21/12/2021

8086 μp Instructions

(1) Data - Transfer Instruction

8086 μp supports :- data movement between registers, registers and memory, registers and immediate data, memory and immediate data.

MOV Destination, Source.

Destination	Source.
Register	Immediate data
memory	Immediate data
Register	Register
Register	memory
memory	Register
Segment	memory.
memory	Segment
Register	Segment
Segment	Register.

All the possible destination and source for 8086.

→ Direct loading of the segment register with immediate data is not allowed.

→ MOV instruction will not be able to set the values of CS and IP register.

→ MOV will not be able to Copy the value of one segment register To another segment register.

MOV Register, Immediate data.
↳ 8-bit / 16-bit.

Eg: ① MOV CL, 82H

It will load CL register with the 8-bit immediate data 82H.

② MOV CX, 2030H.

CX = 2030H.

③ MOV CX, 20H ✗ not valid.

CX ↙ 16-bit reg.

20H ↓ 8-bit value.

→ Size of register and immediate data should be same.

MOV memory, Immediate data.

8086
[ ] Square brackets are used for memory.

Eg: MOV [0345H], 23H.

MA = BA + EA
$= DS \times 16_{10} + 0345H$
$= 2000 \times 16_{10} + 0345H$
$= 20000 + 0345H = \boxed{20345H.}$ ↙ 23H value will be stored.

② MOV [BX], 45H

$MA = BA + EA$

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

$= 2000 \times 16_{10} + 4502H$

$= \boxed{24502H.}$ ← 45H will be stored.

③ MOV [0345H], 2345H.

16-bit immediate data can be stored in memory.

$MA = BA + EA$

$= DS \times 16_{10} + 0345H$

$= 2000 \times 16_{10} + 0345H$

$= \boxed{20345H.}$ ← 45H.

20346H ← 23H.

$\overset{H}{\underbrace{23}}\,\overset{L}{\underbrace{45}}.$

④ MOV [BX], 2104H.

$MA = BA + EA$

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

$= 2000 \times 16_{10} + 2424H.$

$= \boxed{22424H.}$ ← 04H.

$\boxed{22425H}$ ← 21H.

MOV SS:[BX], 4232H.

$MA = BA + EA$

$= SS \times 16_{10} + EA$

→ we have used segment override prefix.

MOV Accumulator, memory.

Eg: 8-bit operation. (always use AL as accumulator).

MOV AL, [0203H]

$MA = BA + EA$

$= DS \times 16_{10} + 0208H.$

AL reg.

$= 2000 \times 16_{10} + 0208H.$

$= \boxed{20208H}$

data will be copied.

16-bit operation (always use AX as accumulator).

MOV AX, [0208H]

$\boxed{20208H} \rightarrow$ AL register.

$\boxed{20209H} \rightarrow$ AH register.

MOV memory, accumulator

Eg: MOV [BX], AL → 8-bit operation

MOV [BX], AX → 16-bit operation.

MOV memory, Register

MOV Register, memory.

MOV Register, Register. → MOV CL, AL

mov CX, AX

size of register must be same.

MOV ~~memory~~, memory. ← not allowed.

MOV CX, [BX]

$$
\begin{aligned}
MA &= BA + EA. \\
&= DS \times 16_{10} + BX. \\
&= 2000 \times 16_{10} + 1200H. \\
&= \boxed{21200H} \rightarrow CL. \\
& \quad \boxed{21201H} \rightarrow CH.
\end{aligned}
$$

MOV CL, [BX]

$\boxed{21200H} \rightarrow CL.$

mov CH, [BX]

$\boxed{21200h} \rightarrow CH.$

Q Move the content of DX register into SS Register.

MOV SS, DX.

Q Load 16-bit data from memory location offset address is 0300H. to AX registe.

MOV AX, [0300]

XCHG Destination, Source.

(Exchange data between source to destination).

Destination	Source.
↓	
Accumulator	Register.
memory	Register.
Register.	Register.

↳ Exchange of contents of two memory locations is not allowed.

Eg: XCHG [4000], AX

MA = BA + EA

= DS × 16₁₀ + 4000

= 24000H.

24000H ⇌ AL

24001H ⇌ AH