. На **Рис. 3.18** показан очень простой программируемый логический блок. Как будет показано в гл. 4, логический блок современной ПЛИС может быть чрезвычайно сложным.

Каждая ПЛИС содержит большое количество таких программируемых блоков. Путем программирования соответствующих ячеек статического ОЗУ каждый логический блок устройства может быть сконфигурирован для выполнения различных функций. В процессе конфигурирования в каждый регистр записывается его начальное значение в виде логического 0 или логической 1, а также определяется, будет ли регистр выполнять функцию триггера или защелки **(Рис. 3.18)**. В первом случае также определяется, по какому фронту тактового сигнала, положительному или отрицательному, будет производиться переключение. Тактовые сигналы являются общими для всех логических блоков. Мультиплексор, подключенный к входу триггера, может конфигурироваться на передачу сигналов с выхода таблицы соответствия или с

1777 г. *Чарльз Стэнхоп (Charles Stanhope) разработал механическую счётную машину.*

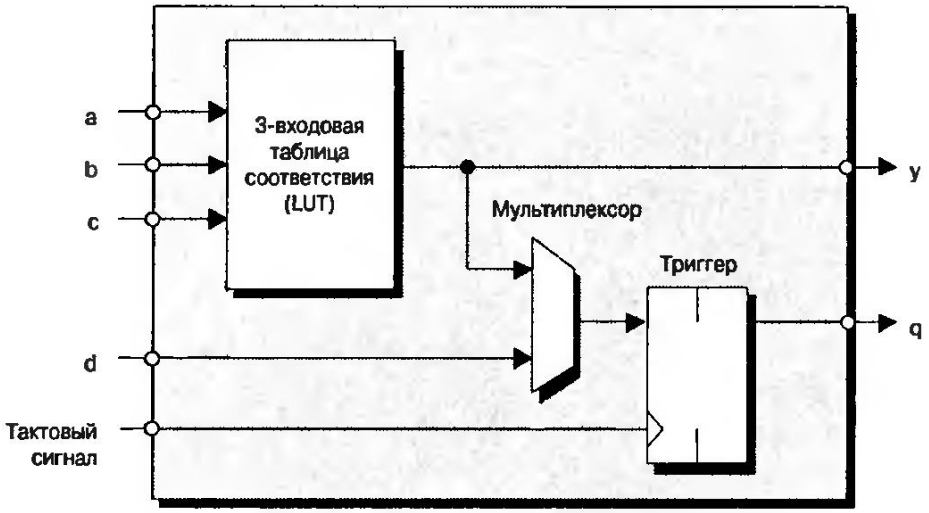


Рис. 3.18. Простой программируемый логический блок

отдельного входа логического блока. Таблица соответствия может программироваться на работу в качестве любой логической функции с тремя входами и одним выходом.

Допустим, что таблице соответствия необходимо сформировать функцию

$$y = (a \,\&\, b) \mid \overline{c}.$$

Для этого в таблицу соответствия необходимо загрузить соответствующие выходные значения этой функции (**Рис. 3.19**).

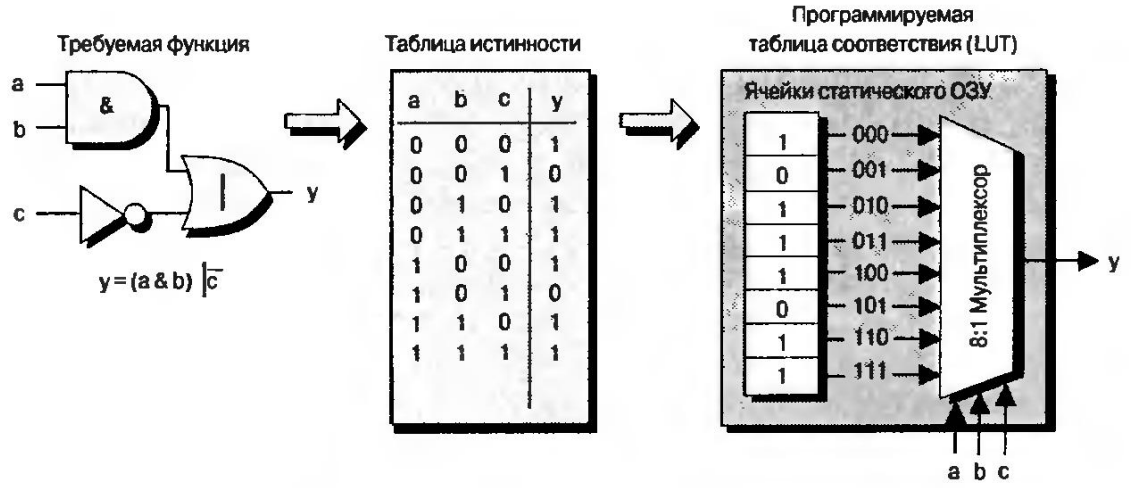


Рис. 3.19. Конфигурирование таблицы соответствия

Таблица соответствия с мультиплексором **8:1**, представленная на **Рис. 3.19**, является упрощенным вариантом существующих таблиц соответствия (*см.* гл. 4 и гл. 5).

ПЛИС состоит из большого числа программируемых логических блоков, или «островов», окруженных «морем» программируемых внутренних соединений (**Рис. 3.20**).

Как обычно, упрощенная схема является всего лишь весьма приближенным представлением архитектуры ПЛИС. В действительности все транзисторы и внутренние соединения выполняются на одном кристалле кремния с помощью стандартных технологий изготовления интегральных микросхем.

Конец 1700-х гг. *Чарльз Стэнхоп (Charles Stanhope) разработал логическую машину, названную демонстратор Стенхопа.*

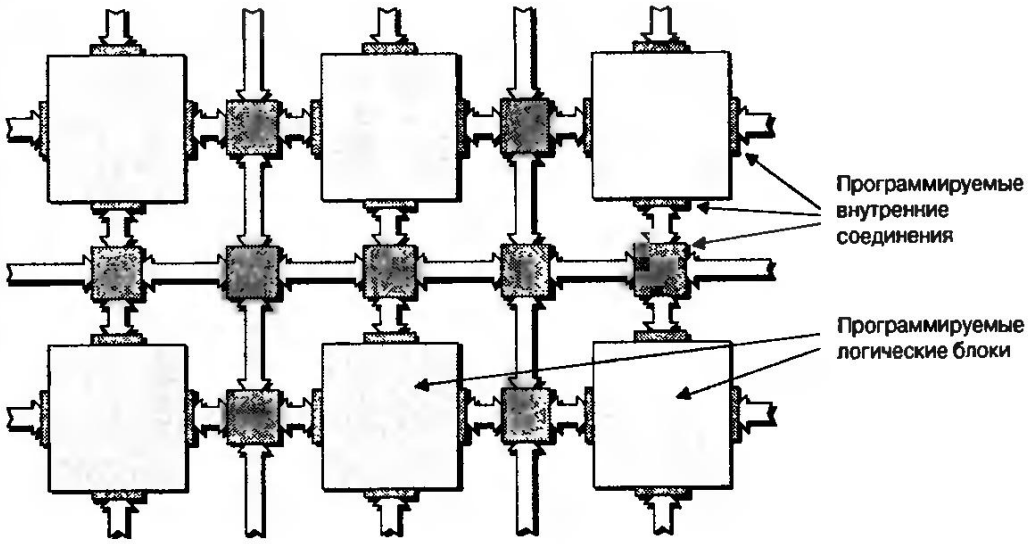


Рис. 3.20. Упрощенная архитектура ПЛИС

Кроме локальных внутренних связей, показанных на **Рис. 3.20**, существуют глобальные, или высокоскоростные, соединения, которые передают сигналы через кристалл без участия многочисленных локальных переключающих элементов.

Устройство также содержит контактные площадки и ножки для ввода/вывода данных (на рисунке не показаны). С помощью ячеек памяти статического ОЗУ внутренние соединения программируются таким образом, чтобы входы микросхемы соединялись с входами одного или нескольких программируемых логических блоков, а выходы этих блоков подключались к входам другим блоков и/или подавались на выходы микросхемы.

В конечном итоге, ПЛИС успешно справились с ролью моста между ПЛУ и заказными специализированными микросхемами. С одной стороны, они обладают высокой степенью конфигурируемости и малым временем изготовления и модификации, как и ПЛУ. С другой стороны, они могут использоваться для реализации невероятно больших и сложных функций, которые раньше реализовывались только с помощью заказных микросхем. Специализированные заказные микросхемы предназначались для действительно больших, сложных и высокотехнологичных систем. Однако на место, отведенное этим микросхемам в сфере программируемой логики, посягают ПЛИС, которые по мере совершенствования начали постепенно их вытеснять.

ПЛИС-платформа

Принцип *разработки по образцу*, или *разработки платформы*, длительное время использовался на уровне печатных плат. Согласно этому принципу на основе базовой конфигурации может быть реализовано множество производных продуктов.

Современные высокотехнологичные ПЛИС, помимо громаднейшего количества программируемой логики, содержат встроенные блоки ОЗУ, встроенные процессорные ядра, высокоскоростные блоки ввода/вывода и многое другое. Кроме того, разработчики имеют доступ к большому набору *блоков интеллектуальной собственности*. Все это способствовало развитию концепции *ПЛИС-платформы*. Суть её заключается в том, что любая компания может использовать уже

спроектированную ПЛИС-платформу для многочисленных изделий внутреннего пользования либо предложить исходные проекты другим компаниям для изготовления заказных устройств или для их видоизменения.

1800 г. *Италия. Алессандро Вольта изобрел первую батарейку.*

Гибриды вида ПЛИС — заказные интеральные схемы

Очевидно, что нет никакого смысла встраивать компоненты заказных интегральных схем внутри ПЛИС, поскольку полученная интегральная схема будет источником классических проблем, которые свойственны технологии создания заказных микросхем: высокие расходы на проектирование и производство, продолжительные сроки выхода на рынок и другие. Однако существует ряд случаев, когда один или несколько компонентов ПЛИС используются как часть заказных микросхем на стандартных элементах.

Одна из причин встраивания компонентов ПЛИС внутрь заказных микросхем — сделать как можно проще принцип разработки платформы. В этом случае платформа будет принадлежать к классу заказных микросхем, но встроенные компоненты ПЛИС будут обеспечивать один из механизмов, используемых для адаптации и модификации изделия.

Другая причинная связана с тем, что последние несколько лет наблюдается расширение сферы применения ПЛИС, в частности они могут использоваться для придания дополнительных свойств изделиям на заказных микросхемах. В этом случае большие и сложные специализированные микросхемы соединяются с ПЛИС, расположенными на одной печатной плате в непосредственной близости друг от друга **(Рис. 3.21)**.

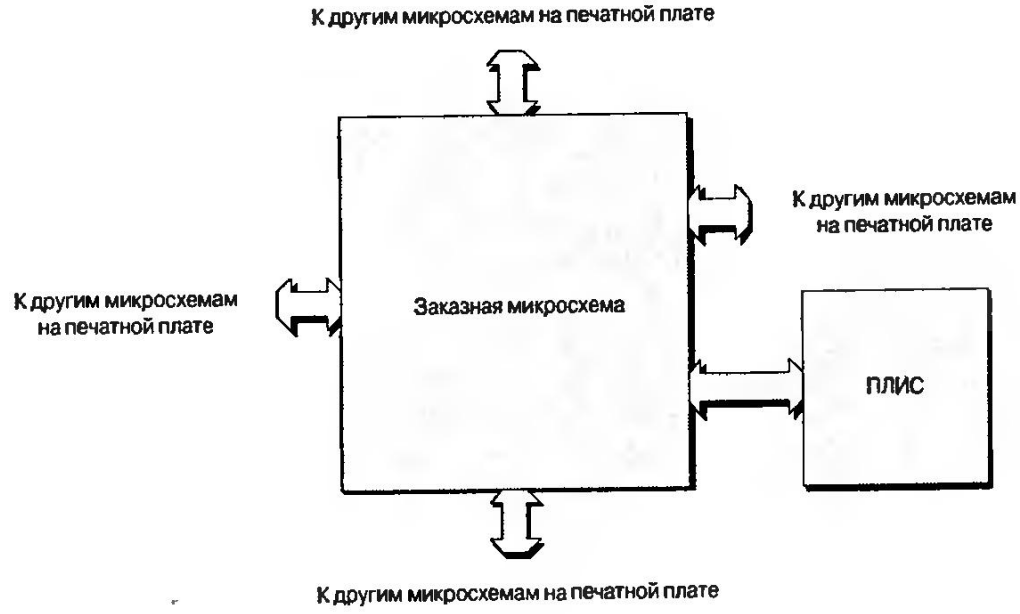


Рис. 3.21. Применение ПЛИС для придания дополнительных свойств изделиям на заказных микросхемах

Применение этой схемы оправданно большой трудоёмкостью и дороговизной операций по устранению ошибок и изменению функциональности системы при использовании заказных микросхем. Даже если изделие на заказных микросхемах не содержит ошибок, применение ПЛИС можно использовать для выполнения низкоуровневых мо-

1801 г. *Франция. Жозеф Жаккар (Joseph Jacquard) разработал ткацкий станок, управляемый перфокартами.*

1820 г. *Франция. Андре Ампер (Andre Ampere) исследовал силу электрического тока в магнитном поле.*

дификаций и модернизации системы. Недостатком такой технологии является длительность прохождения сигналов от заказной микросхемы к ПЛИС и обратно. Устранить этот недостаток можно, встроив ядро ПЛИС внутрь заказной микросхемы. В результате получим *гибрид: ПЛИС — заказная интегральная схема* (FPGA-ASIC).

При этом, однако, надо иметь в виду, что системы автоматизированного проектирования и методы проектирования заказных микросхем и ПЛИС существенно различаются. Например, про заказные микросхемы можно сказать, что они *мелкомодульные*, поскольку в основном они создаются на уровне простейших логических элементов. Это значит, что традиционные методы проектирования, такие как логический синтез, традиционные методы размещения элементов и трассировки соединений также применимы к проектированию мелкомодульных заказных интегральных схем.

Что касается ПЛИС, то про них можно сказать, что они являются *среднемодульными* (или *крупномодульными* в зависимости от того, кто об этом говорит, поскольку физически они реализуются с использованием высокоуровневых блоков, таких как программируемые логические блоки, ранее рассмотренные в этой главе). В этом случае для проектирования лучше использовать специфичные для ПЛИС методы синтеза, размещения элементов и трассировки соединений, поскольку данные методы «воспринимают мир» в терминах высокоуровневых блоков.

Областью применения гибридов вида ПЛИС — заказная микросхема являются структурированные специализированные микросхемы, или структурированные ASIC, поскольку они также соответствуют принципам блочного построения. Когда поставщики структурированных специализированных микросхем выбирают средства проектирования, они чаще общаются с поставщиками ПЛИС–ориентированных средств синтеза, технологий размещения элементов и трассировки соединений, чем с их коллегами, предлагающими традиционные средства. Таким образом, для проектирования гибридов вида ПЛИС-заказная микросхема, основанных на структурированной ASIC, автоматически можно пользоваться едиными средствами разработки, так как одни и те же методы блочного синтеза, размещения и трассировки могут использоваться для изготовления как «заказной», так и «ПЛИС»-части микросхемы.

Как создают ПЛИС

Вопрос, который часто задают, но ответ на который вряд ли можно найти в книге о ПЛИС. Действительно, каким же образом поставщики ПЛИС проектируют новые поколения устройств?

Приходится ли им вручную разрабатывать каждый транзистор и проводник? Придерживаются ли они при этом порядка проектирования, принятого при изготовлении заказных микросхем? Или же они создают описание уровня регистровых передач, преобразуют его в список соединений логических элементов и затем используют программное обеспечение для размещения элементов и трассировки соединений. Другими словами, используют методы, аналогичные тем, которые используются при построении классических заказных микросхем, т. е. вентильных матриц или схем на стандартных элементах (*см.* гл. 2).

Если ответить на все эти вопросы кратко, ответом будет да! Развернутый ответ будет выглядеть приблизительно так:

Некоторые части устройств подобны программируемым логическим блокам и базовым соединительным структурам, и в этом случае поставщики борются за каждый квадратный микрон и каждую частичку наносекунды. Такими частями устройств являются транзисторы и проводники, изготавливаемыми вручную по технологии заказных микросхем. Утешает в этом случае то, что эти части устройства являются относительно малыми и часто повторяющимися, т. е. будучи один раз созданы, они могут тысячи раз воспроизводятся на поверхности кристалла.

Существуют и вспомогательные части устройств, такие как конфигурируемые схемы управления, которые встречаются в единичном экземпляре и к которым не предъявляются жестких требований по размерам и производительности. Эти части изготавливаются методами, которые используются для создания схем на стандартных элементах.