



[1. Функциональное назначение выводов корпуса МК51](#)

[2. Структурная схема МК51](#)

- [2.1. Арифметико-логическое устройство](#)
- [2.2. Резидентная память](#)
- [2.3. Устройство управления и синхронизации](#)

[3. Система команд МК51](#)

- [3.1. Общие сведения о системе команд](#)
- [3.2. Группа команд пересылки данных](#)
- [3.3. Группа команд арифметических операций](#)
- [3.4. Группа команд логических операций](#)
- [3.5. Группа команд операций с битами](#)
- [3.6. Группа команд передачи управления](#)

[4. Порты ввода/вывода информации](#)

[5. Доступ к внешней памяти](#)

[6. Таймер/счетчик](#)

[7. Последовательный интерфейс](#)

- [7.1. Регистр управления/статуса УАПП](#)
- [7.2. Работа УАПП в мультимикроконтроллерных системах](#)
- [7.3. Скорость приёма/передачи](#)
- [7.4. Особенности работы УАПП в различных режимах](#)

[8. Система прерываний](#)

[9. Особые режимы работы МК51](#)

- [9.1. Режим загрузки и верификации прикладных программ](#)
- [9.2. Работа МК51 в пошаговом режиме](#)
- [9.3. Сброс, режим холостого хода и режим пониженного энергопотребления](#)

1. Функциональное назначение выводов корпуса МК51

Микроконтроллер выполнен на основе высокоуровневой n-МОП технологии и выпускается в корпусе БИС, имеющем 40 внешних выводов. Цоколевка корпуса МК51 и наименования выводов показаны на [рис. 3.1](#). Для работы МК51 требуется один источник электропитания +5 В. Через четыре программируемых порта ввода/вывода МК51 взаимодействует со средой в стандарте TTL-схем с тремя состояниями выхода. Корпус МК51 имеет два вывода для подключения кварцевого резонатора, четыре вывода для сигналов, управляющих режимом работы МК, и восемь линий порта 3, которые могут быть запрограммированы пользователем на выполнение специализированных (альтернативных) функций обмена информацией со средой.

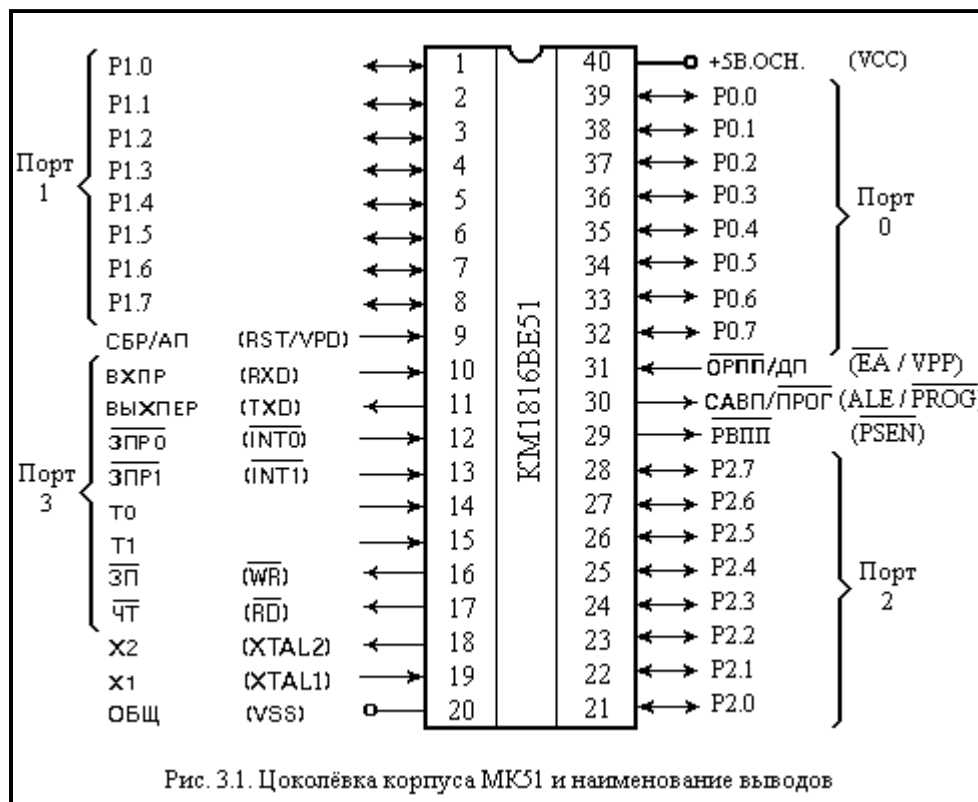


Рис. 3.1. Цоколевка корпуса МК51 и наименование выводов

[Функциональное назначение выводов корпуса МК51](#)

[Начало KM1816BE51](#)

2. Структурная схема МК51

- [2.1. Арифметическо-логическое устройство](#)
- [2.2. Резидентная память](#)
- [2.3. Устройство управления и синхронизации](#)

Основу структурной схемы МК51 (рис. 3.2) образует внутренняя двунаправленная 8-битная шина, которая связывает между собой все основные узлы и устройства: резидентную память, АЛУ, блок регистров специальных функций, устройство управления и порты ввода/вывода. Рассмотрим основные элементы структуры и особенности организации вычислительного процесса в МК51.

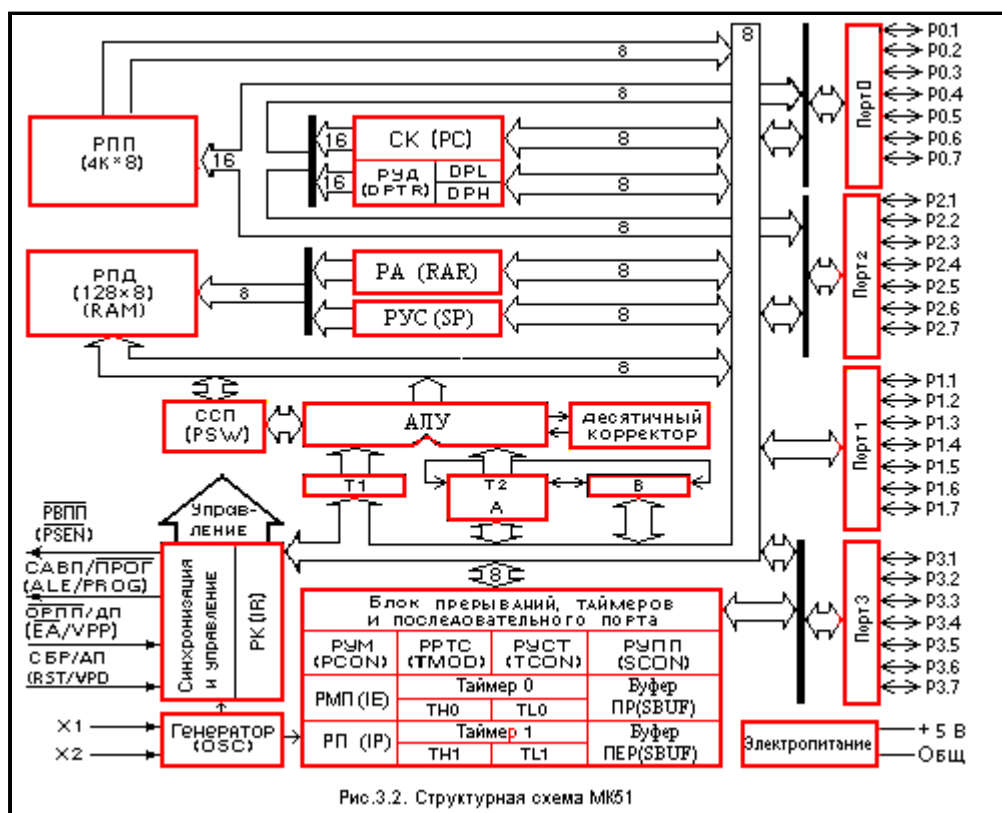


Рис. 3.2. Структурная схема МК51

[Структурная схема МК51](#)

2.1. Арифметическо-логическое устройство

8-битное АЛУ может выполнять арифметические операции сложения, вычитания, умножения и деления; логические операции И, ИЛИ, исключающее ИЛИ, а также операции циклического сдвига, сброса, инвертирования и т.п. В АЛУ имеются программно недоступные регистры T1 и T2, предназначенные для временного хранения операндов, схема десятичной коррекции и схема формирования признаков.

Простейшая операция сложения используется в АЛУ для инкрементирования содержимого регистров, продвижения регистра-указателя данных и автоматического вычисления следующего адреса РПП. Простейшая операция вычитания используется в АЛУ для декрементирования регистров и сравнения переменных.

Простейшие операции автоматически образуют "танделы" для выполнения в АЛУ таких операций, как, например, инкрементирование 16-битных регистровых пар. В АЛУ реализуется механизм каскадного выполнения простейших операций для реализации сложных команд. Так, например, при выполнении одной из команд условной передачи правления по результату сравнения в АЛУ трижды инкрементируется СК, дважды производится чтение из РПД, выполняется арифметическое сравнение двух переменных, формируется 16-битный адрес перехода и принимается решение о том, делать или не делать переход по программе. Все перечисленные операции выполняются в АЛУ всего лишь за 2 мкс.

Важной особенностью АЛУ является его способность оперировать не только байтами, но и битами. Отдельные программно-доступные биты могут быть установлены, сброшены, инвертированы, переданы, проверены и использованы в логических операциях. Эта способность АЛУ, оперировать битами, столь важна, что во многих описаниях МК51 говорится о наличии в нем "булевого процессора". Для управления объектами часто применяются алгоритмы, содержащие операции над входными и выходными булевыми переменными (истина/ложь), реализация которых средствами обычных микропроцессоров сопряжена с определенными трудностями.

Таким образом, АЛУ может оперировать четырьмя типами информационных объектов: булевыми (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В АЛУ выполняется 51 различная операция пересылки или преобразования этих данных. Так как используется 11 режимов адресации (7 для данных и 4 для адресов), то путем комбинирования "операция/ режим адресации" базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

[Структурная схема МК51](#)

2.2. Резидентная память

Память программ и память данных, размещенные на кристалле МК5 физически и логически разделены, имеют различные механизмы адресации, работают под управлением различных сигналов и выполняют разные функции.

Память программ (ПЗУ или СППЗУ) имеет емкость 4 Кбайта и предназначена для хранения команд, констант, управляющих слов инициализации, таблиц перекодировки входных и выходных сменных и т.п. РПП имеет 16-битную шину адреса, через которую обеспечивается доступ из счетчика команд или из регистра-указателя данных. Последний выполняет функции базового регистра при косвенных переходах по программе или используется в командах, оперирующих с таблицами.

Память данных (ОЗУ) предназначена для хранения переменных в процессе выполнения прикладной программы, адресуется одним байтом и имеет емкость 128 байт. Кроме того, к адресному пространству РПД примыкают адреса регистров специальных функций (РСФ), которые перечислены в [табл. 2.1](#).

Память программ, так же как и память данных, может быть расширена до 64 Кбайт путем подключения внешних БИС.

Аккумулятор и ССП. Аккумулятор является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, только с использованием аккумулятора могут быть выполнены операции сдвигов, проверка на ноль, формирование флага паритета и т.п.

Таблица 2.1. Блок регистров специальных функций

Символ	Наименование	Адрес
* ACC	Аккумулятор	0E0H
* B	Регистр-расширитель аккумулятора	0F0H

* PSW	Слово состояния программы	0D0H
SP	Регистр-указатель стека	81H
DPTR	Регистр-указатель данных (DPH)	83H
	(DPL)	82H
* P0	Порт 0	80H
* P1	Порт 1	90H
* P2	Порт 2	0A0H
* P3	Порт 3	0B0H
* IP	Регистр приоритетов	0B8H
* IE	Регистр маски прерываний	0A8H
TMOD	Регистр режима таймера/счетчика	89H
* TCON	Регистр управления/статус таймера	88H
TH0	Таймер 0 (старший байт)	8CH
TL0	Таймер 0 (младший байт)	8AH
TH1	Таймер 1 (старший байт)	8DH
TL1	Таймер 1 (младший байт)	8BH
* SCON	Регистр управления приемопередатчиком	98H
SBUF	Буфер приемопередатчика	99H
PCON	Регистр управления мощностью	87H

Примечание. Регистры, имена которых отмечены знаком (*), допускают адресацию отдельных бит.

При выполнении многих команд в АЛУ формируется ряд признаков операции (флагов), которые фиксируются в регистре ССП. В [табл.2.2](#) приводится перечень флагов ССП, даются их символические имена и описываются условия их формирования.

Таблица 2.2. Формат слова состояния программы (ССП)

Символ	Позиция	Имя и назначение			
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается аппаратурными средствами или программой при выполнении арифметических и логических операций			
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратурными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заем в бите 3			
F0	PSW.5	Флаг 0. Может быть установлен, сброшен или проверен программой как флаг, специфицируемый пользователем			
RS1	PSW.4	Выбор банка регистров. Устанавливается и сбрасывается программой для выбора рабочего банка регистров (см. примечание)			
RS0	PSW.3				
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций			
-	PSW.1	Не используется			
P	PSW.0	Флаг паритета. Устанавливается и сбрасывается аппаратуры в каждом цикле команды и фиксирует нечетное/четное число единичных бит в аккумуляторе, т.е. выполняет контроль по четности			
Примечание.		RS1	RS0	Банк	Границы адресов
		0	0	0	00H-07H
		0	1	1	08H-0FH
		1	0	2	10H-17H

Наиболее "активным" флагом ССП является флаг переноса, которые принимает участие и модифицируется в процессе выполнения множеств операций, включая сложение, вычитание и сдвиги. Кроме того, флаг переноса (C) выполняет функции "булевого аккумулятора" в командах, манипулирующих с битами. Флаг переполнения (OV) фиксирует арифметическое переполнение при операциях над целыми числами со знаком и делает возможным использование арифметики в дополнительных кодах. АЛУ не управляет флагами селекции банка регистров (RS0, RS1), и их значение полностью определяется прикладной программой и используется для выбора одного из четырех регистровых банков.

Широкое распространение получило представление о том, что в микропроцессорах, архитектура которых опирается на аккумулятор, большинство команд работают с ним, используя адресацию "по умолчанию" (неявную). В МК51 дело обстоит иначе. Хотя процессор в МК51 имеет в своей основе аккумулятор, однако, он может выполнять множество команд и без участия аккумулятора. Например, данные могут быть переданы из любой ячейки РПД в любой регистр, любой регистр может быть загружен непосредственным операндом и т.д. Многие логические операции могут быть выполнены без участия аккумулятора. Кроме того, переменные могут быть инкрементированы, декрементированы и проверены (test) без использования аккумулятора. Флаги и управляющие биты могут быть проверены и изменены аналогично.

Регистры-указатели. 8-битный указатель стека (PUS) может адресовать любую область РПД. Его содержимое инкрементируется прежде, чем данные будут запомнены в стеке в ходе выполнения команд PUSH и CALL. Содержимое PUS декрементируется после выполнения команд POP и RET. Подобный способ адресации элементов стека называют прединкрементным/постдекрементным. В процессе инициализации МК51 после сигнала СБР в PUS автоматически загружается код 07H. Это значит, что если прикладная программа не переопределяет стек, то первый элемент данных в стеке будет располагаться в ячейке РПД с адресом 08H.

Двухбайтный регистр-указатель данных (PUD) обычно используется для фиксации 16-битного адреса в операциях с обращением к внешней памяти. Командами МК51 регистр-указатель данных может быть использован или как 16-битный регистр, или как два независимых 8-битных регистра (DPH и DPL).

Таймер/счетчик. В составе средств МК51 имеются регистровые пары с символическими именами TNO, TLO и TNI, TLI, на основе которых функционируют два независимых программно-управляемых 16-битных таймера/счетчика событий.

Буфер последовательного порта. Регистр с символическим именем SBUF представляет собой два независимых регистра - буфер приемника и буфер передатчика. Загрузка байта в SBUF немедленно вызывает начало процесса передачи через последовательный порт. Когда байт считывается из SBUF, это значит, что его источником является приемник последовательного порта.

Регистры специальных функций. Регистры с символическими именами IP, IE, TMOD, TCON, SCON и PCON используются для фиксации и программного изменения управляющих бит и бит состояния схемы прерывания, таймера/счетчика, приемопередатчика последовательного порта и для управления мощностью электропитания МК51. Их организация будет описана ниже при рассмотрении особенностей работы МК51 в различных режимах.

[Структурная схема МК51](#)

2.3. Устройство управления и синхронизации

Кварцевый резонатор, подключаемый к внешним выводам X1 и X2 корпуса МК51, управляет работой внутреннего генератора, который в свою очередь формирует сигналы синхронизации.

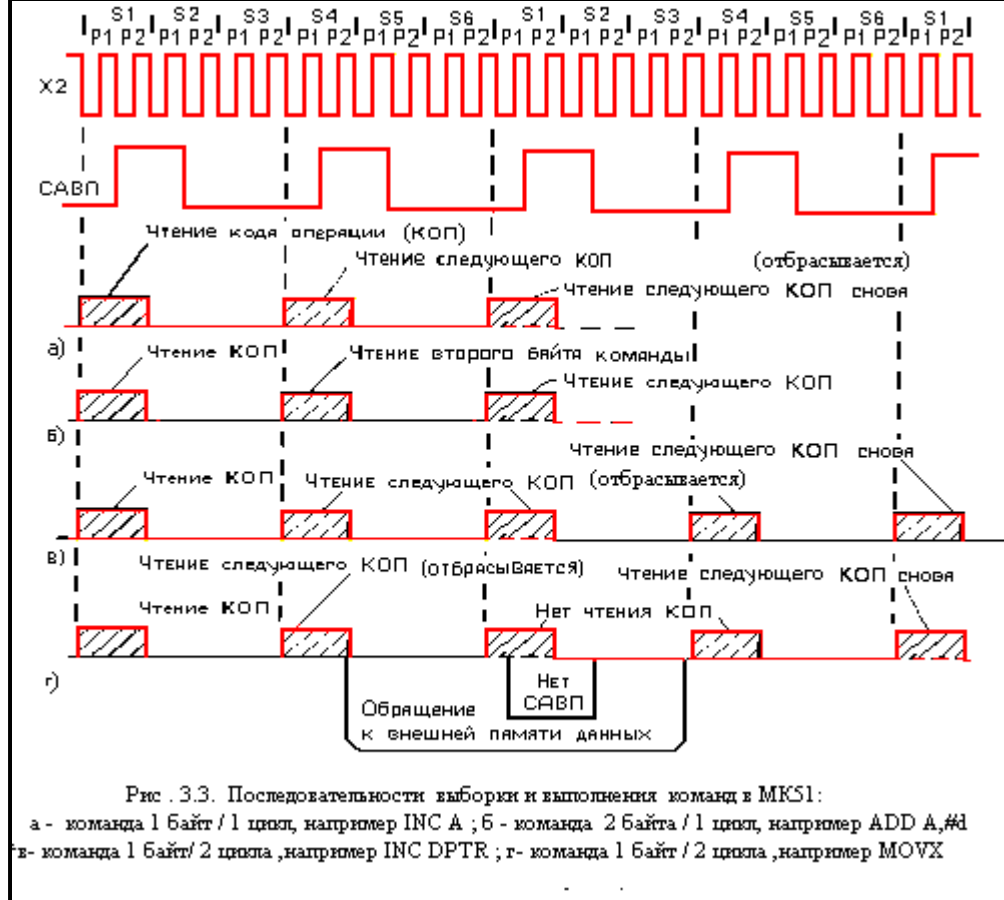


Рис. 3.3. Последовательности выборки и выполнения команд в МК51:

а - команда 1 байт/1 цикл, например INC A; **б** - команда - 2 байта/1 цикл, например ADD A,#d; **в** - команда 1 байт/2 цикла, например INC DPTR; **г** - команда 1 байт/2 цикла, например MOVX.

Устройство управления МК51 на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора или шести состояниям первичного управляющего автомата (S1_S6). Каждое состояние управляющего автомата содержит две фазы (P1, P2) сигналов резонатора. В фазе P1, как правило, выполняется операция в АЛУ, а в фазе P2 осуществляется межрегистровая передача. Весь машинный цикл состоит из 12 фаз, начиная с фазы S1P1 и кончая фазой S6P2, как показано на [рис. 3.3](#). Эта временная диаграмма иллюстрирует работу устройства управления МК51 при выборке и исполнении команд различной степени сложности. Все заштрихованные сигналы являются внутренними и недоступны пользователю МК51 для контроля. Внешними, наблюдаемыми сигналами являются только сигналы резонатора и stroba адреса внешней памяти. Как видно из временной диаграммы, сигнал CSBP формируется дважды за один машинный цикл (S1P2-S2P1 и S4P2-S5P1) и используется для управления процессом обращения к внешней памяти.

Большинство команд МК51 выполняется за один машинный цикл. Некоторые команды, оперирующие с 2-байтными словами или связанные с обращением к внешней памяти, выполняются за два машинных цикла. Только команды деления и умножения требуют четырех машинных циклов. На основе этих особенностей работы устройства управления МК51 производится расчет времени исполнения прикладных программ.

[Структурная схема МК51](#)

[Начало KM1816BE51](#)

3. Система команд МК51

- [3.1. Общие сведения о системе команд](#)
- [3.2. Группа команд пересылки данных](#)
- [3.3. Группа команд арифметических операций](#)
- [3.4. Группа команд логических операций](#)
- [3.5. Группа команд операций с битами](#)
- [3.6. Группа команд передачи управления](#)

3.1. Общие сведения о системе команд

Система команд МК51 содержит 111 базовых команд, которые удобно разделить по функциональному признаку на пять групп: команды передачи данных, арифметических операций, логических операций, передачи управления и операций с битами.

Система команд МК51 много мощнее и шире системы команд МК48, так как кроме всех команд МК48 в ее состав входят команды умножения, деления, вычитания, операций над битами, операций со стеком и расширенный набор команд передачи управления. Большинство команд (94) имеют формат один или два байта и выполняются за один или два машинных цикла. При тактовой частоте 2 МГц длительность машинного цикла составляет 1 мкс. На [рис. 3.19](#) показаны 13 типов команд МК51. Первый байт команды любых типа и формата всегда содержит код операции (КОП). Второй и третий байты содержат либо адреса операндов, либо непосредственные операнды.

Типы операндов. Состав операндов МК51 шире, чем МК48, и включает в себя операнды четырех типов: биты, 4-битные цифры, байты и 16-битные слова.

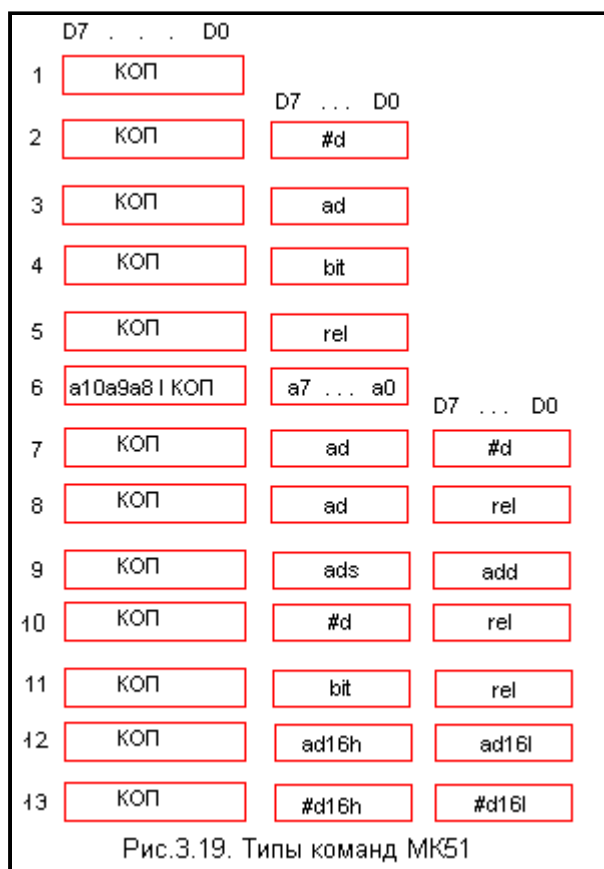


Рис. 3.19. Типы команд МК51

В отличие от МК48, который имеет только три битовых флага, МК51 имеет 128 программно-управляемых флагов пользователя. Имеется также возможность адресации отдельных бит блока регистров специальных функций и портов. Для адресации бит используется прямой 8-битный адрес (bit). Косвенная адресация бит

невозможна. Карты адресов отдельных бит представлены [рис.3.20](#) и [рис.3.21](#). Четырехбитные операнды используются только при операциях обмена (команды SWAP и XCHD). Восьмибитным операндом может быть ячейка памяти программ или данных (резидентной или внешней), константа (непосредственный операнд), регистры специальных функций (РСФ), а также порты ввода/вывода. Порты и РСФ адресуются только прямым способом. Байты памяти могут адресоваться также и косвенным образом через адресные регистры (R0, R1, DPTR и PC). Двухбайтные операнды - это константы и прямые адреса, для представления которых используются второй и третий байты команды.

Способы адресации данных. В МК51 используются такие же способы адресации данных, как и в МК48: прямая, непосредственная, косвенная и неявная. Следует отметить, что при косвенном способе адресации РПД в отличие от МК48 используются все восемь бит адресных регистров R0 и R1.

Система команд МК51 по сравнению с МК48 допускает больше комбинаций способов адресации операндов в командах, что делает ее более гибкой и универсальной.

Флаги результата. Слово состояния программы (PSW) включает в себя четыре флага: С - перенос, АС - вспомогательный перенос, О - переполнение и Р - паритет.

Флаг паритета (отсутствует в МК48) напрямую зависит от текущего значения аккумулятора. Если число единичных бит аккумулятора нечетное, то флаг Р устанавливается, а если четное - сбрасывается попытки изменить флаг Р, присваивая ему новое значение, будут безуспешными, если содержимое аккумулятора при этом останется неизменным.

Флаг АС устанавливается в случае, если при выполнении операции сложения/вычитания между тетрадами байта возник перенос/заем.

Флаг С устанавливается, если в старшем бите результата возникает перенос или заем. При выполнении операций умножения и деления флаг С сбрасывается. Флаг OV (отсутствует в МК48) устанавливается, если результат операции сложения/вычитания не укладывается в семи битах и старший (восьмой) бит результата не может интерпретироваться как знаковый. При выполнении операции деления флаг OV сбрасывается, а в случае деления на нуль устанавливается. При умножении флаг OV устанавливается, если результат больше 255.

В [табл. 3.1](#) перечисляются команды, при выполнении которых модифицируются флаги результата. В таблице отсутствует флаг паритета, так как его значение изменяется всеми командами, которые изменяют содержимое аккумулятора. Кроме команд, приведенных в таблице, флаги модифицируются командами, в которых местом назначения результата определены PSW или его отдельные биты, а также командами операций над битами.

Таблица 3.1. Команды, модифицирующие флаги результата

Команды	Флаги	Команды	Флаги
ADD	C, OV, AC	CLR C	C = 0
ADDC	C, OV, AC	CPL C	C = \overline{MC}
SUBB	C, OV, AC	ANL C, b	C
MUL	C = 0, OV	ANL C, /b	C
DIV	C = 0, OV	ORL C, b	C
DA	C	ORL C, /b	C
RRC	C	MOV C, b	C
RLC	C	CJNE	C
SETB C	C = 1		

Символическая адресация. При использовании ассемблера МК51 (ASM51) для получения объектных кодов программ допускается применение в программах символических имен регистров специальных функций (РСФ), портов и их отдельных бит ([рис. 3.21](#)).

Для адресации отдельных бит РСФ и портов (такая возможность имеется не у всех РСФ) можно использовать символическое имя бита следующей структуры:

Например, символическое имя пятого бита аккумулятора будет следующим: ACC.5. Символические имена РСФ, портов и их бит являются зарезервированными словами для ASM51, и их не надо определять с помощью директив ассемблера.

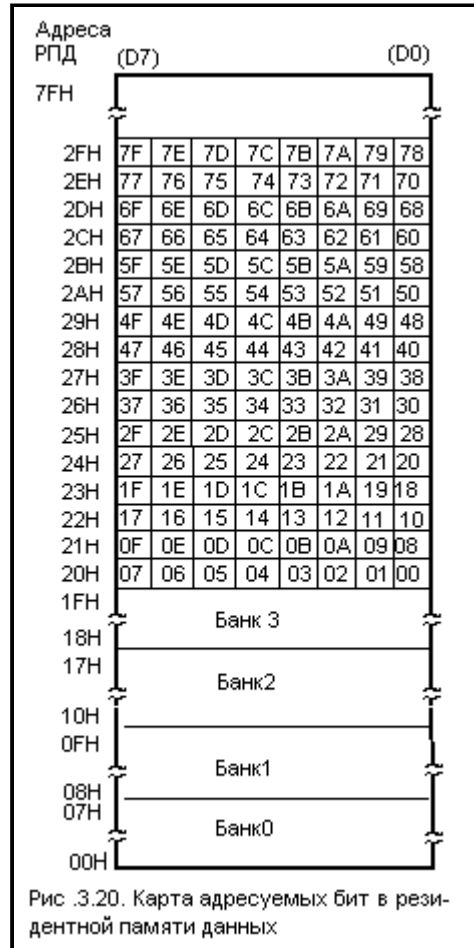
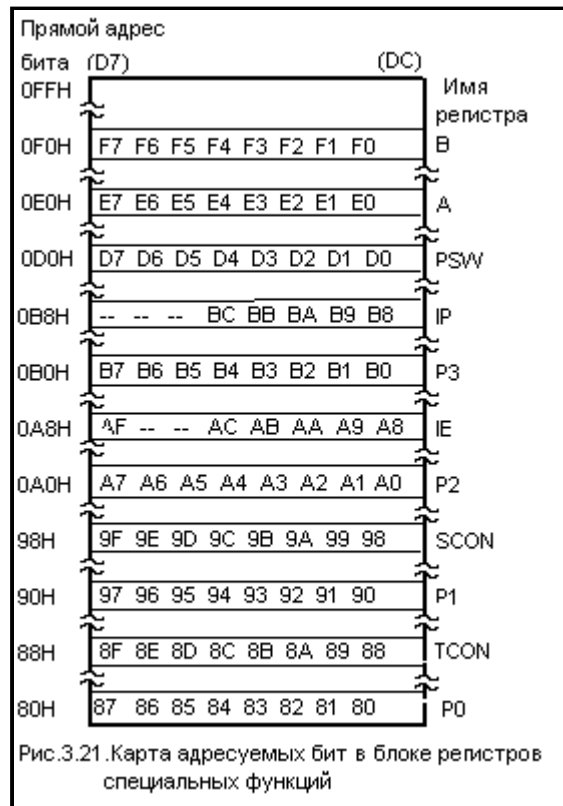


Рис. 3.20. Карта адресуемых бит



3.2. Группа команд пересылки данных

Большую часть команд данной группы (табл. 3.2) составляют команды передачи и обмена байтов. Команды пересылки бит представлены в группе команд битовых операций. Все команды данной группы не модифицируют флаги результата, за исключением команд загрузки PSW и аккумулятора (флаг паритета).

Структура информационных связей. В зависимости от способа адресации и места расположения операнда можно выделить девять типов операндов, между которыми возможен информационный обмен. Граф возможных операций передачи данных показан на рис. 3.22. Аккумулятор (A) представлен на этом графе отдельной вершиной, так как многие команды используют неявную (подразумеваемую) адресацию.

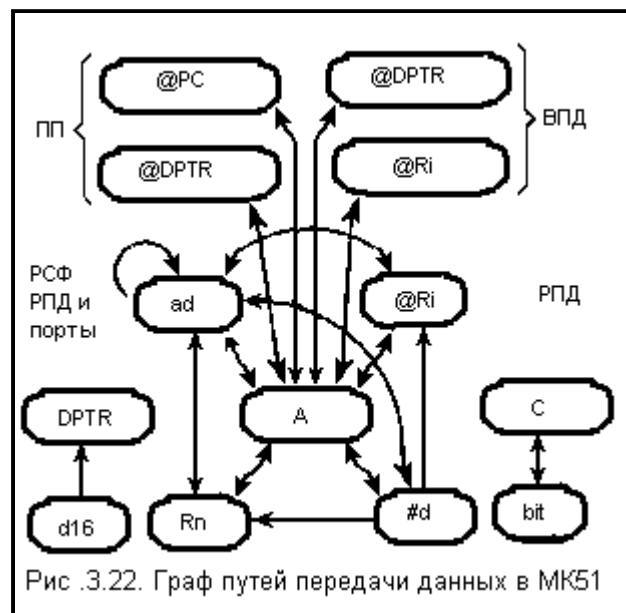


Рис 3.22. Граф путей передачи данных в МК51

В отличие от МК48 передачи данных в МК51 могут выполняться без участия аккумулятора.

Аккумулятор. В отличие от МК48 обращение к аккумулятору может быть выполнено в МК51 с использованием неявной и прямой адресации. В зависимости от способа адресации аккумулятора применяется одно из символических имен: A или ACC (прямой адрес). При прямой адресации обращение к аккумулятору производится как к одному из РСФ, и его адрес указывается во втором байте команды.

Использование неявной адресации аккумулятора предпочтительнее, однако не всегда возможно, например при обращении к отдельным битам аккумулятора.

Обращение к внешней памяти данных. Режим косвенной адресации ВПД, имеющийся в МК48, реализован также и в МК51. При использовании команд MOVX @Ri обеспечивается доступ к 256 байтам внешней памяти данных.

Существует также режим обращения к расширенной ВПД, когда для доступа используется 16-битный адрес,

Таблица 3.2. Группа команд передачи данных

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Пересылка в аккумулятор из регистра ($n = 0 - 7$)	MOV A, Rn	11101rrr	1	1	1	$(A) = (Rn)$
Пересылка в аккумулятор прямоадресуемого байта	MOV A, ad	11100101	3	2	1	$(A) = (ad)$
Пересылка в аккумулятор байта из РДП ($i = 0, 1$)	MOV A, @Ri	1110011i	1	1	1	$(A) = ((Ri))$
Загрузка в аккумулятор константы	MOV A, #d	01110100	2	2	1	$(A) = \#d$
Пересылка в регистр из аккумулятора	MOV Rn, A	11111rrr	1	1	1	$(Rn) = (A)$
Пересылка в регистр прямоадресуемого байта	MOV Rn, ad	10101rrr	3	2	2	$(Rn) = (ad)$
Загрузка в регистр константы	MOV Rn, #d	01111rrr	2	2	1	$(Rn) = \#d$
Пересылка по прямому адресу аккумулятора	MOV ad, A	11110101	3	2	1	$(ad) = (A)$
Пересылка по прямому адресу регистра	MOV ad, Rn	10001rrr	3	2	2	$(ad) = (Rn)$
Пересылка прямоадресуемого байта по прямому адресу	MOV add, ads	10000101	9	3	2	$(add) = (ads)$
Пересылка байта из РДП по прямому адресу	MOV ad, @Ri	1000011i	3	2	2	$(ad) = ((Ri))$
Пересылка по прямому адресу константы	MOV ad, #d	01110101	7	3	2	$(ad) = \#d$
Пересылка в РДП из аккумулятора	MOV @Ri, A	1111011i	1	1	1	$((Ri)) = (A)$
Пересылка в РДП прямоадресуемого байта	MOV @Ri, ad	0110011i	3	2	2	$((Ri)) = (ad)$
Пересылка в РДП константы	MOV @Ri, #d	0111011i	2	2	1	$((Ri)) = \#d$
Загрузка указателя данных	MOV DPTR, #d16	10010000	13	3	2	$(DPTR) = \#d16$
Пересылка в аккумулятор байта из ПП	MOVC A, @A + DPTR	10010011	1	1	2	$(A) = ((A) + (DPTR))$
Пересылка в аккумулятор байта из ПП	MOVC A, @A + PC	10000011	1	1	2	$(PC) = (PC) + 1$ $(A) = ((A) + (PC))$
Пересылка в аккумулятор байта из ВПД	MOVX A, @Ri	1110001i	1	1	2	$(A) = ((Ri))$
Пересылка в аккумулятор байта из расширенной ВПД	MOVX A, @DPTR	11100000	1	1	2	$(A) = ((DPTR))$
Пересылка в ВПД из аккумулятора	MOVX @Ri, A	1111001i	1	1	2	$((Ri)) = (A)$
Пересылка в расширенную ВПД из аккумулятора	MOVX @DPTR, A	11110000	1	1	2	$((DPTR)) = (A)$
Загрузка в стек	PUSH ad	11000000	3	2	2	$(SP) = (SP) + 1$ $((SP)) = (ad)$
Извлечение из стека	POP ad	11010000	3	2	2	$(ad) = (SP)$ $(SP) = (SP) - 1$
Обмен аккумулятора с регистром	XCH A, Rn	11001rrr	1	1	1	$(A) \leftrightarrow (Rn)$
Обмен аккумулятора с прямоадресуемым байтом	XCH A, ad	11000101	3	2	1	$(A) \leftrightarrow (ad)$

Обмен аккумулятора с байтом из РДП	XCH A, @Ri	1100011i	1	1	1	(A) <-> ((Ri))
Обмен младшей тетрады аккумулятора с младшей тетрадой байта РДП	XCHD A, @Ri	1101011i	1	1	1	(A ₀₋₃) <-> ((Ri) ₀₋₃)

[Система команд МК51](#)

3.3. Группа команд арифметических операций

Данную группу образуют 24 команды (табл. 3.3), выполняющие операции сложения, десятичной коррекции, инкремента/декремента байтов. Дополнительно по сравнению с МК48 введены команды вычитания, умножения и деления байтов.

Таблица 3.3. Группа команд арифметических операций

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Сложение аккумулятора с регистром (n = 0 - 7)	ADD A, Rn	00101rrr	1	1	1	(A) = (A) + (Rn)
Сложение аккумулятора с прямоадресуемым байтом	ADD A, ad	00100101	3	2	1	(A) = (A) + (ad)
Сложение аккумулятора с байтом из РПД (i = 0, 1)	ADD A, @Ri	0010011i	1	1	1	(A) = (A) + ((Ri))
Сложение аккумулятора с константой	ADD A, #d	00100100	2	2	1	(A) = (A) + #d
Сложение аккумулятора с регистром и переносом	ADDC A, Rn	00111rrr	1	1	1	(A) = (A) + (Rn) + (C)
Сложение аккумулятора с прямоадресуемым байтом и переносом	ADDC A, ad	00110101	3	2	1	(A) = (A) + (ad) + (C)
Сложение аккумулятора с байтом из РПД и переносом	ADDC A, @Ri	0011011i	1	1	1	(A) = (A) + ((Ri)) + (C)
Сложение аккумулятора с константой и переносом	ADDC A, #d	00110100	2	2	1	(A) = (A) + #d + (C)
Десятичная коррекция аккумулятора	DA A	11010100	1	1	1	Если (A ₀₋₃) > 9 ∨ ((AC) = 1), то (A ₀₋₃) = (A ₀₋₃) + 6, затем если (A ₄₋₇) > 9 ∨ ((C) = 1), то (A ₄₋₇) = (A ₄₋₇) + 6
Вычитание из аккумулятора регистра и заема	SUBB A, Rn	10011rrr	1	1	1	(A) = (A) - (C) - (Rn)
Вычитание из аккумулятора прямоадресуемого байта и заема	SUBB A, ad	10010101	3	2	1	(A) = (A) - (C) - ((ad))
Вычитание из аккумулятора байта РПД и заема	SUBB A, @Ri	1001011i	1	1	1	(A) = (A) - (C) - ((Ri))
Вычитание из аккумулятора	SUBB A, #d	10010100	2	2	1	(A) = (A) - (C) - #d

константы и заема						
Инкремент аккумулятора	INC A	00000100	1	1	1	$(A) = (A) + 1$
Инкремент регистра	INC Rn	00001rrr	1	1	1	$(Rn) = (Rn) + 1$
Инкремент прямоадресуемого байта	INC ad	00000101	3	2	1	$(ad) = (ad) + 1$
Инкремент байта в РПД	INC @Ri	0000011i	1	1	1	$((Ri)) = ((Ri)) + 1$
Инкремент указателя данных	INC DPTR	10100011	1	1	2	$(DPTR) = (DPTR) + 1$
Декремент аккумулятора	DEC A	00010100	1	1	1	$(A) = (A) - 1$
Декремент регистра	DEC Rn	00011rrr	1	1	1	$(Rn) = (Rn) - 1$
Декремент прямоадресуемого байта	DEC ad	00010101	3	2	1	$(ad) = (ad) - 1$
Декремент байта в РПД	DEC @Ri	0001011i	1	1	1	$((Ri)) = ((Ri)) - 1$
Умножение аккумулятора на регистр B	MUL AB	10100100	1	1	4	$(B)(A) = (A)*(B)$
Деление аккумулятора на регистр B	DIV AB	10000100	1	1	4	$(A).(B) = (A)/(B)$

Команды ADD и ADDC аналогичны командам сложения МК48, но допускают сложение аккумулятора с большим числом операндов. Аналогично командам ADDC существуют четыре команды SUBB, что позволяет более просто, чем в МК48, производить вычитание байтов и многобайтных двоичных чисел. В МК51 реализуется расширенный (по сравнению с МК48) список команд инкремента/декремента байтов, введена команда инкремента 16-битного регистра-указателя данных.

[Система команд МК51](#)

3.4. Группа команд логических операций

Данную группу образуют 25 команд ([табл. 3.4](#)), реализующих те же логические операции над байтами, что и в МК48. Однако в МК51 значительно расширено число типов операндов, участвующих в операциях.

Таблица 3.4. Группа команд логических операций

Название команды	Мнемокод	КОП	Г	Б	Ц	Операция
Логическое И аккумулятора и регистра	ANL A, Rn	01011rrr	1	1	1	$(A) = (A) \wedge (Rn)$
Логическое И аккумулятора и прямоадресуемого байта	ANL A, ad	01010101	3	2	1	$(A) = (A) \wedge (ad)$
Логическое И аккумулятора и байта из РПД	ANL A, @Ri	0101011i	1	1	1	$(A) = (A) \wedge ((Ri))$
Логическое И аккумулятора и константы	ANL A, #d	01010100	2	2	1	$(A) = (A) \wedge \#d$
Логическое И прямоадресуемого байта и аккумулятора	ANL ad, A	01010010	3	2	1	$(ad) = (ad) \wedge (A)$
Логическое И прямоадресуемого байта и константы	ANL ad, #d	01010011	7	3	2	$(ad) = (ad) \wedge \#d$
Логическое ИЛИ аккумулятора и регистра	ORL A, Rn	01001rrr	1	1	1	$(A) = (A) \vee (Rn)$
Логическое ИЛИ аккумулятора и прямоадресуемого байта	ORL A, ad	01000101	3	2	1	$(A) = (A) \vee (ad)$
Логическое ИЛИ аккумулятора и байта из РПД	ORL A, @Ri	0100011i	1	1	1	$(A) = (A) \vee ((Ri))$

Логическое ИЛИ аккумулятора и константы	ORL A, #d	01000100	2	2	1	$(A) = (A) \vee \#d$
Логическое ИЛИ прямоадресуемого байта и аккумулятора	ORL ad, A	01000010	3	2	1	$(ad) = (ad) \vee (A)$
Логическое ИЛИ прямоадресуемого байта и константы	ORL ad, #d	01000011	7	3	2	$(ad) = (ad) \vee \#d$
Исключающее ИЛИ аккумулятора и регистра	XRL A, Rn	01101rrr	1	1	1	$(A) = (A) \nabla (Rn)$
Исключающее ИЛИ аккумулятора и прямоадресуемого байта	XRL A, ad	01100101	3	2	1	$(A) = (A) \nabla (ad)$
Исключающее ИЛИ аккумулятора и байта из РПД	XRL A, @Ri	0110011i	1	1	1	$(A) = (A) \nabla ((Ri))$
Исключающее ИЛИ аккумулятора и константы	XRL A, #d	01100100	2	2	1	$(A) = (A) \nabla \#d$
Исключающее ИЛИ прямоадресуемого байта и аккумулятора	XRL ad, A	01100010	3	2	1	$(ad) = (ad) \nabla (A)$
Исключающее ИЛИ прямоадресуемого байта и константы	XRL ad, #d	01100011	7	3	2	$(ad) = (ad) \nabla \#d$
Сброс аккумулятора	CLR A	11100100	1	1	1	$(A) = 0$
Инверсия аккумулятора	CPL A	11110100	1	1	1	$(A) = (\bar{A})$
Сдвиг аккумулятора влево циклически	RL A	00100011	1	1	1	$(A_{n+1}) = (A_n),$ $n = 0 ? 6, (A_0) = (A_7)$
Сдвиг аккумулятора влево через перенос	RLC A	00110011	1	1	1	$(A_{n+1}) = (A_n),$ $n = 0 ? 6, (A_0) = (C),$ $(C) = (A_7)$
Сдвиг аккумулятора вправо циклически	RR A	00000011	1	1	1	$(A_n) = (A_{n+1}),$ $n = 0 ? 6, (A_7) = (A_0)$
Сдвиг аккумулятора вправо через перенос	RRC A	00010011	1	1	1	$(A_n) = (A_{n+1}),$ $n = 0 ? 6, (A_7) = (C),$ $(C) = (A_0)$
Обмен местами тетрад в аккумуляторе	SWAP A	11000100	1	1	1	$(A_{0-3}) \leftrightarrow (A_{4-7})$

В отличие от МК48 имеется возможность производить операцию "исключающее ИЛИ" с содержимым портов. Команда XRL может быть эффективно использована для инверсии отдельных бит портов.

[Система команд МК51](#)

3.5. Группа команд операций с битами

Отличительной особенностью данной группы команд (табл. 3.5) является то, что они оперируют с однокбитными операндами. В качестве таких операндов могут выступать отдельные биты некоторых регистров специальных функций (РСФ) и портов, а также 128 программных флагов пользователя.

Таблица 3.5. Группа команд операции с битами

Название команды	Мnemonic	КОП	Т	Б	Ц	Операция
Сброс переноса	CLR C	11000011	1	1	1	$(C) = 0$
Сброс бита	CLR bit	11000010	4	2	1	$(b) = 0$
Установка переноса	SETB C	11010011	1	1	1	$(C) = 1$
Установка бита	SETB bit	11010010	4	2	1	$(b) = 1$
Инверсия переноса	CPL C	10110011	1	1	1	$(C) = (\overline{C})$
Инверсия бита	CPL bit	10110010	4	2	1	$(b) = (\overline{b})$
Логическое И бита и переноса	ANL C, bit	10000010	4	2	2	$(C) = (C) \wedge (b)$
Логическое И инверсии бита и переноса	ANL C, /bit	10110000	4	2	2	$(C) = (C) \wedge (\overline{b})$
Логическое ИЛИ бита и переноса	ORL C, bit	01110010	4	2	2	$(C) = (C) \vee (b)$
Логическое ИЛИ инверсии бита и переноса	ORL C, /bit	10100000	4	2	2	$(C) = (C) \vee (\overline{b})$
Пересылка бита в перенос	MOV C, bit	10100010	4	2	1	$(C) = (b)$
Пересылка переноса в бит	MOV bit, C	10010010	4	2	2	$(b) = (C)$

Существуют команды сброса (CLR), установки (SETB) и инверсии (CPL) бит, а также конъюнкции и дизъюнкции бита и флага переноса. Для адресации бит используется прямой восьмиразрядный адрес (bit). Косвенная адресация бит невозможна.

Система команд МК51

3.6. Группа команд передачи управления

К данной группе команд (табл. 3.6) относятся команды, обеспечивающие условное и безусловное ветвление, вызов подпрограмм и возврат из них, а также команда пустой операции NOP. В большинстве команд используется прямая адресация, т.е. адрес перехода целиком (или его часть) содержится в самой команде передачи управления. Можно выделить три разновидности команд ветвления по разрядности указываемого адреса перехода.

Таблица 3.6. Группа команд передачи управления

Название команды	Мnemonic	КОП	Т	Б	Ц	Операция
Длинный переход в полном объеме памяти в программ	LJMP ad16	00000010	12	3	2	$(PC) = ad16$
Абсолютный переход внутри страницы в 2 Кбайта	AJMP ad11	$a_{10}a_9a_800001$	6	2	2	$(PC) = (PC) + 2$ $(PC_{0-10}) = ad11$
Короткий относительный переход внутри страницы в 256 байт	SJMP rel	10000000	5	2	2	$(PC) = (PC) + 2$ $(PC) = (PC) + rel$
Косвенный относительный переход	JMP @A+DPTR	01110011	1	1	2	$(PC) = (A) + (DPTR)$ $(PC) = (PC) + 2,$

Переход, если аккумулятор равен нулю	JZ rel	01100000	5	2	2	если (A) = 0, то (PC) = (PC) + rel
Переход, если аккумулятор не равен нулю	JNZ rel	01110000	5	2	2	(PC) = (PC) + 2, если (A) ≠ 0, то (PC) = (PC) + rel
Переход, если перенос равен единице	JC rel	01000000	5	2	2	(PC) = (PC) + 2, если (C) = 1, то (PC) = (PC) + rel
Переход, если перенос равен нулю	JNC rel	01010000	5	2	2	(PC) = (PC) + 2, если (C) = 0, то (PC) = (PC) + rel
Переход, если бит равен единице	JB bit, rel	00100000	11	3	2	(PC) = (PC) + 3, если (b) = 1, то (PC) = (PC) + rel
Переход, если бит равен нулю	JNB bit, rel	00110000	11	3	2	(PC) = (PC) + 3, если (b) = 0, то (PC) = (PC) + rel
Переход, если бит установлен, с последующим сбросом бита	JBC bit, rel	00010000	11	3	2	(PC) = (PC) + 3, если (b) = 1, то (b) = 0 и (PC) = (PC) + rel
Декремент регистра и переход, если не нуль	DJNZ Rn, rel	11011rrr	5	2	2	(PC) = (PC) + 2, (Rn) = (Rn) - 1, если (Rn) ≠ 0, то (PC) = (PC) + rel
Декремент прямоадресуемого байта и переход, если не нуль	DJNZ ad, rel	11010101	8	3	2	(PC) = (PC) + 2, (ad) = (ad) - 1, если (ad) ≠ 0, то (PC) = (PC) + rel
Сравнение аккумулятора с прямоадресуемым байтом и переход, если не равно	CJNE A, ad, rel	10110101	8	3	2	(PC) = (PC) + 3, если (A) ≠ (ad), то (PC) = (PC) + rel, если (A) < (ad), то (C) = 1,

						иначе (C) = 0
Сравнение аккумулятора с константой и переход, если не равно	CJNE A, #d, rel	10110100	10	3	2	(PC) = (PC) + 3, если (A) ? #d, то (PC) = (PC) + rel, если (A) < #d, то (C) = 1, иначе (C) = 0
Сравнение регистра с константой и переход, если не равно	CJNE Rn, #d, rel	10111rrr	10	3	2	(PC) = (PC) + 3, если (Rn) ? #d, то (PC) = (PC) + rel, если (Rn) < #d, то (C) = 1, иначе (C) = 0
Сравнение байта в РПД с константой и переход, если не равно	CJNE @Ri, #d, rel	1011011i	10	3	2	(PC) = (PC) + 3, если ((Ri)) ? #d, то (PC) = (PC) + rel, если ((Ri)) < #d, то (C) = 1, иначе (C) = 0
Длинный вызов подпрограммы	LCALL ad16	00010010	12	3	2	(PC) = (PC) + 3, (SP) = (SP) + 1, ((SP)) = (PC ₀₋₇), (SP) = (SP) + 1, ((SP)) = (PC ₈₋₁₅), (PC) = ad16
Абсолютный вызов подпрограммы в пределах страницы в 2 Кбайта	ACALL ad11	a ₁₀ a ₉ a ₈ 10001	6	2	2	(PC) = (PC) + 2, (SP) = (SP) + 1, ((SP)) = (PC ₀₋₇), (SP) = (SP) + 1, ((SP)) = (PC ₈₋₁₅), (PC ₀₋₁₀) = ad11
						(PC ₈₋₁₅) = ((SP)),

Возврат из подпрограммы	RET	00100010	1	1	2	$(SP) = (SP) - 1,$ $(PC_{0-7}) = ((SP)),$ $(SP) = (SP) - 1$
Возврат из подпрограммы обработки прерывания	RETI	00110010	1	1	2	$(PC_{8-15}) = ((SP)),$ $(SP) = (SP) - 1,$ $(PC_{0-7}) = ((SP)),$ $(SP) = (SP) - 1$
Холостая команда	NOP	00000000	1	1	1	$(PC) = (PC) + 1$
Примечание. Ассемблер допускает использование обобщенного имени команд JMP и CALL, которые в процессе трансляции заменяются оптимальными по формату командами вызова (ACALL, LCALL) или перехода (AJMP, SJMP, LJMP).						

Длинный переход. Переход по всему адресному пространству ПП. В команде содержится полный 16-битный адрес перехода (ad 16). Трех байтные команды длинного перехода содержат в мнемокоде букву L (Long). Всего существует две такие команды: LJMP - длинный переход и LCALL - длинный вызов подпрограммы. На практике редко возникает необходимость перехода в пределах всего адресного пространства и чаще используются укороченные команды перехода, занимающее меньше места в памяти.

Абсолютный переход. Переход в пределах одной страницы памяти программ размером 2048 байт. Такие команды содержат только 11 младших бит адреса перехода (ad 11). Команды абсолютного перехода имеют формат 2 байта. Начальная буква мнемокода - A (Absolute). При выполнении команды в вычисленном адресе следующей по порядку команды $((PC) = (PC) + 2)$ 11 младших бит заменяются на ad11 из тела команды абсолютного перехода.

Относительный переход. Короткий относительный переход позволяет передать управление в пределах -128 - +127 байт относительно адреса следующей команды (команды, следующей по порядку за командой относительного перехода). Существует одна команда безусловного короткого перехода SJMP (Short). Все команды условного перехода используют данный метод адресации. Относительный адрес перехода (rel) содержится во втором байте команды.

Косвенный переход. Команда JMP @A + DPTR позволяет передавать управление по косвенному адресу. Эта команда удобна тем, что предоставляет возможность организации перехода по адресу, вычисляемому самой программой и неизвестному при написании исходного текста программы.

Условные переходы. Развитая система условных переходов предоставляет возможность осуществлять ветвление по следующим условиям: аккумулятор содержит нуль (JZ); содержимое аккумулятора не равно нулю (JNZ); перенос равен единице (JC); перенос равен нулю (JNC); адресуемый бит равен единице (JB); адресуемый бит равен нулю (JNB).

Для организации программных циклов удобно пользоваться командой DJNZ, которая работает аналогично соответствующей команде МК48. Однако в качестве счетчика циклов в МК51 может использоваться не только регистр, но и прямоадресуемый байт (например, ячейка РПД).

Команда CJNE эффективно используется в процедурах ожидания какого-либо события. Например, команда

WAIT: CJNE A,P0,WAIT

будет выполняться до тех пор, пока на линиях порта 0 не установится информация, совпадающая с содержимым аккумулятора.

Все команды данной группы, за исключением CJNE и JBC, не оказывают воздействия на флаги. Команда CJNE устанавливает флаг C, если первый операнд оказывается меньше второго. Команда JBC сбрасывает флаг C в случае перехода.

Подпрограммы. Для обращения к подпрограммам необходимо использовать команды вызова подпрограмм (LCALL, ACALL). Эти команды в отличие от команд перехода (LJMP, AJMP) сохраняют в стеке адрес возврата в основную программу. Для возврата из подпрограммы необходимо выполнить команду RET. Команда RETI отличается от команды RET тем, что разрешает прерывания обслуженного уровня.

[Система команд МК51](#)

[Начало KM1816BE51](#)

4. Порты ввода/вывода информации

Все четыре порта МК51 предназначены для ввода или вывода информации побайтно. Схемотехника портов ввода/вывода МК51 для одного бита показана на [рис. 3.4](#) (порты 1 и 2 имеют примерно такую же структуру, как и порт 3). Каждый порт содержит управляемые регистр-защелку, входной буфер и выходной драйвер.

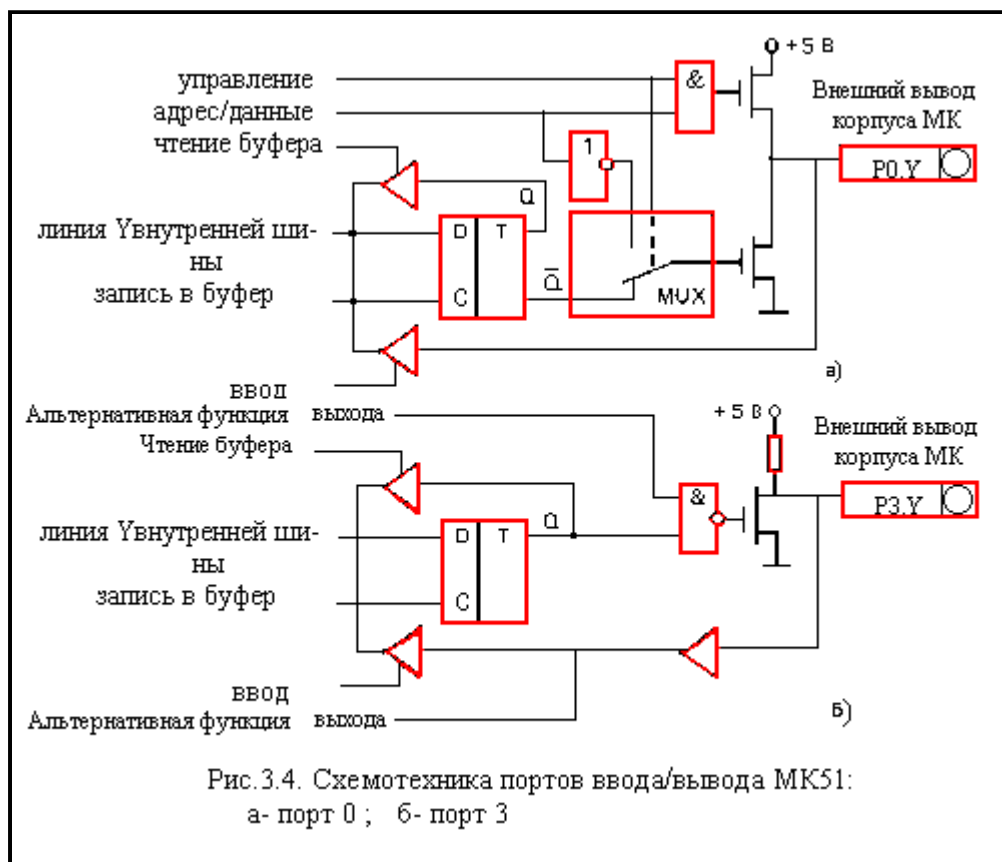


Рис. 3.4. Схематехника портов ввода/вывода МК51: а - порт 0; б - порт 3

Выходные драйверы портов 0 и 2, а также входной буфер порта 0 используются при обращении к внешней памяти (ВП). При этом через 1 порт 0 в режиме временного мультиплексирования сначала выводится младший байт адреса ВП, а затем выдается или принимается байт данных. Через порт 2 выводится старший байт адреса в тех случаях, когда 1 разрядность адреса равна 16 бит.

Все выходы порта 3 могут быть использованы для реализации альтернативных функций, перечисленных в [табл. 8.1](#). Альтернативные функции могут быть задействованы путем записи 1 в соответствующие биты регистра-защелки (P3.0-P3.7) порта 3.

Таблица 8.1. Альтернативные функции порта 3

Символ	Позиция	Имя и назначение
M_{RD}	P3.7	Чтение. Активный сигнал низкого уровня формируется аппаратно при обращении к ВПД
M_{WR}	P3.6	Запись. Активный сигнал низкого уровня формируется аппаратно при обращении к ВПД
T1	P3.5	Вход таймера/счетчика 1 или тест-вход
T0	P3.4	Вход таймера/счетчика 0 или тест-вход
M_{INT1}	P3.3	Вход запроса прерывания 1. Воспринимается сигнал низкого уровня или срез
M_{INT0}	P3.2	Вход запроса прерывания 0. Воспринимается сигнал низкого уровня или срез
TXD	P3.1	Выход передатчика последовательного порта в режиме УАПД. Выход синхронизации в режиме сдвигающего регистра
RXD	P3.0	Вход приемника последовательного порта в режиме УАПД. Ввод/вывод данных в режиме сдвигающего регистра

Порт 0 является двунаправленным, а порты 1, 2 и 3 - квазидвунаправленными. Каждая линия портов может быть использована независимо для ввода или вывода информации. Для того чтобы некоторая линия порта использовалась для ввода, в D-триггер регистра-защелки порта должна быть записана 1, которая закрывает МОП-транзистор выходной цепи.

По сигналу СБР в регистры-защелки всех портов автоматически записываются единицы, настраивающие их тем самым на режим ввода.

Все порты могут быть использованы для организации ввода/вывода информации по двунаправленным линиям передачи. Однако порты 0 и 2 не могут быть использованы для этой цели в случае, если МК-система имеет внешнюю память, связь с которой организуется через общую разделяемую шину адреса/данных, работающую в режиме временного мультиплексирования.

Запись в порт. При выполнении команды, которая изменяет содержимое регистра-защелки порта, новое значение фиксируется в регистре в момент S6P2 последнего цикла команды. Однако опрос содержимого регистра-защелки выходной схемой осуществляется во время фазы P1 и, следовательно, новое содержимое регистра-защелки появляется на выходных контактах порта только в момент S1P1 следующего машинного цикла.

Нагрузочная способность портов. Выходные линии портов 1, 2 и 3 могут работать на одну ТТЛ-схему. Линии порта 0 могут быть нагружены на два входа ТТЛ-схем каждая. Линии порта 0 могут работать и на n-МОП-схемы, однако при этом их необходимо подключать на источник электропитания через внешние нагрузочные резисторы за исключением случая, когда шина порта 0 используется в качестве шины адреса/данных внешней памяти.

Входные сигналы для МК51 могут формироваться ТТЛ-схемами или n-МОП-схемами. Допустимо использование в качестве источников сигналов для МК51 схем с открытым коллектором или открытым стоком. Однако при этом время изменения входного сигнала при переходе из 0 в 1 окажется сильно затянутым.

Особенности работы портов. Обращение к портам ввода/вывода возможно с использованием команд, оперирующих с байтом, отдельным битом и произвольной комбинацией бит. При этом в тех случаях, когда порт является одновременно операндом и местом назначения результата, устройство управления автоматически реализует специальный режим, который называется "чтение-модификация-запись". Этот режим обращения предполагает ввод сигналов не с внешних выводов порта, а из его регистра-защелки, что позволяет исключить неправильное считывание ранее выведенной информации.

Подобный механизм обращения к портам реализован в следующих командах:

ANL - логическое И, например **ANL P1, A**;

ORL - логическое ИЛИ, например **ORL P2, A**;

XRL - исключающее ИЛИ, например **XRL P3, A**;

JBC - переход, если в адресуемом бите единица, и последующий сброс бита, например **JBC P1.1, LABEL**; **CPL** - инверсия бита, например **CPL P3.3**;

INC - инкремент порта, например **INC P2**;

DEC - декремент порта, например **DEC P2**;

DJNZ - декремент порта и переход, если его содержимое не равно нулю, например **DJNZ P3, LABEL**;

MOV PX.Y, C - передача бита переноса в бит Y порта X;

SET PX.Y - установка бита Y порта X;

CLR PX.Y - сброс бита Y порта X.

Совсем не очевидно, что последние три команды в приведенном списке являются командами "чтение-модификация-запись". Однако это именно так. По этим командам сначала считывается байт из порта, а затем записывается новый байт в регистр-защелку.

Причиной, по которой команды "чтение-модификация-запись" обеспечивают отдельный доступ к регистру-защелке порта и к внешним выводам порта, является необходимость исключить возможность неправильного прочтения уровней сигналов на внешних выводах. Предположим для примера, что линия Y порта X соединяется с базой мощного транзистора и выходной сигнал на ней предназначен для его управления. Когда в данный бит записана 1, то транзистор включается. Если для проверки состояния исполнительного механизма (в нашем случае - мощного транзистора) прикладной программе требуется прочитать состояние выходного сигнала в том же бите порта, то считывание сигнала с внешнего вывода порта, а не из D-триггера регистра-защелки порта приведет к неправильному результату: единичный сигнал на базе транзистора имеет относительно низкий уровень и будет интерпретирован в МК как сигнал 0. Команды "чтение-модификация-запись" реализуют считывание из регистра-защелки, а не с внешнего вывода порта, что обеспечивает получение правильного значения 1.

На [рис. 3.5](#) приведены временные диаграммы, иллюстрирующие процесс выполнения операций ввода/вывода информации через порты МК51.

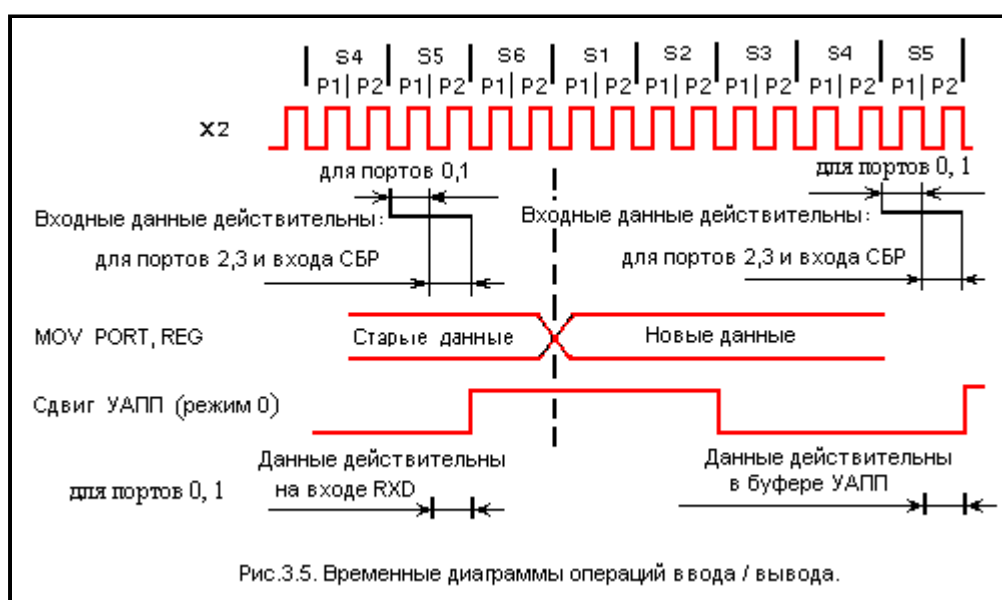


Рис. 3.5. Временные диаграммы операций ввода/вывода

[Порты ввода/вывода информации](#)

[Начало KM1816BE51](#)

5. Доступ к внешней памяти

В микроконтроллерных системах, построенных на основе МК51, возможно использование двух типов внешней памяти: постоянной памяти программ (ВПП) и оперативной памяти данных (ВПД). Доступ к ВПП осуществляется при помощи управляющего сигнала $M^{РВПП}$, который выполняет функцию строб-сигнала чтения. Доступ к ВПД обеспечивается управляющими сигналами $M^{ЧТ}$ и $M^{ЗП}$, которые формируются в линиях P3.7 и P3.6 при выполнении портом 3 альтернативных функций (см. [табл. 8.1](#)).

При обращении к ВПП всегда используется 16-битный адрес. Доступ к ВПД возможен с использованием 16-битного адреса ($MOVX A, @DPTR$) или 8-битного адреса ($MOVX A, @Ri$).

В любых случаях использования 16-битного адреса старший байт адреса фиксируется (и сохраняется неизменным в течение одного цикла записи или чтения) в регистре-защелке порта 2.

Если очередной цикл внешней памяти ($MOVX A, @DPTR$) следует не сразу же за предыдущим циклом внешней памяти, то неизменяемое содержимое регистра-защелки порта 2 восстанавливается в следующем цикле. Если используется 8-битный адрес ($MOVX A, @Ri$), то содержимое регистра-защелки порта 2 остается неизменным на его внешних выводах в течение всего цикла внешней памяти.

Через порт 0 в режиме временного мультиплексирования осуществляется выдача младшего байта адреса и передача байта данных. Сигнал $SABП$ должен быть использован для записи байта адреса во внешний регистр. Затем в цикле записи выводимый байт данных появляется на внешних выводах порта 0 только перед появлением сигнала $M^{ЗП}$. В цикле чтения вводимый байт данных принимается в порт 0 по фронту стробирующего сигнала $M^{ЧТ}$.

При любом обращении к внешней памяти устройство управления МК51 загружает в регистр-защелку порта 0 код 0FFH, стирая тем самым информацию, которая могла в нем храниться.

Доступ к ВПП возможен при выполнении двух условий: либо на вход отключения резидентной памяти программ ($M^{ОРПП}$) подается активный сигнал, либо содержимое счетчика команд превышает значение 0FFFH. Наличие сигнала $M^{ОРПП}$ необходимо для обеспечения доступа к младшим 4К адресам адресного пространства ВПП при использовании МК31 (микроконтроллера без резидентной памяти программ).

Временные диаграммы на [рис. 3.6](#) иллюстрируют процесс генерации управляющих сигналов $SABП$ и $РВПП$ при обращении к внешней памяти.

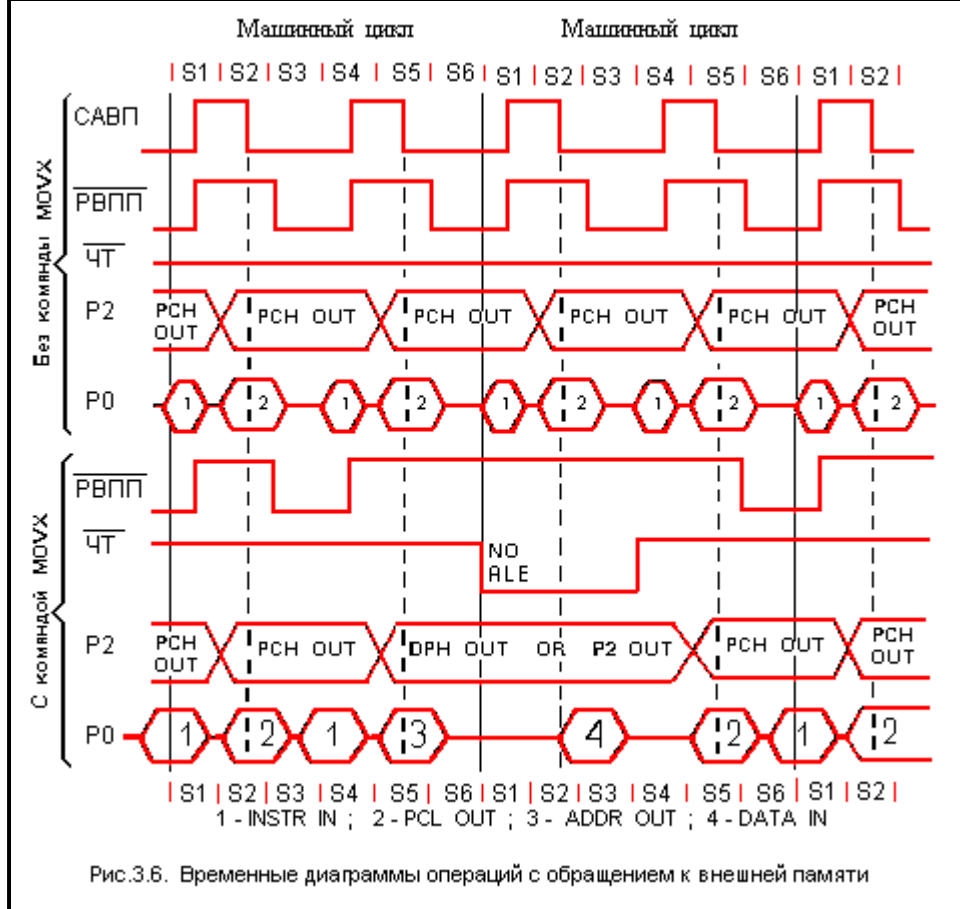


Рис.3.6. Временные диаграммы операций с обращением к внешней памяти

рис. 3.6. Временные диаграммы операций с обращением к внешней памяти

Основная функция сигнала САВП _ обеспечить временное согласование передачи из порта 0 на внешний регистр младшего байта адреса в цикле чтения из ВПП. Сигнал САВП приобретает значение 1 дважды в каждом машинном цикле. Это происходит даже тогда, когда в цикле выборки нет обращения к ВПП. Доступ к ВПД возможен только в том случае, если сигнал САВП отсутствует. Первый сигнал САВП во втором машинном цикле команды MOVX блокируется. Следовательно, в любой МК-системе, не использующей ВПД, сигнал САВП генерируется с постоянной частотой, равной 1/16 частоты резонатора, и может быть использован для синхронизации внешних устройств или для реализации различных временных функций.

При обращении к РПП сигнал RBPP не генерируется, а при обращении к ВПП он выполняет функцию строб-сигнала чтения. Полный цикл чтения ВПД, включая установку и снятие сигнала ЧТ, занимает 12 периодов резонатора.

Временные диаграммы на [рис. 3.7](#) и [рис. 3.8](#) иллюстрируют процесс выборки команды из ВПП и работу с ВПД в режимах чтения и записи соответственно.

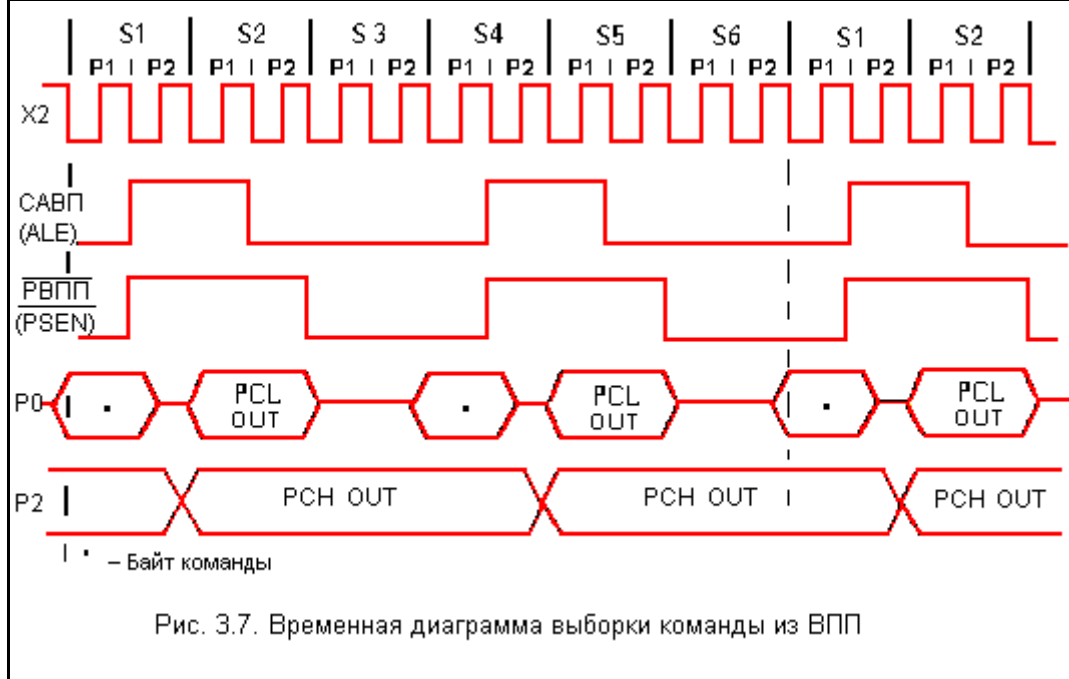


Рис. 3.7. Временная диаграмма выборки команды из ВПП

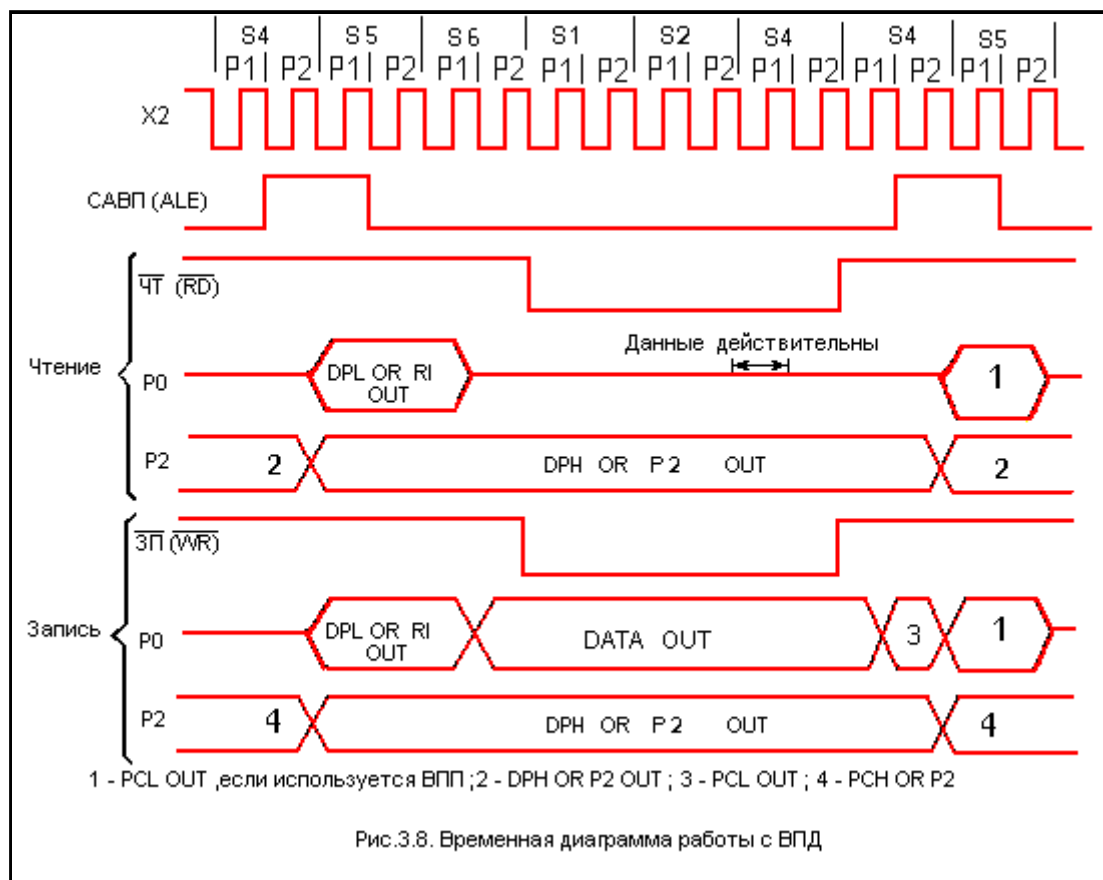


Рис. 3.8. Временная диаграмма работы с ВПД

Особый режим работы МК51. Содержимое памяти программ МК51 заполняется единожды на этапе разработки МК-системы и не может быть модифицировано в завершеном (конечном) изделии. По этой причине микроконтроллеры не являются машинами классической "фон-неймановской" архитектуры. Оперативная память данных (резидентная или внешняя) не может быть использована для хранения кодов программы, так как в МК выборка команд производится только из области адресов памяти программ. Эта особенность архитектуры МК объясняется тем, что в большинстве применений МК требуется наличие одной неизменяемой прикладной программы, хранимой в ПЗУ, наличие ОЗУ небольшой емкости для временного хранения переменных и эффективных, а следовательно, разных методов адресации памяти программ и памяти данных.

Однако на этапе разработки и отладки прикладных программ машина "фон-неймановского" типа оказывается очень удобной, так как позволяет разработчику оперативно изменять коды прикладной программы, размещаемой в ОЗУ. С этой целью МК-система может быть модифицирована для совмещения адресного пространства ВПП и ВПД путем подключения внешней логики, как показано на [рис. 3.9](#).

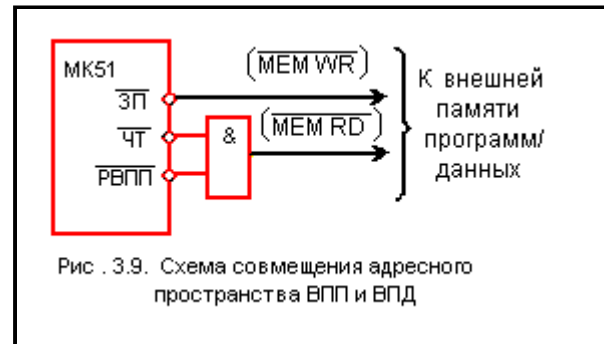


Рис. 3.9. Схема совмещения адресного пространства ВПП и ВПД

Здесь на выходе схемы И формируется строб-сигнал чтения, который может быть использован для объединения памяти программ и памяти данных во внешнем ОЗУ. При этом необходимо учитывать, что в МК51 на схемном уровне реализуются пять различных и независимых механизмов адресации для доступа к ПРР, РПД, ВПП, ВПД и блоку регистров специальных функций. Вследствие этого перемещаемая версия прикладной программы, которая отлаживается в среде внешней памяти программ/данных, будет отличаться от загружаемой в РП (окончательной) версии программы.

Подобный способ организации управления внешней памятью может быть использован в тех применениях МК51, где требуется оперативная перезагрузка или модификация прикладных программ (с помощью УВВ), как в ЭВМ классической архитектуры.

[Доступ к внешней памяти](#)

[Начало KM1816BE51](#)

6. Таймер/счетчик

Два программируемых 16-битных таймера/счетчика (Т/С0 и Т/С1) могут быть использованы в качестве таймеров или счетчиков внешних событий. При работе в качестве таймера содержимое Т/С инкрементируется в каждом машинном цикле, т.е. через каждые 12 периодов резонатора. При работе в качестве счетчика содержимое Т/С инкрементируется под воздействием перехода из 1 в 0 внешнего входного сигнала; подаваемого на соответствующий (Т0, Т1) вывод МК51. Опрос значения внешнего входного сигнала выполняется в момент времени S5P2 каждого машинного цикла. Содержимое счетчика будет увеличено на 1 в том случае, если в предыдущем цикле был считан входной сигнал высокого уровня (1), а в следующем _ сигнал низкого уровня (0). Новое (инкрементированное) значение счетчика будет сформировано в момент S3P1 в цикле, следующем за тем, в котором был обнаружен переход сигнала из 1 в 0. Так как на распознавание перехода требуется два машинных цикла, то максимальная частота подсчета входных сигналов равна 1/24 частоты резонатора. На длительность периода входных сигналов ограничений сверху нет. Для гарантированного прочтения входного считаемого сигнала он должен удерживать значение 1 как минимум течение одного машинного цикла МК51.

Для управления режимами работы Т/С и для организации взаимодействия таймеров с системой прерывания используются два регистра специальных функций (РРТС и РУСТ), описание которых приводите в [табл. 10.1](#) и [табл. 10.2](#) соответственно. Как следует из описания управляющих бит РРТС, для обоих Т/С режимы работы 0, 1 и 2 одинаковы. Режимы для Т/С0 и Т/С1 различны. Рассмотрим кратко работу Т/С во всех четырех режимах.

Таблица 10.1. Регистр режима работы таймера/счетчика

Символ	Позиция	Имя и назначение
GATE	TMOD.7 для Т/С1 и TMOD.3 для Т/С0	Управление блокировкой. Если бит установлен, то таймер/счетчик "х" разрешен до тех пор, пока на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен то Т/С разрешается, как только бит управления "TRx" устанавливается
C/M _T	TMOD.6 для Т/С1 и TMOD.2 для Т/С0	Бит выбора режима таймера или счетчика событий. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если бит установлен, то работает счетчик от внешних сигналов на входе "Tx"
M1	TMOD.5 для Т/С1 и TMOD.1 для Т/С0	Режим работы (см. примечание)
M0	TMOD.4 для Т/С1 и TMOD.0 для Т/С0	
Примечание		
M1	M0	Режим работы
0	0	Таймер МК48. "TLx" работает как 5-битный предделитель
0	1	16-битный таймер/счетчик. "ТНх" и "TLx" включены последовательно
1	0	8-битный автоперезагружаемый таймер/счетчик. "ТНх" хранит значение, которое должно быть перезагружено в "TLx" каждый раз по переполнению
1	1	Таймер/счетчик 1 останавливается. Таймер/счетчик 0: TL0 работает как 8-битный таймер/счетчик, и его режим определяется управляющими битами таймера 0. ТН0 работает только как 8-битный таймер, и его режим определяется управляющими битами таймера 1

Таблица 10.2. Регистр управления/статуса таймера

Символ	Позиция	Имя и назначение
TF1	TCON.7	Флаг переполнения таймера 1. Устанавливается аппаратно при переполнении таймера/счетчика. Сбрасывается при обслуживании прерывания аппаратно
TR1	TCON.6	Бит управления таймера 1. Устанавливается/сбрасывается программой для пуска/останова
TF0	TCON.5	Флаг переполнения таймера 0. Устанавливается аппаратно. Сбрасывается при обслуживании прерывания
TR0	TCON.4	Бит управления таймера 0. Устанавливается/сбрасывается программой для пуска/останова таймера/счетчика
IE1	TCON.3	Флаг фронта прерывания 1. Устанавливается аппаратно, когда детектируется срез внешнего сигнала M3PP1 (MINT1). Сбрасывается при обслуживании прерывания
IT1	TCON.2	Бит управления типом прерывания 1. Устанавливается/сбрасывается программно для

		спецификации запроса $M_{ЗПР1}$ (срез/низкий уровень)
IE0	TCON.1	Флаг фронта прерывания 0. Устанавливается по срезу сигнала $M_{ЗПР0}$. Сбрасывается при обслуживании прерывания
IT0	TCON.0	Бит управления типом прерывания 0. Устанавливается/сбрасывается программно для спецификации запроса $M_{ЗПР0}$ (срез/низкий уровень)

Режим 0. Перевод любого Т/С в режим 0 делает его похожим на таймер МК48 (8-битный счетчик), на вход которого подключен 5-битный предделитель частоты на 32. Работу Т/С в режиме 0 на примере Т/С1 иллюстрирует [рис. 3.10, а](#). В этом режиме таймерный регистр имеет разрядность 13 бит. При переходе из состояния "все единицы" в состояние "все нули" устанавливается флаг прерывания от таймера TF1. Входной синхросигнал таймера 1 разрешен (поступает на вход Т/С), когда управляющий бит TR1 установлен в 1 и либо управляющий бит GATE (блокировка) равен 0, либо на внешний вывод запроса прерывания M_{INT1} поступает уровень 1.

Отметим попутно, что установка бита GATE в 1 позволяет использовать таймер для измерения длительности импульсного сигнала, подаваемого на вход запроса прерывания.

Режим 1. Работа любого Т/С в режиме 1 такая же, как и в режиме 0, за исключением того, что таймерный регистр имеет разрядность 16 бит.

Режим 2. В режиме 2 работа организована таким образом, что переполнение (переход из состояния "все единицы" в состояние "все нули") 8-битного счетчика TL1 приводит не только к установке флага TF1 ([рис. 3.10, б](#)), но и автоматически перезагружает в TL1 содержимое старшего байта (TH1) таймерного регистра, которое предварительно было задано программным путем. Перезагрузка оставляет содержимое TH1 неизменным. В режиме 2 Т/С0 и Т/С1 работают совершенно одинаково.

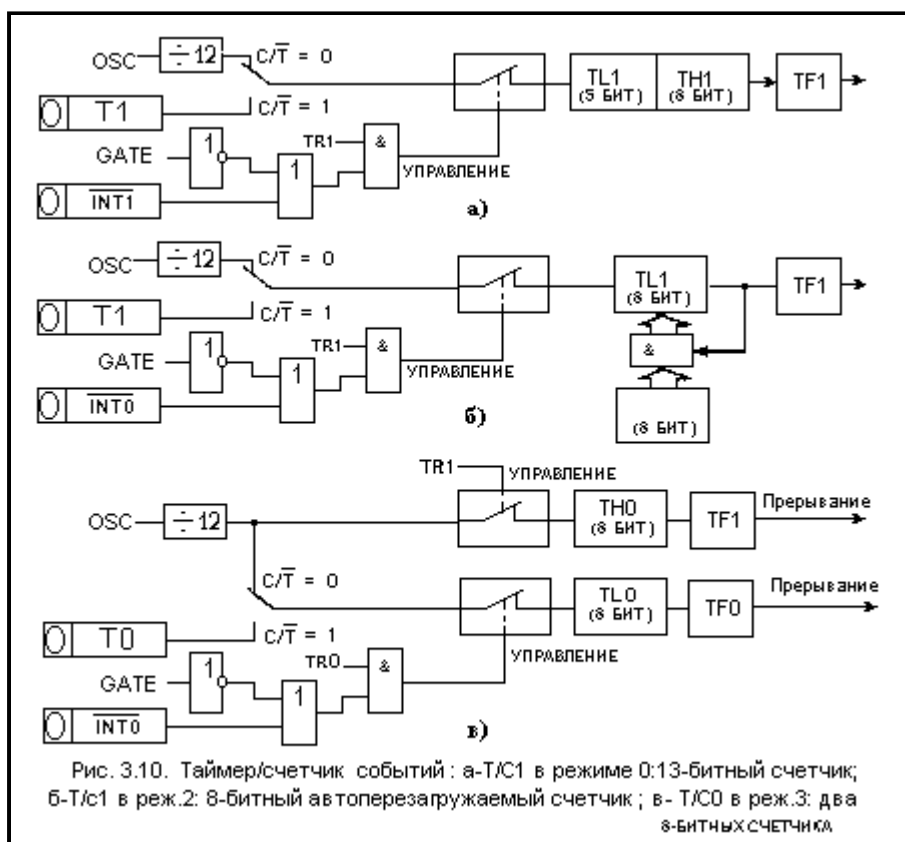


Рис. 3.10. Таймер/счетчик событий: а - Т/С1 в режиме 0: 13-битный счетчик; б - Т/С1 в режиме 2: 8-битный автоперезагружаемый счетчик; в - Т/С0 в режиме 3: два 8-битных счетчика

Режим 3. В режиме 3 Т/С0 и Т/С1 работают по-разному. Т/С1 сохраняет неизменным свое текущее содержимое. Иными словами, эффект такой же, как и при сбросе управляющего бита TR1 в нуль.

Работу Т/С0 в режиме 3 иллюстрирует [рис. 3.10, в](#). В режиме 3 TL0 и TH0 функционируют как два независимых

http://www.msclub.ce.cctpu.edu.ru/MCU_MPU/8051/timer_counter_mk51.htm 13.03.2003 10:39:16

8-битных счетчика. Работу TL0 определяют управляющие биты T/C0 (C/MT, GATE, TR0), входной сигнал MINT0 и флаг переполнения TF0, Работу TH0, который может выполнять только функции таймера (подсчет машинных циклов МК), определяет управляющий бит TR1. При этом TH0 использует флаг переполнения TF1.

Режим 3 используется в тех случаях применения МК51, когда требуется наличие дополнительного 8-битного таймера или счетчика событий. Можно считать, что в режиме 3 МК51 имеет в своем составе три таймера/счетчика. В том случае, если T/C0 используется в режиме 3, T/C1 может быть или включен, или выключен, или переведен в свой собственный режим 3, или может быть использован последовательным портом в качестве генератора частоты передачи, или, наконец, может быть использован в любом применении, не требующем прерывания.

[Таймер/счетчик](#)

[KM1816BE51](#)

7. Последовательный интерфейс

- [7.1. Регистр управления/статуса УАПП](#)
- [7.2. Работа УАПП в мультимикроконтроллерных системах](#)
- [7.3. Скорость приема/передачи](#)
- [7.4. Особенности работы УАПП в различных режимах](#)

Через универсальный асинхронный приемопередатчик (УАПП) осуществляется прием и передача информации, представленной последовательным кодом (младшими битами вперед), в полном дуплексном режиме обмена. В состав УАПП, называемого часто последовательным портом, входят принимающий и передающий сдвигающие регистры, а также специальный буферный регистр (SBUF) приемопередатчика. Запись байта в буфер приводит к автоматической переписи байта в сдвигающий регистр передатчика и инициирует начало передачи байта. Наличие буферного регистра приемника позволяет совмещать операцию чтения ранее принятого байта с приемом очередного байта. Если к моменту окончания приема байта предыдущий байт не был считан из SBUF, то он будет потерян.

Последовательный порт МК51 может работать в четырех различных режимах.

Режим 0. В этом режиме информация и передается и принимается через внешний вывод входа приемника (RXD). Принимаются или передаются 8 бит данных. Через внешний вывод выхода передатчика (TXD) выдаются импульсы сдвига, которые сопровождают каждый бит. Частота передачи бита информации равна 1/12 частоты резонатора.

Режим 1. В этом режиме передаются через TXD или принимаются из RXD 10 бит информации: старт-бит (0), 8 бит данных и стоп-бит (1). Скорость приема/передачи — величина переменная и задается таймером.

Режим 2. В этом режиме через TXD передаются или из RXD принимаются 11 бит информации: старт-бит, 8 бит данных, программируемый девятый бит и стоп-бит. При передаче девятый бит данных может принимать значение 0 или 1, или, например, для повышения достоверности передачи путем контроля по четности в него может быть помещено значение признака паритета из слова состояния программы (PSW.0). Частота приема/передачи выбирается программой и может быть равна либо 1/32, либо 1/64 частоты резонатора в зависимости от управляющего бита SMOD.

Режим 3. Режим 3 совпадает с режимом 2 во всех деталях, за исключением частоты приема/передачи, которая является величиной переменной и задается таймером.

[Последовательный интерфейс](#)

7.1. Регистр управления/статуса УАПП

Управление режимом работы УАПП осуществляется через специальный регистр с символическим именем SCON. Этот регистр содержит не только управляющие биты, определяющие режим работы последовательного порта, но и девятый бит принимаемых или передаваемых данных (RB8 и TB8) и биты прерывания приемопередатчика (RI и TI).

Функциональное назначение бит регистра управления/статуса УАПП приводится в [табл. 11.1](#).

Прикладная программа путем загрузки в старшие биты спецрегистра SCON 2-битного кода определяет режим работы УАПП. Во всех четырех режимах работы передача из УАПП инициируется любой командой, в которой буферный регистр SBUF указан как получатель байта. Прием в УАПП в режиме 0 осуществляется при условии,

что RI = 0 и REN = 1. В режимах 1, 2, 3 прием начинается с приходом старт-бита, если REN = 1.

В бите TB8 программно устанавливается значение девятого бита данных, который будет передан в режиме 2 или 3. В бите RB8 фиксируется в режимах 2 и 3 девятый принимаемый бит данных. В режиме 1, если SM2 = 0, в бит RB8 заносится стоп-бит. В режиме 0 бит RB8 не используется.

Флаг прерывания передатчика TI устанавливается аппаратурно в конце периода передачи восьмого бита данных в режиме 0 и в начале периода передачи стоп-бита в режимах 1,2 и 3. Соответствующая подпрограмма обслуживания прерывания должна сбрасывать бит TI.

Флаг прерывания приемника RI устанавливается аппаратурно в конце периода приема восьмого бита данных в режиме 0 и в середине периода приема стоп-бита в режимах 1, 2 и 3. Подпрограмма обслуживания прерывания должна сбрасывать бит RI.

Таблица 11.1. Регистр управления/статуса УАПП

Символ	Позиция	Имя и назначение
SM0	SCON.7	Биты управления режимом работы УАПП. Устанавливаются/ сбрасываются программно (см. примечание)
SM1	SCON.6	
SM2	SCON.5	Бит управления режимом УАПП. Устанавливается программно для запрета приема сообщения, в котором девятый бит имеет значение 0
REN	SCON.4	Бит разрешения приема. Устанавливается/сбрасывается программно для разрешения/запрета приема последовательных данных
TB8	SCON.3	Передача бита 8. Устанавливается/сбрасывается программно для задания девятого передаваемого бита в режиме УАПП-9 бит
RB8	SCON.2	Прием бита 8. Устанавливается/сбрасывается аппаратурно для фиксации девятого принимаемого бита в режиме УАПП-9 бит
TI	SCON.1	Флаг прерывания передатчика. Устанавливается аппаратурно при окончании передачи байта. Сбрасывается программно после обслуживания прерывания
RI	SCON.0	Флаг прерывания приемника. Устанавливается аппаратурно при приеме байта. Сбрасывается программно после обслуживания прерывания
Примечание		
SM0	SM1	Режим работы УАПП
0	0	Сдвигающий регистр расширения ввода/вывода
0	1	УАПП-8 бит. Изменяемая скорость передачи
1	0	УАПП-9 бит. Фиксированная скорость передачи
1	1	УАПП-9 бит. Изменяемая скорость передачи

[Последовательный интерфейс](#)

7.2. Работа УАПП в мультимикроконтроллерных системах

В системах децентрализованного управления, которые используются для управления и регулирования в топологически распределенных объектах (например, прокатных станах, электроподвижном составе железных дорог и метрополитена, сборочных конвейерах и линиях гибких автоматизированных производств), возникает задача обмена информацией между множеством микроконтроллеров, объединенных в локальную вычислительно-управляющую сеть. Как правило, локальные сети на основе МК51 имеют магистральную архитектуру с разделяемым моноканалом (коаксиальный кабель, витая пара, оптическое волокно), по которому осуществляется обмен информацией между МК.

В регистре специальных функций SCON микроконтроллера имеется управляющий бит SM2, который в режимах 2 и 3 УАПП позволяет относительно простыми средствами реализовать межконтроллерный обмен информацией в локальных управляющих сетях.

Механизм межконтроллерного обмена информацией через последовательный порт МК51 построен на том, что в режимах 2 и 3 программируемый девятый бит данных при приеме фиксируется в бите RB8. УАПП может быть запрограммирован таким образом, что при получении стоп-бита прерывание от приемника будет возможно только при условии $RB8 = 1$. Это выполняется установкой управляющего бита SM2 в регистре SCON.

Поясним процесс межконтроллерного обмена информацией на примере. Пусть ведущему МК требуется передать блок данных некоторому (или нескольким) ведомому МК. С этой целью ведущий МК в протокольном режиме "широковещательной" передачи (всем ведомым МК) выдает в моноканал байт-идентификатор абонента (код адреса МК-получателя), который отличается от байтов данных только тем, что в его девятом бите содержится 1. Программа реализации протокола сетевого обмена информацией должна быть построена таким образом, чтобы при получении Райта-идентификатора ($RB8 = 1$) во всех ведомых МК произошли прерывание прикладных программ и вызов подпрограммы сравнения байта-идентификатора с кодом собственного сетевого адреса. Адресуемый МК сбрасывает свой управляющий бит SM2 и готовится к приему блока данных. Остальные ведомые МК, адрес которых не совпал с кодом байта-идентификатора, оставляют неизменным состояние $SM2 = 1$ и передают управление основной программе. При $SM2 = 1$ информационные байты, передаваемые по моноканалу и поступающие в УАПП ведомых МК, прерывания не вызывают, т.е. игнорируются.

В режиме 1 УАПП автономного МК управляющий бит SM2 используется для контроля истинности стоп-бита (при $SM2 = 1$ прерывание не произойдет до тех пор, пока не будет получено истинное (единичное) значение стоп-бита). В режиме 0 бит SM2 не используется и должен быть сброшен.

[Последовательный интерфейс](#)

7.3. Скорость приема/передачи

Скорость приема/передачи, т.е. частота работы УАПП в различных режимах, определяется различными способами.

В режиме 0 частота передачи зависит только от резонансной частоты кварцевого резонатора $f_0 = f_{рез}/12$. За один машинный цикл последовательный порт передает один бит информации.

В режимах 1, 2 и 3 скорость приема/передачи зависит от значения управляющего бита SMOD в регистре специальных функций РУМ ([табл. 11.2](#)).

В режиме 2 частота передачи определяется выражением $f_2 = (2^{MOD}/64) * f_{рез}$ иными словами, при $SMOD = 0$ частота передачи равна $(1/64) * f_{рез}$, а при $SMOD = 1$ равна $(1/32) * f_{рез}$.

Таблица 11.2. Регистр управления мощностью РУМ

Символ	Позиция	Имя и назначение
SMOD	PCON.7	Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD = 0
----	PCON.6	Не используются
----	PCON.5	
----	PCON.4	
GF1	PCON.3	Флаги, специфицируемые пользователем (флаги общего назначения)
GF0	PCON.2	
PD	PCON.1	Бит пониженной мощности. При установке бит а в 1 МК переходит в режим пониженной потребляемой мощности
IDL	PCON.0	Бит холостого хода. Если бит установлен в 1, то МК переходит в режим холостого хода

Примечание. При одновременной записи 1 в PD и IDL бит PD имеет преимущество. Сброс содержимого РУМ выполняется

В режимах 1 и 3 в формировании частоты передачи кроме управляющего бита SMOD принимает участие таймер 1. При этом частота передачи зависит от частоты переполнения (OVT1) и определяется следующим образом:

$$f_{1,3} = (2^{SMOD}/32) * f_{OVT1}$$

Прерывание от таймера 1 в этом случае должно быть заблокировано. Сам T/C1 может работать и как таймер, и как счетчик событий в любом из трех режимов. Однако наиболее удобно использовать режим таймера с автоперезагрузкой (старшая тетрада TMOD = 0010B). При этом частота передачи определяется выражением

$$f_{1,3} = (2^{SMOD}/32) * (f_{рез}/12) / (256 - (TH1))$$

В [табл. 11.3](#) приводится описание способов настройки T/C1 для получения типовых частот передачи данных через УАПП.

Таблица 11.3. Настройка таймера 1 для управления частотой работы УАПП

Частота приема/передачи (BAUD RATE)	Частота резонатора, МГц	SMOD	Таймер/счетчик 1		
			C/T	Режим (MODE)	Перезагружаемое число
Режим 0, макс: 1 МГц	12	X	X	X	X
Режим 2, макс: 375 кГц	12	1	X	X	X
Режимы 1, 3: 62.5 кГц	12	1	0	2	0FFH
19.2 кГц	11.059	1	0	2	0FDH
9.6 кГц	11.059	0	0	2	0FDH
4.8 кГц	11.059	0	0	2	0FAH
2.4 кГц	11.059	0	0	2	0F4H
1.2 кГц	11.059	0	0	2	0E8H
137.5 Гц	11.059	0	0	2	1DH
110 Гц	6	0	0	2	72H
110 Гц	12	0	0	1	0FE6BH

[Последовательный интерфейс](#)

7.4. Особенности работы УАПП в различных режимах

Режим 0. На [рис. 3.11](#) показаны упрощенная структурная схема УАПП и временная диаграмма его работы в режиме 0. Данные передаются и принимаются через вывод RXD. Через вывод TXD выдаются синхросигналы сдвига.

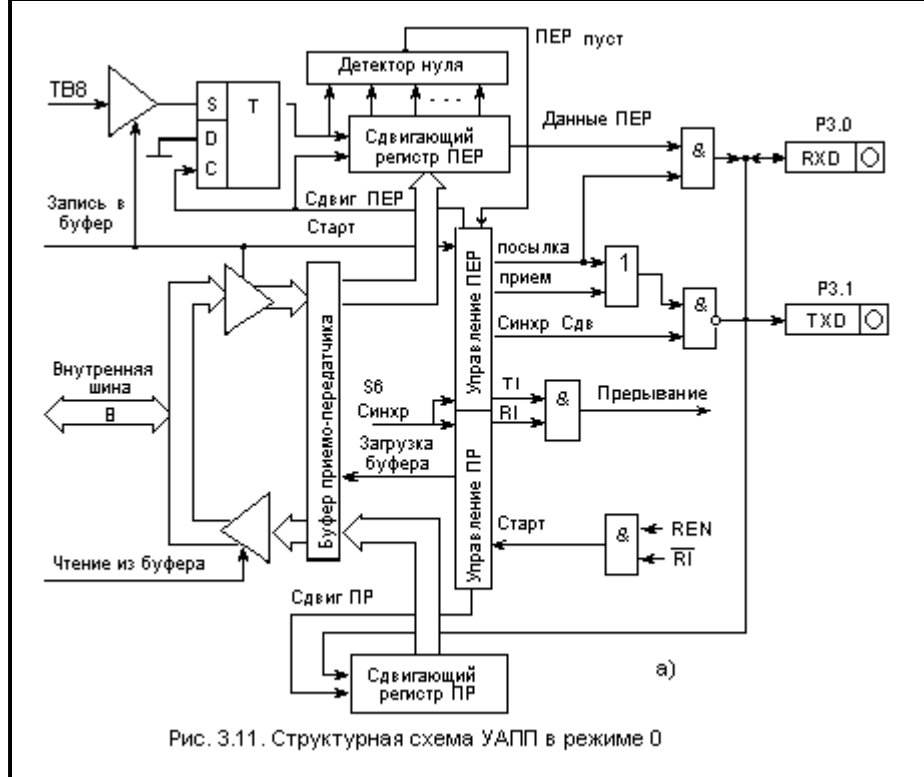


Рис. 3.11. Структурная схема УАПШ в режиме 0

Рис 3.11 а) Структурная схема УАПШ в режиме 0

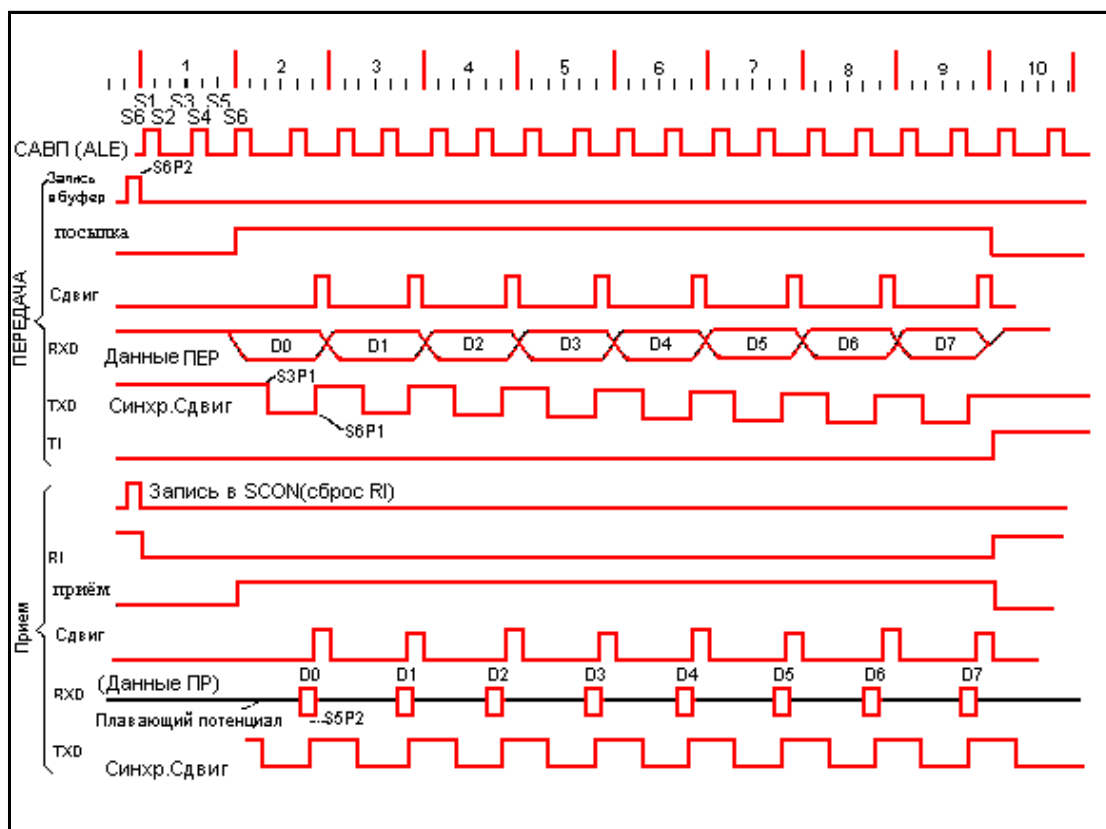


Рис 3.11. б) Временная диаграмма работы УАПШ в режиме 0

Передача начинается любой командой, по которой в SBUF поступает байт данных. В момент времени S6P2 устройство управления МК51 по сигналу *Запись в буфер* записывает байт в сдвигающий регистр передатчика, устанавливает триггер девятого бита и запускает блок управления передачей, который через один машинный цикл вырабатывает разрешающий сигнал *Посылка*. При этом в момент S6P2 каждого машинного цикла содержимое сдвигающего регистра сдвигается вправо (младшими битами вперед) и поступает на вывод RXD. В освобождающиеся старшие биты сдвигающего регистра передатчика записываются нули. При получении от детектора нуля сигнала *Передатчик пуст* блок управления передатчиком снимает сигнал *Посылка* и устанавливает флаг TI (момент S1P1 десятого машинного цикла после поступления сигнала *Запись в буфер*).

Прием начинается при условии $REN = 1$ и $RI = 0$. В момент S6P2 следующего машинного цикла блок управления приемником формирует разрешающий сигнал **Прием**, по которому на выход TXD передаются синхросигналы сдвига и в сдвигающем регистре приемника начинают формироваться значения бит данных, которые считываются с входа RXD в моменты S5P2 каждого машинного цикла. В момент S1P1 десятого машинного цикла после сигнала **Запись в SCON** блок управления приемником переписывает содержимое сдвигающего регистра в буфер, снимает разрешающий сигнал **Прием** и устанавливает флаг RI.

Режим 1. На рис. 3.12 показаны структурная схема (рис. 3.12, а) и временные диаграммы работы УАПП при приеме и передаче данных (рис. 3.12, б, в). Через вывод TXD УАПП передает, а с вывода RXD принимает 10 бит: старт-бит (0), 8 бит данных и стоп-бит (1). При приеме стоп-бит поступает в бит RB8 регистра SCON.

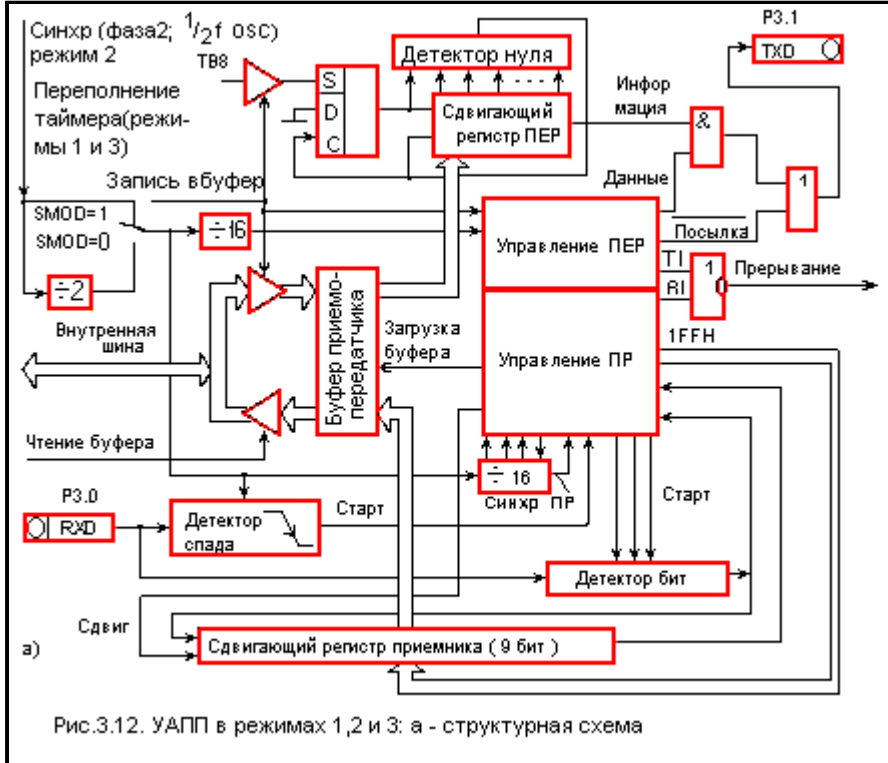


Рис 3.12 а) Структурная схема УАПП в режимах 1, 2 и 3

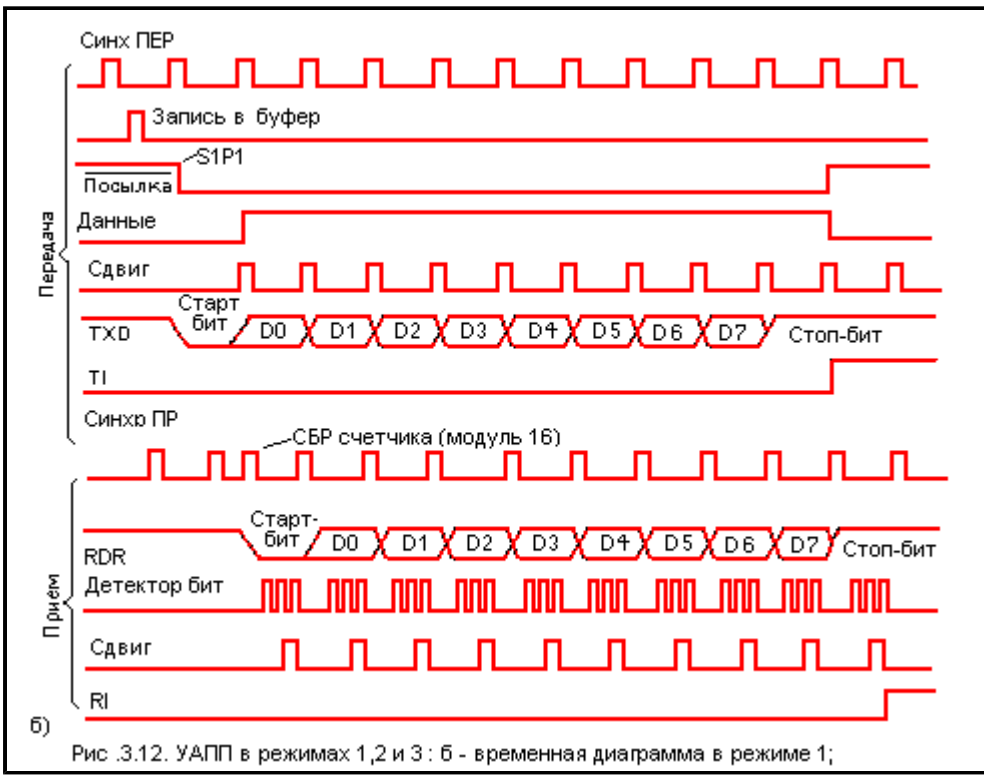


Рис 3.12 б) Временная диаграмма работы УАПП в режиме 1

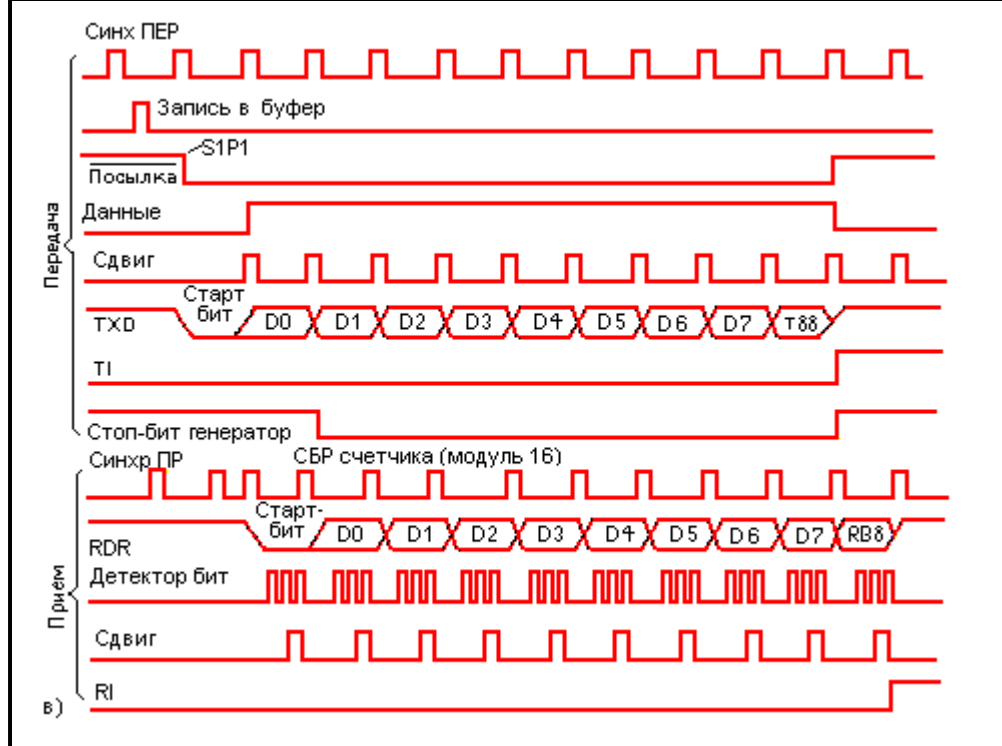


Рис 3.12 в) Временная диаграмма работы УАПП в режимах 2 и 3

Передача инициируется любой командой, в которой получателем байта является регистр SBUF. Генерируемый при этом управляющий сигнал *Запись в буфер* загружает 1 в девятый бит сдвигающего регистра передатчика, запускает блок управления передачей и в момент времени S1P1 формирует разрешающий сигнал *Посылка*. По этому сигналу на вывод TXD сначала поступает старт-бит, а затем (по разрешающему сигналу *Данные*) биты данных. Каждый период передачи бита равен 16 тактам внутреннего счетчика.

Прием начинается при обнаружении перехода сигнала на входе RXD из состояния 1 в состояние 0. Для этого под управлением внутреннего счетчика вход RXD опрашивается 16 раз за период представления бита. Как только переход из 1 в 0 на входе RXD обнаружен, в сдвигающий регистр приемника загружается код 1FFH, внутренний счетчик по модулю 16 немедленно сбрасывается и перезапускается для выравнивания его переходов с границами периодов представления принимаемых бит. Таким образом, каждый период представления бита делится на 16 периодов внутреннего счетчика. В состояниях 7, 8 и 9 счетчика в каждом периоде представления бита производится опрос сигнала на входе RXD. Считанное значение принимаемого бита — это то, которое было получено по меньшей мере дважды из трех замеров (мажоритарное голосование по принципу "два из трех"). Если значение, принятое в первом такте, не равно 0, то блок управления приемом вновь возвращается к поиску перехода из 1 в 0. Этот механизм обеспечивает подавление ложных (сбойных) старт-бит. Истинный старт-бит сдвигается в регистре приемника, и продолжается прием остальных бит посылки. Блок управления приемом сформирует сигнал *Загрузка буфера*, установит RB8 и флаг RI только в том случае, если в последнем такте сдвига выполняются два условия: бит RI = 0, и либо SM2 = 0, либо принятый стоп-бит равен 1. Если одно из этих двух условий не выполняется, то принятая последовательность бит теряется. В это время вне зависимости от того, выполняются указанные условия или нет, блок управления приемом вновь начинает отыскивать переход из 1 в 0 на входе RXD.

Режимы 2, 3. Через вывод TXD УАПП передает или с вывода RXD принимает 11 бит: старт-бит (0), 8 бит данных, программируемый девятый бит и стоп-бит (1). На временной диаграмме (рис. 3.12, в) показана работа УАПП при передаче и приеме данных в режимах 2 и 3. Как видно, режимы 2 и 3 отличаются от режима 1 только наличием девятого программируемого бита. Вследствие этого несколько изменяются условия окончания цикла приема: блок управления приемником сформирует управляющий сигнал *Загрузка буфера*, загрузит RB8 и установит флаг RI только в том случае, если в последнем такте сдвига выполняются два условия: бит RI = 0 и либо SM2 = 0, либо значение принятого девятого бита данных равно 1.

8. Система прерываний

Упрощенная схема прерываний МК51 показана на [рис. 3.13](#).

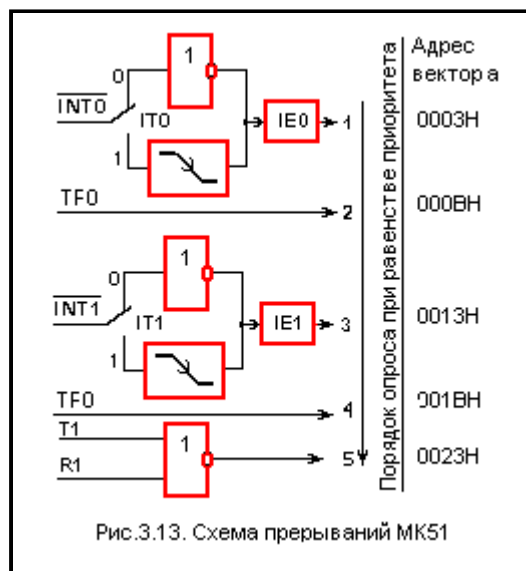


Рис. 3.13. Схема прерываний МК51

Внешние прерывания $MINT0$ и $MINT1$ могут быть вызваны либо уровнем, либо переходом сигнала из 1 в 0 на входах МК51 в зависимости от значений управляющих бит $IT0$ и $IT1$ в регистре $TCON$. От внешних прерываний устанавливаются флаги $IE0$ и $IE1$ в регистре $TCON$, которые инициируют вызов соответствующей подпрограммы обслуживания прерывания. Сброс этих флагов выполняется аппаратурно только в том случае, если прерывание было вызвано по переходу (срезу) сигнала. Если же прерывание вызвано уровнем входного сигнала, то сбросом флага IE управляет соответствующая подпрограмма обслуживания прерывания путем воздействия на источник прерывания с целью снятия им запроса.

Флаги запросов прерывания от таймеров $TF0$ и $TF1$ сбрасываются автоматически при передаче управления подпрограмме обслуживания. Флаги запросов прерывания RI и TI устанавливаются блоком управления УАПП аппаратурно, но сбрасываться должны программой.

Прерывания могут быть вызваны или отменены программой, так как все перечисленные флаги программно-доступны и могут быть установлены/сброшены программой с тем же результатом, как если бы они были установлены/сброшены аппаратурными средствами.

В блоке регистров специальных функций есть два регистра, предназначенных для управления режимом прерываний и уровнями приоритета. Форматы этих регистров, имеющих символические имена IE и IP описаны в [табл. 12.1](#) и [табл. 12.2](#) соответственно. Возможность программной установки/сброса любого управляющего бита в этих двух регистрах делает систему прерываний МК51 исключительно гибкой.

Таблица 12.1. Регистр масок прерывания (РМП)

Символ	Позиция	Имя и назначение
EA	IE.7	Снятие блокировки прерываний. Сбрасывается программно для запрета всех прерываний независимо от состояний IE4 - IE0
----	IE.6	Не используются
----	IE.5	
ES	IE.4	Бит разрешения прерывания от УАПП. Установка/сброс программой для разрешения/запрета прерываний от флагов TI или RI
ET1	IE.3	Бит разрешения прерывания от таймера 1. Установка/сброс программой для разрешения/запрета прерываний от таймера 1
EX1	IE.2	Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерываний
ET0	IE.1	Бит разрешения прерывания от таймера 0. Работает аналогично IE.3

Таблица 12.2. Регистр приоритетов прерываний

Символ	Позиция	Имя и назначение
----	IP.7 - IP.5	Не используются
PS	IP.4	Бит приоритета УАПШ. Установка/сброс программой для присваивания прерыванию от УАПШ высшего/низшего приоритета
PT1	IP.3	Бит приоритета таймера 1. Установка/сброс программой для присваивания прерыванию от таймера 1 высшего/низшего приоритета
PX1	IP.2	Бит приоритета внешнего прерывания 1. Установка/сброс программой для присваивания высшего/низшего приоритета внешнему прерыванию ^M INT1
PT0	IP.1	Бит приоритета таймера 0. Работает аналогично IP.3
PX0	IP.0	Бит приоритета внешнего прерывания 0. Работает аналогично IP.2

Флаги прерываний опрашиваются в момент S5P2 каждого машинного цикла. Ранжирование прерываний по уровню приоритета выполняется в течение следующего машинного цикла. Система прерываний сформирует аппаратно вызов (LCALL) соответствующей подпрограммы обслуживания, если она не заблокирована одним из следующих условий:

- 1) в данный момент обслуживается запрос прерывания равного или более высокого уровня приоритета;
- 2) текущий машинный цикл _ не последний в цикле выполняемой команды;
- 3) выполняется команда RETI или любая команда, связанная с обращением к регистрам IE или IP.

Отметим, что если флаг прерывания был установлен, но по одному из перечисленных выше условий не получил обслуживания и к моменту окончания блокировки уже был сброшен, то запрос прерывания теряется и нигде не запоминается.

По аппаратно-сформированному коду LCALL система прерывания помещает в стек только содержимое счетчика команд (PC) и загружает в счетчик команд адрес вектора соответствующей подпрограммы обслуживания. По адресу вектора должна быть расположена команда безусловной передачи управления (JMP) к начальному адресу подпрограммы обслуживания прерывания. Подпрограмма обслуживания в случае необходимости должна начинаться командами записи в стек (PUSH) спора состояния программы (PSW), аккумулятора, расширителя, указателя данных и т.д. и заканчиваться командами восстановления из стека (POP). Подпрограммы обслуживания прерывания обязательно завершаются командой RETI, по которой в счетчик команд перезагружается из стека сохраненный адрес возврата в основную программу. Команда RET также возвращает управление прерванной основной программе, но при этом не снимает блокировку прерываний, что приводит к необходимости иметь программный механизм анализа окончания процедуры обслуживания данного прерывания.

[Система прерываний](#)

[Начало KM1816BE51](#)

9. Особые режимы работы МК51

- [9.1. Режим загрузки и верификации прикладных программ](#)
- [9.2. Работа МК51 в пошаговом режиме](#)
- [9.3. Сброс, режим холостого хода и режим пониженного энергопотребления](#)

9.1. Режим загрузки и верификации прикладных программ

Под воздействием внешних электрических сигналов МК51 может быть электрически запрограммирован или, иными словами, в РПП микроконтроллера могут быть загружены объектные коды прикладной программы. Содержимое РПП микроконтроллера может быть уничтожено выдержкой под ультрафиолетовым источником света (стирание) для последующего перепрограммирования. Микроконтроллер имеет средство защиты от программного "разбоя", обеспечивающее невозможность прочтения содержимого РПП в конечном изделии и, следовательно, сохранение профессиональных секретов разработчика прикладного программного обеспечения.

Загрузка программ в РПП. В режиме программирования МК51 должен работать на пониженной частоте (с резонатором 4-6 МГц) из-за необходимости мультиплексирования на внутренней шине адресной и кодовой информации. На [рис. 3.14](#) приведена схема подключения МК51 к программатору.

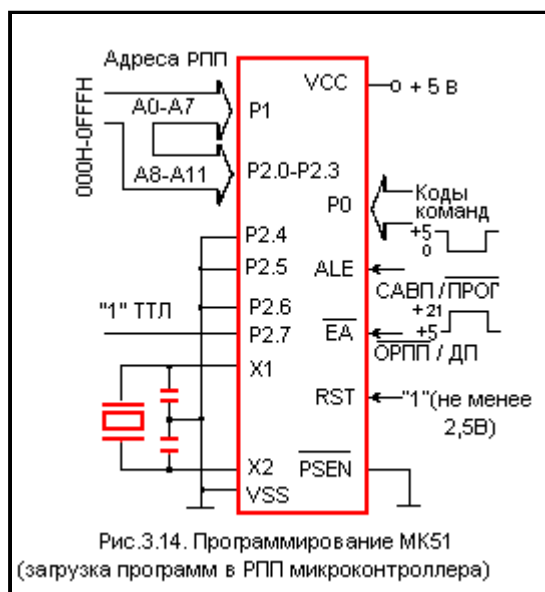


Рис. 3.14. Программирование МК51 (загрузка в РПП микроконтроллера)

Адрес ячейки РПП, в которую должен быть загружен байт прикладной программы, подается на выводы порта 1 и выводы P2.0-P2.3 порта 2. При этом загружаемый байт поступает в МК через порт 0. Выводы P2.4-P2.6 и \overline{PSEN} должны быть заземлены, а на выводы P2.7 и RST необходимо подать уровень логической 1 (для входа RST уровень логической 1 _ не менее 2.5 В, для остальных входов _ стандартный уровень ТТЛ). На входе \overline{EA}/VPP поддерживается уровень +5 В, но в момент загрузки байта он должен быть подключен к источнику напряжения с уровнем +21 В. В это время уровень на входе ALE/ $\overline{M}PROG$ должен быть не менее чем на 50 мс сброшен в 0. После этого напряжение на входе \overline{EA}/VPP возвращается к уровню +5 В. Источник напряжения +21 В (VPP) должен быть очень хорошо стабилизирован, так как превышение предельного значения +21.5 В на входе \overline{EA}/VPP приводит к необратимым повреждениям РПП.

Запись бита защиты. Бит защиты РПП, будучи установлен, запрещает доступ к РПП любыми внешними средствами. Схема записи бита защиты показана на [рис. 3.15](#).

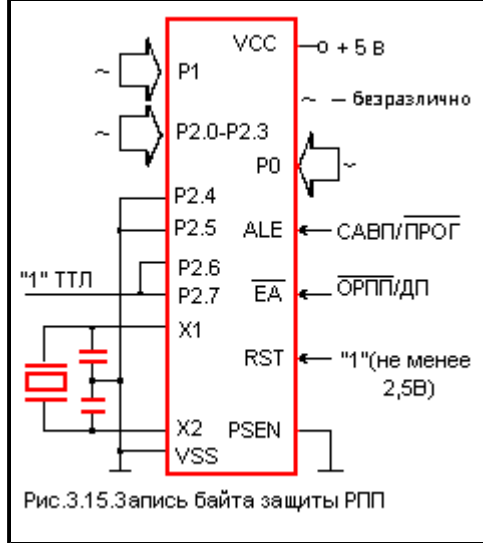


Рис. 3.15. Запись бита защиты РПП

Процедура записи бита защиты такая же, как и при загрузке программ в РПП, но на вывод P2.6 должен подаваться уровень 1. Сигналы на выводах портов P0, P1 и P2.0-P2.3 могут быть в любом состоянии. Однажды установленный бит защиты можно сбросит только путем полного стирания РПП под источником УФ-излучения.

Верификация программ. Если бит защиты не запрограммирован, то содержимое РПП может быть прочитано с целью проверки правильности загрузки прикладной программы либо по ходу программирования либо после окончания программирования МК51. Доступ к ячейкам РПП (рис. 3.16) осуществляется так же, как и при программировании РПП за исключением того, что на вывод P2.7 подается сигнал 0, используемый в качестве строб-сигнала чтения. Считанные коды команд и соответствующие им адреса РПП могут отображаться для визуального контроля на любом внешнем индикаторе и, кроме того, могут быть переданы в ОЗУ инструментальной ЭВМ для дисассемблирования.

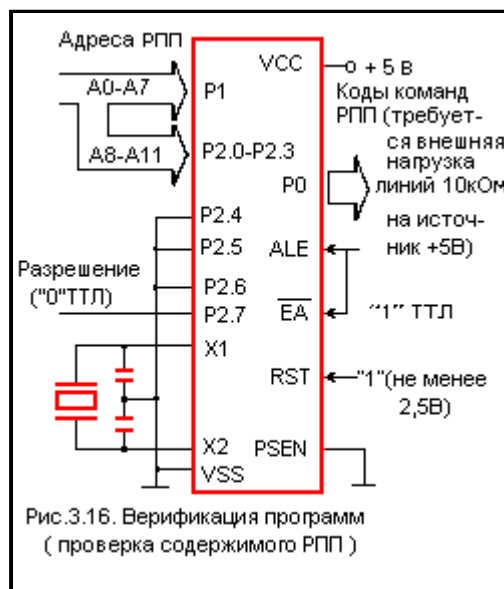


Рис. 3.16. Верификация программ (проверка содержимого РПП)

Стирание РПП. Для стирания содержимого РПП микроконтролер следует поместить под источник УФ-излучения с длиной волны менее $4 \cdot 10^{-7}$ м. Если кварцевая лампа имеет световую мощность 12000 мВт/см^2 , а расстояние между источником света и МК равно 2-3 см, то выдержка 20-30 мин обеспечивает световую дозу, достаточную для надежного стирания РПП. После стирания в матрице РПП содержатся все единицы.

Так как в спектре солнечного света и люминесцентного освещения содержится излучение с длиной волны менее $4 \cdot 10^{-7}$ м, то пребывание МК под воздействием этих источников света дольше установленного предела (около

одной недели на солнечном свете или около трех лет при люминесцентном освещении) может привести к искажению содержимого РПП. Для защиты содержимого РПП от разрушения под воздействием света на окне корпуса МК укрепляют светонепроницаемый экран.

[Особые режимы работы МК51](#)

9.2. Работа МК51 в пошаговом режиме

На этапе отладки прикладной программы в прототипе МК-системы очень удобным для разработчика оказывается режим пошагового (покомандного) исполнения программы. Система прерываний МК51 позволяет реализовать этот режим работы путем использования нескольких дополнительных команд. Как было описано в разделе [Система прерываний](#), внешний запрос прерывания не будет обслужен до тех пор, пока обслуживается прерывание с равным приоритетом. Этот запрос будет воспринят лишь после того, как одна команда после команды RETI будет выполнена. Иными словами, однажды вызвав подпрограмму обслуживания прерывания, вызвать ее вновь невозможно до тех пор, пока хотя бы одна команда основной программы не будет исполнена. Использовать это свойство системы прерываний МК51 для реализации пошагового режима можно следующим образом: одно из внешних прерываний, например $\overline{MINT0}$ (вывод P3.2 микроконтроллера), запрограммировать для представления запроса уровнем сигнала ($IT = 0$), а подпрограмма обслуживания этого прерывания должна заканчиваться последовательностью команд:

JNB	P3.2, \$; Ожидание уровня 1 на входе INT0
JB	P3.2, \$; Ожидание уровня 0 на входе INT1
RETI		; Возврат и выполнение одной команды

Здесь символом \$ обозначено текущее содержимое счетчика команд. Теперь если на вывод $\overline{MINT0}$ подавать сигнал от клавиши ШАГ отладочного модуля системы, то МК51 по сигналу $\overline{MINT0} = 0$ вызовет подпрограмму обслуживания внешнего прерывания 0 и будет в ней находиться до тех пор, пока не обнаружит на входе $\overline{MINT0}$ новый импульс (переход из 0 в 1 и снова в 0). По команде RETI произойдет возврат в основную программу, выполнение в ней одной команды и немедленный вызов подпрограммы обслуживания внешнего прерывания 0.

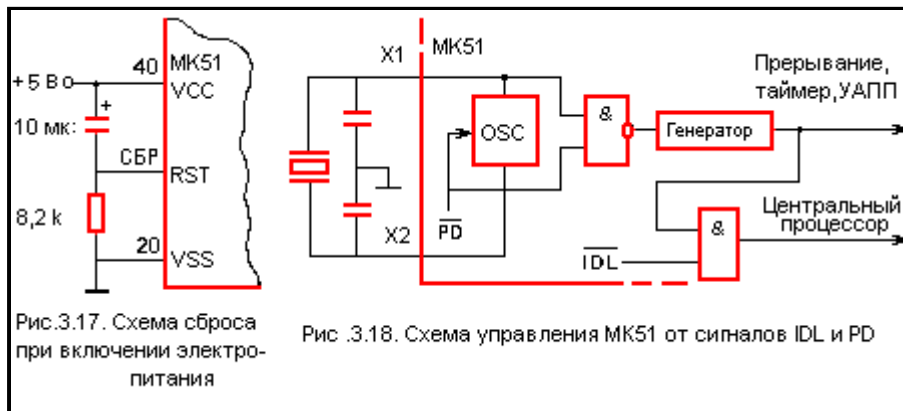
[Особые режимы работы МК51](#)

9.3. Сброс, режим холостого хода и режим пониженного энергопотребления

Сброс. В отличие от МК48 сброс МК51 осуществляется путем подачи на вход СБР (RST) сигнала 1. Для уверенного сброса МК51 этот сигнал 1 должен быть удержан на входе СБР по меньшей мере в течение двух машинных циклов (24 периода резонатора). Квазидвунаправленные буферные схемы внешних выводов ALE и \overline{MPSEN} находятся при этом в режиме ввода. Под воздействием сигнала СБР сбрасывается содержимое регистров: PC, ACC, B, PSW, DPTR, TMOD, TCON, T/C0, T/C1, IE, IP и SCON, в регистре PCON сбрасывается только старший бит, в регистр-указатель стека загружается код 07H, а в порты P0-P3 _ коды 0FFH. Состояние регистра SBUF неопределенное. Сигнал СБР не воздействует на содержимое ячеек РПД. Когда включается

электропитание (VCC), содержащее РПД неопределенно, за исключением операции возврата из режима пониженного энергопотребления.

На [рис. 3.17](#) показана схема автоматического формирования сигнала СБР при включении электропитания. Время, необходимое для полного заряда емкости, обеспечивает уверенный запуск резонатора и его работу в течение более чем двух машинных циклов.



Режим холостого хода. Любая команда, по которой установится управляющий бит IDL (PCON.0) в регистре управления мощностью, переведет МК51 в режим холостого хода. При этом продолжает работу внутренний генератор синхросигналов ([рис. 3.18](#)). Все регистры сохраняют свое значение. На выводах всех портов удерживается то логическое состояние, которое на них было в момент перехода в режим холостого хода. На выводах ALE и $\overline{\text{MPSEN}}$ формируется уровень 1.

Выйти из режима холостого хода можно или по сигналу СБР, или по прерыванию. Любой из разрешенных сигналов прерывания приведет к аппаратному сбросу бита PCON.0 и прекратит тем самым режим холостого хода. После исполнения команды RETI (выход из подпрограммы обслуживания прерывания) будет исполнена команда, которая следует в программе за командой, переведшей МК51 в режим холостого хода. (В некоторых модификациях МК51 этот режим может отсутствовать.)

Режим пониженного энергопотребления. В тех применениях МК-систем, где потребление электроэнергии, а следовательно, габаритные размеры и масса источника электропитания являются одними из основных показателей качества изделия, возможно использование МК51 в режиме пониженного энергопотребления. Перевод МК51 в этот режим возможен по команде, которая установит бит PCON.1 в регистре управления мощностью (см. [табл. 11.2](#)). В этом режиме останавливается генератор синхросигналов, содержимое РПД и регистров специальных функций сохраняется, а на выходных контактах портов удерживаются значения, соответствующие содержимому их буферных регистров. Выходы сигналов ALE и $\overline{\text{MPSEN}}$ сбрасываются. При этом электропитание осуществляется через вывод RST/VPD.

В режиме пониженного энергопотребления напряжение электропитания (VCC) может быть отключено. Перед выходом из режима оно должно быть восстановлено до номинального значения.

Выход из режима пониженного энергопотребления возможен только по сигналу RST. При этом переопределяются все регистры специальных функций, но содержимое РПД не изменяется. (В некоторых модификациях МК51 этот режим может отсутствовать.)

Защита от падения напряжения. При не мгновенных отказах блока электропитания МК-системы можно обеспечить сохранность содержимого РПД с помощью маломощного (батарейного) аварийного источника электропитания, подсоединяемого к выводу RST/VPD. Для этого МК при получении сообщения о грядущем падении напряжения VCC (прерывание от системы контроля электропитания) должен перегрузить в РПД основные параметры прерванного процесса и выдать сигнал, разрешающий подключение к выводу RST/VPD аварийного источника питания. Все эти процедуры МК должен успеть выполнить до того, как напряжение основного источника электропитания упадет ниже рабочего предела. После восстановления номинального значения напряжения в основной цепи электропитания выполняется системный сброс, и источник аварийного электропитания может быть отключен.