Microprocessor

Just Note: Arithmetic instructions

Addition instruction— ADD, ADC, INC, AAA, and DAA

ADD	Add byte or word
ADC	Add byte or word with carry
INC	Increment byte or word by 1
AAA	ASCII adjust for addition
DAA	Decimal adjust for addition

Example

Assume that AX and BX registers contain 1100_{16} and $0ABC_{16}$, respectively. What is the result of executing ADD AX,BX — verify using DEBUG

$AX = AX + BX = 1100_{16} + 0ABC_{16} = 1BBC_{16}$

-r ax
AX 0000
:**1100**
-r bx
BX 0000
:**abc**
-r bx
BX 0ABC
:
-a 0100
13B2:0100 **add ax,bx**
13B2:0102
-r ip
IP 0100
:
-t
AX=1BBC BX=0ABC CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0102 NV UP EI PL NZ NA PO NC
13B2:0102 0000 ADD [BX+SI],AL DS:0ABC=00

ADC— add with carry

ADC works similarly to ADD but in this case, the content of the carry flag is also added

INC— Increment

Execution of the INC instruction adds 1 to the specified operand

Example

The original contents of AX, BL, word-size memory location SUM, and carry flag (CF) are 1234_{16}, AB_{16}, $00CD_{16}$, and 0_{16}, respectively. Describe the results of executing the following sequence of instructions:

```
ADD AX,[SUM]
ADC BL,05H
INC WORD PTR [SUM]
```

$AX = 1234_{16} + 00CD_{16} = 1301_{16}$
$BL = BL + 05 + \text{carry flag} = AB_{16} + 5_{16} + 0_{16} = B0_{16}$
$SUM = SUM + 1_{16} = 00CD_{16} + 1_{16} = 00CE_{16}$

Example

Verify the above example using DEBUG

-r ax
AX 0000
:**1234**
-r bx
BX 0000
:**ab**
-r f
NV UP EI PL NZ NA PO NC -
-e 0 cd 00
-d 0 1
13B2:0000 CD 00 ..
-a 0100
13B2:0100 **add ax,[0000]**
13B2:0104 **adc bl,05**
13B2:0107 **inc word ptr [0000]**

13B2:010B
-r ip
IP 0100
:
-t

AX=1301 BX=00AB CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0104 NV UP EI PL NZ AC PO NC
13B2:0104 80D305 **ADC BL,05**
-t

AX=1301 **BX=00B0** CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0107 NV UP EI NG NZ AC PO NC
13B2:0107 FF060000 **INC WORD PTR [0000] DS:0000=00CD**
-t

AX=1301 BX=00B0 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=010B NV UP EI PL NZ NA PO NC
13B2:010B 0000 ADD [BX+SI],AL DS:00B0=00
-d 0 1
13B2:0000 **CE 00** ..

AAA— ASCII adjust for addition
DAA— Decimal adjust for addition

Whenever the processor does an addition on ASCII format data, an adjustment must be performed on the binary result to convert it to the equivalent decimal number. The AAA instruction should be executed immediately after the ADD instruction that adds ASCII data. If the sum is greater than 9, AL contains the LSD and AH is incremented by 1. Otherwise, AL contains the sum and AH is unchanged

-r ax
AX 0000
:**0035**
-r bx
BX 0000
:**0036**
-a 0100
13B2:0100 **add al,bl**
13B2:0102 **aaa**

13B2:0103
-t
AX=006B BX=0036 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0102 NV UP EI PL NZ NA PO NC
13B2:0102 37 **AAA**
-t
AX=0101 BX=0036 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0103 NV UP EI PL NZ AC PE CY
13B2:0103 0080D305 ADD [BX+SI+05D3],AL DS:0609=00

Subtraction instruction— SUB, SBB, DEC, AAS, DAS, and NEG

SUB	Subtract
SBB	Subtract with borrow
DEC	Decrement
AAS	ASCII Adjust for subtraction
DAS	Decimal adjust for subtraction
NEG	Negate

Solution
Assuming that the contents of registers BX and CX are 1234_{16} and 0123_{16}, respectively, and the carry flag is 0, what will be the result of executing the following instruction?

```
SBB BX,CX
```

BX = BX – CX – carry flag = 1234_{16} - 0123_{16} – 0 = 1111_{16}

-r bx
BX 0000
:**1234**
-r cx
CX 0000
:**0123**
-a 0100
13B2:0100 **sbb bx,cx**
13B2:0102
-t
AX=0000 **BX=1111** CX=0123 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0102 NV UP EI PL NZ NA PE NC
13B2:0102 27 DAA

NEG causes the value of its operand to be replaced by its negative. That is, the contents of the specified operand are subtracted from zero and the result is returned to the operand location

-r ax
AX 0000
:**34fc**
-a 0100
13B2:0100 **neg ax**
13B2:0102
-t
AX=CB04 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0102 NV UP EI NG NZ AC PO CY
13B2:0102 27 DAA

Multiplication and Division Instruction

MUL	Multiply byte or word unsigned
IMUL	Integer multiply byte or word
AAM	ASCII Adjust for multiply
DIV	Divide byte or word unsigned
IDIV	Integer divide byte or word
AAD	ASCII adjust for division
CBW	Convert byte to word
CWD	Convert word to double word in DX and AX
CDQ	Convert double word to quad word

MUL S8	AL*S8 → AX
MUL S16	AX*S16 →DX,AX
DIV S8	AX / S8 → Q: AL R : AH
DIV S16	DX,AX / S16 → Q: AX R: DX

Example
The 2's complement signed data contents of AL equal -1 and the contents of CL are -2. What result will be produced in AX by executing the following instruction?
MUL CL

-r ax
AX 0000
:**00ff**
-r cx

CX 0000
:**00fe**
-a 0100
13B2:0100 **mul cl**
13B2:0102
-t
AX=FD02 BX=0000 CX=00FE DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0102 OV UP EI PL NZ NA PO CY
13B2:0102 27 DAA

Example
What is the result of executing the following sequence of instructions?
MOV AL,A1
CBW ; extend the most significant bit of AL, which is 1, into all bits of AH
CWD ; extend the most significant bit of AX, which is 1, into all bits of DX

Solution

-a 0100
13B2:0100 **mov al,a1**
13B2:0102 **cbw**
13B2:0103 **cwd**
13B2:0104
-t
AX=00A1 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0102 NV UP EI PL NZ NA PO NC
13B2:0102 98 **CBW**
-t
AX=FFA1 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0103 NV UP EI PL NZ NA PO NC
13B2:0103 99 **CWD**
-t
AX=FFA1 BX=0000 CX=0000 **DX=FFFF** SP=FFEE BP=0000 SI=0000 DI=0000
DS=13B2 ES=13B2 SS=13B2 CS=13B2 IP=0104 NV UP EI PL NZ NA PO NC
13B2:0104 80D305 ADC BL,05