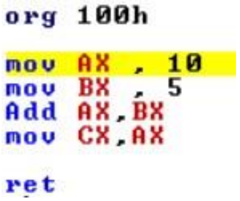
EXPERIMENT - 7

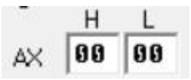
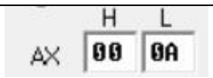
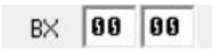
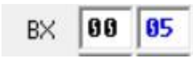
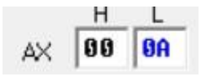
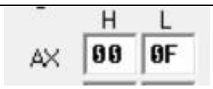
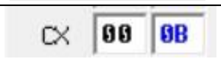
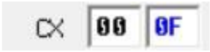
AIM: Perform following operations on 8-bit data.

OBJECTIVE 1 : Perform Addition operation

Code:

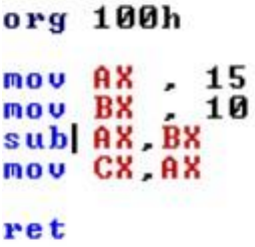


```
org 100h

mov AX , 10
mov BX , 5
Add AX,BX
mov CX,AX

ret
```

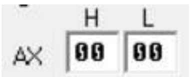
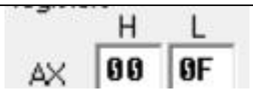
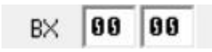
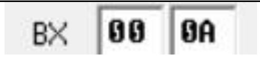
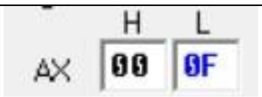
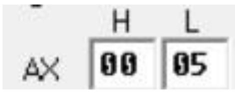
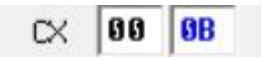
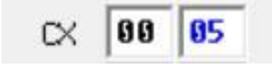
Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 10	AX = 10	 H L AX 00 00	 H L AX 00 0A
Mov BX , 5	BX = 5	 BX 00 00	 BX 00 05
Add AX,BX	AX = AX + BX	 H L AX 00 0A	 H L AX 00 0F
Mov CX, AX	CX = AX	 CX 00 0B	 CX 00 0F

OBJECTIVE 2 : Perform Subtraction operation

Code:

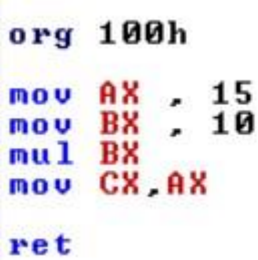


```
org 100h

mov AX , 15
mov BX , 10
sub AX,BX
mov CX,AX

ret
```

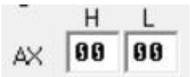
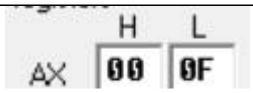
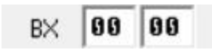
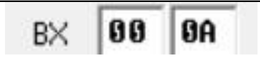
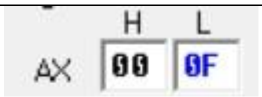
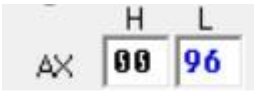
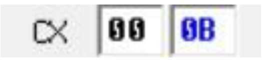
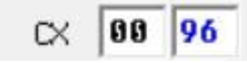
Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 15	AX = 15	 H L AX 00 00	 H L AX 00 0F
Mov BX , 10	BX = 10	 BX 00 00	 BX 00 0A
Sub AX,BX	AX = AX - BX	 H L AX 00 0F	 H L AX 00 05
Mov CX, AX	CX = AX	 CX 00 0B	 CX 00 05

OBJECTIVE 3 : Perform Multiplication operation

Code:

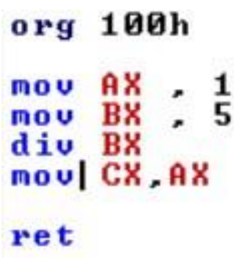


```
org 100h

mov AX , 15
mov BX , 10
mul BX
mov CX,AX

ret
```

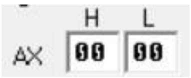
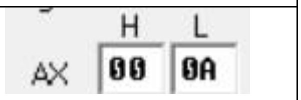
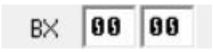
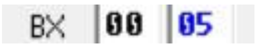
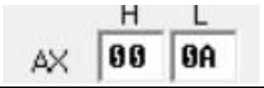
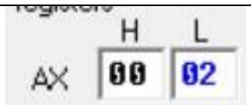
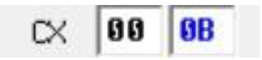
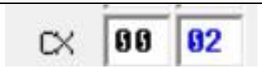
Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 15	AX = 15	 H L AX 00 00	 H L AX 00 0F
Mov BX , 10	BX = 10	 BX 00 00	 BX 00 0A
Mul BX	AX = AX * BX	 H L AX 00 0F	 H L AX 00 96
Mov CX, AX	CX = AX	 CX 00 0B	 CX 00 96

OBJECTIVE 4 : Perform Division operation

Code:

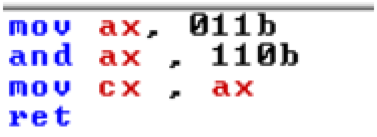


```
org 100h

mov AX , 10
mov BX , 5
div BX
mov| CX,AX

ret
```

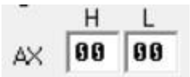
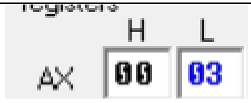
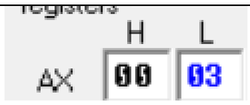
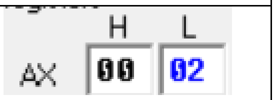
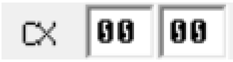
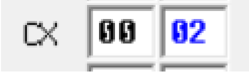
Instruction in assembly language	**Operation executed by instruction**	**Before execution**	**After execution**
Mov AX , 10	AX = 10	 H L AX 00 00	 H L AX 00 0A
Mov BX , 5	BX = 5	 BX 00 00	 BX 00 05
Div AX,BX	AX = AX / BX	 H L AX 00 0A	 H L AX 00 02
Mov CX, AX	CX = AX	 CX 00 0B	 CX 00 02

OBJECTIVE 5 : Perform AND operation

Code:

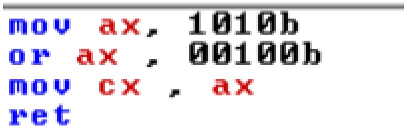


```
mov ax, 011b
and ax , 110b
mov cx , ax
ret
```

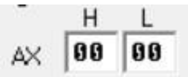
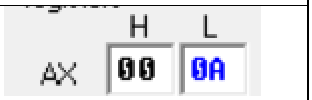
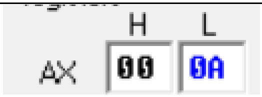
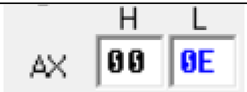
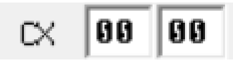
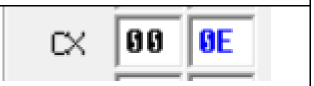
Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 011b	AX = 0003h	 AX H 00 L 00	 AX H 00 L 03
And AX ,	AX = 00006h	 AX H 00 L 03	 AX H 00 L 02
MOV CX,AX	CX = 0006h	 CX 00 00	 CX 00 02

OBJECTIVE 6 : Perform ON operation

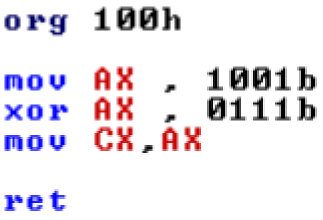
Code:



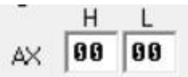
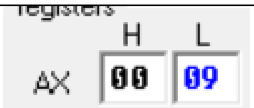
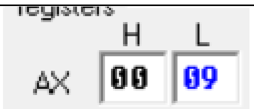
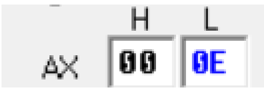
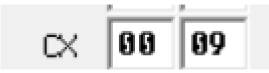
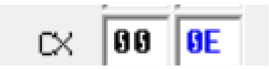
```
mov ax, 1010b
or ax , 00100b
mov cx , ax
ret
```

Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 1010b	AX = 0000Ah	 AX H 00 L 00	 AX H 00 L 0A
And AX ,	AX = 00004h	 AX H 00 L 0A	 AX H 00 L 0E
MOV CX,AX	CX = 0000Eh	 CX 00 00	 CX 00 0E

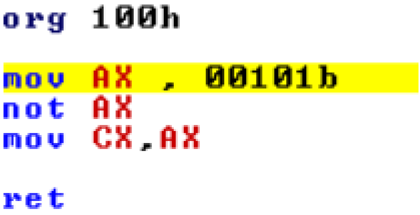
OBJECTIVE 7 : Perform XOR operation Code:



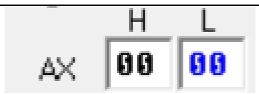
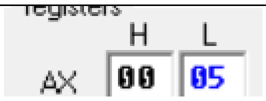
```
org 100h

mov AX , 1001b
xor AX , 0111b
mov CX,AX

ret
```

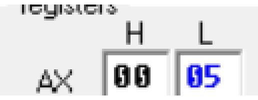
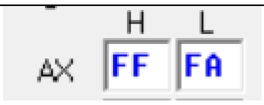
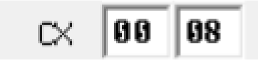
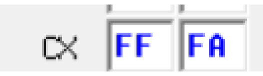
Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 1001b	AX = 1001b	 H L AX 00 00	 H L AX 00 09
XOR AX , 0111b	AX = 0111b	 H L AX 00 09	 H L AX 00 0E
MOV CX ,AX	CX = 1110	 CX 00 09	 CX 00 0E

OBJECTIVE 8 : Perform XOR operation Code:

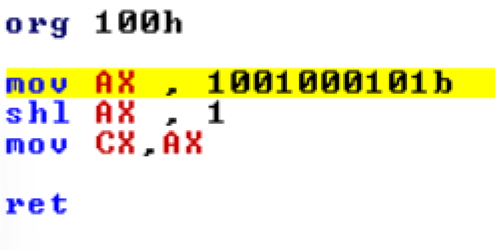


```
org 100h

mov AX , 00101b
not AX
mov CX,AX

ret
```

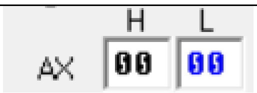
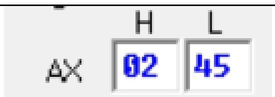
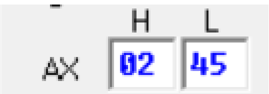
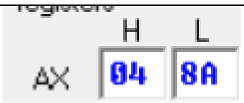
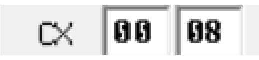
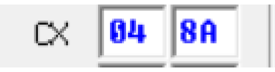
Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 00101b	AX = 00101b	 H L AX 00 00	 H L AX 00 05

NOT AX ,	AX = 11010b	 H L AX 00 05	 H L AX FF FA
MOV CX ,AX	CX = 11010b	 CX 00 08	 CX FF FA

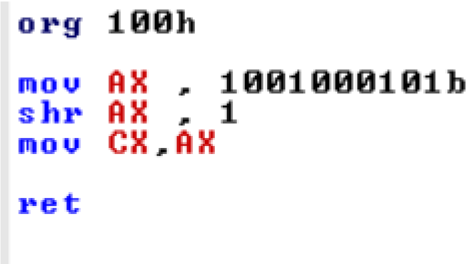
OBJECTIVE 9 : Perform Logical left shift operation Code:



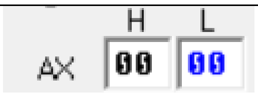
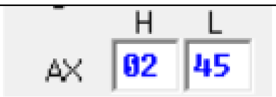
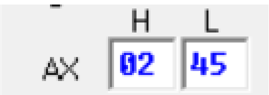
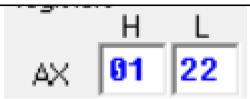
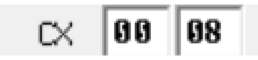
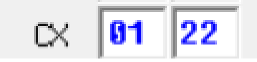
```
org 100h

mov AX , 1001000101b
shl AX , 1
mov CX,AX

ret
```

Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 100100101b	AX = 100100101b	 H L AX 00 00	 H L AX 02 45
SHL AX , 1	AX = 001001010b	 H L AX 02 45	 H L AX 04 8A
MOV CX ,AX	CX =001001010b	 CX 00 08	 CX 04 8A

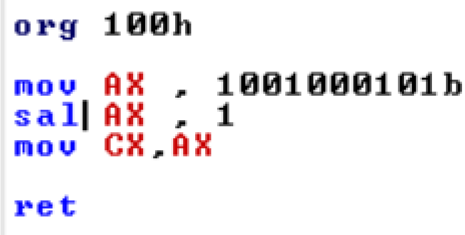
OBJECTIVE 10 : Perform Logical right shift operation Code:



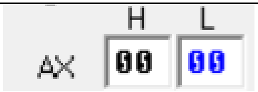
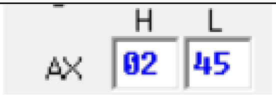
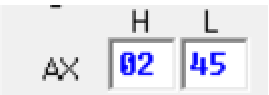
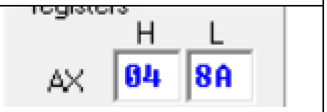
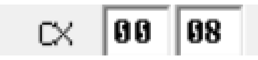
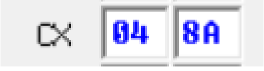
```
org 100h

mov AX , 1001000101b
shr AX , 1
mov CX,AX

ret
```

Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 100100101b	AX = 100100101b	 H L AX 00 00	 H L AX 02 45
SHR AX , 1	AX = 010010010b	 H L AX 02 45	 H L AX 01 22
MOV CX ,AX	CX =010010010b	 CX 00 08	 CX 01 22

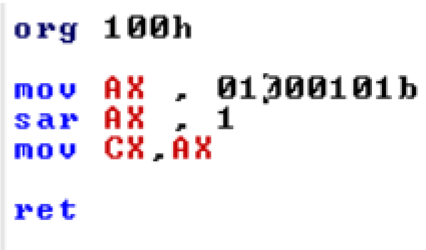
OBJECTIVE 11 : Perform Arithmetic left shift operation Code:



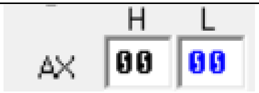
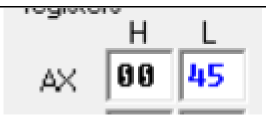
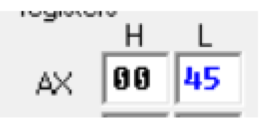
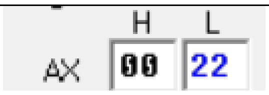
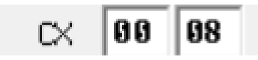
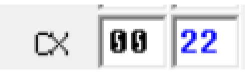
```
org 100h

mov AX , 1001000101b
sal AX , 1
mov CX,AX

ret
```

Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 100100101b	AX = 100100101b	 H L AX 00 00	 H L AX 02 45
SAL AX , 1	AX = 001001010b	 H L AX 02 45	 H L AX 04 8A
MOV CX ,AX	CX =001001010b	 CX 00 08	 CX 04 8A

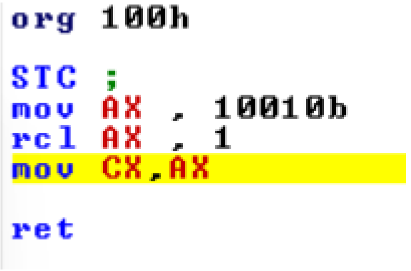
OBJECTIVE 12 : Perform Arithmetic right shift operation Code:



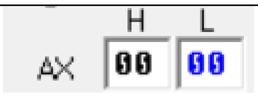
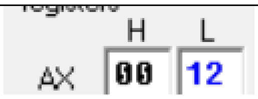
```
org 100h

mov AX , 01000101b
sar AX , 1
mov CX,AX

ret
```

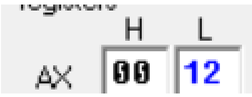
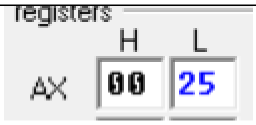
Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX , 01000101b	AX = 01000101b	 H L AX 00 00	 H L AX 00 45
SAR AX , 1	AX = 00100010b	 H L AX 00 45	 H L AX 00 22
MOV CX ,AX	CX =00100010b	 CX 00 08	 CX 00 22

OBJECTIVE 13 : Perform Rotate left with carry operation Code:

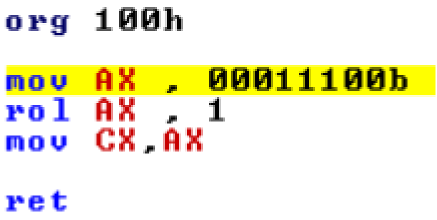


```
org 100h

STC ;
mov AX , 10010b
rcl AX , 1
mov CX,AX

ret
```

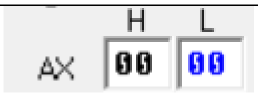
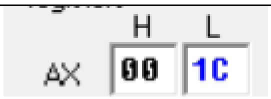
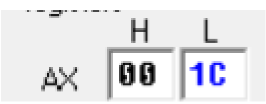
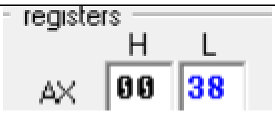
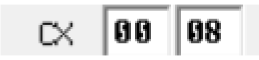
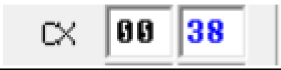
Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX ,	AX = 10010b	 H L AX 00 00	 H L AX 00 12

RCL AX , 1	AX = 100101b	 H L AX 00 12	 registers H L AX 00 25

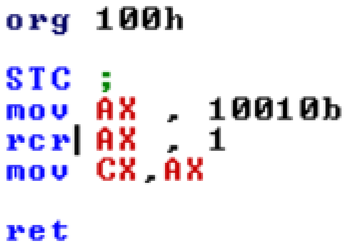
OBJECTIVE 14 : Perform Rotate left without carry operation Code:



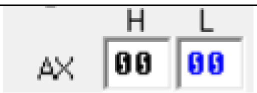
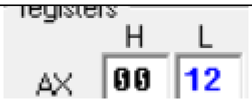
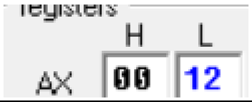
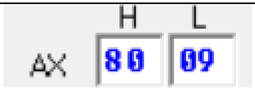
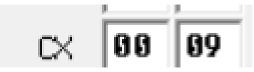
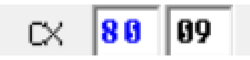
```
org 100h

mov AX , 00011100b
rol AX , 1
mov CX,AX

ret
```

Instruction in assembly language	Operation executed by instruction	Before execution	After execution
Mov AX ,	AX = 00011100b	 H L AX 00 00	 H L AX 00 1C
ROL AX , 1	AX = 00111000b	 H L AX 00 1C	 registers H L AX 00 38
MOV CX ,AX	CX =00111000b	 CX 00 08	 CX 00 38

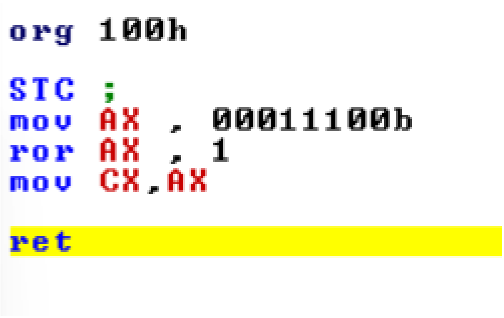
OBJECTIVE 15 : Perform Rotate right with carry operation Code:



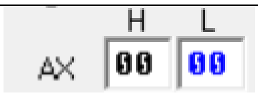
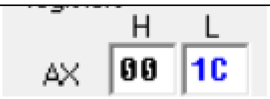
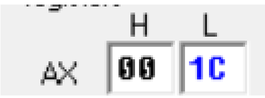
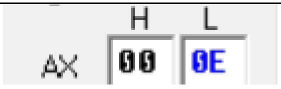
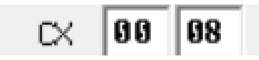
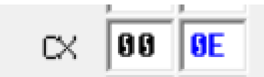
```
org 100h

STC ;
mov AX , 10010b
rcr AX , 1
mov CX,AX

ret
```

Instruction in assembly language	**Operation executed by instruction**	**Before execution**	**After execution**
Mov AX ,	AX = 10010b	 H L AX 00 00	 registers H L AX 00 12
RCR AX , 1	AX = 10001001b	 registers H L AX 00 12	 H L AX 80 09
MOV CX ,AX	CX =10001001b	 CX 00 09	 CX 80 09

OBJECTIVE 16 : Perform Rotate right without carry operation Code:



```
org 100h

STC ;
mov AX , 00011100b
ror AX , 1
mov CX,AX

ret
```

Instruction in assembly language	**Operation executed by instruction**	**Before execution**	**After execution**
Mov AX ,	AX = 00011100b	 H L AX 00 00	 H L AX 00 1C
ROR AX , 1	AX = 00001110b	 H L AX 00 1C	 H L AX 00 0E
MOV CX ,AX	CX =00001110b	 CX 00 08	 CX 00 0E