

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 Капальгина</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн обложки	<i>Марины Дамбиевой</i>

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

ISBN 978-5-9775-4010-0

© ООО "БХВ", 2019
© Оформление. ООО "БХВ-Петербург", 2019

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 вентилях 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
Общезнание 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

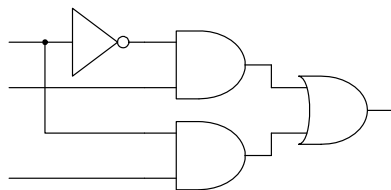
Автор выражает благодарность своей семье: родителям, жене и сыновьям — тем, кто всегда вдохновлял его на поиски нового, поддерживал во всех начинаниях и верил в успех. Без их заботы и терпения ничего бы не получилось.

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 либо 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:

- $10 : 2 = 5$ с остатком 0;
- $5 : 2 = 2$ 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 терморезистора 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

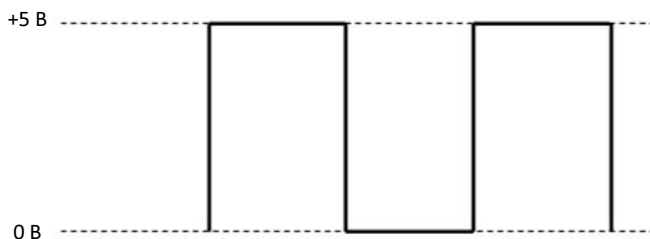


Рис. 1.2. Цифровой сигнал в теории

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.

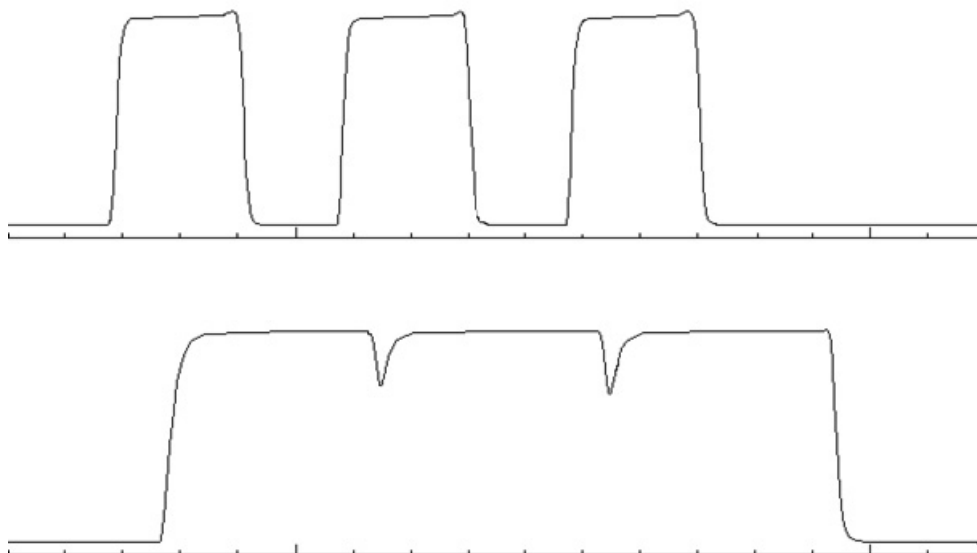


Рис. 1.3. Цифровые сигналы на практике

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 трапециевидную 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-

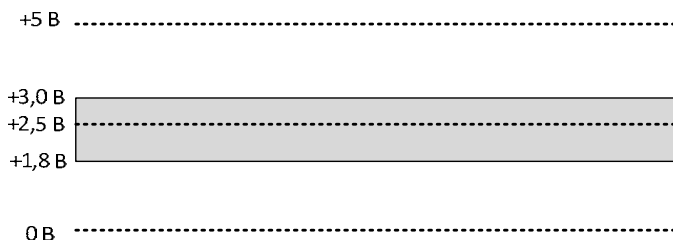


Рис. 1.4. Области нуля, единицы и неопределенности

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?

* * *

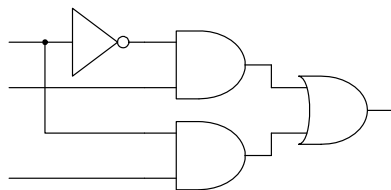
Цифровые схемы в силу того, что в реальности всегда существует некоторый дисбаланс характеристик элементов схемы и проводников, не могут сколь угодно долго находиться в области неопределенности. Обязательно произойдет случайное «сваливание» в ту или другую сторону, которое может вызвать волну переключений 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.

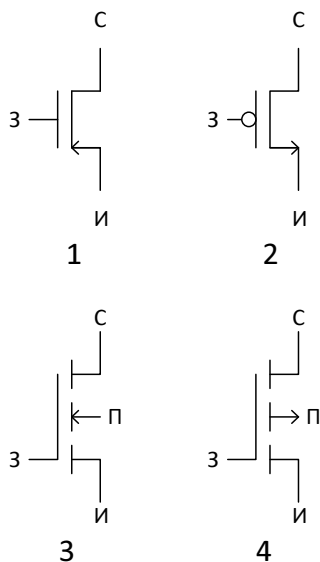


Рис. 2.1. 1 и 3 — n-канальный МОП-транзистор; 2 и 4 — p-канальный МОП-транзистор

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 «n-канальный» и «p-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 n-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 p-n-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!

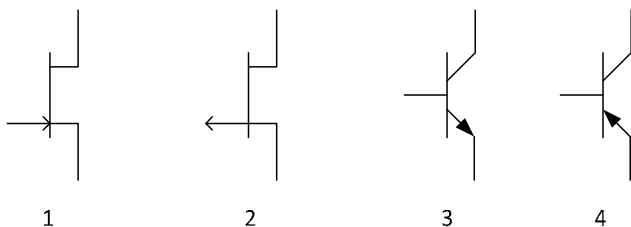


Рис. 2.2. Другие типы транзисторов: 1 — n-канальный полевой транзистор с p-n-переходом; 2 — p-канальный полевой транзистор с p-n-переходом; 3 — npn-биполярный транзистор; 4 — pnp-биполярный транзистор

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 потребуется 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,

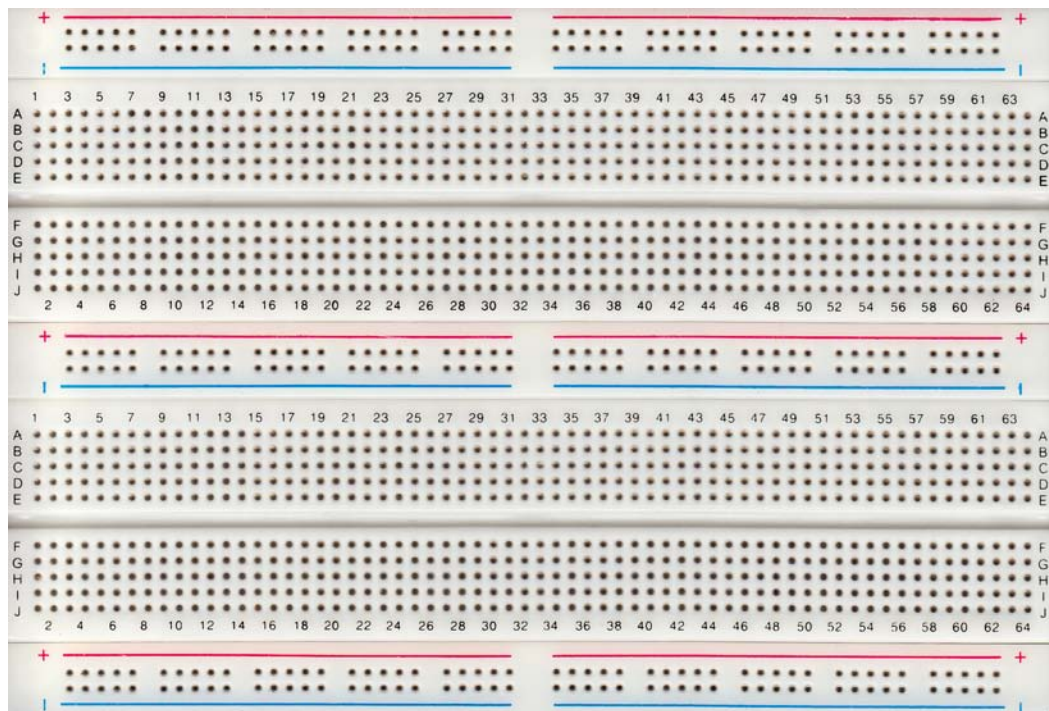


Рис. Ц-2.3. Макетная плата: вид сверху

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 переключки 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 переключки (а); гибкие переключки (б); колодка для батарейки типа «Крона» (в)