

Computer Structure and Language

The 8086/8088 Assembly Language

Hamid Sarbazi-Azad

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



1

(c) Hamid Sarbazi-Azad

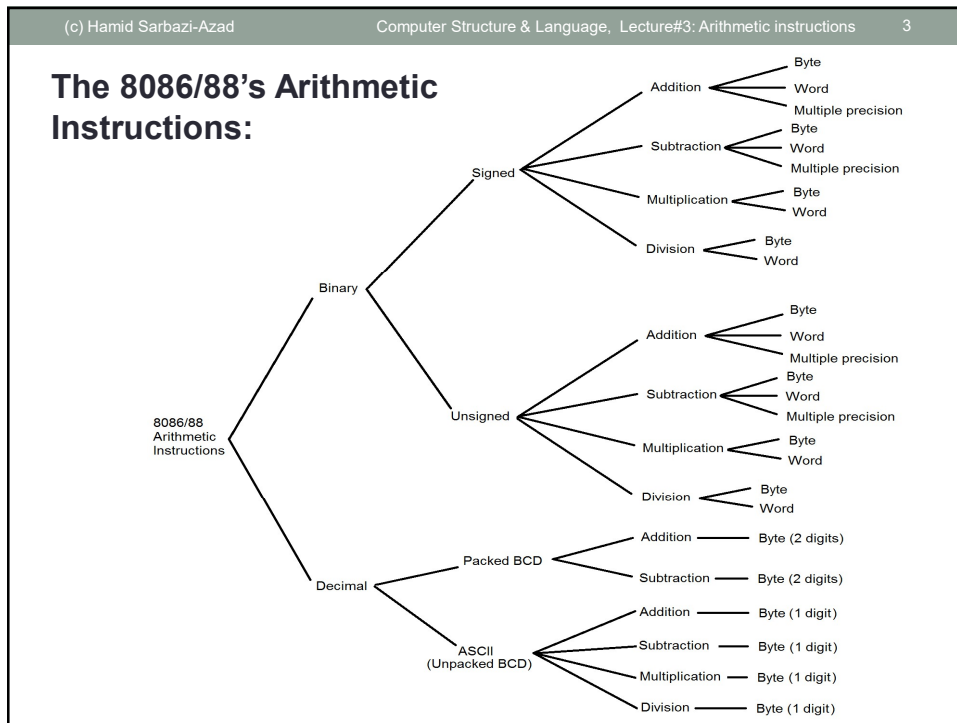
Computer Structure & Language, Lecture#3: Arithmetic instructions

2

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



3

(c) Hamid Sarbazi-Azad Computer Structure & Language, Lecture#3: Arithmetic instructions 4

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  ≡ reg1 ← (reg1)+(reg2); update Flags;
add reg,memory ≡ reg ← (reg)+(EA); update Flags;
add memory,reg ≡ EA ← (EA)+(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]  ≡ cx ← (cx)+(M(si-5)); update Flags;
  
```

Machine code: dw Md Reg R/M Disp. Low ≡ 034CFBh
00000011 01 001 100 11111011

4

(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    ≡  cx ← (cx)+256h; update Flags;
```

Machine code:

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

≡ 81C15602h

Example 2:

```
add    word ptr [bx+2][si],-2  ≡  M(bx)+(si)+2 ← (M(bx)+(si)+2)-2;
                                     update Flags;
```

Machine code:

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

≡ 834002FEh

5

(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    ≡  ax ← (ax)+10; update Flags;
```

Machine code:

0000010	w	Data Low	Data High
101	00001010	00000000	

≡ 050A00h

Example 2:

```
add    al,-1    ≡  al ← (al)-1; update Flags;
```

Machine code:

0000010	w	Data Low
100	11111111	

≡ 04FFh

6

(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    ≡ reg1 ← (reg1)+(reg2)+CF; update Flags;
adc reg,memory   ≡ reg ← (reg)+(EA)+CF; update Flags;
adc memory,reg   ≡ EA ← (EA)+(reg)+CF; 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]  ≡ cx ← (cx)+(M(si)-17)+CF; update Flags;
  
```

Machine code: dw Md Reg R/M Disp. Low ≡ 134CEFh
00010011 01 001 100 11101111

7

(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 Imm. Low-byte Imm. High-byte
for 16-bit displacement If sw=01

Example 1:

```

adc bx,1234h    ≡ bx ← (bx)+1234h+CF; update Flags;
  
```

Machine code:

```

100000 sw Md R/M Data Low Data High    ≡ 81D33412h
01 11 010 011 00110100 00010010
  
```

Example 2:

```

adc word ptr [bx+3][si -1],-5  ≡ M(bx)+(si)+2 ← (M(bx)+(si)+2)-5+CF;
                                update Flags;
  
```

Machine code:

```

10000011 sw Md R/M Disp. Low Data Low    ≡ 835002FBh
01 010 000 00000010 11111011
  
```

8

(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` ≡ $ax \leftarrow (ax) + 10 + CF$; updt e Flags;
Machine code: 00010101 ^w 00001010 ^{Data Low} 00000000 ^{Data High} == 150A00h

Example 2: `adc al,127` ≡ $al \leftarrow (al) + 127 + CF$; update Flags;
Machine code: 00010100 ^w 01111111 ^{Data Low} == 147Fh

9

(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]` ≡ $cx \leftarrow (cx) - (M_{(si)-128})$; update Flags;
Machine code: 00101011 ^{dw} 01 ^{Md} 001 ^{Reg} 100 ^{R/M} 10000000 ^{Disp. Low} == 2B4C80h

10

(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:

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

Machine code:

100000	sw	Md	R/M	Data Low	Data High
01	11	101	001	00100100	00010000

≡ 81E92410h

Example 2:

```
sub    word ptr [bx+7][si],-7    ≡  M(bx)+(si)+7 ← (M(bx)+(si)+7)+7;
                                     update Flags;
```

Machine code:

100000	sw	Md	R/M	Disp. Low	Data Low
11	01	101	000	00000111	11111001

≡ 836807F9h

11

(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:

```
sub    ax,-3    ≡  ax ← (ax)+3; update Flags;
```

Machine code:

0010110	w	Data Low	Data High
101	11111101	11111111	11111111

== 2DFDFFh

Example 2:

```
sub    al,10    ≡  al ← (al)-10; update Flags;
```

Machine code:

0010110	w	Data Low
100	00001010	

≡ 2C0Ah

12

(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(bp)-5)-CF; update Flags;
  
```

Machine code: dw Md Reg R/M Disp. Low ≡ 1B7EFBh
00011011 01 111 110 11111011

13

(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 Imm. Low-byte Imm. High-byte
for 16-bit displacement If sw=01

Example 1:

```

sbb    bx,1    ≡  bx ← (bx) -1 -CF; update Flags;
  
```

Machine code:

```

100000 11 11 011 011 00000001    ≡ 83DB01h
  
```

Example 2:

```

sbb    word ptr [bp+2][si],22    ≡  M(bp)+(si)+2 ← (M(bp)+(si)+2)-22-CF;
                                   update Flags;
  
```

Machine code:

```

10000011 01 011 010 00000010 00010110    ≡ 835A0216h
  
```

14

(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    ≡  al ← (al)-data8-CF; update Flags;   if w=0
sbb    ax,data16   ≡  ax ← (ax)-data16-CF; update Flags; if w=1
```

0001110 w Data Low-byte Data High-byte
If w=1

Example 1: sbb ax,1020h ≡ ax ← (ax)-1020h-CF; update Flags;

Machine code: 00011101 00100000 00010000 ≡ 1D2010h

Example 2: sbb al,33 ≡ al ← (al)-33-CF; update Flags;

Machine code: 00011100 00100001 ≡ 1C21h

15

(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   ≡  Realize (reg1)-(reg2) and update Flags;
cmp    reg,memory  ≡  Realize (reg)-(EA) and update Flags;
cmp    memory,reg  ≡  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]  ≡  Realize (dx)-(M(si)) and update Flags;
```

Machine code: 00111011 00 010 100 ≡ 3B14h

16

(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 \equiv Realize (si)-2 and update Flags;

Machine code:

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

\equiv 83FE02h

Example 2:

cmp byte ptr [bx+20h][si],-2 \equiv Realize $(M_{(bx)+(si)+32})+2$ and update Flags;

Machine code:

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

\equiv 807820FEh

17

(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**

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

Example 1:

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

Machine code:

w	Data Low	Data High
00111101	00011111	00000000

\equiv 3D1F00h

Example 2:

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

Machine code:

w	Data Low
00111100	00010011

\equiv 3C13h

18

(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: 11111111 11 000 011 \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 10 000 100 00010110 00000101 \equiv FE841605h

19

(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 011 \equiv 43h

20

(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 11 001 110 \equiv FFCEh

Example 2:

dec byte ptr array[bx][si] \equiv M_{array+(bx)+(si)} \leftarrow (M_{array+(bx)+(si)})-1;
update Flags;
@array = 500h

Machine code:

11111110 10 001 000 00000000 00000101 \equiv FE880005h

21

(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 110 \equiv 4Eh

22

(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 \equiv bx \leftarrow -(bx); update Flags;

Machine code: 11110111 11 011 011 \equiv F7DBh

Example 2:

neg byte ptr array[si] \equiv M_{array+(si)} \leftarrow -(M_{array+(si)}); update Flags;
@array = 500h

Machine code:

11110110 10 011 100 00000000 00000101 \equiv F69C0005h

23

(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_{byte} \equiv ax \leftarrow (al)*(operand_{byte}); update Flags; if w=0
mul operand_{word} \equiv dx:ax \leftarrow (ax)*(operand_{word}); update Flags; if w=1

Example 1:

mul cl \equiv ax \leftarrow (al)*(cl); update Flags;

Machine code: 11110110 11 100 001 \equiv F6E1h

Example 2:

mul word ptr [si] \equiv dx:ax \leftarrow (ax)*(M_(si)); update Flags;

Machine code: 11110111 00 100 100 \equiv F724h

24

(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**

for 16-bit displacement

1111011	w	Md	101	R/M	Disp. Low-byte	Disp. High-byte

$\text{imul operand}_{\text{byte}} \equiv ax \leftarrow (ax) * (\text{operand}_{\text{byte}}); \text{ update Flags; if } w=0$
 $\text{imul operand}_{\text{word}} \equiv dx:ax \leftarrow (ax) * (\text{operand}_{\text{word}}); \text{ update Flags; if } w=1$

Example 1: $\text{imul dl} \equiv ax \leftarrow (ax) * (dl); \text{ update Flags;}$

Machine code: $\overset{w}{11110110} \overset{Md}{11} \overset{R/M}{101} \overset{010}{} \equiv F6EAh$

Example 2: $\text{imul word ptr [si][bx]} \equiv dx:ax \leftarrow (ax) * (M_{(si)+(bx)}); \text{ update Flags;}$

Machine code: $\overset{w}{11110111} \overset{Md}{00} \overset{R/M}{101} \overset{000}{} \equiv F728h$

25

(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**

for 16-bit displacement

1111011	w	Md	110	R/M	Disp. Low-byte	Disp. High-byte

$\text{div operand}_{\text{byte}} \equiv al \leftarrow (ax) / (\text{operand}_{\text{byte}}); ah \leftarrow \text{Remainder}; \text{ update F; if } w=0$
 $\text{div operand}_{\text{word}} \equiv ax \leftarrow (dx:ax) / (\text{operand}_{\text{word}}); dx \leftarrow \text{Remainder}; \text{ update F; if } w=1$

Example 1: $\text{div bh} \equiv al \leftarrow (ax) / (bh); ah \leftarrow \text{Remainder}; \text{ update Flags;}$

Machine code: $\overset{w}{11110110} \overset{Md}{11} \overset{R/M}{110} \overset{111}{} \equiv F6F7h$

Example 2: $\text{div word ptr [bx]} \equiv ax \leftarrow (dx:ax) / (M_{(bx)}); dx \leftarrow \text{Remainder}; \text{ update Flags;}$

Machine code: $\overset{w}{11110111} \overset{Md}{00} \overset{R/M}{110} \overset{111}{} \equiv F737h$

26

(c) Hamid Sarbazi-Azad Computer Structure & Language, Lecture#3: Arithmetic instructions 27

The 8086/88's Arithmetic Instructions:

Division Instructions:

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

- Integer Divide (Signed): IDIV**

for 16-bit displacement

1111011 w	Md	111	R/M	Disp. Low-byte	Disp. High-byte
-----------	----	-----	-----	----------------	-----------------

$\text{idiv operand}_{\text{byte}} \equiv \text{al} \leftarrow (\text{ax}) / (\text{operand}_{\text{byte}}); \text{ah} \leftarrow \text{Remainder}; \text{update F}; \text{if } w=0$
 $\text{idiv operand}_{\text{word}} \equiv \text{ax} \leftarrow (\text{dx:ax}) / (\text{operand}_{\text{word}}); \text{dx} \leftarrow \text{Remainder}; \text{update F};$
if w=1

Example 1: $\text{idiv dh} \equiv \text{al} \leftarrow (\text{ax}) / (\text{dh}); \text{ah} \leftarrow \text{Remainder}; \text{update Flags};$

Machine code: $11110110 \overset{w}{11} \overset{Md}{111} \overset{R/M}{110} \equiv \text{F6FEh}$

Example 2:

$\text{idiv word ptr [bp][di]} \equiv \text{ax} \leftarrow (\text{dx:ax}) / (\text{M}_{(\text{bp})+(\text{di})}); \text{dx} \leftarrow \text{Remainder};$
update Flags;

Machine code: $11110111 \overset{w}{00} \overset{Md}{111} \overset{R/M}{011} \equiv \text{F73Bh}$

27

(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**

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

10011000

$\text{ax} \leftarrow (\text{ax});$

- Covert Word to Double-Word: CWD**

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

10011001

$\text{dx:ax} \leftarrow (\text{ax});$

28

(c) Hamid Sarbazi-AzadComputer Structure & Language, Lecture#3: Arithmetic instructions29

End of Slides