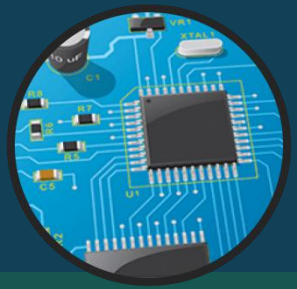
Addressing Mode & Machine Language



Topic 5

Fall 2019

Engr. Taufique-ur-Rehman <taufique.rehman@seecs.edu.pk>
Muhammad Imran Abeel <imran.abeel@seecs.edu.pk>

Machine language basics

A simple instruction set

R = register

@R = register given by a register

 # = Data

M(X) = memory location X

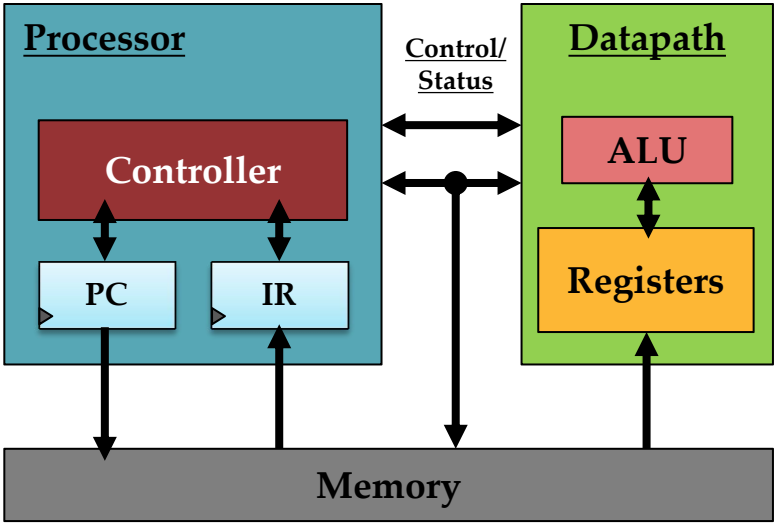
MOV Rn, A	Rn = M(A)
MOV A,Rn	M(A) = Rn
MOV @Rn, Rm	M(Rn) = Rm
MOV Rn, #y	Rn = y
ADD Rn, Rm	Rn = Rn +Rm
SUB Rn, Rm	Rn = Rn-Rm
JZ Rn,X	RC = PC+X

[2]

[ref: Mr. Jameel nawaz]

A simple instruction set

MOV Rn, A
MOV A,Rn
MOV @Rn, Rm
MOV Rn, #y
ADD Rn, Rm
SUB Rn, Rm
JZ Rn,X



Declare width of registers

PC, IR 16-bit

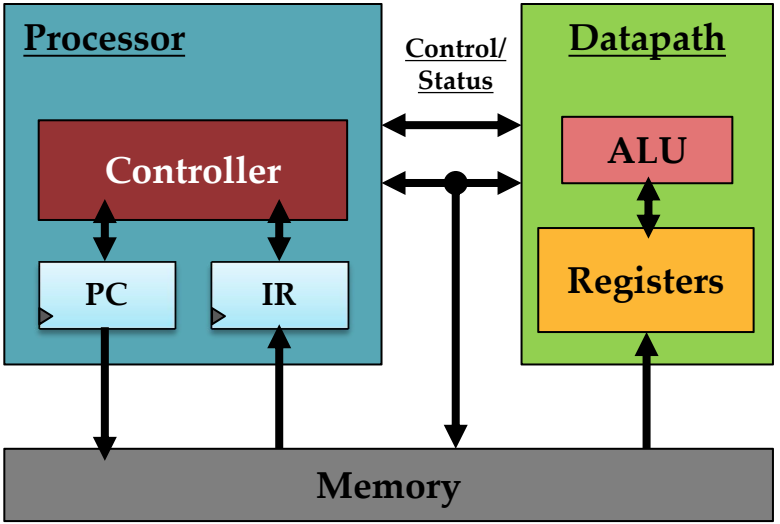
Memory 64K ~ 16 bit

Register file (16 registers) ~ 16 bit

[3]

A simple instruction set

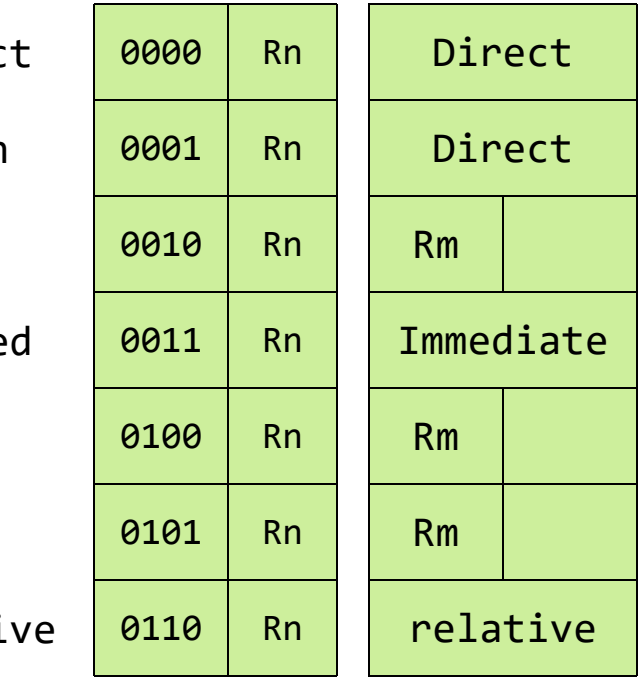
MOV Rn, A
MOV A,Rn
MOV @Rn, Rm
MOV Rn, #y
ADD Rn, Rm
SUB Rn, Rm
JZ Rn,X



IR contents

op	IR[15..12]	dir	IR[7..0]
Rn	IR[11..8]	imm	IR[7..0]
Rm	IR[7..4]	rel	IR[7..0]

A simple instruction set



Assembly language	First byte		second byte	operation
MOV Rn, direct	0000	Rn	Direct	Rn = M(direct)
MOV direct,Rn	0001	Rn	Direct	M(direct) = Rn
MOV @Rn, Rm	0010	Rn	Rm	M(Rn) = Rm
MOV Rn, #immed	0011	Rn	Immediate	Rn = immediate
ADD Rn, Rm	0100	Rn	Rm	Rn = Rn+Rm
SUB Rn, Rm	0101	Rn	Rm	Rn = Rn - Rm
JZ Rn,relative	0110	Rn	relative	PC = PC+relative

opcode operands

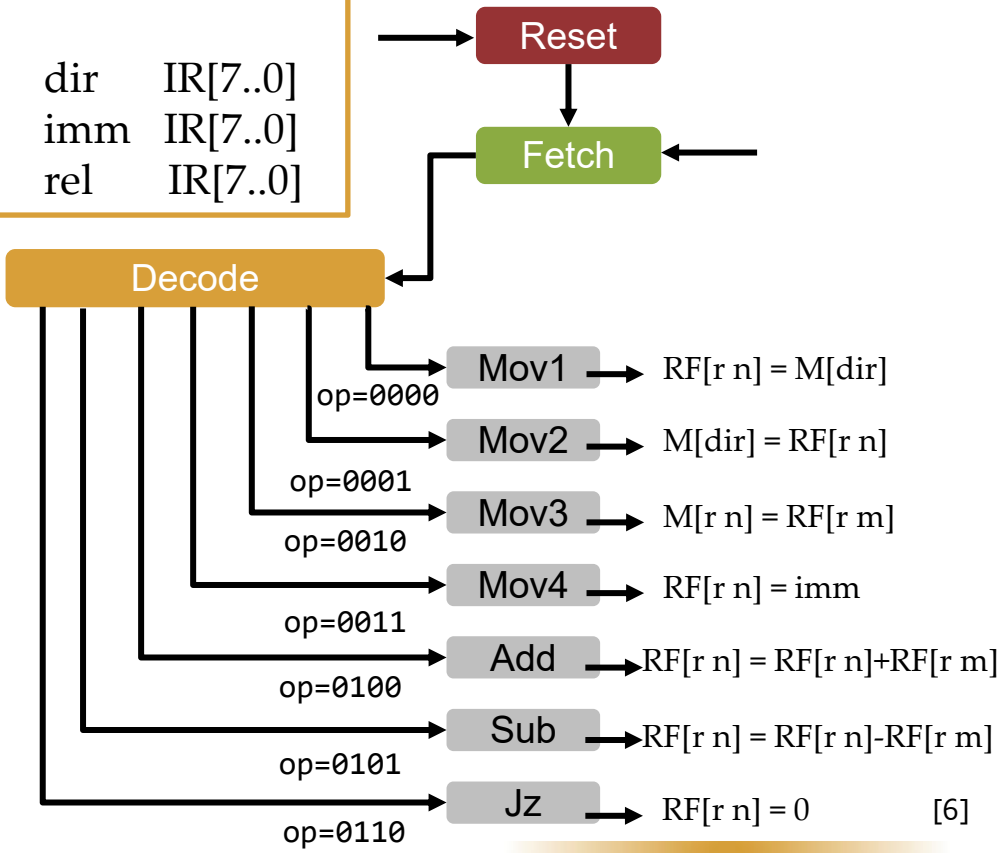
Machine instruction format

IR contents

op	IR[15..12]	dir	IR[7..0]
Rn	IR[11..8]	imm	IR[7..0]
Rm	IR[7..4]	rel	IR[7..0]

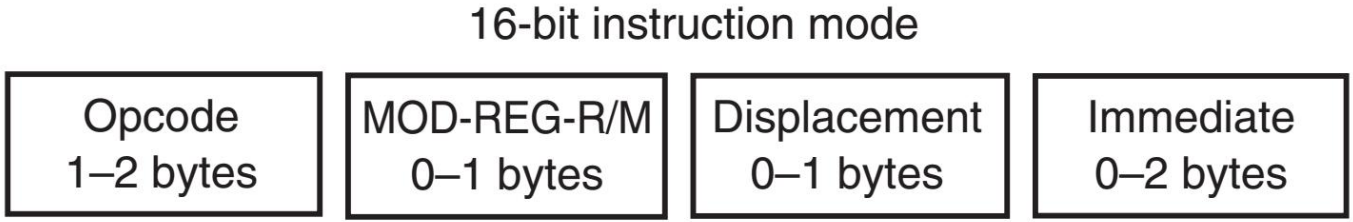
Instruction set

```
MOV Rn, direct
MOV direct,Rn
MOV @Rn, Rm
MOV Rn, #immed
ADD Rn, Rm
SUB Rn, Rm
JZ  Rn,relative
```



[6]

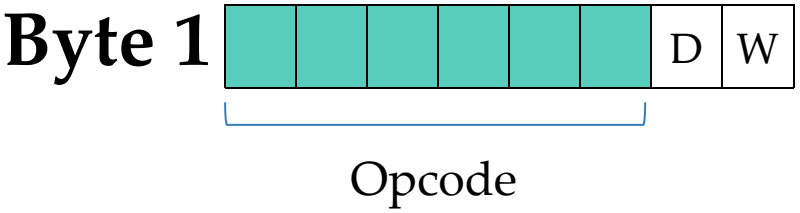
Machine instruction format



Machine language instructions for the 8086 :

1 to 13 bytes

Machine instruction format



Opcode: 6 bits, (Mov, Add, Sub)

D -> direction bit:

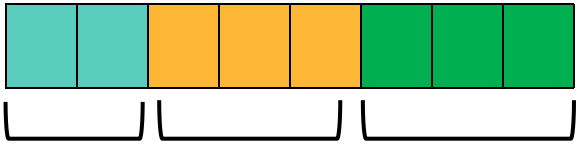
D = 0: REG to R/M and vice versa

W -> Word length:

W= 0: specify Byte (AL) & W=1 specify word or double word

Machine instruction format

Byte 2



MOD	Function
00	No displacement
01	8-bit sign-extended displacement
10	16-bit signed displacement
11	R/M is a register

TABLE 4–1 MOD field for the 16-bit instruction mode.

Machine instruction format

Byte 2

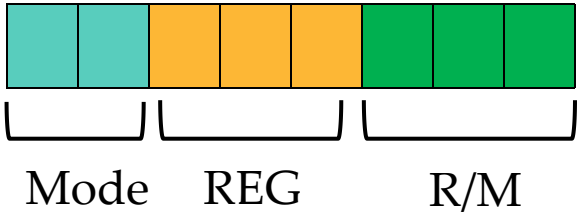
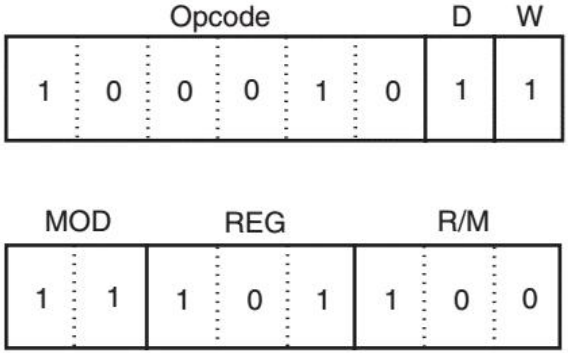


TABLE 4–3 REG and R/M (when) MOD = 11 assignments.

Code	W = 0 *(Byte)*	W = 1 *(Word)*	W = 1 *(Doubleword)*
000	AL	AX	EAX
001	CL	CX	ECX
010	DL	DX	EDX
011	BL	BX	EBX
100	AH	SP	ESP
101	CH	BP	EBP
110	DH	SI	ESI
111	BH	DI	EDI

Machine instruction format

Instruction: 8BECH



Machine instruction format

Instruction: 8BECH

100010 11 11 101 100

Opcode: MOV

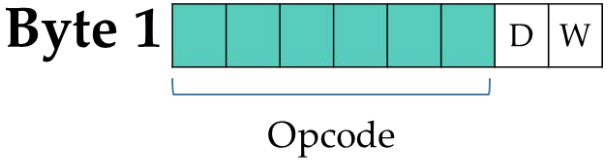
MOD	Function
00	No displacement
01	8-bit sign-extended displacement
10	16-bit signed displacement
11	R/M is a register

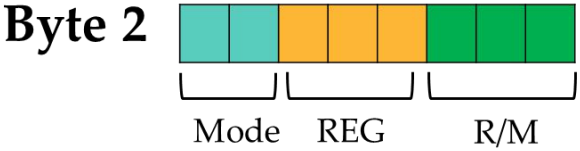
Code	W = 0 *(Byte)*	W = 1 *(Word)*	W = 1 *(Doubleword)*
000	AL	AX	EAX
001	CL	CX	ECX
010	DL	DX	EDX
011	BL	BX	EBX
100	AH	SP	ESP
101	CH	BP	EBP
110	DH	SI	ESI
111	BH	DI	EDI

This instruction means:
MOV BP,SP

[12]

Example





Instruction:

```
100010 10 00 010 101
```

This instruction means:
MOV DL,(DI)

MOD	*Function*
00	No displacement
01	8-bit sign-extended displacement
10	16-bit signed displacement
11	R/M is a register

R/M Code	*Addressing Mode*
000	DS:[BX+SI]
001	DS:[BX+DI]
010	SS:[BP+SI]
011	SS:[BP+DI]
100	DS:[SI]
101	DS:[DI]
110	SS:[BP]*
111	DS:[BX]

Code	W = 0 *(Byte)*	W = 1 *(Word)*	W = 1 *(Doubleword)*
000	AL	AX	EAX
001	CL	CX	ECX
010	DL	DX	EDX
011	BL	BX	EBX
100	AH	SP	ESP
101	CH	BP	EBP
110	DH	SI	ESI
111	BH	DI	EDI

Example

REG or R/M when MOD=11			R/M when MOD≠11			
REG R/M	W=0	W=1	R/M	MOD=00	MOD=01	MOD=10
000	AL	AX	000	BX+SI	BX+SI+D8	BX+SI+D16
001	CL	CX	001	BX+DI	BX+DI+D8	BX+DI+D16
010	DL	DX	010	BP+SI	BP+SI+D8	BP+SI+D16
011	BL	BX	011	BP+DI	BP+DI+D8	BP+DI+D16
100	AH	SP	100	SI	SI+D8	SI+D16
101	CH	BP	101	DI	DI+D8	DI+D16
110	DH	SI	110	*direct*	BP+D8	BP+D16
111	BH	DI	111	BX	BX+D8	BX+D16

MOV [1000h], DL, Opcode ??

REG or R/M when MOD=11			R/M when MOD≠11			
REG R/M	W=0	W=1	R/M	MOD=00	MOD=01	MOD=10
000	AL	AX	000	BX+SI	BX+SI+D8	BX+SI+D16
001	CL	CX	001	BX+DI	BX+DI+D8	BX+DI+D16
010	DL	DX	010	BP+SI	BP+SI+D8	BP+SI+D16
011	BL	BX	011	BP+DI	BP+DI+D8	BP+DI+D16
100	AH	SP	100	SI	SI+D8	SI+D16
101	CH	BP	101	DI	DI+D8	DI+D16
110	DH	SI	110	*direct*	BP+D8	BP+D16
111	BH	DI	111	BX	BX+D8	BX+D16

MOV [1000h], DL

REG or R/M when MOD=11			R/M when MOD≠11			
REG R/M	W=0	W=1	R/M	MOD=00	MOD=01	MOD=10
000	AL	AX	000	BX+SI	BX+SI+D8	BX+SI+D16
001	CL	CX	001	BX+DI	BX+DI+D8	BX+DI+D16
010	DL	DX	010	BP+SI	BP+SI+D8	BP+SI+D16
011	BL	BX	011	BP+DI	BP+DI+D8	BP+DI+D16
100	AH	SP	100	SI	SI+D8	SI+D16
101	CH	BP	101	DI	DI+D8	DI+D16
110	DH	SI	110	*direct*	BP+D8	BP+D16
111	BH	DI	111	BX	BX+D8	BX+D16

- Byte 1 100010 0 0
- Byte 2 00 010 110
- Byte 3 0000 0000
- Byte 4 0001 0000

Opcode example

- MOV [BX+1000h], 1234h opcode ???
  1100011w oo000mmm disp data

REG or R/M when MOD=11			R/M when MOD≠11			
REG R/M	W=0	W=1	R/M	MOD=00	MOD=01	MOD=10
000	AL	AX	000	BX+SI	BX+SI+D8	BX+SI+D16
001	CL	CX	001	BX+DI	BX+DI+D8	BX+DI+D16
010	DL	DX	010	BP+SI	BP+SI+D8	BP+SI+D16
011	BL	BX	011	BP+DI	BP+DI+D8	BP+DI+D16
100	AH	SP	100	SI	SI+D8	SI+D16
101	CH	BP	101	DI	DI+D8	DI+D16
110	DH	SI	110	*direct*	BP+D8	BP+D16
111	BH	DI	111	BX	BX+D8	BX+D16

Opcode example

- MOV [BX+1000h], 1234h opcode ???
  1100011w oo000mmm disp data

- Byte 1 1100011 1
- Byte 2 10 000 111
- Byte 3 0000 0000
- Byte 4 0001 0000
- Byte 5 0011 0100
- Byte 6 0001 0010

REG or R/M when MOD=11			R/M when MOD≠11			
REG R/M	W=0	W=1	R/M	MOD=00	MOD=01	MOD=10
000	AL	AX	000	BX+SI	BX+SI+D8	BX+SI+D16
001	CL	CX	001	BX+DI	BX+DI+D8	BX+DI+D16
010	DL	DX	010	BP+SI	BP+SI+D8	BP+SI+D16
011	BL	BX	011	BP+DI	BP+DI+D8	BP+DI+D16
100	AH	SP	100	SI	SI+D8	SI+D16
101	CH	BP	101	DI	DI+D8	DI+D16
110	DH	SI	110	*direct*	BP+D8	BP+D16
111	BH	DI	111	BX	BX+D8	BX+D16

Questions?

THANK YOU!

Addressing modes