

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение высшего
профессионального образования
“ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

А. Г. Горюнов, С. Н. Ливенцов

АРХИТЕКТУРА МИКРОКОНТРОЛЛЕРА INEL 8051

Учебное пособие

Томск – 2005

УДК 681.322

Горюнов А.Г. Ливенцов С.Н.

Архитектура микроконтроллера Intel 8051: Учеб. пособие. – Томск: Изд-во ТПУ, 2005. – 86 с.

Учебное пособие посвящено самому популярному 8-разрядному семейству микроконтроллеров MCS51. Пособие содержит структурную организацию микроконтроллеров, описание и принцип действия интегрированных периферийных устройств, описание системы команд данного семейства.

Пособие подготовлено на кафедре «Электроника и автоматика физических установок» ТПУ и предназначена для студентов очного обучения специальности 140306.

УДК 681.322

Рекомендовано к печати Редакционно-издательским советом
Томского политехнического университета

Рецензенты

Заведующий кафедрой Северской государственной
технологической академии, кандидат технических наук, доцент
В.Я. Дурновцев

Декан факультета автоматике и
вычислительной техники, кандидат технических наук, доцент
С. А. Гайваронский

ISBN

©Томский политехнический университет, 2005

©Оформление. Издательство ТПУ, 2005

Содержание

ВВЕДЕНИЕ	9
1 Структурная организация микроконтроллера i8051	12
1.1 Общие характеристики.....	12
1.2 Структура микроконтроллера.....	14
2 Организация портов ввода / вывода микроконтроллера 8051	25
3 Доступ к внешней памяти	30
4 Таймеры / счетчики микроконтроллеров семейства 8051	34
4.1 Регистр режима работы таймера/счетчика TMOD	35
4.2 Регистр управления/статуса таймера TCON	36
4.3 Режимы работы таймеров-счетчиков.....	37
5 Последовательный интерфейс микроконтроллера 8051	39
5.1 Регистр управления/статуса приемопередатчика SCON	40
5.2 Работа UART в мультиконтроллерных системах.....	42
5.3 Скорость приема/передачи информации через последовательный порт	43
5.4 Особенности работы UART в различных режимах.....	45
6 Система прерываний микроконтроллера 8051	50
6.1 Регистр масок прерывания (IE)	51
6.2 Регистр приоритетов прерываний (IP).....	52
6.3 Выполнение подпрограммы прерывания	53
7 Режимы микроконтроллера 8051 с пониженным энергопотреблением	54
7.1 Регистр управления мощностью PCON.....	55
7.2 Режим XX.....	56
7.3 Режим ВМП.....	57
8 Система команд микроконтроллера семейства 8051	58
8.1 Общая характеристика.....	58
8.2 Группы команд	62
8.3 Команды передачи данных	64
8.4 Арифметические операции	66
8.5 Логические операции.....	66
8.6 Команды передачи управления	66
8.7 Операции с битами.....	68
9 Развитие микроконтроллерного семейства 8051	69
9.1 Семейство 8052	70
9.2 Расширение внешней памяти данных до 16МБ в микроконтроллерах совместимых с 8051	70
9.3 Увеличение быстродействия в микроконтроллерах совместимых с 8051	71
СПИСОК ЛИТЕРАТУРЫ	72

Список сокращений, символических имён и аббревиатур

Английская нотация

- A** – регистр-аккумулятор
- AC** – вспомогательный перенос (Auxiliary Carry flag in PSW)
- ACC** – символическое имя регистра A
- ad** – прямой 8-битный адрес байта RPD (0-127), порта или SFR
- add** – прямой 8-битный адрес назначения
- ads** – прямой 8-битный адрес источника
- adll** – прямой 11-битный адрес передачи управления
- ad16** – прямой 16-битный адрес передачи управления
- ad 16h** – старший байт прямого 16-битного адреса
- ad 16l** – младший байт прямого 16-битного адреса
- ALE** – управляющий сигнал строба адреса внешней памяти (Address Latch Enable)
- B** – регистр-расширитель аккумулятора
- bit** – прямой 8-битный адрес бита
- C** – флаг переноса
- CE (CS)** – Chip Enable (Chip Select) – выбор корпуса
- CLK** – синхросигнал (Clock)
- C/T** – управляющий бит выбора режима таймера /счетчика (Timer or Counter selector in TMOD)
- #=d** – 8-битный непосредственный операнд (константа)
- #=d16** – 16-битный непосредственный операнд (константа)
- #=d16h** – старший байт 16-битного непосредственного операнда
- #=d16l** – младший байт 16-битного непосредственного операнда
- DPH** – Data Pointer High (старший байт DPTR)
- DPL** – Data Pointer Low (младший байт DPTR)
- DPTR** – Data Pointer – регистр-указатель данных
- EA** – управляющий бит снятия блокировки всех прерываний (Enable All control bit in IE)
- EA/VPP** – External Address/Voltage Power Programming – управляющий сигнал отключения резидентной памяти программ
- EDM** – внешняя память данных (External Data Memory)
- EPROM** – Erasable Programmable Read Only Memory – электрически перепрограммируемая память (только для чтения)
- EPM** – внешняя память программ (External Program Memory)
- ES** – управляющий бит разрешения прерывания от УАПП (Enable Serial port control bit in IE)
- ET** – управляющий бит разрешения прерывания от таймера (Enable Timer control bit in IE)

EX – управляющий бит разрешения внешнего прерывания (Enable External interrupt control bit in IE)
F0, F1– флаги, специфицируемые пользователем
GATE– бит управления блокировкой T/C (Gating control bit in TMOD)
GF0, GF1– флаги пользователя (General Flags in PCON)
i – бит в КОП, определяющий регистр косвенного адреса: $i = 0, 1$ (RO, R1)
IDL – управляющий бит холостого хода (Idle mode in PCON)
IE – 1) Interrupt Enable register – регистр маски прерывания
2) флаг внешнего прерывания, установленный по спаду с сигнала INT (Interrupt Edge flag in TCON)
IP – Interrupt Priority control register – регистр приоритетов прерываний
INT – Interrupt – запрос прерывания
IR – Instruction Register – регистр команд
IT – управляющий бит выбора типа (уровень/спад) сигнала INT (Interrupt Type control bit in TCON)
M0, M1– управляющие биты выбора режима работы T/C (Operating Mode in TMOD)
OV – флаг переполнения (Overflow flag in PSW)
P – флаг паритета (Parity flag in PSW)
PC – Program Counter – счётчик команд
PCON – Power Control register – регистр управления мощностью
PD – бит управления мощностью потребления (Power Done in PCON)
PROG– Programming EPROM – управляющий сигнал для программирования EPROM
PS – управляющий бит приоритета UART (Serial port Priority control bit in IP)
PSEN– Program Store Enable – управляющий сигнал разрешения внешней памяти программ
PSW – Program Status Word – слово состояния программы
PT – управляющий бит приоритета таймера (Timer Priority control bit in IP)
PX – управляющий бит приоритета внешнего прерывания (External interrupt Priority control bit in IP)
PY.X– символическое имя бита X порта Y; $X = 0 \div 7$
RAM– Random Access Memory – ОЗУ
RAR – RAM Address Register – регистр адреса
RB8 – девятый (bit 8) принятый бит (Receive Bit 8 in SCON)

RD – Read – управляющий сигнал чтения
RDM – резидентная память данных (Resident Data Memory)
rel – 8-битный относительный адрес передачи управления (-127 ÷ +128)
REN – управляющий бит разрешения приема в UART (Receiver Enable control bit in SCON)
RI – флаг прерывания от приемника (Received Interrupt flag in SCON)
Ri – обобщенное имя регистра косвенного адреса (R0 или R1)
Rn – обобщенное имя рабочего регистра (n = 0 ÷ 7)
rrr – 3-битное поле в коде операции, определяющее регистр общего назначения (R0-R7)
RPM – резидентная память программ (Resident Program Memory)
RS – управляющий бит выбора банка регистров (Register bank Select in PSW)
RST – Reset – управляющий сигнал сброса
RST/VPD – Reset/Voltage Power Done (см. RST)
RXD – Receive Data pin – вход приёмника UART
Sn – состояние устройства управления, n = 1 ÷ 6 (State)
SCON – Serial port Control/status register – регистр управления последовательного порта
SM0, SM1, SM2 – управляющие биты режима работы UART (Serial port Mode control bits in SCON)
SMOD – управляющий бит двойной скорости передачи (Double Baud rate in PCON)
SFR – регистр специальных функций (Special Function Register)
SXPY – пояснения на временных диаграммах и схемах, привязывающие сигналы к состояниям S устройства управления и фазам P синхросигналов (X = 1 ÷ 6; Y = 1,2) для MCS51
SP – Stack Pointer – регистр указатель стека
SS – Single Step – управляющий сигнал пошагового (покомандного) режима работы
STB – Strobe – стробирующий сигнал
T0, T1 – тест-входы МК
TB8 – девятый (bit 8) передаваемый бит (Transmit Bit8 in SCON)
TCNT – Timer/Counter events – таймер/счётчик событий
TCON – Timer/counter Control/status register – регистр управления/состояния таймера
TF – флаг переполнения таймера (Timer overflow Flag in TCON)
TH – старший байт таймера (Timer High byte)
TI – флаг прерывания от передатчика (Transmit Interrupt flag in SCON)

- TL** – младший байт таймера (Timer Low byte)
TMOD– Timer/counter Mode register – регистр режима таймера/счётчика
TR – управляющий бит пуска таймера (Timer Run control bit in TCON)
TXD – Transmit Data – выход передатчика UART
WR – Write – управляющий сигнал записи

Специальные символы

- ←** – оператор присваивания (замещения)
↔ – оператор взаимного обмена
Λ, V, ∇ – операторы логических операций: И (конъюнкция), ИЛИ (дизъюнкция), исключающее ИЛИ
@ – префикс косвенной адресации
– префикс непосредственного операнда
(Y) – содержимое регистра или ячейки памяти с именем Y
((Y)) – содержимое ячейки памяти, адресуемой содержимым Y (косвенная адресация)
AND – логическая операция "конъюнкция" при ассемблировании
B – суффикс двоичного кода (Binary code)
H – суффикс шестнадцатеричного кода (Hexadecimal code)
HIGH – логическая операция выделения старшего байта из d76 при ассемблировании
LOW– логическая операция выделения младшего байта из d16 при ассемблировании
NOT – логическая операция "инверсия" при ассемблировании
OR – логическая операция "дизъюнкция" при ассемблировании

Русская нотация

- АЛУ** – арифметико-логическое устройство
БИС – большая интегральная схема
БК – выбор корпуса (см. СЕ, англ.)
ВПД – внешняя память данных
ВПП – внешняя память программ
ВХПР – вход приемника УАПП (см. RXD)
ВЫХПЕР– выход передатчика УАПП (см. TXD)
ЗП – управляющий сигнал записи (см. WR)
ЗПР – запрос прерывания (см. INT)
КОП – код операции (поле в теле команды)
МК – микроконтроллер
ОЗУ – оперативное запоминающее устройство (см. RAM)

ОРПП – управляющий сигнал отключения резидентной памяти программ см. EA, англ.)

ПП – память программ

ПРОГ – управляющий сигнал для программирования РПП (см. PROG)

РА – регистр адреса (см. RAR)

РВВ – БИС расширителя ввода/вывода

РВПП – управляющий сигнал разрешения внешней памяти программ (см. PSEN)

РК – регистр команд (см. IR)

РМП – регистр маски прерываний (см. IE)

РП – регистр приоритетов (см. IP)

РПД – резидентная память данных

РПП – резидентная память программ

РРТС – регистр режима таймера/счетчика (см. TMOD)

РСФ – регистры специальных функций (PSW, TMOD, TCON, SCON, PCON, IE, IP)

РУД – регистр-указатель данных (см. DPTR)

РУМ – регистр управления мощностью (см. PCON)

РУПП – регистр управления последовательного порта (см. SCON)

РУС – регистр-указатель стека (см. SP)

РУСТ – регистр управления/состояния таймера (см. TCON)

САВП – управляющий сигнал строба адреса внешней памяти (см. ALE)

СВР – управляющий сигнал сброса (см. RST)

СК – счетчик команд (см. PC, англ.)

ССП – слово состояния программы (см. PSW)

СТБ – стробирующий сигнал (см. STB)

Т/С – таймер/счетчик событий (см. TCNT)

УАПП – универсальный асинхронный приемопередатчик (последовательный порт МК51)

ВВЕДЕНИЕ

Развитие микроэлектроники и широкое применение ее изделий в промышленном производстве, в устройствах и системах управления самыми разнообразными объектами и процессами является в настоящее время одним из основных направлений научно-технического прогресса.

Использование микроэлектронных средств в изделиях производственного и культурно-бытового назначения не только приводит к повышению технико-экономических показателей изделий (стоимости, надежности, потребляемой мощности, габаритных размеров) и позволяет многократно сократить сроки разработки и отодвинуть сроки "морального старения" изделий, но и придает им принципиально новые потребительские качества (расширенные функциональные возможности, модифицируемость, адаптивность и т.д.).

За последние годы в микроэлектронике бурное развитие получило направление, связанное с выпуском однокристальных микроконтроллеров, которые предназначены для "интеллектуализации" оборудования различного назначения. Однокристальные (однокорпусные) микроконтроллеры представляют собой приборы, конструктивно выполненные в виде БИС и включающие в себя все составные части "голой" микроЭВМ: микропроцессор, память программ и память данных, а также программируемые интерфейсные схемы для связи с внешней средой. Использование микроконтроллеров в системах управления обеспечивает достижение исключительно высоких показателей эффективности при столь низкой стоимости (во многих применениях система может состоять только из одной БИС микроконтроллера), что микроконтроллерам, видимо, нет разумной альтернативной элементной базы для построения управляющих и/или регулирующих систем. К настоящему времени более двух третей мирового рынка микропроцессорных средств составляют именно однокристальные микроконтроллеры.

Несмотря на непрерывное развитие и появление все новых и новых 16- и 32-разрядных микроконтроллеров и микропроцессоров, наибольшая доля мирового микропроцессорного рынка и по сей день остается за 8-разрядными устройствами.

В 1976 году экспоненциальное развитие полупроводниковой технологии привело к созданию фирмой Intel первого МК – 8048. Помимо ЦП, в его состав входила память программ, память данных, восьмибитный таймер и 27 линий ввода/вывода. Сегодня 8048 является

уже достоянием истории, а вот следующее изделие, выпущенное Intel в 1980 году, живёт и здравствует поныне. Это – МК 8051.

В настоящее время среди всех 8-разрядных микроконтроллеров - семейство MCS-51 является несомненным чемпионом по количеству разновидностей и количеству компаний, выпускающих его модификации. Оно получило свое название от первого представителя этого семейства - микроконтроллера 8051, выпущенного в 1980 году на базе технологии NMOS. Удачный набор периферийных устройств, возможность гибкого выбора внешней или внутренней программной памяти и приемлемая цена обеспечили этому микроконтроллеру успех на рынке. С точки зрения технологии микроконтроллер 8051 являлся для своего времени очень сложным изделием - в кристалле было использовано 128 тыс. транзисторов, что в 4 раза превышало количество транзисторов в 16-разрядном микропроцессоре 8086.

Важную роль в достижении такой высокой популярности семейства 8051 сыграла открытая политика фирмы Intel, родоначальницы архитектуры, направленная на широкое распространение лицензий на ядро 8051 среди большого количества ведущих полупроводниковых компаний мира.

В результате на сегодняшний день существует более 200 модификаций микроконтроллеров семейства 8051, выпускаемых почти 50-ю компаниями. Эти модификации включают в себя кристаллы с широчайшим спектром периферии: от простых 20-выводных устройств с одним таймером и 1К программной памяти до сложнейших 100-выводных кристаллов с 10-разрядными АЦП, массивами таймеров-счетчиков, аппаратными 16-разрядными умножителями и 64К программной памяти на кристалле. Каждый год появляются все новые варианты представителей этого семейства. Основными направлениями развития являются: увеличение быстродействия (повышение тактовой частоты и переработка архитектуры), снижение напряжения питания и потребления, увеличение объема ОЗУ и FLASH памяти на кристалле с возможностью внутрисхемного программирования, введение в состав периферии микроконтроллера сложных устройств типа системы управления приводами, CAN и USB интерфейсов и т.п.

Все микроконтроллеры из семейства MCS51 имеют общую систему команд. Наличие дополнительного оборудования влияет только на количество регистров специального назначения.

Основными производителями клонов 51-го семейства в мире являются фирмы Philips, Infineon (Siemens), Intel, Acer Labs, Actel, Altium, Atmel, Analog Device, Cast Ins, Chipcon, Dallas, Digital Core Design, Hynix Semiconductor, Maxim, Temic, Oki, AMD, MHS, Gold Star, Winbond, Silicon Systems, Silicon Laboratories (Cygnal) и ряд других.

В рамках СССР производство микроконтроллера 8051 осуществлялось в Киеве, Воронеже (1816BE31/51, 1830BE31/51), Минске (1834BE31) и Новосибирске (1850BE31).

Микроконтроллеры данного семейства выпускаются в PLCC, DIP и QFP корпусах и могут работать в следующих температурных диапазонах:

- коммерческий ($0^{\circ}\text{C} - +70^{\circ}\text{C}$);
- расширенный ($-40^{\circ}\text{C} - +85^{\circ}\text{C}$);
- для военного использования ($-55^{\circ}\text{C} - +125^{\circ}\text{C}$).

Примерами микроконтроллеров семейства MCS51 с расширенными возможностями могут служить расширения микроконтроллеров MCS51/52: 8XC51FA, 8XC51GB, 80C152.

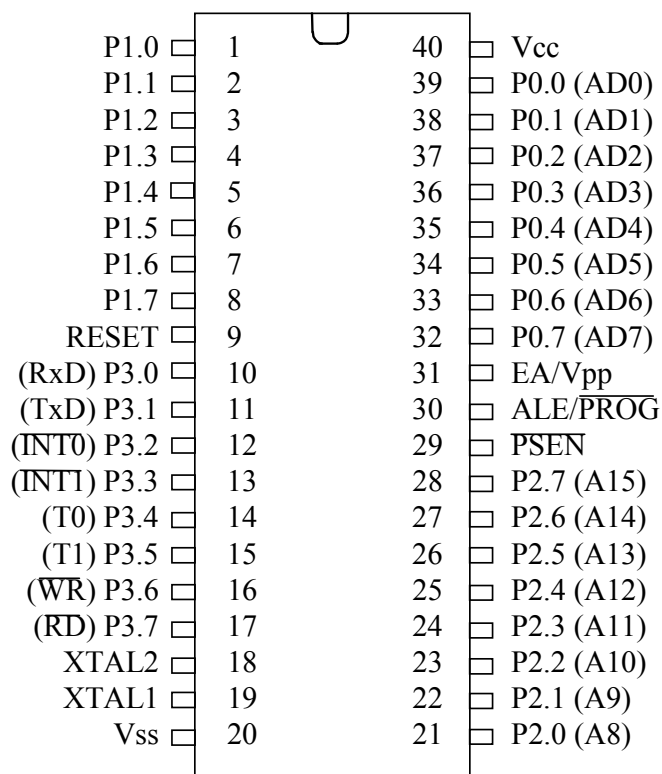
1 Структурная организация микроконтроллера i8051

1.1 Общие характеристики

Классический микроконтроллер i8051 (MCS51) и отечественный аналог КМ1816BE51 выполнены на основе высокоуровневой n-МОП технологии и выпускались в корпусе БИС, имеющем 40 внешних выводов. Цоколевка корпуса MCS51 и наименование выводов показаны на рис. 1 [1]. Для работы MCS51 требуется один источник электропитания +5В. Через четыре программируемых порта ввода/вывода MCS51 взаимодействует со средой в стандарте TTL-схем с тремя состояниями выхода.

Корпус MCS51 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.

Назначение выводов микроконтроллера 8051.



P1.0	□ 1	40	□ Vcc
P1.1	□ 2	39	□ P0.0 (AD0)
P1.2	□ 3	38	□ P0.1 (AD1)
P1.3	□ 4	37	□ P0.2 (AD2)
P1.4	□ 5	36	□ P0.3 (AD3)
P1.5	□ 6	35	□ P0.4 (AD4)
P1.6	□ 7	34	□ P0.5 (AD5)
P1.7	□ 8	33	□ P0.6 (AD6)
RESET	□ 9	32	□ P0.7 (AD7)
(RxD) P3.0	□ 10	31	□ EA/Vpp
(TxD) P3.1	□ 11	30	□ ALE/PROG
(INT0) P3.2	□ 12	29	□ PSEN
(INT1) P3.3	□ 13	28	□ P2.7 (A15)
(T0) P3.4	□ 14	27	□ P2.6 (A14)
(T1) P3.5	□ 15	26	□ P2.5 (A13)
(WR) P3.6	□ 16	25	□ P2.4 (A12)
(RD) P3.7	□ 17	24	□ P2.3 (A11)
XTAL2	□ 18	23	□ P2.2 (A10)
XTAL1	□ 19	22	□ P2.1 (A9)
Vss	□ 20	21	□ P2.0 (A8)

Рис. 1. Назначение выводов 8051

Обозначения на рис. 1:

- Vss – потенциал общего провода ("земли");
- Vcc – основное напряжение литания +5 В;
- XTAL1, XTAL2 – выводы для подключения кварцевого резонатора;
- RESET (RST) – вход общего сброса микроконтроллера;
- PSEN – разрешение внешней памяти программ; выдается только при обращении к внешнему ПЗУ;
- ALE – строб адреса внешней памяти;
- EA – отключение внутренней программной памяти; уровень 0 на этом входе заставляет микроконтроллер выполнять программу только из внешнего ПЗУ; игнорируя внутреннее(если последнее имеется);
- P0 – восьми битный двунаправленный порт ввода-вывода информации: при работе с внешними ОЗУ и ПЗУ по линиям порта в режиме временного мультиплексирования выдается адрес внешней памяти, после чего осуществляется передача или прием данных;
- P1 – восьми битный квази двунаправленный порт ввода/вывода: каждый разряд порта может быть запрограммирован как на ввод, так и на вывод информации, независимо от состояния других разрядов;
- P2 – восьми битный квази двунаправленный порт, аналогичный P1; кроме того, выводы этого порта используются для выдачи адресной информации при обращении к внешней памяти программ или данных (если используется 16-битовая адресация последней).
- P3 – восьми битный квази двунаправленный порт, аналогичный P1; кроме того, выводы этого порта могут выполнять ряд альтернативных функций, которые используются при работе таймеров, порта последовательного ввода-вывода, контроллера прерываний, и внешней памяти программ и данных.

1.2 Структура микроконтроллера

Основу структурной схемы MCS51 (рис. 2) образует внутренняя двуправленная 8-битная шина, которая связывает между собой все основные узлы и устройства: резидентную память программ (RPM), резидентную память данных (RDM), арифметико-логическое устройство (ALU), блок регистров специальных функций, устройство управления (CU) и порты ввода/вывода (P0-P3).

Рассмотрим основные элементы структуры и особенности организации вычислительного процесса в MCS51 [1, 2].

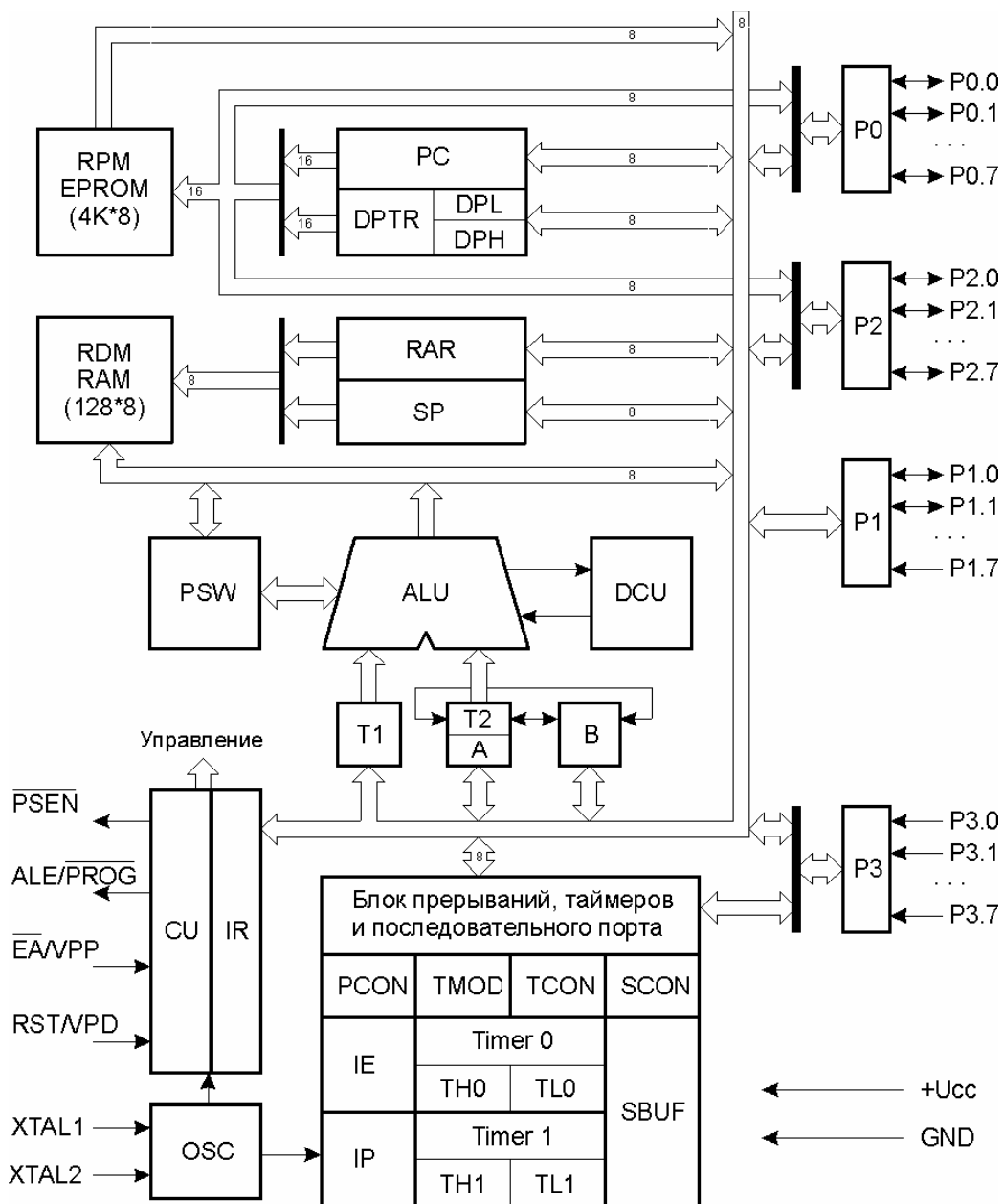


Рис. 2. Структурная схема MCS51

1.2.1 Арифметико-логическое устройство

8-битное арифметико-логическое устройство (ALU) 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. К входам подключены программно-недоступные регистры T1 и T2, предназначенные для временного хранения операндов, схема десятичной коррекции (DCU) и схема формирования признаков результата операции (PSW) [1, 3].

Простейшая операция сложения используется в ALU для инкрементирования содержимого регистров, продвижения регистра-указателя данных (RAR) и автоматического вычисления следующего адреса резидентной памяти программ. Простейшая операция вычитания используется в ALU для декрементирования регистров и сравнения переменных.

Простейшие операции автоматически образуют “танделы” для выполнения таких операций, как, например, инкрементирование 16-битных регистровых пар. В ALU реализуется механизм каскадного выполнения простейших операций для реализации сложных команд. Так, например, при выполнении одной из команд условной передачи управления по результату сравнения в ALU трижды инкрементируется счётчик команд (PC), дважды производится чтение из RDM, 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 всего лишь за 2 мкс.

Важной особенностью ALU 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 достаточно важна, поскольку для управления объектами часто применяются алгоритмы, содержащие операции над входными и выходными булевыми переменными, реализация которых средствами обычных микропроцессоров сопряжена с определенными трудностями.

Таким образом, ALU может оперировать четырьмя типами информационных объектов: булевыми (1 бит), цифровыми (4 бита), байтными (8 бит) и адресными (16 бит). В ALU 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 операции и режима адресации базовое число команд 111 расширяется до 255 из 256 возможных при однобайтном коде операции.

1.2.2 Резидентная память программ / данных и регистры общего назначения

Резидентные (размещённые на кристалле) память программ (**RPM**) и память данных (**RDM**) физически и логически разделены, имеют различные механизмы адресации, работают под управлением различных сигналов и выполняют разные функции [1].

Память программ.

Память программ **RPM** имеет емкость 4 Кбайта и предназначена для хранения команд, констант, управляющих слов инициализации, таблиц перекодировки входных и выходных переменных и т.п. Память имеет 16-битную шину адреса.

При обращении к **внешней памяти программ (EPM)** все микроконтроллеры семейства 8051 всегда используют 16-разрядный адрес, что обеспечивает им доступ к 64 Кбайт ПЗУ. Микроконтроллер обращается к программной памяти при чтении кода операции и операндов (используя счетчик команд **PC**), а также при выполнении команд переноса байта из памяти программ в аккумулятор. При выполнении команд переноса данных адресация ячейки памяти программ, из которой будут прочитаны данные, может осуществляться с использованием как счетчика **PC**, так и специального двухбайтового регистра-указателя данных **DPTR**.

Память данных и регистры общего назначения.

Память данных **RDM** предназначена для хранения переменных в процессе выполнения прикладной программы, адресуется одним байтом и имеет емкость 128 байт. Кроме того, к её адресному пространству примыкают адреса регистров специальных функций, которые перечислены в таблице 2.

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

Первые 32 байта организованы в четыре банка регистров **общего назначения (РОН)**, обозначаемых соответственно банк 0 – банк 3 (см. таблицу 1). Каждый из них состоит из восьми регистров **R0 – R7**. В любой момент программе доступен только один банк регистров, номер которого содержится в третьем и четвертом битах слова состояния программы **PSW** (см. ниже).

Оставшееся адресное пространство может конфигурироваться разработчиком по своему усмотрению: в нем располагаются стек, системные и пользовательские области данных. Обращение к ячейкам памяти данных возможно двумя способами. Первый способ – прямая адресация ячейки памяти. В этом случае адрес ячейки является операндом соответствующей команды. Второй способ – косвенная адресация с помощью регистров **R0** или **R1**: перед выполнением

соответствующей команды в один из них должен быть занесен адрес ячейки, к которой необходимо обратиться.

Для обращения к **внешней памяти данных (EDM)** используется только косвенная адресация с помощью регистров R0 и R1 или с помощью 16-разрядного регистра-указателя DPTR.

Он относится к группе регистров специальных функций, и с его помощью можно адресовать все 64 Кбайта внешней памяти.

Часть памяти данных представляет собой так называемую битовую область, в ней имеется возможность при помощи специальных битовых команд адресовываться к каждому разряду ячеек памяти. Адрес прямо адресуемых битов может быть записан либо в виде (Адрес Байта).(Разряд), например выражение 21.3 означает третий разряд ячейки памяти с адресом 21H, либо в виде абсолютного битового адреса. Соответствие этих двух способов адресации можно определить по таблице.

Таблица 1

Адреса битовых областей памяти микроконтроллера 8051 и регистров общего назначения.

Адрес байта	Адреса битов по разрядам							
Adr	D7	D6	D5	D4	D3	D2	D1	D0
2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70
2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	29	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00
1FH ... 18H	Банк 3 РОН							
17H ... 10H	Банк 2 РОН							

0FH ... 08H	Банк 1 PОН
07H ... 00H	Банк 0 PОН

Примечание. Адрес прямо адресуемых битов может быть записан либо в виде выражения (Адрес Байта).(Разряд), например выражение 21.3 означает адрес третьего разряда ячейки памяти с адресом 21H, либо в виде абсолютного битового адреса, который для данного бита равен (см. таблицу 1) 0B.

1.2.3 Регистры специальных функций

К адресному пространству памяти данных примыкает адресное пространство регистров специальных функций SFR (Special Function Register) [1].

Таблица 2

Размещение регистров специальных функций в пространстве SFR

Адрес	Символ	Наименование
0E0H	*ACC	Аккумулятор (Accumulator)
0F0H	*B	Регистр расширитель аккумулятора (Multiplication Register)
0D0H	*PSW	Слово состояния программы (Program Status Word)
080H	*P0	Порт 0 (SFR P0)
090H	*P1	Порт 1 (SFR P1)
0A0H	*P2	Порт 2 (SFR P2)
0B0H	*P3	Порт 3 (SFR P3)
081H	SP	Регистр указатель стека (Stack Pointer)
083H	DPH	Старший байт регистра указателя данных DPTR (Data Pointer High)
082H	DPL	Младший байт регистра указателя данных DPTR (Data Pointer Low)
08CH	TH0	Старший байт таймера 0 ()
08AH	TL0	Младший байт таймера 0 ()
08DH	TH1	Старший байт таймера 1 ()
08BH	TL1	Младший байт таймера 1 ()
089H	TMOD	Регистр режимов таймеров счетчиков (Timer/Counter Mode Control Register)
088H	*TCON	Регистр управления статуса таймеров (Timer/Counter Control Register)
0B8H	*IP	Регистр приоритетов (Interrupt Priority Control Register)
0A8H	*IE	Регистр маски прерывания (Interrupt Enable Register)
087H	PCON	Регистр управления мощностью (Power Control Register)
098H	*SCON	Регистр управления приемопередатчиком (Serial Port Control Register)
099H	SBUF	Буфер приемопередатчика (Serial Data Buffer)

Примечание. Регистры, символ которых отмечен знаком (*), допускают адресацию своих отдельных бит при использовании команд из группы команд операций над битами.

Адреса, по которым расположены эти регистры, приведены в таблице 3.

Таблица 3
Карта адресуемых битов в блоке регистров специальных функций

Адрес байта	Адреса битов по разрядам								Имя регистра	
	Adr	D7	D6	D5	D4	D3	D2	D1		D0
F0H	F7	F6	F5	F4	F3	F2	F1	F0		B
...
E0H	E7	E6	E5	E4	E3	E2	E1	E0		ACC
...
D0H	D7	D6	D5	D4	D3	D2	D1	D0		PSW
...
B8H	-	-	-	BC	BB	BA	B9	B8		IP
...
B0	B7	B6	B5	B4	B3	B2	B1	B0		P3
...
A8H	AF	-	-	AC	AB	AA	A9	A8		IE
...
A0H	A7	A6	A5	A4	A3	A2	A1	A0		P2
...
98H	9F	9E	9D	9C	9B	9A	99	98		SCON
...
90H	97	96	95	94	93	92	91	90		P1
...
88H	8F	8E	8D	8C	8B	8A	89	88		TCON
...
80H	87	86	85	84	83	82	81	80		P0

Примечание. Адрес прямо адресуемых битов может быть записан либо в виде выражения (Название Регистра).(Разряд), например выражение SCON.3 означает адрес третьего разряда регистра SCON, либо в виде абсолютного битового адреса, который для данного бита равен (см. таблицу 3) 9B. Кроме того, некоторые биты управляющих регистров имеют собственные названия, так например данный бит имеет название TB8.

Отметим, что регистры занимают только часть 128-байтового адресного пространства. Т.е. ячейки памяти с адресами 80H-0FFH, которые не заняты регистрами, физически отсутствуют, на кристаллах

микроконтроллеров семейства 8051 при обращении к ним можно прочитать лишь код команды возврата.

Регистры специальных функций управляют работой блоков, входящих в микроконтроллер.

- Регистры-защелки **SFR** параллельных портов **P0...P3** - служат для ввода-вывода информации.
- Две регистровые пары с именами **TH0, TL0** и **TH1, TL1** представляют собой регистры, двух программно-управляемых 16-битных таймеров-счетчиков.
- Режимы таймеров-счетчиков задаются с использованием регистра **TMOD**, а управление ими осуществляется с помощью регистра **TCON**.
- Для управления режимами энергопотребления микро-ЭВМ используется регистр **PCON**.
- Регистры **IP** и **IE** управляют работой системы прерываний микро-ЭВМ,
- регистры **SBUF** и **SCON** – работой приемопередатчика последовательного порта.
- Регистр-указатель стека **SP** в микро-ЭВМ рассматриваемого семейства – восьми битный. Он может адресовать любую область внутренней памяти данных. В отличие от микропроцессора **KP580BM80**, у микро-ЭВМ семейства 8051 стек «растет вверх», т.е. перед выполнением команды **PUSH** или **CALL** содержимое **SP** инкрементируется, после чего производится запись информации в стек. Соответственно при извлечении информации из стека регистр **SP** декрементируется после извлечения информации. В процессе инициализации микро-ЭВМ после сигнала сброса или при включении питающего напряжения в **SP** заносится код **07H**. Это означает, что первый элемент стека будет располагаться в ячейке памяти с адресом **08H**.
- Регистр-указатель данных **DPTR** чаще всего используют для фиксации 16-битного адреса в операциях обращения к внешней памяти программ и данных. С точки зрения программиста он может выступать как в виде одного 16-битного регистра, так и в виде двух независимых регистров **DPL** и **DPH**.
- Аккумулятор (**ACC**) 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 нуль, формирование

флага паритета и т.п. В распоряжении пользователя имеются 8 регистров общего назначения R0–R7 одного из четырёх возможных банков. При выполнении многих команд в АЛУ формируется ряд признаков операции (флагов), которые фиксируются в регистре PSW.

- Регистр В используется как источник и как приемник при операциях умножения и деления, обращение к нему, как к регистру SFR, производится аналогично аккумулятору.
- При выполнении ряда команд в арифметико-логическом устройстве (АЛУ) формируются признаки операций – флаги, которые фиксируются в регистре PSW.

В таблице 4 приводится перечень флагов PSW, даются их символические имена и описываются условия их формирования.

Таблица 4

Формат слова состояния программы PSW

Символ	Разряд	Имя и назначение
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается аппаратно или программно при выполнении арифметических и логических операций
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратно при выполнении команд сложения и вычитания и сигнализирует о переносе или займе в бите 3
F0	PSW.5	Флаг 0. Может быть установлен, сброшен или проверен программой как флаг, специфицируемый пользователем
RS1 RS0	PSW.4 PSW.3	Выбор банка регистров. Устанавливается и сбрасывается программно для выбора рабочего банка регистров (таблица 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 аппаратно в каждом цикле и фиксирует нечётное/чётное число единичных битов в аккумуляторе, т.е. выполняет контроль по четности

Биты выбора используемого банка регистров могут быть изменены программным путем (см. таблицу 5).

Таблица 5

Выбор рабочего банка регистров

RS1	RS0	Банк	Границы адресов
0	0	0	00H – 07H
0	1	1	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

Наиболее “активным” флагом PSW является флаг переноса, который принимает участие и модифицируется в процессе выполнения множества операций, включая сложение, вычитание и сдвиги. Кроме того, флаг переноса (C) выполняет функции “булева 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. ALU не управляет флагами селекции банка регистров (RS0, RS1), их значение полностью определяется прикладной программой и используется для выбора одного из четырёх регистровых банков.

В микропроцессорах, архитектура которых опирается на аккумулятор, большинство команд работают с ним, используя неявную адресацию. В Intel 8051 дело обстоит иначе. Хотя процессор имеет в своей основе аккумулятор, он может выполнять множество команд и без его участия. Например, данные могут быть переданы из любой ячейки RDM 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 без использования аккумулятора. Флаги и управляющие биты могут быть проверены и изменены аналогично.

1.2.4 Устройство управления и синхронизации

Кварцевый резонатор, подключаемый к внешним выводам микроконтроллера, управляет работой внутреннего генератора, который в свою очередь формирует сигналы синхронизации.

Устройство управления (CU) 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. Эта временная диаграмма иллюстрирует работу устройства управления MCS51 при выборке и исполнении команд различной степени сложности. Все заштрихованные сигналы являются внутренними и недоступны пользователю MCS51 для контроля. Внешними, наблюдаемыми сигналами являются только сигналы резонатора и stroba адреса внешней памяти (ALE). Как видно из временной диаграммы, сигнал ALE формируется дважды за один машинный цикл (S1P2 – S2P1 и S4P2 – S5P1) и используется для управления процессом обращения к внешней памяти [1, 4, 5].

Большинство команд микроконтроллера 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, производится расчёт времени исполнения прикладных программ.

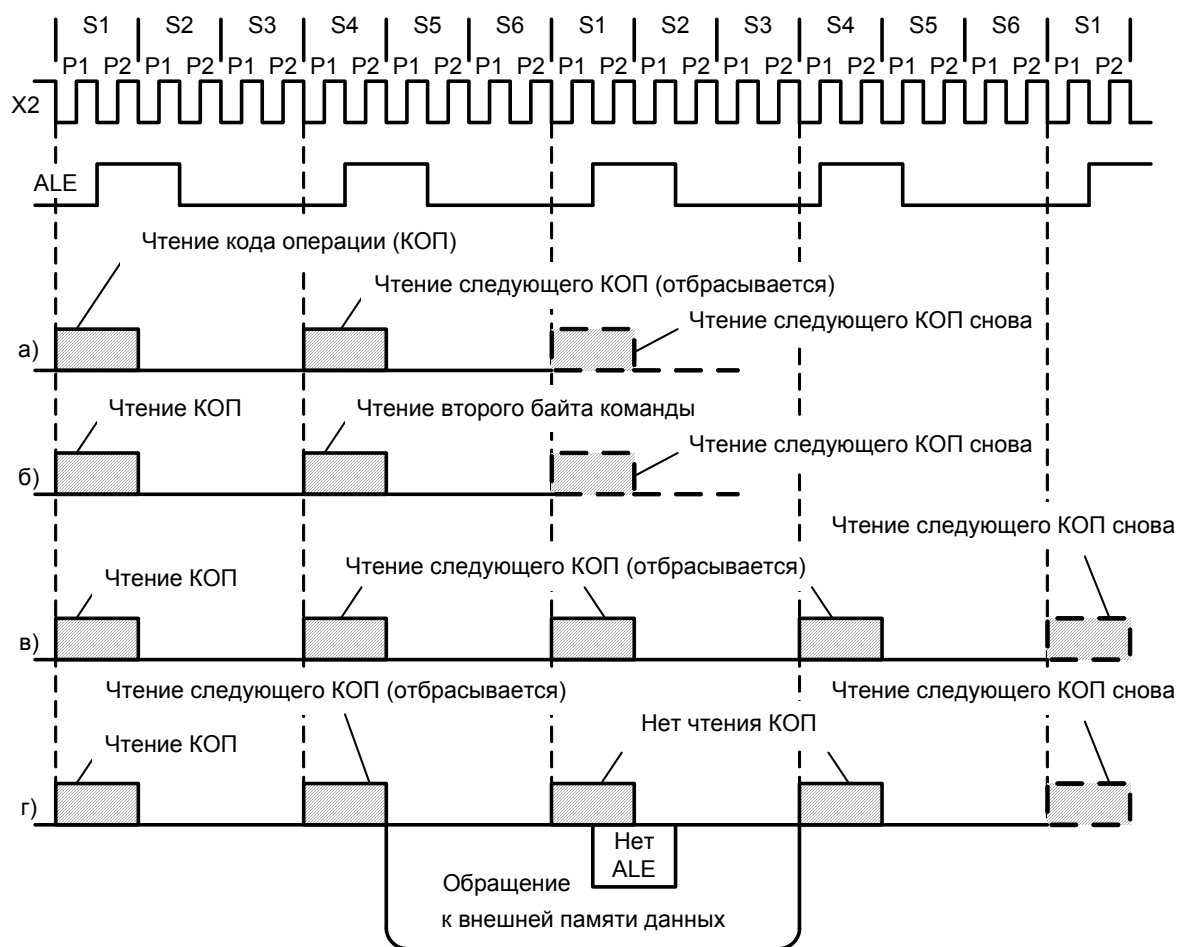


Рис. 3. Последовательности выборки и выполнения команд в MCS51:
 а – команда 1 байт / 1 цикл, например INC A; б – команда 2 байта / 1 цикл, например ADD A,#d; в – команда 1 байт / 2 цикла, например INC DPTR; г – команда 1 байт/ 2 цикла, например MOVX

На схеме микроконтроллера к устройству управления примыкает регистр команд (IR). В его функцию входит хранение кода выполняемой команды.

Входные и выходные сигналы устройства управления и синхронизации:

- PSEN – разрешение программной памяти,
- ALE – выходной сигнал разрешения фиксации адреса,
- PROG – сигнал программирования,
- EA – блокировка работы с внутренней памятью,
- V_{PP} – напряжение программирования,
- RST – сигнал общего сброса,
- V_{PD} – вывод резервного питания памяти от внешнего источника,
- XTAL – входы подключения кварцевого резонатора.

2 Организация портов ввода / вывода микроконтроллера 8051

Все четыре порта (P0-P3) предназначены для ввода или вывода информации побайтно. Схемотехника портов ввода/вывода MCS51 для одного вывода показана на рис. 4 – 7 [1].

Каждый из портов содержит регистр-защелку (SFR P0 — SFR P3), входной буфер и выходной драйвер. Каждый из разрядов регистра-защелки SFR является D-триггером, информация в который заносится с внутренней шины данных микроконтроллера по сигналу «Запись в SFR P_x» (x= 0, 1, 2, 3) от центрального процессорного элемента (CPU). С прямого выхода D-триггера информация может быть выведена на внутреннюю шину по сигналу «Чтение SFR P_x» от CPU, а с вывода микросхемы («из внешнего мира») по сигналу «Чтение выводов P_x». Одни команды активизируют сигнал «Чтение SFR P_i», другие - «Чтение выводов P_i».

Выходные драйверы портов 0 и 2, а также входной буфер порта 0 используются при обращении к внешней памяти (ВП). При этом через 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 разрядность адреса равна 16 бит.

Все выводы порта 3 могут быть использованы для реализации альтернативных функций, перечисленных в таблице 6. Эти функции могут быть задействованы путем записи 1 в соответствующие биты регистра-защелки (P3.0-P3.7) порта 3.

Порт 0 является двунаправленным, а порты 1-3 - квазидвунаправленными. Каждая линия портов может быть использована независимо для ввода или вывода.

По сигналу RST 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 система имеет внешнюю память, связь с которой организуется через общую разделяемую шину адреса/данных, работающую в режиме временного мультиплексирования.

Особенности электрических характеристик портов

Выходные каскады триггеров SFR портов P1 – P3 выполнены на полевых транзисторах с внутренней нагрузкой, в то время как аналогичные каскады триггеров SFR P0 – на транзисторах с открытым стоком. Каждая линия любого из портов может независимо

использоваться как для ввода, так и для вывода информации (для линий портов P0 и P2 это справедливо тогда, когда они не используются для обращения к внешней памяти).

Для перевода любой линии портов P1 – P3 в режим ввода информации необходимо в соответствующий разряд SFR занести 1. При этом выходной полевой транзистор отключается. Внутренний нагрузочный резистор как бы «подтягивает» потенциал вывода к напряжению питания, в то время как внешняя нагрузка может сделать его нулевым. Выходные каскады порта P0 имеют иную структуру. Нагрузочный полевой транзистор линии порта включен только тогда, когда порт выводит 1 при обращении к внешней памяти. В остальных случаях нагрузочный транзистор отключен. Таким образом, при работе в режиме обычного ввода-вывода информации (как, например, порт P1) выходные каскады порта P0 представляют собой ступени на транзисторах с открытым стоком. Запись 1 в соответствующий бит SFR отключает и второй транзистор, что приводит к тому, что вывод БИС оказывается под «плавающим» потенциалом. Это позволяет использовать линии порта P0 как выводы с высоко импедансным состоянием.

Поскольку выходные каскады портов P1 – P3 имеют внутреннюю нагрузку, при переводе в режим ввода информации они становятся источниками тока для микросхемы или транзистора, нагруженных на данный вывод.

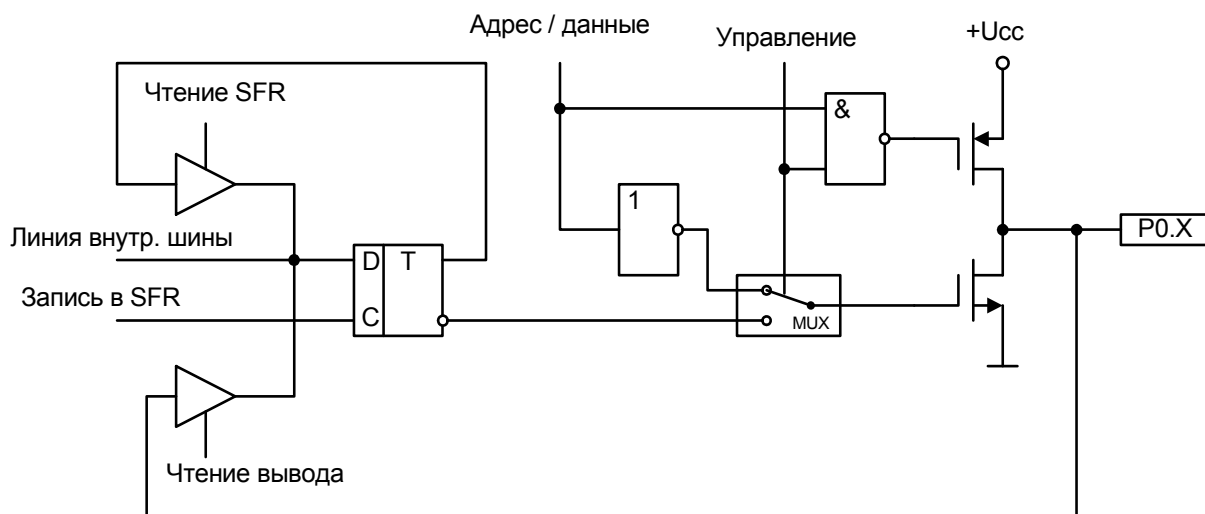


Рис. 4. Порт P0

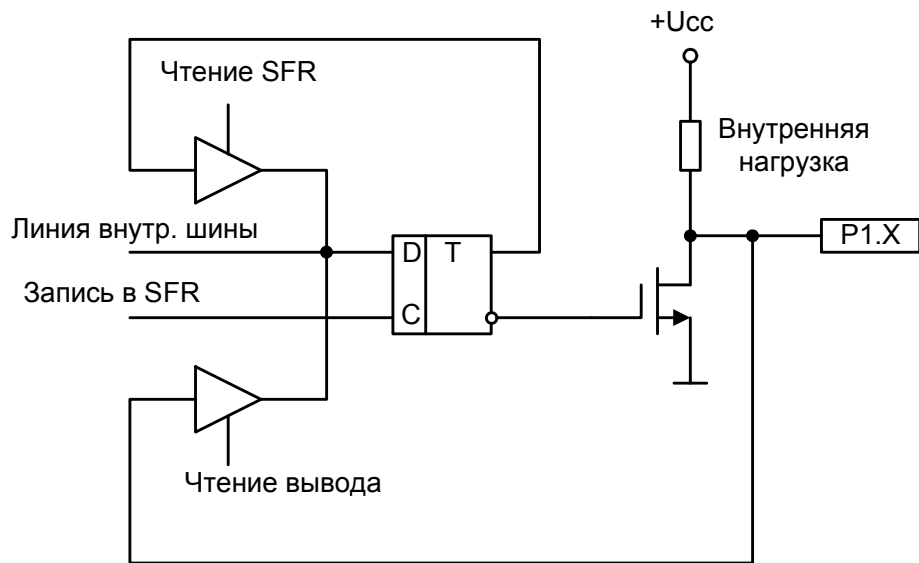


Рис. 5. Порт P1

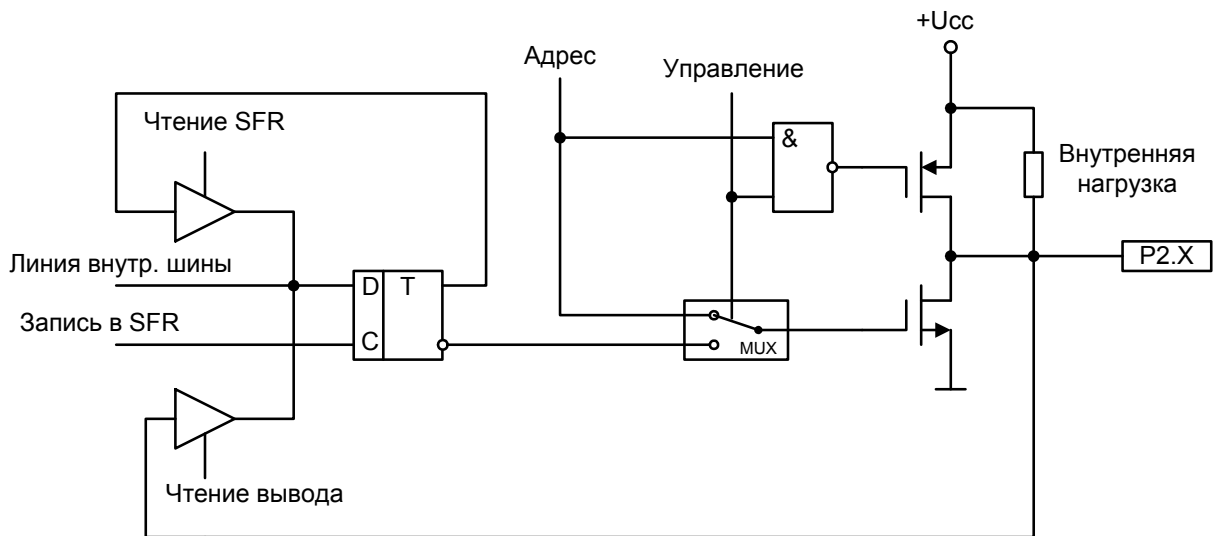


Рис. 6. Порт P2

называется “чтение-модификация-запись”. Этот режим обращения предполагает ввод сигналов не с внешних выводов порта, а из его регистра-защёлки, что позволяет исключить неправильное считывание ранее выведенной информации [1]. Этот механизм обращения к портам реализован в 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, на-пример, DJNZ r, LABEL;
- MOV PX.Y,C – передача бита переноса в бит X порта Y;
- SET PX.Y – установка бита X порта Y;
- CLR PX.Y – сброс бита X порта Y.

Совсем не очевидно, что последние три команды в приведённом 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-защёлку [1].

Причиной, по которой команды “чтение-модификация-запись” обеспечивают отдельный доступ к регистру-защёлке порта и к внешним выводам порта, является необходимость исключить возможность неправильного прочтения уровней сигналов на внешних выводах. Предположим для примера, что линия X порта Y 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 (например, командой MOV ACC, PY) с внешнего вывода порта, а не из D-триггера регистра-защёлки порта приведёт 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 считывания из регистра-защёлки, а не с внешнего вывода порта, что обеспечивает получение правильного значения 1.

3 Доступ к внешней памяти

В микроконтроллерных системах, построенных на основе MCS51, возможно использование двух типов внешней памяти: постоянной памяти программ (ЕРМ) и оперативной памяти данных (ЕDM). Доступ к ЕРМ осуществляется при помощи управляющего сигнала $\overline{\text{PSEN}}$, который выполняет функцию строб-сигнала чтения. Доступ к ЕDM обеспечивается управляющими сигналами $\overline{\text{RD}}$ и $\overline{\text{WR}}$, которые формируются в линиях P3.7 и P3.6 при выполнении портом 3 альтернативных функций (см таблицу 6) [1, 3].

При обращении к ЕРМ всегда используется 16-битный адрес. Доступ к ЕDM 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 течении одного цикла записи или чтения) в регистре защёлке порта 2.

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

Через порт 0 в режиме временного мультиплексирования осуществляется выдача младшего байта адреса и передача байт данных. Сигнал ALE 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 $\overline{\text{WR}}$. В цикле чтения вводимый байт данных принимается в порт 0 по фронту стробирующего сигнала $\overline{\text{RD}}$.

При любом обращении к внешней памяти устройство управления MCS51 загружает в регистр-защёлку порта 0 код 0FFh, стирая тем самым информацию, которая могла в нём храниться.

Обращение к внешней памяти программ ЕРМ возможно в двух случаях:

- когда сигнал $\overline{\text{EA}}$ активен, т.е. имеет нулевой уровень,
- когда программный счетчик РС содержит число больше 0FFh (число больше чем максимальный адрес резидентной памяти программ РРМ).

Временные диаграммы на рис. 8 иллюстрируют процесс генерации управляющих сигналов ALE и $\overline{\text{PSEN}}$ при обращении к внешней памяти [1].

Основная функция сигнала ALE – обеспечить временное согласование передачи из порта 0 на внешний регистр младшего байта адреса в цикле чтения из EPМ. Сигнал ALE приобретает значение 1 дважды в каждом машинном цикле. Это происходит даже тогда, когда в цикле выборки нет обращения к EPМ. Доступ к EPМ возможен только в том случае, если сигнал ALE отсутствует. Первый сигнал ALE во втором машинном цикле команды MOVX блокируется. Следовательно, в любой МК-системе, не использующей EPМ, сигнал ALE 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 EPМ сигнал \overline{PSEN} не генерируется, а при обращении к EPМ он выполняет функцию строб-сигнала чтения. Полный цикл чтения EPМ, включая установку и снятие сигнала \overline{RD} , занимает 12 периодов резонатора.

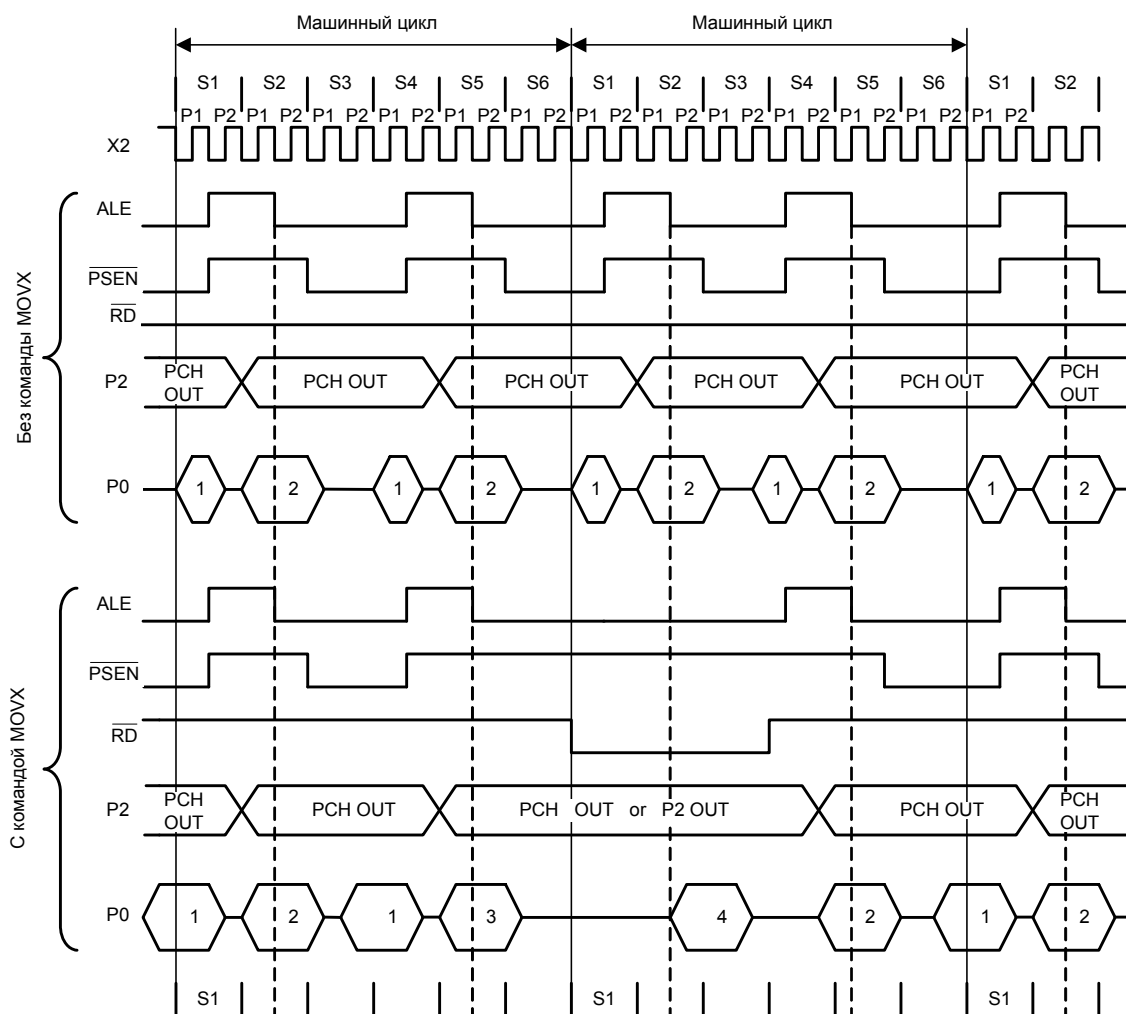


Рис. 8. Временные диаграммы операций с обращением к внешней памяти: 1 – INSTR IN; 2 – PCL OUT; 3 – DPL OUT; 4 – DATA IN

Временные диаграммы на рис. 9 и 10 иллюстрируют процесс выборки команд из EPM и работу с EDM в режимах чтения и записи соответственно [1, 5].

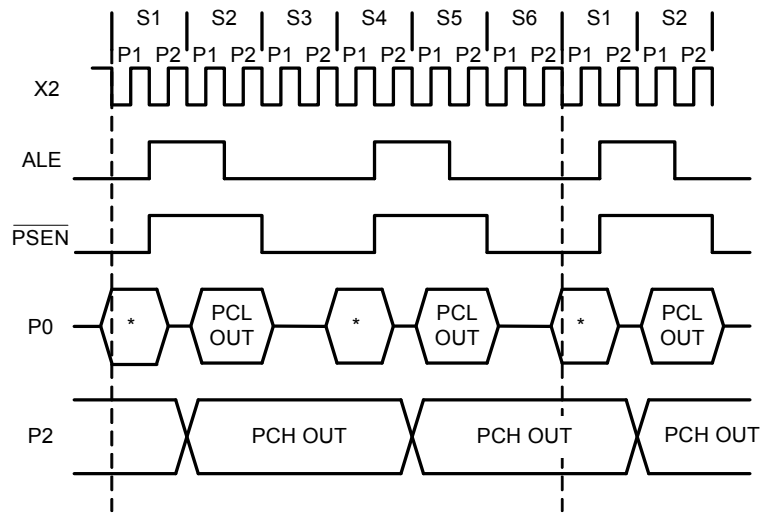


Рис. 9. Временная диаграмма выборки команды из EPM:
* - Байт команды

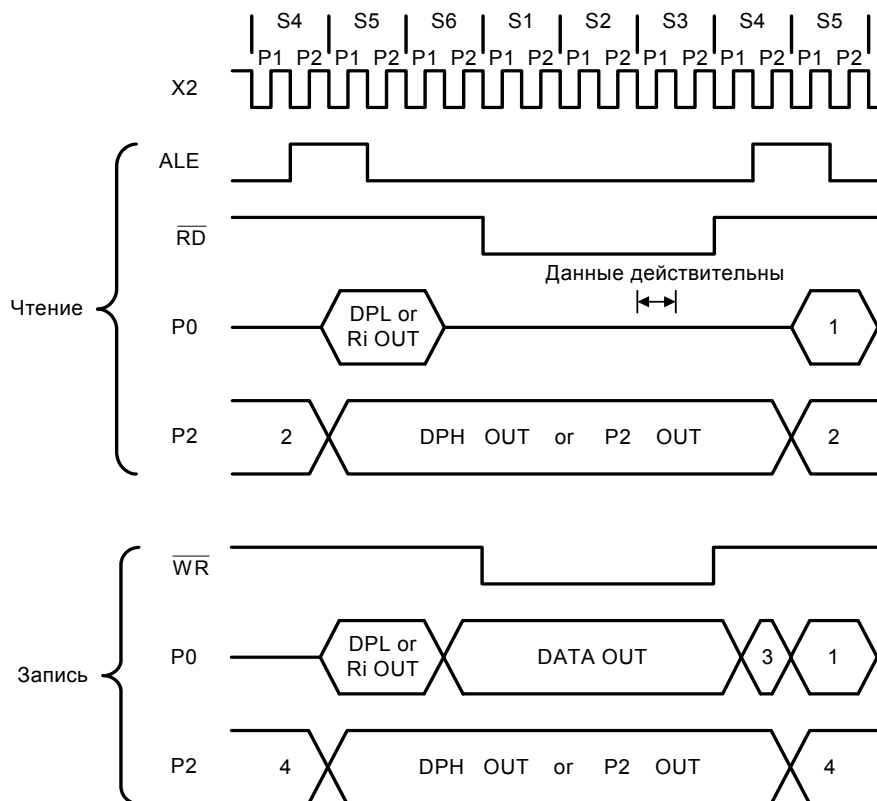


Рис. 10. Временная диаграмма работы с EDM:
1 – PCL OUT, если используется EPM; 2 – DPH or P2 OUT; 3 – PCL OUT;
4 – PCH or P2 OUT

Особый режим работы MCS51

Содержимое памяти программ MCS51 заполняется единожды на этапе разработки МК-системы и не может быть модифицировано в завершённом (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 ёмкости 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 совместного адресного пространства ЕРМ и EDM путём подключения внешней логики, как показано на рис. 11.

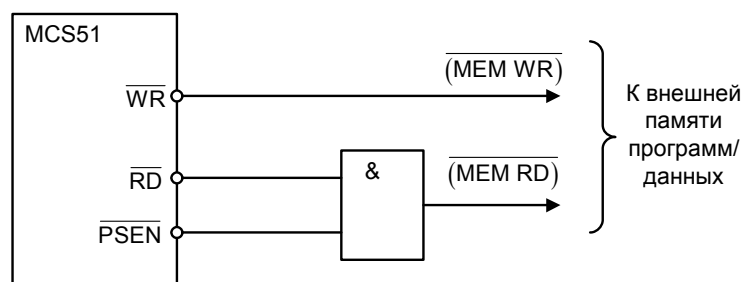


Рис. 11. Схема совмещения адресного пространства ЕРМ и EDM

На выходе схемы И (см. рис. 11) формируется строб-сигнал чтения, который может быть использован для объединения памяти программ и памяти данных во внешнем ОЗУ. При этом необходимо учитывать, что в MCS51 на схемном уровне реализуются пять различных и независимых механизмов адресации для доступа к RPM, RDM, EPM, EDM и блоку регистров специальных функций. Вследствие этого перемещаемая версия прикладной программы, которая отлаживается в среде внешней памяти программ/данных, будет отличаться от загружаемой в RPM (окончательной) версии программы.

Подобный способ организации управления внешней памятью может быть использован в тех применениях MCS51, где требуется оперативная перезагрузка или модификация прикладных программ.

4 Таймеры / счетчики микроконтроллеров семейства 8051

В базовых моделях семейства имеются два программируемых 16-битных таймера/счетчика (T/C0 и T/C1), которые могут быть использованы как в качестве таймеров, так и в качестве счетчиков внешних событий. В первом случае содержимое соответствующего таймера/счетчика (далее для краткости T/C) инкрементируется в каждом машинном цикле, т.е. через каждые 12 периодов колебаний кварцевого резонатора, во втором оно инкрементируется под воздействием перехода из 1 в 0 внешнего входного сигнала, подаваемого на соответствующий (T0,T1) вывод микро-ЭВМ 8051. Опрос значения внешнего входного сигнала выполняется в момент времени S5P2 каждого машинного цикла. Содержимое счётчика 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 S3P3 в цикле, следующем за тем, в котором был обнаружен переход сигнала из 1 в 0. Так как на распознавание периода требуются два машинных цикла, максимальная частота подсчета входных сигналов равна 1/24 частоты резонатора. На длительность периода входных сигналов ограничений сверху нет. Для гарантированного прочтения входной сигнал должен удерживать значение 1, как минимум, в течение одного машинного цикла микро-ЭВМ.

Для управления режимами работы T/C и для организации их взаимодействия с системой прерываний используются два регистра специальных функций (TMOD и TCON), описание которых приведено в табл. 7 и 8 соответственно.

4.1 Регистр режима работы таймера/счетчика TMOD

Таблица 7

Регистр режима работы таймера/счетчика TMOD [1]

Символ	Позиция	Имя и назначение		
GATE	TMOD.7 для T/C1 и TMOD.3 для T/CO	Управление блокировкой. Если бит установлен, то таймер/счетчик "x" разрешен до тех пор, пока на входе "INTx" высокий уровень и бит управления "TRx" установлен. Если бит сброшен, то T/C разрешается, как только бит управления "TRx" устанавливается		
C/T	TMOD.6 для T/C1 и TMOD.2 для T/CO	Бит выбора режима таймера или счетчика событий. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если; установлен, то работает счетчик от внешних сигналов на входе "Tx"		
M1	TMOD.5 для T/C1 и TMOD.1 для T/CO	Режим работы		
		M1	M0	
		0	0	Таймер BE48. "TLx" работает как 5-битный предделитель
		0	1	16 битный таймер/счетчик. "THx" и "TLx" включен последовательно
M0	TMOD.4 для T/C1 и TMOD.0 для T/CO	1	0	8-битный авто перезагружаемый таймер/счетчик. "THx" 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. TH0 работает только как 8 битный таймер, и его режим определяется управляющими битами таймера 1

4.2 Регистр управления/статуса таймера TCON

Таблица 8

Регистр управления/статуса таймера TCON [1]

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 INT1. Сбрасывается при обслуживании прерывания
IT1	TCON.2	Бит управления типом прерывания 1. Устанавливается / сбрасывается программно для спецификации запроса INT1 (срез/низкий уровень)
IE0	TCON.1	Флаг фронта прерывания 0. Устанавливается по срезу сигнала INT0. Сбрасывается при обслуживании прерывания
IT1	TCON.0	Бит управления типом прерывания 0. Устанавливается / сбрасывается программно для спецификации запроса INT0 (срез/низкий уровень)

4.3 Режимы работы таймеров-счетчиков

Как следует из описания управляющих бит TMOD, для обоих T/C режимы работы 0, 1 и 2 одинаковы. Режимы 3 для T/C0 и T/C1 различны. Рассмотрим кратко работу T/C в каждом из режимов [1].

- **Режим 0.** Перевод любого T/C в режим 0 делает его похожим на таймер – восьми битный счетчик, к входу которого подключен 5-битный делитель частоты на 32. Работу T/C в режиме 0 на примере T/C1 иллюстрирует рис. 12,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 T/C1), когда управляющий бит TR1 установлен в 1 либо управляющий бит GATE (блокировка) равен 0, либо на внешний вывод запроса прерывания 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 T/C в этом режиме такая же, как и в режиме 0, за исключением того, что таймерный регистр имеет разрядность 16 бит.
- **Режим 2.** В этом режиме 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 (см. рис. 12,б), но и автоматически перезагружает в TL1 содержимое старшего байта (ТН 1) таймерного регистра, которое предварительно было задано программным путем. Перегрузка оставляет содержимое ТН1 неизменным. В режиме 2 T/C0 и T/C1 также работают совершенно одинаково.
- **Режим 3.** В режиме 3 T/C0 и T/C1 работают по-разному. T/C1 сохраняет неизменным свое текущее содержимое. Иными словами, эффект такой же как и при сбросе управляющего бита TR1 в 0. Работу T/C0 иллюстрирует рис. 12,в. В режиме 3 TL0 и ТН0 функционируют как два независимых 8-битных счетчика. Работу TL0 определяют управляющие биты T/C0 (C/T, GATE TR0), входной сигнал INT0 и флаг переполнения TF0. Работу ТН0, который может выполнять только функции таймера (подсчёт машинных циклов микро-ЭВМ), определяет управляющий бит TR1. При этом ТН0 использует флаг переполнения TF1. Режим 3 используется в тех случаях, когда требуется наличие дополнительного восьми битного таймера или счетчика событий. Можно считать, что в этом режиме микро-ЭВМ 8051 имеет в своем составе три таймера/счетчика. В случае же, если T/C0 используется в

режиме 3, T/C1 может быть или выключен, или переведен в режим 0, 1 или 2, или может быть использован последовательным портом в качестве генератора частоты передачи.

В модернизированных моделях микроконтроллеров семейства MCS-51 может иметься третий таймер счетчик T/C2 и (или) блок программных счетчиков PCA, которые тоже могут быть использованы для отсчета временных интервалов.

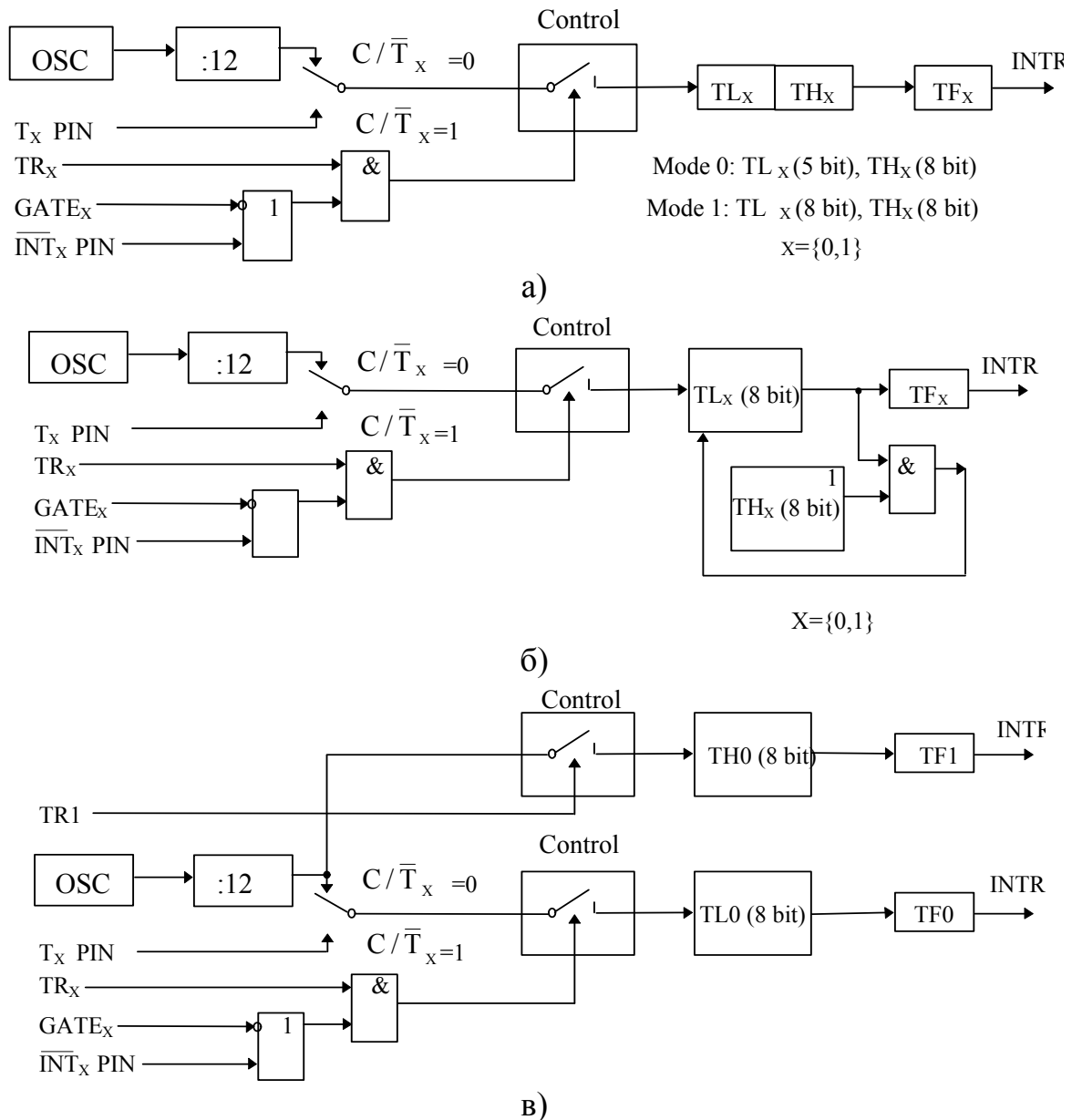


Рис. 12. Таймеры/счётчики T/C0 и T/C1 в режимах 0, 1, 2 и 3:
 а – T/C0 и T/C1 в режимах 0 и 1; б - T/C0 и T/C1 в режиме 2; в - T/C0 в режиме 3

5 Последовательный интерфейс микроконтроллера 8051

Через универсальный асинхронный приемопередатчик **UART** (Universal Asynchronous Receiver-Transmitter) осуществляются прием и передача информации, представленной последовательным кодом (младшими битами вперед), в полном дуплексном режиме обмена. В состав приемопередатчика, называемого часто последовательным портом входят принимающий и передающий сдвигающие регистры, а также специальный буферный регистр (SBUF) приемопередатчика [2, 3, 5].

Кроме того, работой последовательного порта управляют два служебных регистра:

- Регистр управления/статуса приемопередатчика **SCON**;
- Бит **SMOD** регистра управления мощностью **PCON**.

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. Но если к моменту окончания приема байта предыдущий не был считан из **SBUF**, то он будет потерян.

Последовательный порт 8051 может работать в четырех различных режимах.

- **Режим 0.** Информация и передается, и принимается через вывод входа приемника (**RXi TXi**). Принимаются или передается 8 бит данных. Через вывод выхода передатчика (**TXD**); выдаются импульсы сдвига, которые сопровождают каждый бит. Частота передачи бита информации равна $1/12$ частоты кварцевого резонатора
- **Режим 1.** В этом режиме передаются через вывод **TXD** или принимаются через **RXD** 10 бит информации: старт-бит (0), 8 бит данных и стоп-бит (1) при приеме информации в бит **RB8** регистра управления/статуса приемопередатчика **SCON** заносится стоп-бит. Скорость приема/передачи — величина переменная и задается таймером.
- **Режим 2.** В этом режиме через вывод **TXD** передаются или через **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**). При приеме девятый бит данных помещается в бит **RB8 SCON**, а

стоп-бит, в отличие от режима 1, теряется. 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.** совпадает с режимом 2 во всех деталях, за исключением частоты приема/передачи, которая является величиной переменной и задается таймером.

Во всех случаях передача инициализируется инструкцией, в которой данные перемещаются в SBUF. Прием инициализируется при обнаружении перепада из 1 в 0 на входе приемника. При этом в режиме 0 этот переход должен сопровождаться выполнением условий $R1 = 0$ и $REN = 1$ (см. табл. 9), а для остальных режимов – $REN = 1$.

5.1 Регистр управления/статуса приемопередатчика SCON

Управление режимом работы приемопередатчика 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 (R1 и T1).

Функциональное назначение бит регистра управления/статуса UART приводится в таблице 9 [1].

Прикладная программа путем загрузки в старшие биты регистра SCON двухбитного кода определяет режим работы приемопередатчика. Во всех четырех режимах работы передача инициализируется любой командой, в которой буферный регистр SBUF указан как получатель байта. Как уже отмечалось, прием в режиме 0 осуществляется при условии, что $R1 = 0$ и $REN = 1$, в остальных режимах - при условии, что $REN = 1$.

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

Флаг прерывания передатчика TI устанавливается аппаратно в конце периода передачи стоп-бита во всех режимах. Соответствующая подпрограмма обслуживания прерывания должна сбрасывать бит TI.

Флаг прерывания приемника RI устанавливается аппаратно 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 9

Функциональное назначение бит регистра управления/статуса
приемопередатчика SCON

Символ	Позиция	Имя и назначение		
SM0	SCON.7	Биты управления режимом работы приемопередатчика. Устанавливаются/сбрасываются программно		
SM1	SCON.6	SM0	SM1	Режим работы приемопередатчика
		0	0	Сдвигающий регистр расширения ввода/вывода
		0	1	8 битовый приемопередатчик, изменяемая скорость передачи
		1	0	9 битовый приемопередатчик. Фиксированная скорость передачи
		1	1	9 битовый приемопередатчик, изменяемая скорость передачи
SM2	SCON.5	Бит управления режимом приемопередатчика. 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 9-битового передатчика		
RB8	SCON.2	Прием бита 8. Устанавливается/сбрасывается аппаратно для фиксации девятого принимаемого бита в режиме 9-битового приемника		
TI	SCON.1	Флаг прерывания передатчика. Устанавливается аппаратно при окончании передачи байта. Сбрасывается программно после обслуживания прерывания		
RI	SCON.0	Флаг прерывания приемника. Устанавливается аппаратно при приеме байта. Сбрасывается программно после обслуживания прерывания		

5.2 Работа UART в мультиконтроллерных системах

В системах децентрализованного управления, которые используются для управления и регулирования в топологически распределенных объектах, возникает задача обмена информацией между множеством микроконтроллеров, объединенных в локальную вычислительно-управляющую сеть. Как правило, локальные сети на основе Intel 8051 имеют магистральную архитектуру с разделяемым моноканалом (коаксиальный кабель, витая пара, оптическое волокно), по которому осуществляется обмен информацией между контроллерами [5].

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

Механизм обмена построен на том, что в режимах 2 и 3 программируемый девятый бит данных при приёме фиксируется в бите RB8. UART может быть запрограммирован таким образом, что при получении стоп-бита прерывание от приёмника 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. Ведущий МК всем ведомым передаёт широковещательное сообщение с байтом-идентификатором абонента, которое 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 байта-идентификатора ($RB8 = 1$) во всех ведомых МК произошли прерывания прикладных программ и вызов подпрограммы сравнения байта-идентификатора с кодом собственного сетевого адреса. Адресуемый МК сбрасывает свой SM2 и готовится к приёму блока данных. Остальные ведомые микроконтроллеры оставляют неизменными свои $SM2 = 1$ и передают управление основной программе. При $SM2 = 1$ информационные байты в сети, передаваемые по моноканалу и поступающие в UART ведомых МК, прерывания не вызывают, т.е. игнорируются.

В режиме 1 UART автономного микроконтроллера бит SM2 используется для контроля истинности стоп-бита (при $SM2 = 1$ прерывание не произойдёт до тех пор, пока не будет получено

истинное (единичное) значение стоп-бита). В режиме 0 бит SM2 не используется и должен быть сброшен.

5.3 Скорость приема/передачи информации через последовательный порт

Скорость приема/передачи, т.е. частота работы приемопередатчика в различных режимах, определяется различными способами [1].

В режиме 0 частота передачи зависит только от резонансной частоты кварцевого резонатора $f_{рез}$:

$$f_0 = f_{рез} / 12. \quad (1)$$

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 PCON (таблице 10).

Табл. 10

Регистр управления мощностью PCON

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

Примечание. При одновременной записи 1 в PD и IDL бит PD имеет преимущество. Сброс содержимого PCON выполняется путем загрузки в него кода 0XXX0000.

В режиме 2 частота передачи определяется выражением

$$f_2 = (2^{SMOD} / 64) \cdot f_{рез}. \quad (2)$$

Иными словами, при SMOD = 0 частота передачи равна 1/64 частоты $f_{рез}$, а при SMOD = 1 – 1/32 частоты $f_{рез}$.

В режимах 1 и 3 в формировании частоты передачи, кроме управляющего бита SMOD, принимает участие таймер 1. При этом частота передачи f зависит от частоты переполнения f_{OVL1} определяется следующим образом:

$$f_{1,3} = (2^{SMOD}/32) \cdot f_{OVL1} \quad (3)$$

Прерывание от таймера 1 в этом случае должно быть заблокировано. Сам же таймер может работать как в режиме таймера, так и в режиме счетчика. Номер режима (0, 1, 2) роли не играет. Наиболее типично использование его в режиме таймера с автоперезагрузкой (старшая тетрада TMOD = 0010B). При этом частота передачи определяется выражением:

$$f_{1,3} = (2^{SMOD}/32) \cdot (f_{рез}/12) \cdot (256 - (TH1)) \quad (4)$$

В таблице 11 приводится описание способов настройки таймера 1 для получения типовых частот передачи данных через UART.

Табл. 11

Настройка таймера 1 для управления частотой работы приемопередатчика

Частота приема/передачи (BAUD RATE)	Частота резонатора МГц	Таймер/счетчик 1			
		SMOD	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
Режим 1, 3: 62,2 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	0F4H
137,5 Гц	11,059	0	0	2	1DH
110 Гц	6	0	0	2	72H
110 Гц	12	0	0	1	0FEEDH

Отметим, что скорости приема и передачи могут различаться.

Предельно низких частот приемопередачи можно достичь при использовании таймера в режиме 1 (16-битный таймер) и разрешении прерываний от таймера (старший полубайт TMOD = 0001B). Перезагрузка 16-битного таймера должна осуществляться программным путем.

5.4 Особенности работы UART в различных режимах

Режим 0. На рис. 13 показана упрощённая структурная схема UART, а на рис. 14 – временная диаграмма его работы в режиме 0. Данные передаются и принимаются через вывод RXD. Через вывод TXD выдаются синхросигналы сдвига [1].

Передача начинается любой командой, по которой в SBUF поступает байт данных. В момент времени S6P2 устройство управления MCS51 по сигналу *Запись в буфер* записывает байт в сдвигающий регистр передатчика, устанавливает триггер девятого бита и запускает блок управления передачей, который через один

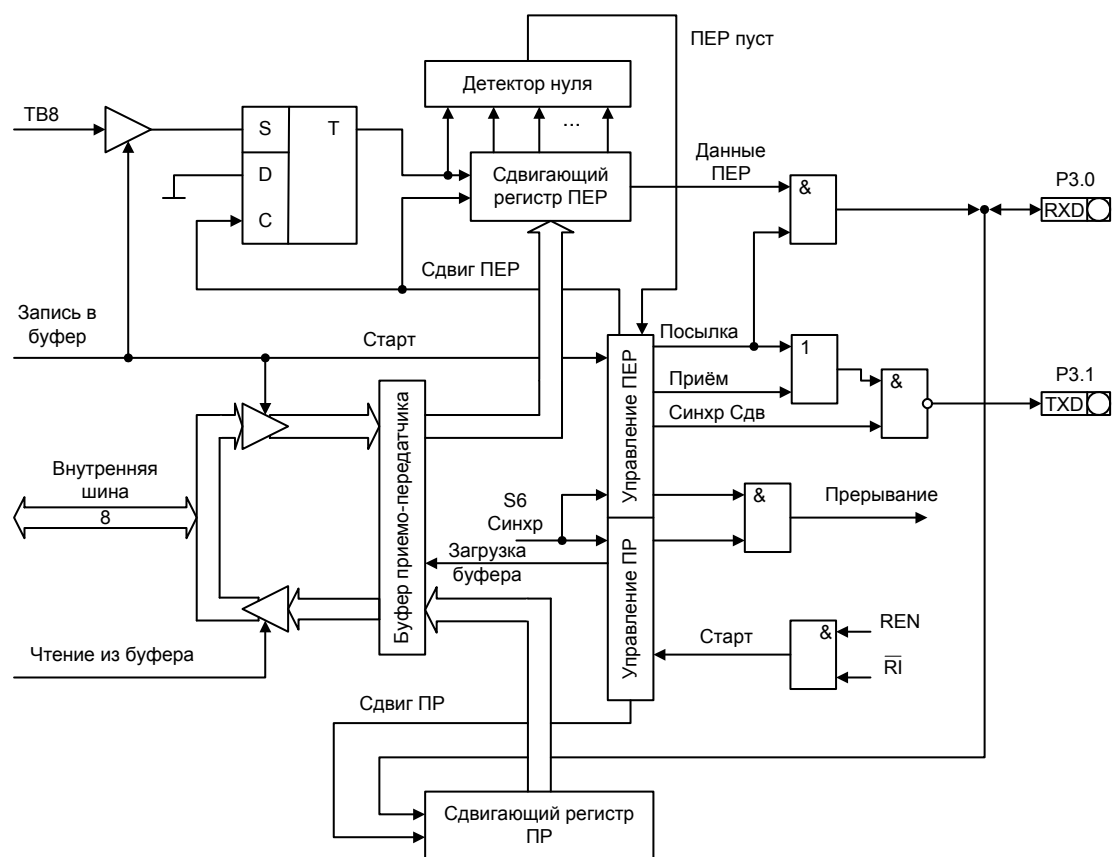


Рис. 13. Структурная схема UART в режиме 0

машинный цикл вырабатывает разрешающий сигнал *Посылка*. При этом в момент S6P2 каждого машинного цикла содержимое сдвигающего регистра сдвигается вправо (младшими битами вперёд) и поступает на выход RXD. В освобождающиеся старшие биты сдвигающего регистра передатчика записываются нули. При получении от детектора нуля *Передатчик пуст*, блок управления передатчиком снимает сигнал *Посылка* и устанавливает флаг TI

(момент S1P1 десятого машинного цикла после поступающего сигнала *Запись в буфер*).

Приём начинается при условии $REN = 1$ и $RI = 0$. В момент S6P2 следующего машинного цикла блок управления приёмником формирует разрешающий сигнал *Приём*, по которому на выход TXD передаются синхросигналы сдвига и в сдвигающем регистре приёмника 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 приёмником переписывает содержимое сдвигающего регистра в буфер, снимает разрешающий сигнал *Приём* и устанавливает флаг RI.

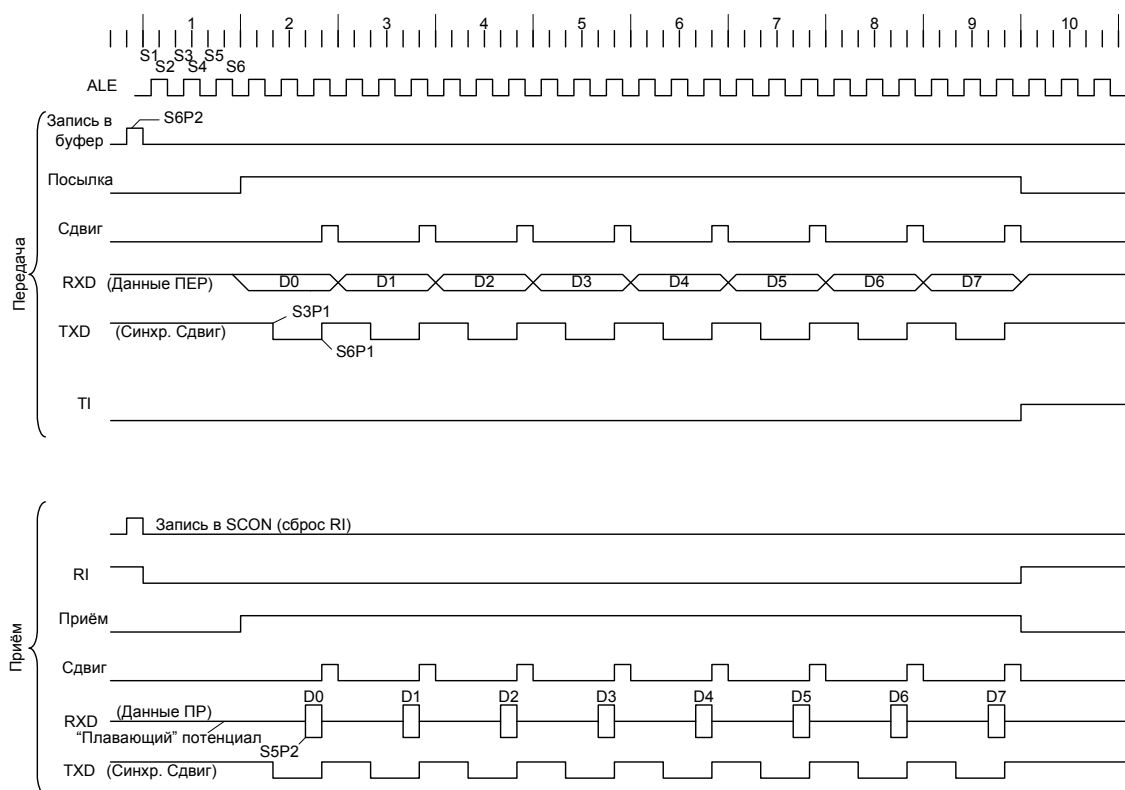


Рис. 14. Временная диаграмма работы UART в режиме 0

Режим 1. На рис. 15 показана структурная схема UART, а на рис. 16 – временные диаграммы работы UART при приёме и передаче данных. Через вывод TXD UART передаёт, а с вывода RXD принимает 10 бит: старт-бит (0), 8 бит данных и стоп-бит (1). При приёме стоп-бит поступает в бит RB8 регистра SCON [1].

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 счётчика.

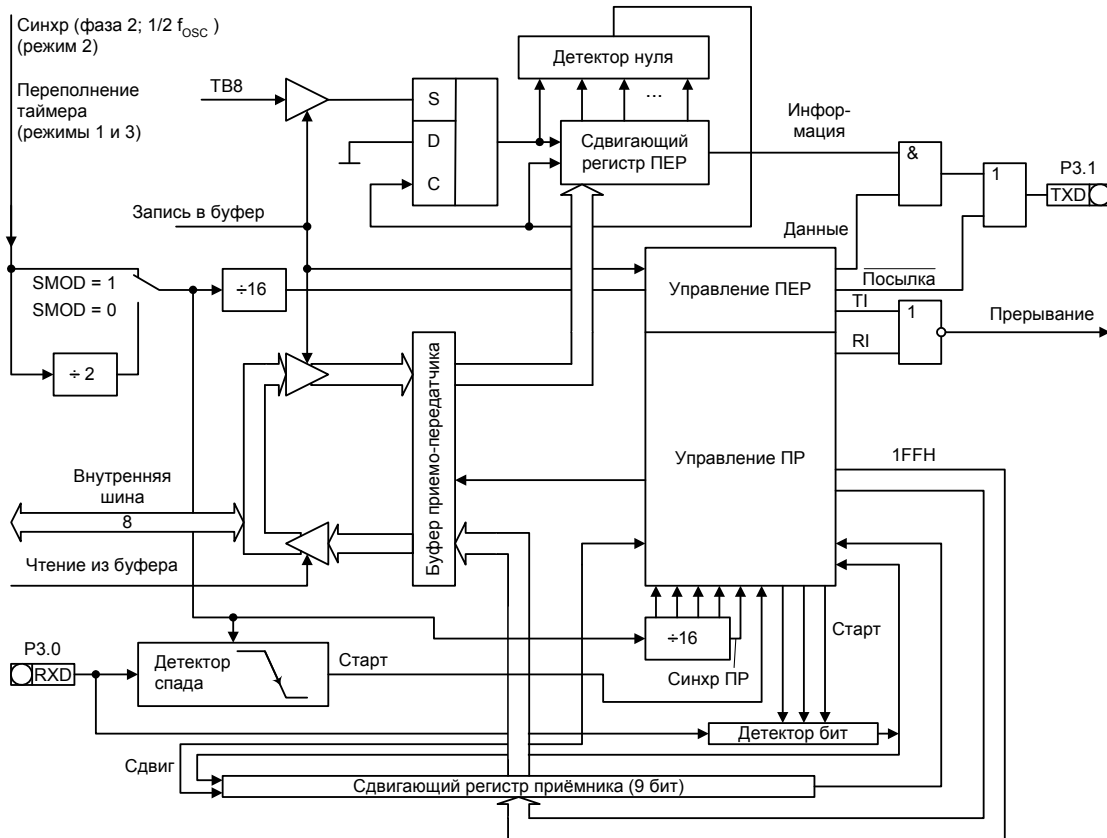


Рис. 15. Структурная схема UART в режимах 1,2 и 3

Приём начинается при обнаружении перехода сигнала на входе RXD из состояния 1 в состояние 0. Для этого под управлением внутреннего счётчика вход RXD опрашивается 16 раз за период представления бита. Как только переход из 1 в 0 на входе RXD обнаружен, в сдвигающий регистр приёмника загружается код 1FFH, внутренний счётчик 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 счётчика. В состояниях 7, 8 и 9 счётчика 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 трёх замеров (мажоритарное

голосование по принципу “два из трёх”). 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 приёмника, и продолжается приём остальных бит посылки. Блок управления приёмом 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 приёмом вновь начинает отыскивать переход из 1 в 0 на входе RXD.

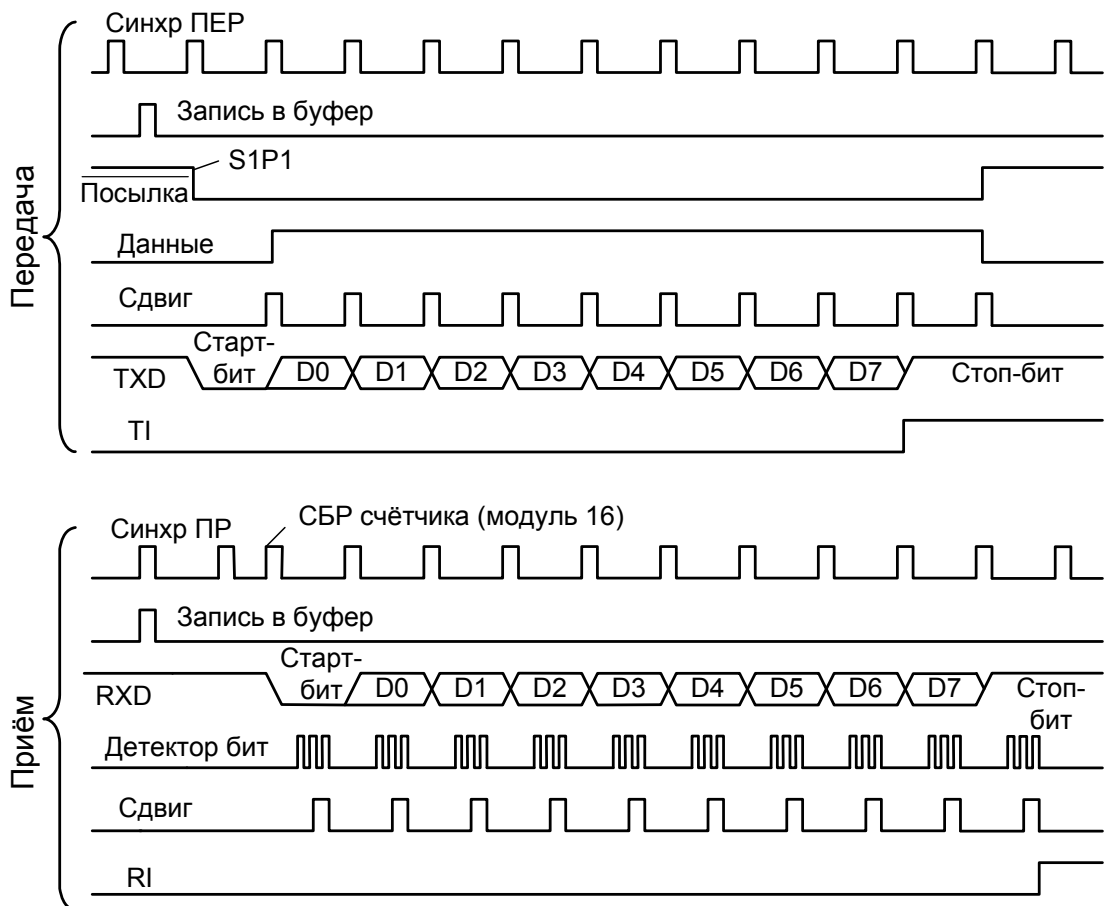


Рис. 16. Временная диаграмма работы UART в режиме 1

Режимы 2, 3. Через вывод TXD UART передаёт или с вывода RXD принимает 11 бит: старт-бит (0), 8 бит данных, программируемый девятый бит и стоп-бит (1). На временной диаграмме (рис. 17) показана работа UART при передаче и приёме данных в режимах 2 и 3. Как видно, режимы 2 и 3 отличается от режима 1 только наличием девятого программируемого бита. Вследствие этого несколько изменяются условия окончания цикла приёма: блок управления приёмником 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 [1].

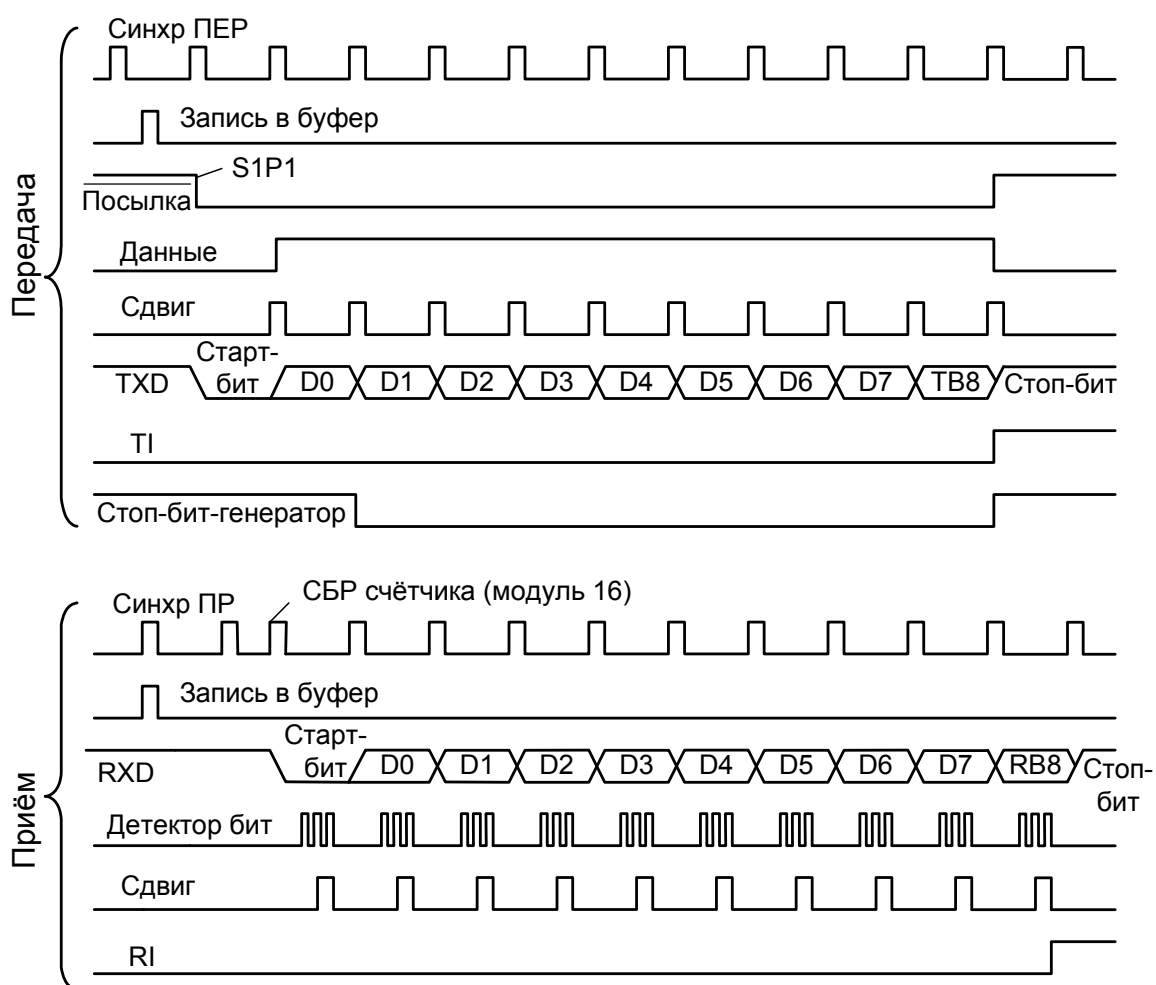


Рис. 17. Временная диаграмма работы UART в режимах 2 и 3

6 Система прерываний микроконтроллера 8051

Упрощенная схема прерываний микро-ЭВМ 8051 показана на рис. 18 [1].

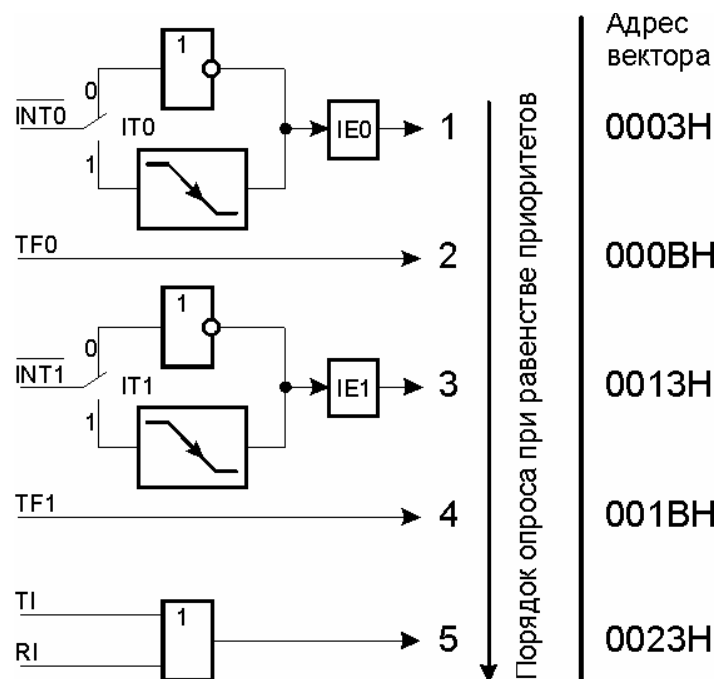


Рис. 18. Схема прерываний

Внешние прерывания INT 0 и INT 1 могут быть вызваны либо уровнем, либо переходом сигнала из 1 в 0 на входах 8051 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 программы обслуживания прерывания. Сброс этих флагов выполняется аппаратно только в том случае, если прерывание было вызвано по переходу (срезу) сигнала. Если же прерывание вызвано уровнем входного сигнала, то сбросом флага I должна управлять 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 передатчика аппаратно, но сбрасываться должны программным путем [1, 2, 5].

Прерывания могут быть вызваны или отменены программой, так как все названные флаги программно доступны и могут быть

установлены/ сброшены программой с тем же результатом, как если бы они были установлены/сброшены аппаратными средствами.

В блоке регистров специальных функций есть два регистра, предназначенных для управления режимом прерываний IE и уровнями приоритета IP, описанные в таблице 12 и 13 соответственно. Возможность программной установки/сброса любого управляющего бита в этих двух регистрах делает систему прерываний 8051 исключительно гибкой.

В более сложных модификациях микроконтроллеров семейства MCS-51 количество периферийных устройств увеличено, что приводит к необходимости использовать один вектор прерывания для нескольких устройств (разделение подпрограмм обслуживания прерываний в этом случае необходимо реализовать программно), либо добавить еще два регистра - режима (маски) и приоритета прерываний.

6.1 Регистр масок прерывания (IE)

Таблица 12

Регистр масок прерывания (IE) [1]

Символ	Позиция	Имя и назначение
EA	IE.7	Снятие блокировки прерывания. Сбрасывается, программно для запрета всех прерываний независимо от состояний IE.4 - IE.0
	IE.6	Не используется
	IE.5	Не используется
ES	IE.4	Бит разрешения прерывания, от приемопередатчика Установка/сброс программой для разрешения/запрета прерываний от флагов TI или RI .
ET1	IE.3	Бит разрешения прерывания от таймера. Установка/сброс программой для разрешения/запрета прерываний от таймера 1
EX1	IE.2	Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерывания 1
ET0	IE.1	Бит разрешения прерывания от таймера 0. Установка/сброс программой для разрешения/запрета прерываний от таймера 0 .
EX0	IE.0	Бит разрешения внешнего прерывания 0. Установка/сброс программой для разрешения/запрета прерывания 0

6.2 Регистр приоритетов прерываний (IP)

Таблица 13

Регистр приоритетов прерываний (IP) [1]

Символ	Позиция	Имя и назначение
-	IP.7 - IP.5	Не используется
PS	IP.4	Бит приоритета приемопередатчика. Установка/сброс программой для присваивания прерыванию от приемопередатчика 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 INT1
PT0	IP.1	Бит приоритета таймера 0. Установка/сброс программой для присваивания прерыванию от таймера 0 высшего/низшего приоритета
PX0	IP.0	Бит приоритета внешнего прерывания 0. Установка/сброс программой для присваивания высшего/низшего приоритета внешнему прерыванию INT0

6.3 Выполнение подпрограммы прерывания

Система прерываний формирует аппаратный вызов (LCALL) соответствующей подпрограммы обслуживания, если она не заблокирована одним из следующих условий [3, 4, 5]:

- в данный момент обслуживается запрос прерывания равного или высокого уровня приоритета;
- текущий машинный цикл – не последний в цикле выполняемой команды;
- 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 указанных выше условий не получил обслуживания и к моменту окончания блокировки уже 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 него адрес вектора соответствующей подпрограммы обслуживания. По адресу вектора должна быть расположена команда безусловной передачи управления (JMP) к начальному адресу подпрограммы обслуживания прерывания. В случае необходимости она должна начинаться командами записи в стек (PUSH) слова состояния программы (PSW), аккумулятора, расширителя, указателя данных и т.д. и должна заканчиваться командами восстановления из стека (POP). Подпрограммы обслуживания прерывания должны завершаться 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 снимут блокировку прерываний, что приводит к необходимости иметь программный механизм анализа окончания процедуры обслуживания данного прерывания [1].

7 Режимы микроконтроллера 8051 с пониженным энергопотреблением

Во многих вариантах применения микро-ЭВМ энергопотребление является одним из основных параметров. В этих случаях целесообразно использовать КМОП версии микро-ЭВМ. В них предусмотрены дополнительные возможности снижения энергопотребления, отсутствующие в стандартных n-МОП изделиях. Ранее выпускались и n-МОП версии микро-ЭВМ, имевшие режимы работы с пониженным энергопотреблением. В настоящее время их выпуск почти повсеместно прекращен. Однако время от времени они все же будут попадать в руки радиолюбителей, поэтому трудно гарантировать наличие или отсутствие этих режимов в тех или иных конкретных n-МОП изделиях.

МКОП микро-ЭВМ имеют два режима с пониженным потреблением тока: режим холостого хода (ХХ) и режим выключенного напряжения питания (ВНП). Последний иногда называют режимом пониженного энергопотребления. По мнению авторов настоящей статьи, это вряд ли корректно, так как режим ХХ также характеризуется пониженным энергопотреблением в сравнении с обычным режимом работы микро-ЭВМ. дословно второй режим, называемый в англоязычной литературе режимом "Power Down Mode", можно перевести так, как указано выше. Этот перевод корректнее отражает суть режима - в нем допустимо вообще отключить от микро-ЭВМ питающее напряжение, подаваемое на вход U_{cc} (вывод 40 микро-ЭВМ). для сохранения содержимого внутреннее ОЗУ в этом случае нужно запитать от резервного источника. Резервное питающее напряжение подают через вход RST (вывод 9).

В режиме ХХ ($IDL = 1$) генератор микро-ЭВМ работает, тактовые сигналы поступают на систему прерываний, последовательный порт и таймеры/счетчики. Все регистры сохраняют свое содержимое, на выводах всех портов удерживаются логические состояния, которые были на них в момент перехода в режим ХХ. Однако сигнал генератора синхронизирующий CPU, отключается. В режиме ВНП ($PD = 1$) генератор останавливается. Прекращается тактирование не только CPU, но и последовательного порта, таймеров/ счетчиков, системы прерываний. Как и в режиме ХХ, состояние регистров, резидентного ОЗУ и выводов портов остается неизменным.

Режимы ХХ и ВНП активизируются при установке соответствующих битов в специальном функциональном регистре-регистре управления мощностью PCON. Адрес этого регистра 87H, его описание приведено в таблице 14 [1].

7.1 Регистр управления мощностью PCON

Табл. 14

Регистр управления мощностью PCON

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

Отметим следующие особенности этого регистра. Если одновременно установлены в 1 биты IDL и PD, то последний имеет преимущество - микро-ЭВМ переходит в режим ВМП. Содержимое регистра PCON после сброса - 0XXX0000.

В n-МОП версиях микроЭВМ регистр PCON содержит, как правило, только SMOD. Остальные четыре бита присутствуют только в КМОП устройствах. Пользовательские программы никогда не должны заносить 1 в незанятые биты (PCON4 - PCON6), так как они могут использоваться в следующих модификациях микроЭВМ. Последнее распространяется на все регистры и на все адреса в области регистров специальных функций, которые не заняты регистрами. Разработчики предполагают использовать их в новых изделиях. Однако для достижения совместимости уже разработанного программного обеспечения с новыми изделиями дополнительные возможности последних будут включаться установкой в 1 битов в соответствующих регистрах. Поэтому пользовательская программа, устанавливающая в 1 неиспользуемые биты, будет нормально работать на микро-ЭВМ, имеющихся в наличии сегодня, но вовсе не обязательно будет работать на новых микро-ЭВМ, несмотря на их полную программную совместимость с семейством 8051.

7.2 Режим ХХ

В этот режим микро-ЭВМ переводится любой командой, устанавливающей в 1 бит PCON.0. Она оказывается последней в цепочке выполняемых команд: в режиме ХХ выполнение программы приостанавливается, так как на CPU перестает поступать сигнал тактового генератора. Однако содержимое внутреннего ОЗУ и регистров специальных функций остается неизменным, выводы портов удерживают значения, которые были на них до перехода в режим ХХ, на таймеры/счетчики, приемопередатчик и на систему прерываний продолжают поступать тактовые сигналы. На выводах ALE и PSEN устанавливаются сигналы единичного уровня.

Состояние выводов портов зависит от типа ОЗУ, с которым микро-ЭВМ обменивалась информацией перед тем как перейти в режим ХХ. При работе с внутренним ОЗУ на выводах портов присутствуют данные из соответствующих SFR (естественно, если порт в режиме вывода информации). При работе с внешним ОЗУ выводы порта 0 переходят в высокоимпедансное состояние, а на выводах порта 2. сохраняется адресная информация. На выводах портов 1 и 3 присутствуют данные из SFR портов.

Вывести микро-ЭВМ из режима ХХ можно двумя способами. Так, вызов любого из прерываний приведет к аппаратному стиранию бита PCON.0, прекращающему ХХ. Прерывание будет обслужено, и очередной после RETI выполняемой командой будет та, которая следует за командой, приведшей к переходу микро-ЭВМ в режим ХХ. флаги GF0 и GF1 могут использоваться для индикации того, произошло ли прерывание во время нормальной работы или во время ХХ. Например, команда, запускающая этот режим, может также устанавливать один или оба флага. Когда ХХ прекращен прерыванием, сервисная программа прерывания может проверять состояние флагов.

Другой способ прекращения ХХ - с помощью аппаратного сброса. Поскольку синхрогенератор продолжает работать, аппаратный сброс должен поддерживаться в активном состоянии, только в течение двух машинных циклов (24 периодов колебаний).

Сигнал сброса стирает бит PCON.0. В этот момент CPU возобновляет выполнение программы с самого начала. Как показано на рис. 14, перед началом отработки алгоритма внутреннего запуска могут иметь место два или три машинных цикла выполнения программы. Встроенное в микросхему устройство в это время препятствует доступу к внутреннему ОЗУ, но доступ к выводам порта не ограничен. Чтобы исключить возможность появления неопределенных выходных сигналов на выводах порта, команда, следующая за вызывающей ХХ,

не должна быть командой, записывающей информацию в SFR порта или во внешнее ОЗУ данных. Напомним, что после аппаратного сброса содержимое SFR переопределяется [1].

7.3 Режим ВНП

Команда, устанавливающая в 1 бит PCON.1, переводит микро-ЭВМ в режим ВНП. В нем генератор микро-ЭВМ, как отмечалось, останавливается, прекращает функционирование не только CPU, но и таймеры/счетчики, приемопередатчик, система прерываний. При наличии основного или резервного источника питающего напряжения встроенное ОЗУ и регистры SFR сохраняют свое содержимое. Состояние портов не отличается от состояния при переходе в режим XX (см. выше). Однако в отличие от режима XX на выводах ALE и PSEN устанавливаются сигналы с нулевым уровнем.

Единственный способ выйти из этого режима - аппаратный сброс. Он переопределяет содержимое всех SFR, но не меняет содержимого встроенного ОЗУ.

В рассматриваемом режиме напряжение питания U_{cc} может снижаться вплоть до 2 В. Следует, однако, позаботиться, чтобы снижалось оно не раньше перехода микро-ЭВМ в режим ВНП и восстанавливалось до прежнего значения прежде, чем она выйдет из этого режима. Сигнал сброса, возвращающий микро-ЭВМ в обычный режим, не должен подаваться раньше, чем U_{cc} достигнет своего рабочего значения, и должен поддерживаться в активном состоянии достаточно долго, чтобы генератор успел запуститься и его колебания стабилизировались (обычно – не менее 10 мс) [1].

8 Система команд микроконтроллера семейства 8051

8.1 Общая характеристика

Микро-ЭВМ рассматриваемого семейства являются типичными микропроцессорными устройствами с архитектурой CISC - со стандартным набором команд. Поэтому их система команд довольно обширна и включает в себя 111 основных команд. Их длина – один, два или три байта, причем большинство из них (94%) – одно- или двухбайтные. Все команды выполняются за один или два машинных цикла (соответственно 1 или 2 мкс при тактовой частоте 12 МГц), исключение – команды умножения и деления, которые выполняются за четыре машинных цикла (4 мкс). Микро-ЭВМ семейства 8051 используют прямую, непосредственную, косвенную и неявную, адресацию данных.

В качестве операндов команд микро-ЭВМ семейства 8051 могут использовать отдельные биты, четырехбитные цифры, байты и двухбайтные слова.

Все эти черты обычны для набора команд любого CISC-процессора и по сравнению с RISC набором команд обеспечивает большую компактность программного кода и увеличение быстродействия при выполнении сложных операций.

В то же время, набор команд семейства 8051 имеет несколько особенностей, связанных с типичными функциями выполняемыми микроконтроллерами - управлением, для которого типичным является оперирование с одноразрядными двоичными сигналами, большое число операций ввода вывода и ветвлений программы.

Наиболее существенная особенность системы команд рассматриваемых микро ЭВМ это возможность адресации отдельных бит в резидентной памяти данных. Кроме того, как отмечалось, некоторые регистры блока регистров специальных функций также допускают адресацию отдельных бит. Карты адресов отдельных бит в резидентной памяти данных и в блоке регистров специальных функций представлены в табл. 1 и 2 соответственно [1, 3, 5].

8.1.1 Типы команд

Всего микро-ЭВМ выполняют 13 типов команд, они приведены в таблице. Как следует из нее, первый байт команды всегда содержит код операции (КОП), а второй и третий (если они присутствуют в команде) – адреса операндов или их непосредственные значения. На рис. 19 показаны 13 типов команд MCS51 [1].

	1-й байт D7 ... D0		
1	КОП	2-й байт D7 ... D0	
2	КОП	#d	
3	КОП	ad	
4	КОП	bit	
5	КОП	rel	
6	a10a9 a8	КОП	a7 ... a0 3-й байт D7 ... D0
7	КОП	ad	#d
8	КОП	ad	rel
9	КОП	ads	add
10	КОП	#d	rel
11	КОП	bit	rel
12	КОП	ad16h	ad16l
13	КОП	#d16h	#d16l

Рис. 19. Типы команд MCS51

8.1.2 Обозначения, используемые при описании команд

R_n ($n = 0, 1, \dots, 7$) – регистр общего назначения в выбранном банке регистров;

@ R_i ($i = 0, 1$) – регистр общего назначения в выбранном банке регистров, используемый в качестве регистра косвенного адреса;

ad – адрес прямоадресуемого байта;

ads – адрес прямо адресуемого байта-источника;

add – адрес прямо адресуемого байта-получателя;

ad11 – 11-разрядный абсолютный адрес перехода;

ad16 – 16-разрядный абсолютный адрес перехода;

rel – относительный адрес перехода;

#d – непосредственный операнд;

#d16 – непосредственный операнд (2 байта);

bit – адрес прямо адресуемого бита;

/bit – инверсия прямо адресуемого бита;

A - аккумулятор;

PC – счетчик команд;

DPTR – регистр указатель данных;

() – содержимое ячейки памяти или регистра.

8.1.3 Типы операндов

Состав операндов включает в себя операнды четырёх типов: биты, 4-битные цифры, байты и 16-битные слова.

Микроконтроллер имеет 128 программно-управляемых флагов пользователя. Имеется также возможность адресации отдельных бит блока регистров специальных функций и портов. Для адресации битов используется прямой 8-битный адрес (bit). Косвенная адресация бит невозможна. Карты адресов отдельных битов представлены в таблице 1 и 2.

Четырёхбитные операнды используются только при операциях обмена (команды SWAP и XCHD).

Восьмибитным операндом может быть ячейка памяти программ (PM) или данных (резидентной (RDM) или внешней (EDM)), константа (непосредственный операнд), регистры специальных функций (SFR), а также порты ввода/вывода. Порты и регистры специальных функций 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.

8.1.4 Способы адресации данных

В микроконтроллере используются следующие способы адресации данных: прямая, непосредственная, косвенная и неявная.

При косвенном способе адресации резидентной памяти данных используются все восемь битов адресных регистров R0 и R1.

8.1.5 Флаги результата

Слово состояния программы PSW включает себя четыре флага: C - перенос, AC - вспомогательный перенос (полуперенос), OV - переполнение и P - паритет.

Флаг паритета напрямую зависит от текущего значения аккумулятора. Если число единичных битов аккумулятора нечётное, то флаг P устанавливается, а если чётное - сбрасывается. Все попытки изменить флаг P, присваивая ему новое значение, бесполезны, если содержимое аккумулятора при этом останется неизменным.

Флаг AC устанавливается, если при выполнении операции сложения или вычитания между тетрадами байта (полубайтами) возник перенос или заем.

Флаг C устанавливается, если в старшем бите результата возникает перенос или заем. При выполнении операций умножения и деления флаг C сбрасывается.

Флаг OV устанавливается, если результат операции сложения или 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 таблице 15 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.

Табл. 15.

Команды, модифицирующие флаги результата

Команды	Флаги	Команды	Флаги
ADD	C, OV, AC	CLR C	C = 0
ADDC	C, OV, AC	CPL C	C = NOT(C)
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		

8.1.6 Символическая адресация

При использовании ассемблера для получения объектных кодов программ допускается применение в программах символических имен регистров специальных функций, портов и их отдельных битов (табл. 2).

Для адресации отдельных битов и портов (такая возможность имеется не у всех регистров специальных функций) можно использовать символическое имя бита следующей структуры: <имя регистра или порта>.<номер бита>.

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

8.2 Группы команд

Система команд семейства MCS-51 содержит 111 базовых команд, которые по функциональному признаку можно подразделить на пять групп:

- пересылки данных;
- арифметических операций;
- логических операций;
- операций над битами;
- передачи управления.

Формат команд - одно-, двух- и трехбайтовый, причем большинство команд (94) имеют формат один или два байта. Первый байт любых типа и формата всегда содержит код операции, второй и

третий байты содержат либо адреса операндов, либо непосредственные операнды.

Состав операндов включает в себя операнды четырех типов: биты, тетрады (4 бит), байты и 16-битные слова. Время исполнения команд составляет 1, 2 или 4 машинных цикла. При тактовой частоте 12 МГц длительность машинного цикла составляет 1 мкс, при этом 64 команды исполняются за 1 мкс, 45 команд - за 2 мкс и 2 команды (умножение и деление) - за 4 мкс.

Набор команд MCS-51 поддерживает следующие режимы адресации.

Прямая адресация (Direct Addressing). Операнд определяется 8-битным адресом в инструкции. Эта адресация используется только для внутренней памяти данных и регистров SFR.

Косвенная адресация (Indirect Addressing). В этом случае инструкция адресует регистр, содержащий адрес операнда. Данный вид адресации может применяться при обращении как к внутреннему, так и внешнему ОЗУ. Для указания 8-битных адресов могут использоваться регистры R0 и R1 выбранного регистрового банка или указатель стека SP.

Для 16-битной адресации используется только регистр "указатель данных" (DPTR - Data Pointer).

Регистровая адресация (Register Instructions). Данная адресация применяется для доступа к регистрам R0...R7 выбранного банка. Команды с регистровой адресацией содержат в байте кода операции трехбитовое поле, определяющее номер регистра. Выбор одного из четырех регистровых банков осуществляется программированием битов селектора банка (RS1, RS0) в PSW.

Непосредственная адресация (Immediate constants). Операнд содержится непосредственно в поле команды вслед за кодом операции и может занимать один или два байта (data8, data16).

Индексная адресация (Indexed Addressing). Индексная адресация используется при обращении к памяти программ и только при чтении. В этом режиме осуществляется просмотр таблиц в памяти программ. 16-битовый регистр (DPTR или PC) указывает базовый адрес требуемой таблицы, а аккумулятор указывает на точку входа в нее. Адрес элемента таблицы находится сложением базы с индексом (содержимым аккумулятора).

Другой тип индексной адресации применяется в командах "перехода по выбору" (Case Jump). При этом адрес перехода вычисляется как сумма указателя базы и аккумулятора.

Неявная адресация (Register-Specific Instructions). Некоторые инструкции используют индивидуальные регистры (например,

операции с аккумулятором, DPTR), при этом данные регистры не имеют адреса, указывающего на них; это заложено в код операции.

8.3 Команды передачи данных

Эта группа представлена 28 командами, их краткое описание приведено в приложении 1, где также указаны тип команды (Т) в соответствии с таблицей, ее длина в байтах (Б) и время выполнения в машинных циклах (Ц). Большую часть команд данной группы составляют команды передачи и обмена байтов. Команды пересылки битов представлены в группе команд битовых операций. Все команды данной группы не модифицируют флаги результата, за исключением команд загрузки PSW и аккумулятора (флаг паритета).

По команде MOV выполняется пересылка данных из второго операнда в первый. Эта команда не имеет доступа ни к внешней памяти данных, ни к памяти программ. Для этих целей предназначены команды MOVX и MOVC соответственно. Первая из них обеспечивает чтение/запись байт из внешней памяти данных, вторая – чтение байт из памяти программ.

По команде XCH выполняется обмен байтами между аккумулятором и ячейкой РПД, а по команде XCHD – обмен младшими тетрадами (битами 0 – 3).

Команды PUSH и POP предназначены соответственно для записи данных в стек и их чтения из стека. Размер стека ограничен лишь размером резидентной памяти данных. В процессе инициализации микро-ЭВМ после сигнала сброса или при включении питающего напряжения в SP заносится код 07H. Это означает, что первый элемент стека будет располагаться в ячейке памяти с адресом 08H.

Группа команд пересылок микроконтроллера имеет следующую особенность - в ней нет специальных команд для работы со специальными регистрами: PSW, таймером, портами ввода-вывода. Доступ к ним, как и к другим регистрам специальных функций, осуществляется заданием соответствующего прямого адреса, т.е. это команды обычных пересылок, в которых вместо адреса можно ставить название соответствующего регистра.

Например, чтение PSW в аккумулятор может быть выполнено командой

MOV A, PSW

которая преобразуется Ассемблером к виду

MOV A, 0D0h (E5 D0),

где E5 – код операции, а D0 – операнд (адрес PSW).

8.3.1 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. 20. Передачи данных могут выполняться без участия аккумулятора [1].

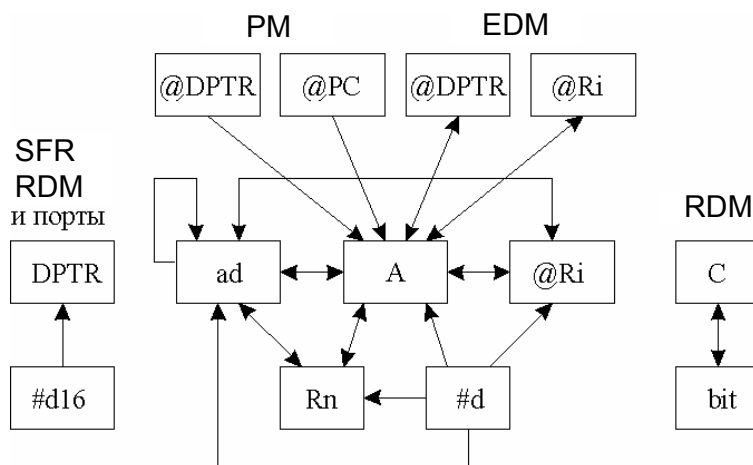


Рис. 20. Граф путей передачи данных

8.3.2 Обращение к аккумулятору

Обращение к аккумулятору может быть выполнено 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 регистров специальных функций, и его адрес указывается во втором байте команды. Использование неявной адресации аккумулятора предпочтительнее, но не всегда возможно, например, при обращении к отдельным битам аккумулятора.

8.3.3 Обращение к внешней памяти данных

При использовании команд `MOVX @Ri` обеспечивается доступ к 256 байтам внешней памяти данных. Существует также режим обращения к расширенной внешней памяти данных, когда для доступа используется 16-битный адрес, хранящийся в регистре-указателе данных `DPTR`. Команды `MOVX @DPTR` обеспечивают доступ к 65536 байтам внешней памяти данных.

8.4 Арифметические операции

Данную группу образуют 24 команды (см. приложение 2), выполняющие операции сложения, десятичной коррекции, инкремента/декремента байтов. Имеются команды вычитания, умножения и деления байтов.

Команды ADD и ADDC допускают сложение аккумулятора с большим числом операндов. Аналогично командам ADDC существуют четыре команды SUBB, что позволяет достаточно просто производить вычитание байтов и многобайтных двоичных чисел.

В микроконтроллере реализуется расширенный список команд инкремента/декремента байтов, команда инкремента 16-битного регистра-указателя данных.

По результату выполнения команд ADD, ADDC, SUBB, MUL и DIV устанавливаются флаги PSW, структура которых приведена в табл. 4.

Флаг C устанавливается при переносе из разряда D7, т. е. в случае, если результат не помещается в восемь разрядов; флаг AC устанавливается при переносе из разряда D3 в командах сложения и вычитания и служит для реализации десятичной арифметики. Этот признак используется командой DAA.

Флаг OV устанавливается при переносе из разряда D6, т. е. в случае, если результат не помещается в семь разрядов и восьмой не может быть интерпретирован как знаковый. Этот признак служит для организации обработки чисел со знаком.

Наконец, флаг P устанавливается и сбрасывается аппаратно. Если число единичных бит в аккумуляторе нечетно, то $P = 1$, в противном случае $P = 0$.

8.5 Логические операции

Данную группу образуют 25 команд (см. приложение 3), реализующих функционально полную систему логических операций над байтами. В микроконтроллере расширено число типов операндов, участвующих в операциях.

Имеется возможность производить операцию “исключающее ИЛИ” с содержимым портов. Команда XRL может быть эффективно использована для инверсии отдельных битов портов.

8.6 Команды передачи управления

К данной группе команд (см. приложение 4) относятся команды, условного и безусловного ветвления, вызова подпрограмм и возврата 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.

8.6.1 Длинный переход

Переход по всему адресному пространству памяти программ. В команде содержится полный 16-битный адрес перехода (ad16). Трехбайтные 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, занимающие меньше места в памяти.

8.6.2 Абсолютный переход

Переход в пределах одной страницы памяти программ размером 2048 байтов. Такие команды содержат только 11 младших битов адреса перехода (ad11). 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 битов заменяются на ad11 из тела команды абсолютного перехода.

8.6.3 Относительный переход

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

8.6.4 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.

8.6.5 Условные переходы

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. В качестве счётчика циклов может использоваться не только регистр, но и прямо-адресуемый байт (например, ячейка резидентной памяти данных).

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 P0 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.

8.6.6 Подпрограммы

Для обращения к подпрограммам необходимо использовать команды вызова подпрограмм LCALL и ACALL. Эти команды в отличие от команд перехода LJMP и 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.

8.7 Операции с битами

Отличительной особенностью данной группы команд (см. приложение 5) является то, что они оперируют с однобитными операндами. В качестве таких операндов могут выступать отдельные биты некоторых регистров специальных функций и портов, а также 128 программных флагов пользователя.

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

9 Развитие микроконтроллерного семейства 8051

Основными направлениями модернизации микроконтроллеров являются:

- увеличение внутренней памяти программ до 64К, причем она может быть масочной, однократно программируемой, электрически перепрограммируемой (EPROM) или с ультрафиолетовым стиранием;
- снижение потребляемой мощности путем применения КМОП-технологии и специальных режимов пониженного энергопотребления;
- переработка архитектуры (снижение числа тактов генератора на один машинный цикл) с целью увеличения быстродействия;
- увеличение тактовой частоты до 100 МГц и выше;
- модификация режимов работы счетчиков-таймеров и последовательного порта;
- размещение на кристалле дополнительного оборудования.

Эти микроконтроллеры выпускаются в PLCC, DIP и QFP корпусах и могут работать в следующих температурных диапазонах:

- коммерческий (0°C — +70°C);
- расширенный (-40°C — +85°C);
- для военного использования (-55°C — +125°C).

Дополнительное оборудование микроконтроллеров:

- Модуль второго UART;
- PCA (Programmable Counter Array) – массив программируемых счётчиков;
- Watchdog timer – сторожевой таймер;
- Модуль АЦП;
- Модуль ЦАП;
- Модуль ШИМ (PWM);
- Интерфейс JTAG;
- Интерфейса CAN;
- Интерфейс LAN (Ethernet);
- Интерфейса USB и т.д.

Перечень микроконтроллеров совместимых с 8051 и фирм изготовителей представлен в приложении 6.

9.1 Семейство 8052

В отличие от 8051 микро-ЭВМ семейства 8052 имеют:

- встроенное ПЗУ или EPROM объемом до 64К;
- встроенное ОЗУ объемом 256 байт;
- дополнительные специальные функциональные регистры;
- таймер/счетчик 2 (далее для краткости – Т/С2), способный работать в режимах защелки, таймера/счетчика, допускающего счет как на увеличение, так и на уменьшение, и генератора скорости передачи в бодах;
- программируемый последовательный интерфейс с детектированием ошибок передачи и автоматическим распознаванием адреса;
- шесть источников прерываний;
- расширенный режим снижения потребляемой мощности; – флаг отключения питания; – режим ONCE.

Микро-ЭВМ 8052 используют стандартный набор команд семейства 8051, их выводы взаимно однозначно соответствуют выводам этих микро-ЭВМ. Отличие заключается лишь в том, что, помимо ввода \ вывода информации, выводы P1.0 и P1.1 8052 могут выполнять альтернативные функции: первый из них играет роль внешнего входа для Т/С2, а второй управляет перезагрузкой/защелкиванием информации в регистры Т/С2.

9.2 Расширение внешней памяти данных до 16МБ в микроконтроллерах совместимых с 8051

Некоторые модели микроконтроллеров совместимые с 8051 (например, линейка микроконтроллеров ADuC8XX фирмы Analog Device) позволяют адресовать 16МБ внешней памяти данных. Данное адресное пространство разбивается на 255 страниц памяти по 64 КБ каждая и условно разделяется на две области. Первая занимает младшие 64 КБ (страница 0) и доступна для выборки команд микроконтроллером. Таким образом, программы могут располагаться только в этих младших 64 КБ. Остальные страницы доступны только для размещения данных. Для адресации ячейки памяти определенной страницы необходимо записать номер страницы в SFR-регистр DPP (Data Pointer Page) – указатель на страницу памяти.

В микроконтроллерной системе с внешним ОЗУ больше 64 КБ необходимо устанавливать второй 8-разрядный регистр, для реализации режима временного мультиплексирования. Данный регистр подключается к порту P2 микроконтроллера для временного хранения старших разрядов адреса, а первый регистр подключается к порту P0

как в случае со стандартным микроконтроллером MCS51 (см. главу «Доступ к внешней памяти»).

9.3 Увеличение быстродействия в микроконтроллерах совместимых с 8051

Изначально наиболее “узким” местом архитектуры MCS51 является относительно медленное выполнение инструкций (для самых “быстрых” из них требуется 12 периодов тактовой частоты). Это ограничивает применение МК семейства в устройствах, требующих повышенного быстродействия и сложных вычислений (16- и 32-битных).

В настоящее время архитектура MCS51 переработана

Однако в настоящее время это «узкое» место классической архитектуры устранено. Многие производители освоили выпуск одноктактных микроконтроллеров полностью совместимых с MCS51, что позволяет при одной и той же частоте тактовых импульсов достичь увеличения производительности микроконтроллера более чем на порядок.

Примерами таких микроконтроллеров являются линейка ADuC84x фирмы Analog Device с быстродействием около 20 миллионов операций в секунду (MIPS), линейка микроконтроллеров DS89C4xx фирмы Dallas Semiconductor с быстродействием до 50 MIPS, линейка микроконтроллеров C8051Fxx фирмы Silicon Laboratories с быстродействием до 100 MIPS и другие.

СПИСОК ЛИТЕРАТУРЫ

1. Сташин В.В., Урусов А.В., Мологонцева О.Ф. Проектирование цифровых устройств на однокристалльных микроконтроллерах. М.: Энергоатомиздат, 1990. – 224 с.
2. Справочник. Микроконтроллеры: архитектура, программирование, интерфейс. Бродин В.Б., Шагурин М.И.М.:ЭКОМ, 1999.
3. Однокристалльные микроЭВМ/ А.В.Боборыкин, Г.П.Липовецкий, Г.В.Литвинский и др. М.: МИКАП, 1994. – 400 с.
4. Хвощ С.Т., Варлинский Н.Н., Попов Е.А. Микропроцессоры и микро-ЭВМ в системах автоматического управления: Справочник/ Под ред. С.Т.Хвоща. Л.: Машиностроение. Ленингр. отделение, 1987. – 640 с.
5. Микропроцессоры. В 3-х кн. Кн. 1. Архитектура и проектирование микро-ЭВМ. Организация вычислительных процессов: Учебник для вузов/ П.В.Нестеров, В.Ф.Шаньгин, В.Л.Горбунов и др.; Под ред. Л.Н.Преснухина. М.: Высшая школа, 1986. – 495 с.

Приложение 1. Система команд MCS51

Команды передачи данных

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Пересылка в аккумулятор из регистра ($n=0\div 7$)	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 РПД ($i=0,1$)	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 РПД 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 РПД из аккумулятора	MOV @Ri, A	1111011i	1	1	1	((Ri)) ← (A)
Пересылка в РПД прямоадресуемого байта	MOV @Ri, ad	0110011i	3	2	2	((Ri)) ← (ad)
Пересылка в РПД 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) ↔ (Rn)
Обмен аккумулятора с прямоадресуемым байтом	XCH A, ad	11000101	3	2	1	(A) ↔ (ad)
Обмен аккумулятора с байтом из РПД	XCH A, @Ri	1100011i	1	1	1	(A) ↔ ((Ri))
Обмен младших тетрад аккумулятора и байта РПД	XCHD A, @Ri	1101011i	1	1	1	(A _{0...3}) ↔ ((Ri) _{0...3})

Примечание: Т – тип команды, Б – длина в байтах, Ц – время выполнения в циклах.

Приложение 2. Система команд MCS51
Арифметические операции

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Сложение аккумулятора с регистром ($n=0\div 7$)	ADD A, Rn	00l01rrr	1	1	1	$(A) \leftarrow (A) + (Rn)$
Сложение аккумулятора с прямоадресуемым байтом	ADD A, ad	00100101	3	2	1	$(A) \leftarrow (A) + (ad)$
Сложение аккумулятора с байтом из РПД ($i = 0, 1$)	ADD A, @Ri	0010011i	1	1	1	$(A) \leftarrow (A) + ((Ri))$
Сложение аккумулятора с константой	ADD A, #d	00100100	2	2	1	$(A) \leftarrow (A) + \#d$
Сложение аккумулятора с регистром и переносом	ADDC A, Rn	00111rrr	1	1	1	$(A) \leftarrow (A) + (Rn) + (C)$
Сложение аккумулятора с прямоадресуемым байтом и переносом	ADDC A, ad	00110101	3	2	1	$(A) \leftarrow (A) + (ad) + (C)$
Сложение аккумулятора с байтом из РПД и переносом	ADDC A, @Ri	0011011i	1	1	1	$(A) \leftarrow (A) + ((Ri)) + (C)$
Сложение аккумулятора с константой и переносом	ADDC A, #d	00110100	2	2	1	$(A) \leftarrow (A) + \#d + (C)$
Десятичная коррекция аккумулятора	DA A	11010100	1	1	1	Если $(A_{0...3}) > 9$ или $((AC)=1)$, то $(A_{0...3}) \leftarrow (A_{0...3}) + 6$, затем если $(A_{4...7}) > 9$ или $((C)=1)$, то $(A_{4...7}) \leftarrow (A_{4...7}) + 6$
Вычитание из аккумулятора регистра и заёма	SUBB A, Rn	10011rrr	1	1	1	$(A) \leftarrow (A) - (C) - (Rn)$
Вычитание из аккумулятора прямоадресуемого байта и заема	SUBB A, ad	10010101	3	2	1	$(A) \leftarrow (A) - (C) - ((ad))$
Вычитание из аккумулятора байта РПД и заема	SUBB A, @Ri	1001011i	1	1	1	$(A) \leftarrow (A) - (C) - ((Ri))$
Вычитание из аккумулятора константы и заема	SUBB A, d	10010100	2	2	1	$(A) \leftarrow (A) - (C) - \#d$
Инкремент аккумулятора	INC A	00000100	1	1	1	$(A) \leftarrow (A) + 1$
Инкремент регистра	INC Rn	00001rrr	1	1	1	$(Rn) \leftarrow (Rn) + 1$
Инкремент прямоадресуемого байта	INC ad	00000101	3	2	1	$(ad) \leftarrow (ad) + 1$
Инкремент байта в РПД	INC @Ri	0000011i	1	1	1	$((Ri)) \leftarrow ((Ri)) + 1$
Инкремент указателя данных	INC DPTR	10100011	1	1	2	$(DPTR) \leftarrow (DPTR) + 1$
Декремент аккумулятора	DEC A	00010100	1	1	1	$(A) \leftarrow (A) - 1$
Декремент регистра	DEC Rn	00011rrr	1	1	1	$(Rn) \leftarrow (Rn) - 1$
Декремент прямоадресуемого байта	DEC ad	00010101	3	2	1	$(ad) \leftarrow (ad) - 1$
Декремент байта в РПД	DEC @Ri	0001011i	1	1	1	$((Ri)) \leftarrow ((Ri)) - 1$
Умножение аккумулятора на регистр B	MUL AB	10100100	1	1	4	$(B)(A) \leftarrow (A)*(B)$
Деление аккумулятора на регистр B	DIV AB	10000100	1	1	4	$(B).(A) \leftarrow (A)/(B)$

Приложение 3. Система команд MCS51
Логические операции

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Логическое И аккумулятора и регистра	ANL A, Rn	01011rrr	1	1	1	$(A) \leftarrow (A) \text{ AND } (Rn)$
Логическое И аккумулятора и прямоадресуемого байта	ANL A, ad	01010101	3	2	1	$(A) \leftarrow (A) \text{ AND } (ad)$
Логическое И аккумулятора и байта из РПД	ANL A, @Ri	0101011i	1	1	1	$(A) \leftarrow (A) \text{ AND } ((Ri))$
Логическое И аккумулятора и константы	ANL A, #d	01010100	2	2	1	$(A) \leftarrow (A) \text{ AND } \#d$
Логическое И прямоадресуемого байта и аккумулятора	ANL ad, A	01010010	3	2	1	$(ad) \leftarrow (ad) \text{ AND } (A)$
Логическое И прямоадресуемого байта и константы	ANL ad, #d	01010011	7	3	2	$(ad) \leftarrow (ad) \text{ AND } \#d$
Логическое ИЛИ аккумулятора и регистра	ORL A, Rn	01001rrr	1	1	1	$(A) \leftarrow (A) \text{ OR } (Rn)$
Логическое ИЛИ аккумулятора и прямоадресуемого байта	ORL A, ad	01000101	3	2	1	$(A) \leftarrow (A) \text{ OR } (ad)$
Логическое ИЛИ аккумулятора и байта из РПД	ORL A, @Ri	0100011i	1	1	1	$(A) \leftarrow (A) \text{ OR } ((Ri))$
Логическое ИЛИ аккумулятора и константы	ORL A, #d	01000100	2	2	1	$(A) \leftarrow (A) \text{ OR } \#d$
Логическое ИЛИ прямоадресуемого байта и аккумулятора	ORL ad, A	01000010	3	2	1	$(ad) \leftarrow (ad) \text{ OR } (A)$
Логическое ИЛИ прямоадресуемого байта и константы	ORL ad, #d	01000011	7	3	2	$(ad) \leftarrow (ad) \text{ OR } \#d$
Исключающее ИЛИ аккумулятора и регистра	XRL A, Rn	01101rrr	1	1	1	$(A) \leftarrow (A) \text{ XOR } (Rn)$
Исключающее ИЛИ аккумулятора и прямоадресуемого байта	XRL A, ad	01100101	3	2	1	$(A) \leftarrow (A) \text{ XOR } (ad)$
Исключающее ИЛИ аккумулятора и байта из РПД	XRL A, @Ri	0110011i	1	1	1	$(A) \leftarrow (A) \text{ XOR } ((Ri))$
Исключающее ИЛИ аккумулятора и константы	XRL A, #d	01100100	2	2	1	$(A) \leftarrow (A) \text{ XOR } \#d$
Исключающее ИЛИ прямоадресуемого байта и аккумулятора	XRL ad, A	01100010	3	2	1	$(ad) \leftarrow (ad) \text{ XOR } (A)$
Исключающее ИЛИ прямоадресуемого байта и константы	XRL ad, #d	01100011	7	3	2	$(ad) \leftarrow (ad) \text{ XOR } \#d$
Сброс аккумулятора	CLR A	11100100	1	1	1	$(A) \leftarrow 0$
Инверсия аккумулятора	CPL A	11110100	1	1	1	$(A) \leftarrow \text{NOT}(A)$
Сдвиг аккумулятора влево циклический	RL A	00100011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0\div 6, (A_0) \leftarrow (A_7)$
Сдвиг аккумулятора влево через перенос	RLC A	00110011	1	1	1	$(A_{n+1}) \leftarrow (A_n), n=0\div 6$ $(A_0) \leftarrow (C), (C) \leftarrow (A_7)$
Сдвиг аккумулятора вправо циклический	RR A	00000011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0\div 6, (A_7) \leftarrow (A_0)$
Сдвиг аккумулятора вправо через перенос	RRC A	00010011	1	1	1	$(A_n) \leftarrow (A_{n+1}), n=0\div 6$ $(A_7) \leftarrow (C), (C) \leftarrow (A_0)$
Обмен местами тетрад в аккумуляторе	SWAP A	11000100	1	1	1	$(A_{0...3}) \leftrightarrow (A_{4...7})$

Приложение 4. Система команд MCS51
Команды передачи управления

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Длинный переход в полном объеме ПП	LJMP ad16	00000010	12	3	2	$(PC) \leftarrow ad16$
Абсолютный переход внутри страницы в 2 Кб	AJMP ad11	$a_{10}a_9a_800001$	6	2	2	$(PC) \leftarrow (PC) + 2, (PC_{0-10}) \leftarrow ad11$
Короткий относительный переход внутри страницы в 256 байт	SJMP rel	10000000	5	2	2	$(PC) \leftarrow (PC) + 2, (PC) \leftarrow (PC) + rel$
Косвенный относительный переход	JMP @A+DPTR	01110011	1	1	2	$(PC) \leftarrow (A) + (DPTR)$
Переход, если аккумулятор равен нулю	JZ rel	01100000	5	2	2	$(PC) \leftarrow (PC) + 2$, если $(A)=0$, то $(PC) \leftarrow (PC) + rel$
Переход, если аккумулятор не равен нулю	JNZ rel	01110000	5	2	2	$(PC) \leftarrow (PC) + 2$, если $(A) \neq 0$, то $(PC) \leftarrow (PC) + rel$
Переход, если перенос равен единице	JC rel	01000000	5	2	2	$(PC) \leftarrow (PC) + 2$, если $(C)=1$, то $(PC) \leftarrow (PC) + rel$
Переход, если перенос равен нулю	JNC rel	01010000	5	2	2	$(PC) \leftarrow (PC) + 2$, если $(C)=0$, то $(PC) \leftarrow (PC) + rel$
Переход, если бит равен единице	JB bit, rel	00100000	11	3	2	$(PC) \leftarrow (PC) + 3$, если $(b)=1$, то $(PC) \leftarrow (PC) + rel$
Переход, если бит равен нулю	JNB bit, rel	00110000	11	3	2	$(PC) \leftarrow (PC) + 3$, если $(b)=0$, то $(PC) \leftarrow (PC) + rel$
Переход, если бит установлен, с последующим сбросом бита	JBC bit, rel	00010000	11	3	2	$(PC) \leftarrow (PC) + 3$, если $(b)=1$, то $(b) \leftarrow 0$ и $(PC) \leftarrow (PC) + rel$
Декремент регистра и переход, если не нуль	DJNZ Rn, rel	11011rrr	5	2	2	$(PC) \leftarrow (PC) + 2, (Rn) \leftarrow (Rn) - 1$, если $(Rn) \neq 0$, то $(PC) \leftarrow (PC) + rel$
Декремент прямоадресуемого байта и переход, если не нуль	DJNZ ad, rel	11010101	8	3	2	$(PC) \leftarrow (PC) + 2, (ad) \leftarrow (ad) - 1$, если $(ad) \neq 0$, то $(PC) \leftarrow (PC) + rel$
Сравнение аккумулятора с прямоадресуемым байтом и переход, если не равно	CJNE A, ad, rel	10110101	8	3	2	$(PC) \leftarrow (PC) + 3$, если $(A) \neq (ad)$, то $(PC) \leftarrow (PC) + rel$, если $(A) < (ad)$, то $(C) \leftarrow 1$, иначе $(C) \leftarrow 0$
Сравнение аккумулятора с константой и переход, если не равно	CJNE A, #d, rel	10110100	10	3	2	$(PC) \leftarrow (PC) + 3$, если $(A) \neq \#d$, то $(PC) \leftarrow (PC) + rel$, если $(A) < \#d$, то $(C) \leftarrow 1$, иначе $(C) \leftarrow 0$
Сравнение регистра с константой и переход, если не равно	CJNE Rn, #d, rel	10111rrr	10	3	2	$(PC) \leftarrow (PC) + 3$, если $(Rn) \neq \#d$, то $(PC) \leftarrow (PC) + rel$, если $(Rn) < \#d$, то $(C) \leftarrow 1$, иначе $(C) \leftarrow 0$
Сравнение байта в РГД с константой и переход, если не равно	CJNE @Ri, d, rel	1011011i	10	3	2	$(PC) \leftarrow (PC) + 3$, если $((Ri)) \neq \#d$, то $(PC) \leftarrow (PC) + rel$, если $((Ri)) < \#d$, то $(C) \leftarrow 1$, иначе $(C) \leftarrow 0$

Продолжение приложения 4

Название команды	Мнемокод	КОП	Т	Б	Ц	Операция
Длинный вызов подпрограммы	LCALL ad16	00010010	12	3	2	$(PC) \leftarrow (PC) + 3, (SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{0..7}), (SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{8..15}), (PC) \leftarrow ad16$
Абсолютный вызов подпрограммы в пределах страницы в 2 Кб	ACALL ad11	$a_{10}a_9a_810001$	6	2	2	$(PC) \leftarrow (PC) + 2, (SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{0..7}), (SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{8..15}), (PC_{0..10}) \leftarrow ad11$
Возврат из подпрограммы	RET	00100010	1	1	2	$(PC_{8..15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1,$ $(PC_{0..7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
Возврат из подпрограммы обработки прерывания	RETI	00110010	1	1	2	$(PC_{8..15}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1,$ $(PC_{0..7}) \leftarrow ((SP)), (SP) \leftarrow (SP) - 1$
Пустая операция	NOP	00000000	1	1	1	$(PC) \leftarrow (PC) + 1$

Примечание. Ассемблер допускает использование обобщенного имени команд *JMP* и *CALL*, которые в процессе трансляции заменяются оптимальными по формату командами перехода (*AJMP*, *SJMP*, *LJMP*) или вызова (*ACALL*, *LCALL*).

Приложение 5. Система команд MCS51
Операции с битами

Название команды	Мnemonic код	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) ← NOT(C)
Инверсия бита	CPL bit	10110010	4	2	1	(b) ← NOT(b)
Логическое И бита и переноса	ANL C, bit	10000010	4	2	2	(C) ← (C) AND (b)
Логическое И инверсии бита и переноса	ANL C, /bit	10110000	4	2	2	(C) ← (C) AND (NOT(b))
Логическое ИЛИ бита и переноса	ORL C, bit	01110010	4	2	2	(C) ← (C) OR (b)
Логическое ИЛИ инверсии бита и переноса	ORL C, /bit	10100000	4	2	2	(C) ← (C) OR (NOT(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)

Приложение 6. Микроконтроллеры семейства MCS51 и его аналоги

Type	Manufacturer	Max. Clock Rate / clock per cycle	RPM (Byte)	RDM / EDM (Byte)	Max. I/O Lines	Timer (16-Bit)	Serial I/O	ADC-In/Res	Others
AT-89C51	Atmel	24	4k Flash	128	32	2	1	-	-
AT-89C52	Atmel	24	8k Flash	256	32	3	1	-	-
AT-89C1051	Atmel	24	1k Flash	64	15	2	1	-	Analog Comp.
AT-89C2051	Atmel	24	2k Flash	128	15	2	1	-	Analog Comp.
AT-894051	Atmel	24	4k Flash	128	15	2	1	-	Analog Comp.
AT-89C5132	Atmel	40 / 6	64k Flash	256 / 2.5 k	44	3	1	2 / 10	USB, SPI, I2C
ADuC812	Analog	12	8k Flash	256	32	3	1	8 / 12	2 12bit DAC, SPI, I2C, WDT
ADuC841	Analog	20	64k Flash	256 / 2304	32	3	1	8 / 12	2 12bit DAC, SPI, I2C, WDT
ADuC841	Analog	0.2 – 16	64k Flash	256 / 2304	32	3	1	8 / 12	2 12bit DAC, SPI, I2C, WDT, on-chip PLL
C8051F042	Cygnal	25	64k Flash	256 / 4352	64	5	2	8 / 10	2 12bit DAC, PCA, CAN, SPI, I2C, WDT
C8051F0122	Cygnal	100	128k Flash	256 / 8.25k	64	5	2	8 / 10	2 12bit DAC, PCA, SPI, I2C, WDT
DS-80C310	Dallas	33	-	256	32	3	1	-	2 Data Pointer
DS-80C320	Dallas	33	-	256	32	3	2	-	Low Power, WDT
DS-80C323	Dallas	18	-	256	32	3	2	-	Low Power, WDT
DS-87C520	Dallas	33	16k	256 / 1k	32	3	2	-	WDT
DS-87C530	Dallas	33	16k	256 / 1k	32	3	2	-	RTC, WDT
DS-89C420	Dallas	50 / 1	16k	256 / 1k	32	3	2	-	WDT
I-8XC51	Intel	24	4k	128	32	2	1	-	-
I-8XC51FX	Intel	20	8k-32k	256	32	2	1	-	PCA
SC-8XC51	Philips	33	4k	128	32	2	1	-	-
P-8XC52	Philips	33	8k	256	32	3	1	-	-
SC-8XC451	Philips	16	4k	128	56	2	1	-	Parallel Interface
P-8XC550	Philips	16	4k	128	24 +8in	2	1	8 / 8	WDT

PCB-8XC552	Philips	24	8k	256	40 +8in	2	2	8 / 10	I2C, WDT, PWM
PCB-8XC562	Philips	16	8k	256	40 +8in	3	1	8 / 8	WDT, PWM, CAPCOM
P-8XC575	Philips	16	8k	256	32	3	1	-	WDT, PCA, 4Analog Comp.
P-8XC576	Philips	16	8k	256	30	3	1	6 / 10	WDT, PWM, 4Analog Comp.
SAB-80C5X5	Siemens	20	8k	256	40 +8in	3	1	8 / 8	WDT, PWM, WDU
SAB-80C515A	Siemens	18	-	256 / 1k	40 +8in	3	1	8 / 10	WDT, PWM, WDU
SAB-80C5X7	Siemens	16	8k	256	56 +12in	4	2	12 / 8	WDT, PWM, WDU
SAB-80C517A	Siemens	18	-	256 / 1k	56 +12in	4	2	12 / 10	WDT, WDU, PWM
SAB-C501	Siemens	40	8k	256	32	3	1	-	-
SAB-C502	Siemens	20	16k	256 / 256	32	3	1	-	progr. XRAM- Start Adr, WDU
SAB-C504	Siemens	40	16k	256 / 256	32	3	1	8 / 10	PWM for motor, WDU
SAB-C509	Siemens	16	-	256 / 3k	64 +15in	5	2	15 / 10	WDT, PWM, WDU
SAB-C511	Siemens	12	2,5k	128	32	2	1 SSC	-	-
SAB-C511A	Siemens	12	4k	256	32	2	1 SSC	-	-
SAB-C513	Siemens	12	8k	256	32	3	2	-	SSC
SAB-C513A	Siemens	12	16k	256 / 256	32	3	2	-	SSC, EEProm available
SAB-C515C	Siemens	10	ROMless / 64k	256 / 1k	49 +8in	3	2	8 / 10	Full CAN Interface, PWM, SSC