

Q) Shift registers are basically of four types :-

i) Serial In parallel out - This register allows serial input and produces a parallel output. The CLR signal is connected in addition to clock signal to all 4 flipflops in order to RESET them and the serial data is connected to flipflop at either end. The main application of Serial In parallel out is to convert serial data into parallel data. Therefore, it is used in communication lines where demultiplexing data lines into several parallel is required.

ii) Serial In Serial out - This register produce serial output when serial input is given. Since there is only one output the data leaves the register one bit at a time in serial pattern. The circuit contains four D flip flops which are connected in serial manner.

(iii) Parallel In Serial out - This register allows parallel input and gives serial output of each flip flop. The clock input is directly connected to all flip flops but the data is connected to each flip flop through a mux at input.

The output of previous flip flops and parallel data input are connected to input of mux and output of mux and output of mux is connected to next flip flops.

(iv) Parallel In Parallel out - This register allows parallel input & produces parallel output. The CLR signal and clock signal are connected to all flip flops. Data is given as input separately for each flip flop and in the same way output is also collected from each flip flop.

Q.2 Truth Table →

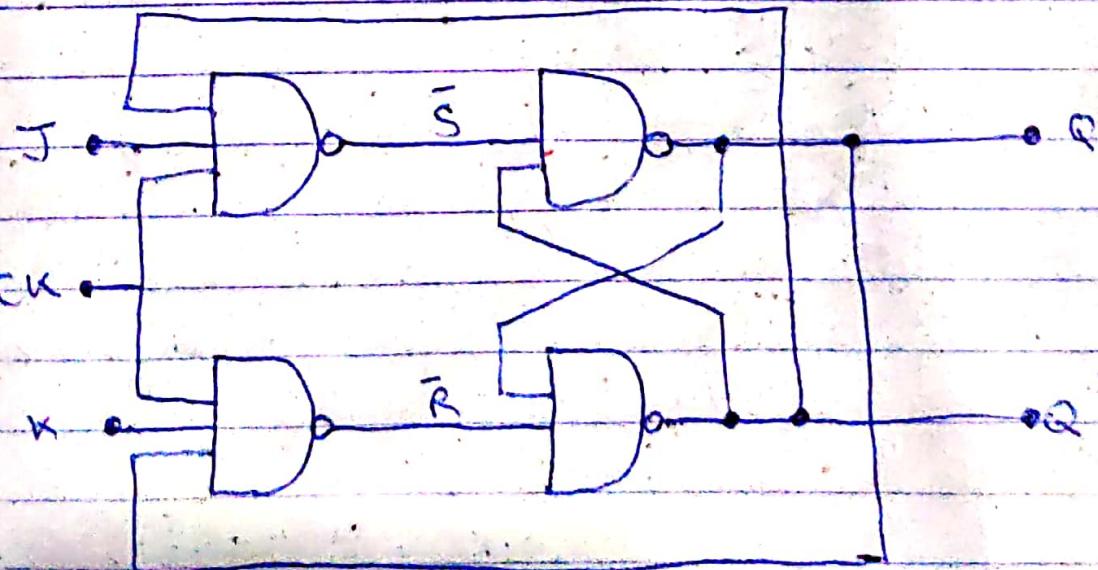
J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q'(t)

ext:

Characteristic Eqn $\rightarrow K'(t)Q(t) + J(t)Q'(t)$

Excitation table \rightarrow

<u>$Q(t)$</u>	<u>$Q(T+1)$</u>	<u>J</u>	<u>K</u>
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0



Q1 The basic NAND flip flop has many advantages and uses in sequential logic circuit but it suffers from two basic switching problems →

- (1) $S = 0$ & $R = 0$ and ($S = R = 0$) must be avoided.
- (2) If set and reset changes state while enable (E_N) input is high, the connect latching may not occur.

Q2

3×8 line decoder

This decoder circuit gives 8 logic output for 3 inputs and has a enable pin
The logical expression for outputs →

$$D_0 = \bar{A} \bar{B} \bar{C}$$

$$D_1 = A \bar{B} \bar{C}$$

$$D_2 = \bar{A} B \bar{C}$$

$$D_3 = \bar{A} B C$$

$$D_4 = A \bar{B} \bar{C}$$

$$D_5 = A B \bar{C}$$

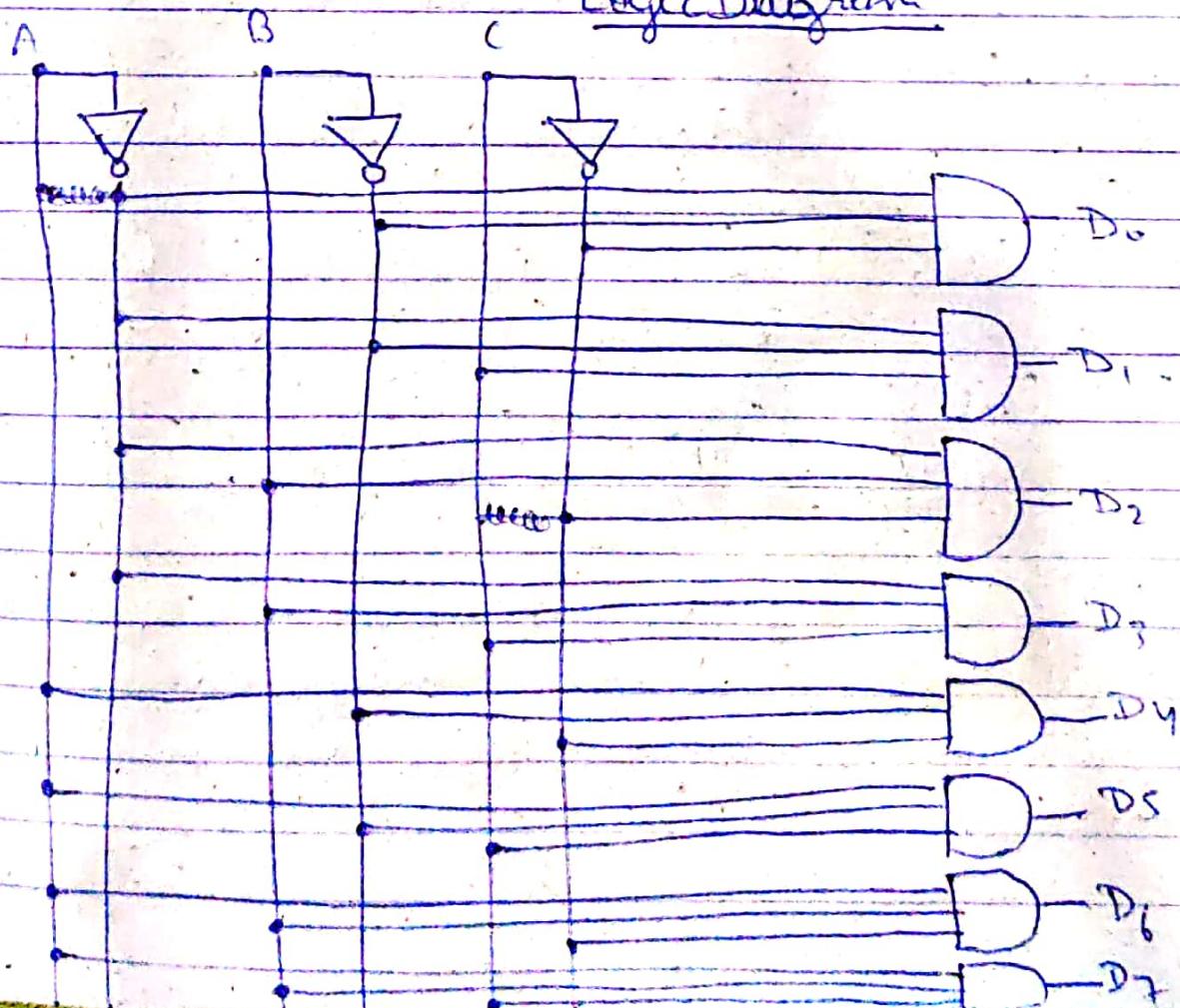
$$D_6 = A B C$$

$$D_7 = A B C$$

Truth Table

A	B	C	D ₀	D ₁	D ₂	D ₃	D ₄	D ₅	D ₆	D ₇
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Logic Diagram



Page No.	
Date	

Q.4

i) No of address lines $\rightarrow 16K = 2^4 \cdot 2^{10} = 2^{14}$

\therefore 14 address lines.

ii) No of data lines $\rightarrow 8\text{ bit} = 8$ data lines.

iii) No of Register $\rightarrow 1$ register. (Size of one reg is 1 bit)

iv) No of memory cell \rightarrow Memory size = 16K

\therefore No of memory cell = $2^4 \times 2^{10}$
 $\times 16$

v) No of pages $\rightarrow 8\text{ bit} = 1B$

\therefore No of pages = $16KB / B = 16K$ pages

Q.5

(i) Kilo Byte (kB)

The kilobyte is the smallest unit of memory measurement but greater than a byte.

A Kilobyte is 10^3 or 1000 bytes abbreviated as 'K' or 'kB'.

$kB = 1K \times 1B = 2^{10} \times 1B$ (where 2^{10} = memory cells)

$m = \text{address lines}$)

(ii) Mega Byte (MB)

One megabyte is equal to 1000 KBS

and antecedes gigabyte (GB) unit of memory measurement. A megabyte is 10^6 or 1000000 bytes and is abbreviated as "MB".

$$MB = 1M \times 1B = 2^{20} \times 1B$$

(iii) Giga Byte (GB)

One gigabyte is equal to 1000 MBs and precedes the terabyte (TB) unit of memory measurement. A gigabyte is 10^9 or 1000000000 bytes and is abbreviated as "GB".

$$GB = 1G \times 1B = 2^{30} \times 1B$$

(iv) Tera Byte (TB)

One terabyte is equal to 1000 GBs and precedes the petabyte (PB) unit of memory measurement. A terabyte is 10^{12} or 1000000000000 bytes.

Page No.	
Date	/ / / /

bytes and is abbreviated as "TB". 1 TB is technically 1 trillion bytes.

$$TB = 1T \times 1B = 2^{40} \times 1B$$

(v) Petabytes (PB)

One petabyte is equal to 1,000 TBs and precedes the exabyte unit of memory measurement. A petabyte is 10^{15} bytes & abbreviated as "PB".

$$PB = 1P \times 1B = 2^{50} \times 1B$$

Q. 6

There are 4 categories of the ROTATE instruction:

Rotate accumulator left (RLC)

Rotate accumulator left through carry (RAL)

Rotate accumulator right (RRC)

Rotate accumulator right through carry (RAR)

Among these four instructions; two are for rotating left and two are for rotating right. Following are -

i) Rotate accumulator left (RLC)

Rotate the content of accumulator
1 bit left with carry (cg)

eg \rightarrow 1 0 1 0 1 0 1 0 $c_g = 0$
after 1st RLC \rightarrow 0 1 0 1 0 1 0 1 $c_g = 1$

ii) RAC -

Rotate the content of accumulator 1 bit
left with carry (cg)

$c_g = 1$, 0 1 0 1 0 1 0 $c_g = 0$
After 1st RAC = 0 1 0 1 0 0 0 $c_g = 1$

iii) RRL -

Rotate the content of accumulator 1 bit
right without carry (cg)

eg \rightarrow 1 0 0 0 0 0 0 1 , $c_g = 0$
After 1st RRL = 1 1 0 0 0 0 0 0 , $c_g = 1$

iv) RAR - Rotate the content of accumulator 1
bit right with carry (cg)

eg \rightarrow 1 0 0 0 0 0 0 1 , $c_g = 0$

After 1st RAR = 01000000, Cg=1

Q3

a)

```
# BEGIN 0000H
LXI H, COS0
MOV A, M
INX H
ADD M
STA COS2
HLT
```

ORG COS0

DB 55H, 66H

b)

```
# BEGIN 0000H
LXI H, COS0
MOV A, M
INX H
SUB M
INX H
MOV M, A
HLT
```

ORG COS0

DB 95H, 65H

c)

#ORG 2000H

#BEGIN 2000H

LHLD 2501

XCHG

LDA 2503

LXI H, 0000

MVI C, 08

LOOP: DAD H

RAL

JNC AHEAD

DAD D

AHEAD: DCR C

JNZ LOOP

SHLD 2504

HLT

#ORG 2501H

#DB 84H, 00H, S6H

d)

#ORG 2000H

#BEGIN 2000H

LHLD 2501

LDA 2503

MOV B, A

MVI C, 08

LOOP: DAD H

MOV A, H

SUB B

JC AHEAD

MOV H, A

INR L

AHEAD: DCR C

JNZ LOOP

SHLD 2504

HLT

#ORG 2501H

#DB 9BH, 48H, 1AH.

O.B.

i) Ascending order-

MVI B, 09 : Initialize counter

START

MVI C, 09H

BACK: MOU A, M

INX

CMP M

JC SKIP

JZ SKIP

MOV D, M
 MOV M, A
 DCX H
 MOV M, D
 INX H
 SKIP: DCR C
 JNZ BACK
 DCR B
 JNZ START
 HALT

2) Descending order:-

LXI H, 5000
 MOV C, M
 DCR C
 REPEAT: MOV D, C
 LXI H, 5001
 LOOP: MOV A, M
 INX H
 CMP M
 JNC SKIP
 MOV B, M
 MOV M, A
 DCX H
 MOV M, B

INX H

SKIP: DCR D

JNZ LOOP

DCR C

JNZ REPEAT

HLT

Q. 9

STA 9000H - F(4T) + R(3T) + R(3T) + W(3T) = 13T

IN 80H - F(4T) + R(3T) + W(3T) = 10T

RST 7 - F(4T) + W(3T) + W(3T) = 10T

INR M - F(4T) + R(3T) + W(3T) = 10T

HLT - F(4) + BUS IDLE (1T) = 5T

LHLD 4050H - F(4T) + R(3T) + R(3T) + R(3T) + R(3T) = 16T

LDA 9000H - F(4T) + R(3T) + R(3T) + R(3T) = 13T

LXI 2000H - F(4T) + R(3T) + R(3T) = 10T

PUSH H - F(6T) + W(3T) + W(3T) = 15T

POP PSW - F(4T) + R(3T) + R(3T) = 10T

BAS B - F(4T) + R(3T) + R(3T) = 10T

F → Opcode fetch

R → Memory Read

W → Memory Write

T → Total T-states

(Q10)

$$FFFFH = 65535_{10}$$

$$F = 3MH_3$$

$$T_{\text{inside loop}} = 6 + 4 + 4 + 10 = 24$$

$$T_{\text{outside loop}} = 10 + 5 = 15$$

$$\text{Time delay in the loop} \Rightarrow T_1 = T \times T_{\text{delay}} \times N_{10}$$

$$= \frac{1}{3} \times 24 \times 65535$$

$$= 524280 \text{ nsec}$$

$$= 524.28 \text{ msec}$$

$$\text{Total time} \Rightarrow T_{1n} = 524.28 \text{ msec} \times (10-7) \times \frac{1}{3} \text{ msec}$$

$$T_{1n} = 524.28 \text{ msec}$$

$$\text{Total delay outside loop} \Rightarrow T_0 = 15 \times \frac{1}{3} \text{ nsec}$$

$$T_0 = 0.0050 \text{ msec}$$

$$\text{Total time delay} \Rightarrow 0.0050 + 524.28 \text{ msec}$$

$$\Rightarrow 524.285 \text{ msec}$$

$$\approx 524.3 \text{ msec}$$

Zojo No.		
Date	/	/

Hence, Total time required by
program = 7.86.4 msec