

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Локтионова Оксана Геннадьевна
Должность: проректор по учебной работе
Дата подписания: 02.02.2021 05:13:54
Уникальный программный ключ:
0b817ca911e6668abb13a5d426d39e5f1c11eabbf73e943df4a4851fda56d089

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра вычислительной техники

УТВЕРЖДАЮ
Проректор по учебной работе
О.Г. Локтионова
«15» 12 2017



ПРОГРАММИРОВАНИЕ МИКРОЭВМ К1816ВЕ35 (i8035)

Методические указания к выполнению лабораторных работ для студентов
направления подготовки 09.03.01 по дисциплине
«Микропроцессорные системы для автоматизации технологических
процессов»

Курск 2017

УДК 681.3.01

Составитель: В.С. Панищев

Рецензент
Кандидат технических наук *Халин Ю.А.*

Программирование микроЭВМ K1816BE35 (i8035): методические указания к выполнению лабораторных работ / Юго-Западный гос. ун-т; сост.: В.С. Панищев. – Курск, 2017. – 39 с.

Излагаются методические указания к выполнению лабораторных работ по дисциплине «Микропроцессорные системы для автоматизации технологических процессов», посвященные основам программирования процессора микроЭВМ K1816BE35 (i8035), K1816BE31 (i8051).

Методические указания соответствуют Федеральному государственному образовательному стандарту высшего образования направления подготовки 09.03.01 Информатика и вычислительная техника, учебному плану направления подготовки 09.03.01 Информатика и вычислительная техника, одобренному Ученым советом университета (протокол № 7 «29» февраля 2016 г.).

Предназначены для студентов направления подготовки 09.03.01 очной и заочной формы обучения

Текст печатается в авторской редакции.

Подписано в печать *15.12.17*. Формат 60*84 1/16.
Усл. печ.л. *1,4*. Уч.-изд.л. *1,6* Тираж 100 экз. Заказ *4788* Бесплатно.
Юго-Западный государственный университет
305040 Курск, ул. 50 лет Октября, 94.

Оглавление

1. Программирование процедур арифметической и логической обработки. Программирование ввода/вывода..... 4

1.1.	Цель работы	4
1.2.	Подготовка к работе	4
1.3.	Краткое описание системы команд.....	4
1.4.	Описание портов ввода/вывод.....	8
1.5.	Архитектура ОМЭВМ ВЕ51(31)	9
1.6.	Краткое описание системы команд ВЕ31(51).....	18
1.7.	Указания к выполнению предварительного домашнего задания....	22
1.8.	Порядок выполнения работы.....	23
1.9.	Содержание отчета.....	23
1.10.	Вопросы для самопроверки	24

2. ПРОГРАММИРОВАНИЕ ТАЙМЕРОВ И ОБРАБОТКА ПЕРЕРЫВАНИЙ 25

2.1.	Цель работы	25
2.2.	Подготовка к работе	25
2.3.	Описание таймеров/счетчиков	25
2.4.	Подсистема прерываний	27
2.5.	Указания к выполнению предварительного домашнего задания....	30
2.6.	Порядок выполнения работы.....	30
2.7.	Содержание отчета.....	31
2.8.	Вопросы для самопроверки	31

1. Программирование процедур арифметической и логической обработки. Программирование ввода/вывода

1.1. Цель работы

Изучение команд пересылки и обработки информации, команд ввода/вывода. Приобретение навыков написания и отладки программ арифметической (логической) обработки и процедур ввода/вывода.

1.2. Подготовка к работе

В процессе подготовки к лабораторной работе необходимо изучить:

- систему команд однокристалльной микроЭВМ K1816BE35, обращая особое внимание на команды пересылки и загрузки, арифметико-логические и ввода/вывода;
- внутреннюю структуру, состав внешних выводов микросхемы K1816BE31, способы подключения к ней внешней памяти и устройств ввода/вывода, организацию портов;
- систему команд однокристалльной микроЭВМ K1816BE31, обращая особое внимание на команды пересылки и загрузки, арифметико-логические и ввода/вывода;
- рабочее задание и методические указания к его выполнению.

1.3. Краткое описание системы команд

Программно-доступными объектами в составе ОМЭВМ серии K1816 являются: регистр - аккумулятор А, таймер/счетчик Т, порты ввода/вывода P0..P2, P4..P7, регистр PSW, два банка по восемь регистров R0..R7, R0'..R7' и 64 ячейки внутренней памяти данных - с адресами 00H..3FH. Кроме того, по специальной команде MOVX может быть доступна внешняя память данных с адресами 00H..FFH.

При программировании следует иметь в виду, что в качестве регистров (банк 0) R0..R7 используются ячейки внутренней памяти данных с адресами 00H..07H соответственно, а в качестве R0'..R7' (банк 1) - ячейки 18H..20H. В ОМЭВМ серии K1816 предусмотрен 8-уровневый стек, в который автоматически (по команде CALL или в процедуре прерываний) загружается 12-разрядный программный счетчик PC и 4 флажка признаков. Таким образом, один уровень стека занимает два байта. Под стек используются регистры внутренней памяти данных с адресами 08H..17H, указатель стека - младшие три бита регистра PSW.

В системе команд ОМЭВМ серии K1816 используются прямая, косвенная и непосредственная адресация. По прямому адресу можно обращаться к аккумулятору А, таймеру/счетчику Т, регистру PSW, портам ввода/вывода P0..P2, P4..P7 и регистрам R0..R7, R0'..R7'. Примеры команд с прямой адресацией: ADD A,R3; MOV T,A; MOV A,PSW; IN A,P1;

Косвенно можно адресоваться к ячейкам памяти данных через регистры R0, R1 текущего банка, например:

MOV A,@R0 чтение в аккумулятор содержимого ячейки внутреннего ОЗУ данных, адрес которой хранится в R0;

INC @R1 добавление 1 к содержимому ячейки внутреннего ОЗУ данных, адрес которой хранится в R1;

MOVX A,@R0 чтение в аккумулятор содержимого ячейки внешнего ОЗУ данных, адрес которой хранится в R0;

MOVX @R0,A запись содержимого аккумулятора в ячейку внешнего ОЗУ данных, адрес которой хранится в R0.

Двухбайтовые команды с непосредственной адресацией позволяют загружать константу, содержащуюся во втором байте в регистры или косвенно адресуемые ячейки внутреннего ОЗУ или задавать второй операнд в двухместных арифметических и логических операциях, например:

MOV A, 5E; MOV R6, 40; MOV @R4, 77; ANL A, 08.

По типу операций можно выделить следующие классы команд: пересылки и загрузки, включая и команды обращения к портам ввода/вывода; арифметические и логические; передачи управления; специальные команды.

Класс команд пересылки и загрузки представлен главным образом разновидностями команд MOV, обеспечивающих пересылку информации между регистрами, ячейками внутренней памяти данных, аккумулятором, таймером, регистром PSW, а так же загрузку перечисленных объектов константами (см. примеры в п.2.3.2). Две команды позволяют считывать в аккумулятор A содержимое ячеек памяти программ:

MOV A,@A считывает с текущей страницы содержимое ячейки памяти программ, косвенно адресуемой через аккумулятор: A:=ЗУ Прг(PC(11:8).A(7:0));

MOVP3 A,@A считывает с третьей страницы содержимое ячейки памяти программ, косвенно адресуемой через аккумулятор: A:=ЗУ Прг(0011.A(7:0)).

Команды обмена XCH меняют местами содержимое аккумулятора и регистра или косвенно адресуемой ячейки памяти. Команды XCHD меняют местами только младшие тетрады аккумулятора и ячейки памяти, при этом старшие тетрады остаются неизменными.

Команда SWAP A меняет местами старшую и младшую тетрады в аккумуляторе.

Чтение в аккумулятор содержимого портов P0..P2 осуществляется по командам IN A,P.; запись содержимого аккумулятора в порт P0..P2 - командами OUTL P.,A.

Для работы с четырехразрядными портами P4..P7 используют команды MOVD A,P.; MOVD P.,A, причем в обмене участвует только младшая тетрада.

Арифметические и логические команды делятся на одноместные и двухместные. В двухместных командах первый операнд размещается в аккумуляторе, туда же помещается результат. Второй операнд может адресоваться разными способами, например:

ADD A,R7; ADDC A,@R1; ANL A,0F.

К двухместным операциям данного класса в ОМЭВМ К1816ВЕ35 относятся:

- ADD - сложение;
- ADDC - сложение с переносом;
- ANL - конъюнкция;
- ORL - дизъюнкция;
- XRL - сумма по модулю два (исключающее ИЛИ).

Следует обратить внимание, что в системе команд отсутствует команда вычитания!

Логические операции с непосредственной адресацией могут выполняться и над содержимым портов P0,P1,P2: ANL Pp, D : Pp := Pp & D; ORL Pp, D : Pp := Pp ∨ D; Pp = [0,1,2].

Для четырехразрядных портов P3..P7 возможно выполнение логических операций с младшей тетрадой аккумулятора A[3:0]: ANLD Pq,A : Pq := Pq & A[3:0]; ORLD Pq,A : Pq := Pq ∨ A[3:0]; Pq ∈ {4,5,6,7}.

Одноместные операции включают:

- DAA - десятичная коррекция аккумулятора;
- CLR A; CPL A - очистка аккумулятора; инверсия аккумулятора;
- INC A; INC R; INC @R - инкремент содержимого аккумулятора, регистра, ячейки памяти;
- DEC A; DEC R - декремент содержимого аккумулятора или регистра;
- RL A; RR A - циклический сдвиг аккумулятора влево и вправо;
- RLC A; RRC A - циклический сдвиг аккумулятора влево и вправо через триггер флага переноса.

Действия команд сдвигов описываются следующими выражениями:

RL A: A[7:0] := A[6:0].A[7]; FC := A[7]

RR A: A[7:0] := A[0].A[7:1]; FC := A[0]

RLC A: FC.A[7:0] := A[7].A[6:0].FC;

RLC A: FC.A[7:0] := A[0].FC.A[7:1].

Группа команд передачи управления включает команды безусловного и условных переходов, команду безусловного вызова подпрограммы и две команды безусловного возврата.

Все команды переходов являются двухбайтовыми, причем только команда безусловного перехода JMP позволяет осуществить т.н. "длинный" переход - в пределах всей памяти программ. Остальные команды организуют переход в пределах текущей страницы (старшие 4 бита программного счетчика не меняются). Среди них:

JMPP	безусловный переход по косвенному адресу PC[7:0]
@A	:= A;
JC/JNC	условные переходы по переносу/отсутствию переноса;
JZ/JNZ	условные переходы по нулю/не нулю аккумулятора;
JT0/JNT	условные переходы по состоянию входа T0;
0	
JT0/JNT	условные переходы по состоянию входа T1;
1	
JF0,JF1	условные переходы по единичному значению флагов пользователя;
JTF	условный переход по переполнению таймера;
JNI	условный переход, если сигнал на входе запроса на внешнее прерывание INT = 0;
JBb	условный переход по единичному значению заданного бита аккумулятора.

К командам передачи управления может быть отнесена команда организации цикла DJNZ Ri,s, где Ri - номер регистра (R0..R7), s - смещение на странице. Команда декрементирует содержимое заданного регистра и, если результат не равен нулю, осуществляется переход.

Команда безусловного вызова подпрограммы CALL позволяет вызвать подпрограмму из любой области памяти программ - она, подобно JMP, загружает все 12 разрядов PC. Предварительно старое значение PC вместе с PSW(7:4) (флаги) загружается в стек.

Во внутреннем языке K1816BE35 существуют две команды безусловного возврата: RET - восстанавливает PC, не меняя PSW; RETR - восстанавливает PC и PSW[7:4].

Специальные команды можно, в свою очередь, разбить на несколько групп.

Команды управления таймером/счетчиком:

- * STRT T - запуск таймера;
- * STRT CNT - запуск счетчика внешних событий;

- * STOP TCNT- останов таймера/счетчика.

Команды управления подсистемой прерываний:

- * EN TSNTI - разрешить прерывание от таймера/счетчика;
- * DIS TSNTI - запретить прерывание от таймера/счетчика;
- * EN I - разрешить внешнее прерывание;
- * DIS I - запретить внешнее прерывание.

Команды выбора блоков памяти:

- * SEL RB0 - выбор банка регистров 0;
- * SEL RB1 - выбор банка регистров 1;
- * SEL RB0 - выбор блока 0 памяти программ;
- * SEL RB1 - выбор блока 1 памяти программ.

Команды управления флагами:

- * CLR C; CLR F0; CLR F1 - очистка признаков переноса и флажков пользователя;

- * CPL C; CPL F0; CPL F1 - инверсия признаков переноса и флажков пользователя.

Прочие команды:

- * ENT0 CLK - разрешение выдачи сигнала синхронизации на вывод T0;

- * NOP - пустая операция.

1.4. Описание портов ввода/вывод

Рассмотрим работу *квазидвунаправленной линии порта*.

Линия постоянно подключена к источнику питания +5В через резистор большого сопротивления (50 кОм), обеспечивая для единичной стандартной ТТЛ-нагрузки достаточный входной ток, соответствующий логической "1". В то же время потенциал в линии может быть приближен к потенциалу "земли" (уровень логического "0") нулевым сигналом стандартной ТТЛ-схемы.

При передаче на выход порта логической "1" по сигналу "Запись" для ускорения переходного процесса от 0 к 1 на короткое время (0,5 мкс) открывается верхний МОП-транзистор с относительно низким сопротивлением (5 кОм).

При передаче на выход порта логического "0" открытым оказывается нижний МОП-транзистор с сопротивлением 3 кОм, что обеспечивает отвод тока из входной цепи ТТЛ-нагрузки.

Для того, чтобы настроить некоторую линию на режим ввода в ОМЭВМ, необходимо перед этим записать в буферный D-триггер "1". При этом нижний МОП-транзистор выходной схемы окажется закрытым и обеспечит ТТЛ-источнику вводимого сигнала высокоимпедансную нагрузку. Сигнал системного сброса RESET записывает во все триггера портов P1 и P2 лог."1".

Таким образом, в процессе ввода информации с портов P1, P2 выполняется операция конъюнкции над вводимыми данными и текущим значением регистра порта. Квазидвунаправленная структура не позволяет выполнить операцию чтения состояния портов P1, P2, зафиксированных по последней команде вывода.

Порт P2 отличается от порта P1 тем, что его младшие 4 бита могут быть использованы для расширения системы ввода/вывода. Через младшую тетраду порта P2 по специальным командам обращения к портам P4..P7 возможен доступ к четырем внешним четырехбитовым портам ввода/вывода, причем адрес порта 4..7 передается по линиям P25,P26:

- 0 0 - порт P4;
- 0 1 - порт P5;
- 1 0 - порт P6;
- 1 1 - порт P7.

Порт P0 (BUS,DB) представляет собой двунаправленный буфер с тремя состояниями и предназначен для побайтового ввода, вывода или ввода/вывода информации, причем двунаправленные передачи обеспечиваются командами MOVX, а однонаправленные - командами IN и OUTL.

1.5. Архитектура ОМЭВМ ВЕ51(31)

Восьмиразрядные высокопроизводительные однокристалльные микроЭВМ (ОМЭВМ) семейства МК51 выполнены по высококачественной n-МОП технологии (серия 1816) и КМОП технологии (серия 1830).

Структурная схема микроЭВМ К1816ВЕ51 показана на рис.1.1.

МикроЭВМ ВЕ51 включает в себя все узлы, необходимые для автономной работы:

- 4 Кбайт резидентного ППЗУ или СППЗУ (для ВЕ31 - отсутствует), расширяемое до 64Кбайт;
- 128 байт резидентного ОЗУ, которое включает 4 банка по 8 общих регистров и стек, используемый при вызове подпрограмм и при прерываниях;
- до 64 Кбайт внешнего ОЗУ;
- шестнадцатиразрядный программный счетчик (счетчик адресов), позволяющий адресовать 64 Кбайт памяти;
- восьмиразрядный указатель стека, который может быть установлен любым адресом резидентной (внутренней) памяти;
- два программируемых шестнадцатиразрядных таймера;
- программируемый асинхронный приемопередатчик (УАПП);

- четыре двунаправленных восьмиразрядных порта параллельного ввода/вывода;
- арифметические и логические узлы, которые реализуют арифметические функции: сложение, вычитание, умножение и деление; и логические функции: логическое умножение, логическое сложение, исключающее или, дополнение.

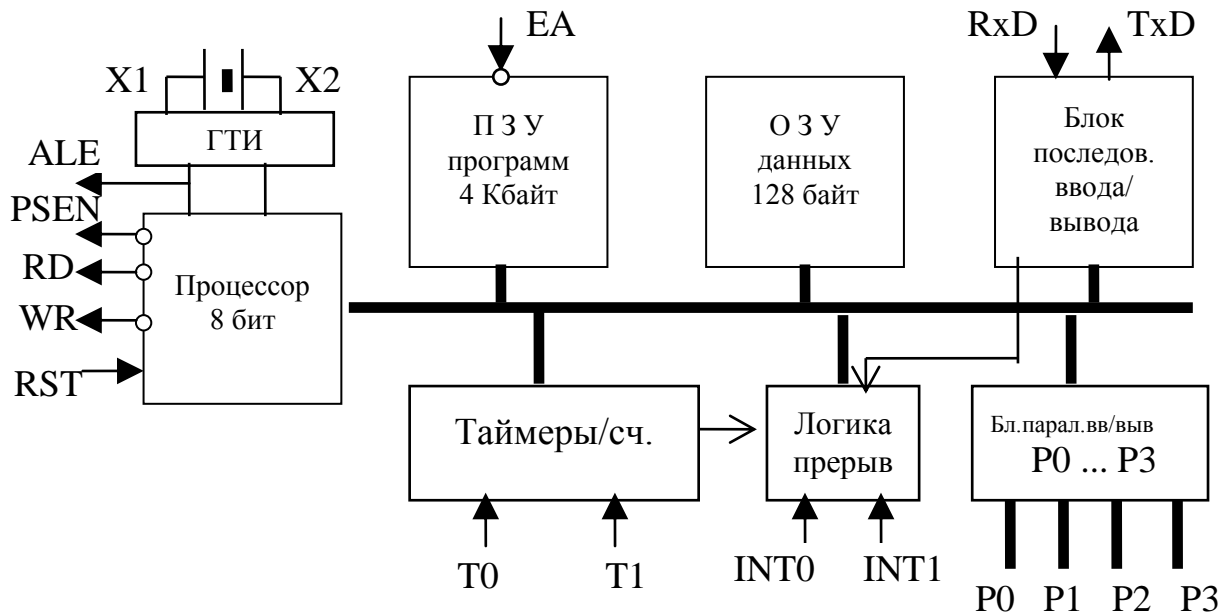
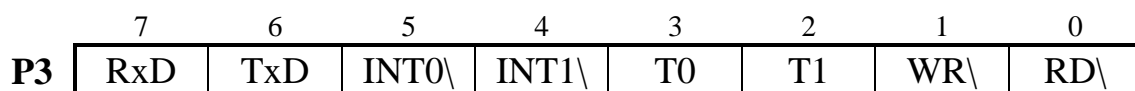


Рис. 1.1. Структура микроЭВМ K1816BE51

В BE51 используются следующие сигналы:

- ALE - строб адреса;
PSEN\ - строб чтения внешней памяти программ;
RD\, WR\ - стробы чтения и записи внешней памяти данных;
EA\ - разрешение обращения во внутреннюю память программ;
T0, T1 - входы счетчиков внешних событий;
INT0, INT1 - запросы внешних радиальных прерываний;
RxD - вход данных последовательного канала;
TxD - выход данных последовательного канала.

В состав BE51 входят параллельные 8-разрядные квазидвунаправленные порты P0..P3. Порт P3 может быть использован для ввода и вывода управляющих сигналов:



Таким образом, в циклах обращения к внешним ресурсам на линиях порта P3 работают управляющие сигналы, а при отсутствии

необходимости управления внешними ресурсами эти же линии могут быть использованы как линии порта.

Для подключения внешних ресурсов можно организовать внешние магистрали микроЭВМ.

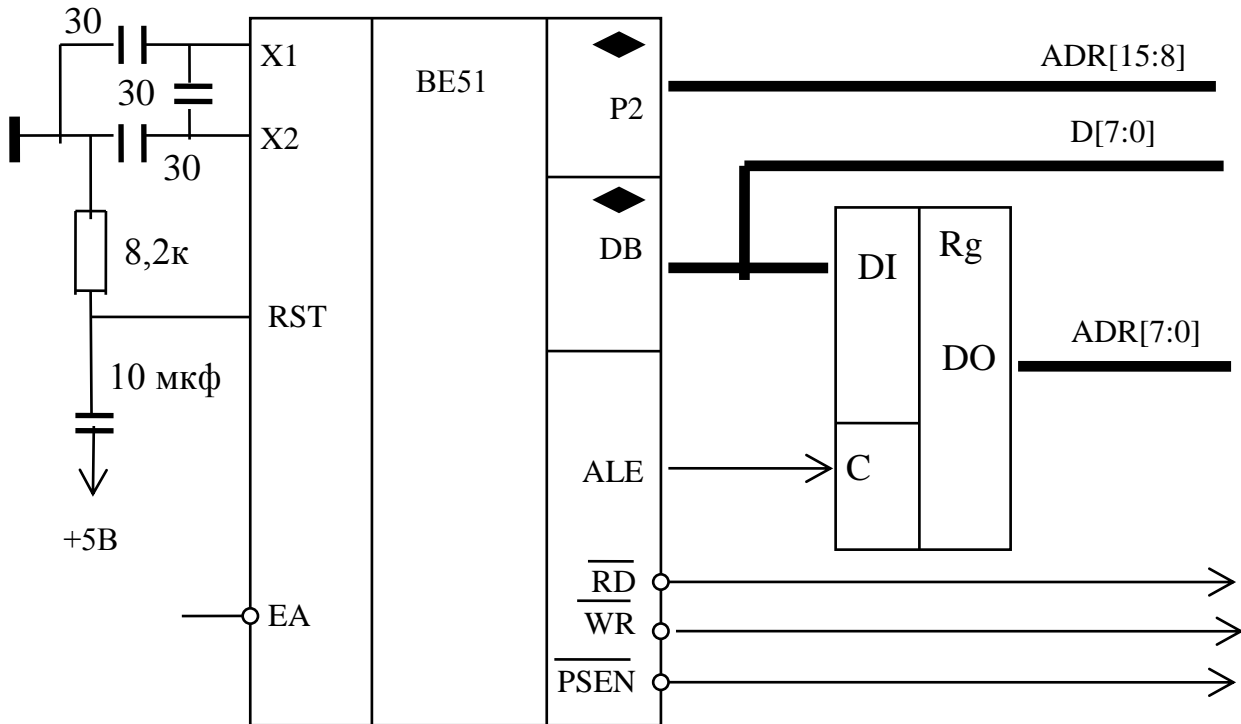


Рис. 1.2. Организация магистрали для подключения внешних ресурсов

МикроЭВМ BE51 позволяет управлять пятью (частично пересекающимися) адресными пространствами памяти, четыре из которых являются областями данных: RSEG - пространство регистров (4×8 байт); DSEG - пространство внутренней памяти данных (256 байт); BSEG - битовое пространство данных (256 бит); XSEG - пространство внешней памяти данных (до 64 Кбайт); CSEG - пространство программного кода (до 64 Кбайт). Пространства RSEG и BSEG частично пересекаются, физически совмещаются с DSEG и образуют единую внутреннюю среду для хранения данных. Это позволяет одни и те же данные рассматривать с разных позиций (ячейка памяти, регистр, битовое поле, порт ввода/вывода и т.п.) и организовывать наиболее удобный для данного случая доступ к ним.

Все порты ввода/вывода, системные регистры, таймеры так же отображены на пространство DSEG.

Пространство регистров представлено четырьмя банками регистров по 8 РОИ в каждом, а так же 16-разрядным программным счетчиком PC и регистром косвенного адреса DPTR, 8-разрядными аккумуляторами A и B, указателем стека SP и регистром PSW (рис. 1.3).

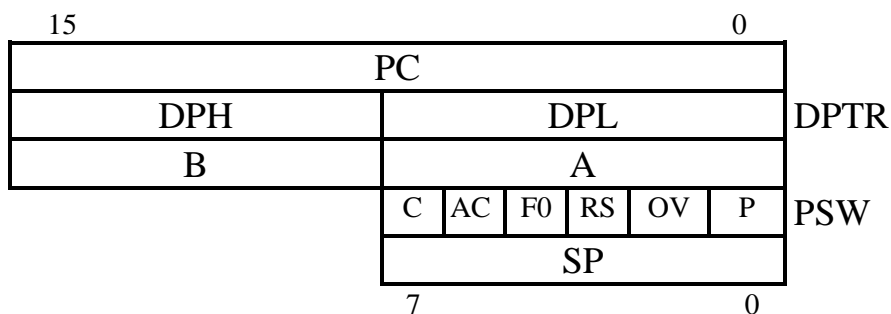


Рис. 1.3. Основные регистры микроЭВМ

Все регистры, показанные на рис.1.3, кроме PC, отображаются на DSEG.

Биты PSW [7:5] имеют следующее значение:

PSW[7]	CY	перенос из старшего (7) разряда АЛБ;
PSW[6]	AC	перенос из третьего разряда АЛБ;
PSW[5]	F0	флаг пользователя;
PSW[4:3]	RS	номер банка регистров;
PSW[2]	OV	флаг арифметического переполнения;
PSW[0])	P	флаг четности.

Флаги CY, AC и OV отражают признаки результата последней арифметической операции, а P - четность содержимого аккумулятора A. Расширение аккумулятора B используется в командах умножения и деления, а в остальных операциях - как обычная ячейка памяти.

Программный счетчик адресует пространство памяти программ CSEG объемом до 64 Кбайт, причем переход из области внутренней памяти программ к внешней осуществляется автоматически.

Указатель данных DPTR используется для обращения к XSEG и при пересылке констант из CSEG в A. Кроме того, содержимое DPTR используется а качестве смещения в команде перехода (см. раздел 1.6).

Указатель стека SP образует системный стек глубиной до 256 байт. SP хранит адрес последнего занесенного байта и растет при записи в сторону больших адресов.

Регистры R0, R1 каждого банка используются в качестве указателей данных. При сбросе машины в A, B, PC и DPTR загружаются 0h, а в SP - 07h.

Организация внутренней памяти данных

Пространство DSEG включает в себя 256 ячеек памяти, часть из которых является одновременно элементами других пространств. Первые 32 байта ОЗУ занимают 4 банка РОНов. Служебные регистры, порты ввода/вывода, таймеры, аккумуляторы и др. так же совмещены с ячейками памяти и полями битового сегмента. Это дает возможность обращения к одному физическому объекту разными способами. Так, к ячейке DSEG[E0]

можно обратиться по прямому адресу, обратиться как к аккумулятору А и как к полю BSEG[E0..E7] (к каждому биту в отдельности). Битовое пространство BSEG располагается в ячейках DSEG как это показано на рис. 1.4. Размещение в DSEG ПОН и служебных регистров показано на рис. 1.5.

DSEG	20	21	22	23	24	25	26	27
BSEG	00..07	08..0F	10..17	18..1F	20..27	28..2F	30..37	38..3F
DSEG	28	29	2A	2B	2C	2D	2E	2F
BSEG	40..47	48..4F	50..57	58..5F	60..67	68..6F	70..77	78..7F
DSEG	80	88	90	98	A0	A8	B0	B8
BSEG	80..87	88..8F	90..97	98..9F	A0..A7	A8..AF	B0..B7	B8..BF
DSEG	C0	C8	D0	D8	E0	E8	F0	F8
BSEG	C0..C7	C8..CF	D0..D7	D8..DF	E0..E7	E8..EF	F0..F7	F8..FF

Рис. 1.4. Размещение битового пространства в DSEG

Память программ адресуется PC[15:0] и может составлять до 64 Кбайт, причем младшие 4 Кбайт могут располагаться непосредственно на кристалле микроЭВМ (РПЗУ или ПЗУ), а остальная память - внешнее ЗУ. С точки зрения программиста внешняя и внутренняя память программ представляют единое адресное пространство.

a)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	R0	R1	R2	R3	R4	R5	R6	R7	R0	R1	R2	R3	R4	R5	R6	R7
1	R0	R1	R2	R3	R4	R5	R6	R7	R0	R1	R2	R3	R4	R5	R6	R7
2																
3																
4																
5																
6																
7																

b)

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
8																
9																
A																
B																
C																
D																
E																
F																

■ - ячейки, занятые битовым пространством BSEG

	0	1	2	3..	..7	8	9	A	B	C	D..
8	P0	SP	DPT L	DPT H	P CON	T CON	T MOD	T0 L	T0 H	T1 L	T1 H
9	P1					S CON	S BUF				
A	P2					IE					
B	P3					IP					
C											
D	PSW										
E	A										
F	B										

Рис. 1.5. Пространство внутренней памяти данных DSEG

Имеются "особые точки" CSEG:

- RESET - 0000h - стартовый адрес;

- EXTIO - 0003h - внешнее прерывание 0;
- TIMER0 - 000Bh - прерывание от таймера/счетчика T0;
- EXTI1 - 0013h - внешнее прерывание 1;
- TIMER1 - 001Bh - прерывание от таймера/счетчика T1;
- SINT - 0023h - прерывание последовательного порта;

Питание микроЭВМ BE51 осуществляется от одного источника +5В. Кварцевый резонатор 3,5..12 МГц подключается к выводам X1, X2 как показано на рис. 1.1; возможно подключение внешнего тактового генератора через X1.

OSC/2 - основная внутренняя тактовая частота CLK. Первая половина периода CLK - фаза P1, вторая - P2. Каждый машинный цикл состоит из шести периодов CLK: состояний S1, S2, .. S6 или двенадцати периодов OSC, называемых фазами S1P1, S1P2, S2P1, .. S6P1, S6P2. Каждый машинный цикл сопровождается генерацией двух стробов ALE длительностью в один период CLK - S1P2..S2P1, S4P2.. S5P1 (рис.1.6).

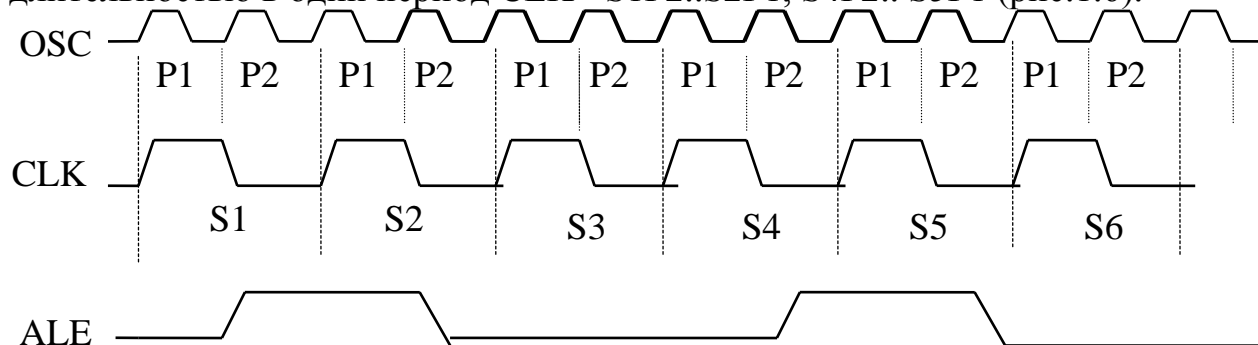


Рис. 1.6. Синхронизация микроЭВМ BE51

Командный цикл (КЦ) содержит один или несколько машинных циклов (МЦ) и отсчитывается от S1P1. По фазе S4P2 в IR фиксируется код операции, второй байт двухбайтовой команды читается в S1P2 следующего МЦ, третий байт трехбайтовой - в S4P2.

Таким образом, для ввода каждого байта команды требуется один полуцикл. Во время всех оставшихся полуциклов в фазах S1P2 и S4P2 читается код операции следующей размещенной в памяти команды, однако он не помещается в IR и не инкрементируется PC. Текущий КЦ всегда завершается в S6P2. Вслед за этим начинается S1, в котором в IR вводится новый байт.

Обращение к внешней памяти данных XSEG по команде MOVX требует второго МЦ, в котором не вырабатывается первый строб ALE. На рис. 1.7 приведена временная диаграмма обращения к XSEG.

МикроЭВМ BE51 содержит 4 параллельных двунаправленных порта ввода/вывода P0..P3 (рис. 1.8).

Поскольку порты ввода/вывода размещены в пространстве DSEG, любая команда с операндом из DSEG применима к содержимому P0-P3, а совмещение с BSEG позволяет иметь доступ к каждому биту портов.

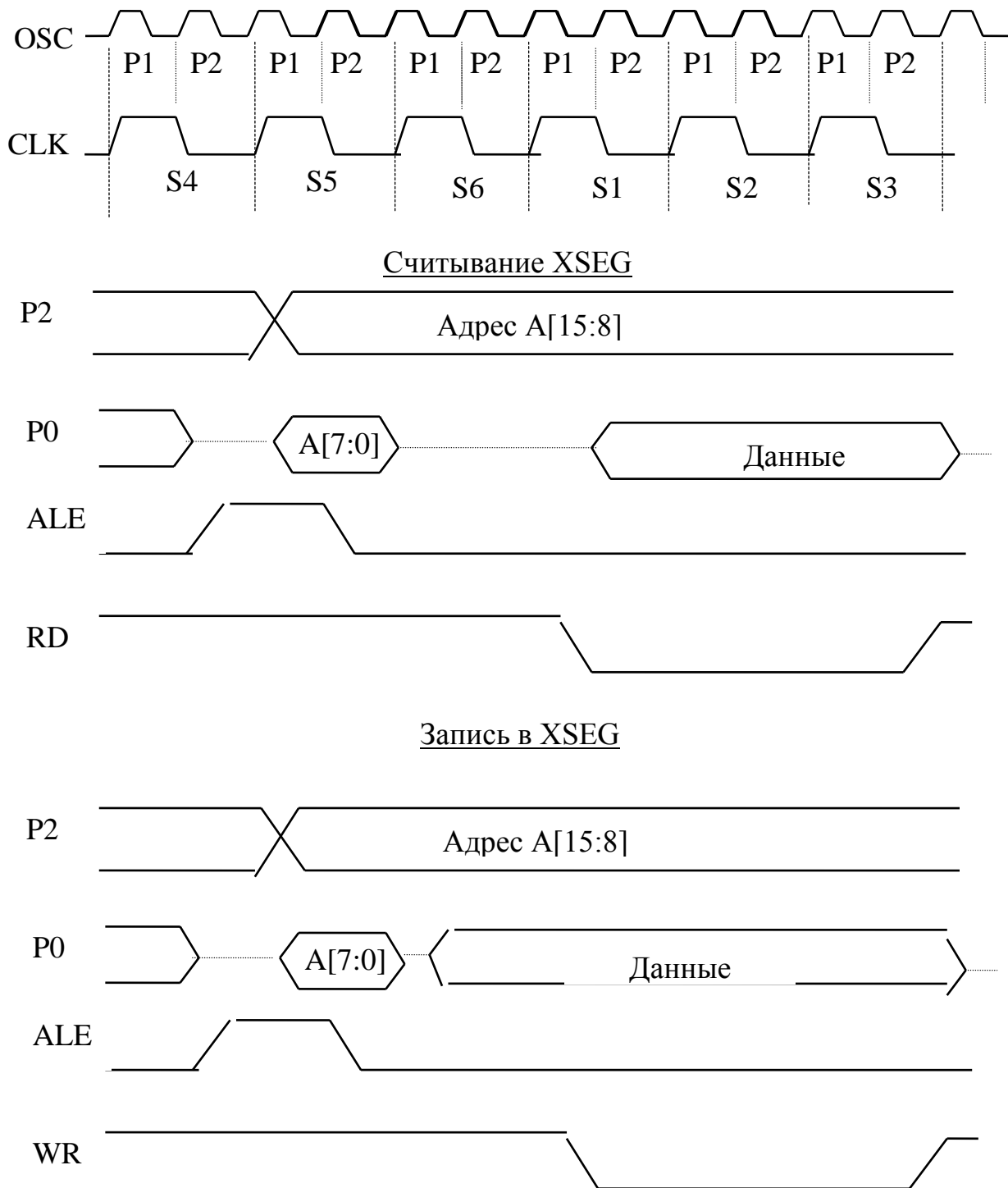


Рис. 1.7. Обращение к внешней памяти данных

При обращении к внешней памяти порты P0 и P2 выполняют функцию системных шин A/D[7:0] и A[15:8] соответственно (см. рис. 1.2). Линии порта P3 используются для передачи управляющих сигналов. При

обращении к внешним объектам содержимое буфера P2 не меняется, а P0 устанавливается в FF.

Каждая линия P0..P3 может быть использована в качестве выходной независимо от других. Для перевода линии в режим ввода в соответствующий разряд выходного регистра должна быть записана "1".

Для использования линий P3 для управления соответствующий разряд должен быть установлен в "1", иначе на выходе всегда будет "0".

	7	6	5	4	3	2	1	0	
Параллельные порты ввода/вывода	87	86	85	84	83	82	81	80	P0 (80)
	97	96	95	94	93	92	91	90	P1 (90)
	A7	A6	A5	A4	A3	A2	A1	A0	P2 (A0)
	RD B7	WR B6	T1 B5	T0 B4	INT1 B3	INT0 B2	TxD B1	RxD B0	P3 (B0)

Буфер и
регистр
управления
/состояния
последователь-
ного порта

SBUF								(99)
SM0 9F	SM1 9E	SM2 D	REN 9C	TB8 9B	RB8 9A	TI 99	RI 98	SCON (98)

Регистр упр.
режимом
Регистр упр.
таймерами

T0 _L								(8A)
T0 _H								(8B)
T1 _L								(8C)
T1 _H								(8D)
T1 MOD				T0 MOD				TMOD (89)
GTE	C/T	M1	M0	GTE	C/T	M1	M0	
TF1 8F	TR1 8E	TF0 8D	TR0 8C	IE1 8B	IT1 8A	IE0 89	IT0 88	TCON (88)

Рис. 1.8. Порты и таймеры

При сбросе микроЭВМ все регистры портов устанавливаются в состояние FF.

Линии порта P0 - с открытым истоком, P1..P3 имеют встроенную нагрузку. Запись результата операции в порты P0..P3 проходит по фазе S6P2, однако на выходе порта новые значения появляются по S1P1.

При записи в порты P1..P3 информации, требующей перехода "0" → "1", через выходную цепь в течение S1 пропускается импульс тока, амплитуда которого в 100 раз превышает номинальное значение, что обеспечивает повышение скорости переключения состояния выходных линий. Порт P0 таким свойством не обладает.

Выходы P1..P3 обеспечивают нагрузку для четырех маломощных ТТЛ-входов (выходной ток - 1,6 мА). Нагрузочная способность выходов P0 - 3,2 мА. Для их работы в качестве порта требуются внешние нагрузочные резисторы.

Считывание данных, присутствующих на входах P0, P1 выполняется по фазе S5P1 последнего машинного цикла команды, а со входов P2, P3 - по фазе S5P2.

Регистр маски прерываний	EA	-	-	ES	ET1	EX1	ET0	EX0	IE (A8)
Регистр приоритетов прерываний	AF	AE	AD	AC	AB	AA	A9	A8	
	-	-	PT2	PS	PT1	PX1	PT0	PX0	IP (B8)
	BF	BE	BD	BC	BB	BA	B9	B8	

Рг.управл.(87)

SMD	-	-	-	GF1	GF0	PD	IDL
-----	---	---	---	-----	-----	----	-----

 PCON

Рис. 1.9. Управление прерываниями

При отсутствии в системе внешней памяти линии управления можно использовать как обычные программируемые линии порта. С другой стороны, пользователь может программировать интерфейс памяти, подавая на выходы RD, WR импульсы произвольной длительности.

Для приведения BE51 в исходное состояние необходимо подать на вход RST импульс Н-уровня длительностью не менее 24 периодов OSC (два МЦ) при условии что генератор запущен.

По сбросу большинство внутренних регистров устанавливается в нулевое состояние. Исключение составляют:

- порты P0..P3 - в FFh;
- указатель стека SP - в 07.

1.6. Краткое описание системы команд BE31(51)

BE51 работает с данными четырех основных типов - битами, полубайтами (4 бита), байтами и адресами (16 бит). Наиболее часто используемой единицей данных является байт: внутренние магистрали данных имеют 8-битный размер. Программная память, внутренняя память и внешняя память данных запоминают и возвращают данные в виде байтов. Кроме этого, имеется много команд, которые работают с одиночными битами. Бит может быть установлен, очищен, проинвертирован, логически скомбинирован с флагом переноса и

проверен для выполнения последующих переходов. Полубайт (двоично-десятичная упакованная цифра) обычно мало применяется в ВЕ51, но двоично-десятичная арифметика может выполняться без преобразования операндов в двоичное представление.

Команды, которые используют 16-битные адреса, имеют дело с указателем данных (16-битный регистр DPTR) и с программным счетчиком (переходы и вызовы подпрограммы). Использование команд сложения с битом переноса (ADDC) и вычитание с заемом (SUBB) делает программирование 16-битной арифметики относительно простым.

Программно-доступными объектами в составе ОМЭВМ являются: аппаратные регистры (см. рис.1.3), четыре банка по восемь регистров R0..R7, R0'..R7 и 128 ячеек внутренней памяти данных - с адресами 00H..7FH. Кроме того, по специальным командам MOVX и MOVC может быть доступна внешняя память данных и программ. В отличие от МК48 в МК51 возможна битовая адресация аппаратных регистров и части внутренней памяти (см. рис. 1.4).

При программировании следует иметь в виду, что в качестве регистров R0..R7 банка 0 используются ячейки внутренней памяти данных с адресами 00H..07H, банка 1 - 08H..0FH, банка 2 - 10H..17H, банка 3 - 18H..1FH.

В системе команд ОМЭВМ серии К1816 используются регистровая, прямая, косвенная и непосредственная адресация.

При *регистровой адресации* операнды находятся в регистрах, имена которых определяются кодом операции команды. Эта адресация используется для обращения к регистрам выбранного банка рабочих регистров, к регистрам A, B, DPTR и к флагу переноса C. Примеры команд с регистровой адресацией:

MOV A,R0 - (пересылка содержимого R0 в аккумулятор);
 MUL AB - (умножение аккумулятора на содержимое регистра B);
 CLR C - (сброс флага C).

По прямому адресу можно обращаться к аппаратным регистрам и внутренней памяти. Используется также прямая битовая адресация. Примеры команд с прямой адресацией:

ADD A,F0H - (сложение аккумулятора и регистра B);
 MOV A,90H - (чтение порта P1 в аккумулятор);
 MOV 90H,A - (вывод содержимого аккумулятора в порт P1).

Косвенно можно адресоваться к ячейкам внутренней и внешней памяти данных через регистры R0, R1 текущего банка, например:

MOV A,@R0 - чтение в аккумулятор содержимого ячейки внутреннего ОЗУ данных, адрес которой хранится в R0;
 MOVX A,@R1 - чтение в аккумулятор содержимого ячейки внешнего ОЗУ данных, младший байт адреса которой хранится в

R1 (старший - в R2).

К любой ячейке внешней памяти (данных и программ) можно обратиться с использованием регистра DPTR:

MOVX @DPTR,A - запись содержимого аккумулятора в ячейку внешнего ОЗУ данных, адрес которой хранится в DPTR;

К командам косвенной адресации, также можно отнести команды PUSH и POP (адрес в регистре SP). В МК51 предусмотрена косвенная адресация памяти программ по сумме базового и индексного (аккумулятора) регистров. Любой байт из памяти программ может быть выбран по адресу, определяемому суммой содержимого DPTR или PC и содержимого A.

Непосредственная адресация позволяет выбрать из адресного пространства памяти программ константы, явно указанные в команде:

MOV A,#8Eh; MOV R6,#40h; MOV @R1,#77h; ANL A,#08h.

1.6.3. По типу операций можно выделить следующие классы команд: пересылки; арифметические и логические; передачи управления; специальные команды.

Класс команд пересылки представлен главным образом разновидностями команд MOV, обеспечивающих пересылку информации между регистрами (в том числе портами), ячейками внутренней памяти данных, аккумулятором, а так же загрузку перечисленных объектов константами (см. примеры в п.1.6.2). Команда MOV применяется и для пересылки отдельных бит. Команда MOVX предназначена для записи и чтения информации из внешней памяти данных. Две команды позволяют считывать в аккумулятор A содержимое ячеек памяти программ:

MOVC A,@A+DPTR - считывает содержимое ячейки памяти программ, косвенно адресуемой через аккумулятор и регистр DPTR

$A \leftarrow \text{ЗУПрг}(\text{DPTR}+A);$

MOVC A,@A+PC - считывает содержимое ячейки памяти программ, косвенно адресуемой через аккумулятор и программный счетчик

$A \leftarrow \text{ЗУПрг}(\text{PC}+A).$

Команды обмена XCH меняют местами содержимое аккумулятора и регистра или косвенно адресуемой ячейки памяти. Команды XCHD меняют местами только младшие тетрады аккумулятора и ячейки памяти, при этом старшие тетрады остаются неизменными.

Команда SWAP A меняет местами старшую и младшую тетрады в аккумуляторе.

Команды PUSH и POP предназначены соответственно для записи и чтения информации из стека.

Арифметические команды включают в себя:

- * ADD - сложение;
- * ADDC - сложение с переносом;
- * SUBB - вычитание с заемом;
- * DA - коррекция 2-х упакованных двоично-десятичных цифр после сложения;
- * MUL - беззнаковое целое умножение;
- * DIV - беззнаковое целое деление;
- * INC - увеличение числа на 1;
- * DEC - уменьшение числа на 1.

В аккумулятор помещается результат функций ADD, ADDC, SUBB и DA и часть результата после MUL и DIV. DEC и INC могут быть применены ко всем байтовым операндам, включая аккумулятор.

Команды, реализующие логические функции, включают в себя:

- * ANL логическое "и" над каждым битом (между двумя байтами или двумя битами);
- * CPL - логическое "не" каждого бита внутри байта или одного бита;
- * ORL - логическое "или" каждого бита (между двумя байтами или двумя битами);
- * XRL - логическое "исключающее или" над каждым битом между двумя байтами;
- * CLR - установка в ноль;
- * SET - установка бита в единицу;
- * RL A и RR A - циклический сдвиг аккумулятора влево и вправо;
- * RLC A и RRC A - циклический сдвиг аккумулятора влево и вправо через триггер флага переноса.

В аккумулятор обычно заносится результат байтовых функций, а во флаг CA обычно заносится результат битовых функций, но некоторые команды помещают результат в указанном байте или бите в области адресов данных.

Действия команд сдвигов описываются следующими выражениями:

```

RL A:   A[7:0] := A[6:0].A[7];
RR A:   A[7:0] := A[0].A[7:1];
RLC A:  C.A[7:0] := A[7].A[6:0].C;
RRC A:  C.A[7:0] := A[0].C.A[7:1].
  
```

При операции умножения содержимое аккумулятора A умножается на содержимое регистра B и результат размещается следующим образом: младший байт в регистре A, старший - в регистре B. В случае выполнения

операции деления целое от деления помещается в аккумулятор А, остаток от деления - в регистр В.

Группа команд передачи управления включает команды безусловных и условных переходов, команды безусловного вызова подпрограмм и две команды безусловного возврата.

Команды безусловного прямого перехода LJMP и безусловного вызова подпрограммы LCALL позволяет осуществить т.н. "длинный" переход - в пределах всей памяти программ. Такой же переход (с косвенной адресацией) позволяет осуществить JMP. Команды AJMP (ACALL) передает управление в пределах области 2 Кбайт. Остальные команды организуют переход в пределах от (PC)-128 до (PC)+127.

Для вычисления смещения в командах коротких переходов необходимо из адреса, на который передается управление вычесть адрес команды, следующей за переходом. Старший байт разности игнорируется, а младший используется в качестве смещения

- SJMP - короткий безусловный переход;
- JC/JNC - условные переходы по переносу/отсутствию переноса;
- JZ/JNZ - условные переходы по нулю/не нулю аккумулятора;
- JB/JNB - условный переход по единичному значению заданного бита;
- JBC - условный переход по единичному значению заданного бита со сбросом этого бита.

К командам передачи управления относится команда CJNE. Команда сравнивает значение первых двух операндов и выполняет ветвление, если результаты не равны.

К командам передачи управления может быть отнесена команда организации цикла DJNZ. Команда декрементирует содержимое заданного регистра и, если результат не равен нулю, осуществляется переход.

Во внутреннем языке K1816BE31 существуют две команды безусловного возврата: RET - из подпрограммы, RETI - из прерывания.

Прочие команды: NOP - пустая операция.

Система команд, упорядоченная по алфавиту, приведена в приложении.

1.7. Указания к выполнению предварительного домашнего задания

Пользуясь "Приложением 1" настоящих методических указаний, составить программу (в мнемосодах и HEX-кодах), обеспечивающую ввод информации с переключателей в микроЭВМ, ее обработку и вывод на индикацию через порт. Ниже приведены различные варианты обработки информации. Программа должна размещаться начиная с адреса 1000H.

1. На переключателях набираются две шестнадцатеричные цифры (по одному на каждой половине). Необходимо перемножить эти цифры и результат вывести на индикацию.
2. Множитель хранится в одном из регистров. Множимое вводится с переключателя. Младший байт результата выводится через порт на индикацию. Старший байт запоминается в любом регистре.
3. Число с переключателей умножается на 10. Младший байт результата выводится на индикацию.
4. К числу с переключателей прибавляется 7. Результат умножается на число, хранимое в R0. Младший байт результата выводится на индикацию. Старший байт помещается в R1.

1.8. Порядок выполнения работы

1. Соединить лабораторный стенд с блоком питания гибким кабелем, обращая особое внимание на правильность соединения разъемов. Подключить блок питания к сети и включить клавишу "Сеть". Если на индикаторах появится "rEAdY", то стенд готов к работе.
2. Просмотреть содержимое аккумулятора и регистров. Изменить содержимое четырех регистров (b0r0, b0r1, b0r2, b0r3), загрузив в них произвольные константы. Зафиксировать значения этих регистров до и после загрузки.
3. Просмотреть содержимое памяти программ. Изменить содержимое нескольких ячеек памяти, начиная с адреса 1000h.
4. Просмотреть содержимое ячеек памяти данных. Записать в четыре ячейки начиная с адреса 20h произвольные константы. Зафиксировать значения этих ячеек.
5. Просмотреть содержимое битов битового процессора. Записать содержимое битов с адресами 00h - 1Fh. Изменить содержимое нескольких следующих битов.
6. Загрузить программу, подготовленную по п.1.7, в память программ и выполнить ее в пошаговом режиме. На каждом шаге программы фиксировать состояние счетчика команд, аккумулятора, регистра PSW, а так же состояние регистра или ячейки памяти, которые меняют свое содержимое в программе.

1.9. Содержание отчета

Отчет должен содержать :
 текст программы, подготовленной в пункте 1.7., в мнемокодах и HEX-кодах с комментариями, представленный в форме таблицы, причем состояние аккумулятора, регистра PSW, а так же состояние регистра или

ячейки памяти, которые меняют свое содержимое в программе следует фиксировать на каждом шаге выполнения программы.

Адрес	HEX-код	Мнемокод	Комментарий
1000	74	MOV A, 00	Очистка аккумулятора
1001	00		
1002	FF	MOV R7,A	R7 := A

и т.д.

1.10. Вопросы для самопроверки

1. Какие классы операций имеются в системе команд ОМЭВМ К1816ВЕ35? Перечислите форматы команд и способы адресации.
2. Над какими объектами К1816ВЕ35 можно осуществлять арифметические и логические операции? Приведите примеры команд.
3. Какими командами можно осуществить ввод байта из портов P1 и P2? В каком состоянии должны при вводе байта находиться триггеры порта? Почему?
1. Какие основные блоки входят в состав однокристалльной микроЭВМ К1816ВЕ31?
2. Охарактеризуйте состав внешнего интерфейса ОМЭВМ К1816ВЕ31.
3. Каким образом подключается клавиатура и дисплей к ОМЭВМ К1816ВЕ31? Охарактеризуйте основные функциональные возможности контроллера К580ВВ79.
4. Как записать новую информацию в регистры банка 1 ОМЭВМ К1816ВЕ31?
5. Как осуществить доступ к внешней и внутренней памяти данных? В чем разница между этими типами памяти?
6. Охарактеризуйте систему команд ОМЭВМ К1816ВЕ31: состав операций, форматы, способы адресации.
7. Какие способы адресации используются для доступа в регистры ОМЭВМ, ячейки внутреннего и внешнего ОЗУ?
8. Какие классы операций имеются в системе команд ОМЭВМ К1816ВЕ31? Перечислите форматы команд и способы адресации.
9. Над какими объектами ВЕ31 можно осуществлять арифметические и логические операции? Приведите примеры команд.
10. Как реализован в ОМЭВМ битовый процессор?
11. Как выполняются команды умножения и деления?
12. Какими командами можно осуществить ввод байта из портов P1 P3? В каком состоянии должны при вводе байта находиться триггеры порта? Почему?

2. ПРОГРАММИРОВАНИЕ ТАЙМЕРОВ И ОБРАБОТКА ПЕРЕРЫВАНИЙ

2.1. Цель работы

Изучение команд, организующих процедуры вызова подпрограмм и возвратов из подпрограмм. Использование таймеров-счетчиков для формирования временных задержек. Изучение подсистемы внешних и внутренних прерываний.

2.2. Подготовка к работе

В процессе подготовки к лабораторной работе необходимо изучить:

- команды передачи управления микроЭВМ K1816BE31, включая команды условных и безусловных переходов, вызовы подпрограмм и возвраты из подпрограмм и прерываний, команды управления таймером-счетчиком и подсистемой прерываний;
- схему подключения кнопок INT0 и T0 к выводам микроЭВМ лабораторной установки УМПК-51;
- рабочее задание и методические указания к его выполнению.

2.3. Описание таймеров/счетчиков

В состав BE51 входят два 16-разрядных таймера/счетчика T0, T1. Состояние таймеров/ счетчиков (далее для краткости - таймеров) отображается регистровыми парами TiH-TiL в пространстве DSEG (см. рис.1.5, 1.8).

В режиме таймеров осуществляется пересчет тактовых сигналов с частотой $OSC/12$, а в режиме счетчиков подсчитываются переходы с H-уровня в L-уровень на соответствующих входах T0, T1. Вход тестируется в течение S5P2 каждого машинного цикла. При обнаружении состояния "1" в одном цикле и "0" - в следующем к соответствующему счетчику добавляется 1. В фазе S3P1 следующего цикла модифицированное значение счетчика отображается в регистровой паре. Т.к. процедура обнаружения перехода длится 2 машинных цикла, максимальная частота счета не должна превышать $OSC/24$ и каждый уровень должен удерживаться неизменным min в течение одного машинного цикла.

Управление режимом работы T0, T1 осуществляет регистр TMOD (89), формат которого приведен на рис. 2.1.

Регистр разбит на два 4-разрядных подрегистра одинакового формата: $T0MOD = TMOD[3:0]$, $T1MOD = TMOD[7:4]$.

Поля регистра TiMOD:

- * M[1:0] - код режима работы таймеров;
- * C/T - выбор функции: 0 - таймер, 1 - счетчик;
- * GATE - флаг управления работой, при GATE = 1 работа разрешена, если (INT=1)&(TR=1) (см. TCON), иначе работа зависит только от TR.

Регистр управления режимом	T1 MOD				T0 MOD				TMOD (89)
	GTE	C/T	M1	M0	GTE	C/T	M1	M0	
Регистр управления таймерами	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON (88)
	8F	8E	8D	8C	8B	8A	89	88	

Рис. 2.1. Форматы регистров TCON и TMOD

Управление работой T0, T1 обеспечивается регистром TCON (88) (аналогично TMOD разбитому на два одинаковых подрегистра).

Поля регистра TiCON:

- * IT - управление типом входа внешнего прерывания, 1 - динамический по срезу, 0 - статический;
- * IE - флаг запроса внешнего прерывания при динамическом входе, при подтверждении прерывания сбрасывается;
- * TR - флаг программного запуска/останова таймера/счетчика;
- * TF - флаг переполнения, вызывающий запрос прерывания, при подтверждении прерывания сбрасывается.

Таймеры T0, T1 могут работать в одном из четырех режимов, схемы работы в которых показаны на рис. 2.2 - 2.4.

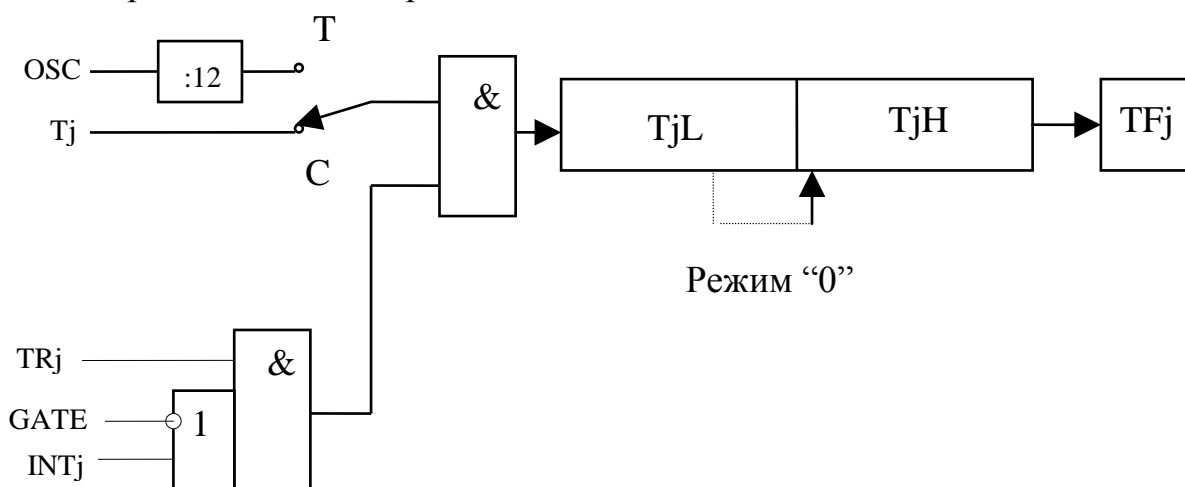


Рис. 2.2. Работа таймеров/счетчиков в режимах "0" и "1"

Режим "0" отличается от режима "1" длиной счетчика (13 бит для "0" и 16 - для "1"). Три старших разряда TjL в режиме "0" игнорируются.

Установка $GATE = 1$ дает возможность таймеру измерить ширину импульса INT_j .

В режиме "2" в восьмиразрядный счетчик T_{jL} производится автозагрузка из T_{jH} . Сигнал переполнения устанавливает флаг TF_j и перезагружает $T_{jH} \rightarrow T_{jL}$.

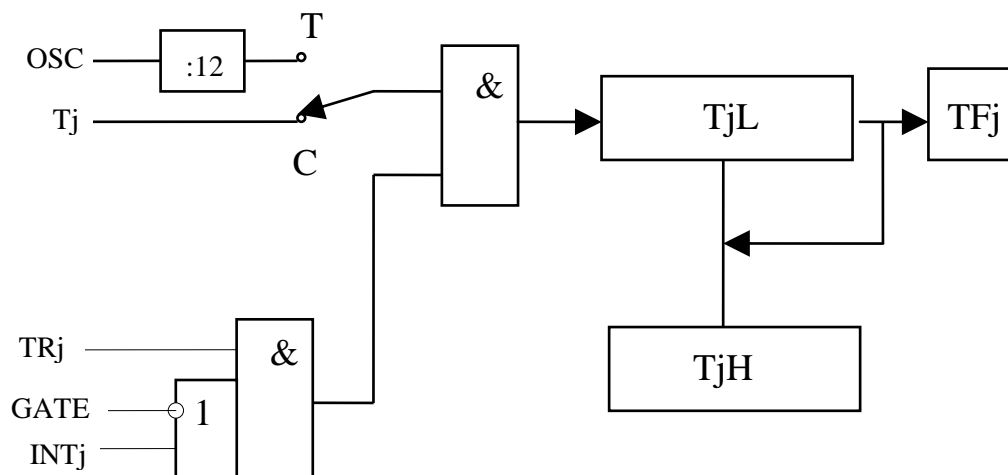


Рис. 2.3. Работа таймеров/счетчиков в режиме "2"

В режиме "3" T_0 и T_1 работают по разному. T_0 работает как два независимых 8-разрядных таймера, причем переполнение T_{0L} устанавливает флаг TF_0 , а переполнение T_{0H} - флаг TF_1 . Таймер T_1 в режиме "3" при $GATE_1 = 0$ постоянно считает. Если $GATE_1 = 1$, то INT_1 используется для разрешения счета. Таймер T_1 тактирует последовательный интерфейс.

2.4. Подсистема прерываний

Архитектура BE51 поддерживает двухуровневую радиальную приоритетную подсистему прерываний (ПП) с шестью источниками запросов. Программное управление ПП осуществляется через два 8-разрядных регистра **IP** (B8) - Interrupt Priority - регистр приоритетов прерываний; **IE** (A8) - Interrupt Enable - регистр разрешения прерываний, форматы которых представлены на рис. 2.5.

Для приема внешних прерываний служат входы INT_0 , INT_1 , которые могут быть независимо друг от друга запрограммированы на срабатывание как по переходу "H" \rightarrow "L", так и по L-уровню. Управление типом входа осуществляется значениями IT_0 и IT_1 - соответственно разряды 0 и 2 регистра TCON. При $IT_i = 1$ устанавливается режим фиксации запроса по спаду (\downarrow) сигнала на входе INT_i , иначе - по низкому уровню на этом входе.

Запросы INT[1:0] устанавливают флажки IE[1:0] - разряды 1 и 3 регистра TCON соответственно.

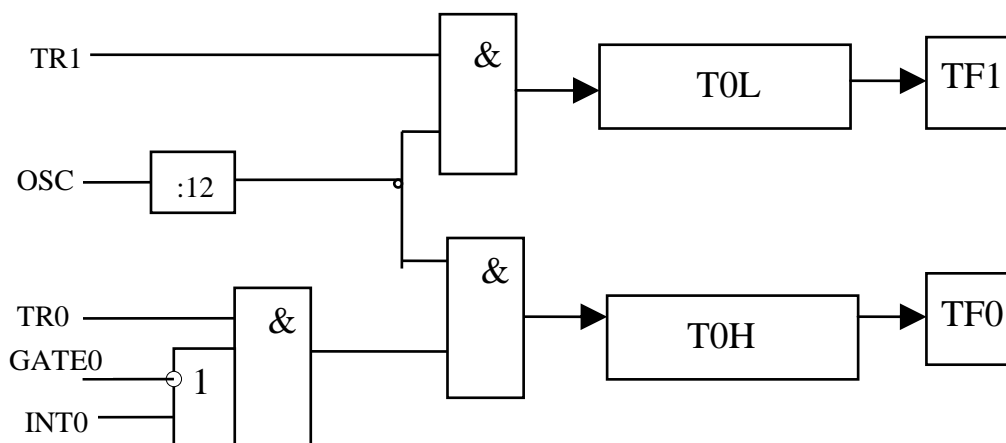


Рис. 2.4. Режим “3” для T0

В случае работы по спаду INT эти флажки сбрасываются автоматически при начале обслуживания прерывания, при работе по уровню флажки отслеживают состояния соответствующих входов INT.

Регистр разрешения прерываний	EA	-	-	ES	ET1	EX1	ET0	EX0	IE (A8)
	AF	AE	AD	AC	AB	AA	A9	A8	
Регистр приоритетов прерываний	-	-	-	PS	PT1	PX1	PT0	PX0	IP (B8)
	BF	BE	BD	BC	BB	BA	B9	B8	

Рис. 2.5. Форматы регистров управления прерываниями

Источниками внутренних запросов могут служить сигналы переполнения таймеров T0, T1 и сигнал окончания работы последовательного канала.

Флажки внутренних запросов от таймеров сбрасываются аппаратно при начале обслуживания прерывания, а флажки последовательного канала TI и RI сохраняются для уточнения источника прерывания программными средствами (т.к. и приемник и передатчик последовательного канала используют общий запрос) и могут быть сброшены программно.

Программно могут быть и установлены все флажки, которые фиксируют запросы, что позволяет активизировать соответствующие процедуры непосредственно из программы. Кроме того, любой флаг запроса может быть проанализирован программно (при выключенной ПП).

Каждый из источников запросов может быть замаскирован с помощью разряда регистра маски IE. В табл. 2.1 приведено назначение

доступа к регистрам IE и IP. При фиксации запроса аппаратно генерируется команда LCALL vect. Процедура обслуживания прерывания должна заканчиваться командой RETI.

Зарезервированные ячейки, предварительно определенные символические имена и соответствующие им прерывающие устройства приведены в табл. 2.2. Программа монитор предусматривает передачу управления при обработке прерываний с использованием команды LJMP на следующие адреса памяти программ:

Таблица 2.2

Источник прерывания	Вектор	Адреса памяти
Внешнее прерывание INT0\	0003H	1750H
Таймер/счетчик T/C 0	000BH	1753H
Внешнее прерывание INT1\	0013H	02F1H
Таймер/счетчик T/C 1	001BH	1759H
Последовательный порт	0023H	175CH

2.5. Указания к выполнению предварительного домашнего задания

Составить программу, обеспечивающую "мигание" светодиодов порта P1 в различных режимах (не менее двух). Режимы мигания:

- мигает только крайний левый светодиод с частотой 0,5Гц;
- мигает только крайний правый светодиод с частотой 2Гц;
- мигают попеременно четные и нечетные светодиоды с частотой 1Гц;
- "бегущий огонек" слева направо с временным интервалом в секунду;
- "бегущий огонек" справа налево с временным интервалом в 2 секунды.

При программировании задержек обязательно использовать внутренние таймеры/счетчики в режиме таймера. Переключение режимов мигания осуществлять путем подачи сигнала внешнего прерывания INT0\.

Первые команды подпрограмм обработки прерываний должны быть командами передачи управления и должны размещаться по адресам, указанным в таблице 2.1.

Подготовленную программу представить в таблице (см. п.1.9 методических указаний к лабораторной работе 1).

2.6. Порядок выполнения работы

2.6.1. Загрузить в ОЗУ подготовленную программу и произвести ее отладку (если необходимо - в пошаговом режиме).

2.6.2. Продемонстрировать работу программы преподавателю на нескольких контрольных примерах.

2.7. Содержание отчета

Отчет должен содержать :

- схему управления внутренними таймерами/счетчиками и расчет значений загружаемых в таймеры констант и параметров программных циклов для получения требуемых задержек;
- блок-схему программы, подготовленной в рамках домашнего задания;
- текст программы в форме таблицы.

2.8. Вопросы для самопроверки

1. В каких режимах могут работать таймеры/счетчики ОМЭВМ?
2. Каков максимальный интервал времени можно отсчитать на таймере при подключении к ОМЭВМ кварцевого резонатора 6МГц?
3. Рассчитайте параметры - константу таймера и количество циклов - для генерации импульсов с частотой 1 сек.
4. Как включить внутренний таймер/счетчик в режим счетчика внешних событий?
5. Каким образом можно программно анализировать состояние таймера/счетчика?
6. Какие прерывания (от каких объектов предусмотрены в ОМЭВМ?
7. Как назначаются приоритеты прерываниям?
8. Как осуществить обработку нескольких внешних прерываний (>2) в ОМЭВМ К1816ВЕ31?

Библиографический список

1. Каган Б.М., Сташин В.В. Основы проектирования МПУ автоматики.- М.: Энергоатомиздат, 1987. -303с.
2. Хвощ С.Т., Варлинский В.В., Попов Е.А. Микропроцессоры и микроЭВМ / Справочник. - Л.: Машиностроение, 1987. -640с.
3. Микропроцессоры и микропроцессорные комплекты интегральных микросхем : Справочник : В 2 т. / Под редакцией В.А.Шахнова - М.: Радио и связь, 1988. - Т1. -368с.,Т2. -368с.
4. Алексенко А.Г., Галицин А.А., Иванников А.Д. Проектирование радиоэлектронной аппаратуры на микропроцессорах. – М.: Радио и связь, 1984. -272с.
5. Однокристалльные микроЭВМ. М.: МИКАП, 1994. 400с.
6. Костров, Б. В. Микропроцессорные системы: учебное пособие / Б. В. Костров, В. Н. Ручкин. - М. : Десс, 2006. - 208 с.
7. Сергиенко, А. Б. Цифровая обработка сигналов: учебное пособие / А. Б. Сергиенко. - 2-е изд. - СПб. : Питер, 2006. - 751 с.

8. Гусев, В. Г. Электроника и микропроцессорная техника: учебник / Ю. М. Гусев. - 3-е изд., перераб. и доп. - М. : Высшая школа, 2004. - 790 с.

Приложение
СИСТЕМА КОМАНД МИКРОЭВМ К1816ВЕ51

MOV	A ,	R _n direc t @R _i #data	XCH	A ,	R _n direc t @R _i		XCHD A, @R_i
	R _n , @R _i ,	A direc t #data					MOVX A, @DPTR MOVX @DPTR, A MOVX A, @R_i MOVX @R_i, A
	direct ,	A R _n direc t @R _i #data					MOVC A, @A+DPTR MOVC A, @A+PC PUSH direct POP direct
<u>Арифметико-логические и специальные команды</u>							

ADD ADC SUB B	A ,	R _n direc t @R _i #data	ANL ORL XRL	A ,	R _n direc t @R _i #data	INC DEC	A , R _n direct @R _i	INC DPTR
	CLR A MUL AB DIV AB			direct ,	A #data	DA A SWAP A RL A RLC A		
			CPL A RR A RRC A					

Команды передачи управления

AJMP ACALL	addr1 1	LJMP LCALL	addr1 6	RET RETI	JMP @A+DPTR SJMP rel
-----------------------------	------------	-----------------------------	------------	---------------------------	---------------------------------------

CJNE	A ,	direct	rel	CJNE	R _n ,	rel	JC	rel	
		#data							direc
					t				JNC
	R _n ,	#data					JZ		
	@R _i ,	#data					JNZ		

БИТОВЫЕ КОМАНДЫ

MOV	C ,	bit	ANL	C ,	bit
	bit ,	C		ORL	

CLR	C ,	rel	JB	bit ,
	bit ,			JNB
				JBC

СИСТЕМА КОМАНД МИКРОЭВМ К1816ВЕ48

МНЕМОКОД		0	1	2	3	4	5	6	7
<i>КОМАНДЫ ПЕРЕСЫЛКИ</i>									
MOV A,R		F8	F9	FA	FB	FC	FD	FE	FF
MOV R,A		A8	A9	AA	AB	AC	AD	AE	AF
MOV R,#D		B8	B9	BA	BB	BC	BD	BE	BF
MOV A,@R		F0	F1						
MOV @R,A		A0	A1						
MOVX A,@R		80	81						
MOVX @R,A		B0	B1						
MOV A,#D	23								
MOV A,PSW	C7								
MOV PSW,A	D7								
MOV A,T	42								
MOV T,A	62								
MOVP A,@A	A3								
MOVP3 A,@A	E3								
XCH A,R		28	29	2A	2B	2C	2D	2E	2F
XCH A,@R		20	21						
XCHD A,@R		30	31						
SWAP	47								
IN A,P		08	09	0A					
OUTL P,A		02	39	3A					
MOVD A,P						0C	0D	0E	0F

МНЕМОКОД		0	1	2	3	4	5	6	7
MOVD P,A						3C	3D	3E	3F
<i>АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ КОМАНДЫ</i>									
ANL A,R		58	59	5A	5B	5C	5D	5E	5F
ORL A,R		48	49	4A	4B	4C	4D	4E	4F
XRL A,R		D8	D9	DA	DB	DC	DD	DE	DF
ANL A,@R		50	51						
ORL A,@R		40	41						
XRL A,@R		D0	D1						
ANL A,#D	53								
ORL A,#D	43								
XRL A,#D	D3								
ANL P,#D		98	99	9A					
ORL P,#D		88	89	8A					
ANLD P,A						9C	9D	9E	9F
ORLD P,A						8C	8D	8E	8F
INC A	17								
INC R		18	19	1A	1B	1C	1D	1E	1F
INC @R		10	11						
DEC A	07								
DEC R		C8	C9	CA	CB	CC	CD	CE	CF
ADD A,R		68	69	6A	6B	6C	6D	6E	6F
ADDC A,R		78	79	7A	7B	7C	7D	7E	7F
ADD A,@R		60	61						
ADDC A,@R		70	71						
ADD A,#D	03								
ADDC A,#D	13								
DJNZ R,Adr		E8	E9	EA	EB	EC	ED	EE	EF

МНЕМОКОД	Код	СЕМАНТИКА КОМАНДЫ
<i>КОМАНДЫ СДВИГОВ, УСТАНОВОК, ИНВЕРСИИ, УПРАВЛЕНИЯ</i>		
RR A	77	Циклический сдвиг A вправо
RL A	E7	Циклический сдвиг A влево
RRC A	67	Циклический сдвиг FC.A вправо
RLC A	F7	Циклический сдвиг FC.A влево
CLR A	27	Сброс A
CPL A	37	Инверсия A
DA A	57	Десятичная коррекция A
CLR C	97	Сброс FC
CPL C	A7	Инверсия FC

МНЕМОКОД	Код	СЕМАНТИКА КОМАНДЫ
CLR F0	85	Сброс F0
CPL F0	95	Инверсия F0
CLR F1	A5	Сброс F1
CPL F1	B5	Инверсия F1
EI	05	Разрешить внешнее прерывание
DIS I	15	Запретить внешнее прерывание
EN TCNTI	25	Разрешить прерывание по T/Сч
DIS TCNTI	35	Запретить прерывание по T/Сч
STRT T	55	Пуск таймера (Т)
STRT CNT	45	Пуск счетчика внешних событий (Сч)
STOP TCNT	65	Стоп таймера/счетчика (Т/Сч)
ENT0 CLK	75	Разрешить выдачу тактового сигнала на T0
SEL MB0	E5	Выбрать банк 0 памяти программ
SEL MB1	F5	Выбрать банк 1 памяти программ
SEL RB0	C5	Выбрать банк 0 регистров
SEL RB1	D5	Выбрать банк 1 регистров
NOP	00	Пустая операция
<i>КОМАНДЫ ПЕРЕДАЧИ УПРАВЛЕНИЯ</i>		
JMP Adr	A ₁₀ A ₉ A ₈ 0 0100	Безусловно
JMPP @A	B3	Безусловно, косвенный
JC Adr	F6	По переносу
JNC Adr	E6	По отсутствию переноса
JZ Adr	C6	По нулевому значению A
JNZ Adr	96	По ненулевому значению A
JF0 Adr	B6	По флагу пользователя F0 = 1
JF1 Adr	76	По флагу пользователя F1 = 1
JT0 Adr	36	По значению входа T0 = 1
JNT0 Adr	26	По значению входа T0 = 0
JT1 Adr	56	По значению входа T1 = 1
JNT1 Adr	46	По значению входа T1 = 0
JTF Adr	16	По флагу T/Сч = 1
JNI Adr	86	По значению входа INT = 0
JBn	nnn1 0010	По значению n-го бита A (=1)
CALL Adr	A ₁₀ A ₉ A ₈ 1 0100	Безусловный вызов подпрограммы
RET	83	Безусловный возврат из подпрограммы (восстановление PC)
RETR	93	Безусловный возврат из прерывания (восстановление PC и PSW)

АЛФАВИТНЫЙ СПИСОК КОМАНД МИКРОЭВМ

МНЕМОКОД		0	1	2	3	4	5	6	7
ADD A,#D	03								
ADD A,@R		60	61						
ADD A,R		68	69	6A	6B	6C	6D	6E	6F
ADDC A,#D	13								
ADDC A,@R		70	71						
ADDC A,R		78	79	7A	7B	7C	7D	7E	7F
ANL A,#D	53								
ANL A,@R		50	51						
ANL A,R		58	59	5A	5B	5C	5D	5E	5F
ANL P,#D		98	99	9A					
ANLD P,A						9C	9D	9E	9F
CALL Adr		A ₁₀ A ₉ A ₈ 1 0100			Безусловный вызов подпрограммы				
CLR A	27	Сброс A							
CLR C	97	Сброс FC							
CLR F0	85	Сброс F0							
CLR F1	A5	Сброс F1							
CPL A	37	Инверсия A							
CPL C	A7	Инверсия FC							
CPL F0	95	Инверсия F0							
CPL F1	B5	Инверсия F1							
DA A	57	Десятичная коррекция A							
DEC A	07								
DEC R		C8	C9	CA	CB	CC	CD	CE	CF
DIS I	15	Запретить внешнее прерывание							
DIS TCNTI	35	Запретить прерывание по T/Сч							
DJNZ R,Adr		E8	E9	EA	EB	EC	ED	EE	EF
EI	05	Разрешить внешнее прерывание							
EN TCNTI	25	Разрешить прерывание по T/Сч							
ENT0 CLK	75	Разрешить выдачу тактового сигнала на T0							
IN A,P		08	09	0A					
INC @R		10	11						
INC A	17								
INC R		18	19	1A	1B	1C	1D	1E	1F
JBn		nnn1 0010			По значению n-го бита A (=1)				
JC Adr	F6	По переносу							
JF0 Adr	B6	По флагу пользователя F0 = 1							
JF1 Adr	76	По флагу пользователя F1 = 1							
JMP Adr		A ₁₀ A ₉ A ₈ 0 0100			Безусловно				

МНЕМОКОД		0	1	2	3	4	5	6	7
JMPP @A	B3	Безусловно, косвенный							
JNC Adr	E6	По отсутствию переноса							
JNI Adr	86	По значению входа INT = 0							
JNT0 Adr	26	По значению входа T0 = 0							
JNT1 Adr	46	По значению входа T1 = 0							
JNZ Adr	96	По ненулевому значению A							
JT0 Adr	36	По значению входа T0 = 1							
JT1 Adr	56	По значению входа T1 = 1							
JTF Adr	16	По флагу T/Сч = 1							
JZ Adr	C6	По нулевому значению A							
MOV @R,A		A0	A1						
MOV A,#D	23								
MOV A,@R		F0	F1						
MOV A,PSW	C7								
MOV A,R		F8	F9	FA	FB	FC	FD	FE	FF
MOV A,T	42								
MOV PSW,A	D7								
MOV R,#D		B8	B9	BA	BB	BC	BD	BE	BF
MOV R,A		A8	A9	AA	AB	AC	AD	AE	AF
MOV T,A	62								
MOVD A,P						0C	0D	0E	0F
MOVD P,A						3C	3D	3E	3F
MOVP A,@A	A3								
MOVP3 A,@A	E3								
MOVX @R,A		B0	B1						
MOVX A,@R		80	81						
NOP	00	Пустая операция							
ORL A,#D	43								
ORL A,@R		40	41						
ORL A,R		48	49	4A	4B	4C	4D	4E	4F
ORL P,#D		88	89	8A					
ORLD P,A						8C	8D	8E	8F
OUTL P,A		02	39	3A					
RET	83	Безусловный возврат из подпрограммы (восстановление PC)							
RETR	93	Безусловный возврат из прерывания (восстановление PC и PSW)							
RL A	E7	Циклический сдвиг A влево							
RLC A	F7	Циклический сдвиг FC.A влево							

МНЕМОКОД		0	1	2	3	4	5	6	7
RR A	77	Циклический сдвиг A вправо							
RRC A	67	Циклический сдвиг FC.A вправо							
SEL MB0	E5	Выбрать банк 0 памяти программ							
SEL MB1	F5	Выбрать банк 1 памяти программ							
SEL RB0	C5	Выбрать банк 0 регистров							
SEL RB1	D5	Выбрать банк 1 регистров							
STOP TCNT	65	Стоп таймера/счетчика (Т/Сч)							
STRT CNT	45	Пуск счетчика внешних событий (Сч)							
STRT T	55	Пуск таймера (Т)							
SWAP	47								
XCH A,@R		20	21						
XCH A,R		28	29	2A	2B	2C	2D	2E	2F
XCHD A,@R		30	31						
XRL A,#D	D3								
XRL A,@R		D0	D1						
XRL A,R		D8	D9	DA	DB	DC	DD	DE	DF