Laboratory work No 4

The principle of segment memory addressing in the x86 family microprocessors

1. The aim of the work

To acquaint with the assignment of segment registers, the calculation algorithm of physical memory addresses and the advantages of the segment memory addressing.

2. General knowledge

The principle of the segment memory addressing is applied in the 16-bit microprocessor Intel® 8086 and in later generation x86 family microprocessors. Since the processor Intel® 8086 uses the 16-bit address registers, the volume of directly addressable memory is equal to 2^{16} = 65536 (FFFFh) bytes or 64 KB. Such unit of directly addressable memory is called a segment. The physical address of the memory cell is formed of the segment address **(**multiple is always 10h) and inside the address segment. The memory cell inside the segment is indicated by the so-called execution (effective) address **EA** or by displacement **Disp**.

The microprocessor calculates the physical 20-bit address by adding the 16-bit segment address shifted by 4 bits to the left (or multiplied by 10h), indicated in one of the segment registers, to the 16-bit execution (effective) address or the displacement with respect to the start of this segment (Fig 1). By using the 20-bit address, the 1 MB memory (1 MB = 1024 KB = 1048576 B) can be addressed.

Segments are not strictly related to certain memory addresses and can partly or completely coincide. The random access memory unit of 16 bytes is called a paragraph. The segment start always coincides with the paragraph boundary (Fig 2).

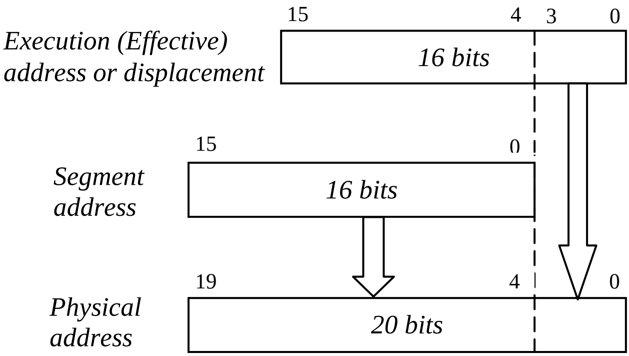

Fig 1. Formation of physical address

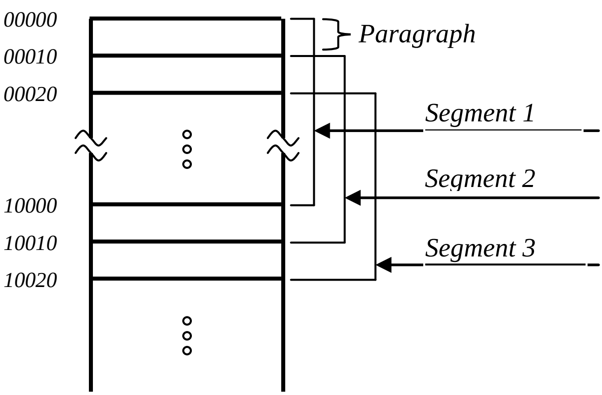

Fig 2. Principle of memory distribution into segments

The memory cell address in the assembler programs is written in the form of logical address:

SSSS:PPPP,

where SSSS is the 16-bit segment address (content of the segment register, e.g., **CS**); PPPP is the displacement (execution address) inside the segment or the cell address with respect to the segment start.

The same physical address of the memory cell can be indicated by logical addresses ambiguously. For example, the physical address 00025 can be written by such logical addresses: 0000:0025 or 0001:0015 or 0002:0005.

Advantages of segment addressing:

- the memory capacity can be up to 1 MB though instructions operate by 16-bit addresses;
- code, data and stack sections can be longer than 64 KB because several segments of the code, data and stack can be used for them;
- the use of certain memory regions is simpler;
- every time the program is started, and (or) its data can be placed in different memory regions without fear that addresses of conditional or unconditional transitions present in the program will confused (Fig 3).

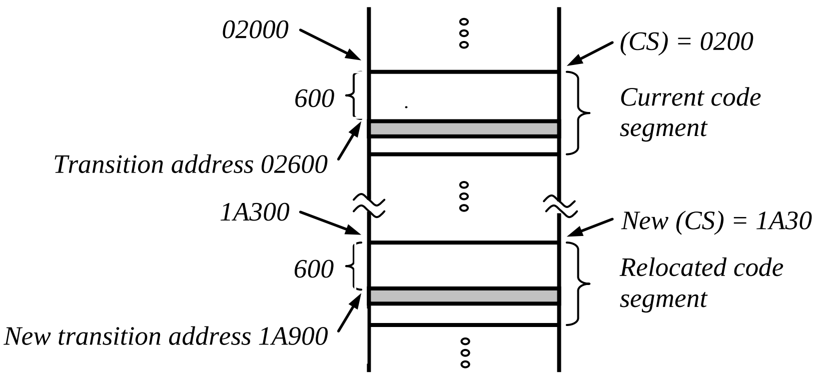

Fig 3. Relocation of the program into another segment

If the program is short (< 64 KB) and occupies only part of the segment, coinciding segments can be used (Fig 4).

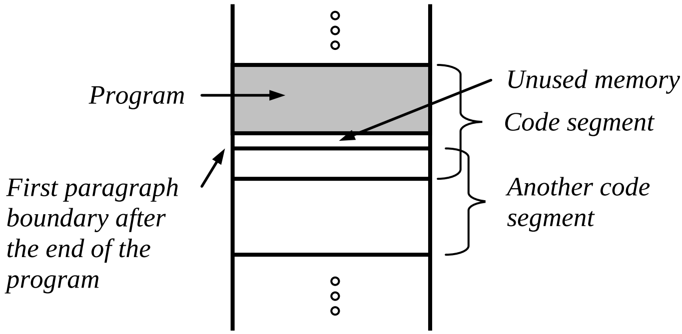

Fig 4. Coinciding segments

In the same way the coinciding and not coinciding segments of data and stack can be formed. The difference is that in this case the segment registers **DS**, **ES** or **SS** will be used to determine the physical address.

3. Task

1. Switch on the MPS and set to the display monitor mode. Determine the influence of pressing the return key „**RESET**“ has on the contents of the random-access memory cells and registers.

2. Press the return key „**RESET**“ and check the content of the segment register **CS**. If **CS** = 0000, enter the same freely chosen numbers at addresses 0000:1000, 0000:1001, 0000:1002 and 0000:1003. Then enter number 0001 to the register **CS** and again enter the same numbers at addresses 0001:1000, 0001:1001, 0001:1002 and 0001:1003. Press the key „**RESET**“ and using command “**D**” check the memory area from 0000:1000 to 0000:1014 address. Draw a conclusion.

3. Enter addresses 0000, 0001, 0002 and 0003 to segment registers **CS**, **DS**, **SS** and **ES**, respectively. Enter to these segments

the same freely chosen numbers for all segments at addresses 1000, 1001, 1002 and 1003. Press the key „**RESET**“ and using command “**D**” check the memory area from 0000:1000 to 0000:1034 address. Draw a conclusion.

Note: *Entering into different segments apply directives with logical addresses. For example, if data should be entered into the DS segment at the address 1000, the following directive is applied:*

E <0001:1000> <data> [.] ↵Enter

4. Press the key „**RESET**“. Enter the same freely chosen numbers to the memory cells at addresses 0000:1200, 0000:1201, 0000:1202 and 0000:1203. Then increase the contents of segment registers by unit and using command “**D**” check the memory area from 0001:1150 to 0001:1250 address. Draw a conclusion.

5. Press the key „**RESET**“, enter the unconditional transition address 1010 to the register **BX** and execute the program presented below:

Memory cell address	Program code		Comment
	Hexadecimal	Mnemonic	
0000:1000	40	**INC AX**	**AX** ← **AX**+1
0000:1001	FF	**JMP BX**	Unconditional transition to the address 1010 indicated in the register **BX**
0000:1002	E3		
...	...	...	...
0000:1010	40	**INC AX**	**AX** ← **AX**+1
0000:1011	CC	**INT 3**	Control transition to the monitor program (this instruction is always written at the end of the program)

Make sure that after execution of the program the content of the register **AX** increases by 2 units.

6. Using the display monitor command „**M**“ move the program of task 5 to another memory location, and to the previous program

location by the command „**F**“ write the constant (delete the program).

7. Execute the moved program. Pay attention to the result – the content of the register **AX**. Draw a conclusion.

8. Delete the used program. Increase the contents of segment registers by unit, beginning with the 0001:1000 address again enter the program of task 5 and execute it. Pay attention to the result. Draw a conclusion.

4. Contents of the report

1. The aim of the work.
2. Algorithm of calculating physical addresses.
3. Results of performing items 1 – 4 of the task.
4. The program text of item 5 of the program by mnemonic and hexadecimal codes.
5. Results and comments of items 5, 7 and 8 of the program execution.
6. Conclusions.

5. Test questions

1. Enumerate segment registers of the microprocessor Intel® 8086.
2. Concepts of the memory segment and the paragraph.
3. Give an example of coinciding segments.
4. Concept of the effective address.
5. Concept of displacement.
6. Concept of the physical address.
7. Concept of the logical address.
8. Algorithm of determining a physical address.

References

1. BERGER, A. S. *Hardware and Computer Organization*. USA, Burlington: Newnes; Book & DVD Edition. May 6, 2005. 512 p. ISBN 0750678860.

2. BREY, B. B. *The Intel Microprocessors 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro Processor, Pentium II, Pentium III, Pentium 4*. Architecture, Programming and Interfacing. USA, New Jersey: Pearson Prentice Hall; 7th Edition. March 23, 2006. 912 p. ISBN 0131974076.

3. TRIEBEL, W. A. *The 8088 and 8086 Microprocessors*: Programming, Interfacing, Software, Hardware and Applications. USA, New Jersey: Pearson Prentice Hall; 4th Edition. August 29, 2002. 1040 p. ISBN 0130930814.