Laboratory work No 1

Investigation of 8-bit microprocessor training system

1. The aim of the work

To acquaint with the structure, control and indication elements of 8-bit microprocessor training system M85-01; to clear up functions of a system monitor.

2. Short description of microprocessor training system

2.1. System purpose

Microprocessor training system M85-01 is a single board computer based on 8-bit microprocessor Intel® 8085, which is widely used to train engineers to develop software or hardware for any industrial process and control. This system is intended for:

- for the acquaintance with the structure of the microprocessor system in practice;
- for the investigation of the microprocessor instruction set;
- for providing basic knowledge about possibilities of practical application of microprocessor systems.

2.2. System specification

Mikroprocessor type	– Intel® 8085 (improved Intel® 8080);
Clock speed	– 3,07 Mhz;
Non-volatile memory	– 8 KB;
Random access memory	– 8 KB (expandable up to 40 KB);
Input / Output lines	– 24;
Timer / Counter	– programmable 3-channel 16-bit;
Serial interface	– RS-232C;
Software	– system monitor program;

Power supply	– +5 V / 1 A and ±12 V / 250 mA;
Physical size	– 260 × 172 × 88 mm;
Operating temperature	– 0 to 50 ^{o}C

2.3. System capabilities

- Examine the contents of any memory location.
- Examine or modify the contents of any of the microprocessor internal register.
- Modify the contents of any of the random access memory location.
- Move a block of data from one location to another location.
- Insert one or more instructions in the user program.
- Delete one or more instructions from the user program.
- Relocate a program written for some memory area to some other memory area.
- Find out a string of data lying at a particular address.
- Fill a particular memory area with a constant.
- Compare two blocks of memory.
- Execute a program at full clock speed.
- Execute a program in single step mode i. e. instruction by instruction.

2.4. Control and indication elements

The microprocessor training system M85-01 has 28 keys and six digits of seven segment display to communicate with the outside world (Fig. 1).

Four most significant digits of seven segment display are intended for displaying the address of any memory location or name of microprocessor internal register, whereas the rest of the two digits are meant for displaying the contents of a memory location or of a register. All the six digits of the display are in hexadecimal notation.

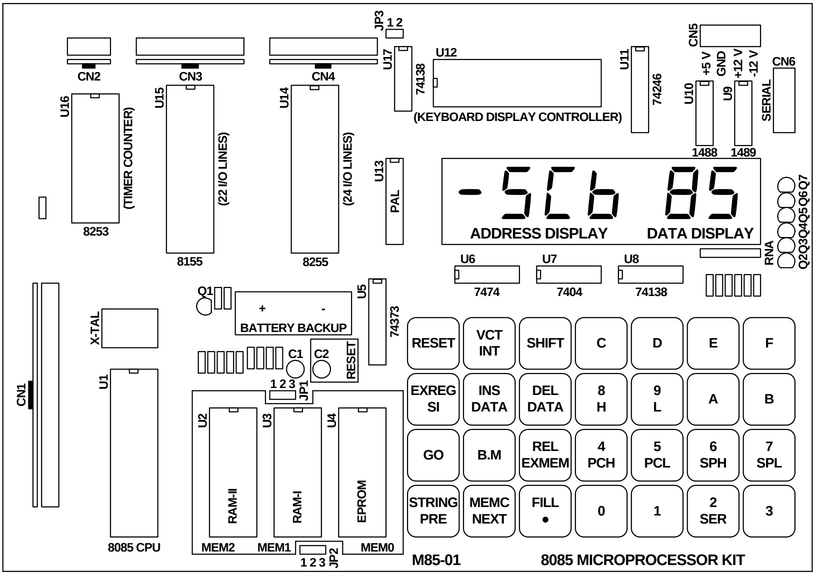


Fig. 1. Control and indication elements of microprocessor training system M85-01

The microprocessor training system M85-01 accepts all data and addresses in hexadecimal form as given in the Table 1.

Table 1. Displaying of hexadecimal digits

Hexadecimal	Decimal	Binary	Seven segment display
0	0	0000	0
1	1	0001	1
2	2	0010	2
3	3	0011	3
4	4	0100	4
5	5	0101	5
6	6	0110	6
7	7	0111	7

Table 1 continuation.

8	8	1000	8
9	9	1001	9
A	10	1010	A
B	11	1011	b
C	12	1100	C
D	13	1101	d
E	14	1110	E
F	15	1111	F

Microprocessor training system M85-01 is controlled by a keyboard, which is formed from groups of informational and command keys.

A group of informational keys is comprised of hexadecimal 4×4 matrix, which intended for the input of data and addresses. The names of microprocessor internal registers are also indicated as group keys.

0	Hexadecimal 0.
1	Hexadecimal 1.
2 **SER**	Hexadecimal 2 or switch on the serial operating mode.
3	Hexadecimal 3.
4 **PCH**	Hexadecimal 4 or higher byte of microprocessor program counter **PC**.

5 **PCL**	Hexadecimal 5 or lower byte of microprocessor program counter **PC**.
6 **SPH**	Hexadecimal 6 or higher byte of microprocessor stack pointer **SP**.
7 **SPL**	Hexadecimal 7 or lower byte of microprocessor stack pointer **SP**.
8 **H**	Hexadecimal 8 or microprocessor register **H**.
9 **L**	Hexadecimal 9 or microprocessor register **L**.
A	Hexadecimal A or microprocessor register **A** (accumulator).
B	Hexadecimal B or microprocessor register **B**.
C	Hexadecimal C or microprocessor register **C**.
D	Hexadecimal D or microprocessor register **D**.
E	Hexadecimal E or microprocessor register **E**.
F	Hexadecimal F or microprocessor flags register **F**.

A group of command keys is comprised of 12 keys. Some of these keys are double, i.e. these keys can perform two commands.

SHIFT	Provides a second level command to all keys.
VCT INT	Hardware interrupt via keyboard.
RESET	Reset the microprocessor system.
GO	To execute the program.
SI	To execute the program in single step mode.
EXREG	Examine register: allows user to examine and modify the contents of different registers.
EXMEM	Examine memory: allows user to examine any memory location and modify any random access location.
PRE	Previous is used as intermediate terminators in case of examine memory. It decrements the program counter **PC** contents and writes the contents of data field to the address displayed in the address location.
NEXT	Increment is used as a intermediate terminator in case of examine memory, examine register and etc. It increments the program counter **PC** contents and writes the data lying in data field at the location displayed at address field.

•	Terminator is used to terminate the command and write the data in data field at the location displayed in address field.
B.M	Move a block: allows user to move a block or data with relocation by one or more bytes.
REL	Relocate: relocates a program written for some memory area and to be transferred to other memory area.
FILL	Allows user to fill random access memory with a constant.
INS DATA	Insert data: inserts one or more data bytes in the user's program data area.
DEL DATA	Delete data: deletes one or more data bytes from the user's program or data area.
STRING	String: finds out the string of data lying at a particular address or addresses.
MEMC	Memory compare: compares two blocks of memory for equality.

All commands are followed by a set of numeric parameters separated by "**PREV**", "**NEXT**" and "•" to work as delimiters.

A "-" on most significant digit of address display indicates that system is waiting for a command. If, instead of a valid command, the user gives a data, the system will display "-Err" or "E". A dot on the least significant digit of address field indicates that the system expects an address. Whenever the data of any memory location is changed, a dot is displayed on the least significant digit of data field.

2.5. Memory organization

Microprocessor training system M85-01 provides 8 KB of random access memory and 8 KB of non-volatile memory. The total onboard memory can be expanded to 40 KB. There are three memory sockets provided on the board. Each socket can be defined to have any address between 0000_{16}–$FFFF_{16}$.

There are three memory sockets namely **MEM0**, **MEM1** and **MEM2** (Fig. 1).

MEM0 is defined as system monitor having 8 KB of powerful system monitor program is embedded in the EPROM. For the system operation the monitor should start from address 0000_{16}. The memory address is from 0000_{16} to $1FFF_{16}$.

MEM1 is defined as user random access memory of 8 KB. The address is from 2000_{16} to $3FFF_{16}$ for the user random access memory. The system monitor uses certain portion of this random access memory for temporary use i.e. scratch pad area. This area is from **2770_{16} to $27FF_{16}$. The user is advised not to use this area for storing data and program.**

MEM2 is defined as expansion for random access memory of 32 KB. The address is from 6000_{16} to $DFFF_{16}$.

The overall mapping of memory is given in the Table 2.

Table 2. Memory map

8 KB of non-volatile memory (**MEM0**)	Address
System monitor	0000_{16}–$1FFF_{16}$
8 KB of random access memory (**MEM1**)	Address
User field System monitor scratch pad	2000_{16}–$3FFF_{16}$ 2770_{16}–$27FF_{16}$
32 KB of random access memory (**MEM2**)	Address
User field	6000_{16}–$DFFF_{16}$

2.6. Work with the system

2.6.1. Power on

After switching on the power supply (power on switch on the rear end) a message – “-5Cb 85” will come on the microprocessor training system display. Press key “**RESET**” if you do not get “-5Cb 85”.

Attention! *Microprocessor training system can be switched on once again only after 10 seconds following the switch off at the earliest.*

2.6.2. Examine or modify register

“SHIFT” “EXREG” R [“NEXT” [D1 D2] ...] “•”,

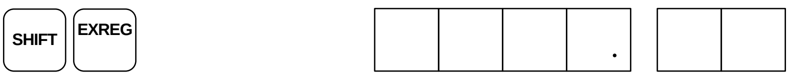
where **R** – register name; **D1**, **D2** – hexadecimal numbers which form the data byte written in register.

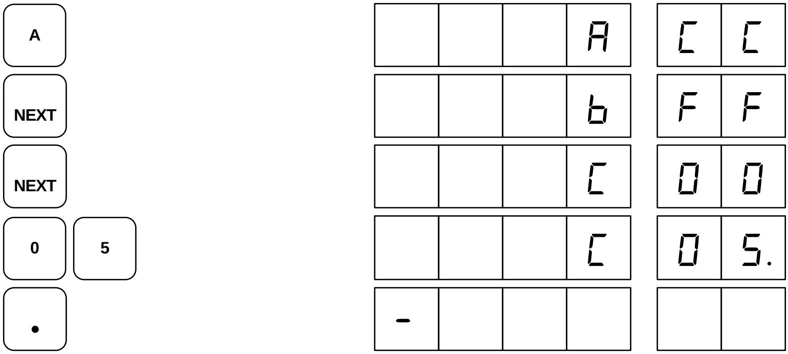
Note: in brackets indicated optional syntax elements.

This command is used to examine or modify any internal register of the microprocessor. If one wants to examine the contents of all the registers, one can start from register **A** and examine all the registers by pressing “**NEXT**” key. Whereas if some specific registers is to be examined, then the key for that register can be entered directly. The contents of any register can be changed.

For example, let us examine the contents of registers **A**, **B** and **C**, and change the content of register **C** to 05_{16}:

Keyboard Address display Data display





On pressing "**SHIFT**" and "**EXREG**" key, a dot is displayed in the address display. Enter the register name (identifier), i. e. **A**. The content of register **A** is displayed. Press "**NEXT**" key to see the content of register **B**, FF_{16} is displayed. Press "**NEXT**" key to see the content of register **C**, 00_{16} is displayed. Press "**0**" key and "**5**" key and then press "•" key to enter 05_{16} in register **B** and terminate the command.

*Note: the initial contents of the registers **A**, **B** and **C** may be different and taken for example only.*

2.6.3. Examine or modify memory

"EXMEM" A1 A2 A3 A4 "NEXT" [D1 D2]
["NEXT" [D1 D2] ...] "•",

where **A1**, **A2**, **A3**, **A4** – hexadecimal numbers which form the address of memory location; **D1**, **D2** – hexadecimal numbers which form the data byte written in memory location.

Note: in brackets indicated optional syntax elements.

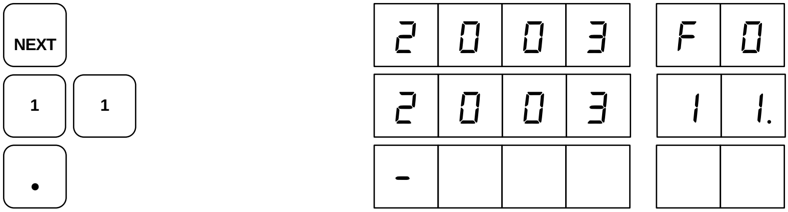
This command is used to examine the contents of any memory location and modify the contents of the random access memory area.

On pressing “**EXMEM**” key, a dot is displayed in the end of address display. On can now enter the address of any location one wants to examine. Enter the desired address and press “**NEXT**” key. The content of this location is displayed in the data display. If one wants to examine the content of next location, just press “**NEXT**” key and the address in the address display will be incremented by one and its content will be displayed in the data display. Same way if one wants to examine the content of previous location just press “**PRE**” key and the address in the address display will be decremented by one and its content will be displayed in the data display.

If one wants to modify the content of any random access location, then enter the data and press “**NEXT**” key. The data display will be written in the address displayed in the address display and simultaneously the content of next location will be displayed.

For example, let us examine the contents of memory locations at addresses 2000_{16}, 2001_{16}, 2002_{16}, 2003_{16} and change the contents of memory locations at addresses 2002_{16} and 2003_{16} to 11_{16}:

Keyboard	Address display	Data display
EXMEM	.	
2 0 0 0	2 0 0 0.	
NEXT	2 0 0 0	F 0
NEXT	2 0 0 1	F 0
NEXT	2 0 0 2	F 0
1 1	2 0 0 2	1 1.



Note: the initial contents of the memory locations at addresses 2000_{16}, 2001_{16}, 2002_{16} and 2003_{16} *may be different and taken for example only.*

2.6.4. Program execution

The user program must be entered from selected address of microprocessor training system random access memory (usually starting from the address 2000_{16}) and executed using the "**SI**" key or "**GO**". **At the end of program the RST 5 instruction (hexadecimal code EF) must be entered.**

2.6.4.1. Program execution in single step mode

"SI" A1 A2 A3 A4 "NEXT" ["NEXT" ...] "•",

where **A1**, **A2**, **A3**, **A4** – hexadecimal numbers which form the start (beginning) address of program.

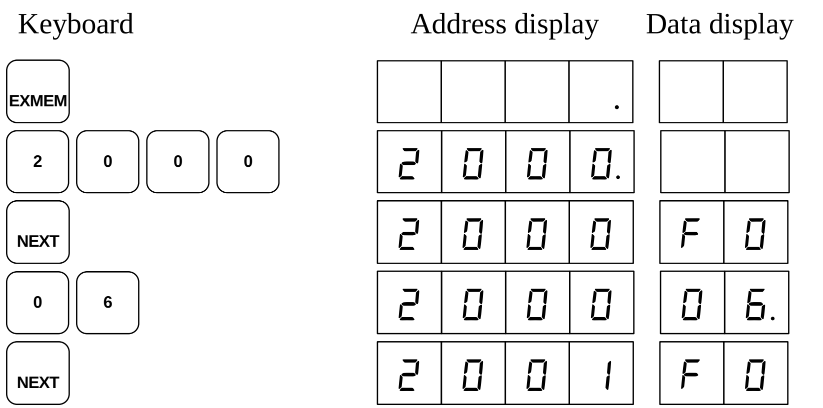
Note: in brackets indicated optional syntax elements.

This command is used to execute the program instruction by instruction. On pressing "**SI**" key, the program counter **PC** content is displayed on the address display and its data in the data display. If one wants to modify the start address of program, one can do that. After entering the start address, press "**NEXT**" key, the content of the entered address is displayed. On pressing "**NEXT**" key, first instruction of program will be executed and the address of the next instruction of program will be displayed with its data in the data display. Each time "**NEXT**" key is pressed, one instruction is executed. If one wants to terminate the command at any stage, one can do that using "•" key. On pressing "•" key, a "-" is displayed in the address display. One can now examine any internal register of microprocessor or any memory location and modify it if desired.

For example, let us enter the following program and run it in single step mode, and examine the contents of registers **A** and **B**:

Address_{16}	Instruction code_{16}	Instruction mnemonic	Comments
2000	06	**MVI B, 11**	The hexadecimal number 11 of one byte (second byte of the instruction) is moved to register B. ($11_{16} \rightarrow$ B, B = 11_{16}).
2001	11		
2002	78	**MOV A, B**	The content of register B is moved to register A (B $\rightarrow$ A, A = 11_{16}).
2003	EF	**RST 5**	Software breakpoint.

First of all, let us enter specified program to a random access memory of microprocessor training system:



1 1	2 0 0 1	1 1.
NEXT	2 0 0 2	F 0
7 8	2 0 0 2	7 8.
NEXT	2 0 0 3	F 0
E F	2 0 0 3	E F.
.	-	

Note: the initial contents of the memory locations at addresses 2000_{16}, 2001_{16}, 2002_{16} and 2003_{16} *may be different and taken for example only.*

Let us execute entered program in single step mode:

Keyboard	Address display	Data display
SI	0 0 7 d.	3 E
2 0 0 0	2 0 0 0.	
NEXT	2 0 0 0.	0 6
NEXT	2 0 0 2.	7 8

The first instruction of two bytes is executed i. e. register **B** has been loaded with hexadecimal number 11. Let us examine the content of register **B**. For this we will have to terminate the command here:

Keyboard	Address display	Data display
.	-	
SHIFT EXREG	.	
B	b	1 1
.	-	

To run the program further presses “**SHIFT**” and “**SI**”:

Keyboard	Address display	Data display
SHIFT SI	2 0 0 2.	7 8

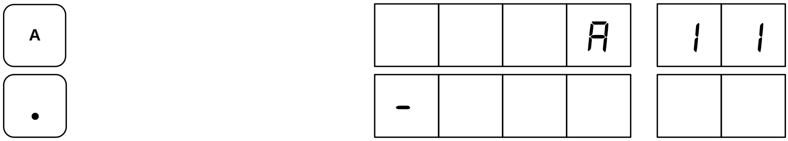
,

Program came back at the same address where we left it.

NEXT	2 0 0 3.	E F
NEXT	0 0 2 8.	F 3

Program is executed to the end. Let us examine the content of register **A**:

Keyboard	Address display	Data display
.	-	
SHIFT EXREG	.	



How you can see the program works correctly.

2.6.4.2. Program execution at full clock speed

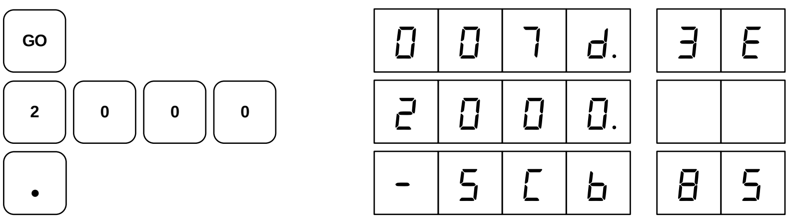
“GO” A1 A2 A3 A4 “•”,

where **A1**, **A2**, **A3**, **A4** – hexadecimal numbers which form the start (beginning) address of program.

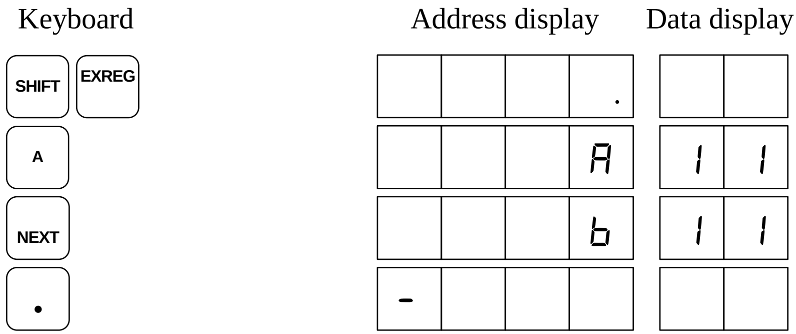
This command is used to execute the program in full clock speed. On pressing “**GO**” key, the program counter PC content are displayed in the address display with the data in the data display. A dot in the address display indicates that the address can be changed, if so desired. Enter the start address of the program. On entering this address, the data display gets blanked out. Press terminate “•” key. The microprocessor will start executing the program and “-5Cb 85” will be displayed in the display.

For example, let us execute the program entered earlier using single step execution command:

Keyboard Address display Data display



The program is executed from beginning to the end. Let us examine the contents of registers **A** and **B**:



How you can see the program works correctly.

2.6.4.3. Execution of additional functions

The microprocessor training system M85-01 has several pre-programmed functions. Let us mention some of them.

To fill a particular random access memory area with a constant:

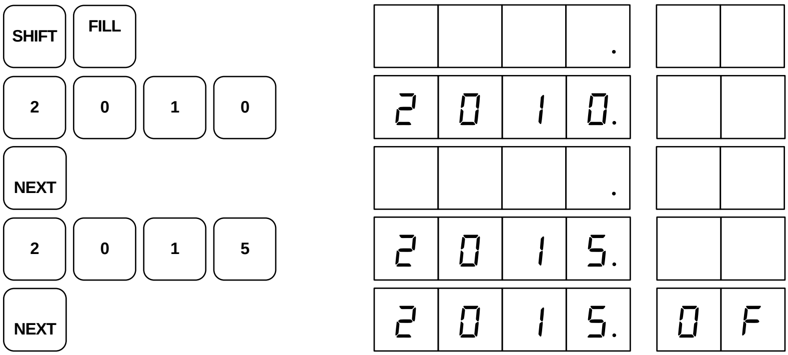
“SHIFT” “FILL” (A1 A2 A3 A4)$_1$ “NEXT” (A1 A2 A3 A4)$_2$ “NEXT” D1 D2 “•”,

where **(A1 A2 A3 A4)$_1$ –** hexadecimal numbers which form the start address of the memory area from where the data should be stored; **(A1 A2 A3 A4)$_2$ –** hexadecimal numbers which form the end address of the memory area till were the data should be stored; **D1, D2** – hexadecimal numbers which form the one byte constant with which the memory area should be filled.

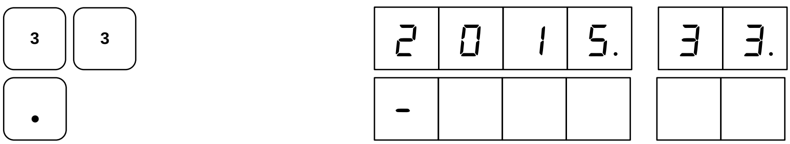
This command allows the user to fill a particular random access memory area with a constant. On pressing “**SHIFT**” key and “**FILL**” key a dot will be displayed at the end of the address display. Enter the start address and press “**NEXT**” key. Again a dot will be displayed. Now enter the end address and press “**NEXT**” key. The present contents of end address will be displayed. Enter the constant and press terminate “•” key.

Let us fill a random access memory area from 2010_{16} to 2015_{16} addresses with constant 33_{16}:

Keyboard Address display Data display

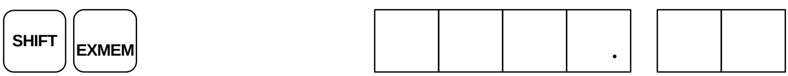


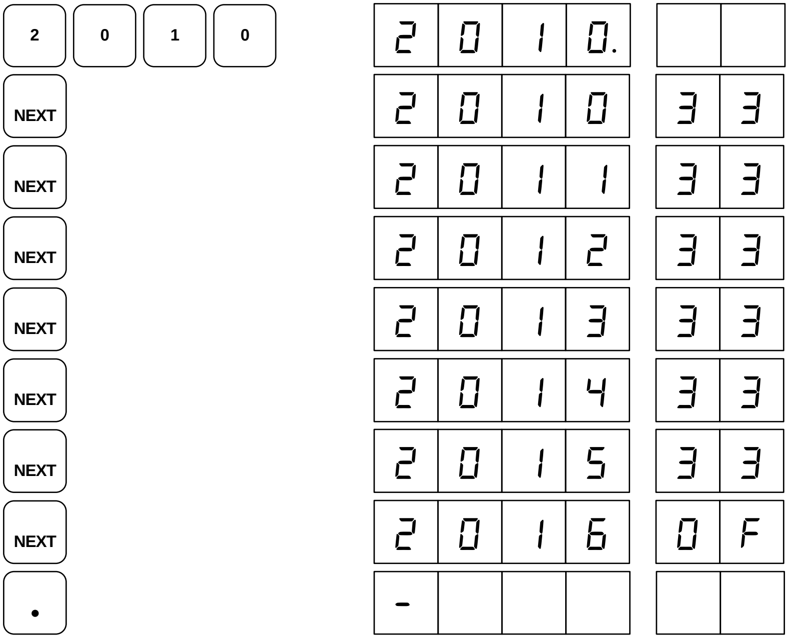
$0F_{16}$ is just some random data which was lying at address 2015_{16}. Press “**3**” key twice and press terminate “•” key.



Let us verify that 33_{16} is filled from 2000_{16} to 2015_{16} addresses:

Keyboard Address display Data display





To move a block of data from one location to another location:

“B.M” (A1 A2 A3 A4)$_1$ “NEXT” (A1 A2 A3 A4)$_2$ “NEXT” (A1 A2 A3 A4)$_3$ “•”,

where **(A1 A2 A3 A4)$_1$ –** hexadecimal numbers which form the start address of the data block which should be moved; **(A1 A2 A3 A4)$_2$ –** hexadecimal numbers which form the end address of the data block which should be moved; **(A1 A2 A3 A4)$_3$ –** hexadecimal numbers which form the destination address of moved data block.

This command allows the user to move the block of data from one memory location to another memory location. On pressing “**B.M**” key a dot is displayed at the end of address display. Enter the start address of the data block to be moved and press “**NEXT**”.

Again a dot is displayed. Now enter the end address of the data block and press “**NEXT**”. Again a dot is displayed. Now enter the destination address and press terminate “•” key. A “-” is displayed on the display.

Let us, the data block from 2010_{16} to 2015_{16} addresses which was created with using fill command, move to another memory location starting at address 2050_{16}:

Keyboard	Address display	Data display
B.M	.	
2 0 1 0	2 0 1 0.	
NEXT	.	
2 0 1 5	2 0 1 5.	
NEXT	.	
2 0 5 0	2 0 5 0.	
•	-	

Let us verify that the data block has moved to address 2050_{16}:

Keyboard	Address display	Data display
EXMEM	.	
2 0 5 0	2 0 5 0.	

NEXT	2050	33
NEXT	2051	33
NEXT	2052	33
NEXT	2053	33
NEXT	2054	33
NEXT	2055	33
NEXT	2056	0F
•	-	

To compare two blocks of memory:

“SHIFT” “MEMC” (A1 A2 A3 A4)$_1$ “NEXT” (A1 A2 A3 A4)$_2$ “NEXT” (A1 A2 A3 A4)$_3$ “•” [“NEXT” ...],

where **(A1 A2 A3 A4)$_1$ –** hexadecimal numbers which form the start address of the first data block; **(A1 A2 A3 A4)$_2$ –** hexadecimal numbers which form the end address of the first data block; **(A1 A2 A3 A4)$_3$ –** hexadecimal numbers which form the start address of the second data block.

Note: in brackets indicated optional syntax elements.

This command allows the user to compare two blocks of memory for equality. If they are not equal, the address of the first data block at which there is a discrepancy will be displayed. On pressing “**SHIFT**” key and “**MEMC**” key a dot will be displayed at the end of the address display. Enter the start address of the first data block and press “**NEXT**” key. Again a dot is displayed. Enter the end address of the first data block and press “**NEXT**” key. A dot is displayed. Now enter the starting address of the second data block and press terminate “•” key.

Let us compare following two blocks of memory for equality (assume that blocks of data have already entered into the memory):

First data block		Second data block	
Address$_{16}$	Data$_{16}$	Address$_{16}$	Data$_{16}$
2020	11	2030	11
2021	12	2031	11
2022	11	2032	11
2023	10	2033	11
2024	11	2034	11

Keyboard	Address display	Data display
SHIFT MEMC	.	
2 0 2 0	2 0 2 0.	
NEXT	.	
2 0 2 4	2 0 2 4.	
NEXT	.	
2 0 3 0	2 0 3 0.	
•	2 0 2 1	1 2
NEXT	2 0 2 3	1 0
NEXT	-	

As we can see two differing memory locations with addresses 2021_{16} and 2023_{16} were found.

When the two data blocks are identical after pressing terminate "•" key a "-" will be displayed.

3. Task

1. To switch on microprocessor training system M85-01 (power on switch on the rear end) and initiate the system monitor (press "**RESET**" key).

2. To draw a Table 3.

Table 3. Results of the registers examination and modification

Internal microprocessor register	Content of register		
	Initial	After modification	After pressing "**RESET**" key
A ... **SPL**	CC	22	CC

Names (identifiers) of registers:

A – 8-bit register **A** or accumulator;
B – 8-bit register **B**;
C – 8-bit register **C**;
D – 8-bit register **D**;
E – 8-bit register **E**;
H – 8-bit register **H**;
L – 8-bit register **L**;
F – 8-bit flags register **F**;
PCH – higher byte of the program counter **PC;**
PCL – lower byte of the program counter **PC**;
SPH – higher byte of the stack pointer **SP**;
SPL – lower byte of the stack pointer **SP**.

3. To examine and modify the contents of all registers of the microprocessor (listed above). After that press “**RESET**” key and once again examine the contents of all registers. Write the results in Table 3. Make a conclusion.

4. To draw a Table 4.

Note: instead of XX write a freely chosen numbers.

Table 4. Results of memory examination and modification

<table>
<tr><td rowspan="2">Memory location address$_{16}$</td><td colspan="3">Content of memory location</td></tr>
<tr><td>Initial</td><td>After modification</td><td>After pressing “RESET” key</td></tr>
<tr><td>20XX
...
1FXX
...</td><td>F0</td><td>33</td><td>33</td></tr>
</table>

5. To examine and modify small memory blocks of 3–5 bytes. After that press “**RESET**” key and once again examine the same memory locations. Write the results in Table 4. Make a conclusion.

6. To draw a Table 5.

Note: instead of X write a freely chosen number.

Table 5. Results of memory filling with a constant

<table>
<tr><td colspan="2">Initial data</td><td colspan="2">Memory locations after filling with a constant (command “FILL”)</td></tr>
<tr><td>Memory location address$_{16}$</td><td>Data$_{16}$</td><td>Memory location address$_{16}$</td><td>Data$_{16}$</td></tr>
<tr><td>20X0
20X1
20X2
20X3
20X4</td><td></td><td>20X0
20X1
20X2
20X3
20X4</td><td></td></tr>
</table>

7. To fill a selected random access memory area with a constant. Write the results in Table 5.

8. To draw a Table 6.

Note: instead of X write a freely chosen number.

Table 6. Results of relocation of data block

Initial data		Memory locations after relocation (command "**B.M**")	
Memory location address$_{16}$	Data$_{16}$	Memory location address$_{16}$	Data$_{16}$
20X0 **20X1** **20X2** **20X3** **20X4**		**20X0** **20X1** **20X2** **20X3** **20X4**	

9. To move the block of data from one memory location to another memory location. Write the results in Table 6.

10. To compare two selected blocks of memory for equality. Write down all necessary commands.

11. To enter the following program and execute it at full clock speed. Write down all necessary commands.

Label	Address$_{16}$	Instruction code$_{16}$	Instruction mnemonic	Comments
	2000	31	**LXI SP, 20FF**	Initialize stack pointer.
	2001	FF		
	2002	20		
	2003	CD	**CALL CLEAR**	Clear the display.
	2004	47		
	2005	03		
START	2006	AF	**XRA A**	A is 00_{16} to display character in the address display.
	2007	47	**MOV B, A**	No dot to be displayed in the data display.

2008	21	**LXI H, 2050**	Starting address where the display is to be started.
2009	50		
200A	20		
200B	CD	**CALL OUTPUT**	Display subroutine to display characters in address display.
200C	D0		
200D	05		
200E	3E	**MVI A, 01**	A is 01_{16} to display character in the data display.
200F	01		
2010	06	**MVI B, 00**	No dot to be displayed in the data display.
2011	00		
2012	21	**LXI H, 2054**	HL points the address where the data to be displayed is lying.
2013	54		
2014	20		
2015	CD	**CALL OUTPUT**	Display last two characters of " CPU85" in data display.
2016	D0		
2017	05		
2018	11	**LXI D, 0000**	Display " CPU85" for about 0,5 seconds.
2019	00		
201A	00		
201B	CD	**CALL DELAY**	Delay subroutine.
201C	BC		
201D	03		
201E	CD	**CALL CLEAR**	Clear the display.
201F	47		
2020	03		
2021	11	**LXI D, 0000**	Clear display for about 0,5 seconds.
2022	00		
2023	00		
2024	CD	**CALL DELAY**	Delay subroutine.
2025	BC		
2026	03		
2027	C3	**JMP START**	Jump to label "START".
2028	06		
2029	20		

Data for display " CPU85"

2050	16		Space.
2051	0C		Letter C.

2052	12	Letter P.
2053	15	Letter U.
2054	08	Number 8.
2055	05	Number 5.

4. Contents of the report

1. The aim of the work.
2. Results of the registers examination and modification.
3. Results of the memory examination and modification.
4. Results of memory filling with a constant.
5. Results of relocation of data block.
6. Results of comparison of two memory blocks.
7. An examples of all executed commands of the system monitor.
8. Conclusions.

5. Test questions

1. What are the main characteristics of microprocessor training system M85-01?
2. Enumerate the possibilities of microprocessor training system M85-01.
3. Explain the purpose of control and indication elements of microprocessor training system M85-01.
4. How the microprocessor training system M85-01 memory is distributed?
5. Explain how to examine or modify the contents of register.
6. Explain how to examine or modify the contents of memory location.
7. Explain how to execute the program in single step mode.
8. Explain how to fill a particular memory area with a constant.
9. Explain how to move a block of data from one location to another location.

References

GRAŽULEVIČIUS, G. 2008. *Mikroprocesorinė technika*: mokomoji knyga. I dalis. Vilnius: Technika, 224 p. ISBN 978-9955-28-280-8.

ROUTT, W. A. 2007. *Microprocessor Architecture, Programming, and Systems Featuring the 8085*. USA, New York: Thomson Delmar Learning. 271 p. ISBN 1-4180-3241-7.