Lecture 06: Arithmetic Instructions. The ADD, ADC, INC, AAA, DAA Instructions

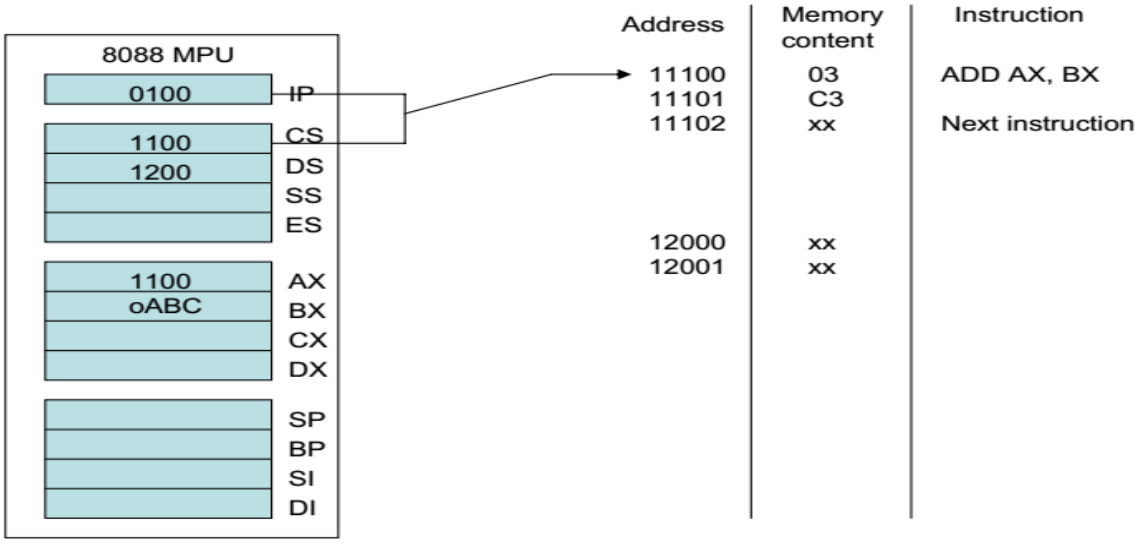
The instruction set of the 8088/8086 microprocessor contains a set of arithmetic instructions. These include instructions for addition, subtraction, multiplication, and division operation. The form of each of the addition instructions is shown in the following table:

Mnemonic	Meaning	Format	Operation	Flags affected
ADD	Addition	ADD D,S	(S)+(D) ⟶ (D) carry ⟶ (CF)	ALL
ADC	Add with carry	ADC D,S	(S)+(D)+(CF) ⟶ (D) carry ⟶ (CF)	ALL
INC	Increment by one	INC D	(D)+1 ⟶ (D)	ALL but CY
AAA	ASCII adjust for addition	AAA	If the sum is >9, AH is incremented by 1	AF,CF
DAA	Decimal adjust for addition	DAA	Adjust AL for decimal Packed BCD	ALL

- The destination on all addition instructions can not be immediate number and no memory to memory can be added.
- Example: Assume that AX and BX registers contain 1100H and 0ABCH, respectively. What is the result of executing the instruction ADD AX, BX

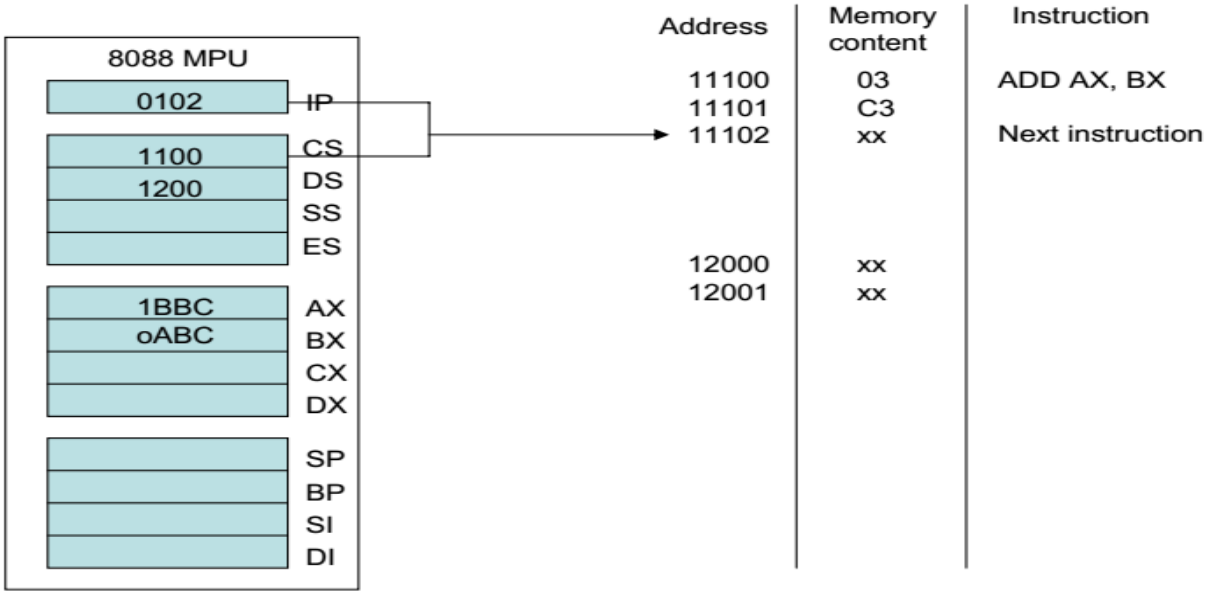
The content of the source (BX) will be added to the content of the destination (AX) to give 0ABC + 1100 = 1BBC in the destination AX. The process of executing this instruction is shown in the following figures.

ADD instruction before execution



Lecture 06: Arithmetic Instructions. The ADD, ADC, INC, AAA, DAA Instructions

After execution of ADD instruction



The instruction ADC works similarly to ADD, but the carry flag is also added; that is

(S)+(D)+(CF) ⟶ (D)

The ADC instruction is used for multiword add operation.

The Increment instruction INC adds 1 to the only operand in the instruction.

Example: The original contents of AX=1234H, BL=ABH, word content of memory location with P.A. of DS:1234H=[DS:1234H] ➔00CDH and carry flag CF=NC=0……

Describe the results of executing the following sequence of instructions:

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

Lecture 06: Arithmetic Instructions. The ADD, ADC, INC, AAA, DAA Instructions

The first instruction add the word in the memory location identified as SUM to the accumulator.

(AX) ⟵ (AX)+([DS:1234H]) = 1234H + 00CDH = 1301H

The carry flag remains 0.

The second instruction adds to the register (BL) the immediate operand 5H and the carry flag to give:

(BL) ⟵ (BL) + IOP + (CF) = ABH + 5H + 0 = B0H

Since no carry out is generated, the CF stays reset.

The last instruction increments the contents of word-size memory location SUM by 1 to give:

([DS:1234H]) ⟵ [DS:1234H] + 1 = 00CDH +1 = 00CEH.

Ex- Perform the following 32-bit binary add operation

(DX,CX) ⟵ (DX,CX) + (BX,AX)

For the following data in the registers

(DX,CX) = FEDCBA98H

(BX,AX) = 01234567H

The least 16 significant bits of the 32-bit numbers are added

ADD CX, AX

To add the most significant 16 bits we must account for the possibility Of a carry out from the addition of the lower 16 bits. Therefore, ADC Instruction must be used

ADC DX, BX

Numbers in addition instructions can be represented in ASCII code. However, An adjustment must be performed on the binary result to convert it to the equivalent decimal number. AAA is provided for such conversion to be used after ADD instruction that adds ASCII data. AAA Causes the contents of AL to be replaced by its equivalent decimal number.

Ex. The ASCII CODE of Numbers 0-9 is 30-39h

```
MOV AX,38H; (ASCII code for number 8)
ADD AL,39H;   (ASCII code for number 9) AL=71h
AAA; used for addition      AH=01, AL=07
ADD AX,3030H; convert answer to ASCII    0107     AX=3137
```

Lecture 06: Arithmetic Instructions. The ADD, ADC, INC, AAA, DAA Instructions

DAA is used to adjust results of adding packed BCD numbers instead of ASCII Numbers.

Ex. Assume AL contains 25 (packed BCD)
BL contains 56 (packed BCD)
ADD AL, BL
DAA

The addition of 25 + 56 would produce 7B, however, DAA will convert This result to the correct answer in packed BCD which is 81

Example:

```
.MODEL SMALL                    ; program that adds two multiword numbers:
.STACK 100
.DATA
DATA1 DW 1D54H,8F99H,63CEH      ; allocate 3 words
ORG 0010h
DATA2 DW 3F73H, 4FA2H,3B8DH     ; ; allocate 2nd 3 words
ORG 0020h
DATA3 DW 3 DUP( ?) ; save results
.CODE
      MAIN PROC FAR
      MOV AX,@DATA              ; receive the starting address for DATA
      MOV DS,AX
      CLC                       ; clear carry
      MOV SI,OFFSET DATA1       ; LEA for DATA1
      MOV DI,OFFSET DATA2       ; LEA for DATA2
      MOV BX,OFFSET DATA3       ; LEA for DATA3
      MOV CX,03
BACK: MOV AX,[SI]
      ADC AX, [DI]              ; add with carry to AX
      MOV [BX],AX
      ADD SI,2
      ADD DI,2
      ADD BX,2
      DEC CX
      JNZ BACK                  ; decrement CX automatically until zero
      MOV AH,4Ch
      INT 21h; halt
 MAIN ENDP
 END MAIN
```