

ОТСЛЕЖИВАНИЕ СОСТОЯНИЯ ВНУТРЕННИХ ТОЧЕК СХЕМЫ В ПЛИС

Отсутствие наглядности

Одна из проблем, связанных с отладкой устройств на программируемых цифровых микросхемах, будь то заказные микросхемы или ПЛИС, заключается в отсутствии возможности увидеть воочию, что же происходит внутри устройства. Давайте рассмотрим простую схему конвейера, состоящую из нескольких регистров и вентилях (Рис. 16.1).

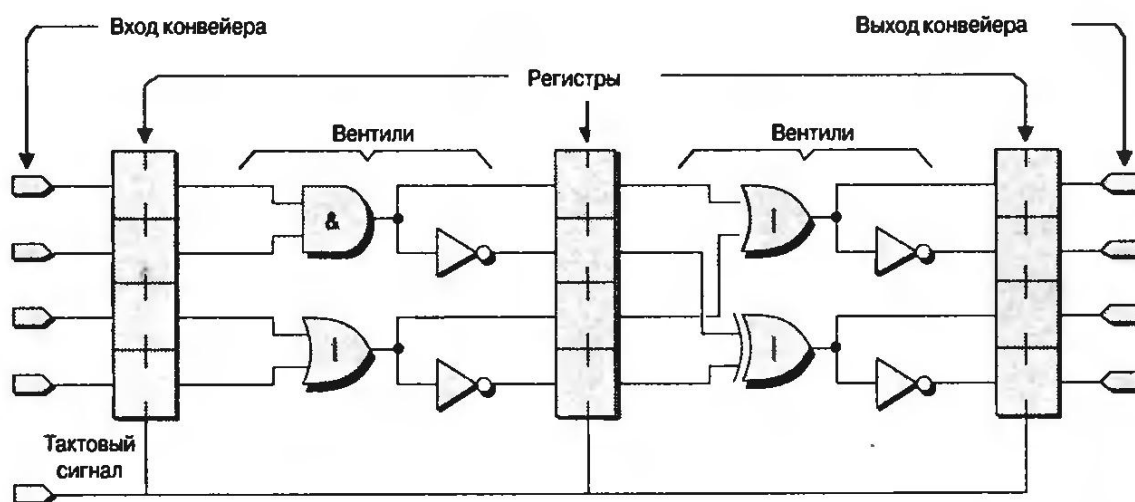


Рис. 16.1. Схема очень простого конвейера

Очевидно, что в некоторой степени эта схема лишена смысла. Читатель, возможно, будет поражен тем, насколько сложными могут быть схемы реальных конвейеров и насколько тяжело бывает изложить их в доступном виде. Поэтому схема, изображенная на Рис. 16.1, удобна для рассмотрения материала этой главы.

Проблема заключается в том, что мы имеем доступ только к входам и выходам конвейера и не можем увидеть, что происходит у него внутри. В этом и нет необходимости, если разработка устройства завершена и проведены все его испытания. Однако отсутствие наглядности происходящего внутри конвейера может вызвать головную боль, если мы пытаемся произвести отладку устройства, чтобы определить, почему оно не выполняет тех функций, которые от него ожидают.

Очевидное решение может быть реализовано путем соединения внутренних точек схемы с внешними выводами устройства, вследствие чего обеспечивается доступ к их содержимому (Рис. 16.2).

Трудности реализации этого подхода заключаются в ограниченном количестве доступных выводов у большинства микросхем. На практике, у многих ПЛИС-устройств остается незадействованной куча внутренних ресурсов, поскольку, как правило, не хватает свободных выводов, необходимых для передачи сигналов управления и данных в устройство и из него, даже если мы не будем использовать какие-либо выводы для доступа к внутренним точкам.

1921 г. Чешский писатель Карел Чапек (Karel Čapek) придумал понятие робот для своей пьесы «РУР — Росsumские универсальные роботы».

1921 г. Для стабилизации частоты начали применять кварцевые кристаллы.

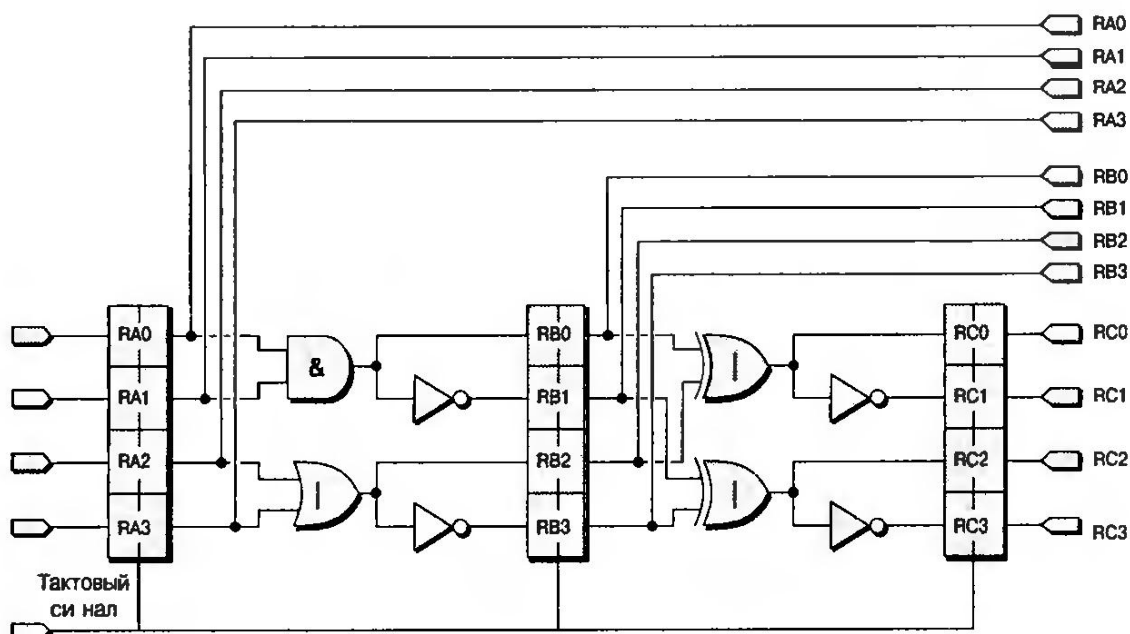


Рис. 16.2. Подсоединение внутренних точек схемы к внешним выводам

Мультиплексирование

Простое решение описанной выше проблемы заключается в мультиплексировании внешних выходов конвейера с внутренними сигналами и в последующей их передаче через одни и те же внешние выводы (Рис. 16.3).

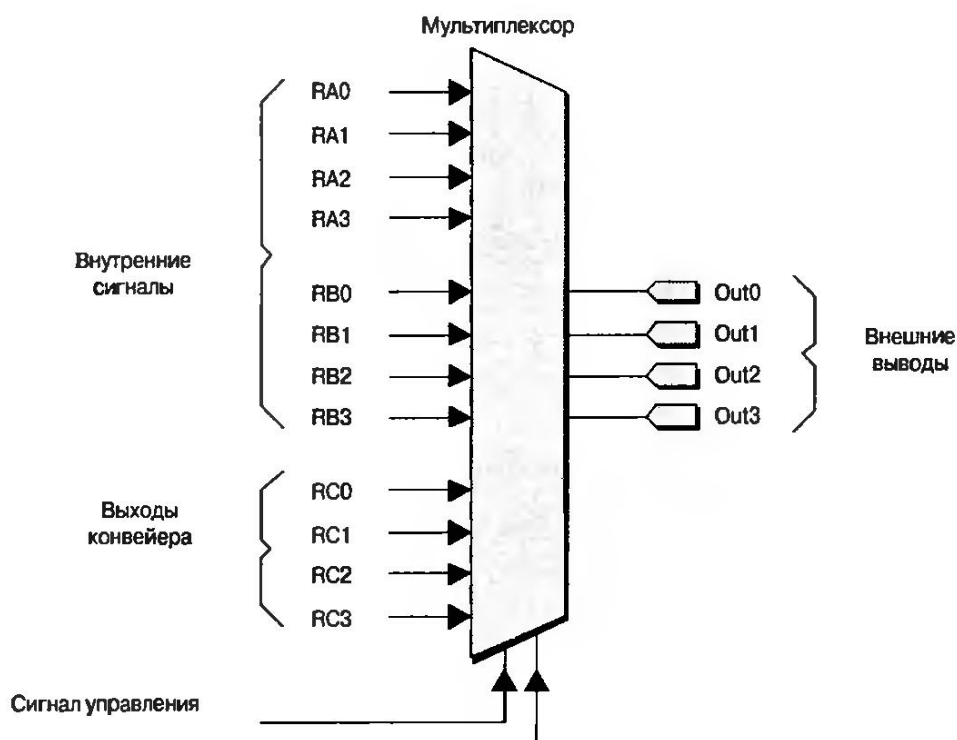


Рис. 16.3. Мультиплексирование сигналов

Разумеется, для сигналов управления мультиплексорами потребуются отдельные выводы. В нашем примере два сигнала управления напрямую соединены с внешним миром, и для этого потребуются два вывода. Однако можно использовать несколько вентилях для реализации простого конечного автомата, который потребует только одного вывода для переключения состояний мультиплексора, причем каждому состоянию будет соответствовать определенный набор сигналов.

Основное преимущество этой схемы заключается в большой степени наглядности и относительно высокой скорости работы. Недостатком является отсутствие гибкости и трудоёмкость реализации, так как если поменять точку наблюдения, придётся модифицировать исходный код устройства и произвести повторный синтез. Аналогично, если изменить режим переключения триггера(ов), используемый в конечном автомате для определения наборов сигналов, выбираемых мультиплексором в определенное время, также придётся изменить исходный код устройства.

Другой недостаток заключается в том, что, если после отладки устройства удалить тестовые структуры из исходного кода, могут возникнуть новые проблемы, например может измениться разводка элементов ПЛИС и, как следствие, величина задержек сигналов.

Специальные цепи отладки

Некоторые ПЛИС содержат специальные цепи отладки, позволяющие вести наблюдение за внутренними точками. Например, микросхема фирмы Actel содержит два специальных вывода, называемых PRA («Probe A» — «пробник А») и PRB («Probe B» — «пробник В»). С помощью встроенной отладочной цепи вместе со специальным отладочным программным обеспечением¹⁾ на эти выводы можно подать любые внутренние сигналы и провести анализ состояния в различных точках схемы устройства.

Основное преимущество такого подхода заключается в том, что исключается необходимость изменять исходный код устройства. Однако двух тестовых выводов не достаточно для полноценных исследований, так как анализу порой должны быть подвергнуты сотни тысяч внутренних сигналов.

Виртуальный логический анализатор

Кроме рассмотренной выше весьма полезной схемы, довольно часто возникает необходимость в средствах обширного логического анализа, позволяющих отслеживать и производить отладку групп внутренних сигналов, производить анализ одних сигналов относительно других или относительно определённых событий. В этом случае решить проблему можно с помощью *внутрикристальных инструментальных средств (OCI — On-chip instrumentation)*, которые упрощают логическую отладку устройства, позволяя пользователю встраивать диагностические блоки интеллектуальной собственности в свои устройства, например, можно встроить виртуальный логический анализатор. Идея заключается в использовании некоторого количества ресурсов ПЛИС для реализации одного или нескольких блоков виртуального логического анализа, фиксирующих активность заданных сигналов. Эти данные могут быть сохранены в одном или нескольких блоках встроенного ОЗУ ПЛИС, доступ к содержимому которых может быть получен с помощью внешнего программного обеспечения через JTAG-порт устройства (Рис. 16.4).

Часть виртуального логического анализатора используется для определения состояния специальных управляющих сигналов, с помо-

1922 г. Начало работы коммерческого радиовещания (по цене 100\$ за 10-минутный репортаж).

1923 г. Появилась первая неоновая рекламная вывеска.

¹⁾ Это программное обеспечение раньше называлось Actionprobe®. Затем появилась «новая и улучшенная» версия Silicon Explorer. Во время написания этой книги наибольшей популярностью пользовалась версия Silicon ExplorerII.

1923 г. Разработана первая фотоэлектрическая ячейка.

1923 г. Осуществлен первый сеанс радиосвязи корабль-корабль.

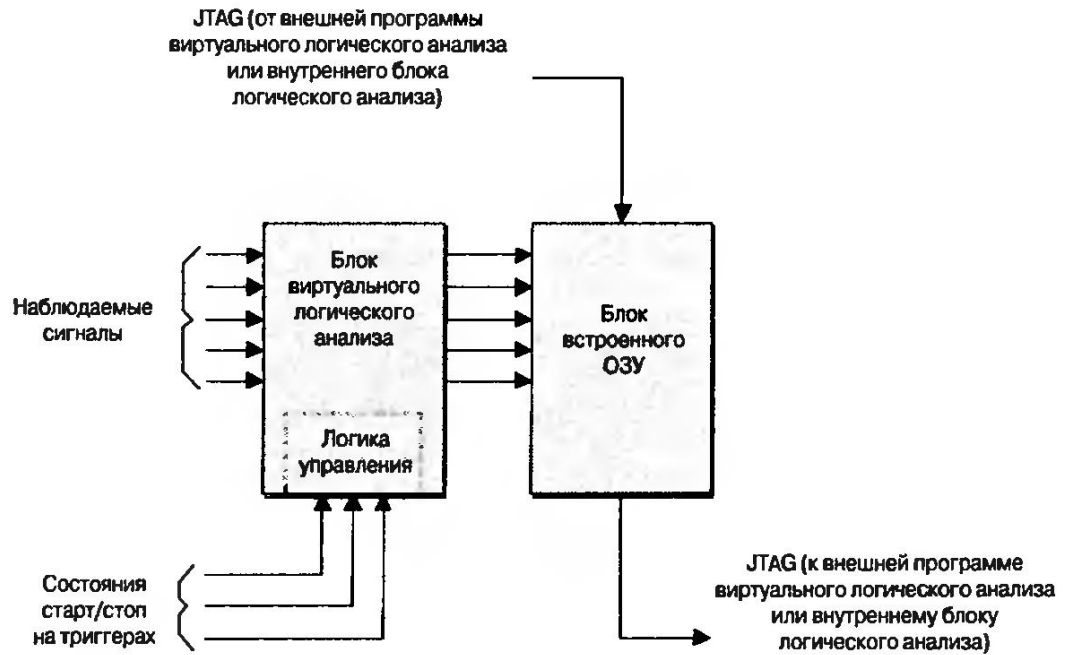


Рис. 16.4. Виртуальный логический анализатор

стью которых производится включение и выключение режима захвата данных с наблюдаемых сигналов.

В зависимости от типа используемого виртуального логического анализатора, либо придётся, либо не придётся модифицировать исходный код устройства для включения этой функциональности. Существенное преимущество схемы данного типа заключается в том, что даже при включении специальных макросов в исходный код системы реализация чрезвычайно сложных возможностей отладки не вызовет каких-либо трудностей.

В некоторых случаях поставщики ПЛИС сами предлагают такие возможности отладки. Хорошим примером средств этого типа могут служить продукты *Chipscope™* Pro компании Xilinx (www.xilinx.com) и *SignalTap®* II компании Altera (www.altera.com). Вместе с тем, при работе с устройством, не содержащим встроенных возможностей отладки, можно обратиться к сторонним разработчикам, например к компании First Silicon Solutions Inc. (www.fs2.com), которая специализируется на внутрикристальных инструментальных средствах и средствах отладки для ПЛИС-логики и встраиваемых процессоров.

В качестве виртуальных логических анализаторов, используемых для трассировки, анализа и отладки внутренних сигналов ПЛИС, компания First Silicon Solutions предлагает *конфигурируемые модули логического анализатора (CLAM — configurable logic analyzer module)*. Эти маленькие штучки состоят из блоков внутрикристальных инструментальных средств, описанных на языках Verilog или VHDL, которые конфигурируются и синтезируются как часть устройства. Этот блок (или блоки, если использовать несколько штук) используется совместно с программным обеспечением управления, тестирования и отображения информации, установленном на персональном компьютере.

VirtualWires

В начале 90-х компания Virtual Machine Works обнародовала свою новую технологию под названием VirtualWire™. Изначально представленная как технология производства больших устройств, для реализа-

ции которых приходится использовать несколько ПЛИС, VirtualWire подготовила почву для различных эмуляторов на основе ПЛИС. Одна из причин, по которой данная технология упоминается в этой главе, заключается в том, что она представляет собой некое подобие рассмотренному ранее методу мультиплексирования. Еще одна причина кроется в том, что это действительно очень хорошее решение.

Проблема

Технология VirtualWire применяется в больших устройствах, размер которых не позволяет разместить их в одной ПЛИС, и поэтому реализацию столь больших устройств приходится разделять на несколько микросхем. В качестве простого примера рассмотрим устройство, которое эквивалентно некоторому количеству системных логических элементов, но наибольшая из ПЛИС содержит только половину требуемого количества таких элементов. Очевидно, что в этом случае устройство будет состоять из двух микросхем (Рис. 16.5).

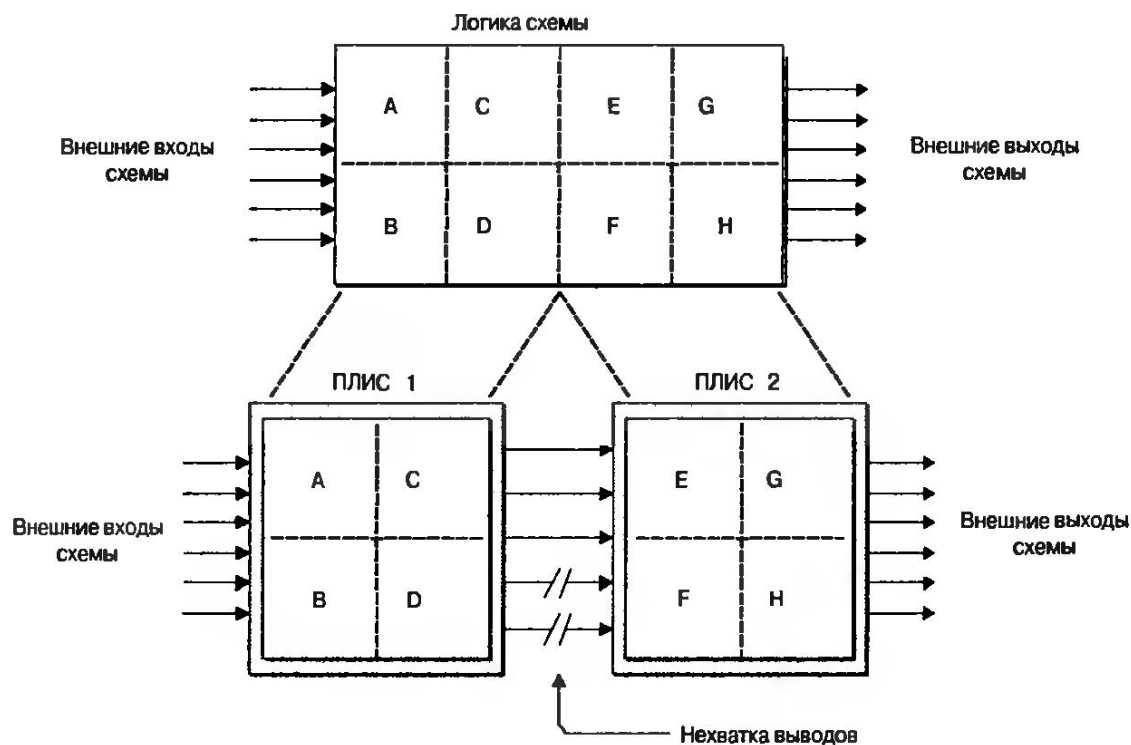


Рис. 16.5. У микросхем не хватает выводов, если мы пытаемся разделить конструкцию на два устройства

На Рис. 16.5 логика показана в виде нескольких субблоков, обозначенных латинскими буквами от А до Н. Такое обозначение использовано исключительно для наглядности метода разбиения устройства.

Недостатком такой реализации является то, что микросхемы обычно не располагают требуемым количеством выводов для удовлетворения потребностей во внешних входах и выходах схемы, а также для связи между различными блоками устройства. Методы, предшествующие технологии VirtualWire или аналогичные предусматривали разбиение конструкции на большее количество частей (Рис. 16.6).

В этом случае возникает очередная проблема, связанная с тем, что огромная часть логических ресурсов ПЛИС остается не задействованной, и в результате для реализации требуемой функциональности устройства потребуется довольно много микросхем.

1925 г. Америка.
Учёный, инженер и политик Ваневар Буш (Vannevar Bush) сконструировал аналоговый компьютер, названный *Product Intergraph*.

1925 г. На территории США начал работать коммерческий сервис по передаче изображений/фоксов по радиоканалу.

1926 г. Америка. Джулиус Лилиенфилд (Dr. Julius Edgar Lilienfeld) из Нью-Йорка получил патент, согласно которому теперь нам необходимо получать разрешение на применение *n-p-n-транзистора* в качестве усилителя.

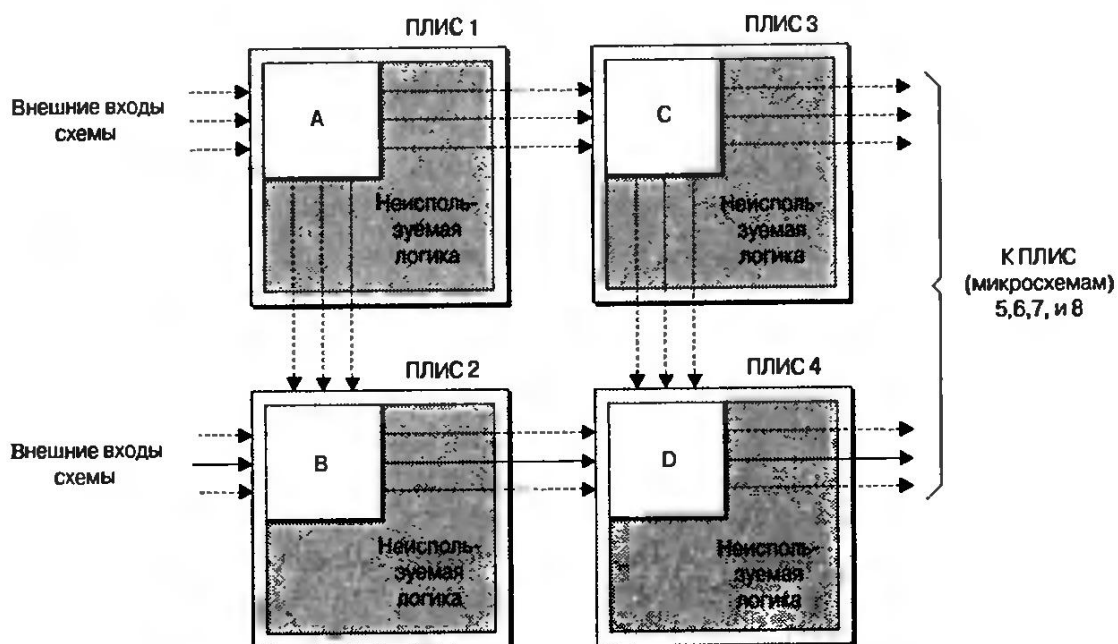


Рис. 16.6. Неиспользованные ресурсы ПЛИС при разбиении устройства на части

Решение VirtualWire

Чтобы понять, как VirtualWire способна решить эту проблему, рассмотрим сначала экзотический случай, когда имеются очень странные микросхемы всего лишь с тремя выводами, из которых 2 вывода — это входы, причём один используется для синхронизации, а третий — это выход. В этом случае, по всей видимости, в каждой микросхеме можно использовать незначительное количество логических ресурсов (Рис. 16.7).

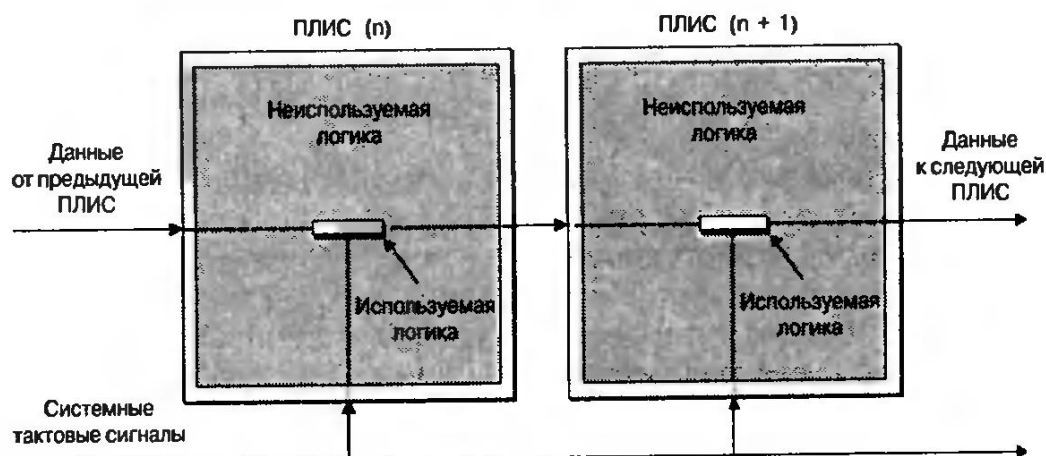


Рис. 16.7. ПЛИС с тремя выводами

Идея, заложенная в основе технологии VirtualWire, заключается в следующем: поскольку огромное количество внутренних ресурсов устройства тратится в пустую, часть из них можно было использовать для реализации специальных цепей. Эти цепи позволят переключать единственный вход микросхемы между несколькими регистрами, каждый из которых подключен к определённому логическому блоку. Выходы этих логических блоков могут быть также мультиплексированы, а выходное значение сохранено в регистре (Рис. 16.8).

В схеме на Рис. 16.8 тактовые сигналы заменены виртуальными тактовыми сигналами, при этом один системный такт делится на несколько виртуальных тактов. Кроме того, внутри каждой ПЛИС реализован

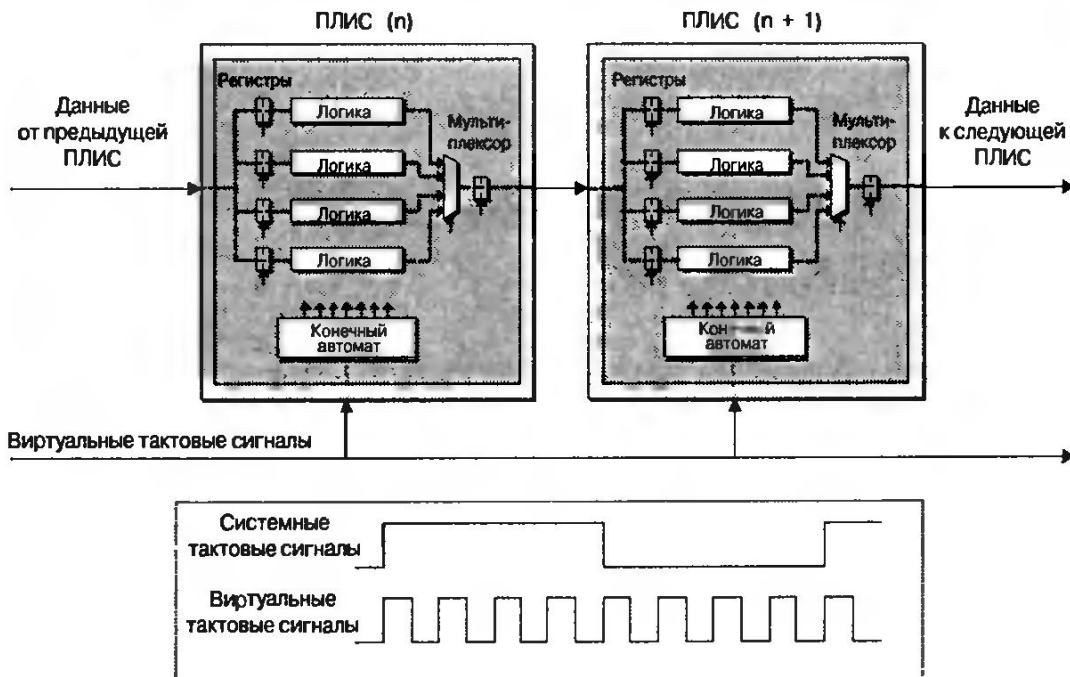


Рис. 16.8. Простой пример технологии VirtualWire

конечный автомат, который используется для включения/выключения регистров, управления мультиплексорами и для других целей. Разумеется, Рис. 16.8 приведён не в масштабе: конечные автоматы и другие структуры VirtualWire на самом деле потребляют относительно небольшое количество логических ресурсов в каждой микросхеме по сравнению с логическими блоками, которые выполняют основные задачи.

С поступлением каждого виртуального тактового сигнала конечные автоматы внутри всех ПЛИС включают определённые регистры, выход которых подключен к входу соответствующих логических блоков. Вследствие этого данные с входа микросхемы записываются в эти регистры и передаются в соответствующие логические блоки. В это же время конечный автомат выдаёт команды мультиплексору, который подключает выход одного из логических блоков к выходному регистру и сохраняет в нём результаты работы. Выход регистра подключен к выходу микросхемы, который, в свою очередь, подключен к входу другой ПЛИС и так далее по цепочке.

Конечно же, в реальном мире ПЛИС имеют сотни или даже тысячи выводов. Следовательно, каждый вход может быть подключен к нескольким логическим блокам, и к каждому физическому выходу с помощью мультиплексора VirtualWire также могут быть подключены несколько логических блоков. Не будем продолжать эту длинную историю, в завершении лишь заметим, что рассмотренные методы стремительно развиваются и усложняются, но принципы, заложенные в их основу, остаются неизменными.

И последнее, но не менее важное. Ключевым элементом концепции VirtualWire является компилятор, который берёт начальное описание устройства в виде таблицы соединения вентилях, разбивает её на несколько ПЛИС, автоматически формирует конечные автоматы и другие структуры VirtualWire, после чего генерирует конфигурационные файлы, предназначенные для загрузки в каждую ПЛИС.

1926 г. Америка.
Разработан первый тестер.

1926 г. Произведена
на передача первого
факсимильного со-
общения по радио-
каналу через Ат-
лантику.

**1926 г. Джон Лоу-
ган Бэрд (John
Logie Baird) проде-
монстрировал ра-
боту электромеха-
нической телевизи-
онной системы.**

1927 г. Изобретена
пятиэлектродная
вакуумная лампа
(пентод).