MSX Article

MARMSX

The MSX Memory (II)

Summary

This article discusses on the expanded slots or sub-slots, as well as the Megaram, Mapper and SRAM.

1- Introduction

Due to the MSX limited memory of 64 KB and the evolution of the games, the developers were forced to increase somehow the amount of MSX memory. By this time, cartridges having 128 or 256 KB of ROM and and a new RAM architecture called Memory Mapper came up in order to solve the problem of large data storage. Furthermore, the new MSXs started to use more the concept of expanded slots, which the slots expansors makes use of. The slot expansion allows each slot to be expanded up to new four sub-slots each. In such way, it is possible to have 16 memory banks of 64 KB each. Figure 1 illustrate the concept of slot expansion, or sub-slots, showing the configuration of a Brazilian MSX machine.

<table>
<tr><th>Page</th><th>Slot 0.0</th><th>Slot 1.0</th><th>Slot 1.1</th><th>Slot 1.2</th><th>Slot 1.3</th><th>Slot 2.0</th><th>Slot 3.0</th></tr>
<tr><td>0</td><td rowspan="2">Main ROM</td><td></td><td rowspan="4">Cartridge “A”</td><td>Sub-ROM</td><td rowspan="4">Reserved</td><td rowspan="4">RAM and Memory Mapper 256 Kb</td><td rowspan="4">Cartridge “B”</td></tr>
<tr><td>1</td><td rowspan="2">MSX-DOS 2</td><td>FM</td></tr>
<tr><td>2</td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>3</td><td></td></tr>
</table>

Figure 1. Brazilian Expert MSX 2+ slots configuration [1].

The MSX 2 SUB-ROM illustrates well the concept of subs-slots. In order to introduce the new MSX 2 features, the ROM size increased from 32 to 48 KB. Thus, the exceeding part could not invade the RAM area on page 2, once it were destined for the Basic programs. So, the new part of 16 KB were introduced on the sub-slot 1.2, also on page 0. Notice that this solution did not hold completely the slot 1, which also may connect 3 other memories.

2. Expanded slots or sub-slots

MSX slots can be expanded up to four new sub-slots, where each one can have up to 64 KB. This theorically allows MSX to have up to 1 MB.

Now, each page holds a sub-slot intead of a slot. The port &HA8 configures the slots level, while the address &HFFFF (page 3) configure the sub-slots level (figure 2).

<table>
<tr><th>Bit</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Page</td><td colspan="2">3</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">0</td></tr>
</table>

Figure 2. Sub-slots configuration on address &HFFFF.

The sub-slot configuration corresponds to the slot set on page 3. For example, if page 3 is set to read slot 1, the returned value from &HFFFF corresponds to the slot 1 sub-slots configuration.

Take a look at the figure 3. If we want to configure page 2 to access the sub-slot 1.1 (Game 2), first we must change page 3 to access slot 1 (primary slot for 1.1). Then, we have to set the sub-slot 1 on page 2. Finally, we must return the original slot on page 3. The next Assembly program shows how to do it.

```
LD A,&B01110000      ; Set page 3 to primary slot 1
OUT (&HA8),A         ;
LD HL,&HFFFF         ;
LD (HL),&B00010000 ; Game 2 is located at the expansor's sub-slot 1
LD A,&B11010000      ; Return page 3 to RAM and
OUT (&HA8),A         ; set page 2 to primary slot 1
```

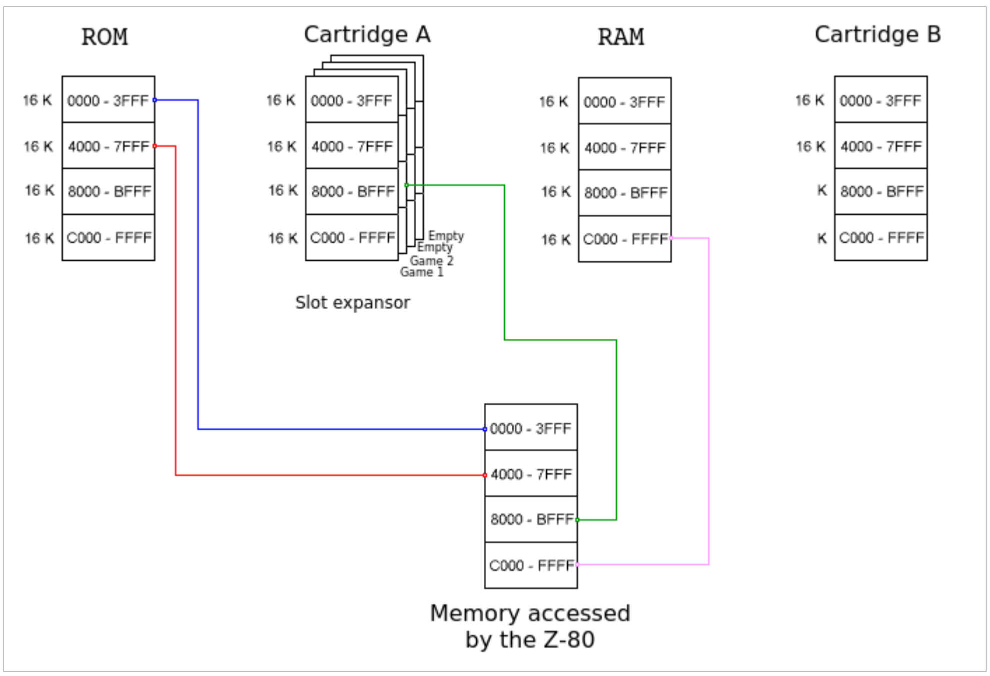

Figure 3. An example of MSX slots configuration.

Remember that the program cannot run from the same pages that will be modified.

3- Memory Mapper

This new architecture allows RAM memory to be expanded up to 4 MB, holding only one sub-slot. An internal switching schema makes possible the access of a maximum 256 banks of 16 KB [1].

Memory Mapper access is controlled using 4 ports: &HFC, &HFD, &HFE, &HFF. Each one coorresponds to a RAM page, indicating the number of current Memory Mapper bank set for that page. So, the port &HFC indicates the bank number for page 0, &HFD for the page 1, &HFE for the page 2 and &HFF for the page 3.

Figure 4 shows an example of a 256 KB Memory Mapper configuration.

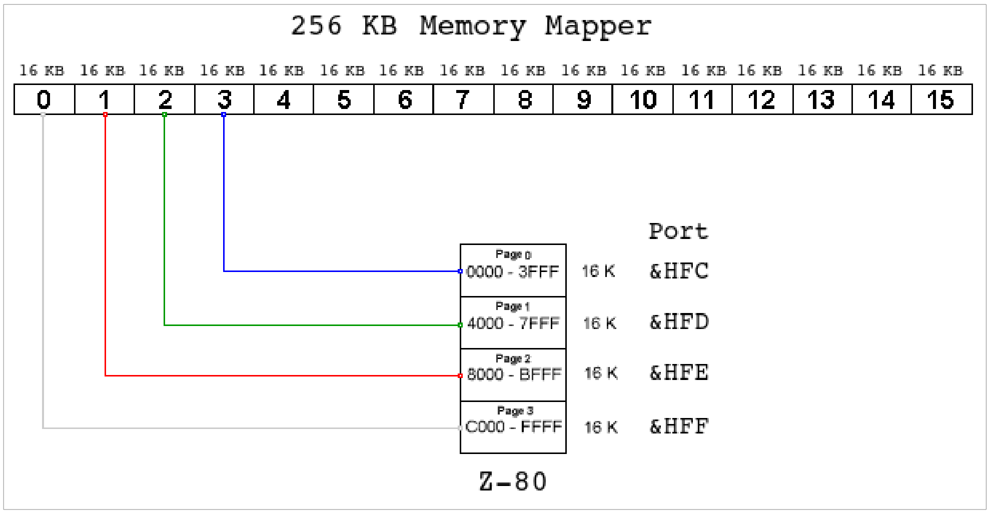

Figure 4. Example of Memory Mapper configuration.

For machines with internal Memory Mapper, the 64 KB memory is replaced by a Memory Mapper. So, when the MSX starts, the initial configuration for that Mapper [1] is shown on figure 4.

Memory Mapper is limited to 4 MB, once it is possible to address until 256 pages of 16 KB, resulting on 4 MB.

It is possible to add more than one Memory Mapper to the system, each one using a different sub-slot. The system recognizes automatically the memories and correctly numbers the pages. For example, if two Mappers of 256 KB are connected to the system, the MSX will recognize the 512 KB and give the bank numbers from 0 to 15 to Mapper 1 and 16 to 31 to Mapper 2 [1].

The following program in Basic illustrates the bank switch on Memory Mapper. Try it on a machine that has a Memory Mapper.

```
10 rem Pagina 1
```

Write the following line:

```
OUT &HFE,5
```

Type the Basic command “list”. What happened? The program disappeared. Once we changed the page 2 to access the Memeory Mapper bank #5 instead of #1, this new memory is “empty”. To return to the original bank, type:

```
OUT &HFE,1
```

Try to list the program again. Voilà, the program is back.

It is possible to create many programs in Basic on page 2 using different banks.

Finding out the Mapper size

When a Memory Mapper port is read, the number of the current bank is combined with the total number of banks. The highest numbers tell us the total number of the banks, while the lowest numbers hold the current bank for that page.

Mapper capacity is calculated using the following program in Basic, where the Mapper were not modifyed yet:

```
PRINT 256 – INP(&HFF)
```

Table 1 presents the common Mapper sizes and the respective values read from a given Mapper port.

Size (bytes)	Blocks	Returned value from ports &HFC - &HFF
256	16	&B1111xxxx
512	32	&B111xxxxx
1024	64	&B11xxxxxx
2048	128	&B1xxxxxxx
4096	256	&Bxxxxxxxx

Table 1. Returned values from ports &HFC - &HFF and the respect memory sizes.

Once we calculate the Mapper size, we can stablish the number of bits that represents the number of the bank. For example, the 256 KB Mapper uses 4 bits while the 4096 KB Mapper uses 8 bits. The “x” symbols on table 1 depicts the bits used to represent a bank.

The following program in Basic prints which bank is being used by each page.

```
10 M = (NOT INP(&HFF)) AND &HFF
20 FOR P=0 TO 3
30 PRINT"Page";P;"is set to bank"; INP(&HFC+P) AND M;"from Mapper."
40 NEXT P
```

4- MegaROM and MegaRAM

MagaROM is a cartrige with memory greater than 64 KB, created to hold games with higher memory use. MegaRAM is a RAM version of such cartridges. Both memories are accessed through the correspond slot that they were connected.

This memory works similar to the Memory Mapper, once it switch internally memory banks. Nevertheless, only pages 1 and 2 are available. In addition, it can make use of 8 KB banks instead of 16 KB. For 8 KB banks, each page is divided into 2 parts. The maximum memory available is 2 MB (8 KB banks) or 4 MB (16 KB banks)

Figure 5 describes the MagaROM or MegaRAM memory structure.

Page 0	0000 - 3FFF	ROM BIOS
Sub-page 1.0	4000 - 5FFF	MegaROM or MegaRAM
Sub-page 1.5	6000 - 7FFF	
Sub-page 2.0	8000 - 9FFF	
Sub-page 2.5	A000 - BFFF	
Page 3	C000 - FFFF	RAM Memory

Figure 5. 8 KB bank MegaROM and MegaRAM schema.

Most of MegaROM are 8 KB type. Thus, some games like R-Type use 16 KB. In that case, the page 1 and 2 is not divided.

Four registers controls the MegaROM active page. They are accessed when we write some data in any sub-page that we want to modify. For example, if we want to change the active block on sub-page 1.0 to #7, we must write the value 7 in any address from &H4000 to &H5FFF. The data reading is done naturally reading the desired address.

Once MegaRAM is writable, an aditional register is necessary to allow writing on memory. This register controls two access modes:

- Mode 0 - MegaROM, where the write changes the bank of MegaROM / MegaRAM.
- Mode 1 - MegaRAM, where the write changes a value on memory address.

To change the mode:

- Mode 0 – send any value to port &H8E.
- Mode 1 – read a data from port &H8E.

The following example shows how to write the value &H45 on MegaRAM bank #20 address &H10. The sub-page 2.0 is used to achieve that. So, the absolute address to be written is &H8000 + &H10 = &H8010.

```
OUT (&H8E),A      ; Set mode 0 to select the block
LD HL,&H8000      ;
LD (HL),20        ; Block 20 at 2.0
IN A,(&H8E)       ; Set mode 1 to start to write
LD (&H8010),&H45  ; Write data
```

MegaRAM Disk is a MegaRAM that includes the MSX-DOS operating system inside a 16 KB ROM and can also behave like a disk-drive. To access the ROM, send an OUT to &H8F, and the RAM, send an out to &H8E.

5- SRAM

SRAM is a memory designed to hold data information on cartridges, even when the MSX is turned off. We can find SRAM, for example, in FM-PAC cartridge.

In order to make use of SRAM, the game must be designed for it. Dragon Slayer 6 is an example of a game that is compatible with SRAM and can save game progress on it. This is a interesting alternative, once some games consumes a whole 360 or 720 KB diskette to save data.

SRAM has 8 KB and it is divided into 8 banks of 1 KB each. To access SRAM, change active slot on page 1 to the corresponding SRAM slot and then send the following data to memory:

```
Address   Value
 &H5FFE   &H4D
 &H5FFF   &H69
```

After that, the SRAM is enabled and can be accessed from &H4000 to &H5FFF. If you want to access FM-PAC ROM, change the addresses &H5FFE or &H5FFF to values different from that used to activate SRAM.

6- Credits and references

This article was originally written in portuguese and translated by Marcelo Silveira, Systems and Computer Engineer.

Written in: May, 2004.
Revised on: July, 2017.
E-mail: flamar98@hotmail.com
Homepage: http://marmsx.msxall.com

References:

[1] - CPU MSX número 30, Bonus Rio editora, Artigo de Julio Marchi e André Tupinambá.
[2] – MSX 2 Technical Handbook, ASCII Corporation – Japan, março de 1987.
[3] - Livro Vermelho do MSX, editora Mc Graw Hill.
[4] - CPU MSX número 36, Bonus Rio editora, Artigo de R. Pontes e Roberto Silva.
[5] - CPU MSX número 33, Bonus Rio editora, Artigo de Julio Marchi e André Tupinambá.
[6] - MegaROMs em http://www.msxnet.org/tech/megaroms.