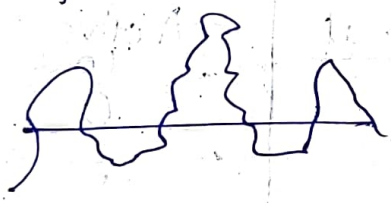
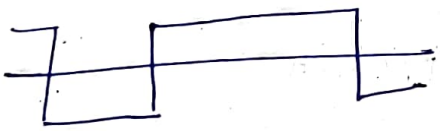
→ Digital electronics is a field of electronics involving the study of digital signals & the engineering of devices that use or produce them

Analog Signals	Digital Signals
- Continuous signals	- Discrete signals
- Sine wave	- Square wave
- Human voice, natural sound are few ex.	- Computers, optical drives & other electronic device.
- Cont. reanges of value	- Discontinuous values
- Record sound waves as they are	- Converts into binary waveform



→ Analog



→ Digital (Asynchronous)

→ Digital (synchronous)

✱ Digital Signal

Binary value represented:
(i) 0 & 1 (ii) T & F (iii) H & L (iv) ON & OFF

Represented by values or range of values of physical qty

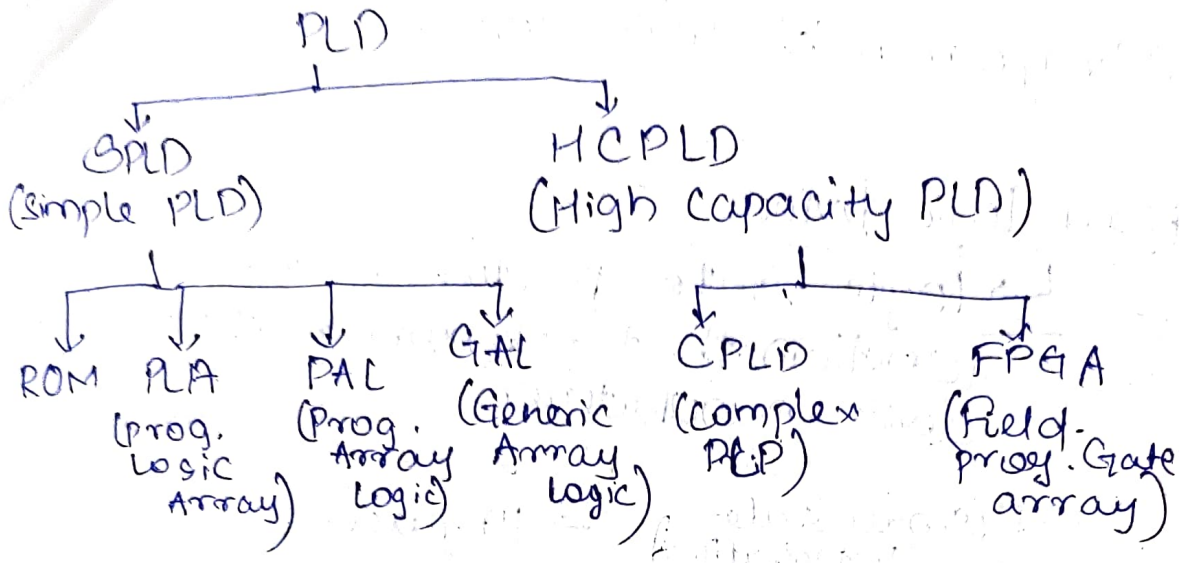
CPU: Voltage level

Disk: Magnetic field direc^{n}

CD: surface pits

Dynamic RAM: Electric Charge

Types of PLD:



PLD config:

1. EPROM
2. EEPROM
3. Static RAM
4. flash memory
5. Antifuse

* FPGA

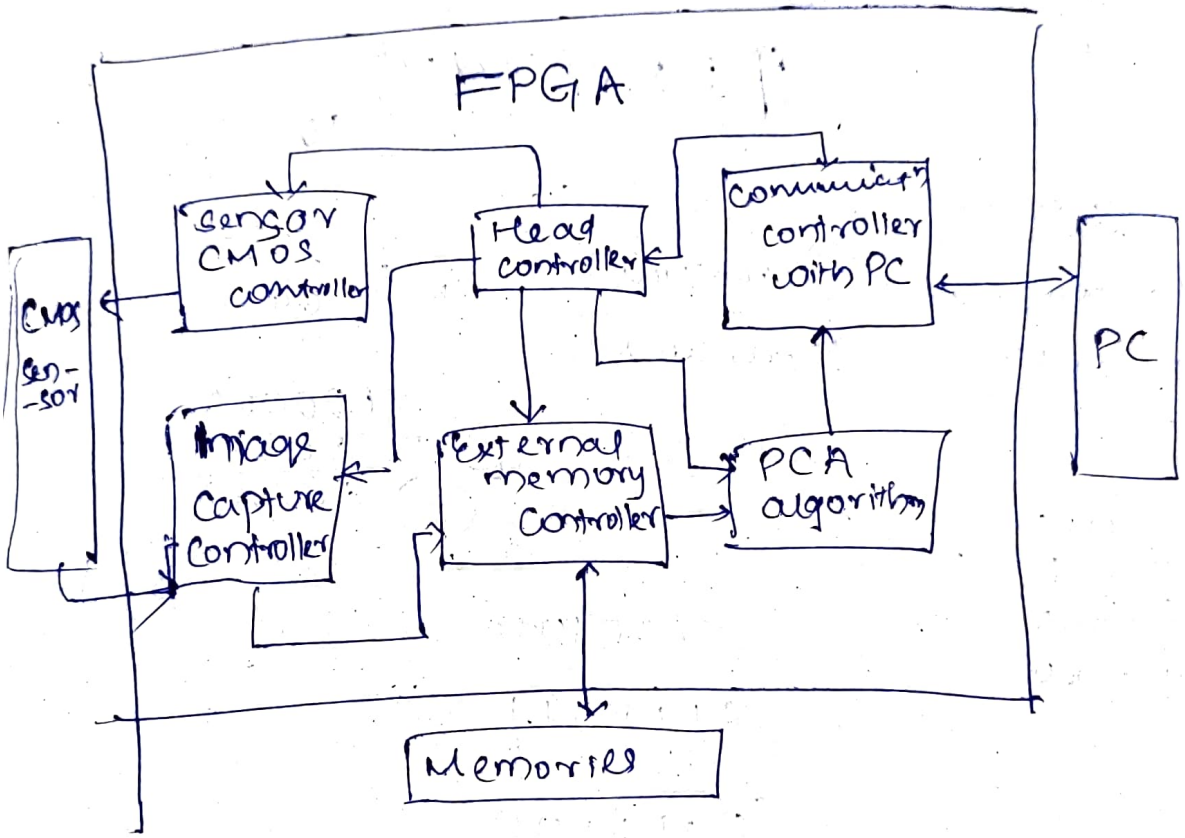
Stands for Field Programming Gate Array is a of configurable integrated ckt that can be repeatedly manufactured after manufacturing. These are subsets of logic gates. They consists of programmable logic device/blocks with a connect . FPGA's are flexible & can be reprogrammed to perform various tasks or functions.

→ They are prog. using hardware description language.

→ Used in diverse appl. including telecommunications, automative, aeroplace, etc.

Working Principle

- FPGA work based on a reconfigurable logic architechture that allows users to program the hardware functionability after manufacturing. They consists of a matrix of configurable logic block, interconnect & I/O pads



Octal → 3 bits
Hexadecimal → 4 bits

A → 10
B → 11 13
C → 12
D → 13

$(35305523624)_8$

3 5 3 0 5 5 2 3 6 2 4
011101011000101101010011100100100

EB16A794

$(EB16A794)_{16}$

Assignment

9's comple. - (5) problems
10's compl. - (5) problems

99999
30718
69282

100000
- 30718
69282

$$\begin{array}{r} 1111 \\ +\quad\ 1 \\ \hline 10000 \\ 10000 \end{array}$$

$$\begin{array}{r} 1000100 \\ +\quad\ 0111011 \\ \hline 0111100 \end{array}$$

(*) Signed Binary Number

(+) Positive : MSB = 0
(-) Negative : MSB = 1

- Signed magnitude
- Signed 1's comp.
- Signed 2's comp.

n bits

Signed magnitude & 1's : $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

2's : $-(2^{n-1})$ to $(2^{n-1}-1)$

BCD → Binary Coded Decimal

Add (6)

5 + 5

```
  0101
+ 0101
------
  1010
+ 0110
------
(1) 0000
```

```
  1010
+ 0011
------
  1101
```

```
  100
0001 0000
```

0001 0000

1 0

Greater than 9 or carry none se Add (6)

```
  7    0111
+ 6    0110
------------
       1101
     + 0110
------------
(1)    0011
```

13

0001 0011

Assignment - 03	Excess -3 code is self complement? Justify
Take your own number and go for BCD	

EBCDIC code

8-bit alphanumeric code

Excess-3 Binary + 0011

```
  1  3
+ 3  3
-------
  4  6
-------
0100 0110
```

13 → excess -3

↓

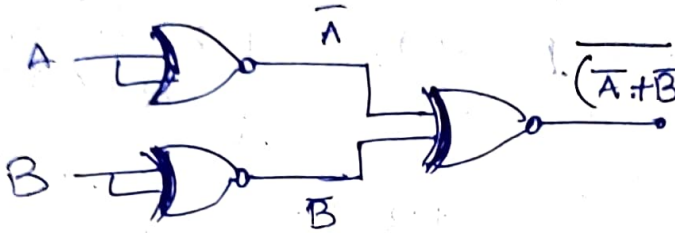
0100 0010

ASCII → American Standard Code for Information Interchange.

Short the $0 \to 0 \times 1^{\wedge}$

Bubbled NAND Gate is same as OR gate

NOR to AND gate



$\overline{(\bar{A}+\bar{B})} = \bar{\bar{A}} \cdot \bar{\bar{B}} = A \cdot B$

XOR

ODD no. of inputs are 1. then output is 1

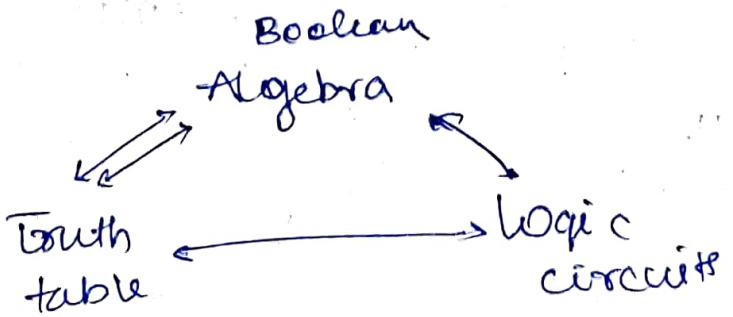
XNOR For 2 input only

If both inp. are same, the output will be 1.

↳ Equivalence gate

Assignment	Realisation of all logic gates by Proteus
Design diff. logic gates using transistor & explain	Date: 30/7/2024

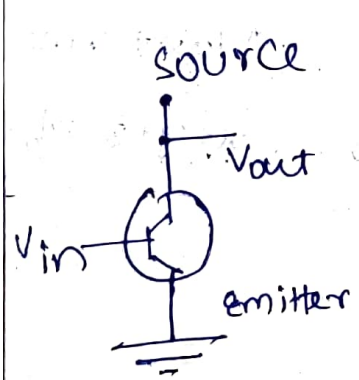
Gate Delay Propagation Delay



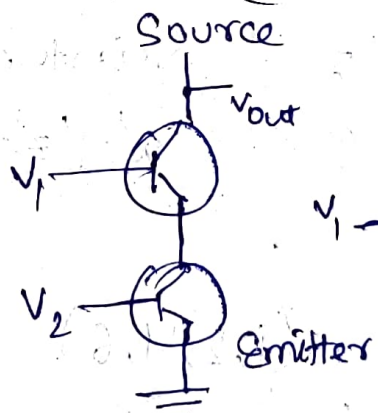
$(x+y)(x+\bar{y})(\overline{x\bar{z}})$

1 → 0
0 → 1
(+) → (·)
(·) → (+)

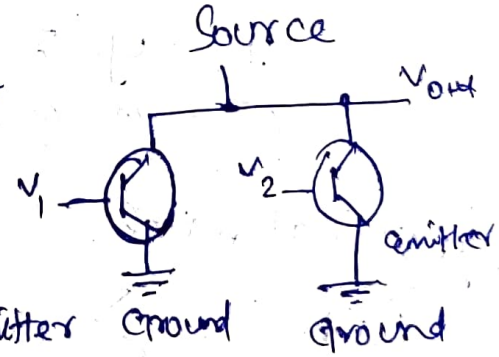
Identity Name	AND form	OR form
Identity law	$1x=x$	$0+x=x$
NULL Law	$0x=0$	$1+x=1$
Idempotent law	$xx=x$	$x+x=x$
Inverse law	$x\bar{x}=0$	$x+\bar{x}=1$
Absorption law	$x(x+y)=x$	$x+xy=x$
Demorgan	$\overline{(x\cdot y)}=\bar{x}+\bar{y}$	$\overline{(x+y)}=\bar{x}*\bar{y}$
Distributive law	$x*(yz)$ $=(x+y)(x+z)$	$x(y+z)$ $xy+xz$



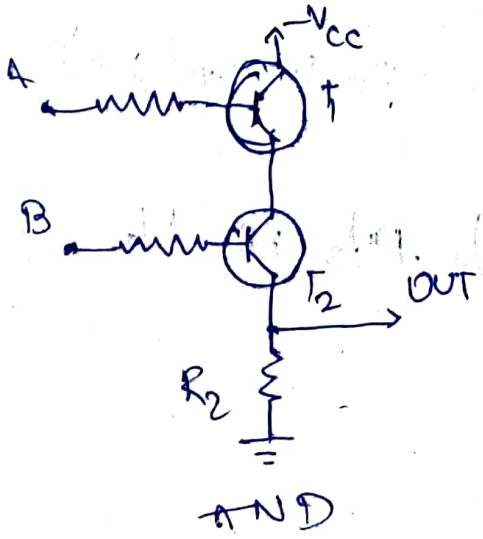
NOT



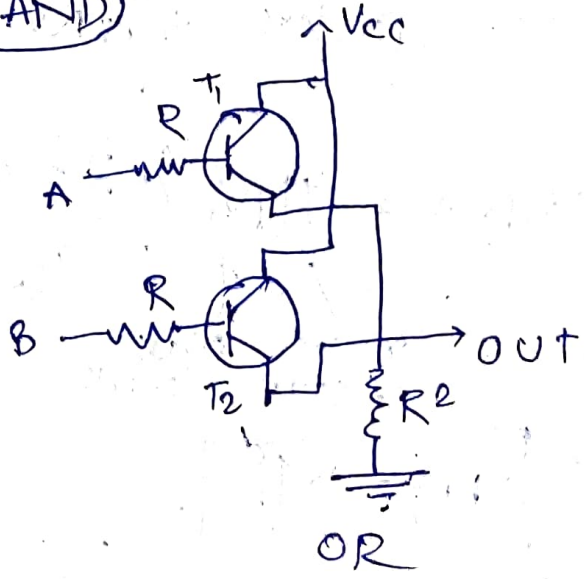
NAND



NOR



AND



OR

SOP and POS

↑ Sum of prdcts.

↑ Product of sums

$F(x,y,z)$

↳ $xy + yz + xz$

↳ $F(x,y,z) = (x+y)(y+z)(z+x)$

literal: A variable or its complement.

Prdct term: •

Sum term: +

Minterm: A prdct term in which all variables appear exactly once, either complemented or uncomplemented

$(abc) + (\bar{a}b\bar{c})$

Maxterm: A sum term

$(a+b+c) \cdot (\bar{a}+b+c)$

x	y	z	Minterm	Maxterm
0	0	0	$x'y'z' = m_0$	$x+y+z = M_0$

Canonical forms

$$f(a,b,c) = \sum m(1,2,4,6)$$

↑ Sum of prdcts ↑ Min terms ↳ $m_1\ m_2\ m_4\ m_6$

$$f(a,b,c) = \prod M(0,2,5,7)$$

↳ $M_0\ M_3\ M_5\ M_7$

↑ Prdcts of sum ↑ Max terms

$$m_j = M_j'$$

Standard form

No need that all variable appear.

SOP → $a + \bar{a}$

POS → $a\bar{a}$

$a'b'c + bc' + ac'$

$a'b'c' + (a+\bar{a})bc' + a(b+b')c'$

$a'b'c' + \underline{abc'} + a'bc' + \underline{abc'} + ab'c'$

⇒ $ab'c' + abc' + a'bc' + ab'c'$

$(a+b+c).(b'+c').(a'+c')$

$(a+b+c)((a\bar{a})b'+c').(a'+c'+b\bar{b})$

$(a+b+c)(a+b'+c')\underline{(a'+b'+c')} + (a'+c'+b) + \underline{(a'+c'+b')}$

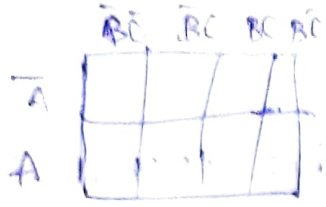
$(a+b+c)(a+b'+c')(a'+c'+b)(a'+b'+c')$

(*) Minimization Techniques

	CD'	CD	CD	CD'
A'B'	0	1	3	2
A'B	4	5	7	6
AB	12	13	15	14
AB'	8	9	11	10

x \ y	(y') 0	(y) 1
(x') 0	0	1
(x) 1	1	1

Adjacent
Octet > Quad > pairing



$\bar{x}\bar{y}\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z} + xy\bar{z}$

$\bar{y}\bar{z} + y\bar{z}$

$\bar{z}(\bar{y} + y) = \bar{z}$

~~Or~~ ~~$\bar{x}\bar{z} + x\bar{z}$~~

$\bar{x}(\bar{y}\bar{z} + \bar{y}z + y\bar{z} + yz)$

① $\bar{x}$

(+)

$x(\bar{y}\bar{z} + y\bar{z})$

$F(w,x,y,z) = \bar{w}\bar{x}\bar{y}\bar{z} + \bar{w}x\bar{y}z + \bar{w}\bar{x}y\bar{z}$
$+ \bar{w}xy\bar{z} + w\bar{x}\bar{y}\bar{z} + w\bar{x}\bar{y}z$
$+ w\bar{x}y\bar{z}$

wx \ yz	00	01	11	10
00	1	1	0	1
01	0	0	0	1
11	0	0	0	0
10	1	1	0	1

corner can be a group

$\bar{w}\bar{x}\bar{y} + y\bar{w}yz$ ~~$\bar{w}\bar{x}\bar{y} + w\bar{x}yz$~~

$x\bar{y} +$ $\bar{w}\,y\bar{z}$

Total: 3 group

Corner

$w\bar{x} + \bar{w}\bar{x} + \bar{y}\bar{z} + y\bar{z}$

$(\bar{x})$ + ~~$(\bar{z})$~~

Quad

$\bar{w}\bar{x} + w\bar{x}$

~~$\bar{x}$~~ + $\bar{y}\bar{z}$

$\bar{w}\bar{x}\,\bar{y}\bar{z} + \bar{w}\bar{x}\,yz + \bar{w}\,x\bar{y}\bar{z} + wx\bar{y}\bar{z} + w\bar{x}\,\bar{y}\bar{z} + \bar{w}\,xyz + \bar{w}\,xy\,\bar{z}$

$\bar{w}\bar{x} + \bar{w}\,x$

$\bar{w}\bar{y}\bar{z}$ $\quad \bar{w}\,(\bar{x}+x) = \bar{w}$

$\bar{y}\bar{z} + y\bar{z}$ $\quad \dfrac{\bar{w}\bar{x} + w\bar{x} + wx + \bar{w}x}{\bar{x} + x \;\; \cancel{x}}$

~~$(\bar{z})$~~ $\quad \bar{w}y\bar{z} + \bar{z}$ (1)

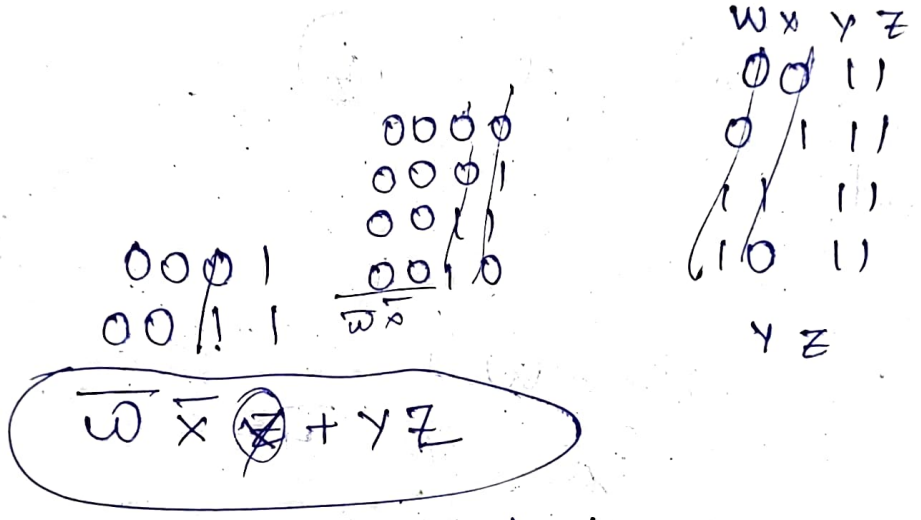
$y\bar{z} + \bar{y}\bar{z}$

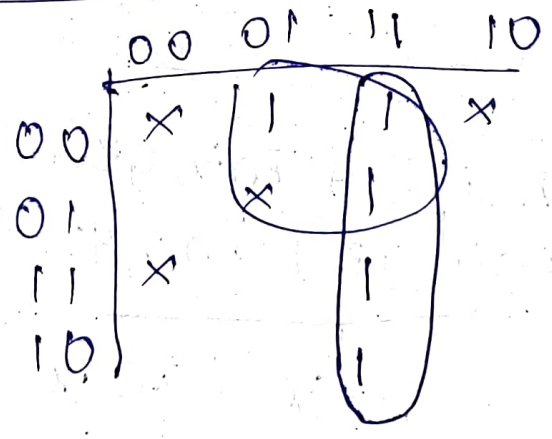
$y\bar{z} + \bar{w}yz$ $\quad \bar{w}\bar{x} + \bar{w}\,x$ (y)

$+ \bar{w}xy$ $\quad (\bar{w})$ # $\bar{w} + y$

Don't care Condition

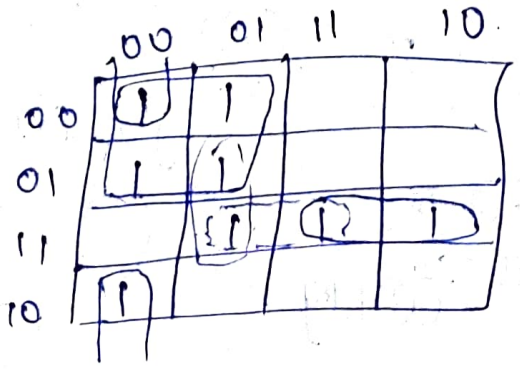
00 00 11 10



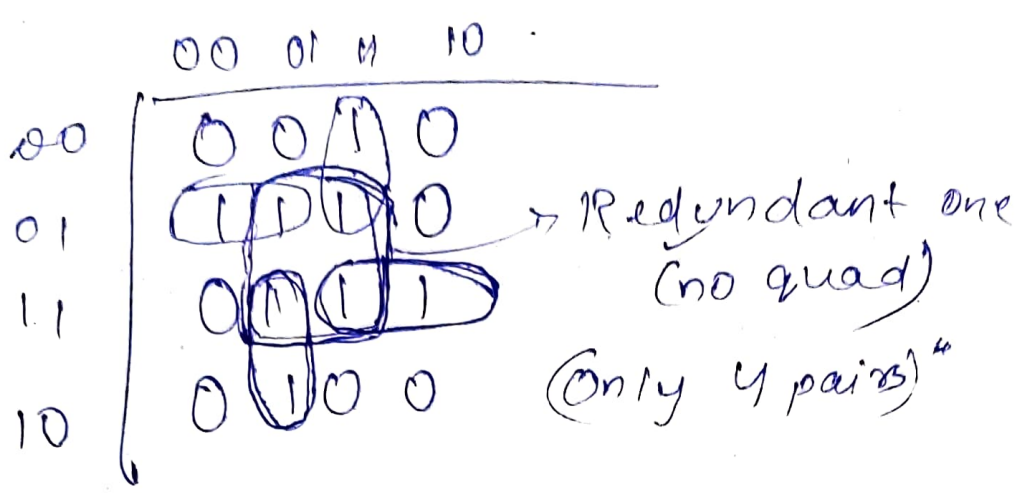


Consider as don't care cond?

Redundancy



One ungrouped can be grouped once



Q) Using Boolean Algebra, solve

(i) $x+x = x$

(ii) $x \cdot x = x$

(iii) $x + 1 = 1$

(iv) $x + xy = x$

(v) $x(x+y) = x$

(vi) $x + \bar{x}y = x+y$.

(vii) $x(\bar{x}+y) = xy$

Soln

(i) $x+x$

$\Rightarrow x(1+x)$ (Null law)

$\Rightarrow x \cdot 1$

$\Rightarrow x$

(ii) $x \cdot x = x$

Module-3

2-Bit

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sum

A \ B	$\bar{B}$	B
(0) $\bar{A}$	0	1
(1) A	1	0

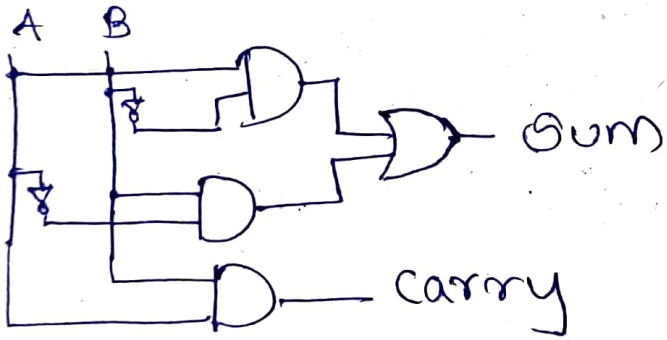
$\bar{A}B + A\bar{B}$

(XOR gate)

Carry

A \ B	$\bar{B}$	B
$\bar{A}$	0	0
A	0	1

AB



A	B	C-in	Sum	C-out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A \ BC	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	0	1	0	1
A	1	0	1	0

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$BC + AC + AB$

$\bar{A}B\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$

$\Rightarrow A \oplus B \oplus C$

Half Subtracter

2-Bit

A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

A \ B	0	1
0	0	1
1	1	0

$D = A\bar{B} + \bar{A}B$

A \ B	0	1
0	0	1
1	0	0

$B = \bar{A}B$

Full Subtracter (3-Bit)

A	B	B-in	Difference	B-out
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

A \ BC	$\bar{B}\bar{C}$ (00)	$\bar{B}C$ (01)	BC (11)	$B\bar{C}$ (10)
0	0	1	0	1
1	1	0	1	0

$D = A \oplus B \oplus C$

A \ BC	00	01	11	10
0	0	1	1	1
1	0	0	1	0

$B = BC + \bar{A}C + \bar{A}B$

Assignment

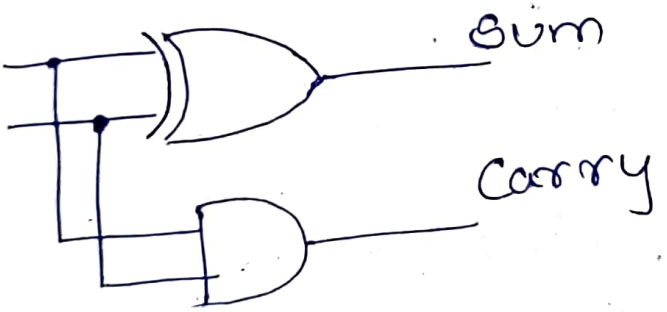
Full subtracter using Half subtracter

Adder

Half

$S = A\bar{B} + \bar{A}B$ → XOR gate

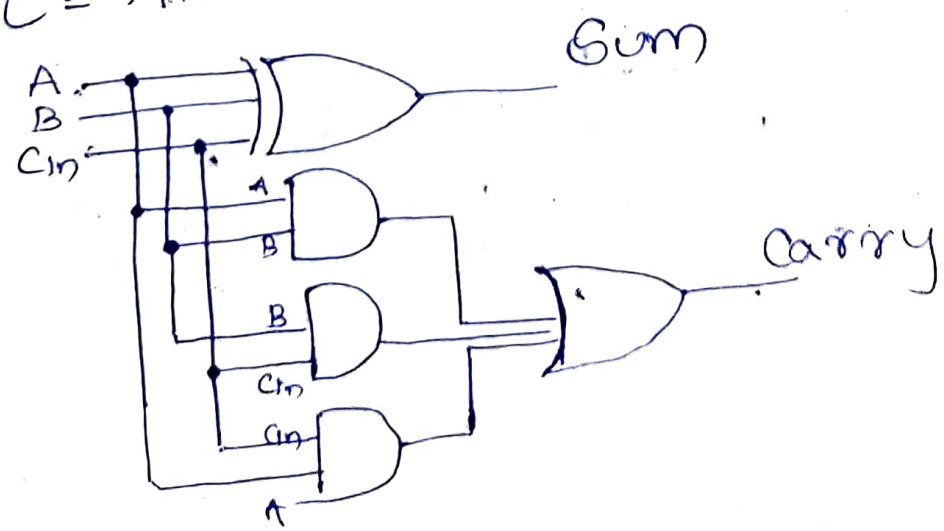
$C = AB$ → AND gate



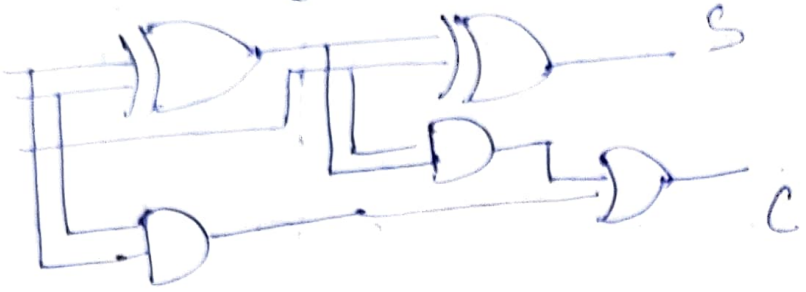
Full

$S = A \oplus B \oplus C_{in}$

$C = AB + BC_{in} + C_{in}A$



Full Adder using half adder

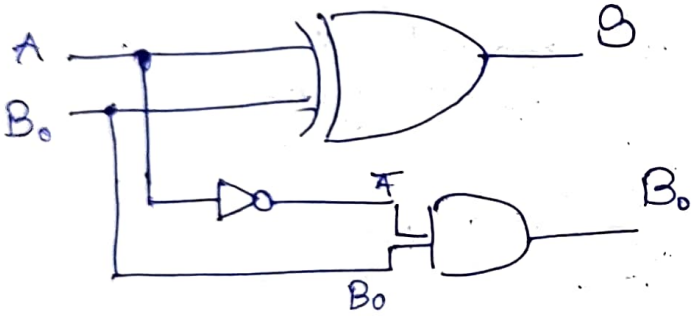


Subtractor

Half

$D = \bar{A}B + A\bar{B}$

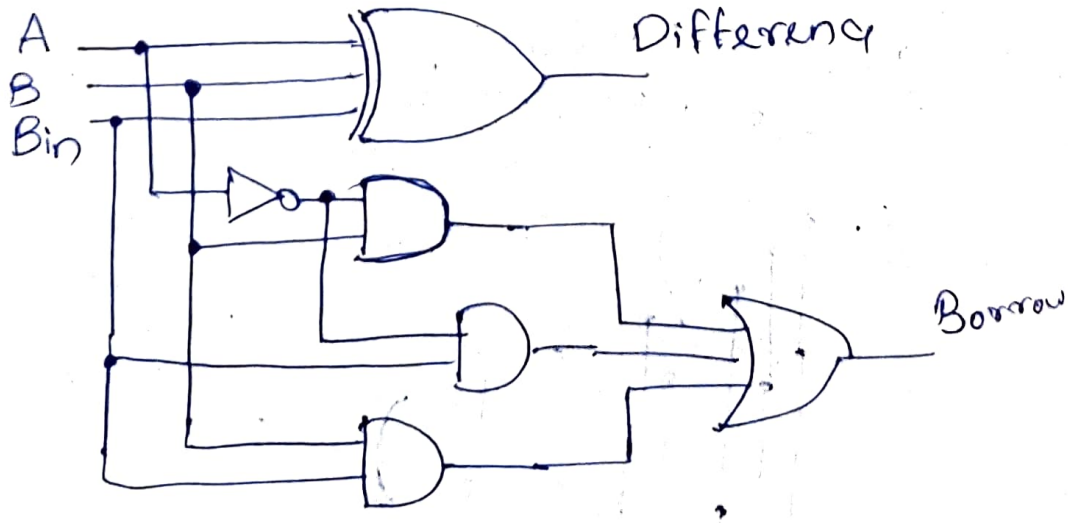
$B_o = \bar{A}B$



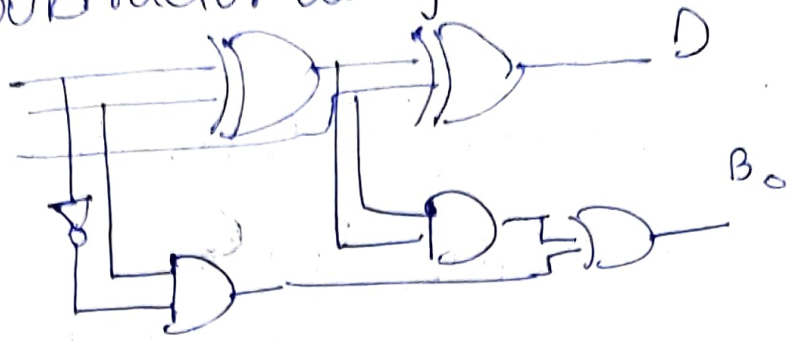
Full

$D = A \oplus B \oplus B_{in}$

$B_o = \bar{A}B + \bar{A}B_{in} + B B_{in}$

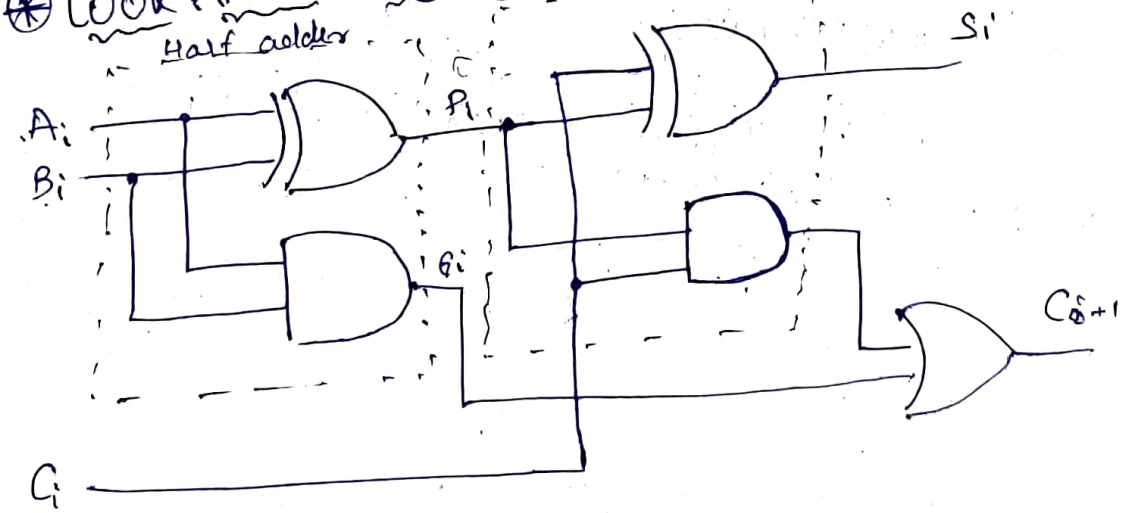


Full Subtractor using half subtractor



4-bit Adder Subtractor ckt.
4-bit Ripple Carry Adder
One's complement circuit (XOR gate)
Adder-Subtractor (4-bit)
Gate Delay & Propagation time

✱ Look-Ahead Carry Adder



$P_i = A_i \oplus B_i$

$G_i = A_i B_i$

$S_i = P_i \oplus G_i$

$C_{i+1} = G_i + (P_i \cdot C_i)$

$G_i \rightarrow$ Carry generator

$P_i \rightarrow$ Carry propagator

Carry signal will be generated if
a) Both A_i & B_i are 1
b) Either A_i & B_i are 1 but C_i must be 1

$C_{i+1} = G_i + P_i C_i$

$C_1 = G_0 + P_0 C_0$

$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0)$

$= G_1 + P_1 G_0 + P_1 P_0 C_0$

$C_3 = G_2 + P_2 C_2 = G_2 + P_2 (G_1 + P_1 G_0 + P_1 P_0 C_0)$

$C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0$

There are 4 gate delay

- C_2, C_3, C_4 doesnot depend on its previous carry-in
- C_4 doesnot ~~depend~~ need to wait for C_3 to propagate
- As soon as C_0 is computed, C_4 can reach steady state.

```
 1 1
 01010
+00110
-------
 10000
```

```
 0 1 1 0 1
+0 0 1 1 0
---------
 1 0 0 1 1
```

```
  1 1
 01110
+00110
-------
 10100
```

```
  1 1
 01011
+00110
-------
 10001
```

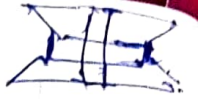
```
  1 1 1
 10010
+00110
-------
 11000
```

```
  1 1 1
 01111
+00110
-------
 10101
```

Combinational Circuits

BCD

Decimal	Binary Sum K Z_8 Z_4 Z_2 Z_1	BCD Sum C S_8 S_4 S_2 S_1	
0	0 0 0 0 0	0 0 0 0 0	
1	0 0 0 0 1	0 0 0 0 1	
2	0 0 0 1 0	0 0 0 1 0	
3	0 0 0 1 1	0 0 0 1 1	
4	0 0 1 0 0	0 0 1 0 0	
5	0 0 1 0 1	0 0 1 0 1	
6	0 0 1 1 0	0 0 1 1 0	
7	0 0 1 1 1	0 0 1 1 1	
8	0 1 0 0 0	0 1 0 0 0	
9	0 1 0 0 1	0 1 0 0 1	
10	0 1 0 1 0	1 0 0 0 0	Add +6
11	0 1 0 1 1	1 0 0 0 1	
12	0 1 1 0 0	1 0 0 1 0	
13	0 1 1 0 1	1 0 0 1 1	
14	0 1 1 1 0	1 0 1 0 0	
15	0 1 1 1 1	1 0 1 0 1	
16	1 0 0 0 0	1 0 1 1 0	
17	1 0 0 0 1	1 0 1 1 1	
18	1 0 0 1 0	1 1 0 0 0	
19	1 0 0 1 1	1 1 0 0 1	



$$C = K + Z_8 Z_4 + Z_8 Z_2$$

Cascading of BCD Adder

87 − 39

↳ 9's complement

87	1000	0111
+ 60	0110	0000
	1110	0111
	+ 0110	
	(1) 0100	0111
		1
	0100	1000
	4	8

BCD to Excess-3 code converter

BCD + 3 = Excess-3

13 → excess-3 ⇒ 46

1	3
+ 3	+ 3
4	6

Decimal	BCD	Excess-3
0	0000	
1	0001	
2	0010	
3	0011	
4	0100	
5	0101	
6	0110	
7	0111	
8	1000	
9	1001	

Decimal	BCD				Excess-3			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0
10					x	x	x	x
11					x	x	x	x
12					x	x	x	x
13					x	x	x	x
14					x	x	x	x
15					x	x	x	x

BCD after addition exists

wx \ yz	00	01	11	10
00		1	1	
01	1		1	1
11	1			
10	1	1	1	1

K-Map

W

AB \ CD	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	x	x	x	x
10	1	1	x	x

$W = A + BC + BD$

X

AB \ CD	00	01	11	10
00	0	1	1	1
01	1	0	0	0
11	x	x	x	x
10	0	1	x	x

$X = \bar{B}C + \bar{B}D + B\bar{C}\bar{D}$

Y

AB \ CD	00	01	11	10
00	1	0	1	0
01	1	0	1	0
11	x	x	x	x
10	1	0	x	x

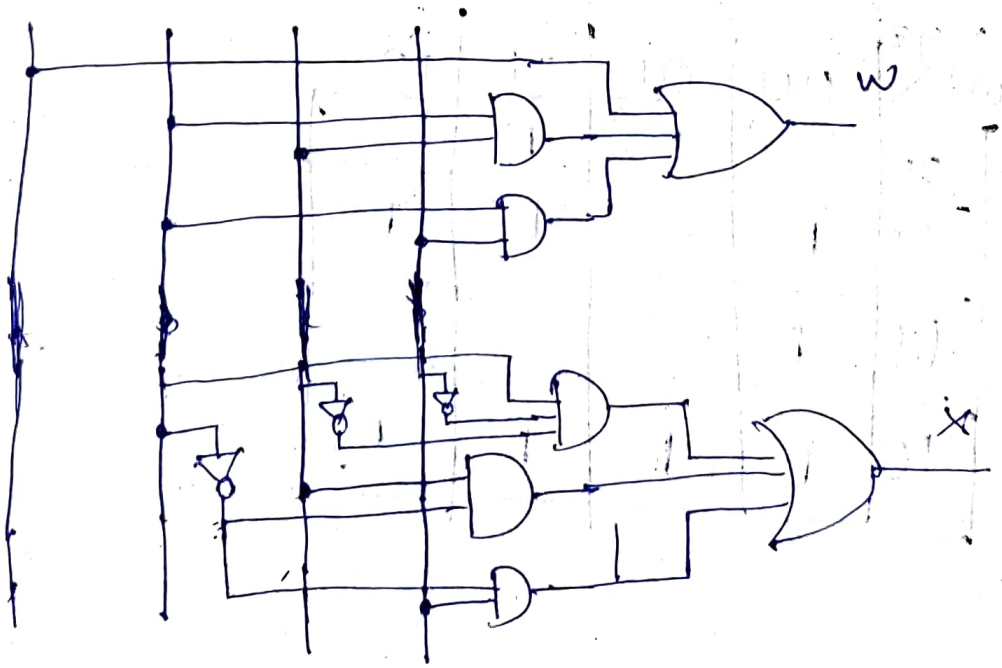
$Y = \bar{C}\bar{D} + CD$

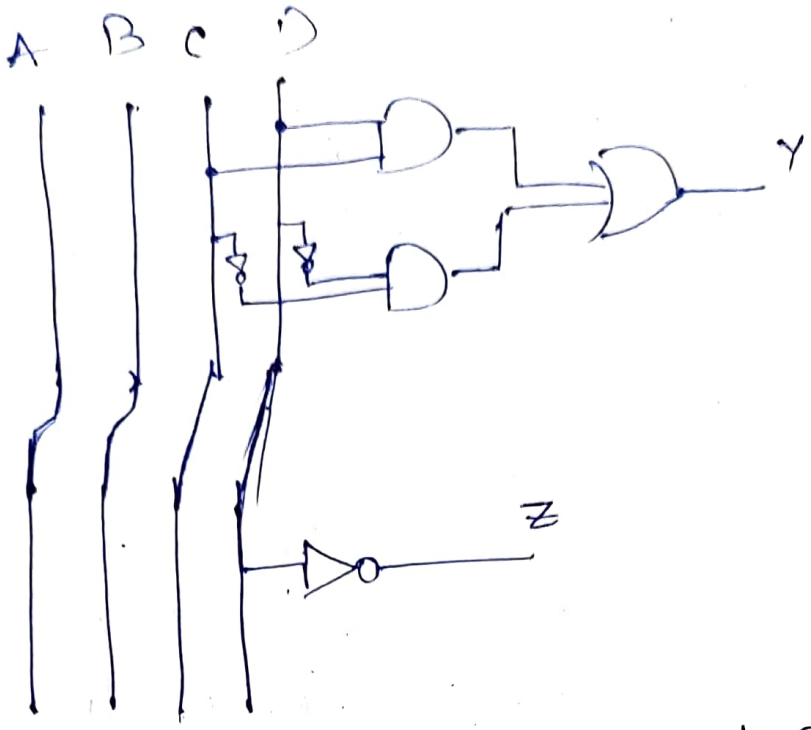
Z

AB \ CD	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	x	x	x	x
10	1	0	x	x

$Z = \bar{D}$

A B C D





Excess-3 to BCD converter

Decimal	Excess-3 A	B	C	D	BCD W	X	Y	Z
0	0	0	0	0	x	x	x	x
1	0	0	0	1	x	x	x	x
2	0	0	1	0	x	x	x	x
3								
4								
5								
6								
7								
8								
9								
10								
11								
12					x	x	x	x
13					x	x	x	x
14					x	x	x	x
15								

① Binary to Gray code

$b_3\ b_2\ b_1\ b_0$ $\quad$ $g_3\ g_2\ g_1\ g_0$

② Gray to Binary

$g_3\ g_2\ g_1\ g_0$ $\quad$ $b_3\ b_2\ b_1\ b_0$

	b_3	b_2	b_1	b_0	g_3	g_2	g_1	g_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

$g_3 = b_3$

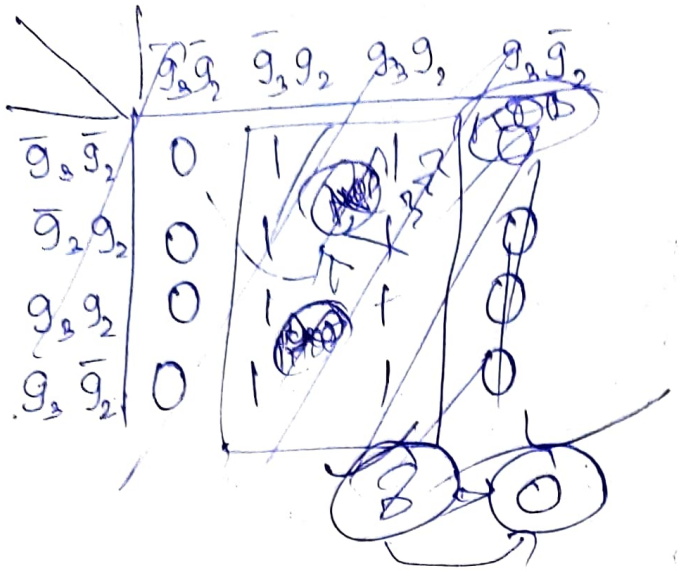
$g_2 = b_3 \oplus b_2$

$g_1 = b_2 \oplus b_1$

$g_0 = b_1 \oplus b_0$

	g_3	g_2	g_1	g_0	b_3	b_2	b_1	b_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	1	0	0	1	0
3	0	0	1	0	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	0	1	0	1
6	0	1	1	1	0	1	1	0
7	0	1	1	0	0	1	1	1
8	1	0	0	0	0	1	0	0
9	1	0	0	1	1	1	0	1
10	1	1	0	1	1	1	1	0
11	1	1	1	0	1	1	1	1
12	1	0	1	0	1	1	0	0
13	1	0	1	1	1	1	0	1
14	1	0	0	1	1	1	1	0
15	1	0	0	0	1	1	1	1

$$b_3 = g_3$$
$$b_2 = g_3 \oplus g_2$$
$$b_1 = g_3 \oplus g_2 \oplus g_1$$
$$b_0 = g_3 \oplus g_2 \oplus g_1 \oplus g_0$$



b_0 =

	$\bar{q}_1\bar{q}_0$	$\bar{q}_1 q_0$	$q_1 q_0$	$q_1\bar{q}_0$
$\bar{q}_3\bar{q}_2$	0	1	0	1
$\bar{q}_3 q_2$	1	0	1	0
$q_3 q_2$	0	1	0	1
$q_3\bar{q}_2$	1	0	1	0

$$\bar{q}_3\bar{q}_2\bar{q}_1 q_0 + \bar{q}_3\bar{q}_2 q_1\bar{q}_0 + \bar{q}_3 q_2\bar{q}_1\bar{q}_0 + \bar{q}_3 q_2 q_1 q_0 + q_3 q_2\bar{q}_1 q_0 + q_3 q_2 q_1\bar{q}_0 + q_3\bar{q}_2\bar{q}_1\bar{q}_0 + q_3\bar{q}_2 q_1 q_0$$

$$\Rightarrow q_3 \oplus q_2 \oplus q_1 \oplus q_0$$

Excess-3 to Gray code

Gray to Excess-3

	Gray				Excess 3			
	g_3	g_2	g_1	g_0	A	B	C	D
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	1	0	0	1	1	1
5	0	1	1	1	1	0	0	0
6	0	1	0	0	1	0	0	1
7	0	1	0	1				
8	1	1	0	0				
9	1	1	0	1				
10	1	1	1	0				
11	1	1	1	1				
12	1	0	1	0				
13	1	0	1	1				
14	1	0	0	0				
15	1	0	0	1				

upto 9

Not mox

Binary to BCD
BCD to Binary
} H.W.

• Decimal

Comparator

Compares two digital or Binary number

1-Bit

A	B	A>B	A=B	A<B
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

A \ B	0	1
0	0	0
1	1	0

A>B : $A\bar{B}$

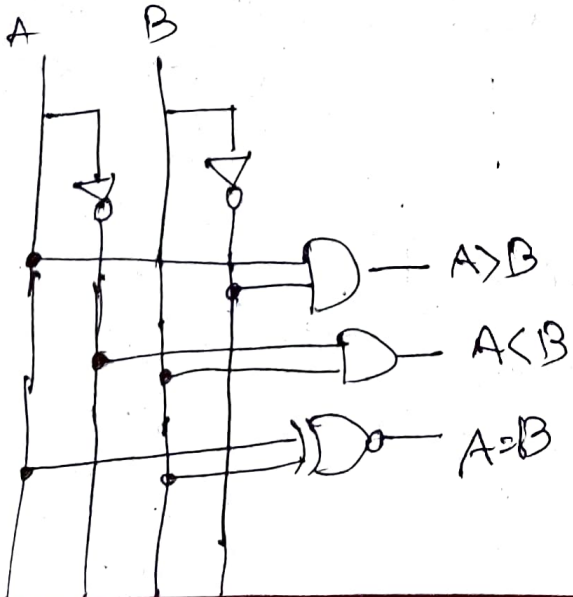
A \ B	0	1
0	0	1
1	0	0

A<B : $\bar{A}B$

A \ B	0	1
0	1	0
1	0	1

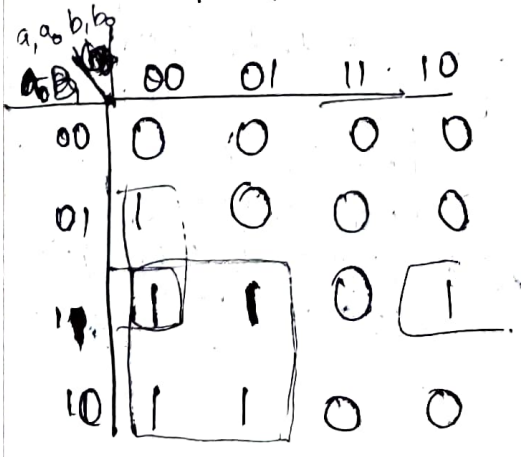
A=B :

$\bar{A}\bar{B} + AB$



2-bit

a_1	a_0	b_1	b_0	A>B	A<B	A=B
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1



$A > B$

$A\bar{B} + B\bar{C}\bar{D} + A\bar{C}\bar{D}$

$\hookrightarrow a_1\bar{b}_1 + a_0\bar{b}_1\bar{b}_0 + a_1 a_0 \bar{b}_0$

q_1q_0 \ b_1b_0 (AB \ CD)	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	0	0

$A < B$

$\bar{A}C + \bar{A}\bar{B}D + CD\bar{B}$

$\bar{a}_1 b_1 + \bar{a}_1 \bar{a}_0 b_0 + b_1 b_0 \bar{a}_0$

q_1q_0 \ b_1b_0 (AB \ CD)	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

$A = B$

$\bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}D + AB\bar{C}D + A\bar{B}C\bar{D}$

$\bar{a}_1\bar{a}_0\bar{b}_1\bar{b}_0 + \bar{a}_1 a_0 \bar{b}_1 b_0 + a_1 a_0 b_1 b_0 + a_1\bar{a}_0 b_1 \bar{b}_0$

4 Bit Magnitude Comparator

$2^4 = 16$

$x_i = A_iB_i + \bar{A}_i\bar{B}_i \quad i = 0, 1, 2, 3 \ldots$

So $(A = B) = x_2 \cdot x_1 \cdot x_0$

(*) $A > B$ | $A_3\ A_2\ A_1\ A_0$ / $B_3\ B_2\ B_1\ B_0$

① $A_3 = 1 \quad B_3 = 0$

② $A_3 = B_3$ & $A_2 = 1 \quad B_2 = 0$

$A_3\bar{B}_3 + x_3 A_2 \bar{B}_2 + x_3 x_2 A_1 \bar{B}_1 + x_3 x_2 x_1 A_0 \bar{B}_0$

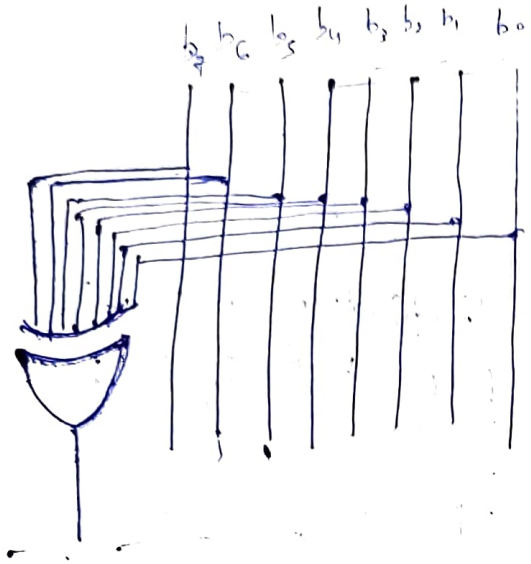
(*) For $A < B$ | $A_3\ A_2\ A_1\ A_0$ / $B_3\ B_2\ B_1\ B_0$

$\bar{A}_3 B_3 + x_3 \bar{A}_2 B_2 + x_3 x_2 \bar{A}_1 B_1 + x_3 x_2 x_1 \bar{A}_0 B_0$

$A > B$	$A < B$
1000	0000
1100	1000
1110	1100
1111	1110
	1111

Parity Code??

Parity generator and checker



→ Limited to 1 bit

Hamming code is used to detect the error.

⊛ Hamming Code

→ Multiple parity bits

$2^k \geq n+k+1$

k = no. of parity bits
n = no. of bits

let $n=4$

$2^k \geq 4+k+1 \Rightarrow 2^k \geq 5+k$

So (k=3)

Thus there will be 4 bits in binary code & 3 parity bits.

d7 d6 d5 p4 d3 p2 p1 ← Hamming Code

$p1 = d3 \oplus d5 \oplus d7$
$p2 = d3 \oplus d6 \oplus d7$
$p3 = d5 \oplus d6 \oplus d7$

Q. 1011

d7 d6 d5 p4 d3 p2 p1
1 0 1 ? 1 ? ?

$p1 = d3 \oplus d5 \oplus d7 = 1$
$p2 = d3 \oplus d6 \oplus d7 = 0$
$p4 = d5 \oplus d6 \oplus d7 = 0$

1010101

$x = d3 \oplus d5 \oplus d7 \oplus p1$

$y = d3 \oplus d6 \oplus d7 \oplus p2$

$z = d5 \oplus d6 \oplus d7 \oplus p4$

Recieved data: 1010001

d7	d6	d5	p4	d3	p2	p1
1	0	1	0	0	0	1

$x = 0 \oplus 1 \oplus 1 = 1$

$y = 1$

$z = 0$

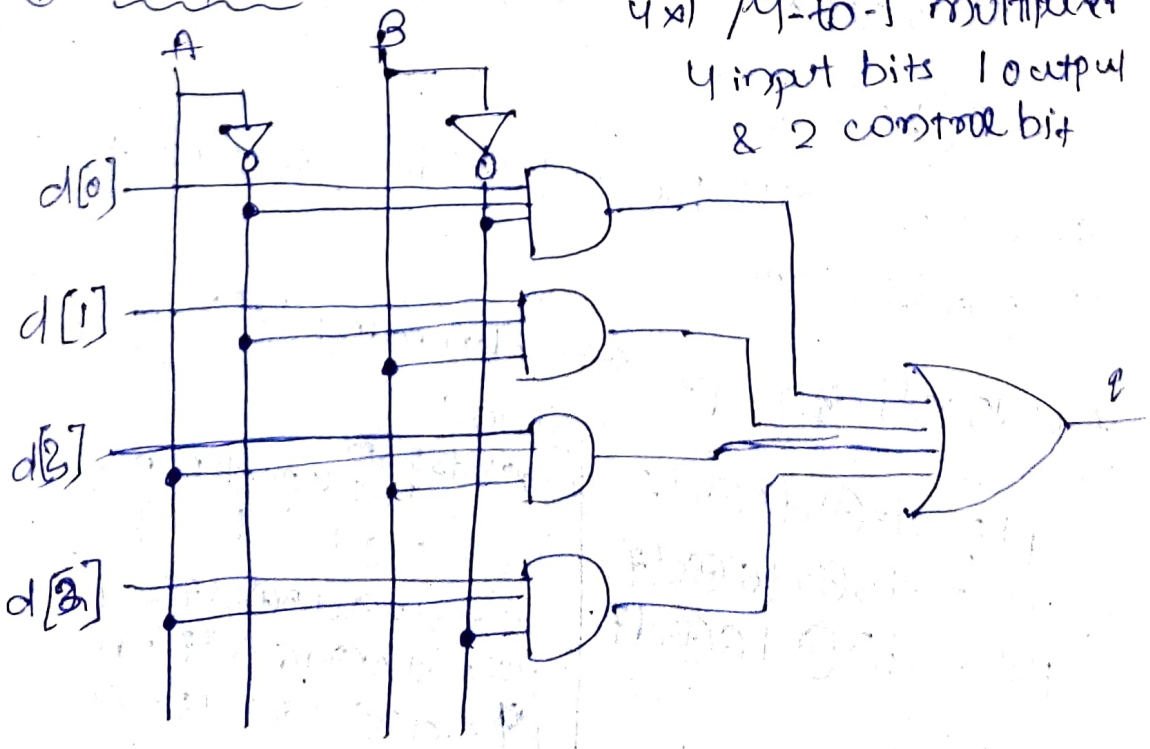
$zyx = 011$ Equal to 3

↳ 3rd bit is erroneous that is d3

Correct code: 1010101

Multiplexer (MUX)

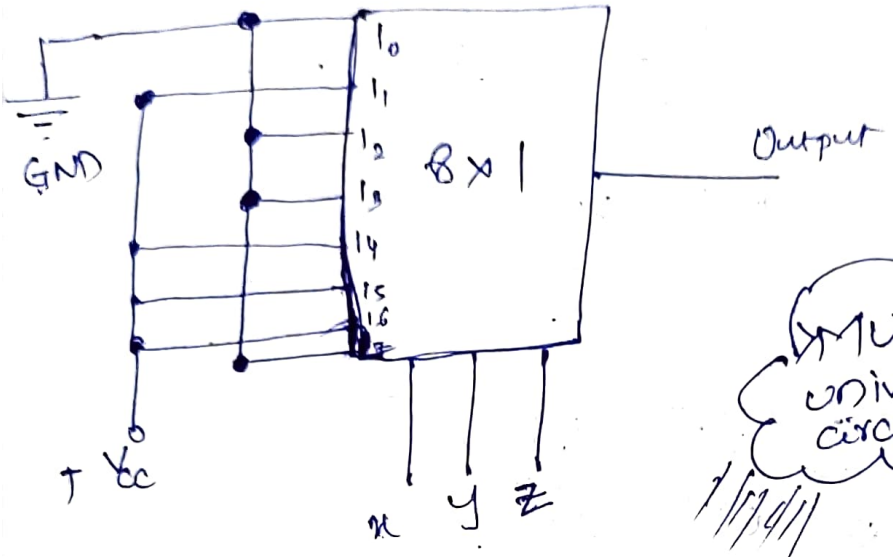
4×1 / 4-to-1 multiplexer
4 input bits 1 output
& 2 control bit



V_{cc} for 1

GND for 0.

Q) $\Sigma(1,4,5,6) = F(x,y,z)$

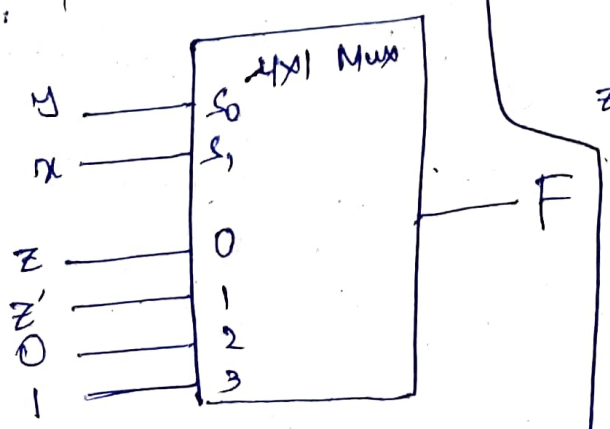


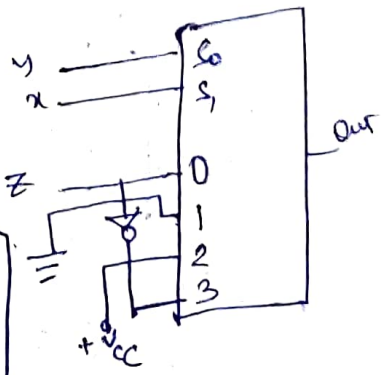
Multiplexer is universal logic circuit

Q) $\Sigma(1,2,6,7)$

x	y	z	F	
0	0	0	0	F = z
0	0	1	1	
0	1	0	1	F = z'
0	1	1	0	
1	0	0	0	F = 0
1	0	1	0	
1	1	0	1	F = 1
1	1	1	1	

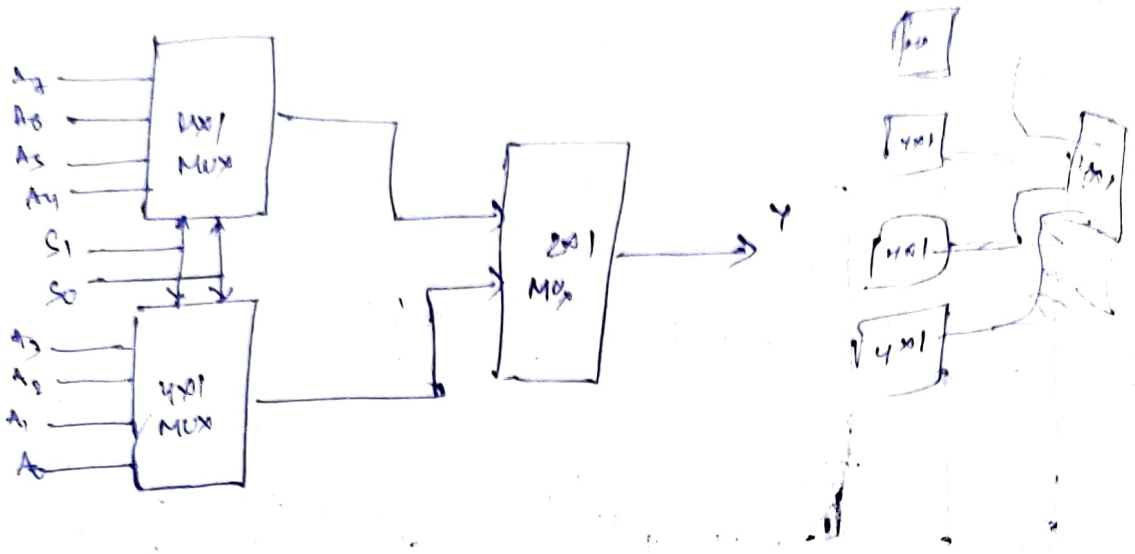
x	y	z	F	
0	0	0	0	F = z
0	0	1	1	
0	1	0	0	F = 0
0	1	1	0	
1	0	0	1	F = 1
1	0	1	1	
1	1	0	1	F = z'
1	1	1	0	





8×1

16 to 1

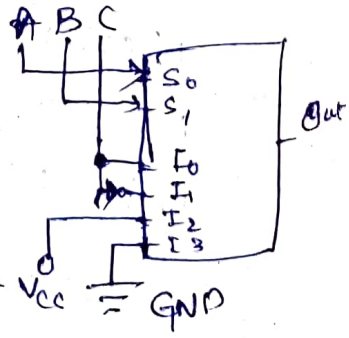


$F(A, B, C) = \Sigma 1, 2, 4, 5$

A	B	C	F	
0	0	0	0	F = C
0	0	1	1	
0	1	0	1	F = C'
0	1	1	0	
1	0	0	1	F = 1
1	0	1	1	
1	1	0	0	F = 0
1	1	1	0	

	I_0	I_1	I_2	I_3
C'	0	(2)	(4)	6
C	(1)	3	(5)	7

C C' 1 0



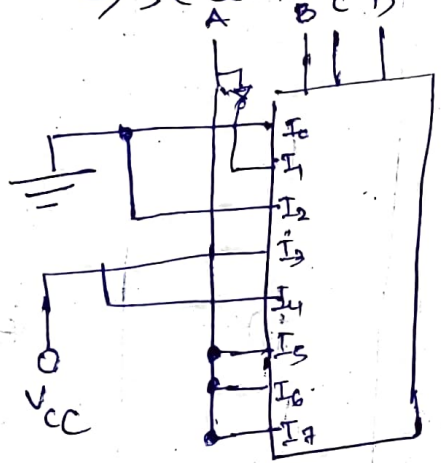
$F(A,B,C,D) = \Sigma(1, 3, 4, 11, 12, 13, 14, 15)$

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	0	(1)	2	(3)	(4)	5	6	7
A	8	9	10	(11)	(12)	(13)	(14)	(15)
	0	A'	0	1	1	A	A	A

(A) B C D

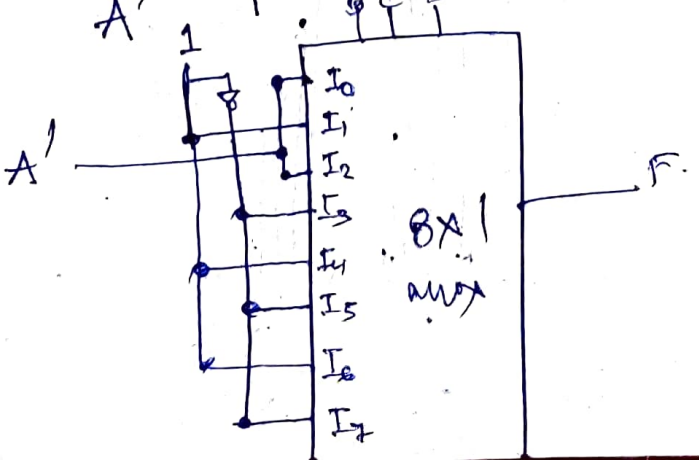
↳ selection line

outside

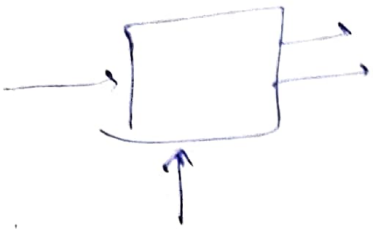


$F(A,B,C,D) = \Sigma(0, 1, 2, 4, 6, 9, 12, 14)$

	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
A'	(0)	(1)	(2)	3	(4)	5	(6)	7
A	8	(9)	10	11	(12)	13	(14)	15
	A'	1	A'	0	1	0	1	0



DeMultiplexer

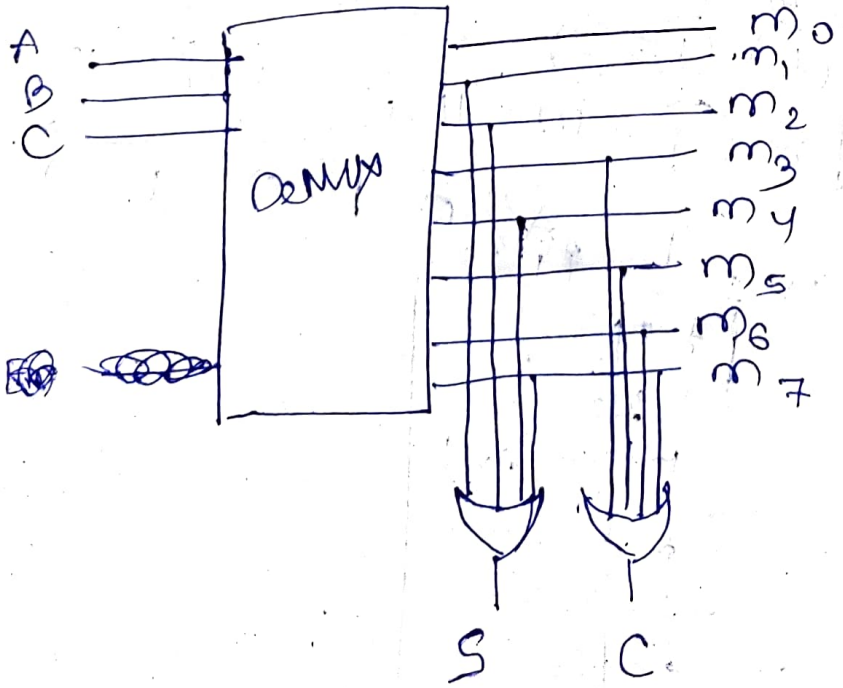


Decoder

Whose purpose is to decode the information

Design a full adder using decoder.

A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Encoder

Performs the reverse operation of decoder

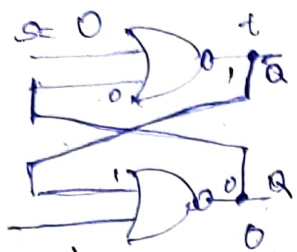
1) Priority Encoder

Higher value gets the priority upto 8 inputs

4 to 2 priority encoder:

Module-04

SR latch

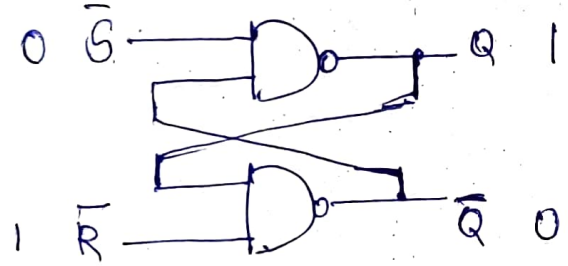


R	S	Q	$\bar{Q}$	Comment
0	0	?	?	Stored state unknown
0	1	1	0	Set Q to 1
0	0	1	0	Now Q remembers 1
1	0	0	1	"Reset" Q to 0
0	0	0	1	Now Q remembers 0
1	1	0	0	Both go low
0	0	?	?	One table

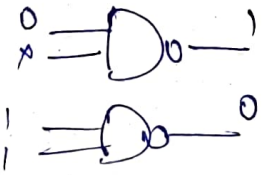
Forbidden state

R	S	Q
0	0	Prev state
0	1	set Q to 1
1	0	set Q to 0
1	1	Forbidden state

SR Latch (NAND version)



$\bar{S}$	$\bar{R}$	Q	$\bar{Q}$
0	0	1	1 Forbidden
0	1	1	0 Set
1	0	0	1 Reset
1	1	1	0 Last state
		0	1 Last state

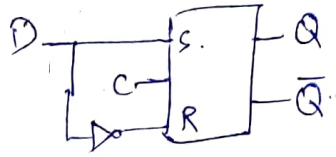


R	S	Q
0	0	Last state
0	1	Q=0 Reset
1	0	Q=1 Set state
1	1	Forbidden state

$Q(t+1) = S + \bar{R}Q$

$SR = 0$

D-Flipflop



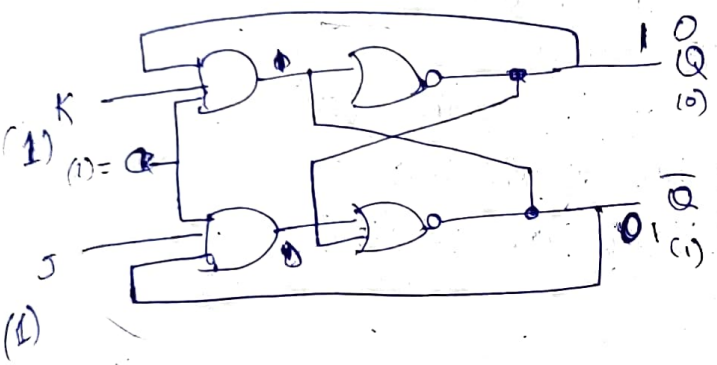
C	D	Next state of Q
0	x	No change
1	0	Q=0 Reset
1	1	Q=1 Set

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

$Q_{(n+1)} = D$

JK Flip flop

J	K	
0	0	No change
0	1	Reset $Q = 0$
1	0	Set $Q = 1$
1	1	Toggles



$Q = 0$

$\bar{Q} = 1$

Master slave flip flop: Toggling race around.

Q	J	K	Q (t+1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$Q_{(t+1)} = J\bar{Q} + \bar{K}Q$

T-Flip flop J & K are shorted.

$Q_{(t+1)} = T\bar{Q} + \bar{T}Q = T \oplus Q$

Assignment:

① Count the sequence from 0 to 7 using SR Flip Flop, JK Flip Flop, D-flip flop

(2) n-mod (6) n-mod (3) n-mod

Sequential ckt: Clock must.

SIS

* Sequential Logic Circuits

- The output state of a "sequential logic circuit" depends on present input as well as past output
  - → It is a bi-stable device
  - → Outputs from the system are "fed back" as new inputs.

⊙ Clock

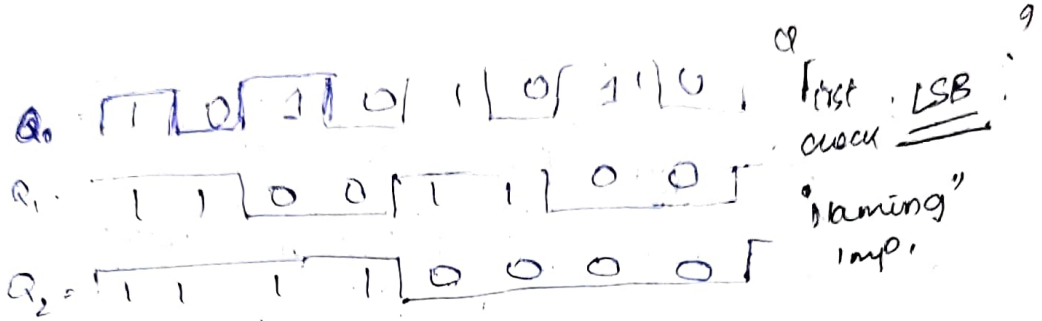
↳ Asynchronous: Circuit change thier state and output values whenever a change in input value occurs.
  ↳ clockless.

↳ Synchronous: Change their value at fixed points of time
  ↳ Clock

Lec 05

Counter (Module 5)

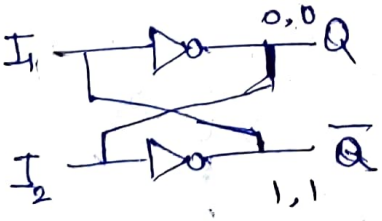
Synchronous Mod 12 counter using JK flipflop, SR flipflop, D flipflop.



Module-4

① Latches

- Building blocks of seq. circuits.
- → Built from logic gates
- → Without clock



- → level triggered

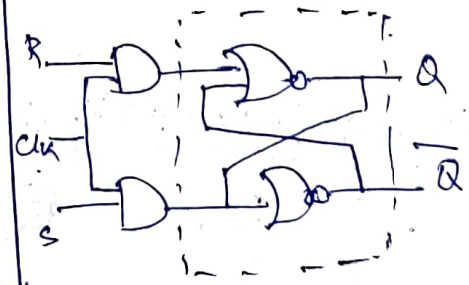
[Set stage : 1
Reset stage : 0]

Latch is sensitive only when C = 1

Flip flop

- → Built from latches
- → with clock.

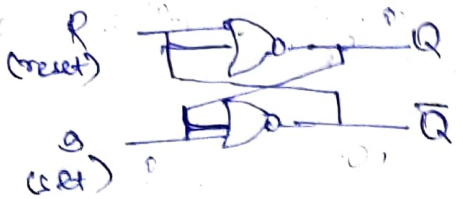
(JK, SR, D, T)



- → Edge triggered

SR latch (NOR gates)

S → $\bar{Q}$ R → Q



Nor gate

eak '1' aane se output '0' ho jaata h

S	R	Q	$\bar{Q}$	
1	0	1	0	(set state)
0	0	1	0	(Previous state)
0	1	0	1	(Reset)
0	0	0	1	(Prev state)
1	1			

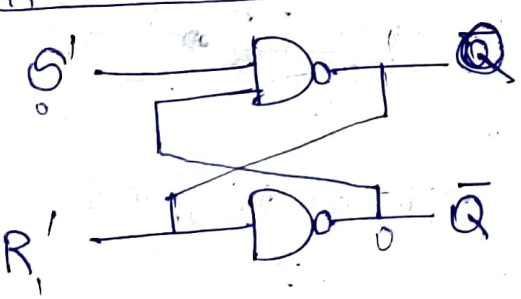
$0+Q = (1)(\bar{Q}) = \bar{Q}$

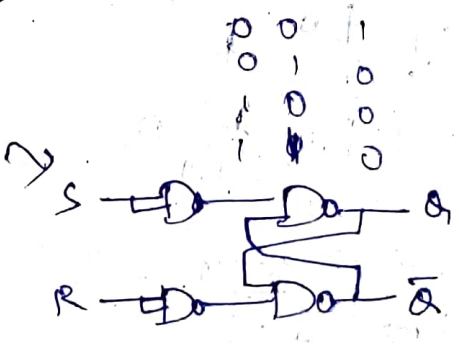
$(0+\bar{Q}) = (1)\,Q = Q$

S	R	Q	$\bar{Q}$	
0	0			No change (Hold)
0	1	0	1	
1	0	1	0	
1	1			Invalid Output (Not defined)

SR latch (NAND gates)

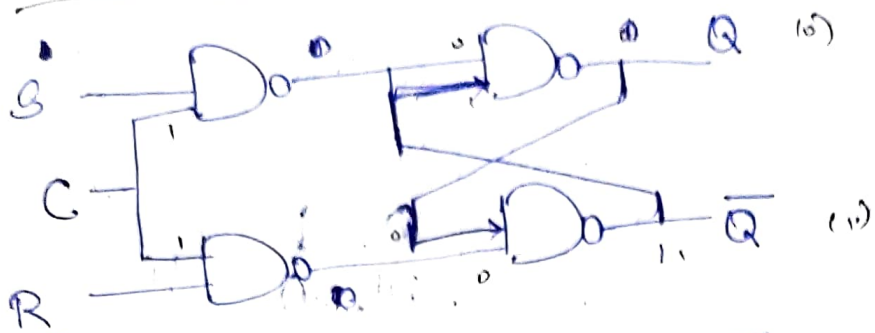
S→Q R→$\bar{Q}$





S'	R'	Q(n+1) Q	Q(n+1) $\bar{Q}$	
0	0	1	1	Forbidden
0	1	1	0	
1	0	0	1	
1	1	0	1	Last state

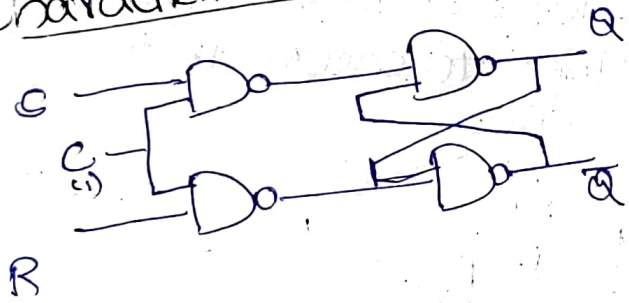
SR Flipflop (Nand gate)



Nand
① same se
② data hu

Clk	S	R	Q	$\overline{Q}$	condi
0	x	x	x	x	No change
1	0	0	x	x	
1	0	1	0	1	set Q=0 (Reset state)
1	1	0	1	0	set Q=1 (Set state)
1	1	1	1	1	Undefined

Characteristic eqn for SR flip flop



Truth table

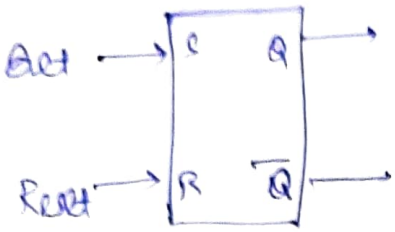
Clk	S	R	Q_{n+1}
0	x	x	x
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	Invalid

Q	S	R	$Q_{(n+1)}$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	x
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	x

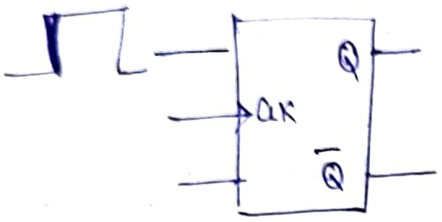
Q \ SR	00	01	11	10
0	0	0	x	1
1	1	0	x	1

$Q_{(n+1)} = S + QR$

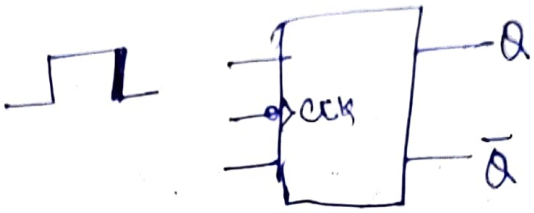
Triggering of flip flop



S	R	Q	$\bar{Q}$
0	0	Q_0	$\bar{Q}_0$
0	1	0	1
1	0	1	0
1	1	Not allow	



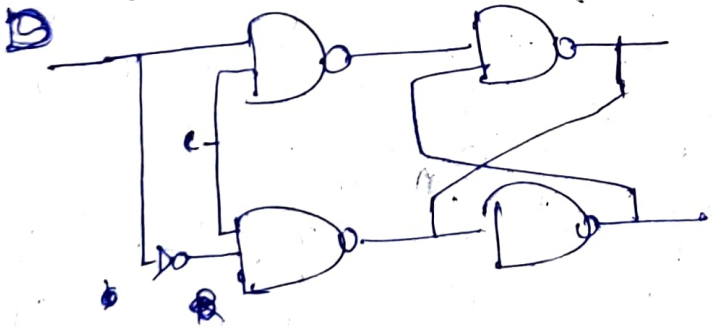
Positive edge trigg.



Negative edge trigg

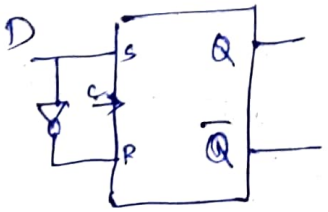
✳ D-flip flop:

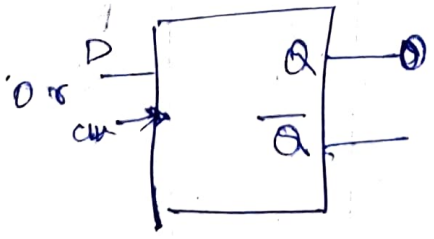
It is way to eliminate the undesirable indeterminate state in the RS flipflop to ensure that S and R are never 1 simultaneously



D=0
S=0

D=1
S=1





Clk	D	Q_{n+1}	
0	x	No change	
1	0	0	(Reset)
1	1	1	(Set)

Characteristic eqn (D flip flop)

Clk	D	Q_{n+1}
0	×	Q_n
1	0	0
1	1	1

Q_n	D	Q_{n+1}
0	0	0
0	1	1
1	0	0
1	1	1

C \ D	0	1
0	0	1
1	0	1

$Q_{n+1} = D$

Excitation state

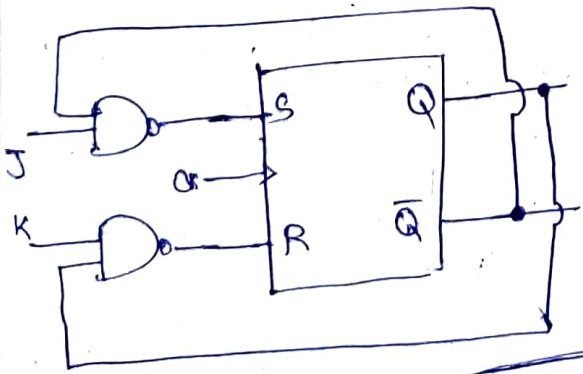
Present state Q_n	Next State Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

Q_n \ Q_{n+1}	0	1
0	0	1
1	0	1

$D = Q_{n+1}$

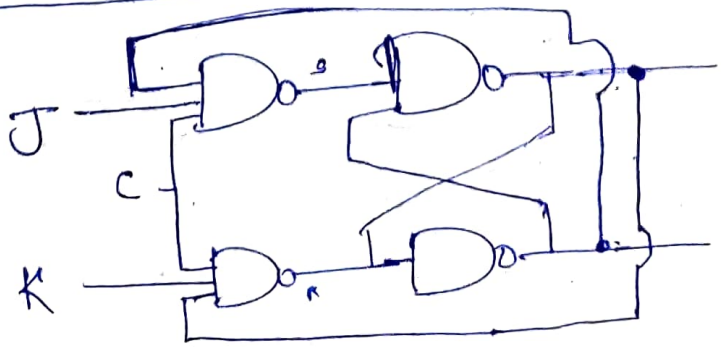
JK Flip Flop:

In SR flipflop S=R=1 should be avoided
To overcome this JK flipflop is used.



or

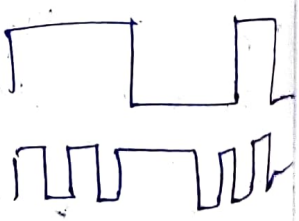
$S = J\bar{Q}_n$
$R = KQ$



J	K	Q_n	$\bar{Q}_n$	S = J$\bar{Q}_n$	R = KQ_n	Q_{n+1}	
0	0	0	1	0	0	0	Q_n
0	0	1	0	0	0	1	
0	1	0	1	0	0	0	0
0	1	1	0	0	1	0	
1	0	0	1	1	0	1	1
1	0	1	0	0	0	1	
1	1	0	1	1	0	1	$\bar{Q}_n$
1	1	1	0	0	1	0	

J	K	Q_{n+1}	
0	0	Q_n	(No change)
0	1	0	(Reset Q to 0)
1	0	1	(Set Q to 1)
1	1	$\bar{Q}_n$	(Toggles)

* When J = K = Clk = 1 then the state of flip flop keeps on toggling which leads to uncertainty of the output which is also called as Race around Condition.



Q \ JK	00	01	11	10
0			1	1
1	1			1

$$Q(n+1) = \bar{Q}J + Q\bar{K}$$

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Master - Slave JK flipflop

→ Design using two separate flip flops. One act as master other as slave.

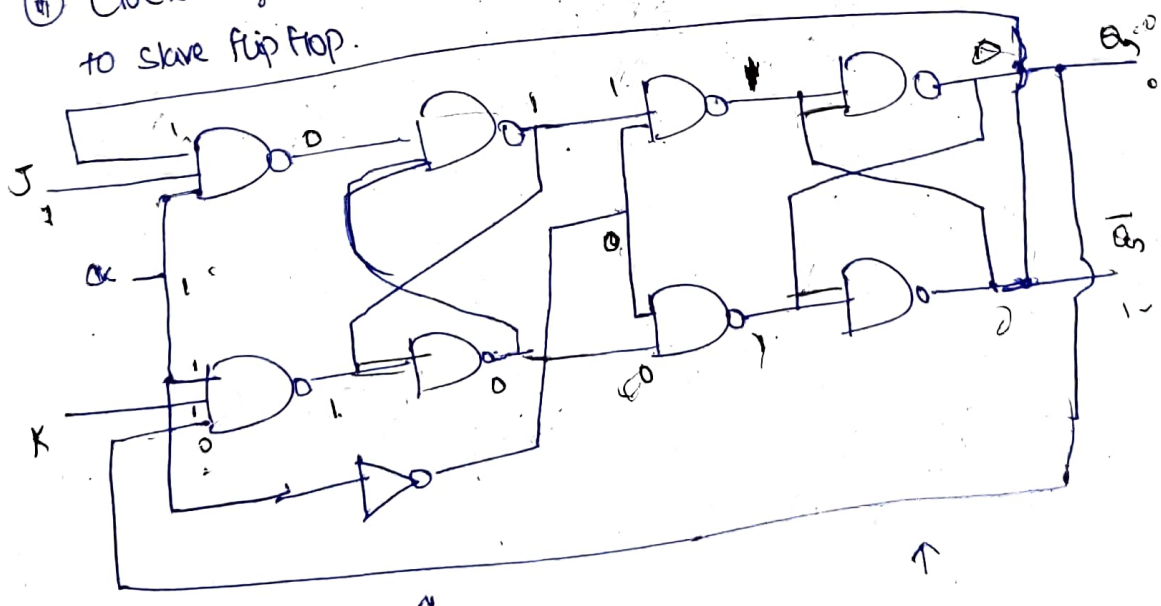
→ JK flipflop is rep in series connection

→ avoids race around condn

(i) Output of master JK flip flop is fed to input of slave JK flip flop

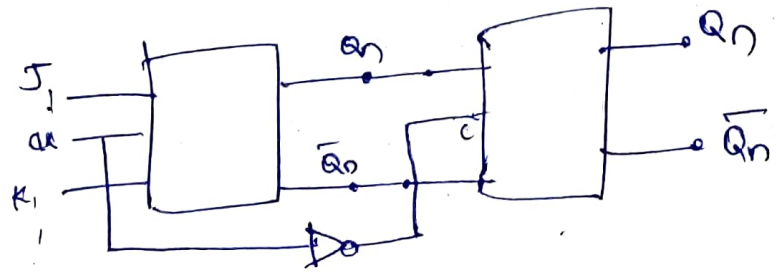
(ii) Output of slave JK flip flop is feedback to input of master JK

(iii) Clock is given to JK flipflop & is passed through not gate to slave flip flop.



Master

Clock

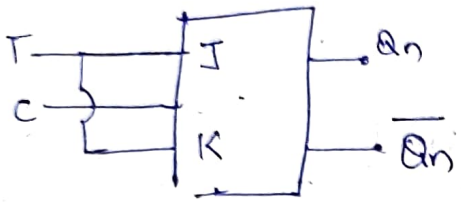


J=K=1

Clk=1

⊛ T-Flip flop

For using as toggling
J K को short कर देते है।



Clk	J	K	Q_{n+1}
0	x	x	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	$\overline{Q_n}$

Clk	T	Q_{n+1}
0	x	Q_n
1	0	Q_n
1	1	$\overline{Q_n}$

(J=K=0)

Q_n	I	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

Q_n \ T	0	1
0	0	(1)
1	(1)	0

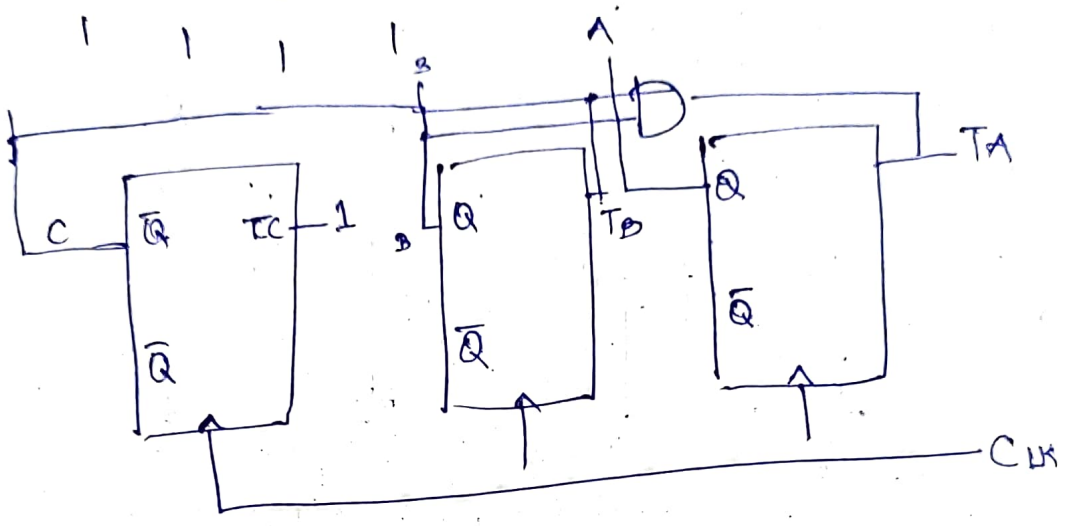
$$Q_{n+1} = \overline{Q_n}T + Q_n\overline{T}$$

A	B	C	TA	TB	TC
0	0	0	0	0	1
0	0	1	0	1	1
0	1	0	0	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	1	1	1

$TA = BC$

$TB = C$

$TC = 1$



—x—

Example

$A(t+1) = \bar{A}BCD + \bar{A}\bar{B}C + ACD + A\bar{C}\bar{D}$

$B(t+1) = \bar{A}C + C\bar{D} + \bar{A}B\bar{C}$

$C(t+1) = B$

$D(t+1) = D'$

$Q(t+1) = J\bar{Q} + Q\bar{K}$

$A(t+1) = (\bar{B}CD + \bar{B}C)A' + A(CD + \bar{C}\bar{D})$

$J = (\bar{B}CD + \bar{B}C) = \bar{B}C$

$K = (CD + \bar{C}\bar{D})' = \bar{C}\cdot D + CD'$

$B(t+1) = \bar{A}C + C\bar{D} + \bar{A}B\bar{C}$

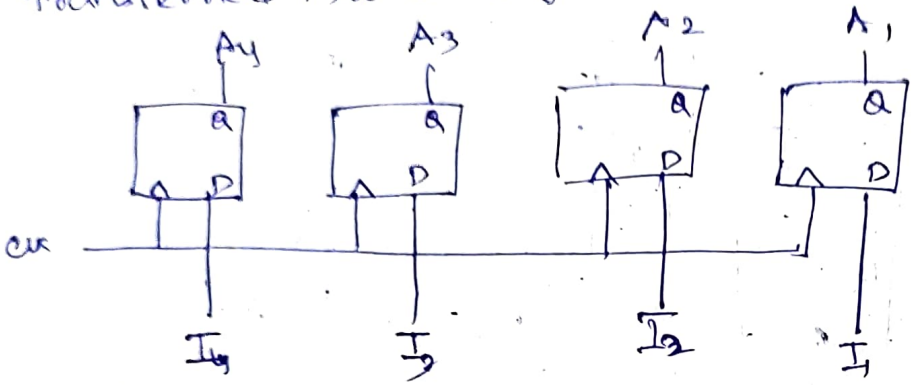
$B(t+1) = (\bar{A} + \bar{D})C + (\bar{A}B)\bar{C}$

$J = \bar{A} + \bar{D}$

$K = (\bar{A}B)' = A\bar{B}$

⊛ Register

→ Flip-flops are essential comp.
→ Circuit that include flip flop
→ Register is a collection of flipflop
↳ used to store single bit digital data.
→ It also contain combinational gates that perform data processing tasks.
→ The gate control when & how new info is transferred into the registrar



→ A register capable of shifting its binary infor in both or one direction is called shift register

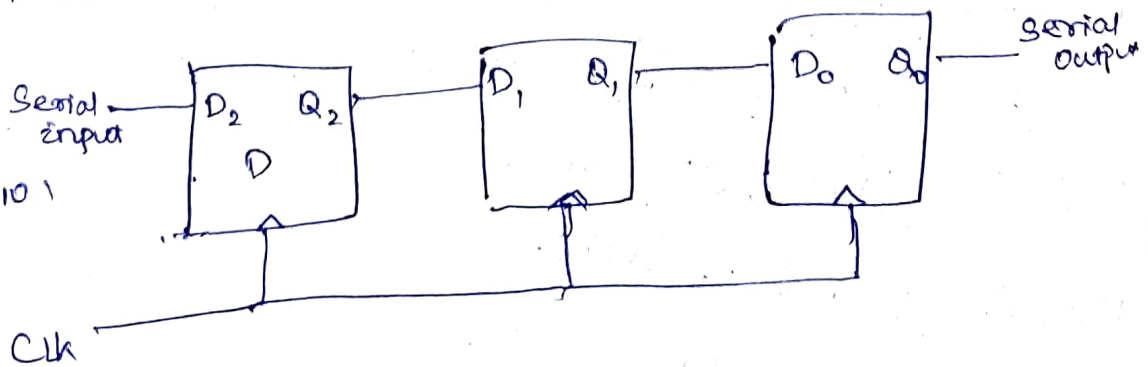
⊛ Shift Register :

→ used to implement arithmetic operation

4 types
① SISO ② SIPO ③ PISO ④ PIPO

① SISO

Allows serial input and produces serial output



	CLK	Q_2	Q_1	Q_0
	0	0	0	0
1	↑	1	0	0
0	2	0	1	0
1	3	1	0	1
0	4	0	1	0
0	5	0	0	1

shift right

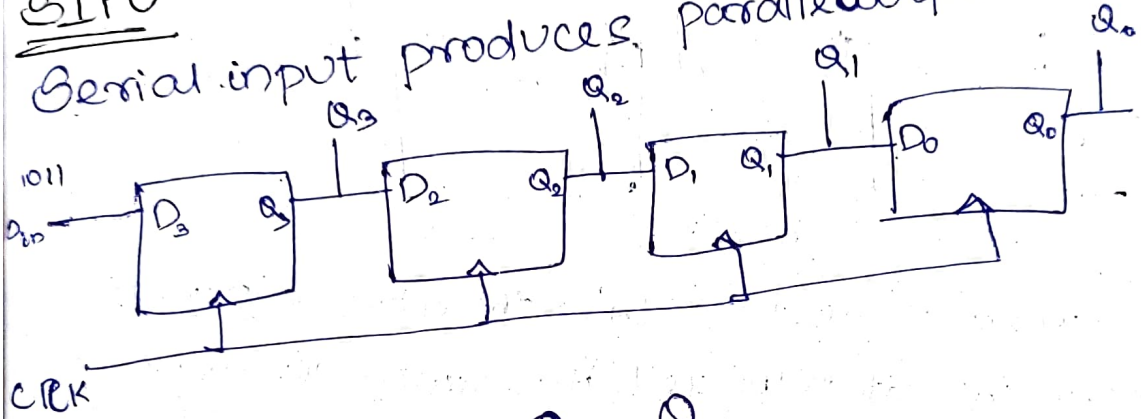
n → bit required to store data

n−1 bit to read the data

2n−1 for total proc

SIPO

Serial input produces parallel output



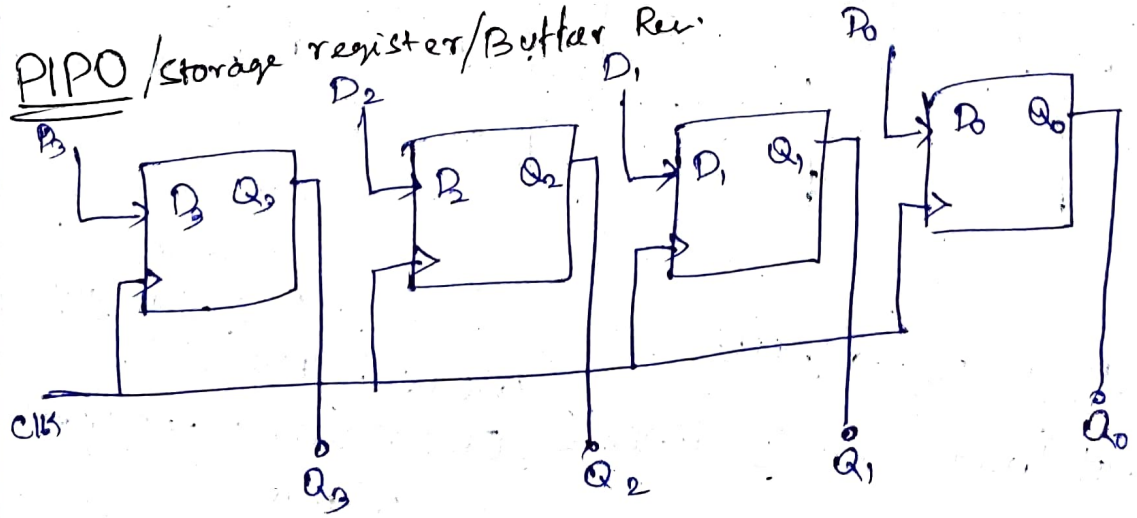
CLK	Q_3	Q_2	Q_1	Q_0
Initially	0	0	0	0
↑	1	0	0	0
↑	0	1	0	0
↑	1	0	1	0
↑		1	0	1

n-bits to store data

0 clock for use of clock

n for total proc.

PIPO / Storage register / Buffer Reg



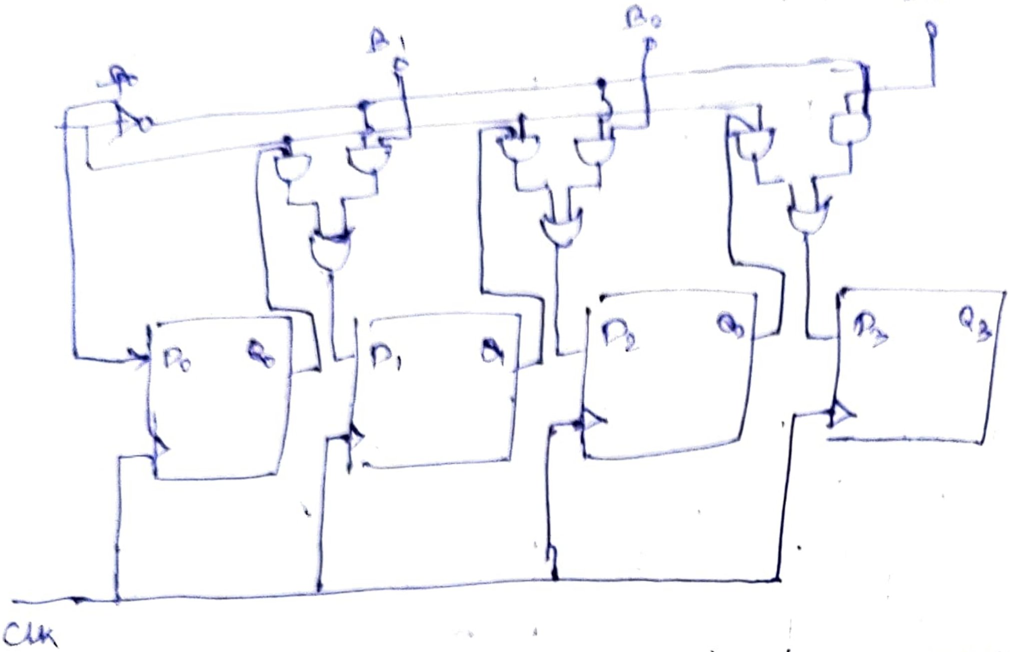
1 clock to store data

0 clock to read data

Total processing = 1

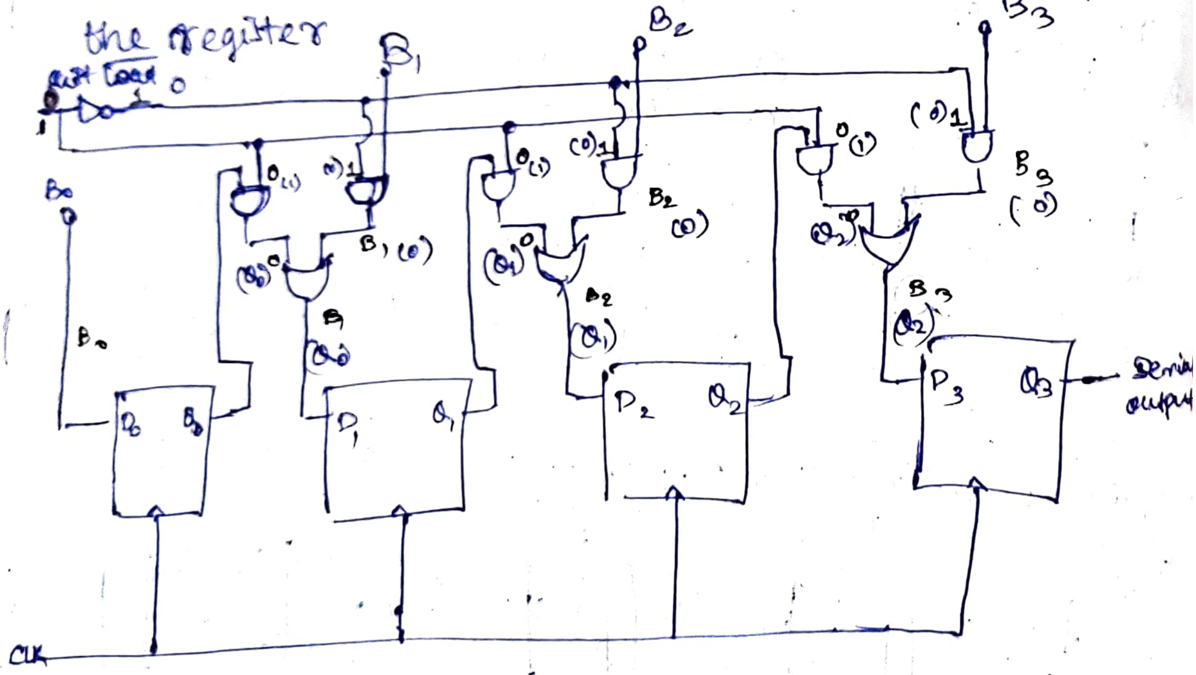
PISO

Operates in 2 modes (i) Load mode → Active low
(ii) Shift mode → Active high



→ Load mode is used to load the data in parallel.

→ After the data is loaded, it is shifted serially out of the register



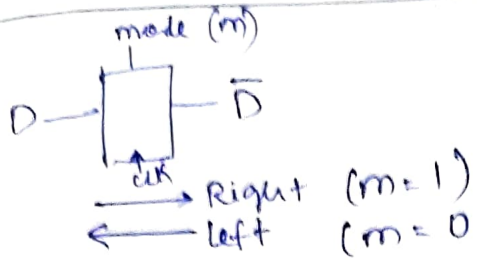
(B_0, B_1, B_2, B_3) → Inputs

$Shift/\overline{Load}$ → 2 lines – One with not → Input – One and gate
– One normal → one and gate – Outputs

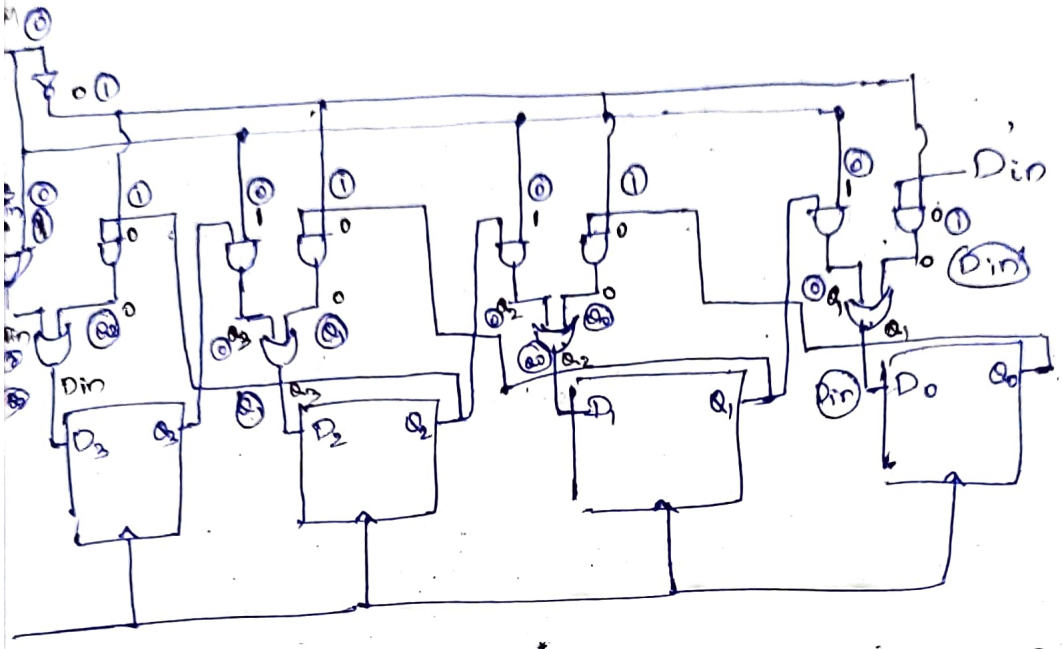
Load mode = 0

Shift mode = 1

Bidirectional Shift register



(See video)



m = 1 → right shift

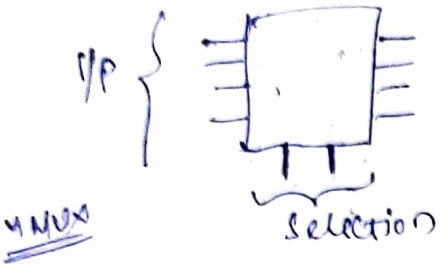
$D_{in} \rightarrow Q_3 \rightarrow Q_2 \rightarrow Q_1 \rightarrow Q_2$

m = 0

$Q_2 \leftarrow Q_1 \leftarrow Q_0 \leftarrow D_{in}$

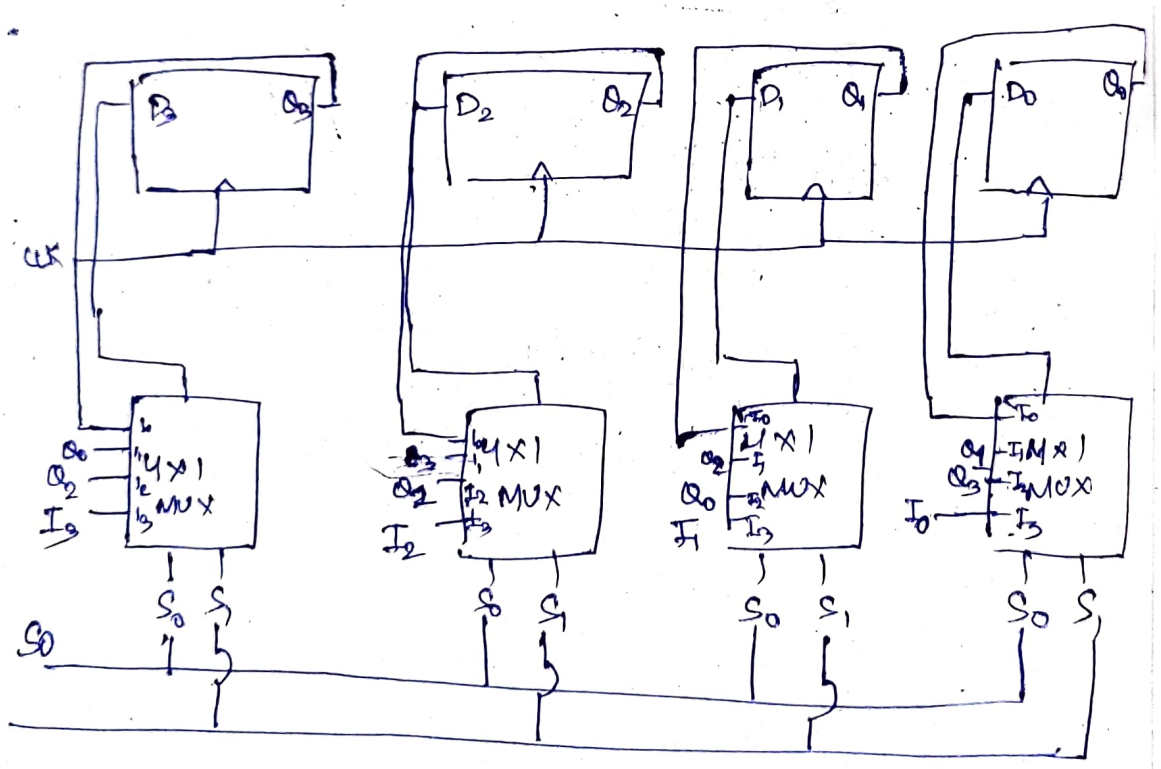
⊛ Universal Shift register

Shift left
Shift right
PIPO
HOLD data
} 4 func^n



4 MUX

S_0	S_1	func^n
I_0 → 0	0	Hold
I_1 → 0	1	Right shift
I_2 → 1	0	Left shift
I_3 → 1	1	PIPO



Counter : used to count pulses as well as frequency dividers

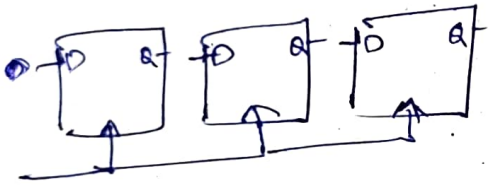
→ No. of times the specific time has been occured
→ Input must be digital.
→ Single clock input & Multiple output
→ The output rep. BCD
→ Modulus is the no. of diff. output states through which counter goes through before returning to its first state

(For n bits max counts = 2^n from 0 to n-1)

For 4 Bit MOD 16
3 Bit MOD 8

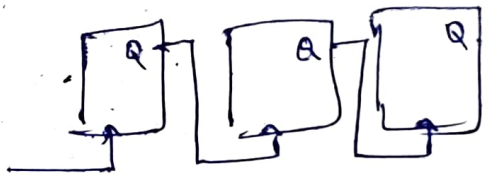
Synchronous

→ Parallel counter
→ Circuit delay is prop. delay of one flip flop.
→ All flip-flops of the counter driven by same clock.



Asynchronous

→ Ripple counter
→ Circuit delay is prop delay of all flip flops.
→ Output of one flipflop is used as clock input of other flip flop

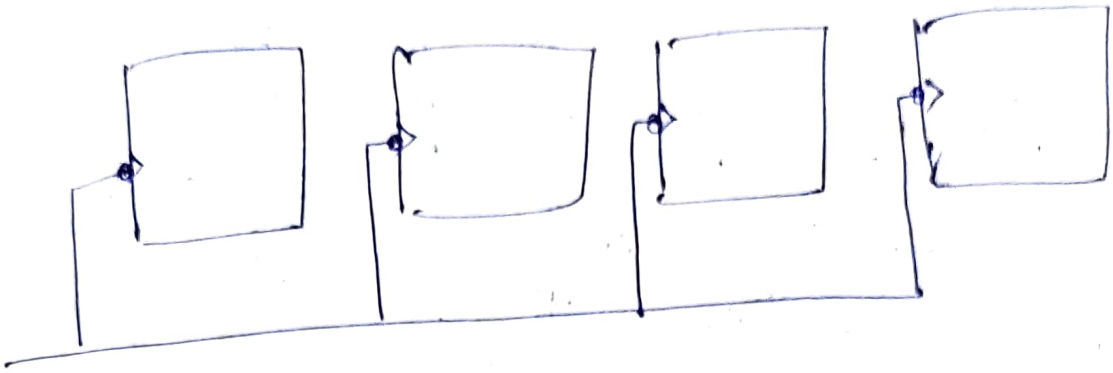


BCD (Decade) Counter

MOD 10 counter

→ counts 0 to 9 (4 bits)

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

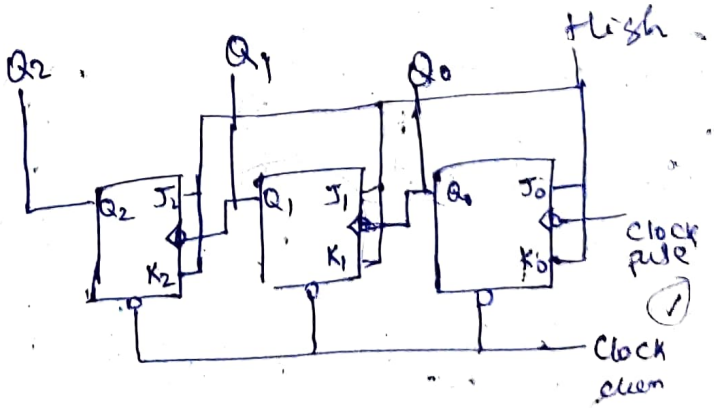


Asynchrous counter

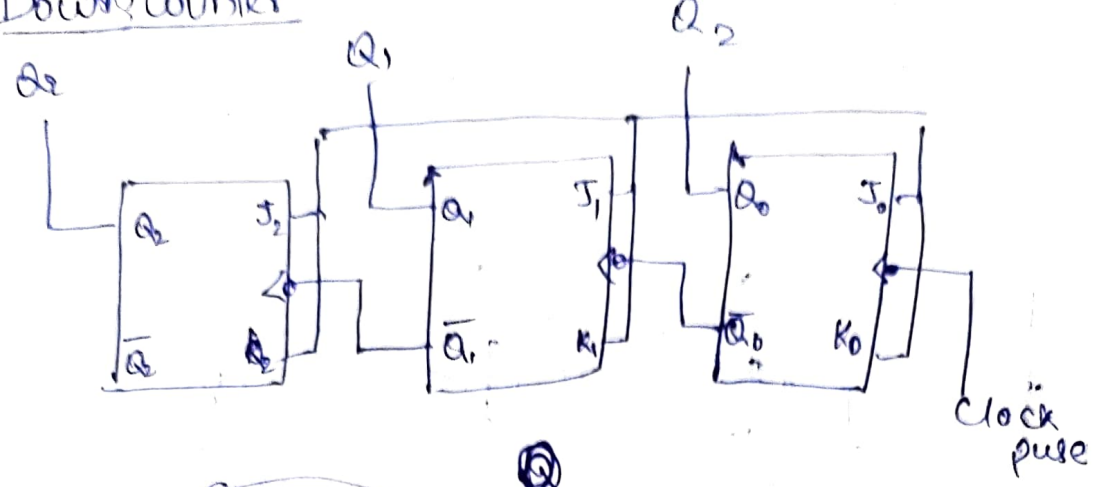
→ Ripple counter
→ Clock pulse ripples throughout the ckt
→ Diff. flipflops are used with a diff. clock pulse
→ All flip flops are used in toggle mode.
→ The flipflop applied with external clock pulse act as LSB.
→ It may be up or down counter

Up counter

State	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



Down Counter



	→	↓
Q	Down	UP
$\bar{Q}$	UP	Down

Ring Counter : (Synchronous counter)

→ Composed of flip flop working as shift register

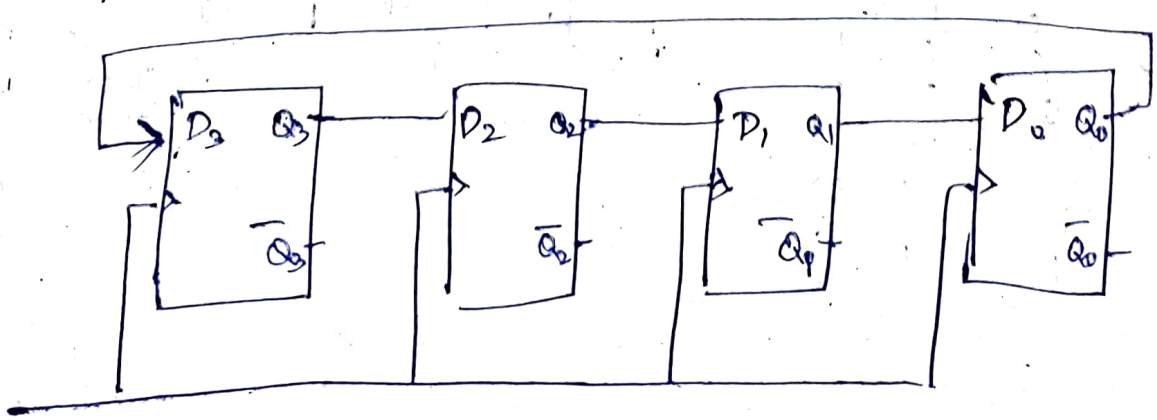
→ The output of the last flip flop fed to the input of the first

Straight

Output of last flipflop connects to the first flip flop of the shift register input

Twisted (Johnson)

→ Complement output of the last flip flop of shift register to input of the first flip flop

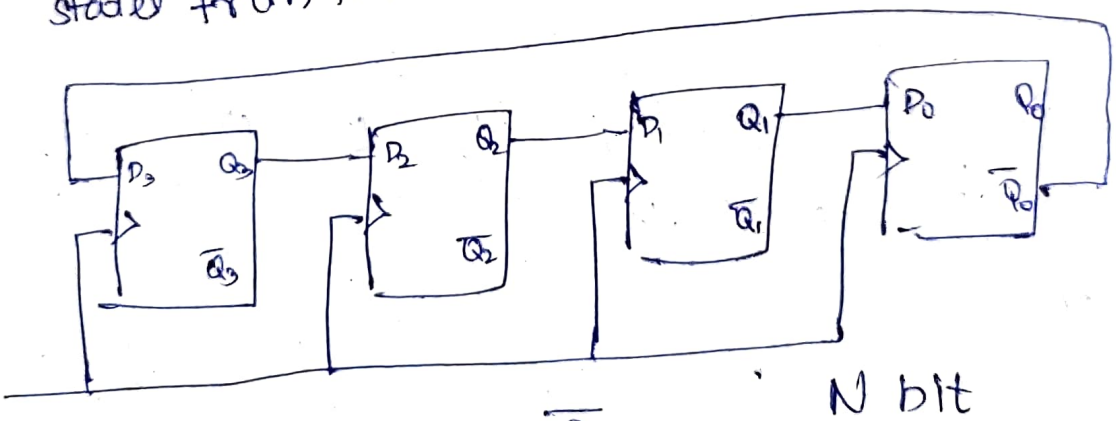


$$Q_3^+(n+1) = D_3 = Q_0(n)$$

State	Q_3	Q_2	Q_1	Q_0	
0	0	0	0	0	n bit reg
1	0	0	0	1	IN| states
2	0	0	1	0	usable ⇒ n
3	0	1	0	0	Unusable ⇒ $2^n - n$
4	1	0	0	0	
5	3				
6	0				
7	1				
	2	1			
	3		1		

Johnson Counter

→ favoured as they offer twice as many count states from the same no. of flip flops.



N bit

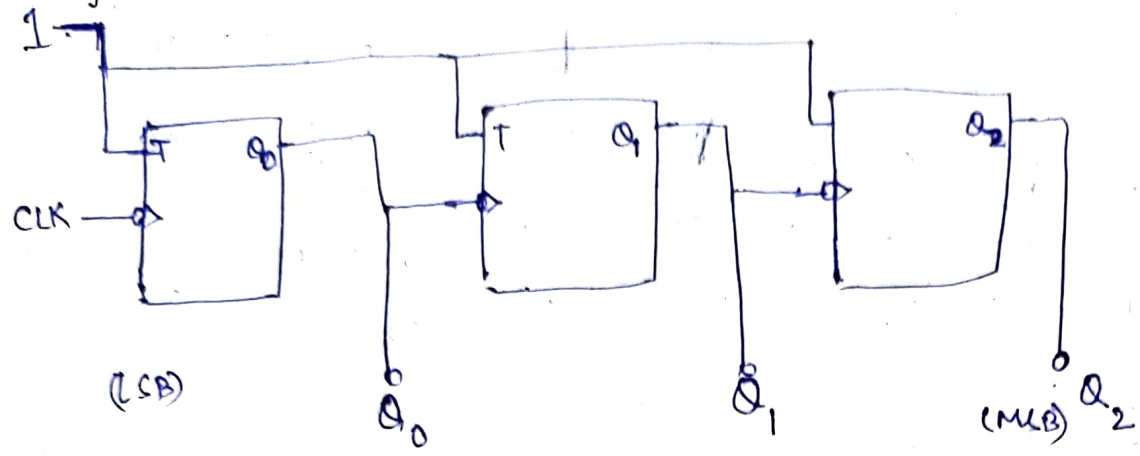
2N states

$Q_3^+(n+1) = D_3 = \overline{Q_0}(n)$

$Q_2(n+1) = D_2 = Q_3(n)$

0	0	0	0
0	0	0	1
0	0	1	1
0	1	1	1
1	1	1	1
1	1	1	0
1	1	0	0
1	0	0	0
0	0	0	0

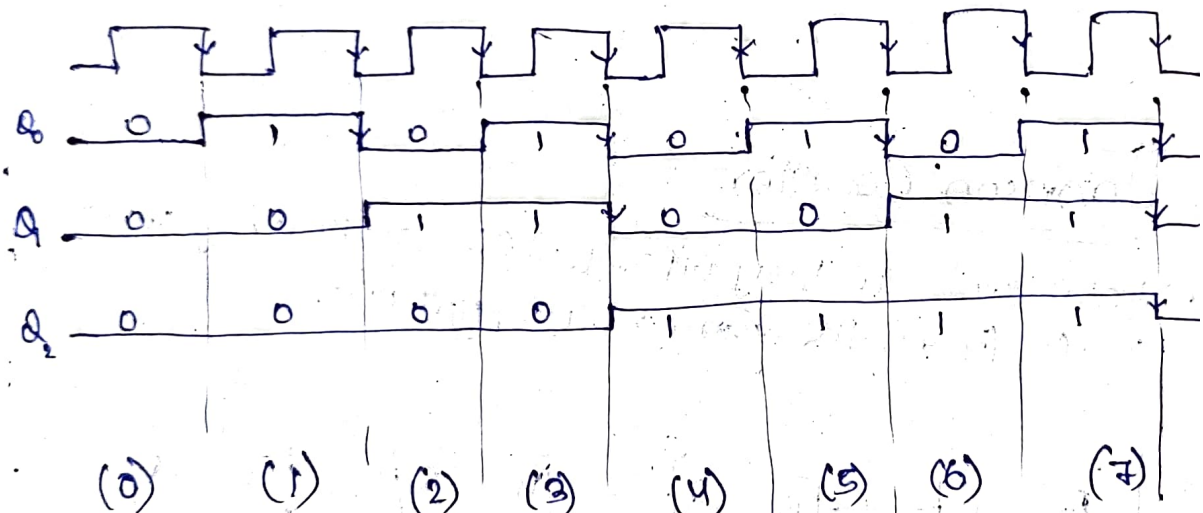
Asynchronous UP counter



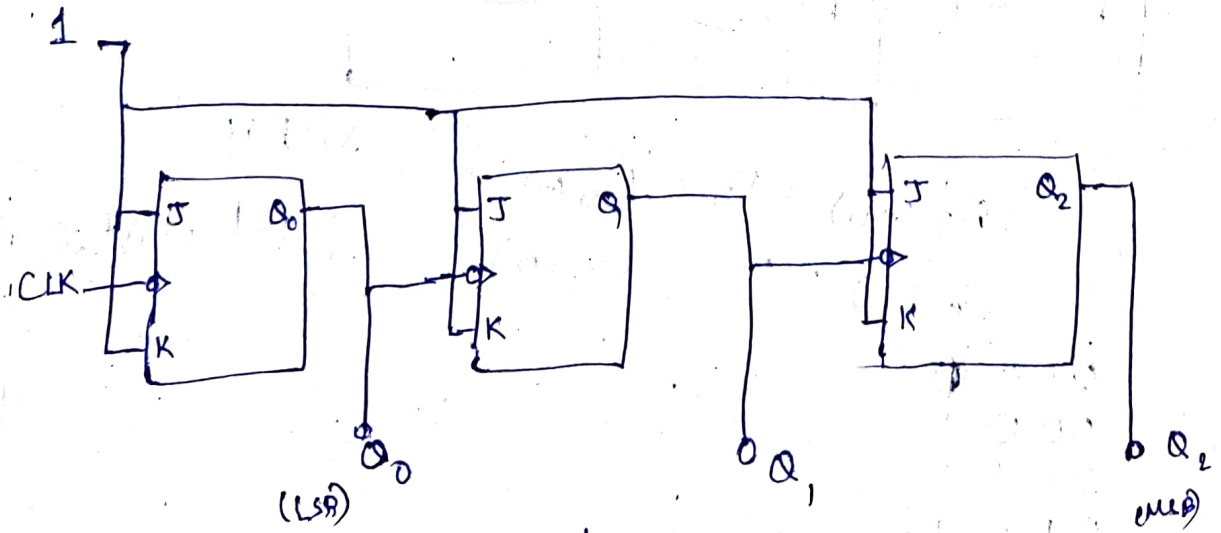
In t-flip flop when input is 1, it will toggle

Note: there is negative edge triggered

OR

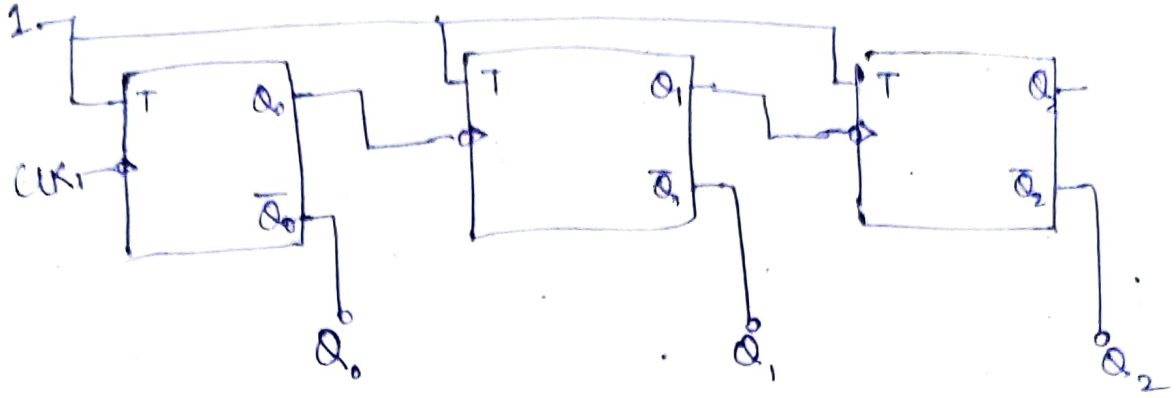


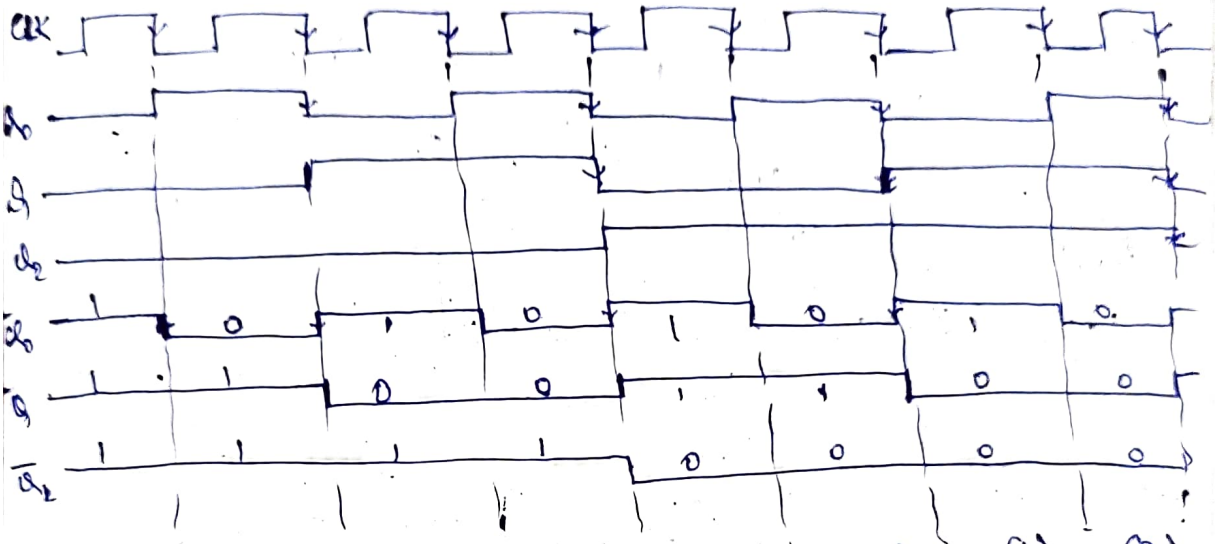
In case of J-K flip flop



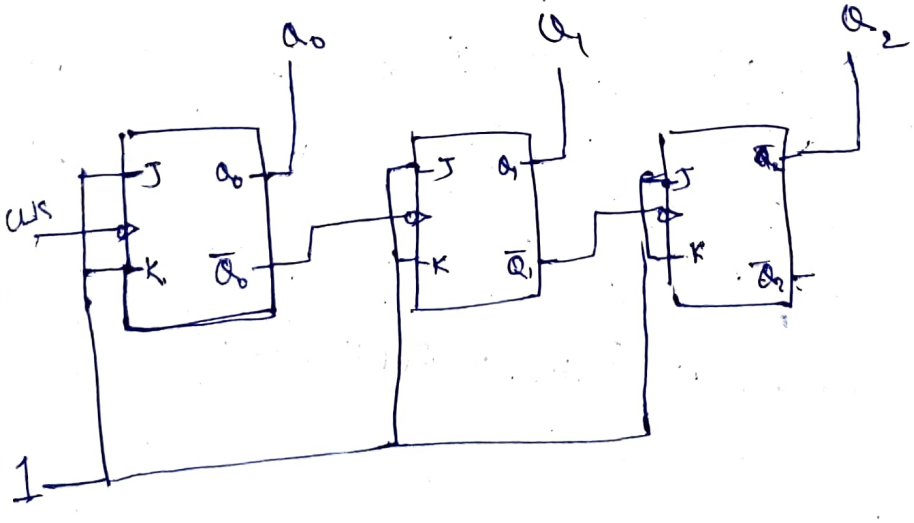
Short कर देना है बस

Asynchronous (Ripple) down Counter





(7) (6) (5) (4) (3) (2) (1) (0)



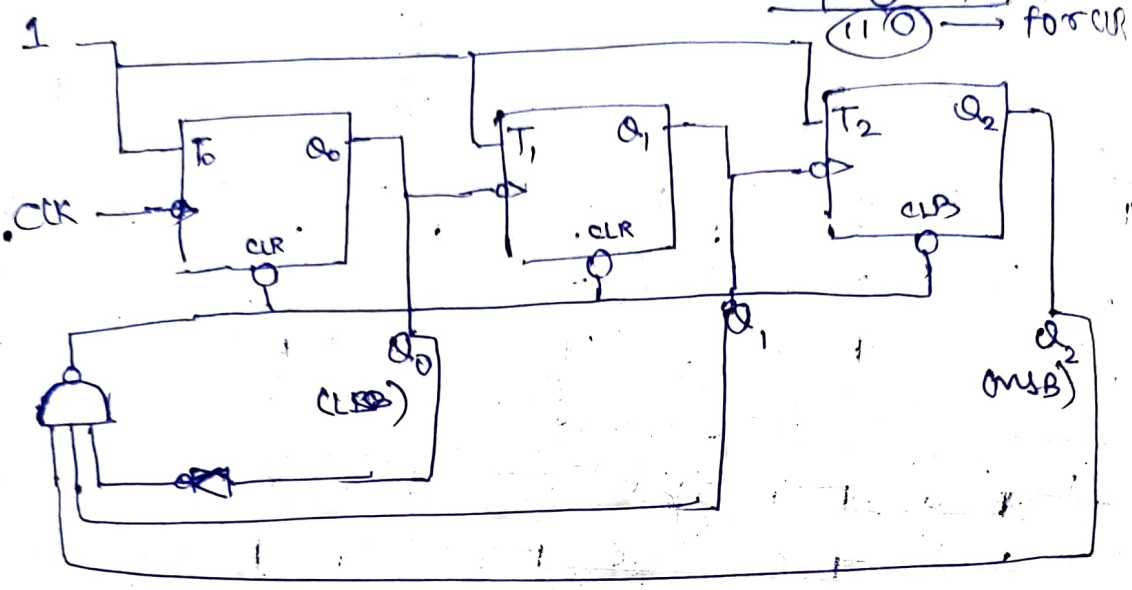
✱Modulo Counter by Asynchronous counter:

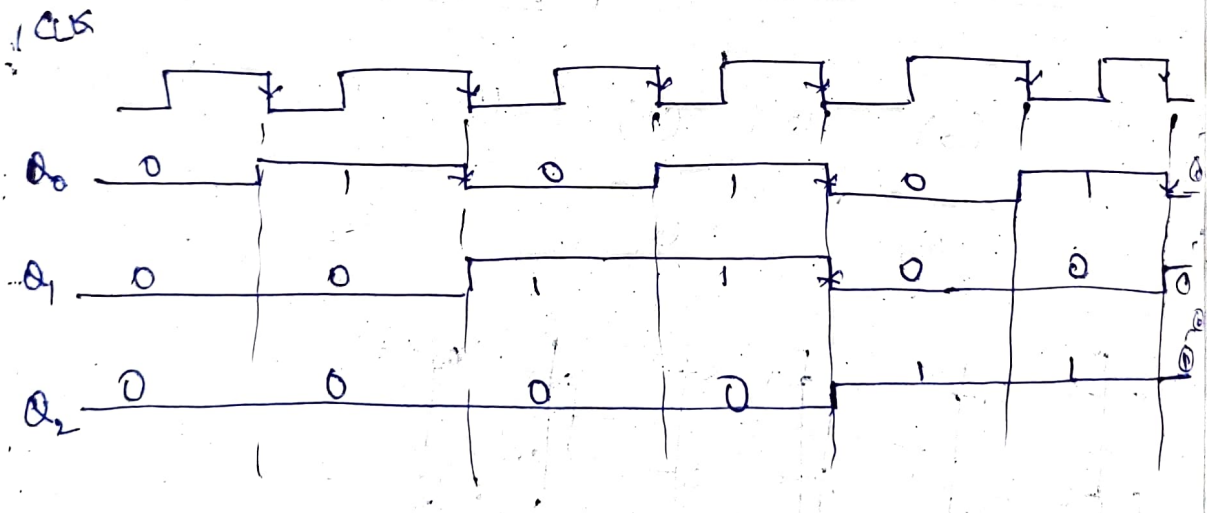
Eg. Modulo 6. counter

To count from 0 to 5

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0

for CLR





CLR is connected to NAND gate

BCD (Decade) Counter

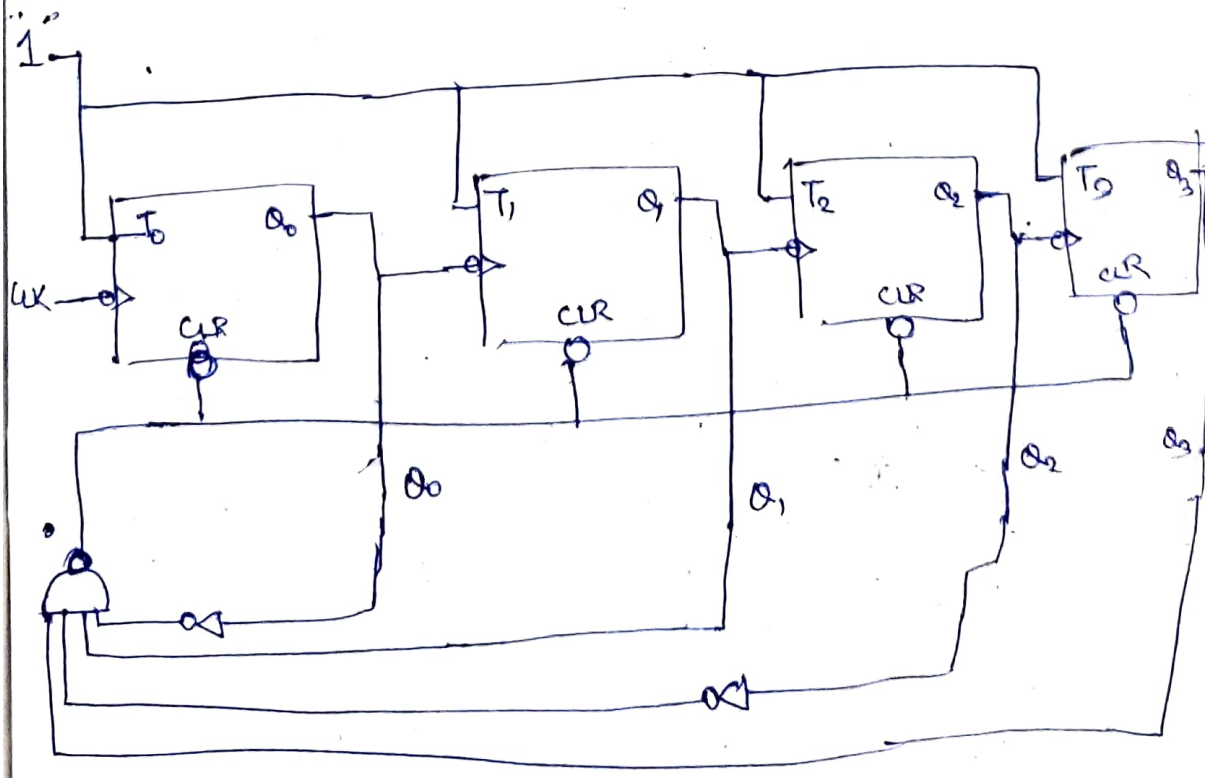
In BCD counter it counts 0 to 9

To count (0 to 9) it takes 4 bits

Q_3	Q_2	Q_1	Q_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0

CLR terminal is Active (0) low

CLR



2-bit synchronous counter by JK flip flop

Ring Counter

Module - 5

Counter:

It is a device used to count the no. of times a particular event occurred.

→ The output represents binary or BCD numbers.

→ Single clock input & multiple output.

→ Used as frequency divider

Modulus

Total no. of state through which a counter has progress.

2^N

$(0 \text{ to } 2^n - 1)$ State

Synchronous

→ Parallel counter

→ The clock signal is common to all flip flop.

→ Simpler design

→ Circuit delay is propag. delay of one flip flop.

→ Faster

→ High cost

→ large no. of logic gates are req.

Asynchronous

→ Ripple counter

→ Output of one flip flop act as clock input of another flip flop.

→ Complex design

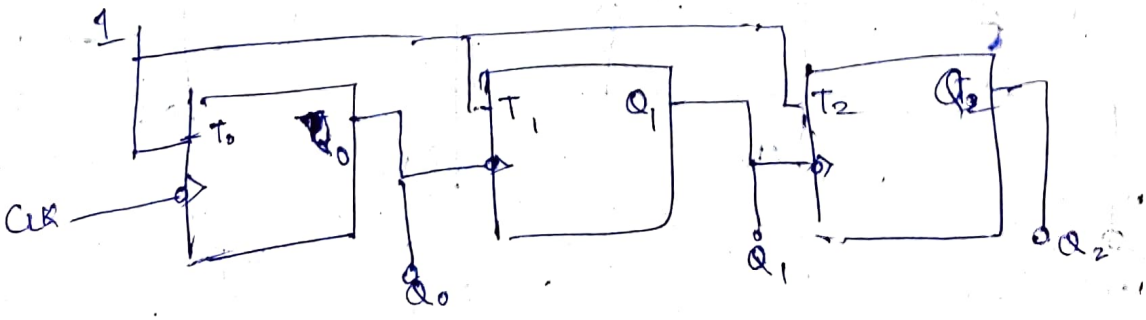
→ Circuit delay is propag. delay of all flip flop

→ slower

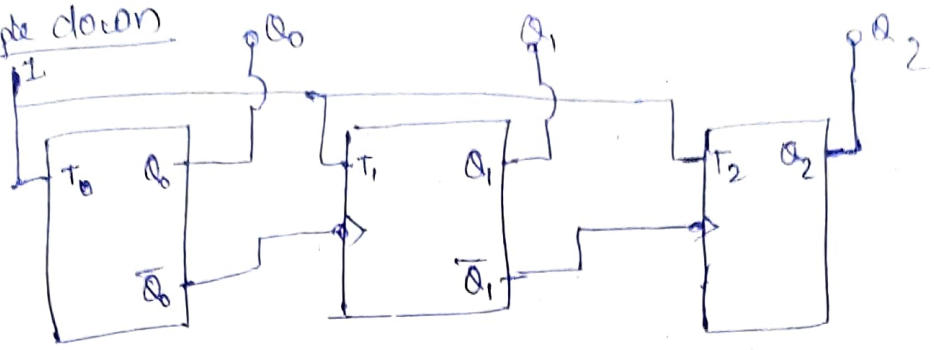
→ low cost

→ Less no. of logic gates are required.

Ripple Up



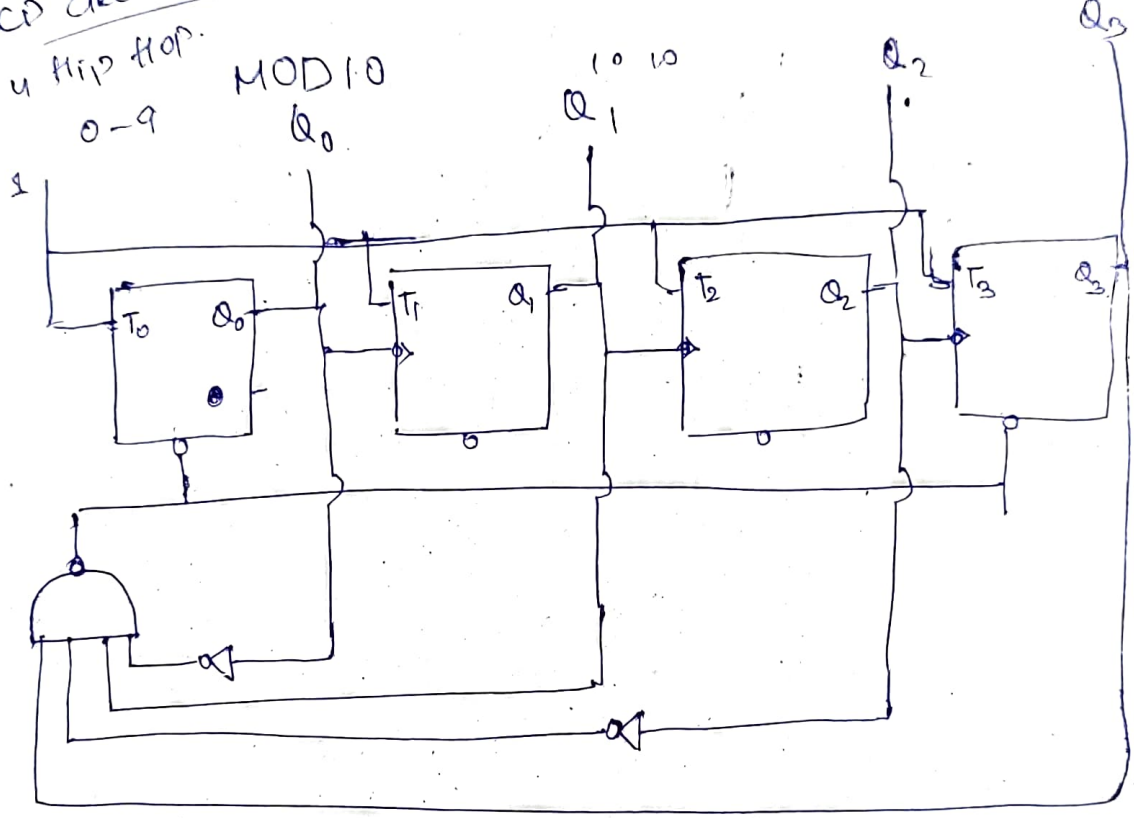
Ripple down



BCD decade counter :-

4 flip flop.

0-9 MOD 10



Q) 0 → 1 → 3 → 7 → 6 → 4 Ans circuit (T), (J)(K)

Q_2	Q_1	Q_0	Q_2^*	Q_1^*	Q_0^*	T_2	T_1	T_0
0	0	0						
0	0	1						
0	1	0						
0	1	1						

Ring Counter :-

Collection of flip flop working as shift register, with the output of last flip-flop fed to the input of the first. making a circular or ring-like structure.

Binary counter → 2^n states for n bits

Ring → n states

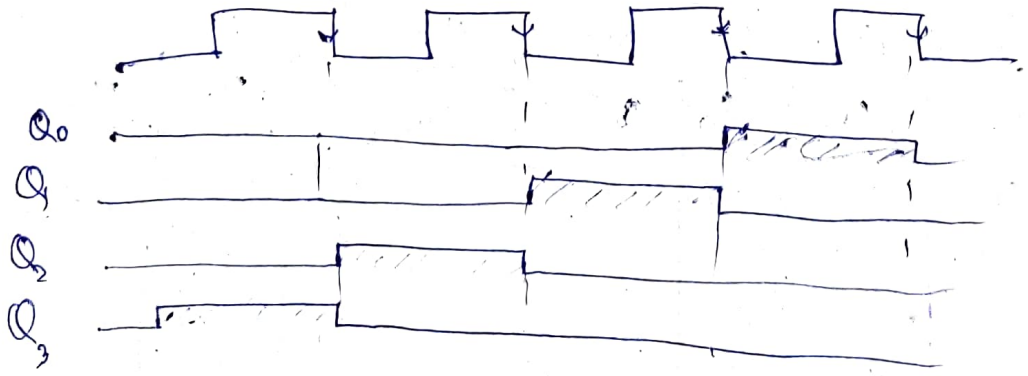
Johnson → 2n states

Clear & preset is used

Over riding input

		Q_3	Q_2	Q_1	Q_0
⊓	↑	1	0	0	0
1	↓	1	1	0	0
1	↓	0	1	1	0
1	↓	1	0	1	1

		Q_3	Q_2	Q_1	Q_0
⊔	↑	1	0	0	0
	↓	0	1	0	0
		0	0	1	0
		0	0	0	1



Johnson

2n State (twice as many states from the same no. of flip flops).

Output of the complement of last flip flop is fed as input to the first flip flop.

Used in comm. system as adj. bit are differ by only one bit

CLK	Q_0	Q_1	Q_2	Q_3	
↓	0	0	0	0	(0)
↓	1	0	0	0	(1)
↓	1	1	0	0	(2)
↓	1	1	1	0	(3)
↓	1	1	1	1	(4)
↓	0	1	1	1	(5)
↓	0	0	1	1	(6)
↓	0	0	0	1	(7)
↓	0	0	0	0	(0)