

От автора

Выход в 2005 году моей книги «Разработка цифровых устройств на основе ПЛИС Xilinx™ с применением языка VHDL» [8] показал, что изложенный в ней материал в целом востребован российскими читателями. Полученный за более чем 20 лет опыт разработки цифровых устройств различного назначения и преподавания в учебном центре компании Xilinx в России позволил существенно переработать материал книги, которая задумывалась как практическое пособие для инженеров, не претендующее на справочную полноту изложения материала, однако поясняющее наиболее актуальные вопросы проектирования. Общение со специалистами и консультации по выполнению проектов показывают, что в области работы с ПЛИС существует целый ряд вопросов, нуждающихся в иллюстрациях, схемах и подробном разъяснении источников наблюдаемых эффектов и способов решения проблем. Издания справочного характера или техническая документация обычно не способствуют формированию системного взгляда на вопросы проектирования, который для практикующего разработчика должен включать всесторонний анализ решаемых им задач. Вместо этого книга задумывалась и писалась как сборник ответов на вопросы «что конкретно делать с проектом сейчас?».

В процессе подготовки материала и создания примеров активно использовались результаты обсуждений на лекциях учебного центра Xilinx. Российские инженеры — слушатели курсов — наглядно демонстрировали высокий уровень технической подготовки и ставили интересные и актуальные вопросы. Многие из них получили свое отражение в материале книги. Также важным источником материалов и местом проверки идей является инженерная группа, возглавляемая профессором Дмитрием Станиславовичем Потехиным. Деятельность этого коллектива стала источником интересных задач, позволивших на практике исследовать многие вопросы, отраженные в материалах книги.

Отдельное спасибо человеку, имеющему уникально тесную связь с моей деятельностью по созданию книг. Выход в свет моей первой книги и его рождение отделяют всего несколько дней. В итоге мой сын Антон стал первым читателем материала, опробовав на себе освоение некоторых проблемных вопросов проектирования. Хочется надеяться, что и технические специалисты найдут изложение достаточно простым и понятным.

Введение

Данная книга предлагает читателю сведения о проектировании цифровых систем на базе ПЛИС Xilinx. В главе 1 приведены начальные сведения о цифровой электронике и описание ПЛИС семейств Spartan-6, Spartan-7, Artix-7, Kintex-7 и Zynq-7000, которые могут быть использованы в широком спектре устройств, в том числе и для расширения возможностей микроконтроллера. Материалы книги могут быть использованы для изучения порядка проектирования с использованием семейств верхнего уровня, таких как Virtex и Zynq MPSOC.

Глава 2 содержит описание «быстрого старта» в САПР ПЛИС Vivado и ISE.

Главы 3 и 4 описывают возможности языков описания аппаратуры VHDL и Verilog. Главы имеют приблизительно одинаковую организацию, поскольку рассмотренные языки обладают сопоставимыми возможностями. С учетом того, что разработчики могут выбирать один из языков в качестве приоритетного, главы могут быть изучены выборочно или совместно. Рекомендуется начинать изучение с языка VHDL.

Глава 5 посвящена применению дополнительных инструментов проектирования Xilinx. В ней приводятся примеры реализации процессорных систем на базе так называемых софт-процессоров (т.е. процессоров, которые образуются путем программирования ячеек ПЛИС) MicroBlaze и PicoBlaze. Также приводятся краткие сведения о применении языков высокого уровня (HLS, High Level Synthesis).

Глава 6 приводит примеры реализации отдельных модулей на HDL. Знакомство с данной главой может быть полезно для эффективного использования возможностей ПЛИС Spartan-6 и создания устойчиво работающих схем, учитывающих технологические особенности производства современных цифровых микросхем.

В главе 7 приведены сведения, касающиеся подходов оптимизации проектов на ПЛИС и используемых для этого инструментов САПР. В первую очередь это касается эффективного использования проектных ограничений и анализа отчетов о результатах работы САПР.

Глава 8 рассматривает вопросы проектирования устройств на базе ПЛИС, включая вопросы организации питания, программирования, разработки печатной платы.

В приложении приведены сведения о языке System Verilog, который обладает широкими возможностями при моделировании цифро-

вых схем. Этот язык на момент издания книги ограниченно поддерживается в САПР Vivado, поэтому его практическое использование следует рассматривать при использовании других инструментов моделирования схем. В целом владение языком пока нельзя считать критически важным навыком при разработке систем на базе ПЛИС, поэтому информация о нем приведена в приложении.

Практическое использование ПЛИС Xilinx можно начать с установки бесплатно распространяемой версии САПР Vivado Webpack (или ISE Webpack для Spartan-6), загружаемой официального сайта компании-производителя <http://www.xilinx.com/> В России также функционирует учебный центр Xilinx на базе официального дистрибьютора Xilinx КТЦ «Инлайн Групп», <http://www.plis.ru>, реализующий программы повышения квалификации по учебным материалам, разработанным Xilinx.

1 Начальные сведения о цифровой электронике и архитектуре ПЛИС Xilinx

Возможность провести оперативное и недорогое натурное моделирование крайне полезно во многих областях человеческой деятельности. Разработка программ является наглядным примером процесса создания нового продукта, который не требует затрат компонентов или сырья для каждой итерации. Имея компьютер, программист может реализовать широчайший спектр идей, при этом он не ограничен требованием каждый раз безвозвратно расходовать материальные носители информации. В области механики подобные возможности предоставляют конструкторы Lego. В области дискретной электроники такую роль играет возможность монтажа компонентов пайкой на макетную плату или накруткой проводов. Во всех случаях комбинирование компонентов базового набора (команды компьютерного процессора, кубики Lego) открывает достаточно широкие возможности для построения устройств.

Микроэлектроника предполагает исполнение всех компонентов и связей между ними на полупроводниковой пластине. После изготовления *интегральной микросхемы* промышленным способом изменение ее схемы уже невозможно, а стоимость подготовки производства существенно возрастает с каждым поколением технологических процессов. Поэтому производители микросхем должны выбирать такие схемы, которые могли бы быть востребованы тиражами десятки и сотни тысяч штук. Это практически исключает возможность экспериментов с микросхемами, требующих их постоянного исправления.

Обеспечить относительную гибкость интегральной микросхемы можно, если управлять соединениями не механически, путем добавления или разрыва проводников, а электрически, программируя заранее предусмотренные на кристалле соединители. Микросхемы с набором таких соединителей (ключей) на кристалле относятся к классу программируемых устройств. В зависимости от того, переключаются цифровые или аналоговые сигналы, микросхемы относят к программируемым логическим интегральным схемам (ПЛИС) или к программируемым аналоговым интегральным схемам (ПАИС).

Микросхемы ПЛИС на протяжении своего существования претерпели достаточно заметную эволюцию. От простых программируемых устройств (SPLD, Simple Programmable Logic Device), которые имели ограниченные возможности модификации, они прошли путь до комплексных устройств с гибкими возможностями построения схемы — CPLD (complex Programmable Logic Devices) и FPGA (Field Programmable Gate Array). Эти микросхемы в настоящее время производятся с применением наиболее современных технологических процессов, что обеспечивает высокие технические характеристики и большой объем в логических элементах. В данном издании будут рассмотрены ПЛИС небольшого логического объема (преимущественно Spartan-6, серия 7 компании Xilinx), с помощью которых тем не менее можно разработать достаточно широкий спектр устройств.

1.1. Основы схемотехники цифровых устройств

В основе цифровой электроники лежит использование ключевых режимов работы полупроводниковых приборов. В таком режиме полупроводниковый транзистор может рассматриваться как устройство, сопротивление между двумя электродами которого может управляться при помощи третьего. В биполярном транзисторе ток базы управляет сопротивлением между эмиттером и коллектором, а в полевом напряжение затвор-исток управляет сопротивлением между истоком и стоком. В одном из состояний сопротивление близко к нулю (ключ замкнут), а в другом — бесконечно велико (ключ разомкнут). Механическая модель транзистора и варианты его включения показаны на рис. 1.1.

Из подобных элементов может быть построена схема, состояние которой может быть описано двумя уровнями напряжения — низким и высоким. С целью увеличения помехоустойчивости для каждого уровня выбирается некоторый диапазон значений напряжения, обеспечивающий устойчивое считывание состояния цифрового выхода входами других цифровых узлов (рис. 1.2).

Напряжения, меньшие некоторого минимального значения $U_{0\max}$, трактуются как логическая величина «ноль», а большие не-

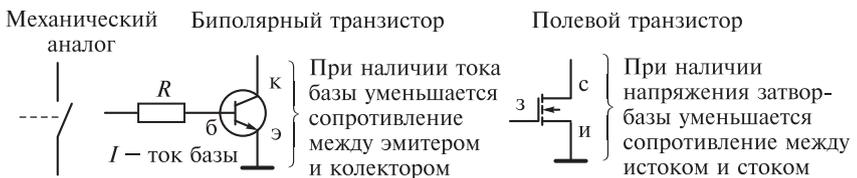


Рис. 1.1. Механическая модель транзистора и транзисторы в роли ключей.

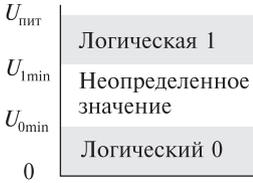


Рис. 1.2. Логические уровни в цифровых устройствах

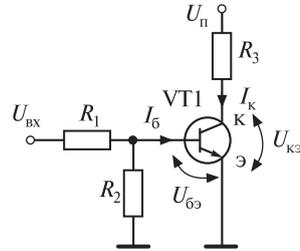


Рис. 1.3. Простейший инвертор на биполярном транзисторе

которого $U_{1\min}$ — как «логическая единица». Напряжение питания естественным образом ограничивает диапазон возможных значений напряжения — нижним пределом логического нуля является нулевое напряжение, а верхним пределом логической единицы — напряжение питания. Таким образом, с помощью значений напряжения можно представить простейшие сигналы, принимающие только два значения. Однако кроме представления требуется и преобразование сигналов, что также обеспечивается полупроводниковыми устройствами.

На рис. 1.3 изображена упрощенная схема инвертора на биполярном транзисторе. Принцип работы данной схемы можно понять, если учесть, что при наличии тока базы транзистор уменьшает сопротивление между выводами эмиттера и коллектора («открывается»), а при отсутствии — увеличивает («закрывается»). Резисторы R_1 и R_2 служат для ограничения тока базы, а R_3 ограничивает максимальный ток коллектора. Теперь, если на вход данной схемы будет подано напряжение, соответствующее логической единице, управляющий ток базы откроет транзистор и сопротивление между эмиттером и коллектором резко уменьшится. Следовательно, напряжение на коллекторе будет близко к нулю. Если же напряжение на входе будет близко к нулю, ток базы будет слишком мал для открывания транзистора и напряжение на коллекторе будет определяться током, протекающим от источника питания через резистор R_3 . Таким образом, логический уровень на выходе данной схемы всегда противоположен логическому уровню на ее входе. Такое устройство называется инвертором (рис. 1.4).

Показанная на рис. 1.3 схема является сильно упрощенной и не соответствует схемотехнике промышленных цифровых микросхем. Тем не менее она позволяет ознакомиться с принципом использования биполярных транзисторов для создания цифровых схем. В реальных устройствах используется первичное преобразование элек-

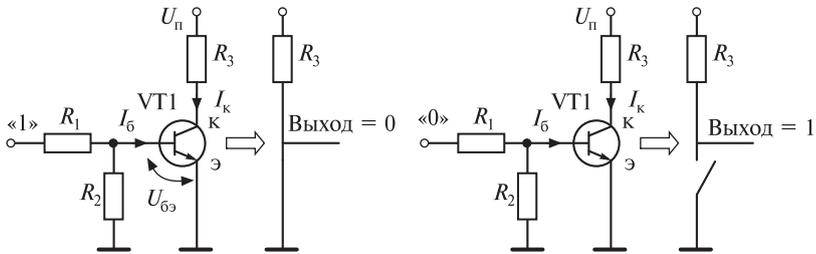


Рис. 1.4. Работа инвертора

трических сигналов, которое обычно также выполняется с помощью транзисторов. Вследствие этого одна из наиболее распространенных архитектур цифровых систем носит название «транзисторно-транзисторная логика», или ТТЛ. Такое название отражает тот факт, что транзисторы используются как для первичного преобразования сигналов, так и для формирования выходного напряжения. Предшественниками такой архитектуры являются резисторно-транзисторная (показанная на рис. 1.3) и диодно-транзисторная логика.

Существенным недостатком микросхем ТТЛ является наличие статического потребления — для удержания биполярного транзистора в открытом состоянии требуется протекание постоянного тока базы, поэтому использование биполярных транзисторов в цифровых устройствах ограничено максимальной рассеиваемой мощностью.

Альтернативным вариантом является построение цифровых устройств на полевых транзисторах, которые управляются потенциалом затвора и не требуют постоянного протекания управляющего тока. Удобнее использовать полевые транзисторы, выполненные по технологии «металл–окисел–полупроводник» (МОП). В зависимости от типа проводимости канала такие транзисторы могут открываться либо низким логическим уровнем, либо высоким. Использование транзисторов, аналогичных по характеристикам, но имеющих разный тип проводимости канала (*комплементарных*, т. е. «дополняющих») позволяет реализовать эффективный инвертор (рис. 1.5). Архитектура микросхем, основанных на комплементарных полевых транзисторах, обозначается КМОП (также CMOS — Complementary Metal–Oxide–Semiconductor).

На схеме рис. 1.5 верхний транзистор открывается при низком логическом уровне, подключаемая к выходу напряжение питания (т. е. обеспечи-

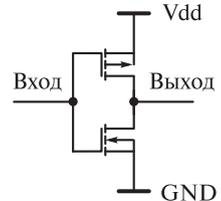


Рис. 1.5. Принципиальная электрическая схема инвертора КМОП

вая высокий уровень выходного напряжения). При высоком входном уровне верхний транзистор закрывается, а нижний — открывается и подключает к выходу уровень нуля. Статическое потребление такой схемы определяется утечками тока через слой диэлектрика между затвором и каналом, т. е. весьма мало. Большую роль играет так называемое динамическое потребление: в процессе переключения входного напряжения существует период времени, в течение которого оба транзистора обладают конечным сопротивлением и наблюдается протекание тока через оба канала (в статическом состоянии один из транзисторов почти полностью открыт, а другой — почти полностью закрыт). При этом мощность, потребляемая схемами КМОП, оказывается приблизительно пропорциональной частоте переключения внутренних сигналов.

На практике различие между ТТЛ и КМОП проявляется в уровне напряжений логической единицы. Сравнение схем инверторов позволяет увидеть, что в коллекторной цепи выходного транзистора ТТЛ имеется резистор. Поэтому если транзистор открыт, то выход подключается не к напряжению питания, а к этому резистору. Чем больше выходной ток, требуемый от элемента, тем большим будет падение напряжения на резисторе и, соответственно, меньше выходное напряжение. Поэтому для микросхем ТТЛ уровень, воспринимаемый как логическая единица, устанавливается достаточно низким, чтобы учесть возможное падение напряжения. Для схем с питанием 5 В или 3,3 В этот уровень составляет 2,4 В. В ряде случаев применяется порог 2,0 В.

В отличие от ТТЛ, микросхемы КМОП имеют на выходе симметричный каскад из комплементарных полевых транзисторов (комплементарность и отражена в аббревиатуре — символом «К»). Поэтому уровни выходных напряжений оказываются близки к нулю и напряжению питания соответственно. Порог срабатывания для логической единицы устанавливается в 80 % от напряжения питания. Таким образом, при питании в 5 В микросхема КМОП требует подачи на вход напряжения 4,0 В для надежного распознавания логической единицы.

Уровень логического нуля для обоих вариантов одинаков и составляет 0,8 В. Однако несоответствие выходного напряжения ТТЛ и входного уровня логической единицы КМОП может вызвать проблему при их соединении. Это касается только случая, когда выход ТТЛ подается на вход КМОП, поскольку ТТЛ может выдавать 2,4 В, рассчитывая, что это напряжение будет достаточным для распознавания логической единицы, но вход микросхемы КМОП потребует

4,0 В. В подобных случаях необходимо использовать специальные микросхемы согласования уровней. Можно отметить, что в ряде случаев микросхемы КМОП выполняются с учетом особенностей работы ТТЛ и их входной уровень логической единицы равен 2,4 или даже 2,0 В.

ТТЛ и КМОП являются наиболее распространенными, но не единственными подходами к реализации цифровых устройств. Ранее использовались также интегральная инжекционная (I^2L) и эмиттерно-связанная логика (ЭСЛ, также Emitter Coupled Logic, ECL). Последняя позволяет обеспечить достаточно высокое быстродействие, однако имеет повышенное энергопотребление по сравнению со схемами ТТЛ и КМОП. Кроме того, элемент ЭСЛ использует отрицательное напряжение питания. Это усложняет его соединение с компонентами ТТЛ/КМОП, однако с развитием полупроводниковой технологии преимущества отрицательного питания стали не такими существенными. Поэтому существует и современная разновидность PECL (Positive Emitter-Coupled Logic).

Для повышения помехоустойчивости цифровых линий на высоких рабочих частотах используются также дифференциальные электрические интерфейсы. Одним из наиболее распространенных является LVDS (Low-Voltage Differential Signaling). Логический уровень при этом передается не абсолютным уровнем напряжения, а их разностью на положительном (positive, P) и отрицательном (negative, N) проводниках (рис. 1.6). Повышение помехоустойчивости при этом достигается за счет того, что оба проводника обычно прокладываются на печатной плате параллельно, поэтому помехи, наведенные на один из них, будут практически в том же виде наведены и на другой. В конечном счете разность потенциалов между ними останется примерно одинаковой, т. е. определяемой микросхемой-источником.

Применение интерфейса LVDS требует специальных микросхем: драйвера (источника) и приемника. Также можно обратить внимание, что для микросхемы-приемника используется резистор, вклю-

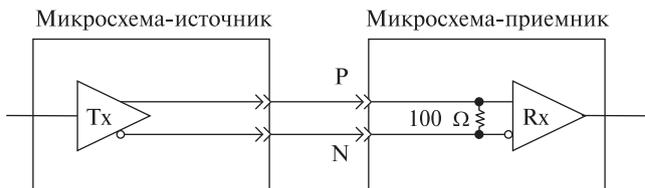


Рис. 1.6. Соединение микросхем с помощью интерфейса LVDS

ченный между линиями. В соответствии со стандартом его номинал составляет 100 Ом, поэтому LVDS при работе потребляет относительно большую мощность по сравнению с КМОП. Микросхемы ПЛИС имеют возможность конфигурировать выводы как однопроводные (КМОП, ТТЛ) или дифференциальные (LVDS и др.). Тем не менее произвольное образование дифференциальных пар в ПЛИС не допускается, каждому выводу P соответствует определенный вывод N (обычно расположенный рядом).

Приведенные выше сведения о схемотехнике цифровых устройств являются сильно упрощенными и не отражают целый ряд вопросов. Например, уменьшение норм технологического процесса и размеров транзисторов приводит к тому, что сопротивление диэлектрика между затвором и каналом также уменьшается, что приводит к резкому росту утечек тока. Не рассмотрено также поведение транзисторных ключей в динамическом режиме, подключение нагрузки к выходам, особенности технологических процессов изготовления цифровых устройств и многое другое. Для подробного изучения схемотехники цифровых элементов следует обратиться к специальной литературе.

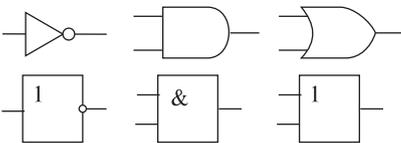


Рис. 1.7. Базовые логические элементы: логическое НЕ (инвертор), логическое И, логическое ИЛИ. Верхний ряд — изображение по ИЕЕЕ, нижний ряд — по ЕСКД

Набор базовых логических элементов приведен на рис. 1.7.

Логические элементы называются также вентилями (в англоязычной литературе *gate*). Необходимо обратить внимание, что таким же термином обозначается и затвор полевого транзистора, хотя смысловая нагрузка в этих случаях различна.

Удобным способом описания работы логических вентиляей является составление таблиц истинности. В такую таблицу записывается состояние выхода в зависимости от комбинации входных сигналов. Для элементов, показанных на рис. 1.7, таблицы истинности будут выглядеть так, как показано в табл. 1.1.

Число входов для логических элементов И и ИЛИ может быть больше двух. При этом сохраняется принцип работы, показанный в табл. 1.1: на выходе элемента И логическая 1 появляется только в

Инвертор не является единственным устройством для преобразования логических сигналов. Более того, интерес представляют как раз не одноходовые, а многоходовые устройства, позволяющие сформировать сигнал, представляющий собой сложную функцию от состояния

Таблица 1.4

Таблица истинности для элемента
2И-НЕ с инвертированными входами

A	B	Q
0 (1)	0 (1)	0
0 (1)	1 (0)	1
1 (0)	0 (1)	1
1 (0)	1 (0)	1

Таблица 1.3

Таблица истинности для элемента
2И-НЕ с объединенными входами

A	B	Q
0	0	1
1	1	0

венно. Для упрощения графического изображения символ инверсии (небольшой кружок) размещают непосредственно на выходе такого элемента. Часто в обозначении элемента указывается и число его входов. Например, 3ИЛИ соответствует трехвходовому элементу ИЛИ.

Символы инверсии могут быть размещены и на входах логических элементов. В этом случае их работу необходимо анализировать так, как если бы сигнал на этот вход подавался через элемент НЕ.

Рассмотрим двухвходовой элемент И-НЕ (2И-НЕ). Если объединить его входы, то таблицу истинности можно представить так, как показано в табл. 1.3. Из этой таблицы следует, что элемент 2И-НЕ в данном случае выполняет функцию инвертора.

Соединив элементы 2И-НЕ и инвертор, созданный из такого же элемента с объединенными входами, получаем элемент 2И.

Наконец, вводя инверторы для входных сигналов, получаем следующую таблицу истинности (в скобках показаны значения инвертированных сигналов, которые в действительности воспринимаются элементом 2И-НЕ).

Итак, для элемента 2И логическая единица на выходе должна появиться только в том случае, если оба сигнала, стоящие в скобках, равны 1. В остальных случаях на выходе присутствует логический ноль. С учетом инвертора, стоящего на выходе, получаем результаты, показанные в табл. 1.4. Нетрудно убедиться, что они соответствуют логике работы элемента 2ИЛИ.

Таким образом, имея достаточное число элементов 2И-НЕ, можно реализовать любой из элементов И, ИЛИ, НЕ. Поэтому элемент 2И-НЕ может выступать в роли некоторого эквивалента сложности комбинационной схемы. Аналогичные рассуждения можно провести и для элемента 2ИЛИ-НЕ, с помощью которого также реализуются любые схемы комбинационной логики. Элементы 2И-НЕ (называемый также базисом Шеффера) и 2ИЛИ-НЕ (базис Пирса) являются эквивалентными логическими вентилями (*equivalent logic gates*), с

помощью которых можно оценить сложность того или иного цифрового устройства.

Особенностью работы представленных логических элементов является то, что соответствующий таблице истинности сигнал на их выходе появляется с незначительной задержкой относительно смены состояний на входе. Эта задержка обусловлена только процессами переключения элементов схемы и при рассмотрении идеальных элементов считается, что она равна нулю. Таким образом, изменение уровня выходного сигнала может произойти в любой момент времени, поэтому такие схемы относят к классу асинхронных. Однако более точным названием, отражающим использование логических вентилях, состояние которых определяется только уровнем входного сигнала, является термин «комбинационная логика». Действительно, для точного определения состояния выхода такой схемы достаточно знать комбинацию состояний ее входов. При этом предыстория их изменений не играет роли. Существуют также устройства, состояние которых может также измениться в любой момент времени, но определяется не только текущим состоянием входов, но и историей изменения этих состояний.

На базе устройств комбинационной логики возможно решение многих задач вычислительной техники. В частности, с их помощью производится выполнение базовых арифметических операций.

Рассмотрим операцию сложения, выполняемую над числами, представленными в двоичной системе счисления. В этой системе возможны всего четыре варианта: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = 10$ (2_{10}). Для представления числа 2, которое в двоичном виде запишется как 10, требуется уже два разряда.

Запишем таблицу истинности для элемента, устанавливающего выход в соответствии с младшим разрядом результата сложения двух чисел, представленных в двоичной системе (табл. 1.5).

В последнем случае в таблице записан ноль, поскольку произошел перенос в следующий разряд. Из таблицы видно, что операция сложения реализуется элементом ИСКЛЮЧАЮЩЕЕ ИЛИ.

Для представления числа, переносимого в следующий разряд, потребуется еще один выход. Нетрудно убедиться, что значение этого выхода равно 1 только в том случае, когда оба входа равны 1 — сигнал переноса, таким образом он формируется элементом И.

Таблица 1.5
Таблица истинности
одноразрядного сумматора:
результат сложения
чисел A и B

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

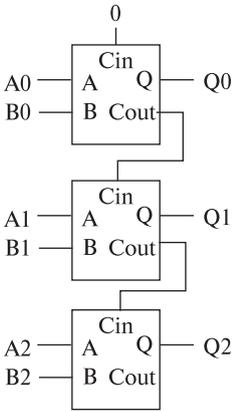


Рис. 1.9. Многоразрядный сумматор

Таблица 1.6
Таблица истинности одного разряда
многоразрядного сумматора

A	B	Cin	Q	Cout
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

При сложении многоразрядных двоичных чисел (рис. 1.9) необходимо учитывать сигналы переноса, приходящие из предыдущих разрядов, и формировать такие сигналы для последующих разрядов. С учетом сигнала переноса каждый элемент сумматора, начиная со второго, должен иметь три входа вместо двух. Обозначим вход переноса как Cin, а выход как Cout и составим таблицу истинности для такого блока (табл. 1.6).

При составлении таблицы истинности использовалось свойство коммутативности сложения: поскольку порядок слагаемых не влияет на результат, можно ориентироваться на то, что нечетное число единиц в трех входных сигналах обуславливает появление единицы на выходе Q, а выход Cout принимает значение 1 в тех случаях, когда два или три входа находятся в состоянии единицы. Можно подобрать несколько вариантов реализации устройства комбинационной логики, реализующего сложение.

Элементы комбинационной логики реализуются цифровыми микросхемами малой степени интеграции. Это серии 155, 555, 1533 и другие российского производства и их аналоги 74хх, выполненные по технологии ТТЛ, а также КМОП-устройства серий 176, 561 и 54хх соответственно. В составе этих серий присутствует большое число элементов рассмотренных выше типов с различным числом входов.

Схема, показанная на рис. 1.9, называется сумматором с последовательным переносом (*ripple-carry adder*) и не является единственным вариантом реализации сумматора. Однако она представляет интерес как своей простотой, так и тем, что в рассматриваемых в данной книге микросхемах FPGA имеются аппаратные реализации именно такой схемы.

В проектах для FPGA не следует реализовывать сумматоры путем описания их схемы в соответствии с рис. 1.9. В этих микросхемах существуют специальные цепи для реализации переноса, которые эффективно используются САПР. Самостоятельное повторение схемы сумматора приведет к менее эффективной реализации.

В противоположность асинхронным, синхронные цифровые устройства изменяют состояние выходных сигналов в строго определенные моменты времени. Эти моменты, как правило, соответствуют положительному перепаду специального синхронизирующего сигнала, называемого проще тактовым сигналом. Для описания работы синхронных устройств может использоваться такое представление, как временные диаграммы работы. Они представляют собой графики зависимости логических уровней от времени, на которых показывают типичные комбинации и события (перепады уровня) на входных линиях, а также соответствующие изменения выходных сигналов. Базовым синхронным элементом является D-триггер. Его графическое изображение и временные диаграммы работы показаны на рис. 1.10.

На рис. 1.10 видно, что выход триггера Q принимает то же состояние, что было на входе данных D в момент положительного перепада (фронта) тактового сигнала. В остальные моменты времени изменения логического уровня на входе D никак не влияют на состояние выхода. Такое поведение позволяет, в частности, использовать D-триггер в качестве устройства хранения данных. После подачи на вход D требуемого уровня сигнала и запоминания его в триггере по фронту тактового сигнала значение с входа данных D может быть убрано. При этом возникает проблема: как добиться того, чтобы последующие фронты тактового сигнала не переписывали запомненное в триггере значение? Одним из способов является формирование вспомогательного тактового сигнала с помощью элемента 2И. На первый из входов этого вентиля подается собственно тактовый сигнал, а на второй — сигнал разрешения. Обращаясь к таблице истинности 2И, нетрудно убедиться, что если на втором входе будет присутствовать сигнал логического нуля, то выход этого элемента будет оставаться в нуле при любом уровне тактового сигнала. Если же на вход разрешения подан сигнал логической единицы, то состояние выхода пол-

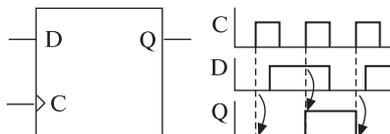


Рис. 1.10. Графическое изображение и временные диаграммы работы D-триггера

ностью определяется состоянием входа, на который подан тактовый сигнал. Таким образом, элемент 2И может запрещать прохождение тактового сигнала на вход С триггера в те периоды времени, когда обновление его состояния не требуется.

В ряде случаев введение в проект вспомогательных вентилях, запрещающих прохождение тактового сигнала, ведет к труднообнаружимым ошибкам. Их источником является тот факт, что тактовый сигнал, прошедший через логический вентиль, оказывается задержанным относительно исходного сигнала на время, требуемое для переключения внутренних схем элемента 2И. Это приводит к тому, что фронт тактового сигнала приходит на разные триггеры в разные моменты времени и часть триггеров проекта, срабатывая раньше, может исказить логические сигналы, требующие записи в остальные триггеры. Этот неприятный эффект носит название «гонки фронтов», и его следует всячески избегать. Для исключения записи в неподходящие моменты времени в состав триггеров (и других синхронных узлов) вводят специальный вход разрешения записи *CE* (от *Clock Enable* — «разрешение тактового сигнала»). Перезапись состояния триггера происходит только в том случае, если в момент прихода фронта тактового сигнала на входе *CE* присутствовал высокий логический уровень. Правильное формирование тактового сигнала внутри ПЛИС должно быть обеспечено с помощью специальных компонентов — аппаратных формирователей и тактовых сетей.

Необходимо иметь в виду, что действительный уровень сигнала, разрешающего работу синхронных устройств, может быть как логическим нулем, так и логической единицей. В схемотехнике ТТЛ существует целый ряд соображений, по которым некоторые сигналы считаются активными, если их уровень соответствует логическому нулю. Такие сигналы обозначают символом инверсии (небольшой кружок на входе, аналогично выходу инвертора), чертой над именем сигнала на его графическом изображении, а в текстовых описаниях — символами \sim , #, n перед именем сигнала (например, $\sim CE$, # CE , nCE). Уровень сигнала, при котором соответствующий вход считается влияющим на работу устройства, называют активным уровнем сигнала. Пример изображения D-триггера с входом разрешения тактового сигнала, имеющим активный низкий уровень, приведен на рис. 1.11.

Аналогично активный низкий уровень можно назначить и другим входам и выходам цифровых устройств. Например, если инвертировать тактовый вход D-триггера, то перезапись выхода Q будет