

ЗаНуматеЛьНо о МИКРОКОНТРОЛЛЕРАХ

Описание
внутреннего устройства
микропроцессоров

Разработка схем
на микроконтроллерах

Разработка и отладка программ
для микроконтроллеров
на языках Си и ассемблере

Александр Микушин



УДК 681.3.06
ББК 32.973-04
М59

Микушин А. В.

М59 Занимательно о микроконтроллерах. — СПб.:
БХВ-Петербург, 2006. — 432 с.: ил.

ISBN 5-94157-571-8

Собраны материалы, затрагивающие различные аспекты проектирования микропроцессорной техники: от сведений о простейших логических элементах до изложения принципов разработки микропроцессорных систем и достаточно сложных многомодульных программ для них. Из всего многообразия микропроцессоров в качестве примера рассматриваются "классические", доступные и распространенные микроконтроллеры семейства MCS-51, поддерживаемые такими крупнейшими производителями, как Analog Devices и Texas Instruments. Большое внимание уделено построению структуры, принципам написания и отладке программ для микроконтроллеров на языках Си и ассемблере. Приведены готовые шаблоны для написания программ на ассемблере и показаны особенности применения языка Си для реализации конкретных устройств. При подготовке книги были использованы материалы лекций по микропроцессорам, читаемых автором в течение ряда лет в Сибирском государственном университете телекоммуникаций и информатики.

Для широкого круга радиолюбителей и программистов

УДК 681.3.06
ББК 32.973-04

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Игорь Шишигин</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Владимир Харитонов</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 31.05.06.

Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 34,83.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04
от 11.11.2004 г. выдано Федеральной службой по надзору
в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-571-8

© Микушин А. В., 2006
© Оформление, издательство "БХВ-Петербург", 2006

Оглавление

Введение.....	1
Как пользоваться книгой	2
Глава 1. Что такое микроконтроллеры, микропроцессоры и сигнальные процессоры	3
Классификация микропроцессоров.....	3
Универсальные процессоры.....	7
Микроконтроллеры	10
Сигнальные процессоры	12
Итак, подведем итоги.....	14
Глава 2. Цифровая техника	15
Простейшие логические элементы	16
Принципы реализации цифровых устройств по произвольной таблице истинности.....	20
Сумматоры.....	23
Дешифраторы.....	28
Мультиплексоры	31
Демультимплексоры	33
Шинные формирователи	34
Итак, подведем итоги.....	39
Глава 3. Запоминающие устройства	41
Постоянные запоминающие устройства.....	41
Триггеры	52
Регистры.....	56
Статические оперативные запоминающие устройства (ОЗУ)	60
Динамические оперативные запоминающие устройства (ОЗУ).....	65
Итак, подведем итоги.....	71

Глава 4. Принципы работы микропроцессора	73
Виды двоичных кодов	73
Беззнаковые двоичные коды	74
Прямые знаковые двоичные коды	74
Знаковые обратные двоичные коды	75
Знаковые дополнительные двоичные коды	76
Представление рациональных чисел в двоичном коде с фиксированной запятой	80
Представление рациональных чисел в двоичном коде с плавающей запятой	81
Представление десятичных чисел	84
Суммирование двоично-десятичных чисел	85
Представление текстовых данных в памяти процессора	86
Арифметико-логические устройства	86
Классификация микропроцессоров	91
Операционный блок микропроцессора	94
Блок микропрограммного управления	97
Команды микропроцессора	98
Микропрограммирование	101
Итак, подведем итоги	105
 Глава 5. Принципы работы микропроцессорной системы	 107
Системная шина	108
Адресное пространство микропроцессорного устройства	109
Способы расширения адресного пространства микропроцессора	112
Согласование быстродействия памяти и универсальных микропроцессоров	117
Подключение внешних устройств к микропроцессору	119
Принципы построения параллельного порта	123
Принципы построения последовательного порта	127
Синхронные последовательные порты	127
Асинхронные последовательные порты	131
Принципы построения таймеров	134
Итак, подведем итоги	138
 Глава 6. Принципы работы микроконтроллеров	 139
Семейство микроконтроллеров MCS-51	140
Архитектура микроконтроллеров MCS-51	142

Система команд микроконтроллеров MCS-51	147
Арифметические команды.....	148
Логические команды с байтовыми переменными.....	149
Команды пересылки данных	149
Битовые команды	150
Команды ветвления и передачи управления	151
Способы адресации операндов	154
Устройство параллельных портов микроконтроллеров MCS-51.....	156
Особенности построения памяти микроконтроллеров семейства MCS-51.....	164
Память программ микроконтроллеров MCS-51.....	165
Внешняя память данных микроконтроллеров MCS-51	167
Внутренняя память данных микроконтроллеров MCS-51	169
Регистры специальных функций.....	172
Внутренние таймеры микроконтроллера, особенности их применения	173
Режим 0	175
Режим 1	176
Режим 2	178
Режим 3	179
Управление таймерами/счетчиками	180
Использование таймера в качестве измерителя длительности импульсов	182
Использование таймера в качестве частотомера.....	183
Последовательный порт микроконтроллеров семейства MCS-51	185
Скорость приема/передачи информации через последовательный порт.....	187
Режим 0. Синхронный режим работы последовательного порта.....	189
Режим 1. Асинхронный 8-битовый режим	192
Режим 2. Асинхронный 9-битовый режим с фиксированной скоростью передачи.....	197
Режим 3. Асинхронный 9-битовый режим	199
Итак, подведем итоги.....	200
 Глава 7. Принципы создания программ для микроконтроллеров	201
Языки программирования для микроконтроллеров.....	202
Виды программ-трансляторов.....	204
Виды компиляторов.....	205
Применение подпрограмм	206
Стек, его организация и структура.....	208

Подпрограммы-процедуры и подпрограммы-функции	210
Применение комментариев	212
Структурное программирование.....	214
Линейная цепочка операторов.....	217
Условное выполнение операторов.....	220
Конструкция управления циклическим выполнением оператора с проверкой условия после тела цикла	225
Структурная конструкция циклического выполнения оператора с проверкой условия до тела цикла.....	228
Понятие многофайлового и многомодульного программирования	230
Многофайловые программы.....	231
Многомодульные программы.....	237
Программа-монитор	241
Использование таймера для организации параллельных программных потоков	258
Использование прерываний для ввода информации о кратковременных сигналах и событиях, наступающих в произвольный момент времени	261
Итак, подведем итоги.....	265
Глава 8. Язык программирования ASM-51	267
Исходный текст программы на языке программирования ASM-51	269
Символы языка ASM-51	271
Идентификаторы	272
Ключевые слова	273
Встроенные имена	274
Определяемые имена	274
Числа и литеральные строки.....	275
Директивы языка программирования ASM-51.....	277
Управляющие команды.....	283
Реализация подпрограмм на языке ASM-51	285
Реализация подпрограмм-процедур на языке ASM-51.....	285
Передача переменных-параметров в подпрограмму.....	286
Реализация подпрограмм-функций на языке ASM-51	289
Реализация подпрограмм обработки прерываний на языке ASM-51	290
Структурное программирование на языке ASM-51	293
Многомодульные программы	298
Использование сегментов в языке программирования ассемблер.....	301
Итак, подведем итоги.....	308

Глава 9. Язык программирования C-51	309
Структура программ C-51	314
Элементы языка C-51	317
Используемые символы алфавита	317
Лексические единицы, разделители и использование пробелов	320
Идентификаторы	321
Ключевые слова	323
Константы	323
Использование комментариев в тексте программы	327
Типы данных языка программирования C-51 и их объявление	328
Категории типов данных	331
Целочисленный тип данных	332
Числа с плавающей запятой	333
Переменные перечислимого типа	333
Указатели	336
Указатели общего вида	337
Специализированные указатели	339
Массивы	340
Структуры	342
Битовые поля	344
Объединения (смеси)	345
Определение типов	346
Инициализация данных	347
Выражения	349
Операнды и операции	350
Преобразования типов при вычислении выражений	357
Операции унарного минуса, логического и поразрядного отрицания	358
Операции разадресации и вычисления адреса	359
Операция <i>sizeof</i>	360
Мультипликативные операции	360
Аддитивные операции	361
Операции сдвига	363
Поразрядные операции	363
Логические операции	364
Операция последовательного вычисления	365
Условная операция	365
Операции инкремента и декремента	366
Простое присваивание	367

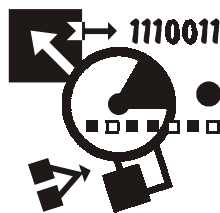
Составное присваивание	367
Приоритеты операций и порядок вычислений	368
Побочные эффекты	368
Преобразование типов	369
Операторы	372
Оператор-выражение	372
Пустой оператор	373
Составной оператор	373
Оператор <i>if</i>	374
Оператор <i>switch</i>	376
Оператор <i>break</i>	379
Оператор цикла <i>for</i>	379
Оператор цикла <i>while</i>	381
Оператор цикла <i>do-while</i>	382
Оператор <i>continue</i>	383
Оператор возврата из функции <i>return</i>	383
Оператор безусловного перехода <i>goto</i>	384
Использование функций в языке программирования С-51	385
Определение и вызов функций	385
Итак, подведем итоги	393

Приложение. Справочные данные по системе команд микроконтроллера MCS-51 и кодировке символов	395
---	------------

Литература	418
-------------------------	------------

Предметный указатель	420
-----------------------------------	------------

Глава 2



Цифровая техника

Итак, рассмотрев какие виды микропроцессоров бывают и для решения каких видов задач они применяются, можно приступить к решению вопроса — как же они устроены? Как уже говорилось, микропроцессорная техника является частью цифровой техники. Поэтому, не зная основ цифровой техники, невозможно понять, как работает микропроцессор.

Начнем с самых элементарных вопросов: из каких элементов строятся цифровые схемы и как они устроены? Затем научимся реализовывать на основе этих простейших элементов цифровые устройства любой сложности. Следует отметить, что в данной книге вы не получите обзора всего разнообразия цифровых устройств. Будут рассмотрены только те цифровые устройства, которые используются непосредственно в микропроцессорной технике.

В данной главе будут рассмотрены только сумматоры и устройства коммутации цифровых сигналов. Однако, при необходимости, можно и разобраться в оставшихся за рамками данной книги разделах цифровой техники, применив рассмотренные методы построения цифровых устройств.

Обычно любые устройства предназначены для преобразования входных сигналов в выходные. Свойства аналоговых схем описываются рядом общепринятых параметров (например, коэффициентом усиления и динамическим диапазоном) и характеристик (амплитудно-частотные и фазо-частотные характеристики).

Иначе обстоит дело с параметрами цифровых микросхем. Для идеализированных цифровых устройств коэффициент усиления не нормируется — он реализуется достаточным для того, чтобы сигнал на выходе не затухал.

И только! Логические уровни на входе и выходе цифровых микросхем одинаковы. Если логический элемент обладает запасом по коэффициенту усиления, то выходной сигнал просто ограничивается. Конкретное зна-

чение логических уровней зависит от напряжения питания цифровых микросхем и примененной схемотехники, но это не меняет принципов работы цифрового устройства.

Цифровые схемы наиболее полно описываются таблицей истинности. Таблица истинности позволяет поставить выходные сигналы в соответствие входным сигналам. Обычно каждый из выходных сигналов цифрового устройства зависит от нескольких входных сигналов этого цифрового устройства. Поэтому в таблице истинности перечисляются все возможные комбинации входных сигналов и записывают соответствующий каждой комбинации входных сигналов выходной сигнал.

Достаточно часто одним и тем же комбинациям входных цифровых сигналов соответствуют несколько выходных сигналов. Тогда для всех выходных сигналов записывается одна таблица истинности.

Для простейших цифровых логических элементов таблица истинности состоит из одного выходного и одного или двух входных сигналов. Рассмотрим эти элементы.

Простейшие логические элементы

Любые цифровые устройства строятся на основе простейших логических элементов: «НЕ», «ИЛИ», «И». Самым простым логическим элементом является инвертор (элемент «НЕ»), который работает в соответствии с табл. 2.1. Он просто изменяет значение входного сигнала на прямо противоположное. В качестве инвертора можно использовать обычный транзисторный усилитель, построенный по схеме с общим эмиттером или общим истоком. Схемы, позволяющие реализовать функцию логического инвертирования, изображены на рис. 2.1. На рис. 2.1, *а* приведена схема инвертора на обычном биполярном транзисторе, а на рис. 2.1, *б* приведена схема инвертора, выполненного на комплементарных МОП-транзисторах.

Таблица 2.1. Таблица истинности логического инвертора

Вход	Выход
0	1
1	0

Условное графическое обозначение инвертора на схемах не зависит от схемотехники, использованной для его реализации, оно приведено на

рис. 2.2. С этого момента инвертор будет изображаться исключительно в таком виде.

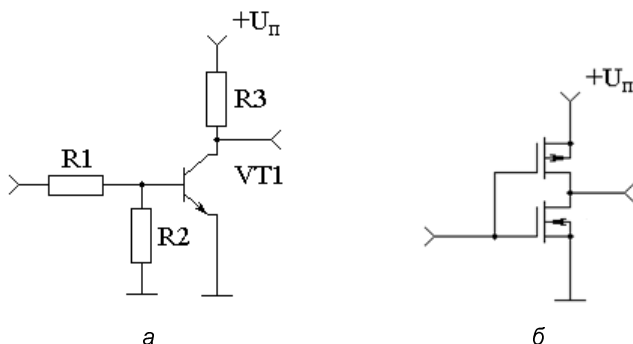


Рис. 2.1. Схемы, реализующие функцию логического инвертирования

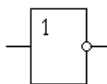


Рис. 2.2. Условное графическое обозначение инвертора

Следующий распространенный элемент цифровой техники реализует логическую операцию «И», однако чаще всего в качестве готовых микросхем существуют не отдельные схемы логического «И», а более сложные устройства, выполняющие одновременно две логические функции: «И» и «НЕ». Таблица истинности элемента, выполняющего логическую функцию «2И-НЕ», приведена в табл. 2.2.

Проще всего построить такой элемент на самых обыкновенных ключах, как это показано на рис. 2.3, *а*. В этой схеме ток будет протекать только в том случае, если оба ключа окажутся замкнутыми (будем считать, что такое их состояние достигается при управлении логической единицей). Это означает, что нулевой уровень на выходе схемы появится только при двух логических единицах на входе, т. е. приведенная схема реализует логическую функцию «2И-НЕ» (табл. 2.2). Точно таким же образом выполняется элемент «2И-НЕ» и в микросхемах, построенных на КМОП-транзисторах, только в качестве ключа используется транзистор. Схема

логического элемента «2И-НЕ», выполненного на комплементарных МОП-транзисторах, приведена на рис. 2.3, б.

Таблица 2.2. Таблица истинности схемы, выполняющей логическую функцию «2И-НЕ»

Вход X	Вход Y	Выход
0	0	1
0	1	1
1	0	1
1	1	0

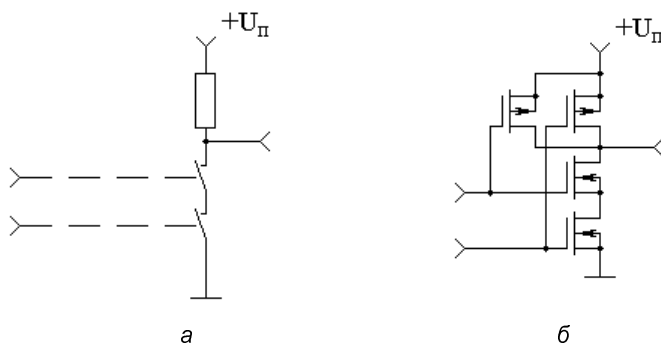


Рис. 2.3. Принципиальные схемы цифровых элементов, реализующих логическую функцию «2И-НЕ»

Условное графическое обозначение элемента, выполняющего логическую функцию «2И-НЕ», приведено на рис. 2.4, и с этого момента элементы, выполняющие данную функцию, будут изображаться именно в таком виде. Это обозначение не зависит от конкретной схемы построения цифрового элемента.

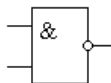


Рис. 2.4. Условное графическое обозначение цифрового элемента, выполняющего логическую функцию «И-НЕ»

Точно так же, как редко можно встретить отдельный элемент логического «И», практически не производятся отдельные элементы логического «ИЛИ». Чаще встречаются элементы «2ИЛИ-НЕ», таблица истинности которых приведена в табл. 2.3.

Таблица 2.3. Таблица истинности цифрового элемента, выполняющего логическую функцию «2ИЛИ-НЕ»

Вход X	Вход Y	Выход
0	0	1
0	1	0
1	0	0
1	1	0

Как и в предыдущем случае, воспользуемся для реализации элемента «2ИЛИ-НЕ» ключами. На этот раз соединим ключи параллельно. Схема, реализующая таблицу истинности табл. 2.3, приведена на рис. 2.5, а. Схема логического элемента «2ИЛИ-НЕ», выполненного на КМОП-транзисторах, показана на рис. 2.5, б. Как видно из приведенных схем, уровень логического нуля появится на выходе любой из этих схем, как только любой из ключей будет замкнут, т. е. приведенные схемы реализуют таблицу истинности табл. 2.3.

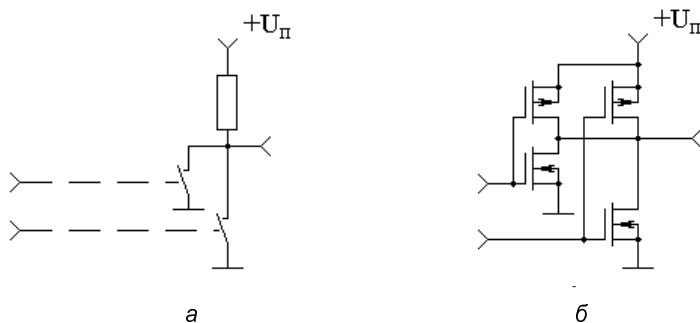


Рис. 2.5. Принципиальные схемы элемента, реализующего логическую функцию «2ИЛИ-НЕ»

Так как один и тот же логический элемент может быть реализован различными способами, для его изображения на схемах используется специальное условное графическое обозначение, приведенное на рис. 2.6.

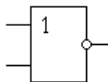


Рис. 2.6. Условное графическое обозначение элемента, выполняющего логическую функцию «2ИЛИ-НЕ»

Принципы реализации цифровых устройств по произвольной таблице истинности

Любое цифровое устройство полностью описывается таблицей истинности. При построении сложных устройств с произвольной таблицей истинности используется сочетание простейших элементов: «И», «ИЛИ», «НЕ». Если устройство имеет несколько выходов, то формирование сигнала для каждого из них анализируется отдельно и для каждого из них строится отдельная схема.

Для реализации устройства можно воспользоваться как элементами «И», так и элементами «ИЛИ». В настоящее время наиболее распространены микросхемы, совместимые с ТТЛ, а в ТТЛ проще всего получить элементы «И», выходы которых объединены по функции «ИЛИ», поэтому рассмотрим способ реализации произвольной таблицы истинности, основанный на комбинации логических элементов «И-ИЛИ».

Для реализации таблицы истинности при помощи логических элементов «И» достаточно рассмотреть только те ее строки, которые содержат логические единицы в выходном сигнале. Строки, содержащие в выходном сигнале логический ноль, в построении схемы не участвуют. Каждая строка, содержащая в выходном сигнале логическую единицу, реализуется элементом логического «И» с количеством входов, совпадающим с количеством входных сигналов в таблице истинности.

Таблица 2.4. Произвольная таблица истинности

Входы				Выходы	
In0	In1	In2	In3	Out0	Out1
0	0	0	0	0	0
0	0	0	1	1	0

Таблица 2.4 (окончание)

Входы				Выходы	
0	0	1	0	0	0
0	0	1	1	0	0
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	1	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0
1	1	1	1	0	1

Входные сигналы, описанные в таблице истинности логической единицей, подаются на вход этого элемента непосредственно, а входные сигналы, описанные в таблице истинности логическим нулем, подаются на вход этого же элемента «И» через инверторы. Объединение сигналов с выходов элементов «И», реализующих отдельные строки таблицы истинности, производится при помощи элемента логического «ИЛИ». Количество входов элемента «ИЛИ» определяется количеством строк таблицы истинности, в которых в выходном сигнале присутствует логическая единица.

Для сокращения количества инверторов имеет смысл выделить их в отдельный блок, который сразу сформирует сигналы, инверсные по отношению к входным сигналам цифрового устройства. Теперь для реализации строки таблицы истинности достаточно соединить входы логического элемента «И» с соответствующими инвертированными и неинвертированными входными сигналами.

Рассмотрим конкретный пример. Пусть необходимо реализовать устройство с таблицей истинности, приведенной в табл. 2.4. Для построения схемы, реализующей сигнал Out1, достаточно рассмотреть строки, выделенные жирным шрифтом. Эти строки реализуются микросхемой D2 на

рис. 2.7. Каждая строка реализуется своим многовходовым элементом «И», затем выходы этих элементов объединяются по «ИЛИ». Количество входов элемента «И» однозначно определяется числом входных сигналов в таблице истинности. Количество этих элементов, а значит и входов в логическом элементе «ИЛИ», определяется количеством строк с единичным сигналом на реализуемом выходе цифрового устройства.

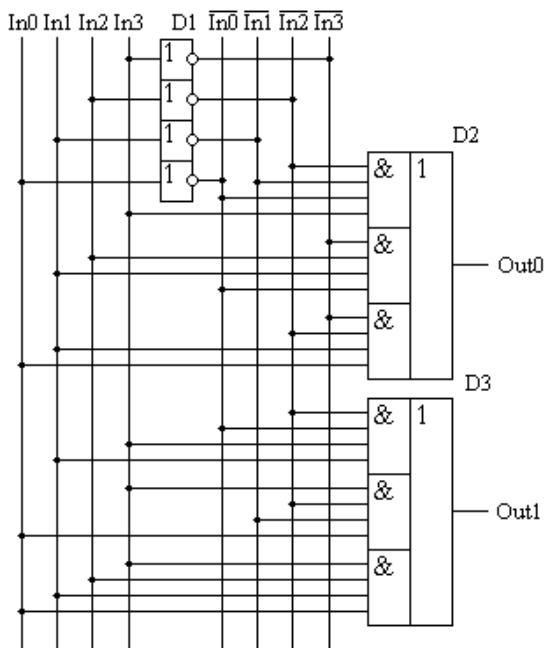


Рис. 2.7. Принципиальная схема устройства, реализующего таблицу истинности, приведенную в табл. 2.4

Для построения схемы, реализующей сигнал Out2, достаточно рассмотреть строки, выделенные курсивом. Соответствующая логическая функция реализуется микросхемой D3. Принцип построения этой схемы такой же, как в примере, рассмотренном выше, и поэтому повторяться не будем.

Обычно при построении цифровых устройств после реализации таблицы истинности производится минимизация схемы, но для упрощения изложения материала в этой книге она выполняться не будет. Отказ от минимизации оправдан еще и тем, что неминимизированные схемы обычно обладают максимальным быстродействием.

Сумматоры

Важным элементом цифровых устройств, выполняющих арифметическую обработку цифровой информации, является сумматор. Построение двоичных сумматоров обычно начинается с сумматора по модулю 2.

В табл. 2.5 приведена таблица истинности этого сумматора. Ее можно получить, исходя из правил суммирования в двоичной арифметике. Предполагается, что читатель знаком с основами двоичной арифметики. Более подробно операции над двоичными числами будут рассмотрены позднее.

Таблица 2.5. Таблица истинности сумматора по модулю 2

Вход X	Вход Y	Выход
0	0	0
0	1	1
1	0	1
1	1	0

В соответствии с принципами построения произвольной таблицы истинности, рассмотренными в предыдущей главе, получим схему сумматора по модулю 2. Эта схема приведена на рис. 2.8.

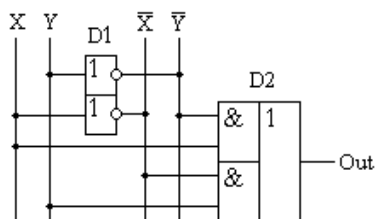


Рис. 2.8. Принципиальная схема устройства, реализующего таблицу истинности сумматора по модулю 2

Сумматор по модулю 2 (для двоичной арифметики его функцию реализует элемент исключающего «ИЛИ») изображается на схемах с использованием условного графического обозначения, показанного на рис. 2.9.

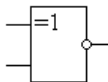


Рис. 2.9. Условное графическое обозначение элемента, выполняющего логическую функцию исключающего «ИЛИ»

Сумматор по модулю 2 выполняет суммирование без учета переноса. В полном двоичном сумматоре его необходимо учитывать, поэтому требуются элементы, позволяющие формировать перенос в следующий двоичный разряд. Таблица истинности такого устройства, называемого полусумматором, приведена в табл. 2.6.

Обратите внимание, что сигналы в приведенной таблице истинности расположены в порядке, принятом для схем, т. е. в соответствии с тем, что сигнал распространяется слева направо. В результате перенос, который имеет двоичный вес, больший по сравнению с суммируемыми разрядами, записан правее. В математике принят другой порядок разрядов числа. Старший разряд на бумаге записывается самым левым, а младший разряд записывается самым правым. В результате может возникнуть путаница. Чтобы этого не произошло, приведу десятичный эквивалент каждой строки таблицы истинности полусумматора (табл. 2.6).

Первая строка получена из выражения $0 + 0 = 0_{10} (00_2)$. Вторая строка получена из выражения $0 + 1 = 1_{10} (01_2)$. Третья строка получена из выражения $1 + 0 = 1_{10} (01_2)$. Четвертая строка получена из выражения $1 + 1 = 2_{10} (10_2)$.

Таблица 2.6. Таблица истинности полусумматора

Вход А	Вход В	Выход S	Выход РО
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

В соответствии с принципами построения произвольной таблицы истинности получим схему полусумматора. Она приведена на рис. 2.10. Условное графическое обозначение полусумматора показано на рис. 2.11.

Полусумматор формирует перенос в следующий разряд, но не может учитывать перенос из предыдущего разряда, поэтому он и называется полусумматором.

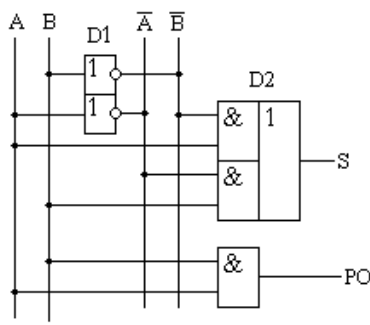


Рис. 2.10. Принципиальная схема цифрового устройства, реализующего таблицу истинности полусумматора

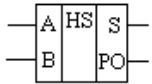


Рис. 2.11. Условное графическое обозначение полусумматора

Таблицу истинности полного двоичного одноразрядного сумматора (табл. 2.7) можно получить из правил суммирования двоичных чисел. В обозначении входов и выходов полного сумматора использовано следующее правило: в качестве входов использованы одноразрядные двоичные числа *A* и *B*; сумма — это одноразрядное двоичное число *S*; перенос обозначен буквой *P*; для обозначения входа переноса используется сочетание букв *PI* (*I* — сокращение от английского слова *input*, вход); для обозначения выхода переноса используется сочетание букв *PO* (*O* — сокращение от английского слова *output*, выход).

Таблица 2.7. Таблица истинности полного двоичного одноразрядного сумматора

PI	A	B	S	PO	Математическое выражение
0	0	0	0	0	$0 + 0 + 0 = 0_{10} (00_2)$
0	0	1	1	0	$0 + 0 + 1 = 1_{10} (01_2)$
0	1	0	1	0	$0 + 1 + 0 = 1_{10} (01_2)$
0	1	1	0	1	$0 + 1 + 1 = 2_{10} (10_2)$

Таблица 2.7 (окончание)

PI	A	B	S	PO	Математическое выражение
1	0	0	1	0	$1 + 0 + 0 = 1_{10} (01_2)$
1	0	1	0	1	$1 + 0 + 1 = 2_{10} (10_2)$
1	1	0	0	1	$1 + 1 + 0 = 2_{10} (10_2)$
1	1	1	1	1	$1 + 1 + 1 = 3_{10} (11_2)$

В соответствии с правилами построения принципиальной схемы по произвольной таблице истинности получим схему полного двоичного одноразрядного сумматора. Она приведена на рис. 2.12.

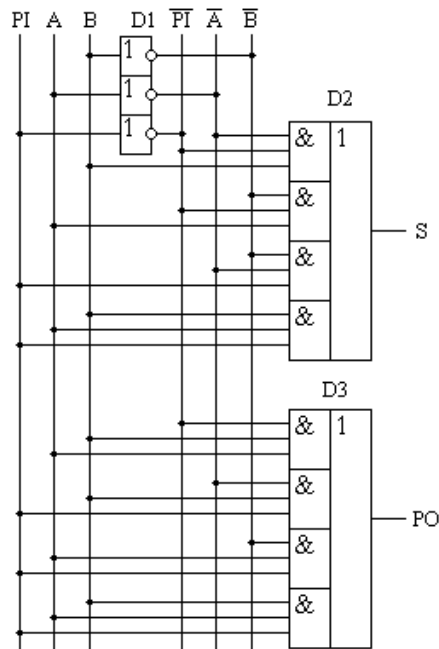


Рис. 2.12. Принципиальная схема цифрового устройства, реализующая функцию полного двоичного одноразрядного сумматора

Ее можно минимизировать, но, как уже оговаривалось, минимизация в данной книге рассматриваться не будет. Условное графическое обо-