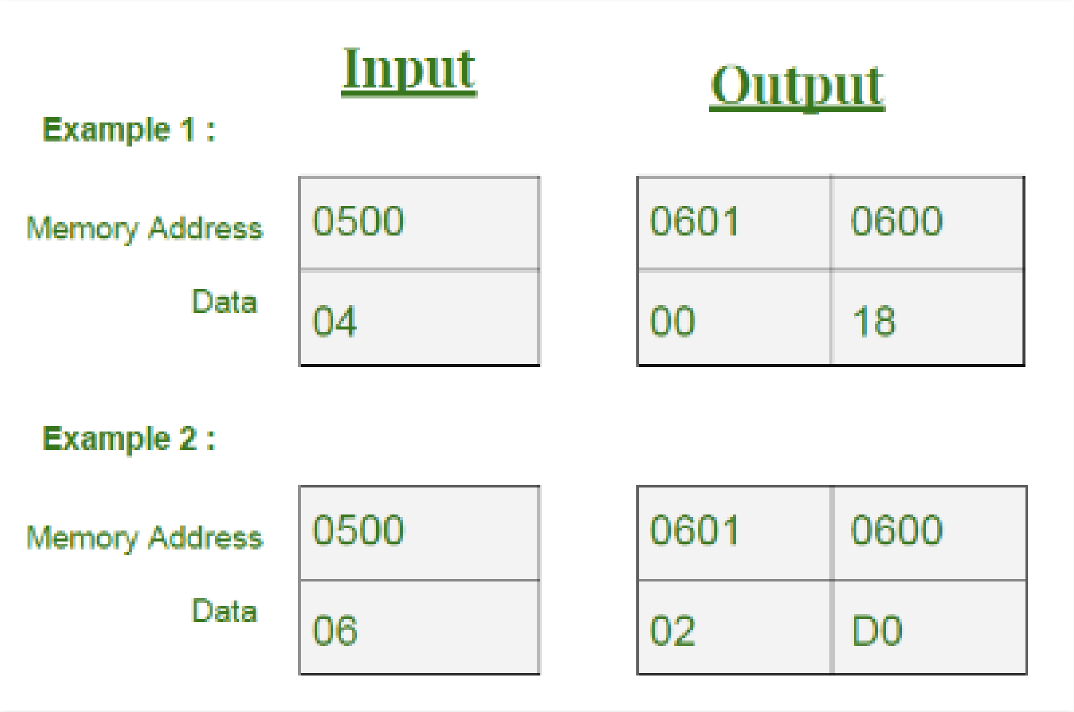
UNIT-II
8086 Assembly language programme

Write an assembly language program for calculating the factorial of a number using 8086 microprocessor.

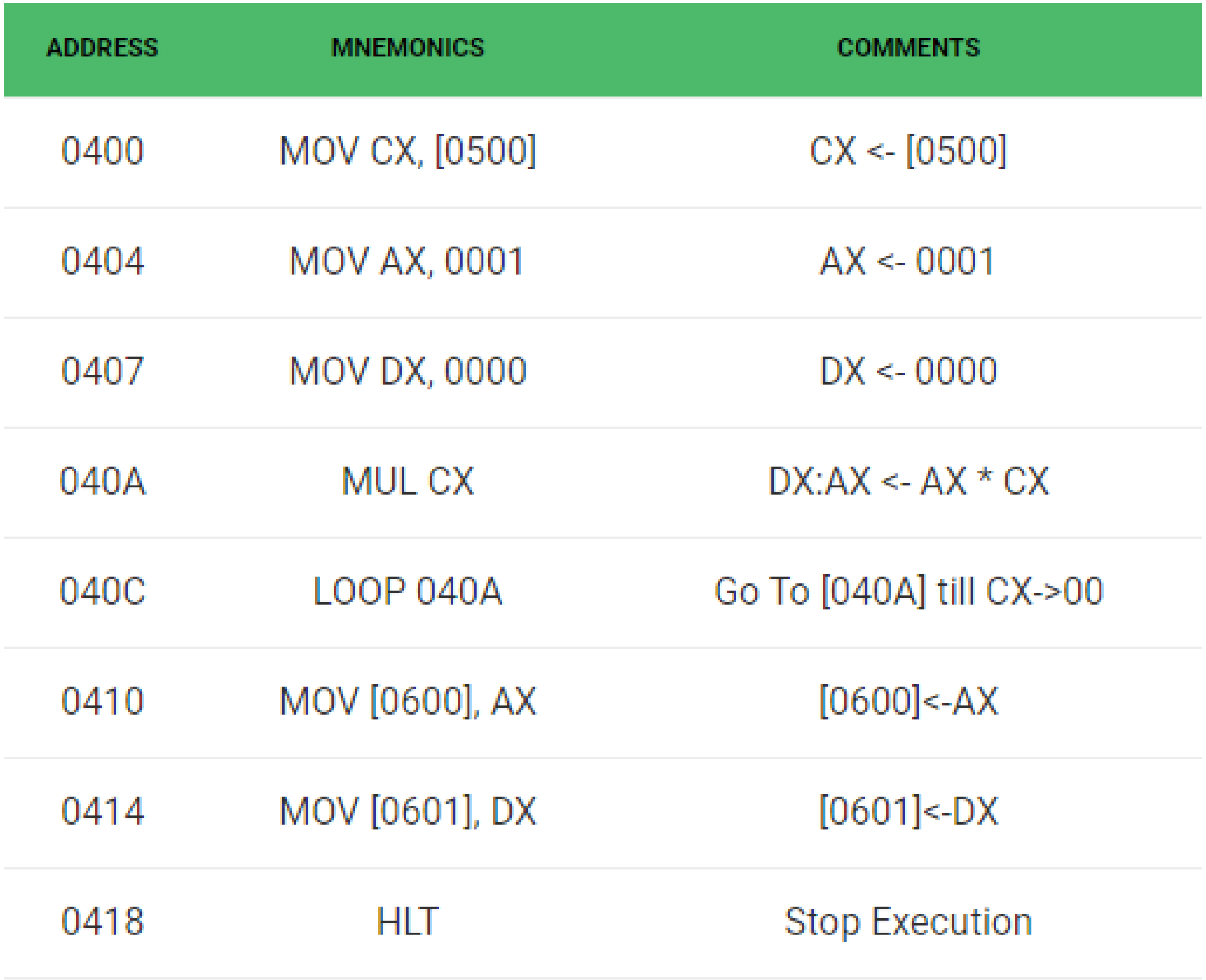


Example 1 :

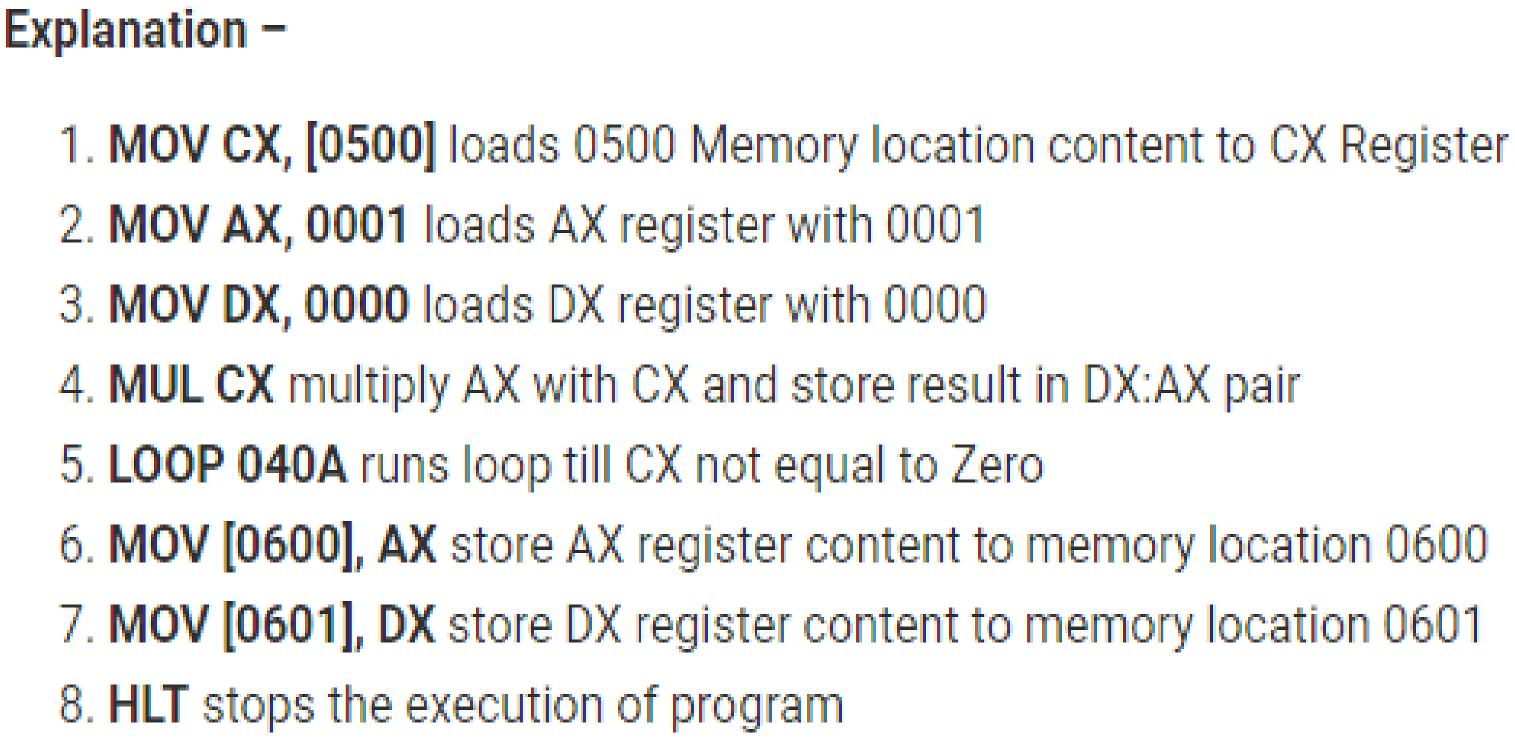
	Input	Output	
Memory Address	0500	0601	0600
Data	04	00	18

Example 2 :

	Input	Output	
Memory Address	0500	0601	0600
Data	06	02	D0



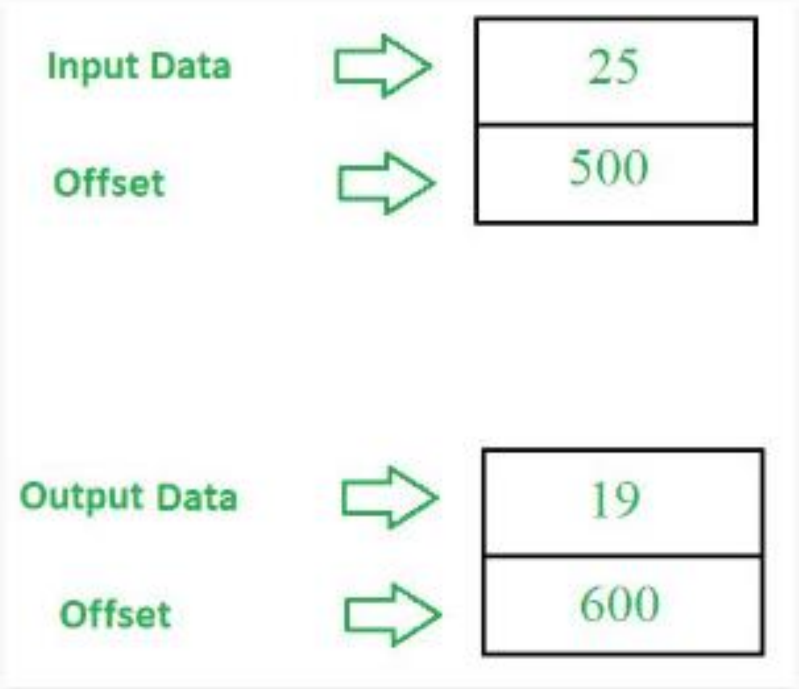
ADDRESS	MNEMONICS	COMMENTS
0400	MOV CX, [0500]	CX <- [0500]
0404	MOV AX, 0001	AX <- 0001
0407	MOV DX, 0000	DX <- 0000
040A	MUL CX	DX:AX <- AX * CX
040C	LOOP 040A	Go To [040A] till CX->00
0410	MOV [0600], AX	[0600]<-AX
0414	MOV [0601], DX	[0601]<-DX
0418	HLT	Stop Execution



Explanation –

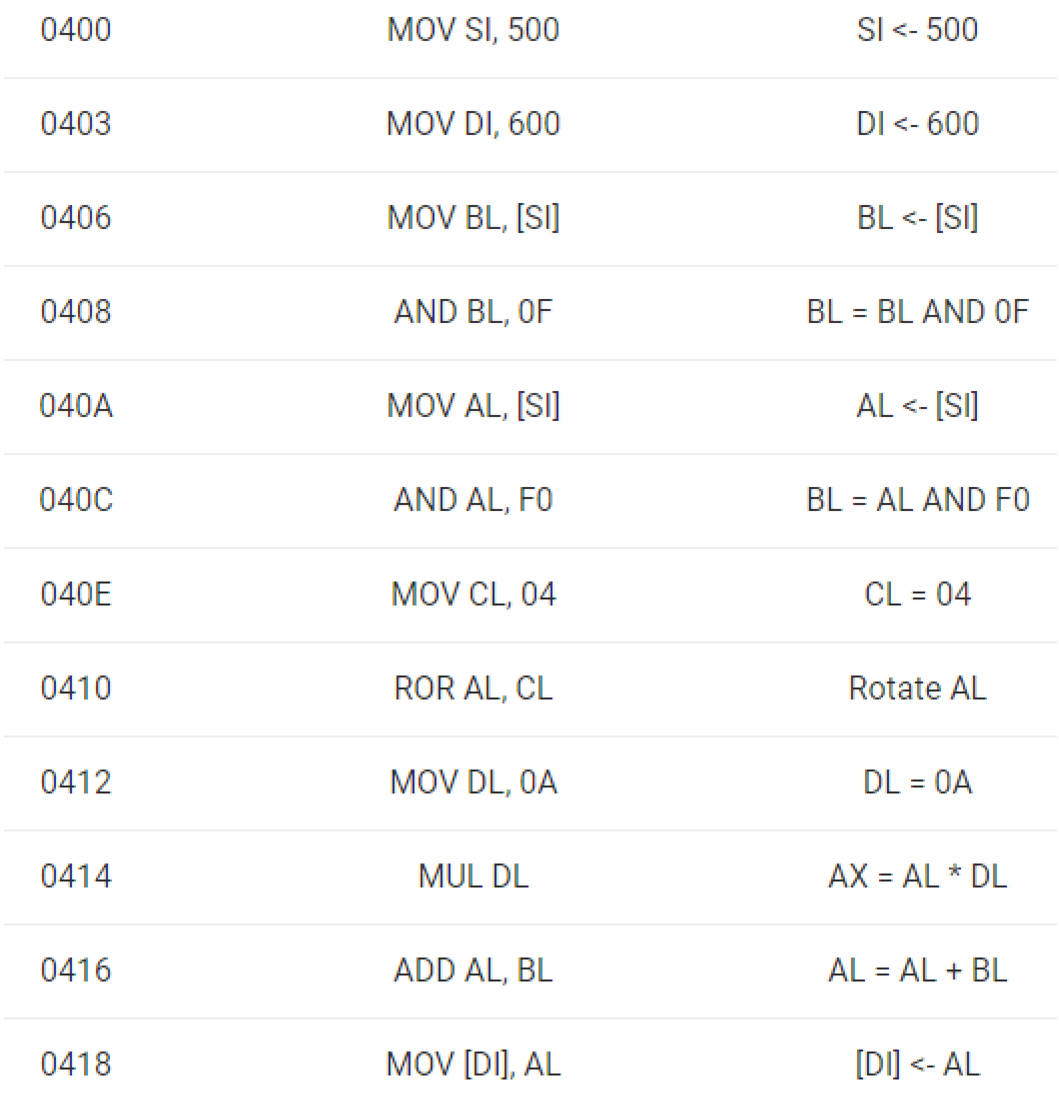
1. **MOV CX, [0500]** loads 0500 Memory location content to CX Register
2. **MOV AX, 0001** loads AX register with 0001
3. **MOV DX, 0000** loads DX register with 0000
4. **MUL CX** multiply AX with CX and store result in DX:AX pair
5. **LOOP 040A** runs loop till CX not equal to Zero
6. **MOV [0600], AX** store AX register content to memory location 0600
7. **MOV [0601], DX** store DX register content to memory location 0601
8. **HLT** stops the execution of program

2. Write an assembly language program in 8086 microprocessor to convert an 8 bit BCD number into hexadecimal number.

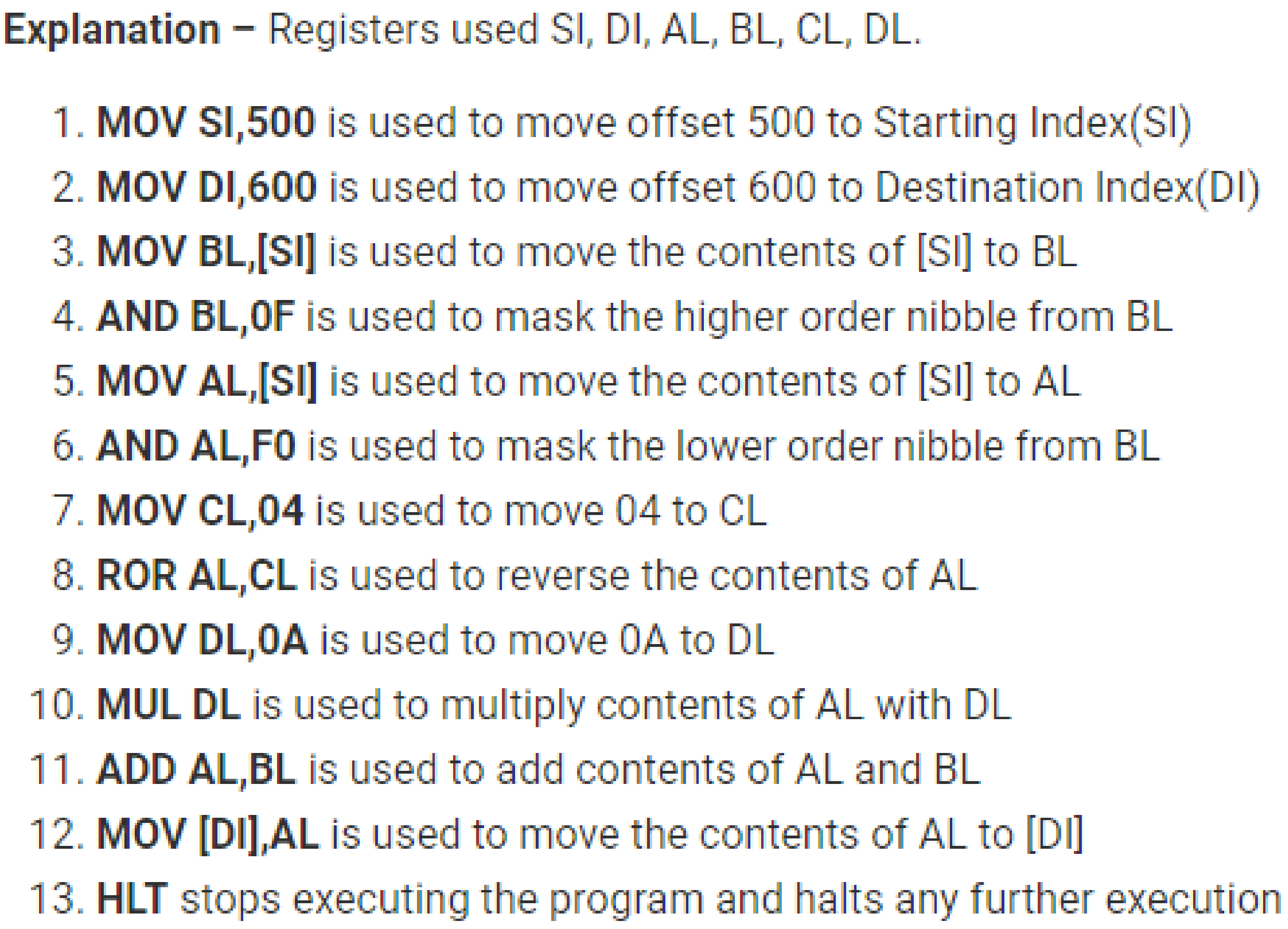


Algorithm –

Assign value 500 in SI and 600 in DI.
Move the contents of [SI] in BL.
Use **AND** instruction to calculate AND between 0F and contents of BL.
Move the contents of [SI] in AL.
Use **AND** instruction to calculate AND between F0 and contents of AL.
Move 04 in CL.
Use **ROR** instruction on AL.
Move 0A in DL.
Use **MUL** instruction to multiply AL with DL.
Use **ADD** instruction to add AL with BL.
Move the contents of AL in [DI].
Halt the program.



0400	MOV SI, 500	SI <- 500
0403	MOV DI, 600	DI <- 600
0406	MOV BL, [SI]	BL <- [SI]
0408	AND BL, 0F	BL = BL AND 0F
040A	MOV AL, [SI]	AL <- [SI]
040C	AND AL, F0	BL = AL AND F0
040E	MOV CL, 04	CL = 04
0410	ROR AL, CL	Rotate AL
0412	MOV DL, 0A	DL = 0A
0414	MUL DL	AX = AL * DL
0416	ADD AL, BL	AL = AL + BL
0418	MOV [DI], AL	[DI] <- AL



Explanation – Registers used SI, DI, AL, BL, CL, DL.

1. **MOV SI,500** is used to move offset 500 to Starting Index(SI)
2. **MOV DI,600** is used to move offset 600 to Destination Index(DI)
3. **MOV BL,[SI]** is used to move the contents of [SI] to BL
4. **AND BL,0F** is used to mask the higher order nibble from BL
5. **MOV AL,[SI]** is used to move the contents of [SI] to AL
6. **AND AL,F0** is used to mask the lower order nibble from BL
7. **MOV CL,04** is used to move 04 to CL
8. **ROR AL,CL** is used to reverse the contents of AL
9. **MOV DL,0A** is used to move 0A to DL
10. **MUL DL** is used to multiply contents of AL with DL
11. **ADD AL,BL** is used to add contents of AL and BL
12. **MOV [DI],AL** is used to move the contents of AL to [DI]
13. **HLT** stops executing the program and halts any further execution

THANK YOU