Date: ______ Day: ______

$\Rightarrow$ AND DL, 0F $\rightarrow \boxed{\text{AND DL, 0F}}$

$\boxed{\text{Reverse order}}$

1011 0010

0000 1111

$\boxed{\text{DL} = 0000\ 0010}$

Date: 10-10-2023

Day: Tuesday

(Lecture 11)

⇒ At end of the program $\boxed{BL}$ = all zeros.

Print bit of 0,1 and count 1's in it

```
       MOV  BL, 47h
       MOV  CH, 8
       MOV  CL, 0
             ROL
again: ~~SHL~~  BL, 1
       JC one
       MOV  DL, 30H
       MOV  AH, 02
       Int  21H
       Jmp exist
one:     MOV  DL, 31H
         IncCL  → count 1's.
         MOV AH, 02
         Int 21H
exist:   DecCH
         JNZ again
         Mov ah, 4ch
         int 21h
```

∴ Inc cl → count 0's

Date:

Day:

Rotates:-

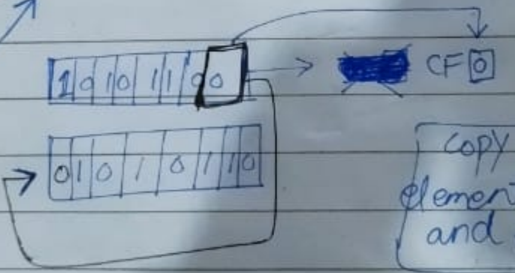
$$\begin{tikzcd}[column sep=small]
 & & \text{Rotates} \arrow[d] & & \\
 & & \text{circular shift} \arrow[dl, dash] \arrow[dr, dash] & & \\
 & \text{Rotate without carry} \arrow[dl, dash] \arrow[d, dash] & & \text{Rotate with carry} \arrow[d, dash] \arrow[dr, dash] & \\
\text{Right (ROR)} & \text{Left (ROL)} & & \text{Right (RCR)} & \text{Left (RCL)}
\end{tikzcd}$$

SHR AL, 1 → ROL RCL

① ROR AL, 1

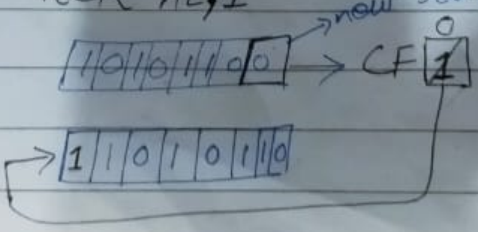
~~Resu~~

Reuse



copy last element in CF and first position.

② RCR AL, 1



stored in CF and CF value is rotated

Date 10-10-2023 Day Tuesday

(1) print Numbers in Hexadecimal :-

BX = 24B6h $\boxed{\text{one way}}$

$\boxed{0010\ 0100\ \cancel{1000}\ 1011\ 0110}$ $\boxed{\text{loop 4 tim}}$

↓
MSB move into registers.

again ~~DL, 0010~~

8bit ← DL, 0000 0010

```
CMP DL, 0  ──→ [Label]
JE Zero
CMP DL, 1
JE one
```

0000
0001
0010
0011
⋮
1001 → 9
1111 → 15

```
Zero :  DL, 30H
        Mov ah, 02
        ~~Mov ah, 4ch~~

        int 21h
        [Jmp exist]  ──→ like break in python.
one :   DL, 31H
        mov ah, 02
        ~~mov ah, 4ch~~
        int 21h
        Jmpexist
Two :

exist : DEC CX
JNZ again
```

Date 10-10-2023 Day Tuesday

1) print Numbers in Hexadecimal :-

Bx = 24B6h $\boxed{\text{one way}}$

$\boxed{0010 \; 0100 \; 1011 \; 0110}$ $\boxed{\text{loop 4 times}}$

↓
MSB move into registers.

again ~~DL, 0010~~

8bit ← DL, 0000 0010

CMP DL, 0 → $\boxed{\text{Label}}$
JE Zero
CMP DL, 1
JE one

0000	0
0001	1
0010	2
0011	3
⋮	
1001	→ 9
1111	→ 15

Zero : DL, 30H
Mov ah, 02
~~Mov ah, 4ch~~

int 21h
$\boxed{\text{Jmp exist}}$ ⟶ like break in python.

one : DL, 31H
mov ah, 02
~~mov ah, 4ch~~

int 21h
Jmp exist

Two :

exist : DEC CX
JNZ again

Date: Day:

another way:

$\boxed{0010\ 0011\ 1010\ 1011}$

```
again:
CMP DL, 09
JBE aa
aa: Add DL, 30H
    mov AH, 02
    int 21H
    Dec CX
    Jnz again
```

Second code

```
CMP DL, 09
JBE aa
aa:  Add DL, 37H
     mov AH, 02
     int 21H
     Jmp exist
exist : Dex CX
        JNZ again
```

first code

0	→ 30H
1	→ 31H
2	→ 32H
3	→ 33H
9	→ 39H
A	→ 41H
B	→ 42H
C	→ 43H
D	
E	
F	

$\boxed{0 \ldots 9}$

$\boxed{10-15}$

$\boxed{\text{Rotate bits}}$

```
MOV  BX, 23ABh
Mov  CX, 4
RoL  BX, 4
Mov  DL, BL
```

$\boxed{0010\ 0011\ 1010\ 1011}$

$\to \boxed{0011\ 1010\ 1011\ 0010}$

$\overset{\downarrow}{\boxed{\text{AND DL, 0}}}$

(COAL LECTURE 9)

Previous code :-

```
Mov BX, 246Bh
Mov Cx, 4
again : ROL BX, 4
        Mov DL, BL
        AND DL, 0Fh
```

Huzaifa

⇒ Input in Binary :-

```
Mov BX, 0000H
Mov CX, 16
Mov AH, 01
int 21h
SUB AL, 30H
if [OR BL, AL
   [SHL BX, 1   (No bit is lost)
DEC CX
JNZ again
```

(DIFFERENTIAL EQUATION

(LECTURE 9) :-

⇒ Switch is changed to position 2 :-

$$L\frac{di}{dt} + iR = 0$$

$$2\frac{di}{} + 4i = 0 \quad \text{——} \quad (A)$$

ln

e

```
da : ADD DL, 30H
     mov AH, 02
     int 21H
[exit : DEC CX
[       JNZ again
[ → optional
```

Q) Input from the user :

(i) Input in binary

(ii) Input in hexadecimal

→ $\boxed{BX}$

```
Mov BX, 0000H
Mov CX, 16
[Mov AH, 01
[int 21h        ⇐ (1) → AL = 31H
SUB AL, 30H
CMP AL, 39H
JLE da
```

Counters :-

For Hex ; CX, 4

For Binary : CX, 16

31H - 30H

↳ 1H

values

↓

add

ASCII values

↓

subtract

```
JLE aa
ADD DL, 37h
mov AH, 02
int 21h
JMP exit
aa: ADD DL, 30H
    mov AH, 02
    int 21h
exit: ROR BX, 4
      DEC CX
      JNZ again
```

Q) Print in binary :

```
2    4    6    8
0010 0100 0110 1011
```

```
Mov BX, 246BH
mov CX, 16
again : ROL BX, 1
        mov DL, BL
        AND DL, 01
        CMP DL, 09  ┐
        JLE aa      │
        ADD DL, 37h │ optional
        mov AH, 02  │
        int 21H     │
        JMP exit    ┘
```

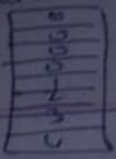
```
CMP DL, 09
JLE ad
ADD DL, 37h
Mov AH, 02
INT 21H
JMP exit
ad : ADD DL, 30H
     Mov AH, 02
     INT 21H
exit : DEC CX
       JNZ again
```

Q) Print numbers in hexadecimal in reverse order.

```
Mov BX, 246Bh
mov CX, 4
again : ROL BX, 4
mov DL, BL
AND DL, 0Fh
CMP DL, 09
JLE ad
ADD DL, 37h
mov AH, 02
int 21h
```

$$\boxed{\begin{array}{c}246Bh\\ \Downarrow\\ B642h\end{array}}$$

var1 dd 1234h



A2FA4

10215104 — 115104

0100000

Compare Instruction

↳ It is like if-else instruction

Syntax:- CMP, AL, BL

JG → ~~If~~ Jump if greater than
JGE → Jump if greater equal
JL → Jump if less than
JLE → Jump if less equel

} signed number

JA → Jump if above
JAE → Jump if above or equal
JB → Jump if below
JBE → Jump if below or equal.

} unsigned number

Example:-

```
        ≡
        CMP AL, BL
        JMP aa
        ≡
        ≡
        JMP exit
aa. ≡
exit. ≡
```

carry

c)

t

Conversi

To print Hexadecimal value

```
CMP dl, 9
JMP aa
```

12/10/23 Classwork (C.W) GOAL (H.W) Thursday

246B

24

42BG

Input in binary:-

```
       mov  bx, 0000
       mov  cx, 16
again: mov  Ah, 01
       int  21h
```

(1)(b)? (39)

```
sub AL,30h
OR BL,AL
SHL BX,1
DEC CX
JNZ again
```

```
  0000 0001
  0000 0000
  ---------
  0000 0001
```

```
mov Ah,01
int 21h
CMP AL,39H
JLE aa
SUB AL,37H
aa: Add AL,
```

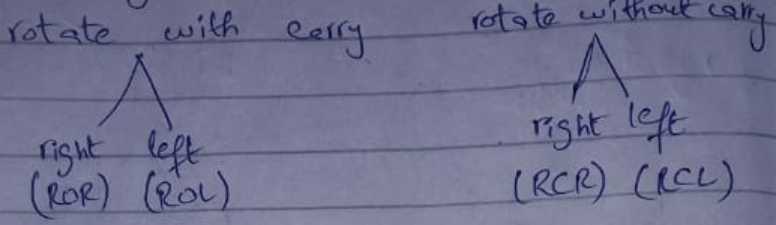
Rotate Instruction

↳ It is circular shift

↳ Two types:

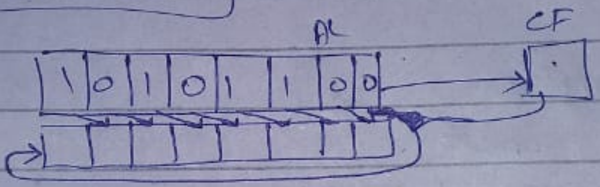
rotate with carry — right (ROR), left (ROL)

rotate without carry — right (RCR), left (RCL)



ROR Instruction:-

$\boxed{\text{ROR. AL, 1}}$



RCR Instruction

↳ Every bit moves 1 right, extreme right moves to flag. and the value that was present previously in carry flag will move to extreme left. position.

~~0110 0001~~

$\boxed{9 \Rightarrow 0110\ 0001}$ 3

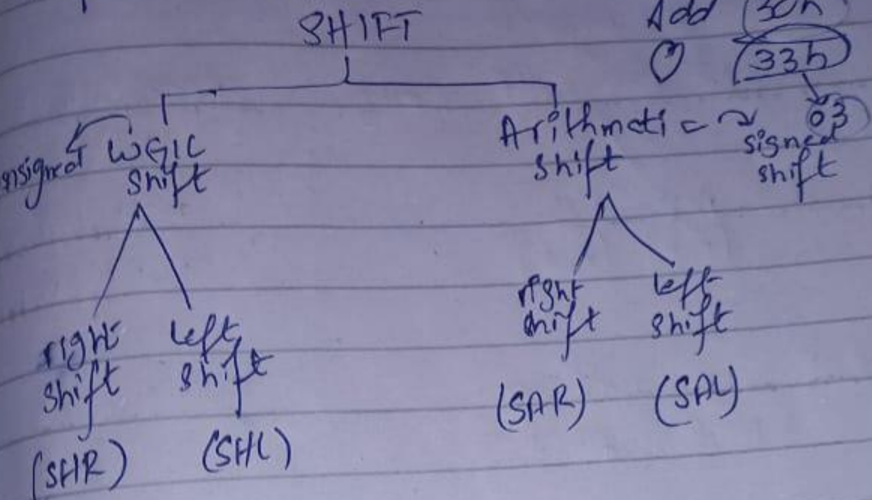
0 $\boxed{3}$h

Add 30h

0 33h

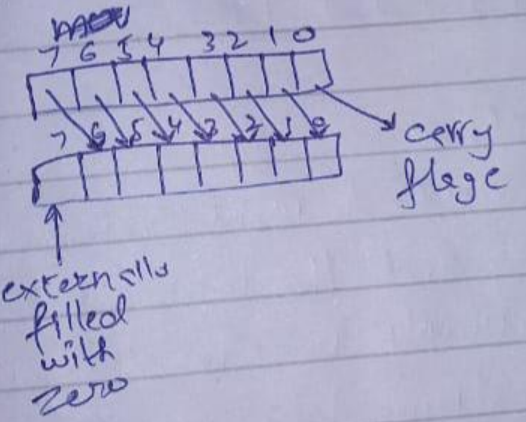
03

Shif Instructions



Examples ~~SHR~~ SHR BL,1

[6][1]



$\boxed{1011\ 0000}$

0

~~SR~~ SHR BL,1 ⇒ division by two

SHL BL,1 ⇒ multiplication by two