

# Computer Structure and Language

## The 8086/8088 Assembly Language

Hamid Sarbazi-Azad

Department of Computer Engineering  
Sharif University of Technology (SUT)  
Tehran, Iran



### 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

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

3

The 8086/88's Logic Instructions:

And Instructions: AND

OF	DF	IF	TF	SF	ZF	AF	PF	CF
0	-	-	-	X	X	U	X	0

Register/Memory with Register to Either

Possible combinations:

and

reg1,reg2

≡

reg1 ← (reg1)^(reg2); update Flags;

and

reg,memory

≡

reg ← (reg)^(EA); update Flags;

and

memory,reg

≡

EA ← (EA)^(reg); update Flags;

001000 d w

Md

Reg

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example:

and

cx,word ptr [si -5]

≡

cx ← (cx)^(M<sub>(si-5)</sub>); update Flags;

Machine code:

dw

Md

Reg

R/M

Disp. Low

00100011 01 001 100 11111011

≡

234CFBh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

4

The 8086/88's Logic Instructions:

And Instructions: AND

OF	DF	IF	TF	SF	ZF	AF	PF	CF
0	-	-	-	X	X	U	X	0

Immediate to Register/Memory

1000000 w

Md

100

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Imm. Low-byte

Imm. High-byte

If w=1

Example 1:

and

bx,25h

≡

bx ← (bx)^(0025h); update Flags;

Machine code:

w

Md

R/M

Data Low

Data High

10000001 11 100 011 00100101 00000000

≡

81E32500h

Example 2:

and

word ptr [bx+20][di],1234h

≡

M<sub>(bx)+(di)+20</sub> ← (M<sub>(bx)+(di)+20</sub>)^(1234h); update Flags;

Machine code:

w

Md

R/M

Disp. Low

Data Low

Data High

10000001 01 100 001 00010100 00110100 00010010

≡

8161143412h

2

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

5

The 8086/88's Logic Instructions:

And Instructions: AND

OF	DF	IF	TF	SF	ZF	AF	PF	CF
0	-	-	-	X	X	U	X	0

• Immediate to Accumulator

and al,data8 ≡ al ← (al)∧data8; update Flags; if w=0

and ax,data16 ≡ ax ← (ax)∧data16; update Flags; if w=1

0010010 w

Data Low-byte

Data High-byte

If w=1

Example 1:

and ax,100 ≡ ax ← (ax)∧100; update Flags;

Machine code: 00100101 01100100 00000000 ≡ 256400h

Example 2:

and al,-1 ≡ al ← (al)∧FFh; update Flags;

Machine code: 00100100 11111111 ≡ 24FFh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

6

The 8086/88's Logic Instructions:

Or Instructions: OR

OF	DF	IF	TF	SF	ZF	AF	PF	CF
0	-	-	-	X	X	U	X	0

• Register/Memory with Register to Either

Possible combinations:

or reg1,reg2 ≡ reg1 ← (reg1)∨(reg2); update Flags;

or reg,memory ≡ reg ← (reg)∨(EA); update Flags;

or memory,reg ≡ EA ← (EA)∨(reg); update Flags;

000010 d w

Md

Reg

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example:

or dx,word ptr [bx + 5] ≡ dx ← (dx)∨(M<sub>(bx)+5</sub>); update Flags;

Machine code: 00001011 01 010 111 00000101 ≡ 0B5705h

3

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

7

The 8086/88's Logic Instructions:

Or Instructions: OR

OF	DF	IF	TF	SF	ZF	AF	PF	CF
0	-	-	-	X	X	U	X	0

• Immediate to Register/Memory

1000000 w	Md	001	R/M	Disp. Low-byte	Disp. High-byte	Imm. Low-byte	Imm. High-byte
				for 16-bit displacement		If w=1	

Example 1:

or        dl,2         $\equiv$      $dl \leftarrow (dl) \vee 02h$ ; update Flags;

Machine code:

	w	Md	R/M	Data Low	
10000000	11	001	010	00000010	$\equiv$ 80CA02h

Example 2:

or    word ptr [bx+1][si+1],8000h     $\equiv$      $M_{(bx)+(si)+2} \leftarrow (M_{(bx)+(si)+2}) \vee 8000h$ ;  
update Flags;

Machine code:

	w	Md	R/M	Disp. Low	Data Low	Data High	
10000001	01	001	000	00000010	00000000	10000000	$\equiv$ 8148020080h

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

8

The 8086/88's Logic Instructions:

Or Instructions: OR

OF	DF	IF	TF	SF	ZF	AF	PF	CF
0	-	-	-	X	X	U	X	0

• Immediate to Accumulator

0000110 w	Data Low-byte	Data High-byte
	If w=1	

Example 1:

or        ax,1000h         $\equiv$      $ax \leftarrow (ax) \vee 1000$ ; update Flags;

Machine code:

	w	Data Low	Data High	
00001101	00000000	00010000	$\equiv$ 0D0010h	

Example 2:

or        al,-128         $\equiv$      $al \leftarrow (al) \vee 80h$ ; update Flags;

Machine code:

	w	Data Low	
00001100	10000000	$\equiv$ 0C80h	

4

(c) Hamid Sarbazi-AzadComputer Structure & Language, Lecture#4: Bit manipulation instructions9

The 8086/88's Logic Instructions:

Exclusive-Or Instructions: XOR

OFDFIFTFSSFZFAFPFCF

0- - - X X U X 0

• Register/Memory with Register to Either

Possible combinations:

xorreg1,reg2≡reg1←(reg1)⊕(reg2); update Flags;

xorreg,memory≡reg←(reg)⊕(EA); update Flags;

xormemory,reg≡EA←(EA)⊕(reg); update Flags;

001100dwMdRegR/MDisp. Low-byteDisp. High-byte

for 16-bit displacement

Example:

xorbp,word ptr [bx]≡bp←(bp)⊕(M<sub>(bx)</sub>); update Flags;

Machine code:

dwMdRegR/M

00100111

≡332Fh

(c) Hamid Sarbazi-AzadComputer Structure & Language, Lecture#4: Bit manipulation instructions10

The 8086/88's Logic Instructions:

Exclusive-Or Instructions: XOR

OFDFIFTFSSFZFAFPFCF

0- - - X X U X 0

• Immediate to Register/Memory

1000000wMd110R/MDisp. Low-byteDisp. High-byteImm. Low-byteImm. High-byte

for 16-bit displacementIf w=1

Example 1:

xorcl,60h≡cl←(cl)⊕60h; update Flags;

Machine code:

wMdR/MData Low

100000001111000101100000

≡80F160h

Example 2:

xorword ptr [bx+32],4000h≡M<sub>(bx)+32</sub>←(M<sub>(bx)+32</sub>)⊕4000h; update Flags;

Machine code:

wMdR/MDisp. LowData LowData High

1000000101110111001000000000000001000000

≡8177200040h

5

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

11

The 8086/88's Logic Instructions:

Exclusive-Or Instructions: XOR

OF

DF

IF

TF

SF

ZF

AF

PF

CF

0

-

-

-

X

X

U

X

0

• Immediate to Accumulator

xor

al,data8

≡

al ← (al)⊕ data8; update Flags;

if w=0

xor

ax,data16

≡

ax ← (ax)⊕ data16; update Flags;

if w=1

0011010 w

Data Low-byte

Data High-byte

If w=1

Example 1:

xor

ax,7432h

≡

ax ← (ax)⊕ 7432; update Flags;

Machine code:

00110101

w

Data Low

00110010

Data High

01110100

≡

353274h

Example 2:

xor

al,127

≡

al ← (al)⊕ 7Fh; update Flags;

Machine code:

00110100

w

Data Low

01111111

≡

347Fh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

12

The 8086/88's Logic Instructions:

Test Instructions: TEST

OF

DF

IF

TF

SF

ZF

AF

PF

CF

0

-

-

-

X

X

U

X

0

• Test Register/Memory and Register

Possible combinations:

test reg1,reg2

≡

Realize (reg1) ∧(reg2) and update Flags;

test memory,reg

≡

Realize (EA) ∧(reg) and update Flags;

test reg,memory

≡

Realize (Reg) ∧(EA) and update Flags;

1000010 w

Md

Reg

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example:

test cx,word ptr [si -5]

≡

Realize (cx) ∧(M<sub>(si)-5</sub>) and update Flags;

Machine code:

10000101

w

Md

01

Reg

001

R/M

100

Disp. Low

11111011

≡

854CFBh

6

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

13

The 8086/88's Logic Instructions:

Test Instructions: TEST

OF

DF

IF

TF

SF

ZF

AF

PF

CF

0

-

-

-

X

X

U

X

0

• Test Immediate and Register/Memory

1111011 w

Md

000

R/M

Disp. Low-byte

Disp. High-byte

Imm. Low-byte

Imm. High-byte

for 16-bit displacement

If w=1

Example 1:

test cx,256h ≡ Realize (cx)∧ 256h and update Flags;

Machine code:

11110111

w

Md

11

000

R/M

001

Data Low

01010110

Data High

00000010

≡ F7C15602h

Example 2:

test word ptr [bx+21][si],1004h ≡ Realize (M<sub>(bx)+(si)+21</sub>)∧ 1004h  
and update Flags;

Machine code:

11110111

w

Md

01

000

R/M

000

Disp. Low

00010101

Data Low

00000100

Data High

00010000

≡ F740150410h

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

14

The 8086/88's Logic Instructions:

Test Instructions: TEST

OF

DF

IF

TF

SF

ZF

AF

PF

CF

0

-

-

-

X

X

U

X

0

• Test Immediate and Accumulator

1010100 w

Data Low-byte

Data High-byte

If w=1

Example 1:

test ax,-1 ≡ Realize (ax)∧ FFFFh and update Flags;

Machine code:

10101001

w

Data Low

11111111

Data High

11111111

≡ A9FFFFh

Example 2:

test al,10101010B ≡ Realize (al)∧ AAh and update Flags;

Machine code:

10101000

w

Data Low

10101010

≡ A8AAh

7

w	Md	R/M	Disp. Low	Disp. High	
11110110	10 010	100	00011110	00000011	≡ F6941E03h

11010010<sup>vw</sup> 01<sup>Md</sup> 100<sup>R/M</sup> 100<sup>Disp. Low</sup> 00000010 ≡ D26402h



(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

17

The 8086/88's Logic Instructions:

Shift Instructions:

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

X

X

U

X

X

• Shift Logical Right: SHR

110100

v w

Md

101

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

0 → 

Operand

 → 

CF

shr operand,1 ≡ operand ← (operand)>>1; update Flags; if v=0

shr operand,cl ≡ operand ← (operand)>>(cl); update Flags; if v=1

Example 1:

shr bl,1 ≡ bl ← (bl)>>1; update Flags;

Machine code: 

11010000

vw

Md

11

R/M

011

 ≡ D0EBh

Example 2:

shr word ptr [bx],cl ≡ M<sub>(bx)</sub> ← (M<sub>(bx)</sub>)>>(cl); update Flags;

Machine code:

11010011

vw

Md

00

R/M

111

 ≡ D32Fh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

18

The 8086/88's Logic Instructions:

Shift Instructions:

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

X

X

U

X

X

• Shift Arithmetic Right: SAR

110100

v w

Md

111

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Operand

 → 

CF

sar operand,1 ≡ operand ← (operand)>>1; update Flags; if v=0

sar operand,cl ≡ operand ← (operand)>>(cl); update Flags; if v=1

Example 1:

sar cx,1 ≡ cx ← (cx)>>1; update Flags;

Machine code: 

11010001

vw

Md

11

R/M

001

 ≡ D1F9h

Example 2:

sar byte ptr [si+20],cl ≡ M<sub>(si)+20</sub> ← (M<sub>(si)+20</sub>)>>(cl); update Flags;

Machine code:

11010010

vw

Md

01

R/M

100

Disp. Low

00010100

 ≡ D27C14h

(c) Hamid Sarbazi-AzadComputer Structure & Language, Lecture#4: Bit manipulation instructions19

The 8086/88's Logic Instructions:

Shift Instructions:

OFDFIFTFSFZFAPFCF

X-- -- -- -- --X

• Rotate Left: ROL

110100vwMd000R/MDisp. Low-byteDisp. High-byte

for 16-bit displacement

CF

Operand

rol operand,1 ≡ rotate operand left one bit; update Flags; if v=0

rol operand,cl ≡ rotate operand left (cl) bits; update Flags; if v=1

Example 1:

rol bx,1 ≡ rotate (bx) left one bit; update Flags;

Machine code: 1101000111000011 ≡ D1C3h

Example 2:

rol byte ptr [di-1],cl ≡ rotate (M<sub>(di)-1</sub>) left (cl) bits; update Flags;

Machine code:

110100100100010111111111 ≡ D245FFh

(c) Hamid Sarbazi-AzadComputer Structure & Language, Lecture#4: Bit manipulation instructions20

The 8086/88's Logic Instructions:

Shift Instructions:

OFDFIFTFSFZFAPFCF

X-- -- -- -- --X

• Rotate Right: ROR

110100vwMd001R/MDisp. Low-byteDisp. High-byte

for 16-bit displacement

Operand

CF

ror operand,1 ≡ rotate operand right one bit; update Flags; if v=0

ror operand,cl ≡ rotate operand right (cl) bits; update Flags; if v=1

Example 1:

ror dl,1 ≡ rotate (dl) right one bit; update Flags;

Machine code: 1101000011001010 ≡ D0CAh

Example 2:

ror byte ptr [bx+21],cl ≡ rotate (M<sub>(bx)+21</sub>) right (cl) bits; update Flags;

Machine code:

11010010010011100010101 ≡ D24F15h

10

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

21

The 8086/88's Logic Instructions:

Shift Instructions:

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

-

-

-

-

X

• Rotate Through Carry Left: RCL

110100

v w

Md

010

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

CF

Operand

rcl operand,1 ≡ rotate CF:operand left one bit; update Flags; if v=0

rcl operand,cl ≡ rotate CF:operand left (cl) bits; update Flags; if v=1

Example 1:

rcl si,1 ≡ rotate CF:(si) left one bit; update Flags;

Machine code: 

11010001

vw

Md

R/M

11 010 110

 ≡ D1D6h

Example 2:

rcl byte ptr [si+2],cl ≡ rotate CF:(M<sub>(si)+2</sub>) left (cl) bits; update Flags;

Machine code: 

11010010

vw

Md

R/M

Disp. Low

01 010 100 00000010

 ≡ D25402h

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

22

The 8086/88's Logic Instructions:

Shift Instructions:

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

-

-

-

-

X

• Rotate Through Carry Right: RCR

110100

v w

Md

011

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Operand

CF

rcr operand,1 ≡ rotate CF:operand right one bit; update Flags; if v=0

rcr operand,cl ≡ rotate CF:operand right (cl) bits; update Flags; if v=1

Example 1:

rcr ax,1 ≡ rotate CF:(ax) right one bit; update Flags;

Machine code: 

11010001

vw

Md

R/M

11 011 000

 ≡ D1D8h

Example 2:

rcr byte ptr [di],cl ≡ rotate CF:(M<sub>(di)</sub>) right (cl) bits; update Flags;

Machine code: 

11010010

vw

Md

R/M

00 011 101

 == D21Dh

11

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#4: Bit manipulation instructions

23

End of Slides