

Электроника

Павел Кириченко

bhv®



ЦИФРОВАЯ ЭЛЕКТРОНИКА для начинающих

- Теория и практика цифровой электроники
- Разнообразные устройства на макетной плате
- Схемы, идеи и конструкции
- Электронные игрушки
- Цифровые сигналы и двоичная система счисления
- МОП-транзисторы и логические микросхемы
- Устройство и работа основных частей микропроцессора



Материалы
на www.bhv.ru

Павел Кириченко

ЦИФРОВАЯ ЭЛЕКТРОНИКА ДЛЯ НАЧИНАЮЩИХ

Санкт-Петербург
«БХВ-Петербург»
2019

УДК 621.3
ББК 32.85
К43

Кириченко П. Г.

К43 Цифровая электроника для начинающих. — СПб.: БХВ-Петербург, 2019. — 176 с.: ил. — (Электроника)

ISBN 978-5-9775-4010-0

Основы цифровой электроники изложены простым и доступным для начинающих способом — путем создания на макетной плате забавных и познавательных устройств на транзисторах и микросхемах, которые сразу после сборки начинают работать, не требуя пайки, наладки и программирования. Набор необходимых деталей сведен к минимуму как по количеству наименований, так и по стоимости. По ходу изложения даются вопросы для самопроверки и закрепления материала, а также творческие задания на самостоятельную разработку схем. Рассмотрены цифровые сигналы и двоичная система счисления, простейшие схемы на МОП-транзисторах, устройства цифровой логики на транзисторах и микросхемах, комбинационные и последовательностные схемы, сдвиговые регистры и счетчики.

Книга дает необходимые и достаточные знания о работе аппаратной части современных микропроцессоров и микроконтроллеров.

Для интересующихся электроникой

УДК 621.3
ББК 32.85

Группа подготовки издания:

Руководитель проекта	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капалыгина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн обложки	<i>Марины Дамбиевой</i>

Подписано в печать 31.08.18.

Формат 70×100^{1/8}. Печать офсетная. Усл. печ. л. 14,19.

Тираж 1300 экз. Заказ № 7489.

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Отпечатано с готового оригинал-макета

ООО "Принт-М", 142300, М.О., г. Чехов, ул. Полиграфистов, д. 1

Оглавление

Предисловие	7
Что ждет читателя в книге: теория, практика, задачи и идеи для самостоятельного творчества	7
Электронный архив	9
Благодарности	9
Глава 1. Двоичная система в цифровой электронике	11
Числа, состоящие только из нулей и единиц.....	11
Такие разные электрические сигналы.....	14
Как перевести электрические сигналы на язык цифр?	16
Ответы на вопросы для самопроверки.....	17
Глава 2. Первые схемы на МОП-транзисторах	19
Транзистор — что это такое?	19
Что потребуется для сборки схем из этой книги?	22
Как выбрать подходящие для экспериментов транзисторы и другие детали?	24
Первая практическая схема, которая помогает увидеть нули и единицы	28
Игра «Кто быстрее?» на двух транзисторах	33
Ответы на вопросы для самопроверки.....	35
Ответы на задания по конструкциям для самостоятельной разработки	35
Глава 3. Цифровая логика на транзисторах	37
Собираем элемент, который переворачивает сигналы наоборот	37
Основные логические схемы: И, ИЛИ и все-все-все	40
Логическое И.....	41
Логическое ИЛИ	42
Вентиль ИЛИ-НЕ	42
Вентиль И-НЕ.....	43
Исключающее ИЛИ и Исключающее ИЛИ-НЕ	44
Конструирование всех базовых логических элементов: от таблиц истинности к схемам на транзисторах	45
Практическая схема ИЛИ-НЕ на транзисторах.....	45
Алгоритм создания цифровых логических схем на транзисторах.....	47

Практическая схема вентиля И-НЕ на транзисторах	48
Практические схемы вентиляей И и ИЛИ на транзисторах	48
Практическая схема Исключающего ИЛИ на транзисторах	49
Добавляем входы в логические схемы	51
Создание логического вентиля по произвольной таблице истинности	52
Ответы на вопросы для самопроверки	54
Ответы на задания по конструкциям для самостоятельной разработки	55
Глава 4. Микросхемы цифровой логики	57
Общежитие для транзисторов: что важно знать о микросхемах цифровой логики	57
Приступаем к опытам с микросхемами	60
Соединим вместе входы логических элементов	60
Добавим недостающие входы	61
Инвертируем входы	65
Что делать с ненужными входами?	67
Кодовый замок: от идеи до готовой конструкции	68
Замысел	68
Приступаем к конструированию схемы	69
Доработка схемы: экономим энергию	70
Оборудуем замок сигнализацией	71
Игра «Угадай число» на базе схемы кодового замка	72
Ответы на вопросы для самопроверки	73
Ответы на задания по конструкциям для самостоятельной разработки	74
Глава 5. Основные комбинационные схемы	75
Как процессор выполняет вычисления: конструируем сумматор	75
Тайны двоичных кодов: шифраторы и дешифраторы	81
Схема дешифратора для клавиатуры кодового замка	81
Схема дешифратора	84
Учим цифровые схемы показывать десятичные числа	87
Мультиплексоры и демультимплексоры — регулировщики цифрового движения	93
Практическая схема мультиплексора	93
Демультимплексор	95
Транзисторные мультиплексоры	96
Ответы на вопросы для самопроверки	98
Глава 6. Последовательностные схемы	101
Два пишем, три в уме: как запомнить результат вычислений?	101
Практическая схема RS-триггера на вентилях И-НЕ	104
Входить только по сигналу: как в триггерах разрешают и запрещают запись данных?	106
Практическая схема тактируемого триггера	107
Триггеры без запрещенных состояний	110
Приказано задержать! D-триггер	113
Двухступенчатые триггеры: теория и практическая схема	114
T-триггер, или как посчитать импульсы?	118
Практическая схема D-триггера на транзисторах	119
От одного триггера к целому регистру, и зачем процессору конвейер	122

Ответы на вопросы для самопроверки.....	124
Ответ на задание по конструкции для самостоятельной разработки	125
Глава 7. Сдвиговые регистры и счетчики	127
Эстафета для триггеров: конструкция сдвигового регистра.....	127
Забавный светильник «Бегущие огни».....	129
Считалочка для цифровых схем	132
Схема простейшего счетчика на Т-триггерах	132
Вычитающий счетчик.....	134
Как научить двоичный счетчик десятичной системе?	135
Две схемы электронного кубика для игр	137
Синхронные и асинхронные счетчики.....	142
Ответы на вопросы для самопроверки.....	144
Ответ на задание по конструкции для самостоятельной разработки	144
Глава 8. Макет оперативного запоминающего устройства.....	145
Как устроена память цифрового мозга?	145
Собираем память на макетной плате	150
Заключение.....	157
ПРИЛОЖЕНИЯ	159
Приложение 1. Список дополнительной литературы для изучения электроники и двоичной логики.....	161
Приложение 2. Цоколевки транзисторов и микросхем, применяемых в схемах этой книги	163
Приложение 3. Цветовая маркировка резисторов	167
Приложение 4. Сводная таблица используемых компонентов.....	169
Приложение 5. Аналоги импортных микросхем серии CD4000BE	171
Предметный указатель	173

Предисловие

Что ждет читателя в книге: теория, практика, задачи и идеи для самостоятельного творчества

Современная жизнь немыслима без огромного количества электронных устройств, окружающих нас. Кажется, не осталось ни одной области деятельности, ни одного предмета, в которые бы не внедрили или не попытались внедрить какую-нибудь электронику. Инженеры постоянно изобретают и разрабатывают новые устройства, которые позволяют людям приобрести ранее недоступные возможности: от ночного зрения до сверхбыстрых вычислений и сверхточных перемещений. Многие уже в школьном возрасте не только пользуются электронными приборами, но и сами пишут для них программы. Так, в наши дни большую популярность приобрели занятия робототехникой. Доступность микроконтроллеров, а также разнообразных датчиков и приводов к ним, позволяет ученикам создавать самые неожиданные и оригинальные конструкции роботов, которые удивляют взрослое жюри конкурсов и олимпиад.

Но активно пользоваться чем-то не означает понимать, как оно устроено и работает. Для многих детей и взрослых микроконтроллер остается неким магическим черным ящиком, на который они подают определенные воздействия, а внутри него каким-то образом рождаются необходимые отклики в соответствии с программой, записанной в его память. Многим хотелось бы пойти дальше в изучении техники и понять, как же подаваемые данные обрабатываются внутри этой загадочной многоножки на плате в работе, или в калькуляторе, или еще в чем-нибудь.

Существует большое количество книг, которые в доступной форме объясняют принцип действия самых разных электронных схем. Но когда речь заходит об обработке цифровых данных, рассказ начинается сразу с базовых логических элементов — без объяснения, как же они устроены и функционируют. Такие объяснения, видимо, считаются сложными и откладываются на потом, на период обучения в высших учебных заведениях. Автор этой книги считает, что это не совсем верный подход. Для создания целостной картины и глубокого понимания принципов работы цифровых схем можно в доступной и детям, и взрослым форме рассказать все с самого начала, т. е. с главного элемента любой микросхемы — транзистора. И не

только рассказать, но и помочь самостоятельно собрать самые разные и интересные цифровые схемы. Например, вы узнаете, как устроен электронный кодовый замок. Разберетесь в схеме бегущих огней и переключателей елочных гирлянд. Сможете создать цифровой кубик с индикатором на замену обычному для настольных игр. Эти и многие другие устройства рассматриваются от идеи до практической реализации без единой строки программного кода, сложных вычислений и компьютерного моделирования. При этом современные технологии позволяют обойтись и без умения паять. Все предлагаемые конструкции были в обязательном порядке проверены на практике, и читатель может собрать их сам на своем столе, провести с ними самые разнообразные эксперименты и увидеть, что — при должном терпении и внимательности — никаких загадок и тайн в мире цифровой электроники для него не останется.

Сразу надо сказать, что в этой книге не будет ни одной описывающей работу транзистора сложной формулы на полстраницы. Во-первых, базовые принципы, изложенные в доступной и учащемуся старших классов форме, достаточны для разработки огромного количества разнообразных цифровых схем. Во-вторых, те уравнения, за незнание которых профессор ставит студенту «неуд» на экзамене, уже давно не позволяют вычислять с необходимой точностью токи и напряжения во время работы транзистора в микросхемах, изготовленных по современным технологиям. В моделях нынешних полупроводниковых приборов учитывается не одна сотня параметров. Поэтому все устройства — от небольших исследовательских до дорогих коммерческих — сегодня немислимо разработать без применения мощных систем автоматизированного проектирования (САПР), которые способны рассчитать сложнейшие уравнения для многих тысяч и даже миллионов транзисторов, содержащихся в микросхемах. Конечно, формулы, которые изучают в университетах, дают более глубокое понимание работы схем. И грамотный специалист хорошо их знает. Но сложные уравнения не требуются для решения главной задачи этой книги — обучения тому, как самостоятельно создавать свои первые цифровые устройства и иметь ясное представление об их работе. Тем не менее предполагается, что читатель уже знаком с основными понятиями электротехники, и ему не нужно объяснять, что такое электрический ток, напряжение и сопротивление. И закон Ома для него не загадка, так же как и последовательные и параллельные электрические цепи. Больше для освоения теоретического материала этой книги ничего не потребуется.

Книга станет полезным подспорьем и руководителям робототехнических кружков, которые желают дать своим ученикам более обширные познания. В начале каждой главы, содержащей практические примеры, приведен перечень необходимых деталей для всех рассмотренных схем. Разнообразие используемых компонентов сведено к минимуму. При этом они выбирались максимально доступными по цене самому широкому кругу читателей. В каком магазине или на интернет-сайте их приобрести, каждый решит сам, исходя из своих условий. Современное развитие интернет-торговли позволяет купить самые разнообразные детали, инструменты и материалы. Нужно активно пользоваться этой возможностью! Также к каждой главе обязательно дается набор теоретических вопросов или практических заданий для

самопроверки с ответами. А в *приложении 1* к книге приведен список литературы для тех, кто желает глубже изучить и понять электронику.

Электронный архив

Поскольку книга выходит в черно-белом варианте, а для лучшего понимания ряда моментов некоторые приведенные в ней схемы хорошо бы увидеть в цвете, комплект необходимых цветных иллюстраций приведен в сопровождающем книгу электронном архиве, который можно загрузить с FTP-сервера издательства «БХВ-Петербург» по ссылке <ftp://ftp.bhv.ru/9785977540100.zip> или со страницы книги на сайте www.bhv.ru.

Благодарности

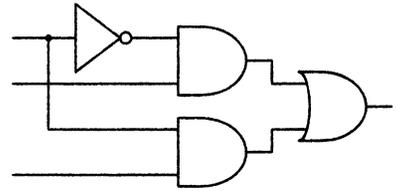
Автор выражает благодарность своей семье: родителям, жене и сыновьям — тем, кто всегда вдохновлял его на поиски нового, поддерживал во всех начинаниях и верил в успех. Без их заботы и терпения ничего бы не получилось.

Педагоги и руководители детских летних образовательных лагерей (МКШ в Дубне и GoTo Camp в Москве) дали автору возможность поделиться своими знаниями с талантливыми подростками, с которыми было интересно проводить теоретические и практические занятия. Нельзя не признать, что выступать перед такой аудиторией гораздо сложнее, чем на «взрослой» конференции. Это был бесценный опыт, в результате которого и появилась идея написать книгу.

Безусловно, спасибо издателю, который поверил в автора и выпустил эту книгу в свет.

И огромная благодарность всем читателям за то, что приобрели книгу в свою библиотеку или в подарок друзьям. Надеюсь, она вас не разочарует.

ГЛАВА 1



Двоичная система в цифровой электронике

Числа, состоящие только из нулей и единиц

Прежде чем приступить к изучению и конструированию цифровых схем, необходимо познакомиться с двоичной системой счисления и соответствующими ей электрическими сигналами. Даже тем читателям, которые, как принято говорить, «в теме», рекомендуется прочесть эту короткую главу — не исключено, что вы тоже узнаете что-то новое.

Как известно, в жизни мы используем позиционную десятичную систему счисления. Существуют десять цифр: от 0 до 9, которые пишутся в нужную позицию, чтобы сформировать требуемое число. Это делается по следующим правилам:

- цифра в самой правой позиции указывает количество единиц в числе, т. е. умножается на $10^0 = 1$;
- цифра на второй позиции справа определяет количество десятков и умножается на $10^1 = 10$, на третьей — на $10^2 = 100$ и т. д.;
- затем результаты всех умножений складываются.

В соответствии с этими правилами, например, запись 984 раскладывается на $9 \cdot 10^2 + 8 \cdot 10^1 + 4 \cdot 10^0 = 900 + 80 + 4 = 984$.

Это всем объясняли в школе на уроках математики, а если кто уже вдруг успел забыть, то наверняка вспомнил.

А на уроках информатики, вероятно, вам рассказывали, что в вычислительной технике десятичная система неудобна, зато удобна двоичная. В ней используется всего две цифры: 0 и 1. И они образуют числа по тому же принципу, что и в десятичной системе, только вместо степени десяти для вычисления значения нужной позиции используется степень двойки. Например, двоичная запись 101 означает $1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 4 + 0 + 1 = 5$. В табл. 1.1 показан процесс перевода первых двенадцати десятичных чисел в двоичный код.

Общий принцип перевода из десятичной системы в двоичную таков: находим максимально близкую к десятичному числу степень двойки, которая будет меньше или равна исходному десятичному числу. И записываем единицу в соответствующий

Таблица 1.1. Преобразование десятичных чисел в двоичные

Десятичное число	Пересчет в двоичный код	Двоичное число
0	0×2^0	0
1	1×2^0	1
2	$1 \times 2^1 + 0 \times 2^0$	10
3	$1 \times 2^1 + 1 \times 2^0$	11
4	$1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$	100
5	$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$	101
6	$1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$	110
7	$1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$	111
8	$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$	1000
9	$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$	1001
10	$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$	1010
11	$1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$	1011

разряд двоичного числа. Затем повторяем эту операцию для разности исходного числа и найденной степени двойки до тех пор, пока не дойдем до последнего разряда. Вот, скажем, число девять. Ближайшая степень двойки — это 8, два в кубе. Значит, пишем в четвертую позицию единицу (помним, что первая позиция соответствует двойке в нулевой степени, а куб — третья степень, т. е. четвертая позиция справа). Остаток — единица — не делится ни на два в квадрате ни просто на два, значит, пишем нули в третьей и второй позициях. А вот двойке в нулевой степени остаток как раз равен, поэтому в первой позиции в двоичной записи появляется единица. Так и выполняется перевод из одной системы в другую самым очевидным, но не самым удобным способом. Перевод из двоичной системы в десятичную проще: умножаем единицы в записи числа на соответствующие им степени двойки и складываем все произведения. Программисты и разработчики цифровой электроники наизусть назовут, даже если их ночью разбудить, первые десять степеней двойки: 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 — потому что эти числа постоянно используются в работе. Полезно их выучить для быстрого пересчета из двоичной системы в десятичную и обратно.

Менее очевидный, но более простой по применению вариант перевода из десятичной системы в двоичную таков. Исходное десятичное число делим на 2. Какой получится остаток от деления — 0 или 1 — то число и записываем в первый (самый правый) разряд двоичного числа. Затем частное снова делим на 2 и остаток записываем во второй разряд. И так продолжаем до тех пор, пока очередное частное не станет равным 0. Для примера переведем десятичное число 10 в двоичное этим способом:

$$\square 10 : 2 = 5 \text{ с остатком } 0;$$

$$\square 5 : 2 = 2 \text{ с остатком } 1;$$

□ $2 : 2 = 1$ с остатком 0;

□ $1 : 2 = 0$ с остатком 1.

Результат в двоичной системе будет записан так: 1010.

Вопросы для самопроверки

1. Переведите десятичные числа 35, 13, 47, 127, 25, 211 в двоичные любым удобным вам способом без применения электронных вычислительных средств.
2. Переведите двоичные числа 1000101, 1110, 10110, 11100101, 10101, 101101 в десятичные без применения электронных вычислительных средств.

* * *

Поскольку самый правый разряд числа имеет самое маленькое значение по величине, то его называют *младшим значащим разрядом* (сокращенно — МЗР). Еще про него говорят, что он имеет наименьший вес в числе. А самый левый разряд «весит» больше остальных, поэтому он именуется *старшим значащим разрядом*, или СЗР. То есть в двоичном числе 1000 младший значащий разряд равен нулю, а старший значащий разряд — единице. А в десятичном числе 357 МЗР равен 7, а СЗР — 3.

Один разряд двоичного числа называется *битом*. Число, содержащее в себе восемь битов, называется *байтом*. Легко сосчитать, что одним байтом можно закодировать числа от 0 до 255. Применяется в технике и байтовая запись — в ней число 5 выглядит как 00000101, а 15 представляется как 00001111, т. е. нули в старших неиспользуемых разрядах не отбрасываются. Если нужны еще большие значения, то используют несколько байтов.

Столь длинная запись на практике неудобна, поэтому существуют такие системы счисления, как восьмеричная (использует только цифры от 0 до 7) и шестнадцатеричная. В последней задействованы все цифры десятичной системы от 0 до 9, а к ним еще добавляются буквенные обозначения чисел $10 = A$, $11 = B$, $12 = C$, $13 = D$, $14 = E$, $15 = F$. Обе эти системы являются компромиссом между привычной нам десятичной и применяемой в цифровой электронике двоичной. Но физической реализации в устройствах они не имеют и служат только для удобства записи, поэтому здесь нет смысла останавливаться на них подробнее.

Все программисты по своей работе хорошо знают двоичную систему. Зато мало кто из программистов и даже не все инженеры-электронщики уверенно назовут больше одной причины, почему такая громоздкая система была избрана для вычислительной техники. А она и правда совсем не компактная — достаточно взглянуть на таблицу перевода первых двенадцати чисел. Пока в десятичной системе запись остается в пределах одного разряда для чисел от 0 до 9, в двоичной используются уже четыре разряда. Это значит, что для обмена данными между устройствами будет нужно передать либо четыре цифры поочередно по одному проводу, либо по одной цифре одновременно по четырем проводам. А если двоичное число содержит 64 разряда? Чтобы понять, чем же приглянулась такая система разработчикам цифровых схем, нужно разобраться, как происходит обмен данными в различных устройствах.

Такие разные электрические сигналы

Внутри электронной системы мы можем передавать информацию от одного устройства к другому при помощи электрических сигналов. Они бывают самыми разными. Например, акустические колонки издают звук за счет сжатия или разрежения воздуха с помощью подвижного элемента динамика — мембраны или диффузора. Сигнал, подаваемый на динамик и заставляющий его мембрану вибрировать, непрерывно меняется, пока играет музыка. И если с помощью специального прибора — осциллографа — вывести на экран форму этого сигнала, чтобы посмотреть, как он меняется во времени, то можно будет увидеть примерно такую картину, как на рис. 1.1. На нем по направлению горизонтальной координатной оси отсчитывается время, а по вертикальной — напряжение на контактах динамика.

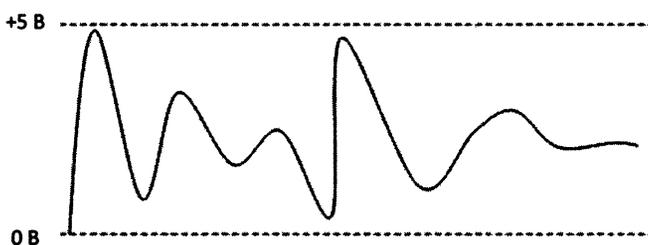


Рис. 1.1. Аналоговый сигнал

Этот сигнал представляет собой переменное по величине напряжение, подаваемое с аудиовыхода воспроизводящего устройства на динамик. Оно может меняться в пределах от минимального до максимального, задаваемого в устройстве источником питания и электрической схемой. Непрерывно и гладко меняющиеся во времени сигналы в электронике называют *аналоговыми*. Но с их помощью сложно придумать, как передавать числовые данные. Звук — пожалуйста, изменения температуры с термопары в виде напряжения на ее концах — запросто, уровень освещенности как силу тока от фотодатчика, включающего уличные фонари, — легко. Но изменения всех этих величин, даже очень сильные, не происходят мгновенно — всегда есть нарастание или спад на определенном отрезке времени. А одно число от другого при передаче между устройствами должно отличаться самым решительным образом, иначе сложно сказать, что за цифру мы считали, например, из карты памяти — еще семерку или уже восьмерку.

Поэтому для передачи цифровых данных используют сигналы, меняющиеся скачком между заданными крайними значениями, — *импульсные* сигналы. Так, можно условиться для двоичной системы считать низкий уровень напряжения 0 вольт нулем, а высокий, равный напряжению питания, — единицей. Тогда передать число 5 можно так, как показано на рис. 1.2.

Как просто оказывается передавать нужную последовательность цифр в двоичном коде — следует всего лишь задать правильную последовательность импульсов. И эту причину — простоту различения цифр 0 и 1 — обычно уверенно называют в ответ на вопрос, почему же используется именно двоичная система. Но эта причина

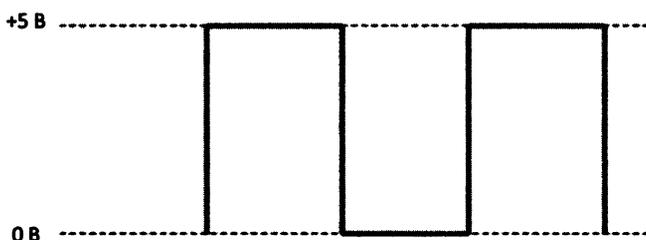


Рис. 1.2. Цифровой сигнал в теории

вместе с упомянутым ранее недостатком — длиной кода — делает выбор в пользу двоичных чисел не столь очевидным. Вот вы будете питать рассматриваемые в следующих главах схемы от батарейки «Крона», которая выдает напряжение 9 вольт между плюсом и минусом. Почему бы не поделить этот диапазон не на две, а на 10 равных частей по 0,9 вольт каждая и не передавать ступенчатый сигнал десятичного кода? Создать такую схему возможно. Но так не делают.

СИСТЕМЫ НА ТРОИЧНОМ КОДЕ

Правда, известна попытка создания в нашей стране вычислительного комплекса «Сетунь», работающего в троичном коде, т. е. в коде, основанном на степенях тройки. Но дальше небольшого количества образцов дело не пошло тоже. Хотя в Интернете вы можете найти сайты, посвященные и этой машине, и преимуществам троичного кода перед десятичным и двоичным.

Почему же двоичный код в технике вытеснил остальные? Дело в том, что в реальной вычислительной системе цифровые сигналы не выглядят такими идеальными прямоугольными импульсами, как их обычно изображают в книгах. Вот вам для примера на рис. 1.3 изображение двух сигналов, полученное моделированием реального устройства в микропроцессоре.

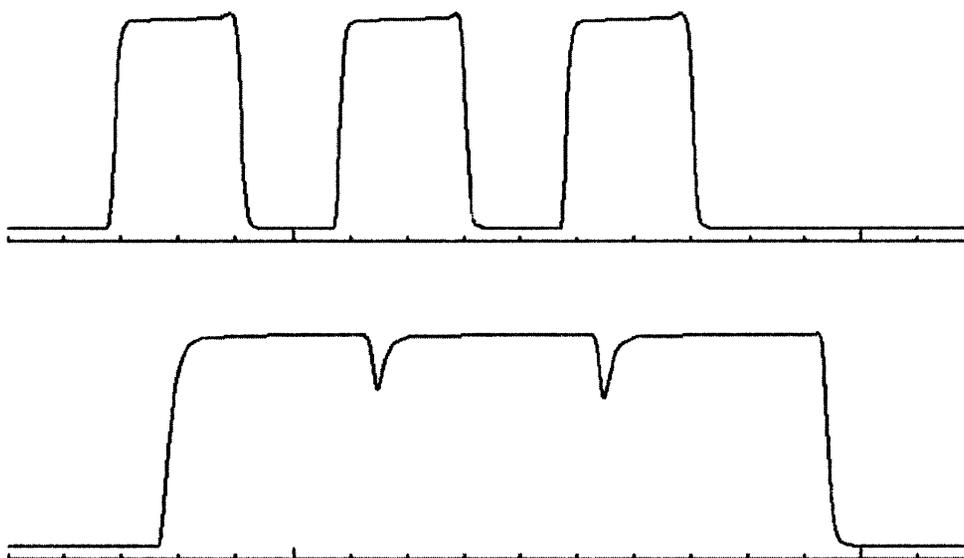


Рис. 1.3. Цифровые сигналы на практике

Форма отличается от красивых прямоугольных нулей и единиц с предыдущего рисунка, правда? В верхней серии импульсов хорошо видно, что не идеальная, а вполне определенная конечная скорость переключения сигнала из одного состояния в другое приводит к тому, что импульсы приобретают трапецевидную форму. На нижнем графике сигнал явно подвергается воздействию какой-то помехи, дважды искажающей его плоскую вершину и приводящей к внезапным периодическим спадам уровня напряжения примерно на четверть от максимальной. Большая нагрузка на какой-то элемент схемы способна замедлить скорость нарастания импульса так, что он больше становится похож на аналоговый сигнал, чем на цифровой. Да еще напряжение питания в современных микропроцессорах постоянно стремятся снижать для экономии потребляемой электрической мощности, и в передовых разработках оно уже давно меньше одного вольта. Если такой незначительный диапазон мы начнем делить еще на много уровней для передачи всех возможных цифр, то не сможем отличить пятерку от восьмерки — вся полезная информация потонет в искажениях и помехах, подобных рассмотренным. То есть на первый план выходит такая характеристика, как *помехоустойчивость*. И тут двоичному коду нет равных.

Как перевести электрические сигналы на язык цифр?

Если мы договоримся все напряжения ниже половины напряжения питания называть нулем, а выше — единицей, то такой код будет труднее всего исказить. Чтобы не путать двоичные нули и единицы с нулями и единицами вольт, используют названия «логический ноль» и «логическая единица». Еще их называют «низкий уровень» и «высокий уровень» — понятно почему. Строго говоря, вблизи середины полного диапазона напряжений есть некоторая область неопределенности. В ней узел цифровой схемы находится во время переключения из одного состояния в другое, когда нельзя сказать точно, произошел ли уже переход из высокого уровня в низкий и наоборот, или еще нет. Так, на рис. 1.4 области неопределенности, отмеченной серым цветом, соответствуют напряжения от 1,8 до 3 В, т. е. области логической единицы и логического нуля в этом примере несколько несимметричны относительно половины напряжения питания. Такое тоже бывает, симметрию никто гарантировать не может. Но если вы мысленно совместите этот рисунок с пре-

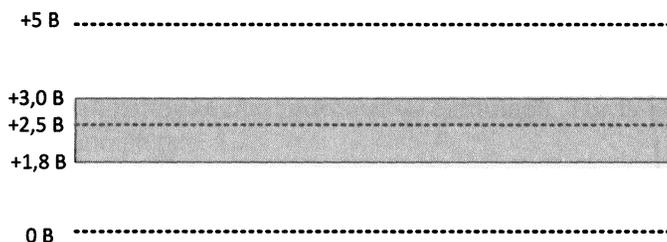


Рис. 1.4. Области нуля, единицы и неопределенности

дыдущими графиками реальных сигналов, то увидите, что никакие помехи не искажают данные, которые мы передаем, даже несмотря на существование области неопределенности. Логические нули остаются нулями, а единицы — единицами.

Вопросы для самопроверки

3. Пусть напряжение питания равно 9 вольт. Область неопределенности находится в интервале от 3,5 до 5,5 вольт. Какому цифровому сигналу соответствуют следующие напряжения: 0,1 вольт; 1,5 вольт; 7,8 вольт; 4,7 вольт; 5,6 вольт; 3,9 вольт?
4. Пусть напряжение питания равно 5 вольт. Область неопределенности расположена симметрично относительно половины напряжения питания и занимает 40% всего диапазона. Какие границы у области неопределенности?
5. Распределение областей логического нуля, логической единицы и области неопределенности такое, как на рис. 1.4. Сигнал логической единицы подвергся воздействию помехи и опустился вниз на 30% от номинального напряжения 5 В. Произошло ли искажение передаваемых цифровых сигналов?

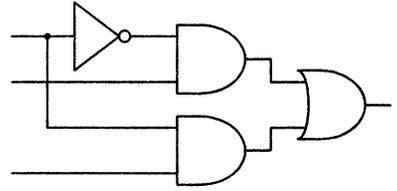
Цифровые схемы в силу того, что в реальности всегда существует некоторый дисбаланс характеристик элементов схемы и проводников, не могут сколь угодно долго находиться в области неопределенности. Обязательно произойдет случайное «сваливание» в ту или другую сторону, которое может вызвать волну переключений далее по цепочке элементов. Но грамотно спроектированная схема никогда не окажется в неизвестном разработчику состоянии, нарушив тем самым работу всей системы. Инженеры на этот случай предусматривают механизмы принудительного выставления нужной точной информации в ключевых точках схемы при включении питания и по сигналу сброса.

Теперь становится понятно, почему, несмотря на громоздкость двоичной системы, она применяется уже не одно десятилетие. Когда в технике возникнут какие-то принципиально новые способы передачи и хранения информации, то, возможно, двоичной системе придется подвинуться, уступив место какой-нибудь другой. Но пока что нет повода за нее волноваться, поэтому ее нужно хорошо изучить и применять на практике.

Ответы на вопросы для самопроверки

1. 100011, 1101, 101111, 1111111, 11001, 11010011.
2. 69, 14, 22, 229, 21, 45.
3. 0, 0, 1, не определено, 1, не определено.
4. 3,5 и 5,5 В.
5. 30% от 5 В равно 1,5 В. Значит, уровень сигнала опустился с 5 до 3,5 В. Верхняя граница области неопределенности равна 3 В. Искажения данных не произошло.

ГЛАВА 2



Первые схемы на МОП-транзисторах

Вам потребуются:

- 2 транзистора типа n-МОП;
- 1 транзистор типа p-МОП;
- 2 светодиода;
- 2 резистора сопротивлением 100 Ом;
- 2 резистора сопротивлением 1 кОм;
- 2 резистора сопротивлением 10 кОм;
- две кнопки.

Транзистор — что это такое?

Основной «кирпичик» при построении большинства цифровых микросхем даже самого невероятного масштаба — МОП-транзистор. Чтобы уверенно себя чувствовать при изучении цифровой электроники, достаточно простейшего понимания того, как он работает и для чего применяется в этих схемах. Так что же такое транзистор? Это сконструированный из особых материалов, называемых *полупроводниками*, электронный прибор, который при определенных условиях начинает пропускать через себя электрический ток. И в зависимости от изменения внешних воздействий ток через транзистор может становиться больше или меньше, а может и совсем прекратиться.

Транзисторы бывают самых разных видов, но мы из всего их многообразия будем рассматривать только один — МОП- (он же МДП-) транзистор. Эти три буквы расшифровываются как Металл-Оксид-Полупроводник (или Металл-Диэлектрик-Полупроводник) и описывают структуру прибора и материалы, из которых он изготовлен. В зарубежной литературе и справочных материалах МОП-транзисторы обозначаются как MOSFET (Metal-Oxide-Semiconductor Field Effect Transistor) или иногда просто MOS transistor. Бывают эти транзисторы двух типов: p-канальные и n-канальные. Полезно запомнить, как каждый тип транзисторов обозначается на

схемах. Наиболее часто встречаются два способа (рис. 2.1). В этой книге будут использоваться условные обозначения из верхнего ряда (поз. 1 и 2), как более простые и наглядные. Они очень часто применяются в научной литературе, где соответствие обозначений стандартам не так важно, как удобство восприятия. Более строгий технический стандарт приведен в нижнем ряду (поз. 3 и 4). Его можно встретить в справочных материалах и в официальных изданиях схем.

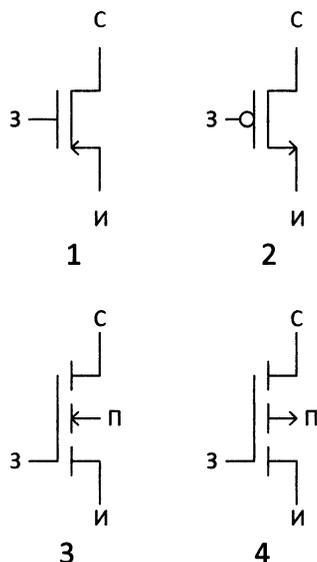


Рис. 2.1. 1 и 3 — п-канальный МОП-транзистор; 2 и 4 — р-канальный МОП-транзистор

Физика работы этих полупроводниковых устройств здесь рассматриваться не будет, чтобы не перегружать материал. Если эта тема вызывает у вас интерес, то в доступной форме ознакомиться с ней можно, например, в главах 1 и 2 книги М. Х. Джонса или в первых главах книги В. Г. Борисова (см. приложение 1). А для изучения основ цифровой электроники вполне достаточно представления о том, что МОП-транзистор — это электронный переключатель. Как же он работает? На условных обозначениях видно, что у него есть несколько выводов, к которым можно подключить что-нибудь. Эти выводы имеют вполне понятные названия: *исток*, *сток* и *затвор* (source, drain, gate в англоязычной литературе) и подписаны на рис. 2.1 первыми буквами своих названий. Транзистор устроен так, что ток может протекать между истоком и стоком лишь в том случае, когда на затвор подается нужное напряжение, отпирающее транзистор. И прекращает течь, когда напряжение на затворе запирает его. Между затвором и остальными двумя выводами ток практически не течет, и его величиной можно пренебречь. При приложении отпирающего напряжения в исходно плохо проводящем кристалле полупроводника как будто создается невидимый канал для электрического тока. Отсюда и произошли названия «п-канальный» и «р-канальный». А первая, латинская, буква определяет один из двух типов полупроводникового материала, использованных при создании транзистора.

МОП-транзистор р-типа отпирается напряжением, соответствующим логическому нулю. Об этом нам напоминает похожий на ноль кружок возле затвора в выбранном условном графическом обозначении этого транзистора. А п-канальный проводит ток, когда на его затвор подается логическая единица. Строго говоря, каждый из этих приборов начинает проводить ток, когда напряжение на затворе превысит значение такого важного параметра транзистора, который называется *пороговым напряжением*. Но со схемами, которые работают с напряжениями ниже пороговых напряжений, вы на протяжении всей книги дела иметь не будете. Во всех рассматриваемых примерах разница между уровнями логического нуля и логической единицы заведомо выше пороговых напряжений, поэтому при подаче нужного сигнала на затвор транзистора этот электронный переключатель будет гарантированно включен или выключен.

Есть у МОП-транзистора обязательно еще и четвертый контакт — *подложка* (bulk). Но в транзисторах, выпускаемых в виде отдельных деталей, — их называют *дискретными компонентами* — подложка намертво соединена с истоком. Поэтому отдельного вывода из корпуса у подложки нет. В транзисторах, находящихся внутри микросхем, — *интегральных транзисторах* — подложка обычно подключается к одному из контактов источника питания. То есть в любом варианте однозначно понятно, с чем соединена подложка, поэтому, чтобы не усложнять схемы, ее часто не изображают на условных обозначениях. В том числе и на том, которое будет применяться здесь далее.

Наверно, кто-то из читателей знает, что существуют еще *биполярные* транзисторы и *полевые* с р-п-переходом. Ранее для создания цифровых схем использовались и они (и кое-где до сих пор используются), но уже давно в этой области всех остальных с разгромным счетом победили МОП-транзисторы как наиболее технологичные и удобные в производстве для создания микросхем, содержащих миллионы и десятки миллионов этих компонентов. Поэтому далее приставка МОП или МДП будет опускаться, подразумевая, что только с такими транзисторами и будет идти работа на протяжении всей книги. Но на всякий случай на рис. 2.2 приведены условные обозначения других типов транзисторов. Встретив их в какой-либо схеме в литературе или в Интернете, вы будете понимать, что это не МОП-прибор, с которым вам довелось познакомиться в этой книге, и для понимания работы таких схем нужны дополнительные знания. Электроника интересная и разнообразная наука!

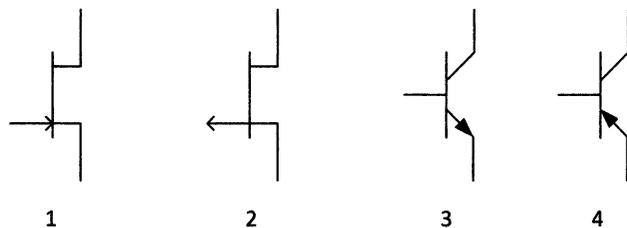


Рис. 2.2. Другие типы транзисторов: 1 — п-канальный полевой транзистор с р-п-переходом; 2 — р-канальный полевой транзистор с р-п-переходом; 3 — npn-биполярный транзистор; 4 — pnp-биполярный транзистор

Что потребуется для сборки схем из этой книги?

Для создания предложенных далее устройств желательно использовать *контактную макетную плату* (их еще называют breadboard) и *соединительные провода*. В таком случае можно обойтись совсем без пайки и получить возможность легко изменять и настраивать любую схему. Плата должна иметь не менее чем 800 контактных отверстий, т. к. наши схемы будут большими. А в последней главе рассматривается макет оперативного запоминающего устройства, для которого требуется еще более крупная плата или две по 800 отверстий в каждой. Поэтому макетную плату рекомендуется приобрести максимальную по размеру из доступных. На рис. Ц-2.3¹ приведен пример платы на полторы тысячи отверстий, на рис. Ц-2.4, а — набор жестких перемычек и на рис. Ц-2.4, б — набор гибких проводов с контактами-штырьками на обоих концах. Жесткие перемычки удобны тем, что не торчат подобно джунглям над деталями, размещенными на плате. Но их невозможно проложить поверх компонентов, что может усложнить соединения на плате. Гибкие провода могут подключаться произвольным способом, но при их большом количестве собранную схему становится сложнее анализировать. Поэтому,

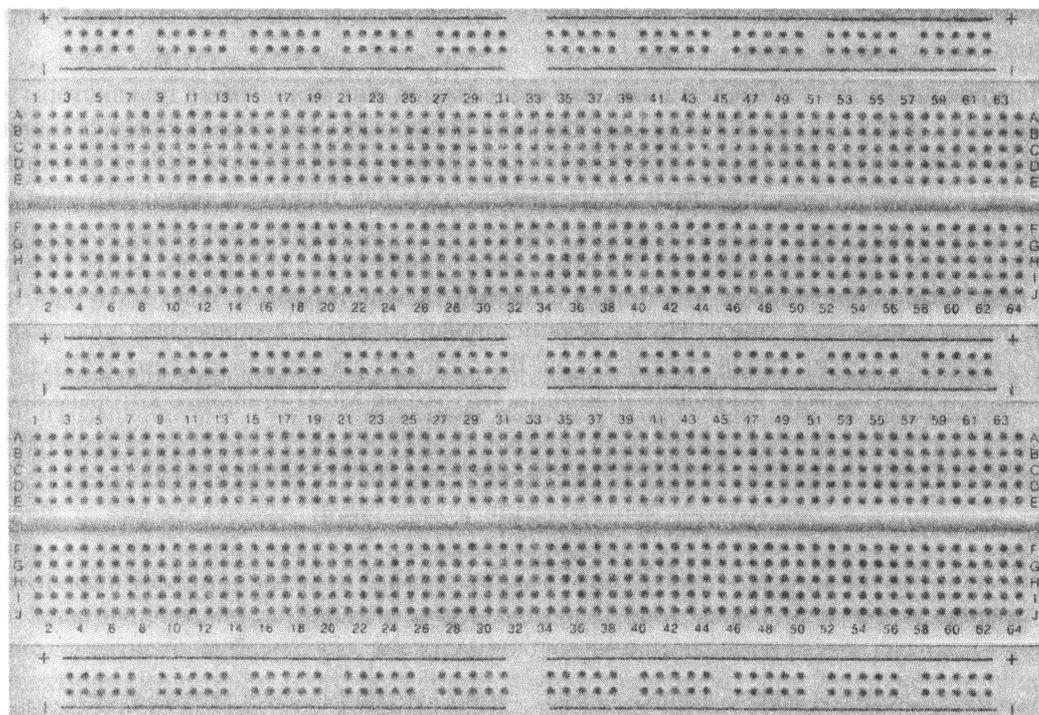


Рис. Ц-2.3. Макетная плата: вид сверху

¹ Префиксом «Ц-» помечены цветные иллюстрации, вынесенные в электронный архив, сопровождающий книгу.

если средства позволяют, желательно приобрести оба типа соединителей. Но для начала будет достаточно набора гибких проводов. Также вы можете, затратив пару часов времени, сами создать жесткие перемычки из отрезков одножильного провода, зачистив по пять миллиметров изоляции с каждого конца. Кроме платы и проводов непременно потребуется батарея питания на 9 В типа «Крона» и контактная колодка для подсоединения ее к плате, показанная на рис. Ц-2.4, в.

Если на макетную плату посмотреть с обратной стороны, сняв ее заднюю панель, то станет видно, что каждый вертикальный столбец из пяти контактных отверстий

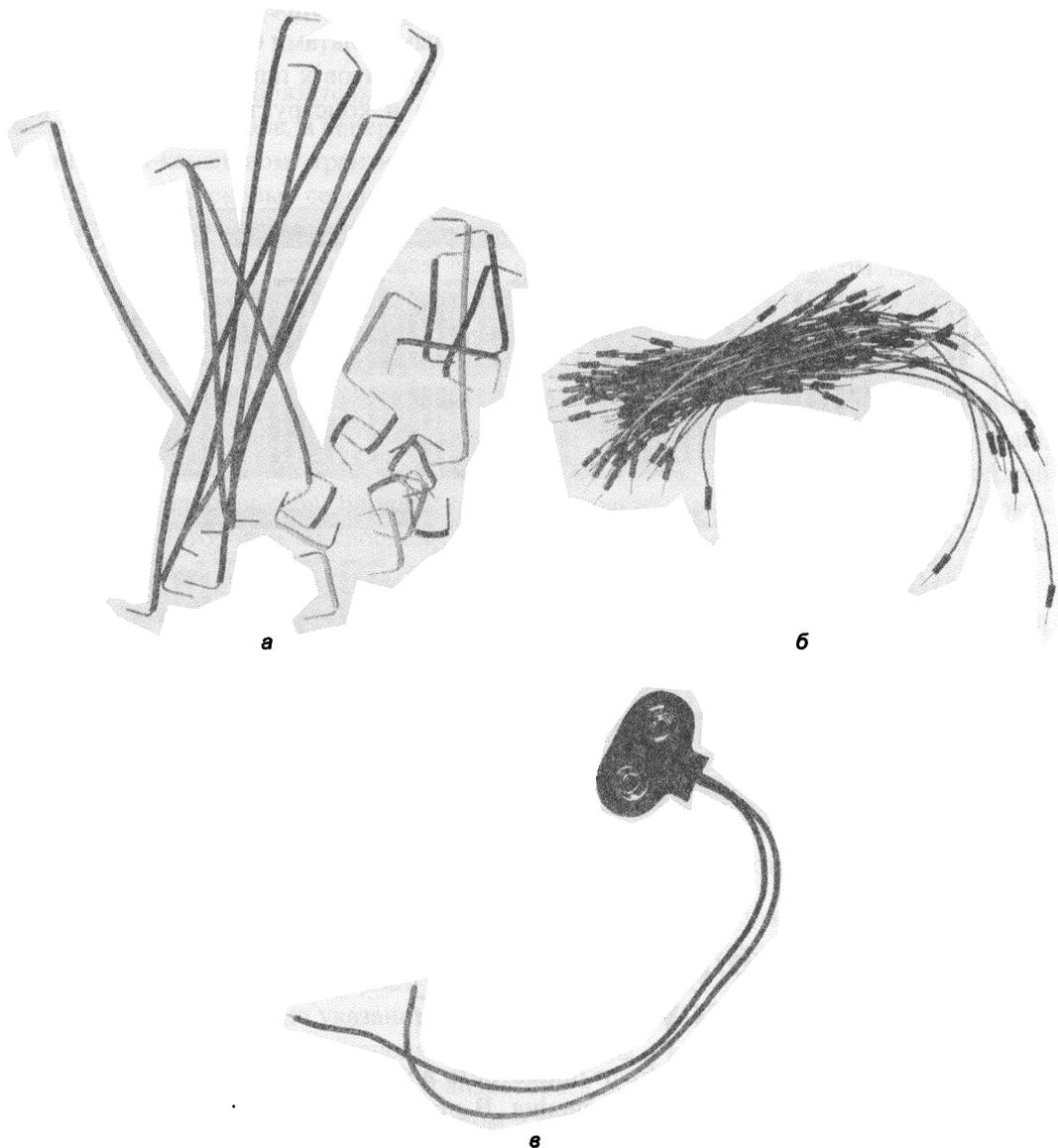


Рис. Ц-2.4. Жесткие перемычки (а); гибкие перемычки (б); колодка для батарейки типа «Крона» (в)

содержит металлическую полосу, соединяющую эти отверстия, как показано на рис. 2.5. То есть выводы деталей, стоящих в одном вертикальном ряду, всегда будут подключены друг к другу. Канавка между каждыми пятью горизонтальными рядами отверстий разрывает вертикальную связь. Так что отверстия сверху от канавки никак не связаны с отверстиями снизу от нее. Это нужно для установки и удобного подключения микросхем и обязательно пригодится далее. Еще на макетной плате могут быть горизонтальные проводники для плюса и минуса питания. Их называют *шинами питания* и отмечают красной и синей линиями и соответствующими символами (см. рис. Ц-2.3). В том месте, где цветная линия прерывается, разрывается и шина питания. Если необходимо сделать ее непрерывной, нужно применять перемычки. Можно пользоваться и макетными платами без шин питания, но в этом случае подключение большого числа компонентов к плюсу и минусу батареи станет неудобным. *Компонентами* инженеры и конструкторы называют все детали, установленные на плате, кроме проводов.

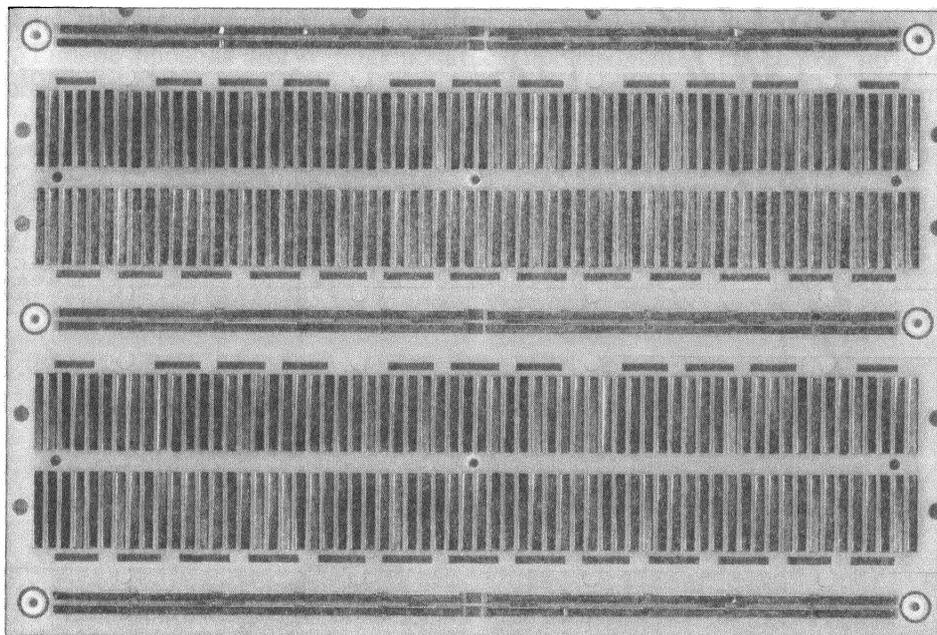


Рис. 2.5. Макетная плата: вид снизу

Как выбрать подходящие для экспериментов транзисторы и другие детали?

Вернемся к МОП-транзистору. Если посмотреть внимательно на его корпус, то на нем можно увидеть условные обозначения. В примерах этой книги используются транзисторы, которые выглядят, как показано на рис. 2.6 (n-канальный — *слева*, p-канальный — *справа*).

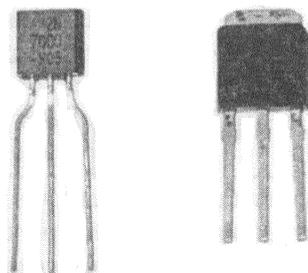


Рис. 2.6. n-канальный (слева) и p-канальный (справа) транзисторы, используемые в схемах этой книги

Раньше инженеру обязательно были нужны справочники с описанием всех возможных транзисторов, а теперь эту информацию можно найти в Интернете. На выбранном n-канальном транзисторе имеется маркировка с его наименованием: 2N7000TA. Поисковые сервисы сразу же выдают множество ссылок на документацию по этому транзистору. Лучше всего, конечно, взять справочные данные от производителя. Не стоит пугаться, если они окажутся на английском языке. Нужно только убедиться, что это, действительно, транзистор типа MOS или MOSFET и найти изображение с порядком его выводов. Иногда такое изображение называют *цоколёвкой*. Слово это родом из эпохи радиоламп, у которых был цоколь с выводами. У транзисторов цоколя нет, а есть корпус и три вывода, которые по-английски обозначаются, как вы уже знаете, так: исток — source, сток — drain, затвор — gate. Или первыми буквами этих названий. Для 2N7000TA, расположенного плоской стороной корпуса к себе, порядок слева направо таков: исток, затвор, сток. Цоколевки транзисторов и микросхем, применяемых в схемах этой книги, приведены в *приложении 2*.

В качестве p-канального прибора был взят транзистор IRFU9024NPBF, который заметно отличается от n-канального типом корпуса, чтобы вы могли легко отличать их друг от друга на иллюстрациях. Порядок его выводов, если расположить транзистор маркировкой к себе: затвор, сток, исток. Металлическая пластина позади корпуса — теплоотвод, подключенный к стоку. Если через транзистор протекает большой ток, то прибор перегревается и может выйти из строя. Чтобы этого не произошло, его теплоотвод размещают на специальном радиаторе. Но в наших экспериментах больших токов не будет, поэтому радиатор не требуется.

Не обязательно для опытов покупать именно эти транзисторы — достаточно знать, на какие параметры нужно смотреть, чтобы выбрать подходящий прибор из доступных в обычном или интернет-магазине. Как уже упоминалось ранее, в документации должно быть явно указано, что это МОП- (MOS) транзистор. Либо приведено одно из тех условных обозначений, которые даны в начале этой главы. Как правило, к обозначению дискретного транзистора в документации добавляют еще символ, изображающий диод между стоком и истоком. Тогда схема транзистора в описании может выглядеть примерно так, как изображено на рис. 2.7.

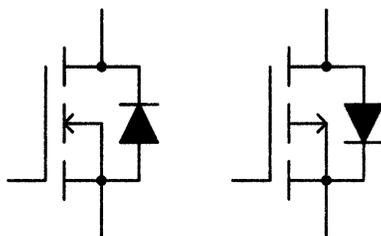


Рис. 2.7. МОП-транзисторы с закороченными истоком и подложкой

ПАРАЗИТНЫЕ ДИОДЫ

Диод — это полупроводниковый прибор, который проводит ток только в одном направлении. Если тот его вывод, который называется *анодом* и обозначается треугольником, будет направлен в сторону плюса питания, а второй, который называется *катодом* и обозначается чертой, — в сторону минуса, то через диод будет течь ток. Такое подключение называется *прямым*. Если же включить диод в обратном направлении, ток через него течь не будет. Если вы вдруг захотите провести эксперимент с прямым включением первого попавшегося диода, не надо присоединять его анод и катод напрямую к плюсу и минусу батарейки без последовательно включенного ограничивающего резистора — через диод может потечь слишком большой ток, на который он не рассчитан. Возникнет короткое замыкание, и батарея разрядится, а диод может и сгореть. Так откуда вдруг взялся диод в МОП-транзисторе? А прямо из структуры полупроводникового прибора — никто его не проектировал, он сам собой получился. Поэтому он и называется *паразитным диодом*. Вообще, их там получается при производстве сразу два: между подложкой и стоком и между подложкой и истоком. Но за счет соединения подложки с истоком анод и катод одного из паразитных диодов оказываются соединенными друг с другом. В результате этот диод, как говорят инженеры, становится *коротко замкнутым*, т. е. нейтрализованным. А второй остается. Поэтому нужно помнить, что у р-канальных транзисторов исток должен всегда «смотреть» в сторону плюса питания, а у п-канальных, наоборот, в сторону минуса. Тогда паразитные диоды в этих транзисторах окажутся обратно включенными и заметного влияния на работу схем оказать не смогут.

Второй важный параметр: *предельно допустимое напряжение между стоком и истоком*, которое в документации обозначается V_{dss} . Оно положительное для п-канальных транзисторов и отрицательное для р-канальных. По модулю величина этого напряжения должна несколько превышать напряжение питания. Так как мы используем для питания схем батарейку «Крона», выдающую 9 В, то значения V_{dss} для выбранных транзисторов, равные 60 и -55 В, более чем подходят. Третий параметр, требующий внимания: *максимальный ток стока* I_d . Схемы из этой книги будут управлять светодиодами, через которые будет протекать ток в несколько десятков миллиампер. Поэтому величина $I_d = 200...300$ мА по модулю более чем достаточна. Если допустимый ток, указанный в документации, сильно больше, то это транзистор, предназначенный для работы на большую нагрузку, и он, конечно, подойдет для схем, приведенных в книге, просто будет стоить дороже. Значения всех рассмотренных параметров не должны подбираться с позиции «как можно больше» — они должны быть подходящими для решаемой задачи с небольшим запасом. Тот р-канальный транзистор, который показан в примерах, обладает заметно большим максимально допустимым током стока, чем п-канальный, но это будет использоваться далее в ряде схем. Ну и корпус выбранных транзисторов должен иметь

выводы в виде длинных тонких штырьков, чтобы можно было их устанавливать в контактные отверстия на макетной плате. Конечно, при необходимости слишком длинные выводы укорачивают, отрезая лишнее кусачками. Если же выводы для установки деталей на плату требуется изгибать, то лучше всего это делать с помощью металлического пинцета.

Сборку схемы лучше всегда начинать с установки самых «многоногих» компонентов. А батарея питания подключается последней к уже собранной схеме. Это делается для того, чтобы избежать риска повреждения полупроводниковых приборов случайно возникшими неправильными потенциалами на их выводах в процессе сборки. Существует еще несколько выработанных инженерами за много лет работы правил, которых желательно придерживаться. Например, при наличии разноцветных проводов плюс батареи питания предпочтительно подключать проводами красного цвета, а минус — черного. Цвет проводов никак не влияет на протекание тока, но делает рассмотрение схемы и поиск в ней неполадок более удобным благодаря определенности. С гибкими соединителями это правило соблюдать легко. А вот жесткие переключки обычно изготавливаются по принципу «одинаковый цвет — одинаковая длина», поэтому компоненты на плате подключаются к шинам питания теми переключками, которые подходят по длине, невзирая на цвет.

Другое важное правило в устранении неисправностей: если что-то не работает, первым делом надо проверить, подключено ли питание! Ну а потом уже, если оно есть, необходимо убедиться, что все компоненты установлены правильно. Просток, затвор и исток у транзисторов уже сказано. Нужно также обращать внимание на полярность подключения светодиода — диода, способного светиться при прямом включении. У него два вывода: они разной длины — анод длинный, а катод — короткий. Анод всегда подключается в сторону положительного уровня напряжения и обозначается на схеме, как вы уже знаете, треугольником, а катод — отрицательного и обозначается чертой, напоминающей минус. Если полярность светодиода не соблюдать, то он не будет загораться. Без резистора, ограничивающего ток до нескольких десятков миллиампер, подключать светодиод не стоит, если вы не хотите, чтобы он перегорел. У резисторов же оба вывода абсолютно равнозначны. Полоски на корпусе резистора кодируют величину его сопротивления. В *приложении 3* приведена таблица, позволяющая определить сопротивление по цветовому коду. Если сопротивление резистора находится в диапазоне до 1 кОм (читается «один килоом» или одна тысяча ом), то номинал на схемах или в документации подписывается цифрами без единиц измерений, от 1 до 999 кОм — рядом с цифрами пишется буква к, а для мегаомных значений (т. е. от одного миллиона ом) используется буква М. Внешний вид и условные графические обозначения резистора и светодиода показаны на рис. 2.8.

В процессе освоения материала книги схемы будут усложняться, и в них появятся другие детали. Там, где они встретятся впервые, о них будет рассказано подробно. А для того чтобы начать изучение цифровой электроники на практике, достаточно уже рассмотренных компонентов. Сводная таблица используемых в книге компонентов приведена в *приложении 4*.

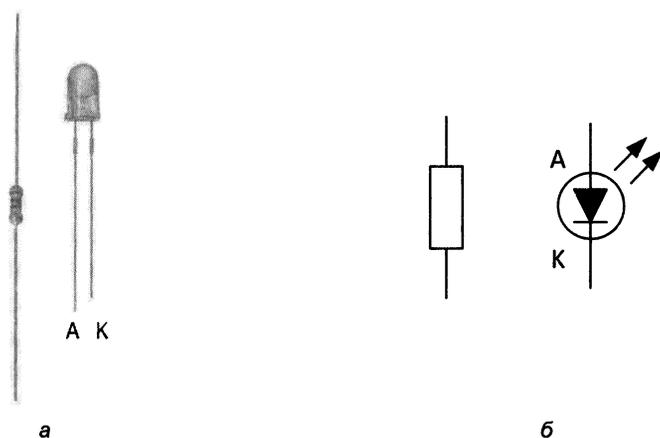


Рис. 2.8. Резистор и светодиод (а) и их обозначения на схемах (б)

Вопросы для самопроверки

1. Какой сигнал: логический ноль или логическую единицу — нужно подать на затвор р-канального транзистора, чтобы между стоком и истоком потек ток?
2. Можно ли менять местами сток и исток у дискретных МОП-транзисторов?

Первая практическая схема, которая помогает увидеть нули и единицы

Настало время перейти к практическим конструкциям. С помощью первой вы самостоятельно убедитесь, что транзисторы работают как электронные ключи, причем р-канальный открывается логическим нулем, а n-канальный — логической единицей. На бумаге условные графические обозначения компонентов размещаются в виде, который называется *принципиальной электрической схемой*. Она показывает, каким образом компоненты должны быть соединены друг с другом. При этом на плате они могут размещаться совершенно в ином порядке — важно только выполнить все подключения правильно.

На первой схеме, изображенной на рис. 2.9, вы можете увидеть обозначения компонентов, которые для определенности подписываются цифробуквенным кодом. Так, номера транзисторов на схеме имеют перед собой букву Т, светодиодов — VD, резисторов — R. А номер позволяет отличить компоненты одного типа друг от друга при описании схемы. Кроме уже известных вам деталей здесь показаны и два переключателя: SA1 и SA2, которые подсоединяют затворы транзисторов к плюсу или минусу питания в зависимости от положения рычажка. Но на рис. Ц-2.10, показывающем, как выглядит собранная на макетной плате схема, вместо них вы обнаружите простые гибкие провода. Так нагляднее, потому что хорошо видно, куда именно они подключены, т. е. какой уровень — высокий или низкий — подается на затвор. Жирная точка на схеме в месте пересечения проводов играет роль вообра-

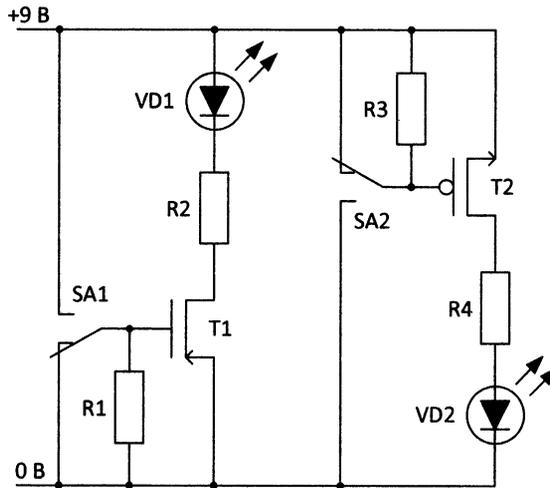


Рис. 2.9. Первая электрическая схема на МОП-транзисторах

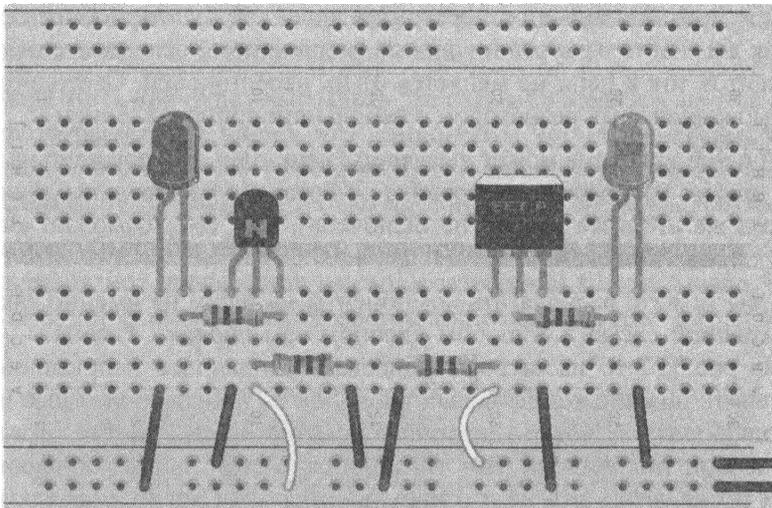


Рис. Ц-2.10. Первая монтажная схема на МОП-транзисторах

жаемой капли припоя, т. е. она показывает, что эти выводы компонентов должны быть обязательно соединены друг с другом. Если же провода на схеме пересекаются без точки, то соединения между ними нет, просто их было удобно начертить таким образом, как будто один проходит над другим. Напряжения при питании от батарейки отсчитываются от ее минуса, поэтому 0 В подключаем именно к этому контакту. А про подключение точки +9 В к плюсу вы и сами уже догадались.

Рассмотрим работу схемы пока на бумаге, чтобы собирать ее на макетной плате с полным пониманием. Как уже упоминалось ранее, затворы транзисторов Т1 и Т2 подключаются либо к плюсу батареи питания (логическая единица), либо к минусу (логический ноль) при помощи гибких проводов, используемых в качестве пере-

ключателей. Резисторы R1 и R3 задают напряжение на затворах в отсутствие входного сигнала. У n-канального T1 затвор через резистор R1 «привязан» к минусу, а у р-канального T2 через R3 — к плюсу батареи. Таким образом, при положении переключателей, показанном на схеме, при включении питания оба транзистора сразу оказываются в выключенном состоянии, потому что затворы транзисторов соединяются через резисторы с теми контактами питания, которые держат их закрытыми. Если гибкий провод, используемый вместо переключателя, вообще вынуть из платы, то потенциал обоих выводов каждого из резисторов R1 и R3 не изменится. Значит, не изменятся напряжения на затворах обоих транзисторов, и они не включатся. А вот если поставить SA1 и SA2 в другое положение или, в нашем случае, подключить гибкий провод к противоположной шине питания, то оба транзистора откроются, через них, светодиоды и резисторы R2 и R4 потечет ток, и светодиоды загорятся. При этом резисторы R1 и R3 оказываются подключены между плюсом и минусом батареи, т. е. и через них тоже есть электрический ток. Для того чтобы он не был слишком велик, и не разрядил быстро батарею, сопротивление обоих резисторов выбрано 1 кОм. Резисторы R2 и R4 нужны для ограничения тока через светодиоды, которые иначе могут сгореть. Их сопротивление равно 100 Ом. Причем эти токоограничивающие резисторы можно менять местами со светодиодами, которые они защищают от перегрузки, если так будет удобнее при монтаже на плате. Как вы знаете из курса школьной физики, при последовательном включении компонентов ток в цепи не меняется от их перестановки. Светодиод в прямом включении можно считать резистором с небольшим сопротивлением.

На макетной плате эта схема может выглядеть, например, как показано на рис. Ц-2.10 (здесь и на других изображениях провода от батарейки подведены сбоку к шинам питания, сама же «Крона» в кадр не включена). Рис. Ц-2.10 можно считать *монтажной электрической схемой*.

Разумеется, вы можете располагать компоненты на плате не именно так, как показано на монтажной схеме, а в любом удобном вам порядке, главное — чтобы все работало. На рис. Ц-2.10 на затворы подается отпирающее напряжение. Если теперь переставить нижние концы белых проводов на противоположные шины питания, т. е. подключить затвор р-канального транзистора к единице, а n-канального к нулю, — то они выключатся. Вот вы и собрали свою первую цифровую схему, можете себя поздравить! Это не что иное, как пробник логических уровней, очень полезная вещь. Даже сразу два пробника: один загорается, когда на его входе логический ноль, и гаснет, когда единица. А второй ведет себя противоположным образом. В дальнейшем вы можете использовать любой из них для того, чтобы визуально определять состояние какого-нибудь узла цифрового устройства. Достаточно подключиться гибким проводом с затвора как щупом к интересующей вас точке, и станет ясно, единица там или ноль в данный момент времени. Конечно, при этом плюс и минус питания и тестируемой вами схемы, и пробника должны быть подключены к одной и той же батарейке.

На этой монтажной схеме и на всех подобных далее будут применяться как уже упомянутые ранее правила цветовой кодировки проводов, так и новые:

- красный провод подает плюс питания;
- черный провод — минус питания;

- белый провод — входной сигнал, он может быть подключен как к плюсу, так и к минусу питания попеременно для изменения состояния устройства;
- остальные цвета проводов равнозначны и применяются для удобства рассмотрения схем;
- провод электрически соединяет между собой только точки на своих концах, а контакты, провода и компоненты, оказавшиеся под ним на схеме, никак с ним не связаны;
- р-МОП-транзистор вы всегда устанавливаете лицевой стороной к себе;
- n-МОП-транзистор, установленный лицевой (плоской) стороной к себе, имеет на монтажной схеме видимую букву N на корпусе, установленный лицевой стороной от себя — буквы не имеет;
- анод светодиода (более длинный вывод) имеет на схеме изогнутую форму;
- порядок шин плюса и минуса питания на монтажной схеме может отличаться от их расположения на вашей макетной плате, будьте внимательны;
- цвет полосок на корпусе резистора имеет значение, поэтому сверяйтесь с цветным вариантом монтажной схемы, если на принципиальной ее версии используются несколько разных номиналов резисторов.

А для примера того, как будет выглядеть эта схема «в железе», на рис. Ц-2.11 приведена ее фотография, где видно, что компоненты установлены несколько иначе, но, тем не менее, это все те же пробники нуля и единицы (в дальнейшем к принципиальным схемам будут даваться только их монтажные варианты, без фотографий).

Из схемы можно было бы исключить резисторы R1 и R3, т. к. транзисторы станут реагировать на разное напряжение на своих затворах и в этом случае. Но тогда заранее нельзя будет сказать, в каком состоянии, открытом или закрытом, окажется

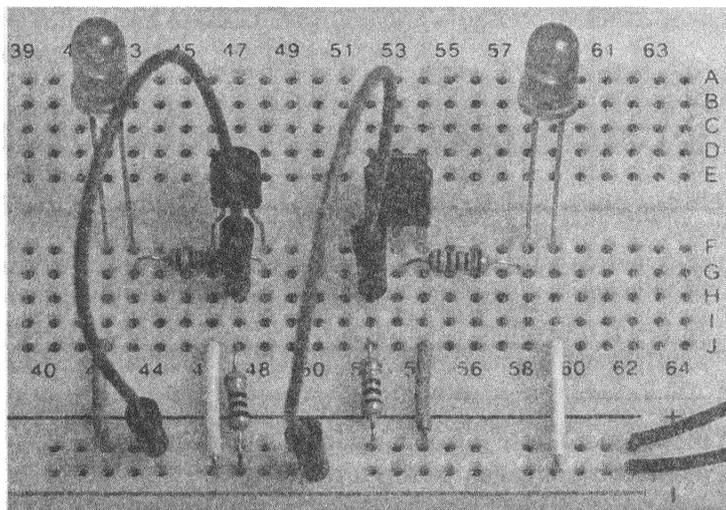


Рис. Ц-2.11. Фотография первой монтажной схемы на МОП-транзисторах

каждый транзистор при включении питания и при еще не подключенном куда-либо гибком проводе, — ведь потенциал затвора в таком случае ничем не задается. Вы можете убедиться в этом, убрав R1 и R3 с платы и подключая затворы к плюсу или минусу питания, а потом вынимая гибкий провод из контактного отверстия. Загоревшийся светодиод будет продолжать гореть даже при никуда не подключенном затворе за счет оставшегося на затворе электрического заряда. И если питание выключить, а потом снова включить, то светодиод может опять загореться, хотя затвор по-прежнему никуда не подключен, а, как говорят, «висит в воздухе». Этот эксперимент показывает то, о чем говорилось в начале главы: ток через затвор МОП-транзистора не течет. При подключении затвора к какому-либо потенциалу на нем накапливается соответствующий электрический заряд. И если нет привязки затвора к шине питания через резистор, заряду некуда стекать. Поэтому при вынимании белого провода из гнезда состояние пробника не изменяется. Этот опыт показывает еще и справедливость утверждения, сделанного в конце первой главы: если вам важно иметь строго заданное исходное состояние, то вы должны позаботиться об этом сами. Что и было сделано благодаря включению в схему резисторов R1 и R3.

У вас может возникнуть вопрос, а почему транзисторы в схемах расположены по-разному: исток р-канального не просто направлен в сторону плюса питания, а подключен прямо к нему, а исток n-канального подключен к минусу? Нельзя ли, например, поменять их местами, чтобы n-канальный передавал логическую единицу, а р-канальный — логический ноль, т. е. подключить, как показано на рис. 2.12.

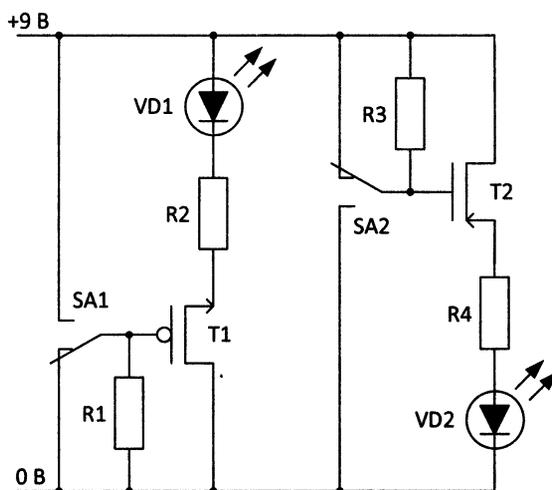


Рис. 2.12. Схема, в которой р-канальный транзистор передает ноль, а n-канальный — единицу

В данном случае можно, и на первый взгляд все будет работать точно так же, сами проверьте. Но если вы возьмете прибор для измерения напряжений — вольтметр — и замерите разность потенциалов между стоком и истоком каждого транзистора, когда светодиоды горят, то увидите, что в исходной схеме она была около нуля вольт, а в новой стала равной приблизительно 0,7 В. Если у вас другие транзисторы

или батарейка уже не совсем свежая, то эта разница будет иной, но все равно будет. Почему же так происходит?

Дело в том, что физика работы *n*-канального транзистора такова, что он без потерь передает от истока на сток напряжение логического нуля, а напряжение логической единицы — с потерями, зависящими от величины порогового напряжения. А *p*-канальный, наоборот, бережно обращается с логической единицей, но портит логический ноль. То есть при неправильном включении транзисторов не все напряжение питания падает на нагрузку, а теряется частично на самом полупроводниковом ключе. И получается, что на выходе такой схемы логический ноль уже и не равен нулю вольт, а единица не равна напряжению питания. В результате помехоустойчивость цифровой схемы ухудшается. Так как надежность устройства — самое главное, то нужно помнить, что для создания логического нуля следует применять только *n*-канальные транзисторы, а для логической единицы — только *p*-канальные. Этот принцип и ляжет в основу всех схем, которые изучаются далее.

Конструкции для самостоятельной разработки

1. Внесите изменения в схему пробника на *n*-канальном транзисторе так, чтобы в исходном состоянии (при неподключенном ни к чему щупе) светодиод VD1 горел, а гас при нуле на затворе T1.
2. Внесите изменения в схему пробника на *p*-канальном транзисторе так, чтобы в исходном состоянии (при неподключенном ни к чему щупе) светодиод VD2 горел, а гас при единице на затворе T2.

Вопросы для самопроверки

3. Для чего вместе с переключателями или кнопками, задающими сигналы на затворах транзисторов, используются резисторы?
4. Для чего нужны резисторы, включенные последовательно со светодиодами?
5. Что будет, если анод и катод светодиодов в рассмотренных выше схемах поменять местами?

Игра «Кто быстрее?» на двух транзисторах

На основе пробника логических сигналов можно за пару минут создать интересную и динамичную игру «Кто быстрее?» Для этого потребуются две кнопки, четыре резистора, два светодиода и два *n*-канальных транзистора, стоки и затворы которых соединены перекрестно через кнопки, как показано на рис. 2.13. По команде два игрока нажимают каждый на свою кнопку. Кто сделает это быстрее — покажет загоревшийся светодиод. Номиналы токоограничивающих резисторов R1 и R4 точно такие же, как и в схемах пробников, — 100 Ом. А резисторы R2, R3, устанавливающие транзисторы в исходно закрытое состояние, имеют сопротивление 10 кОм, в отличие от резисторов в пробниках.

Эта схема работает очень просто. Изначально оба транзистора закрыты, и ток через них не течет. Поэтому на их стоках устанавливается напряжение 9 В. Допустим,

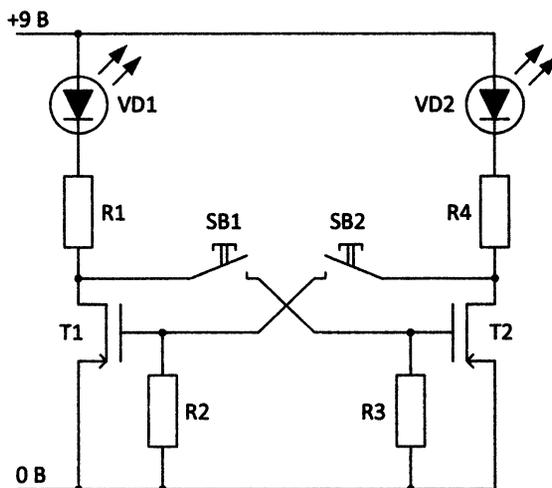


Рис. 2.13. Принципиальная схема игры «Кто быстрее»

первой была нажата кнопка SB1. В этот момент ток начинает течь от плюса батареи к минусу через VD1, R1, SB1, R3. Величина его слишком мала, чтобы зажечь светодиод VD1, потому что сопротивление R3 велико. При этом на затворе T2 устанавливается напряжение чуть менее 9 В, но, тем не менее, соответствующее логической единице. Транзистор T2 открывается, и на его стоке оказывается то же напряжение, что и на истоке — 0 В. Через VD2 и R4 начинает течь ток, заставляющий светодиод VD2 ярко гореть, поскольку сопротивление R4 всего 100 Ом, а сопротивление открытого транзистора пренебрежимо мало. Если теперь при нажатой кнопке SB1 нажать на SB2, то состояние схемы никак не изменится, поскольку на затвор T1 будет подаваться логический ноль со стока уже открытого T2. Так как транзистор — очень быстрый переключатель тока, то даже разницы в несколько сотых долей секунды между нажатиями двух кнопок достаточно, чтобы произошло все, что написано про работу этой схемы. В случае, если более быстрым окажется игрок, нажимающий кнопку SB2, произойдет то же самое, только в зеркальном отображении. На монтажной схеме (рис. Ц-2.14) кнопка SB2 расположена рядом с зажигаемым ею светодиодом VD1, а кнопка SB1 — рядом с VD2.

Кнопки: два вывода или четыре?

Здесь используются кнопки с двумя выводами, расположенными по середине противоположных сторон корпуса. Они занимают мало места на плате и удобно вставляются в контактные отверстия. При этом их выводы, конечно же, располагаются на всех схемах вдоль горизонтальной стороны платы — в ином случае они окажутся коротко замкнутыми. Существуют похожие внешне сдвоенные кнопки с четырьмя выводами, расположенными по углам корпуса. Они чуть больше, и при их использовании монтажные схемы будут выглядеть иначе. Поэтому, если вы хотите в точности повторить именно те схемы, которые приведены в этой книге, приобретайте двухвыводные кнопки.

На этом от рассмотрения компонентов и простейших схем можно переходить к более сложным базовым элементам цифровой электроники.

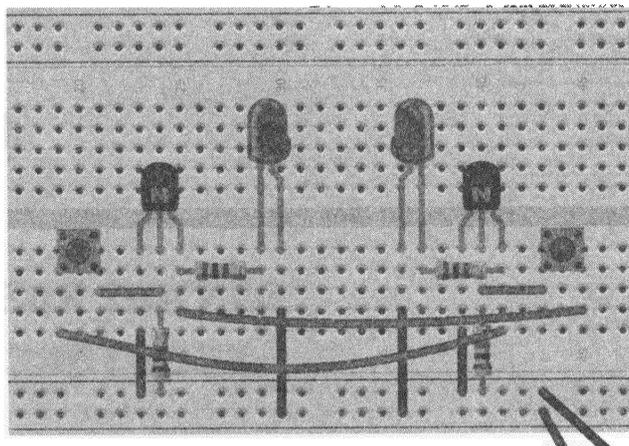
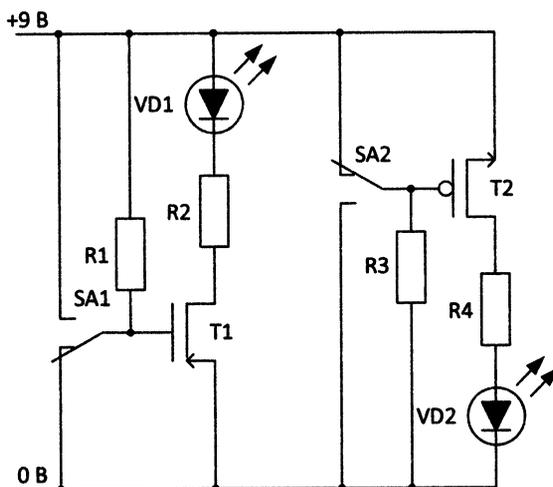


Рис. Ц-2.14. Игра «Кто быстрее» на макетной плате

Ответы на вопросы для самопроверки

1. Логический ноль.
2. Нельзя.
3. Чтобы на затворах транзисторов всегда было определенное напряжение.
4. Они ограничивают ток, защищая светодиоды от перегорания.
5. Светодиод не будет гореть ни при каких логических сигналах на затворах транзисторов в этих схемах.

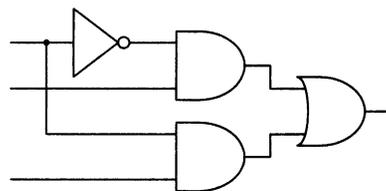
Ответы на задания по конструкциям для самостоятельной разработки



Слева — ответ на задание 1; справа — ответ на задание 2

ГЛАВА 3

Цифровая логика на транзисторах



Вам потребуются:

- 6 n-канальных и 6 p-канальных транзисторов;
- 2 светодиода;
- 2 резистора сопротивлением 100 Ом.

Собираем элемент, который переворачивает сигналы наоборот

В предыдущей главе вы создали на транзисторных ключах два пробника, реагирующих зажиганием светодиода на разные логические уровни входного сигнала. А нельзя ли объединить их в одну схему? Можно попробовать. Вы уже знаете, что открываемый низким логическим уровнем p-канальный транзистор хорошо передает на выход единицу от источника питания. А n-канальный прибор ведет себя противоположным образом. Значит, если объединить их затворы и стоки, а истоки подключить к плюсу и минусу питания, а потом присоединить к выходу такой конструкции два светодиода с ограничивающими ток резисторами, то получится схема универсального пробника, показанная на рис. 3.1 и рис. Ц-3.2. Светодиод VD1 будет загораться в том случае, если на стоках транзисторов логический ноль, а VD2 — при логической единице в этом же узле. Кстати, от установки начального состояния здесь можно отказаться — все равно один из светодиодов в любом случае будет гореть, поэтому не важно, в каком состоянии окажется схема при включении питания.

Давайте проанализируем работу транзисторов в этой схеме. Предположим, что на ее вход, т. е. на оба затвора, подан ноль. Тогда n-канальный ключ T2 будет закрыт, а p-канальный T1 — открыт. И на выход, т. е. на оба стока, с плюса питания будет выдаваться единица без потерь. Если же ситуация изменится и на затворы придет единица, то p-канальный транзистор T1 закроется, а n-канальный T2 откроется, и на выходе окажется ноль с минуса питания. То есть из двух транзисторов удалось создать устройство, у которого на выходе сигнал всегда обратный входному. Это самый простой логический элемент — *инвертор* (от лат. *inversio*, переворачивание).

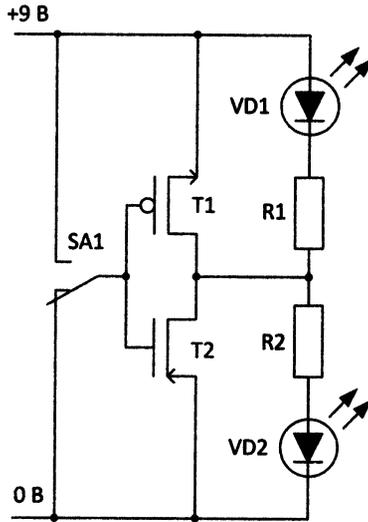


Рис. 3.1. Схема универсального пробника-индикатора

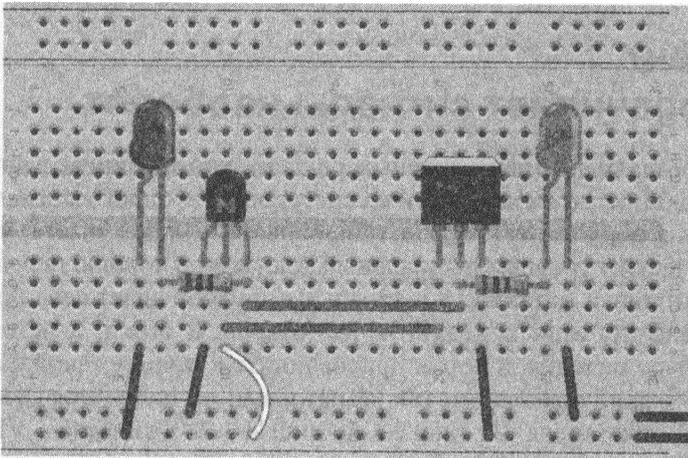


Рис. Ц-3.2. Универсальный пробник-индикатор на макетной плате

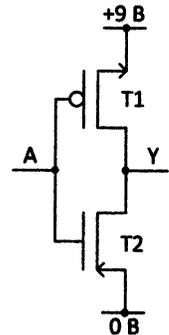


Рис. 3.3. Схема инвертора

На рис. 3.3 его схема изображена отдельно от остальных, вспомогательных, компонентов: вход инвертора обозначен буквой А, а выход — буквой Y.

Этот и другие цифровые элементы, которые будут рассмотрены далее, называют *вентильми* — по аналогии с трубопроводными устройствами, которые пропускают воду или перекрывают ее в зависимости от положения рукоятки. Только здесь не сантехнические вентили, а логические. И управляются они не вручную, а электрическими цифровыми сигналами. И пропускают или блокируют они не поток жидкости, а поток двоичных данных. Условные обозначения инвертора в отечественной и зарубежной литературе, как и изображения всех прочих вентилей, различаются. Поэтому на рис. 3.4 приведены: *слева* — обозначение, принятое в российских

источниках, *справа* — в зарубежных), чтобы вы могли легко ориентироваться в схемах, независимо от их происхождения. В этой же книге будут использоваться зарубежные, которые своей формой сразу говорят о типе вентиля, без необходимости рассматривать символ внутри прямоугольника. Тем более, что такие обозначения уже часто применяют не только в переводных, но и в российских научно-технических изданиях.



Рис. 3.4. Условное графическое обозначение инвертора: слева — принятое в российской литературе, справа — в зарубежной

В схеме инвертора используются оба типа транзисторов. Причем, как вы знаете, р-канальный обеспечивает на выходе единицу без потерь по уровню напряжения, с чем не может справиться его n-канальный собрат. Зато второй прекрасно формирует ноль на выходе. Они дополняют друг друга, выполняя ту часть работы, для которой каждый из них подходит наилучшим образом. Поэтому логические элементы, построенные по такому принципу — с использованием обоих типов транзисторов вместе — называются комплементарными МОП-вентильми. Или сокращенно — КМОП. Обратите внимание, что «комплементарный» пишется через букву Е, а не через И, — в отличие от приятного слова «комплимент». И означает этот термин «взаимодополняющий».

Если записать в одну колонку все возможные состояния входа инвертора А, а в другую — соответствующие им состояния выхода Y, то получится табл. 3.1, состоящая из двух колонок:

Таблица 3.1. Сигналы инвертора

A	Y
0	1
1	0

Она называется *таблицей истинности* и в краткой и емкой форме рассказывает о принципе работы вентиля. В данном случае из таблицы ясно, что этот элемент имеет один вход, один выход и выполняет инверсию входного сигнала, т. е. его превращение в свою противоположность. Далее вы познакомитесь с более сложными вентилями и научитесь как составлять для них таблицы истинности, так и решать обратную задачу: по заданной таблице конструировать схему на транзисторах. Кроме таблиц, для описания цифровых схем используются формулы и теоремы из особого раздела математики, называющегося *булевой алгеброй* или *двоичной логикой*. Это довольно обширная область знаний, которую изучают в высших учебных заведениях. В этой книге приводится только небольшая ее часть, необходимая

для понимания материала. А в список литературы, приведенный в *приложении 1*, включена книга «Дискретная математика и комбинаторика», которая будет полезна тем, кто хочет глубже освоить эту науку. Функцию, выполняющую инверсию, в формулах булевой алгебры обозначают чертой над обозначением сигнала. То есть $Y = \bar{A}$. Далее в книге будет часто встречаться подобное обозначение инвертированных сигналов.

Еще раз посмотрим на схему вентиля, приведенную на рис. 3.3. Можно обратить внимание на то, что при его работе в любом устойчивом состоянии, будь то ноль на входе или единица, всегда открыт только один транзистор, а второй закрыт. То есть ток в такой схеме между плюсом и минусом питания в стабильном режиме не течет, энергия не расходуется. Только в момент переключения оба транзистора, проходя через область неопределенности в противоположных направлениях, оказываются ненадолго открытыми, и через них в это время течет сквозной ток. Эта особенность — отсутствие энергозатрат при отсутствии переключений — наряду с технологическим удобством сделала КМОП-схемы чрезвычайно привлекательными для разработчиков. Все другие типы цифровой логики гораздо менее бережно относятся к энергоресурсам.

Вопросы для самопроверки

1. Рассмотрите схему из двух инверторов, показанную на рис. 3.5, и составьте таблицу истинности для входа А и выхода Y всей схемы. Промежуточный сигнал, обозначенный буквой А с чертой наверху и соответствующий инверсии входного сигнала А, в таблицу включать не обязательно. Такая схема применяется, например, для очистки сигнала, приходящего на вход А, от искажений и помех и передачи его в другие части цифровой системы. А также для усиления слабого сигнала, если транзисторы второго инвертора будут мощнее, чем первого.

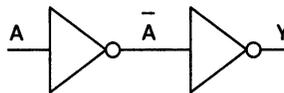


Рис. 3.5. Два последовательно включенных инвертора

Основные логические схемы: И, ИЛИ и все-все-все

Теперь настало время познакомиться с устройством других вентилях. Возможно, вы уже знаете про логические функции И, ИЛИ, И-НЕ, ИЛИ-НЕ и Исключающее ИЛИ, особенно если занимались программированием. Если же нет, то ничего страшного. Далее будет рассказано не только об этих функциях, но и о том, как они реализованы на транзисторах.

Логическое И

Представьте, что вы конструируете стиральную машинку, которая среди прочей электроники содержит в себе два датчика. Один определяет, открыт ли кран на трубе, подающей воду в машинку. А другой — закрыта ли дверца, через которую в барабан загружается белье. И процесс стирки будет разрешено запустить, только если оба датчика ответят «ДА». Если же какой-то из них ответит «НЕТ», то нажимай на кнопку «Старт» или не нажимай, стирка не начнется. Для реализации такой схемы работы прекрасно подходит цифровая электроника. В булевой алгебре принято считать, что ответу «ДА» соответствует логическая единица, а ответу «НЕТ» — логический ноль. Значит, для решения задачи требуется применить такой цифровой вентиль, у которого два входа и один выход. И он должен выдавать единицу на выходе только в единственном случае: если на его первый вход И на его второй вход пришли единицы. Поэтому он называется «логическое И» или AND. Обозначается такой вентиль на схемах, как показано на рис. 3.6 (снова два варианта: *слева* — российский, *справа* — зарубежный).

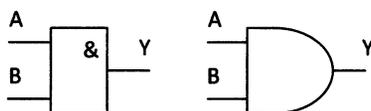


Рис. 3.6. Условное графическое обозначение логического И

А его таблица истинности (табл. 3.2) содержит уже четыре строки, а не две.

Таблица 3.2. Таблица истинности логического И

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Так как в схеме стало больше входов, чем было у инвертора, то количество возможных комбинаций их состояний увеличилось, поэтому и таблица стала больше. Видно, что логической единицы только лишь на одном входе недостаточно для появления высокого уровня на выходе. Как изготовить этот вентиль из транзисторов, вы узнаете чуть позже.

Логическое И сигналов A и B обозначается знаком умножения: $A \cdot B = Y$. Часто еще применяется позаимствованный у программистов символ, называемый *амперсандом*: $A \& B = Y$. Его можно увидеть на отечественном условном графическом обозначении вентиля И (см. рис. 3.6). В зависимости от предпочтений разработчиков на электронных схемах и в формулах двоичных сигналов можно встретить оба варианта.

Логическое ИЛИ

Теперь представим, что вы разрабатываете систему пожарной сигнализации на два помещения, в каждом из которых стоят датчики дыма. Неважно, какой из них срабатывает — сигнал о пожаре должен появиться в любом случае, когда дым обнаружен в одной комнате **ИЛИ** во второй. Вентиль, который обеспечивает работу по схеме «логического ИЛИ» (он же OR), обозначается, как показано на рис. 3.7 (слева — российский вариант, справа — зарубежный).

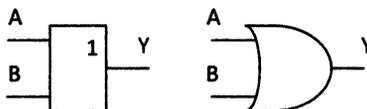


Рис. 3.7. Условное графическое обозначение логического ИЛИ

А выходной сигнал в его таблице истинности (табл. 3.3) понятным образом отличается от сигнала Y логического И.

Таблица 3.3. Таблица истинности логического ИЛИ

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Как только на любом входе такого вентиля появится единица, она тут же будет передана на выход. Логическое ИЛИ сигналов A и B записывается в виде операции сложения: $A + B = Y$.

Вентиль ИЛИ-НЕ

Теперь снова возвращаемся к стиральной машинке. С точки зрения безопасности дверца для загрузки белья должна быть заблокирована для открывания, если внутри налита вода, **ИЛИ** барабан крутится. То есть защелка дверцы **НЕ** открывается в любом из двух случаев. Это означает, что нужен логический вентиль, который выдает единицу только тогда, когда на оба входа подаются нули. Он называется, как уже, наверно, стало понятно, **ИЛИ-НЕ** (NOR в английской версии) и изображается подобно вентилю ИЛИ, только с добавлением кружка, обозначающего инверсию выхода (рис. 3.8).

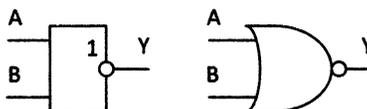


Рис. 3.8. Условное графическое обозначение логического ИЛИ-НЕ:
слева — российское, справа — зарубежное

Его поведение описывается таблицей истинности (табл. 3.4), в которой сигнал на выходе прямо противоположен выходному сигналу для логического ИЛИ.

Таблица 3.4. Таблица истинности логического ИЛИ-НЕ

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Вы уже знаете элемент, который превращает сигналы в противоположные, и даже успели изучить его схему на транзисторах. Речь про инвертор, он же логическое НЕ (NOT). Действительно, ИЛИ-НЕ может быть получен из ИЛИ путем добавления инвертора. И наоборот, ИЛИ может быть получено из ИЛИ-НЕ также добавлением инвертора (рис. 3.9).

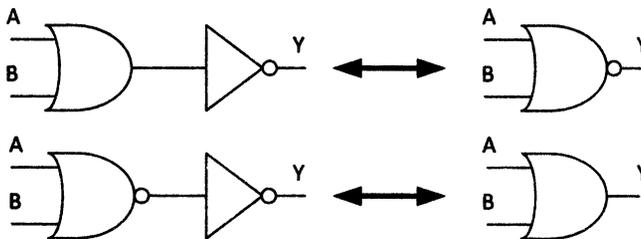


Рис. 3.9. Преобразование логических элементов: *вверху* — ИЛИ-НЕ из ИЛИ путем добавления инвертора; *внизу* — ИЛИ из ИЛИ-НЕ также путем добавления инвертора

Вентиль И-НЕ

Представьте теперь ситуацию, как в шпионском фильме, — секретный завод, на котором есть система непрерывного видеонаблюдения. Отключить ее можно только одновременно нажав две кнопки, одна из которых находится в кабинете директора производства, а вторая — у начальника охраны. Значит, сигнализация НЕ работает, только если директор И начальник охраны нажали кнопки. Вентиль, управляющий такой системой, будет реализовывать функцию логического И-НЕ (он же NAND) и выглядеть, как показано на рис. 3.10.

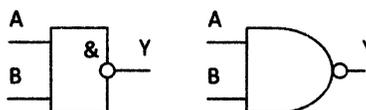


Рис. 3.10. Условное графическое обозначение логического И-НЕ: *слева* — российское, *справа* — зарубежное

В формулах ИЛИ-НЕ и И-НЕ обозначаются добавлением черты инверсии над всей формулой сложения или умножения входных сигналов:

$$Y = \overline{A + B}$$

$$Y = \overline{A \cdot B}.$$

Вопросы для самопроверки

2. Составьте сами таблицу истинности логического И-НЕ.
3. Нарисуйте схемы преобразования вентиля И в И-НЕ и наоборот.
4. Какой вентиль выдает на выходе единицу, когда на оба входа подаются логические нули?
5. На один вход И-НЕ подается логический ноль, на другой — логическая единица. Каким будет состояние выхода?

Исключающее ИЛИ и Исключающее ИЛИ-НЕ

Есть еще один важный логический вентиль. Это Исключающее ИЛИ (XOR). Он используется для сравнения данных. Задача его состоит в том, чтобы выдавать ноль, если сигналы на входах одинаковые, и единицу, если они разные. Его таблица истинности приведена в табл. 3.5, а обозначение — на рис. 3.11.

Таблица 3.5. Таблица истинности Исключающего ИЛИ

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

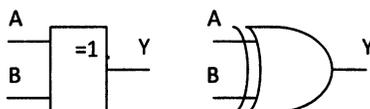


Рис. 3.11. Условное графическое обозначение Исключающего ИЛИ: слева — российское, справа — зарубежное

В булевой алгебре Исключающее ИЛИ обозначается знаком плюса в круге:

$$Y = A \oplus B.$$

Конечно, существует и Исключающее ИЛИ-НЕ (он же XNOR), но с его таблицей истинности (табл. 3.6) и обозначениями (рис. 3.12), наверняка, вам уже и так все должно быть понятно.

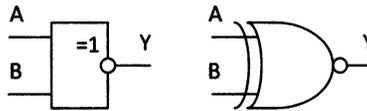


Рис. 3.12. Условное графическое обозначение Исключающего ИЛИ-НЕ: слева — российское, справа — зарубежное

Таблица 3.6. Таблица истинности Исключающего ИЛИ-НЕ

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Вы уже узнали, решая задания для самопроверки, что двойная инверсия сигнала равна самому сигналу, и что с помощью инвертора можно выполнять взаимное преобразование функции И в И-НЕ. Но булева алгебра позволяет выполнять и более сложные преобразования. Например, в ней строго доказывается, что из функции И-НЕ можно получить вообще любую другую логическую функцию. То же самое справедливо и для функции ИЛИ-НЕ. Такие и множество других интересных преобразований можно узнать, изучая этот раздел математики.

Конструирование всех базовых логических элементов: от таблиц истинности к схемам на транзисторах

До сих пор вы рассматривали логические вентили только с точки зрения выполняемых ими функций. И только инвертор был сконструирован из двух транзисторов и проверен на практике. Надо разобраться, как устроены все остальные вентили, и собрать их на макетной плате. Кстати, обратили ли вы внимание, что условные обозначения логических элементов содержат только контакты для входных и выходных сигналов? Для простоты на них не показаны контакты к шинам питания — считается, что инженер-схемотехник и так об этом знает и не забудет подключить вентили к питанию.

Практическая схема ИЛИ-НЕ на транзисторах

Учиться создавать транзисторные схемы проще всего, начиная с ИЛИ-НЕ. Обратимся еще раз к его таблице истинности (см. табл. 3.4).

Как вы уже знаете, логическая единица на выходе схем КМОП без потерь передается р-канальными транзисторами. Как видно из таблицы, она может появиться там

только в одном случае: когда оба входных сигнала А и В одновременно равны нулю. Такое поведение можно обеспечить, если установить два р-канальных транзистора друг за другом, т. е. последовательно. Тогда ноль на любом входе А или В будет открывать только свой транзистор, и ток от плюса питания к выходу Y потечет, лишь когда сразу два транзистора будут открыты. Эта цепочка изображена на рис. 3.13.

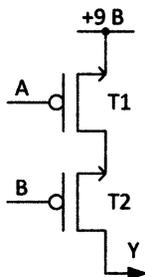


Рис. 3.13. Цепочка р-канальных транзисторов в логическом ИЛИ-НЕ

Теперь нужно в таблице истинности найти, при каких условиях на выходе Y появится логический ноль. Для этого любой из входных сигналов А или В должен быть равным единице. Транзистор, открывающийся единицей и передающий ноль на выход, — это п-канальный прибор. Чтобы реализовать нужное поведение вентиля, два транзистора должны быть подключены параллельно, т. е. их стоки объединяются на выходе Y, а истоки — на минусе питания (рис. 3.14). Тогда единица на любом из входов откроет свой п-канальный транзистор, и ток потечет между выходом и минусом батареи питания.

Соединив полученные цепочки р- и п-канальных транзисторов в одну, получают схему вентиля ИЛИ-НЕ, изображенную на рис. 3.15.

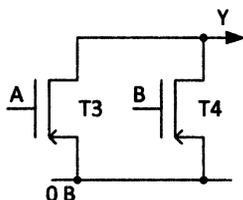


Рис. 3.14. Цепочка п-канальных транзисторов в логическом ИЛИ-НЕ

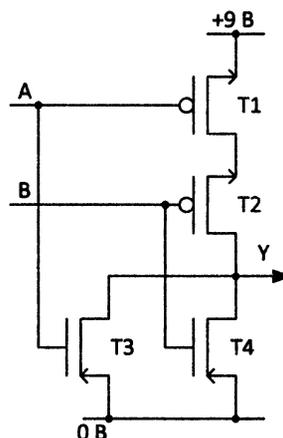


Рис. 3.15. Схема логического ИЛИ-НЕ на транзисторах

А если к выходу вентиля добавить светодиод с резистором, как вы уже умеете, то можно наблюдать состояние выхода Y без вольтметра. Это реализовано на монтажной схеме, приведенной на рис. Ц-3.16. Здесь светодиод подключен к минусу питания, т. е. загорается при единице на выходе Y .

Обращайте на монтажных схемах внимание на положение корпусов транзисторов, если вы повторяете схему вентиля на тех же самых компонентах, что и в книге! На приведенной на рис. Ц-3.16 схеме n -канальные приборы развернуты друг к другу плоской стороной, чтобы их стоки можно было подключить к минусу одной перемычкой, а не двумя. Белые провода установлены в такое положение, чтобы на входы вентиля подавались два нуля — единственное состояние, при котором будет единица на выходе.

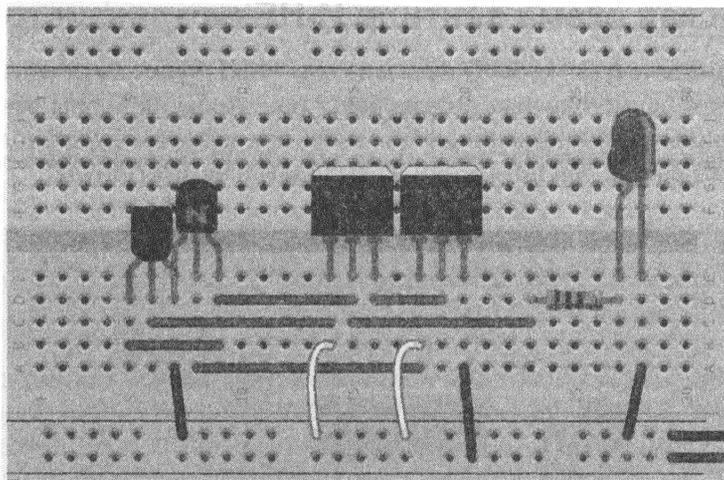


Рис. Ц-3.16. Монтажная схема вентиля ИЛИ-НЕ

Если вы внимательно посмотрите на таблицу истинности (см. табл. 3.4) и на полученную схему ИЛИ-НЕ (см. рис. 3.15), то увидите, что входы A и B равнозначны. Можно поменять их местами — работа вентиля с точки зрения выполняемой функции от этого никак не изменится. Дальше вы увидите, что это же замечание справедливо и для всех остальных базовых вентилях.

Алгоритм создания цифровых логических схем на транзисторах

Итак, еще раз по шагам рассмотрим процесс создания схемы на транзисторном уровне по таблице истинности.

1. Определяем, какая комбинация входных сигналов должна приводить к появлению единицы на выходе. Эту электрическую цепь схемы надо реализовывать на p -канальных транзисторах.
2. Определяем, какая комбинация входных сигналов должна приводить к появлению нуля на выходе. Эту цепь надо реализовывать на n -канальных транзисторах.

3. Если нужный сигнал на выходе появляется, лишь только когда ВСЕ или несколько входных сигналов цепи требуют этого, то данная цепь схемы составляется из последовательно включенных транзисторов.
4. Если нужный сигнал на выходе появляется, даже когда лишь ОДИН из нескольких входных сигналов требует этого, то данная цепь схемы составляется из параллельно включенных транзисторов.

Конструкции для самостоятельной разработки

1. Попробуйте самостоятельно с помощью этого алгоритма создать схему И-НЕ по таблице истинности и собрать ее.

Практическая схема вентиля И-НЕ на транзисторах

На монтажной схеме (рис. Ц-3.17) показано, как И-НЕ будет выглядеть на макетной плате. При этом белые провода подключены так, чтобы на входы вентиля подавалось две единицы — единственное состояние, при котором будет ноль на выходе, а светодиод загорается при появлении на выходе логического нуля.

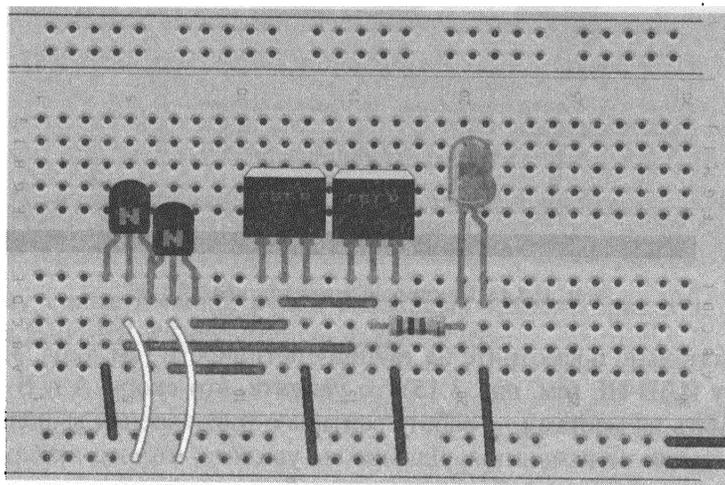


Рис. Ц-3.17. Монтажная схема вентиля И-НЕ

Практические схемы вентилях И и ИЛИ на транзисторах

Если вспомнить про взаимное преобразование ИЛИ-НЕ и ИЛИ с помощью инвертора, то, добавив по одному n- и p-канальному транзистору в схему ИЛИ-НЕ, можно получить вентиль ИЛИ (рис. 3.18).

То же самое справедливо и для логического И (рис. 3.19).

Надо полагать, вам не составит труда собрать эти два вентиля на макетной плате и проверить их работу.

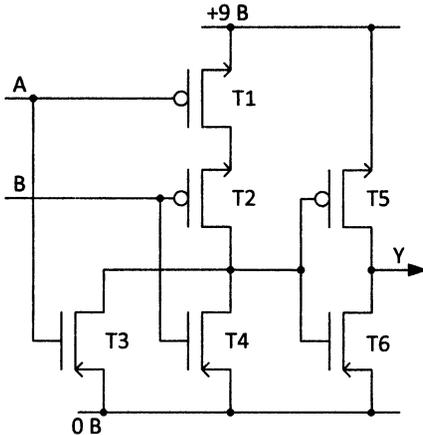


Рис. 3.18. Схема логического ИЛИ на транзисторах

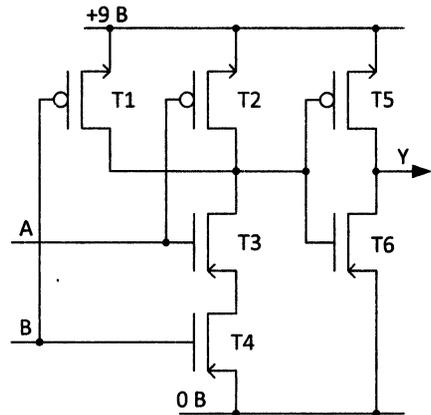


Рис. 3.19. Схема логического И на транзисторах

Именно такие схемы используются для реализации логических вентилях в микросхемах КМОП.

Практическая схема Исключающего ИЛИ на транзисторах

Из базовых элементов не была рассмотрена только схема Исключающего ИЛИ. Она не самая простая, и существует несколько вариантов ее создания с разным количеством транзисторов.

Рассмотрим классический способ. Единица на выходе должна появиться в случае, когда $A = 0$ и $B = 1$ или $A = 1$ и $B = 0$. Уже сразу возникает вопрос, как реализовать эту цепь схемы на р-канальных транзисторах, которые открываются логическим нулем, если у нас один из сигналов обязательно будет единицей. Не лучше ситуация и с n-канальной цепью, которая должна активироваться при $A = 0$ и $B = 0$ или $A = 1$ и $B = 1$. Если со вторым условием все понятно, то первое не реализовать на приборах, открывающихся логической единицей.

Выход из этой затруднительной ситуации следующий: нужно инвертировать входные сигналы и подавать на затворы транзисторов прямой или инверсный сигнал в зависимости от требуемой ситуации. Тогда схема вентиля Исключающее ИЛИ будет выглядеть так, как показано на рис. 3.20.

Здесь каждой из четырех возможных комбинаций входных сигналов соответствует одна цепочка из двух последовательно соединенных транзисторов: T1 и T2 открываются при $A = 1$ и $B = 0$, T3 и T4 — при $A = 0$ и $B = 0$, T5 и T6 — при $A = 0$ и $B = 1$, T7 и T8 — при $A = 1$ и $B = 1$. Таким способом реализуется таблица истинности этого вентиля.

Схема, воспроизводящая Исключающее ИЛИ, на макетной плате (рис. Ц-3.21) занимает весьма много места. Инверторы входных сигналов на монтажной схеме

находятся слева. От них идут связи далее. Четыре n-канальных транзистора самого вентиля сгруппированы в центре. Правее их расположены четыре р-канальных прибора. При этом разные цвета сигнальных проводников (коричневый, синий, зеленый) используются, как упоминалось ранее, лишь для удобства чтения схемы.

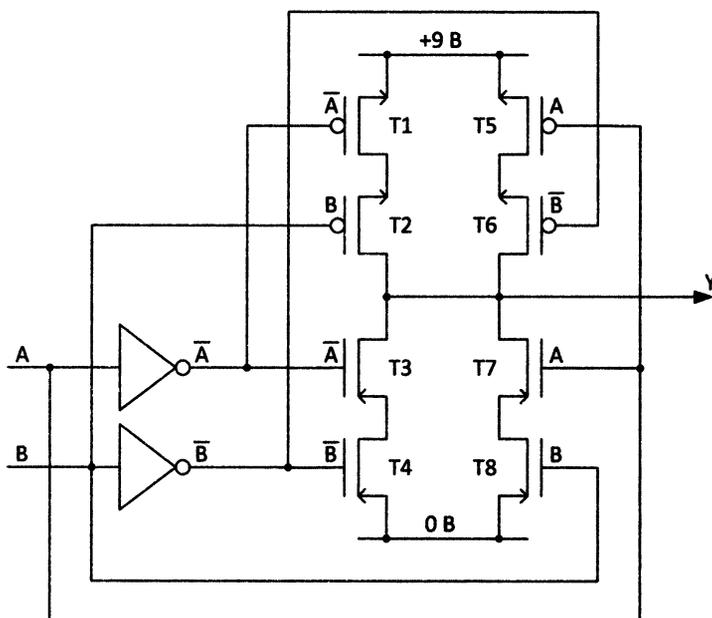


Рис. 3.20. Принципиальная схема логического Исключающего ИЛИ на транзисторах

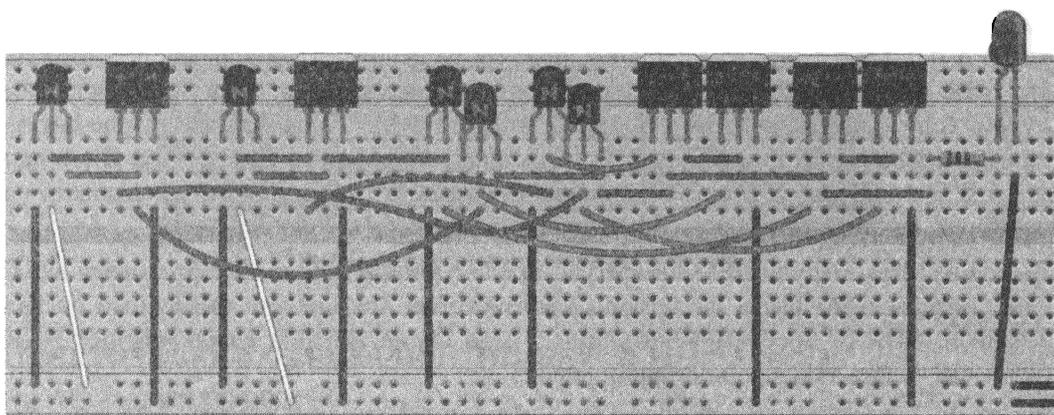


Рис. Ц-3.21. Исключающее ИЛИ на макетной плате

В отличие от способа получения ИЛИ из ИЛИ-НЕ Исключающее ИЛИ-НЕ не требует добавления транзисторов на выход схемы. Достаточно перераспределить сигналы, приходящие на затворы приборов с T1 по T8, чтобы реализовать ее на тех же 12 транзисторах.

Конструкции для самостоятельной разработки

2. Попробуйте самостоятельно создать схему Исключающего ИЛИ-НЕ на базе Исключающего ИЛИ и собрать ее на макетной плате.

Добавляем входы в логические схемы

До сих пор нами рассматривались вентили, которые получают сигналы из двух источников, т. е. двухвходовые схемы. Но как быть, если нужно объединить функцией И-НЕ три входа? Да все так же! Сначала составляется таблица истинности (табл. 3.7) — возможных комбинаций входов уже будет 8.

Таблица 3.7. Таблица истинности вентиля ЗИ-НЕ

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

А затем в соответствии с этой таблицей в схему добавляются еще два транзистора (рис. 3.22). Условное обозначение такого элемента приведено на рис. 3.23.

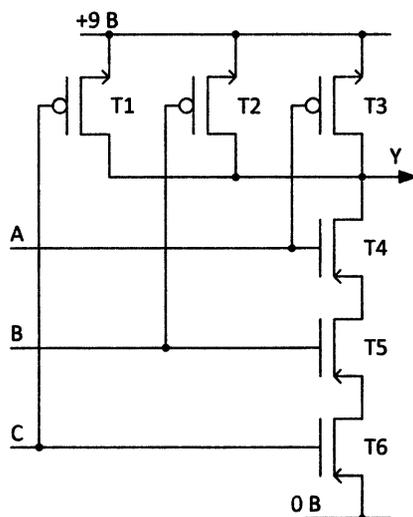


Рис. 3.22. Схема вентиля И-НЕ с тремя входами

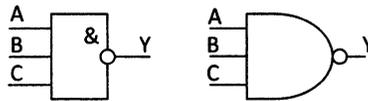


Рис. 3.23. Условное графическое обозначение вентиля И-НЕ с тремя входами: слева — российское, справа — зарубежное

УКАЗЫВАЕМ КОЛИЧЕСТВО ВХОДОВ

Как вы уже, возможно, заметили, количество состояний схемы, описываемое таблицей истинности, равняется степени двойки от числа входов. Чтобы отличать друг от друга вентили, выполняющие одну логическую функцию, но имеющие разное количество входов, в их наименование добавляют цифру, обозначающую число входов. То есть на рис. 3.22 показана схема, а на рис. 3.23 — условное обозначение элемента ЗИ-НЕ (NAND3). Аналогичным образом можно получить и схемы ЗИ, ЗИЛИ, ЗИЛИ-НЕ из ранее рассмотренных 2И, 2ИЛИ, 2ИЛИ-НЕ. Можно не ограничиваться тремя входами и добавить еще. На практике больше четырех входов в таких вентилях не применяется, потому что цепочка из большого количества последовательно включенных транзисторов переключается слишком медленно, чтобы ее можно было использовать в быстродействующих системах. Вентили типа Исключающее ИЛИ вообще используются только двухвходовые.

Создание логического вентиля по произвольной таблице истинности

Вообще, по таблице истинности можно на транзисторном уровне создать устройство, реализующее самую замысловатую логическую функцию. Рассмотрим простой пример создания вентиля для произвольной таблицы истинности (табл. 3.8).

Таблица 3.8. Таблица истинности произвольного вентиля

A	B	Y
0	0	0
0	1	1
1	0	0
1	1	0

Видно, что она не соответствует никакому вентилю из ранее рассмотренных. Значит, нужно вновь воспользоваться уже опробованным алгоритмом построения логических элементов.

P-канальные транзисторы должны соединяться последовательно и открываться при $A = 0$, $B = 1$. Значит, потребуется инверсия сигнала B. N-канальные приборы открываются в остальных трех случаях. Можно реализовать это в виде трех параллельных цепочек, состоящих из двух транзисторов (рис. 3.24). Выходы инверторов и затворы транзисторов не соединены друг с другом явным образом на схеме, но одинаковые подписи означают, что эти контакты на плате соединяются. Так иногда

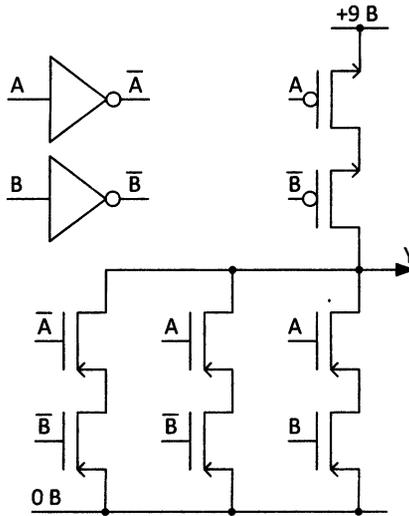


Рис. 3.24. Первая попытка построить вентиль по произвольной таблице истинности

делают, чтобы не запутаться в большом количестве проводов на схеме. Важно, чтобы все «оборванные» цепи были аккуратно подписаны.

Но такая схема содержит в себе избыточные транзисторы. Для их исключения можно отметить, что в таблице истинности ясно прослеживается при внимательном рассмотрении: если $A = 1$, то сигнал на входе B не имеет значения, выход Y будет равен 0. И еще одна аналогичная особенность: если $B = 0$, то $Y = 0$ при любом A. Значит, в p-канальной цепи остаются всего два транзистора. И внутри схемы тогда требуется лишь один инвертор вместо двух (рис. 3.25).

Возможно, вы уже сами увидели, что у нас получился вентиль 2ИЛИ-НЕ с инверсией входа B, который так и обозначается (рис. 3.26).

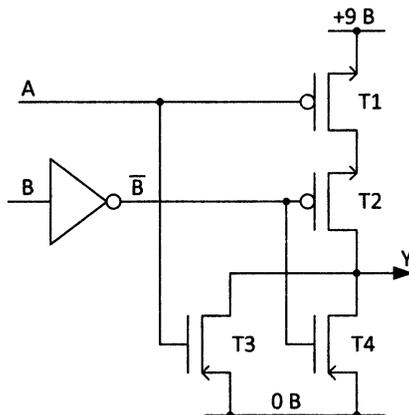


Рис. 3.25. Результат упрощения схемы вентиль по произвольной таблице истинности

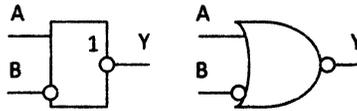


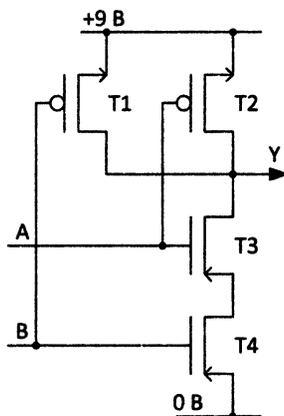
Рис. 3.26. Условное графическое обозначение вентиля 2ИЛИ-НЕ с инверсией входа В: слева — российское, справа — зарубежное

То, что было сделано со схемой в результате анализа таблицы истинности, называется *минимизацией логической функции*. Здесь она оказалась легкой и очевидной, т. к. был выбран простой двухвходовой элемент. Научиться же минимизации в сложных задачах с большим количеством входных сигналов лучше, узнав хорошо булеву алгебру. Ее раздел по преобразованию и минимизации логических функций содержит очень изящные методы с применением карт Карно и других интересных способов. Пока же на время нужно отложить рассмотрение цифровых схем на транзисторном уровне и перейти к созданию более крупных устройств из готовых вентилях.

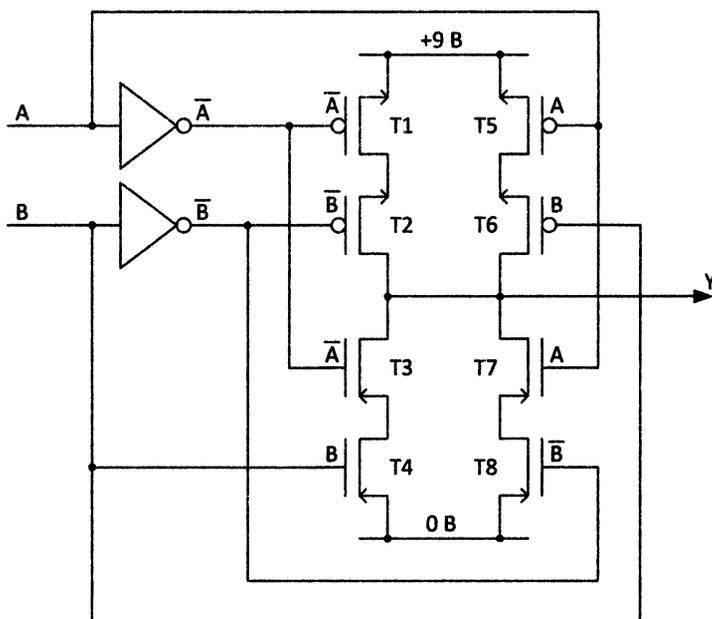
Ответы на вопросы для самопроверки

Номер вопроса	Ответ															
1	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>A</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	Y	0	0	1	1									
A	Y															
0	0															
1	1															
2	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>A</th> <th>B</th> <th>Y</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Y														
0	0	1														
0	1	1														
1	0	1														
1	1	0														
3																
4	ИЛИ-НЕ															
5	Логическая единица															

Ответы на задания по конструкциям для самостоятельной разработки

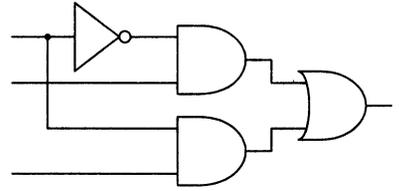


Ответ на задание 1



Ответ на задание 2

ГЛАВА 4



Микросхемы цифровой логики

Вам потребуются:

- один транзистор типа n-МОП и 2 транзистора типа p-МОП;
- 2 светодиода;
- 2 резистора сопротивлением 100 Ом;
- 5 резисторов сопротивлением 10 кОм;
- одна кнопка;
- один счетверенный DIP-переключатель;
- по одной микросхеме CD4001BE, CD4011BE, CD4012BE, CD4069UBE, CD4072BE, CD4077BE, CD4081BE, CD4511BE;
- соленоид TAU-0520.

Общежитие для транзисторов: что важно знать о микросхемах цифровой логики

Построение вентилях на макетной плате в предыдущей главе завершилось созданием Исключающего ИЛИ. 12 транзисторов вместе с многочисленными проводами заняли очень много места. Разумеется, на дискретных приборах даже на большой макетной плате никакое устройство, содержащее более двух-трех вентилях, собрать нельзя. Чтобы двигаться дальше в практическом изучении цифровой электроники, необходимо воспользоваться возможностями, которые предоставляют *интегральные микросхемы*. У них внутри одного корпуса со сравнительно небольшим количеством выводов находится много транзисторов, формирующих сразу несколько логических вентилях. Поскольку вы уже разобрались с тем, как они функционируют на транзисторном уровне, и даже научились сами их конструировать, можете быть уверены, что переход на следующий уровень проектирования не вызовет затруднений.

Микросхемы цифровой логики изготавливаются в виде серий, имеющих свое обозначение. Что подразумевается в микроэлектронике под словом «серия»? Во-

первых, тип логики. В этой книге рассматриваются только устройства КМОП. Поэтому все другие микросхемы (ТТЛ, ТТЛШ, ЭСЛ, И²Л и прочие) и соответствующие им серии остаются за бортом. Во-вторых, даже внутри одного типа бывают подгруппы интегральных микросхем, отличающиеся по самым разным параметрам, среди которых минимальное время переключения, допустимое напряжение питания, максимальная емкость нагрузки и др. Поэтому между собой микросхемы из разных серий могут взаимодействовать не очень хорошо. А вот внутри одной серии они адаптированы к работе друг с другом идеально. То есть можно быть уверенным, что если схема работает не так, как ожидается, то причина кроется исключительно в ошибке разработчика, а не в проблемах сопряжения разнородных микросхем.

В качестве объекта для экспериментов была выбрана недорогая КМОП-серия CD4000BE (ее отечественный аналог — КР1561). Нули в наименовании каждой микросхемы заменяются цифровым кодом, указывающим на ее функциональное назначение. В *приложении 5* приведена таблица соответствия обозначений разных видов микросхем, применяемых в этой книге, в российском и зарубежном стандартах. Сразу нужно отметить, что выходной ток у микросхем этой серии слишком слаб, чтобы зажечь светодиод от выхода вентиля, как это делалось в предыдущей главе. Поэтому для индикации состояния узлов схем нужно использовать те самые пробники, которые были сконструированы в самом начале (см. *главы 2 и 3*), подключаясь гибким проводом к интересующей точке. При соединении микросхем друг с другом выходного тока одной из них достаточно, чтобы переключать транзисторы сразу в нескольких подобных, если потребуется. А вот для управления такой нагрузкой, как светодиод, не говоря уже о чем-то более мощном, требуется использовать усилитель сигнала — например, на МОП-транзисторе, как в пробнике. В этом случае можно управлять даже небольшим электродвигателем для моделей. Лишь бы максимально допустимый ток стока усилительного транзистора был больше тока нагрузки. Напряжение питания КМОП-логики может меняться в широком диапазоне — как правило, от 5 до 15 вольт. Поэтому все устройства в этой и последующих главах по-прежнему будут получать питание от батарейки «Крона».

Микросхемы серии CD4000BE, которые используются в примерах этой книги, имеют пластиковые корпуса типа DIP (dual in-line package) — двухрядные. Это прямоугольные коробочки с выводами, расположенными вдоль длинных сторон. Возле одной из коротких сторон сделана полукруглая выемка — ключ, обозначающий первый вывод. Если расположить микросхему ключом вверх, то нумерация выводов будет начинаться слева от ключа и продолжаться против часовой стрелки, как показано на рис. 4.1. Такой корпус позволяет как использовать макетную плату, так и припаивать выводы микросхемы к контактным площадкам печатной платы, вставив их в просверленные для этого отверстия. Существуют корпуса и для поверхностного монтажа микросхем с помощью пайки (тип SO и другие). Их невозможно установить в макетную плату, поэтому при покупке надо обязательно обращать внимание на тип корпуса, обозначаемый буквами BE в конце названия микросхем импортной серии или КР в начале названия отечественной.

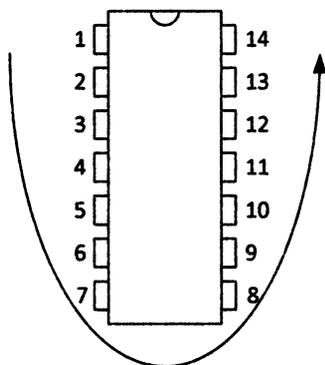


Рис. 4.1. Нумерация выводов DIP-микросхемы

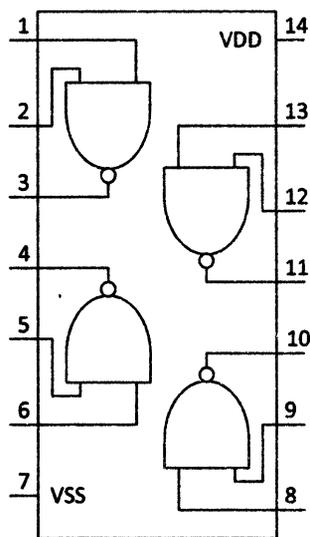


Рис. 4.2. Цоколевка микросхемы CD4011BE

Так же, как и в случае с транзисторами, в документации на микросхему необходимо найти ее цоколевку. Для простых логических элементов ее иногда изображают в виде корпуса, внутри которого находятся условные обозначения вентиляей, а их входы и выходы маркированы числами в соответствии с нумерацией выводов. При этом, если не указано иное, все вентиля в микросхеме одинаковы. Для примера на рис. 4.2 показана цоколевка микросхемы CD4011BE. Обозначение VSS соответствует минусу питания, а VDD — плюсу. Эти обозначения, применяемые в зарубежной литературе, запомнить просто. Буква V означает напряжение, удвоенная S — направление истока (source) n-канального транзистора, а удвоенная D — направление его стока (drain). Как вы помните, исток такого типа приборов «смотрит» на минус питания, а сток — на плюс. Аналогично расположены входы и выходы для микросхем других двухвходовых вентиляей: CD4001BE, CD4072BE, CD4081BE, CD4070BE, CD4077BE.

На макетной плате DIP-микросхема располагается над канавкой вдоль, чтобы два ряда выводов оказались по разные стороны от нее. Кстати, сразу следует отметить два момента, связанных с установкой этих компонентов на плату и их демонтажем. Во-первых, выводы новой микросхемы необходимо слегка подогнуть, т. к. они отходят от корпуса вниз под непрямым углом и изначально не совпадают с нужными отверстиями. Для этого можно взять ее за короткие стороны, положить длинной стороной на стол и с легким нажимом вниз довернуть корпус до прямого угла с поверхностью стола. Потом повторить эту же процедуру для выводов с другой стороны. Во-вторых, при вынимании микросхемы из зажимов контактных отверстий ее обязательно нужно тянуть или поддевать за обе короткие стороны одновременно — иначе при извлечении крайних выводов произойдет их изгибание. Конечно, их можно выпрямить пинцетом. Но после нескольких подобных процедур они просто-напросто отломятся. А поддевание сразу за оба края направляет общее усилие вертикально вверх и не приводит к деформации выводов.

В литературе также упоминается, что КМОП-компоненты могут быть необратимо повреждены разрядом статического электричества, который произойдет при касании вывода микросхемы пальцем наэлектризованного человека. Проскочившая искра может создать разность потенциалов в единицы киловольт, что приведет к пробое затворного диэлектрика и к выходу МОП-транзистора из строя. Поэтому выводов микросхемы лучше не касаться. Впрочем, на практике в бытовых условиях с такими повреждениями встретиться вряд ли доведется. Особенно учитывая тот факт, что все входы современных микросхем в обязательном порядке снабжаются схемами защиты от статического электричества. Поэтому можно не бояться касания выводов микросхем руками.

Приступаем к опытам с микросхемами

Соединим вместе входы логических элементов

В качестве первого эксперимента над микросхемами можно посмотреть, что произойдет с вентилями И-НЕ и ИЛИ-НЕ, если их входы накоротко замкнуть между собой. В таблице истинности такой схемы исчезнут все строки, в которых входные сигналы разные. И останутся только две: «все нули» и «все единицы». Понятно, что при этом оба типа вентиляей независимо от количества входов превратятся в обычные инверторы.

На рис. Ц-4.3 приведена такая схема на базе микросхемы 2И-НЕ CD4011BE, о чем говорит номер на ее корпусе (здесь и далее на всех монтажных схемах микросхемы установлены над канавкой ключом влево, т. е. номер микросхемы виден на иллюстрациях рядом с ключом). Шины питания над микросхемой и под ней объединяются при помощи двух вертикальных перемычек, что позволяет удобно подавать плюс и минус питания на любые точки схемы. Проводник от объединенных входов вентиля подведен к минусу питания, чтобы на выходе была логическая единица.

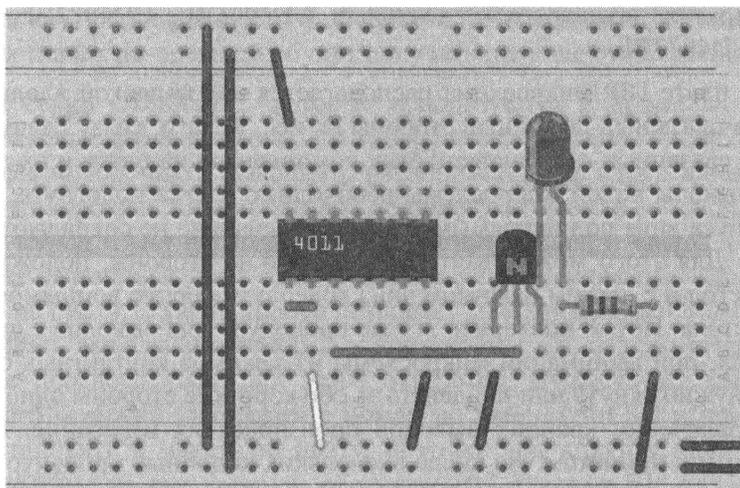


Рис. Ц-4.3. Инвертор на базе микросхемы 2И-НЕ CD4011BE

Тогда пробник, сконструированный, как можно увидеть на схеме, на базе n-канального транзистора, включит светодиод. Заменяв на плате микросхему CD4011BE на CD4001BE и оставив остальную часть без изменений, можно получить точно такой же вентиль.

Знание того, что инвертор можно получить объединением входов И-НЕ или ИЛИ-НЕ, позволяет гибко использовать имеющуюся элементную базу. И если на плате находится одна из таких микросхем, а в ней есть свободный вентиль, то можно не применять логический элемент НЕ из шести, входящих в состав CD4069UBE, а задействовать уже установленные компоненты. И вот еще что: хотя каждая из примененных микросхем (CD4011BE и CD4001BE) содержит по четыре вентиля своего типа, в нашем эксперименте было задействовано по одному из них, а остальные не потребовались. Выводы, соответствующие полностью неиспользованным логическим элементам, можно оставлять не подключенными ни к чему.

Добавим недостающие входы

В предыдущей главе рассматривались принципы построения многоходовых вентилях на транзисторах. Существуют и логические микросхемы с элементами, имеющими от двух до четырех входов. Но что делать, если требуется, например, микросхема 3И, а в наличии есть только 2И? Внутри не проникнуть, транзисторы не добавить. Как же быть? Для увеличения числа входов можно соединять элементы в каскады вентилях, следующих друг за другом. В таких случаях говорят, что применяется *каскадирование*. Так, если с помощью операции И объединить сначала два входа, а потом к этому объединению добавить с помощью логического И третий, то это будет то же самое, что применить операцию И ко всем трем входам одновременно. Например, из двух вентилях 2И можно получить 3И, как показано на рис. 4.4.

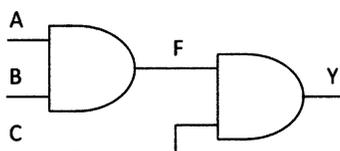


Рис. 4.4. Вентиль 3И на базе 2И

Промежуточный сигнал, обозначенный F, будет равен единице только в том случае, если единицы поданы на входы A и B. В свою очередь, выходной сигнал Y будет единицей, только если и $F = 1$, и $C = 1$. Поэтому такая схема по выполняемой функции: $Y = F \cdot C = A \cdot B \cdot C$ — соответствует вентилю 3И. При этом, как легко можно убедиться, в каком именно порядке подключать сигналы к такой схеме, с точки зрения логики ее работы значения не имеет. Точно так же из трех вентилях 2И микросхемы CD4081BE нетрудно получить 4И, как показано на рис. 4.5. На монтажной схеме (рис. Ц-4.6) входы A, B, C и D подключены к плюсу питания, что обеспечивает единственную комбинацию сигналов, создающую логическую единицу на выходе.

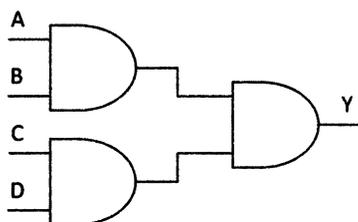


Рис. 4.5. Вентиль 4И на базе 2И

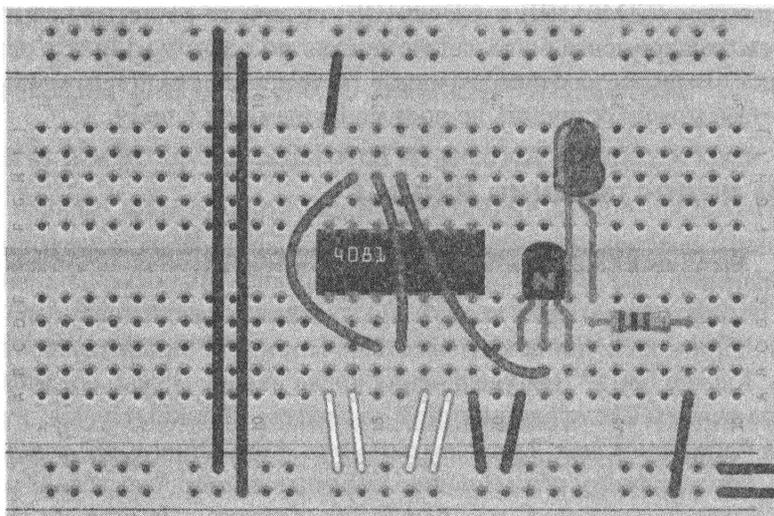


Рис. Ц-4.6. Монтажная схема вентиля 4И на базе 2И

Вентиль 4И также легко конструируется из одного 2И и одного 3И двумя способами, показанными на рис. 4.7.

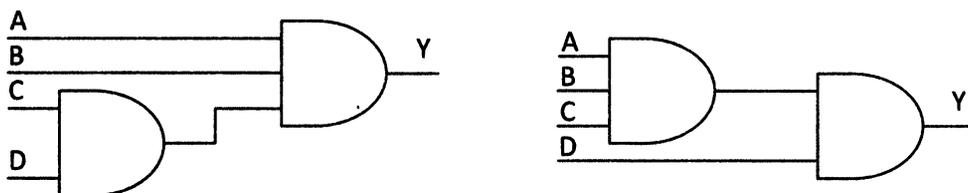


Рис. 4.7. Способы получения 4И из вентилях 2И и 3И

Пользуясь подобными правилами каскадирования, можно создавать вентили с любым количеством входов — например, объединяющий один байт данных вентиль 8И, древовидная схема которого показана на рис. 4.8.

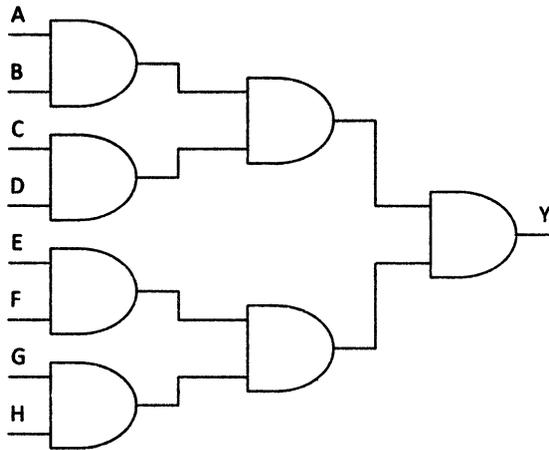


Рис. 4.8. Вентиль 8И

Вопросы для самопроверки

1. Составьте схему получения 4ИЛИ из нескольких вентилях 2ИЛИ.
2. Составьте схему получения 3ИЛИ из нескольких вентилях 2ИЛИ.

* * *

А вот добавить входы в элементы И-НЕ и ИЛИ-НЕ с помощью вентилях будет несколько сложнее. Например, как сконструировать 4И-НЕ из 2И-НЕ? Трех вентилях 2И-НЕ для этого недостаточно. Если поставить их по такой же каскадной схеме, как это делалось для 4И, то на входы второго каскада сигналы придут инвертированными, и таблица истинности такой схемы будет соответствовать элементу, попарно объединяющему входы по И, а потом результаты для каждой пары по ИЛИ. Проверьте это, составив таблицу истинности и перебрав все возможные 16 состояний схемы, приведенной на рис. 4.9.

Знание булевой алгебры и формул преобразования двоичных выражений позволяет обойтись без построения больших таблиц и перебора значений. Но метод с таблицами, как наиболее наглядный, используется в технике наравне с формулами, и в дальнейшем к нему не раз придется обращаться. Когда создается какая-то сложная схема, состоящая из множества вентилях, всегда полезно выполнять проверку ее логической функции, составив таблицу истинности.

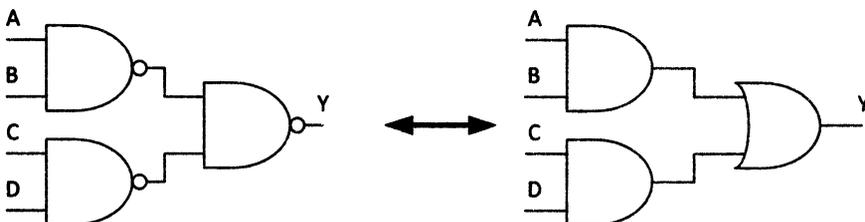


Рис. 4.9. Неудачная попытка получить 4И-НЕ из трех 2И-НЕ

Поскольку для создания 4ИЛИ-НЕ необходимо нейтрализовать инверсию выходов вентилях, образующих первый каскад обработки сигналов, придется между имеющимися каскадами добавить еще один, промежуточный, состоящий из инверторов (рис. 4.10). Как вы помните, они превратят 2И-НЕ первого каскада в 2И. Инверторы можно взять как из отдельной микросхемы CD4069UBE, так и уже известным путем преобразования 2И-НЕ в НЕ. Аналогичным способом из двух 2И-НЕ и одного инвертора можно получить 3И-НЕ, а из 2ИЛИ-НЕ и одного инвертора — 3ИЛИ-НЕ.

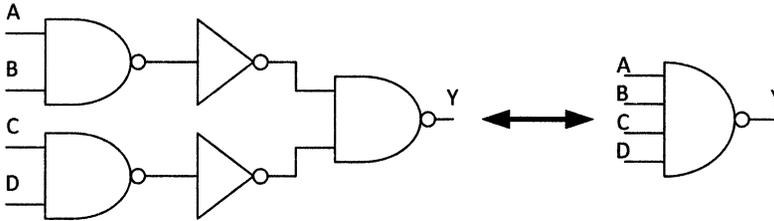


Рис. 4.10. Правильная схема получения 4И-НЕ из 2И-НЕ

Вопросы для самопроверки

3. Составьте схему получения 4ИЛИ-НЕ только из 2ИЛИ-НЕ.
4. Составьте схему получения 3ИЛИ-НЕ только из 2ИЛИ-НЕ.

* * *

Другой очевидный вариант построения многовходовых И-НЕ и ИЛИ-НЕ — собрать аналогичные по числу входов вентили И или ИЛИ, а затем добавить инвертор на выход. Как видите, существует масса способов добиться требуемого результата с помощью ограниченного набора микросхем. Это удобно и позволяет в ряде случаев не добавлять лишние компоненты на плату, а использовать имеющиеся.

Из всего многообразия базовых функций остались нерассмотренными только Иключающее ИЛИ и Иключающее ИЛИ-НЕ. Как за счет каскадирования увеличить у них количество входов? Если последовательно включить вентили Иключающее ИЛИ, как показано на рис. 4.11, то сигнал на выходе этой схемы будет равен единице, только если на нечетное число входов были поданы единицы.

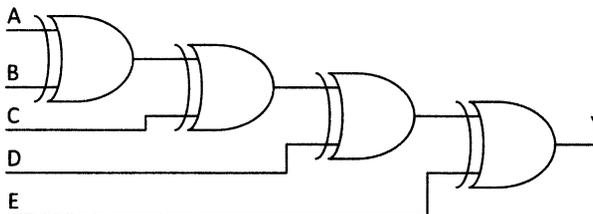


Рис. 4.11. Схема четности на 5 битов

Такую схему будет более точно назвать *схемой контроля четности*. Подобная последовательная многокаскадная структура будет пригодной при любом количестве входов, но при этом она окажется несбалансированной для быстродействующих систем. Поскольку каждый вентиль имеет определенную задержку срабатывания (ее в официальной документации называют *задержкой распространения*), получается, что сигналы с разных входов добираются до выхода всей схемы за разное время. В случае с четным количеством сравниваемых битов более эффективной оказывается древовидная схема, показанная на рис. 4.12.

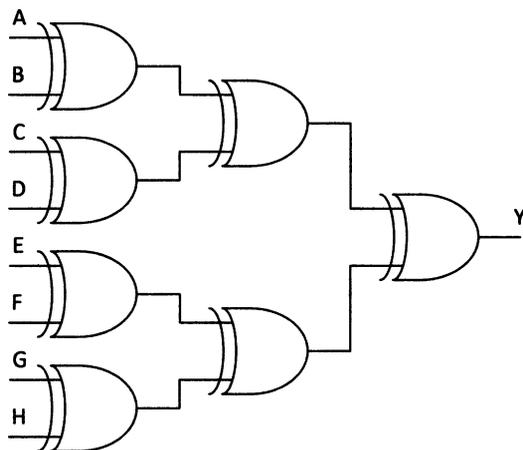


Рис. 4.12. Схема четности на 8 битов

Можно при необходимости применять комбинацию этих двух подходов. Многовходовое Исключающее ИЛИ-НЕ строится тоже на базе вентилей Исключающее ИЛИ, а на ее выход добавляется инвертор.

ЕЩЕ О КОНТРОЛЕ ЧЕТНОСТИ

Контроль четности применяется для проверки правильности передачи и хранения данных. К каждому многобитному слову данных добавляется *бит четности*, вычисляемый подобными схемами. Он же называется *контрольным разрядом*. Если в результате какого-либо сбоя произойдут искажения в одном или нескольких битах слова данных, то при одиночной, тройной или любой другой нечетной ошибке изменится значение контрольного разряда. Это обнаруживается схемой сравнения, проверяющей полученную информацию. К сожалению, при четном количестве ошибок значение бита четности не изменяется. Но вероятность двойных и многократных ошибок намного ниже, чем вероятность одиночных сбоев, поэтому такой довольно простой метод эффективен для обнаружения проблем и часто применяется на практике.

Инвертируем входы

До сих пор в этой главе рассматривались вопросы каскадирования логических вентилей одинаковых типов. Разумеется, совсем не обязательно ограничиваться только ими. Можно вернуться к ИЛИ-НЕ с инверсией на входе В, схема которого была разработана на транзисторном уровне в *главе 3*. Его условное обозначение повторно приведено на рис. 4.13.

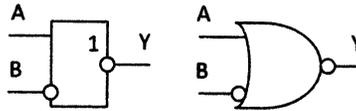


Рис. 4.13. ИЛИ-НЕ с инверсией входа В

Такого вентиля среди микросхем серии CD4000BE нет. Но его можно сконструировать из инвертора и стандартного ИЛИ-НЕ. Причем, обойтись можно всего одной микросхемой CD4001BE, но для примера была задействована еще и CD4069UBE (рис. 4.14 и Ц-4.15).

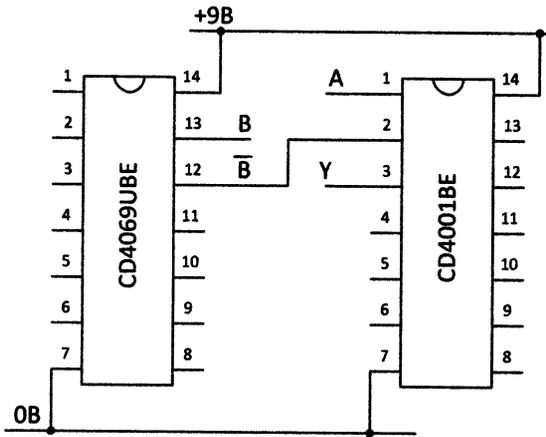


Рис. 4.14. Вентиль 2ИЛИ-НЕ с инверсией входа В на микросхемах

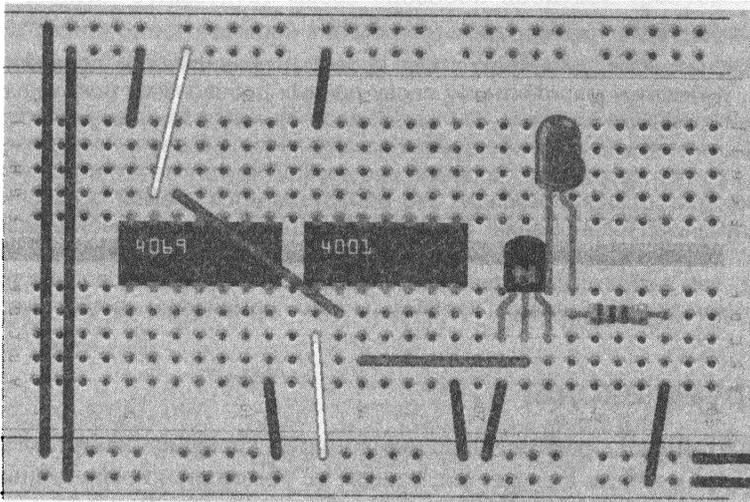


Рис. Ц-4.15. Монтажная схема вентиля ИЛИ-НЕ с инверсией входа В

Конструкции для самостоятельной разработки

1. Составьте схему получения 3И на базе CD4081BE.
2. Составьте схему получения 2ИЛИ на базе CD4001BE.

Что делать с ненужными входами?

Бывает так, что в одном корпусе остаются неиспользованными не целые вентили, а только часть их входов. Например, микросхема CD4012BE состоит из двух четырехвходовых И-НЕ. А вам требуется трехвходовой вентиль. Как быть в этом случае? Неиспользуемые входы КМОП-логики должны быть присоединены к одному из используемых или к питанию. Если оставить их неподключенными, то неизвестно, какой потенциал будет у затвора транзистора на этом входе в момент включения. Да и в процессе работы свободный вход будет вести себя как антенна, улавливая помехи, возникающие от изменения сигналов на плате. И вполне возможно, что произойдет незапланированное переключение вентиля. Поэтому лишние входы используемых вентилях однозначно не должны, как говорят инженеры, «болтаться в воздухе». Вопрос лишь в том, какой сигнал на них нужно подать, чтобы не нарушить логическую функцию. Так, например, если свободный вход упомянутой микросхемы CD4012BE подключить к нулю, то на выходе в соответствии с таблицей истинности И-НЕ никаким образом не удастся получить ноль. Поэтому у вентилях И и И-НЕ неиспользованные выводы должны быть привязаны к логической единице, а у ИЛИ и ИЛИ-НЕ, соответственно, к нулю. Здесь не нужен никакой резистор, можно подключить их непосредственно к питанию. Другой способ решения проблемы лишних входов — подать один из входных сигналов не на один, а на несколько входов сразу. В этом случае несколько возрастет нагрузка на источник сигнала, но это играет заметную отрицательную роль только в быстродействующих применениях. Поэтому вы во всех примерах этой книги можете использовать тот способ, который окажется наиболее удобным. У Исключающих ИЛИ неподключенных выводов быть не может, поскольку это элемент сравнения двух сигналов. А вот выходы КМОП-схем можно оставлять неподключенными без всяких последствий. Правда, тогда в случае с простыми вентилями с одним выходом возникает резонный вопрос: а зачем он нужен в схеме, если сигнал с его выхода никуда дальше не идет? Однако более сложные элементы, с которыми вам еще предстоит познакомиться далее, имеющие несколько выходов, часто подключаются так, что часть из них не используется.

ЗАКАЗНОЕ ПРОЕКТИРОВАНИЕ И БИБЛИОТЕКИ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ

Вернемся немного назад к разработке устройств на транзисторном уровне. Легко подсчитать, что для создания 4И нужно 10 полупроводниковых приборов-транзисторов. Если же вентиль 4И собирается из трех готовых 2И, то количество транзисторов в такой схеме будет равно 18. Тем не менее, когда имеешь дело с дискретными компонентами, где отдельный транзистор по размеру очень большой, то реализация многовходовых элементов на микросхемах все равно выигрывает по площади, потому что все транзисторы там без корпусов, и провода проложены над ними в нескольких слоях, называемых *слоями металлизации*. Впрочем, особенно важные или специфические блоки могут по интегральной технологии создаваться из заданных комбинаций

транзисторов. Такой метод, называемый *заказным проектированием*, позволяет получить параметры блоков по площади, мощности или быстродействию лучше, чем в случае создания их из готовых вентиляей. Но при этом он требует больших трудозатрат и очень высокой квалификации разработчиков. Поэтому в разработке микросхем с десятками миллионов транзисторов вместе с заказным проектированием используются и готовые вентиляи — под разные технологические процессы создаются целые библиотеки логических элементов, содержащие по несколько сотен разных их типов и размеров. А из этих вентиляей с помощью специализированных программ синтезируется вся микросхема. Так что, изучив цифровую электронику на макетной плате, вы в будущем можете стать инженерами-проектировщиками новых микропроцессоров и суперкомпьютеров.

Вопросы для самопроверки

5. Можно ли оставлять неподключенными входы логических элементов?
6. Можно ли оставлять неподключенными выходы логических элементов?

Кодовый замок: от идеи до готовой конструкции

Замысел

До сих пор в этой главе не было собрано ни одного законченного устройства на вентиляях, а только изучались их свойства и некоторые правила преобразования. Настало время попробовать собрать из рассмотренных микросхем кодовый замок для сейфа. Конечно, он весьма простой, но, тем не менее, самый настоящий, и его можно вмонтировать в подходящий ящик с крышкой или дверцей на петлях. Правда, для этого придется приобрести устройство, которое будет выполнять роль электромагнитного запора, управляемого цифровой схемой, — соленоид с подвижным сердечником.

Те, кто уже изучал электромагнетизм, знают, что если в катушку из провода, через который протекает ток, поместить стальной сердечник, то в зависимости от направления тока в катушке сердечник может втягиваться в катушку или выталкиваться из нее. Такая катушка-соленоид продается в самых разных исполнениях. Подойдет, например, модель ТАУ-0520, продаваемая Амперкой. Этот соленоид потребляет ток порядка 300 мА, поэтому n-канальный транзистор 2N7000ТА с его предельно допустимым током стока 200 мА можно использовать, только поставив сразу две штуки параллельно. То есть соединив друг с другом попарно их стоки, затворы и истоки. А можно воспользоваться одним р-канальным транзистором IRFU9024NPBF с более мощным предельным током стока, не забыв про полярность сигнала, поступающего на его затвор.

В разрабатываемом замке ток будет протекать через соленоид только в случае набора на переключателях правильной кодовой последовательности. При этом сердечник, выполняющий роль ригеля (язычка) замка, втянется внутрь на несколько миллиметров, позволяя открыть дверцу. Если ток через соленоид не течет, то возвратная пружина выталкивает сердечник обратно, запирая сейф. В документации на

соленоид и в маркировке на его корпусе указано, что напряжение, приложенное к его выводам, должно быть 12 вольт. Однако он работает и от свежей батарейки «Крона». В случае, когда батарея уже использовалась, и напряжение на ней стало меньше 9 В, соленоид начинает плохо втягивать сердечник. Поэтому его придется расположить вертикально — так, чтобы сила тяжести при срабатывании помогала соленоиду сжимать возвратную пружину. Для изучения и испытания схемы такой способ годится, а для создания надежно работающего окончательного варианта устройства лучше подыскать источник питания на 12 В. КМОП-микросхемы серии CD4000BE могут работать в широком диапазоне напряжений от 5 до 15 вольт без каких-либо ограничений. В этом заключается одно из преимуществ этого класса цифровых схем в сравнении с ТТЛ, ЭСЛ и другими, требующими строго выдерживать только одно значение напряжения питания.

Приступаем к конструированию схемы

Итак, очевидно, что электронный замок должен сравнивать значения, задаваемые переключателями, с предварительно установленной кодовой последовательностью. Код будет четырехбитным. Конечно, его легко подобрать простым перебором, ведь у такого замка существует всего 16 возможных комбинаций. Но на базе этой конструкции можно потом разработать и собрать более сложные варианты. Как вы помните, идеальные вентили для сравнения цифровых сигналов — Исключающее ИЛИ и Исключающее ИЛИ-НЕ. В микросхеме CD4077BE как раз находятся четыре вентили Исключающего ИЛИ-НЕ. В зависимости от того, какой код вы хотите запрограммировать в замке, один вход каждого вентилля жестко подключается проводником к плюсу или к минусу питания. А второй — при помощи переключателей, как показано на схеме (рис. 4.16). На ней, если считать вентилля сверху вниз, задан двоичный код 1010, т. е. десять. Одному положению переключателя соответствует

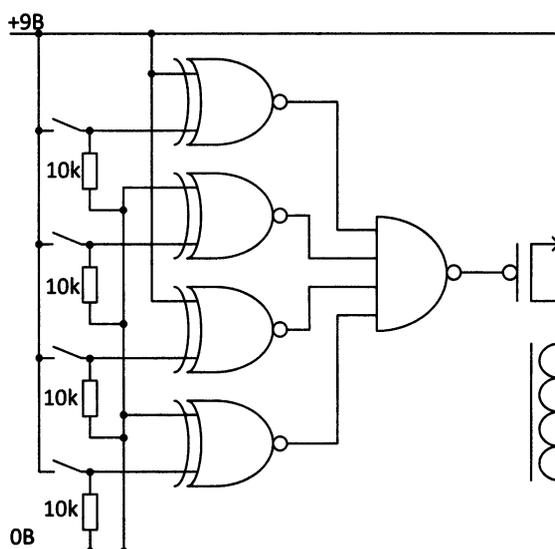


Рис. 4.16. Базовая схема кодового замка

логический ноль, а другому — единица. При подаче правильной комбинации на все четыре входа на выходах каждого вентиля будет единица. Сигналы с микросхемы CD4077BE поступают на входы микросхемы 4И-НЕ CD4012BE. На ее выходе ноль появится только в том случае, если все переключатели стоят в правильном положении. Этот сигнал уже можно непосредственно подавать на затвор р-канального транзистора, управляющего током соленоида. Его стандартное обозначение на схеме — волнистая линия из четырех полуокружностей с расположенной рядом чертой — соответствует его физическому воплощению: катушка из провода с сердечником внутри.

Доработка схемы: экономим энергию

Однако показанное на рис. 4.16 решение имеет серьезный недостаток. При наборе правильной кодовой комбинации ток непрерывно течет через соленоид, а так не рекомендуется делать, чтобы не испортить его и не разрядить батарею за несколько минут. Поэтому схему необходимо немного доработать: добавить в нее сигнал, который будет называться «разрешение на введение кода» (рис. 4.17). Пока кнопка, устанавливающая этот сигнал в активное состояние, не нажата, даже правильная кодовая комбинация не откроет замок. Сигнал, генерирующийся вентилем 4И-НЕ, имеет активный уровень, т. е. логический уровень, открывающий замок, равный нулю. Если и сигналу разрешения (по-английски enable, поэтому его обозначение на схеме — En) задать активный уровень, равный нулю, то с помощью вентиля ИЛИ можно объединить эти два сигнала в общий, передающийся на затвор р-канального мощного транзистора. Как вы помните, ИЛИ способен выдать ноль только тогда, когда на все его входы подан низкий уровень. Чтобы явно указать на значение активного уровня, обозначение сигнала En на схеме (см. рис. 4.17) дополняется

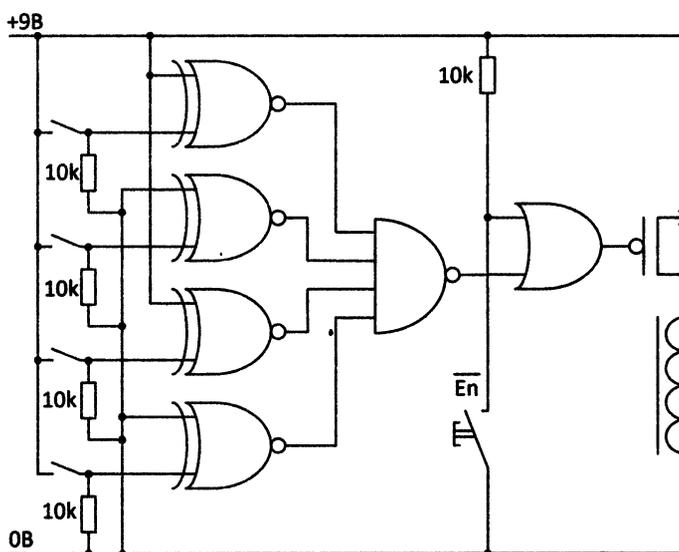


Рис. 4.17. Схема кодового замка с сигналом разрешения открывания

чертой сверху. Такая схема будет подавать ток на соленоид только в том случае, если одновременно введен правильный код и нажата кнопка разрешения. Стоит отпустить кнопку после открывания дверцы сейфа, как на вход вентиля ИЛИ поступит единица, она же появится на его выходе, и транзисторный ключ закроется.

В распоряжении автора книги не было микросхемы CD4071BE, содержащей 4 вентиля 2ИЛИ, зато имелась CD4072BE с двумя четырехходовыми ИЛИ. Поэтому ее выводы с номерами 2 и 4 были присоединены к минусу питания. Выводы кнопки, вставленной в макетную плату, совпадают с пятым и седьмым выводами последней микросхемы. На монтажной схеме (рис. Ц-4.18) в кодовом замке используется так называемый *DIP-переключатель*. Такие переключатели бывают с разнообразным количеством тумблеров, здесь — с четырьмя. По приставке к названию понятно, что расположение выводов у этого переключателя такое же, как и у микросхем. В верхнем положении тумблера соответствующий ему переключатель замыкается, о чем напоминает надпись «ON» с этой стороны корпуса. Конечно, вместо него можно использовать и обычные переключатели или даже просто провода, но такой вариант исполнения будет более аккуратным и компактным.

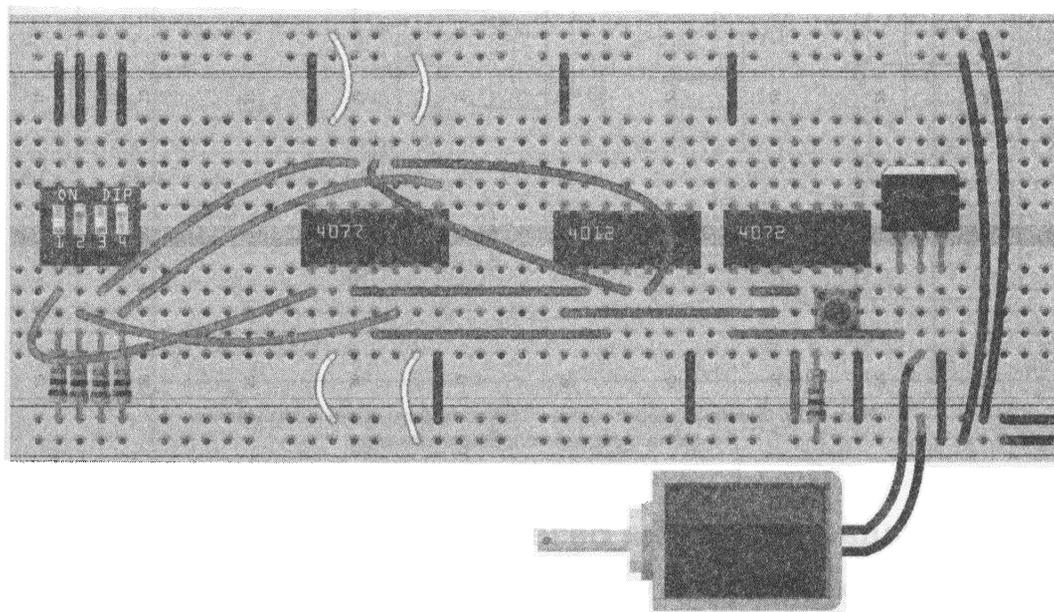


Рис. Ц-4.18. Монтажная схема кодового замка

Оборудуем замок сигнализацией

Схему замка можно дополнить световой сигнализацией на случай попытки открыть его неправильным кодом. Светодиод будет загораться в случае, если кнопка разрешения нажимается при неверном коде, т. е. когда на выходе 4И-НЕ — логическая единица. Этот сигнал нужно инвертировать и подать на вход другого вентиля ИЛИ, который уже будет управлять транзистором, подающим ток на светодиод (рис. 4.19).

Чтобы не добавлять лишних деталей на плату, инвертор можно получить из незадействованного вентиля 4И-НЕ микросхемы CD4012BE.

Если злоумышленник попытается открыть сейф, вынув батарею питания, то своей цели он не достигнет. Соленоид втягивает сердечник только при срабатывании схемы кодового замка. А в отсутствие питания он находится в положении, запирающем дверцу. Так что и разрядившаяся батарея не станет причиной утраты секретных материалов. Конструкцию замка можно улучшить, если вместо перемычек, устанавливающих правильный код, установить второй комплект из DIP-переключателя и резисторов внутри сейфа. Тогда код можно будет легко менять каждый раз перед закрыванием дверцы. Главное при этом — не забыть его самому!

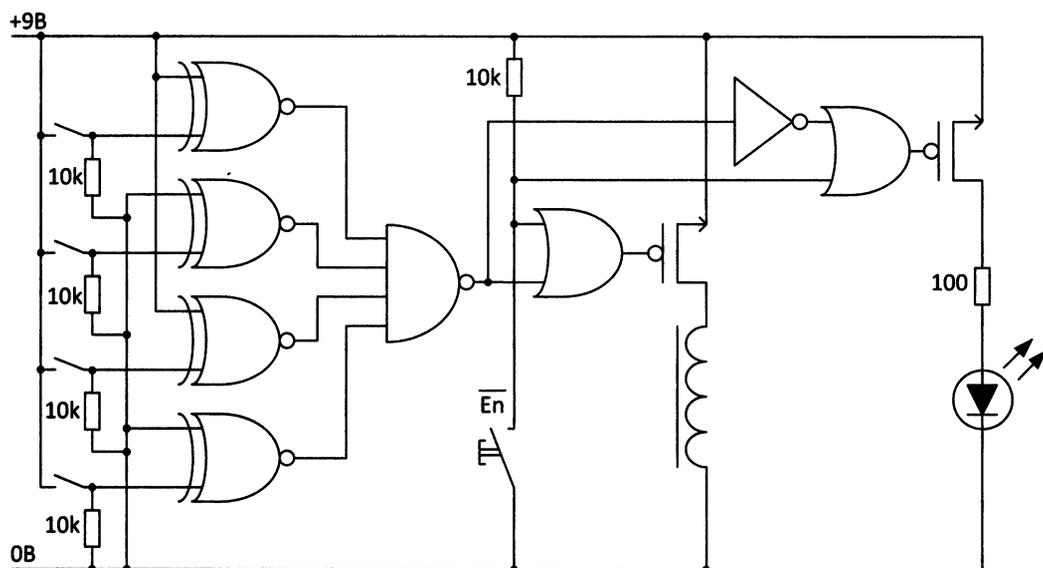


Рис. 4.19. Схема кодового замка с сигналом разрешения открывания и сигнализацией неправильного кода

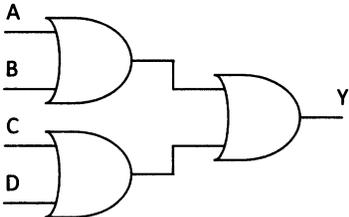
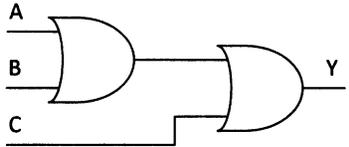
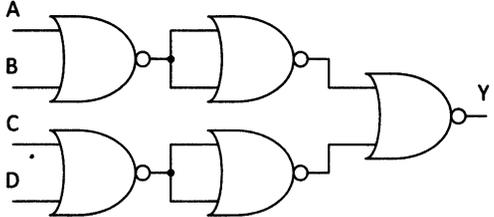
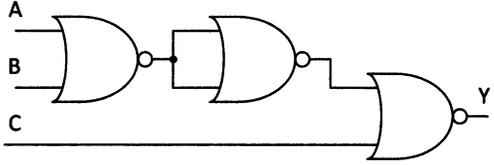
Игра «Угадай число» на базе схемы кодового замка

А если вам не требуется что-то скрывать, зато вы не прочь весело провести время с товарищами, то замените соленоид в последней схеме (см. рис. 4.19) на еще один светодиод другого цвета с резистором 100 Ом. Такая переделка превратит кодовый замок в игру «Угадай код». Один игрок с помощью DIP-переключателя устанавливает некоторую комбинацию и прикрывает ее, а второй пытается угадать за три попытки, набирая код на втором переключателе. В случае правильного ответа при нажатии кнопки загорается светодиод одного цвета, а при неудаче — другого.

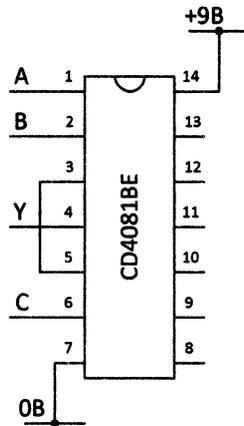
Как видите, такая довольно большая цифровая схема, из которой можно сделать сразу два интересных устройства, появилась в результате последовательного движения от простого к сложному. Сначала была сформулирована главная задача, а по мере ее решения добавлялись все новые функции. При желании вы сообразите, как

расширить код для большего количества битов или добавить новые свойства в этот электронный замок. А на основании изученных в этой главе правил применения вентилях и преобразования одних логических схем в другие вы сможете создавать и свои схемы, сразу выполняющие какие-либо действия в зависимости от изменения на входах. Такие устройства называются *комбинационными* — в них нет элементов для запоминания внутреннего состояния, и их поведение задается лишь комбинацией входных сигналов. В следующей главе будут рассмотрены основные схемы этого класса, применяемые в вычислительной технике.

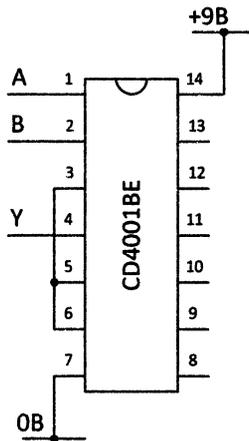
Ответы на вопросы для самопроверки

Номер вопроса	Ответ
1	
2	
3	
4	
5	Нет
6	Да

Ответы на задания по конструкциям для самостоятельной разработки

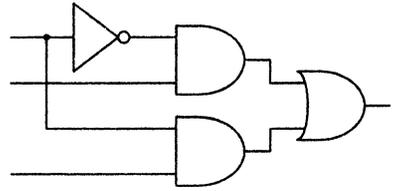


Ответ на задание 1 (один из возможных вариантов)



Ответ на задание 2 (один из возможных вариантов)

ГЛАВА 5



Основные комбинационные схемы

Вам потребуются:

- 4 транзистора типа n-МОП;
- 4 светодиода;
- 3 резистора сопротивлением 100 Ом;
- 4 резистора сопротивлением 10 кОм;
- 7 резисторов сопротивлением 470 Ом;
- одна кнопка;
- один счетверенный DIP-переключатель;
- 2 двойных DIP-переключателя;
- по одной микросхеме CD4011BE, CD4069UBE, CD4070BE, CD4071BE, CD4072BE, CD4081BE, CD4511BE;
- семисегментный индикатор SC56-11.

Как процессор выполняет вычисления: конструируем сумматор

Если задуматься над предназначением микропроцессора, то станет ясно, что его основное дело — быстро обрабатывать числа. Все остальное подчинено этой задаче. Поэтому надо разобраться, как с помощью логических вентилей выполняются арифметические операции. В целом, двоичные числа складываются так же, как и десятичные, только процессор не бормочет себе под нос: «пять пишем, три в уме». Во-первых, ему нечем бормотать, а во-вторых, в двоичной системе можно писать только ноль или единицу, а в уме нужно держать только единицу. При сложении двух таких чисел в каждом разряде возможны всего четыре комбинации: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$ и $1 + 1 = 10$. Но поскольку в последнем случае единица перешла в соседний разряд, то результат описывают так: $1 + 1 = 0$ с переносом 1. То самое: «ноль пишем, один в уме». Перенесенная единица, так же как и в случае с десятичными числами, складывается со знаками в следующем разряде. Если повнимательнее приглядеться ко всем четырем комбинациям, то можно увидеть, что

сумма однобитных чисел формируется не чем иным, как элементом Иключающее ИЛИ. А бит переноса вычисляется логической функцией И. Таким образом появляется схема (рис. 5.1) и таблица истинности (табл. 5.1) для устройства, складывающего два однобитных числа.

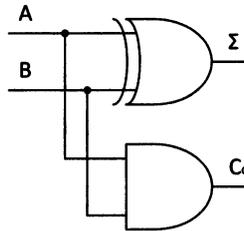


Рис. 5.1. Схема сложения двух однобитных чисел

Таблица 5.1. Сложение однобитных чисел

A	B	Σ	C_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Это устройство называется *полусумматором*. Выход вентиля Иключающее ИЛИ обозначается греческой заглавной буквой Σ (читается «сигма»), которая в математике используется для обозначения суммы. А бит переноса — C_0 (его обозначение происходит от английского carry output, выход переноса). Почему же приведенная схема — лишь полусумматор, а не полный? Дело в том, что у него нет входа для бита переноса из предыдущих разрядов. То есть он может использоваться лишь для сложения однобитных чисел, а такое устройство мало для чего годится. Полный сумматор должен иметь вход переноса C_i (от английского carry input). Его таблица истинности (табл. 5.2) выглядит следующим образом:

Таблица 5.2. Сложение двух однобитных чисел с переносом из предыдущего разряда

C_i	A	B	Σ	C_0
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Первые четыре строки с точки зрения выходных сигналов ничем не отличаются от таблицы истинности полусумматора. Следующие четыре строки заполняются понятным образом: ненулевой бит переноса суммируется с битами слагаемых. Особый интерес представляет последняя строка. В ней складываются три единицы, что дает единицу и в сумме, и в C_o . Все, как при сложении десятичных чисел.

Рассмотрим примеры сложения четырехбитных чисел по этим правилам:

$$\begin{array}{r} 0111 \\ +0001 \\ \hline 1000 \end{array} \quad \begin{array}{r} 0101 \\ +0011 \\ \hline 1000 \end{array} \quad \begin{array}{r} 0011 \\ +0011 \\ \hline 0110 \end{array}$$

Вопросы для самопроверки

1. Вычислите без применения электронных устройств суммы следующих четырехбитных двоичных чисел (бит переноса из четвертого разряда, если он есть, записывайте в пятый разряд): $0101+1010$, $1110+0001$, $1001+0011$, $1010+0011$, $1010+0111$.

* * *

Полный сумматор можно собрать из двух полусумматоров и логического ИЛИ так, как показано на рис. 5.2.



Рис. 5.2. Схема полного сумматора

Если вы знаете булеву алгебру, то легко можете преобразовать эту схему к такому виду, чтобы в ней использовались не три, а всего два типа вентилей: Исключающее ИЛИ и И-НЕ (рис. 5.3). А если не знаете, то можете проверить правильность предложенного преобразования, перебрав все восемь комбинаций входных сигналов.

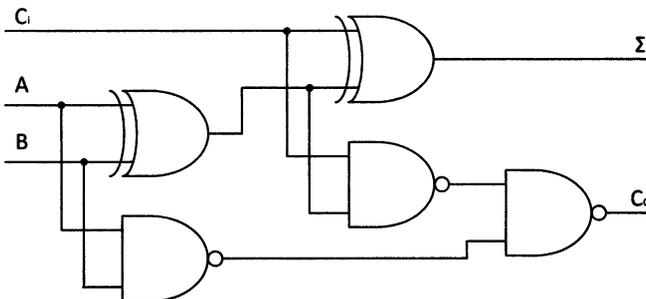


Рис. 5.3. Схема полного сумматора на вентилях Исключающее ИЛИ и И-НЕ

В любом случае сумматор, изображенный на рис. 5.3, можно собрать на макетной плате с помощью микросхем CD4070BE и CD4011BE, как показано на рис. Ц-5.4, и проверить его работу не только на бумаге, но и «в железе», как говорят между собой инженеры.

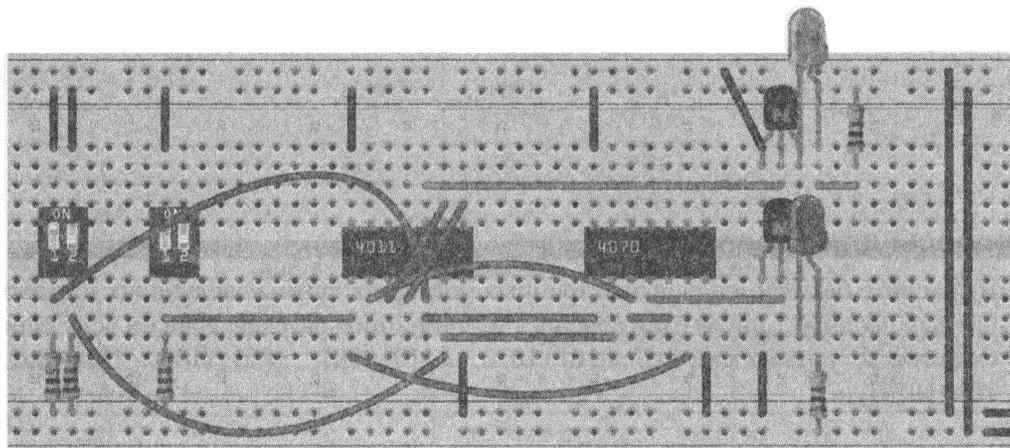


Рис. Ц-5.4. Монтажная схема полного сумматора на микросхемах CD4070BE и CD4011BE

Если теперь взять столько полных сумматоров, сколько битов в каждом из слагаемых, и соединить их, как показано далее для четырехбитной схемы, то получится так называемый *параллельный сумматор* (рис. 5.5). Называется он так потому, что в нем все биты слагаемых подаются на входы одновременно друг с другом, т. е. па-

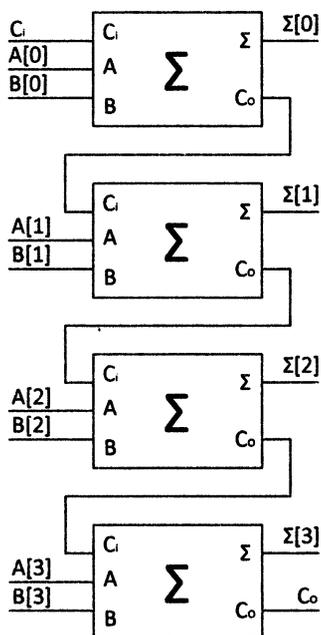


Рис. 5.5. Четырехбитный полный сумматор

параллельно. На рис. 5.5 вы впервые встречаетесь с обозначением битов в многоразрядных сигналах. Их обычно сопровождают квадратными скобками, в которых пишут номер старшего бита и младшего через двоеточие: $A[3:0]$ и $B[3:0]$. Такие многобитные сигналы называют *шинами*. То есть на вход изображенного на рис 5.5 сумматора подаются два слагаемых по четырехбитным (или четырехразрядным) шинам А и В. И есть еще выходная шина суммы $\Sigma[3:0]$

О БЫСТРОДЕЙСТВИИ ПАРАЛЛЕЛЬНЫХ СУММАТОРОВ

Это самый простой вид устройства для сложения, однако довольно неэффективный с точки зрения быстродействия. Вы видите, что бит переноса всего сумматора S_0 генерируется при последовательном прохождении сигналов от младшего значащего разряда к старшему. Поэтому задержка срабатывания сумматора будет определяться длиной всей этой цепочки. Так, при сложении 64-битных чисел сигнал от входа переноса до выхода будет проходить последовательно через сто двадцать восемь логических вентилях. Поэтому параллельная схема применяется очень ограниченно. Учеными-кибернетиками разработаны самые разнообразные алгоритмы ускоренной генерации переноса, а инженерами предложены многочисленные схемы их реализации. В микроэлектронике они получили широкое распространение. Но в этой книге мы не будем на них останавливаться, потому что они довольно сложны, и их не получится реализовать на макетной плате. Но все они так или иначе базируются на рассмотренных здесь сумматоре и полусумматоре.

По аналогии с сумматорами создают и двоичные вычитатели. Только в них вместо бита переноса в старший разряд есть бит заёма из него. А можно выполнить вычитание и с помощью сумматора и поразрядного дополнения до единицы. Есть самые разные варианты двоичных кодов для представления отрицательных чисел. Существуют такие алгоритмы, реализуемые схемотехнически, которые позволяют использовать сумматор как для сложения, так и для вычитания. Просто во втором случае вычитаемое сначала требуется преобразовать, а уже потом подавать на вход В сумматора. Это гораздо удобнее, чем делать сразу два громоздких устройства в микропроцессоре.

Любопытно рассмотреть, как будет выполняться умножение в двоичной системе. Здесь, как и в сложении, действуют те же правила, что и для десятичных чисел. Только все заметно проще, т. к. цифр всего две. Если записать привычным образом, в столбик, два перемножаемых двоичных числа, а потом умножить поразрядно первое на второе, то можно отметить некоторые особенности.

$$\begin{array}{r}
 111 \\
 \times 101 \\
 \hline
 111 \\
 000 \\
 111 \\
 \hline
 100011
 \end{array}$$

Как и должно быть, при умножении на единицу в каком-либо разряде второго множителя в соответствующее частичное произведение записывается сам первый множитель со сдвигом на нужное количество битов. А при умножении на ноль в эту строку записываются все нули. В приведенном примере умножения 7 на 5 первое

частичное произведение в двоичном виде равно 111, второе 000, а третье 111, записанное со сдвигом на два разряда влево. При сложении $111 + 0000 + 11100$ получается 100011.

Вопросы для самопроверки

2. Вычислите без применения электронных устройств произведения следующих четырехбитных двоичных чисел: 0101·1010, 1110·0001, 1001·0011, 1010·0011.

* * *

Исходя из обнаруженных особенностей, вы уже могли догадаться, что умножитель тоже нетрудно сделать из сумматора! Для этого один из множителей нужно побитно сдвигать влево, и если соответствующий бит другого множителя равен единице, то прибавлять сдвинутый первый множитель к частичной сумме. А если этот бит равен нулю, то пропускать прибавление. Конечно, существуют и другие схемы умножителей, но алгоритм сложения со сдвигом довольно эффективен. В этой теме нам пришлось забежать немного вперед, потому что еще не были рассмотрены вопросы устройства схем хранения промежуточных результатов для частичных сумм и схемы побитового сдвига. Но к ним мы еще вернемся, главное — понять сам принцип выполнения различных арифметических действий в цифровых системах. Можно отметить, что умножению на 2 соответствуют сдвиг числа на один разряд влево. А делению на 2 — вправо. Точно так же, как это происходит в десятичной системе при умножении и делении на 10. Это свойство также часто применяется для выполнения операций над числами.

Таким образом, на базе сумматора с разнообразными дополнительными блоками формируется *арифметико-логическое устройство* (АЛУ), являющееся вычислительным ядром микропроцессора. Оно бывает как для целочисленных вычислений, так и для операций над дробями, называемыми *вычислениями с плавающей точкой*. Обозначается АЛУ на схеме хорошо узнаваемым символом (рис. 5.6) и имеет целый набор входов, задающих выполняемую *операцию* (сложение, умножение и др.) и *операнды* (слагаемые, множители и др.), над которыми она выполняется. Кроме того, в АЛУ поступают и выходят служебные сигналы, иногда называемые *флагами*, которые дополняют или уточняют то или иное действие, выполняемое в нем. Операнды, как правило, многоразрядные. Их разрядность указывается на схеме в виде косой черты, пересекающей обозначение входа, и написанного рядом с ней количества разрядов.

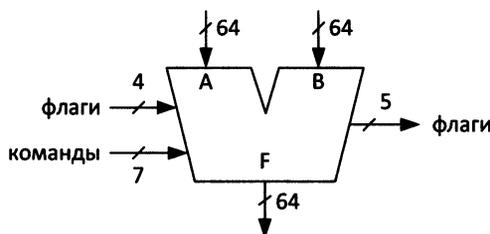


Рис. 5.6. Условное графическое обозначение арифметико-логического устройства (АЛУ)

Тайны двоичных кодов: шифраторы и дешифраторы

Следующими по важности за АЛУ схемами комбинационной логики можно считать дешифраторы и шифраторы. Названия их совсем не означают, что предстоит изучить что-то секретное. Речь идет о преобразовании чисел из одной системы счисления в другую. Предположим, что у вас есть клавиатура замка сейфа с десятью кнопками, на каждой из которых написано число от 0 до 9. А в цифровом устройстве, конечно, сигналы с этих кнопок должны быть представлены в двоичном коде. Между прочим, заблуждается тот, кто думает, что это будет преобразование из десятичного кода в двоичный. В таком замке нет одного провода, по которому передается сигнал от всех кнопок разом десятью способами, а есть десять проводов, по каждому из которых передается сигнал о том, нажата та или иная кнопка или не нажата. Но это, получается, и не двоичный код. Какой же? Код, в котором каждому числу, состоянию или устройству соответствует свой отдельный разряд или провод и в данный момент времени только на одном проводе из этого набора может быть активный логический уровень, называется *унитарным* (от латинского *uno*, один, единица). Кнопки на замке сейфа нажимаются по очереди. Значит, нам нужен преобразователь унитарного кода в двоичный. Устройство для перевода числа из более громоздкого кода в более компактный называется *шифратором*. А обратное преобразование выполняется *дешифраторами*. И никаких секретов!

Код Грея, «ТЕМПЕРАТУРНЫЙ КОД» И ДРУГИЕ...

Между прочим, уже знакомый вам двоичный код тоже не единственный в своем роде. Одних только способов для представления отрицательных двоичных чисел существует несколько. А еще есть *код Грея* — это такой особый способ записи, в котором каждое следующее число отличается от предыдущего ровно в одном разряде. В нем первые десять чисел выглядят так: 0000, 0001, 0011, 0010, 0110, 0111, 0101, 0100, 1100, 1101. Учеными были придуманы еще «температурный» код, код 8421 и другие. Каждый из них имеет свои преимущества в определенных областях техники, но в конечном итоге для обработки в компьютере их нужно привести к известному вам двоичному. Тут без шифраторов и дешифраторов тоже никак не обойтись.

Схема дешифратора для клавиатуры кодового замка

Итак, построим таблицу истинности (табл. 5.3) для клавиатуры сейфового замка. В ней будет всего 11 строк — ведь цифры кодовой последовательности вводятся поочередно, т. е. каждый раз положено нажимать только одну кнопку. Поэтому первая под заголовками столбцов строка, содержащая одни нули, описывает исходную ситуацию, когда клавиатуру не трогают. Слева в таблице в заголовках столбцов расположены унитарные коды клавиатуры, номер которых соответствует номеру кнопки, а справа — кодирующие их двоичные числа. Ноль в левой части таблицы означает, что кнопка не нажата, 1 — нажата. Клавиша 0 кодируется двоичным значением, соответствующим десятичному числу 10.

Таблица 5.3. Кодирование клавиатуры замка сейфа

A1	A2	A3	A4	A5	A6	A7	A8	A9	A0	Y3	Y2	Y1	Y0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	0	0	1	0	0	0	0	0	1	1	0
0	0	0	0	0	0	1	0	0	0	0	1	1	1
0	0	0	0	0	0	0	1	0	0	1	0	0	0
0	0	0	0	0	0	0	0	1	0	1	0	0	1
0	0	0	0	0	0	0	0	0	1	1	0	1	0

Для начала предположим, что нет никаких ограничений по количеству входов логических вентилях, и функции выходов могут быть любой длины. Тогда единица в младшем разряде будет формироваться следующим образом:

$$Y_0 = A_1 + A_3 + A_5 + A_7 + A_9.$$

Как вы помните, символ плюс в двоичной логике означает операцию ИЛИ. Поэтому составленное выражение означает, что разряд Y_0 будет равен единице, если нажата клавиша 1, или 3, или 5, или 7, или 9. На четные номера, как и должно быть, Y_0 не реагирует. Таким же образом записываются и выражения для остальных выходных сигналов:

$$Y_1 = A_0 + A_2 + A_3 + A_6 + A_7,$$

$$Y_2 = A_4 + A_5 + A_6 + A_7,$$

$$Y_3 = A_0 + A_8 + A_9.$$

То есть для реализации потребуются два 5ИЛИ, один 4ИЛИ и один 3ИЛИ. Но микросхем с вентилями 5ИЛИ не существует, поэтому придется преобразовать функции выходов под то, что есть в наличии.

Например, на вентилях 2ИЛИ схема шифратора замка будет выглядеть так, как показано на рис. 5.7. Обратите внимание, что функция Y_1 использует один из вентилях, задействованных для генерации функции Y_0 . Так сделано потому, что выражение $A_3 + A_7$ присутствует в обеих из них. В главе 3 при рассмотрении базовых вентилях было показано, что порядок записи входных сигналов в них совершенно не важен, т. е. сигналы в функциях для каждого из четырех выходов нашего дешифратора можно перегруппировать любым способом. Поэтому нет необходимости ставить два вентиля там, где можно обойтись одним. Как можно видеть, в результате для создания этого устройства нужно воспользоваться тремя микросхема-

ми CD4072. Если бы исключение избыточных членов в функциях не было произведено, то потребовалось бы использовать 4 микросхемы, т. к. общее число вентиляей стало бы равным тринадцати.

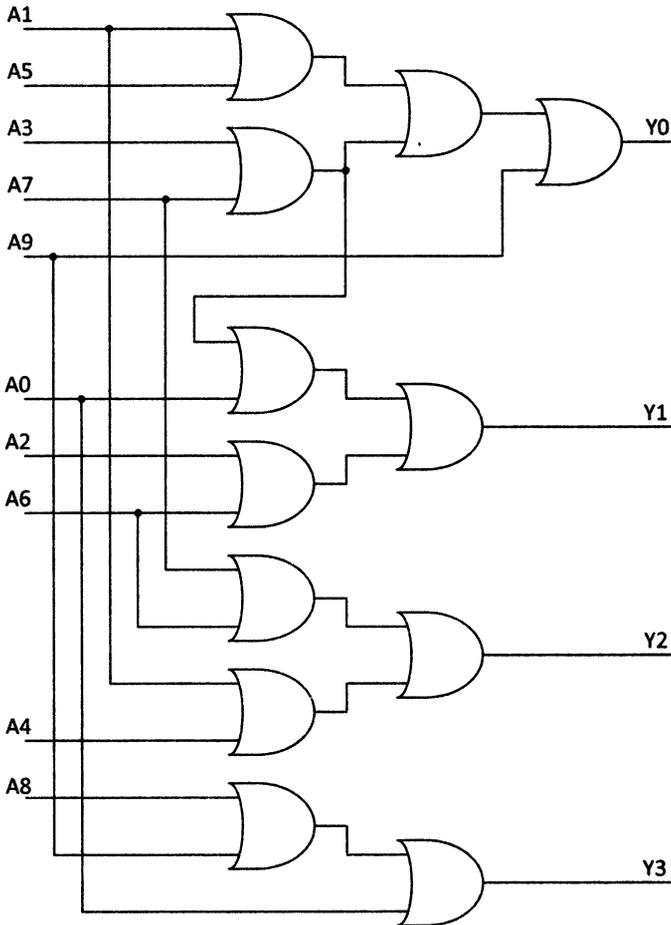


Рис. 5.7. Схема шифратора клавиатуры

Можно заметить еще одно одинаковое выражение в функциях $Y1$ и $Y2$: $A6 + A7$. Но так как оно имеет общий член $A7$ с ранее найденным выражением $A3 + A7$, то минимизация по обоим выражениям не приведет к сокращению количества вентиляей. Вы можете выбрать любой из двух вариантов и переделать схему или оставить ее, как она приведена здесь. А еще никто не запрещает взять за основу при создании схемы, например, ЗИЛИ. Тогда потребуется всего 7 вентиляей, правда, это все равно будет 3 корпуса на плате.

С помощью схемы шифратора (в нашем случае он называется *шифратором 10-в-4* — по количеству значащих входов и выходов) вы можете при желании дополнить механизм открывания цифрового замка, который был сконструирован в предыду-

щей главе. Тогда на входы компараторов, сделанных на базе исключающего ИЛИ, надо подавать сигнал не с механических переключателей, а с выходов шифратора. При этом код будет задаваться нажатием одной кнопки, что по-прежнему не самый лучший вариант с точки зрения безопасности. Но для того, чтобы иметь возможность вводить несколько цифр друг за другом, нужно изучить материал следующих глав, в которых рассматриваются схемы, запоминающие и хранящие значения цифровых сигналов.

Так как в шифраторе не предусмотрена защита от одновременного нажатия нескольких кнопок, то по выражениям для логических функций или по таблице истинности можно вычислить, что будет передано с клавиатуры замка в его электронную схему при нажатии, например, на цифры 2 и 3 вместе. Это значение 3. А если пальцем задеть сразу две кнопки 7 и 8, то будет считано двоичное число 15, которого на клавиатуре вообще нет. Для сейфа такое поведение системы, скорее, является правильным. Пользователь должен быть аккуратен и точен, а в любой неточности в деле защиты ценностей полезно подозревать злоумышленника. Тем не менее, в цифровой электронике существуют так называемые *шифраторы унитарного кода с приоритетом*. В них заранее определяется, что при поступлении сразу нескольких значений на входы либо меньшие числа блокируют большие, либо наоборот. Такие варианты уже существенно более громоздкие из-за усложнения логических функций их выходов. Для них выпускаются готовые микросхемы, содержащие приоритетные шифраторы в интегральном исполнении.

Схема дешифратора

В том случае, когда требуется обратное преобразование двоичного кода в унитарный, применяются *дешифраторы*. В микроэлектронике самое частое их использование наблюдается в устройствах памяти разного типа, имеющих матрицу запоминающих ячеек. Ячейки образуют строки и столбцы, как в таблице. Строки матрицы называют *словами*, которые имеют длину, измеряемую в битах. Например, матрица 1024×64 содержит 1024 слова по 64 бита каждое. Чтобы прочитать содержимое того или иного слова или изменить его, нужно подать на вход памяти номер нужной строки. Номера строк, как и положено в цифровой системе, кодируются двоичным числом, в данном случае 10-битным. А при обращении к определенному слову необходимо подать соответствующий сигнал выборки только на него. То есть преобразовать двоичный номер строки в унитарный. Поэтому без дешифратора 10-в-1024 никак не обойтись. Конечно, такое сложное устройство на макетной плате не собрать, но для изучения вполне достаточно рассмотреть существенно более простую схему дешифратора 2-в-4.

Снова работа начинается с составления таблицы истинности. В таком дешифраторе есть два входа, кодирующих номер строки: A_0 и A_1 , и четыре унитарных выхода: Y_0 , Y_1 , Y_2 , Y_3 . Так как входы по сути являются битами двоичного числа, то их для удобства лучше записать слева направо по убыванию номера, так же, как и выходы. Это сделано в табл. 5.4.

Таблица 5.4. Таблица истинности дешифратора 2-в-4

A1	A0	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Глядя на эту таблицу, легко составить логические функции для выходных сигналов дешифратора:

$$Y0 = \overline{A1} \cdot \overline{A0};$$

$$Y1 = \overline{A1} \cdot A0;$$

$$Y2 = A1 \cdot \overline{A0};$$

$$Y3 = A1 \cdot A0.$$

Как можно видеть, схема, реализующая дешифратор в соответствии с логическими выражениями для его выходов, будет содержать вентили 2И и инверторы (рис. 5.8).

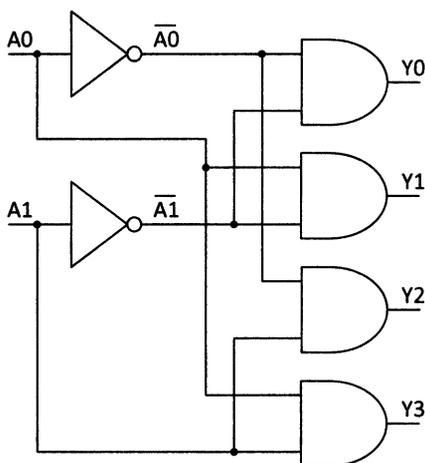


Рис. 5.8. Схема дешифратора 2-в-4

На макетной плате дешифратор 2-в-4 на микросхемах CD4081BE и CD4069UBE с добавлением переключателей на входах и светодиодов на выходах будет иметь вид, показанный на рис. Ц-5.9.

У такой схемы есть очевидный недостаток: т. к. на входах A1 и A2 всегда есть какое-то определенное состояние, то один из выходов всегда будет активным. А такая ситуация чаще всего не является правильной. В устройствах памяти — в самом частом месте применения дешифраторов — такое просто противопоказано, чтобы не перезаписать случайным образом информацию, хранящуюся в той строке мат-

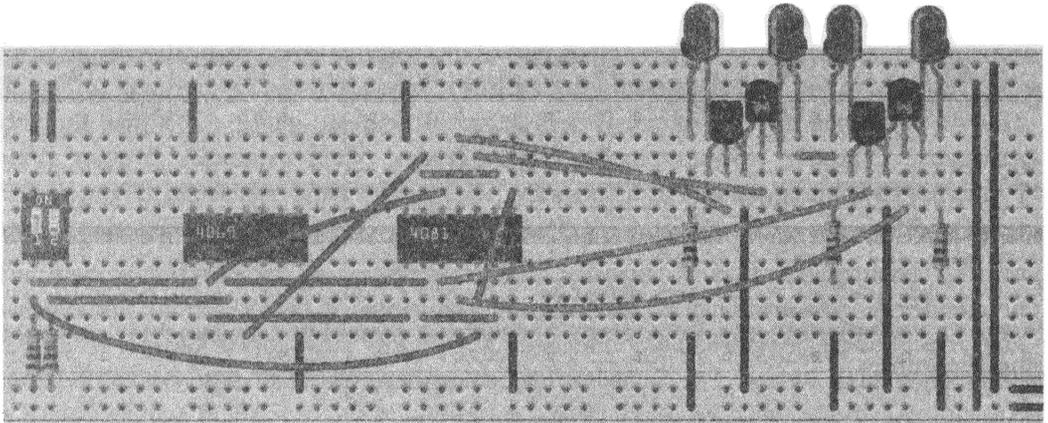


Рис. Ц-5.9. Монтажная схема дешифратора 2-в-4

рицы, чей номер выставлен дешифратором. Значит, нужно дополнить его схему входным сигналом, разрешающим или запрещающим активацию выходов. Такой сигнал называется *разрешением выхода* — OE (от англ. output enable). Он может быть активным как по уровню нуля, так и по уровню единицы — это определяется требованиями конкретной цифровой системы. Предположим, что выходы активны при OE = 1. Таблица истинности дешифратора (см. табл. 5.4) с добавлением этого сигнала немного изменяется: когда он равен нулю, то состояние остальных входов не имеет значения, что отображается латинской буквой X в соответствующих позициях (табл. 5.5).

Таблица 5.5. Таблица истинности дешифратора 2-в-4 с разрешением выходов

OE	A1	A0	Y3	Y2	Y1	Y0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Логические функции выходов тоже становятся чуть сложнее:

$$Y0 = OE \cdot \overline{A1} \cdot \overline{A0};$$

$$Y1 = OE \cdot \overline{A1} \cdot A0;$$

$$Y2 = OE \cdot A1 \cdot \overline{A0};$$

$$Y3 = OE \cdot A1 \cdot A0.$$

А вот принципиальную схему такого дешифратора вам предлагается составить самостоятельно.

Вопросы для самопроверки

3. Составьте схему дешифратора 2-в-4 с сигналом разрешения выхода, активным в высоком логическом уровне.

* * *

В серии CD4000 есть две микросхемы дешифраторов 2-в-4: CD4555BE и CD4556BE. У обеих активный уровень сигнала разрешения OE низкий. Каждая содержит по два дешифратора. Только у первой выбранный выход находится в высоком уровне, а остальные — в низком, как в созданных вами схемах на вентилях. А у второй — на активном выходе будет логический ноль, а на остальных — единицы. Таким образом, у разработчика есть возможность выбора, какую микросхему предпочесть в зависимости от решаемой задачи.

Учим цифровые схемы показывать десятичные числа

Существует особый вид дешифраторов, которые используются для отображения цифр на *семисегментных индикаторах*. Любой из вас постоянно встречался с подобными индикаторами в своей жизни — это, например, указатель этажа, на котором находится лифт. Цифры от 0 до 9 отображаются на нем с помощью включения или выключения подсветки от двух до семи полос, каждая из которых и называется *сегментом* (рис. 5.10).



Рис. 5.10. Цифры на семисегментном индикаторе

Такой способ представления чисел прост и удобен, что и определило широкое распространение семисегментных индикаторов всяких форм, цветов, размеров и использующих разнообразные физические принципы в своей работе: от светодиодных излучателей до жидких кристаллов. Поэтому очевидно, что для них были созданы микросхемы специализированных дешифраторов, чтобы не строить каждый раз одну и ту же схему из логических элементов. Полезно рассмотреть устройство подобной микросхемы. Сегменты индикатора обозначаются латинскими буквами от *a* до *g* (рис. 5.11). Как вы знаете, для передачи десяти цифр в двоичном коде требуется 4 двоичных разряда. Обозначим их от *A0* (младший) до *A3* (старший). Сопоставим двоичные коды цифр и их изображения на рис. 5.10 с зажженными или погашенными сегментами индикатора. Активные сегменты обозначим единицами, остальные — нулями и получим для такого индикатора таблицу истинности (табл. 5.6).

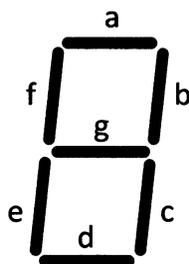


Рис. 5.11. Кодирование сегментов индикатора

Таблица 5.6. Преобразование двоичного кода в 7-сегментный

A3	A2	A1	A0	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

В этой таблице истинности единиц намного больше, чем нулей. Поэтому при составлении выражений для логических функций выходов можно воспользоваться инверсией значений. То есть удобнее создать формулы для тех случаев, когда соответствующий сегмент НЕ должен светиться. Это будут сигналы гашения, а не зажигания сегментов. Это гораздо проще, чем пытаться напрямую реализовать довольно громоздкие выражения для логических единиц по таблице. Обратите внимание на две особенности полученных формул. Во-первых, их слагаемые были сгруппированы, как это делается при упрощении выражений в школьной алгебре. Во-вторых, объединение по ИЛИ одного и того же бита в прямом и инверсном значении всегда будет равно единице. А объединение по И любого логического выражения с единицей дает само это выражение. Поэтому громоздкие изначально формулы для выходов e, f, g удалось сильно сократить. А в-третьих, важно помнить, что эти выражения описывают функции гашения сегментов, о чем свидетельствует знак инверсии над ними. Они выдают логическую единицу там, где по таблице истинности должен быть логический ноль! Чтобы их преобразовать в функции зажигания, надо проинвертировать полученные сигналы прежде, чем подавать их на входы индикатора. Иначе он будет показывать загадочные знаки, совсем не похо-

жие на цифры. Учитывая сказанное, напомним следующие выражения для логических функций выходов:

$$\bar{a} = \bar{A3} \cdot \bar{A1} \cdot (\bar{A2} \cdot A0 + A2 \cdot \bar{A0})$$

$$\bar{b} = \bar{A3} \cdot A2 \cdot (\bar{A1} \cdot A0 + A1 \cdot \bar{A0})$$

$$\bar{c} = \bar{A3} \cdot \bar{A2} \cdot A1 \cdot \bar{A0}$$

$$\bar{d} = \bar{A3} \cdot (\bar{A2} \cdot \bar{A1} \cdot A0 + A2 \cdot \bar{A1} \cdot \bar{A0} + A2 \cdot A1 \cdot A0)$$

$$\begin{aligned} \bar{e} &= \bar{A2} \cdot \bar{A1} \cdot A0 \cdot (\bar{A3} + A3) + \bar{A3} \cdot A1 \cdot A0 \cdot (\bar{A2} + A2) + \bar{A3} \cdot A2 \cdot \bar{A1} \cdot (\bar{A0} + A0) = \\ &= \bar{A2} \cdot \bar{A1} \cdot A0 + \bar{A3} \cdot A1 \cdot A0 + \bar{A3} \cdot A2 \cdot \bar{A1} \end{aligned}$$

$$\begin{aligned} \bar{f} &= \bar{A3} \cdot (\bar{A2} \cdot \bar{A1} \cdot A0 + \bar{A2} \cdot A1 \cdot \bar{A0} + A1 \cdot A0 \cdot (\bar{A2} + A2)) = \\ &= \bar{A3} \cdot (\bar{A2} \cdot (\bar{A1} \cdot A0 + A1 \cdot \bar{A0}) + A1 \cdot A0) \end{aligned}$$

$$\begin{aligned} \bar{g} &= \bar{A3} \cdot (A2 \cdot A1 \cdot A0 + \bar{A2} \cdot \bar{A1} \cdot (\bar{A0} + A0)) = \\ &= \bar{A3} \cdot (A2 \cdot A1 \cdot A0 + \bar{A2} \cdot \bar{A1}) \end{aligned}$$

В результате схема дешифратора семисегментного индикатора будет выглядеть не просто (рис. 5.12). Существуют и другие варианты преобразования его таблицы истинности в логические функции. Но ни один из них не позволяет уменьшить схему как-нибудь существенно. Обратите внимание на то, что инверсия выхода для получения сигналов зажигания сегментов из их сигналов гашения произведена не добавлением к выходам инвертора, а простой заменой И на И-НЕ. Кроме того, так же, как и в схеме шифратора замка, были найдены одинаковые выражения в функциях управляющих сигналов для сегментов b и f. Эти части были сделаны общими для обеих функций. Тот же самый подход использовался для сигналов d и e.

К полученной схеме необходимо добавить общий сигнал гашения для тех случаев, когда на индикаторе не отображается ничего. Обозначим этот сигнал VL и договоримся считать его активным по высокому уровню, т. е. при VL = 1 на всех семи выходах — логический ноль. Тогда у выходных вентилях в схеме нужно снова убрать инверсию, затем после каждого добавить 2ИЛИ-НЕ, на вторые входы которых и будет подведен сигнал VL. Добавленные ИЛИ-НЕ позволят выдавать логический ноль как в случае выставления индивидуального сигнала гашения для сегмента, так и в случае активного VL.

Конечно, на плате такое устройство, собранное на вентилях, будет занимать очень много места, поэтому их выпускают в интегральном исполнении. При этом в микросхему зачастую добавляются другие функции и возможности, и она уже называется не дешифратором, а *драйвером* семисегментного индикатора (от английского глагола to drive, управлять). Какие-то из них имеют всего один вход для отображаемого числа, которое задается не параллельным четырехбитным кодом, а последовательностью импульсов. Другие содержат входной счетчик, и отображаемая цифра задается нужным числом импульсов, подаваемых на этот вход по сигналу

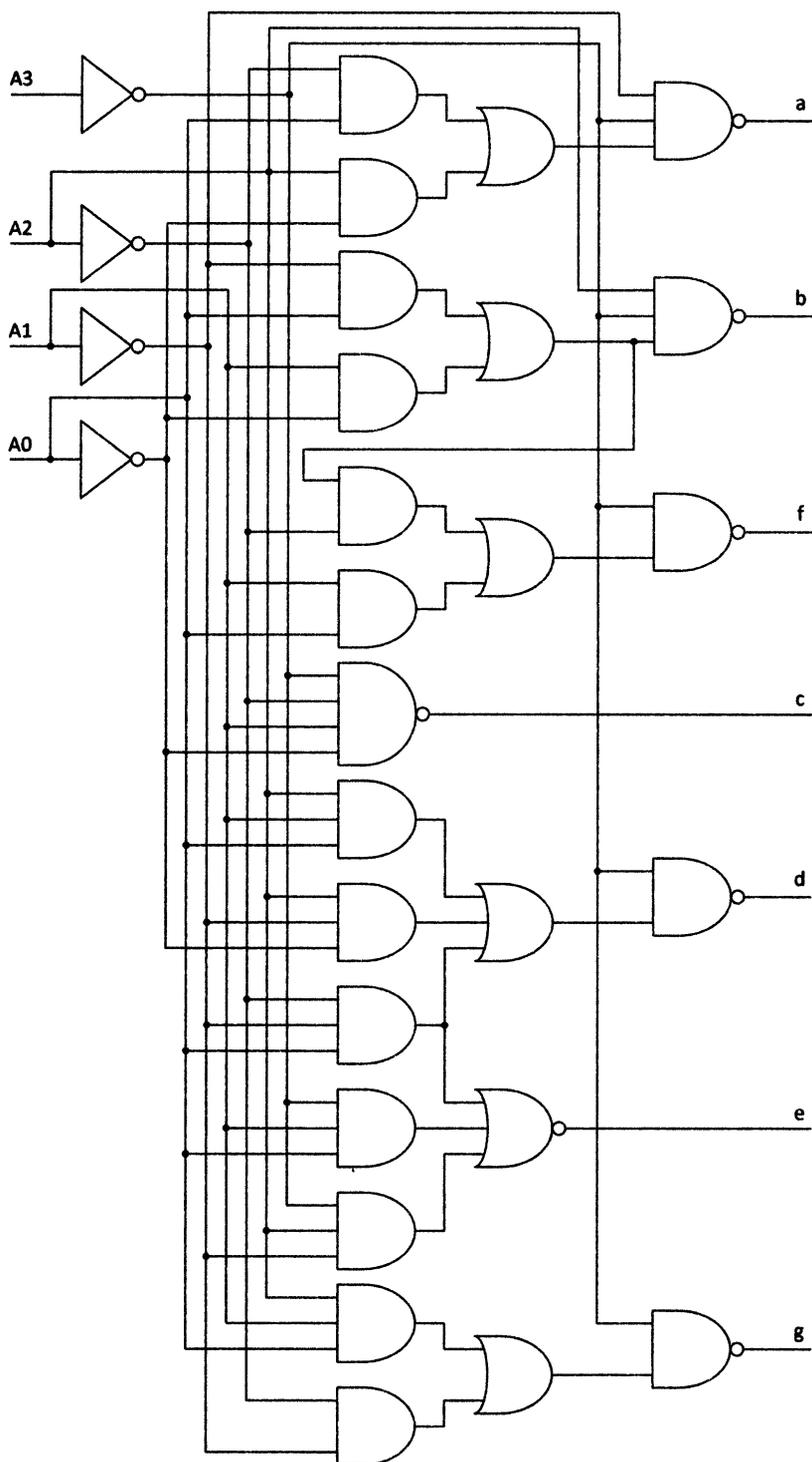


Рис. 5.12. Дешифратор семисегментного индикатора

разрешения. Но, в конечном итоге, все выходные сигналы генерируются с помощью схемы дешифратора, подобной той, что была только что рассмотрена.

В качестве практического примера использования семисегментного индикатора можно создать схему на базе драйвера CD4511BE, семи резисторов сопротивлением 470 Ом и светодиодного индикатора SC56-11. При сборке реальных схем с индикаторами на светодиодах нужно учитывать две важные особенности их конструкции. Первая — общая для любых светодиодов — необходимость ограничивать ток. Поэтому входы *a, b, c, d, e, f, g* обязательно должны подключаться через резисторы, сопротивление которых указано в документации на микросхему. Некоторые модели индикаторов имеют еще вход для десятичной точки — DP (от английского *decimal point*). Она служит для отделения целой части числа от дробной и, если нужна, активируется напрямую, без сложных схем дешифрации. Но ее тоже нужно подключать к драйверу через резистор, как и сегменты числа. Вторая особенность заключается в учете того, какие выводы светодиодов — аноды или катоды — подключены ко входам индикатора. Если аноды как у SC56-11, то катоды всех сегментов изначально «намертво» соединены с минусом питания, и для активации на входы должен быть подан высокий логический уровень. Такая микросхема называется *схемой с общим катодом*. В обратном случае, в *схеме с общим анодом*, ситуация ровно противоположная.

Помимо уже известных вам основных сигналов: четырех входных (A0, A1, A2, A3), задающих число для отображения, и семи выходных, зажигающих свои сегменты индикатора, — драйвер содержит еще 3 вспомогательных: LE, LT и BL. Первый активирует режим записи входных данных во внутреннюю память — так называемое *защелкивание* данных, и расшифровывается как разрешение защелкивания (*latch enable*). Если этот сигнал равен единице, то изменение данных на входе никак не отражается на выходе, а следовательно, и на показаниях индикатора. В рассматриваемой схеме (рис. 5.13) такой режим не задействуется, поэтому этот вход подключен к минусу питания, чтобы данные беспрепятственно шли от входов драйвера к выходам. Но вы в любой момент можете подать на LE единицу и убедиться, что при этом режим защелкивания активируется. Вход LT (*lamp test*) — тестовый, для зажигания всех сегментов разом. Он активируется нулем. А вход BL (*blanking*) имеет обратное действие — он гасит все сегменты, если установлен в ноль.

У индикатора SC56-11 всего десять выводов. Два из них: восьмой и третий — подключаются к минусу питания, еще семь управляют сегментами, а последний, номер пятый, — десятичной точкой. Так как это не вход КМОП, а анод светодиода, то его можно оставить неподключенным. Изменяя состояния входов A0, A1, A2, A3 драйвера можно наблюдать смену цифр на индикаторе. Для этого соответствующие входы микросхемы CD4511BE изначально подключены по схеме через резисторы сопротивлением 10 кОм к логическому нулю, а переключатель или кнопка при нажатии меняет состояние входа на логическую единицу. Если вы соберете эту схему на макетной плате (рис. Ц-5.14), то следует иметь в виду, что у цифр 6 и 9 этот драйвер не зажигает горизонтальные сегменты в «хвостике» цифры. Эстетически это смотрится не лучшим образом, но разработчики схемы драйвера почему-то решили сделать именно так. В случае входных сигналов, соответствующих числам

от 10 до 15, которые невозможно отобразить на одном индикаторе, драйвер не активирует ни один сегмент. Кстати, эта схема впоследствии пригодится в главе 7 при рассмотрении счетчиков и создании электронного игральнго кубика.

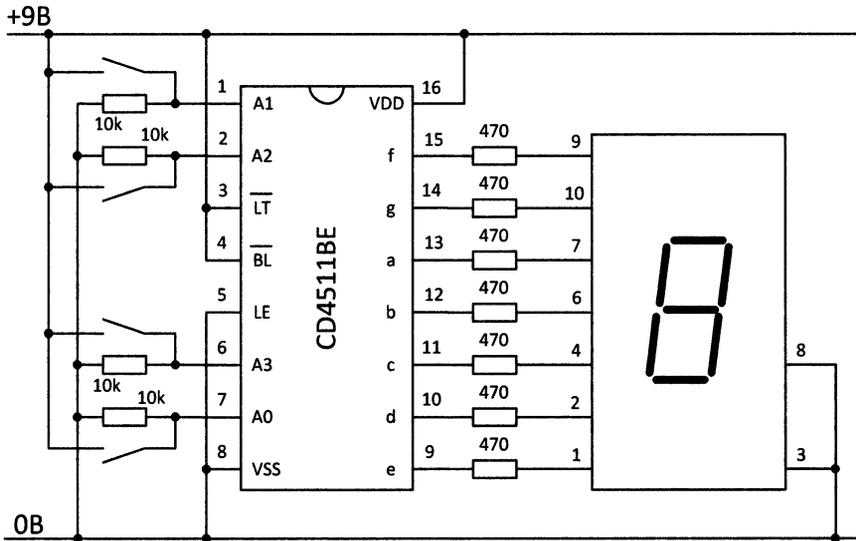


Рис. 5.13. Принципиальная схема управления 7-сегментным индикатором

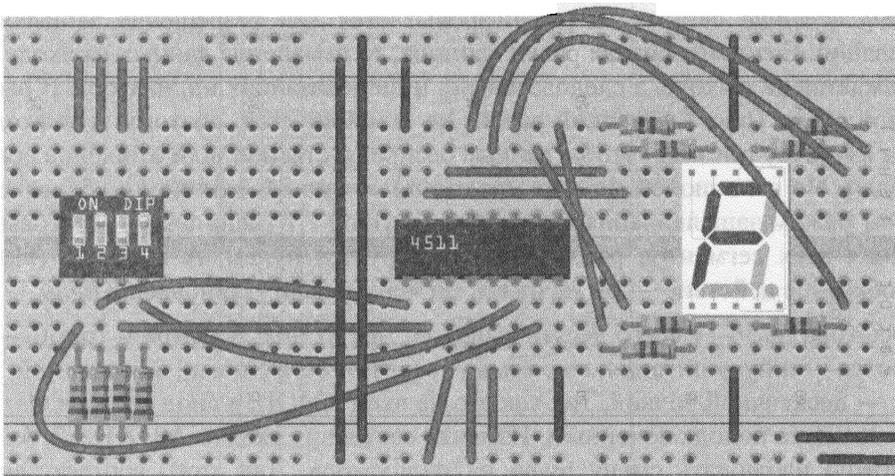


Рис. Ц-5.14. Монтажная схема управления 7-сегментным индикатором

Схемы шифраторов и дешифраторов не ограничиваются рассмотренными примерами, т. к. разнообразных двоичных кодов существует великое множество. Но с наиболее популярными представителями этого класса схем вы уже познакомились и научились применять. Если потребуется, то теперь сможете самостоятельно разобраться и с остальными.

Мультиплексоры и демультиплексоры — регулировщики цифрового движения

В списке наиболее часто применяющихся комбинационных устройств в микропроцессорах важное место занимают мультиплексоры и демультиплексоры. В чем состоит их задача? Предположим, что операнды в АЛУ поступают из нескольких источников. Но одновременно обработать сразу все данные невозможно — необходимо установить очередность. Как регулировщик на загруженной перекрестке дает разрешение на проезд по улице сначала с одного направления, потом с другого, так и цифровой системе подчас требуется такой электронный регулировщик. Эта роль возложена на *мультиплексор*. Его условное обозначение на рис. 5.15 приведено для случая, когда входов данных два: $DI0$ и $DI1$, а выход один — DO , и для выбора того входа, данные с которого будут передаваться на выход, используется *сигнал адреса* — A (его еще называют *сигналом выборки*).

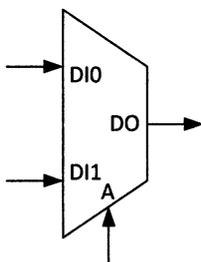


Рис. 5.15. Условное графическое обозначение мультиплексора

Практическая схема мультиплексора

Внутреннее устройство такого мультиплексора очень простое: при $A = 0$ на выход могут пройти только данные со входа $DI0$, т. к. нижний по схеме на рис. 5.16 вентиль И будет заблокирован. А при $A = 1$ активным становится вход $DI1$. Это также отражено в таблице истинности мультиплексора (табл. 5.7).

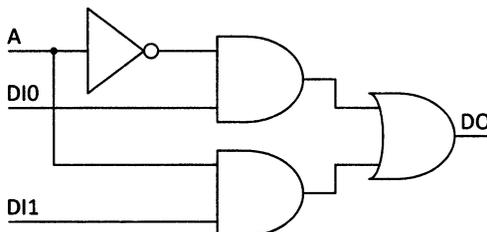


Рис. 5.16. Схема двухвходового мультиплексора

Практическая реализация мультиплексора на два входа показана на рис. Ц-5.17. У микросхемы CD4072BE сигналы с предыдущего каскада подаются на два входа каждый, чтобы реализовать на 4ИЛИ функцию 2ИЛИ. DIP-переключатель задает

Таблица 5.7. Таблица истинности двухходового мультиплексора

A	D11	D10	D0
0	X	0	0
0	X	1	1
1	0	X	0
1	1	X	1

данные на входах D10 и D11 мультиплексора, а кнопка — адрес A. При нажатой кнопке на состояние выхода будет влиять только один из двух переключателей, а при отпущенной — другой.

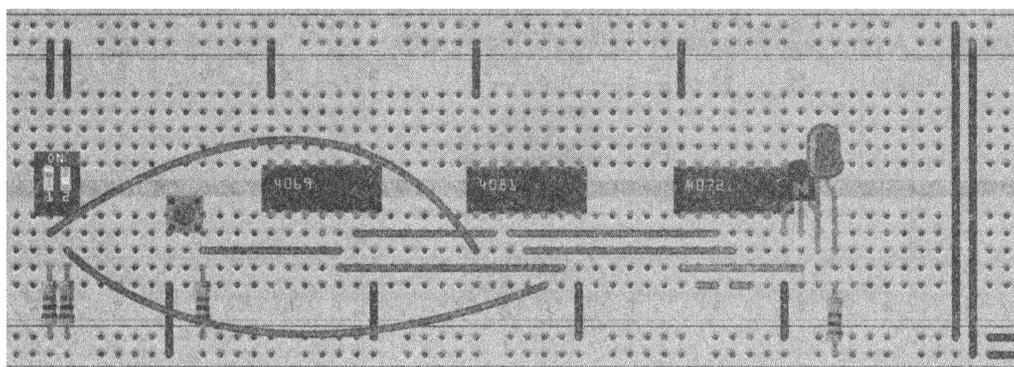


Рис. Ц-5.17. Монтажная схема двухходового мультиплексора

Если воспользоваться правилами преобразования логических функций булевой алгебры, то двухходовый мультиплексор можно построить всего лишь на одной микросхеме CD4011BE из входящих в ее состав четырех вентилей 2И-НЕ (рис. 5.18). Вы легко сможете сами собрать этот вариант на макетной плате и проверить, как он работает.

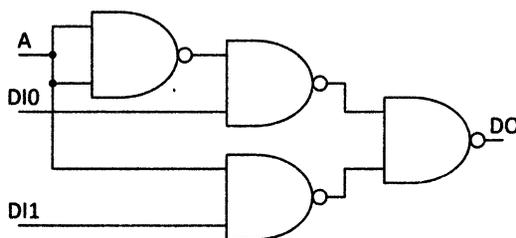


Рис. 5.18. Схема двухходового мультиплексора на вентилях 2И-НЕ

Несложно догадаться, как будет выглядеть схема мультиплексора с четырьмя входами данных (рис. 5.19). При этом адресных входов должно быть уже 2. По сути мультиплексор строится на базе из дешифратора, выходы которого объединяются по И с соответствующими данными, а затем собираются вместе на вентиле 4ИЛИ.

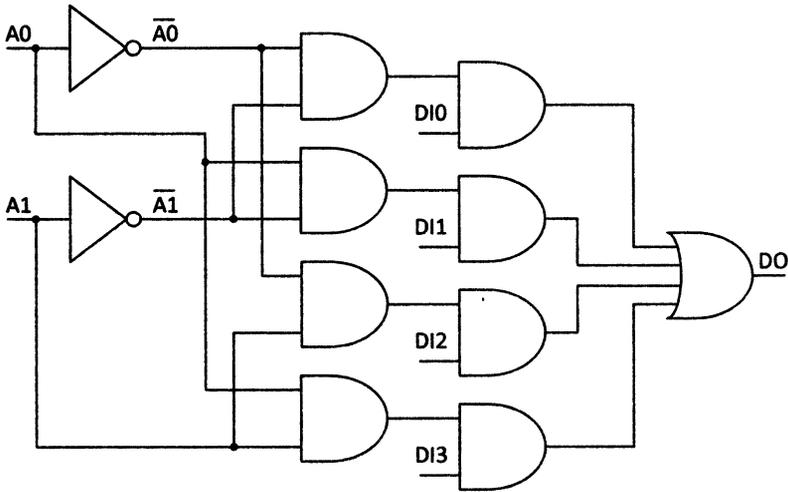


Рис. 5.19. Схема мультиплексора с четырьмя входами

Если учесть, что два двухвходовых И подряд образуют вентиль 3И, то схема становится проще (рис. 5.20).

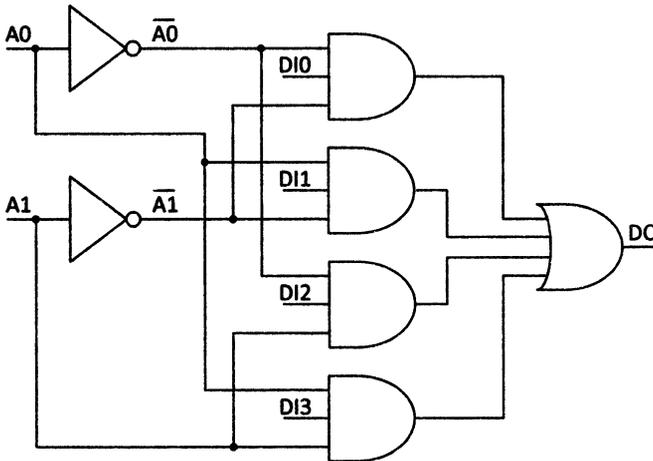


Рис. 5.20. Более простая версия мультиплексора с четырьмя входами

Демультимплексор

Как вы уже, наверно, догадались, *демультимплексор* выполняет обратную задачу. Подобно душевой лейке, разбивающей поток воды на множество струй, он может направить поток данных сразу в несколько направлений. А вот какое именно одно направление из всех доступных должно быть выбрано в тот или иной момент — определяется с помощью сигналов на адресных входах демультимплексора. Обозначается на схемах он сходным с мультиплексором образом. Вам предлагается самостоятельно разработать схему, соответствующую условному графическому обозначению, приведенному на рис. 5.21.

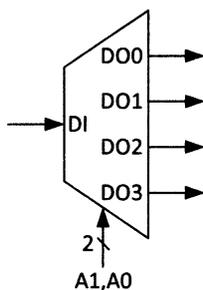


Рис. 5.21. Условное графическое обозначение демультиплексора с четырьмя выходами

Вопросы для самопроверки

- Составьте схему демультиплексора 2-в-4 и сравните ее со схемой дешифратора 2-в-4 с сигналом разрешения выхода OE, которую требовалось создать при ответе на *вопрос 3*.

Транзисторные мультиплексоры

А теперь полезно вернуться к схемам на транзисторах. Как вы помните, МОП-транзистор в цифровой электронике используется как ключ. Причем n-канальный передает без потерь низкий логический уровень, а p-канальный — высокий. Если объединить попарно стоки и истоки двух разных транзисторов и подать на затвор одного транзистора прямой сигнал разрешения, а на затвор другого — инвертированный (рис. 5.22), то получится так называемый *двунаправленный ключ* (от англ. transmission gate или TG). Он, действительно, может передавать сигнал в обе стороны: как в направлении от I к O, так и обратно. А самое главное — делает это без потерь уровня сигнала. Более того, через такой ключ можно передавать не только цифровые, но даже и аналоговые сигналы, максимальная величина которых не больше напряжения питания схемы ключа.

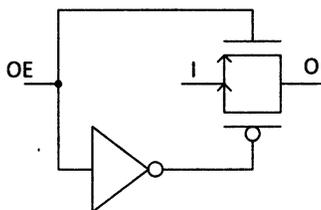


Рис. 5.22. Двунаправленный ключ

Несложно догадаться, как из него собрать мультиплексор на транзисторном уровне. А используя двунаправленность ключа можно даже создать универсальный мультиплексор-демультиплексор. Двухходовый вариант такого устройства показан на следующей схеме (рис. 5.23). При передаче сигналов данных слева направо получается мультиплексирование. А для демультиплексирования нужно подавать сигнал справа, а снимать с нужного выхода слева. Именно по такому принципу устроены микросхемы CD4051BE, CD4052BE и CD4053BE. Они отличаются друг от друга

разным коэффициентом мультиплексирования: 2, 4 или 8 универсальных входо-выходов. Можно также отметить, что схема мультиплексора на вентилях содержит заметно большее количество транзисторов, чем вариант на двунаправленных ключах. Обратите внимание, что выводы двунаправленных ключей объединяются друг с другом напрямую, без применения логического вентиля ИЛИ. Такая схема подключения называется *проводное ИЛИ* и возможна здесь потому, что когда ключ разомкнут, на его выходе нет никакого напряжения: ни нуля, ни единицы. То есть он разрывает провод, но не механически, как тумблер или кнопка, а электрически. Поэтому при размыкании на его выходах возникает так называемое *третье состояние*, которое в таблицах истинности обозначается символом не 0, и не 1, а Z. Тот ключ, который находится в третьем состоянии, никак не может помешать своему активному соседу передать его бит данных.

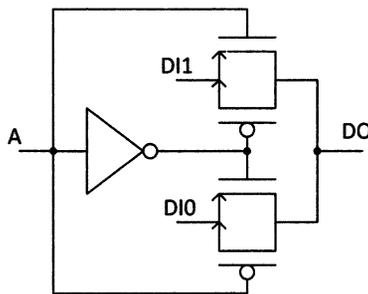


Рис. 5.23. Мультиплексор на двунаправленных ключах

К сожалению, заставить работать эти интересные схемы на дискретных транзисторах не получится. Как уже упоминалось в *главе 2*, у них подложка не имеет своего вывода, а «намертво» соединена с истоком, чтобы избавиться от четвертого вывода и одного из двух паразитных диодов. Но другой-то диод никуда не пропадает. Поэтому в двунаправленном ключе возможно протекание тока через него даже при выключенном транзисторе. Ток этот, конечно, слабый, но вполне достаточен, чтобы сбить работу чувствительных МОП-приборов. В интегральных микросхемах подложка р-канального транзистора, как правило, подключена к плюсу питания, а n-канального — к минусу. В результате паразитные диоды оказываются всегда в обратном включении и не влияют на правильность работы схемы, что позволяет выпускать упомянутые ранее универсальные микросхемы мультиплексоров-демультиплексоров.

Раз уж мы коснулись третьего состояния Z и проводного ИЛИ, то нельзя пройти мимо других устройств, способных их использовать. Самый простой вентиль с третьим состоянием — инвертор, дополненный двумя транзисторами, которые управляются сигналом разрешения выхода OE. Как можно видеть на его схеме (рис. 5.24), при $OE = 0$ два транзистора, стоящие посередине, отключены, и на выходе нет никакого сигнала. А при $OE = 1$ ноль или единица данных D беспрепятственно проходят через добавленные транзисторы. Такая схема работает даже на дискретных транзисторах, и вы встретите ее в следующей главе при рассмотрении триггеров. Вот на базе этой схемы уже можно построить мультиплексоры и демультиплексоры на дискретных транзисторах, но свойством двунаправленности

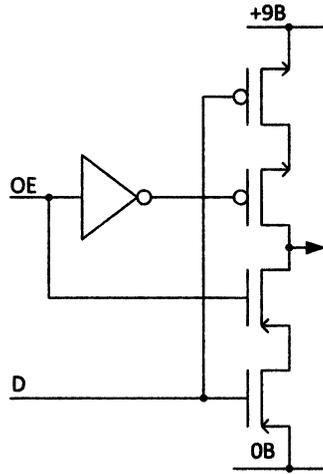


Рис. 5.24. Инвертор с третьим состоянием

они уже обладать не будут. К этим схемам нам еще предстоит вернуться в следующей главе.

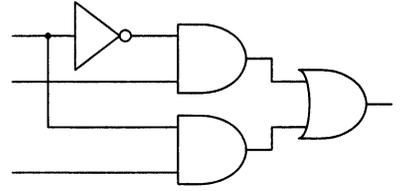
На этом изучение наиболее часто применяющихся комбинационных устройств можно считать законченным. Полезно еще раз повторить, что так называются те схемы, состояние выходов которых целиком зависит от комбинации данных на их входах и практически сразу изменяется при изменении входных сигналов. Теперь можно перейти к устройствам, которые обладают очень полезным свойством запоминания и хранения данных.

Ответы на вопросы для самопроверки

Номер вопроса	Ответ
1	1111, 1111, 1100, 1101, 10001
2	110010, 1110, 11011, 11110
3	

Номер вопроса	Ответ
4	<p>The diagram shows a logic circuit with three inputs: A0, A1, and D1. Inputs A0 and A1 are each connected to an inverter, producing outputs $\overline{A0}$ and $\overline{A1}$. There are four AND gates. The inputs to the AND gates are as follows: <ul style="list-style-type: none"> AND gate DO0: inputs A0 and $\overline{A0}$. AND gate DO1: inputs A0 and $\overline{A1}$. AND gate DO2: inputs A1 and $\overline{A1}$. AND gate DO3: inputs A1 and D1. </p>

ГЛАВА 6



Последовательностные схемы

Вам потребуются:

- 7 транзисторов типа n-МОП и 6 транзисторов p-МОП;
- 2 светодиода;
- 2 резистора сопротивлением 100 Ом;
- один резистор сопротивлением 1 кОм;
- один электролитический конденсатор емкостью 1000 мкФ;
- две микросхемы CD4011BE и по одной CD4001BE и CD4012BE.

Два пишем, три в уме: как запомнить результат вычислений?

В предыдущей главе вы познакомились с наиболее часто применяемыми комбинационными устройствами, предназначенными для преобразования и обработки данных. При этом упоминалось существование неких схем, которые в состоянии запоминать полученную информацию. Настало время разобраться в их работе детально.

Почему такие схемы называются *последовательностными*? Потому что состояние их выходов зависит от последовательности состояния входов в разные моменты времени. Что не похоже на поведение комбинационных схем, состояние выходов которых в какой-либо момент времени зависит только от комбинации входных сигналов в этот же момент. Это означает, что внутри последовательностных схем обязательно есть некий элемент, который умеет запоминать логический уровень сигнала и хранить его.

Самый простой запоминающий элемент можно собрать из двух инверторов, соединенных в кольцо (рис. 6.1). Действительно, если в одном узле такой схемы окажется логический ноль, то в другом будет логическая единица, которая через инвертор поддержит логический ноль в первом узле. И храниться эти значения будут столько времени, сколько подается питание. Изменять состояние узлов можно, подключая любой из них к тому выводу батареи, напряжение на котором противоположно на-

пряжению на узле в данный момент. То есть в схеме, получается, есть два входа и совпадающие с ними два выхода, логические уровни на которых всегда противоположны друг другу.

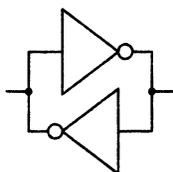


Рис. 6.1. Простейший триггер

Такое запоминающее устройство называется *триггером*. Но все разработчики электроники предпочитают не переключателем щелкать, а управлять при помощи электрических сигналов. Кроме того, не всегда удобно, когда выходы схемы одновременно являются и ее входами. Поэтому инженерами был изобретен триггер на двух вентилях 2И-НЕ, соединенных перекрестно (рис. 6.2). У него тоже два входа, обозначенных буквами R и S, и два выхода: Q и Qn, принимающие противоположные значения. Два сигнала, которые всегда имеют противоположные логические уровни, называют *парафазными*. Так что выходы Q и Qn — парафазные. Но теперь входы и выходы — совершенно отдельные узлы. Такая схема называется *RS-триггером* по названию его входов. Чуть позже станет понятно, почему были выбраны именно эти обозначения. У входов активные уровни низкие, поэтому на условном обозначении они отмечены кружками. Обязательно нужно обращать внимание на обозначения выводов внутри прямоугольника триггера — в разных их видах они могут быть разными. Рассмотрим подробно таблицу истинности триггера (табл. 6.1), поскольку она выглядит не так, как для комбинационных схем.

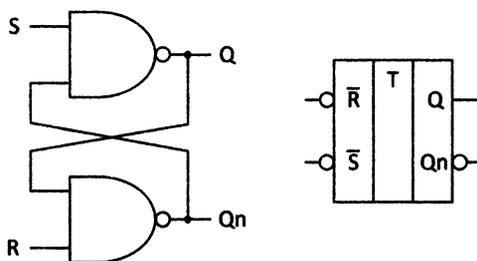


Рис. 6.2. RS-триггер: схема (слева) и условное графическое обозначение (справа)

Таблица 6.1. Таблица истинности RS-триггера на вентилях И-НЕ

S	R	Q(t)	Qn(t)	Режим работы
0	0	1	1	Запрещенное состояние, не используется
0	1	1	0	Установка Q = 1
1	0	0	1	Сброс Q = 0
1	1	Q(t-1)	Qn(t-1)	Хранение

При низком логическом уровне на входе S и высоком на входе R выход Q принимает значение единицы. Поэтому вход S так и обозначен — от английского слова *set*, установка. При низком уровне на R и высоком на S выход Q сбрасывается в ноль. Отсюда и обозначение входа R — от английского *reset*, сброс. Нули на оба входа одновременно подавать нельзя — такое состояние является запрещенным, потому что тогда оба выхода триггера примут одинаковое значение. При этом в триггере и вне его ничего не сломается и не сгорит, но нарушится логика работы схемы: выходы Q и Q_n , которые должны быть всегда в противоположных состояниях, одновременно переключатся в высокий логический уровень. А вот две единицы подавать не только можно, но и нужно — в тех случаях, когда триггер должен перейти в режим хранения. Рассмотренные первые три режима ничем не отличают триггер от комбинационных схем — изменение сигналов на входах тут же отражается на выходах. Но совсем иначе обстоит дело при поданных одновременно единицах на входы сброса и установки. Здесь вы сталкиваетесь с новым обозначением состояния схемы. Так как оно зависит от предшествующего, то к названиям выходных сигналов добавлен еще отсчет текущего времени t . Предыдущее изменение состояния произошло в момент, обозначаемый $t-1$. А следующее произойдет в $t+1$. В последней строке в таблице истинности вы видите, что выходы при $R = S = 1$ будут хранить то значение, которое они приняли во время предыдущего изменения состояния входов. Если в момент времени $t-1$ комбинация была, например, $S = 0$, $R = 1$, то выходы установились в $Q = 1$, $Q_n = 0$. И при переходе в момент t сигнала S в логическую единицу выходы продолжают оставаться в том же состоянии, что и раньше. А если в момент времени $t-1$ был подан сигнал сброса, то при переходе R в высокий логический уровень в момент t выходы будут иметь состояние $Q = 0$, $Q_n = 1$. Поведение триггеров удобно иллюстрировать с помощью особого вида графиков, который называется *временной диаграммой*. На этих графиках по горизонтальной оси откладывается время, а сигналы располагаются один над другим по вертикали. В результате можно наглядно представить, в каком состоянии в любой произвольный момент времени находятся все интересующие узлы схемы.

Рассмотрим временную диаграмму RS-триггера, показанную на рис. 6.3. В момент времени t_0 сигнал $R = 0$, а $S = 1$. Триггер находится в режиме сброса, поэтому выходы принимают состояние $Q = 0$, $Q_n = 1$. В момент t_1 ситуация на входах меняется на противоположную, что соответствует режиму установки, и поэтому $Q = 1$, $Q_n = 0$. После этого в момент t_2 на оба входа подается единица, и триггер переходит в режим хранения тех данных, которые были на его выходах в предыдущий момент времени t_1 , т. е. $Q = 1$, $Q_n = 0$. Затем снова поочередно следуют сигналы сброса, хранения и установки, что отражается на выходных сигналах так, как показано на временной диаграмме.

Важная особенность внутреннего устройства триггеров, которая в комбинационных схемах не встречалась, состоит в том, что сигнал с выхода поступает обратно на входной элемент. Такое подключение сигналов называется *обратной связью*. В триггере, состоящем из двух инверторов, из-за смешивания входов с выходами она не столь очевидна, как во второй рассмотренной схеме на базе вентилях И-НЕ. Без обратной связи невозможно создать элемент памяти в цифровой логике, храня-

ший нужное значение сколь угодно долго. Именно обратная связь, пока не отключено питание схемы, заставляет выходной сигнал как бы «бегать по кругу», поддерживая самого себя в стабильном состоянии.

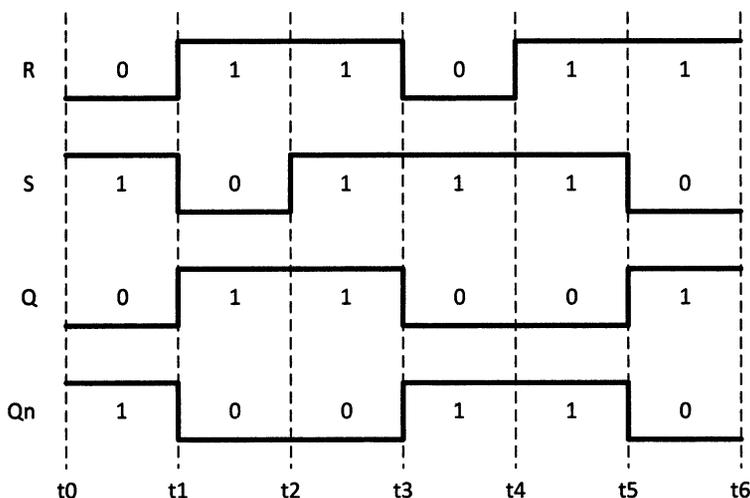


Рис. 6.3. Временная диаграмма RS-триггера на вентилях И-НЕ

Практическая схема RS-триггера на вентилях И-НЕ

На монтажной схеме (рис. Ц-6.4) показан RS-триггер, собранный на микросхеме CD4011BE, вместе с логическими пробниками, которые подключены к выходам схемы. Белыми проводниками задается состояние входов R и S, и светодиоды сразу же показывают, как изменяется сигнал на выходе. Все четыре режима работы, включая запрещенное состояние, легко проверяются на этом макете.

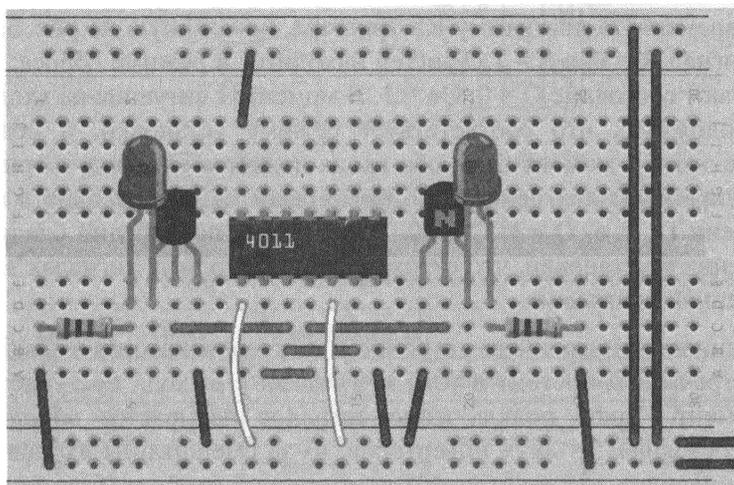


Рис. Ц-6.4. Монтажная схема RS-триггера на микросхеме CD4011BE

Можно создать RS-триггер и на базе вентилей 2ИЛИ-НЕ (рис. 6.5). В этом случае входы установки и сброса меняют активный уровень с низкого на высокий. Меняются местами и запрещенное состояние с режимом хранения. Таблица истинности такого триггера приведена в табл. 6.2.

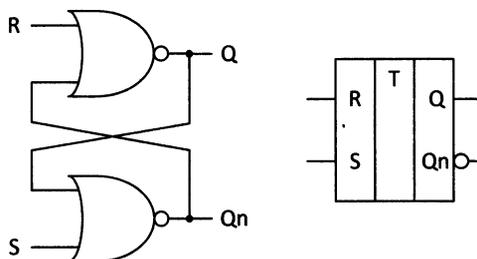


Рис. 6.5. Схема (слева) и условное графическое обозначение (справа) RS-триггера на вентилях ИЛИ-НЕ

Таблица 6.2. Таблица истинности RS-триггера на вентилях ИЛИ-НЕ

S	R	Q(t)	Qn(t)	Режим работы
0	0	Q(t-1)	Qn(t-1)	Хранение
0	1	0	1	Сброс Q = 0
1	0	1	0	Установка Q = 1
1	1	0	0	Запрещенное состояние, не используется

А вот изобразить временную диаграмму для заданной последовательности входных сигналов вам предлагается самостоятельно. Вы также можете сами собрать этот вариант RS-триггера из микросхемы CD4001BE на макетной плате и проверить его работу в деле.

Вопросы для самопроверки

1. Составьте временную диаграмму выходных сигналов для RS-триггера на базе вентилей ИЛИ-НЕ, подавая на его входы сигналы R и S, показанные на рис. 6.6.

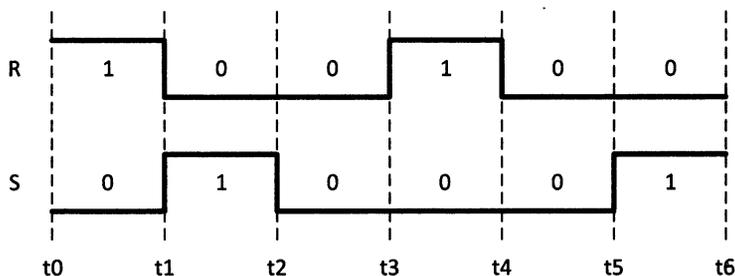


Рис. 6.6. Временная диаграмма для вопроса 1

Входить только по сигналу: как в триггерах разрешают и запрещают запись данных?

В реальных применениях рассмотренные нами RS-триггеры встречаются редко. Дело в том, что добиться идеальной одновременности переключения входных сигналов не получится: где-то будет немного отличаться длина проводов, где-то окажется разной скорость прохождения сигналов в предшествующих частях схемы. Представьте, что на последней временной диаграмме (см. рис. 6.3) сигнал R сдвинут немного вправо, что соответствует его большей задержке относительно сигнала S. Тогда в момент времени t_1 триггер перейдет ненадолго в запрещенный режим, и на его обоих входах появятся нули. И только потом он окажется в режиме установки. Из-за этого может сбиться работа следующих за ним частей цифровой системы. Чтобы избавиться от таких ситуаций, в триггер добавляется еще один вход, называемый *тактовым* или *входом синхронизации* (в переводе с греческого языка синхронный — значит одновременный). То есть добавляется новый сигнал, обеспечивающий одновременность переключения узлов схемы. Он обозначается C — от английского слова clock, часы. Часто тактовый сигнал для четкого отличия от остальных обозначается не одной буквой этого слова, а двумя — СК, а то и тремя — CLK.

Схема (рис. 6.7) и временная диаграмма (рис. 6.8) триггера немного усложняются, и теперь он называется *синхронным* или *тактируемым* RS-триггером — в отличие от всех ранее рассмотренных, которые именуются *асинхронными*. А в его таблице истинности (табл. 6.3) появляется дополнительный столбец для тактового сигнала C.

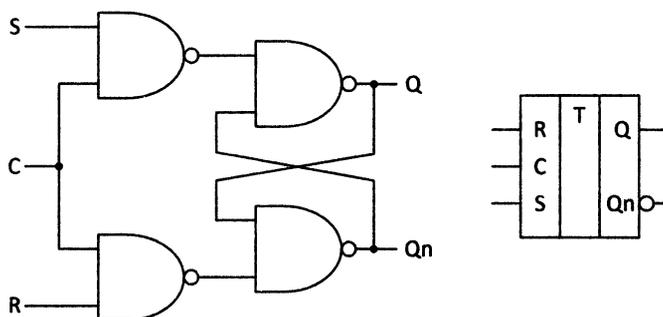


Рис. 6.7. Схема (слева) и условное графическое обозначение (справа) синхронного RS-триггера

Таблица 6.3. Таблица истинности синхронного RS-триггера

C	S	R	Q(t)	Qn(t)	Режим работы
0	X	X	Q(t-1)	Qn(t-1)	Хранение
1	0	0	Q(t-1)	Qn(t-1)	Хранение
1	0	1	0	1	Сброс Q = 0
1	1	0	1	0	Установка Q = 1
1	1	1	1	1	Запрещенное состояние, не используется

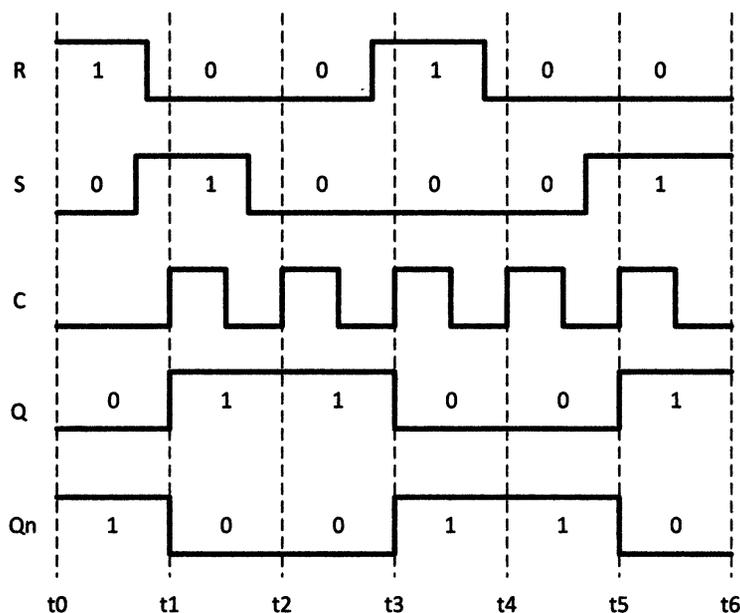


Рис. 6.8. Временная диаграмма синхронного RS-триггера

Видно, что в отсутствие высокого логического уровня на входе С никакие изменения на входах сброса и установки не оказывают влияния на выходы триггера. И только переход синхросигнала в высокий уровень как бы «открывает дверь» для прохода сигналов R и S внутрь схемы. Инженеры говорят, что этот триггер *срабатывает по высокому уровню тактового сигнала*. Бывают и схемы, которые срабатывают по низкому уровню. Важно здесь то, что у входных сигналов появляется некоторое время на то, чтобы «добежать» извне до входов триггера и подождать того момента, когда их будут готовы принять внутри. И, конечно, необходимо, чтобы в течение того времени, пока синхросигнал активен, состояние входов не изменялось. Тогда ситуации, в которой триггер вдруг окажется в запрещенном состоянии из-за того, что R и S переключились с задержкой друг относительно друга, больше не возникнет.

Практическая схема тактируемого триггера

Вы можете на практике проверить работу тактируемого RS-триггера. Конечно, формировать на плате тактовый сигнал можно, переставляя провод то на плюс питания, то на минус, или нажимая и отпуская кнопку. Но это неудобно. Поэтому предлагается сконструировать на макетной плате простейший генератор синхриимпульсов, который будет сам переключаться из нуля в единицу и обратно. Он состоит, как показано на рис. 6.9, из двух инверторов, резистора сопротивлением 1 кОм и электролитического конденсатора 1000 мкФ.

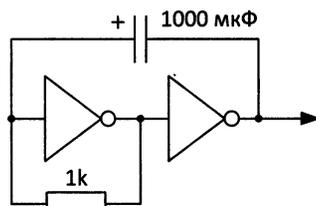


Рис. 6.9. Простейший генератор импульсов

КОНДЕНСАТОРЫ

Конденсатор — это компонент, обеспечивающий накопление электрического заряда, а его основной параметр — емкость — измеряется в *фарадах* и показывает, заряд какой величины он способен накопить в себе при заданном напряжении питания. Одна фарада (обозначается буквой Φ) — это огромная величина, поэтому конденсаторы имеют емкость, измеряемую в ее малых долях: пикофарады (10^{-12} Φ или пФ), нанофарады (10^{-9} Φ или нФ) и микрофарады (10^{-6} Φ или мкФ). Эти электронные детали бывают самых разных видов и типов, но такую большую емкость, как 1000 мкФ, способны обеспечить лишь *электролитические* конденсаторы, которые, в отличие от остальных своих собратьев, обладают полярностью. Поэтому на схемах обязательно обозначается, куда должен быть подключен их плюсовой вывод. А на корпусе конденсатора имеется маркировка: либо на его донце есть выпуклый знак плюс, либо на боковой поверхности в белой полосе печатается знак минуса у соответствующего вывода. В монтажных схемах здесь используется изображение конденсатора по второму варианту, поэтому по положению белой полосы на корпусе вы легко определите минусовой вывод.

У изображенного на рис. 6.9 генератора импульсов примерно две секунды на выходе будет логический ноль и столько же — логическая единица. Вы можете проверить это, подключив к его выходу логический пробник. В таких случаях говорят, что *период следования* импульсов равен четырем секундам. То есть период повторяющегося цифрового сигнала — это время между его последовательными переключениями в одно и то же состояние, например, в единицу. Эта схема единственная в книге, работа которой рассматриваться не будет, поскольку тут не обойтись без знания теории переходных процессов. Но она настолько проста, что — если все сделано правильно — начинает работать сразу после сборки. Выходные импульсы такого генератора нельзя считать прямоугольными — они имеют сильно криволинейную форму. Но для практических задач, решаемых в этой книге, схема будет пригодна и намного удобнее тумблера, кнопки или провода, переключаемых вручную. Следует иметь в виду, что при использовании уже подсевшей батарейки светодиод на выходе генератора начинает загораться и гаснуть не резко, а как бы нехотя, с замедлением. В этом случае батарею просто следует заменить на свежую, иначе синхронизация схемы нарушится из-за сильного искривления формы тактовых импульсов.

На принципиальной схеме, приведенной на рис. 6.10, показано, как на двух микросхемах CD4011BE можно собрать генератор и тактируемый триггер. Один корпус использован для входных вентилях триггера, а второй — для выходных и генератора синхросигнала. Это позволило уменьшить количество проводников, пересекающих корпуса микросхем на макетной плате, и повысить наглядность монтажной

схемы (рис. Ц-6.11). Разумеется, вы можете перекомпоновать устройство так, как вам будет удобно. Сигналы S и R подаются от шин питания на входы триггера, и вы можете менять их состояние, переставляя соответствующие провода. На плате также есть два логических пробника: верхний отображает состояние тактового сигнала, а нижний — выхода Q. Большая длительность периода сигнала C позволяет отчетливо увидеть, что изменения логических уровней на входах приводят к переключениям выходов только в тот момент, когда синхросигнал находится в высоком уровне. Можно также заметить, что нельзя изменить состояние входов, пока тактовый сигнал не перейдет в низкий уровень, — т. е. значения S и R должны быть постоянными при $C = 1$.

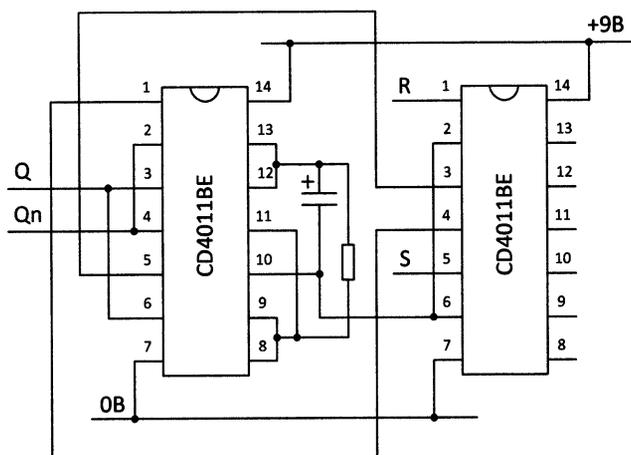


Рис. 6.10. Принципиальная схема синхронного RS-триггера с генератором тактовых импульсов

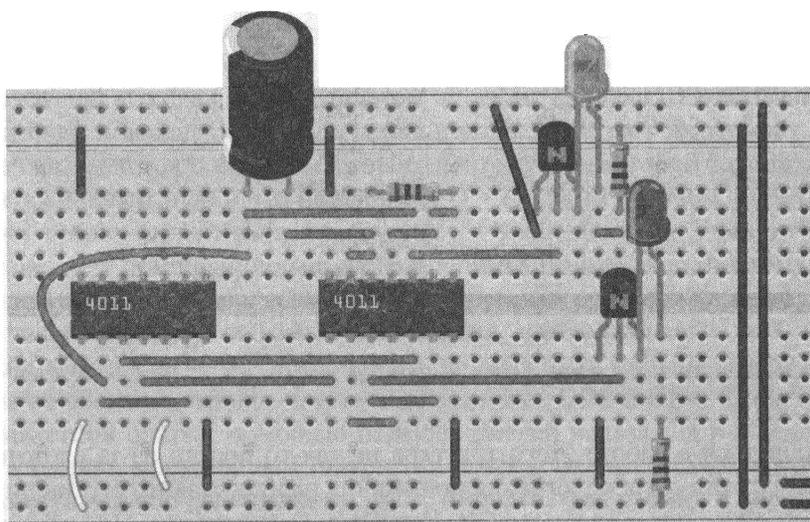


Рис. Ц-6.11. Монтажная схема синхронного RS-триггера с генератором тактовых импульсов

Триггеры без запрещенных состояний

Итак, вы на практике смогли убедиться, что собрать запоминающий элемент всего на нескольких вентилях — решаемая задача. Тем не менее триггер, у которого есть какие-то запрещенные состояния, несмотря на простоту, не может доставить большой радости инженерам-схемотехникам. Поэтому в дополнение к рассмотренным схемам был разработан JK-триггер. Его таблица истинности (табл. 6.4) подобна таблице RS-триггера на базе вентиля ИЛИ-НЕ (см. табл. 6.3) за исключением последней строки: установка обоих входов в высокий логический уровень вместо запрещенного состояния переводит схему в переключательный режим. В нем сигнал на выходах меняется на противоположный тому, что был ранее. Это очень полезное свойство, которое используется, например, при создании схем счетчиков.

Таблица 6.4. Таблица истинности JK-триггера

J	K	Q(t)	Qn(t)	Режим работы
0	0	Q(t-1)	Qn(t-1)	Хранение
0	1	0	1	Сброс Q = 0
1	0	1	0	Установка Q = 1
1	1	Qn(t-1)	Q(t-1)	Переключательный режим

В литературе иногда можно встретить схему асинхронного JK-триггера (рис. 6.12). На ней каждый вентиль имеет свое наименование, начинающееся с двойной латинской D, принятой в качестве обозначения для цифровых микросхем. Надписаны и внутренние узлы. Эти обозначения потребуются при анализе схемы.

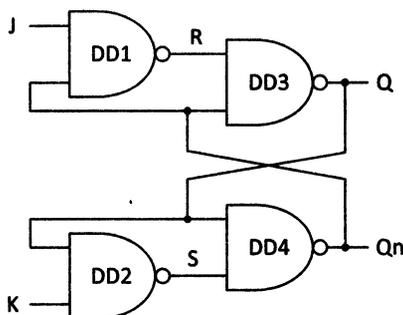


Рис. 6.12. Асинхронный JK-триггер

Сразу разобраться в работе этого триггера не так-то просто из-за сложных обратных связей. Поэтому для облегчения понимания надо составить его таблицу истинности (табл. 6.5), считая сигналы Q(t-1) и Qn(t-1) входными. По ней можно убедиться в формальной правильности схемы триггера.

Таблица 6.5. Таблица истинности JK-триггера с учетом предыдущего состояния

$Q(t-1)$	$Q_n(t-1)$	J	K	$Q(t)$	$Q_n(t)$
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	0
1	0	1	1	0	1

Однако если вы попытаете построить такой триггер, например, на микросхеме CD4011BE, то работать он не будет! Вместо переключательного режима при $J=K=1$ схема будет упорно оказываться в запрещенном, выдавая единицы на обоих выходах. В чем же дело? А в том, что в реальности вентили имеют пусть и небольшую, но задержку срабатывания. То есть изменение сигналов на входе отражается на состоянии выходов не мгновенно, а через некоторое время, измеряемое десятками наносекунд. А в современных микропроцессорах задержка переключения одного вентиля вообще составляет доли наносекунды! Мы такие интервалы заметить не можем, а транзисторы к ним чувствительны. И не учитывать это при рассмотрении работы схемы нельзя, поэтому для лучшего понимания цифровой электроники надо в этом разобраться.

Предположим, изначально триггер находился в режиме хранения ($J=K=0$), а состояние выходов было $Q=1$, $Q_n=0$. Это значит, что на обоих внутренних узлах схемы был высокий уровень: $S=R=1$. Как только на оба входа была подана единица, начался процесс переключения транзисторов в вентиле DD2. И после этого сигнал S стал равным нулю. Это, в свою очередь, привело к переключению вентиля DD4 и состоянию выхода $Q_n=1$. При этом на обоих входах DD3 оказывается логическая единица, и выход Q переключается в ноль. Но в этот же момент оба входа DD1 тоже оказываются в высоком уровне, поэтому с некоторой задержкой сигнал R становится равным нулю, что приводит к $Q=1$. Таким образом, переключательный режим практически мгновенно сменяется запрещенным. Увидеть столь быстрое изменение с помощью логического пробника невозможно, поэтому вы можете наблюдать лишь итог процесса переключения: две единицы на выходах. Чтобы избежать такого развития событий, необходимо, чтобы сигналы J и K имели вид очень узких импульсов — их длительность должна быть меньше задержки передачи сигнала от входа до выхода, т. е. суммы задержек DD2 и DD4. Сконструировать такое на макетной плате с помощью переключателей нельзя. Да и в реальных схемах, а не в теории, асинхронный JK-триггер никому не интересен, потому что идеально совместить два очень узких входных импульса для работы в переключательном режиме практически невозможно. Поэтому JK-триггеры на практике бывают только синхронными, а узкий входной импульс обеспечивается специальным спо-

собом внутри микросхемы. Кроме того, при ее создании тщательно выверяются размеры всех транзисторов, чтобы обеспечить наилучшее согласование и гарантировать отсутствие сбоев. Схема синхронного JK-триггера без такого генератора узкого синхросигнала показана на рис. 6.13, а его временная диаграмма — на рис. 6.14.

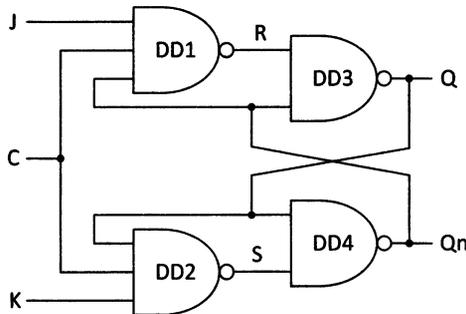


Рис. 6.13. Синхронный JK-триггер

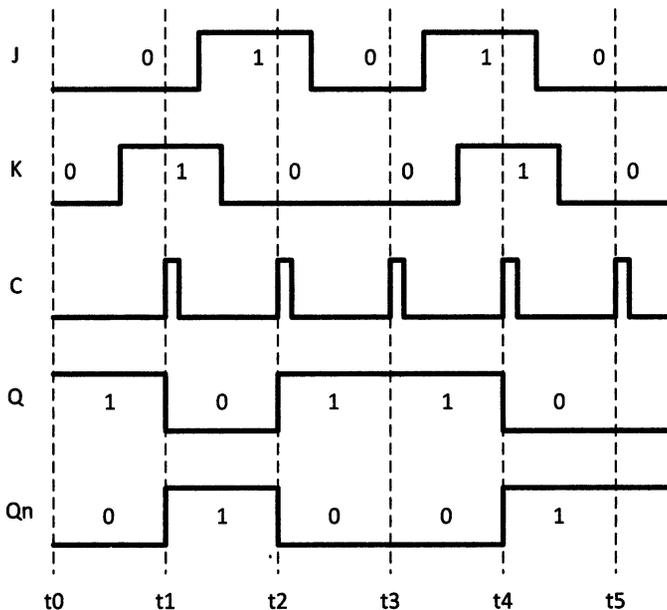


Рис. 6.14. Временная диаграмма синхронного JK-триггера

Зачастую, кроме синхронных входов подобные триггеры снабжают асинхронными входами сброса и установки, чтобы можно было записать в триггер строго определенное состояние. Это бывает нужно не только перед стартом цифровой системы, но и в процессе ее работы. Такие входы имеют наивысший приоритет, т. е. отменяют действие всех других входных сигналов, включая тактовый.

Вопросы для самопроверки

2. В чем отличие синхронного триггера от асинхронного с точки зрения логики работы?
3. В чем отличие JK-триггера от RS-триггера с точки зрения логики работы?

Приказано задержать! D-триггер

Если взять синхронный RS-триггер и подать сигнал на вход S напрямую, а на вход R — через инвертор, как показано на рис. 6.15, то получится элемент, который задерживает входной сигнал до очередного высокого уровня тактового импульса. Поэтому он называется D-триггером — от английского delay, задержка. Можно считать, что у него нет больше входов установки и сброса, а вместо них есть один вход данных, который обозначается символом D. Таблица истинности этого триггера очень проста, и вам предлагается составить ее самостоятельно. А его временная диаграмма приведена на рис. 6.16.

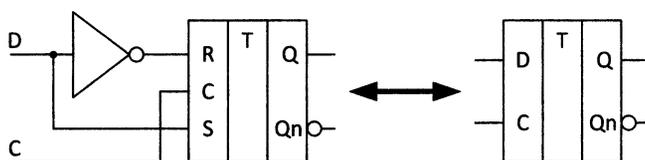


Рис. 6.15. Схема (слева) и условное графическое обозначение (справа) D-триггера

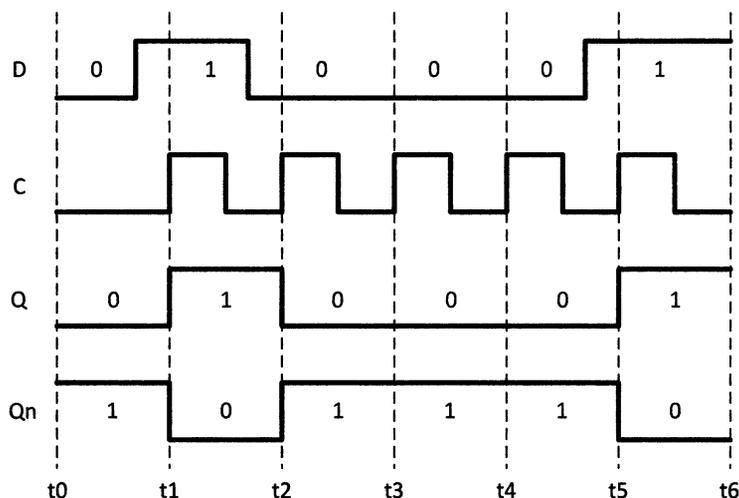


Рис. 6.16. Временная диаграмма D-триггера

Вопросы для самопроверки

4. Составьте таблицу истинности D-триггера в соответствии со схемой (см. рис. 6.15) и временной диаграммой (см. рис. 6.16).

* * *

В электронике есть общепринятые названия различных частей временной диаграммы цифровых сигналов. Вы уже знаете два из них: высокий логический уровень, он же единица, и низкий, он же ноль. Но есть еще два участка, речь о которых до сих пор не шла. Изменение сигнала при переключении из нуля в единицу называется *фронтом импульса*. А обратное, из единицы в ноль, — *срезом*.

Надо также отметить, что все работоспособные схемы синхронных триггеров, которые вы уже изучили, срабатывают по высокому уровню тактового сигнала, т. е., чтобы переключение выходов произошло правильно, необходимы переход С из нуля в единицу и стабильность входных сигналов все время, пока $C = 1$. Это отражено и в условных обозначениях триггеров — вход синхроимпульса не имеет кружка инверсии возле своего вывода или черты над названием. Если же внутрь схемы добавить инвертор на сигнал С, то триггер будет работать по низкому уровню, и его условное обозначение изменится (рис. 6.17).

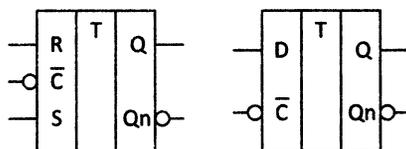


Рис. 6.17. Условные графические обозначения триггеров, срабатывающих по низкому уровню

Двухступенчатые триггеры: теория и практическая схема

Достоинством триггеров, срабатывающих по уровню, является их простота. Недостатком — необходимость держать входной сигнал неизменным в течение интервала времени, в котором синхросигнал находится в активном уровне. Вы, возможно, сами успели проверить на собранной ранее схеме синхронного RS-триггера, что изменение состояния входов во время высокого уровня тактового сигнала изменяет и состояние выходов. В таком случае говорят, что триггер *прозрачный* в течение высокого уровня сигнала С. Чтобы этого избежать, необходимо каким-то образом сузить то временное «окно», в которое сигнал может проникнуть внутрь триггера. Тогда не придется заботиться о стабильности напряжений на управляющих входах в течение всего активного уровня синхроимпульса. Один из способов решения этой задачи — добавить в схему преобразователь широких синхроимпульсов в узкие, использующий присущую всем вентилям задержку срабатывания для формирования сигнала. Для одной из таких схем, показанной на рис. 6.18, ширина выходного импульса будет равна задержке инвертора, умноженной на три.

Однако задержка любого вентиля сильно зависит от внешних условий, таких как температура окружающей среды и напряжение питания, и отклонений в технологическом процессе во время изготовления микросхемы. Поэтому ширина импульса в реальности будет нестабильной, что может привести к сбою в работе всей схемы.

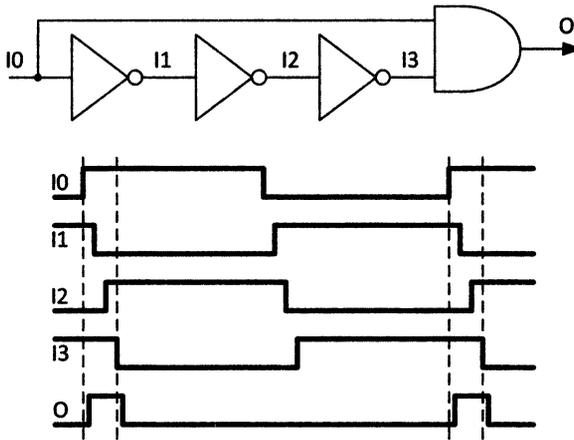


Рис. 6.18. Схема получения узких импульсов

Гораздо лучше заставить триггер срабатывать по фронту или срезу, а не по уровню. Для этого применяют двухступенчатые схемы триггеров, называемых *ведущими-ведомыми* или MS-триггерами (от английского Master-Slave). Для примера можно взять два синхронных RS-триггера и поставить их друг за другом, а входы синхронизации объединить через инвертор, как показано на рис. 6.19.

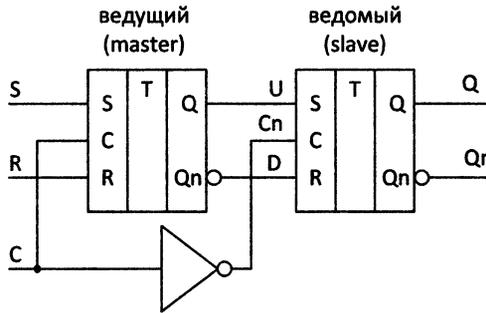


Рис. 6.19. Двухступенчатый триггер

Как работает двухступенчатый триггер? Посмотрите на его временную диаграмму (рис. 6.20), на которой показаны все сигналы, кроме инвертированного тактового C_n . При высоком уровне синхросигнала первый триггер (ведущий) может принимать команды с R и S входов и передавать их на свои выходы, обозначенные на схеме U и D. В это время второй триггер (ведомый) находится в режиме хранения записанных в него ранее данных, и ему как бы все равно, что происходит на его входах, т. е. в точках U и D. В момент среза синхросигнала, т. е. когда происходит его переключение из единицы в ноль, ведущий перестает реагировать на изменения уровней на входах до следующего фронта. А ведомый, наоборот, начинает воспринимать информацию с выходов ведущего и переключается в новое состояние в соответствии с ней.

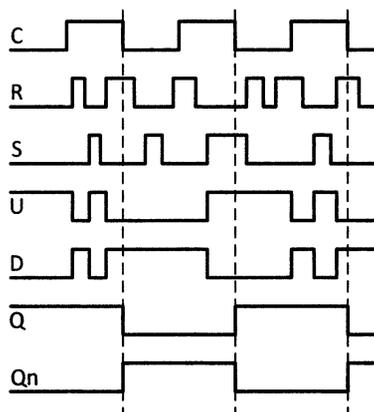


Рис. 6.20. Временная диаграмма двухступенчатого триггера

Получается, что если данные на входах MS-триггера менялись в течение высокого уровня, то это никак не отражалось на выходах Q и Qn всего триггера, — значение имело лишь то состояние входов S и R, которое было последним перед срезом синхросигнала. После среза C, во время его низкого уровня, входные данные тоже могут изменяться произвольным образом, т. к. ведущий триггер находится в режиме хранения, и на входах ведомого никаких переключений происходить не может, хотя он в это время прозрачный. Таким образом, временное окно для проникновения входных данных на выход оказывается очень узким, хотя никаких специальных коротких сигналов для этого создавать не пришлось.

Обратите внимание на кружок и треугольник возле наименования тактового входа на условном обозначении двухступенчатого триггера (рис. 6.21): треугольник указывает на работу по фронту или срезу, а кружок показывает, что этот RS-триггер переключается именно по срезу. Заставить триггер срабатывать по фронту можно или за счет добавления инвертора на входе сигнала синхронизации, или поменяв местами подключения уже имеющегося инвертора к тактовым входам ведущего и ведомого триггеров. Еще одно отличие условного обозначения MS-триггеров — наличие на нем не одной, а двух букв T, показывающих, что в состав этого триггера входят два: ведущий и ведомый. В английской литературе одноступенчатые триггеры, прозрачные по одному уровню и закрывающиеся по другому, называются latch — т. е. *триггер-защелка*. А MS-триггер именуется flip-flop.

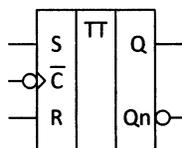


Рис. 6.21. Условное графическое обозначение двухступенчатого триггера

На базе MS-триггера типа RS можно создать надежно работающий синхронный JK-триггер. Его схема (рис. 6.22) содержит много вентилях, но теперь вы в ней не запутаетесь, а сразу найдете и ведущий триггер, и ведомый, а также обратные связи

от ведомого к ведущему, которые и превращают RS-триггер с синхронизацией по срезу в JK-триггер.

Поскольку схема весьма сложная, на ней подписаны номера выводов микросхем возле входа или выхода вентиля. А также каждый логический элемент несет свое обозначение, в котором цифра перед точкой нумерует микросхему, а цифра после точки — вентиль в этой микросхеме. То есть DD1.1 и DD1.2 расположены в одном корпусе, а DD1.1 и DD2.1 — в разных.

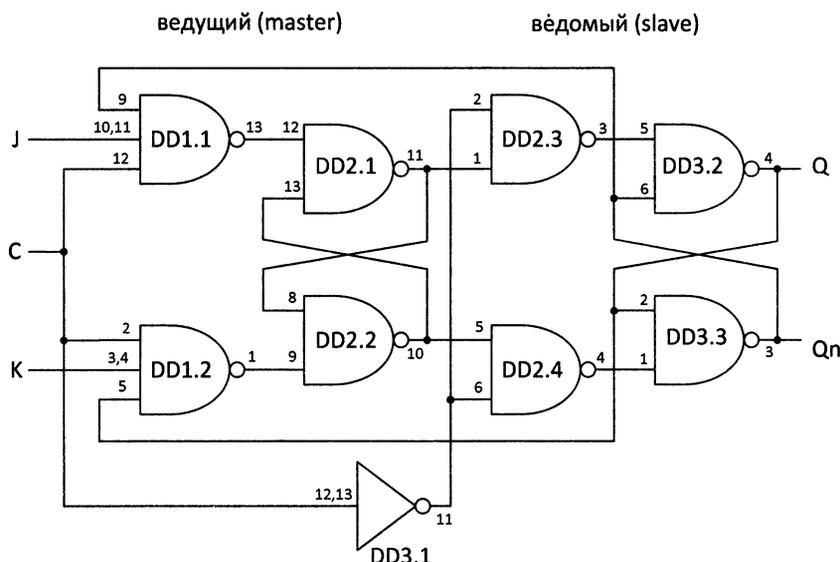


Рис. 6.22. Схема двухступенчатого JK-триггера

В распоряжении автора имелись только микросхемы 2И-НЕ CD4011BE и 4И-НЕ CD4012BE, поэтому входные трехвходовые элементы вместо CD4023BE реализованы на четырехвходовых CD4012BE с объединением двух входов в один. Этим объясняется, почему на схеме сигналам J и K соответствуют сразу два входа микросхемы. Генератор синхросигнала на макетной плате точно такой же, как и показанный ранее, но только собран он на отдельной микросхеме CD4001BE, т. к. две CD4011BE уже задействованы для создания триггера. Это позволяет обойтись меньшим набором компонентов, и не имеет принципиального значения для работы схемы. Как вы помните, инвертор можно одинаково эффективно получить как из вентилей И-НЕ, так и из ИЛИ-НЕ. Монтажная схема устройства показана на рис. Ц-6.23.

На базе схем «ведущий-ведомый» инженерами были также разработаны и D-триггеры. Как вы понимаете, они будут непрозрачными в любое время, кроме того фронта, по которому переключаются. Они строятся так же, как и одноступенчатые, — за счет добавления инвертора между входами S и R или между J и K, — тут сгодится любой из двух типов, поскольку инвертор между входами гарантирует, что их состояние всегда будет противоположным.

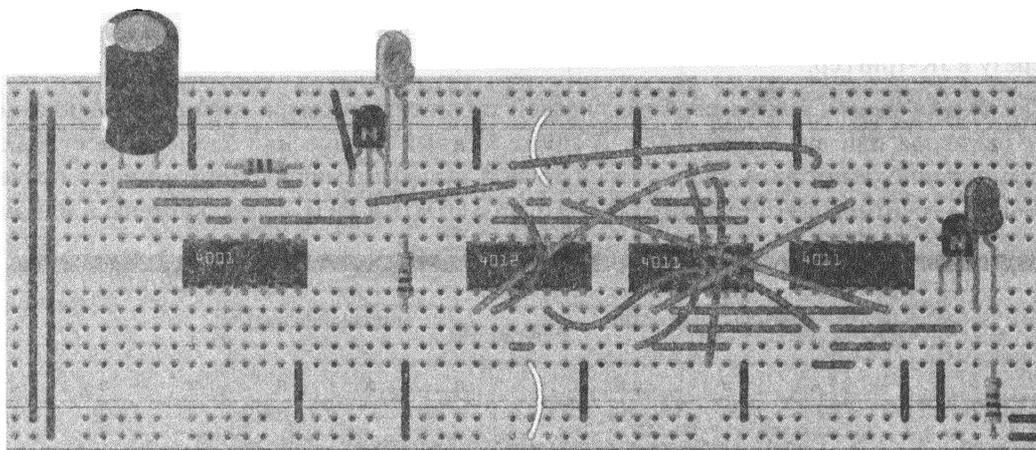


Рис. Ц-6.23. Монтажная схема двухступенчатого JK-триггера

Вопросы для самопроверки

5. В чем отличие двухступенчатого синхронного триггера от одноступенчатого синхронного?

T-триггер, или как посчитать импульсы?

Если сигнал с выхода Q_n двухступенчатого D-триггера подать на вход, то получится схема, в которой каждый такт значение выхода меняется на обратное (рис. 6.24). Выходит T-триггер — очень простая схема подсчета импульсов, пришедших на тактовый вход. Правда, один триггер «умеет» считать только от нуля до единицы, но в следующей главе вы увидите, как на базе этой простой схемы строятся много-разрядные счетчики. На временной диаграмме (рис. 6.25) отчетливо видно, что за то время, пока тактовый импульс совершает два переключения, выход T-триггера переключается один раз. В этом случае говорят, что триггер делит частоту следования импульсов в два раза. В самом деле, по определению частота повторяющегося импульсного сигнала равна количеству импульсов за единицу времени. Измеряется она в герцах (Гц). Один герц соответствует одному импульсу в секунду. Тысяча импульсов в секунду — это уже один килогерц (кГц), а миллион — мегагерц (МГц). На временной диаграмме T-триггера видно, что, например, между t_1 и t_5 у сигнала C было 4 импульса, а у Q — всего два. То есть частота синхросигнала в два раза выше, чем частота выходного.

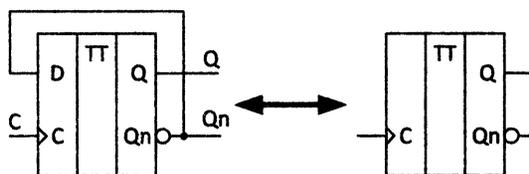


Рис. 6.24. Схема (слева) и условное графическое обозначение (справа) T-триггера

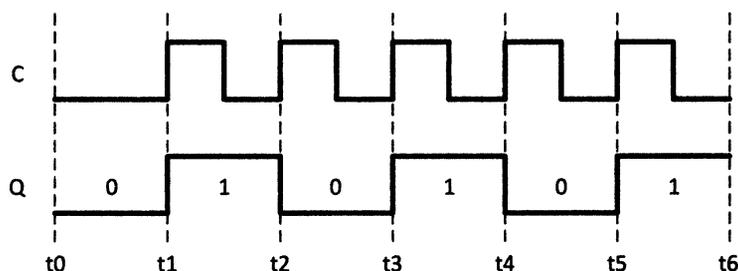


Рис. 6.25. Временная диаграмма Т-триггера

Можно создать Т-триггер из JK и другим способом, без добавления инвертора между его входами и заведения обратной связи с Qn. Попробуйте придумать, каким образом.

Конструкция для самостоятельной разработки

Составьте схему преобразования JK-триггера в Т-триггер без использования дополнительных вентилях. Проверьте правильность этой переделки, доработав соответствующим образом монтажную схему, изображенную на рис. Ц-6.23.

* * *

Монтажные схемы D- и Т-триггеров здесь не приводятся — вы легко сможете модифицировать ранее созданный синхронный RS- или JK-триггер и провести все опыты с ними самостоятельно. Инверторы можно создавать из свободных вентилях микросхем CD4001BE и CD4011BE, уже установленных на плате.

Практическая схема D-триггера на транзисторах

Тем, кого не испугать созданием схем на транзисторах, можно предложить попробовать сделать одноступенчатый D-триггер, срабатывающий по высокому уровню. Он, как обычно, будет содержать гораздо меньше транзисторов, чем D-триггер на вентилях. Хотя, конечно, займет на макетной плате больше места. Одна из классических схем, применяемых при разработке современных микропроцессоров, состоит из трех инверторов и двух двунаправленных ключей, уже знакомых вам из предыдущей главы, и приведена на рис. 6.26. В то время, пока на входе синхронизации С находится высокий уровень, ключ, через который проходит сигнал данных D, открыт, и данные беспрепятственно поступают на выход Q через два инвертора. А верхний по схеме ключ, стоящий в петле обратной связи, закрыт. Триггер в этих условиях прозрачный. Как только уровень синхросигнала сменяется на низкий, путь данным внутрь триггера перекрывается. Но при этом замыкается петля обратной связи. Два выходных инвертора работают через ключ по кругу друг на друга, обеспечивая тем самым надежное защелкивание данных внутри, — вспомните схему, которая рассматривалась самой первой в этой главе. С приходом следующего высокого уровня на вход С все повторяется заново. Обрыв петли обратной связи

нужен для того, чтобы поступившему на вход D сигналу не требовалась «пересилить» бит, хранящийся внутри триггера, если они противоположны друг другу по значению. Тем самым уменьшается задержка переключения триггера. В этой схеме всего 12 транзисторов, против 18 в D-триггере, созданном из синхронного RS-триггера на вентилях, и срабатывающего по уровню.

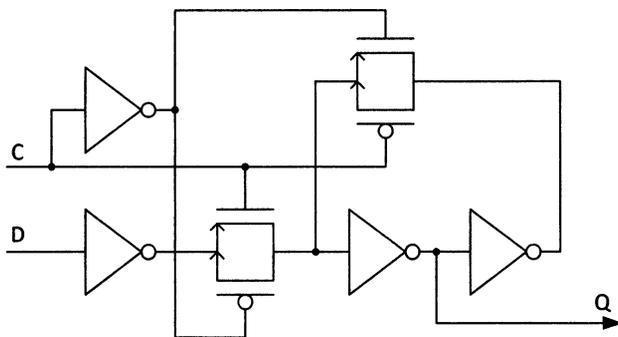


Рис. 6.26. D-триггер на двунаправленных ключах

Но, как уже обсуждалось ранее, двунаправленные ключи на дискретных транзисторах не работают, как от них ожидается, из-за диода между стоком и истоком. Тем не менее знать эту схему необходимо, поскольку в документации на микросхемы D-триггеров она часто встречается. В CD4013BE, например, находятся два двухступенчатых синхронных D-триггера, использующих именно такую схемотехнику и дополненных асинхронными входами сброса и установки, т. е. входами, срабатывающими независимо от состояния синхросигнала.

К счастью, существует и другой вариант схемы триггера с тем же количеством транзисторов, который будет правильно функционировать на макетной плате. На схеме, приведенной на рис. 6.27, вы видите, что в качестве входного инвертора и инвертора в петле обратной связи используются схемы с *третьим состоянием*, рассмотренные в предыдущей главе. Первая пропускает сигнал через себя только по высокому уровню, делая триггер прозрачным, а по низкому уровню переходит в третье состояние с высоким сопротивлением: ни ноль, ни единица. Вторая защелкивает поступившие данные в петле обратной связи по низкому уровню, обеспечивая их хранение, и обрывает обратную связь в то время, когда триггер прозрачный. Такой триггер уже можно собрать на макетной плате, и он будет работать.

На монтажной схеме (рис. Ц-6.28) первая группа из двух транзисторов — инвертор синхросигнала, следующая из четырех — инвертор входного сигнала D с третьим состоянием выхода, затем на плате расположены четыре транзистора такого же инвертора с третьим состоянием выхода в петле обратной связи, потом два транзистора выходного инвертора. Последним установлен пробник выходного сигнала. Значения на входах C и D задаются с помощью гибких перемычек, подключаемых к плюсу или минусу питания. Конечно, вы можете задействовать для создания синхрои импульсов и ранее собранный генератор на двух инверторах.

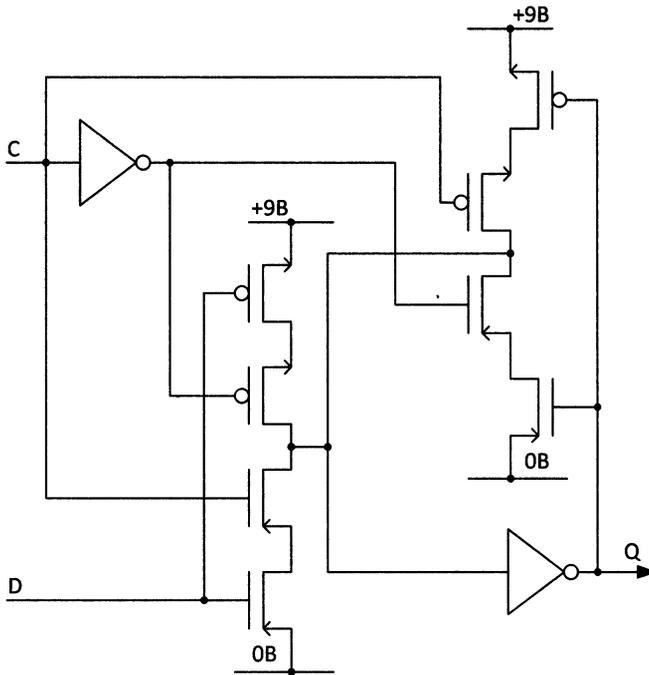


Рис. 6.27. Принципиальная схема D-триггера на инверторах с третьим состоянием

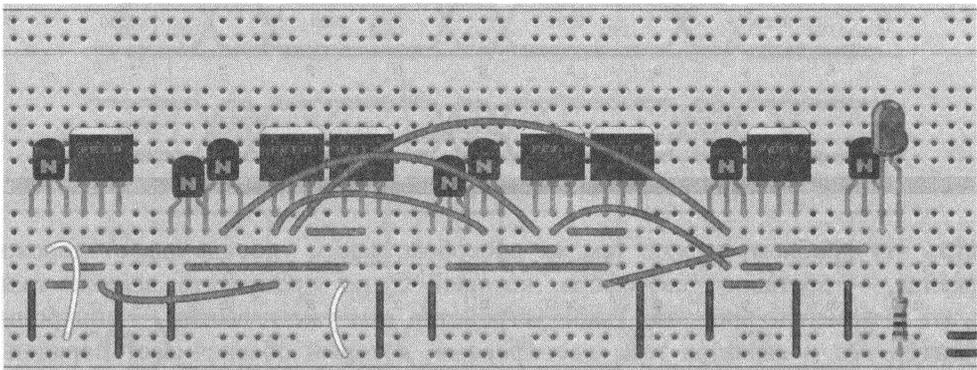


Рис. Ц-6.28. Монтажная схема D-триггера на транзисторах

Поставив два таких триггера друг за другом, и получают вариант «ведущий-ведомый». Также можно добавить и асинхронные входы сброса и установки. В микроэлектронике одинаково часто применяются оба типа схем: и на двунаправленных ключах, и на инверторах с третьим состоянием. Между прочим, на одном кристалле могут находиться одновременно несколько миллионов синхронных D-триггеров. Зачем же их нужно такое количество?

От одного триггера к целому регистру, и зачем процессору конвейер

Рассмотрим, например, четырехбитный сумматор, на вход которого поступают два операнда: $A[3:0]$ и $B[3:0]$. В квадратных скобках, как вы узнали из предыдущей главы, рядом с названием многоразрядных данных указываются номера старшего и младшего разрядов, разделенные двоеточием. На временной диаграмме (рис. 6.29), чтобы не рисовать каждый сигнал отдельно, многобитные данные изображают в виде двух параллельных линий одновременно по уровням нуля и единицы с указанием между ними текущего значения. В тот момент, в который произошло изменение хотя бы одного бита, линии перекрещиваются, а затем между ними вписывается новое значение. Операнды поступают в сумматор из разных устройств микропроцессора, поэтому, например, $A[3:0]$ может идти по более короткому пути и приходиться раньше, чем $B[3:0]$, что и показано на временной диаграмме. Если оба операнда сразу пропускать на вход сумматора, то он будет периодически выдавать ложную информацию, отмеченную штриховкой. И это мы еще не рассматривали вполне реальные случаи, когда разные биты одного операнда приходят с разной скоростью!

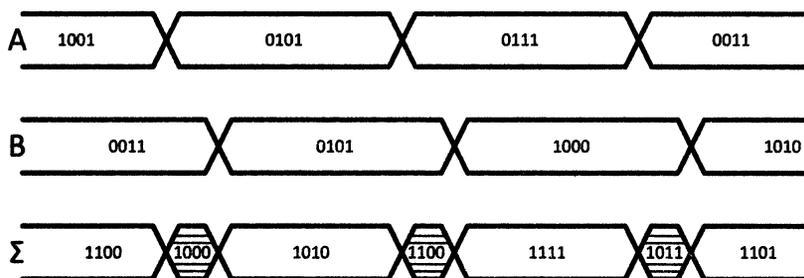


Рис. 6.29. Временная диаграмма четырехбитного сумматора

Решить эту проблему можно тем же способом, каким упорядочивают дорожное движение, — поставив «светофор», который будут пропускать данные только в нужные моменты времени. Такое устройство вам уже известно — синхронный D-триггер. Он и является тем «светофором», который обеспечивает порядок на «магистральных» микропроцессора: *шинах* адресов и данных. Так называются не только проводники питания, но и многоразрядные сигналы, объединенные общим назначением. Соответственно, $A[3:0]$ и $B[3:0]$ в нашем примере — две четырехбитные шины данных. На каждый бит операнда нужно поставить свой триггер. Но так как они обслуживают одну шину и запускаются одним синхросигналом, то их удобно ставить рядом, объединяя в большую схему, которая называется *параллельным регистром* или просто *регистром*.

Более того, регистры ставят не только на входах устройств, но и на их выходах, чтобы комбинационные схемы раньше времени не отправляли результаты своей работы дальше. И даже посередине пути от устройства к устройству — везде, где требуется обеспечить упорядоченность продвижения данных. Если входы и выхо-

ды сумматора дополнить регистрами данных, то изменятся и схема (рис. 6.30), и временная диаграмма этого блока (рис. 6.31).

Как видите, теперь никакой путаницы в вычислениях возникнуть не может. Сигналы на входах А и В ждут, пока синхриимпульс CLK не откроет им путь, чтобы они могли пройти внутрь регистра и на входы операндов сумматора OPA и OPB. Изменения значений операндов приводит к изменению состояния выхода Σ сумматора с некоторой задержкой. К этому моменту регистр, стоящий на выходе, закрыт, поэтому хранит предыдущее вычисленное значение. Какое оно было в первый такт временной диаграммы, нам неизвестно, поэтому состояние выхода S не определено и обозначается как XXXX. Открывается выходной регистр, как и входные, тоже по фронту CLK, передавая тем самым уже готовые и достоверные результаты суммирования дальше.

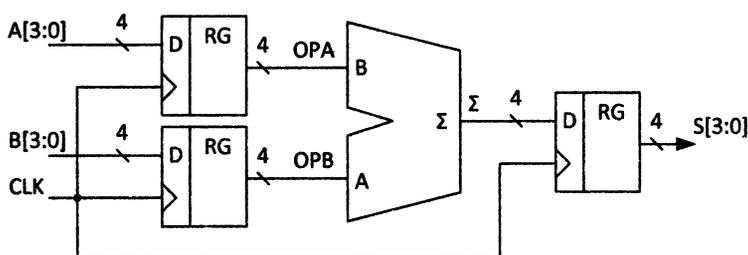


Рис. 6.30. Регистры на входах и выходах сумматора

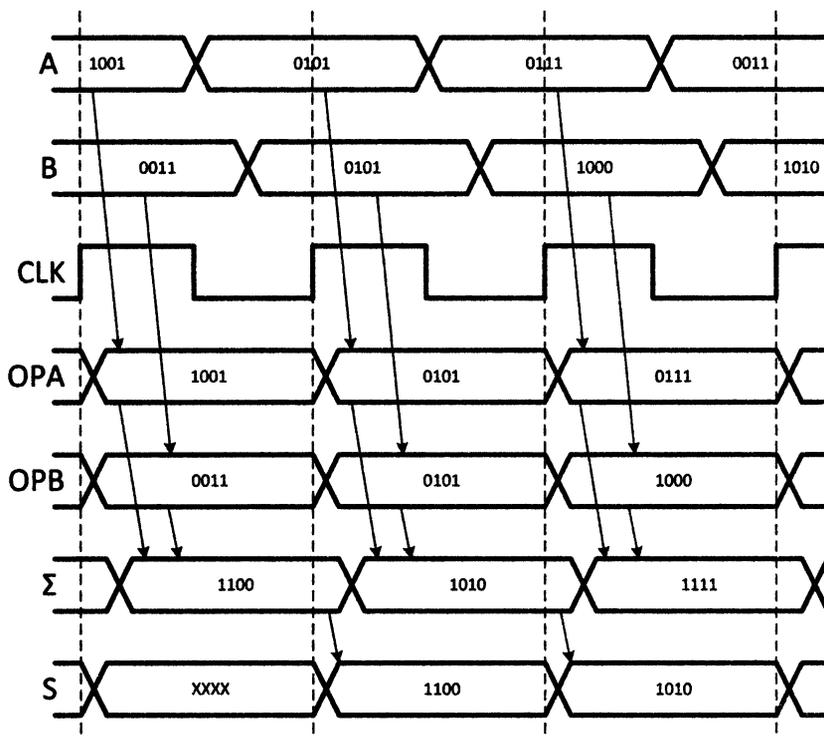


Рис. 6.31. Временная диаграмма сумматора с регистрами на входах и выходах

Этот принцип работы очень похож на фабричный конвейер: выполненная часть общего задания с одного участка передается на следующий, а предыдущий освобождается для новой задачи. Как полусобранный автомобиль движется из цеха в цех, постепенно обрастая колесами, двигателем, дверцами, бамперами, так и данные в микропроцессорах и микроконтроллерах перемещаются от одного устройства к другому с регистра на регистр. Поэтому такая система *регистровых передач* тоже называется *конвейером*. Этот термин часто употребляется применительно к внутреннему устройству микропроцессора. А частота изменения тактового сигнала, приходящего на регистры внутри него, является одной из основных характеристик вычислительной системы. Но, как вы понимаете, сама по себе тактовая частота не является мерилем быстродействия процессора. Не менее важно то, сколько логических операций успевают выполниться за каждый такт. Только зная оба этих параметра, можно сравнивать производительность двух разных процессоров. Если сумматор из нашего примера еле-еле успевает выполнить вычисления за один такт синхросигнала длительностью 2 нс, то при уменьшении периода тактового сигнала до 1 нс придется подождать два такта, прежде чем достоверный результат сложения можно будет получить на выходе S[3:0]. А отсюда следует, что производительность такой системы с ростом частоты никак не увеличивается.

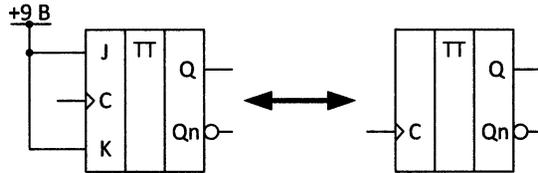
Рассмотренный регистр, данные в который поступают одновременно во все разряды и так же выходят дальше, напомним, называется *параллельным*. Он самый простой: это несколько триггеров с общим синхросигналом и, если они требуются, с общими сигналами сброса и установки. А бывают и другие виды регистров. Здесь уже от рассмотрения схем отдельных триггеров настала пора переходить к сложным последовательностным устройствам.

Ответы на вопросы для самопроверки

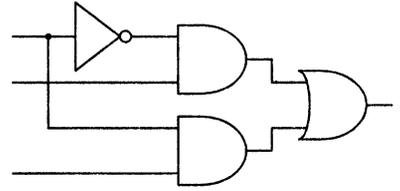
Номер вопроса	Ответ																																								
1	<p>The diagram shows the timing of a 4-bit parallel register. The signals are R, S, Q, and Qn. The time axis is marked from t0 to t6. R and S are active-low signals. Q and Qn are active-low signals. The diagram shows that the output Q changes only when both R and S are high (active-low) and a clock edge occurs.</p> <table border="1"> <thead> <tr> <th>Signal</th> <th>t0</th> <th>t1</th> <th>t2</th> <th>t3</th> <th>t4</th> <th>t5</th> <th>t6</th> </tr> </thead> <tbody> <tr> <td>R</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>S</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>Q</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>Qn</td> <td>1</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>0</td> <td>1</td> </tr> </tbody> </table>	Signal	t0	t1	t2	t3	t4	t5	t6	R	1	0	0	1	0	0	0	S	0	1	0	0	0	1	0	Q	0	1	1	0	0	1	0	Qn	1	0	0	1	1	0	1
Signal	t0	t1	t2	t3	t4	t5	t6																																		
R	1	0	0	1	0	0	0																																		
S	0	1	0	0	0	1	0																																		
Q	0	1	1	0	0	1	0																																		
Qn	1	0	0	1	1	0	1																																		

Номер вопроса	Ответ																				
2	Синхронный триггер пропускает управляющие сигналы только по активному уровню тактового импульса, а асинхронный переходит в другой режим работы сразу в момент переключения управляющего сигнала																				
3	У JK-триггера нет запрещенного состояния, вместо него есть переключательный режим																				
4	<table border="1" data-bbox="242 388 1066 569"> <thead> <tr> <th data-bbox="242 388 392 435">C</th> <th data-bbox="392 388 530 435">D</th> <th data-bbox="530 388 657 435">Q(t)</th> <th data-bbox="657 388 783 435">Qn(t)</th> <th data-bbox="783 388 1066 435">Режим работы</th> </tr> </thead> <tbody> <tr> <td data-bbox="242 435 392 482">0</td> <td data-bbox="392 435 530 482">X</td> <td data-bbox="530 435 657 482">Q(t-1)</td> <td data-bbox="657 435 783 482">Qn(t-1)</td> <td data-bbox="783 435 1066 482">Хранение</td> </tr> <tr> <td data-bbox="242 482 392 529">1</td> <td data-bbox="392 482 530 529">0</td> <td data-bbox="530 482 657 529">0</td> <td data-bbox="657 482 783 529">1</td> <td data-bbox="783 482 1066 529">Передача нуля</td> </tr> <tr> <td data-bbox="242 529 392 569">1</td> <td data-bbox="392 529 530 569">1</td> <td data-bbox="530 529 657 569">1</td> <td data-bbox="657 529 783 569">0</td> <td data-bbox="783 529 1066 569">Передача единицы</td> </tr> </tbody> </table>	C	D	Q(t)	Qn(t)	Режим работы	0	X	Q(t-1)	Qn(t-1)	Хранение	1	0	0	1	Передача нуля	1	1	1	0	Передача единицы
C	D	Q(t)	Qn(t)	Режим работы																	
0	X	Q(t-1)	Qn(t-1)	Хранение																	
1	0	0	1	Передача нуля																	
1	1	1	0	Передача единицы																	
5	Двухступенчатый синхронный триггер переключается по фронту или срезу тактового сигнала, а одноступенчатый делает это по уровню																				

Ответ на задание по конструкции для самостоятельной разработки



ГЛАВА 7



Сдвиговые регистры и счетчики

Вам потребуются:

- 7 транзисторов типа n-МОП;
- 8 светодиодов;
- 8 резисторов сопротивлением 100 Ом;
- 7 резисторов сопротивлением 470 Ом;
- 2 резистора сопротивлением 1 кОм;
- один резистор сопротивлением 10 кОм;
- одна кнопка;
- по одному электролитическому конденсатору емкостью 10, 22, 100, 470 и 1000 мкФ;
- две микросхемы CD4013BE;
- по одной микросхеме CD4001BE, CD4011BE, CD4081BE и CD4511BE;
- семисегментный индикатор CS56-11.

Эстафета для триггеров: конструкция сдвигового регистра

Помимо уже изученных вами регистров с параллельной загрузкой данных, т. е. одновременной во все триггеры регистра, существуют и *сдвиговые*, с последовательной загрузкой. Вы сталкивались с их работой, например, при наборе числа на калькуляторе. При нажатии первой цифры она отображается на дисплее справа. Когда вы вводите вторую цифру, то первая сдвигается влево, а на ее месте появляется вторая. Это продолжается до тех пор, пока все число не появится на экране. Именно так работает сдвиговый регистр — данные приходят в него с одной стороны и с каждым тактом синхросигнала смещаются дальше. Его схема весьма проста и состоит из цепочки D-триггеров, в которой выход предыдущего триггера подключается ко входу данных следующего. Как вы помните, D-триггер задерживает

поступивший на него бит до очередного тактового импульса. За счет этого и происходит последовательное продвижение информации по всей цепочке. Если триггеры имеют асинхронные входы сброса и установки, то через них в сдвиговый регистр можно загружать данные и параллельно, т. е. во все биты сразу. На схеме, представленной на рис. 7.1, все сигналы установки объединены в одну шину, и то же самое сделано с сигналами сброса. То есть в данном случае можно в произвольный момент времени или обнулить все биты регистра, или записать в каждый из них единицу. Но при необходимости можно комбинировать асинхронные входы самым произвольным образом, загружая в триггеры любое наперед выбранное значение битов в качестве исходных.

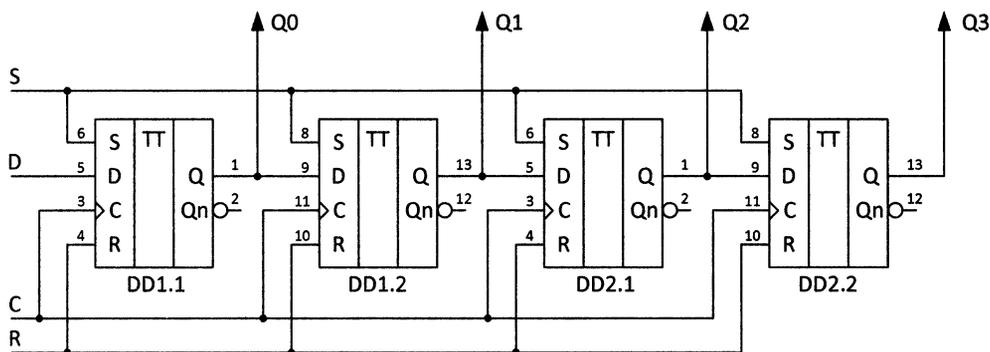


Рис. 7.1. Сдвиговый регистр

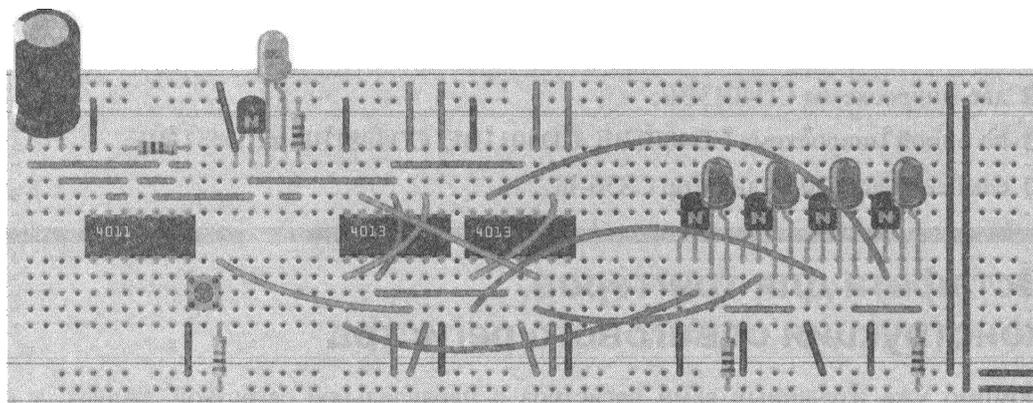


Рис. Ц-7.2. Монтажная схема сдвигового регистра с генератором и светодиодной индикацией

Приведенный на рис. 7.1 четырехразрядный сдвиговый регистр был сконструирован на базе двух микросхем CD4013BE, каждая из которых содержит по два D-триггера с асинхронными входами сброса и установки. На схеме дана нумерация выводов для удобства сборки, а на макетной плате (рис. Ц-7.2) эти две микросхемы находятся рядом друг с другом справа. В CD4013BE, как и в логических элементах, плюс питания подается на четырнадцатый вывод, а минус — на седьмой. Входы сброса и установки имеют активный высокий уровень, и так как они не использу-

ются в этом простейшем примере, то на макетной плате подключены к минусу питания. Это приводит к тому, что при включении схемы выходы триггеров оказываются в случайном состоянии, но для иллюстрации работы сдвигового регистра это непринципиально. Каждый выход Q снабжен светодиодной индикацией. На вход синхронизации С сигнал поступает с уже известной вам из предыдущей главы схемы генератора импульсов на базе двух вентилях 2И-НЕ из состава микросхемы CD4011BE, установленной на плате слева. Нажатием и отпусканием кнопки, расположенной на плате возле седьмого выхода этой микросхемы, вы можете передавать на вход данных сдвигового регистра (вывод 5 DD1.1) ноль или единицу и затем наблюдать, как заданный бит проходит по цепочке светодиодов с каждым тактовым импульсом.

Можно сделать так, чтобы сдвиговый регистр передавал за каждый такт не один бит, а сразу несколько. Для этого нужно вместо триггеров поставить параллельные регистры требуемой разрядности. Например, с помощью сдвигового регистра, организованного на базе четырехбитных параллельных регистров, можно запоминать десятичные коды кнопок с шифратора кодового замка. В этом случае нажатие любой кнопки на цифровой клавиатуре не только активирует работу шифратора унитарного кода в двоичный, но и вызывает появление одиночного синхроимпульса, который записывает в первый каскад сдвигового регистра очередную цифру в двоичном коде и сдвигает ранее введенные числа в следующие каскады. Легко посчитать, что для кода, состоящего из трех десятичных цифр, всего потребуется 12 триггеров. То есть одной макетной платы на 800 точек будет мало.

Вопросы для самопроверки

1. Как будет работать сдвиговый регистр, если на схеме, изображенной на рис. 7.1, выход Q3 подать на вход D?

Забавный светильник «Бегущие огни»

Если вместо кнопки в уже собранную на плате схему сдвигового регистра поставить еще один генератор импульсов, то можно сделать устройство с эффектом бегущих огней. В микросхеме CD4011BE, имеющейся на плате, остались еще два недействующих вентиля. Их и нужно использовать для задания последовательности битов, как показано на схеме, приведенной на рис. 7.3. При этом емкость конденсатора в генераторе последовательности данных должна отличаться от емкости конденсатора в генераторе синхросигнала. Тогда несовпадение в частотах переключения приведет к разнообразным световым эффектам. Поэкспериментируйте сами, меняя емкости обоих конденсаторов в пределах 10...1000 мкФ в любых сочетаниях.

К каждому транзистору на выходе сдвигового регистра можно подключить не один, а несколько светодиодов разных цветов. Поместив плату внутри корпуса, а светодиоды — на его передней панели в виде какого-нибудь узора, можно сделать устройство для украшения интерьера с интересными световыми эффектами, которое можно использовать в качестве подсветки комнаты или ночника для младших чле-

нов семьи, если скорость переключения огней будет подобрана невысокой, успокаивающей. Максимальное количество светодиодов определяется предельно допустимым током стока транзистора. При сопротивлении токоограничивающего резистора 100 Ом, как на схеме, и напряжении питания 9 В максимальный для p-канального транзистора 2N7000TA ток стока величиной 200 мА не позволяет подключать более двух светодиодов. Можно увеличить сопротивление резисторов до 220 Ом — это позволит за счет небольшой потери яркости удвоить количество огней. Но лучше установить p-канальный прибор IRFU9024NPBF, который способен работать на гораздо более высокую нагрузку. Конечно же, учитывая другой тип проводимости транзистора, его затвор должен подключаться не к прямому выходу триггеров, а к инверсному. И светодиоды в цепи стока p-канального транзистора, как вы помните, подсоединяются иначе.

Если вы захотите увеличить количество триггеров в сдвиговом регистре, разнообразив эффект еще больше, то вашу фантазию может ограничивать только размер

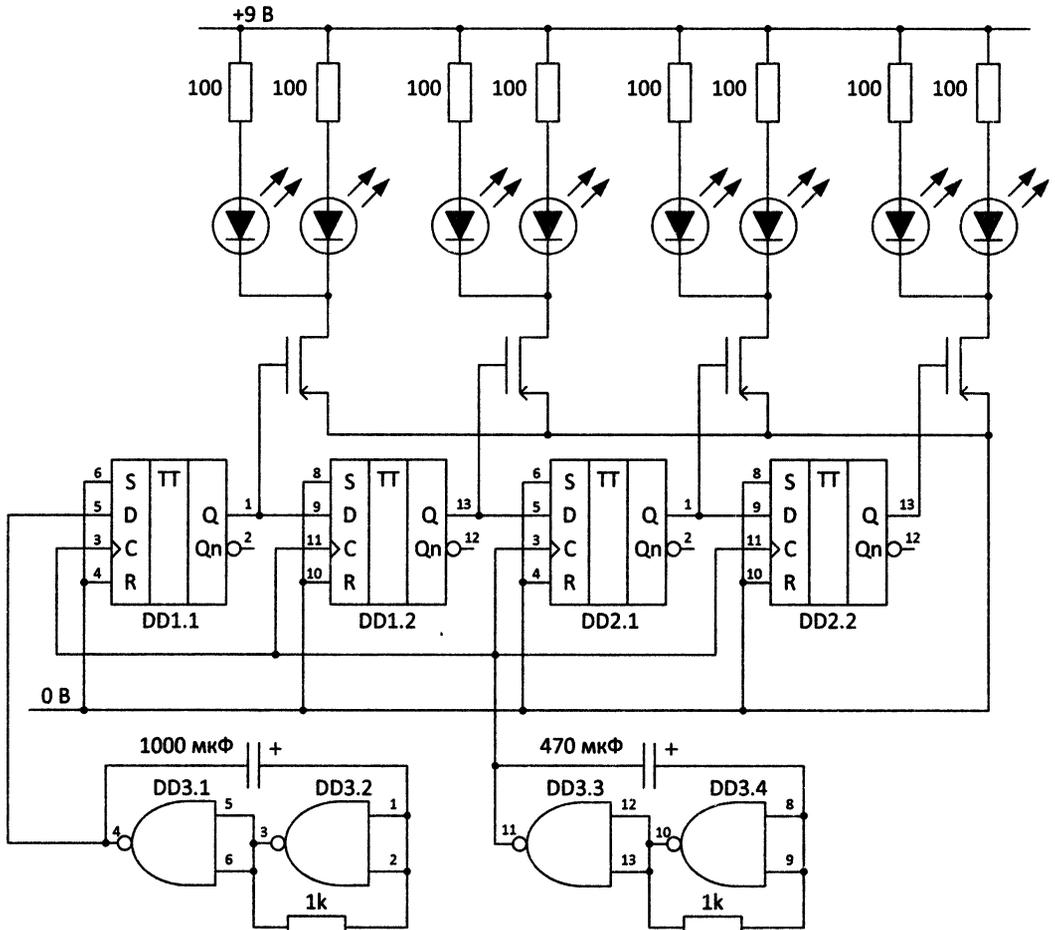


Рис. 7.3. Схема светильника «Бегущие огни»

используемой платы. А в качестве источника питания большой схемы вместо «Кроны» лучше взять любой сетевой адаптер, выдающий постоянное напряжение 9...12 В и рассчитанный на ток необходимой величины. Эти параметры обычно указаны на корпусе адаптера. Маркировка на нем содержит два важных реквизита — например: «AC 210-240V» и «DC 12V, 1A». Первый указывает напряжение сети переменного тока (от англ. AC, alternate current), на которое рассчитан адаптер. А второй — постоянное выходное напряжение (от англ. DC, direct current) и предельный ток нагрузки, в данном случае 12 В и 1 А соответственно. Желательно, чтобы ваша схема не потребляла более 80% от максимально допустимого тока, чтобы не перегружать выходной каскад адаптера. Потреблением цифровой части в сравнении с током светодиодов можете пренебречь.

Те из вас, кто уже знаком с параллельными и последовательными электрическими цепями, могут удивиться тому, что на этой схеме каждый светодиод подключен к питанию через отдельный резистор. Ведь через каждый резистор сопротивлением 100 Ом (т. е. и через каждый светодиод) протекает ток примерно 90 мА, и теоретически можно было бы поставить один резистор номиналом 50 Ом, а к нему подключить аноды обоих светодиодов. И через каждый из них тек бы тот же ток 90 мА. А можно было бы и вообще все 8 анодов объединить на одном резисторе с сопротивлением 12,5 Ом. Это уменьшило бы размер и количество компонентов устройства, не изменив величины токов, протекающих через светодиоды. Так, на монтажной схеме простого сдвигового регистра (см. рис. Ц-7.2) светодиоды, показывающие состояние выходов триггеров, подключены попарно через один резистор сопротивлением 100 Ом. Да и для других устройств в этой главе можно увидеть, что индикация зачастую сделана так же. Чтобы только разобраться в работе схемы и уменьшить количество компонентов на плате, такой подход оправдан. Но если вы планируете собрать устройство, рассчитанное не на несколько часов занятий с целью отладки и изучения с последующим демонтажем, а на длительное применение, то необходимо позаботиться об обеспечении максимальной надежности. Предположим, что через единственный резистор к питанию будут подключены все восемь светодиодов. И один из них вышел из строя, или нарушился контакт между его выводом и платой. В результате ток через отключившийся светодиод не идет, а распределяется теперь на меньшее количество цепей. Значит, каждый оставшийся светодиод получает теперь больше тока и перегревается. От этого может выйти из строя еще один, и ситуация для остальных станет еще хуже. Чтобы этого не произошло, необходимо полностью разделить цепи питания светодиодов, как и сделано на схеме (см. рис. 7.3). В случае выхода из строя любого из них остальные просто ничего не заметят.

Промышленностью в серии CD4000BE выпускаются и готовые микросхемы разнообразных сдвиговых регистров в сборе, которые вы можете использовать в этом и других своих устройствах, чтобы сделать их более компактными. Такие регистры могут иметь самую разную разрядность, набор входов для установки начальных значений битов и даже возможность менять направление сдвига на противоположное. Однако изучение сдвиговых регистров, собранных самостоятельно на отдельных D-триггерах, позволяет в мельчайших подробностях разобраться в работе этих устройств и применять их в дальнейшем осознанно.

Считалочка для цифровых схем

Схема простейшего счетчика на Т-триггерах

С помощью триггеров создаются не только последовательные и параллельные регистры, но и счетчики импульсов. Рассмотрим четырехбитный счетчик на базе той же микросхемы CD4013BE, что была использована для сборки сдвигового регистра (рис. 7.4). Часть проводников между триггерами придется подключить иначе. Обратите внимание, что инверсный выход каждого триггера теперь подключается к его же входу, образуя таким образом Т-триггер. Как вы помните, он меняет значение своего выхода на противоположное при каждом очередном фронте синхросигнала.

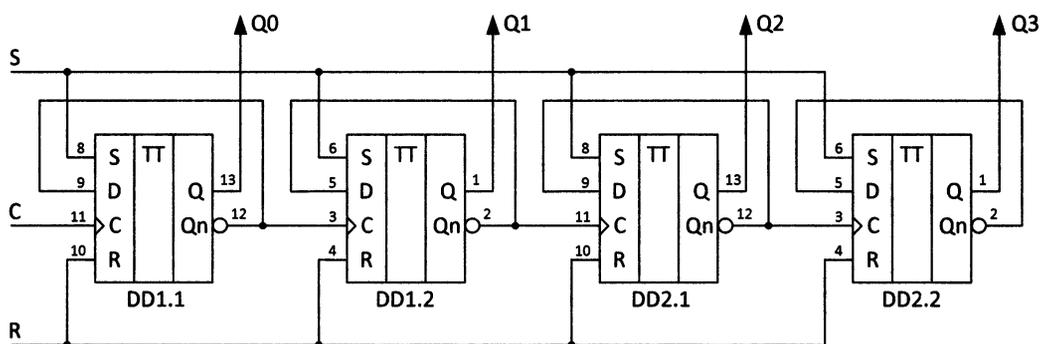


Рис. 7.4. Принципиальная схема двоичного счетчика

Временная диаграмма, приведенная на рис. 7.5, поясняет принцип работы двоичного счетчика на примере сигналов для первых трех разрядов. Стрелками на ней показано, как фронт сигнала с предыдущего каскада управляет переключением следующего. Значения кодовой последовательности Q_2 , Q_1 , Q_0 в двоичном представлении приведены в самом низу диаграммы. Видно, что по мере поступления тактовых импульсов последовательно перебираются числа от 0 до 7 и далее по кругу в том же порядке. Это называется *циклическим счетом по модулю 8 в прямом направлении*, т. к. перебираются 8 последовательных чисел от меньшего к большему со сбросом в 0 при достижении максимального значения. Асинхронными входами сброса и установки можно задавать начальные значения счетчика, как это делалось в сдвиговом регистре. Нетрудно догадаться, что приведенная на рис. 7.4 схема из четырех триггеров представляет собой циклический счетчик по модулю 16, считающий в прямом направлении. Собрав на монтажной плате такое устройство вместе с генератором синхросигналов, как это показано на рис. Ц-7.6, вы сможете наблюдать последовательно перебираемые двоичные числа в виде последовательно зажигающихся светодиодов.

С помощью JK-триггеров также создают счетчики. Для этого используется переключательный режим работы триггеров. Надо полагать, что объяснять работу такого счетчика, схема которого приведена на рис. 7.7, не требуется. Его можно

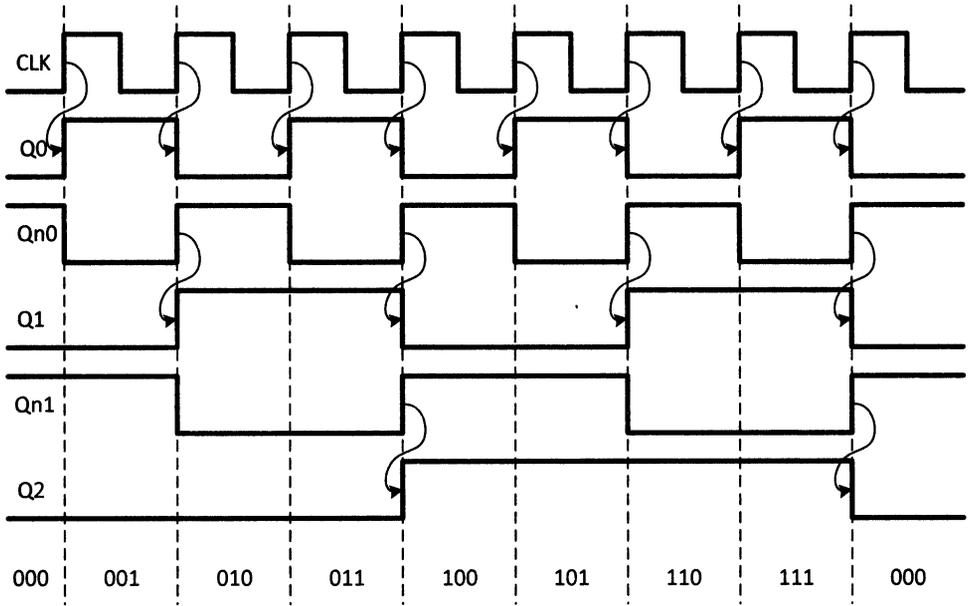


Рис. 7.5. Временная диаграмма двоичного счетчика

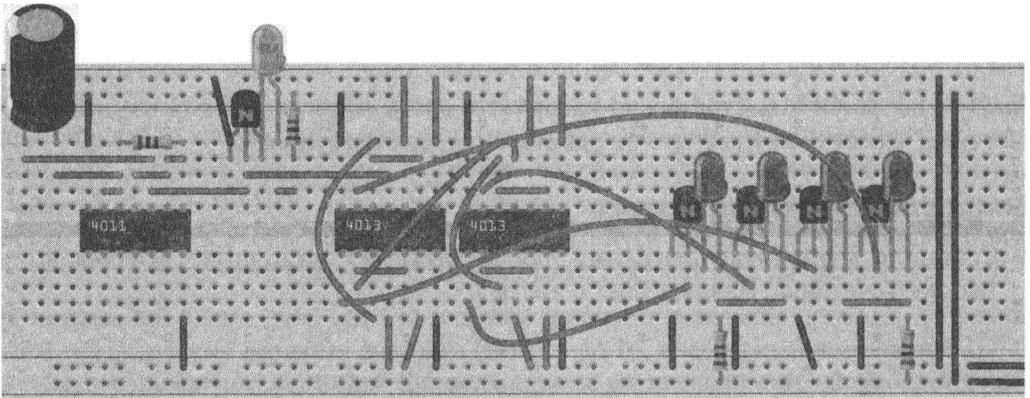


Рис. Ц-7.6. Монтажная схема двоичного счетчика с генератором и светодиодной индикацией

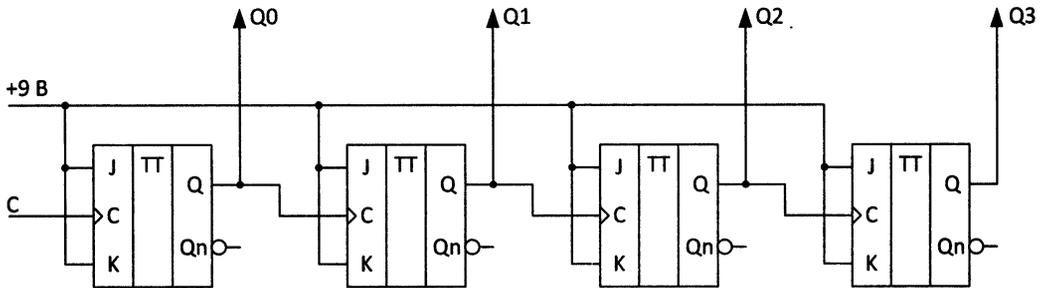


Рис. 7.7. Двоичный счетчик на JK-триггерах

собрать, например, на базе микросхемы CD4027BE, содержащей два триггера с асинхронными входами сброса и установки, которые на схеме не показаны для простоты.

Вычитающий счетчик

А можно ли построить счетчик, который «отматывает» числа в обратном направлении: от большего к меньшему? Да, такой счетчик называется *вычитающим*, и для его создания нужно всего лишь подавать сигналы на следующий каскад не с инверсного выхода предыдущего, а с прямого, как это показано на рис. 7.8.

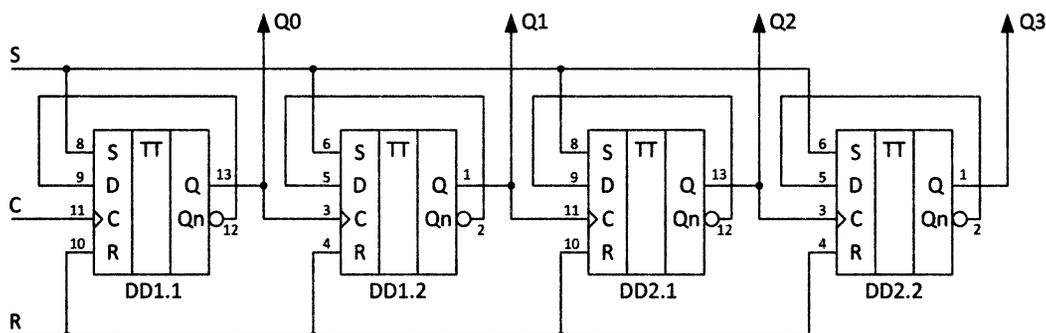


Рис. 7.8. Вычитающий счетчик

Вам предлагается самостоятельно изобразить временную диаграмму вычитающего счетчика, чтобы убедиться, что он работает так, как требуется.

Вопросы для самопроверки

- Составьте временную диаграмму для первых трех выходных сигналов вычитающего счетчика. Начните счет с числа 7.

* * *

Переставив всего три провода на макете прямого счетчика, который был сделан ранее (см. рис. Ц-7.6), вы легко проверите на практике правильность функционирования показанной вычитающей схемы.

Можно модифицировать ее таким образом, чтобы счетчик смог считать и в прямом, и в обратном направлении по команде. Такой вариант называется *реверсивным счетчиком*, и для его реализации требуется добавить несколько вентилях между каждой парой каскадов, чтобы получился мультиплексор, или взять уже готовый мультиплексор с двумя входами — например, CD4053BE. На рис. 7.9 сигнал U/D (от англ. Up, вверх и Down, вниз) задает направление счета.

Счетчики можно каскадировать, т. е. ставить последовательно друг за другом, для увеличения их разрядности. Если с выхода Q3 четырехразрядного счетчика подать сигнал на вход следующего такого же, то, как нетрудно вычислить, модуль счета такой схемы будет уже 256.

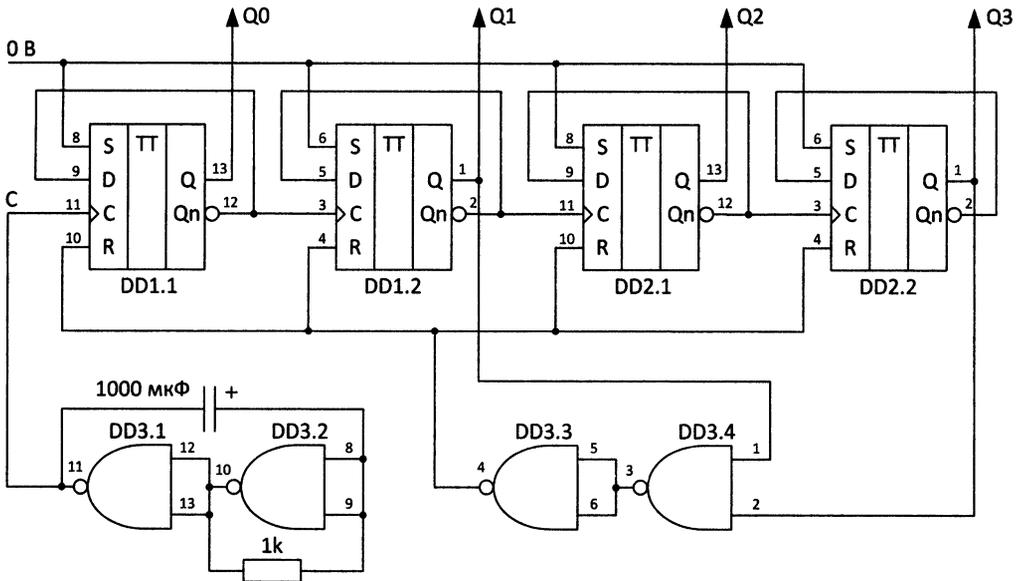


Рис. 7.10. Принципиальная схема счетчика по модулю 10

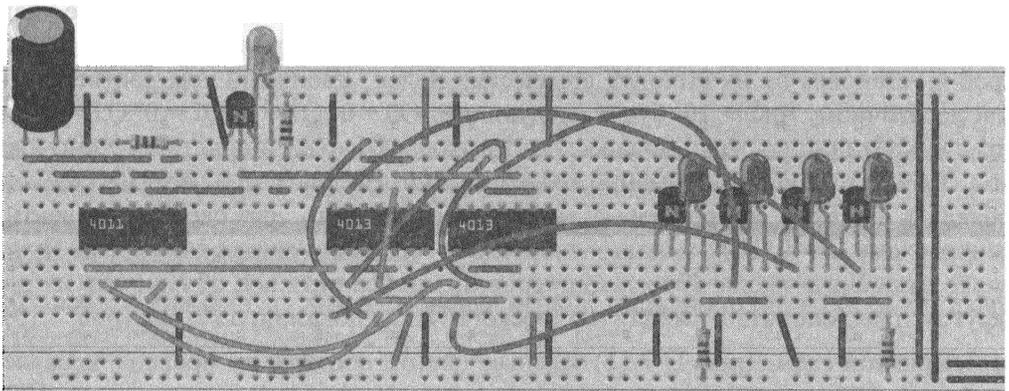


Рис. Ц-7.11. Монтажная схема счетчика по модулю 10

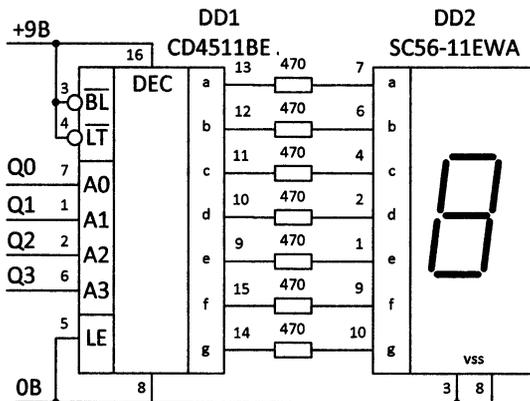


Рис. 7.12. Схема цифровой индикации для счетчика

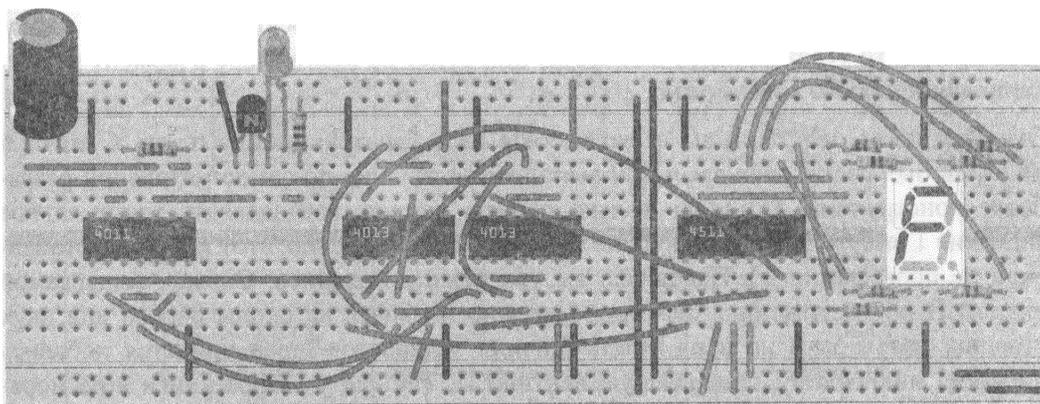


Рис. Ц-7.13. Монтажная схема счетчика по модулю 10 с цифровой индикацией

вариантом обозначены зеленым цветом, а также изменены точки подключения коричневых проводников, соединяющих выходы счетчика с входами драйвера индикатора.

Конструкция для самостоятельной разработки

Составьте схему счетчика по модулю 7. Генератор синхросигнала в схему включать не нужно.

Вопросы для самопроверки

3. Сколько триггеров необходимо для построения счетчиков по модулю 19, 8, 45 и 128?
4. Можно ли из трех триггеров получить счетчики по модулям 5, 12, 8, 21?
5. Каково максимальное число, до которого может считать счетчик по модулю 7, 16, 32?

Две схемы электронного кубика для игр

Просто перебирать цифры хоть в двоичном, хоть в десятичном представлении — быстро надоедает. Гораздо интереснее собрать что-нибудь, имеющее практическое применение. Например, можно модифицировать изготовленный счетчик таким образом, чтобы он перебирал цифры от 1 до 6, останавливаясь в случайном месте. Такая схема будет электронным имитатором игрового кубика. Для его создания, во-первых, нужно увеличить частоту следования счетных импульсов так, чтобы на индикаторе нельзя было успевать различить цифры. Для этого конденсатор 1000 мкФ заменяется на 10 мкФ. Но при отладке схемы лучше оставить конденсатор большой емкости, чтобы видеть, правильно ли она функционирует, а замену на 10 мкФ произвести уже после того, как вы убедитесь, что все в порядке. Во-вторых, требуется разорвать кнопкой провод, идущий от генератора импульсов ко входу счетчика. Тогда нажатие кнопки будет запускать счет, а ее отпускание — останавливать его.

Так как цифры при малой емкости времязадающего конденсатора мелькают слишком быстро, то угадать, какая именно из них будет на индикаторе в момент размыкания цепи, просто невозможно.

Эти изменения просты, и вы легко внесете их в конструкцию на плате. Счет от 0 до 6 уже был реализован вами при выполнении задания по самостоятельной разработке конструкции. Осталось только решить вопрос, как задавать минимальное значение 1 вместо 0. Ведь на кубике нуля нет (если, конечно, этот кубик изготавливали не программисты). Наверняка вы уже сами догадались. Сигнал, выдаваемой схемой И по достижению значения 7, нужно подавать на входы сброса для двух старших разрядов и на вход установки для младшего. Здесь придется добавить лишнюю микросхему, содержащую вентили И. Так как у автора книги в распоряжении была только микросхема CD4081BE с четырьмя вентилями 2И, то она и применялась для генерации сигнала сброса. Также можно использовать CD4082BE, в состав которой входят два 4И или CD4073BE с тремя 3И. В любом случае установленные на плате компоненты придется заставить потесниться, чтобы добавить еще один DIP-корпус. Принципиальная схема электронного кубика на трех D-триггерах показана на рис. 7.14, а монтажная — на рис. Ц-7.15.

Как можно видеть, на плате остался индикатор переключений синхросигнала, состоящий из транзистора, светодиода и резистора. Но вы можете его убрать, т. к. после наладки схемы он становится совершенно не нужным. Дальнейшее упрощение электронного кубика может быть связано с использованием готовых микросхем счетчиков, выпускаемых промышленностью, — например, CD4029BE. Этот четырехразрядный счетчик имеет вход переключения между двоичным и десятичным счетом, между прямым и обратным направлением, а также входы установки начального состояния. При его использовании на плате вместо двух сдвоенных триггеров CD4013BE и вентиля 2И CD4081BE понадобится всего одна микросхема счетчика. Если вы владеете навыками пайки, то после отладки устройства на макетной плате сможете спаять его и разместить в подходящем компактном корпусе.

А может быть, вы предпочли бы изменить цифровую индикацию на традиционную для игральных костей и кубиков — в виде точек на его грани, роль которых будут выполнять светодиоды? Попробуем реализовать и такой вариант, почему бы и нет. В этом случае потребуется 7 светодиодов, расположенных, как показано на рис. 7.16. Официального названия для такого устройства отображения информации нет, и мы с полным правом назовем его «семиточечным индикатором». Светодиоды для определенности обозначим латинскими буквами от а до g, как и в семисегментном индикаторе. Теперь нужно составить таблицу истинности для дешифратора нашего «кубика».

Выходы триггеров счетчика обозначим по порядку: А1, А2, А3. Формируемый ими двоичный код числа должен зажигать или гасить необходимые светодиоды. Как и раньше для семисегментного индикатора, для семиточечного высокий уровень сигнала включает светодиод, а низкий выключает. Все эти рассуждения традиционно представлены в виде таблицы истинности (табл. 7.1).

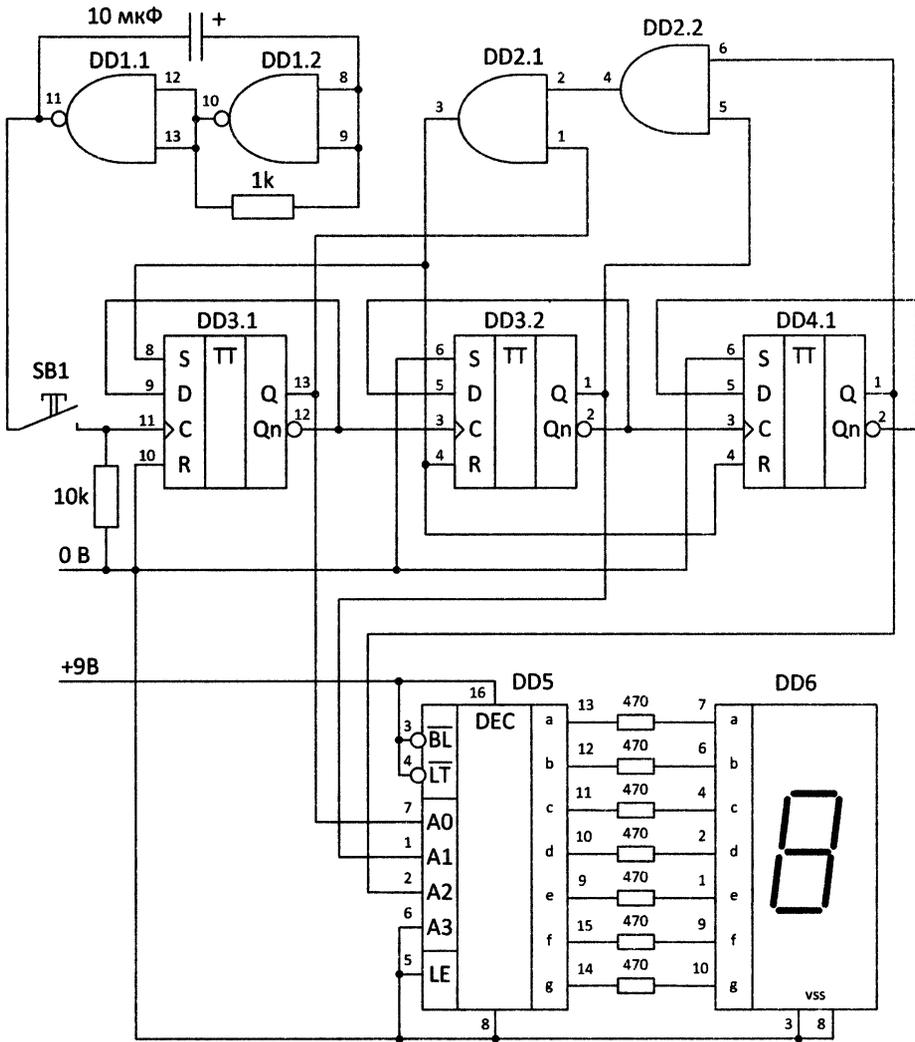


Рис. 7.14. Принципиальная схема электронного кубика

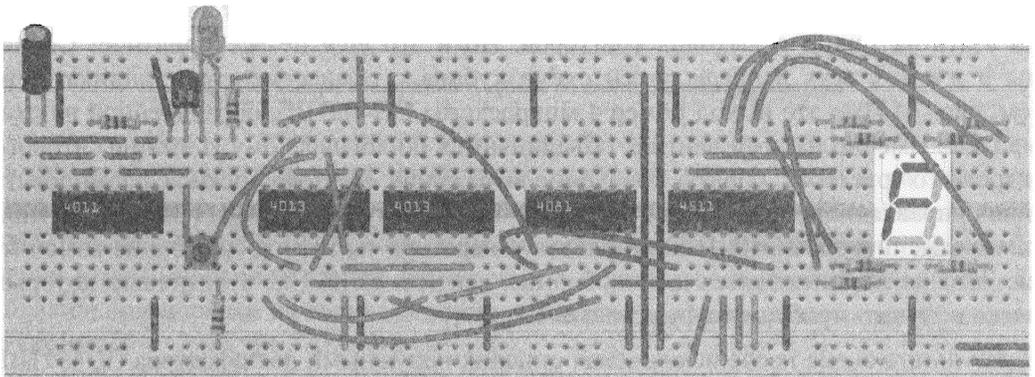


Рис. Ц-7.15. Электронный кубик на макетной плате

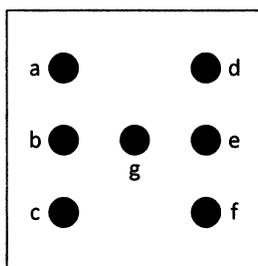


Рис. 7.16. Порядок обозначения светодиодов в «семиточечном индикаторе»

Таблица 7.1. Таблица истинности электронного кубика

Число	A3	A2	A1	a	b	c	d	e	f	g
1	0	0	1	0	0	0	0	0	0	1
2	0	1	0	0	0	1	1	0	0	0
3	0	1	1	0	0	1	1	0	0	1
4	1	0	0	1	0	1	1	0	1	0
5	1	0	1	1	0	1	1	0	1	1
6	1	1	0	1	1	1	1	1	1	0

При рассмотрении табл. 7.1 видно, что в силу принятых на игральными кубиках правилах обозначения чисел, сигналы f и a одинаковы. Аналогичным образом $b = e$ и $c = d$. Дальнейшее внимательное изучение таблицы позволяет сформировать крайне простые выражения для всех логических функций дешифратора кубика:

$$a = f = A3$$

$$b = e = A3 \cdot A2$$

$$c = d = A3 + A2$$

$$g = A1$$

То есть два из четырех сигналов специальным образом получать не нужно — можно взять их прямо с выходов триггеров. Но и это еще не все приятное! Сигналы b и e уже есть в предыдущей схеме кубика с семисегментным индикатором (см. рис. 7.14) — это выход номер 4 микросхемы CD4081BE, обозначенной на схеме DD2. Именно на этом вентиле объединяются по логическому И выходы второго и третьего триггеров, образуя требуемое выражение. Поэтому нужно сформировать лишь управляющие импульсы c и d . Если микросхему DD1 CD4011BE на макетной плате заменить на CD4001BE, то работа генератора тактового сигнала не изменится. Зато на плате появятся два свободных вентиля ИЛИ-НЕ, с помощью которых легко получить требуемую функцию ИЛИ.

В результате схема после исключения ставших ненужными дешифратора CD4511BE и семисегментного индикатора приобретает вид, показанный на рис. 7.17.

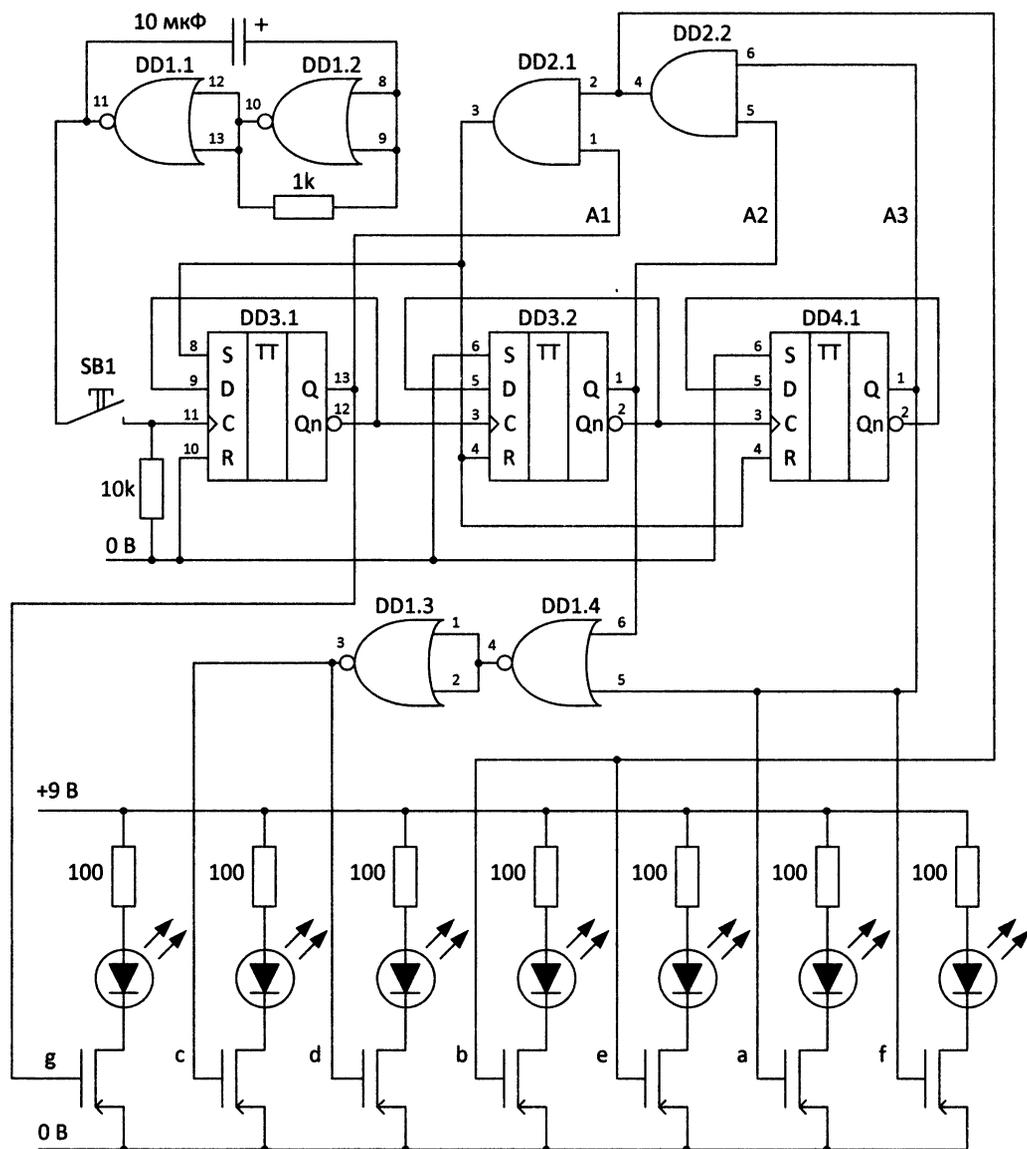


Рис. 7.17. Схема электронного кубика со светодиодной индикацией

На схеме обозначены сигналы на выходах триггеров (A1, A2, A3) и на затворах транзисторов (a, b, c, d, e, f, g), чтобы было легче разобраться в подключении компонентов друг к другу. Вы можете выбрать любой удобный вам корпус для размещения в нем собранной схемы таким образом, чтобы светодиоды располагались в правильном порядке на верхней стороне, создавая привычный всем вид игрального кубика.

Синхронные и асинхронные счетчики

Все рассмотренные в этой главе счетчики называются *асинхронными*, поскольку тактовый сигнал подается только на первый триггер в цепочке, а следующие переключаются уже по мере срабатывания предыдущих. Каждый элемент схемы переключается не мгновенно, а в течение некоторого времени, пусть и малого, но вполне измеряемого и указываемого в документации на микросхему или транзистор. В результате задержка между подачей синхроимпульса и появлением сигнала на выходах счетчика будет увеличиваться от младшего разряда к старшему пропорционально количеству триггеров. Для высокопроизводительных систем, например, микропроцессоров, такое положение дел недопустимо. Да и в многоразрядных счетчиках из-за большой задержки переключения последнего разряда требования, предъявляемые к быстродействию, могут не выполняться. Поэтому были разработаны схемы *синхронных* счетчиков. В них тактовый сигнал подается на все триггеры одновременно, а данные для переключения каждого разряда формируются с помощью комбинационных схем, на входы которых поступают данные с предыдущих триггеров. Общий принцип работы формулируется просто: следующий триггер должен переключиться только тогда, когда все предыдущие разряды равны единице: первый разряд срабатывает при единице на нулевом, для второго триггера нужно двухвходовое И с нулевого и первого разрядов, для третьего — трехвходовое и т. д., как показано на рис. 7.18. Поэтому, например, для 20-разрядного счетчика 19-входовое И нужно будет конструировать из нескольких каскадов более простых вентилях.

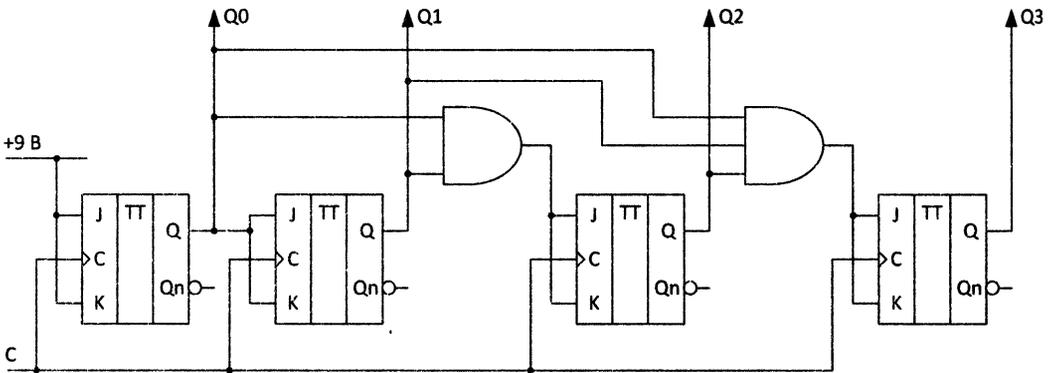


Рис. 7.18. Синхронный счетчик

Иногда в литературе можно встретить следующий вариант синхронного счетчика, в котором применяется каскадирование вентилях 2И (рис. 7.19).

Казалось бы, такая схема очень проста и удобна. Но при внимательном рассмотрении видно, что за счет последовательного включения 2И между разрядами период тактового импульса не может быть меньше суммарной задержки всех вентилях в этой цепочке. Иначе сигнал от младшего значащего разряда просто не успеет добежать до последнего триггера к тому моменту, когда придет следующий син-

хроимпульс. Для быстродействующих многоразрядных счетчиков такой подход может оказаться неприменим, а вот предыдущая схема (см. рис. 7.18), хоть и более громоздкая, обладает лучшим быстродействием. Этот пример показывает, что грамотный инженер должен обладать широким кругозором, чтобы хорошо ориентироваться в преимуществах и недостатках разных методов и выбирать для решения очередной задачи именно тот, который подходит для нее наилучшим образом. Руководствуясь этим соображением, можно утверждать, что при тех частотах тактовых сигналов, которые используются в схемах этой книги, разницу в скорости между разными типами счетчиков заметить не получится. Поэтому для всех применений с частотами вплоть до единиц, а то и десятков мегагерц, простейшие асинхронные счетчики, подобные тем, что вы уже собирали на макетной плате, более чем пригодны, не говоря уже о готовых микросхемах счетчиков серии CD4000BE.

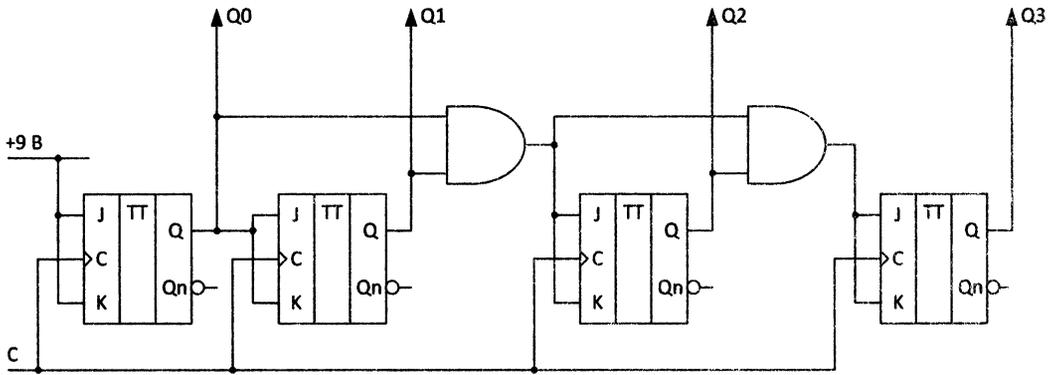


Рис. 7.19. Другой вариант синхронного счетчика

Вопросы для самопроверки

6. Какой счетчик работает быстрее: синхронный или асинхронный?
7. Какой счетчик наиболее простой по конструкции: синхронный или асинхронный?

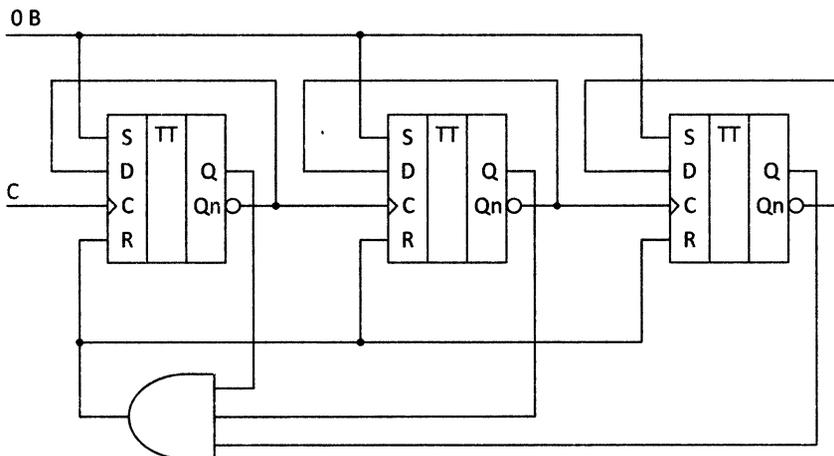
* * *

Изучение основных последовательностных блоков на этом можно считать законченным. Теперь полученных знаний вам достаточно, чтобы успешно разбираться в работе самых разнообразных цифровых схем, которые вам могут встретиться в литературе или Интернете. И даже больше — вы можете смело приступать к проектированию своих собственных устройств. В качестве завершающего примера, подводящего черту под изученным материалом, в последней главе рассматривается процесс разработки макета одного из видов оперативного запоминающего устройства.

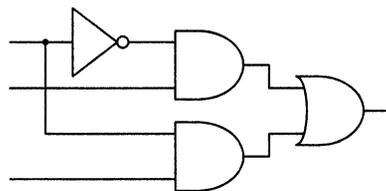
Ответы на вопросы для самопроверки

Номер вопроса	Ответ
1	Сдвиговый регистр будет непрерывно передавать внутри себя по кругу данные, которые появились в нем случайным образом при включении питания либо были заданы извне через асинхронные входы сброса и установки триггеров
2	<p>111 110 101 100 011 010 001 000 111</p>
3	5, 3, 6 и 7
4	Да, нет, да, нет
5	6, 15, 31
6	Синхронный
7	Асинхронный

Ответ на задание по конструкции для самостоятельной разработки



ГЛАВА 8



Макет оперативного запоминающего устройства

Вам потребуются:

- 2 транзистора типа n-МОП;
- 2 светодиода;
- 2 резистора сопротивлением 100 Ом;
- 6 резисторов сопротивлением 10 кОм;
- две кнопки;
- 2 сдвоенных DIP-переключателя;
- 5 микросхем CD4013BE;
- по одной микросхеме CD4555BE, CD4052BE.

Как устроена память цифрового мозга?

Теперь, когда базовые правила разработки и основные виды цифровых схем изучены, можно создать устройство, которое займет довольно большую макетную плату целиком и для которого потребуются все полученные знания. Речь идет об *оперативном запоминающем устройстве* (ОЗУ) или, в просторечии, об *оперативной памяти*. Существует большое количество разных типов ОЗУ. Оно может представлять собой как отдельную планку с микросхемами оперативной памяти, установленную на материнской плате, так и составную часть микропроцессора. ОЗУ может допускать одновременную работу сразу с несколькими разными блоками вычислительной системы или иметь возможность поиска данных внутри себя по заданному образцу. У всех подобных устройств, тем не менее, несмотря на различия, есть масса общих признаков. Изучив их, можно разобраться в работе любого ОЗУ.

Основная его часть — матрица запоминающих ячеек, или матрица памяти. Ее можно представить как таблицу, состоящую из запоминающих ячеек — D-триггеров специфической для памяти конструкции, срабатывающих по высокому уровню синхросигнала. Схемы «ведущий-ведомый» в запоминающих ячейках не используются, чтобы сделать ОЗУ как можно меньше по площади. Минимальная пло-

щадь — самый важный параметр ячеек памяти после помехоустойчивости. Следует отметить, что разновидностей ячеек памяти существует несколько десятков, если не сотен. Это весьма своеобразная область электроники, находящаяся уже на стыке цифровой и аналоговой схемотехники. Все входы синхронизации триггеров в одной строке объединены проводником, называемым *словарной шиной*. Таким образом, строка матрицы — не что иное, как параллельный регистр. Его часто называют *словом данных*, отсюда и название шины. А входы и выходы данных объединяются вертикальными *битовыми, или разрядными, шинами*. Понятно, что они получили свое название из-за того, что каждый из этих проводников проходит через один и тот же бит, или разряд, во всех регистрах. Специфика схем запоминающих ячеек заключается именно в том, что для объединения выходов одного столбца можно использовать всего одну битовую шину. Если применять стандартный D-триггер, то в каждом столбце потребуется столько битовых шин, сколько регистров в матрице, — ведь выходы таких триггеров объединять друг с другом непосредственно одним проводником нельзя, нужно ставить схемы мультиплексирования. А запоминающая ячейка при отсутствии обращений к ней имеет на выходе третье состояние, что и позволяет обойтись всего одним проводником в качестве битовой шины независимо от количества регистров.

Для записи данных в один из регистров матрицы необходимо выставить на битовых шинах нужное двоичное число, а затем подать высокий уровень на словарную шину того регистра, в который производится запись. При чтении данные на битовые шины выставляются не извне матрицы, а из запоминающих ячеек того регистра, словарная шина которого активна. В зависимости от требований, предъявляемых к памяти, используются разные конструкции запоминающих ячеек. Так, например, чтение и запись могут производиться через одни и те же битовые шины. В режиме чтения данные на них выставляются из регистра матрицы. А в режиме записи — одновременно и из запоминающих ячеек, и из регистра входных данных, имеющего настолько мощные выходные каскады, что они «пересиливают» слабый сигнал из ячеек и записывают в них новую информацию. Другой вариант — отдельные битовые шины для чтения и для записи. В этом случае требуются отдельные и словарные шины для каждого из режимов работы памяти. Именно такой тип ОЗУ с регистрами матрицы, имеющими отдельные входы и выходы, мы будем рассматривать в качестве примера. Он называется *регистровым файлом*.

Функциональная схема такого устройства показана на рис. 8.1. Она, в отличие от ранее приведенных в книге принципиальных схем, изображает ОЗУ «крупными мазками», позволяющими не потонуть в деталях, а выделить основные узлы и рассмотреть связи между ними. В центре схемы находится уже известная вам по описанию матрица запоминающих ячеек. Для простоты она содержит всего лишь четыре слова по два бита каждое. Сверху над ней расположен параллельный двухбитный регистр входных данных WD[1:0] (от Write Data, данные для записи), а снизу — такой же регистр выходных RD[1:0] (от Read Data, считанные данные). Тем самым реализуется принцип регистровых передач, о котором шла речь в *главе 6*. Из регистра входных данных к ячейкам идут две битовые шины записи WBL[1:0] (Write Bit Line), а из ячеек выходят битовые шины чтения RBL[1:0] (Read

Bit Line). Чтобы в режиме записи нужный регистр открылся для приема данных с $WBL[1:0]$, на его словарную шину записи WWL (Write Word Line) нужно подать высокий уровень напряжения. А для чтения из регистра нужно выставить единицу на требуемую словарную шину чтения RWL (Read Word Line).

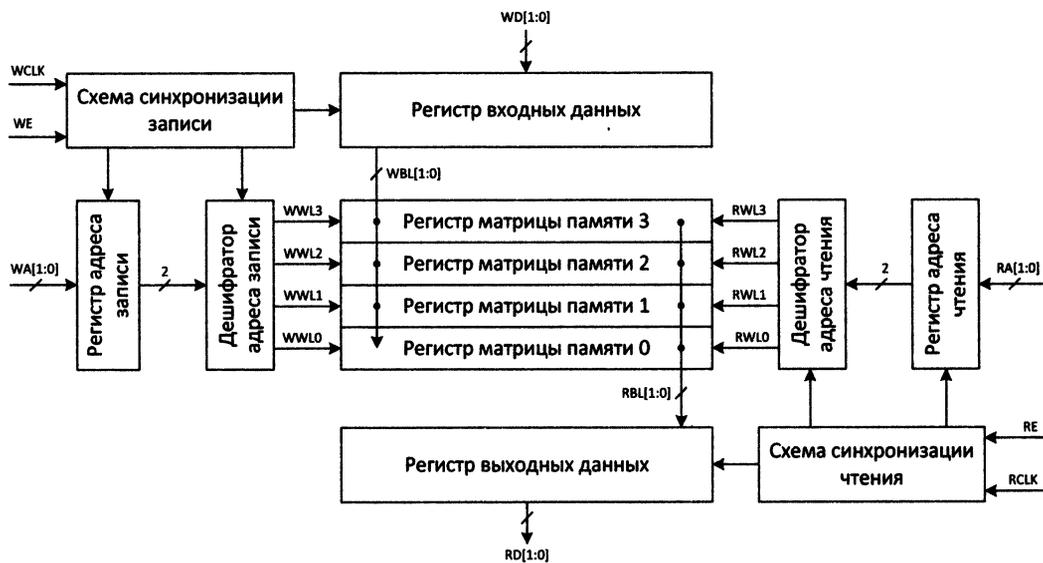


Рис. 8.1. Функциональная схема ОЗУ

Унитарные адреса четырех регистров для уменьшения количества проводов, идущих извне к ОЗУ, можно закодировать двумя битами. В рассматриваемом случае экономия проводов получается несущественной, но на практике используются блоки памяти со значительно большим количеством регистров. И тогда кодирование, например, 256 адресов восьмью битами просто необходимо, иначе невозможно будет использовать ОЗУ на плате или в микросхеме. Это означает, что поступившие извне адреса записи WA (Write Address) и чтения RA (Read Address) требуется дешифровать. Поэтому слева и справа от матрицы на функциональной схеме оказались дешифраторы. А для соблюдения принципа регистровых передач адреса слов ОЗУ должны приниматься на свои регистры.

В углах схемы расположены схемы синхронизации. На них приходят тактовые импульсы чтения $RCLK$ и записи $WCLK$, а также сигналы разрешения записи WE (Write Enable) и чтения RE (Read Enable). Так как эти операции требуется проводить далеко не всякий раз, когда приходит очередной тактовый импульс, то RE и WE сообщают о том, требуется ли от памяти какая-то активность, или она может в текущий момент «отдохнуть». На рис. 8.2 показана типичная схема, выбирающая из непрерывной последовательности синхроимпульсов CLK именно тот, на который указывает сигнал разрешения EN . Для этого используется D-триггер, прозрачный по низкому уровню и защелкивающийся по высокому. Сигнал EN поступает заранее, до защелкивания триггера, и с некоторой задержкой передается на его выход Q . При переходе CLK в высокий уровень следующий за триггером вентиль И

пропускает этот импульс CLK дальше. При переходе синхросигнала в логический ноль то же самое происходит и с CKE. Если к следующему такту CLK сигнал разрешения не был выставлен, то очередной высокий уровень синхросигнала не может попасть внутрь ОЗУ.

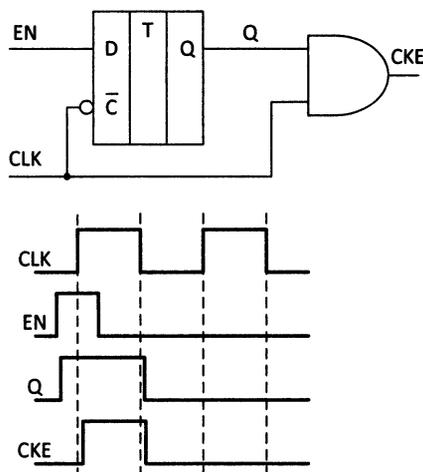


Рис. 8.2. Синхронизация сигнала разрешения

Таким способом из непрерывного потока тактовых импульсов выделяют только нужные и с их помощью вырабатывают сигналы, которые управляют работой всех остальных частей памяти. Так, например, данные на битовые шины записи должны быть выставлены немного раньше сигнала на словарной шине записи, а измениться они могут только после переключения WWL обратно в низкий уровень. При этом длительность WWL должна быть достаточной, чтобы запись прошла успешно. Если какое-либо из этих условий окажется нарушенным, то в регистр памяти будет записано непонятно что. За соблюдение всех временных соотношений между сигналами в ОЗУ и отвечают схемы синхронизации. RCLK и WCLK часто объединяют в один общий CLK так, что запись запускается, например, по его фронту, а чтение — по срезу или наоборот. Набор разрешающих сигналов и их активные уровни также могут отличаться от рассмотренных в этом примере. Здесь все зависит от требований, предъявляемых к конкретному ОЗУ со стороны цифровой системы, в которой они будут применяться.

Теперь нужно обратиться к временной диаграмме всего регистрового файла, показанной на рис. 8.3. В разрабатываемом макете ОЗУ и чтение, и запись происходят по фронту своих тактовых сигналов. Перед началом записи извне должны быть поданы данные для записи $WD[1:0]$, адрес регистра $WA[1:0]$, в который будут записаны эти данные, и активный (высокий в нашем примере) уровень WE. По первому на диаграмме фронту WCLK схема синхронизации обнаруживает $WE = 1$ и запускает работу блоков памяти, отвечающих за запись. Так, двоичное значение $WA[1:0] = 00$ принимается регистром адреса записи и проходит на дешифратор, где преобразуется в унитарный номер 0 регистра в матрице, активируя, т. е. переводя

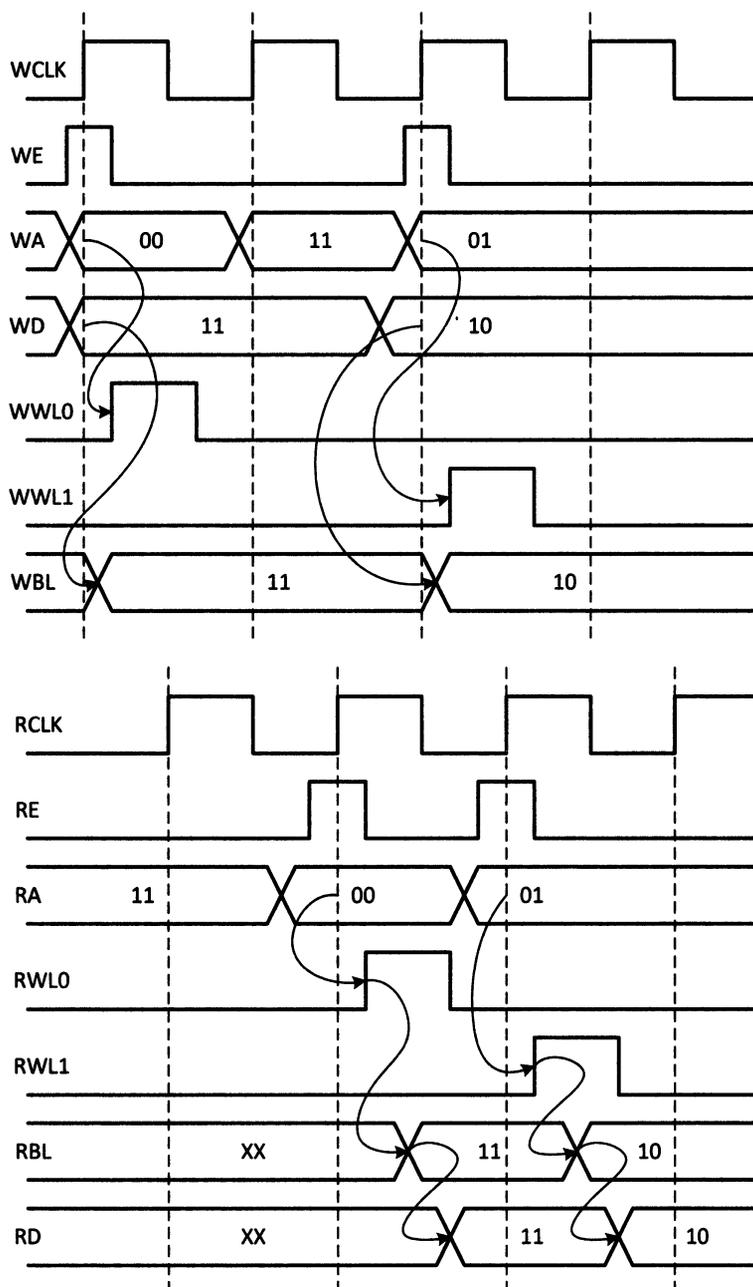


Рис. 8.3. Временная диаграмма O3Y

в высокий уровень, словарную шину записи с этим номером. В это же время данные $WD[1:0] = 11$ поступают в регистр входных данных, усиливаются и передаются на битовые шины записи $WBL[1:0]$. Оба этих процесса для наглядности показаны на временной диаграмме стрелками от вызвавших их сигналов. В то время, когда словарная шина $WWL0 = 1$, запоминающие ячейки регистра с номером 0 принимают

поступившие на их входы по битовым шинам записи значения. Остальные словарные шины записи остаются в низком логическом уровне, поэтому, несмотря на то, что данные с $WBL[1:0]$ приходят на все регистры матрицы памяти одновременно, они их игнорируют и находятся в режиме хранения записанной в них ранее информации. К следующему фронту $WCLK$ на адресный вход $WA[1:0]$ поступило новое значение, но так как в этот момент $WE = 0$, запись не производится. На третьем такте в регистр номер 1 записывается значение 10 аналогично тому, как это происходило в самом начале. Следует отметить, что запоминающие ячейки крайне редко содержат входы сброса или установки, поэтому при включении питания все регистры хранят случайные значения. Это нормальная ситуация для большинства устройств памяти. В программировании обычно считается ошибкой чтение данных из тех регистров ОЗУ, в которые еще ни разу не производилась запись чего-либо осмысленного.

Чтение запускается при $RE = 1$. В первый такт $RCLK$ активного уровня сигнала разрешения чтения нет, поэтому на битовых шинах RBL и на выходе RD остаются ранее присутствовавшие там данные. Так как мы не знаем, какие они были, то на временной диаграмме стоит обозначение XX , принятое для случайной или неизвестной комбинации данных. На втором такте $RCLK$ происходит чтение из слова с номером 0. Последним в него в соответствии с рассмотренной ранее временной диаграммой записи было записано значение 11, оно и оказывается сначала на битовых шинах $RBL[1:0]$, а затем и на выходе ОЗУ $RD[1:0]$. Там эти данные хранятся до следующего такта $RCLK$, во время которого происходит чтение двоичного числа 10 из слова номер 1. Регистр выходных данных в этом примере, в отличие от регистра адреса чтения, срабатывает не по фронту своего синхросигнала, а по срезу. Так делается потому, что в реальных ОЗУ требуется заметное время на то, чтобы дешифратор сформировал сигнал на словарной шине чтения, и маленькие транзисторы запоминающей ячейки передали сигнал по длинным битовым шинам чтения до регистра выходных данных. Зачастую для ускорения процесса чтения сигнал с битовых шин регистрового файла принимается на одноступенчатые триггеры, срабатывающие по уровню. Тогда в тот интервал времени, пока триггер прозрачный, данные из ячейки памяти без лишней задержки передаются на выход, а затем происходит их защелкивание в регистре.

Собираем память на макетной плате

Прежде чем приступить к рассмотрению принципиальной схемы макета регистрового файла и к его сборке на плате, нужно отметить, что полностью реализовать его функциональную схему на микросхемах серии $CD4000BE$ не получится. Во-первых, не существует точного соответствия между некоторыми узлами регистрового файла, изготавливаемыми по интегральной технологии, и DIP-компонентами этой серии. Во-вторых, если бы и было такое соответствие, то необходимое количество корпусов и проводов даже на большой макетной плате просто не удалось бы разместить. Это подобно тому ограничению, с которым вы сталкивались при создании цифровых схем на дискретных транзисторах при переходе к устройствам, содер-

жащим много вентилях. Поэтому неизбежно придется несколько упростить схему, не нарушая при этом принципов ее функционирования, описанных ранее.

Для начала нужно определиться с тем, как с помощью использованных ранее простейших D-триггеров CD4013BE реализовать матрицу запоминающих ячеек с битовыми шинами. Так как в самих триггерах нет третьего состояния, то им на выход нужно добавить элемент, который может это обеспечить. Здесь пригодится упомянутая ранее микросхема CD4052BE, устройство которой показано на рис. 8.4. Как можно видеть, она содержит два четырехканальных мультиплексора-демультиплексора на двунаправленных ключах и общий для них дешифратор 2-в-4. Два дополнительных входа: подавления выходов IN (шестой вывод микросхемы) и аналогового питания VE (седьмой вывод) — в макете ОЗУ не используются и поэтому подключены к минусу питания. Выводы от X0 до X3 подведены к выходам триггеров одного столбца, а от Y0 до Y3 — к триггерам другого.

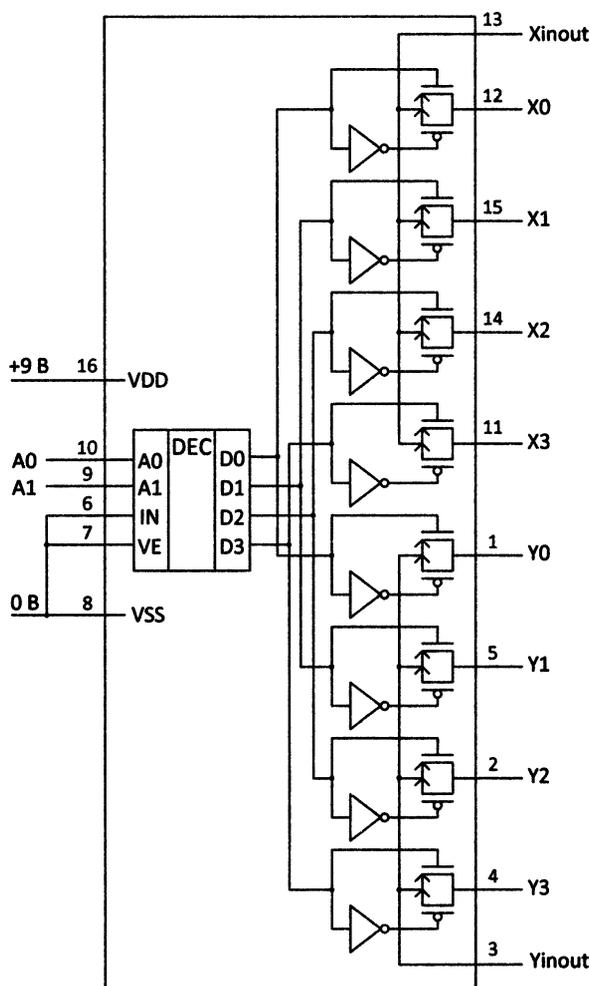


Рис. 8.4. Внутреннее устройство микросхемы мультиплексора CD4052BE

тор, встроенный в микросхему мультиплексора, будет выполнять функцию дешифратора чтения, а общие выходы X_{inout} и Y_{inout} станут битовыми шинами $RBL[1:0]$. То есть в результате такого подхода умозрительная граница между двумя функциональными частями регистрового файла: матрицей запоминающих ячеек и дешифратором адреса чтения — будет проходить не между какими-то двумя микросхемами, а внутри микросхемы CD4052BE. Четыре словарные шины чтения окажутся при этом недоступными для прямого контроля их состояния, и правильность работы можно будет оценить лишь косвенно — по сигналам, появляющимся на битовых шинах чтения. При подаче на вход IN (от inhibition, подавление) высокого логического уровня все выходы встроенного дешифратора оказываются в низком уровне, т. е. все выходы мультиплексора — в третьем состоянии. Но так как это состояние, по сути, является обрывом проводника, а не нулем или единицей, то логическим пробником при наладке схемы его отследить нельзя. Поэтому нет смысла использовать вход IN в качестве сигнала разрешения выхода дешифратора чтения. В результате такого упрощения одна словарная шина чтения в соответствии с поданным адресом всегда будет активна. Так не делается в реальных ОЗУ, но это упрощение не приведет к нарушению логики работы макета, потому что регистр выходных данных будет открыт для приема данных с этих шин только по своему синхросигналу.

В качестве дешифратора записи используется микросхема CD4555BE. Она содержит два дешифратора 2-в-4, из которых задействуется один. От обоих регистров адреса и регистра входных данных на макете можно отказаться по двум причинам. Во-первых, так схема станет намного проще для сборки. Во-вторых, конечно, исключение этих элементов из ОЗУ в реальной микросхеме нарушило бы принцип регистровых передач, но разработанная конструкция будет функционировать сама по себе, без взаимодействия с чем бы то ни было извне. Поэтому и такое упрощение никак не скажется на решении задачи создания действующего макета сложнофункционального блока регистрового файла.

Итак, рассмотрим теперь его принципиальную схему, показанную на рис. 8.5. Переключатели SA1 и SA2 подают на дешифратор DD1 (микросхема CD4555BE) адрес регистра, в который будет производиться запись. Вход разрешения OE у дешифратора активирует по низкому уровню выход в соответствии с заданным адресом. Поэтому кнопка SB1 подключена таким образом, чтобы при отжатом положении на этот вход подавалась единица. Нажатие кнопки моделирует получение дешифратором запускающего импульса от схемы синхронизации при $WE = 1$.

Матрица запоминающих ячеек собрана из четырех микросхем CD4013BE, обозначенных на схеме DD2–DD5. На вход синхронизации каждой из них подается сигнал со своего выхода дешифратора DD1, играющего роль словарной шины записи. Одна из битовых шин записи объединяет все пятые выводы микросхем триггеров, а вторая — все девятые. На схеме, чтобы ее не загромождать, не показаны входы сброса и установки DD2–DD5, но их все необходимо подключить к минусу питания, чтобы они не влияли на работу макета. При этом будет моделироваться поведение реальной памяти: хранение в ней случайных значений при включении питания. Мультиплексор DD6 на микросхеме CD4052BE, как уже обсуждалось, содержит в себе дешифратор чтения и выходные цепи матрицы запоминающих ячеек.

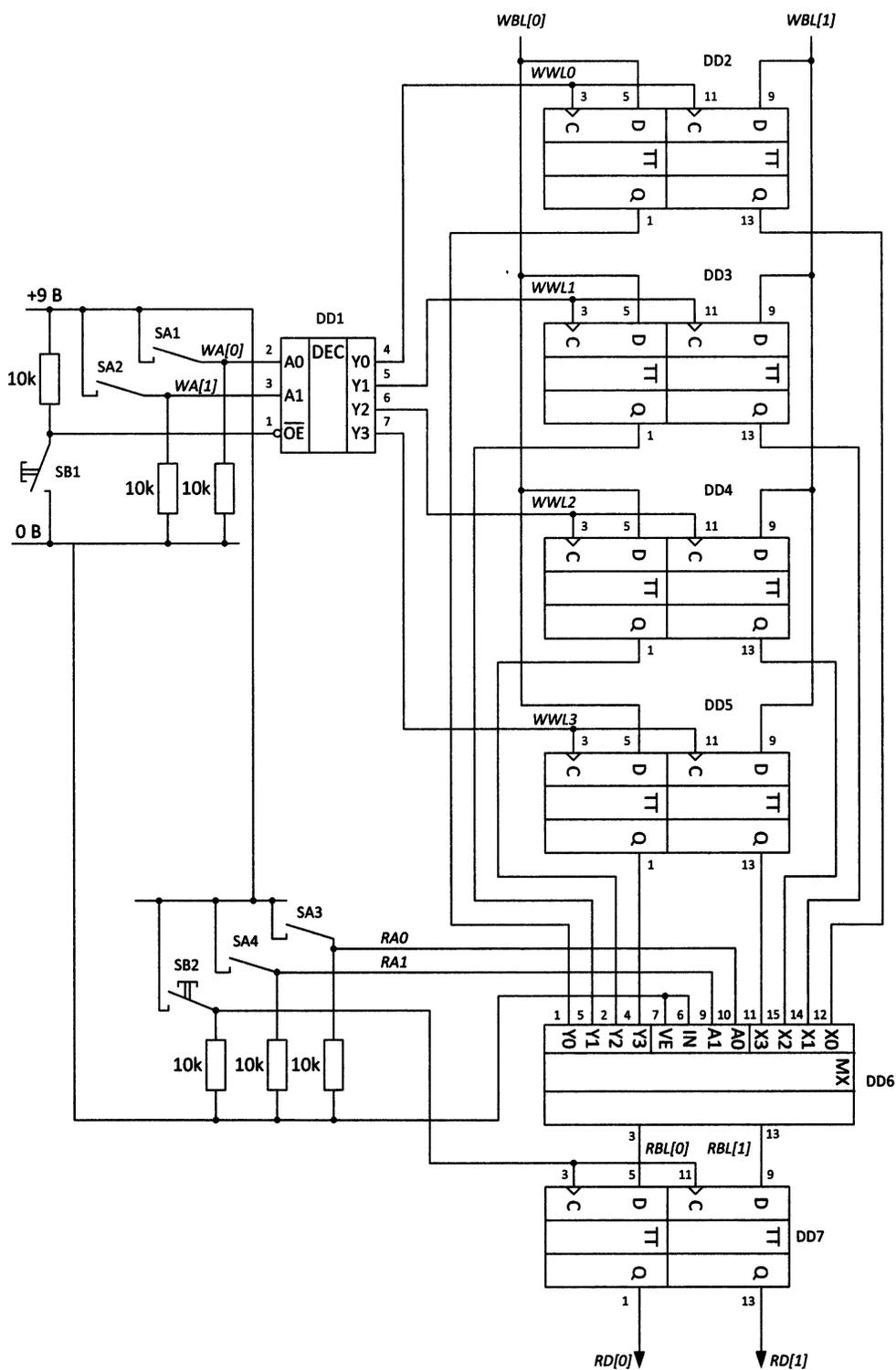


Рис. 8.5. Принципиальная схема ОЗУ

Адрес регистра, из которого производится чтение, задается переключателями SA3 и SA4. Выходы 3 и 13 DD6 образуют битовые шины RBL[1:0], которые подключаются к входам регистра выходных данных DD7 (CD4013BE). По нажатию кнопки SB2 происходит запись сигналов с битовых шин чтения в этот регистр и передача их наружу. То есть кнопка имитирует работу схемы синхронизации чтения.

Сборку и отладку схемы следует начинать с установки DD1, SA1, SA2 и SB1 и резисторов привязки кнопок и переключателей. Проверив, что при удержании кнопки нажатой активным становится именно тот выход дешифратора, который задается адресом записи, можно переходить к сборке матрицы. В каждый регистр следует записать свое двоичное число, т. к. двумя битами как раз можно закодировать четыре разных значения. Тем самым будет проверена полностью работа макета в режиме записи. После этого на плате устанавливается мультиплексор DD6 с переключателями SA3 и SA4 и проверяется состояние его выходов при подаче разных адресов чтения на вход. Последним добавляется регистр выходных данных с кнопкой SB2.

Для сборки монтажной схемы макета (рис. Ц-8.6) использованы две макетные платы типового размера по 800 точек. В верхнем ряду слева направо находятся дешифратор записи DD1, мультиплексор с дешифратором чтения DD6, регистр выходных данных DD7 и подключенные к его выходам пробники логических сигналов. В нижнем ряду располагаются регистры матрицы запоминающих ячеек. Сигналы WBL[1:0] задаются гибкими проводниками, подключаемыми к шинам питания, а адреса чтения и записи — DIP-переключателями. Не забудьте, кстати, подключить шины питания обеих плат к одной и той же батарее!

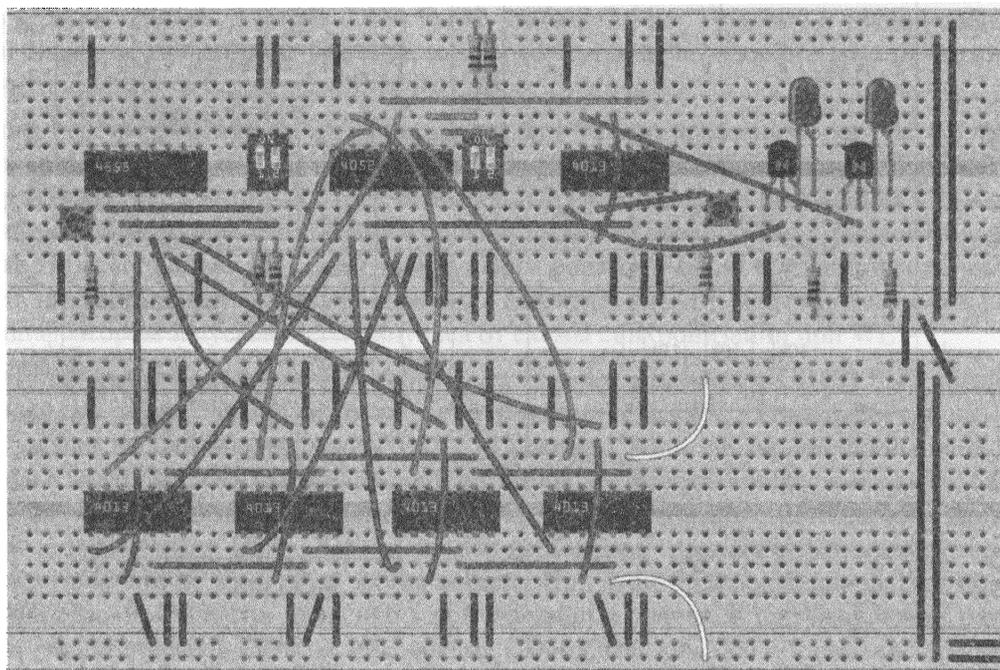


Рис. Ц-8.6. Монтажная схема макета ОЗУ на двух макетных платах

Итак, установив гибкими проводниками данные на битовой шине записи и задав на DIP-переключателях SA1 и SA2 адрес записи, нажатием кнопки SB1 можно отправить требуемое значение в один из регистров. А считываются данные по нажатию кнопки SB2 из того регистра, адрес которого определяется положением переключателей SA3 и SA4. Светодиоды пробников наглядно покажут, что было последним записано в тот или иной регистр матрицы ОЗУ размерностью 4 слова по 2 бита каждое. То есть во все регистры можно записать разные значения и убедиться, что ОЗУ работает правильно, считав их поочередно.

Если не разобраться в принципиальной схеме создаваемого макета ОЗУ и в логике ее работы, то внешний вид монтажной схемы будет казаться бессмысленной мешаниной из микросхем и проводов. Но, разобравшись в работе спроектированного макета ОЗУ, вы можете быть уверенными, что сможете самостоятельно не только повторить его на плате, но и — при желании — внести в него изменения и дополнения. Например, на фото этого же устройства на другой макетной плате (рис. Ц-8.7) та же схема имеет иное подключение компонентов к питанию и друг к другу, но это все тот же макет ОЗУ.

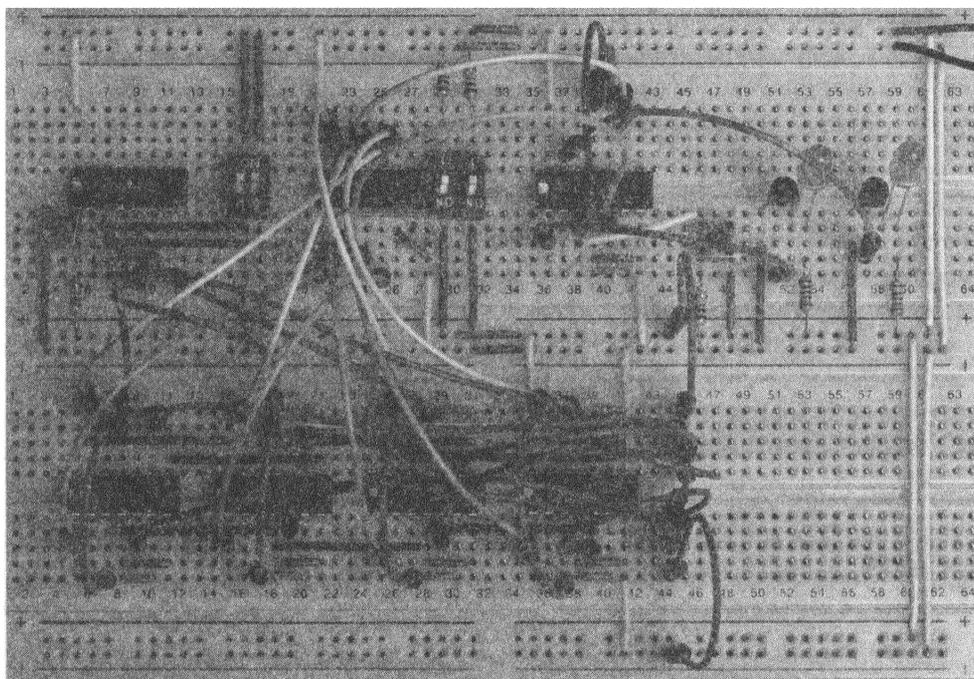


Рис. Ц-8.7. Фото макета ОЗУ, выполненного на одной большой плате

Как видите, для создания такого устройства вам, действительно, пригодились все знания, полученные при чтении этой книги. Более того, конструирование макета ОЗУ показало, что даже в случае отсутствия полностью подходящих компонентов задачу бывает возможно решить каким-то иным способом. Но чтобы видеть другие пути, необходимо обладать широким кругозором. А для этого требуется постоянно совершенствоваться и стремиться узнавать что-то новое. В чем вам и хочется пожелать удачи, упрства и успехов!

Заключение

На этом заканчивается изложение основ цифровой электроники. Вы узнали, из чего состоят микросхемы, как они работают, можно ли одни вентиля заменять другими, как двигаться от идеи к ее практическому воплощению. Освоено много, но это только самое начало пути. Все КМОП-схемы, рассмотренные в книге, относятся к классу, который называется *статической логикой*. Кроме нее существуют еще и динамическая, и дифференциальная логика, и вентиля на проходных транзисторах и двунаправленных ключах, — все это и многое другое тоже применяют весьма широко в цифровых устройствах и интегральных микросхемах.

Можно в дальнейшем двигаться в сторону изучения процесса проектирования и изготовления микросхем, и тогда в будущем вы можете стать разработчиками новых процессоров и целых систем на кристалле. А можно вместо погружения в микромир осваивать конструирование крупных систем робототехники и сверхпроизводительных вычислительных устройств. Обладая хорошим пониманием основ, вы будете лучше и быстрее проектировать сложные схемы. Если после экспериментов с «цифрой» вы решите заняться и аналоговой схемотехникой, то можете быть уверены — у вас получится. Не бойтесь экспериментировать, а некоторые отличные книги, необходимые для изучения этого огромного раздела, вы найдете в списке литературы, приведенном в *приложении 1*.

Электроника не только современная и динамично развивающаяся область техники, но еще обширная и разнообразная. В ней любому найдется занятие по душе. Желаю успехов!

ПРИЛОЖЕНИЯ

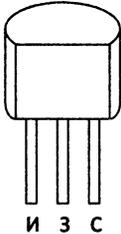
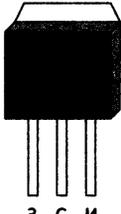
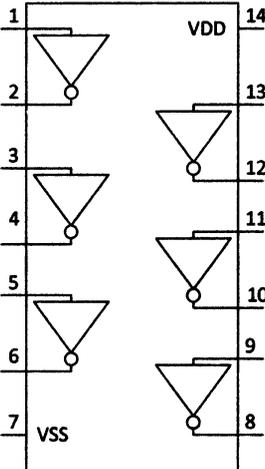
ПРИЛОЖЕНИЕ 1

Список дополнительной литературы для изучения электроники и двоичной логики

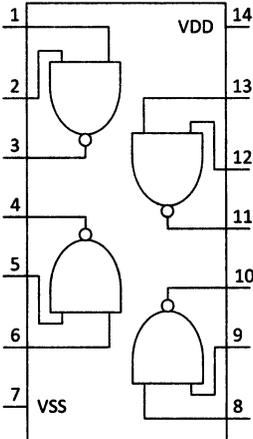
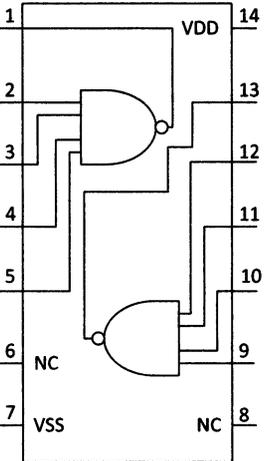
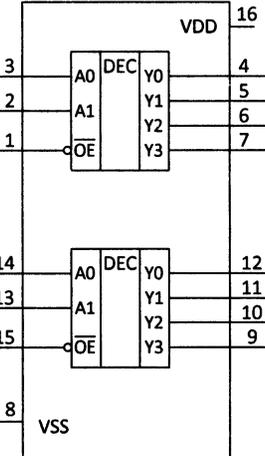
1. Борисов В. Г. Юный радиолюбитель. — 8-е изд., перераб. и доп. — М.: Радио и связь, 1992. — 409 с.
2. Джонс М. Х. Электроника — практический курс. — Пер. с англ. — М.: Постмаркет, 1999. — 528 с.
3. Хоровиц П., Хилл У. Искусство схемотехники. В трех томах. — Пер. с англ. — М.: Мир, 1993.
4. Токхейм Р. Основы цифровой электроники. — Пер. с англ. — М.: Мир, 1988. — 392 с.
5. Андерсон Дж. Дискретная математика и комбинаторика. — Пер. с англ. — М.: Издательский дом «Вильямс», 2004. — 960 с.

ПРИЛОЖЕНИЕ 2

Цоколевки транзисторов и микросхем, применяемых в схемах этой книги

Транзистор, микросхема	Цоколевка
п-канальный транзистор 2N7000TA	 <p>И З С</p>
р-канальный транзистор IRFU9024NPBF	 <p>З С И</p>
Микросхема CD4069UBE	 <p>1 14 VDD</p> <p>2 13</p> <p>3 12</p> <p>4 11</p> <p>5 10</p> <p>6 9</p> <p>7 VSS 8</p>

(продолжение)

Транзистор, микросхема	Цоколевка
<p>Микросхемы CD4001BE, CD4011BE, CD4070BE, CD4071BE, CD4077BE, CD4081BE (изображение вентилей внутри микросхемы приведено для CD4011BE)</p>	
<p>Микросхемы CD4012BE, CD4072BE, CD4082BE (изображение вентилей внутри микросхемы приведено для CD4012BE)</p>	
<p>Микросхема CD4055BE (номера выводов сгруппированы не по порядку их расположения на корпусе, а по функциональному назначению)</p>	

(окончание)

Транзистор, микросхема	Цоколевка
<p>Микросхема CD4013BE (номера выводов сгруппированы не по порядку их расположения на корпусе, а по функциональному назначению)</p>	<p>The diagram shows two CD4013BE flip-flops. The top flip-flop has inputs S (pin 5), D (pin 6), C (pin 3), and R (pin 4). Its outputs are Q (pin 1) and Qn (pin 2). The bottom flip-flop has inputs S (pin 8), D (pin 9), C (pin 11), and R (pin 10). Its outputs are Q (pin 12) and Qn (pin 13). Both flip-flops are connected to VDD (pin 14) and VSS (pin 7).</p>

ПРИЛОЖЕНИЕ 3

Цветовая маркировка резисторов

На постоянные резисторы в соответствии с требованиями российского и зарубежных стандартов наносится маркировка в виде цветных колец (табл. ПЗ.1), кодирующая их номинал (сопротивление). Цветные кольца, если позволяют размеры резистора, сдвинуты к одному из его выводов. Первым кольцом считается то, которое расположено ближе к выводу. Остальные считываются от него слева направо. Если размеры резистора малы для подобного смещения, то первое кольцо делается шире остальных.

Количество колец может меняться от трех до пяти. Если их пять, то первые три — численная величина сопротивления в омах, четвертое — множитель, пятое — допуск, определяющий максимально возможное отклонение действительного сопротивления резистора от номинального. Если колец четыре, то это означает, что есть

Таблица ПЗ.1. Цветовая маркировка резисторов

Цвет кольца	Номинальное сопротивление, Ом			Множитель	Допуск
	Первая полоса	Вторая полоса	Третья полоса		
Серебристый				0,01	±10
Золотистый				0,1	±5
Черный		0	0	1	
Коричневый	1	1	1	10	±1
Красный	2	2	2	100	±2
Оранжевый	3	3	3	1000	
Желтый	4	4	4	10 ⁴	
Зеленый	5	5	5	10 ⁵	±0,5
Голубой	6	6	6	10 ⁶	±0,25
Фиолетовый	7	7	7	10 ⁷	±0,1
Серый	8	8	8	10 ⁸	
Белый	9	9	9	10 ⁹	

незначущий ноль в третьем разряде, который в этом случае не маркируется. В резисторах с большим допуском — до $\pm 20\%$ — он не маркируется никаким цветом, поэтому у них могут быть всего три цветовых кольца.

В проектах этой книги используются всего три номинала резисторов: 100 Ом, 1 и 10 кОм. Они несут кольца следующих цветов:

- первый: коричневого, черного, черного, черного, коричневого. Это соответствует номиналу $100 \cdot 1 = 100$ Ом с допуском $\pm 1\%$;
- второй: коричневого, черного, красного, золотистого. Что дает в результате номинал $10 \cdot 100 = 1$ кОм с допуском $\pm 5\%$;
- третий: коричневого, черного, оранжевого, золотистого. Эта комбинация соответствует номиналу $10 \cdot 1000 = 10$ кОм с допуском $\pm 5\%$.

ПРИЛОЖЕНИЕ 4

Сводная таблица используемых компонентов

Для сборки всех схем из глав 2–7 достаточно стандартной макетной платы на 860 контактных точек. Для сборки макета ОЗУ из главы 8 нужны или две макетные платы на 860 точек, или одна крупная на 1500 точек и более.

Таблица П4.1. Таблица компонентов, необходимых для создания схем, рассмотренных в книге

Позиция	Глава							Итого
	2	3	4	5	6	7	8	
Колодка для батареи 9 В типа «Крона»	1	1	1	1	1	1	1	1
Набор гибких проводов «папа-папа»	1	1	1	1	1	1	1	1
Набор жестких перемычек	1	1	1	1	1	1	1	1
n-канальный транзистор 2N7000TA	2	6	1	4	7	7	2	7
p-канальный транзистор IRFU9024NPBF	1	6	2		6			6
Светодиод	2	2	2	4	2	8	2	8
Резистор 100 Ом	2	2	2	3	2	8	2	8
Резистор 470 Ом				7		7		7
Резистор 1 кОм	2				1	2		2
Резистор 10 кОм	2		5	4		1	6	6
Конденсатор электролитический 1000 мкФ					1	1		1
Конденсатор электролитический 470 мкФ						1		1
Конденсатор электролитический 220 мкФ						1		1
Конденсатор электролитический 100 мкФ						1		1

Таблица П4.1 (окончание)

Позиция	Глава							Итого
	2	3	4	5	6	7	8	
Конденсатор электролитический 10 мкФ						1		1
Кнопка тактовая с двумя выводами	2		1	1		1	2	2
Счетверенный DIP-переключатель			1	1				1
Сдвоенный DIP-переключатель				2			2	2
Соленоид ТАУ-0520			1					1
Микросхема CD4001BE			1		1	1		1
Микросхема CD4011BE			1	1	2	1		2
Микросхема CD4012BE			1		1			1
Микросхема CD4013BE						2	5	5
Микросхема CD4052BE							1	1
Микросхема CD4069UBE			1	1				1
Микросхема CD4070BE				1				1
Микросхема CD4071BE				1				1
Микросхема CB4072BE			1	1				1
Микросхема CD4077BE			1					1
Микросхема CD4081BE			1	1		1		1
Микросхема CD4511BE				1		1		1
Микросхема CD4555BE							1	1
Семисегментный индикатор SC56-11				1		1		1

ПРИЛОЖЕНИЕ 5

Аналоги импортных микросхем серии CD4000BE

*Таблица П5.1. Аналоги микросхем, применяемых в примерах и задачах этой книги,
в отечественной серии КР1561*

Серия CD4000BE	Серия КР1561	Назначение
CD4069UBE	КР1561ЛН1	Шесть логических элементов НЕ
CD4001BE	КР1561ЛЕ5	Четыре логических элемента 2ИЛИ-НЕ
CD4002BE	КР1561ЛЕ6	Два логических элемента 4ИЛИ-НЕ
CD4025BE	КР1561ЛЕ10	Три логических элемента 3ИЛИ-НЕ
CD4011BE	КР1561ЛА7	Четыре логических элемента 2И-НЕ
CD4012BE	КР1561ЛА8	Два логических элемента 4И-НЕ
CD4023BE	КР1561ЛА9	Три логических элемента 3И-НЕ
CD4071BE	Нет	Четыре логических элемента 2ИЛИ
CD4072BE	Нет	Два логических элемента 4ИЛИ
CD4075BE	Нет	Три логических элемента 3ИЛИ
CD4081BE	КР1561ЛИ2	Четыре логических элемента 2ИЕ
CD4082BE	Нет	Два логических элемента 4И
CD4073BE	Нет	Три логических элемента 3И
CD4070BE	КР1561ЛП14	Четыре элемента Исключающее ИЛИ
CD4077BE	нет	Четыре элемента Исключающее ИЛИ-НЕ
CD4053BE	КР1561КП5	Тройной двухканальный мультиплексор/демультиплексор
CD4052BE	КР1561КП1	Два четырехканальных мультиплексора/демультиплексора
CD4555BE	КР1561ИД6	Два дешифратора 2-в-4 с прямыми выходами
CD4556BE	КР1561ИД7	Два дешифратора 2-в-4 с инверсными выходами
CD4013BE	КР1561ТМ2	Два триггера
CD4511BE	Нет	Драйвер семисегментного индикатора

Предметный указатель

А

Арифметико-логическое устройство (АЛУ)
80

Б

Батарея питания 23
Битовая шина 146
Булева алгебра 39

В

Вентиль 38
Временная диаграмма 103

Г

Генератор синхронимпульсов 107

Д

Двоичный вычитатель 79
Двунаправленный ключ 96
Демультимплексор 93
Дешифратор 81
Диод 26

З

Заказное проектирование 68

К

Каскадирование 61
КМОП 39

Комбинационные устройства 73
Конвейер микропроцессорный 124
Конденсатор 108

Л

Логическая единица 16
Логический ноль 16
Логический элемент
◇ алгоритм разработки 47
◇ библиотека 68
◇ И 41
◇ ИЛИ 42
◇ ИЛИ-НЕ 42
◇ инвертор 37
◇ И-НЕ 43
◇ исключающее ИЛИ 44
◇ исключающее ИЛИ-НЕ 44

М

Макетная плата 22
Матрица памяти 145
Микросхема интегральная 57
◇ DIP-корпус 58
◇ серия CD4000BE 58
Младший значащий разряд (МЗР) 13
Мультиплексор 93

О

Область неопределенности 16
Обратная связь 103
Операнд 80
Оперативное запоминающее устройство (ОЗУ) 145

П

Период 108
 Полный сумматор 76
 Полусумматор 76
 Последовательностные устройства 101
 Пробник логических уровней 30

Р

Регистр
 ◇ параллельный 122
 ◇ сдвиговой 127
 Регистровый файл 146
 Резистор 27

С

Светодиод 27
 Семисегментный индикатор 87
 Сигнал
 ◇ аналоговый 14
 ◇ разрешения 70
 ◇ цифровой 14
 ◇ электрический 14
 Синхронизация 106
 Система счисления
 ◇ двоичная 11
 ◇ десятичная 11
 ◇ перевод из двоичной в десятичную 12
 ◇ перевод из десятичной в двоичную 11
 Словарная шина 146
 Соленоид 68
 Срез сигнала 114
 Старший значащий разряд (СЗР) 13
 Схема контроля четности 65
 Схема электрическая
 ◇ монтажная 30
 ◇ принципиальная 28
 ◇ функциональная 146
 Счетчик 132
 ◇ асинхронный 142
 ◇ вычитающий 134
 ◇ по модулю 10 135
 ◇ по модулю 16 132
 ◇ реверсивный 134
 ◇ синхронный 142

Т

Таблица истинности 39
 Тактовый сигнал 106
 Транзистор 19
 ◇ биполярный 21
 ◇ МОП (МДП) 19
 ◇ полевой с р-n-переходом 21
 Третье состояние 97
 Триггер 102
 ◇ D-типа 113
 ◇ JK-типа 110
 ◇ RS-типа 102
 ◇ T-типа 118
 ◇ асинхронный 106
 ◇ ведущий-ведомый 115
 ◇ запрещенное состояние 103
 ◇ переключаемый режим 110
 ◇ синхронный 106



Умножитель 80
 Унитарный код 81

Фронт сигнала 114

Ч

Частота 118

Ш

Шина
 ◇ данных 79
 ◇ питания 24
 Шифратор 81



www.bhv.ru

Отдел оптовых поставок

E-mail: opt@bhv.spb.su

Платт Ч.

Электроника для начинающих, 2-е изд.

Мировой бестселлер для начинающих радиолюбителей!



Создание работающих устройств, от простых до сложных

В своей книге автор исходит из того, что вы приступаете к процессу изучения, не имея каких-либо предварительных знаний в области электроники. Поэтому первые эксперименты очень простые, и вам даже не придется использовать паяльник или плату для монтажа схем, а вы будете соединять провода с помощью «крокодилов». Но очень скоро вы начнете выполнять эксперименты с транзисторами и интегральными микросхемами.

Пошаговые инструкции и более 500 фотографий и рисунков помогут вам легко освоить излагаемый материал.

Новое во втором издании!

- Существенно переработан текст, добавлены новые проекты и внесены изменения в старые.
- Переработан состав компонентов для выполнения экспериментов с целью уменьшения их количества и цены.
- Включены проекты с использованием платы Arduino.

Прочитав книгу, вы сможете:

- открыть для себя новый удивительный мир, извлекая уроки из удачных и неудачных экспериментов с электронными компонентами;
- создать дома рабочее место, удобное для выполнения проектов и оснащенное приборами и инструментами;
- узнать больше об основных электронных компонентах и функциях, которые они выполняют в электрической цепи;
- сделать охранную сигнализацию для защиты от проникновения в дом, елочные огни, электронные украшения для одежды, устройство преобразования звука, тестер измерения скорости реакции человека и кодовый замок;
- получить ясное, доступное для понимания объяснение того, что вы делаете и зачем.

Первым проектом Чарльза Платта был телефонный автоответчик, который он сделал в возрасте 15 лет. Впоследствии он был писателем-фантастом, преподавал компьютерную графику и работал ведущим автором в журнале Wired, но всю жизнь сохранял свою любовь к электронике — главному хобби своей жизни. В настоящее время Чарльз работает редактором в журнале MAKE и занимается разработкой новых образцов медицинского оборудования.



www.bhv.ru

Ревич Ю.

Занимательная электроника, 5-е изд.

Отдел оптовых поставок

E-mail: opt@bhv.spb.su

Электроника — это искусство, которое основано на нескольких основных законах и включает в себя большое количество практических правил и приемов.

П. Хоровиц, У. Хилл



- Начала начал электроники
- Оборудуем домашнюю лабораторию
- Резисторы, конденсаторы, транзисторы, микросхемы...
- Аналоговые и логические схемы
- Arduino — современная электроника для чайников
- Разнообразие применений Arduino на практике

За последние несколько десятков лет в мире произошла революция. Мы и не успели опомниться, как компьютеры, мобильные телефоны, «умные» электронные игрушки стали частью нашего быта. В том, что эти чаяния фантастов прошлого стали явью, главную роль сыграла электроника — искусство конструирования и производства электронных схем. Автор вводит читателя в увлекательный мир электроники и схемотехники, на простых примерах показывая, что изобретать и изготавливать электронные приборы самостоятельно — вовсе не такая сложная наука, как кажется. Основной упор при изложении материала делается на творческий подход — так, чтобы читатель всегда знал, как можно улучшить ту или иную конструкцию или включить ее в свой собственный проект. В новом издании существенно расширен раздел про основу современной любительской электроники — платформу Arduino. Книга рассчитана на старших школьников, учащихся техникумов, студентов и широкий круг любителей электроники.

Ревич Юрий Всеволодович — инженер-электронщик с многолетним стажем. Занимался автоматизацией производств, конструированием измерительных приборов для изучения океана и других научных исследований. В настоящее время журналист. Основной круг интересов — информационные технологии, их влияние на современное общество, история компьютеров и электронных приборов. Имеет несколько сотен публикаций в журналах, газетах и сетевых изданиях. Автор 10 книг, среди которых «Практическое программирование микроконтроллеров Atmel AVR на языке ассемблера», «Информационные технологии в СССР. Создатели советской вычислительной техники» и др.