Lecture 4

Program Execution

Text: Chapter 3

The Basics of Program Execution

Program execution is a two step process:

FETCH an instruction from memory (CS:IP)

EXECUTE the instruction

This cycle is repeated until the program terminates.

Suppose the following memory and register contents:

Addr	53040	53041	53042	53043	53044	53045	53046	53047	53048
	B8	23	01	8B	D8	03	06	56	02

CS	5300	DS	5834	IP	0040	AX	0000	BX	0000

1. FETCH:

The next instruction is at CS:IP

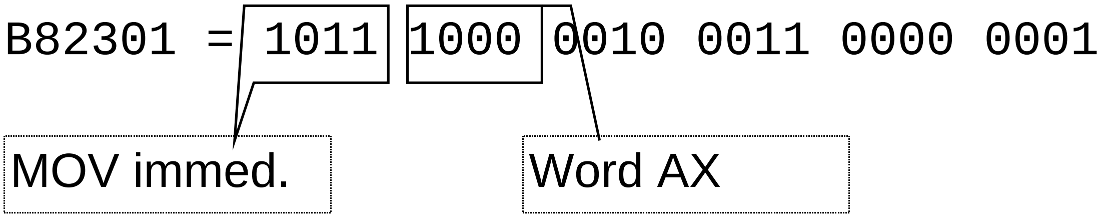
$$\begin{array}{r} \mathbf{5300}0 \\ +0040 \\ \hline 53040 \end{array}$$

The byte at 53040h is B8.

The Control Unit understands this as a MOV immediate operand instruction and the register is AX (a word). It therefore presumes the word to move is in the next two bytes (reversed). The entire instruction, therefore, is B82301.

Addr	53040	53041	53042	53043	53044	53045	53046	53047	53048
	B8	23	01	8B	D8	03	06	56	02

CS 5300 DS 5834 IP 0040 AX 0000 BX 0000



B82301 = 1011 1000 0010 0011 0000 0001

MOV immed.

Word AX

Now the CPU knows what the instruction is.
The second part of the FETCH cycle is to change the IP so that it points to the next instruction.

This instruction was at 0040h and was three bytes long.
The next instruction comes from 0043h (0040+3).

Addr	53040	53041	53042	53043	53044	53045	53046	53047	53048
	B8	23	01	8B	D8	03	06	56	02

CS 5300 DS 5834 IP **0043** AX 0000 BX 0000

Now the instruction can be executed, copying the immediate operand into the AX register (note that it is reversed).

Addr	53040	53041	53042	53043	53044	53045	53046	53047	53048
	B8	23	01	8B	D8	03	06	56	02

CS 5300 DS 5834 IP 0043 AX **0123** BX 0000

Now the next instruction can be fetched from CS:IP

CS 53000
IP +0043
53043 is the address of the instruction.

It contains the byte 8Bh, which is the code to MOV, and the next byte D8h specifies AX to BX.

This is, therefore, a two-byte instruction.

Addr	53040	53041	53042	53043	53044	53045	53046	53047	53048
	B8	23	01	8B	D8	03	06	56	02

CS	5300	DS	5834	IP	0043	AX	0123	BX	0000

IP increases by 2:

Addr	53040	53041	53042	53043	53044	53045	53046	53047	53048
	B8	23	01	8B	D8	03	06	56	02

CS	5300	DS	5834	IP	**0045**	AX	0123	BX	0000

And the instruction is executed:

Addr	53040	53041	53042	53043	53044	53045	53046	53047	53048
	B8	23	01	8B	D8	03	06	56	02

CS	5300	DS	5834	IP	0045	AX	0123	BX	**0123**

The next instruction is fetched from CS:IP

CS 53000

IP +0045
53045 which has the instruction 03065602 (4 byte)

IP increases by 4:

Addr	53040	53041	53042	53043	53044	53045	53046	53047	53048
	B8	23	01	8B	D8	03	06	56	02

CS	5300	DS	5834	IP	**0049**	AX	0123	BX	0123

The instruction 03565602 is broken down as

0306 add to what is in AX from memory word 0256

Memory word 0256 is fetched from the **DATA** Segment using DS:[0256] to determine its address:

DS 58340
+0256
58596 is the address of the word we want

DATA SEGMENT

Addr	58590	58591	58592	58593	58594	58595	58596	58597	58598
	01	00	02	00	03	00	04	00	05

CS	5300	DS	5834	IP	0049	AX	0123	BX	0123

The contents (0004h) is added to AX (0123h), and the sum (0127h) is placed in AX.

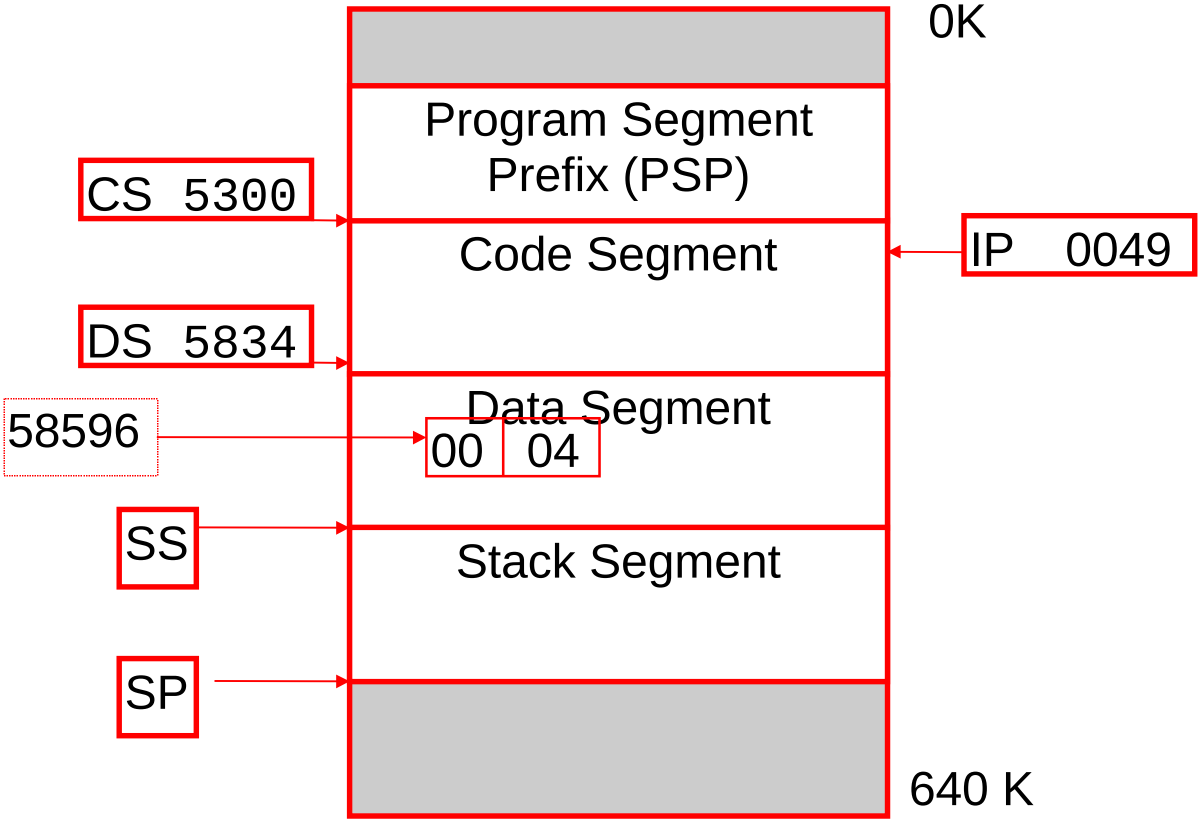
Addr	53040	53041	53042	53043	53044	53045	53046	53047	53048
	B8	23	01	8B	D8	03	06	56	02

CS	5300	DS	5834	IP	0049	AX	**0127**	BX	0123

In assembly language, these three instructions would be written:

Machine code   Assembly code

```
b82301                 mov  ax,0123h
8bd8                   mov bx,ax
03065602               add ax,[0256h]
```



So, after this instruction is executed, the contents of the CX register will be 0058h.

A Quick Look at Symbolic Instructions

MOV	AX,BX	Move contents of BX into AX
MOV	AX,TWO	Move contents of a memory location called TWO into AX
MOV	AX,50	Move the number 50 into AX
MOV	AX,[BX]	Move the number which is at the address in BX into AX