

СОДЕРЖАНИЕ

Введение	6
----------------	---

1 Внутренние ресурсы ОВМ семейства x51	8
1.1. Общее описание ОВМ x51	9
1.1.1. Внутренняя структура ОВМ	9
1.1.2. Процессорное ядро	9
1.1.3. Внутренняя периферия	10
1.1.4. Условное графическое обозначение	10
1.1.5. Служебные выводы общего назначения	12
1.1.6. Выводы для обслуживания внешней памяти	13
1.1.7. Альтернативные функции порта P3	14
1.2. Особенности выходных цепей портов	15
1.2.1. Выходные цепи порта P0	15
1.2.2. Выходные цепи портов P1, P2, P3	16
1.2.3. Особенности маловыводного варианта x51	17
1.2.4. Подключение нагрузки к выводу порта	18
1.2.5. Внешний ключ на p-p-p-транзисторе	20
1.2.6. Внешний ключ на p-p-p-транзисторе	22
1.3. Работа ОВМ и структура памяти	22
1.3.1. Общие сведения	22
1.3.2. Структура памяти команд (ПЗУ)	23
1.3.3. Способы занесения кодов во внутреннее ПЗУ	24
1.3.4. Структура памяти данных	25
1.3.5. Навигация по ячейкам памяти	26
1.4. Спецрегистры ОВМ x51	27
1.4.1. Аккумулятор	27
1.4.2. Регистр признаков	28
1.4.3. «Нефиксируемые» признаки результата	29
1.4.4. Особые биты регистра признаков	29
1.4.5. Регистры-указатели	30
1.4.6. Спецрегистры внутренней периферии	30
1.5. Память данных – регистры и операнды	31
1.5.1. Структура основного ОЗУ	31
1.5.2. Определение операндов пользователя	32
1.5.3. Использование зарегистрированных имен	33
1.6. Методы адресации операндов	34
1.6.1. Непосредственная адресация	34
1.6.2. Регистровая адресация	34
1.6.3. Прямая адресация	35
1.6.4. Косвенная адресация	35
1.6.5. Относительная адресация	36

2 Перечень команд ОВМ x51	37
2.1. Особенности ассемблерных команд	38
2.1.1. Мнемокоды команд	38
2.1.2. Особенности записи мнемокода команды	38

2.2. Команды пересылки данных.....	39
2.2.1. Общий вид команд пересылки.....	39
2.2.2. Регистровые пересылки	40
2.2.3. Пересылки с косвенно-регистровой адресацией.....	41
2.2.4. Пересылки с прямой адресацией	42
2.2.5. Пересылка (загрузка) констант	43
2.2.6. Загрузка констант из ПЗУ с адресацией по сумме регистров.....	43
2.3. Арифметические команды.....	44
2.3.1. Общие особенности	44
2.3.2. Команды сложения (ADD).....	45
2.3.3. Команды вычитания.....	45
2.3.4. Команда десятичной коррекции аккумулятора.....	46
2.3.5. Команда умножения.....	47
2.3.6. Команда деления.....	47
2.4. Логические команды	47
2.4.1. Команды логического умножения.....	47
2.4.2. Команды логического сложения	48
2.4.3. Команды «исключающего ИЛИ»	48
2.4.4. Команды ротации (сдвига)	49
2.4.5. Команды очистки и инвертирования аккумулятора	49
2.4.6. Команды побитовой обработки	50
2.5. Команды передачи управления	51
2.5.1. Общие сведения	51
2.5.2. Разновидности команд перехода.....	52
2.5.3. Безусловные переходы	52
2.5.4. Условные переходы.....	54
2.5.5. Сдвоенные условные команды перехода	55
2.5.6. Вызовы подпрограмм	56
2.6. Оформление ассемблерной программы	57
2.6.1. Структура ассемблерной программы	57
2.6.2. Определение констант и распределение внешних выводов.....	58
2.6.3. Резервирование ячеек для переменных в памяти данных	59
2.6.4. Заполнение особых ячеек в ПЗУ	59
2.6.5. Текст основной части программы	60
2.6.6. Подпрограмма и макрос	60
2.6.7. Тексты используемых процедур.....	61
2.6.8. Размещение массивов констант в ПЗУ	62
2.6.9. Общие советы	62

3 Управление внутренней аппаратурой ОВМ х51.....	64
3.1. Запись (вывод) информации битовыми командами.....	65
3.1.1. Особенности адресации битов.....	65
3.1.2. Установка, сброс и инвертирование отдельных битов	66
3.1.3. Вывод – копирование групп битов	66
3.1.4. Вывод битов в последовательном формате.....	67
3.2. Чтение (ввод) информации битовыми командами	68
3.2.1. Ввод (чтение) одиночного бита.....	68
3.2.2. Чтение и анализ бита.....	69
3.2.3. Чтение групп битов из разных портов	70
3.2.4. Чтение последовательного формата битов.....	70

3.3. Ввод-вывод информации байтовыми командами	71
3.3.1. Вывод (запись) байтовых констант и переменных	72
3.3.2. Установка групп битов байтовыми командами	72
3.3.3. Сброс групп битов байтовыми командами	74
3.3.4. Инвертирование групп битов байтовыми командами	74
3.3.5. Ввод (чтение) байтовой информации	74
3.3.6. Ввод (чтение) информации групп битов байтовыми командами	75
3.4. Управление прерываниями	75
3.4.1. Общие сведения	75
3.4.2. Регистр разрешения прерываний IE	76
3.4.3. Регистр управления приоритетами IP	76
3.4.4. Управление типом внешних прерываний	77
3.4.5. Запросы прерываний	77
3.5. Пример программы для внешнего прерывания	78
3.5.1. Организация внешнего прерывания	78
3.5.2. Пример процедуры внешнего прерывания	79
3.5.3. Использование процедуры прерывания	79
3.6. Регистры и возможности таймеров	80
3.6.1. Регистр управления таймерами TCON	80
3.6.2. Формат регистра режима TMOD	81
3.6.3. Инициализация (загрузка) регистра TMOD	82
3.6.4. Загрузка регистров данных	82
3.6.5. Чтение регистров данных	83
3.7. Счет времени при помощи таймеров	83
3.7.1. Общие сведения	83
3.7.2. 16-битный счет времени	84
3.7.3. 8-битный счет времени с перезагрузкой	86
3.7.4. Отсчет времени без использования прерывания	87
3.7.5. Отсчет времени с использованием прерываний	87
3.8. Управление последовательным портом	89
3.8.1. Общие сведения	89
3.8.2. Назначение битов регистра управления SCON	91
3.8.3. Управление режимами последовательного порта	92
3.8.4. Режим аппаратного различия 9-го бита при приеме	94
3.8.5. Бит последовательного порта в регистре управления мощностью PCON	95
3.9. Особенности инициализации последовательного порта	96
3.9.1. Выбор режима тактирования последовательного порта	96
3.9.2. Использование таймера T1 для тактирования последовательного порта	97
3.9.3. Погрешность задания стандартной скорости	98
3.9.4. Пример инициализации последовательного порта	99
3.10. Процедуры для последовательного порта	100
3.10.1. Программное ожидание посылки	100
3.10.2. Прием по прерыванию	100
3.10.3. Передача байта без использования прерываний	101
3.10.4. Передача с ожиданием окончания посылки	102
3.10.5. Передача пакета данных по прерыванию	102
Список используемых сокращений	103
Литература	103

Введение

Как известно, экономическую целесообразность производства микросхем определяет их тираж, который, в свою очередь, зависит от степени универсальности изделия. Поэтому в настоящее время стараются по возможности унифицировать элементную базу, используя два пути реализации универсального схемотехнического элемента.

Первый – *аппаратный*, когда изготавливается интегральная схема в виде набора логических и аналоговых элементов, которые коммутируются пользователем при программировании, определяя заданную схему. Заметим, что в этом случае достигается предельное для используемых элементов быстродействие.

Второй путь – *программный*, когда необходимая функция по обработке данных в виде программного алгоритма заносится в память однокристалльной ВМ (ОВМ), которая под управлением программы реализует необходимые операции. Быстродействие в этом случае существенно хуже, поскольку необходимые функции приходится реализовывать последовательно во времени.

Однако на практике в диапазоне относительно низких рабочих частот существует огромный круг задач, где применение ОВМ экономически и технически оправдано благодаря вычислительному потенциалу процессорного ядра.

Отметим также, что в настоящее время происходит естественное слияние двух путей создания универсальных микросхем, аппаратного и программного, в рамках концепции «система на кристалле» (СнК), когда реализуют программируемые цифровые и аналоговые цепи вместе с процессорным ядром на одном кристалле. Поэтому изучение основ схемотехники и принципа действия ОВМ совершенно необходимо современному специалисту.

В настоящее время существует множество процессорных семейств, отличающихся назначением, внутренней структурой, системой команд, набором внутренних узлов и средствами общения с внешним миром. Но первое знакомство с ОВМ недаром начинают с изучения неустаревающего 8-разрядного семейства *MCS-51*, разработанного фирмой *Intel* много лет назад и до сих пор успешно конкурирующего со своими 16- и 32-битными собратьями.

Десятки зарубежных компаний производят сотни модификаций (клонов) этого семейства – от простых маловыводных, которые за ограниченность вычислительных ресурсов иногда называют микроконтроллерами, до сложнейших многвыводных кристаллов с массивами таймеров-счетчиков, АЦП, интерфейсными схемами и зна-

чительными объемами памяти. И самое интересное, что ядро *x51* по-прежнему используется в новых разработках (например, в СпК), поскольку его архитектура функционально достаточна для множества применений и имеет существенный резерв для повышения быстродействия, снижения потребляемой мощности и увеличения объемов памяти.

Как показывает опыт, освоение общих механизмов работы основных узлов ОВМ *x51* оказывается весьма полезным в дальнейшем при изучении более сложных процессорных семейств.

Чтобы более наглядно иллюстрировать процессы, происходящие внутри ОВМ, принципы действия ее внутренних узлов будут поясняться при помощи средств языка ассемблера. В дальнейшем будем придерживаться в основном рекомендаций, изложенных в ОСТ 110342.2–87 «Язык ассемблера АСМ51».

Имена переменных и процедур, вводимые пользователем в примерах программных фрагментов, будут начинаться с префикса «ту», чтобы отличать их от стандартных имен спецрегистров, операторов и директив, которые для большей заметности будут набраны жирным шрифтом.

1 ВНУТРЕННИЕ РЕСУРСЫ ОВМ СЕМЕЙСТВА x51

2	Перечень команд ОВМ x51	37
3	Управление внутренней аппаратурой ОВМ x51	64

В этой главе приводятся основные сведения для первоначального знакомства с функциональными возможностями ОВМ, которые, не подменяя справочных, дают опорный материал для использования в дальнейшем при изучении процессоров других семейств.

1.1. Общее описание ОВМ x51

1.1.1. Внутренняя структура ОВМ

Под ОВМ обычно понимается функционально законченная БИС, имеющая в своем составе (рис. 1.1) все необходимые узлы для самостоятельной работы по программе с целью обслуживания внешних устройств.

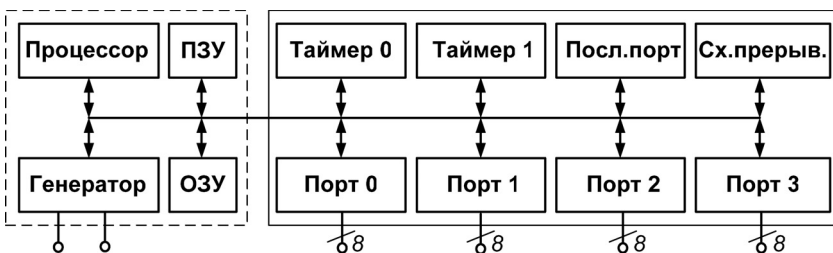


Рис. 1.1

Как правило, в ОВМ выделяют две части, основную (собственно ВМ) и дополнительную (набор вспомогательных устройств).

1.1.2. Процессорное ядро

Основой ОВМ x51 являются:

- 8-разрядный процессор, обеспечивающий вычисления и управление по программе;
- память команд в виде постоянного запоминающего устройства (ПЗУ), где хранится программа;
- память данных, которую часто для простоты называют оперативным запоминающим устройством (ОЗУ);
- тактовый генератор (с внешним кварцем), обеспечивающий внутреннюю синхронизацию.

Раздельная память команд и данных (гарвардская архитектура) применяется в большинстве ОВМ, поскольку позволяет при модер-

низации процессора увеличивать размер памяти без опасности наложения адресных пространств ПЗУ и ОЗУ друг на друга.

Процессор взаимодействует со своим обрамлением через внутреннюю системную магистраль, но обычно этот обмен скрыт от пользователя.

1.1.3. Внутренняя периферия

Важной частью ОВМ является набор так называемых внутренних периферийных устройств, предназначенных для взаимодействия с внешними объектами и аппаратной реализации некоторых функций – счета и измерения времени, цифрового и аналогового ввода-вывода, организации последовательной связи и т. д. Следует отметить, что набор этих устройств для разных семейств различен, но в любом случае имеются регистры, называемые портами, для внешнего ввода-вывода цифровых сигналов.

В стандартном варианте ОВМ x51 имеет 4 порта, два таймера-счетчика, последовательный порт и схему прерываний, особенности которых будут рассмотрены далее.

В каждом из подобных устройств имеются несколько регистров, организующих их работу и называемых специальными регистрами (*SFR* – *SPECIAL FUNCTION REGISTERS*), в отличие регистров общего назначения (РОН), необходимых для вычислений. Все спецрегистры имеют стандартные символические имена, которые понимаются инструментальными программами и используются программистами. Например, регистры четырех портов обозначаются как **P0**, **P1**, **P2**, **P3**, регистр управления для последовательного порта имеет имя **SCON** (*SERIAL CONTROL*). Другие стандартные имена спецрегистров будут рассмотрены по ходу пояснения.

Отметим только, что в состав внутренней периферии более сложных ОВМ дополнительно входят энергонезависимая память данных, АЦП, ЦАП, цифровые часы, интерфейс (например, *USB*) и другие устройства, необходимые микропроцессорной системе для обслуживания внешних объектов. Эти дополнительные устройства имеют свой набор и имена спецрегистров.

1.1.4. Условное графическое обозначение

Выполнено, как обычно, в виде прямоугольника (рис. 1.2), в центральном поле которого пишется знак процессора – *CPU*. Как известно, обозначения внешних *входов* должны размещаться слева,

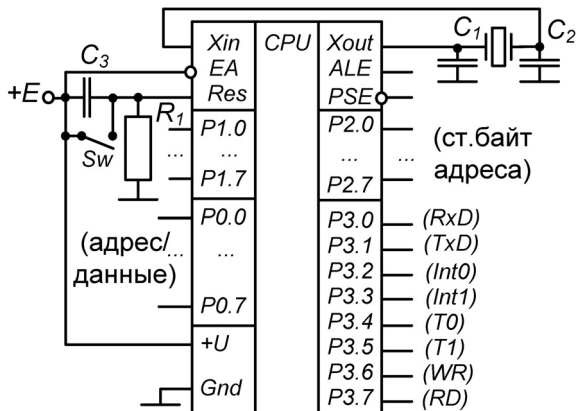


Рис. 1.2

а выходов – справа. Однако это правило неприменимо для портов **P0...P3**, поскольку биты портов могут служить как для ввода, так и для вывода сигналов. Поэтому внешние выводы портов рисуют и слева, и справа.

Каждый из 4 портов имеет 8 линий, то есть всего в распоряжении разработчика имеются 32 внешних вывода, которые можно использовать независимо друг от друга, побайтно или побитно, для ввода или вывода сигналов.

Выводы *отдельных битов* портов нумеруются числами от 0 до 7 через точку после имени порта, например **P1.5**. Особенности схемотехники выходных цепей портов приведены в параграфе 1.2.

Замечательной особенностью семейства *x51* является наличие команд управления отдельными битами портов или переменных (см. п. 2.4.6).

Общее число внешних выводов, включая общий вывод и питание, в стандартном варианте равно 40. Но существует множество разновидностей ОВМ семейства *x51*, отличающихся размером внутренней памяти, наличием дополнительных устройств (АЦП, интерфейсов). Такие сложные кристаллы могут иметь значительно большее число внешних выводов и соответствующую цену. Для простых применений выпускают дешевые кристаллы с усеченным набором внутренней периферии и малым числом внешних выводов (см. п. 1.2.3).

Важно помнить, что многие биты портов ОВМ имеют иные (альтернативные) назначения, которые указаны на рис. 1.2 в скобках и могут быть при необходимости запрограммированы (см. пп. 1.1.6 и 1.1.7).

Кроме 32 линий портов для общения с внешними устройствами, ОВМ имеет несколько служебных выводов, назначение которых рассмотрено далее.

1.1.5. Служебные выводы общего назначения

Обеспечивают работоспособность ОВМ и служат для подключения внешних элементов для генератора и цепи сброса.

Xin, Xout – вход и выход внутреннего генератора для подключения внешнего кварцевого резонатора, задающего тактовую частоту работы ОВМ.

Частоту кварца выбирает пользователь, исходя из условий технического задания. Например, на предельных рабочих частотах (от 12 до 50 МГц для разных кристаллов) достигается максимальная производительность, но существенно увеличивается энергопотребление. Переносные устройства, питающиеся от батареек, для снижения потребляемой мощности используют пониженные частоты от 1 МГц и менее. Иногда, если позволяет процессор, применяют даже кварц от цифровых часов 32 кГц. Очевидно, что при этом производительность системы снижается.

Следует отметить, что кварц обязательно обрамляется внешними конденсаторами C_1 и C_2 , обеспечивающими фазовое условие возникновения генерации. Величина этих конденсаторов указывается производителем в технической документации и находится в пределах от 15 до 50 пФ. Причем меньшая цифра относится к природным кварцам, а большая – к искусственным (керамическим).

В некоторых ОВМ можно программно включить генератор с внутренней RC-цепочкой, задающей фиксированную частоту, например 1 МГц. Но стабильность тактовой частоты в этом случае невысока.

RES (от **RESet** – сбросить) – вход, который служит для установки ОВМ в начальное состояние. При этом большинство битов спецрегистров сбрасываются в нуль, а биты портов устанавливаются в единичное состояние (ориентируются на ввод, см. п. 1.2.2). В программный счетчик записывается нулевой код, и выполнение программы после окончания импульса сброса начинается поэтому с нулевой ячейки (см. п. 1.3.1).

Единичный сигнал сброса должен иметь длительность, большую времени разгона генератора. Время разгона (старта) генератора зависит от частоты и добротности кварца. Для природных кварцев это время равно примерно 10 000 тактовых периодов. Например,

на частоте 12 МГц время разгона – 1,2 мс, на 1 МГц – 10 мс. Для страховки длительность импульса сброса делают большей 20–50 мс. В простейшем случае импульс сброса получают от кнопки Sw или при помощи дифференцирующей RC-цепочки, как показано на рис. 1.2. При подаче напряжения питания на схему разряженный ранее конденсатор C_3 представляет в первом приближении закоротку. Поэтому на выводе сброса возникает единичный сигнал, который по мере заряда конденсатора спадает до нуля. Учитывая, что большой точности для импульса сброса не требуется, берут постоянную времени с запасом, например $R_1 C_3 \approx 50$ мс. Поэтому если $R_1 = 10$ кОм, то $C_3 = 50 \cdot 10^{-3} / 10 \cdot 10^3 = 5$ мкФ.

Отметим, что в некоторых микросхемах есть внутренний твердотельный резистор между входом и общей точкой, поэтому внешний резистор в цепи сброса можно в этом случае исключить.

Для дальнейшего уменьшения внешних элементов внутри некоторых ОВМ иногда размещают и сам формирователь импульса сброса, который в ряде случаев делают в виде так называемого сторожевого таймера (Watchdog Timer). Это специализированное устройство сброса предохраняет программу от «зависаний» в тех случаях, когда некому перезапустить программу. При нормальной работе сторожевой таймер периодически обнуляется программой, чтобы сигнал сброса был пассивным. При нештатной работе программы таких обнулений нет и сторожевой таймер спустя некоторое время вырабатывает сигнал сброса, возвращая программу в исходное состояние.

1.1.6. Выводы для обслуживания внешней памяти

В то время когда создавалось семейство x51, технология изготовления внутренней памяти была недостаточно развита. Поэтому базовая архитектура предусматривала подключение внешней памяти, для чего пришлось пожертвовать двумя портами **P0** и **P2**, которые были задействованы для обслуживания внешних ПЗУ и ОЗУ. Например, порт **P0** в этом режиме используется для мультиплексной передачи младшего байта адреса – данных, а **P2** – для передачи старшего байта адреса. Подробности здесь не рассматриваются, при необходимости их можно найти, например, в [1]. Ниже указаны минимальные сведения, касающиеся назначения внешних выводов для обслуживания внешней памяти.

EA (**EXTERNAL ADDRESS** – внешний адрес) – *инверсный вход*, сигнал на этом выводе определяет, какое ПЗУ будет использоваться в ра-

боте, внешнее или внутреннее. В большинстве случаев используется внутренняя память команд, и этот инверсный сигнал пассивен и равен 1.

ALE (**A**DDRESS **L**ATCH **E**NABLE – «защелка» адреса разрешена) – *выход*, на котором формируется тактовый импульс для синхронизации внешних устройств, служит чаще всего для фиксации во внешнем регистре младшего байта адреса, выводимого через порт **P0**.

PSE (**P**ROGRAM **S**TORAGE **E**NABLE) – *инверсный выход*, на котором образуется синхросигнал чтения из внешней памяти команд. Он возникает тогда, когда происходит выборка из внешнего ПЗУ, определяя момент приема кода очередной команды через порт **P0**.

Отметим, что выходы синхросигналов чтения-записи (RD и WR) для внешнего ОЗУ используют при необходимости биты порта **P3** (см. п. 1.1.7).

Порт **P1** в базовой конфигурации альтернативных функций не имеет. У кристаллов, не предназначенных для подключения внешней памяти, сигналы EA, ALE и PSE отсутствуют.

Кроме того, отметим, что при занесении двоичных кодов во внутреннее ПЗУ (программировании кристалла) практически все выводы также меняют свое назначение. Но эти вопросы специфичны и здесь не рассматриваются.

1.1.7. Альтернативные функции порта **P3**

Чтобы не увеличивать числа внешних выводов ОВМ, некоторые из битов порта **P3** могут использоваться различной внутренней аппаратурой, то есть являются многофункциональными.

Перечень альтернативных названий битов порта **P3**, которые внутренняя периферия может после программной активации использовать для общения с внешними устройствами, дан ниже.

RxD (**R**ECEIVE **D**ATA – принимаемые данные) – *вход* приемника последовательного порта.

TxD (**T**RANSMITTE **D**ATA – передаваемые данные) – *выход* передатчика последовательного порта.

INT0, **INT1** (**I**NTERRUPT – прерывание) – *входы* внешних запросов на прерывание работы процессора по программе для экстренного обслуживания внешних устройств. Механизм прерывания будет рассмотрен в параграфе 3.4.

T0, **T1** – внешние *входы* для управления таймерами-счетчиками (см. параграф 3.6).

RD (READ) и *WR (WRITE)* – выходы инверсных синхросигналов чтения и записи, которые формируются в момент, когда процессор обращается к внешнему ОЗУ, производя чтение или запись.

Интересной особенностью некоторых ОВМ является возможность программно назначать внешний вывод порта для обслуживания внутренней периферии.

1.2. Особенности выходных цепей портов

При подключении внешних схем к ОВМ нужно помнить, что выходные каскады портов в *x51* имеют схемотехнику двух типов: классическую двунаправленную (с двумя ключами и третьим состоянием) и квазидвунаправленную (с одним ключом и внутренним резистором нагрузки).

1.2.1. Выходные цепи порта P0

Порт *P0* имеет двунаправленную структуру с третьим состоянием, свойственную практически всем микропроцессорным магистралям для обмена информацией с множеством абонентов. В первом приближении выходной каскад, обслуживающий отдельный вывод порта *P0*, можно представить в виде двух противофазно работающих ключей (рис. 1.3), которые управляются командами процессора.

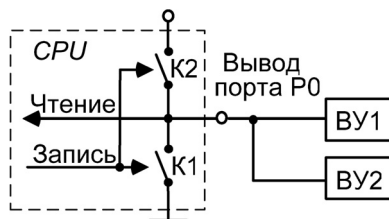


Рис. 1.3

При выводе информации один из ключей проводит ток, другой – не проводит, обеспечивая на внешнем выводе сигналы 0 или 1, например для внешнего устройства ВУ1. При вводе информации оба ключа находятся в непроводящем (третьем) состоянии, а данные на выводе под управлением не показанного на схеме синхросигнала могут устанавливаться внешним устройством ВУ2.

Такая структура порта позволяет подключать к ОВМ внешнюю память, и в этом режиме порт **P0** выполняет функции мультиплексированной шины адреса/данных. Отметим также, что большую роль в организации обмена через порт **P0** играют синхросигналы чтения-записи *PSE*, *RD* и *WR* (см. пп. 1.1.5 и 1.1.6), разрешающие ввод или вывод *байтовой* информации.

Очень важно отметить, что в том случае, когда внешняя память отсутствует, ключ **K2** порта **P0** всегда находится в непроводящем состоянии, работает только ключ **K1**, то есть двухключевой каскад превращается в каскад с открытым стоком. Эту особенность нужно обязательно учитывать при выводе информации (см. пп. 1.2.4 и 1.2.5) и включать **внешний** «подтягивающий» резистор (см. далее).

1.2.2. Выходные цепи портов **P1**, **P2**, **P3**

Эти порты имеют так называемую квазидвухнаправленную структуру (рис. 1.4), основанную на использовании однотранзисторной выходной цепи с внутренним транзисторным ключом **K1** и резистором R_B , который часто называют «подтягивающим» (английский синоним *Pull-up*).

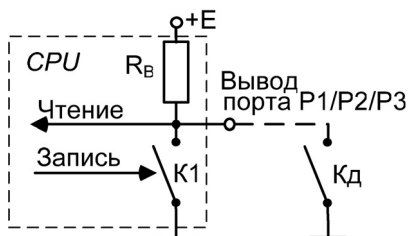


Рис. 1.4

Подобная конфигурация выходной цепи позволяет организовать ввод или вывод отдельно по каждой из линий без использования разрешающих синхросигналов направления.

При работе линии порта на вывод информация (в виде 0 или 1) записывается процессором во внутренний триггер порта и при помощи буферного ключа **K1** устанавливается на внешнем выводе. Таким образом, активная роль принадлежит ключу **K1**, и внешние сигналы на вывод, вообще говоря, подавать не рекомендуется (хотя и не возбраняется).

Информацию с этого же вывода можно также прочитать при помощи процессора. Однако если ранее триггер был установлен в 0, то ключ K1 будет проводить. Поэтому процессор в этом случае будет принимать нулевой сигнал независимо от состояния внешнего датчика (ключа Кд). И чтобы настроить (запрограммировать) определенный бит порта на ввод, нужно предварительно установить его в «1». Тогда сигнал от ключевого или логического датчика будет правильно восприниматься процессором.

Примечания: 1. Внутренний резистор $R_B \approx 40$ кОм в моменты записывания ключа K1 переключается на короткое время не показанным на схеме ключом, чтобы форсировать установку единичного сигнала путем ускоренного заряда паразитной емкости нагрузки.

2. Для уменьшения потребляемого тока внутренний резистор R_B отсоединяется от питания, если напряжение на выводе становится меньшим примерно 1,5 В (см. п. 1.2.5).

3. В современных ОВМ тип выходного каскада можно программировать, выбирая двухключевой выход, квазидвунаправленный вход/выход с одним ключом (с внутренним сопротивлением или без) или только вход.

1.2.3. Особенности маловыводного варианта x51

В простых микропроцессорных системах не все ресурсы и внешние выводы («ножки») стандартного варианта ОВМ оказываются задействованы. В таких случаях можно использовать так называемые «малоножечные» кристаллы с несколько ограниченными функциональными возможностями. Особенности их рассмотрим на примере ОВМ AT89C2051 фирмы *Atmel*, УГО которой приведено на рис. 1.5.

Ядро и состав такой ОВМ почти полностью соответствуют стандартной конфигурации. Отличие заключается только в том, что исключены

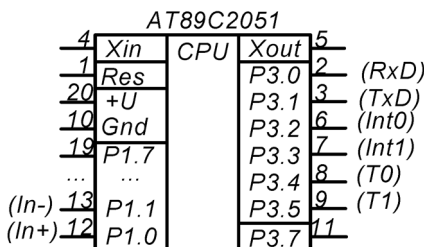


Рис. 1.5

служебные выводы и два порта (**P0**, **P2**), за счет чего число внешних выводов уменьшено до 20. Очевидно, что при этом теряется возможность подключения внешней памяти и перестают работать некоторые команды по ее обслуживанию.

Но вся внутренняя периферия семейства *x51*, как и прежде, остается доступной через альтернативные выводы (показаны в скобках) порта **P3**. И более того, дополнительно в состав данного кристалла введен внутренний аналоговый *компаратор* (рис. 1.6), который весьма полезен на практике для многих применений.

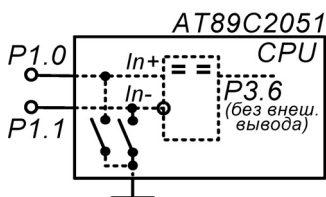


Рис. 1.6

Для внешнего подключения компаратора задействованы квази-двунаправленные *выводы P1.0* (прямой вход $In+$) и **P1.1** (инверсный вход $In-$), ключи которых в данной микросхеме лишены *внутренних подтягивающих резисторов*. Если компаратор используется, то эти разряды должны иметь единичное значение (внутренние ключи не проводят), иначе компаратор не будет правильно работать.

Выход внутреннего компаратора скрыт от пользователя, поскольку подключен к не имеющему внешнего вывода биту **P3.6**. Неудобств при этом не возникает, потому что указанный бит обычно читают и анализируют программно.

1.2.4. Подключение нагрузки к выводу порта

Нагрузочная способность внешних выводов ОВМ зависит от модификации. В стандартном варианте считается, что к выводу порта можно подсоединить один ТТЛ-вход, то есть

$$I_{\text{МАКС}}^0 \approx I_{\text{ВХ}}^0 \approx -1,6 \text{ мА} \quad \text{и} \\ I_{\text{МАКС}}^1 \approx I_{\text{ВХ}}^1 \approx 0,1 \text{ мА.}$$

Напомним, что в соответствии с несимметричной входной характеристикой ТТЛ-элемента ток $I_{\text{МАКС}}^0$ втекает в выходной зажим, $I_{\text{МАКС}}^1$ вытекает из него.

Более современные кристаллы имеют увеличенную нагрузочную способность. Например, маловыводная микросхема *AT89C2051* допускает выходной ток до 20 мА.

Некоторые микросхемы могут иметь специализированные выводы с еще большим током для прямого управления, например светодиодными индикаторами. Но общий ток всех портов ОВМ обычно ограничивается на уровне 60–80 мА.

При подключении нескольких входов внешних микросхем к одному выходу ОВМ вычисляют обычно суммы входных токов $I_{ВХ}^0$ и $I_{ВХ}^1$, которые должны быть меньше соответствующих максимальных выходных токов ОВМ – $I_{МАКС}^0$ и $I_{МАКС}^1$.

Нередко к внешнему выводу ОВМ подключают светодиод для индикации. В этом случае нужно обязательно включать последовательно внешний резистор $R_{ОГР}$ для ограничения выходного тока на уровне, меньшем максимального для выходной цепи ($I_{МАКС}^0$ или $I_{МАКС}^1$).

Известны два способа подключения светодиода. В первом из них (рис. 1.7 и 1.9) свечение диода должно происходить при единичном выходном сигнале, а во втором (рис. 1.8 и 1.10) – при нулевом. Однако схема рис. 1.7 для бита порта *P0* оказывается *нерабочей*, поскольку ключ *K2* функционирует только при обслуживании внешней памяти (см. п. 1.2.1). В этом случае следует использовать схему рис. 1.9 с внешним резистором $R_{ОГР}$.



Рис. 1.7

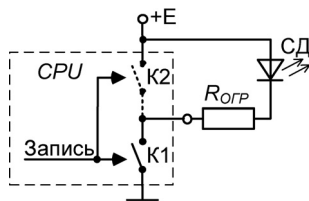


Рис. 1.8

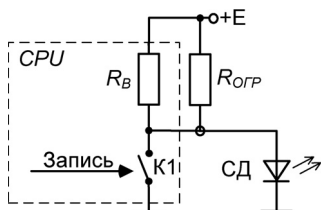


Рис. 1.9

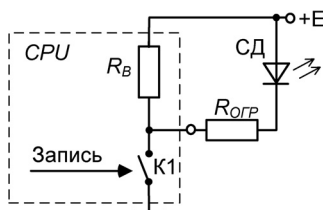


Рис. 1.10

Величину резистора для схем рис. 1.8 и 1.10 ограничивают снизу, используя неравенство

$$R_{OГР} \geq U_{Rop}/I_{max} = (E - U_{ВЫХ}^0 - U_{СДip})/I_{МАКС}^0, \quad (1.1)$$

а для схемы рис. 1.9:

$$R_{OГР} \geq U_{Rop}/I_{max} = (E - U_{ВЫХ}^0)/I_{МАКС}^0. \quad (1.2)$$

Кроме светодиодов, к выводам ОВМ бывает нужно подсоединять и другую более низкоомную нагрузку, для которой нагрузочной способности ОВМ может не хватить. В таких случаях применяют ключевые усилители на основе внешних микросхем или транзисторов.

1.2.5. Внешний ключ на *n-p-n*-транзисторе

Как и в случае со светодиодом, при подключении внешнего транзистора нужно учитывать структуру выходного каскада порта ОВМ. При использовании порта *P0* не следует надеяться на двухключевой выходной каскад, который образуется только при обслуживании внешней памяти. Таким образом, схема на рис. 1.11 оказывается нерабочей, так как ключ *K2* не проводит, и нет единичного сигнала, необходимого для насыщения транзистора. Надо использовать внешний резистор $R_{ББ}$, как на рис. 1.12.

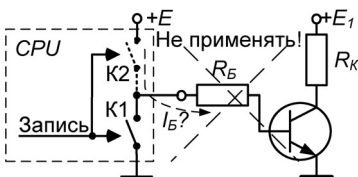


Рис. 1.11

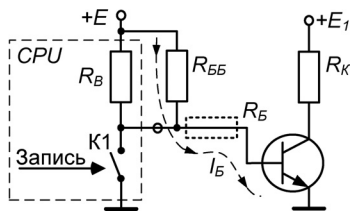


Рис. 1.12

В квазидвунаправленной схеме (рис. 1.12) на первый взгляд подобной проблемы нет благодаря наличию внутреннего резистора, который можно использовать для задания насыщающего базового тока транзистора (внешний резистор R_B практически не нужен).

Но данная схема все равно оказывается ограниченно работоспособной. Дело не только в том, что внутренний резистор обеспечивает малый базовый ток ($\approx 0,1$ мА), но и в том, что единичный уровень напряжения на выводе порта становится малым и равным напря-

жению на прямосмещенном переходе $U_{БЭ}$. А при таком напряжении внутренний резистор отсоединяется от питания (см. *примечание 2* к п. 1.2.2), базовый ток внешнего транзистора исчезает и ключ становится неуправляемым.

Поэтому рекомендуется не надеяться на внутренний резистор и подключить параллельно внешний $R_{ББ} \approx 4-10$ к (рис. 1.12), выбираемый из условия насыщения транзистора, но не менее

$$R_{ББ} \geq (E - U_{ВЫХ}^0) / I_{МАКС}^0 \quad (1.3)$$

Напряжение питания ключа E , может отличаться от напряжения питания ОВМ и иметь значение до сотен вольт.

Другим способом восстановления управляемости внешнего ключа является использование составного транзистора VT_C , (рис. 1.13), напряжение на базе которого равно сумме напряжений на эмиттерных переходах отдельных транзисторов. Эта сумма превышает внутренний порог, и твердотельный подтягивающий резистор R_B в такой схеме не отсоединяется от питания. Кроме того, составной транзистор имеет большой коэффициент усиления B_C , равный произведению коэффициентов составляющих транзисторов, что позволяет коммутировать ток нагрузки, который в B_C раз больше малого тока внутреннего резистора ($\approx 0,1$ мА).

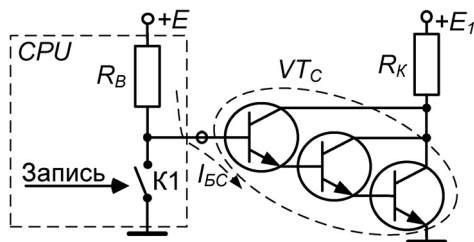


Рис. 1.13

Выходные транзисторы составного работают на границе насыщения и имеют поэтому увеличенное напряжение в проводящем состоянии. Для устранения этого недостатка и для увеличения управляющего тока внешнего ключа обычно меняют тип транзистора (см. далее), чтобы использовать для насыщения нулевой сигнал с большим выходным током $I_{МАКС}^0$.

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине
«Электронный универс»
e-Univers.ru