MIT Assignment/Lab 3

SAHIL BONDRE: U18CO021

Assignment 3:

1 . WALP to check the forth bit of a byte stored at location 3000H is 0 or 1. If 0 store 00h else store FFH at location 3002H.

```
;<Check 4th bit set>

jmp start

;data
;code
start: nop
lda 3000H
rrc
rrc
rrc
rrc
lxi H, 3002H
mvi B, 0
mov M, B
jnc end
mvi B, 0FFH
mov M, B
end: hlt
```

Address (Hex)	Address	Data
3000	12288	8
3001	12289	0
3002	12290	255
3003	12291	0

4th bit set

Address (Hex)	Address	Data
3000	12288	4
3001	12289	0
3002	12290	0
3003	12291	0
3004	12292	0

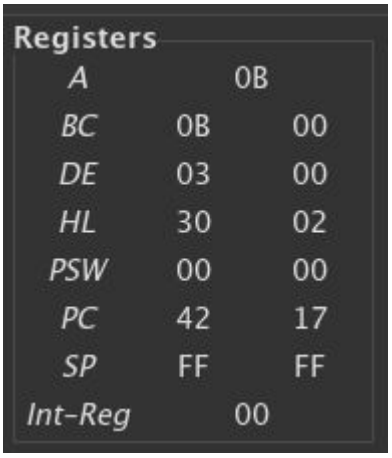
4th bit not set

2. Write Assembly language program to count the number of 1s in 8-bit number stored in register B.

```
;<Number of set bits in B>

jmp start

;data
;code
start: nop
mvi B, 0BH
; loop 8 times
mvi C, 8
mov A, B
; counter for number of bits
mvi D, 0
loop: nop
rrc
jnc skip
; increase count if bit is set
inr D
skip: nop
; decrease loop conuter
dcr C
jnz loop
hlt
```



0BH has 3 bits set. So D is 03

3. There is an array of some elements. Write Assembly language program to count number of elements that are lesser than 09H.

```
;<Count Number of elements less than 09H>

jmp start

;data
;code
start: nop
; size of array
mvi C, 08H
; array location
lxi H, 3000H
; counter
mvi B, 0
; given number
mvi D, 09H
loop: nop
mov A, M
cmp D
jnc skip
inr B
skip: nop
dcr C
inx H
jnz loop
hlt
```

Address (Hex)	Address	Data
3000	12288	4
3001	12289	5
3002	12290	10
3003	12291	12
3004	12292	1
3005	12293	1
3006	12294	9
3007	12295	2
3008	12296	0

Given array of size 8

Registers		
A	02	
BC	05	00
DE	09	00
HL	30	08
PSW	00	00
PC	42	1B
SP	FF	FF
Int-Reg	00	

B with 5 elements less than 09H

Lab 3 Solutions:

1. Store the data byte 32H into memory location 4000H

```
;code
start: nop
lxi H, 4000H
mvi M, 32H

hlt
```

Address (Hex)	Address	Data
4000	16384	50
4001	16385	0

2. Exchange the contents of memory locations 2000H and 4000H

```
start: nop
; swap
mov E, M
lxi H, 2000H
mov L, M
xchg
mov D, L

; store swapped values
lxi H, 4000H
mov M, E
lxi H, 2000H
```

```
mov M, D

hlt
```

Address (Hex)	Address	Data
4000	16384	12

Address (Hex)	Address	Data
2000	8192	20

initial value

Address (Hex)	Address	Data
4000	16384	20

Address (Hex)	Address	Data
2000	8192	12

final value

(3) Add two 8-bit numbers: Add the contents of memory locations 4000H and 4001H and place the result in memory location 4002H.

```
start: nop
mvi A, 0
lxi H, 4000H
add M
inx H
add M
sta 4002H
hlt
```

Address (Hex)	Address	Data
4000	16384	12
4001	16385	15
4002	16386	27

(4) Subtract two 8-bit numbers: Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.

```
;code
start: nop
lda 2000H
lxi H, 4000H
mov B, M
sub B
sta 4002H

hlt
```

Address (Hex)	Address	Data
2000	8192	20

Address (Hex)	Address	Data
4000	16384	8
4001	16385	0
4002	16386	12

[4002] = [2000] - [4000]

(5) Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.

```
;code
start: nop
lxi H,4000H
mov E,M
inx H
mov D,M
inx H
mov C,M
inx H
mov B,M
mov H,B
mov L,C
dad D
shld 4004H

hlt
```

Address (Hex)	Address	Data
4000	16384	1
4001	16385	12
4002	16386	4
4003	16387	8
4004	16388	5
4005	16389	20

(6) Add contents of two memory locations: Add the contents of memory locations 4000H and 4001H and place the result in the memory locations 4002H and 4003H.

```
lda 4000H
lxi H, 4001H
add M
sta 4003H
jnc end
lxi H, 4002H
mvi M, 1
end: htl
```

Address (Hex)	Address	Data
4000	16384	12
4001	16385	45
4002	16386	0
4003	16387	57

(7) Write a program for one's complement of 8 bit number.

```
;code
start: nop
mvi A, 0
lda 4000H
cma
sta 4001H
hlt
```

Address (Hex)	Address	Data
4000	16384	5
4001	16385	250

(8) Write a program for two's complement of 8 bit number.

```
;code
start: nop
mvi A, 0
lda 4000H
cma
adi 01
sta 4001H
hlt
```

Address (Hex)	Address	Data
4000	16384	5
4001	16385	251

(9) Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.

```
lhld 4002H
xchg
lhld 4000H
mov a,e
sub l
sta 4004H
mov a,d
sbb H
sta 4005H
hlt
```

Address (Hex)	Address	Data
4000	16384	1
4001	16385	57
4002	16386	12
4003	16387	34
4004	16388	11
4005	16389	233

(10) Write a program using the ADI instruction to add the two hexadecimal numbers 3AH and 48H and store the result in memory location 2100H.

```
;code
start: nop
```

```
mvi A, 0
adi 3AH
adi 48H
sta 2100H
hlt
```

Address (Hex)	Address	Data
2100	8448	130

(11) Write an assembly language program that AND, OR and XOR together the contents of register B, C and E and place the result into memory location 3000H, 3001H and 3002H.

```
mvi B, 12H
mvi C, 34H
mvi E, 42H
mvi A, 00H
mov A, B
ana C
ana E
sta 3000H
mov A, B
ora C
ora E
sta  3001H
mov A, B
xra C
xra E
sta 3002H
hlt
```

Address (Hex)	Address	Data
3000	12288	0
3001	12289	119
3002	12290	99

(12) Program to Find 1's Complement of 16-bit Number

```
lhld 3000H
mov A, L
cma
mov L, A
```

```
mov A, H
cma
mov H, A
shld 3002H
hlt
```

Address (Hex)	Address	Data
3000	12288	12
3001	12289	2
3002	12290	243
3003	12291	253

(13) Program to Find 2's Complement of 16-bit Number

```
lhld 3000H
mov A, L
cma
mov L, A
mov A, H
cma
mov H, A
inx H
shld 3002H
hlt
```

Address (Hex)	Address	Data
3000	12288	12
3001	12289	2
3002	12290	244
3003	12291	253