

# Computer Structure and Language

## The 8086/8088 Assembly Language

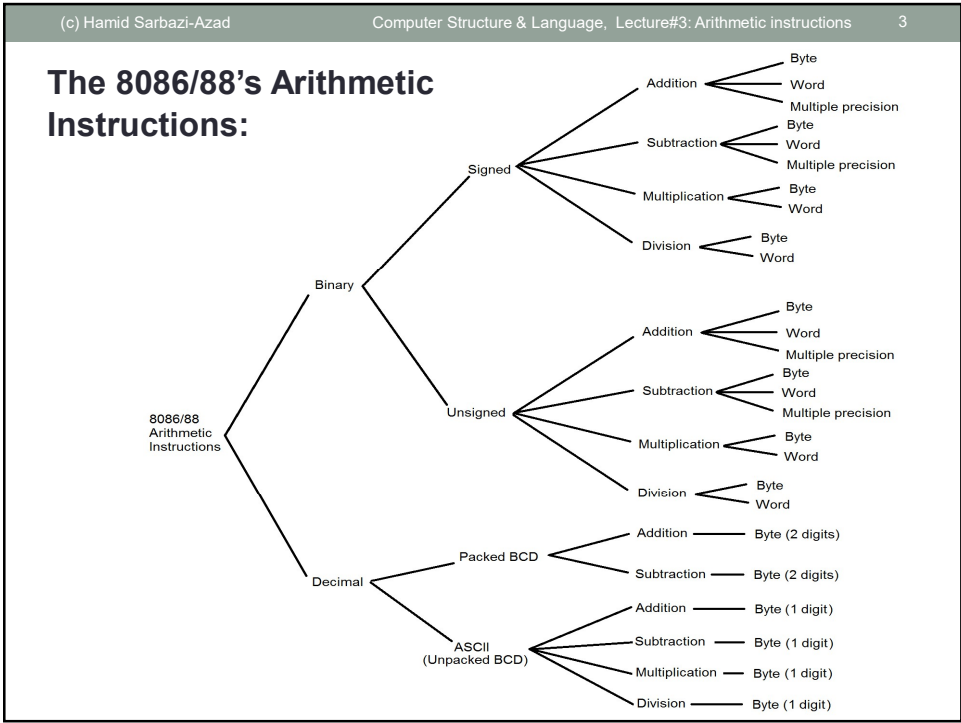
Hamid Sarbazi-Azad

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-AzadComputer Structure & Language, Lecture#3: Arithmetic instructions4

### The 8086/88's Arithmetic Instructions:

#### Add Instructions: ADD

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

x: set or cleared according to the result

- Add Register/Memory with Register to Either**

Possible combinations:

add reg1,reg2

$\equiv$

$\text{reg1} \leftarrow (\text{reg1}) + (\text{reg2});$

update Flags;

add reg,memory

$\equiv$

$\text{reg} \leftarrow (\text{reg}) + (\text{EA});$

update Flags;

add memory,reg

$\equiv$

$\text{EA} \leftarrow (\text{EA}) + (\text{reg});$

update Flags;

000000 d w

Md

Reg

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

**Example:**

add cx,word ptr [si -5]

$\equiv$

$\text{cx} \leftarrow (\text{cx}) + (\text{M}_{(\text{si}-5)});$

update Flags;

Machine code:

dw

00000011

Md

01

Reg

001

R/M

100

Disp. Low

11111011

$\equiv$

034CFBh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

5

The 8086/88's Arithmetic Instructions:

Add Instructions: ADD

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

Add Immediate to Register/Memory

100000	s	w	Md	000	R/M	Disp. Low-byte	Disp. High-byte	Imm. Low-byte	Imm. High-byte
						for 16-bit displacement		If sw=01	

Example 1:

add      cx,256h       $\equiv$     $cx \leftarrow (cx)+256h$ ; update Flags;

Machine code:

	sw	Md	R/M	Data Low	Data High	
100000	01	11	000	001	01010110	00000010

 $\equiv$  81C15602h

Example 2:

add      word ptr [bx+2][si],-2       $\equiv$     $M_{(bx)+(si)+2} \leftarrow (M_{(bx)+(si)+2})-2$ ;  
update Flags;

Machine code:

	sw	Md	R/M	Disp. Low	Data Low	
100000	11	01	000	000	00000010	11111110

 $\equiv$  834002FEh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

6

The 8086/88's Arithmetic Instructions:

Add Instructions: ADD

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

Add Immediate to Accumulator

0000010	w	Data Low-byte	Data High-byte
		If w=1	

Example 1:

add      ax,10       $\equiv$     $ax \leftarrow (ax)+10$ ; update Flags;

Machine code:

	w	Data Low	Data High	
00000101	00001010	00000000		050A00h

Example 2:

add      al,-1       $\equiv$     $al \leftarrow (al)-1$ ; update Flags;

Machine code:

	w	Data Low		
00000100	11111111			04FFh

3

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

7

The 8086/88's Arithmetic Instructions:

Add with Carry Instructions: ADC

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

Add with Carry Register/Memory with Register to Either

Possible combinations:

adc reg1,reg2

$\equiv \text{reg1} \leftarrow (\text{reg1})+(\text{reg2})+\text{CF}; \text{ update Flags};$

adc reg,memory

$\equiv \text{reg} \leftarrow (\text{reg})+(\text{EA})+\text{CF}; \text{ update Flags};$

adc memory,reg

$\equiv \text{EA} \leftarrow (\text{EA})+(\text{reg})+\text{CF}; \text{ update Flags};$

000100 d w

Md

Reg

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example:

adc cx,word ptr [si -11h]

$\equiv \text{cx} \leftarrow (\text{cx})+(\text{M}_{(\text{si})-17})+\text{CF}; \text{ update Flags};$

Machine code:

00010011

dw

01

Md

001

Reg

100

R/M

11101111

Disp. Low

$\equiv 134\text{CEh}$

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

8

The 8086/88's Arithmetic Instructions:

Add with Carry Instructions: ADC

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

Add with Carry Immediate to Register/Memory

100000 s w

Md

010

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Imm. Low-byte

Imm. High-byte

If sw=01

Example 1:

adc bx,1234h

$\equiv \text{bx} \leftarrow (\text{bx})+1234\text{h}+\text{CF}; \text{ update Flags};$

Machine code:

100000

sw

01

Md

11

010

R/M

011

00110100

Data Low

00010010

Data High

$\equiv 81\text{D33412h}$

Example 2:

adc word ptr [bx+3][si -1],-5

$\equiv \text{M}_{(\text{bx})+(\text{si})+2} \leftarrow (\text{M}_{(\text{bx})+(\text{si})+2})-5+\text{CF};$

update Flags;

Machine code:

10000011

sw

01

Md

010

R/M

000

00000010

Disp. Low

11111011

Data Low

$\equiv 835002\text{FBh}$

4

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

9

The 8086/88's Arithmetic Instructions:

Add with Carry Instructions: ADC

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

X

X

X

X

X

• Add with Carry Immediate to Accumulator

adc al,data8 ≡ al ← (al)+data8+CF; update Flags; if w=0

adc ax,data16 ≡ ax ← (ax)+data16+CF; update Flags; if w=1

0001010 w

Data Low-byte

Data High-byte

If w=1

Example 1: adc ax,0001010b ≡ ax ← (ax)+10+CF; updt e Flags;

Machine code: 00010101 <sup>w</sup>00001010 <sup>Data Low</sup>00000000 <sup>Data High</sup> == 150A00h

Example 2: adc al,127 ≡ al ← (al)+127+CF; update Flags;

Machine code: 00010100 <sup>w</sup>01111111 <sup>Data Low</sup> == 147Fh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

10

The 8086/88's Arithmetic Instructions:

Subtract Instructions: SUB

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

X

X

X

X

X

• Subtract Register/Memory and Register to Either

Possible combinations:

sub reg1,reg2 ≡ reg1 ← (reg1)-(reg2); update Flags;

sub reg,memory ≡ reg ← (reg)-(EA); update Flags;

sub memory,reg ≡ EA ← (EA)-(reg); update Flags;

001010 d w

Md

Reg

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example:

sub cx,word ptr [si -128] ≡ cx ← (cx) - (M<sub>(si)-128</sub>); update Flags;

Machine code: 00101011 <sup>dw</sup>01 <sup>Md</sup>001 <sup>Reg</sup>100 <sup>R/M</sup>10000000 <sup>Disp. Low</sup> == 2B4C80h

5

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

11

The 8086/88's Arithmetic Instructions:

Subtract Instructions: SUB

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

X

X

X

X

X

• Subtract Immediate from Register/Memory

100000

s

w

Md

101

R/M

Disp. Low-byte

Disp. High-byte

Imm. Low-byte

Imm. High-byte

for 16-bit displacement

If sw=01

Example 1:

subcx,1024h≡cx←(cx)-1024h; update Flags;

Machine code:

100000

sw

01

Md

11

R/M

101

001

Data Low

00100100

Data High

00010000

≡

81E92410h

Example 2:

subword ptr [bx+7][si],-7≡M<sub>(bx)+(si)+7</sub>←(M<sub>(bx)+(si)+7</sub>)+7; update Flags;

Machine code:

100000

sw

11

Md

01

R/M

101

000

Disp. Low

00000111

Data Low

11111001

≡

836807F9h

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

12

The 8086/88's Arithmetic Instructions:

Subtract Instructions: SUB

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

X

X

X

X

X

• Subtract Immediate from Accumulator

0010110

w

Data Low-byte

Data High-byte

If w=1

Example 1:

subax,-3≡ax←(ax)+3; update Flags;

Machine code:

0010110

w

101

Data Low

11111101

Data High

11111111

==

2DFDFFh

Example 2:

subal,10≡al←(al)-10; update Flags;

Machine code:

0010110

w

100

Data Low

00001010

≡

2C0Ah

6

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

13

The 8086/88's Arithmetic Instructions:

Subtract with Borrow Instructions: SBB

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

Subtract with Borrow Register/Memory and Register to Either

Possible combinations:

sbb reg1,reg2

≡

reg1 ← (reg1)-(reg2)-CF; update Flags;

sbb reg,memory

≡

reg ← (reg)-(EA)-CF; update Flags;

sbb memory,reg

≡

EA ← (EA)-(reg)-CF; update Flags;

000110 d w

Md

Reg

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example:

sbb di,word ptr [bp -5]

≡

di ← (di)-(M<sub>(bp)-5</sub>)-CF; update Flags;

Machine code:

dw

Md

Reg

R/M

Disp. Low

00011011 01 111 110 11111011

≡

1B7EFBh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

14

The 8086/88's Arithmetic Instructions:

Subtract with Borrow Instructions: SBB

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

Subtract with Borrow Immediate from Register/Memory

100000 s w

Md

011

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Imm. Low-byte

Imm. High-byte

If sw=01

Example 1:

sbb bx,1

≡

bx ← (bx) -1 -CF; update Flags;

Machine code:

sw

Md

R/M

Data Low

100000 11 11 011 011 00000001

≡

83DB01h

Example 2:

sbb word ptr [bp+2][si],22

≡

M<sub>(bp)+(si)+2</sub> ← (M<sub>(bp)+(si)+2</sub>)-22-CF; update Flags;

Machine code:

sw

Md

R/M

Disp. Low

Data Low

10000011 01 011 010 00000010 00010110

≡

835A0216h

7

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

15

The 8086/88's Arithmetic Instructions:

Subtract with Borrow Instructions: SBB

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

- Subtract with Borrow Immediate from Accumulator

sbb al,data8

$\equiv$

$al \leftarrow (al) - data8 - CF$ ; update Flags; if w=0

sbb ax,data16

$\equiv$

$ax \leftarrow (ax) - data16 - CF$ ; update Flags; if w=1

0001110 w

Data Low-byte

Data High-byte

If w=1

Example 1:

sbb ax,1020h

$\equiv$

$ax \leftarrow (ax) - 1020h - CF$ ; update Flags;

Machine code:

00011101

<sup>w</sup>00100000

<sup>Data Low</sup><sup>Data High</sup>00010000

$\equiv$

1D2010h

Example 2:

sbb al,33

$\equiv$

$al \leftarrow (al) - 33 - CF$ ; update Flags;

Machine code:

00011100

<sup>w</sup><sup>Data Low</sup>00100001

$\equiv$

1C21h

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

16

The 8086/88's Arithmetic Instructions:

Compare Instructions: CMP

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

- Compare Register/Memory and Register

Possible combinations:

cmp reg1,reg2

$\equiv$

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

cmp reg,memory

$\equiv$

Realize (reg)-(EA) and update Flags;

cmp memory,reg

$\equiv$

Realize (EA)-(reg) and update Flags;

001110 d w

Md

Reg

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example:

cmp dx,word ptr [si]

$\equiv$

Realize (dx)-(M<sub>(si)</sub>) and update Flags;

Machine code:

00111011

<sup>dw</sup>00

<sup>Md</sup>010

<sup>Reg</sup>100

$\equiv$

3B14h

8



(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

17

The 8086/88's Arithmetic Instructions:

Compare Instructions: **CMP**

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

• Compare Immediate with Register/Memory

100000	s	w	Md	111	R/M	Disp. Low-byte	Disp. High-byte	Imm. Low-byte	Imm. High-byte
						for 16-bit displacement		If sw=01	

Example 1:

cmp si,2 ≡ Realize (si)-2 and update Flags;

Machine code:

	sw	Md	R/M	Data Low	
100000	11	11	111	110	00000010

≡ 83FE02h

Example 2:

cmp byte ptr [bx+20h][si],-2 ≡ Realize (M<sub>(bx)+(si)+32</sub>)+2 and update Flags;

Machine code:

	sw	Md	R/M	Disp. Low	Data Low	
10000000	01	111	000	00100000	11111110	

≡ 807820FEh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

18

The 8086/88's Arithmetic Instructions:

Compare Instructions: **CMP**

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

• Compare Immediate with Accumulator

cmp al,data8 ≡ Realize (al)-data8 and update Flags; if w=0  
cmp ax,data16 ≡ Realize (ax)-data16 and update Flags; if w=1

0011110	w	Data Low-byte	Data High-byte
		If w=1	

Example 1:

cmp ax,31 ≡ Realize (ax)-31 and update Flags;

Machine code:

	w	Data Low	Data High	
00111101	00011111	00000000		

≡ 3D1F00h

Example 2:

cmp al,19 ≡ Realize (al)-19 and update Flags;

Machine code:

	w	Data Low	
00111100	00010011		

≡ 3C13h

9

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

19

The 8086/88's Arithmetic Instructions:

Increment Instructions: INC

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

X

X

X

X

-

• Increment Register/Memory

1111111 w

Md

000

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example 1:

inc      bx       $\equiv$      $bx \leftarrow (bx)+1$ ; update Flags;

Machine code:    1111111<sup>w</sup> 11 000<sup>Md</sup> 011<sup>R/M</sup>       $\equiv$     FFC3h

Example 2:

inc      byte ptr array+22[si]       $\equiv$      $M_{array+(si)+22} \leftarrow (M_{array+(si)+22})+1$ ;  
update Flags;  
@array = 500h

Machine code:

11111110<sup>w</sup> 10<sup>Md</sup> 000<sup>Reg</sup> 100<sup>R/M</sup> 00010110<sup>Disp. Low</sup> 00000101<sup>Disp. High</sup>       $\equiv$     FE841605h

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

20

The 8086/88's Arithmetic Instructions:

Increment Instructions: INC

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

X

X

X

X

-

• Increment Register

01000 Reg

Example:

inc      bx       $\equiv$      $bx \leftarrow (bx)+1$ ; update Flags;

Machine code:    01000<sup>Reg</sup> 011       $\equiv$     43h

10

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

21

The 8086/88's Arithmetic Instructions:

Decrement Instructions: DEC

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

• Decrement Register/Memory

1111111 w

Md

001

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example 1:

dec si  $\equiv$  si  $\leftarrow$  (si)-1; update Flags;

Machine code: 

11111111<sup>w</sup> 11001<sup>Md</sup> 110<sup>R/M</sup>

 $\equiv$  FFCEh

Example 2:

dec byte ptr array[bx][si]  $\equiv$  M<sub>array+(bx)+(si)</sub>  $\leftarrow$  (M<sub>array+(bx)+(si)</sub>)-1; update Flags;  
@array = 500h

Machine code:

11111110<sup>w</sup> 10<sup>Md</sup> 001<sup>Reg</sup> 000<sup>R/M</sup> 00000000<sup>Disp. Low</sup> 00000101<sup>Disp. High</sup>

 $\equiv$  FE880005h

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

22

The 8086/88's Arithmetic Instructions:

Decrement Instructions: DEC

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

• Decrement Register

01001 Reg

Example:

dec si  $\equiv$  si  $\leftarrow$  (si)-1; update Flags;

Machine code: 

01001<sup>Reg</sup> 110

 $\equiv$  4Eh

11

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

23

The 8086/88's Arithmetic Instructions:

Change Sign: NEG

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

X

X

X

X

X

• Negate (2's complement) Register/Memory

1111011

w

Md

011

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

Example 1:

neg    bx    ≡    bx ← -(bx);    update Flags;

Machine code:    

1111011

<sup>w</sup>

<sup>Md</sup>11

<sup>R/M</sup>011

    ≡    F7DBh

Example 2:

neg    byte ptr array[si]    ≡    M<sub>array+(si)</sub> ← -(M<sub>array+(si)</sub>); update Flags;

@array = 500h

Machine code:

11110110

<sup>w</sup>

<sup>Md</sup>10

<sup>Reg</sup>011

<sup>R/M</sup>100

<sup>Disp. Low</sup>00000000

<sup>Disp. High</sup>00000101

    ≡    F69C0005h

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

24

The 8086/88's Arithmetic Instructions:

Multiplication Instructions:

OF

DF

IF

TF

SF

ZF

AF

PF

CF

X

-

-

-

U

U

U

U

X

u: undefined

• Multiply Unsigned: MUL

1111011

w

Md

100

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

mul operand<sub>byte</sub>    ≡    ax ← (al)\*(operand<sub>byte</sub>); update Flags;    if w=0

mul operand<sub>word</sub>    ≡    dx:ax ← (ax)\*(operand<sub>word</sub>); update Flags;    if w=1

Example 1:

mul    cl    ≡    ax ← (al)\*(cl);    update Flags;

Machine code:    

11110110

<sup>w</sup>

<sup>Md</sup>11

<sup>R/M</sup>100

<sup>R/M</sup>001

    ≡    F6E1h

Example 2:

mul    word ptr [si]    ≡    dx:ax ← (ax)\*(M<sub>(si)</sub>); update Flags;

Machine code:    

11110111

<sup>w</sup>

<sup>Md</sup>00

<sup>R/M</sup>100

<sup>R/M</sup>100

    ≡    F724h

12

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

25

The 8086/88's Arithmetic Instructions:

Multiplication Instructions:

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

Integer Multiply (Signed): IMUL

1111011 w

Md

101 R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

imul operand<sub>byte</sub>

$\equiv ax \leftarrow (al) * (operand_{byte});$

update Flags;

if w=0

imul operand<sub>word</sub>

$\equiv dx:ax \leftarrow (ax) * (operand_{word});$

update Flags;

if w=1

Example 1:

imul dl

$\equiv ax \leftarrow (al) * (dl);$

update Flags;

Machine code:

11110110

<sup>w</sup>

<sup>Md</sup>

11

<sup>R/M</sup>

010

$\equiv$

F6EAh

Example 2:

imul word ptr [si][bx]

$\equiv dx:ax \leftarrow (ax) * (M_{(si)+(bx)});$

update Flags;

Machine code:

11110111

<sup>w</sup>

<sup>Md</sup>

00

<sup>R/M</sup>

101 000

$\equiv$

F728h

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

26

The 8086/88's Arithmetic Instructions:

Division Instructions:

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

Divide Unsigned: DIV

1111011 w

Md

110 R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

div operand<sub>byte</sub>

$\equiv al \leftarrow (ax) / (operand_{byte});$

ah  $\leftarrow$  Remainder;

update F;

if w=0

div operand<sub>word</sub>

$\equiv ax \leftarrow (dx:ax) / (operand_{word});$

dx  $\leftarrow$  Remainder;

update F;

if w=1

Example 1:

div bh

$\equiv al \leftarrow (ax) / (bh);$

ah  $\leftarrow$  Remainder;

update Flags;

Machine code:

11110110

<sup>w</sup>

<sup>Md</sup>

11

<sup>R/M</sup>

110 111

$\equiv$

F6F7h

Example 2:

div word ptr [bx]

$\equiv ax \leftarrow (dx:ax) / (M_{(bx)});$

dx  $\leftarrow$  Remainder;

update Flags;

Machine code:

11110111

<sup>w</sup>

<sup>Md</sup>

00

<sup>R/M</sup>

110 111

$\equiv$

F737h

13

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

27

The 8086/88's Arithmetic Instructions:

Division Instructions:

OFDFIFTFSFZFAPFCF

U

-

-

-

U

U

U

U

U

Integer Divide (Signed): IDIV

1111011w

Md

111

R/M

Disp. Low-byte

Disp. High-byte

for 16-bit displacement

idiv operand<sub>byte</sub> ≡ al←(ax)/(operand<sub>byte</sub>); ah←Remainder; update F; if w=0

idiv operand<sub>word</sub> ≡ ax←(dx:ax)/(operand<sub>word</sub>); dx←Remainder; update F;

if w=1

Example 1:

idiv dh ≡ al ←(ax)/(dh); ah ←Remainder; update Flags;

Machine code: 1111011011111110110≡F6FEh

Example 2:

idiv word ptr [bp][di] ≡ ax ←(dx:ax)/(M<sub>(bp)+(di)</sub>); dx ←Remainder;

update Flags;

Machine code: 111101100111011011≡F73Bh

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

28

The 8086/88's Arithmetic Instructions:

Division Related Instructions:

Covert Byte to Word: CBW

OFDFIFTFSFZFAPFCF

-

-

-

-

-

-

-

-

-

10011000

ax ← (al);

Covert Word to Double-Word: CWD

OFDFIFTFSFZFAPFCF

-

-

-

-

-

-

-

-

-

10011001

dx:ax ← (ax);

14

(c) Hamid Sarbazi-Azad

Computer Structure & Language, Lecture#3: Arithmetic instructions

29

End of Slides