Computer Structure and Language

The 8086/8088 Assembly Language

Hamid Sarbazi-Azad
Department of Computer Engineering
Sharif University of Technology (SUT)
Tehran, Iran

1

8086/88 has 7 types of instructions:

1. Data Transfer Instructions
2. Arithmetic Instructions
3. Bit Manipulation Instructions
4. String Instructions
5. Program Execution Transfer Instructions
6. Processor Control Instructions
7. Interrupt Instructions

2

The 8086/88's Data Transfer Instructions:

Move Instructions: mov

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Register/Memory to/from Register**

```
mov     reg1,reg2       ≡  reg1 ← (reg2);
mov     reg,memory      ≡  reg1 ← (EA);
mov     memory,reg      ≡  EA ← (reg);
```

Example 1:

```
mov     ax,si     ≡    ax ← (si);          word transfer
                  dw   Md  Reg  R/M
Machine code:   10001011  11  000  110   ≡   8BC6h
```

dw Md Reg R/M

3

Memory address calculation using Md and R/M fields

R/M	Memory mode			Register mode		Reg
	Md = 00	Md = 01	Md = 10	Md = 11		
	SR : EA	SR : EA	SR : EA	W = 0	W = 1	
000	DS : (BX)+(SI)	DS : (BX)+(SI)+d8	DS : (BX)+(SI)+d16	AL	AX	000
001	DS : (BX)+(DI)	DS : (BX)+(DI)+d8	DS : (BX)+(DI)+d16	CL	CX	001
010	SS : (BP)+(SI)	SS : (BP)+(SI)+d8	SS : (BP)+(SI)+d16	DL	DX	010
011	SS : (BP)+(DI)	SS : (BP)+(DI)+d8	SS : (BP)+(DI)+d16	BL	BX	011
100	DS : (SI)	DS : (SI)+d8	DS : (SI)+d16	AH	SP	100
101	DS : (DI)	DS : (DI)+d8	DS : (DI)+d16	CH	BP	101
110	DS : d16	SS : (BP)+d8	SS : (BP)+d16	DH	SI	110
111	DS : (BX)	DS : (BX)+d8	DS : (BX)+d16	BH	DI	111

SR: Segment Register **EA:** Effective Address

4

The 8086/88's Data Transfer Instructions:

Move Instructions: mov

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Register/Memory to/from Register**

```
mov   reg1,reg2     ≡  reg1 ← (reg2);
mov   reg,memory    ≡  reg1 ← (EA);
mov   memory,reg    ≡  EA ← (reg);
```

Example 1:

```
mov   ax,si     ≡   ax ← (si);          word transfer
```

```
                    dw   Md  Reg  R/M
Machine code:   10001011 11  000  110   ≡  8BC6h
```

Example 2:

```
mov   cl,ah     ≡   cl ← (ah);   byte transfer
```

```
                    dw   Md  Reg  R/M
Machine code:   10001000 11  100  001   ≡  88E1h
```

5

The 8086/88's Data Transfer Instructions:

Move Instructions: mov

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Register/Memory to/from Register (cont.)**

Example 3:

```
mov   cx,word ptr [si -5]   ≡  cx ← (M(si)-5);     word transfer
```

```
                    dw   Md  Reg  R/M   Disp. Low
Machine code:   10001011 01  001  100   11111011   ≡  8B4CFBh
```

Example 4:

```
mov   array[bx+2][di-6],dx   ≡  Marray+(bx)+(di)-4 ← (dx);
                                 @array = 260  of word type
```

Machine code:

```
                    dw   Md  Reg  R/M   Disp. Low   Disp. High
                10001001 10  010  001   00000000   00000001   ≡  89910001h
```

6

Memory address calculation using Md and R/M fields

R/M	Memory mode: Md = 00	Memory mode: Md = 01	Memory mode: Md = 10	Register mode: Md = 11, W = 0	Register mode: Md = 11, W = 1	Reg
	SR : EA	SR : EA	SR : EA	W = 0	W = 1	
000	DS : (BX)+(SI)	DS : (BX)+(SI)+d8	DS : (BX)+(SI)+d16	AL	AX	000
001	DS : (BX)+(DI)	DS : (BX)+(DI)+d8	DS : (BX)+(DI)+d16	CL	CX	001
010	SS : (BP)+(SI)	SS : (BP)+(SI)+d8	SS : (BP)+(SI)+d16	DL	DX	010
011	SS : (BP)+(DI)	SS : (BP)+(DI)+d8	SS : (BP)+(DI)+d16	BL	BX	011
100	DS : (SI)	DS : (SI)+d8	DS : (SI)+d16	AH	SP	100
101	DS : (DI)	DS : (DI)+d8	DS : (DI)+d16	CH	BP	101
110	DS : d16	SS : (BP)+d8	SS : (BP)+d16	DH	SI	110
111	DS : (BX)	DS : (BX)+d8	DS : (BX)+d16	BH	DI	111

SR: Segment Register **EA:** Effective Address

7

The 8086/88's Data Transfer Instructions:

Move Instructions: mov

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Immediate to Register/Memory**

1100011 w	Md	000	R/M	Disp. Low-byte	Disp. High-byte	Imm. Low-byte	Imm. High-byte
					for 16-bit displacement		for 16-bit (word) data

Example 1:

mov cx,256h ≡ cx ← 256h; i.e. (ch) = 2 (cl) = 56h

Machine code:

w	Md	R/M	Data Low	Data High	
11000111	11	000 001	01010110	00000010	≡ C7C15602h

Example 2:

mov byte ptr [bx+2][si],-2 ≡ $M_{(bx)+(si)+2}$ ← -2; byte transfer

Machine code:

w	Md	R/M	Disp. Low	Data Low	
11000110	01	000 000	00000010	11111110	≡ C64002FEh

8

The 8086/88's Data Transfer Instructions:

Move Instructions: mov

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Immediate to Register**

1011 w reg	Imm. Low-byte	Imm. High-byte
		If w=1

Example 1:

mov dx,-1 ≡ dx ← FFFFh; word transfer

Machine code:

	w reg	Data Low	Data High	
1011	1 010	11111111	11111111	≡ BAFFFFh

Example 2:

mov bh,100 ≡ bh ← 100; byte transfer

Machine code:

	w reg	Data Low	
1011	0 111	01100100	≡ B764h

9

The 8086/88's Data Transfer Instructions:

Move Instructions: mov

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Memory to Accumulator**

mov al,address ≡ al ← $(M_{address})_{byte}$; if w=0

mov ax,address ≡ ax ← $(M_{address})_{word}$; if w=1

1010000 w	Address Low	Address High

Example 1:

mov ax,array+10 ≡ ax ← $(M_{array+10})_{word}$; @array = 1000h

Machine code:

	w	Addr. Low	Addr. High	
10100001		00001010	00010000	≡ A10A10h

10

The 8086/88's Data Transfer Instructions:

Move Instructions: mov

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Accumulator to Memory**

mov address,al ≡ $M_{address}$ ← (al); if w=0
mov address,ax ≡ $M_{address}$ ← (ax); if w=1

1010001 w	Address Low	Address High

Example 1:
mov array-2,al ≡ $M_{array-2}$ ← (al); @array = 1000h

Machine code: 10100010 11111110 00001111 ≡ A2FE0Fh
(w | Addr. Low | Addr. High)

11

The 8086/88's Data Transfer Instructions:

Move Instructions: mov

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Register/Memory to Segment register**

10001110	Md	0SR	R/M	Disp. Low-byte	Disp. High-byte (for 16-bit displacement)

Segment Register	SR
ES	00
CS	01
SS	10
DS	11

Example 1:
mov ES,AX ≡ ES ← (AX);

Machine code: 10001110 11 000 000 ≡ 8EC0h
(Md | SR | R/M)

Example 2:
mov SS,array[bx] ≡ SS ← ($M_{array+(bx)}$); @array = 100

Machine code:

10001110 01 010 111 01100100 ≡ 8E5764h
(Md | SR | R/M | Disp. Low)

12

The 8086/88's Data Transfer Instructions:

Move Instructions: mov

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Segment Register to Register/Memory**

Segment Register	SR
ES	00
CS	01
SS	10
DS	11

Example 1:

mov AX,ES ≡ AX ← (ES);

Machine code: 10001100 (Md SR R/M: 11 000 000) ≡ 8CC0h

Example 2:

mov myarr+3[di],DS ≡ $M_{myarr+(di)+3}$ ← (DS); @myarr = 500h

Machine code:

	Md	SR	R/M	Disp. Low	Disp. High	
10001100	10	011	101	00000011	00000101	≡ 8C9D0305h

13

The 8086/88's Data Transfer Instructions:

Exchange Instructions: xchg

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Register/Memory with Register**

Example 1:

xchg si,bx ≡ (si) ↔ (bx);

Machine code: 10000111 (w Md Reg R/M: 11 110 011) ≡ 87F3h

Example 2:

xchg byte ptr cam+30[si],dl ≡ ($M_{cam+(si)+30}$) ↔ (dl);
@cam = 500h

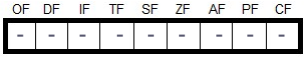
Machine code:

	w	Md	Reg	R/M	Disp. Low	Disp. High	
10000110	10	010	100	00011110	00000101	≡ 86941E05h

14

The 8086/88's Data Transfer Instructions:

Exchange Instructions: xchg



- **Register with Accumulator**

10010 Reg

Example:

xchg ax,bx ≡ (ax) ↔ (bx);

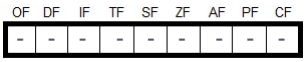
Machine code: 10010 011 (Reg) ≡ 93h

15

The 8086/88's Data Transfer Instructions:

Special Instructions:

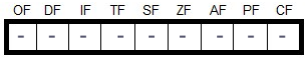
- **Translate Byte to AL: XLAT**



11010111

xlat ≡ al ← $(M_{(bx)+(al)})$;

- **Load Effective Address: LEA**



10001101 | Md | Reg | R/M | Disp. Low-byte | Disp. High-byte (for 16-bit displacement)

Example:

lea ax,[di][bx-1] ≡ ax ← (di)+(bx)-1;

Machine code:

	Md	Reg	R/M	Disp. Low	
10001101	01	000	001	11111111	≡ 8D41FFh

16

The 8086/88's Data Transfer Instructions:

Special Instructions:

- **Load Pointer to DS: LDS**

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

Example 1:

lds bx,p1[si] ≡ bx ← $(M_{p1+(si)})$; DS ← $(M_{p1+(si)+2})$;
@p1 = 1234h

Machine code:

	Md	Reg	R/M	Disp. Low	Disp. High	
11000101	10	011	100	00110100	00010010	≡ C59C3412h

17

The 8086/88's Data Transfer Instructions:

Special Instructions:

- **Load Pointer to ES: LES**

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

Example 1:

les bx,pip+10[di] ≡ bx ← $(M_{pip+(di)+10})$; ES ← $(M_{pip+(di)+2+10})$;
@pip = 1234h

Machine code:

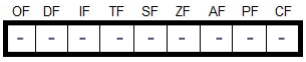
	Md	Reg	R/M	Disp. Low	Disp. High	
11000100	10	011	101	00111110	00010010	≡ C49D3E12h

18

The 8086/88's Data Transfer Instructions:

Special Instructions:

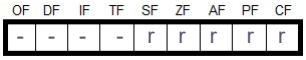
- **Load AH with Flags: LAHF**



10011111

lahf ≡ ah ← ($F_{7..0}$);

- **Store AH into Flags: SAHF**



r: restored from prev. saved value

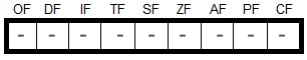
10011110

sahf ≡ $F_{7..0}$ ← (ah);

19

The 8086/88's Data Transfer Instructions:

Push Instructions: PUSH



- **Push Register/Memory into Stack**

Example 1:

push dx ≡ sp ← (sp)-2; $M_{(sp)}$ ← (dx);

Machine code: 11111111 (Md 11, 110, R/M 010) ≡ FFF2h

Example 2:

push array+2 ≡ sp ← (sp)-2; $M_{(sp)}$ ← ($M_{array+2}$);

@array = 11h of word type

Machine code:

	Md		R/M	Disp. Low	Disp. High	
11111111	00	110	110	00010011	00000000	≡ FF361300h

20

The 8086/88's Data Transfer Instructions:

Push Instructions: PUSH (cont.)

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Push Register into Stack** 01010 Reg

Example: push dx ≡ sp ← (sp)-2; $M_{(sp)}$ ← (dx);

Machine code: 01010 010 (Reg) ≡ 52h

Segment Register	SR
ES	00
CS	01
SS	10
DS	11

- **Push Segment Register into Stack** 000 SR 110

Example: push DS ≡ sp ← (sp)-2; $M_{(sp)}$ ← (DS);

Machine code: 000 11 110 (SR) ≡ 1Eh

- **Push Flags into Stack: PUSHF** 10011100

pushf ≡ sp ← (sp)-2; $M_{(sp)}$ ← ($F_{15..0}$);

21

The 8086/88's Data Transfer Instructions:

Pop Instructions: POP

OF	DF	IF	TF	SF	ZF	AF	PF	CF
-	-	-	-	-	-	-	-	-

- **Pop Register/Memory from Stack**

Example 1:

pop dx ≡ dx ← ($M_{(sp)}$); sp ← (sp)+2;

Machine code: 10001111 11 000 010 (Md, R/M) ≡ 8FC2h

Example 2:

pop arc-2 ≡ M_{arc-2} ← ($M_{(sp)}$); sp ← (sp)+2;

@arc = 100h of word type

Machine code:

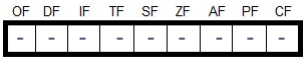
10001111 00 000 110 11111110 00000000 (Md, R/M, Disp. Low, Disp. High) ≡ 8F06FE00h

22

The 8086/88's Data Transfer Instructions:

Pop Instructions: POP (cont.)

- **Pop Register from Stack** 01011 Reg



Example: pop dx ≡ dx ← $(M_{(sp)})$; sp ← (sp)+2;

Machine code: 01011 010 (Reg) ≡ 5Ah

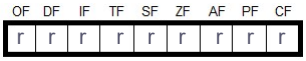
Segment Register	SR
ES	00
CS	01
SS	10
DS	11

- **Pop Segment Register from Stack** 000 SR 111

Example: pop DS ≡ DS ← $(M_{(sp)})$; sp ← (sp)+2;

Machine code: 000 11 111 (SR) ≡ 1Fh

- **Pop Flags from Stack: POPF** 10011101



popf ≡ $F_{15..0}$ ← $(M_{(sp)})$; sp ← (sp)+2;

23

End of Slides

24