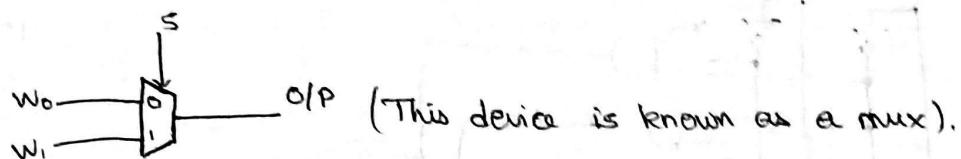


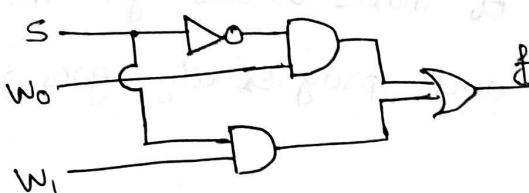
MUX



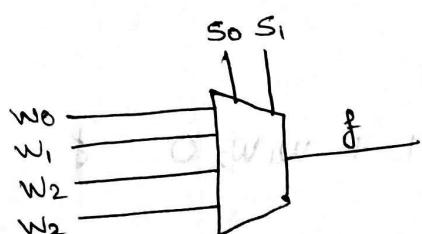
- If selection bit is 0, O/P is W_0 and if selection bit is 1, O/P is W_1 .
- For 2^n inputs if one output is required, 'n' number of selections should be made by picking 1 out of 2^n inputs.

S	f	(for $n=1$)
0	W_0	
1	W_1	

$$\therefore f = \bar{S}W_0 + SW_1$$

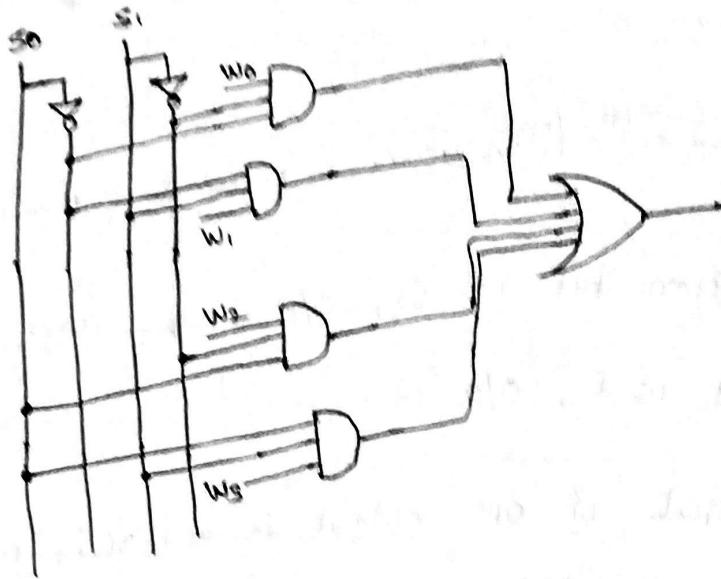


S_0	S_1	f	(for $n=2$)
0	0	W_0	
0	1	W_1	
1	0	W_2	
1	1	W_3	

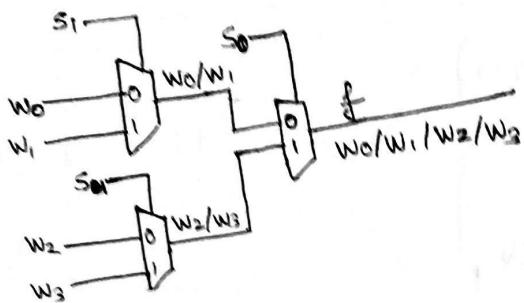


$$f = \bar{S}_0 \bar{S}_1 W_0 + \bar{S}_0 S_1 W_1 + S_0 \bar{S}_1 W_2 + S_0 S_1 W_3$$

- The function variable can also be different (i.e. not necessarily W_0, W_1, W_2, \dots).



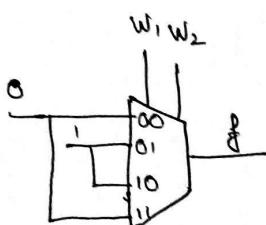
→ Realising 4:1 mux using 2:1 mux:-



→ The leftmost column of muxes should get the highest index of s and progressively goes rightwards upto s_0 .

$$\rightarrow f = w_1 \oplus w_2$$

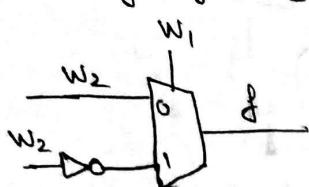
w_1	w_2	f
0	0	0
0	1	1
1	0	1
1	1	0



$$\overline{w_1} \overline{w_2} 0 + \overline{w_1} w_2 1 + w_1 \overline{w_2} 1 + w_1 w_2 0 = f$$

$$f = \overline{w_1} w_2 + w_1 \overline{w_2}$$

→ Implementing $f = w_1 \oplus w_2$ using 2:1 mux.



Q1 $f_1 = \sum m(3, 5, 6, 7)$

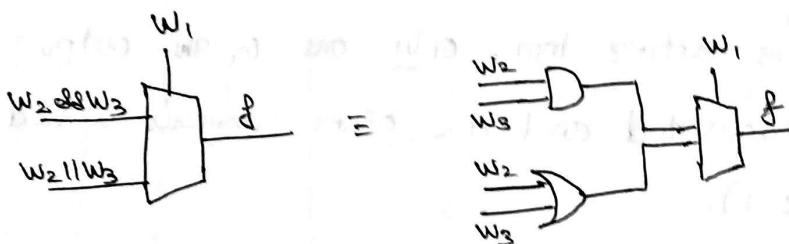
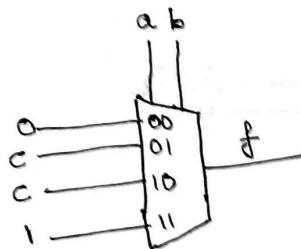
no	a	b	c	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

bc	00	01	11	10
0	0	1	1	0
1	1	0	0	1

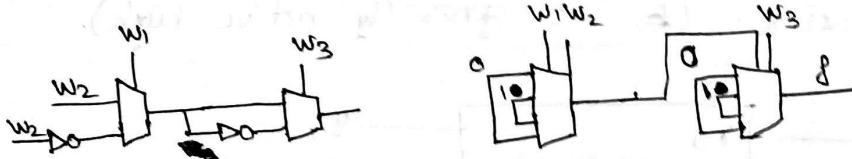
$$bc + ac + ab$$



$$f = ab + bc + ac$$



Q2 $f = w_1 \oplus w_2 \oplus w_3$ using 4:1 and 2:1 muxes. (only using muxes),



Q3 $f = \overline{w_1 w_3} + w_1 w_2 + w_1 w_3$ using 4:1 and 2:1

Q4 $f = w_1 w_2 + w_1 w_3 + w_2 w_3$ using 2:1

Q5 $f = \overline{w_2 w_3} + \overline{w_1 w_2} \overline{w_3} + w_2 \overline{w_3} w_4 + w_1 \overline{w_2} \overline{w_4}$

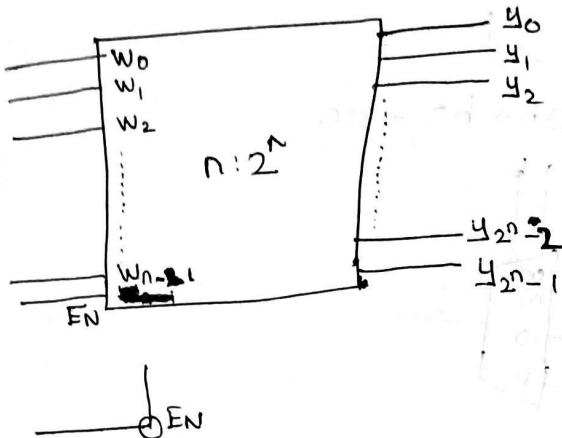
Q6 When is common cathode, common anode preferred?

Justify?

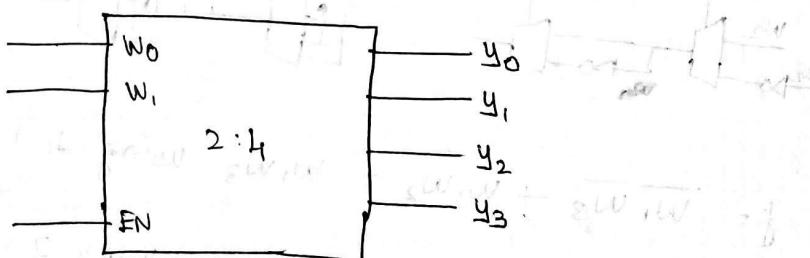
Decoder

~~given~~
in
↓
out

→ For n inputs to the decoder, 2^n output signals are emitted.



- If ~~decoder~~ is active low, only one of the output will be deactivated and all other signals would be active (ie. 1).
- If ~~decoder~~ is active high, only one of the output will be active (ie. 1) and all other signals would be inactive. (~~decoder~~ is generally active high).

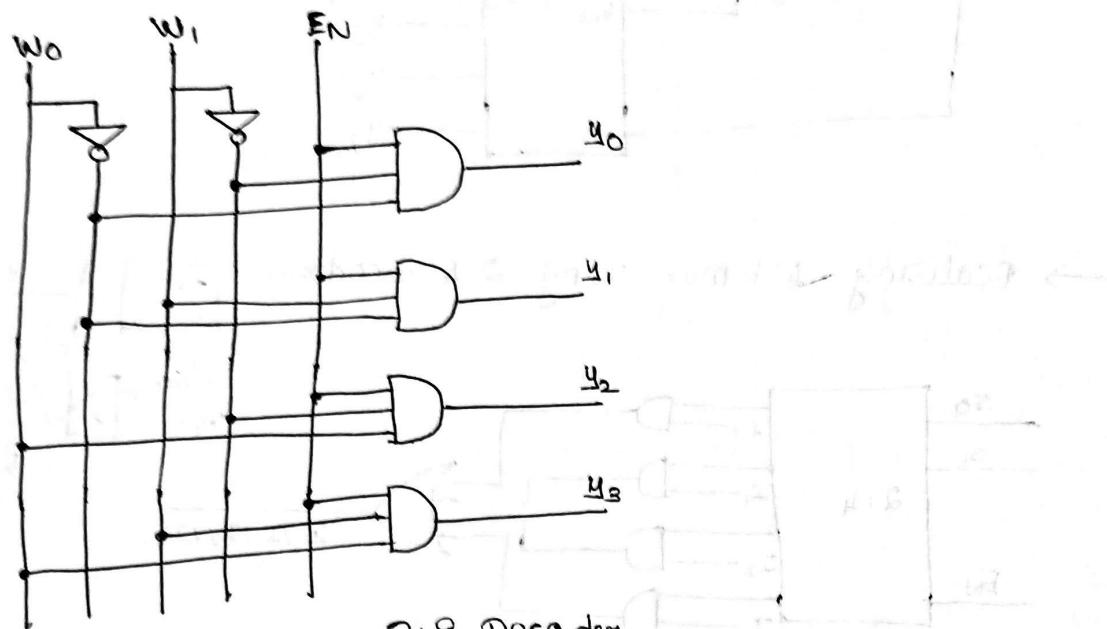
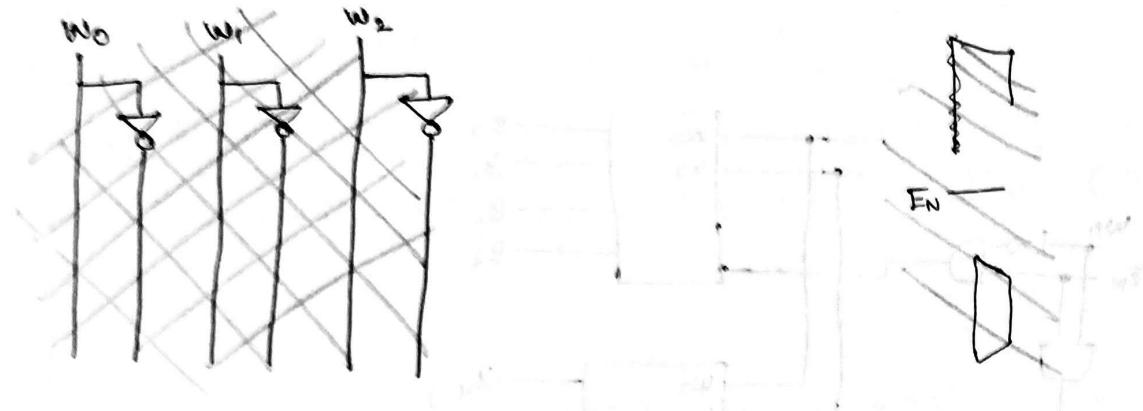


- EN should be 1 for the decoder to function. If EN is 0, the output will all be 0 irrespective of input.

TRUTH TABLE

91

E_2	π_0	π^+	π^-	π^0	π^+	π^-	π^0	π^+	π^-
1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
0	x	x	0	0	0	0	0	0	0



3:8 Decoder

$$y_0 = EN \overline{W_0} \overline{W_1} \overline{W_2}$$

$$y_1 = EN \overline{W_0} \overline{W_1} W_2$$

$$y_2 = EN \overline{W_0} W_1 \overline{W_2}$$

$$y_3 = EN \overline{W_0} W_1 W_2$$

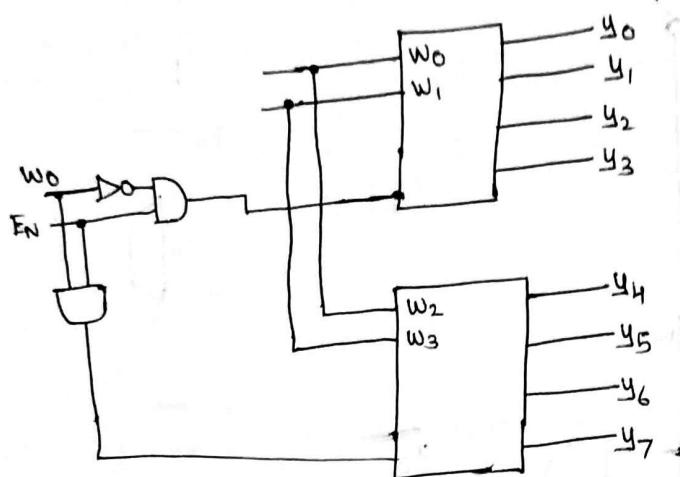
~~$y_4 = EN W_0 \overline{W_1} \overline{W_2}$~~

$$y_5 = EN W_0 \overline{W_1} W_2$$

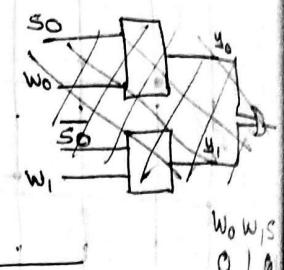
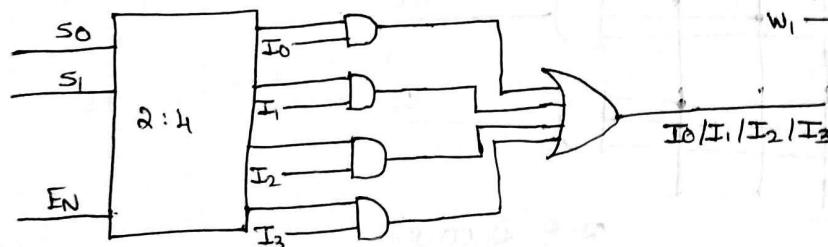
$$y_6 = EN W_0 W_1 \overline{W_2}$$

$$y_7 = EN W_0 W_1 W_2$$

→ Realising 3:8 decoder using 2 2:4 decoders



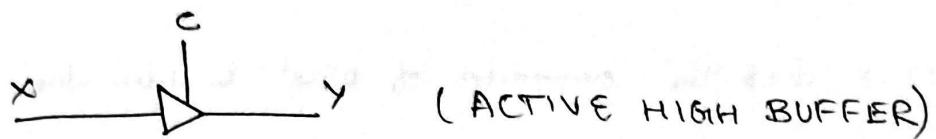
→ Realising 4:1 mux using 2:4 decoder:-



~~Realising 4:1 mux using 2:1:~~

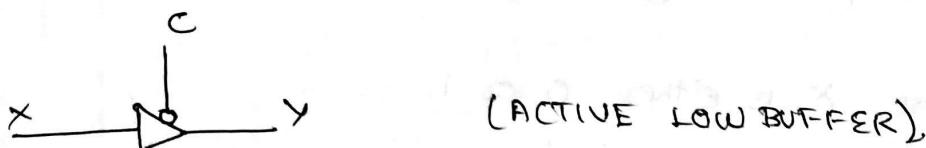
On Realise 4:16 decoder using 2:4 decoders.

BUFFER



X	C	Y
0	0	1
0	1	0
1	0	1
1	1	1

→ The input is equal to output only when $C = 1$.

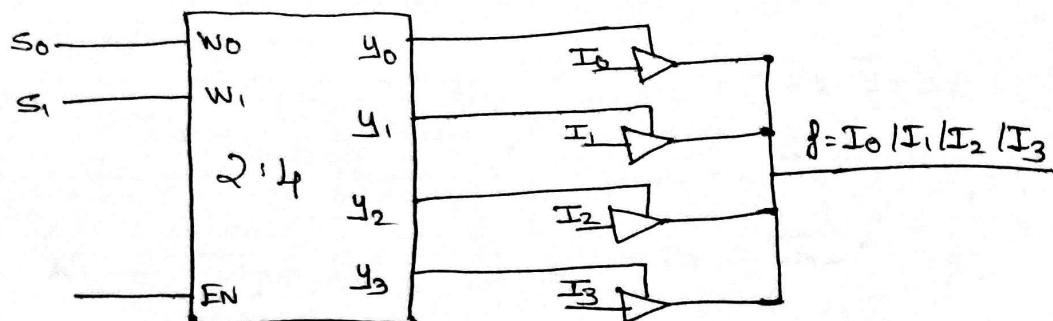


X	C	Y
0	0	0
0	1	1
1	0	1
1	1	1

→ There are three possible outputs for this buffer.

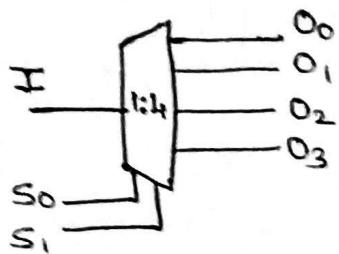
→ 1 is called the high impedance state.

→ This buffer is also known as transmission gate or tri-state buffer.



DEMUX

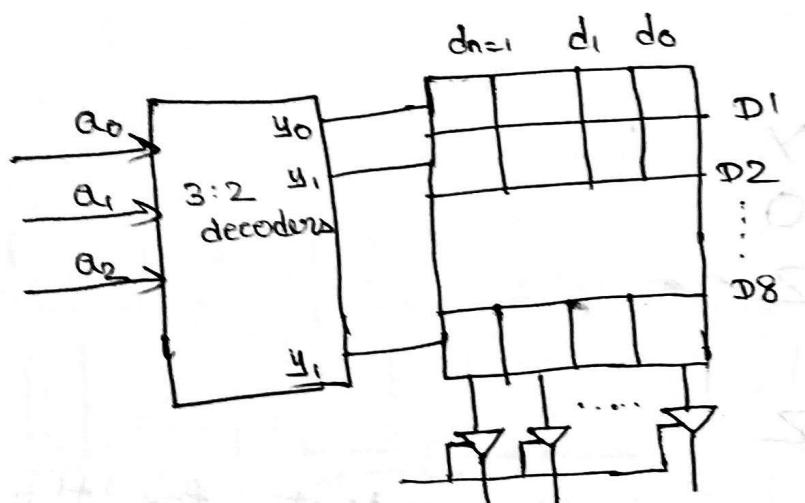
→ A demux does the opposite of what a mux does.



— Symbolic representation of demux.
 $(1:2^n)$.

I	S ₀	S ₁	O ₀	O ₁	O ₂	O ₃
X	0	0	X	0	0	0
X	0	1	0	X	0	0
X	1	0	0	0	X	0
X	1	1	0	0	0	X

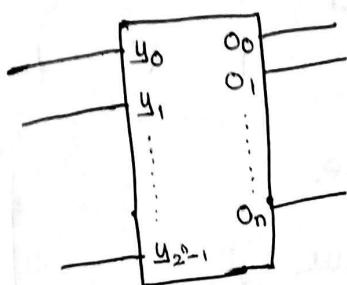
Where X is either 0 or 1.



ENCODER

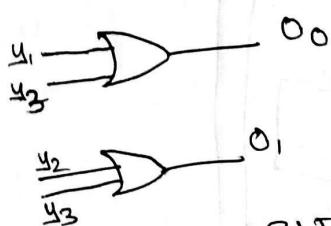
$2^n : n$

y_3	y_2	y_1	y_0	o_1	o_2
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1



$2^n : n$ encoder

→ Encoder does the opposite of what decoder does.



PRIORITY ENCODER

y_3	y_2	y_1	y_0	o_1	o_0	Z
0	0	0	0	1	1	0
0	0	0	1	I ₀	0	1
0	0	1	X	I ₁	1	1
0	0	X	X	I ₂	1	1
0	1	X	X	I ₃	1	1
1	X	X	X			

$$I_0 = \overline{y_3} \overline{y_2} \overline{y_1} y_0$$

$$I_1 = \overline{y_3} \overline{y_2} y_1$$

$$O_0 = I_3 + I_1$$

$$I_2 = \overline{y_3} y_2$$

$$I_3 = y_3$$

$$O_1 = I_2 + I_3$$

$$Z = I_1 + I_0 + I_2 + I_3$$

B_3	B_2	B_1	B_0	G_{13}	G_{12}	G_{11}	G_{10}
0	0	0	0	0	0	0	0
0	0	0	1	0	0	1	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	0	1	1	0	1	1	0
0	1	0	0	0	0	1	1
0	1	0	1	0	0	0	1
0	1	1	0	0	0	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1

COMPARATOR

→ If $(a_i \oplus b_i)$ is true for

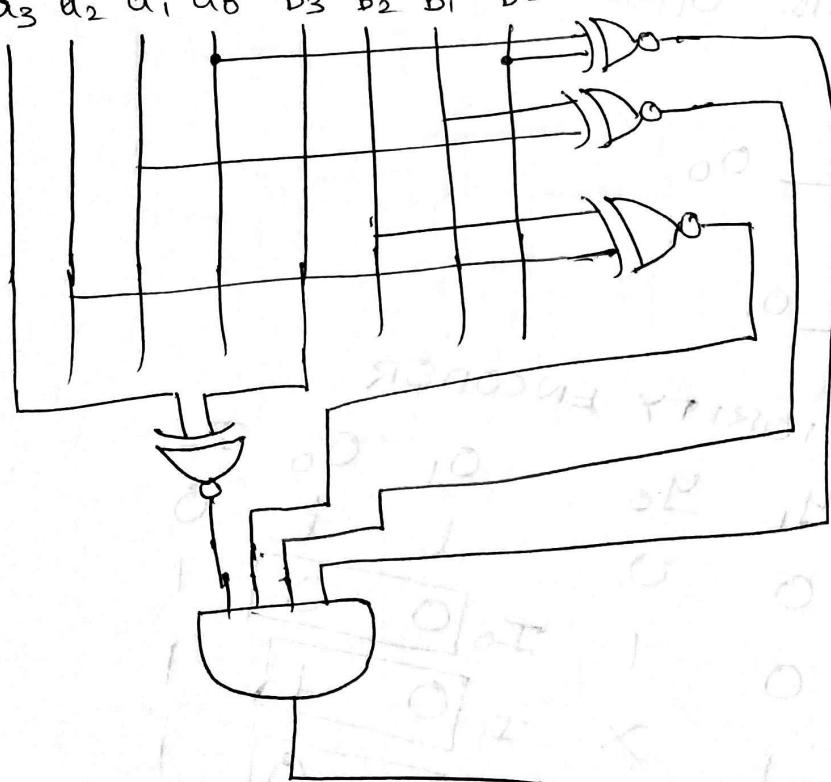
all i , then $A = B$.

$B_3 B_2 G_3$	00	01	11	10
$B_3 B_2$	00	01	10	11
00	0	0	0	0
01	0	0	0	0
11	X	X	X	X
10	1	1	X	X

$$G_{13} = B_3$$

$$G_{12} = B_2 \oplus B_1$$

$$G_{11} = B_3 \oplus B_1$$



→ if $A > B$

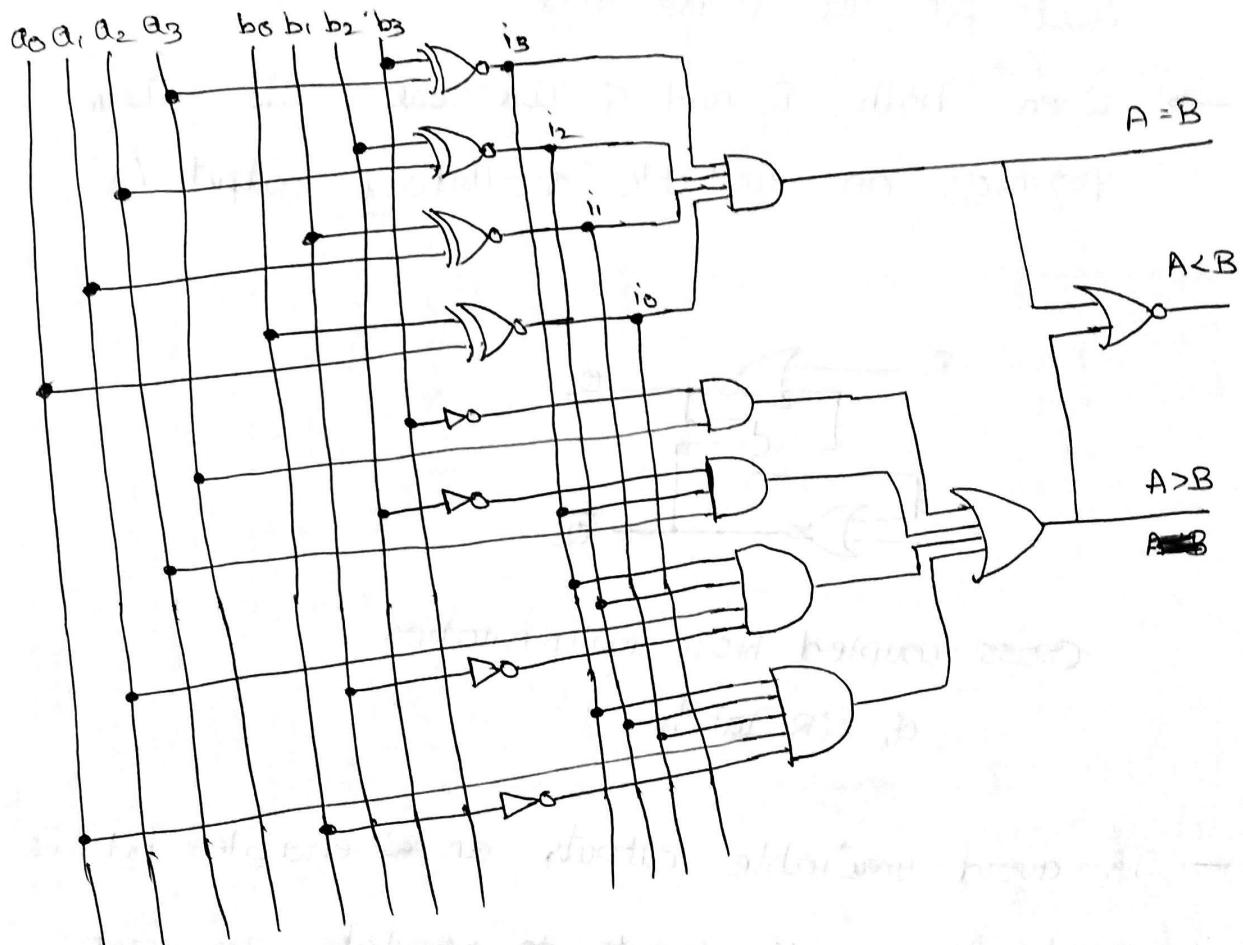
$$A > B = \overline{a_3 b_3} + \overline{a_2 b_2} i_3 + i_3 i_2 \overline{a_1 b_1} + i_3 i_2 i_1 \overline{a_0 b_0}$$

where $i_j = (\overline{a_j \oplus b_j})$

→ if $A < B$

$$A < B = \overline{(A = B + A > B)}$$

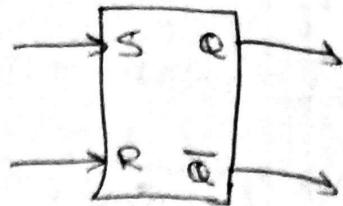
The complete implementation is:-



→ A circuit that stores a single bit is known as a latch.

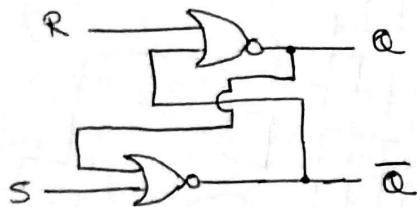
→ Eg:- Alarm clocks (~~SR~~ SR latch).

S	R	Q
0	0	NC
1	0	1
0	1	0
1	1	X



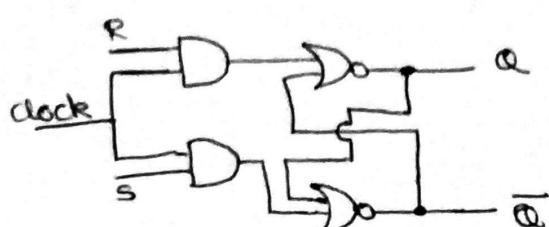
NC - NO Change

- Where S represents set and R represents Reset.
- Both set and Reset bits cannot be on at the same time as a clock can't be set and reset at the same time.
- When both R and S are one, the alarm produces an unstable oscillatory output (X).



Cross coupled NOR implementation
of SR latch.

- To avoid unreliable outputs an enable bit is used along with input to regulate the times where the output is to be read and the times where the input is to be feeded.



Gated SR latch.

→ The clock periodically inverts the bit.

தினங்கள்

→ The input which has to be given such that we obtain the required outputs are found out by using the excitation table.

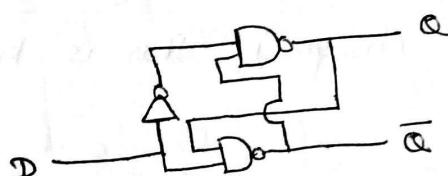
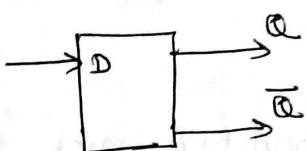
பெஸ்வாஷ்
45 ந்
வியாழன்
பெவர்ஸி
சனி
நாயிரு

EXCITATION TABLE FOR SR LATCH

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

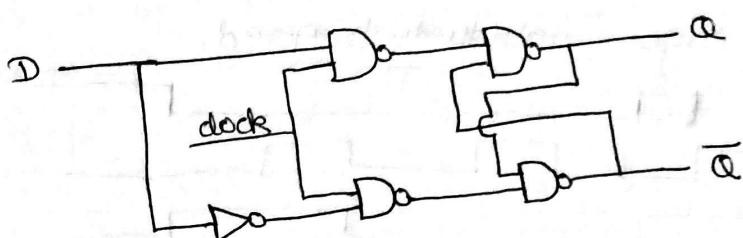
where $Q(t)$ is the first output $Q(t+1)$ is the next output required and S,R are the inputs

D-LATCH



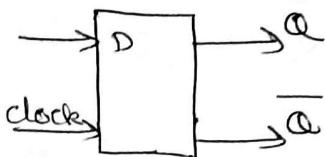
Truth Table	
D	$Q(t)$
0	0
1	1

clock	D	$Q(t+1)$
0	1/0	NC/D
1	0	0
1	1	1



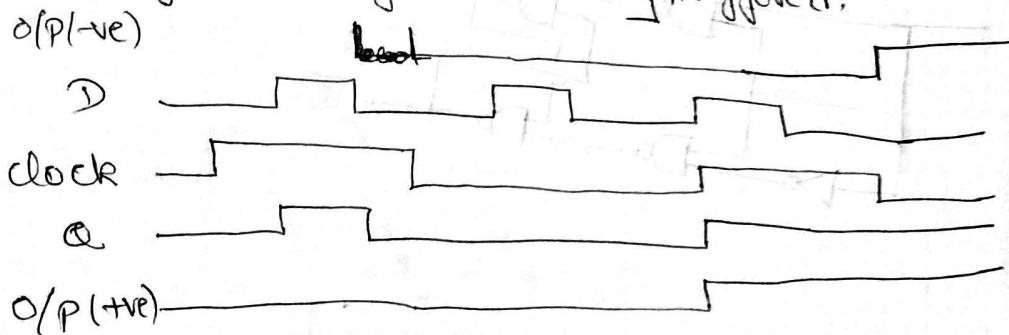
EXCITATION TABLE FOR D-LATCH

clock	$Q(t)$	$Q(t+1)$	D
0	x	x	x
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Gated D-latch

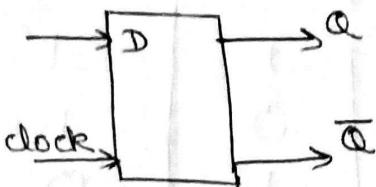
- Clock is used in a circuit so that the output varies only when the clock is high. This is known as level sensitivity.
 - The circuit can also be modified such that the output is varied only at the edges (ie. when the value of clock changes). This is known as edge sensitivity.
 - When the clock moves from 0 to 1 it is known as positive edge triggered and when the clock moves from 1 to 0 it is known as negative edge triggered.



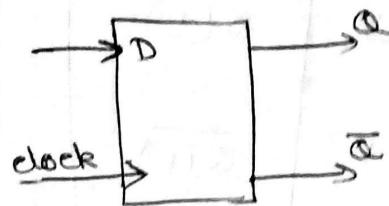
~~O/P (out)~~

edge sensitive

→ These kind of latches are known as flip-flops.

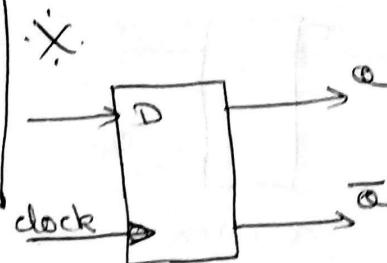


level triggered
D-latch



positive edge triggered
D-latch

- All latches are level sensitive
- ✗ and all flip-flops are
- ✗ edge-sensitive.



- In actuality the transistors take some time to change value from 0 to 1 and vice versa.

- So, a small amount of time is given as a buffer before and ~~and~~ after the clock's value changes known as setup time and hold time respectively.

- The buffer time is required to prevent misreading the value of clock when its' value is in transition.

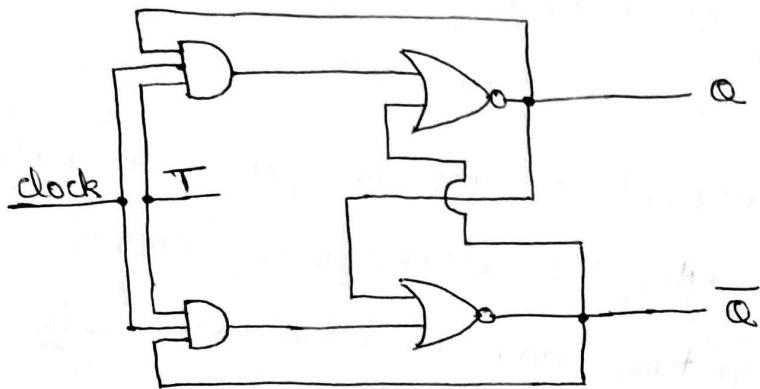
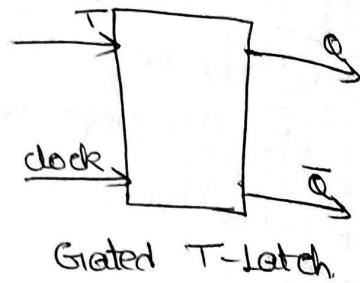
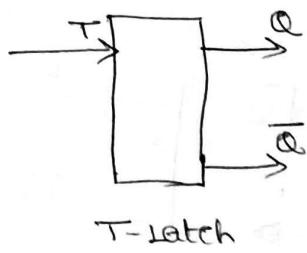
T-LATCH

- The difference between D-latch at T-latch is that, D-latch's output is the same as its' input whereas T-latch maintains its previous state if Input is 0 and the output is reversed if input is 1.

Truth Table

T	$Q(t+1)$
0	$Q(t)$
1	$\overline{Q(t)}$

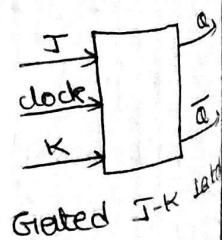
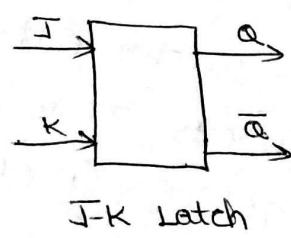
$Q(t)$	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

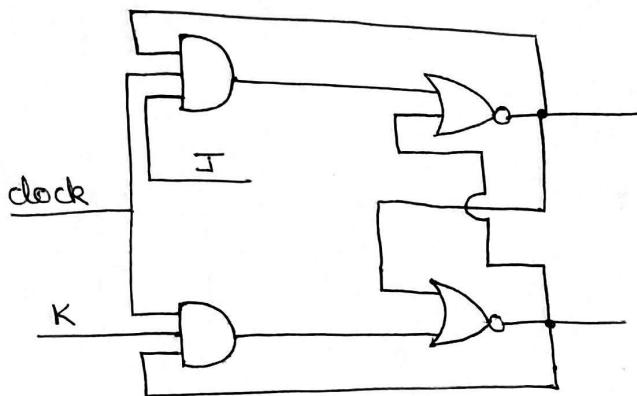


J-K LATCH

→ The disadvantage of SR latch is that it becomes unusable if set and reset are left the same time.

T	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$\overline{Q(t)}$





~~* → Flip-flops have clocks whereas latches don't *~~

→ is represented in clock's column for positive edge triggered.

→ is represented in clock's column for negative edge triggered.



clock	$Z(t)$	$Z(t+1)$	T
↑	0	0	0
↑	0	1	1
↑	1	0	1
↑	1	1	0

$Z(t)$	$Z(t+1)$	T	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0