28/12/2021

8086 Microprocessor

Arithmetic Instructions

ADD , ADC , SUB , SBB , INC , DEC , MUL , DIV

① Addition Instruction :-

It support 8-bit and 16-bit addition.

ADD destination, Source.

Register, memory
memory, Register
Register, Register
Register, Immediate data.
memory, Immediate data.

Algorithm for its execution :-

operand at destination = operand at destination + operand at Source.

Note : Result is not stored in accumulator as was the case in 8085 rather result is always stored in the destination operand.

For eg: ① ADD AL, BL

AL ← (AL) + (BL)

② ADD BX, 2401H

BX ← (BX) + 2401

Size of Source and destination must be Same.

③ ADD CL, [0428H]

↳ memory.

How will you find MA?.

$MA = BA + EA$

$= DS \times 16_{10} + 0428H$

$= 2000 \times 16_{10} + 0428H$

$= 20428H.$

(assumed that DS = 2000H)

$CL = CL +$ 8-bit data at MA

④ ADD CX, [BX]

↳ memory.

$CX \leftarrow CX +$ 16-bit data at MA

$MA = BA + EA$

$= DS \times 16_{10} + BX$

⑤ ADD [0410H], 0425H.

destination is memory ; source is immediate data.

16-bit addition.

$MA = BA + EA$

$= DS \times 16_{10} + 0410H$

$MA \leftarrow MA + 0425H$

→ You cannot give destination and source both as memory.

ADD [BX], [0210H] ✗ not allowed.

ADC destination, Source — Addition with Carry flag.

Algorithm for its operation :-

destination operand ← destination operand + Source operand + Carry flag value.

All other things of ADC are Similar to ADD.

(2) Subtraction Instruction :-

SUB destination, Source.

Register, memory.
memory, Register
Register, Register
memory, immediate data
Register, immediate data.

Algorithm for its operation :-

destination operand ← destination operand − Source operand.

→ Both 8-bit and 16-bit operations are supported.
→ But Size of destination operand must match with the Size of source operand
→ SUB [BX], [BX+SI] ✗ not allowed.

SBB destination, Source.

destination operand ← destination operand − Source operand − value of carry flag.

All other things of SBB as similar to SUB.

③ Multiplication Instruction

MUL Source

destination is by default accumulator

* for 8-bit multiplication the result is stored in AX register.
* for 16-bit multiplication the result is stored in DX and AX register.

8-bit multiplication

MUL Register → AX = AL X Register

MUL memory → AX = AL X memory.

For eg: ① MUL BL

AX ← (AL) X (BL)

② MUL [0420H]

AX ← (AL) X (MA)

MA = BA + EA

= DS X 16_{10} + 0420H

16-bit multiplication

MUL Register → DX (H) | AX (L) = AX × 16-bit Register

MUL memory → DX | AX = AX × 16-bit memory.

Higher-order 16-bit will be stored in DX

Lower-order 16-bit will be stored in AX.

For eg: ↑MUL BX

DX | AX ← (AX) × (BX)

MUL → it is unsigned multiplication instruction

iMUL → it is Signed multiplication instruction

MUL → flags that are affected : Overflow
Carry flag.

S, Z, A, P indeterminate.

iMUL → flags that are affected : S, O, C

(4) Increment Instruction :-

INC destination (Add 1 to destination value)

destination can be Register or memory.
(8), (16)

① INC BX

$BX \leftarrow (BX) + 1$

assume BX = 0200H.

INC BX

after execution; BX = 0201H.

② INC [0204H].

memory → MA = BA + EA

= DS X 16_{10} + EA.

$MA \leftarrow (MA) + 1$

⑤ Decrement Instruction :-

DEC destination. (Decrements destination value by 1).

destination can be register or memory.

For eg:- DEC BX.

$BX \leftarrow (BX) - 1$

⑥ Negate Instruction :- makes operand negative.

NEG destination (performs 2's complement of destination)

destination can be register or memory.