



## 1.8 Возможность расширения программного обеспечения на MSX-компьютерах

Удачное аппаратное и программное решение определили популярность MSX-компьютеров в нашей стране. Однако широкие возможности MSX практически не описаны в отечественной литературе.

—В. Лашук

В подготовке этого материала принимали непосредственное участие: В.С.Лашук (Владимирский государственный педагогический институт), Ю.В.Юзифович (учащийся 8-го класса, г.Куйбышев).

### 1.8.1 Подробная карта памяти

Самый простой способ заново сложить дорожную карту — это сложить ее по-другому.

—Из дорожных правил Джоунса

Каждый, кому приходится работать с большим объемом информации, неизбежно сталкивается с проблемой нехватки памяти. Обычно считается, что объем доступной для пользователя памяти на MSX-компьютерах (RAM) составляет всего 32 Кбайта. Однако оказывается, что действительный объем RAM гораздо больше: 64 Кбайта на компьютерах серии MSX-1 и 134 Кбайта на компьютерах серии MSX-2 !

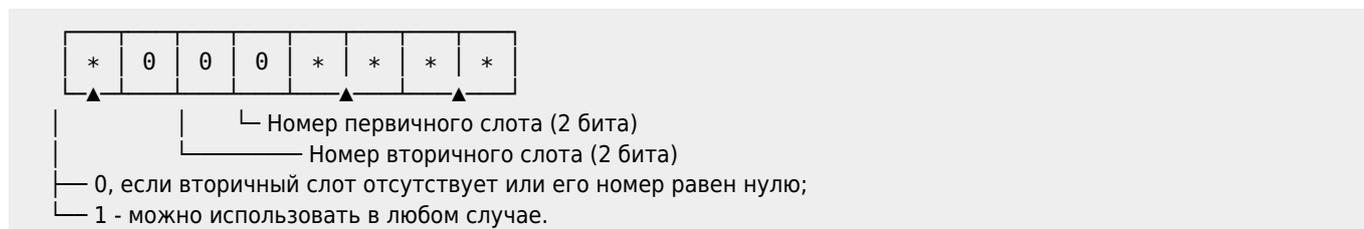
Вся память хранится в особых устройствах, называемых *слотами* («slot»—«позиция», «поле», «участок»). На схеме изображена слотовая карта памяти компьютера серии MSX-1 (ROM локальной сети располагается в слоте 3-1 по адресам 4000h÷53FFh):

Адреса	Слот 3					
	Слот 0	Слот 1	Слот 2	Слот 3-0	Слот 3-1	Слот 3-2   Слот 3-3
0000h 3FFFh (16 Кбайтов)	ROM BASIC					RAM
4000h 7FFFh (16 Кбайтов)	ROM BASIC (16 Кбайтов)	ROM диско-вода A (16 Кбайтов)	ROM диско-вода B (16 Кбайтов)	ROM диско-вода C (5 Кбайтов)	ROM локальной сети (5 Кбайтов)	ROM RAM
8000h BFFFh дополн.	дополн.	ROM дополн. дисков.	ROM дополн. дисков.			RAM

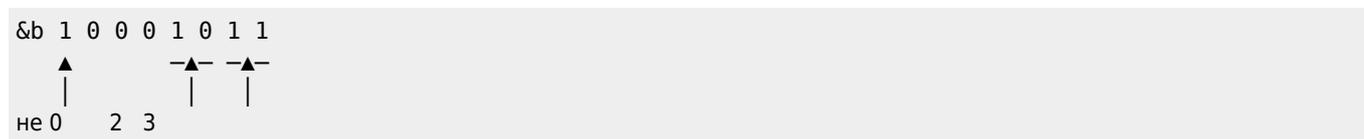


Итак, в MSX-компьютерах имеется *четыре* слота. В свою очередь, к каждому слоту можно подключить расширитель, который содержит до *четырёх* новых (*вторичных*) слотов. Обозначение «слот 3-1» подразумевает, что мы рассматриваем «подключение» к первичному слоту с номером 3 вторичного слота с номером 1.

Каждый слот имеет свою *метку (указатель)*. Указатель слота — это двоичное число, биты которого имеют следующий смысл:



Например, для слота 3-2 указатель будет таким:



Напомним Вам, что для вызова подпрограммы, написанной в машинных кодах и расположенной в некотором слоте, применяется команда ассемблера:

```
RST 30h [Указатель слота/Адрес подпрограммы]
```

На языке **MSX BASIC** эту команду можно «моделировать» следующим способом:

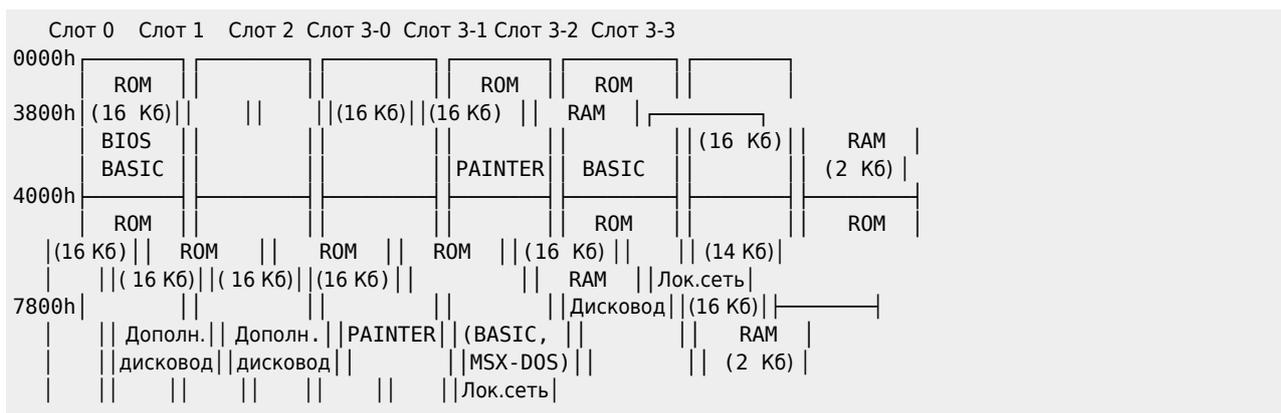
[1081-01.bas](#)

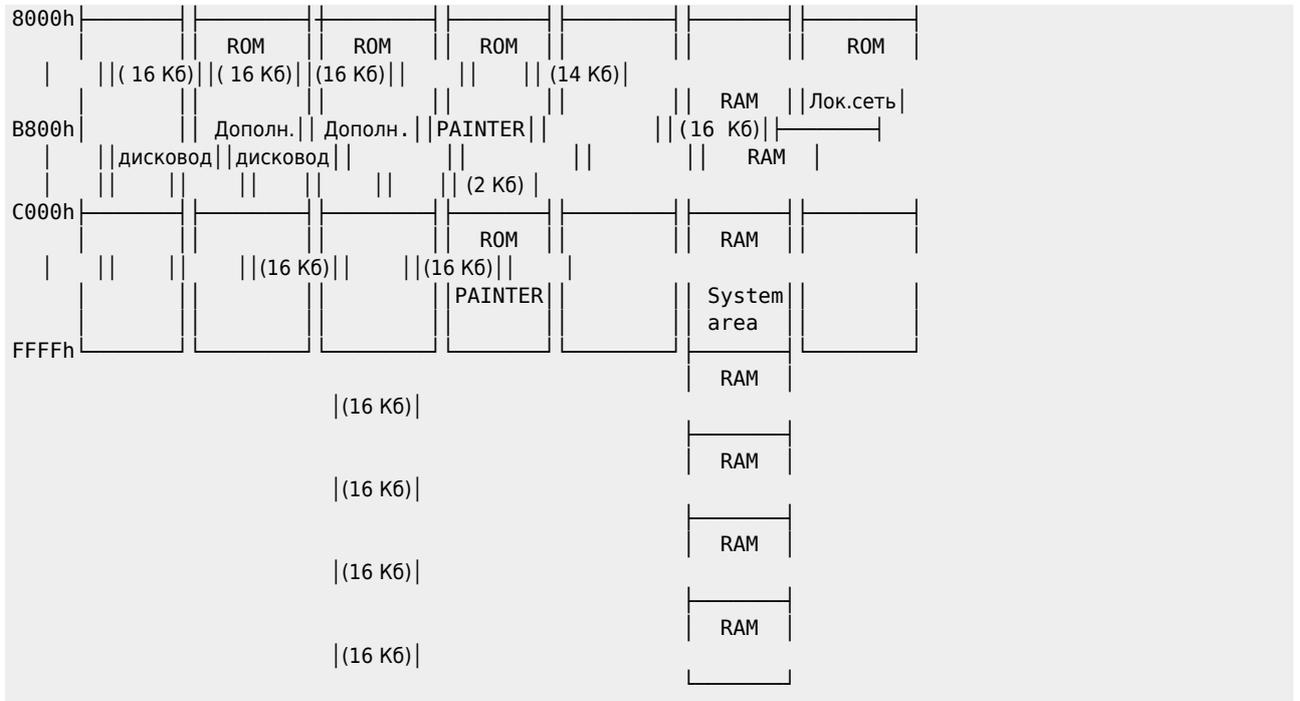
[1081-01.bas](#)

```
10 CLEAR 200,&HF300:DEFUSR=&HF300
20 INPUT"Номер первичного слота";N1
30 INPUT"Номер вторичного слота";N2
40 POKE &HF300,&HF7 ' F7 - код команды RST 30h
50 POKE &HF301,&H80+N2*4+N1 ' Указатель слота
60 INPUT"Адрес ячейки";AD%
70 POKE &HF302,PEEK(VARPTR(AD%)):POKE &HF303,PEEK(VARPTR(AD%)+1)
80 POKE &HF304,&HC9:A=USR(A) ' Код команды RET и запуск программы
```

Покажем расположение памяти в слотах для компьютеров серии MSX-2:

- а) *учительский компьютер*





Записав в соответствующий порт номер переключаемой страницы можно установить эту страницу для доступа, например:

&hFB	→	порт с адресом &hFC
&hFA	→	порт с адресом &hFD
&hF9	→	порт с адресом &hFE
&hF8	→	порт с адресом &hFF

В отключаемых страницах данные сохраняются!

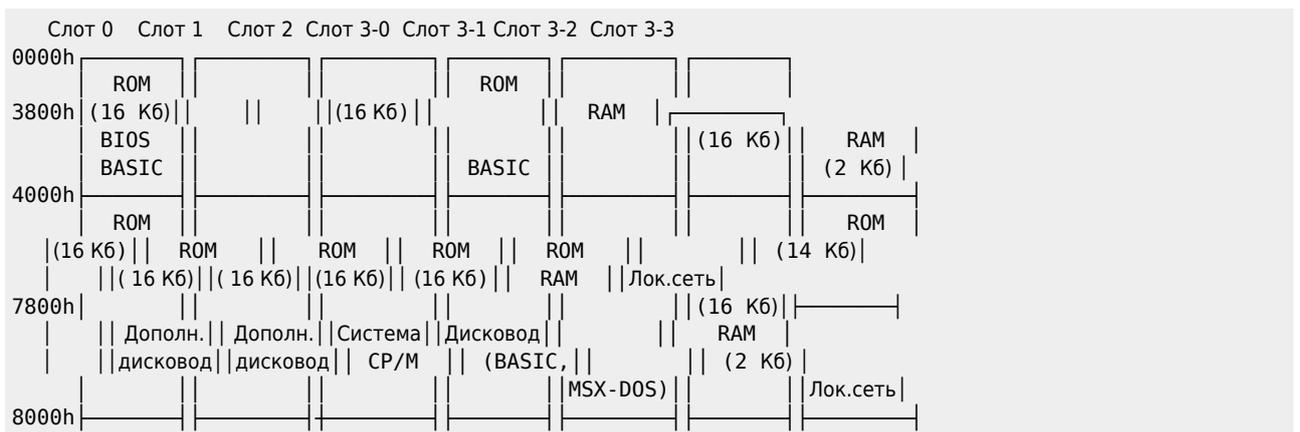
#### Логические страницы

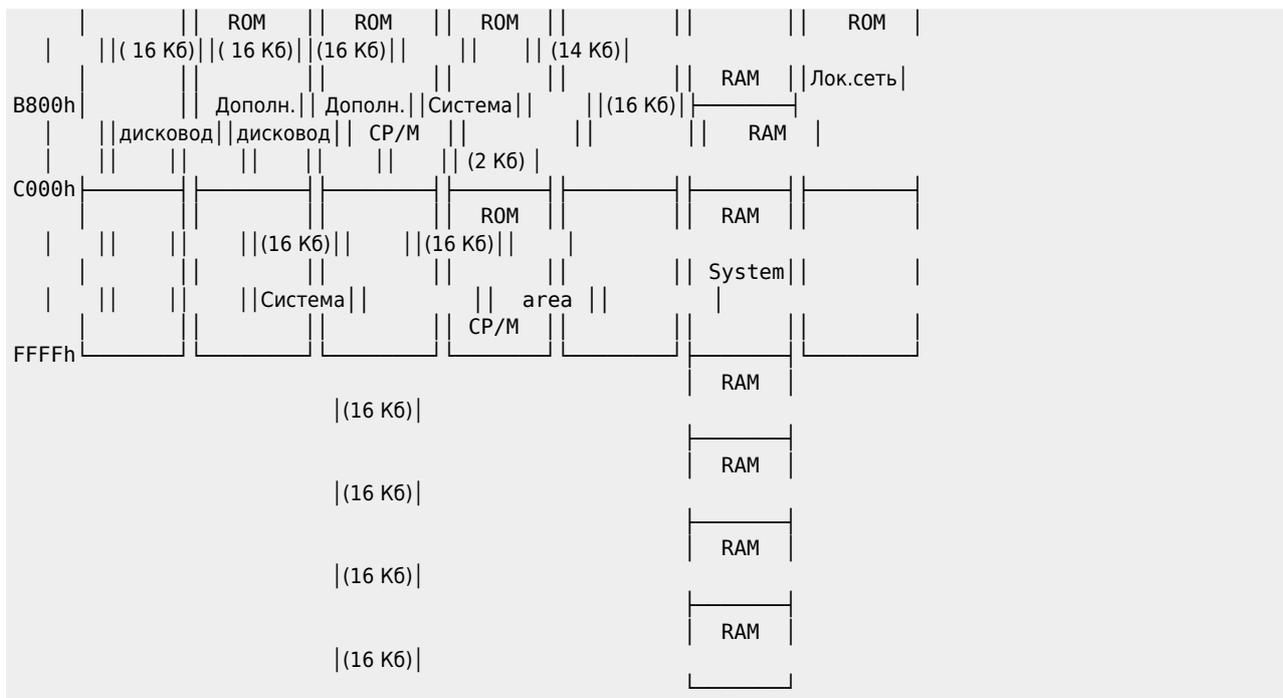


&hC000- &hFFFF	&h8000- &hBFFF	&h4000- &h7FFF	&h0000- &h3FFF				
Номера страниц			RAM	слота 3-2			
&hF8	&hF9	&hFA	&hFB	&hFC	&hFD	&hFE	&hFF
0	1	2	3	4	5	6	7

Физические страницы слота 3-2 и их номера

- в) *ученический* компьютер





Нумерация физических и логических страниц слота 3-2, а также адреса соответствующих портов в ученическом компьютере аналогичны учительскому компьютеру.

Таким образом, непосредственным сложением нетрудно получить, что общий объем памяти MSX-компьютеров составляет:

Компьютер	ROM	RAM
MSX-1 (без дисководов)	37 Кбайтов	64 Кбайта
MSX-2 (ученический)	92 Кбайта	134 Кбайта
MSX-2 (учительский)	156 Кбайтов	134 Кбайта

В слоте 0 располагаются подпрограммы BIOS, используемые как в компьютерах серии MSX-1, так и в компьютерах серии MSX-2. Дополнительные подпрограммы BIOS («расширенный BIOS»), которые используются только в компьютерах серии MSX-2, располагаются:

- для учительского компьютера - в слоте 3-1,
- для ученического компьютера - в слоте 3-0.

## 1.8.2 Работа со слотами

— Боюсь, что эти подробности утомительны, но вам не разобраться в ситуации, если вы не будете в курсе моих затруднений.

—А.Конан Дойль. Скандал в Богемии

Все адресное пространство MSX-компьютера разобьем на *четыре* т.н. *логические* страницы (объемом по 16 Кбайтов каждая).

Пронумеруем их следующим образом:

Адреса	0÷3FFFh	4000h÷7FFFh	8000h÷BFFFh	C000h÷FFFFh
--------	---------	-------------	-------------	-------------

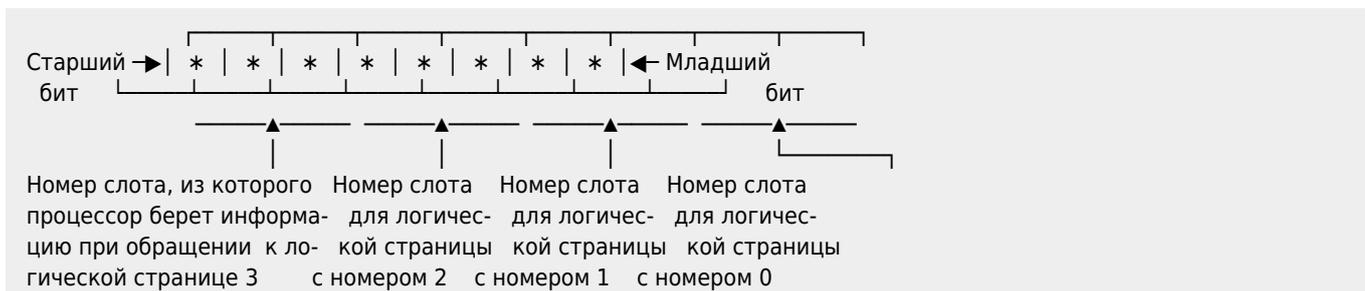
Логическая страница	0	1	2	3
---------------------	---	---	---	---

Заметим, что каждый слот также можно разбить на участки объемом по 16 Кбайтов каждый, которые мы будем называть *физическими* страницами памяти. Прежде чем начать работу с физической страницей памяти, ее необходимо *подключить* к некоторой логической странице адресного пространства, для чего выполнить следующие операции:

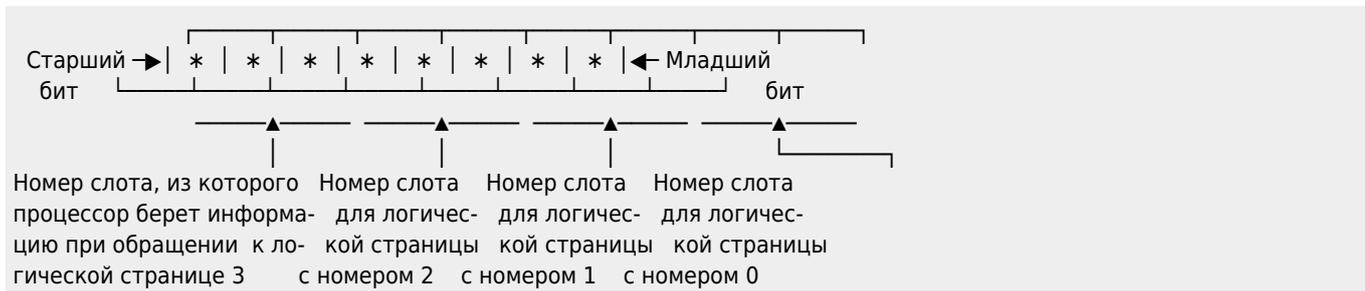
- α) указать компьютеру *номер* подключаемой логической страницы (тем самым мы присвоим физической странице некоторый начальный адрес). Заметим, что при включении компьютера некоторым физическим страницам памяти начальный адрес присваивается автоматически;
- β) указать *номера* первичного и вторичного слотов, которым принадлежит выбранная физическая страница памяти.

Номера первичных слотов записываются в порт ввода-вывода с адресом A8h, а номера вторичных слотов - в ячейку FFFFh рабочей области (слот 3-2).

Порт ввода-вывода A8h: (номера *первичных* слотов)



Ячейка памяти FFFFh: (номера *вторичных* слотов)

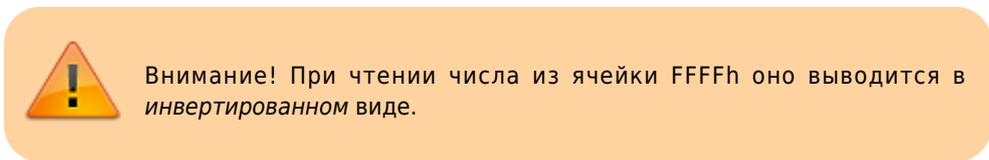


Например, если содержимое порта A8h равно &B11001001, а содержимое ячейки FFFFh равно &B01101100, то в этом случае:

- логическая страница 3 рассматривается как участок памяти из слота 3-1,
- логическая страница 2 рассматривается как участок памяти из слота 0-2,
- логическая страница 1 рассматривается как участок памяти из слота 2-3, а
- логическая страница 0 рассматривается как участок памяти из слота 1-0.

Теперь ясно, что подпрограмма подключения памяти к адресному пространству выглядит следующим образом:

```
DI      ; Запретить прерывания при работе с портами ввода-вывода
LD      A, Номера первичных слотов
OUT     (A8h),A
LD      A, Номера вторичных слотов
LD      (FFFFh),A
EI      ; Разрешить системные прерывания
RET
```



Пример 1.

[1082-01.bas](#)

 [1082-01.bas](#)

```
10 POKE &HFFFF, &B01101100
20 PRINT BIN$(PEEK(&HFFFF))
run
10010011
Ok
```

Теперь для того, чтобы узнать, какие вторичные слоты подключены к адресному пространству, необходимо инвертировать полученный результат (заменить нули — единицами, а единицы — нулями).



Отметим, что при включении компьютера в логические страницы 0 и 1 «помещаются» физические страницы памяти из слота 0, а в логические страницы 2 и 3 «помещаются» физические страницы памяти из слота 3-2.

Содержимое порта ввода-вывода с адресом A8h при этом становится равным &B11110000, а истинное содержимое ячейки памяти FFFFh становится равным &B10100000 (не забудьте про инвертирование!)

Если до выключения компьютера на странице 1 слота 3-2 создать идентификатор ROM и разместить на данной странице Вашу подпрограмму, то она будет «стартовать» при включении компьютера, т.к. первое обращение идет к этой странице.

Пример 2. А теперь небольшой фрагмент на макроассемблере...

[1082-02.asm](#)

```
DEFB 0FEH ; Программа запускается с адреса &H9000
DEFW START,KONEC+END-NACH,START
ORG 09000H

CLS EQU 00C3H ; Очистка экрана
KEYOFF EQU CLS+09H ; Отключение функциональных клавиш
FORCLR EQU 0F3E9H ; Адрес цвета изображения
COLOR EQU 0062H ; Установка цвета
CONSOL EQU 00A2H ; Вывод символа на консоль.

START: LD HL,PROM ; Адрес старта реставрируемой подпрограммы
PUSH HL ; Сохранили в стеке и при встрече RET
DI ; переходим на этот адрес.
LD A,0A8H ; Организация страниц RAM и ROM:
LD (0FFFFH),A ; 0 страница (0000H-4000H) -BASIC (BIOS)
LD A,0FCH ; 1 страница (4000H-8000H) -СЛОТВ 3-2
OUT (0A8H),A ; 2 страница (8000H-C000H) -СЛОТВ 3-2
LD HL,KONEC+1 ; 3 страница (C000H-FFFFH) -СЛОТВ 3-2
LD DE,NACH ; Пересылка подпрограммы на начало
LD BC,END-NACH ; первой страницы слота 3-2
LDIR ;
KONEC: RET ;
;
ORG 04000H ;
;
NACH: DEFB "AB" ; Идентификатор псевдо-ROM
DEFW PROM ; Адрес старта подпрограммы
DEFS 12 ; Зарезервировано стандартом
PROM: XOR A ;
CALL CLS ;
CALL KEYOFF ; Установка цвета,CLS и KEYOFF
LD HL,FORCLR ; CLS:KEYOFF:COLOR 15,9,9
LD (HL),0FH ;
```

```

INC HL ;
LD (HL),09H ;
INC HL ;
LD (HL),09H ;
CALL COLOR ;
CALL INLINE ; Печать строки на мониторе
; ... Здесь может размещаться Ваша программа.
; ... Она будет работать даже при отключении
LABEL: JP LABEL ; компьютера с помощью кнопки "RESET"

INLINE: LD HL,STROKA ; Подпрограмма вывода символов на консоль
LD A,(HL) ;
LD C,A ;
OR A ;
RET Z ;
LD A,C ;
PUSH HL ;
CALL CONSOL ;
POP HL ;
INC HL ;
JR INLINE+3 ;

STROKA: DEFM "ВЫКЛЮЧИ КОМПЬЮТЕР НА МИНУТУ."
NOP
END: NOP

```

#### Внимание !



1. Логическая страница с номером 3 ни при каких условиях не должна подвергаться изменениям, так как в ней расположены подпрограммы-ловушки и область системных переменных. Нарушение этого условия приведет к отказу системы от дальнейшего выполнения программы и ее сбросу.
2. Если для физической страницы Вы используете начальный адрес 8000h, то прежде чем подключать ее к адресному пространству, позаботьтесь о перемещении программы на [MSX BASIC](#) и стека в «безопасное» место, каковым является логическая страница с номером 3. Для этого выполните программу:

```
10 POKE &HF676,1:POKE &HF677,&HC0:POKE &HC000,0:NEW
```

Существует еще один способ подключения физической страницы памяти к адресному пространству. Для этого воспользуйтесь подпрограммой BIOS с именем ENASLT, расположенной в слоте 0 по адресу 0024h.

Обращение к ней выглядит следующим образом:

```
LD A, Указатель слота
LD HL, Начальный адрес (0000h, 4000h или 8000h)
CALL ENASLT
RET
```

В качестве примера приведем программу подключения физической страницы из некоторого слота к логической странице 2 адресного пространства.

#### Пример 3.

[1082-03.bas](#)

 [1082-03.bas](#)

```

10 DATA 3E,00 : 'LD A, Указатель слота
20 DATA 21,00,80 : 'LD HL,8000h ; Начальный адрес
30 DATA CD,24,00 : 'CALL ENASLT
40 DATA C9 : 'RET
50 CLEAR 200,&HF300:DEFUSR=&HF300

```

```

60 FOR T=0 TO 8:READ Z$:POKE &HF300+T,VAL("&h"+Z$):NEXT
70 INPUT"Номер первичного слота";N1
80 INPUT"Номер вторичного слота";N2
90 POKE &HF301,&H80+N2*4+N1 ' ← Указатель слота
120 A=USR(A)

```

Кстати, подобного эффекта можно достичь путем воздействия на ячейку памяти FFFFh в слоте 3-2:

- POKE &HFFFF,&H80 — активизирован слот 3-0 (для учительского компьютера);
- POKE &HFFFF,&HA0 — активизирован слот 3-2 (по умолчанию);
- POKE &HFFFF,&HV0 — активизирован слот 3-3.



Итак, прежде чем работать с памятью, ее необходимо явно распределить в адресном пространстве компьютера!

Существует, однако, другой способ работы со слотами, заключающийся в использовании подпрограмм BIOS, расположенных в слоте 0. В этом случае проблема распределения памяти снимается с пользователя и целиком возлагается на систему.

Опишем работу с некоторыми из этих подпрограмм.

Запись числа в RAM любого слота осуществляется путем обращения к подпрограмме с именем WRSLT, расположенной по адресу 0014h.

#### Пример 4.

```

10 DATA 3E,00          : 'F300 LD  A, Указатель слота
20 DATA 21,00,00      : 'F302 LD  HL, Адрес ячейки
30 DATA 1E,00         : 'F305 LD  E, Число
40 DATA CD,14,00     : 'F307 CALL WRSLT
50 DATA C9            : 'F30A RET
60 CLEAR 200,&HF300:DEFUSR=&HF300
70 FOR T=0 TO 10:READ Z$:POKE &HF300+T,VAL("&h"+Z$):NEXT
80 INPUT"Номер первичного слота";N1
90 INPUT"Номер вторичного слота";N2
100 POKE &HF301,&H80+N2*4+N1 ' ← Указатель слота
110 INPUT"Адрес ячейки";AD%
120 POKE &HF303,PEEK(VARPTR(AD%)):POKE &HF304,PEEK(VARPTR(AD%)+1)
130 INPUT"Какое число запишете";Q%:POKE &HF306,Q%:A=USR(A)

```

Чт е н и е числа из любой физической страницы, имеющей некоторый начальный адрес, осуществляется путем обращения к подпрограмме с именем RDSLТ ,

расположенной по адресу 000Ch.

#### Пример 5.

```

10 DATA 3E,00          : 'F300 LD  A, Указатель слота
20 DATA 21,00,00      : 'F302 LD  HL, Адрес читаемой ячейки
30 DATA CD,0C,00     : 'F305 CALL RDSLТ
40 DATA 32,10,F3      : 'F308 ;Запись результата в доступную ячейку
                        ;памяти (в нашем случае &HF310)
50 DATA C9            : 'F30B RET
60 CLEAR 200,&HF300:DEFUSR=&HF300
70 FOR T=0 TO 11:READ Z$:POKE &HF300+T,VAL("&h"+Z$):NEXT
80 INPUT"Номер первичного слота";N1
90 INPUT"Номер вторичного слота";N2
100 POKE &HF301,&H80+N2*4+N1 ' ← Указатель слота
110 INPUT"Адрес ячейки";AD%
120 POKE &HF303,PEEK(VARPTR(AD%)):POKE &HF304,PEEK(VARPTR(AD%)+1)
130 A=USR(A):PRINT"Вот Ваше число: ";PEEK(&HF310)

```

Посмотрев на схему расположения памяти в слотах, Вы можете заметить, что некоторые физические страницы памяти слота 3-2 в компьютерах серии MSX-2 не имеют физических адресов. Пронумеруем все физические страницы слота 3-2 от 0 до 7 сверху вниз (в порядке следования на рисунке).

Любую из этих физических страниц можно подключить к любой логической странице адресного пространства.

\/i

При этом не рекомендуется изменять содержимое логической страницы с номером 3, так как это приводит к перезагрузке системы!

\/i-

Подключить физические страницы слота 3-2 можно путем воздействия на порты ввода-вывода с адресами 252, 253, 254 и 255. При этом, если Вы используете порт с адресом  $252+N$ , то процессор подключит распределяемую физическую страницу к логической странице с номером  $N$ .

```
Для подключения физической страницы к логической выполните команду |
                                OUT 252+N1,&hF8+N2    , |
```

```
где: N1   - номер логической страницы, |
      N2   - номер физической страницы, |
      &hF8 - число, полученное при вычитании  $256 - M$ , |
      M    - число физических страниц в слоте 3-2 . |
```

Таким образом Вы получите доступ к любой ячейке памяти в слоте 3-2 (внимательно изучите текст на схеме расположения памяти в слотах)!

В заключение отметим, что для доступа к физическим страницам памяти, подключенным к адресному пространству, применяются оператор POKE и функция PEEK. Кроме того, для работы с памятью компьютера серии MSX-2, подключенного к локальной сети, применяются операторы CALL POKE и CALL PEEK.

Если адрес принадлежит отрезку  $[\&h4000, \&h7FFF]$ , то операторы CALL POKE и CALL PEEK "работают" с памятью, которая находится в слоте 3-3 и отвечает за работу локальной сети. Во всех остальных случаях действие оператора CALL POKE совпадает с действием оператора POKE, а действие оператора CALL PEEK совпадает с действием функции PEEK.

Следует стремиться увидеть в каждой вещи то, чего еще никто не видел, и над чем еще ни кто не думал.

—Г.Лихтенберг

То, что я понял,прекрасно, из этого я заключаю, что остальное, чего я не понял, тоже прекрасно.

—Сократ

### 1.8.3. Создание новых операторов

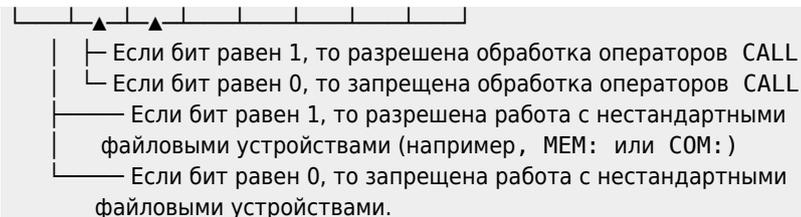
В языке MSX BASIC допускается использование так называемых «встроенных» подпрограмм. Каждая из этих подпрограмм имеет свое имя. Для обращения к встроенным подпрограммам используется оператор

```
CALL Имя подпрограммы [(Список параметров)] ,
```

где:

- CALL («call»-«звать») - служебное слово, которое можно заменять знаком «\_»;
- список параметров - необязательный список,содержащий одно или несколько выражений, отделенных друг от





Приведем теперь пример программы, позволяющей «изготовить» *новые* операторы CALL. После запуска этой программы с адреса 9000h становится возможным использование операторов

- CALL FATPRINT («утолщение» символов),
- CALL SCR0N (включение экрана дисплея),
- CALL SCROFF (выключение экрана дисплея).

Заметим, что число возможных подпрограмм для обработки нестандартных операторов ограничивается только размерами логической страницы!

Читающий, если не усвоишь - перечти,  
переждав.

—Е.Рерих. Живая вода

Итак, смотрите:

Пример.

[1083-01.asm](#)

```

;Выбор слота 3-2 для всех логических страниц памяти
9000: 3E AA      LD    A,AAh      ;Выберем 2-й слот в качестве вто-
9002: 32 FF FF    LD    (FFFFh),A      ; ричного для всех страниц
;
9005: F3        DI          ;Запретим прерывания
9006: 3E FF      LD    A,FFh      ;Выберем 3-й слот в качестве пер-
9008: D3 A8      OUT   (A8h),A      ; вичного для всех страниц
;Заполнение псевдо-ROM
900A: 21 20 90   LD    HL,9020h     ; Откуда берутся данные
900D: 11 00 40   LD    DE,4000h     ; Куда помещаются
9010: 01 00 01   LD    BC,0100h     ; Длина копируемого блока
9013: ED B0      LDIR          ; Групповая пересылка
;Восстановление конфигурации BASIC
9015: 3E F0      LD    A,F0h        ;
9017: D3 A8      OUT   (A8h),A      ;
9019: FB        EI          ;Разрешим прерывания
;Заполнение буфера SLTATR
;(64 байта, начиная с адреса FCC9h)
901A: 3E 20      LD    A,00100000b ; 5-й бит - разрешена обработка
; операторов расширения
; 6-й бит - запрещена обработка
; нестандартных устройств I/O
901C: 32 02 FD   LD    (FD02h),A    ;Псевдо-ROM в слоте 3-2, страница 1
901F: C9        RET          ;Возврат управления в BASIC
; Далее идет блок содержимого псевдо-ROM
;Заголовок псевдо-ROM ("карта ROM")
9020: 41        ; Код символа A
9021: 42        ; Код символа B
9022: 00 00     ; Адрес подпрограммы инициализации ROM
9024: 11 40     ; Адрес начала обработки операторов расширения
9026: 00 00     ; Адрес начала обработки нестандартных устройств I/O
9028: 00 00     ; Адрес текста на BASIC в ROM
902A-9030: 00 ; 7 байтов, зарезервированных стандартом
;Блок подпрограмм обработки операторов расширения
9031: 37        SCF          ;Установим флаг C (необходим для

```

```

;генерации сообщения об ошибке)
9032: E5      PUSH HL      ;
9033: 06 03    LD      B,03      ;Будут обработаны только три первых
;оператора расширения в псевдо-ROM
9035: 21 51 40  LD      HL,4051h  ;Адрес блока идентификаторов
; операторов расширения
9038: E5      PUSH HL      ;
9039: CD 38 40  CALL   4038h     ;Переход на блок сравнения 2 имен
903C: E1      POP      HL     ;
903D: 30 09    JR      NC,9048h ;Если имя не найдено, то переход на
; блок поиска адресов
903F: 11 10 00  LD      DE,0010h ;Указатель на
9042: 19      ADD     HL,DE     ; следующее имя
9043: 10 F3    DJNZ   9038h     ;Имена кончились?
9045: E1      POP      HL     ;Если имя не найдено, то возврат
9046: 37      SCF                    ; управления в BASIC с сообщением
9047: C9      RET                    ; "Syntax error"
;Блок поиска в таблице адресов
9048: 3E 03    LD      A,03     ;Число подпрограмм в псевдо-ROM
904A: 90      SUB     B        ;
904B: 21 4B 40  LD      HL,404Bh ;Начало таблицы адресов подпрограмм
944E: 87      ADD     A,A      ;Вычислим смещение
904F: 16 00    LD      D,00     ;Вычислим положение
9051: 5F      LD      E,A      ; адреса программы
9052: 19      ADD     HL,DE     ; в таблице
9053: 5E      LD      E,(HL)   ;Помещаем адрес
9054: 23      INC     HL       ; перехода из
9055: 56      LD      D,(HL)   ; таблицы в регистр DE
9056: EB      EX     DE,HL    ;Адрес -из регистра DE в регистр HL
9057: E9      JP     (HL)     ;Переход на выбранную подпрограмму
; расширения
;Сравнение вводимого имени оператора расширения
;с именем оператора в псевдо-ROM
9058: 11 89 FD  LD      DE,FD89h ;Начало области, в которой хранится
; имя набираемого оператора
905B: 1A      LD      A,(DE)   ;
905C: A7      AND     A        ;Это 0?
905D: 28 07    JR      Z,9066h  ;Если да, то произошел конец набора
; оператора с клавиатуры
905F: BE      CP     (HL)     ;Указатель на имя в псевдо-ROM
9060: 37      SCF                    ;Если имена различны, то выход в
9061: C0      RET     NZ      ; BASIC с сообщением
; "Syntax error"
9062: 23      INC     HL       ;
9063: 13      INC     DE       ;
9064: 18 F5    JR      905Bh    ;Имя кончилось?
9066: BE      CP     (HL)     ;Имя в псевдо-ROM тоже кончилось?
9067: 37      SCF                    ;Если оно длиннее, чем набранное
9068: C0      RET     NZ      ; имя, то выход в BASIC с сообщени-
; ем "Syntax error"
9069: 3F      CCF                    ;Имена совпали, возврат без сообще-
906A: C9      RET                    ; ния об ошибке
;Таблица адресов операторов расширения
906B: 81 40    ;Адрес подпрограммы CALL FATPRINT
906D: 9C 40    ;Адрес подпрограммы CALL SCRON
906F: A1 40    ;Адрес подпрограммы CALL SCROFF
;Блок имен операторов (по 16 символов на имя)
9071: 46      ;F
9072: 41      ;A
9073: 54      ;T
9074: 50      ;P
9075: 52      ;R
9076: 49      ;I
9077: 4E      ;N

```

```

9078: 54 ;T
9079-9080: 00
9081: 53 ;S
9082: 43 ;C
9083: 52 ;R
9084: 4F ;O
9085: 4E ;N
9086-9090: 00
9091: 53 ;S
9092: 43 ;C
9093: 52 ;R
9094: 4F ;O
9095: 46 ;F
9096: 46 ;F
9097-90A0: 00
;Вот они! Наши операторы CALL !
90A1: CD 6F 00 CALL 006Fh ;Установка режима SCREEN 1
90A4: 21 00 01 LD HL,0100h ;
90A7: 01 57 07 LD BC,0757h ;
90AA: 07 RLCA ;
90AB: CD 4A 00 CALL 004Ah ;Чтение из видеопамати
90AE: 57 LD D,A ;
90AF: 1F RRA ;Утолщение символа
90B0: B2 OR D ;
90B1: CD 4D 00 CALL 004Dh ;Запись в видеопамать
90B4: 0B DEC BC ;
90B5: 23 INC HL ;
90B6: 78 LD A,B ;
90B7: B1 OR C ;
90B8: 20 F1 JR NZ,90ABh ;
90BA: E1 POP HL ;
90BB: C9 RET ;Возврат в BASIC
90BC: CD 44 00 CALL 0044h ;Включение экрана дисплея
90BF: E1 POP HL ;
90C0: C9 RET ;
90C1: CD 41 00 CALL 0041h ;Выключение экрана дисплея
90C4: E1 POP HL ;
90C5: C9 RET ;

```

[https://sysadminmosaic.ru/msx/basic\\_dialogue\\_programming\\_language/108?rev=1676478628](https://sysadminmosaic.ru/msx/basic_dialogue_programming_language/108?rev=1676478628)

2023-02-15 19:30

