

## 5. ОДНОКРИСТАЛЬНЫЕ МИКРОЭВМ

### 5.1. Однокристальные микроЭВМ К1816ВЕ48/49/35

#### 5.1.1. Структура ОМЭВМ

Восьмиразрядные ОМЭВМ К1816ВЕ48, К1816ВЕ49 и К1816ВЕ35 отличаются лишь объемом и типом памяти программ: ..ВЕ48 - 1К ППЗУ с УФ - стиранием, ..ВЕ49 - 2К ПЗУ с масочным программированием, ..ВЕ35 не содержит на кристалле памяти программ. Все остальные архитектурные и структурные особенности этих микро-ЭВМ идентичны; ниже будет рассмотрена ОМЭВМ ..ВЕ48, структурная схема которой показана на рис. 5.1.

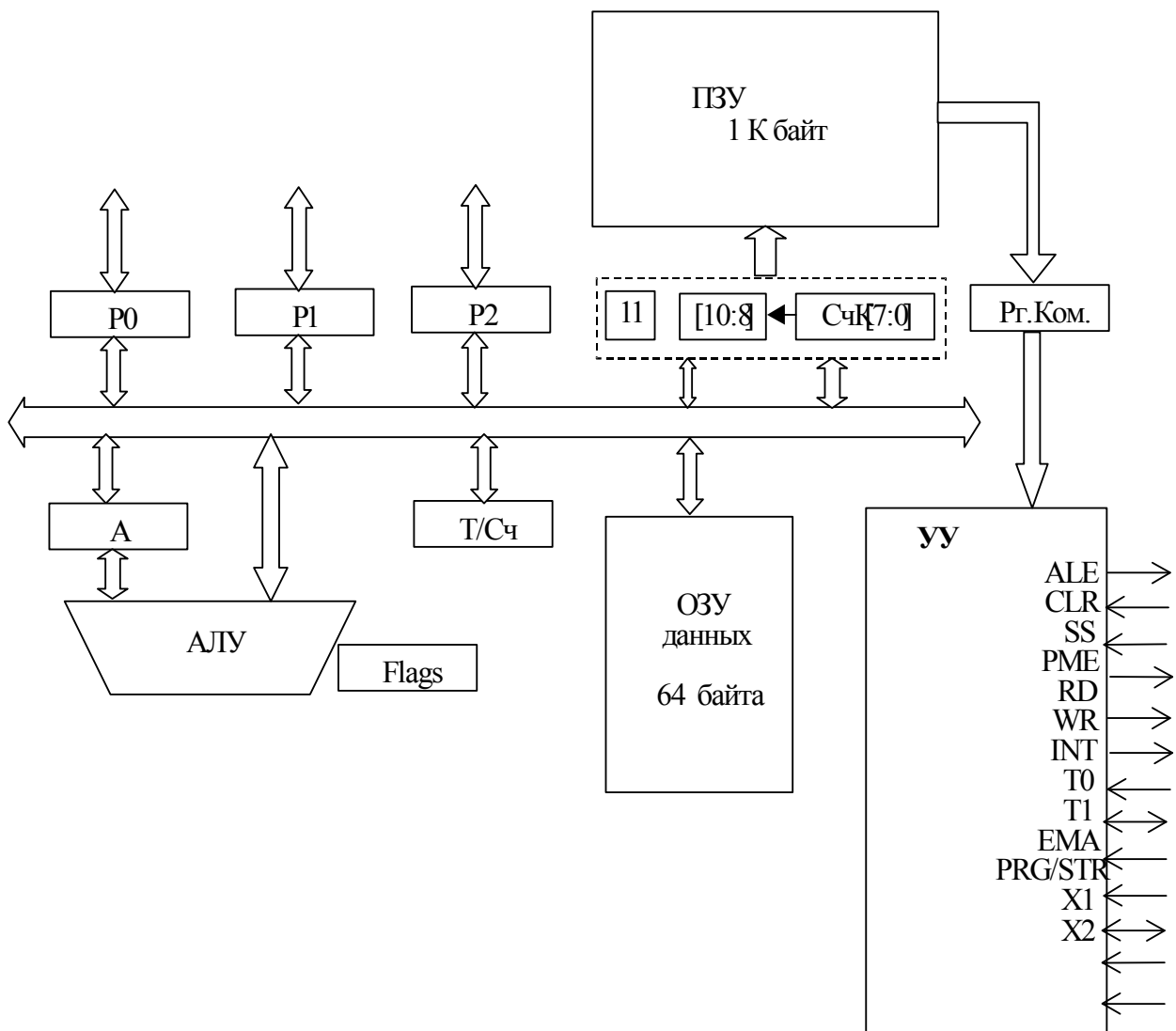


Рис. 5.1. Структурная схема ОМЭВМ К1816ВЕ48

ОМЭВМ включает в себя 8-разрядное АЛУ, ОЗУ программ объемом 64 байта, 8-разрядный регистр команд и 12-разрядный счетчик команд, три 8-разрядных квазидвунаправленных порта ввода/вывода, 8-разрядные регистр-аккумулятор А, таймер/счетчик Т и регистр слова состояния PSW. Устройство управления УУ включает тактовый генератор, автомат управления операциями на кристалле и схему связи с внешней средой. Кроме того, ОМЭВМ ..BE48 содержит на кристалле ППЗУ программ объемом 1К байт.

12-разрядный СчК позволяет процессору управлять памятью команд до 4К, поэтому в системе предусмотрена возможность подключения внешней памяти.

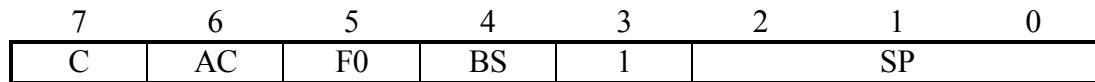
Назначение выводов ОМЭВМ :

DB(P0), P1, P2 - восьмиразрядные двунаправленные порты ввода/вывода. Кроме того, по линиям DB передается младший байт адреса и код команды или данные при связи ОМЭВМ с внешней памятью, а по линиям P2[3:0] - старшие разряды адреса;

- ◆ ALE - строб адреса внешней памяти, сопровождает появление адреса на линиях DB и позволяет "защелкнуть" адрес во внешнем регистре адреса;
- ◆ CR1, CR2 - входы для подключения кварцевого резонатора (1..6 МГц), на вход CR1 можно подавать синхроимпульсы от внешнего тактового генератора;
- ◆ CLR - сброс;
- ◆ SS - управление пошаговым режимом работы ОМЭВМ;
- ◆ PME - строб чтения из внешней памяти программ;
- ◆ RD, WR - стробы чтения и записи во внешнюю память данных;
- ◆ ЕМА - режим внешнего доступа, при ЕМА = 1 производится выборка команд только из внешней памяти программ;
- ◆ INT - радиальный запрос внешнего прерывания, при появлении на входе INT L-уровня и установленном бите разрешения прерывания процессор переходит на выполнение подпрограммы с адреса 003h;
- ◆ T0 - тестируемый вход (имеется команда условного перехода по состоянию T0), может использоваться как выход тактовой частоты внутреннего генератора;
- ◆ T1 - тестируемый вход, может использоваться как вход счетчика внешних событий;
- ◆ PROG/ST - в режиме программирования ППЗУ на этот вход подается импульс записи амплитудой 25 В, в режиме работы ОМЭВМ используется для стробирования передачи данных через порты расширения.

Формат слова состояния PSW показан на Рис.5.2. Трехразрядный указатель стека SP определяет один из восьми уровней стека, расположенного во внутреннем ОЗУ данных. Селектор банка регистров определяет один из

двух банков РОН, являющийся активным. Бит BS является программно-доступным, как и бит F0, по значению которого можно выполнить команду условного перехода. Флаги переноса (из 7 разряда) и дополнительного переноса (из 3 разряда) формируются в арифметических операциях.



- |                          |                                |
|--------------------------|--------------------------------|
| C - флаг переноса;       | BS - селектор банка регистров; |
| AC - флаг доп. переноса; | SP - указатель стека.          |
| F0 - флаг пользователя;  |                                |

Рис.5.2. Формат PSW

Распределение адресного пространства внутренней памяти данных показано на рис. 5.3,а. К каждой ячейке памяти 00..3Fh можно обратиться по косвенному адресу и, кроме того, некоторые ячейки выполняют роль РОНов и стека.

К РОНам можно обращаться по прямому адресу, причем в зависимости от значения BS роль РОНов выполняют ячейки 00..07h или 18..1Fh. Каждый уровень стека занимает 2 байта, в которые записывается 12-разрядный СчК и старшая тетрада PSW (см. Рис.5.3,б). Стек загружается только по команде CALL или по прерыванию, а разгружается - по команде RET.

В ОМЭВМ серии К1816 существует четкое разделение памяти на область программ и область данных. В область программ можно обратиться только по адресу, содержащемуся в СчК. Невозможно разместить фрагмент программы в свободной области памяти данных, т.к. процедуры обращения к ЗУ программ и данных проводятся с выработкой разных управляющих сигналов.

Память программ ОМЭВМ ..BE48/49/35 может составлять до 4К байт, причем часть этой памяти (1-2К) может располагаться на кристалле. ОМЭВМ автоматически генерирует циклы обращения к внутренней памяти программ, если адрес команды (содержимое СчК) лежит в пределах внутренней памяти и на управляющий вход ЕМА подан L-уровень. При ЕМА = 1 независимо от содержимого СчК генерируется цикл обращения к внешней памяти программ. Таким образом, подав H-уровень на вход ЕМА, можно отключить внутреннюю память программ.

К системным ресурсам ОМЭВМ относится 8-разрядный таймер/счетчик внешних событий Т. Это программно-доступный регистр (имеются команды MOV А,Т и MOV Т,А), который может работать (аппаратный суммирующий счетчик) в режиме таймера или счетчика внешних событий.

В режиме таймера подсчитываются импульсы тактового генератора после делителя на 480. Таким образом, одна единица таймера соответст-

вует 80 мкс (при тактовой частоте 6 МГц), а максимальное время счета - 20,4 мс. Запуск таймера осуществляется по команде STRT T, а прекращение счета - по команде STOP TCNT.

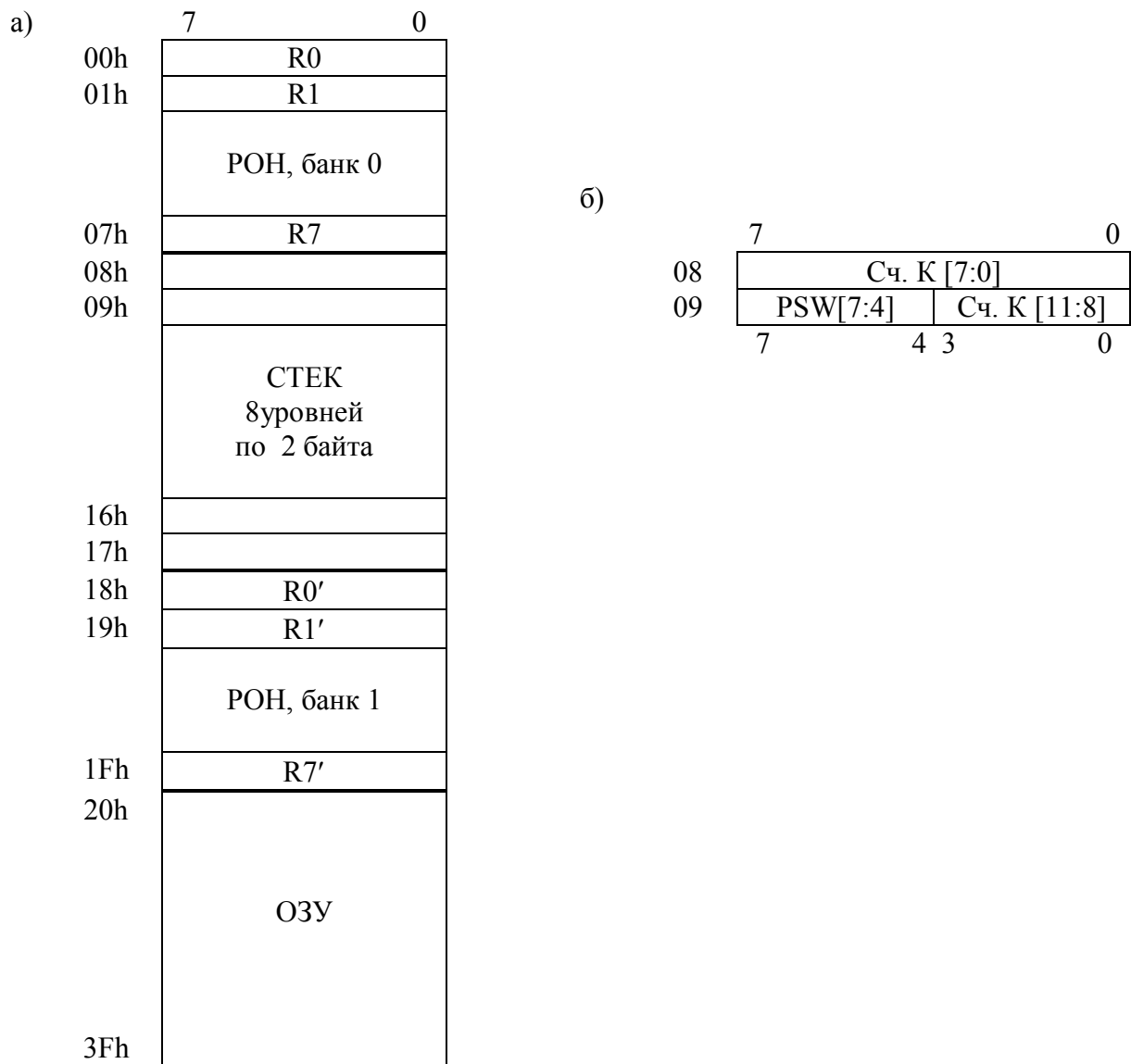


Рис.5.3. Распределение памяти данных и формат уровня стека

В режиме счетчика внешних событий подсчитывается число отрицательных перепадов ( $H \rightarrow L$ ) на входе T1; запуск режима счета внешних событий осуществляется по команде STRT CNT, а прекращение счета – по команде STOP TCNT.

Переполнение таймера/счетчика во всех режимах вызывает установку флага таймера TF, и внутреннее прерывание по адресу 007, если прерывание по таймеру не запрещено.

Содержимое таймера/счетчика может быть в любой момент считано в аккумулятор A и там проанализировано. Предварительной загрузкой в T

константы (из аккумулятора) можно выбрать произвольное время отсчета (в пределах диапазона 0,08 .. 20,4 мС с шагом 0.08 мС)

*Подсистема прерываний* ОМЭВМ включает два радиальных прерывания: внешнее - по входу INT (с вектором 003) и внутреннее - по переполнению таймера/счетчика (вектор 007). Каждое прерывание может быть запрещено программно. Команды управления подсистемой прерываний приведены в разделе (специальные команды).

При необходимости можно использовать вход счетчика внешних событий T1 для подачи второго внешнего радиального прерывания (с вектором 007). В этом случае таймер следует загрузить кодом FF, тогда появление импульса на T1 вызовет переполнение счетчика и прерывание с вектором 007.

### 5.1.2. Порты ввода/вывода

В состав ОМЭВМ входят три 8-разрядных двунаправленных порта ввода/вывода, причем порты P1 и P2 являются фактически квазидвунаправленными. Кроме того, с помощью линий порта P2 можно обращаться к четырем 4-разрядным портам P4..P7.

**Порт P0 (DB)** снабжен двунаправленным буфером и предназначен для связи с внешней памятью. Через DB в сопровождении stroba ALE выдается 8-разрядный адрес внешней памяти данных или младший байт адреса памяти программ. Кроме того, через DB принимаются команды (в сопровождении stroba PМЕ\ ) или принимаются и выдаются данные (по stroбам RD\ и WR\ соответственно. При отсутствии внешней памяти P0 можно использовать для связи с ВУ.

**Порт P0** содержит 8-разрядный регистр, в который записывается информация при выводе в порт. При этом формируется stroб записи WR\ . Информация сохраняется в регистре до поступления новой информации. При считывании из порта информация снимается непосредственно с входных линий и не фиксируется в регистре порта. Считывание сопровождается выдачей stroба WR\ . Над входными данными и содержимым регистра могут производиться логические операции И и ИЛИ.

**Порты P1 и P2** так же содержат 8-разрядные регистры, предназначенные для фиксации выводимой информации (или маски для вводимой). Однако, порты не имеют двунаправленных буферных схем, поэтому для приема байта через P1 (P2) необходимо предварительно записать в регистр порта константу FFh, иначе разряды регистра, хранящие "0", передадут L-уровень на линии внутренней шины и в этих разрядах будут считаны лог."0" независимо от состояния входных линий. На Рис.5.4 представлена схема одного разряда квазидвунаправленного порта.

Блок LB на Рис.5.4 реализует логические операции над содержимым регистра порта и входным словом.

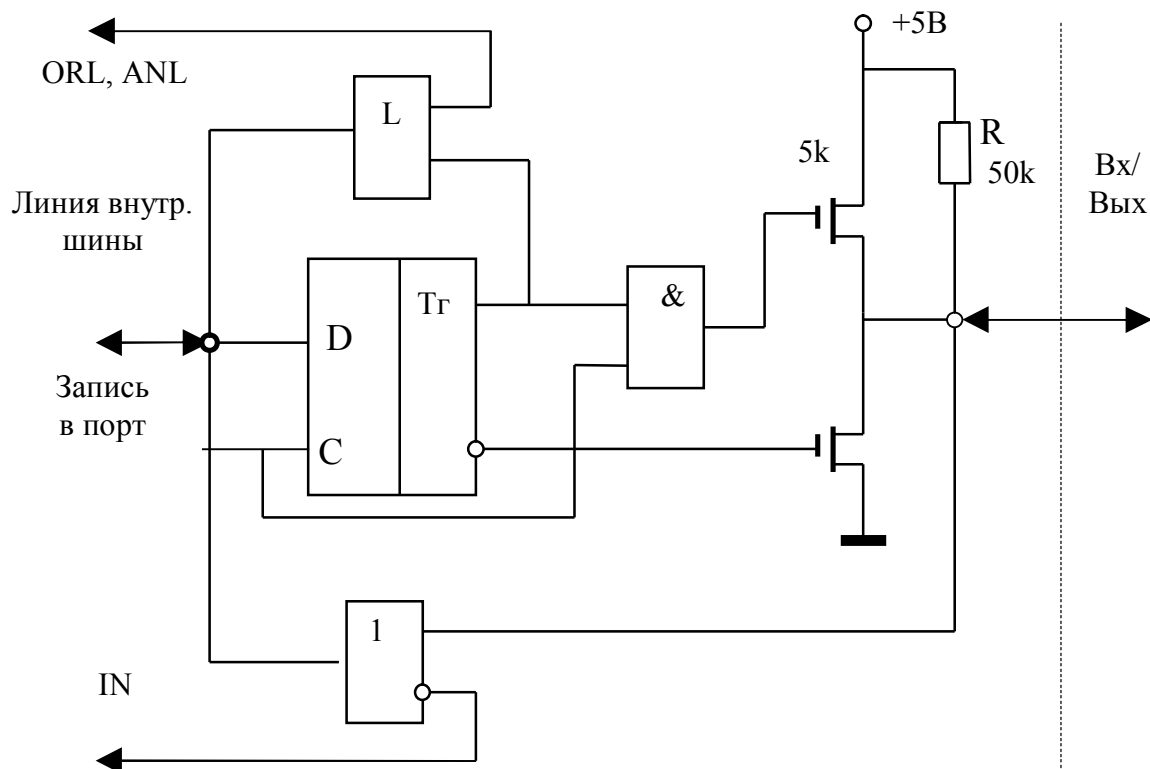


Рис.5.4. Структура разряда квазидвунаправленного порта

В системе команд предусмотрены команды ввода (IN) и вывода (OUT) для портов P0..P2, а так же конъюнкция и дизъюнкция содержимого регистра порта и непосредственного операнда (второй байт команды).

Кроме трех 8-разрядных портов, в ОМЭВМ предусмотрено подключение четырех 4-разрядных портов P4..P7, причем линии P2[3:0] используются для передачи 1/2-байта данных, линии P2[7:6] определяют номер порта, а P2[5:4] - операцию по команде:

0 0	- чтение	- MOVD A, Pp;
0 1	- запись	- MOVD Pp, A;
1 0	- конъюнкция	- ANLD Pp, A;
1 1	- дизъюнкция	- ORLD Pp, A.

В командах с 4-разрядными портами используется младшая тетрада аккумулятора A[3:0]. Синхронизация передачи осуществляется сигналом ST по выходу PROG/ST.

### 5.1.3. Система команд ОМЭВМ

Система команд ОМЭВМ серии К1816 ориентирована на эффективную реализацию процедур управления: маскированный ввод/вывод, логические операции, операции с битами, большое число команд условных переходов, арифметические операции в двоичном и десятичном кодах.

**Программно-доступными объектами** в составе ОМЭВМ серии К1816 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 К1816 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 К1816 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. Две команды позволяют считывать в аккумулятор А 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;

\* RRC 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 @A - безусловный переход по косвенному адресу PC[7:0] := A;

\* JC/JNC - условные переходы по переносу/отсутствию переноса;

\* JZ/JNZ - условные переходы по нулю/не нулю аккумулятора;

\* JT0/JNT0 - условные переходы по состоянию входа T0;

\* JT0/JNT1 - условные переходы по состоянию входа T1;

\* 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,a; где Ri - номер регистра (R0..R7), а - смещение на странице. Команда декрементирует содержимое заданного регистра и, если результат не равен нулю, осуществляется переход.

Команда безусловного вызова подпрограммы CALL позволяет вызывать 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;

*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.

#### 5.1.4. Расширение ресурсов ОМЭВМ

При работе с внешней памятью программ ОМЭВМ выдает младший байт адреса команды через DB в сопровождении строба ALE и старший полубайт адреса СчК[11:8] - через P2[3:0]. В следующем такте снимается младший байт адреса и формируется строб чтения команды из внешней памяти программ РМЕ. Считанный байт команды должен поступать на

линии DB, с которых и передается в регистр команд. На Рис.5.5 показана временная диаграмма обращения ОМЭВМ во внешнюю память программ, а на Рис.5.6 – схема подключения памяти.

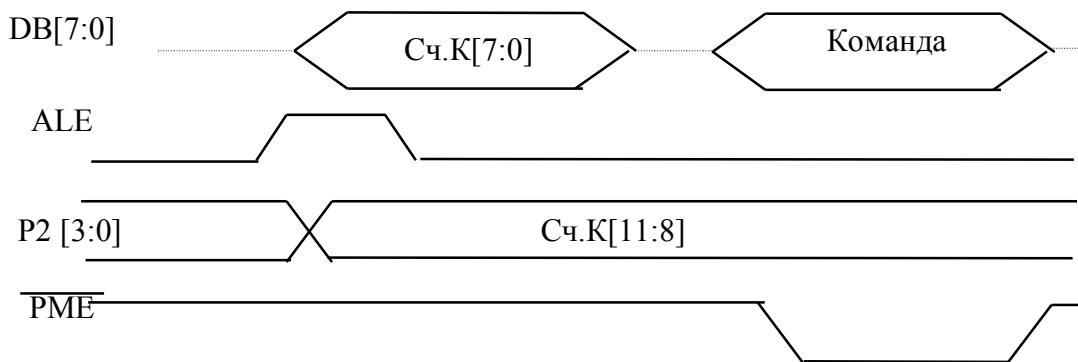


Рис.5.5. Обращение к внешней памяти программ

Независимо от физического воплощения памяти программ, она рассматривается в ОМЭВМ как сплошной массив адресов 000..FFFh, причем адреса 000..7FF составляют нулевой блок памяти, а 800..FFF - первый. Переход из блока в блок путем естественного наращивания адреса СчК невозможен. Так, после команды (однобайтовой) по адресу 7FF будет выполнена команда по адресу 000, а не 800. Перенос в СчК[11] отсутствует, а его изменение возможно только специальными командами SEL MB0 и SEL MB1.

Наличие команд выбора блока памяти (программ) и выбора банка регистров SEL RB0 и SEL RB1 позволяет иметь в системе две в какой-то мере защищенных друг от друга области системных ресурсов. Эти области можно использовать для работы двух независимых программ или области программ и области подпрограмм.

Архитектура ОМЭВМ позволяет расширить *память данных*, подключив дополнительно 0,25К байт внешней памяти. В системе команд предусмотрены специальные команды обращения к внешней памяти данных:

\*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.

Внешняя память данных адресуется только косвенно (через регистры R0 и R1), причем, поскольку мнемоника команд обращения к внешней и внутренней памяти данных различна (MOVX и MOV соответственно), допустимо иметь одинаковые адреса в этих типах памяти. Подключение внешней памяти данных показано на Рис.5.7.

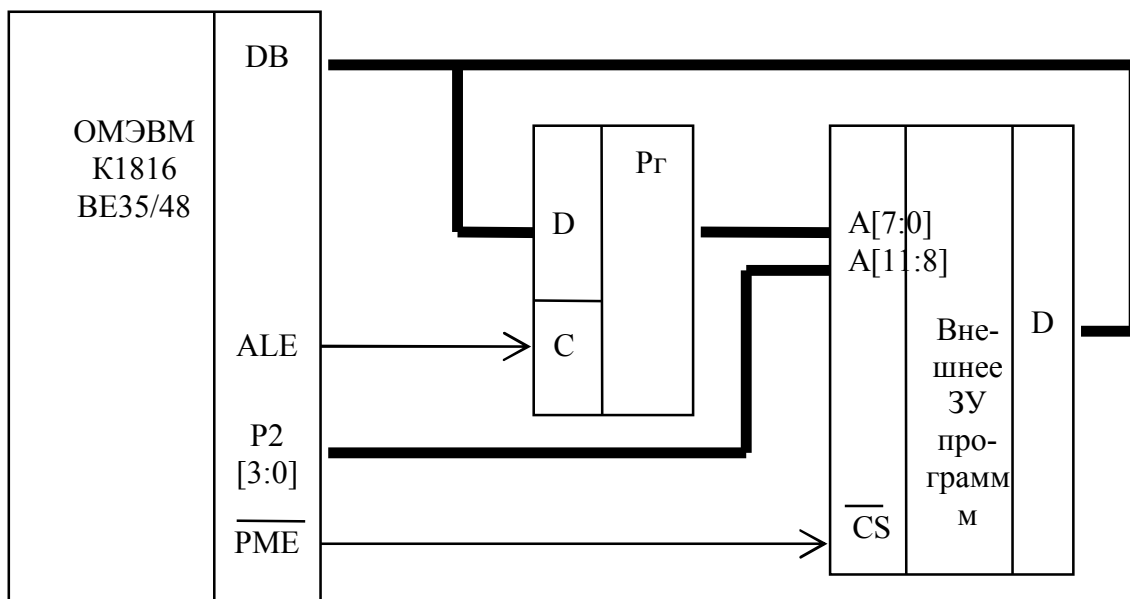


Рис.5.6. Подключение внешней памяти программ

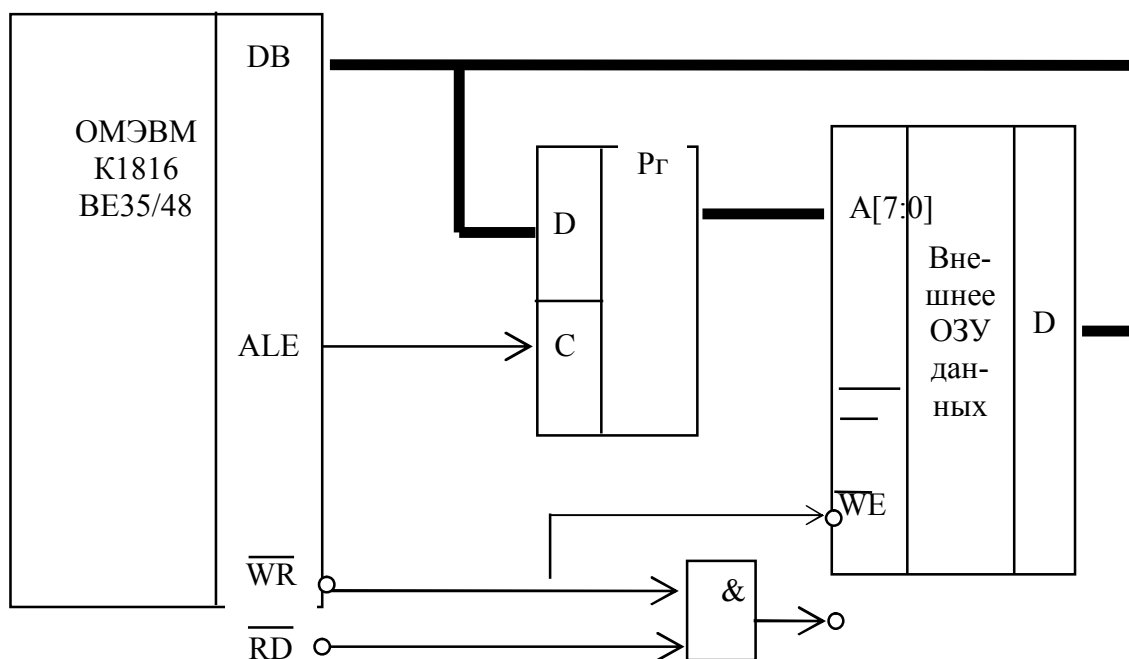


Рис.5.7. Подключение внешней памяти данных

Если системе недостаточно портов микро-ЭВМ (P0..P2, P4..P7), то возможно подключение *внешних контроллеров параллельного обмена*, например K580BB55. При этом допустимы различные варианты подклю-

чения: 1) подключить как внешнюю память данных с обращением по командам MOVX; 2) без использования адресного регистра, обращение по командам MOVX, однако используются только стробы RD и WR при работе команд MOVX. Адрес команды (содержимое регистра R0 или R1) безразличен, а для адресации используются линии порта P1 или P2. Соответствующие разряды должны быть загружены в порт до выполнения команды обращения к внешнему порту MOVX.

## 5.2. Однокристалльная микроЭВМ К1816ВЕ51/52

Однокристалльные микро-ЭВМ К1816ВЕ51/52 (зарубежный аналог – i8051, i8052) представляют как бы второе поколение микро-ЭВМ серии К1816 со значительно большими вычислительными мощностями. 16-разрядный программный счетчик и 16-разрядные регистры косвенного адреса позволяют управлять внешней памятью программ и данных по 64К байт. Значительно расширены возможности системы команд, увеличена номенклатура встроенных ВУ.

МикроЭВМ ..ВЕ52 отличается от ..ВЕ51 большей внутренней памятью данных (256 байт вместо 128) и наличием третьего встроенного таймера. Все остальные ресурсы кристаллов идентичны. Ниже будем рассматривать микроЭВМ К1816ВЕ52.

МикроЭВМ (микроконтроллер) семейства ..ВЕ51(8051) имеют следующие аппаратные особенности:

- внутреннее ОЗУ объемом 128 байт;
- четыре двунаправленных побитно настраиваемых восьмиразрядных порта ввода-вывода;
- два 16-разрядных таймера-счетчика;
- встроенный тактовый генератор;
- адресация 64 Кбайт памяти программ и 64 Кбайт памяти данных;
- две линии запросов на прерывание от внешних устройств;
- интерфейс для последовательного обмена информацией с другими микроконтроллерами или персональными компьютерами.

Микроконтроллер 8751 снабжен УФ ПЗУ объемом 4 Кбайт.

Микроконтроллер 8051 выполнен на основе высокоуровневой n-МОП технологии. Через четыре программируемых параллельных порта ввода/вывода и один последовательный порт микроконтроллер взаимодействует с внешними устройствами. Основу структурной схемы образует внутренняя двунаправленная 8-битная шина, которая связывает между собой основные узлы и устройства микроконтроллера: 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).

### **5.2.1. Семейство однокристальных микроЭВМ MCS-51**

История этого семейства началась с базового микроконтроллера i8051 и за последние 15 лет семейство пополнялось не только изделиями INTEL, но и микроконтроллерами других фирм. Однако почти все модели строго соблюдали архитектурные особенности, предложенные фирмой INTEL.

Но, наряду с существенными преимуществами микроконтроллеры INTEL обладают и недостатками, например: невозможностью построения операционных систем внутри кристалла или сложностью межпроцессорного обмена. Кроме того, часто хотелось бы иметь более высокую производительность при том же потреблении энергии или, наоборот, уменьшить потребление энергии не снижая производительности процессора. Поэтому многие производители семейства MCS51 вносят различные изменения в архитектуру, позволяющие оптимизировать отдельные характеристики микроконтроллера.

Фирма **SIEMENS COMPONENTS INC.** выпускает микроконтроллеры, ориентированные на использование в бытовой технике, в том числе видеотехнике. Микросхемы содержат разнообразные встроенные устройства (АЦП, массивы счетчиков, дополнительные блоки умножения и деления, расширение портов ввода/вывода).

Фирма **PHILIPS SEMICONDUCTORS** выпускает микроконтроллеры, ориентированные на применение в бытовой или автомобильной технике. Благодаря аппаратной реализации шин I<sup>2</sup>C и CAN легко могут быть использованы в инструментальных комплексах. Микросхемы содержат разнообразные встроенные устройства (АЦП, массивы счетчиков, расширение портов ввода/вывода). Имеются варианты с пониженным питающим напряжением и в компактных корпусах.

Фирма **OKI SEMICONDUCTOR** производит полностью статические микросхемы по технологии CMOS, являющиеся функциональным аналогом микросхем фирмы INTEL с дополнительными возможностями. Имеют улучшенные характеристики по цепям питания. Применен корпус, в котором внутренняя шина вынесена наружу в виде разъема для установки ПЗУ.

Фирма **MATRA MHS** производит статические микросхемы, представляющие собой аналоги стандартных моделей семейства MCS51 с улучшенными выходными цепями портов ввода/вывода.

Фирма **ADVANCED MICRO DEVICES (AMD)** выпускает по технологиям CMOS и NMOS функциональные аналоги микросхем фирмы INTEL с дополнительными возможностями.

Фирма **FUJITSU** выпускает функциональные аналоги микросхем i8031, i8051 и i8751 по технологии NMOS.

Объединение ATME1 производит стандартные микроконтроллеры с ПЗУ по технологии Flash объемом 4 Кбайт в обычных и уменьшенных корпусах.

В рамках бывшего 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°C — +70°C);
- расширенный (-40°C — +85°C);
- для военного использования (-55°C — +125°C).

Примерами микроконтроллеров семейства MCS-51 с расширенными возможностями могут служить 8XC51FA, 8XC51GB, 80C152.

### 5.2.2. Структура микроЭВМ К1816BE52

Структурная схема микро-ЭВМ К1816BE52 показана на Рис.5.8.

Большинство сигналов ..BE52 имеют назначение, аналогичное соответствующим сигналам ..BE48:

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,T2 - входы счетчиков внешних событий;

INT0,INT1 - запросы внешних радиальных прерываний; Дополнительно введен сигнал управления счетчиком T2 и последовательные линии связи:

T2EX - управление режимом таймера/счетчика T2;

RxD - вход данных последовательного канала;

TxD - выход данных последовательного канала.

В состав ..BE52, как и ..BE48, входят параллельные 8-разрядные квазидвухнаправленные порты P0..P2. Кроме того, большинство управляющих линий “собрано” в порт P3:

	7	6	5	4	3	2	1	0
P3	RxD	TxD	INT0\	INT1\	T0	T1	WR\	RD\

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.

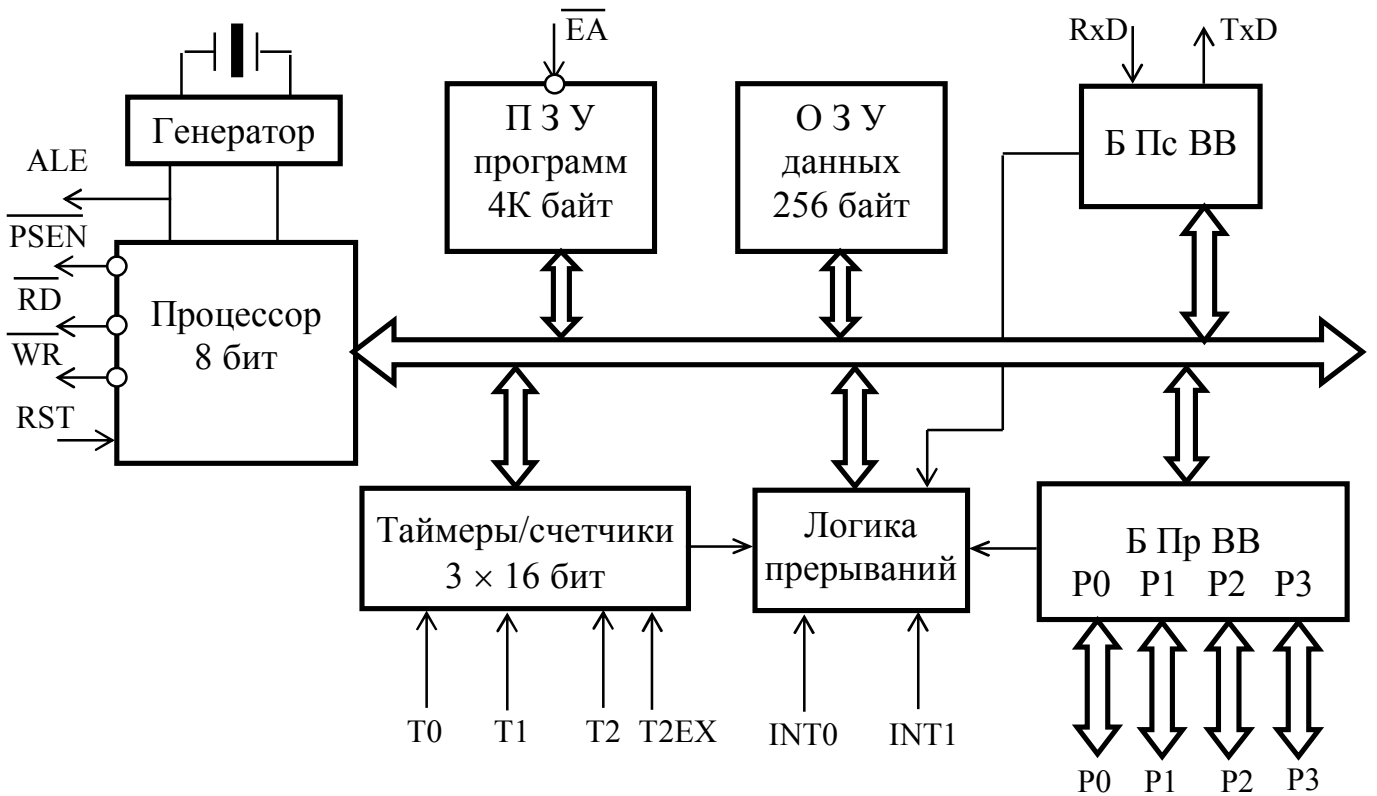


Рис.5.8. Структура микро-ЭВМ К1816ВЕ52

Для управления счетчиком Т2 использованы две линии порта P1: T2 = P1.0, а T2EX = P1.1.

Для подключения внешних ресурсов можно организовать внешние магистрали микро-ЭВМ (рис. 5.9), причем наличие отдельных сигналов для

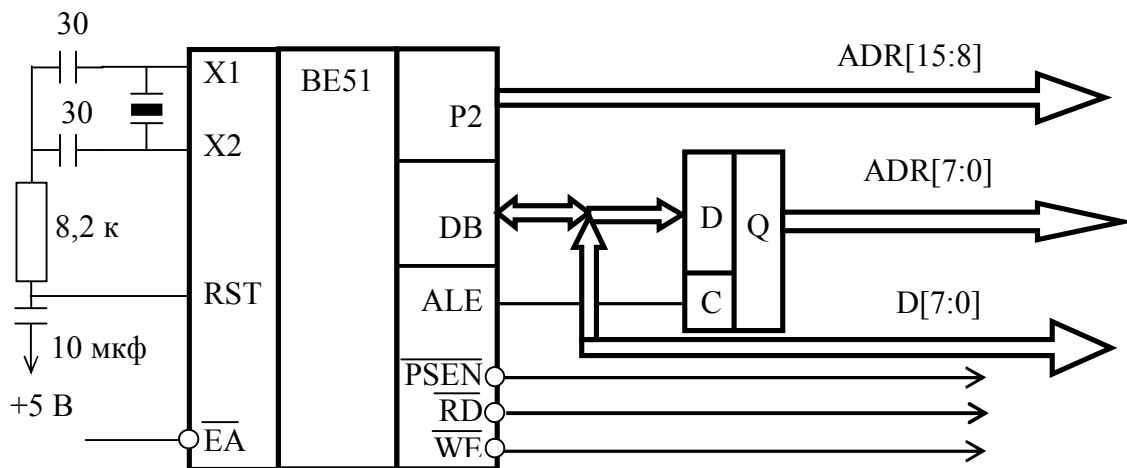


Рис. 5.9. Организация ..BE52 при подключении внешних ресурсов  
 стробирования чтения из памяти программ (PSEN) и обращения к памяти данных (RD, WR) позволяет иметь непересекающиеся пространства для этих типов внешней памяти.



Назначение выводов микроЭВМ K1816BE 51/52 приведен на рис. 5.10. На рисунке приняты следующие обозначения:

- $U_{ss}$  — потенциал общего провода ("земли");
- $U_{cc}$  — основное напряжение литания +5 В;
- X1, X2 — выводы для подключения кварцевого резонатора;
- 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 память; уровень 0 на этом входе заставляет микроконтроллер выполнять программу только внешнее ПЗУ; игнорируя внутреннее(если последнее имеется);

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.5.10. Назначение выводов K1816BE51/52.

- P1 — восьмибитный квазидвунаправленный порт ввода/вывода: каждый разряд порта может быть запрограммирован как на ввод, так и на вывод информации, независимо от состояния других разрядов;
- P2 — восьмибитный квазидвунаправленный 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). Выводы порта используются при программировании 8751 для ввода в микроконтроллер старших разрядов адреса:

- P3 — восьмибитный квазидвунаправленный порт, аналогичный P1; кроме того, выходы этого порта могут выполнять ряд альтернативных функций, которые используются при работе таймеров, порта последовательного ввода-вывода, контроллера прерываний, и внешней памяти программ и данных;
- P0 — восьмибитный 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.

### 5.2.3. Архитектурные особенности микроЭВМ

Микро-ЭВМ ..BE52 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 64К байт);
- CSEG - пространство программного кода (до 64К байт).

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.

Характерно, что все порты ввода/вывода, системные регистры, таймеры так же отображены на пространство DSEG.

Пространство регистров представлено четырьмя банками регистров по 8 РОН в каждом, а так же 16-разрядными программным счетчиком PC и регистром косвенного адреса DPTR, 8-разрядными аккумуляторами A и B, указателем стека SP и регистром PSW (Рис.5.11)

Все регистры, показанные на Рис.5.11, кроме PC, отображаются на DSEG.

PSW[7:5] имеют тот же смысл, что и в ..BE48:

PSW[7] - CY - перенос из старшего (7) разряда АЛБ;

PSW[6] - AC - перенос из третьего разряда АЛБ;

PSW[5] - F0 - флаг пользователя.

Кроме того, формат PSW включает двухбитовое поле PSW[4:3] - RS номера банка регистров и флаги арифметического переполнения OV (PSW[2]) и четности P (PSW[1]).

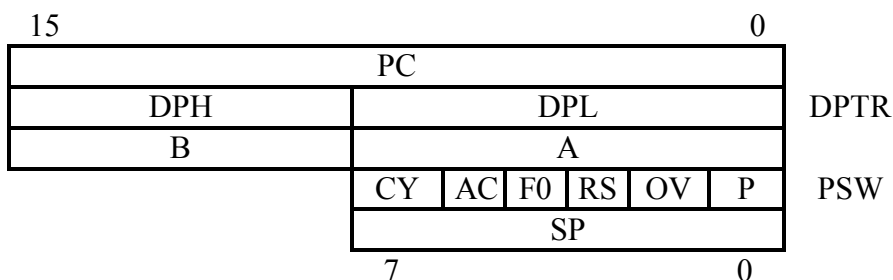


Рис.5.11. Основные регистры микро-ЭВМ

Флаги CY, AC и OV отражают признаки результата последней арифметической операции, а P - четность содержимого аккумулятора А.

Формат регистра флагов PSW представлен на рис. 5.12.

Символ	Позиция	Имя и назначение																				
P	PSW.0	Флаг приоритета. Устанавливается и сбрасывается аппаратурно в каждом цикле команды и фиксирует нечетное/четное число единичных бит в аккумуляторе																				
-	PSW.1	Не используется																				
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратурно при выполнении арифметических операций																				
RS0 - RS1	PSW.3 - PSW.4	Биты выбора используемого банка регистров. Могут быть изменены программным путем <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>RS0</th> <th>RS1</th> <th>Банк</th> <th>Границы адресов ОЗУ</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>00H - 07H</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>08H - 0FH</td> </tr> <tr> <td>0</td> <td>1</td> <td>2</td> <td>10H - 17H</td> </tr> <tr> <td>1</td> <td>1</td> <td>3</td> <td>18H - 1FH</td> </tr> </tbody> </table>	RS0	RS1	Банк	Границы адресов ОЗУ	0	0	0	00H - 07H	1	0	1	08H - 0FH	0	1	2	10H - 17H	1	1	3	18H - 1FH
RS0	RS1	Банк	Границы адресов ОЗУ																			
0	0	0	00H - 07H																			
1	0	1	08H - 0FH																			
0	1	2	10H - 17H																			
1	1	3	18H - 1FH																			
F0	PSW.5	Флаг пользователя. Может быть установлен, сброшен или проверен программой пользователя																				
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратными средствами при выполнении команд сложения и вычитания и сигнализирует о переносе или заеме в бите 3 аккумулятора																				
CY	PSW.7	Флаг переноса. Устанавливается и сбрасывается как аппаратурно, так и программным путем																				

Рис. 5.12. Перечень флагов, их символические имена и условия формирования

Расширение аккумулятора В используется в командах умножения и деления, а в остальных операциях - как обычная ячейка памяти.

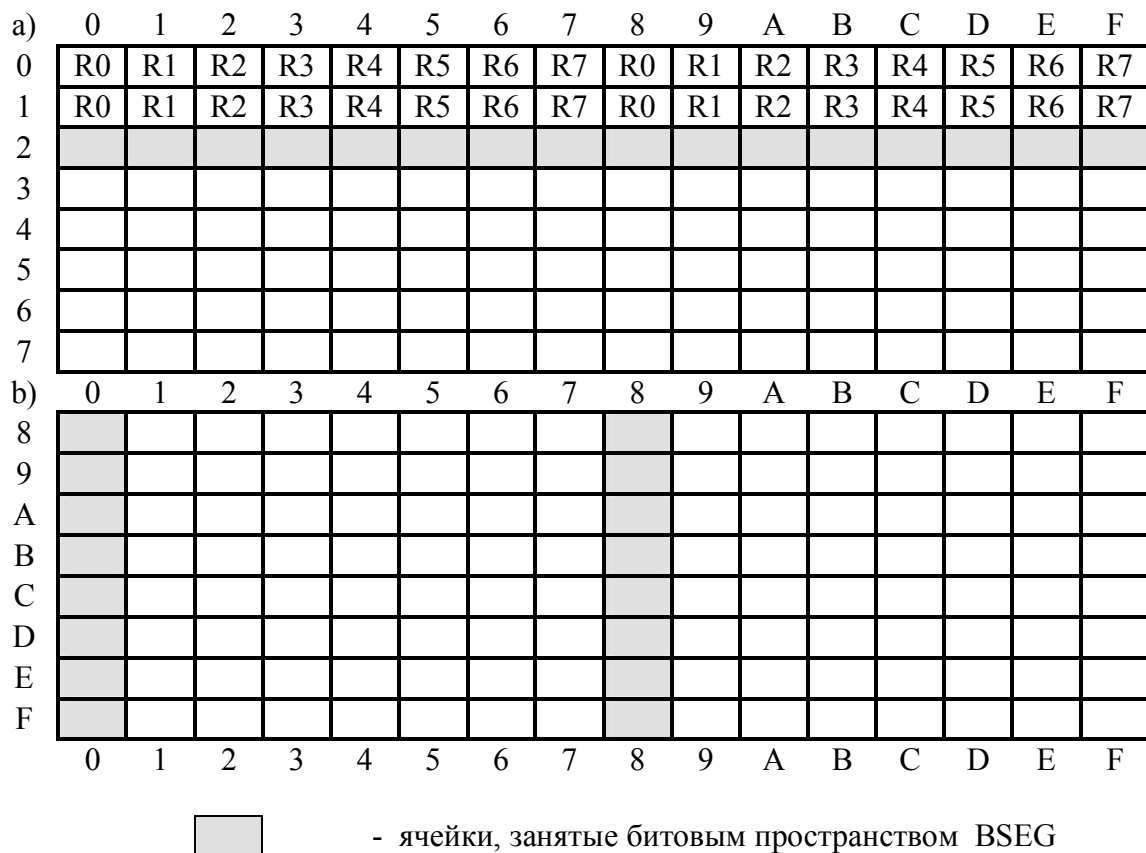
Программный счетчик адресует пространство памяти программ CSEG объемом до 64К байт, причем переход из области внутренней памяти программ к внешней осуществляется, как и в ..BE48, автоматически.

Указатель данных DPTR используется для обращения к XSEG и при

пересылке констант из CSEG в А. Кроме того, содержимое DPTR используется в качестве смещения в команде перехода.

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 А, В, PC и DPTR загружаются 0h, а в SP <07h. Таким образом обеспечивается преемственность с ..BE48.

#### 5.2.4. Организация внутренней памяти данных



	0	1	2	3 ..	..7	8	9	A	B	C	D..
8	P0	SP	DPTL	DPTH	PCON	TCON	TMOD	T0L	T0H	T1L	T1H
9	P1					SCON	SBUF				
A	P2					IE					
B	P3					IP					
C						T2CON		RCAP 2 <sub>L</sub>	RCAP 2 <sub>H</sub>	T2 <sub>L</sub>	T2 <sub>H</sub>
D	PSW										
E	A										
F	B										

Рис.5.13. Пространство внутренней памяти данных DSEG

Пространство DSEG включает в себя 256 ячеек памяти, часть из которых является одновременно элементами других пространств (Рис.5.13). Так, 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 и косвенному адресу, обратиться как к аккумулятору А и как к полю BSEG[E0..E7] (к каждому биту в отдельности).

Адресация к каждому отдельному биту пространства BSEG (строка 2 пространства DSEG) показана на рис. 5.14.

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

Рис. 5.14. Адресация битового пространства BSEG (строка 2 пространства DSEG)

Адресация к каждому отдельному биту пространства BSEG (столбцы [08...0F] и [88...8F] пространства DSEG) показана на рис. 5.15.

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. 5.15. Адресация битового пространства BSEG (столбцы [08...0F] и [88...8F] пространства DSEG)

Столбец [08...0B] пространства DSEG является одновременно пространством RSEG.

Обратиться к каждому отдельному биту пространства BSEG можно с помощью команд побитовой обработки.

Память программ адресуется PC[15:0] и может составлять до 64К байт, причем младшие 4..8К могут располагаться непосредственно на кристалле микро-ЭВМ (РПЗУ или ПЗУ), а остальная память - внешнее ЗУ. С точки зрения программиста внешняя и внутренняя память программ представляют единое адресное пространство. Имеются “особые точки” CSEG :

RESET	- 0000h	- стартовый адрес;
EXTI0	- 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;
TIMER2	- 002Bh	- прерывание от таймера/счетчика T2.

### 5.2.5. Внешние устройства микроЭВМ

К “внешним” устройствам микроЭВМ К1816BE52 отнесем:

- 4 параллельных двунаправленных порта ввода/вывода P0..P3;
- буфер SBUF и регистр управления SCON последовательного канала;
- таймеры/счетчики T0, T1, их регистр управления TCON и регистр режимов TMOD;
- таймер/счетчик T2, его буферный регистр RCAP2 и регистр управления T2CON;
- регистры управления подсистемой прерываний: регистр приоритетов IP и регистр маски IE;
- регистр управления машиной PCON.

Форматы этих регистров и их размещение в пространствах DSEG и BSEG показаны на рис. 5.16.

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.

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

Характерно, что 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\, PSEN\ импульсы произвольной длительности.

Буферы портов P0..P2 ..BE52 отличаются от соответствующих буферов ..BE48. Структура разрядов порта P0 и P3 приведены на

Рис.5.17 и Рис.5.18. 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” (Рис.5.8), иначе на выходе всегда будет “0”.

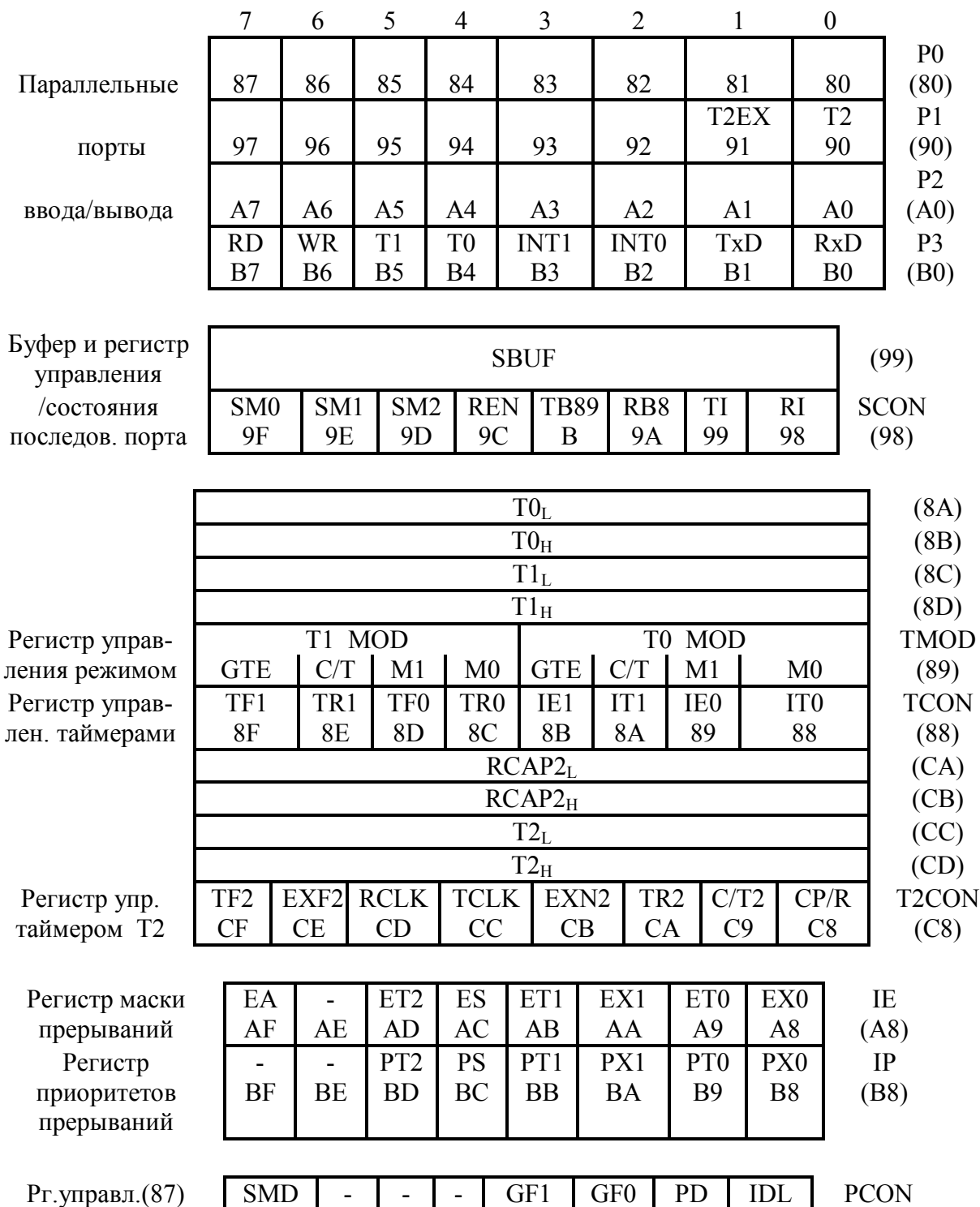


Рис.5.16. Размещение регистров внешних устройств в DSEG и BSEG и форматы регистров

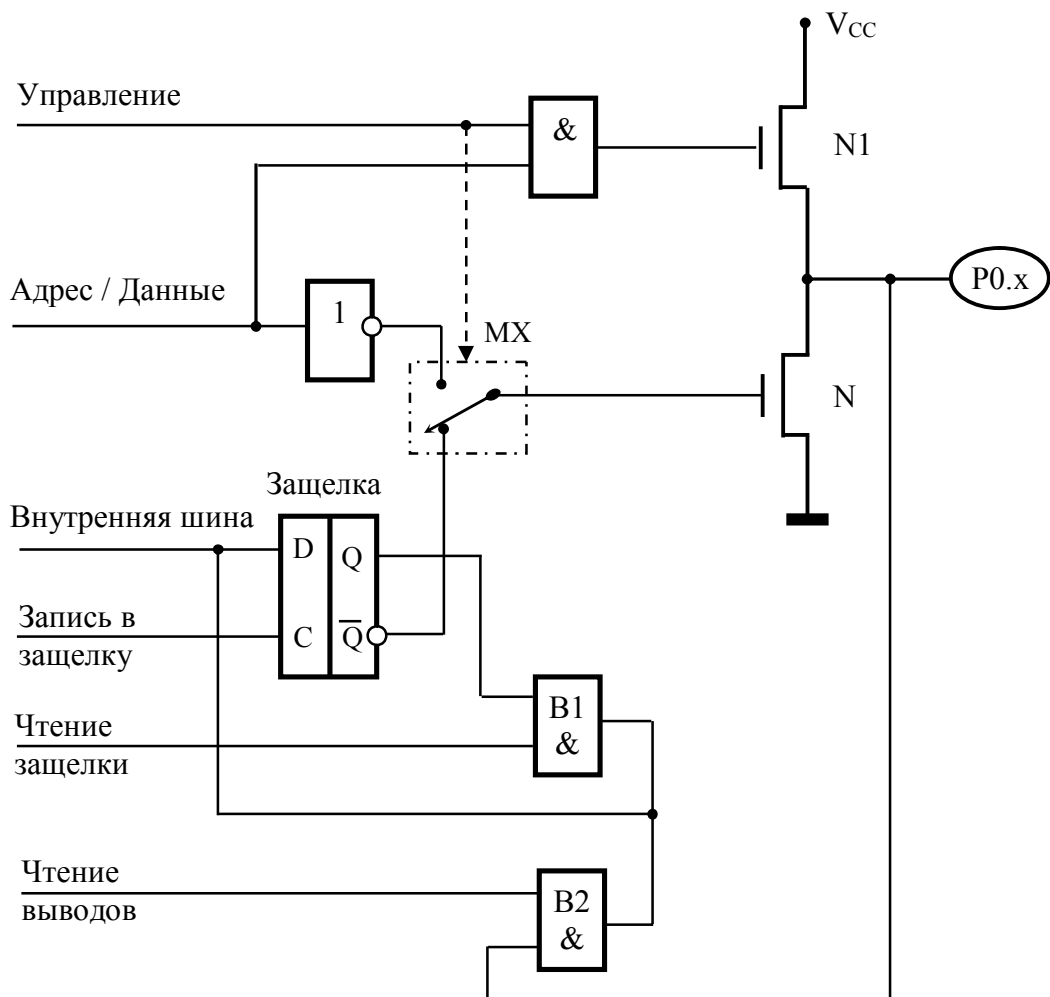


Рис.5.1 7. Разряд портов P0...P2

При сбросе микро-ЭВМ все регистры портов устанавливаются в состоянии FF.

Линии порта P3 - с открытым стоком, P0...P2 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, однако для их работы 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.

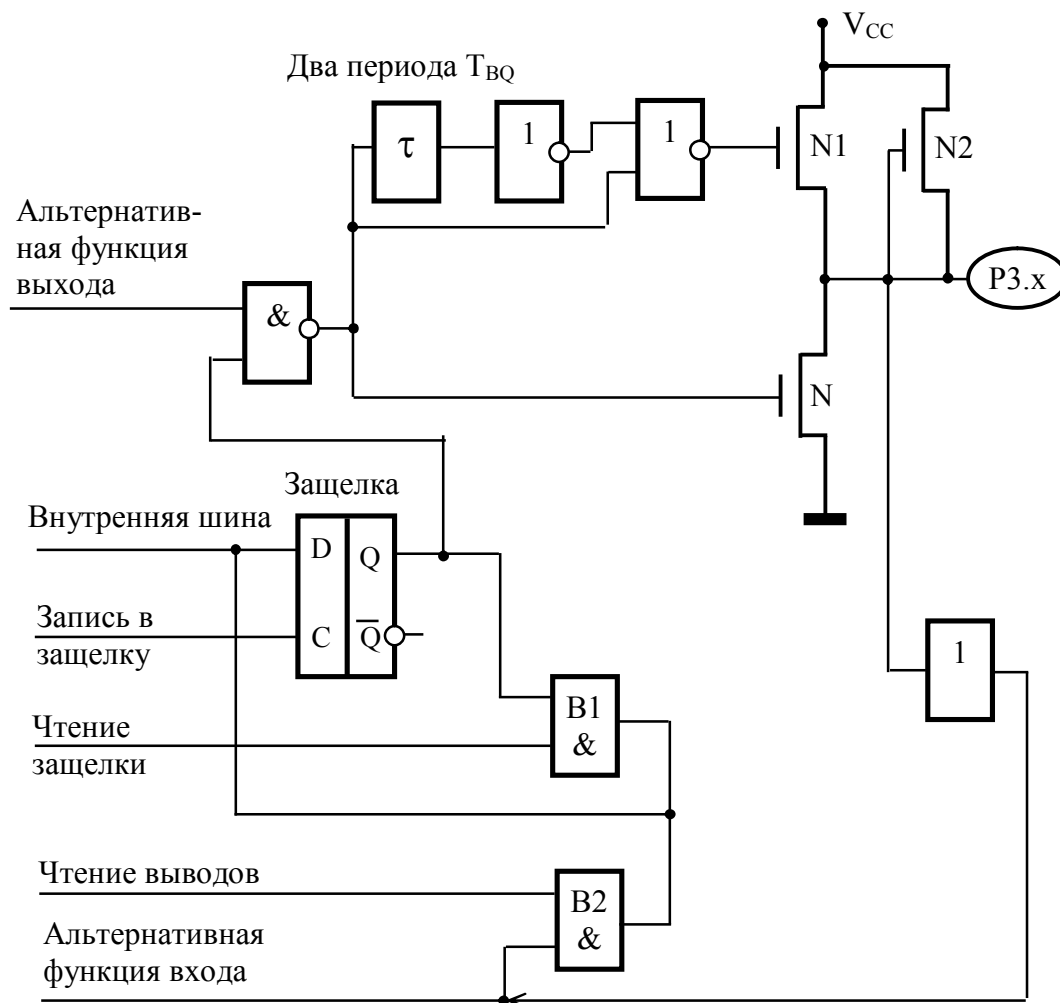


Рис.5.18. Разряд порта P3

### 5.2.6. Описание последовательного порта

В состав ..VE51 входит дуплексный последовательный канал связи с буферизацией, который может быть запрограммирован для работы в одном из четырех режимов:

- режим "0" - синхронный ввод/вывод с частотой  $OSC/12$ ;
  - режим "1" - асинхронный с 8-бит. кадром, частота  $k \times f$ ;
  - режим "2" - асинхронный с 9-бит. кадром, частота  $k \times OSC/32$ ;
  - режим "3" - асинхронный с 9-бит. кадром, частота  $k \times f$ ;
- где  $k \in \{1, 1/2\}$ ,  $f$  - частота переполнения T1 ( $f_{ov}$ ), деленная на 16.

Входные и выходные данные хранятся в регистре SBUF(99), управление работой приемопередатчиков осуществляется словом состояния и управления SCON (98), формат которого представлен в Таблица 5.1.

Таблица 5.1

№ бита	Имя	Назначение	
0	RI	Флаг прерывания приемника	
1	TI	Флаг прерывания передатчика	
2	RB8	8 бит приемника (реж. 2 и 3)	
3	TB8	8 бит передатчика (реж. 2 и 3)	
4	REN	Разрешение приема	
5	SM2	Запрещение приема кодов с b8 = 0	
6	SM1	00 - "0"	10 - "2"
7	SM0	01 - "1"	11 - "3"

В режиме "0" (синхронный) данные передаются или принимаются по линии RxD со скоростью OSC/12 младшими битами вперед по 8 бит за операцию. Для синхронизации внешних средств используется линия TxD (Рис.5.19).

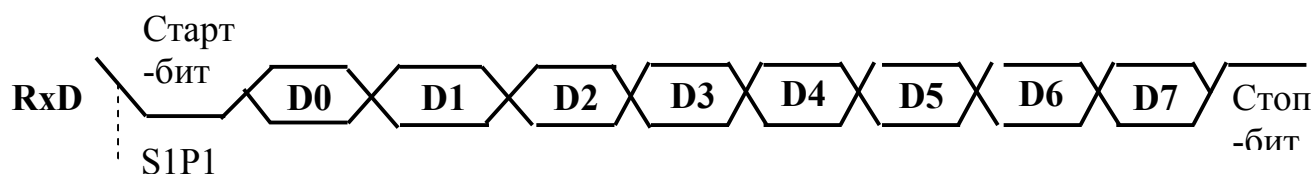


Рис.5.19. Прием/передача в синхронном режиме

Для повышения помехоустойчивости приема в режимах 1..3 каждый бит данных опрашивается трижды. Для этого период передачи бита делится на 16 интервалов и опросы осуществляются в 7, 8, 9 интервалах; значение бита определяется мажорированием. До прихода стартового бита приемник RxD проверяется 16 раз за период. При обнаружении перехода "H" → "L" начинается отсчет. Если стартовый бит не подтверждается (опрос 3 раза), то переход принимается за помеху. При обнаружении стартового бита осуществляется прием следующих 8 или 9 битов (в зависимости от режима) и размещение их в SBUF и флажке RB8. При приеме стопового бита устанавливается флаг RI, свидетельствующий об окончании приема очередного кадра.

Передача данных на выход TxD начинается после записи их в SBUF независимо от флага TI. Передача стартового бита начинается в фазе S1P1 МЦ, следующего за сигналом переполнения того счетчика, который служит генератором скорости. Кадр завершается выдачей стопового бита. Пе-

ред началом передачи стоп-бита устанавливается флаг TI - окончание передачи данных.

Флажки RI и TI способны осуществлять запросы на прерывания, причем в режимах 1..3 при SM2 = 1 установка флага RI (и генерация запроса) возможна только при приеме кадра с D8 = 1. Для режима 1 это соответствует приему стопового бита. Таким образом имеется возможность (в режимах 2 и 3) разделять принимаемые кадры на два класса (по D8 → RB8). Кадры, принимаемые с RB8 = 0 теряются. Это дает возможность эффективно управлять межмашинной связью в сетях с единой последовательной магистралью. Кадры с RB8 = 1 - широковещательные и содержат адрес приемника, информации которому будет передаваться кадрами с RB8 = 0. Для вызова другого абонента следует передать новый кадр с RB8 = 1.

Скорость передачи зависит от состояния флага SMOD в регистре PCON (87), режима работы канала и длительности периода пересчета таймера-генератора скорости.

Для режима 2: при SMOD = 1 - скорость OSC/32;  
SMOD = 0 - скорость OSC/64.

В режимах 1 и 3 скорость передачи определяется частотой переполнения таймера T1( $f_{ov}$ ) следующим образом:

$$f = (2^{SMOD}/32) \times f_{ov}.$$

При использовании таймера T1 в режиме автозагрузки скорость последовательного обмена определяется по следующей формуле:

$$f = (2^{SMOD} \times f_{OSC}) / (32 \times 12 \times [256 - (TH)]).$$

В случае SMOD = 1 скорость совпадает с частотой переполнения таймера, при SMOD = 0 - скорость в два раза ниже.

### **5.2.7. Таймеры-счетчики**

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.

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  $t_{in}$  в течение одного машинного цикла.

Функционирование таймера T2 несколько отличается от работы T0, T1.

Управление режимом работы T0, T1 осуществляет регистр TMOD (89), формат которого приведен на Рис.5.20.

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.5.20. Форматы регистров TCON и TMOD.

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

Поля регистра T1MOD:

- 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.

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

Поля регистра TiCON:

- IT - управление типом входа внешнего прерывания, 0 - динамический по срезу, 1 - 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.5.21, Рис.5.22, Рис.5.23.

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 INTj.

В режиме "2" в восьмиразрядный счетчик TjL производится автозагрузка из TjH. Сигнал переполнения устанавливает флаг TFj и перезагружает TjH → TjL. Следует иметь в виду, что программная установка флага TFj не вызывает перезагрузки. Перезагрузка TjH → TjL не меняет содержимого TjH.

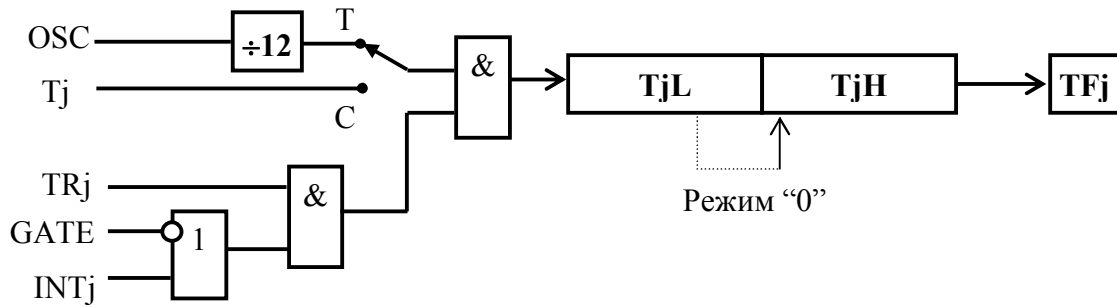


Рис.5.21. Работа таймеров/счетчиков в режимах "0" и "1"

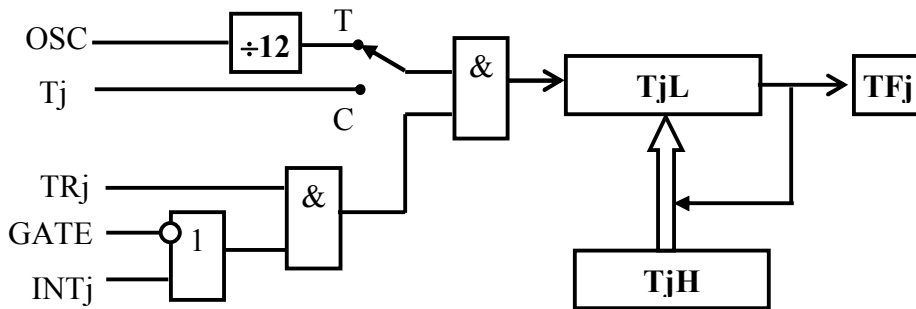


Рис.5.22. Работа таймеров/счетчиков в режиме "2"

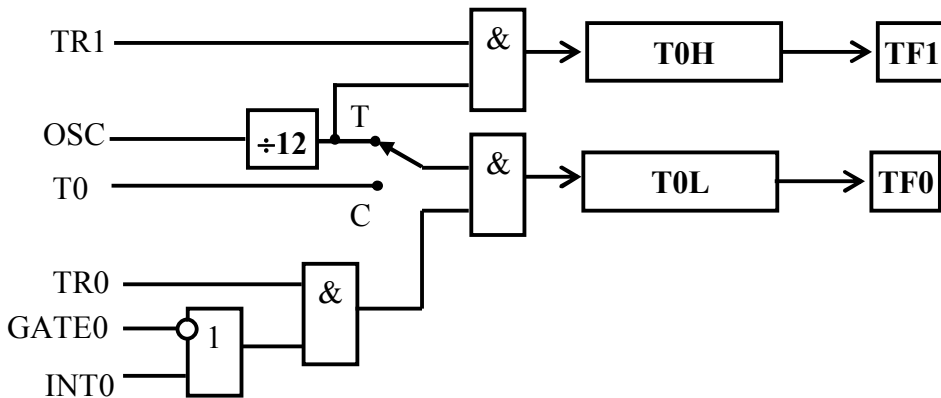


Рис.5.23. Режим "3" для T0

В режиме "3" T0 и T1 работают по-разному. T1 в этом режиме заблокирован и просто сохраняет свое состояние (значение в регистрах T1L и T1H), как и в случае TR1 = 0.

T0 работает как два независимых 8-разрядных таймера, причем T0L может работать как в режиме таймера, так и в режиме счетчика. За ним сохраняются все биты управления T0, он реагирует на воздействия по входам T0, INT0, устанавливает флаг TF0.

T0H может работать только в режиме таймера. Он использует бит включения TR1, при переполнении выставляет флаг TF1. Других битов управления устройство в этом режиме не имеет.

Установка T0 в режим “3” лишает T1 бита включения TR1, поэтому в этом случае T1 в режимах “0”, “1”, “2” при GATE = 0 всегда включен и при переполнении в режимах “0” и “1” обнуляется, а в режиме “2” перезагружается, не устанавливая флаг. Управление от входов INT\, T1, биты управления C/T1, GATE1 не зависят от режима T0.

T1 аппаратно связан с блоком синхронизации последовательного канала. При работе в режимах “0”, “1”, “2” T1 всегда вырабатывает импульс тактирования последовательного канала (который канал может и не использовать).

16-разрядный таймер/счетчик T2 обладает дополнительными возможностями по сравнению с T0, T1.

Помимо регистровой пары T2H.T2L (8B.8A), отображающей состояние таймера/счетчика, предусмотрена регистровая пара RCAP2 (H - CBh, L - CAh), в которую осуществляется *автозахват* и из которой осуществляется *автозагрузка* T2. Для внешнего управления процессами автозахвата/автозагрузки предусмотрен дополнительный вход T2EX.

Управление работой T2 осуществляется с помощью регистра T2CON (C8), формат которого приведен на Рис.5.24.

		RCAP2 <sub>L</sub>								(CAh)
		RCAP2 <sub>H</sub>								(CBh)
		T2 <sub>L</sub>								(CCh)
		T2 <sub>H</sub>								(CDh)
Регистр управл. таймером T2	TF2	EXF2	RCLK	TCLK	EXN2	TR2	C/T2	CP/R	T2CON	
	CF	CE	CD	CC	CB	CA	C9	C8	(C8h)	

Рис.5.24. Регистры таймера T2

Разряды регистра T2CON имеют следующее назначение:

CP/RL2 - флаг захвата/автозагрузки T2. При CP/RL2 = 1 захват текущего состояния осуществляется по срезу T2EX, если EXEN2 = 1. При CP/RL = 0 разрешается автозагрузка T2, если T2 переполнен или по срезу T2EX, если EXEN2 = 1. Этот флаг не принимается во внимание, если RCLK = 1 или TCLK = 1, т.е. когда T2 работает в качестве генератора скорости последовательного канала;

C/T2 - управление функцией таймера (0) или счетчика (1);

TR2 - программный запуск T2 при TR2 = 1 и останов по TR2 = 0;

EXEN2 - флаг разрешения (по “1”) динамическому (\\_) входу осуществлять захват/автозагрузку, если он не работает в режиме генератора скорости последовательного порта;

TCLK - при TCLK = 1 T2 служит генератором скорости для передатчика последовательного канала, работающего в режиме “1” или “3”. При TCLK = 0 генератором скорости служит T1;

RCLK - то же для приемника последовательного канала;

EXF2 - флаг выполнения захвата/автозагрузки, вызванного изменением состояния на входе T2EX при EXEN2 = 1. При разрешенном прерывании вызывает запрос. Сбрасывается только программно;

TF2 - флаг переполнения T2. Вызывает запрос на прерывание. Функции установки подавляются при TCLK = 1 или RCLK = 1.

Режимы работы T2 (их три) определяются состоянием разрядов 4, 5 и 0 регистра T2CON.

RCLK  $\vee$  TCLK = 1 - режим генератора скорости последовательного канала. В этом случае остальные флаги режимов игнорируются, а выработка запроса прерывания по переполнению подавляется;

CP/RL = 1 - режим захвата;

CP/RL = 0 - режим автозагрузки.

*Режим захвата* осуществляется путем копирования текущего состояния T2 в регистры RCAP2 по отрицательному перепаду ( $\setminus$ ) импульса на входе T2EX при условии EXEN2 = 1. Захват сопровождается установкой флага EXF2.

В *режиме автозагрузки* каждое переполнение T2 вызывает установку TF2 и перезагрузку RCAP2  $\rightarrow$  T2. Перезагрузку можно осуществить и от внешнего источника - по отрицательному перепаду сигнала на входе T2EX (при условии EXEN2 = 1), причем такая перезагрузка сопровождается установкой флага EXF2.

### 5.2.8. Подсистема прерываний

Архитектура ..BE51/52 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.5.5.25.

Для приема внешних прерываний служат входы INT0 $\setminus$ , INT1 $\setminus$ , 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 IT0 и IT1 - соответственно разряды 0 и 2 регистра TCON. При TI = 1 устанавливается режим фиксации запроса по спаду ( $\setminus$ ) сигнала на входе INT, иначе - по низкому уровню на этом входе.

Регистр разрешения прерываний	EA	-	ET2	ES	ET1	EX1	ET0	EX0	IE (A8)
	AF	AE	AD	AC	AB	AA	A9	A8	
Регистр приоритетов прерываний	-	-	PT2	PS	PT1	PX1	PT0	PX0	IP (B8)
	BF	BE	BD	BC	BB	BA	B9	B8	

Рис.5.25. Форматы регистров управления прерываниями

Запросы INT[1:0] устанавливают флажки IE[1:0] - разряды 1 и 3 регистра TCON соответственно. В случае работы по спаду INT эти флажки сбрасываются автоматически при начале обслуживания прерывания, при работе по уровню флажки отслеживают состояния соответствующих входов INT.

Источниками внутренних запросов могут служить сигналы переполнения таймеров T0, T1, T2 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 Таблица 5.2 приведено назначение разрядов этого регистра. Прерывание разрешается при наличии "1" в соответствующем разряде регистра IE.

Таблица 5.2

Разряд IE	Имя	Назначение
0	EX0	Разрешение IE0 или INT0\
1	ET0	Разрешение TF0 (таймер T0)
2	EX1	Разрешение IE1 или INT1\
3	ET1	Разрешение TF1 (таймер T1)
4	ES	Разрешение TI ∨ RI (послед. канал)
5	ET2	Разрешение TF2 ∨ EXF2 (таймер T1)
6	-	Не используется
7	EA	Общее разрешение прерывания

Таким образом, в ПП ..VE51 имеют место следующие радиальные запросы (каждый из которых, как отмечалось выше, может быть установлен и программно):



Таблица 5.3

Флаг	Тип прерывания	Адрес подпрограммы
IE0	внешнее прерывание INT0\	0003h
TF0	прерывание от таймера T0	000Bh
IE1	внешнее прерывание INT1\	0013h
TF1	прерывание от таймера T1	001Bh
RI\TI	флаг последовательного порта	0023h
TF2\ EXF2	прерывание от таймера T2 или по факту захвата/автозагрузки	002Bh

Эти запросы могут быть:

- 1) запрещены все - при  $IE[7] = 0$ ;
- 2) произвольно замаскированы разрядами регистра  $IE[4:0]$ ;
- 3) произвольно разбиты на два непересекающихся подмножества прерываний с высшим и низшим приоритетом обслуживающих подпрограмм.

Разбиение на два подмножества по приоритетам программ осуществляется с помощью разрядов регистра  $IP[4:0]$ . В разряды  $IP$ , соответствующие прерываниям, отнесенным к высшему приоритету, следует установить "1", к низшему - "0".

Процедура обслуживания низкоприоритетного запроса может быть прервана высокоприоритетным запросом; обслуживание высокоприоритетного запроса не прерывается.

При одновременном появлении нескольких одинаково приоритетных запросов выбирается для обслуживания запрос согласно строгому приоритету:

$$IE0 \rightarrow TF0 \rightarrow IE1 \rightarrow TF1 \rightarrow RI\backslash TI \rightarrow TF2\backslash EXF2.$$

высший

низший

Запросы могут быть приняты на обслуживание в конце каждого КЦ, кроме циклов исполнения команды  $RETI$  или команд с любым видом доступа к регистрам  $IE$  и  $IP$ . При фиксации запроса аппаратно генерируется команда  $LCALL \text{ vect}$ . Процедура обслуживания прерывания должна заканчиваться командой  $RETI$ .

В состав  $..BE52$  входит *регистр управления* -  $PCON$  (Power Control), формат которого зависит от технологии изготовления БИС. В БИС, реализованных по n-МОП технологии, задействован только один разряд  $PCON[7]$  -  $SMOD$ , определяющий скорость работы последовательного канала ("1" - двойная скорость).

В КМОП-БИС предусмотрены разряды управления режимом потребления мощности. При выключенном источнике питания  $+5V ..BE52/51$  может сохранять содержимое внутренней памяти данных (пространства  $DSEG$ ). При этом напряжение резервного источника подается на вход  $RST$ , а ток, потребляемый от него, очень мал. При восстановлении основ-

ного питания резервный источник должен оставаться включенным еще два машинных цикла, в течение которых выполняется сброс системы.

Для перехода в режим с малым потреблением используется флаги PD (PCON[1]) и IDL (PCON[0]). При установке IDL работа генератора не прекращается, а следовательно работают все внутренние устройства. При установке PD останавливается генератор.

Выход из IDL - по сбросу или прерыванию, выход из PD только по сбросу.

В состав регистра PCON включены так же два общецелевых флажка GF0 и GF1.

Все управляющие регистры, кроме PCON, совмещены с битовым пространством BSEG. При сбросе все управляющие регистры обнуляются, кроме нижеследующих:

IP(..BE51) - xxx0 0000; SBUF - xxxx xxxx;  
IP(..BE52) - xx00 0000; PCON (n-MOP)- 0xxx xxxx;  
IE(..BE51) - 0xx0 0000; PCON (KMOP) - 0xxx 0000.  
IE(..BE52) - 0x00 0000;

### 5.2.9. Система команд

Система команд включает 111 команд - однобайтовых (49), двухбайтовых (45) и трехбайтовых (17). Все команды выполняются за 1 или 2 МЦ (12 тактов CLK), за исключением команд умножения и деления MUL и DIV, для выполнения которых требуется 4 МЦ. Большинство двухбайтовых команд - одноцикловые, а все трехбайтовые двухцикловые. За один МЦ в ..BE52 можно вводить до двух байтов программного кода.

В микро-ЭВМ предусмотрена прямая (direct), косвенная (через R0, R1 или DPTR) и непосредственная (# data) адресация элементов DSEG и RSEG, прямая адресация элемента BSEG, косвенная (через R0, R1 или DPTR) адресация XSEG.

Все множество команд разбито на 5 групп: пересылки (28), логические (25), арифметические (24), передачи управления (17) и битового процессора (17).

В Таблица 5.4 – Таблица 5.7 приведены мнемкокоды команд, длина команды в МЦ, HEX-коды и описание команды. В таблицах использованы следующие обозначения:

A,B	- аккумулятор, дополнительный аккумулятор;
Rn	- регистр общего назначения (n = 0..7);
@Ri	- косвенный адрес через Ri (i = 0,1);
direct	- прямой адрес DSEG (8 бит);
#data	- непосредственный операнд длиной 8 бит;
#data16	- непосредственный операнд длиной 16 бит;
addr11	- адрес перехода длиной 11 бит;
addr16	- адрес перехода длиной 16 бит;

- rel - смещение на странице CSEG длиной 8 бит;  
bit - прямой адрес BSEG (номер бита).

В командах возможны следующие варианты формирования признаков

- [1] - P
- [2] - CY, P
- [3] - CY, AC, OV, P
- [4] - OV, P, CY ← 0
- [5] - CY

В Таблица 5.4 – Таблица 5.7 в описании команд указаны соответствующие обозначения. Отсутствие обозначения говорит о том, что данная команда не меняет значений признаков. Символ в фигурных скобках означает, что он относится ко всем командам блока.

Таблица 5.4. Команды пересылок

Мнемокод	МЦ	HEX-коды	Описание команды
MOV A, R <sub>n</sub>	1	E8 .. EF	A ← R <sub>n</sub>
MOV A, direct	1	E5	A ← (direct)
MOV A, @R <sub>i</sub>	1	E6 .. E7	A ← (R <sub>i</sub> )
MOV A, #data	1	74	A ← data
MOV R <sub>n</sub> , A	1	F8 .. FF	R <sub>n</sub> ← A
MOV R <sub>n</sub> , direct	2	A8 .. AF	R <sub>n</sub> ← (direct)
MOV R <sub>n</sub> , #data	1	78 .. 7F	R <sub>n</sub> ← data
MOV direct, A	1	F5	(direct) ← A
MOV direct, R <sub>n</sub>	2	88 .. 8F	(direct) ← R <sub>n</sub>
MOV direct, direct	2	85	(direct) ← (direct)
MOV direct, @R <sub>i</sub>	2	86 .. 87	(direct) ← (R <sub>i</sub> )
MOV direct, #data	2	75	(direct) ← data
MOV @R <sub>i</sub> , A	1	F6 .. F7	(R <sub>i</sub> ) ← A
MOV @R <sub>i</sub> , direct	2	A6 .. A7	(R <sub>i</sub> ) ← (direct)
MOV @R <sub>i</sub> , #data	1	76 .. 77	(R <sub>i</sub> ) ← data
MOV DPTR, #data16	2	90	DPTR ← data16
MOVC A, @A+DPTR	2	93	A ← CSEG(A+DPTR)
MOVC A, @A+PC	2	83	A ← CSEG(A+PC)
MOVX A, @R <sub>i</sub>	2	E2 .. E3	A ← XSEG(P2.R <sub>i</sub> )
MOVX A, @DPTR	2	E0	A ← XSEG(DPTR)
MOVX @R <sub>i</sub> , A	2	F2 .. F3	XSEG (P2.R <sub>i</sub> ) ← A
MOVX @DPTR, A	2	F0	XSEG (DPTR) ← A
PUSH direct	2	C0	+(SP) ← (direct)
POP direct	2	D0	(direct) ← (SP)-
XCH A, R <sub>n</sub>	1	C8 .. CF	A ↔ R <sub>n</sub>
XCH A, direct	1	C5	A ↔ (direct)
XCH A, @R <sub>i</sub>	1	C6 .. C7	A ↔ (R <sub>i</sub> )
XCHD A, @R <sub>i</sub>	1	D6 .. D7	A[3:0] ↔ (R <sub>i</sub> [3:0])

Таблица 5.5. Логические и арифметические команды

Мнемокод	МЦ	HEX-коды	Описание команды
ANL A, R <sub>n</sub>	1	58 .. 5F	$A \leftarrow A \& R_n$
ANL A, direct	1	55	$A \leftarrow A \& (\text{direct})$
ANL A, @R <sub>i</sub>	1	56 .. 57	$A \leftarrow A \& (R_i)$
ANL A, #data	1	54	$A \leftarrow A \& \text{data}$
ANL direct, A	1	52	$(\text{direct}) \leftarrow (\text{direct}) \& A$
ANL direct, #data	2	53	$(\text{direct}) \leftarrow (\text{direct}) \& \text{data}$
ORL A, R <sub>n</sub>	1	48 .. 4F	$A \leftarrow A \vee R_n$
ORL A, direct	1	45	$A \leftarrow A \vee (\text{direct})$
ORL A, @R <sub>i</sub>	1	46 .. 47	$A \leftarrow A \vee (R_i)$
ORL A, #data	1	44	$A \leftarrow A \vee \text{data}$
ORL direct, A	1	42	$(\text{direct}) \leftarrow (\text{direct}) \vee A$
ORL direct, #data	2	43	$(\text{direct}) \leftarrow (\text{direct}) \vee \text{data}$
XRL A, R <sub>n</sub>	1	68 .. 6F	$A \leftarrow A \oplus R_n$
XRL A, direct	1	65	$A \leftarrow A \oplus (\text{direct})$
XRL A, @R <sub>i</sub>	1	66 .. 67	$A \leftarrow A \oplus (R_i)$
XRL A, #data	1	64	$A \leftarrow A \oplus \text{data}$
XRL direct, A	1	62	$(\text{direct}) \leftarrow (\text{direct}) \oplus A$
XRL direct, #data	2	63	$(\text{direct}) \leftarrow (\text{direct}) \oplus \text{data}$
CLR A	1	E4	$A \leftarrow 0000\ 0000$ (Очистка акк.)
CLP A	1	F4	$A \leftarrow \bar{A}$ (Инверсия аккумулятора)
RL A	1	23	Сдвиг левый циклический
RLC A	1	33	Сдвиг левый с переносом
RR A	1	03	Сдвиг правый циклический
RRC A	1	13	Сдвиг правый с переносом
SWAP A	1	C4	$A[7:4] \leftrightarrow A[3:0]$
ADD A, R <sub>n</sub>	1	28 .. 2F	$A \leftarrow A + R_n$
ADD A, direct	1	25	$A \leftarrow A + (\text{direct})$
ADD A, @R <sub>i</sub>	1	26 .. 27	$A \leftarrow A + (R_i)$
ADD A, #data	1	24	$A \leftarrow A + \text{data}$
ADDC A, R <sub>n</sub>	1	38 .. 3F	$A \leftarrow A + R_n + CY$
ADDC A, direct	1	35	$A \leftarrow A + (\text{direct}) + CY$
ADDC A, @R <sub>i</sub>	1	36 .. 37	$A \leftarrow A + (R_i) + CY$
ADDC A, #data	1	34	$A \leftarrow A + \text{data} + CY$
SUBB A, R <sub>n</sub>	1	98 .. 9F	$A \leftarrow A - R_n - CY$
SUBB A, direct	1	95	$A \leftarrow A - (\text{direct}) - CY$
SUBB A, @R <sub>i</sub>	1	96 .. 97	$A \leftarrow A - (R_i) - CY$
SUBB A, #data	1	94	$A \leftarrow A - \text{data} - CY$
INC A	1	04	$A \leftarrow A + 1$
INC R <sub>n</sub>	1	08 .. 0F	$R_n \leftarrow R_n + 1$
INC direct	1	05	$(\text{direct}) \leftarrow (\text{direct}) + 1$
INC @R <sub>i</sub>	1	06 .. 07	$(R_i) \leftarrow (R_i) + 1$
INC DPTR	1	A3	$DPTR \leftarrow DPTR + 1$

DEC A	1	14	$A \leftarrow A - 1$
DEC R <sub>n</sub>	1	18 .. 1F	$R_n \leftarrow R_n - 1$
DEC direct	1	15	$(direct) \leftarrow (direct) - 1$
DEC @R <sub>i</sub>	1	16 .. 17	$(R_i) \leftarrow (R_i) - 1$
MUL AB	4	A4	$B.A \leftarrow A \times B$
DIV AB	4	84	$A.B \leftarrow A : B$
DA A	1	74	Десятичная коррекция сложения

Таблица 5.6. Команды передачи управления

Мнемокод	МЦ	HEX-коды	Описание команды
ACALL addr11	2	aaa10001	$+(SP) \leftarrow PC; PC[10:0] \leftarrow addr11$
LCALL addr16	2	12	$+(SP) \leftarrow PC; PC \leftarrow addr16$
RET	2	22	$PC \leftarrow (SP)-$
RETI	2	32	$PC \leftarrow (SP)-$ ; конец прерывания
AJMP addr11	2	aaa00001	$PC[10:0] \leftarrow addr11$
LJMP addr16	2	02	$PC \leftarrow addr16$
JMP @A + DPTR	2	73	$PC \leftarrow DPTR + A$
SJMP rel	2	80	$PC \leftarrow PC + rel$
JZ rel	2	60	if $A = 0$ then $PC \leftarrow PC + rel$
JNZ rel	2	70	if $A \neq 0$ then $PC \leftarrow PC + rel$
CJNE A, direct, rel	2	B5	if $A \neq (direct)$ then $PC \leftarrow PC + rel$
CJNE A, #data, rel	2	B4	if $A \neq data$ then $PC \leftarrow PC + rel$
CJNE R <sub>n</sub> , #data, rel	2	B8 .. BF	if $R_n \neq data$ then $PC \leftarrow PC + rel$
CJNE @R <sub>i</sub> , #data, rel	2	B6 .. B7	if $(R_i) \neq data$ then $PC \leftarrow PC + rel$
CJNZ R <sub>n</sub> , rel	2	D8 .. DF	$R_n \leftarrow R_n - 1$ if $R_n \neq 0$ then $PC \leftarrow PC + rel$
CJNE direct, rel	2	D5	$(direct) \leftarrow (direct) - 1$ if $(direct) \neq 0$ then $PC \leftarrow PC + rel$
NOP	1	00	

Таблица 5.7. Булевы команды

Мнемокод	МЦ	HEX-коды	Описание команды
MOV C, bit	1	A9	$CY \leftarrow BSEG(bit)$
MOV bit, C	1	92	$BSEG(bit) \leftarrow CY$
CLR C	1	C3	$CY \leftarrow 0$
CLR bit	1	C2	$BSEG(bit) \leftarrow 0$
SETB C	1	D3	$CY \leftarrow 1$
SETB bit	1	D2	$BSEG(bit) \leftarrow 1$
CPL C	1	B3	$CY \leftarrow CY \setminus$
CPL bit	1	B2	$BSEG(bit) \leftarrow BSEG(bit) \setminus$
ANL C, bit	2	82	$CY \leftarrow CY \& BSEG(bit)$
ANL C, bit\	2	B0	$CY \leftarrow CY \& BSEG(bit) \setminus$
ORL C, bit	2	72	$CY \leftarrow CY \vee BSEG(bit)$
ORL C, bit\	2	A0	$CY \leftarrow CY \vee BSEG(bit) \setminus$
JC rel	2	40	if $CY = 1$ then $PC \leftarrow PC + rel$

JNC rel	2	50	if CY = 0 then PC ← PC + rel
JB bit, rel	2	20	if BSEG(bit) = 1 then PC ← PC + rel
JNB bit, rel	2	30	if BSEG(bit) = 0 then PC ← PC + rel
JBC bit, rel	2	10	if BSEG(bit) = 1 then PC ← PC + rel и BSEG(bit) ← 0

### 5.3. Однокристалльная микроЭВМ MC68HC11E9

Изготовленный по КМОП-технологии с высокоплотной структурой восьмиразрядный микроконтроллер **MC68HC11E9** предназначается для выполнения широкого круга прикладных задач. Для достижения номинальной частоты шины 2 МГц были использованы новые технологии. Кроме того, полностью статическая схемотехника позволяет работать на очень низких частотах, что позволяет при необходимости уменьшать потребление энергии.

ОЭВМ имеет ряд особенностей в аппаратном и программном обеспечении, которые перечислены ниже:

*Особенности аппаратного обеспечения:*

- 12 Кбайт ПЗУ;
- 512 байт ЭСППЗУ;
- 512 байт ОЗУ;
- 16-разрядный таймер с расширенными функциями:
  - ◊ 4-разрядный предделитель частоты.
  - ◊ Три функции входной фиксации и пять - выходного сравнения или
  - ◊ Четыре функции входной фиксации и четыре - выходного сравнения.
- восьмиразрядный счетчик внешних импульсов;
- последовательный асинхронный интерфейс связи расширенного формата NRZ (SCI);
- последовательный периферийный интерфейс (SPI);
- восьмиканальный, восьмиразрядный АЦП;
- система прерываний реального времени;
- система слежения за правильностью работы ОЭВМ (COP-Watchdog);
- 52-выводной квадратный пластиковый корпус.

*Особенности программного обеспечения:*

- система команд представляет собой надмножество системы команд семейства M6800;

- операции дробного и целочисленного 16x16 деления;
- операции манипуляцией отдельными битами данных;
- режимы малого потребления энергии (WAIT и STOP);

В Таблица 8 представлены основные микроконтроллеры семейства MC68HC11. Они отличаются главным образом по типу и объему встроенной памяти. Каждый представитель семейства MC68HC11 имеет вариант исполнения x8, x9, x1, или x0. Эти варианты имеют одинаковую конструкцию. Регистр конфигурации встраивается как ячейка ЭСППЗУ и используется для запрещения ОЗУ в варианте x1. В варианте x0 ОЗУ и ЭСППЗУ отсутствует.

Таблица 5.8

Номер чипа	ПЗУ	ЭСППЗУ	ОЗУ	CONFIG <sup>2)</sup>	Комментарий
MC68HC11A8	8 Кбайт	512 байт	256 байт	0F	базовый представитель семейства
MC68HC11A1	0	512 байт	256 байт	0D	как 'A8, но без ПЗУ
MC68HC11A0	0	0	256 байт	0C	как 'A8, но без ПЗУ и ЭСППЗУ
MC68HC11E9	12 Кбайт	512 байт	512 байт	0F	4 функции входного сравнения, большее ОЗУ и 12 Кбайт ПЗУ
MC68HC11E1	0	512 байт	512 байт	0D	как 'E9, но без ПЗУ
MC68HC11E0	0	0	512 байт	0C	как 'E9, но без ПЗУ и ЭСППЗУ
MC68HC11E2	0	2 Кбайта <sup>1)</sup>	256 байт	FF	без ПЗУ, для работы в расширенном режиме
MC68HC11D3	4 Кбайта	0	192 байта	-	упрощенная версия: нет АЦП и меньшая память

<sup>1)</sup> Эти 2 Кбайта ЭСППЗУ можно переместить на границу каждых 4 Кбайт памяти. Перемещение управляется четырьмя битами регистра CONFIG.

<sup>2)</sup> Величина в регистре CONFIG, которая программируется фирмой Motorola при изготовлении.

### Программная модель MC68HC11E9

Программная модель (машина пользователя) включает в себя семь регистров общего назначения (Рис.5.26), 64 системных и служебных регистра и различные блоки памяти:

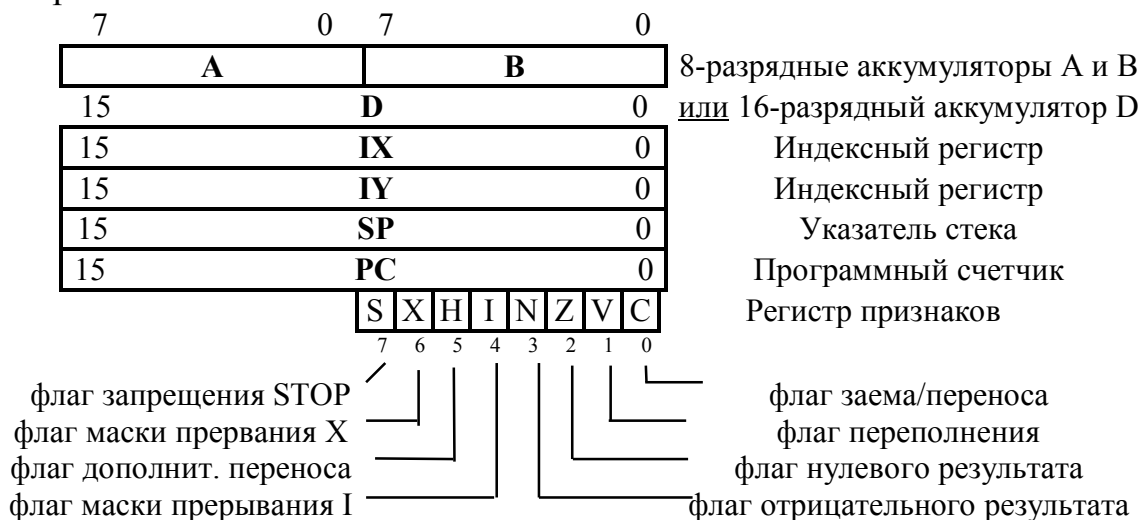


Рис.5.26. Программно-доступные регистры процессора

Структура памяти микроЭВМ будет описываться ниже. Память микроЭВМ и регистры внешних устройств составляют единое адресное пространство объемом 64К байт, в котором достаточно произвольно могут располагаться различные блоки внутренней и внешней памяти и регистров.

### ***5.3.1. Внутренняя структура и назначение выводов***

Внутренняя структура микроконтроллера МС68НС11Е9 показана на рис 5.27. На кристалле микроЭВМ располагается 8-разрядный центральный процессор CPU11, 12 Кбайт однократно программируемого ПЗУ, 512 байт ППЗУ с электрическим стиранием, статическое ОЗУ объемом 512 байт, два устройства последовательных интерфейсов, пять параллельных портов ввода/вывода, 8-разрядный 8-канальный АЦП с входным мультиплексором, блок таймеров и прерываний, тактовый генератор и блок управления режимами работы.

Внутренний генератор может работать как в режиме автогенерации (при подключении кварцевого резонатора), так и от внешнего генератора.

Назначение линий портов **A**, **D** и **E** не зависит от режима работы. Порт **B** является портом вывода общего назначения в однокристалльном режиме работы и является источником старшего байта адреса в расширенном. Порт **C** является портом ввода/вывода общего назначения в однокристалльном режиме работы. В расширенном режиме работы порт **C** используется как мультиплексированная шина адреса/данных.

В Таблица 5.5.9 приведены краткие характеристики и описание выводов ОМЭВМ.



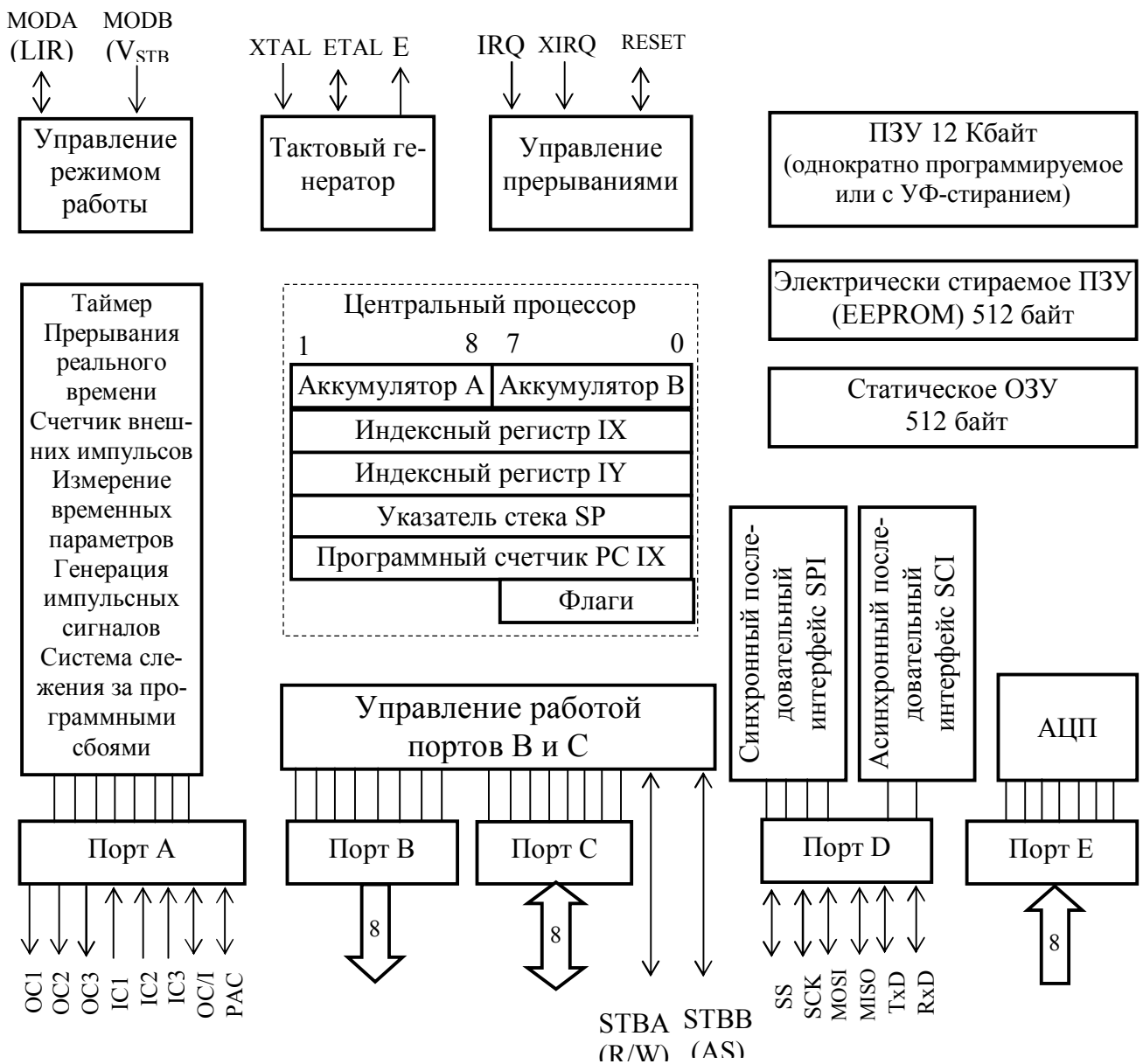


Рис.5.27. Внутренняя структура ОЭВМ MC68HC11E9

Таблица 5.9

Сигнал	Тип	Назначение
V <sub>DD</sub> V <sub>SS</sub>	питание “земля”	+5 В 0 В
RESET\	вх./вых.	Вход начальной установки, выход “системного сброса” (открытый сток) после сбоя системы
XTAL ETAL		Для подключения кварцевого резонатора (4×E) Внешнюю частоту 4×E подают на ETAL
E	вых.	Тактовая частота E = ETAL/4
IRQ\	вх.	Запрос асинхронного прерывания
XIRQ\	вх.	Запрос немаскируемого прерывания
MODA/LIR	вх./вых.	MODA, MODB - в момент действия RESET входы выбора режима работы. LIR - выход, отмечающий начало выполнения
MODB/V <sub>STBY</sub>	вх./вх.	командного цикла; V <sub>STBY</sub> - автономное питание ОЗУ
V <sub>RH</sub> , V <sub>RL</sub>		Опорные напряжения АЦП
STRB/ R/W\	вых.	Строб порта В <sup>1)</sup> ; определяет направление передачи данных по внешней шине <sup>2)</sup> (высокий уровень - чтение с шины)
STRA/AS	вх./вых.	Вход stroba записи в А <sup>1)</sup> ; строб адреса на мультиплексированной шине адреса/данных порта С <sup>2)</sup>

**Порт А** можно настроить для выполнения четырех функций входной фиксации (IC1, IC2, IC3 и IC4) и трех функций выходного сравнения (OC2, OC3 и OC4) либо трех функций входной фиксации (IC1, IC2, и IC3) и четырех функций выходного сравнения (OC2, OC3, OC4 и OC5), а также организовать вход для счетчика импульсов (импульсного аккумулятора PAI) или выход пятой функции выходного сравнения (OC1). Любые выводы порта А, которые не используются как линии специального назначения, можно использовать как линии ввода/вывода общего назначения.

**Порт В.** При работе в *однокристалльном режиме* все линии порта В являются линиями ввода/вывода общего назначения. При чтении регистра данных этого порта считываются уровни на входах соответствующих выходных буферов. Порт В может также быть использован в режиме простого стробируемого вывода, где каждый раз при записи в порт В на выводе STRB появляется стробирующий импульс. В *расширенном режиме* все линии порта В действуют как старшие восемь линий адреса. В течение каждого цикла микроконтроллера разряды адреса с 8 по 15 выводятся на линии PB0-PB7 соответственно.

**Порт С.** В *однокристалльном режиме* работы все линии порта С являются линиями ввода/вывода общего назначения. Также значение, поступившее в порт, может быть зафиксировано по поступлении сигнала STRA. Порт С можно использовать в режиме параллельного ввода/вывода с полным квитированием установления связи, где ввод STRA и вывод STRB яв-

ляются управляющими сигналами обмена. При работе в *расширенном режиме* все выходы порта С настраиваются как мультиплексированные линии адреса/данных. В течение той фазы каждого цикла микроконтроллера, когда производится адресация, биты [7:0] адреса выводятся на линии PC7-PC0. В течение фазы выбора данных линии порта являются двунаправленными линиями данных, причем направление передачи данных индицируется сигналом R/W/.

**Порт D.** Выводы [0:5] порта D можно использовать как линии ввода/вывода общего назначения, либо как линии обмена и управления для последовательного интерфейса связи (SCI) и последовательного периферийного интерфейса (SPI), если работа этих подсистем разрешена. Вывод PD0 является входом приемника данных (RxD) последовательного интерфейса связи (SCI). Вывод PD1 является выходом передатчика данных (TxD) SCI. Выводы PD2-PD5 предназначаются для SPI. PD2 является входом ведущего и выходом ведомого устройства (MISO). PD3 является выходом ведущего и входом ведомого устройства (MOSI). PD4 - это сигнал синхронизации (SCK) для SPI и PD5 - это вход выбора ведомого устройства (SS/).

**Порт E.** Линии порта E можно использовать либо как входы общего назначения, либо как входные каналы системы АЦП. Чтение порта E может вызвать помехи и повлиять на точность результата при работе системы АЦП. Если вам необходима высокая точность, избегайте чтения порта E во время выполнения преобразований системой АЦП.

### 5.3.2. Режимы работы

Всего существует четыре режима работы микроконтроллера MC68HC11E9: однокристалльный, расширенный, а также специальные режимы загрузки и теста. В Таблица 5.10 показано, какие логические уровни следует подать на выходы MODA, MODB в момент действия сигнала RESET для выбора соответствующего режима

Таблица 5.10

MODB	MODA	Выбираемый режим
1	0	Однокристалльный режим
1	1	Расширенный режим
0	0	Режим загрузки
0	1	Режим теста

В *однокристалльном режиме* микроконтроллер работает без использования внешних шин адреса и данных. Порты А и В и выходы STRA и STRB функционируют как входы/выводы общего назначения и сигналы квитирования.

В *расширенном режиме* работы микроконтроллер MC68HC11E9 способен адресовать пространство объемом до 64 Кбайта. Все это адресное

пространство включает в себя те же участки памяти, что и в однокристалльном режиме, плюс внешние память и периферию. Адресация производится с помощью портов В и С, а также управляющих сигналов AS и R/W/. Адрес и сигналы R/W/ и AS действуют на протяжении всех циклов шин, включая доступ и ко внутренним участкам памяти.

*Специальный режим загрузки* позволяет произвести загрузку программы во встроенное ОЗУ по последовательному асинхронному интерфейсу и запустить программу на выполнение. Программа загрузчика находится в 192-байтовом участке ПЗУ в ячейках BF40-BFFF. Этот участок ПЗУ становится доступным только в специальном режиме загрузки. Программа загрузчика использует последовательный интерфейс связи SCI для загрузки во внутреннее ОЗУ программы длиной вплоть до 512 байт по адресам ОЗУ 0000-01FF. После загрузки последнего байта управление автоматически передается по адресу 0000.

В специальном режиме загрузки вектора прерываний отображаются на ОЗУ. Это позволяет использовать эти прерывания с помощью таблицы переходов.

*Режим теста* - это специальный режим работы, предназначенный для производственного тестирования микроконтроллера при изготовлении. Этот режим очень похож на расширенный режим работы. В режиме теста вектора прерываний выбираются с адресов BFC0-BFFF, а не с FFC0-FFFF. В этом режиме нет защиты регистров TMSK2, OPTION, BPROT и INIT, так что эти регистры можно перезаписывать неоднократно в любой момент времени. Также становится доступным регистр TEST1, который позволяет вызвать некоторые функции заводского тестирования.

### **5.3.3. Карта памяти**

Карта памяти ОЭВМ MC68HC11E9 определяет расположение в адресном пространстве 512 байт внутреннего ОЗУ, 64 регистров внешних устройств, 512 байт ЭСППЗУ и 12К байт внутреннего ПЗУ. Кроме того, в определенных режимах в адресном пространстве становится возможным подключать внешнюю память или обращаться к «теневому» ПЗУ загрузчика.

По умолчанию указанные элементы адресного пространства располагаются, как показано на Рис.5.28.

Для каждого режима работы существуют небольшие отличия в конфигурации памяти.

В однокристалльном режиме работы микроконтроллер не осуществляет адресацию внешней памяти.

В расширенном режиме карта памяти аналогична однокристалльному режиму, но участки памяти, расположенные между затененными областями являются памятью, адресуемой на внешние устройства.

Расположение памяти в специальном режиме загрузки похоже на однокристалльный режим, за исключением того, что в данном режиме доступная область программы загрузки располагается по адресам BF40-BFFF. В режиме загрузки все вектора прерываний адресуются на участок памяти BFC0-BFFF. Эти вектора вместе с 192-байтным участком памяти используются программой загрузки.

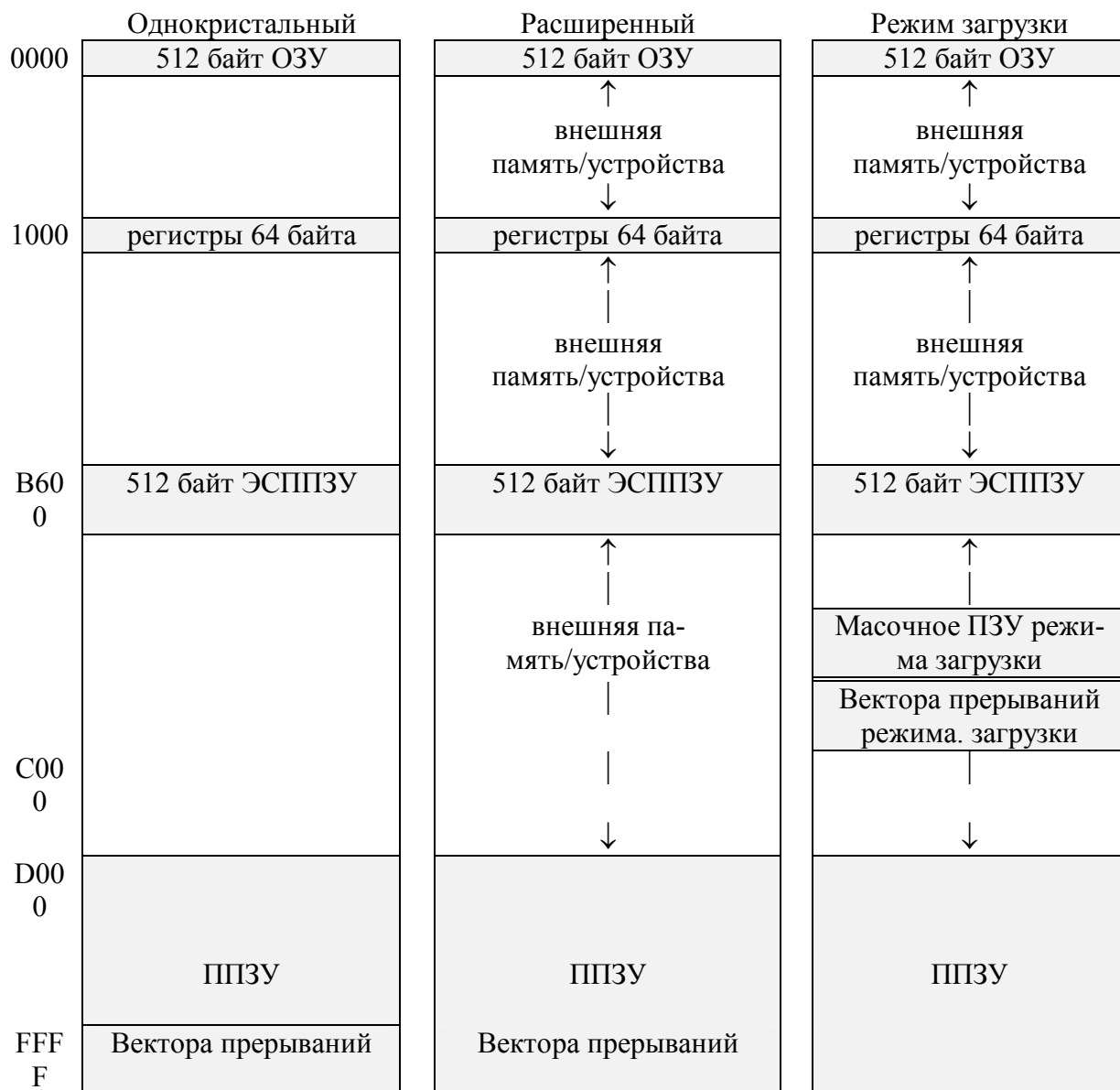


Рис.5.28. Карта памяти

В специальном режиме теста карта памяти аналогична расширенному режиму, за исключением того, что все вектора прерываний и RESET расположены по адресам BFC0-BFFF.

В специальном режиме теста карта памяти аналогична расширенному режиму, за исключением того, что все вектора прерываний и RESET расположены по адресам BFC0-BFFF.

При необходимости пользователь может переместить в адресном пространстве блок регистров и внутреннее ОЗУ, а так же отключить внутреннее ПЗУ и/или ЭСПЗУ.

Блок из 64 внутренних регистров может быть перемещен в любое место адресного пространства на границу одной из 4 Кбайтных областей, путем перепрограммирования регистра INIT

Блок из 64 внутренних регистров, которые используются для контроля за работой процессора может быть перемещен в любое место адресного пространства на границу одной из 4 Кбайтных областей, путем перепрограммирования регистра INIT (Рис.5.19).

Регистр INIT - это специальный 8-разрядный регистр, который может быть использован при инициализации для изменения мест расположения ОЗУ и блока внутренних регистров. Запись в этот регистр разрешена только в первые 64 цикла сигнала синхронизации E после RESET; далее возможно только чтение этого регистра.

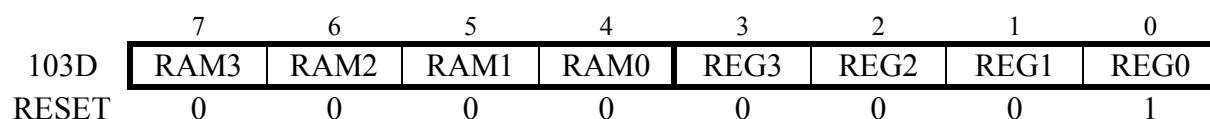


Рис.5.19. Формат регистра INIT

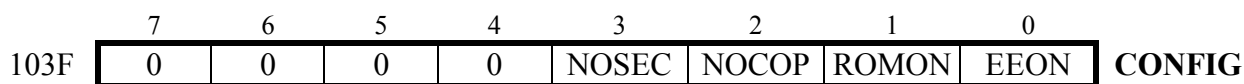
Начальный адрес ОЗУ по умолчанию определяется как 0000, а начальный адрес блока внутренних регистров -1000. Четыре старших бита регистра INIT определяют начальный адрес 512 байт ОЗУ, а четыре младших - блока внутренних регистров. Каждые четыре бита совпадают со старшими четырьмя битами полного 16-разрядного адреса начала соответствующего участка памяти. Вследствие этого отображение возможно лишь на границу 4 Кбайт.

Следует отметить, что если ОЗУ размещать по адресам D000, E000 или F000, то возникнет конфликт с внутренним ПЗУ (за исключением случая, когда бит ROMON в регистре конфигурации (CONFIG) сброшен в нуль). В случае конфликта ОЗУ будет иметь более высокий приоритет и соответствующая область ПЗУ станет недоступной. В случае подобного конфликта блока внутренних регистров с каким-либо участком ОЗУ или ПЗУ блок регистров будет иметь более высокий приоритет. Аналогично, при совпадении адресов внутренних и внешних областей конфликта не возникает, т.к. данные от внешних устройств, в этом случае, не подаются на внутреннюю шину данных.

В случае конфликта совмещения адреса от внешнего устройства и внутреннего участка памяти, приоритет будет иметь внутренняя память. При чтении данных по любому такому адресу, данные на входах порта C игнорируются и таким образом не приводят к нежелательным конфликтам. При записи по таким адресам данные выводятся также и на линии порта C.

Во время записи во внутреннюю память ни одно из внешних устройств не должно выставлять информацию на линии порта С; однако подобного конфликта обычно не возникает, так как внешняя схема дешифрации адреса должна использовать сигнал R/W/ для определения направления передачи данных. Сигналы R/W/, AS, адрес и сигналы записи данных действительны для всех видов доступа, включая доступ ко внутренним участкам памяти и регистрам. Следует заметить, что некоторые из 64 регистров не используются и при чтении будут возвращать данные с внутренней шины данных (неактивной на момент их чтения), а не какого-либо другого источника, который может случайно располагаться по тому же самому адресу.

Отключение областей внутренних ПЗУ и ЭСППЗУ осуществляется с помощью перепрограммирования регистра CONFIG, реализованного как ячейка ЭСППЗУ.



Биты EEON и ROMON, будучи сброшенными, запрещают работу внутренних ЭСППЗУ и ПЗУ соответственно; при этом соответствующее адресное пространство можно использовать под внешнюю память. При NOCOP = 0 разрешается работа системы контроля правильности работы программы.

Бит NOSEC предусматривается в системе только в случае реализации специальных требований защиты. Будучи сброшенным, он предотвращает выбор специальных режимов работы контроллера.

Внутреннее ПЗУ занимает старшие 12 Кбайт адресного пространства. ПЗУ запрещено, когда бит ROMON регистра конфигурации CONFIG сброшен. В однокристалльном режиме работы внутреннее ПЗУ доступно независимо от состояния бита ROMON. В MC68HC11E9 есть также маскируемая область размером 192 байта, в которой размещается программа-загрузчик, управляющая работой ОЗВМ в специальном режиме загрузки и доступная сразу после начальной установки в этом режиме.

Внутреннее ОЗУ объемом 512 байт по умолчанию, после RESET устанавливается по адресам 0000-01FF, но может быть переадресовано в течение инициализации. ОЗУ состоит из статических ячеек памяти. Информация в ячейках сохраняется в течение режимов WAIT и STOP. Содержимое ОЗУ сохраняется также при подаче поддерживающего напряжения на вывод MODB/VSTBY. Когда используется дополнительный источник питания напряжение с вывода VDD можно убрать, однако уровень сигнала на входе RESET\ должен стать низким до спада VDD и оставаться низким до тех пор, пока питание на вывод VDD снова не будет подано.

### 5.3.4. ЭСППЗУ и его программирование

512 байт ЭСППЗУ размещаются по адресам с В600 по В7FF, однако если, бит EEON регистра CONFIG сброшен в нуль, то ЭСППЗУ запрещено. ЭСППЗУ может работать в трех режимах:

- чтение
- стирание
- запись.

Время доступа к ЭСППЗУ при чтении не отличается от времени доступа к ПЗУ.

Время записи и стирания ячейки ЭСППЗУ существенно превышает время чтения

Записи в ячейку ЭСППЗУ обычно должно предшествовать стирание, причем возможно запустить процедуру стирания

- одного байта;
- выровненного блока 16 байт;
- всего накопителя ЭСППЗУ.

Содержимое стертого байта ЭСППЗУ - FF. При программировании производится сброс определенных бит в нули. Если какой-нибудь бит следует перепрограммировать из нуля в единицу, то перед программированием отдельной операцией должен быть стерт весь байт.

Программирование и стирание ЭСППЗУ осуществляется с помощью встроенного генератора подкачки заряда. Если тактовая частота E падает ниже 2 МГц, эффективность работы генератора уменьшается, в то время как увеличивается время, требуемое для программирования или стирания байта. Рекомендуемое время программирования 10 мс при частоте 2 МГц и должна быть увеличена до 20 мс при частоте от 1 до 2 МГц. При частоте ниже 1 МГц генератор должен быть переключен с системной синхронизации на встроенный R-C генератор путем установки бита CSEL регистра OPTION. После переключения следует выждать период в 10 мс для стабилизации работы генератора. Следует заметить, что бит CSEL также управляет синхронизацией для встроенного АЦП.

На режим программирования ЭСППЗУ оказывают влияние два регистра: регистр управления ЭСППЗУ - PPROG и регистр защиты ЭСППЗУ - BPROG. На Рис.5.30 показаны форматы этих регистров.

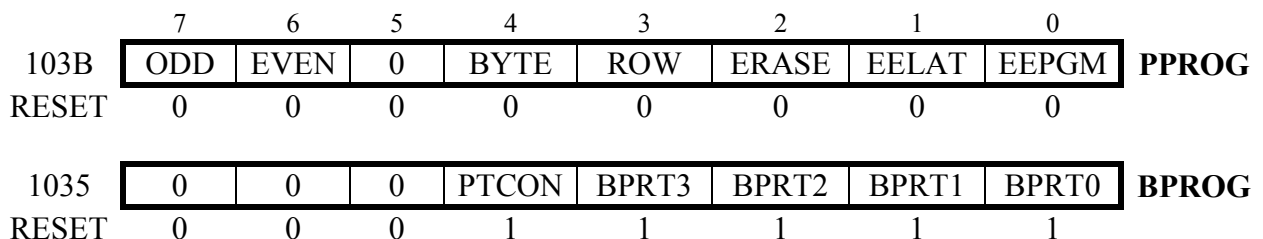


Рис.5.30. Форматы регистров управления ЭСППЗУ



Биты *ODD* и *EVEN* регистра **PPROG** работают только в режиме теста. *BYTE* определяет режим стирания: 1 - стирает один байт, 0 - стирает строку (16 байт) - при *ROW* = 0 или все ЭСППЗУ - при *ROW* = 1, причем при *BYTE* = 1 значение *ROW* безразлично. *ERASE* определяет режим стирания (0 - чтение или программирование, 1 - стирание), а *EELAT* - режим работы (0 - чтение ЭСППЗУ, 1 - программирование или стирание ЭСППЗУ). Бит *EERGM* выключает (при 1) программирующее напряжение.

ERASE	EELAT	Режим ЭСППЗУ	BYTE	ROW	Объект стирания
0	0	Чтение	0	0	Строка 16 байт
0	1	Запись	0	1	Все ЭСППЗУ
1	1	Стирание	1	x	Один байт

Бит *PTCON* регистра **BPROG**, будучи установленным в 1, запрещает программирование /стирание регистра *COFIG*, а биты *BPRT[3:0]* аналогично защищают блоки ЭСППЗУ.

Для операции **чтения** из ЭСППЗУ требуется, чтобы бит *EELAT* был сброшен. Само чтение не отличается от обычного обращения к ОЗУ или ПЗУ.

Записи информации в ЭСППЗУ должно в общем случае предшествовать стирание байта, блока или всего ЭСППЗУ. Для **стирания** достаточно установить в **PPROG** значения разрядов *BYTE* и *ROW* в соответствии с требуемым объектом стирания, а разряд *EELAT* - в "1", выполнить команду записи любых данных по (любому) адресу стираемого объекта и на 10 мс включить программирующее напряжение установкой бита *EERGM*.

В режиме **записи** (программирования) ЭСППЗУ биты *BYTE* и *ROW* не используются. Для записи байта следует установить разрешение программирования (*EELAT* =1), выполнить команду записи нужного байта по выбранному адресу и на 10 мс включить программирующее напряжение установкой бита *EERGM*. Недопустимо устанавливать оба бита - *EELAT* и *EERGM* - одной командой!

### 5.3.5. Параллельный ввод/вывод

Микроконтроллер **MC68HC11E9** имеет 40 линий ввода/вывода объединенных в пять 8-разрядных портов А, В, С, D, Е. Все эти линии могут выполнять множество функций, зависящих от режима работы и данных, содержащихся в регистрах управления.

Регистры, имеющие отношение к параллельному вводу/выводу, представлены на Рис.5.32.

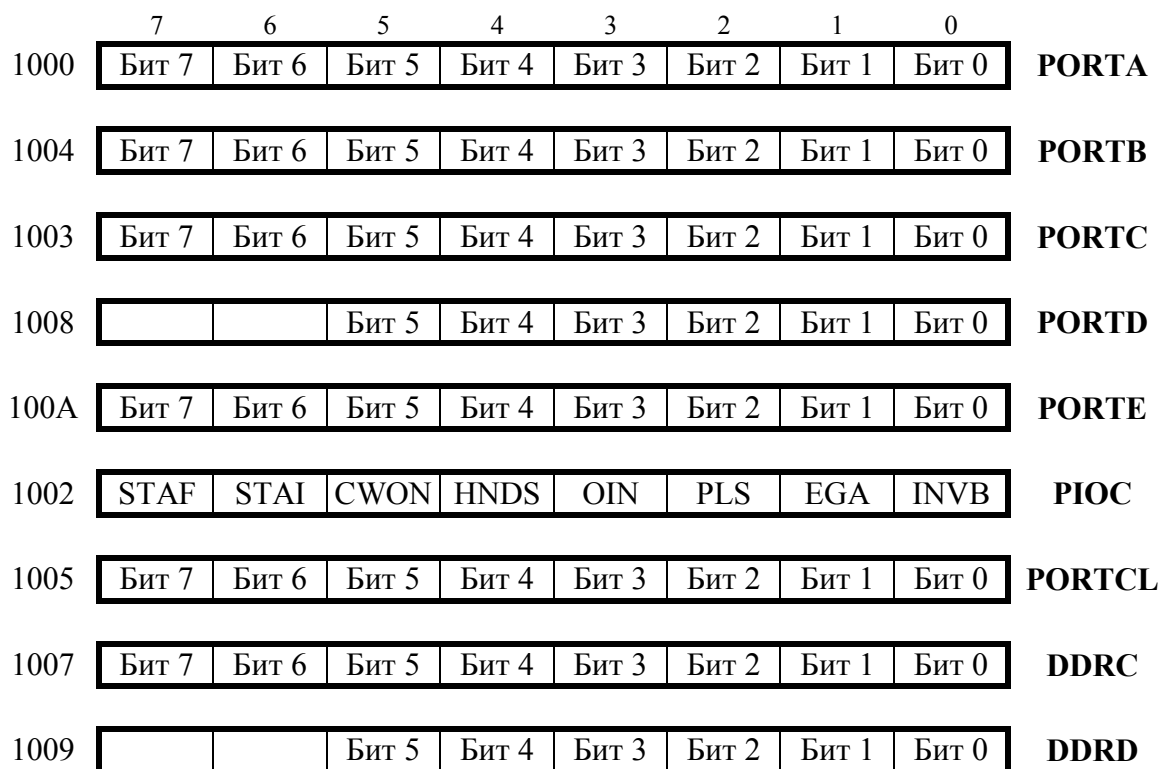


Рис.5.32. Регистры подсистемы параллельного ввода/вывода

Порты С и D используются как порты общего назначения и направление передачи данных по ним управляется регистрами направления передачи данных (DDRC и DDRD соответственно). Порты А, В и Е, имеют фиксированное направление пересылки данных и, следовательно, не имеют регистров направления передачи данных. В расширенном режиме порты В и С используются для организации связи с внешней памятью: PORTB - шина старшего байта адреса, PORTC - мультиплексированная шина младшего байта адреса/данных. Остальные порты, помимо функций обычного цифрового обмена, могут выполнять и специальные функции: линии PORTA используются подсистемой контроля времени, PORTD - последовательными каналами, а PORTE - подсистемой аналого-цифрового преобразования.

Порты В и С могут работать не только в синхронном, но и в асинхронном режиме, причем порт С - в режиме “с полным квитиowaniem” - двунаправленном асинхронном режиме.

#### *Синхронный параллельный обмен*

В однокристалльном режиме все порты могут использоваться как порты общего назначения и работать на обмен в синхронном режиме.

Направление передачи для каждого разряда портов С и D независимо программируется значением соответствующего разряда регистров DDRC и DDRD (“0” - ввод данных, “1” - вывод).

При настройке линии на вывод данные, записанные в триггере, подаются на выход. При попытке прочитать состояние линии, настроенной на

вывод, возвращается величина присутствующая на входе выходного буфера линии.

При настройке линии на ввод, соответствующий вывод устанавливается в высокоимпедансное состояние. Попытка произвести запись в такую линию не изменяют ее состояние, но записываемые данные сохраняются в выходном триггере линии.

При RESET биты регистра управления направлением передачи данных сбрасываются, то есть все линии настраиваются на ввод данных.

Линии портов А, В и Е (исключая биты 3 и 7 порта А) имеют фиксированное направление передачи данных. При использовании порта В как порта общего назначения он настраивается исключительно на вывод и чтение его линий возвращает состояние на входах соответствующих выходных буферов.

Линии порта Е соответствуют восьми каналам встроенного АЦП, но также могут быть использованы для обычного дискретного ввода. Запись в PORTE не дает никакого эффекта.

В расширенном режиме порты В и С используются только для связи с внешней памятью, линия AS/STRA стробирует передачу адреса на шину адреса/данных (порт С), а R/W/STRB определяет направление передачи данных.

#### *Асинхронный параллельный обмен*

В асинхронных режимах могут работать только порты В и С, причем функции квитирования доступны только в однокристалльном режиме. Различают несколько простых режимов с квитированием, а также режимы с полным квитированием (одно- и двунаправленный обмен). Управление режимами работы портов сосредоточено главным образом в регистре PIOC (Parallel Input/Output Control).

	7	6	5	4	3	2	1	0	
1002	STAF	STAI	CWON	HNDS	OIN	PLS	EGA	INVB	PIOC
RESET	0	0	0	0	0	U	1	1	

Рис.5.32. Формат регистра управления вводом/выводом

◆ **STAF**-флаг статуса прерывания строба А. Этот бит устанавливается при появлении выбранного сигнала на линии STRA. Процесс его сброса зависит от состояния битов HNDS и OIN. В режиме просто стробирования или режиме ввода с квитированием флаг сбрасывается при чтении регистра PIOC (флаг STAF должен быть установлен) с последующим чтением регистра PORTCL. В режиме вывода с квитированием флаг сбрасывается последовательно чтением регистра PIOC и записью в регистр PORTCL.

◆ **STAI**-маска разрешения прерывания по стробу А. При установленном бите I регистра признаков и бите STAI регистра PIOC установка флага STAF будет вызывать запрос прерывания.

◆ **CWON**-вид выводов порта C; действует сразу на все выводы: 0 = обычные КМОП- выводы; 1 = выводы с открытым стоком.

◆ **HNDS**-режим квитирования: 0 = режим простого стробирования; 1 = режим ввода/вывода с полным квитированием. При сброшенном бите строб А работает как входной строб для записи данных в регистр PORTCL, а строб А - как строб вывода, который выдается после записи в порт В. Если HNDS = 1 порт С работает в режиме полного квитирования: режимы ввода/вывода выбираются битом OIN.

◆ **OIN**-направление передачи в режиме с квитированием: 0 = режим вывода; 1 = режим ввода. Этот бит не используется при HNDS = 0.

◆ **PLS**-импульсный режим/режим блокировки: 0 = статический режим; 1 = импульсный режим. При выборе статического режима однажды установленный строб В остается активным до тех пор, пока не будет обнаружен выбранный фронт строба А. В импульсном режиме строб В устанавливается на время двух циклов тактовой частоты Е. Не используется при HNDS = 0

◆ **EGA**-активный фронт строба А: 0 = отрицательный фронт; в режиме вывода по сигналам квитирования при низком уровне STRA линии порта С согласуются с регистром направления, но при высоком уровне на STRA все линии порта С настраиваются на вывод; 1 = положительный фронт; в режиме вывода по сигналам квитирования при высоком уровне STRA линии порта С согласуются с регистром направления, но при низком уровне на STRA все линии порта С настраиваются на вывод;

◆ **INVT**-активный уровень строба В: 0 = активный уровень - логический ноль; 1 = активный уровень - логическая единица.

Биты регистра PIOC доступны как для чтения, так и для записи, исключая бит 7, который можно только читать.

#### *Простой стробируемый ввод/вывод*

Режим простого стробируемого параллельного ввода/вывода устанавливается при сброшенном бите HNDS регистра PIOC. Порт С настраивается на ввод со стробирующим сигналом на линии STRA, при этом запись информации во входные триггер производится по фронту сигнала на STRA. Аналогично порт В стробируется на вывод линией STRB. Логический уровень на выходе STRB выбирается битом INVВ регистра PIOC.

#### *Стробируемый ввод в порт С*

В этом режиме существует два адреса, по которым возможно чтение данных из порта С: регистр данных порта С PORTC (1003h) и регистр - защелка PORTCL (1005h). Регистр направления передачи данных управляет направлением передачи всех линий порта С. Даже при выборе режима стробируемого ввода любые линии порта С все еще могут использоваться для ввода/вывода общего назначения.

Линия STRA используется как вход по фронту сигнала и бит выбора фронта стробирования порта А (EGA) регистра ПИОС определяет, положительный или отрицательный фронт используется для стробирования. При обнаружении на линии STRA выбранного фронта текущее состояние битов порта С копируется в регистр PORTCL и в регистре ПИОС устанавливается флаг стробирования А (STAF). Если в регистре ПИОС установлен бит разрешения прерываний по стробу А, производится запрос внутреннего прерывания. При чтении регистра ПИОС (с установленным флагом STAF) и последующим чтением регистра PORTCL флаг STAF автоматически сбрасывается. В регистр PORTCL данные помещаются независимо от того, был ли флаг STAF ранее сброшен или нет.

#### *Стробируемый вывод из порта В*

В этом режиме вывод STRB является выходом сигнала стробирования, на который при каждой записи в порт В в течение двух периодов тактовой частоты Е подается активный уровень. Бит INVВ регистра ПИОС управляет полярностью активного уровня на линии STRB.

#### *Ввод/вывод с полным квитированием установления связи*

Этот режим параллельного ввода/вывода использует порт С и линии STRA и STRB. Существуют два основных режима (ввод и вывод) и дополнительно к ним вариант режима вывода с квитированием установления связи, который допускает работу порта С в трех состояниях. Во всех режимах с полным квитированием установления связи вывод STRA является входом, работающим по фронту, а STRB - выходом.

Когда установлен протокол квитирования также сохраняется возможность работы порта С как порта ввода/вывода общего назначения. Здесь имеются две особенности взаимодействия режима вывода с полным квитированием установления связи и ввода общего назначения.

Первая: в режиме вывода с полным квитированием установления связи линии порта С являются выходами всякий раз, когда появляется активный уровень на STRA независимо от состояния битов регистра направления передачи данных. Здесь потенциально возможен конфликт с любым внешним устройством, пытающимся управлять портом С, за исключением тех внешних устройств, которые имеют выходы с открытым стоком.

Вторая: величина, возвращаемая при чтении порта С, является состоянием внутреннего триггера вывода порта С независимо от состояния бит регистра направления, так что данные, записанные для вывода с полным квитированием установления связи могут быть прочитаны даже если выходы находятся в третьем (высокоимпедансном) состоянии.

#### *Режима ввода с полным квитированием установления связи*

В этом протоколе регистр С является портом ввода, STRA - линия стробирования ввода данных в порт С из внешней системы и линия STRB - линия вывода сигнала готовности, управляемая логикой микроконтроллера.

При определении сигнала готовности, внешнее устройство помещает данные на линии порта C, после чего подает сигнал на линию STRA. По фронту на линии STRA данные на входе порта C копируются в регистр PORTCL, устанавливается флаг STAF (не обязательно вызывая прерывание) и сбрасывается сигнал готовности на линии STRB. Сброс сигнала готовности автоматически препятствует стробированию внешними устройствами новых данных в порт C. Чтение порта C (независимо от состояния флага STAF) устанавливает сигнал готовности на линии STRB, сигнализируя о том, что в порт C можно записывать новые данные.

Линия STRB может быть настроена (битом управления PLS) как импульсный выход (импульсный режим) или статический выход (статический режим).

В регистре направления DDRC должны быть сброшены биты, соответствующие разрядам порта C, которые используются для ввода с квити́рованием. Однако, некоторые линии порта C можно использовать в качестве входов с квити́рованием и в тоже время некоторые линии могут использоваться как статические входы, а некоторые - как статические выходы. Протокол режима ввода не оказывает никакого влияния на использование линий порта C в качестве статических входов или статических выходов. Чтение регистра данных PORTC всегда возвращает логические уровни на выводах порта C (для линий, настроенных на ввод). Запись как в регистр PORTC, так и в регистр PORTCL посылает информацию в тот же самый выходной регистр порта C, не влияя на стробирование ввода с квити́рованием.

#### *Режима вывода с полным квити́рованием установления связи*

В этом протоколе порт C настроен на вывод, на линию STRB выводится сигнал готовности и линия STRA является входной линией подтверждения (работающая по фронту), сообщающая микроконтроллеру о том, что выходные данные приняты внешним устройством. В разных вариантах этого протокола линия STRA используется также и в качестве линии разрешения вывода.

Микроконтроллер помещает данные на выходные линии порта C и подтверждает это действие сигналом на линии STRB. После этого внешнее устройство считывает данные с последующим подтверждением по линии STRA, означающим, что на порт C уже можно выводить новые данные. Активный перепад на линии STRA позволяет сбросить линию STRB и установить флаг STAF. В ответ на установку флага STAF программа выводит через порт C новые данные. Запись данных в регистр PORTCL вызывает появление этих данных на выходных линиях порта C и подачу сигнала подтверждения по линии STRB.

#### *Режима двунаправленного обмена*

Существуют варианты протокола вывода, которые допускают режим работы порта C в третьем состоянии, то есть возможно прямое соединение

этого 8-разрядного порта с другими устройствами, имеющими трехстабильные выходы без использования других вспомогательных схем.

Во время нахождения линии STRA в неактивном состоянии направление передачи на всех линиях порта C определяется содержимым регистра направления, то есть линии, настроенные как входы находятся в высокоимпедансном состоянии. При активизации линии STRA все линии порта C становятся выходами несмотря на содержимое регистра направления. Следует заметить, что в рассматриваемом протоколе чтение порта C всегда будет возвращать величину на входе выходных буферов порта, независимо от состояния регистра направления, поскольку эти линии могут не иметь значащих данных в варианте данного протокола с третьим состоянием. Эта особенность делает непрактичным использование некоторых линий порта C для ввода во время протокола вывода, но не мешает им являться статическими выходами. У линий порта C, которые используются в качестве статических выходов или обычных выходов со стробированием, соответствующие биты регистра направления должны быть установлены, а у линий, использующихся как выходы с высокоимпедансным состоянием в протоколе вывода с квитиowaniem - сброшены.

#### *Синхронный параллельный обмен*

В однокристалльном режиме все порты могут использоваться как порты общего назначения и работать на обмен в синхронном режиме, причем направление передачи для каждого разряда портов C и D независимо программируется значением соответствующего разряда регистров DDRC и DDRD ("0" - ввод данных, "1" - вывод). При попытке прочитать состояние линии, настроенной на вывод, возвращается величина присутствующая на входе выходного буфера линии. При настройке линии на ввод, соответствующий вывод устанавливается в высокоимпедансное состояние. Попытка произвести запись в такую линию не изменяют ее состояние, но записываемые данные сохраняются в выходном триггере линии. При настройке линии на вывод данные, записанные в триггере, подаются на выход. При RESET биты регистра управления направлением передачи данных сбрасываются, то есть все линии настраиваются на ввод данных.

Линии портов A, B и E (исключая биты 3 и 7 порта A) имеют фиксированное направление передачи данных.

Когда порт A используется в качестве порта общего назначения, биты [0:2] настраиваются только на ввод данных и запись информации по этим линиям не дает никакого эффекта. Биты [4:6] этого порта настроены исключительно на вывод данных и их чтение возвращает уровень на входе соответствующих выходных буферов. Каждый из битов 3, 7 может быть настроен как на ввод, так и на вывод данных с помощью битов DDRA3 и DDRA7 регистра управлением счетчиком внешних импульсов.

При использовании порта В как порта общего назначения он настраивается исключительно на вывод и чтение его линий возвращает состояние на входах соответствующих выходных буферов.

Линии порта Е соответствуют восьми каналам встроенного АЦП, но также могут быть использованы для обычного дискретного ввода. Запись в PORTE не дает никакого эффекта.

В расширенном режиме порты В и С используются только для связи с внешней памятью, линия AS/STRA стробирует передачу адреса на шину адреса/данных (порт С), а R/W/STRB определяет направление передачи данных.

Таблица 5.11. Работа портов ввода/вывода в асинхронных режимах

Режимы работы	HNDS	OIN	PLS	Способ очистки STAF*	Порт С	Порт В
Режим с простым стробированием	0	X	X	Считать PIOC (STAF=1), потом прочитать PORTCL	Входное значение копируется в PORTCL по активному фронту на STRA	Запись в порт В стробируется сигналом на STRB
Режим ввода с квитированием	1	0	0=STRB работает по уровню 1=STRB работает по импульсу	Считать PIOC (STAF=1), потом прочитать PORTCL	Входное значение копируется в PORTCL по активному фронту на STRA	Обычный порт вывода, на который в этом режиме не оказывается никакого влияния
Режим вывода с квитированием	1	1	то же	Считать PIOC (STAF=1), потом записать в PORTCL	Выход, если на STRA активный уровень; иначе - согласно DDRC	Обычный порт вывода, на который в этом режиме не оказывается никакого влияния

\*Установка по активному фронту на STRA

Запись данных по адресу выходного порта (линии) устанавливает соответствующие значения триггеров выходного буфера, открытого на внешние выводы. Чтение данных по адресу входного порта возвращает состояние внешних выводов.

Противоречия: попытка чтения по адресу порта (линий), объявленного выходным, возвращает состояние соответствующих выходных буферов. Попытка записи по адресам входных портов (линий) не меняет состояния выходных выводов, причем для двунаправленных (программируемых) линий записываемые данные сохраняются в выходных триггерах линий, а для однонаправленных линий ввода запись не дает никакого эффекта.

1. В синхронном режиме могут работать все порты, причем A[7,3], C[7:0], D[5:0] – каждый разряд может независимо настраиваться на ввод или вывод. A[2:0], E[7:0] – только на ввод; A[6:4], B[7:0] – только на вывод;



2. При  $PIOC[4]=HNDS=0$  устанавливается стробируемый ввод из порта С и стробируемый вывод в порт В. В этом режиме по фронту сигнала STRA осуществляется фиксация состояния линий порта С1 в регистра-защелку PORTCL, которая всегда сопровождается установкой флага STAF и запросом прерывания, если установлен бит STAI. Сброс STAF осуществляется аппаратно при чтении PORTCL. Кроме того, всегда можно прочитать текущее состояние входных линий из PORTC.

Всякая запись данных в PORTB1 в этом режиме сопровождается выходным стробом STRB, длительностью  $2 \times E$ .

3. Порт С может работать в полностью асинхронном режиме (с полным квитированием установления связи) на ввод и вывод.

Асинхронный ввод : по STRA данные с входных линий С защелкиваются в PORTCL, устанавливается флаг STAF, и сбрасывается сигнал готовности STRB. Последнее препятствует выработке нового строба STRA. Чтение из устанавливает сигнал готовности STRB.

Асинхронный вывод : выдача данных на линии порта С (запись данных в регистр PORTCL) сопровождается формированием сигнала готовности STRB. Когда ВУ считывает данные с линий порта С, оно выставляет подтверждение по STRA. Активный перепад на STRA сбрасывает STRB и устанавливает флаг STAF. После записи в PORTCL устанавливается STRA и т.д.

### **5.3.6. Система контроля временных интервалов**

Система контроля временных интервалов (СКВИ) включает в себя (Рис.5.33):

- 16-разрядный таймер;
- предделитель частоты E для таймера с программируемым коэффициентом деления;
- три 16-разрядных регистра входной фиксации;
- четыре 16-разрядных регистра выходного сравнения;
- 16-разрядный регистр, который может работать как четвертый регистр входной фиксации или пятый регистр выходного сравнения;
- 8-разрядный счетчик внешних событий или измеритель длительности внешних импульсов;
- генератор прерываний реального времени с программируемым периодом;
- блок 8-разрядных программно-доступных регистров (10) управления и состояния СКВИ.

Для взаимодействия с СКВИ используются внешние выходы порта А, причем  $A[2:0]$  используются для ввода сигналов входной фиксации,  $A[4:7]$  - для вывода сигналов выходного сравнения, а линия  $A3$  – в зависимости от назначенных функций регистра I4/O5. Кроме того, линия  $A7$  может использоваться в качестве входной для счетчика внешних событий.

	7	6	5	4	3	2	1	0	
100B	FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0	<b>CFORG</b>
100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0	<b>OC1M</b>
100D	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0	<b>OC1D</b>
1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	<b>TCTL1</b>
1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	<b>TCTL2</b>
1022	OOC1I	OOC2I	OOC3I	OOC4I	I4O5I	IC1I	IC2I	IC3I	<b>TMSK1</b>
1023	OOC1F	OOC2F	OOC3F	OOC4F	I4O5F	IC1F	IC2F	IC3F	<b>TFLG1</b>
1024	TOI	RTI	PAOVI	PAI	0	0	PR1	PR0	<b>TMSK2</b>
1025	TOF	RTIF	PAOVF	PAF	0	0	0	0	<b>TFLG2</b>
1026	DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0	<b>PACTL</b>

Рис.5.33. Регистры состояния и управления системы контроля времени

Регистры СКВИ имеют следующее назначение:

- **CFORG** – регистр принудительного сравнения, запись “1” в разряд(ы) которого вызывает соответствующее событие сравнения (но без установки флага);
- **OC1M** – определяет те биты порта A[7:3], на которые будет влиять OC1;
- **OC1D** – содержит данные, которые будут выводиться на разрешенные линии A[7:3] при OC1;
- **TCTL1** – четыре двухбитовых поля (OM:OL) определяют действия, выполняемые на выходах OC при сравнении;
- **TCTL2** – четыре двухбитовых поля (EDGxB:EDGxA) определяют фронт(ы) сигналов на входах, по которым производится входная фиксация;
- **TFLG1, TFLG2** – содержат флаги событий СКВИ;
- **TMSK1, TMSK2** – содержат биты маски прерываний соответствующих флагов СКВИ. Кроме того, TMSK2 содержит поле, управляющее предделителем таймера;
- **PACTL** – регистр управления счетчиком импульсов. Кроме того, TMSK2 содержит поле, определяющее период прерывания реального времени.

Таймер представляет собой 16-разрядный суммирующий счетчик, на вход которого поступают импульсы с выхода предделителя, делящего внутреннюю частоту генератора E на один из четырех делителей (1, 4, 8, 32), выбираемый полем PR[1:0] регистра TMSK2, причем программирование этого поля возможно только в первые 64 такта после RESET.

Переполнение таймера – переход из состояния FFFFh в состояние 0000h – вызывает установку флага переполнения и может вызвать прерывание, если соответствующий разряд регистра маски установлен. Программа может в любое время считывать показания счетчика, никак не влияя на его состояние, поскольку его считывание и инкремент происходит в противоположных фазах сигнала E.

### *Входная фиксация*

При подаче на один из входов системы фиксации фронта импульса, текущее содержимое таймера защелкивается в соответствующем регистре входной фиксации. При этом устанавливается связанный с этим регистром флаг входной фиксации и может быть вызвана прерывающая подпрограмма, если прерывание разрешено установкой в “1” соответствующего бита маски.

Каждому входу входной фиксации соответствует пара бит в регистре управления TCTL2, которые определяют, по какому фронту внешнего импульса произойдет фиксация:

EDG <sub>i</sub> B	EDG <sub>i</sub> A	Конфигурация
0	0	Функция входной фиксации запрещена
0	1	Фиксация только по положительному фронту
1	0	Фиксация только по отрицательному фронту
1	1	Фиксация по любому из двух фронтов

Регистры входной фиксации могут быть прочитаны в любой момент времени.

### *Выходное сравнение*

Четыре (пять) регистра выходного сравнения TOS<sub>i</sub> доступны по записи и чтению, причем по сбросу RESET в них заносится значение FFFFh. Каждый регистр снабжен собственной схемой сравнения, обеспечивающей непрерывное сравнение текущего состояния TCNT с содержимым регистра TOS<sub>i</sub>. В момент совпадения устанавливается флаг выходного сравнения в регистре TFLG1 и автоматически выполняется одно из действий, определяемых разрядами OM<sub>i</sub> OL<sub>i</sub> регистра TCTL1:

OM <sub>i</sub>	OL <sub>i</sub>	Действия, предпринимаемые в случае успешного сравнения
0	0	Таймер отключен от логики вывода
0	1	Инвертировать состояние выходной линии OC <sub>i</sub>
1	0	Сбросить состояние выходной линии OC <sub>i</sub> в ноль
1	1	Установить состояние выходной линии OC <sub>i</sub> в единицу

### *Принудительное сравнение*

В СКВИ предусмотрена возможность искусственного формирования событий выходного сравнения в произвольном подмножестве регистров выходного сравнения. В состав регистров управления входит доступный по записи регистр CFORC, пять разрядов которого соответствуют пяти регистрам и выводам выходного сравнения. Установка в “1” бита в этом регистре вызывает событие “принудительного сравнения” соответствующего регистра с изменением состояния выхода. Однако содержимое регистра выходного сравнения не меняется, поэтому “в свое время” может произойти обычное сравнение. Записью в регистр CFORC нескольких “1” можно вызвать “действия совпадения” сразу на нескольких выходах.

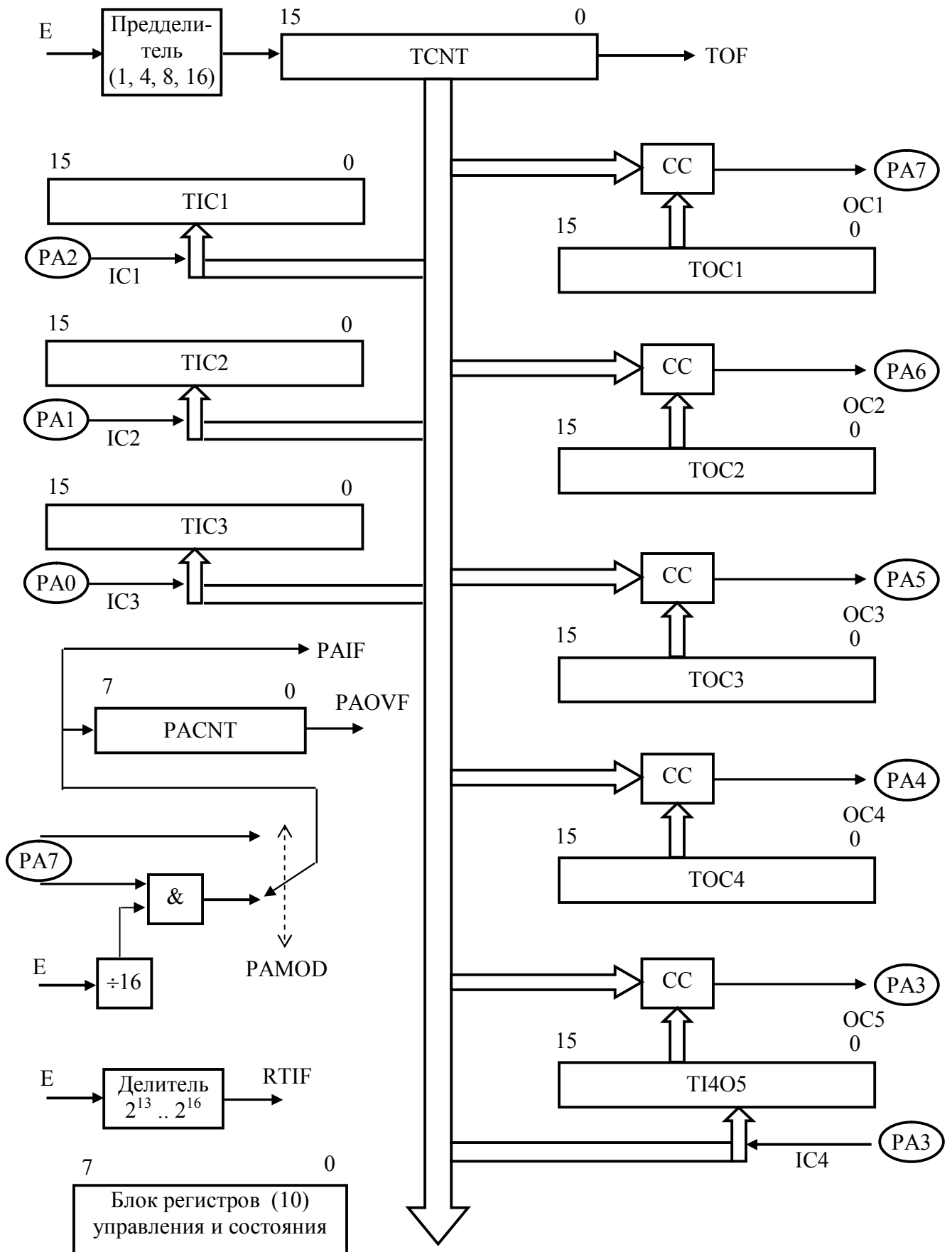


Рис.5.34. Система контроля времени

### *Особенности выходного сравнения 1:*

Особую роль играет *таймер выходного сравнения 1*, который может влиять сразу на все выходы (биты 3-7 порта А) выходного сравнения (или любое их подмножество). Для реализации этой функции предусмотрены два пятиразрядных регистра - регистр маскирования выходного сравнения 1 (OC1M) и регистр данных выходного сравнения 1 (OC1D).

Регистр OC1M используется для определения тех битов порта А (порт ввода/вывода и таймера), на которые будет произведено воздействие в результате успешного сравнения OC1. Регистр OC1D используется для определения данных, которые будут выведены на линии порта А в результате успешного сравнения, если разрешено воздействие на соответствующий бит. В случае совпадения значения в регистре сравнения OC1 и любом другом в течение одного цикла E, и если результатом обоих будет попытка изменить состояние одних и тех же линий порта А, доминирует выходное сравнение OC1.

Эта функция позволяет выполнять автоматическое управление несколькими вводами/выводами с помощью единственного выходного сравнения.

Другой вариант использования специального управления вводами/выводами состоит в том, чтобы позволить более одного сравнения для управления одним выводом. В этом случае можно формировать импульсы каждый E цикл.

### *Счетчик внешних событий:*

8-разрядный суммирующий счетчик внешних событий изменяет свое состояние под действием фронта (положительного или отрицательного – программируется) на линии PA7 или под действием тактовой частоты E, умноженной на значение потенциала линии PA7 (высокого или низкого – программируется). Всякий входной импульс, изменяющий состояние счетчика, устанавливает флаг PAIF и может вызвать прерывание. Другой флаг PAVF устанавливается по переполнению счетчика, которое тоже имеет свой вектор прерывания.

Таким образом, счетчик внешних событий может, во-первых, подсчитывать внешние импульсы (с частотой до E/2) и, во-вторых, измерять длительность внешнего импульса (положительного или отрицательного) в периодах E/64.

Содержимое счетчика в любой момент доступно как по чтению, так и по записи.

### *Генератор прерываний реального времени*

СКВИ включает в себя программируемый делитель тактовой частоты E (с коэффициентами деления  $2^{13}$ ,  $2^{14}$ ,  $2^{15}$ ,  $2^{16}$ ), выход которого вызывает радиальное прерывание RTI реального времени. Как и любое прерывание, оно может быть замаскировано. Коэффициент деления программируется в

поле RTR[1:0] регистра PACTL в первые 64 такта после RESET, и в дальнейшем не может быть изменен.

Например, при использовании частоты XTAL = 8 МГц (E = 2 МГц) периоды прерываний реального времени будут составлять для кодов RTR[1:0] = 00, 01, 10, 11 соответственно 3,91 7,81 15,62 и 31,25 мс.

### 5.3.7. Подсистема аналого-цифрового преобразователя

Подсистема АЦП (Рис.5.35) включает в себя:

- ◆ 8-разрядный АЦП;
- ◆ мультиплексор аналогового сигнала 16→1;
- ◆ четыре регистра результатов преобразования;
- ◆ регистр управления и состояния АЦП.

Для ввода аналоговых сигналов используются линии порта PE[7:0]. Кроме того, для управления включением питания и тактовой частоты для АЦП используются два разряда регистра OPTION.

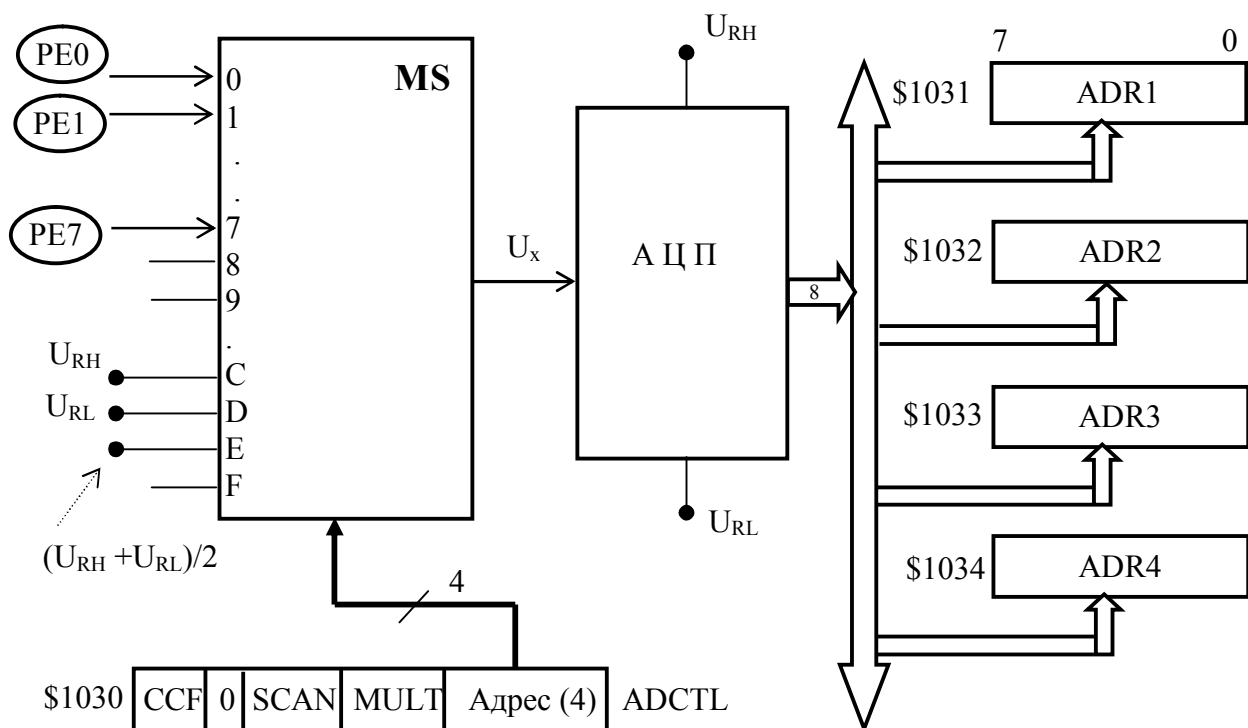


Рис.5.35. Подсистема аналого-цифрового преобразования

Кроме приведенных на Рис.5.34 объектов, к системе АЦП относятся два бита регистра OPTION (1039h). Бит 7 - ADPU - будучи установленным включает питание АЦП, а бит 6 - CSEL - определяет источник синхронизации для аналого-цифрового преобразования. Если бит CSEL сброшен, то система АЦП использует для синхронизации сигнал внутренней тактовой частоты E. Когда этот бит установлен, система АЦП синхронизируется от

внутреннего генератора, который работает на частоте около 1.5 МГц. АЦП не может нормально функционировать при частоте сигнала синхронизации микроконтроллера ниже 750 КГц, поэтому если в микроЭВМ выбрана низкая частота синхронизации  $E$  (ниже 750 КГц), следует выбирать внутренний генератор. Следует заметить, что бит CSEL управляет выбором источника синхронизации также и для генератора накачки заряда ЭСПЗУ.

АЦП измеряет напряжение на входе относительно опорного напряжения  $U_{RL}$  и  $U_{RH}$ . Входное напряжение, равное  $U_{RL}$ , преобразуется в 00, а напряжение, равное  $U_{RH}$  - в FF, никак не регистрируя выход за этот диапазон. Для правильного преобразования источник аналогового сигнала должен использовать напряжения  $V_{RH}$  как напряжение питания относительно  $V_{RL}$ . Обычно  $U_{RH} = U_{DD} = +5V \pm 10\%$  а  $U_{RL} = U_{SS} = 0V$ , однако, для увеличения точности преобразования допускается уменьшать диапазон  $U_{RH} - U_{RL}$  до 2,5... 3 В.

Мультиплексор позволяет выбирать для единственного АЦП один из шестнадцати аналоговых сигналов. Восемь из этих шестнадцати сигналов соответствуют восьми выводам порта  $E$ , три связаны с внутренними точками и предназначены для тестирования, а оставшиеся пять - зарезервированы для последующего использования.

Регистр управления и статуса АЦП ADCTL включает в себя четырехразрядное поле адреса преобразуемого канала, биты режимов преобразования MULT и SCAN и флаг завершения преобразований CCF.

Преобразование начинается после записи в ADCTL. Каждое преобразование требует 32 периода частоты  $E$ . Система всегда выполняет подряд четыре преобразования и размещает результаты в регистрах ADR1 .. ADR4 соответственно. По окончании четвертого преобразования устанавливается флаг CCF. Сбрасывается CCF после перезаписи регистра ADCTL.

В *одноканальном* режиме все четыре преобразования выполняются над значениями одного выбранного поля адреса канала, а в *многоканальном* - над группой из четырех последовательных входов мультиплексора, определяемых двумя старшими разрядами поля адреса. При этом значения двух младших разрядов адреса игнорируются.

Если определен режим *сканирования* (SCAN = 1), то последовательно из четырех преобразований повторяются, независимо от состояния флага CCF. При SCAN = 0 после заполнения ADR1 .. ADR4 преобразование прекращается и может возобновиться только после перезаписи регистра ADCTL.

### 5.3.8. Прерывания

В микроЭВМ MC68HC11E9 предусмотрено два входа запросов внешних радиальных прерываний - IRQ и XIRQ. Кроме того, MC68HC11E9 поддерживает большое число внутренних прерываний. С каждым источ-

ником аппаратных (внутренних и внешних) и программных прерываний связан свой вектор, размещенный по фиксированному адресу в таблице векторов прерываний (табл. 5.12). К внутренним прерываниям относятся и различные варианты RESET. Прерывания MC68HC11E9 можно разделить на две категории: маскируемые и немаскируемые. Пятнадцать прерываний могут быть замаскированы битом I регистра признаков. Дополнительно к этому, все встроенные источники прерываний можно маскировать специальными управляющими битами. В приведены все источники прерываний, соответствующие им адреса векторов и биты локальных масок.

Программное прерывание (команда SWI) является скорее немаскируемой командой, чем маскируемым прерыванием. Прерывание по неверному коду команды - это немаскируемое прерывание. Внешний вход XIRQ\ рассматривается как источник немаскируемого прерывания, потому что будучи однажды разрешенным это прерывание не может быть запрещено программно; однако оно маскируется в течение RESET и при получении сигнала на выводе XIRQ.

Таблица 5.12. Назначение векторов прерываний.

Адрес вектора	Источник прерывания	Локальная маска
FFC0,C1	Зарезервировано	-
FFD4,D5	Зарезервировано	-
FFD6,D7	Система SCI	1)
FFD8,D9	Завершение передачи SCI	SPIE <sup>1)</sup>
FFDA,DB	Входной фронт счетчика импульсов	PAPI <sup>1)</sup>
FFDC,DD	Переполнение счетчика импульсов	PAOVI <sup>1)</sup>
FFDE,DF	Переполнение таймера	TOI <sup>1)</sup>
FFE0,E1	Выходное сравнение 5 таймера	OC5I <sup>1)</sup>
FFE2,E3	Выходное сравнение 4 таймера	OC4I <sup>1)</sup>
FFE4,E5	Выходное сравнение 3 таймера	OC3I <sup>1)</sup>
FFE6,E7	Выходное сравнение 2 таймера	OC2I <sup>1)</sup>
FFE8,E9	Выходное сравнение 1 таймера	OC1I <sup>1)</sup>
FFEA,EB	Входная фиксация 3 таймера	IC3I <sup>1)</sup>
FFEC,ED	Входная фиксация 2 таймера	IC2I <sup>1)</sup>
FFEE,EF	Входная фиксация 1 таймера	IC1I <sup>1)</sup>
FFF0,F1	Прерывания реального времени	RTI <sup>1)</sup>
FFF2,F3	IRQ\ (Внешний вывод или параллельный ввод\вывод)	FI <sup>1)</sup>
FFF4,F5	XIRQ\ вывод (псевдо-немаскируемое прерывание)	Нет <sup>2)</sup>
FFF6,F7	SWI	Нет <sup>3)</sup>
FFF8,F9	Прерывание по неверному коду команды	Нет <sup>3)</sup>
FFFA,FB	RESET, вызываемый системой слежения	NOCOP <sup>3)</sup>
FFFC,FD	RESET, тактового монитора системы слежения	CME <sup>3)</sup>
FFFE,FF	RESET	Нет <sup>3)</sup>



Примечания:

- 1) Маскируется разрядом I регистра состояния СС
- 2) Маскируется разрядом X регистра состояния СС
- 3) Не маскируется

Некоторые источники могут генерировать запрос по нескольким различным причинам. Дополнительная идентификация причины в этом случае осуществляется программно

Таблица 5.13. Прерывания последовательного интерфейса связи.

Причина прерывания	Локальная маска
Регистр принимаемых данных полон	RIE
Приемник переполнен	RIE
Обнаружена свободная линия	ILIE
Регистр передаваемых данных пуст	TIE
Передача завершена	TCIE

Таблица 5.14. Прерывания по IRQ\.

Причина прерывания	Локальная маска
Внешний вывод	Нет
Квитирование параллельного ввода/вывода	STAI

При возникновении внешнего или внутреннего прерывания, оно не обрабатывается до тех пор, пока не будет закончено выполнение текущей команды. Прерывание ожидает обработки до полного выполнения команды. При обработке прерывания в стек аппаратно помещается содержимое регистров процессора в порядке, показанном на рис. , после чего устанавливается флаг маски прерывания I.

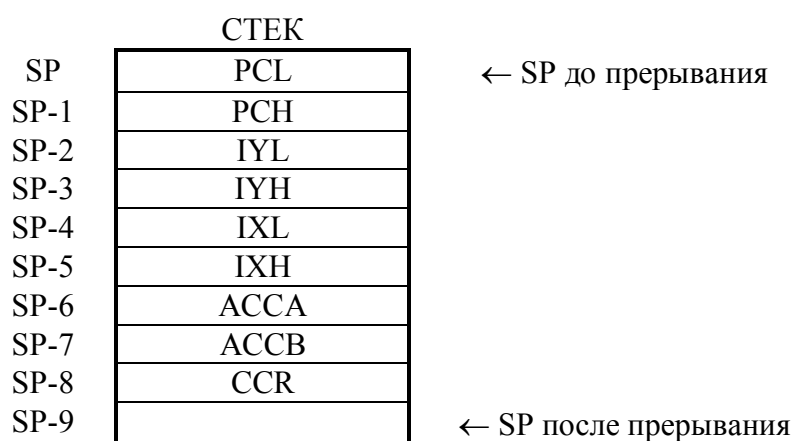


Рис.5.36. Порядок заполнения стека при обслуживании прерывания.

### *Дисциплина обслуживания прерываний*

После **RESET** устанавливаются биты I и X для того, чтобы запретить все маскируемые прерывания и прерывание XIRQ\ . Программно можно

сбросить бит X командой TAP, разрешая таким образом прерывания XIRQ\ . После этого уже нельзя установить бит X, то есть это прерывание на самом деле является немаскируемым. Изменение бита I, связанного с системой прерываний, также не оказывает никакого влияния на бит X. В системе прерываний XIRQ\ имеет более высокий приоритет относительно всех источников, которые маскируются битом I. Все связанные с битом I прерывания действуют в соответствии со своей системой приоритетов. При возникновении прерывания, связанного с битом I, этот бит автоматически устанавливается после сохранения в стеке регистра признаков, но бит X не изменяется. При возникновении прерывания, связанного с битом X, аппаратно устанавливаются оба бита - X и I - после сохранения в стеке регистра признаков. Команда RTI (возврат из прерывания) восстанавливает состояние битов X и I на момент запроса прерывания.

### *Приоритеты запросов*

Для того, чтобы разрешить ситуацию одновременного запроса нескольких прерываний, у различных прерываний аппаратно установлен разный приоритет. Однако один из источников прерывания маскируемый битом I можно программно сделать прерыванием с наивысшим приоритетом. Первые шесть источников прерываний не маскируются битом I регистра признаков и имеют следующий фиксированный приоритет: RESET, RESET монитора тактовой частоты, RESET системы слежения, прерывание по неверному коду команды и XIRQ\ . (SWI - является командой и имеет наивысший приоритет после RESET в том смысле, что после выборки кода SWI никакое другое прерывание не обрабатывается до тех пор, пока не будет выбран вектор SWI, то есть выполнена команда). Каждый из этих источников прерываний подключен на вход системы арбитража. Наивысший приоритет источнику, маскируемому битом I назначается программно - регистром HPRIО, при этом вход прерывания с наивысшим приоритетом соединяется с выбранным источником прерывания. Для того, чтобы избежать конфликтов, регистр HPRIО можно перезаписывать только если прерывания маскируемые битом I запрещены (бит I регистра признаков установлен в логическую единицу). Прерывание, которому назначается наивысший приоритет, можно запретить соответствующим битом управления или битом I регистра признаков. При назначении прерыванию наивысшего приоритета его вектор не изменяется.

Регистр HPRIО используется для присвоения наивысшего приоритета одному из источников прерывания, маскируемого битом I. Дополнительно к этому в регистр включено четыре системных управляющих бита.

	7	6	5	4	3	2	1	0	
103C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIО
RESET	-	-	-	-	0	1	0	1	

**RBOOT** - чтение ПЗУ режима загрузки. Этот бит имеет значение только если бит SMOD установлен в единицу (выполняется режим загрузки или режим тест). Все остальное время этот бит сброшен и не может быть перезаписан. Установленный бит в режиме загрузки разрешает чтение небольшой программы загрузчика. Сброшенный во всех трех других режимах бит запрещает доступ к данному ПЗУ и обращение к этим адресам памяти интерпретируется как доступ к внешней памяти.

**SMOD** - специальный режим. Бит является инверсией состояния вывода MODB во время положительного фронта RESET. Бит устанавливается, если во время RESET на выводе MODB установлен низкий уровень сигнала. Если во время RESET на выводе MODB установлен высокий уровень, то SMOD сбрасывается. Бит может быть программно сброшен при работе в специальных режимах, изменяя таким образом режим работы микроконтроллера. Установить бит программно невозможно.

**MDA** - выбор режима A. Бит отображает состояние вывода MODA во время положительного фронта сигнала RESET. Когда бит SMOD установлен (микроконтроллер находится в режиме теста или загрузки), бит MDA может быть перезаписан, изменяя таким образом режим работы микроконтроллера. При сброшенном бите SMOD бит MDA доступен только для чтения и режим работы не может быть изменен без выполнения RESET.

**IRV** - отображение внутреннего чтения информации. Данный бит используется в специальных режимах (SMOD=1) для отображения процесса внутреннего чтения на внешнюю шину данных.

**PSEL[3:0]** - выбор приоритета. Эти четыре бита используются для присвоения наивысшего приоритета одному из маскируемых битом I источнику прерываний. Эти биты могут быть перезаписаны только если бит I регистра CCR установлен в единицу, то есть соответствующие прерывания запрещены.

**Программное прерывание** выполняется аналогично другим командам. Прерывание SWI имеет приоритет над другими прерываниями, если они замаскированы (биты I и X регистра признаков установлены). Процесс обработки SWI похож на процесс обработки других маскируемых прерываний: устанавливается бит I, регистры процессора сохраняются в стеке и т.д.

Так как не все возможные коды команд или последовательности команд определены, то микроконтроллер включает в себя схемы **определения неверного кода команды**. Когда обнаруживается неверный код команды, то запрашивается прерывание. Вектор этого прерывания никогда не следует оставлять неинициализированным. Можно посоветовать инициализировать указатель стека при обработке данного прерывания, так что повторяющаяся последовательность неверных команд не вызовет переполнения стека.

## RESET

Частью подсистемы прерываний MC68HC11E9 является подсистема начальной установки - RESET. Микроконтроллер имеет четыре варианта реализации RESET :

- по активному внешнему сигналу на выводе RESET\;
- RESET при включении питания;
- RESET, производимый системой слежения;
- RESET тактового монитора,

причем каждому варианту RESET соответствует<sup>1</sup> вектор прерывания.

**Внешний вывод RESET\** используется для начальной установки (инициализации) микроконтроллера. При RESET микроконтроллер аппаратно устанавливает низкий уровень сигнала на этом выводе на 4 E цикла, после чего освобождает его. Еще через два цикла E вывод тестируется. Если вывод все еще находится в состоянии низкого уровня, это означает, что произошел внешний RESET. Если же на выводе высокий уровень, то RESET был инициирован или системой слежения или тактовым монитором. Этот способ распознавания внутреннего и внешнего RESET предполагает, что вывод RESET\ возвратится в состояние "1" не менее чем за два E цикла и что внешне генерируемый сигнал RESET должен находиться в активном состоянии не менее восьми E циклов.

**RESET при включении питания** происходит при положительном перепаде на выводе  $V_{DD}$ . Такой RESET используется исключительно при включении питания, и его не следует использовать для обнаружения падения напряжения питания. Схема формирователя обеспечивает задержку в 4064 цикла, начиная с первого импульса генератора. В системах с частотой  $E=2$  МГц RESET при включении питания длится около двух миллисекунд. Если после задержки на выводе RESET\ все еще присутствует низкий уровень, то микроконтроллер продолжает находиться в состоянии RESET до появления на выводе RESET\ высокого уровня.

В контроллере MC68HC11E9 при подаче внешнего RESET\ или RESET при включении питания производятся следующие действия:

- в течение первых трех циклов ЦП загружает вектор из ячеек FFFE и FFFF (в режимах загрузки или теста - из ячеек BFFE и BFFF) и начинает выполнять команды начиная с адреса, соответствующего этому вектору;
- указатель стека и другие регистры ЦП не определены непосредственно после RESET, однако флаги I и X маскирования прерываний в регистре признаков устанавливаются, тем самым запрещая обработку любых прерываний. Кроме того, бит S регистра признаков устанавливается для запрета режима STOP;
- регистр INIT инициализируется значением 01, перенося таким обра-

---

<sup>1</sup> первым двум - FFFE, третьему - FFFA, четвертому - FFFC.

зом 512 байт ОЗУ на адреса 0000-01FF и регистры управления на адреса 1000-103F; состояние регистра CONFIG при этом не меняется, следовательно, не меняется и место 12 Кбайт ПЗУ и/или 512 байт ЭСППЗУ на карте памяти;

- системный таймер инициализируется значением 0000. Биты предделителя обнуляются и все регистры выходного сравнения инициализируются значением FFFF. Содержание всех регистров входной фиксации после RESET не определено. Регистр 1 маскирования выходного сравнения (OC1M) обнуляется, так что успешное выходное сравнение не окажет никакого влияния на выводы. Другие четыре функции выходного сравнения настраиваются так, чтобы не влиять на выводы при успешном сравнении. Все три схемы обнаружения фронта для входной фиксации настраиваются на "запрещение фиксации". Флаг переполнения таймера и флаги прерываний всех восьми функций таймера обнуляются. Запрещаются все девять видов прерываний от таймера;

- флаг прерываний реального времени сбрасывается и автоматически маскируются аппаратные прерывания. Биты управления периодом сбрасываются после RESET и могут быть инициализированы перед использованием системы прерываний реального времени;

- система счета импульсов во время RESET запрещается, так что вывод PAI определяется как вывод общего назначения, настроенный на ввод;

- Условия RESET для SCI независимы от режима работы. После RESET скорость обмена SCI не определена и должна быть программно установлена записью в регистр BAUD. Все прерывания передачи и приема маскируются, и как приемник, так и передатчик запрещаются, а выводы порта определяются как линии общего назначения. Формат передачи SCI определяется для длины восемь бит. Посылка символа останова прекращается, запрещается режим слежения. Биты статуса TDRE и TC регистра статуса SCI устанавливаются, означая таким образом, что передаваемых данных нет ни в регистре данных, ни в регистре последовательного сдвига. Биты RDRF, IDLE, OR, NF и FE, связанные с передатчиком, сбрасываются;

- система SPI во время RESET запрещается и выводы, связанные с SPI определяются как линии ввода/вывода общего назначения;

- конфигурация системы АЦП не определена. Флаг завершения преобразования сбрасывается. Бит ADPU во время RESET сбрасывается, запрещая таким образом систему АЦП;

- Программирование ЭСППЗУ запрещено, так что память системы настроена для работы в режиме обычного чтения. Самый высокий приоритет прерывания I определяется для внешнего вывода IRQ битами PSEL[3:0] = 0101. Вывод прерывания настраивается для работы по уровню (для возможности объединения по "монтажному ИЛИ"). Биты RBOOT, SMOD и MDA регистра HPRIO отображают состояние вводов MODA и MODB на

положительном фронте сигнала RESET. Бит управления DLY устанавливается для определения того, что генератору нужна стартовая задержка после выхода из режима STOP. Система тактового монитора запрещается сбросом бита CME в нуль.

В состав микроконтроллера включена система слежения за правильностью работы программы (Computer Operating Properly - COP). При использовании COP, **RESET, вызванный системой слежения**, выполняется через регулярные периоды времени, так что COP никогда не допустит заклинивания программы дольше определенного времени. Система управления COP включает специальные биты, которые позволяют определить один из четырех периодов времени или полностью запретить работу COP. COP имеет отдельный вектор RESET.

Бит управления NOCOP, который определяет, будет ли COP вызывать RESET или нет, встраивается как ячейка ЭСППЗУ в регистр CONFIG. Однажды запрограммированный, этот бит остается установленным (или сброшенным) даже при отключении питания, и работа COP разрешена или запрещена независимо от выполняемой программы. В режиме теста бит NOCOP может быть переустановлен для предотвращения генерации аппаратного RESET от COP. Два других управляющих бита регистра OPTION выбирают один из четырех периодов работы таймера COP, определяя коэффициент деления частоты синхронизации E: - 1, 4, 16 или 64. По умолчанию оба эти управляющих бита сброшены, что соответствует самому короткому периоду.

Для того, чтобы инициализировать COP, необходимо выполнить следующую последовательность действий:

- 1) Записать значение 55 в регистр сброса COP (COPRST) по адресу 103A;
- 2) Записать значение AA в тот же самый регистр.

Команды записи должны происходить в указанном порядке. Между двумя записями можно выполнить любое количество других команд, но обе команды записи должны быть выполнены до следующего RESET, вызванного системой COP. Время между двумя последовательностями программных сигналов RESET никогда не должно превышать величину периода COP. Чтение регистра COPRST не возвращает значащих данных и не оказывает никакого влияния на работу таймера системы слежения.

Работа **монитора тактовой частоты** разрешается установкой управляющего бита CME регистра OPTION. Когда бит CME сброшен, работа монитора запрещена. Если же бит CME установлен, то тактовый монитор определяет отсутствие сигнала синхронизации E за определенный промежуток времени. Величина этого промежутка зависит от параметров работы и может быть от 5 до 100 микросекунд. Это означает, что при тактовой частоте E 200 КГц и больше монитор никогда не вызовет сбой и, наоборот, при частоте менее 10 КГц монитор будет однозначно вызывать сбой. Та-

ким образом, системы, работающие на частоте около 200 КГц или ниже не должны использовать монитор тактовой частоты.

При обнаружении сбоя частоты синхронизации схемы монитора вызывают системный RESET. Этот RESET передается внешней системе через двунаправленный вывод RESET\-. Система тактового монитора имеет отдельный вектор RESET.

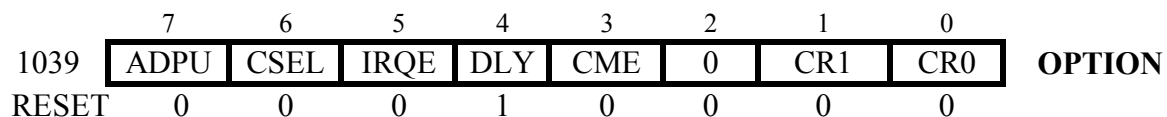
При использовании в одной и той же системе режима STOP и тактового монитора следует внимательно планировать процесс работы, так как после остановки генератора тактовый монитор будет вызывать RESET, при условии, что работа монитора разрешена на момент перехода в режим STOP.

Тактовый монитор обычно используется для дублирования работы системы слежения. Так как система слежения требует для своей работы тактовых импульсов от таймера, то она не обнаружит сбой системы при отсутствии тактовых импульсов. В этом случае монитор сигнала синхронизации обнаружит данный сбой и принудительно переведет микропроцессор в состояние RESET. Следует заметить, что для перехода микропроцессора в состояние RESET не требуется наличия тактовых импульсов, хотя они необходимы для возвращения из состояния RESET к нормальному режиму работы.

### ***5.3.9. Специальные средства микроконтроллера***

#### *Регистр выбора конфигурации (OPTION).*

Восьмиразрядный регистр специального назначения (рис. 5.37) может использоваться в течение инициализации для настройки системы МК. Исключая биты 7, 6 и 3, которые можно как читать, так и перезаписывать в любое время, запись в этот регистр разрешена только сразу после RESET, и, следовательно, он является регистром, доступным только для чтения. Если в течение первых 64 циклов E после RESET не произошло записи в этот регистр, то биты 5, 4, 1 и 0 (IRQE, DLY, CR1 и CR0) становятся доступны только для чтения, чтобы уменьшить вероятность любого случайного изменения конфигурации (запись просто игнорируется). В режиме теста механизм защиты этого регистра отключается и все биты регистра OPTION могут быть перезаписаны в любое время.



Регистр выбора конфигурации OPTION		
Разряд	Имя	Назначение
7	ADPU	Управление питанием АЦП. Когда бит ADPU сброшен, система АЦП отключена, и запросы на преобразование не возвращают значащей информации. Для использования АЦП необходимо, чтобы этот бит был установлен. После включения системы АЦП требуется задержка в 100 мкс для стабилизации работы системы АЦП.
6	CSEL	Выбор источника синхронизации для генератора накачки заряда ЭСППЗУ и для системы АЦП. Когда этот бит равен нулю, система АЦП и генератор накачки заряда тактируются от частоты синхронизации E. Если же бит CSEL равен единице, то работает отдельный RC - генератор на частоте около 2 МГц.
5	IRQE	Если этот бит установлен, то вывод IRQ\ настраивается для обнаружения уровня сигнала (низкого), а если бит сброшен - то фронта сигнала (отрицательного).
4	DLY	Этот бит устанавливается в течение RESET и управляет наличием определенной задержки перед началом работы после выхода из режима STOP. При установленном бите DLY на стабилизацию нормальной работы генератора отводится 4064 циклов E. При сброшенном бите DLY задержка пропускается (бит устанавливается в "0", если система использует внешний тактовый генератор).
3	CME	Разрешение монитора тактовой частоты. Работа монитора запрещена при сброшенном бите CME и разрешена в противном случае. Системы, работающие на частотах ниже 200 КГц не должны использовать тактовый монитор. Бит CME сбрасывается во время RESET.
2	0	Не используется, при чтении всегда возвращает нуль.
1	CR1	Служат для выбора периода системы слежения за временем выполнения программы.
0	CR0	

Рис. 5.37. Формат регистра специального назначения OPTION

### *Режимы пониженного энергопотребления.*

Микроконтроллер включает в себя два программируемых режима с пониженным энергопотреблением: WAIT и STOP.

**Команда WAIT** переводит микроконтроллер в режим пониженного энергопотребления, оставляя работающим тактовый генератор. После выполнения команды WAIT в стеке сохраняется состояние микроконтроллера, после чего выполнение программы прекращается. Выйти из режима WAIT можно только посредством RESET или немаскируемого прерывания. Если установлен бит I (соответствующие прерывания замаскированы) и работа системы слежения за временем работы программы запрещена, то выключается система таймера, что позволяет дополнительно уменьшить



энергопотребление. Количественно экономия энергоресурсов зависит как от конкретного применения микроконтроллера, в том числе и от подключенных к микроконтроллеру дополнительных схем, так и подсистем (то есть SPI, SCI, таймера), которые были активны при переходе в режим WAIT. Выключение подсистемы АЦП сбросом бита ADPU дополнительно снижает энергопотребление микроконтроллера.

**Команда STOP** переводит микроконтроллер в режим наименьшего энергопотребления, но только в случае, если сброшен бит S регистра признаков. Если же бит S установлен, то режим STOP запрещен и команда STOP интерпретируется как команда NOP (нет операции). В режиме STOP все источники синхронизации, включая внутренний генератор тактовых импульсов, останавливаются, что вызывает прекращение всех внутренних процессов. Выход из режима STOP осуществляется посредством RESET\, XIRQ\ или немаскируемым IRQ\ . При использовании XIRQ\ микроконтроллер выходит из режима STOP, независимо от состояния бита X. Если бит X сброшен, то микроконтроллер начинает с последовательного сохранения в стеке регистров с последующим обычным обслуживанием запроса XIRQ\ . Если же бит X установлен, то работа микроконтроллера продолжится с команды, следующей непосредственно за командой STOP и запроса на обработку прерывания XIRQ не будет. Начальная установка посредством RESET\ всегда ведет к выходу из режима STOP и работе микроконтроллера с команды, определяемой вектором RESET.

### 5.3.10. Система команд микроЭВМ MC68HC11E9

1. Команды пересылки и загрузки			
Мнемокод	Адресация	Операция	Примечание
CLRA		$0 \rightarrow A$	
CLRB		$0 \rightarrow B$	
CLR	П, OXY	$0 \rightarrow M$	
LDAA	H8, П0, П, OXY	$M \rightarrow A$	
LDAB	H8, П0, П, OXY	$M \rightarrow B$	
LDD	H16, П0, П, OXY	$M \rightarrow A; M + 1 \rightarrow B$	
LDS	H16, П0, П, OXY	$M.M+1 \rightarrow SP$	
LDX	H16, П0, П, OXY	$M.M+1 \rightarrow IX$	
LDY	H16, П0, П, OXY	$M.M+1 \rightarrow IY$	
STAA	П0, П, OXY	$A \rightarrow M$	
STAB	П0, П, OXY	$B \rightarrow M$	
STD	П0, П, OXY	$A \rightarrow M; B \rightarrow M + 1$	
STS	П0, П, OXY	$SP \rightarrow M.M+1$	
STX	П0, П, OXY	$IX \rightarrow M.M+1$	
STY	П0, П, OXY	$IY \rightarrow M.M+1$	
TAB		$A \rightarrow B$	
TBA		$B \rightarrow A$	

1. Команды пересылки и загрузки			
TAP		$A \rightarrow CCR$	
TPA		$CCR \rightarrow A$	
XGDX		$IX \rightarrow D; D \rightarrow IX$	
XGDY		$IY \rightarrow D; D \rightarrow IY$	
TSX		$SP+1 \rightarrow IX$	
TSY		$SP+1 \rightarrow IY$	
TXS		$IX-1 \rightarrow SP$	
TYS		$IY-1 \rightarrow SP$	
PSHA		$A \rightarrow \text{Стек}; SP \leftarrow SP-1$	
PSHB		$B \rightarrow \text{Стек}; SP \leftarrow SP-1$	
PSHX		$IX \rightarrow \text{Стек}; SP \leftarrow SP-2$	Сначала - младший байт
PSHY		$IY \rightarrow \text{Стек}; SP \leftarrow SP-2$	Сначала - младший байт
PULA		$A \leftarrow \text{Стек}; SP \leftarrow SP+1$	
PULB		$B \leftarrow \text{Стек}; SP \leftarrow SP+1$	
PULX		$IX \leftarrow \text{Стек}; SP \leftarrow SP+2$	Сначала - старший байт
PULY		$IY \leftarrow \text{Стек}; SP \leftarrow SP+2$	Сначала - старший байт

2. Арифметические, логические и специальные команды			
Мнемокод	Адресация	Операция	Примечание
ABA		$A + B \rightarrow A$	
ABX		$IX + 00.B \rightarrow IX$	
ABY		$IY + 00.B \rightarrow IY$	
ADDA	H8, П0, П, OXY	$A + M \rightarrow A$	
ADDB	H8, П0, П, OXY	$B + M \rightarrow B$	
ADCA	H8, П0, П, OXY	$A + M + C \rightarrow A$	
ADCB	H8, П0, П, OXY	$B + M + C \rightarrow B$	
ADDD	H16, П0, П, OXY	$D + M.M+1 \rightarrow D$	
SUBA	H8, П0, П, OXY	$A - M \rightarrow A$	
SUBB	H8, П0, П, OXY	$B - M \rightarrow B$	
SUBD	H16, П0, П, OXY	$D - M.M+1 \rightarrow D$	
SBA		$A - B \rightarrow A$	
SBCA	H8, П0, П, OXY	$A - M - C \rightarrow A$	
SBCB	H8, П0, П, OXY	$B - M - C \rightarrow B$	
MUL		$A \times B \rightarrow D$	
IDIV		$D \div IX \rightarrow IX; r \rightarrow D$	Целочисленное деление
FDIV		$D \div IX \rightarrow IX; r \rightarrow D$	Дробное деление
INCA		$A + 1 \rightarrow A$	
INCB		$B + 1 \rightarrow B$	

2. Арифметические, логические и специальные команды			
Мнемокод	Адресация	Операция	Примечание
INC	П, ОХУ	$M + 1 \rightarrow M$	
INS		$SP + 1 \rightarrow SP$	
INX		$IX + 1 \rightarrow IX$	
INY		$IY + 1 \rightarrow IY$	
DECA		$A - 1 \rightarrow A$	
DECB		$B - 1 \rightarrow B$	
DEC	П, ОХУ	$M - 1 \rightarrow M$	
DES		$SP - 1 \rightarrow SP$	
DEX		$IX - 1 \rightarrow IX$	
DEY		$IY - 1 \rightarrow IY$	
ANDA	H8, П0, П, ОХУ	$A \& M \rightarrow A$	
ANDB	H8, П0, П, ОХУ	$B \& M \rightarrow B$	
BITA	H8, П0, П, ОХУ	$A \& M$	Тест битов
BITB	H8, П0, П, ОХУ	$B \& M$	Тест битов
ORAA	H8, П0, П, ОХУ	$A \vee M \rightarrow A$	
ORAB	H8, П0, П, ОХУ	$B \vee M \rightarrow B$	
EORA	H8, П0, П, ОХУ	$A \oplus M \rightarrow A$	
EORB	H8, П0, П, ОХУ	$B \oplus M \rightarrow B$	
CBA		$A - B$	Сравнение А и В
CMPA	H8, П0, П, ОХУ	$A - M$	Сравнение А с памятью
CMPB	H8, П0, П, ОХУ	$B - M$	Сравнение В с памятью
CPD	H16, П0, П, ОХУ	$D - M.M+1$	Сравнение D с памятью 16 бит
CPX	H16, П0, П, ОХУ	$IX - M.M+1$	Сравнение IX с памятью 16 бит
CPY	H16, П0, П, ОХУ	$IY - M.M+1$	Сравнение IY с памятью 16 бит
COMA		$FFh - A \rightarrow A$	Дополнение А до “минус единицы”
COMB		$FFh - B \rightarrow B$	Дополнение В до “минус единицы”
COM	П, ОХУ	$FFh - M \rightarrow M$	Дополнение байта памяти до “минус 1”
NEGA		$0 - A \rightarrow A$	Дополнение А до нуля
NEGB		$0 - B \rightarrow B$	Дополнение В до нуля
NEG	П, ОХУ	$0 - M \rightarrow M$	Дополнение байта памяти до нуля
TSTA		$A - 0$	Тест на $A \leq 0$
TSTB		$B - 0$	Тест на $B \leq 0$
TST	П, ОХУ	$M - 0$	Тест на $M \leq 0$

2. Арифметические, логические и специальные команды			
Мнемокод	Адресация	Операция	Примечание
ASLA		Арифметический сдвиг левый	$b_0 \leftarrow 0;$
ASLB			$b_7 \leftarrow b_7;$
ASL	П, OXY		$C \leftarrow b_7$
ASLD		Двойной арифметический сдвиг левый D	$d_0 \leftarrow 0; d_{15} \leftarrow d_{15}$ $C \leftarrow d_{15}$
LSLA		Логический сдвиг левый	$b_0 \leftarrow 0;$
LSLB			$b_7 \leftarrow b_6;$
LSL	П, OXY		$C \leftarrow b_7$
LSLD		Двойной логический сдвиг левый D	$C \leftarrow d_{15}; d_{15} \leftarrow d_{14}$ $d_0 \leftarrow 0;$
LSRA		Логический сдвиг правый	$0 \rightarrow b_7;$
LSRB			$b_7 \rightarrow b_6;$
LSR	П, OXY		$b_0 \rightarrow C$
LSRD		Двойной логический сдвиг правый D	$0 \rightarrow d_{15}; d_{15} \rightarrow d_{14};$ $d_0 \rightarrow C;$
ROLA		Циклический сдвиг левый	$b_0 \leftarrow C;$
ROLB			$b_7 \leftarrow b_6;$
ROL	П, OXY		$C \leftarrow b_7$
RORA		Циклический сдвиг правый	$C \rightarrow b_7;$
RORB			$b_7 \rightarrow b_6;$
ROR	П, OXY		$b_0 \rightarrow C$
BSET	П, OXY	$M \vee mm \rightarrow M$	Устанавливает биты по маске
BCLR	П, OXY	$M \& mm \rightarrow M$	Сбрасывает биты по маске
DAA		Десятичная коррекция	аккумулятора A

3. Команды передачи управления			
Мнемокод	Условие перехода	Операция	Примечание
JMP	Безусловно	Переход	П, OXY
JSR	Безусловно	Вызов подпрограммы	ПО, П, OXY
SWI		Программное прерывание	Вектор FFF6h
BSR	Безусловно	Ветвление на подпрограмму	OPC
RTS	Безусловно	Возврат из подпрограммы	
RTI	Безусловно	Возврат из прерывания	
BRA	Безусловно	Ветвление всегда	OPC
BRN	Безусловно	Ветвление никогда	OPC
BCC	$C = 0$	Ветвление, если флаг C сброшен	OPC
BCS	$C = 1$	Ветвление, если флаг C установлен	OPC
BEQ	$Z = 1$	Ветвление, если	OPC

3. Команды передачи управления			
Мнемокод	Условие перехода	Операция	Примечание
		результат = 0	
BGE	$N \oplus V = 0$	Ветвление, если результат $\geq 0$	ОПС
BGT	$Z \vee (N \oplus V) = 0$	Ветвление, если результат $> 0$	ОПС
BHI	$C \vee Z = 0$	Ветвление, если результат $>$	ОПС
BHS	$C = 0$	Ветвление, если результат $\geq$	ОПС
BLE	$Z \vee (N \oplus V) = 1$	Ветвление, если результат $\leq 0$	ОПС
BLO	$C = 1$	Ветвление, если результат $<$	ОПС
BLS	$C \vee Z = 1$	Ветвление, если результат $\leq$	ОПС
BLT	$N \oplus V = 1$	Ветвление, если результат $< 0$	ОПС
BMI	$N = 1$	Ветвление, если результат $< 0$	ОПС
BNE	$Z = 0$	Ветвление, если результат $\neq 0$	ОПС
BPL	$N = 0$	Ветвление, если результат $> 0$	ОПС
BVC	$V = 0$	Ветвление, если не было переполнения	ОПС
BVS	$V = 1$	Ветвление, если было переполнение	ОПС
BRCLR	$M \& mm = 0$	Ветвление, если биты сброшены	ПО, ОХУ
BRSET	$M \& mm = 1$	Ветвление, если биты установлены	ПО, ОХУ

4. Системные команды		
Мнемокод	Операция	
CLC	$0 \rightarrow C$	Сброс флага переноса
CLI	$0 \rightarrow I$	Очистка маски прерываний
CLV	$0 \rightarrow V$	Сброс флага переполнения
SEC	$1 \rightarrow C$	Установка флага переноса
SEI	$1 \rightarrow I$	Установка маски прерываний
SEV	$1 \rightarrow V$	Установка флага переполнения
STOP	Останов внутреннего таймера	
TEST	Счет шины адреса (только в режиме теста)	
WAI	Ожидание прерывания - сохранить регистры в стеке и ждать	
NOP	Нет операции	

