Lecture on Arithmetic Instructions (cont'd)

The SUB, SBB, DEC, AAS, DAS, NEG instructions

Subtraction instructions are similar to the addition instructions

Mnemonic	Meaning	Format	Operation	Flags affected
SUB	Subtract	SUB D,S	(D)-(S) → (D) Borrow → (CF)	All
SBB	Subtract with borrow	SBB D,S	(D)-(S)-(CF) → (D)	All
DEC	Decrement by one	DEC D	(D)-1 → (D)	All but CF
NEG	Negate	NEG D		All
DAS	Decimal adjust for subtraction	DAS	Convert the result in AL to packed decimal format	All
AAS	ASCII adjust for subtraction	AAS	(AL) difference (AH) dec by 1 if borrow	CY,AC

The subtract (SUB) instruction is used to subtract the source from the destination
The subtract with borrow (SBB) is similar to SUB but also subtract the carry flag
From the destination.

EX: Assume the contents of registers BX and CX are 1234H and 0123H, respectively, and the carry flag is 0, what is the result of executing the following instruction SBB BX, CX

The instruction implements the operation

$$(BX) - (CX) - (CF) \longrightarrow (BX)$$

Therefore, (BX) = 1234H – 0123H – 0 = 1111H
The carry flag remains cleared since no borrow is needed.

The decrement (DEC) instruction is used to subtract 1 from its operand. It does not affect the carry flag.

EX: Assume BX has 0000H,What is the results of executing the following instruction: DEC BX

Subtracting 1 from 0000H would produce FFFFH However, the carry flag will remain unchanged.

The negate (NEG) instruction replaced its operand by its negative. The carry flag will be become 1 as a result of this instruction.

- Assume (BX)=003AH, what is the result of executing the instruction

    NEG BX

(BX)=0000000000111010

2’s comp = 1111111111000110 =

= FFC6H and (CF)=1

The SUB and SBB instruction can subtract numbers represented in ASCII and BCD. Just like in addition, the results must be adjusted To produce the corresponding decimal numbers. The instructions AAS, and DAS are provided to perform the adjustments in AL.

```
EX: MOV BL, 28H
MOV AL, 83H
SUB AL,BL;  AL=5BH
DAS             ;  adjust as AL=55H

EX: MOV AX, 38H
SUB AL,39H;  AX=00FF
AAS ;  AX=FF09 ten's complement of -1(Borrow one from AH )
OR AL,30H  ;   AL=39
```

EX: 32-bit subtraction of two 32 bit numbers X and Y that are stored in the memory as

X = (DS:203h)(DS:202h)(DS:201h)(DS:200h)

Y = (DS:103h)(DS:102h)(DS:101h)(DS:100h)

- The result X - Y to be stored where X is saved in the memory

```
MOV SI, 200h
MOV DI, 100h
MOV AX, [SI]
SUB AX, [DI]
MOV [SI], AX ;save the LS word of result
MOV AX, [SI] +2
SBB AX, [DI]+2
MOV [SI] +2, AX
```

Ex. 12 34 56 78 – 23 45 67 89 = EF EE EE EE

Multiplication and Division

Multiplication and division can be performed on signed or unsigned numbers For unsigned numbers, MUL and DIV instructions are used, while for signed numbers IMUL and IDIV are used.

The format of the multiplication and division instructions does not specify the multiplicand as it is implicitly specified depending on the size of the source As shown in the following tables.

Multiplication (MUL or IMUL)	Multiplicand	Operand (Multiplier)	Result
Byte*Byte	AL	Register or memory	AX
Word*Word	AX	Register or memory	DX :AX

Division (DIV or IDIV)	Dividend	Operand (Divisor)	Quotient: Remainder
Word/Byte	AX	Register or Memory	AL : AH
Dword/Word	DX:AX	Register or Memory	AX : DX

For the unsigned division, if the quotient is larger than FF in the division By a byte and FFFF in the division by a word type 0 interrupt occurs.

For the signed division, if the quotient is positive and exceeds 7F for the byte division or 7FFF for the word division type 0 interrupts occurs. However, if the quotient is negative and less than 81H for byte division and 8001H for word division, then type 0 interrupt occurs.

The following examples explains the steps to perform multiplication and division Using calculators to give same results as the 88/86 processors.

Ex1: Assume that each instruction starts from these values:

AL = 85H, BL = 35H, AH = 0H

1. MUL BL → AL . BL = 85H * 35H = 1B89H → AX = 1B89H

2. IMUL BL → AL . BL = 2'S AL * BL = 2'S (85H) * 35H
= 7BH * 35H = 1977H→ 2's comp → E689H → AX.

3. DIV BL → $\frac{AX}{BL} = \frac{0085H}{35H}$ = 02 (85-02*35=1B) →

AH	AL
1B	02

4. IDIV BL → $\frac{AX}{BL} = \frac{0085H}{35H} =$

AH	AL
1B	02

Ex2: AL = F3H, BL = 91H, AH = 00H

1. MUL BL $\rightarrow$ AL * BL = F3H * 91H = 89A3H $\rightarrow$ AX = 89A3H

2. IMUL BL $\rightarrow$ AL * BL = 2'S AL * 2'S BL = 2'S (F3H) * 2'S(91H) = 0DH * 6FH = 05A3H $\rightarrow$ AX.

3. IDIV BL $\rightarrow$ $\dfrac{AX}{BL} = \dfrac{00F3H}{2'S(91H)} = \dfrac{00F3H}{6FH}$ = 2 $\rightarrow$ (00F3 – 2*6F=15H)

AH	AL
15	02
R	Q

$\rightarrow \dfrac{POS}{NEG} = NEG$ $\rightarrow$ 2's(02) = FEH $\rightarrow$

AH	AL
15	FE

4. DIV BL $\rightarrow$ $\dfrac{AX}{BL} = \dfrac{00F3H}{91H}$ = 01 $\rightarrow$ (F3-1*91=62) $\rightarrow$

AH	AL
62	01
R	Q

Ex3: AX= F000H, BX= 9015H, DX= 0000H

1. MUL BX = F000H * 9015H =

DX	AX
8713	B000

2. IMUL BX = 2’S(F000H) * 2’S(9015H) = 1000 * 6FEB =

DX	AX
06FE	B000

3. DIV BL = $\frac{F000H}{15H}$ = B6DH → More than FFH → Divide Error.

4. IDIV BL → $\frac{2'S(F000H)}{15H} = \frac{1000H}{15H}$ = C3H > 7F → Divide Error.

Ex4: AX= 1250H, BL= 90H

1. IDIV BL $\rightarrow \dfrac{AX}{BL} = \dfrac{1250H}{90H} = \dfrac{POS}{NEG} = \dfrac{POS}{2'sNEG} = \dfrac{1250H}{2's(90H)} = \dfrac{1250H}{70H}$

= 29H (Q) $\rightarrow$ (1250 – 29 * 70) = 60H (REM)

29H (*POS*) $\rightarrow$ 2'S (29H) = D7H $\rightarrow$

R	Q
60H	D7H

2. DIV BL $\rightarrow \dfrac{AX}{BL} = \dfrac{1250H}{90H}$ = 20H$\rightarrow$1250-20*90 =50H $\rightarrow$

R	Q
50H	20H
AH	AL

To divide an 8-bit dividend by and 8-bit divisor by extending the sign bit Of Al to fill all bits of AH. This can be done automatically by executing the Instruction (CBW).

In a similar way 16-bit dividend in AX can be divided by 16-bit divisor. In this case the sign bit in AX is extended to fill all bits of DX. The instruction CWD perform this operation automatically.

Note that CBW extend 8-bit in AL to 16-bit in AX while the value in AX will Be equivalent to the value in AL. Similarly, CWD convert the value in AX to 32-bit In (DX,AX) without changing the original value.