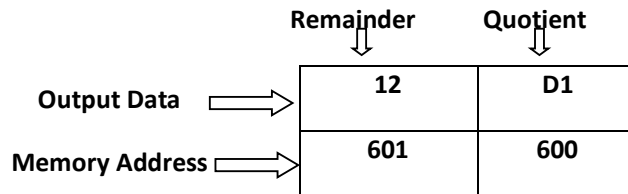
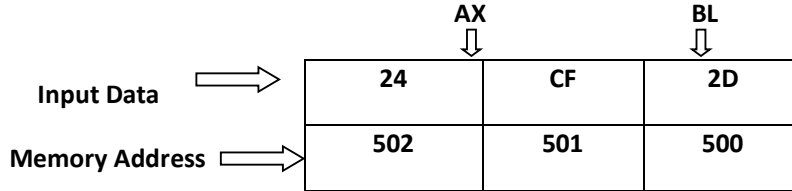


## ALP to divide a 16 bit number 24CF by an 8 bit number 2D in 8086 $\mu$ P



### Algorithm –

1. Assign value 500 in **SI** and 600 in **DI**.
2. Move the contents of **[SI]** in **BL** and increment **SI** by 1.
3. Move the contents of **[SI]** and **[SI+1]** in **AX**.
4. Use **DIV** instruction to divide **AX** by **BL**.
5. Move the contents of **AX** in **[DI]**.
6. Halt the program.

Assumption – Initial value of each segment register is 0000H.

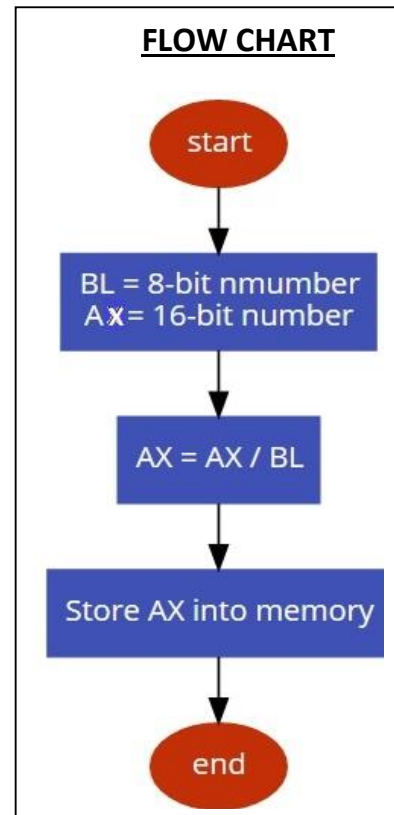
### Calculation of physical memory address –

Memory Address = Segment Register \* 10(H) + Offset,

where Segment Register and Offset are decided on the basis of following table :

OPERATIONS	SEGMENT REGISTER	OFFSET
Instruction fetching	Code Segment	Instruction Pointer
Data operation	Data Segment	Base Register <b>[BX]</b> , Displacement <b>[DISP]</b>
Stack operation	Stack Segment	Stack Pointer <b>[SP]</b> , Base Pointer <b>[BP]</b>
String as a source	Data Segment	Source Indexed ( <b>SI</b> )
String as a destination	Extra Segment	Destination Indexed ( <b>DI</b> )

### FLOW CHART



## Program –

MEMORY ADDRESS	MNEMONICS	COMMENTS
0400	MOV SI, 500	SI←500
0403	MOV DI, 600	DI←600
0406	MOV BL, [SI]	BL←[SI]
0408	INC SI	SI←SI + 1
0409	MOV AX, [SI]	AX←[SI]
040B	DIV BL	AX←AX/BL
040D	MOV [DI], AX	[DI]←AX
040F	HLT	End of program

**Explanation** - Registers used are AX, BL, SI and DI.

1. **MOV SI, 500** assigns **500** to **SI**.
2. **MOV DI, 600** assigns **600** to **DI**.
3. **MOV BL,[SI]** moves the content of **[SI]** to **BL** register i.e. value of divisor will be stored in **BL**.
4. **INC SI** increments the content of **SI** by **1**.
5. **MOV AX, [SI]** moves the content of **[SI]** and **[SI+1]** to **AX** register i.e. value of dividend is stored in **AX**.
6. **DIV BL** divides the content of **AX** by **BL**, after execution of this instruction the quotient gets stored in **AL** and remainder in **AH**.
7. **MOV [DI], AX** moves the content of **AX** to **[DI]**.
8. **HLT** stops executing the program.