DAY 12 – 100 DAYS VERIFICATION CHALLENGE

Topic: **Flip Flop & Latches**

Homework: Deep dive into Flip Flop & Latches

DAY 12 CHALLENGE:

1. Explain functioning of JK & SR Flip Flop
2. Difference between Flip Flop & Latch
3. Why are latches faster than flip-flops?
4. Explain the use of:
    - Flip Flop
    - Latch
5. Why is the Gated SR Flip Flop called Asynchronous Latch?
6. Implement D-FF using NAND Gate
7. Design D-FF using 2:1 MUX.

Copyright @Sweety Pinjani

Adityagi_ digital Electronic C
#Hoodaye Verification

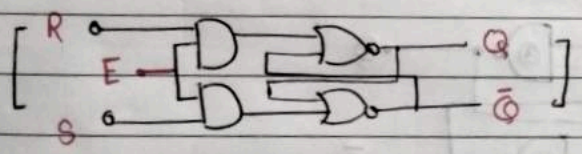
DAY : (12)

Topic : Flip-flop & LATCHES

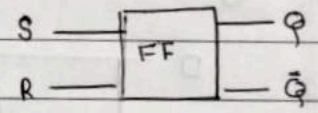
Sol^n ① Explain functioning of JK & SR Flip-flop.
Sol^n ② difference Between Flip-flop & latches
Sol^n ③ why are latches faster than flip-flops ?
Sol^n ④ Explain the use of : flip flop . latch.
Sol^n ⑤ why the Gated SR Flip-flop called Asynchronous latch ?
Sol^n ⑥ Implement D-FF using NAND gate
Sol^n ⑦ Design D-FF using 2:1 MUX.

Sol^n ⑤ why the Gated SR Flip-flop called Asynchronous latch ?

: Gated S-R flip flops are called asynchronous because the output responds immediately to input change.



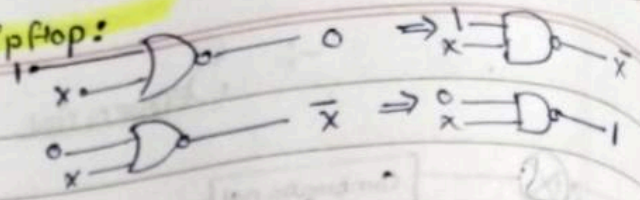
:- Gated SR latch



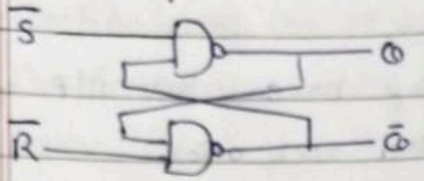
unclock SR-ff called latches

- latches that can change its state instantaneously on the application of it required inputs condition are known as asynchronous latches.
- That means the output can only act upon when the latches is enabled otherwise there will be no change in output state even required inputs are applied. in other words,
- The latch is active when enable signal is HIGH & its inactive when ENABLE is LOW . This HIGH LOW Enable signal is applied to the gated latch in form of clocked pulses. so gated SR latch as called . clocked SR FLIPFlop or Synchronous SR latch.
- Unclocked flip-flop called Asynchronous latch because clock signal is absent

Sol^n ① Explain function of JK Flip & SR Flipflop:



• **SR USING NAND LATCH**



∴ IN NAND, enable = 1 and disable = 0.
NOR Enable = 0 disable = 1

→ also called Active low SR latch
→ then we chang Q and $\bar{Q}$ Position

TRUTH TABLE :-

S	R	Q	
0	0	invalid ($Q = \bar{Q} = 1$)	} oscillation
0	1	1	→ Set
1	0	0	→ Reset
1	1	previous state (no change)	→ Hold

→ SR latch is used to eliminate Switch bouncing.
- Bouncing means vibration of switching when ON or OFF.
→ SR latch low → latch remain in it present state (no change)

S	R	Q_n	$\bar{Q}_n$	
0	0	x	x	Prohibited [disabled]
0	1	1	0	→ Set
1	0	0	1	→ Reset
1	1	Q_n	$\bar{Q}_n$	→ Hold

→ For SR latch using NOR, we cannot apply both input as 11.
It is suffered as invalid input which lead to osilation.
→ using SR Flip is voilation and more macanism.

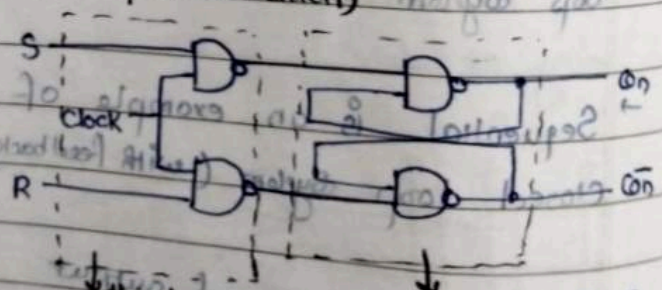
NAND:
enable = 1
disable = 0

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

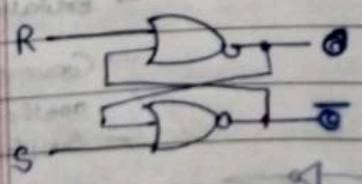
→ In SR latch if both gates are enable o/p remain same previous state and both are disable the o/p remain same invalid state

⊙ NAND GATE is Compliment of NOR.

⊙ **S-R Flip Flop** :- OR clocked SR FF.
→ is called Set-Reset Flip-Flop.
→ SR latch with enable or gated SR latch
→ (USING NAND LATCH)

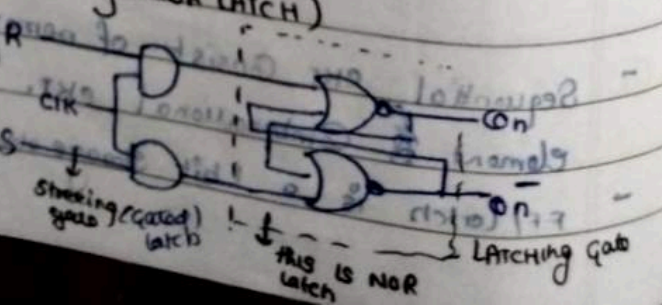


• **SR USING NOR LATCH** Active high



S	R	Q_n	$\bar{Q}_n$	
0	0	Q_n	$\bar{Q}_n$	"Hold"
0	1	0	1	OFF / Reset
1	0	1	0	ON / Reset
1	1	x	x	invalid

→ (USING NOR LATCH)



→ **TRUTH TABLE :-**

Truth Table is same for both NOR and NAND implementation shown in.

Clock	S	R	Qn+1	
0	X (any value)	X (any value)	Qn	Previous State — Hold State
1	0	0	Qn	
1	0	1	0	→ Reset
1	1	0	1	→ Set
1	1	1	X	→ Invalid OR unused

SR	Qn+1
00	Qn
01	0
10	1
11	Invalid

← important

- **CHARACTERISTIC TABLE :-**

in this table we determine upcoming output | Next output for present input and Past output.

S R (Present input)	Qn (Past output)	Qn+1 (Next output)
0 0	0	0
0 0	1	1
0 1	0	0
0 1	1	0
1 0	0	1
1 0	1	1
1 1	0	X
1 1	1	X

S \ Qn	Qn	Qn	Qn	Qn
0				
1	1	1	X	X

$Q_{n+1} = S + Q_n\bar{R}$

∴ Since S=1 R=1, the o/p is invalid because S·R = 1 not satisfy the above condition.

- **EXCITATION TABLE :-**

Qn	Qn+1	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

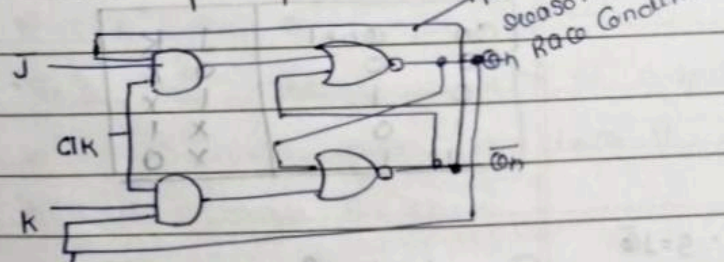
→ in SR, if $\boxed{Q=0}$ the o/p is Reset & set $\boxed{Q=1}$

→ disadvantage of SR FF is invalid state present when S=1, R=1

→ To avoid this JK FF is used.

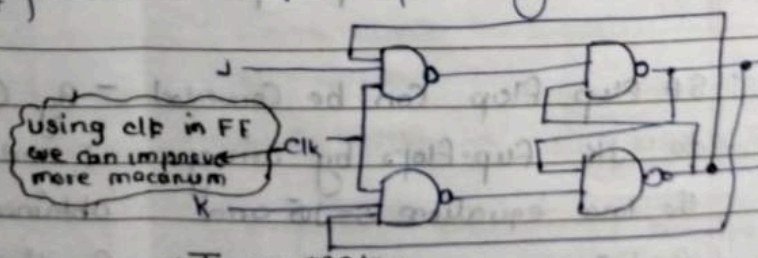
→ resetting means storing 0 and 1 bit.

② **JK Flip Flop :-**



This JK Flip-Flop using NOR GATE

JK Flip-Flop using NAND GATE :-



→ **TRUTH TABLE :**

Clock	J	K	Qn+1	
0	X	X	Qn	Hold
1	0	0	Qn	Hold
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	$\overline{Qn}$	Toggle

Proof :-

$Q_{n+1} = S + Q_n\bar{R}$

$J\overline{Q_n} + Q_n(\overline{K \cdot Q_n})$

$J\overline{Q_n} + Q_n(\overline{K} + \overline{Q_n})$

$= \boxed{J\overline{Q_n} + \overline{K}Q_n}$

Characteristic Table :-

J	K	Q_n	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

(iii) Logical expression minimization →

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

$$Q_{n+1} = Q_n\bar{K} + J\bar{Q}_n$$

$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

(iv) Exitation Table :-

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

once during the clock Pulse.

→ The Race around Problem in JK flip-flop can be eliminated by using edge triggered flip-flop or master slave JK flip-flop or by using the clock signal whose pulse width is less than or equal to the propagation delay of flip flop.

→ Master-slave Flip-flop is cascading of two JK Flip-flops Positive or direct clock pulses are applied to master and these are inverted and applied to the slave flip-flop.

→ difference Between latch and flip-flop :-

	Latch	Flip-flop
i.)	it checks the inputs but changes the output only at times defined by the clock signal or any other control signal.	i.) It checks the inputs continously and responds to the changes in inputs immediatly.
ii.)	it is a edge triggered device.	ii.) it is a level triggered device.
iii.)	Flip-flop always have a clock signal.	iii.) There is no such classification in latches
iv.)	flip-flop can be build from latches. Ex:- D Flip-flop, JK flip-flop.	iv.) latches can be build from gates. ex:- SR-LATCH, D-LATCH

⊙ Explain the use of : ⊙ Flip-flop ⊙ LATCH

:- use cases of flip flop and where latches are Beneficial :

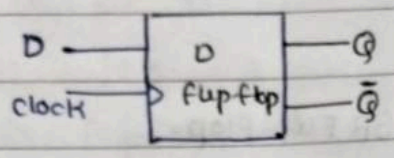
flipflop have a clock input that allows them to change state only when a clock signal is present. This makes flip-flop useful in synchronous circuits.

- Latches are single bit storage elements which are widely used in computing as well as data storage
- latches are used in the circuits like power gating & clock as a storage device.
- Latches is used on clock gating gate enable path to avoid glitches.
- and widely used to store the data and the codes for computations.

Sol^n ⊛

Sol ⑥ implement D-flip flop using NAND Gate.

D-flip flop:



⊙ TRUTH TABLE OF NAND gates

A	B	$\overline{AB}$
0	0	1
0	1	1
1	0	1
1	1	0

⊙ TRUTH Table D-flip-flop

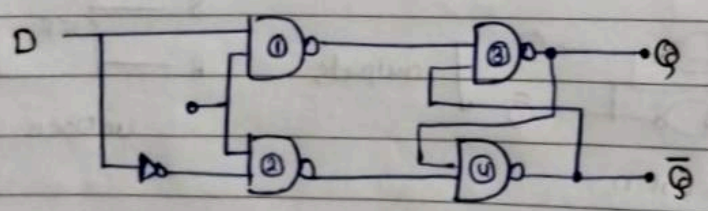
CLK	D	Q	$\bar{Q}$
1	0	0	1
1	1	1	0

⊙ Excitation table of D flip flop

Q(t)	Q(t+1)	D
0	0	0
0	1	1
1	0	0
1	1	1

⇒ characteristic equation : $\boxed{Q(t+1) = D}$

⊙ Logic circuit D-flip using NAND gate



Sol⁰ ⑦ Design D-flip-flop using 2:1 mux.

- D-flip flop, which stores a single bit of information and has a clock input to control when the data is stored.
- D-flip-flop has two inputs: D(data) input & clock input. The D input represents the data to be stored & the clck input determines when the data is latched and stored.

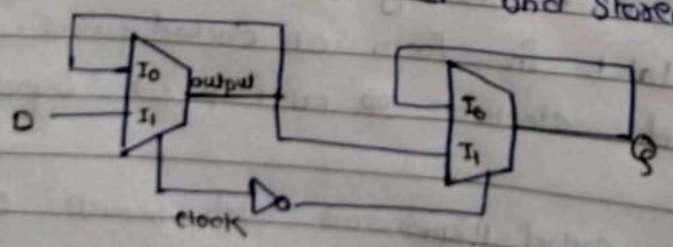


fig: D-flip-flop using multiplexer.

DAY 13 – 100 DAYS VERIFICATION CHALLENGE

Topic: **Logic Gates**

Homework: Understand about Logic Gates in depth

DAY 13 CHALLENGE:

1. Which are basic logic gates. Explain working with truth table.
2. Why are NAND & NOR called the Universal Gates? What are advantages of universal gates.
3. Design 4:1 MUX using universal gates:
    a. NAND
    b. NOR
4. What are applications of these gates:
    a. NAND
    b. NOR
5. Design OR, AND, XOR, NOT using:
    a. NAND
    b. NOR
6. Design below logic using Logic Gates
    a. Inverter
    b. Full adder
7. What happens if more than one input is applied to a logic gate?

Copyright @Sweety Pinjani

Aditiyagi_digital
#100DAYSVERIFICATION

classmate
Date
Page

DAY : 13

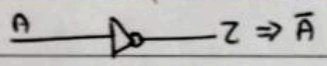
Topic : LOGIC GATES

Sol ① which are basic logic gates. Explain working with TRUTH TABLE.

:- AND, OR, NOT gates are basic gates.

- The logic gates which are derived from the Basic gates like AND, OR, NOT gates are known as derived gates. NAND, NOR, XOR & XNOR are derived gates.
- A universal gate is gate which can be implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates.

① NOT GATE : output of NOT gate is an invert of the input.

- Symbol : 

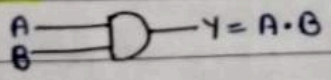
- TRUTH TABLE :

input	output
0	1 (High)
1	0 (low)

output Equation :

$$\boxed{Y = \bar{A}}$$

② AND GATE :

- Symbol : 

- TRUTH TABLE

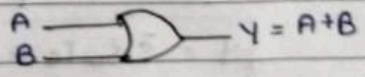
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

- output Equation :

$$\boxed{Y = A \cdot B}$$

⊙ for AND gate: The output is high only when both the input are high.

③ OR GATE :

- Symbol 

- TRUTH TABLE :

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

output Equation : $\boxed{Y = A + B}$

⊙ for OR : The output is low only when both the input are low in OR GATE.

Sol ② why are NAND & NOR called the universal gates? what are advantages of universal gates.

∴ universal gates is a gate which can implement any Boolean function without need to use any other gate type. The NAND and NOR gates are universal gates.

- NAND & NOR are called universal gates because all the other gates like, and, or, not, XOR, XNOR gate can be derived from it.

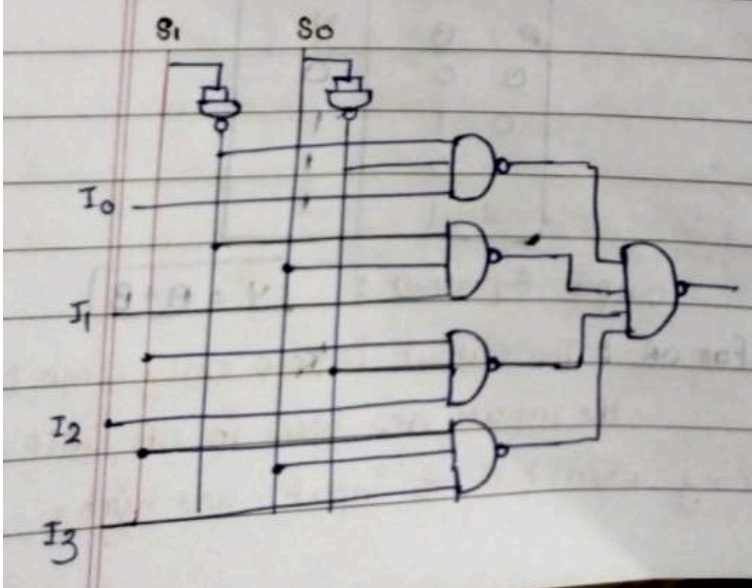
- main advantage of universal gates is that we can be used to made any kind of logic gate using universal gates. in practice this is advantageous since NAND & NOR gates are economical and easier to fabricate and are the Basic gates used in all IC digital logic families. :

Sol ③ Design 4:1 mux using universal gates:
ⓐ NAND ⓑ NOR

ⓐ 4:1 mux using NAND gates:

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

$$\boxed{Y = \bar{S}_1\bar{S}_0 I_0 + \bar{S}_1 S_0 I_1 + S_1\bar{S}_0 I_2 S_1 S_0 I_3}$$

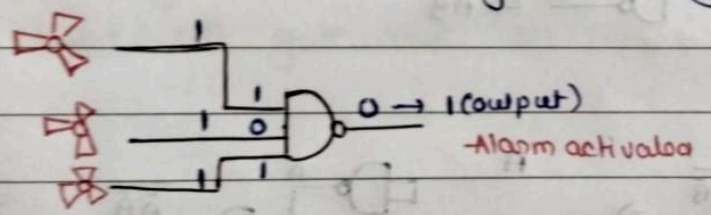


Sol^n ④ what are Application of these gates:

ⓐ NAND ⓑ NOR

- in Real life implementing with cmos technology you cant implement and you can only complement NAND and NOR gate then you can construct any other function using these two gate & inverter.

1.) Industrial Ware House Alarm System using NAND gate.



② Refrigerator Alarm System

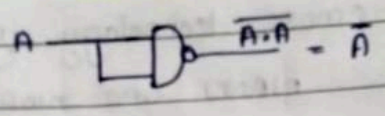
- NOR gate have numerous application in digital circuits:
- NOR gate used in Building digital System like calculators, computers & digital appliances.

Sol^n Design OR, AND, XOR, NOT using :

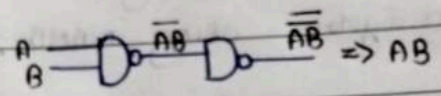
ⓐ NAND ⓑ NOR

- **Implomentation OF all gates using NAND**

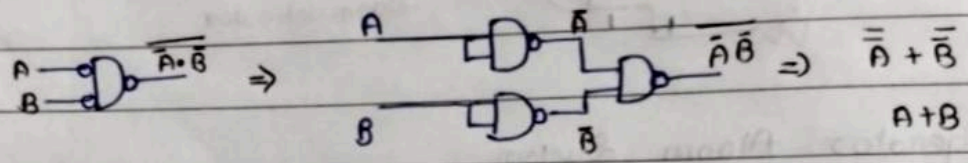
i.) NOT :-



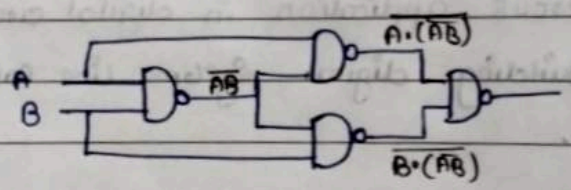
ii.) AND :



iii.) OR :



iv.) XOR :



$\overline{A \cdot (\overline{AB})} \cdot \overline{(B \cdot (\overline{AB})}$

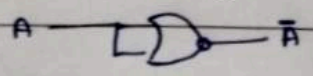
$A \cdot \overline{AB} + B \cdot \overline{AB}$

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

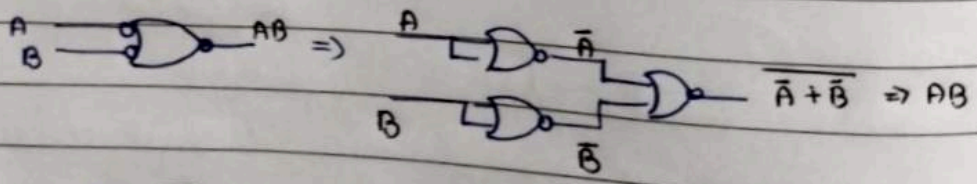
$(A\bar{B} + \bar{A}B)$

- **Implomont of all the gate using NOR:**

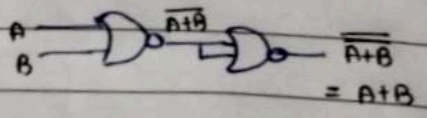
i.) NOT :



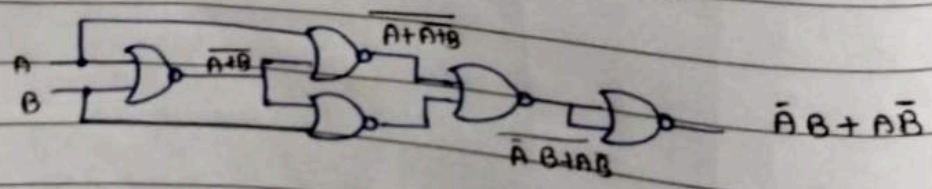
ii.) AND:



iii.) OR :



iv.) XOR :



Sol^n ⑦ what happens if more than one input is applied to a logic gate?

- adding more input terminals to a logic gate increase the number of input state possibilities.

- when multiple inputs are applied to a logic gate, the gate performs its specific logical operation on those inputs to generate an output according to its behaviour. The exact behaviour depends on the type of gate and the logical operation it performs.

DAY 15 – 100 DAYS VERIFICATION CHALLENGE

Topic: Encoders, Decoders.

Homework: Deep dive into encoders, decoders

DAY 15 CHALLENGE:

1. Design & explain working of:
    a. BCD to Decimal encoder
    b. Octal to Binary encoder using OR gates.
    c. 3:8 Decoder
    d. 4: 2 Priority Encoder
2. Explain difference between:
    a. Encoder & Decoder
    b. Encoder & Priority Encoder
3. Explain applications of:
    a. Priority encoder
    b. Priority Decoder
4. What are the basic functions of:
    a. Digital Encoder
    b. Digital Decoder

Copyright @Sweety Pinjani

-Nitiyogi
#100dayschallenge

classmate
Date
Page

DAY : 15

Topic : ENCODERS, DECODERS :

Sol^n ① Design & Explain working of :

ⓐ BCD To Decimal encoder :

	BCD A B C D	Output y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
0	0 0 0 0	1	0	0	0	0	0	0	0	0	0
1	0 0 0 1	0	1	0	0	0	0	0	0	0	0
2	0 0 1 0	0	0	1	0	0	0	0	0	0	0
3	0 0 1 1	0	0	0	1	0	0	0	0	0	0
4	0 1 0 0	0	0	0	0	1	0	0	0	0	0
5	0 1 0 1	0	0	0	0	0	1	0	0	0	0
6	0 1 1 0	0	0	0	0	0	0	1	0	0	0
7	0 1 1 1	0	0	0	0	0	0	0	1	0	0
8	1 0 0 0	0	0	0	0	0	0	0	0	1	0
9	1 0 0 1	0	0	0	0	0	0	0	0	0	1

→ The above decoder has 4 input lines for BCD code & 10 output line for deimal digit.

→ $y_0 = \bar{A}\bar{B}\bar{C}\bar{D}$

$y_1 = \bar{A}\bar{B}\bar{C}D$

$y_2 = \bar{A}\bar{B}C\bar{D}$

$y_3 = \bar{A}\bar{B}CD$

$y_4 = \bar{A}\bar{B}CD$

$y_5 = \bar{A}B\bar{C}\bar{D}$

$y_6 = \bar{A}BC\bar{D}$

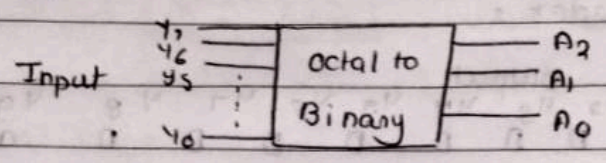
$y_7 = \bar{A}BCD$

$y_8 = A\bar{B}\bar{C}\bar{D}$

$y_9 = A\bar{B}\bar{C}D$

Sol ① (b) Octal to Binary Encoder using OR gates.

- 8 input line and 3 output lines. each input line corresponds to each octal digit & three outputs generate corresponding Binary code.



input, $n = 2^n$ output

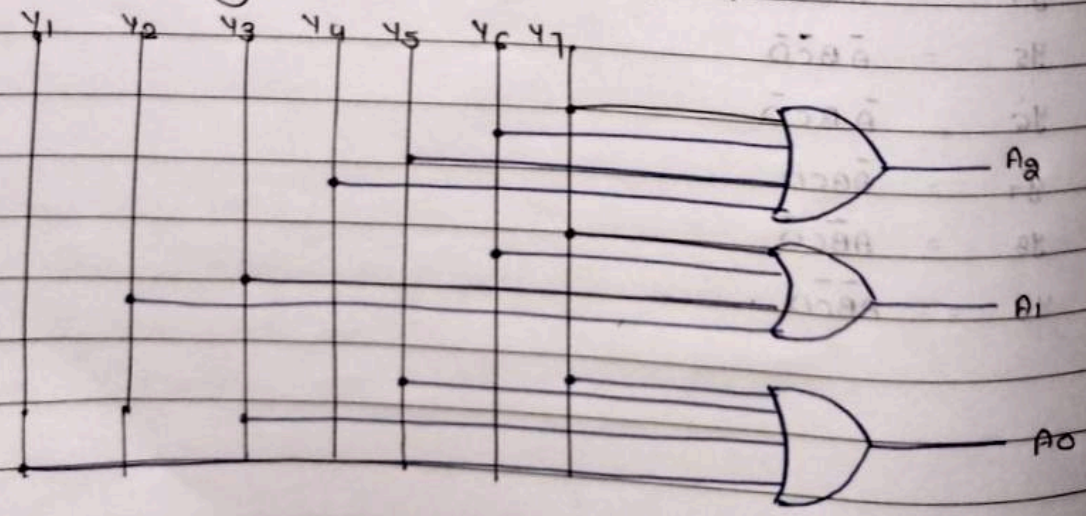
TRUTH TABLE FOR Octal to Binary Encoder (or 8 to 3 Encoder):-

input								outputs		
Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

→ Logical expression :-

$$A_2 = Y_7 + Y_6 + Y_5 + Y_4$$
$$A_1 = Y_7 + Y_6 + Y_3 + Y_2$$
$$A_0 = Y_7 + Y_5 + Y_3 + Y_1$$

→ implemented using four input OR GATE :

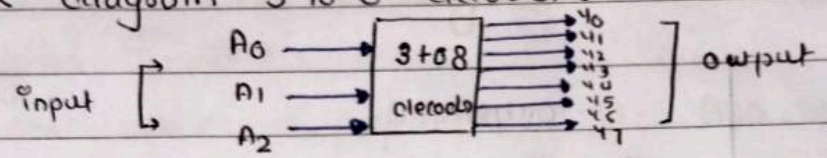


Sol^n ① © 3:8 Decoder :

: 3 to 8 line decoder is also known as Binary to octal Decoder.

- in 3 to 8 line decoder there is a total of eight outputs. Y_0 to Y_7 & 3 outputs. A_0 to A_2. The circuit has an Enable input 'E'.
- when Enable 'E' is set to 1. one of these four outputs will be 1.

:- Block diagram 3 to 8 decoder :-

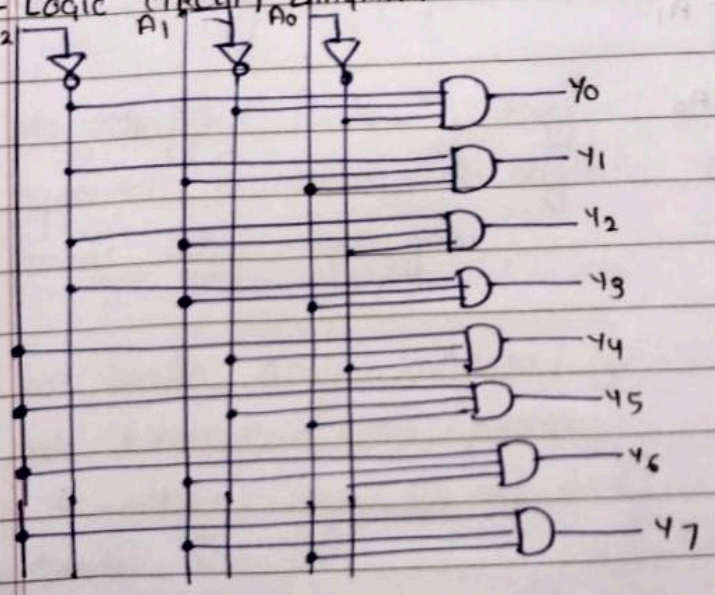


:- TRUTH TABLE :-

Enable	inputs			outputs							
E	A_2	A_1	A_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

:- Logic circuit DIAGRAM

:



:- logical expression of the Terms Y_0 to Y_7 :-

$Y_0 \Rightarrow \bar{A_0} \cdot \bar{A_1} \cdot \bar{A_2}$

$Y_1 \Rightarrow \bar{A_0} \cdot \bar{A_1} \cdot \bar{A_2}$

$Y_2 \Rightarrow \bar{A_0} \cdot A_1 \cdot \bar{A_2}$

$Y_3 \Rightarrow A_0 \cdot A_1 \cdot \bar{A_2}$

$Y_4 \Rightarrow \bar{A_0} \cdot \bar{A_1} \cdot A_2$

$Y_5 \Rightarrow A_0 \cdot \bar{A_1} \cdot A_2$

$Y_6 \Rightarrow \bar{A_0} \cdot A_1 \cdot A_2$

$Y_7 \Rightarrow A_0 \cdot A_1 \cdot A_2$

Date
Page

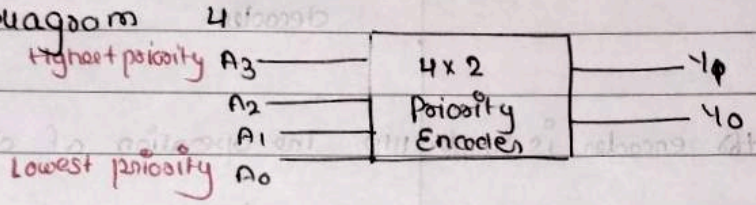
Sol^n ① (a) 4:2 PRIORITY ENCODER

→ Priority Encoder is a combinational logic circuit that can be take 2^n inputs & produced n outputs. &

→ represents the highest priority input among all the inputs.

→ when multiple inputs are active high at the same time then the input that has highest priority is considered first to generate the output.

: Block diagram



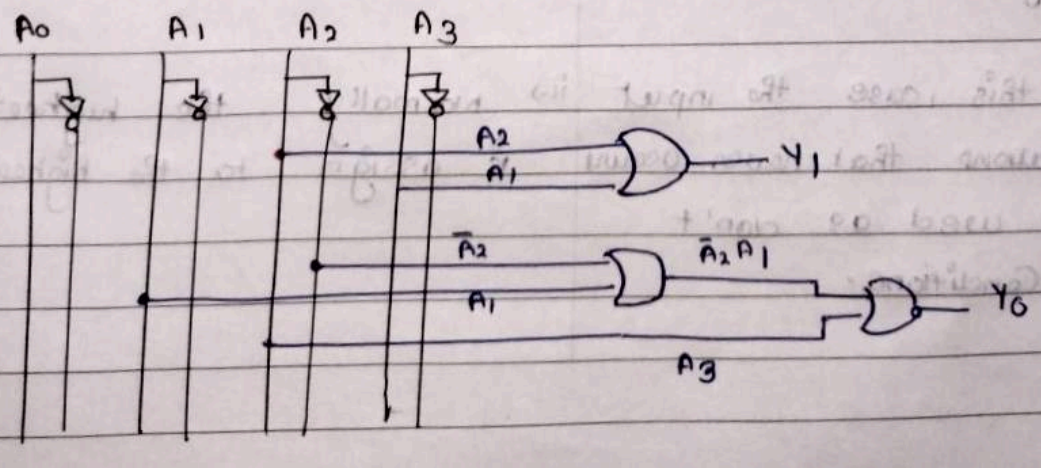
: TRUTH TABLE 4:2 PRIORITY Encoder

inputs				out	
A_3	A_2	A_1	A_0	Y_1	Y_0
0	0	0	0	X	X
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

: Boolean Expression:

$$Y_0 = A_3 + \bar{A}_2 A_1$$

$$Y_1 = A_2 + A_3$$



Sol^n (2) Explain difference Between :-

(a) Encoder & Decoder

(b) Encoder & Priority Encoder

(a) Encoder and decoder ?

Encoder	decoder
i.) Its a Combinal ckt that basically Converts the information signal to the coded digital bitstream.	i.) its also a Combinational ckt that Converts the Coded bits into original information. It performs the reverse operation of the decoder
ii.) There are 2^n input lines in the encoder.	ii.) There are n input lines in the decoder.
iii.) operation of the encoder is Simple.	iii.) The operation of decoder is complex.
iv.) Basic logic element that is used in encoders is OR Gates.	iv.) in decoder, The basic logic element is AND gates along with the NOT gates.
v.) Encoders are Commonly used in video, E-mail etc.	v.) Decoders are Commonly used in memory chips, microprocessors etc.

(b) Encoder & Priority Encoder :

Encoder	Priority Encoder
i.) An active encoder has a no. of input lines but only one of them to actuated at a given time.	i.) A priority encoder can have more than one input activated at the same time.
ii.) in this case the input Combinations that never occurs way be used as don't care Conditions.	ii.) Normally the highest priority is assign to the higher Subscript

Aditi_tyagi_digital

Sol^n ③ Explain application of :-

ⓐ Priority Encoder ⓑ Priority decoder

ⓐ Application : Priority Encoder :-

i. used to encode the analog to digital Converter output.

ii.) its used to reduce the no. of wires and Connection required for electronic ckt design that have multiple input lines. eg: Keypads & Keyboards.

iii.) Used in the detection of highest priority input in various applications of microprocessor interrupt Controllers.

iv.) it is used in robotic Vehicles.

ⓑ Application : Priority decoder :-

i.) address decoding

ii.) data multiplexing

iii.) priority detectors

iv.) code Conversion

Sol^n ④ What are Basic functions of :-

ⓐ digital Encoder ⓑ digital decoder

digital Encoder	digital decoder
i.) Convert parallel input into Serial codes.	i.) Convert an analogue Signal in to digital data.
ii.) Encoder is a Combinational ckt that performs the reverse operation of a decoder	ii.) digital decoder can take a binary input and Convert it into its decimal equivalent which is useful is various application :- error detection & Correction - Network routing etc.
- Convert code one form to another decrose the number of Signals	

DAY 16 – 100 DAYS VERIFICATION CHALLENGE

Topic: Counters, Timers

Homework: Deep dive into counters, timers

DAY 16 CHALLENGE:

1. Design & explain working of:
    a. 4-bit Ring counter
    b. 4-bit Johnson counter
    c. 3-bit Ripple counter
    d. Decade counter
2. List the difference between:
    a. Timers & Counters
    b. Synchronous & Asynchronous Counter
3. Explain working of 555 Timer IC

Copyright @Sweety Pinjani

Aati_tyagi_digital
#100daysverification

classmate
Date
Page

DAY : 16

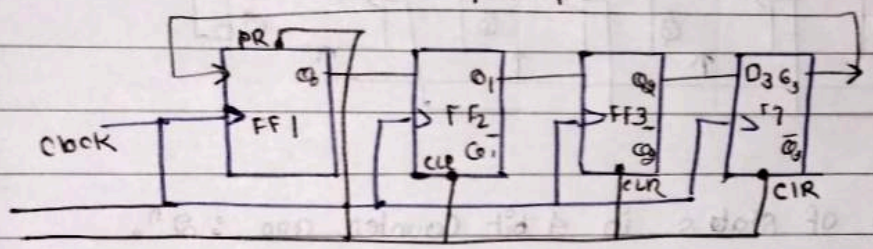
Topic : COUNTER'S , TIMERS.

Sol^n ① ⓐ **4-bit Ring COUNTER :-**

- Ring Counter is a typical application of Shift Register.
- The only change is the output of last flip flop is connected to the input of first flip-flop.

No. of states in Ring counter = No of flip-flop used

∴ we use 4 D-flip-flop.



- its a Synchronous Counter
- Also here we use Overridding input for each ffs.
- Overridding input is, Preset (PR) & clear (CLR) are used.

∴ when preset (PR) is 0, then the output is 1.

∴ when clear (CLR) is 0, then the output is 0.

- PR & CLR are active low signal that always work in value 0.

PR = 0 , Q = 1 ∴ These two value always
CLR = 0 , Q = 0 fixed.

ⓑ TRUTH TABLE : OF 4 Bit ring counter :

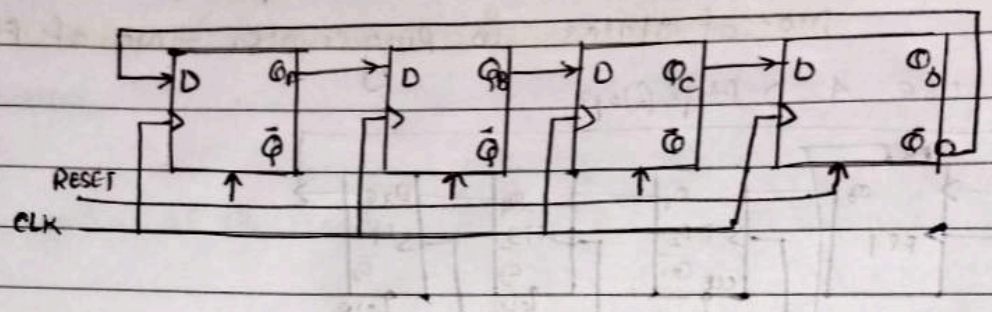
CLR/PR	clock	Q_0	Q_1	Q_2	Q_3
0	No clk	1	0	0	0
1	↓	0	1	0	0
1	↓	0	0	1	0
1	↓	0	0	0	1
1	↓	1	0	0	0

NOTE :- Can be implemented using D & JK FF. its a self-decoding ckt.

Sol^n ① ⓑ 4-bit Johnsons Counter :-

Johnsons Counter :- it also known as a twisted Ring Counter, Switch-tail Ring Counter. or walking ring Counter.

- The inverted output of the last flip-flop is connected to the input of first flip-flop.
- If we use n ff to design the Johnson Counter, its known as 2n bit Johnson counter or mod 2n Johnson Counter



∴ Total number of states in 4 bit Counter are : 2^n.

Total states in johnson Counter are : 2n.

unused states are : $2^n - 2n$ states.

- Johnson Couter Truth Table :-

State	Q_A	Q_B	Q_C	Q_D
1	0	0	0	0
2	1	0	0	0
3	1	1	0	0
4	1	1	1	0
5	1	1	1	1
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1

Working of Johnson Counter :

The Johnson Counter design with D FF. its has four state i.e four ff connected in series type or cascaded. initially zero is feed to the johnson counter & on apply clk signal, o/p will change to 1000, 1100, 1110, 1111, 0111, 0011, 0001, 000 in a sequence & the sequence will repeat for next clk signal.

⊙ Asynchronous Counter :-

Ⓐ Ripple Counter :-

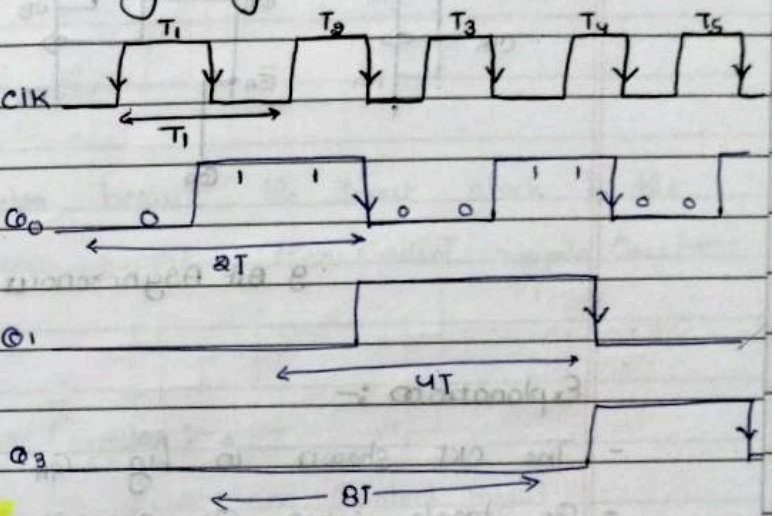
- It is a Asynchronous Counter.
- different FF used different clock pulse.
- for designing of Asynchro nous Counter only T and JK FF are used (Toggle mode).
- Only one FF is applied with external clk and other its clk is from previous ff output (whether Q_0 or $\bar{Q}_0$).
- The ff applied with external clk will act as LSB.

4	1	0	0	mod-8 up counter
5	1	0	1	
6	1	1	0	
7	1	1	1	
8	0	0	0	

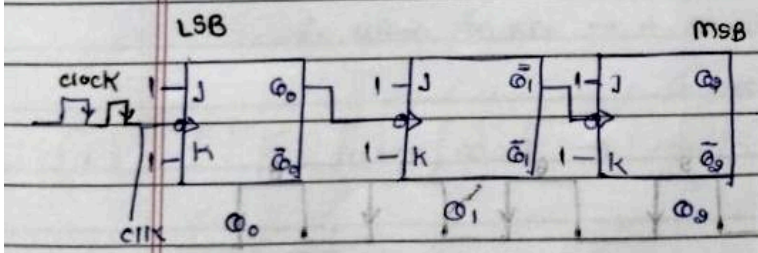
→ this is up counter.

- Its also called mod-8 ripple counter

⊙ Timing diagram -



- **3 bit ripple Counter : (UP Counter)**



→ The CKT shown in fig Q_0 toggle to every clock pulse

→ Q_n change when Q_{n-1} change from 1 → 0 ie. Q_2 changes when Q_1 changes from 1 – 0.

Truth Table :-

clk	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1

$T_{out} = 8 T_{IN}$

$$\boxed{F_{out} = \frac{F_{IN}}{N} = \frac{f_{IN}}{8}}$$

→ in n bit Ripple Counter propagation delay of each ff is t_{pdff} then time Period of clk is.

$$\boxed{\begin{aligned} T_{clk} &\geq n t_{pdff} \\ F_{clk} &\leq \frac{1}{n t_{pdff}} \end{aligned}}$$

- Ripple Counter is slower in operation Compared to Synchronous Counter

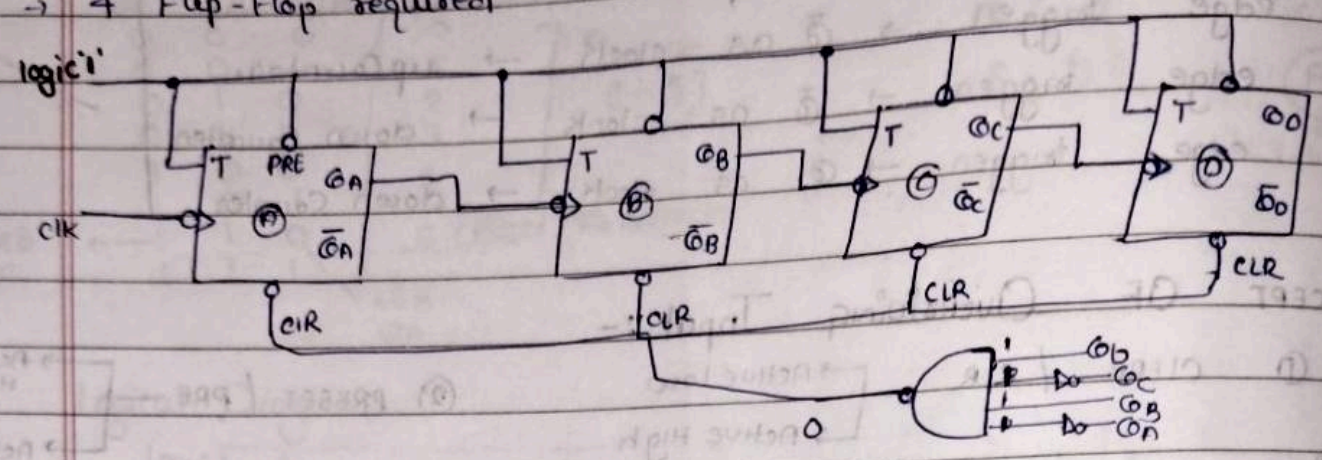
$[T_0 = n \cdot T_{pd}]$

n → no of ff used

T_{pd} → Propagation delay of each ff

MOD-10 Counter (Decade Counter) :-

→ 4 Flip-Flop required



→ The NAND gate output is zero when the count reaches 10 (1010)

- A 4 bit binary counter will act as decade counter.

→ Truth Table

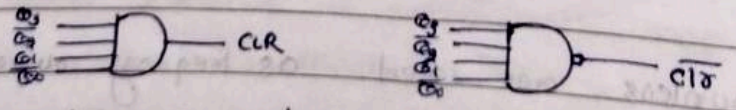
clock	Q_D	Q_C	Q_B	Q_A	
0	0	0	0	0	
1	0	0	0	1	
2	0	0	1	0	
3	0	0	1	1	
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	
10	1	0	1	0	(Q_3 $\bar{Q}_2$ Q_1 $\bar{Q}_0$)
0	0	0	0	Reset	

Application of BCD Counter or decade Counter :-

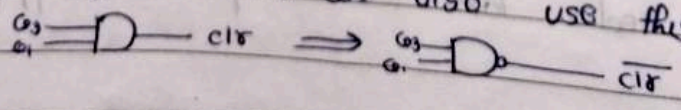
- Clock generator
- TTL Compatible input
- clock division
- in frequency Counting ckt.

→ duty cycle is 80 %.

→ Output frequency of BCD Counter is f/10.

NOTES → 

if is use only Q_3 and Q_1. we also use this as clear ckt.



Division:-

COUNTER

Counter:

- Counter are used to count the clock pulses. The clock pulses occur at regular intervals.
- Counter are Sequential circuit which change their predefined states with the help of clock pulses.
- Counter are Constructed Flip flops and logic gates.

→ There are two types:-

i) Asyncronous Counter

ii) Syncronous Counter

- in Asyncronous Counter, an external clock pulse is provided by first flip-flops.
- output of first flip-flop Connected of second flip-flop of a clock pulse.
- 3 Bit down Counter

∴ for 3 Bit counter, we require 3 flip-flop and

∴ we can generate $2^3 = 8$ state

⊙ Difference Between Syncronous and Asynchronous Counter?

- Synchronous Counter :-

i.) in Synchronous Counter we use a universal clock that is common to all ff. through out the CKT.

ii.) Syncronous Counter is faster in operation as Compared to Asynchronous Counter.

iii.) Syncronous Counter is also called parallel Counter.

iv.) in syncronous Counter propogation delay is less.

v.) Syncronous Counter will operate in any desired Count Sequence.

⊙ Asynchronous Counter :-

i.) in Asynchronous Counter main clock is only applied to the first-flip-flop and then for rest of flipflop the o/p of previous flipflop is taken as clock.

ii.) Asyncronous Counter is slower as Compared to Synchronous Counter in operation.

iii.) Asynchronous Counter is also called Serial Counter.

iv.) in Asynchronous Counter, there is high propogation delay.

v.) Asynchronous Counter will operate only in fixed Count sequence (up/down).

Sol^n ② List The difference Between:

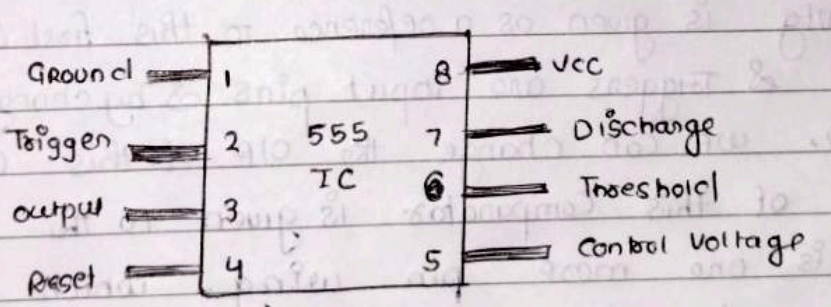
ⓐ Timers & Counters

i.> Timers	ii.> Counters
The register incremented for every machine cycle.	The register is incremented 1 to 0 transition at its corresponding to an external input pin.
ii.> Timers uses the frequency of the internal clk & generate delay.	ii.> A Counter uses an external signal to count pulses.
iii.> maximum count rate is 1/12 of the oscillator frequency	iii.> A Counter uses an external signal count pulses & maximum count rate is 1/24 of the oscillator frequency.

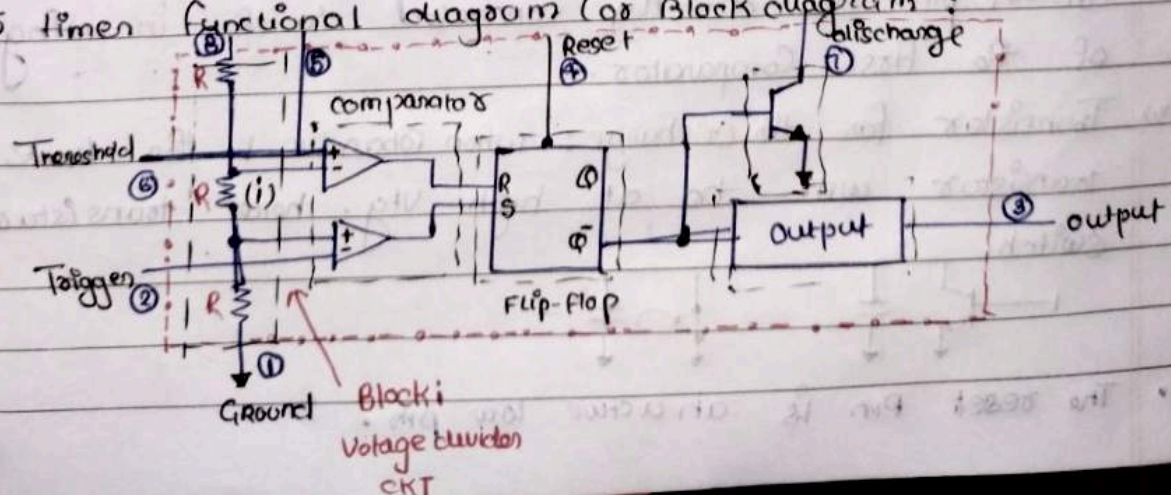
Sol^n ③ Explain working OF 555 timer IC?

∴ The 555 timer IC got its name from The three 5kΩ registers that are used in its voltage divider network.

- This IC is useful for generating accurate time delay & oscillation
- Pin Diagram:
- The 555 Timer IC is an 8 pin mini-Dual inline package (DIP).



555 timer functional diagram (or Block diagram):



∴ The purpose of pin number 1, 3 and 8 is pretty straightforward.
- pin number ⑧ is the supply voltage pin. & usually, The supply voltage in the range of 4.5 to 15V is applied at this pin.

- pin number ① is the ground pin. and its connected to the circuit GND.
- If you see pin number ③ is the output pin. So this pin is capable of driving the TTL loads & it can source or sink up to the amount of current.

i.) Voltage divider circuit: first Block of this 555 times IC is Vtg divider CKT. the volage at two nodes are going to the Comparator. & here we are assuming that the op-amp which is used as a

ii.) Comparator is an ideal op-amp. So no current is following in to the op-amp terminals. & here we assuming the no current is following into the op-amp terminals. Current is only flowing through these 3 Registers. the we directly apply Vtg divider Rule.

$$\text{Vtg at node} = \frac{2R \times VCC}{2R+R}$$

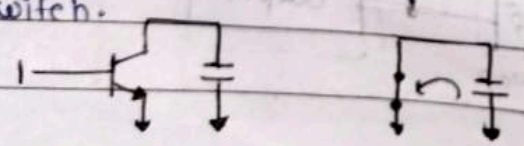
& this vtg is given as a reference to this first Comparator.

- Threshold & Triggers are input pins & by changing the Vtg at the two pins. we can change the o/p of this Comparator.

iii.) The o/p of this Comparator is given to the ff- CKT.

- there is one more pin using which we can also control the timing. that is pin number ⑤ This is called control pin. its directly connected to the inverting terminal of the first Comparator.

v.) Transistor for the Discharge: when Cobar is 1, the base of this transistor will be at high vtg. here transistor act like a switch.



- The reset Pin is an active low pin.

DAY 17 – 100 DAYS VERIFICATION CHALLENGE

Topic: Multiplexers, Demultiplexers

Homework: Let's learn about muxes & demuxes

DAY 17 CHALLENGE:

1. Design 16:1 MUX using 2:1 MUX
2. Design D-FF using MUX
3. Design a 1:2 DEMUX that can be used as an inverter & also as a buffer.
4. Design NAND Gate using 1:2 DEMUX
5. What is the difference between Decoder & DEMUX?
6. If D0 input of a 2:1 MUX is connected to ground, what is the output?
7. If D1 input of 2:1 MUX is connected to 1, what is the output?
8. List the applications of:
    a. Multiplexer
    b. Demultiplexer

Copyright @Sweety Pinjani

Aditi_tyagi_2023
#100daysverification

classmate
Date
Page

DAY: 17

Topic: Multiplexer's, Demultiplexers

Sol^n ① Design **16:1 mux using 2:1 mux:**

No. of selection lines required :-

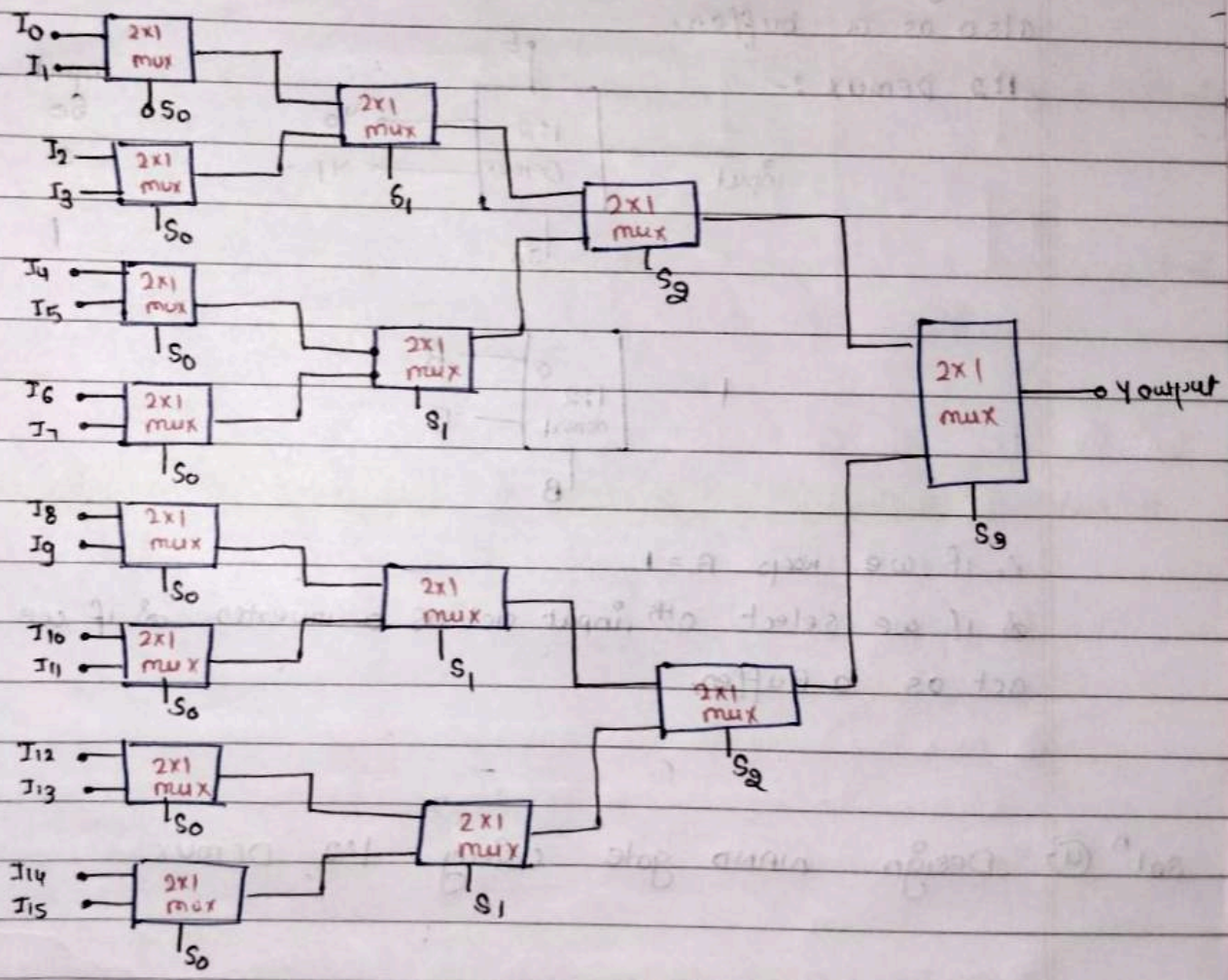
Input line $\leq 2^{\text{selection line}}$

$16 \leq 2^n$ $\quad 2^4 \leq 2^n$

$\boxed{n = 4 \text{ selection line}}$

∵ no of input line = 16

∵ Output line = 1



→ To implement $2^n \times 1$ mux using 2×1 mux, The total number of 2×1 mux required is $(2^n - 1)$.

∴ The number of 2×1 multiplexer required to implement 16×1 mux will be ; $n = 16 - 1 \Rightarrow 15$ mux required

1st stage $= 16/2 = 8$

2nd stage $= 8/2 = 4$

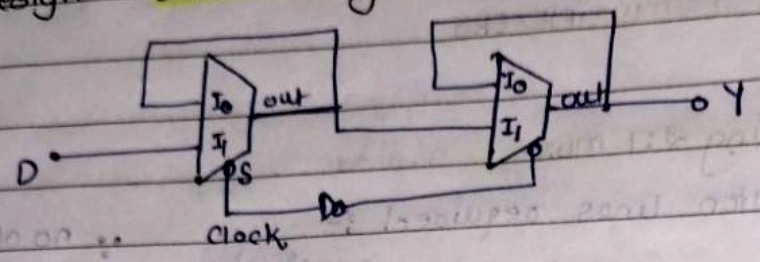
3rd stage $= 4/2 = 2$

4th stage $= 2/2 = 1$

$n = 8 + 4 + 2 + 1$

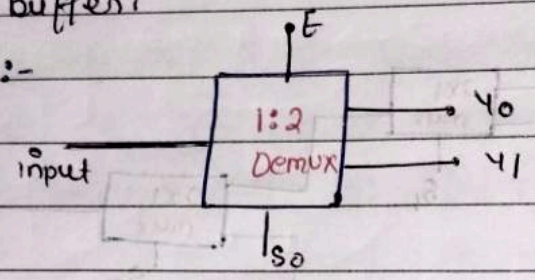
$= 15$ mux.

Sol^n ② Design D-ff using mux :-

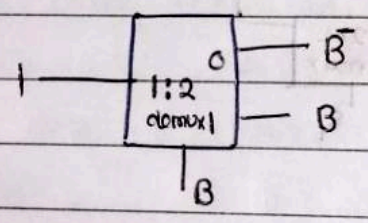


Sol^n ③ Design a 1:2 Demux that can be used as an inverter & also as a buffer.

1:2 Demux :-



Input	Outputs	
S_0	Y_0	Y_1
0	A	0
1	0	A



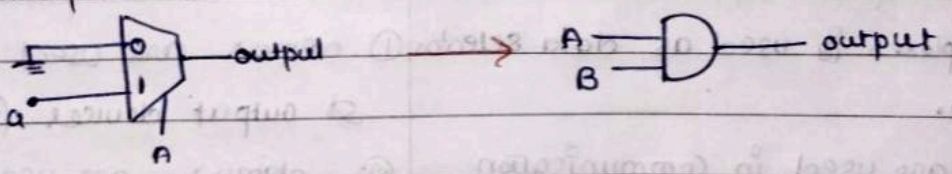
∴ if we keep A = 1

& if we select 0th input act as a inverter & if we select 1 input act as a Buffer.

Sol^n ⑤ what is difference Between decoder & DEMUX ?

decoder	demux
① A decoder decodes an encrypted input signal to multiple output signal from one format to another.	① A demultiplexer routes an input signal to multiple output signals.
② A decoder has input line and a maximum of 2^n output lines.	② A demultiplexer has single input selection lines and maximum of 2^n outputs.
③ decoders are used to detect bits encoding of data.	③ Demultiplexers are used in switching data, distribution.
④ decoders are heavily used in networking application.	④ demultiplexers are employed in communication systems.

Sol^n ⑥ if D_0 input of a 2:1 mux is connected to ground, what is the output ?



A	B	out
0	0	0
0	1	0
1	0	0
1	1	1

when A = 0, out = 0

when A = 1, out = B

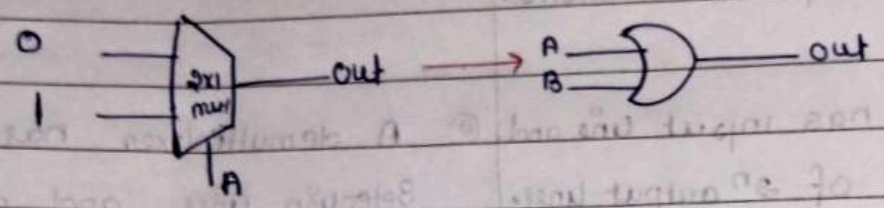
∴ AND gate will be implemented if we connect D_0 to 0 and D_1 to 'B'. This is also called AND gate using 2x1 mux.

∴ here we connect A to select pin of 2x1 mux.

Sol^n ⑦ if D_1 input of 2x1 mux is connected to 1, what is output

Sol^n

if we connect select pin of mux to A, Do pin of mux to 1 then it will



	A	B	output
0	0	0	0
	0	1	1
1	1	0	1
	1	1	1

when A = 0 out is B ← Condition

when A = 1 out = 1

∴ This implemtation of 2-input OR gate using 2x1 mux.

Sol^n ⑧ List the application of:

ⓐ multiplexer ⓑ demultiplexer

multiplexer application

① multiplexer is used as data selector device.

② mux are used in communication system to increse the efficency of the system

③ multiplexer are used in Satellite communication & GPS.

④ To maintain large amount of data, multiplexers are also used in computer memory system.

demultiplexer application

① demux are used in several input & output devices for data routing.

② demux are used in digital control system to select one signal from a mutual of signals.

③ demux can be used in Serial to parallel converts.

④ demux are used for broadcasting of ATM Packets.

DAY 18 – 100 DAYS VERIFICATION CHALLENGE

Topic: FSM, Sequence Detectors

Homework: In-depth knowledge on Finite State machines & sequence detectors

DAY 18 CHALLENGE:

1. What is Finite state machine? What are different types of FSMs?
2. What is the difference between Moore & Mealy machine?
3. What are applications & limitations of FSM?
4. How do you handle exceptions or error states in a Finite State Machine?
5. Design a Finite state machine for:
   i. serial two's complement block
   ii. detect a sequence 10110 (overlapping)
   iii. detect a sequence 11001100 (non-overlapping)
   iv. detect more than one "1"s in last 3 samples.
6. Explain below concepts in FSM:
   i. Trigger
   ii. Transition
   iii. State Diagram
   iv. Super state
   v. Guard conditions
   vi. Deadlock
   vii. Determinism
   viii. epsilon transitions

Copyright © Sweety Pinjani

Aditi_Tyagi_digital_2023

DAYS : 18

Topic : FSM . Sequence DETECTORS.

Sol^n ① what is Finite state machine ? what are different type OF FSMs?

∴ finite state machine can be viewed as an abstract representation of digital logic with limited or finite states.

- its a mathematical model of computation.
- FSM can defined by a list states, its initial state & the input that trigger each transition.
- The machine states are represented by a collection of state variables.

⊙ Finite State machine Types :-

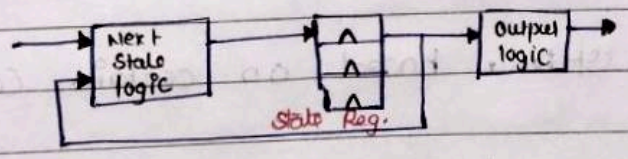
→ finite state machine can be classified :-

① mealay state machine

② moore state machine

① mealay state machine :-

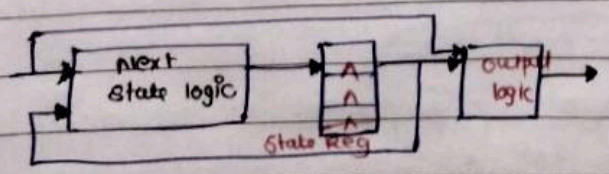
→ A finite state machine is said to be mealy state machine, if output depend on both present Inputs & present states.



→ in case of melay required n state.

② moore state machine

A finite state machine is said to be moore state machine, if outputs depends only on present states.



→ in case of moore required (n+1) states.

Sol^n ② what are application & limitation of FSM?

:- Application :-

① Software for desiging & checking the Behaviour of digital circuit

② lexical analyzer of a tipical Compiler.

③ Software for Scanning large bodies of text example web page pattern matching.

④ Software for verifying system of all type that have a finite no of states eg :- Stock market transaction, Communication N/w protocol.

◉ limitation of FSM :-

- The major limitation of an FSM is that it cannot be produced for an infinite sequence, nor can it multiply to arbitrary large binary numbers.
- It cannot rememeber state transitions & arbitrarily large amount of information.

Sol^n ④ How do you handle exceptions or error state in a finite state machine

(1) Define Error states.

② transition to error states, based on certain condition when the errors occurs.

③ Implement error handling logic like notifying the error or taking corrective actions to recover from error.

④ And from the error state based on corrective active, transition to appropriate state to proceed.

Sol^n ⑤ Design a finite state machine for:

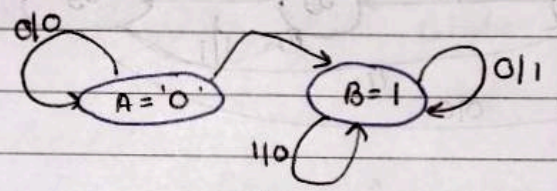
i.) Serial two's Complement block

- Design a one input, one output serial 2's complement. The circuit accepts a string of bits from the input & generate the 2's complement at the output
- The CKT can be synchronously to start & end the operation.

→ 2's Complement of a number can be obtained by keeping the least significant bits as such untill the first 1. & complemented all bits.

eg: 001010 ⟶ 110110

∴ State diagram: let A, B the two states representing the output states



Sol^n ⑤ ii.) detect a sequence 10110 (overlapping).

→ The stream of bit has been feed as input, when the clock is high and a particular pattern / sequence is detected.

→ as soon as sequence is detected the output becomes high & then again becomes low

Sequence = 10110 No of states = No of bits = 5

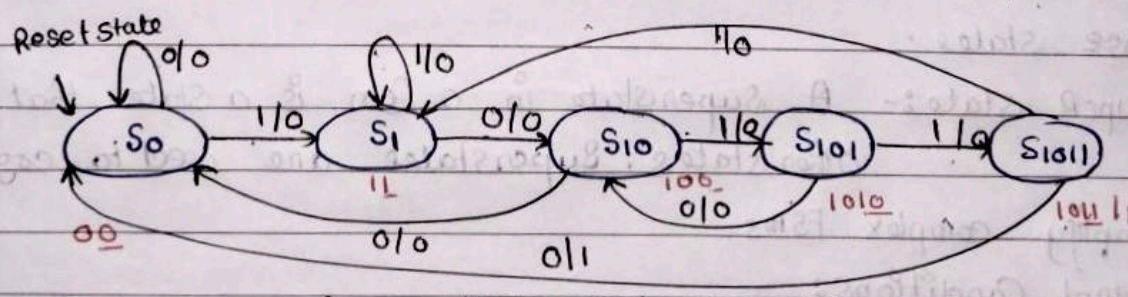


fig :- Non-over lapping 10110

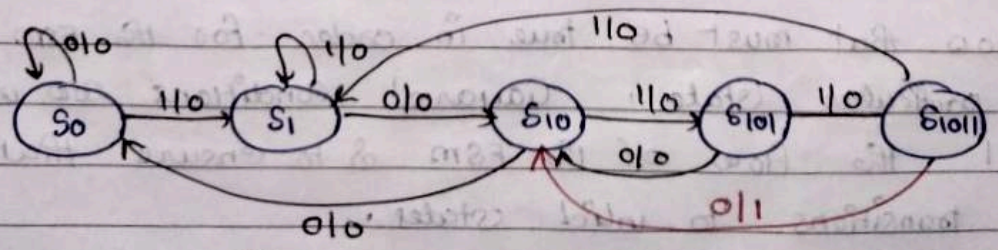
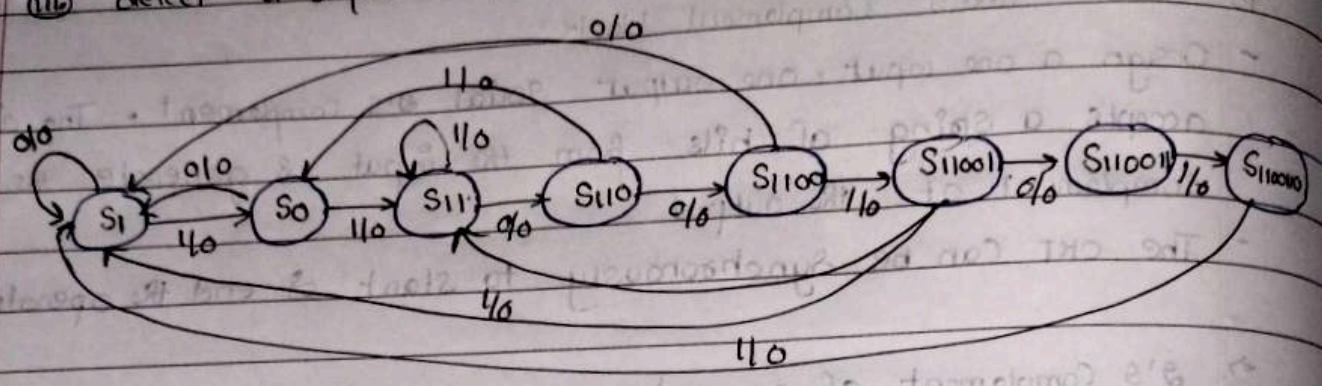
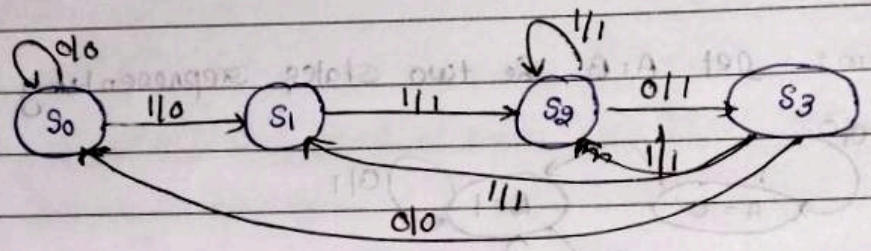


fig:- 10110 overlapping using mealy FSM

Sol^n ⑤ (iii) detect a Sequence 11001100 (non-overlapping)



Sol^n ⑤ iv) detect more than one "1"s in last 3 Samples.



Sol^n ⑥ Explain below Concepts in FSM:

① Trigger :- A Trigger in a finite state machine is an event that cause the FSM to transition from one state to another.

② Transition :- A transition in a FSM is a change from one state to another. Transition are triggered by events.

③ State diagram :- A state diagram in a FSM is a graphical representation of the states of the FSM & the transition btw those states.

④ SuperR state :- A superstate in a fsm is a state that contains other states. Superstates are used to organize & simplify complex FSMs.

⑤ Guard Conditions:

Gaurd Condition in a FSM is a boolean expression that must be true in order for the FSM to transition to a particular state. Gaurd conditions are used to control the flow of the FSM & to ensure that the FSM only transitions to valid states.

⑥ Deadlock :

A deadlock in a FSM is a Condition in which the FSM Cannot transition to any other state because all of the possible transition are Blocked.

- Deadlock Can occur in FSMs when there are multiple Concurrent process that are Competing for resources, or when there is a race Condition Btw two or more process.

⑦ Determinism :-

Determinism in a FSM is a Condition in which any given state and input, there is only one possible next State. This is non deterministic FSMs. where there may be multiple possible next states for a given state & input.

⑧ Epsilon Transitions:

An epsilon transition in a FSM is a transition that Can occur without Consuming any input. Epsilon transition are also known as Spontaneous transitions or null transitions.

FINITE STATE MACHINE (FSB)

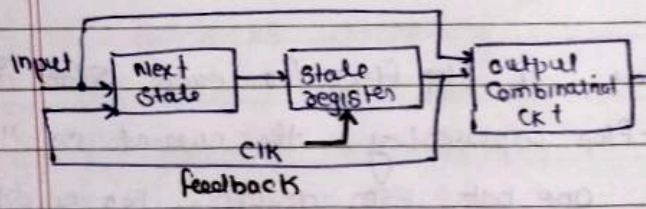
No. of states = finite

FSM
- moore
- mealay

→ moore, output is function of present state (it means value stored in the system)

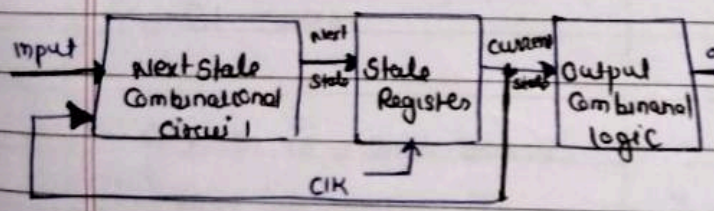
→ mealay, output is function of present state as well input.

→ mealy Circuit :-
its depends on present input and present state



its called melay circuit

→ MOORE Circuit :-
"its depend on present state.



→ moore machine has output that depen on the state only (thus FSM has the o/p written in the state itself)

→ Comparision of MEALY & MOORE CKT

MEALY CKT	MOORE CKT
1> O/P depend on present state and present I/P.	1> O/P depend only Present State.
2> Input change affect the output	2> Input changes doesn't affect the O/P.
3> it Requires less no. of states	3> it requires more States.
Present I/P, present State → mealy CKT → O/P	Present State → moore CKT → O/P
(4) it requires very little hardware for designing a melay machine	(4) one requires more hardware to design a moore machine.
(5) if input changes, O/P does not change	(5) if input changes. output also changes

→ mealy machine :-

(1) mealy machine has output that depend on the state and input (thus, the FSM has the output written on edge)

→ mealy machine give you O/P instantly that is immedially upon receiving I/P but the O/P is not half after that clk cycle.

Ashi_tyagi_Verilog.

classmate
Date
Page

Finite State Machines

◎ what is state machine ?

- every Sequential circuit compulsorily has a state machine.
- Sequential design represented using various states will all possible state transitions shown based on the input applied.

- A Synchronous Sequential circuit is also called as Finite state machine FSM. if it has finite number of states.
- There are two types of FSMs.

① mealy state machine :
:- output depends on current states and inputs applied.

② moore state machine :-
:- output depends only on current state.

: in digital logic design FSM are used to design a Controller (Vending machine Controller, traffic Controller, Controller for processor)
ex :- ATM machine
∴ inside ATM machine atm Controller working which is Controlling all the circuits of events that has to be occurred.

• in ATM card :
- Card / insert
- Banking
- Pin
- Saving / current
- Amount
- cash

All are predetermined action happens. and all those things happen sequentially.

∴ So these finite statement is used to design all these Controllers.

• FSM has finite number of states and based on its current inputs and history (states - present). it determines what to do next."

◎ Every FSM has 3 parts :

1.) state memory :
[it stores current state that FSM is in]

② next state logic :-
[determines the next state for FSM and which depends on current state it is in and input signals.]

3.) output logic :
[generate required output signals from Controller from Controlling that we want].

• FSM is in only one state at a time. The state it is in at any given time called current state.

DAY 19 – 100 DAYS VERIFICATION CHALLENGE

Topic: Sequential & combinational circuits

Homework: Understand about Sequential & combinational circuits.

DAY 19 CHALLENGE:

1. List the difference between Sequential & Combinational Circuits.
2. List the applications of Sequential & Combinational Circuits.
3. Why does sequential circuit contain memory element?
4. What is an asynchronous sequential circuit?
5. What are the two types of asynchronous sequential circuits?
6. What are the types of hazards in asynchronous sequential circuits?
7. What is race condition in asynchronous sequential circuits? How to avoid this race conditions?
8. What is the difference between race condition and deadlock?
9. How do you prevent deadlocks?
10. What is a glitch? How to avoid & fix glitches?

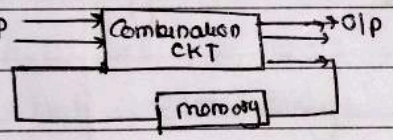
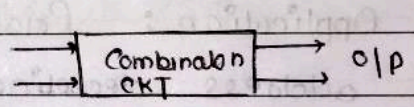
Copyright @Sweety Pinjani

Arti_tyagi_digital_2018

DAY : (19)

Topic : Sequential & Combinational Circuits.

Sol^n ① List The difference Between Sequential & Combinational Circuits.

Sequential CKT	Combinational CKT
① in this output depends upon present as well as past input.	① In this output depends only upon present input.
② There exists a feedback path Between input and output.	② There is no feedback Btw input & output.
③ Sequential CKT, mainly make use of these types of CKT for storing data.	③ One can use these types of CKT for both Boolean as well as Arithmatic operations.
④ 	⑤ 
ex: flip-flop, counter, Registers.	eg:- Half adder, full adder, mux.
⑤ clock signal is required.	⑤ No clock signal is applied.

Sol^n → ② List The application of Sequential & Combinational Circuits.

∴ Application of Sequential CKT	Application Combinational CKT
① Shift Registers & flip-flop	① Arithmatic circuits multiplexer & demux.
② Counters	② digital calculators, computers
③ used as Registers inside microprocessor & Controller to store temporary information	③ Combinational ckt are used in digital electronics, such as computer, to perform operation on data.
④ Analog to digital & digital to analog Converter.	④ Error Correction and detection.

Sol^n → ③ why does Sequential circuit Contain memory element?

∴ Any sequential circuit Consists of logic gates whose outputs at any time are determined from both the present combination of input and previous outputs. That means Sequential circuit use memory elements to store the value of previous output.

→ The memory elements are circuits capable of storing Binary Information.

Sol^n → ④ what is an asynchronous Sequential circuit?

∴ Asynchronous sequential circuit are digital sequential CKT in which the feedback to the input for next output generation is not governed by a universal clock signal.

→ Asynchronous Sequential CKT works without clock.

→ Asynchronous Sequential CKT worked based on memory concept.

→ The memory of the asynchronous sequential circuit may include flip-flops or time-delay devices.

Sol^n ⑤ what are the two types of asynchronous sequential circuits?

∴ According to input variable there are two types:

① fundamental mode circuit:

→ The input variable change only when the CKT is stable. Only one input can change at a given time.

→ Input are levels (0, 1) and not pulses.

② Pulse mode circuit:

→ The input variables are pulses (True, False) instead of levels.

→ The width of the pulses is long enough for the circuit to respond to the input.

→ The pulse width must not be so long that that it is still present after the new state is reached.

Sol^n ⑥ what are the types of hazards in asynchronous Sequential Circuits?

Essential hazards may occur in Asynchronous Sequential CKT:

HAZARDS:- There are two main type Hazard: Static Hazard & Dynamic.

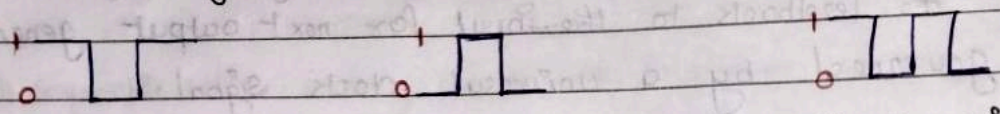
→ Unwanted switching appears at the output of a circuit.

∴ due to different propagation delay in different paths.

→ may Cause the circuit to mal-function.

- Cause temporary false-output values in Combinational CKT.
- Cause a transition to a wrong state in asynchronous CKT

- Three types of hazards:



ⓐ static 1-hazards ⓑ static 0-hazards ⓒ Dynamic hazard

- Static and Dynamic Hazard:-
- when the I/P to the logic CKT changes the O/P has to remain at a particular logic. but instead it may change its value momentarily then there exist a Static Hazard.
- A hazard is said to be dynamic if the O/P change two or more times when it should change from 1 to 0 or from 0 to 1.

Sol^n → ① what is race Condition in asynchronous Sequential CKT? How To avoid this race Conditions?

race condition occurs when a device is designed assuming a particular sequence of without taking steps to ensure it. in logic gates, it happens when the inputs arrive at the gate in a sequence not assumed while driving the function Sometimes Result anomalous behaviour of the logic gate because of unexpected dependence on the sequence of arrival of inputs. Since the function of the logic gate is calculated assuming static I/P The order of arrival may impact on the O/P. we can consider race Condition as a situation in which two or more signal are racing to have their effect on the O/P.

→ To avoid race Condition :-

→ we need to have proper sequencing of signals so as to have the desired o/p.

→ we Can use edge trigger.

→ By adding delay elements.

Sol^n ⑧ what is the difference Between race Condition & deadlock?

- A deadlock is a situation in which more than one process is Blocked.
- Deadlock happen in multithreaded programs, race condition happens in a single-Threaded
- A race Condition is when two & more threads try to change the shared data at the same time.
- deadlock cause a crash, race Condition does not.

Sol^n ⑨ How do you prevent deadlocks?

- when two or more process try to access the critical Section at the same time and they fail to access simultaneously or stuck while accessing the critical section then this is known as deadlocks.

∴ we can prevent a deadlocks by eliminating any of the above Conditions.

i) Eliminate mutual Exclusion

ii) Eliminate hold and wait

iii) Eliminate No preemption

iv) Eliminate circular wait

Sol^n (10) What is glitch? How to avoid & fix glitches?

- Glitches can be a major issue for digital circuits and can lead to unpredictable and unexpected behavior. Glitches can occur due to a variety of causes, such as noise, slow transition times and improper circuit design.
- A Glitch is an unexpected and usually short-lived fault in a system, such as a transient fault that corrects itself, or an error in a program that cause it to freeze or crash.

→ To remove glitches:

i) Sample the signal the glitch appears on via a flop.
ii) Gray encode the flops that may be causing the glitch.
iii) Delay insertion & having hazard free logic.
iv) adding to much delay also cause glitch so minimize no. of delays.
v) By Reducing switching activity.

DAY 20 – 100 DAYS VERIFICATION CHALLENGE

Topic: Boolean Algebra, Number System & Digital codes

Homework: Let's learn about Boolean algebra, number system & digital codes.

DAY 20 CHALLENGE:

1. How many types of number system are there? Describe each in detail.
2. What are the fundamental properties of Boolean algebra?
3. State the disadvantage of a 8-4-2-1 code
4. Explain below codes:
   i. BCD code
   ii. Gray code
5. What are the applications of:
   i. Weighted code
   ii. Non-weighted code
   iii. Excess-3 code.
   iv. BCD code
   v. Gray code
   vi. ASCII code
   vii. Octal Number System
   viii. Hexadecimal Number System
6. What are different ways to represent a negative number?
7. What is meant by isomorphic Boolean algebra?

Copyright @Sweety Pinjani

Date
Page

antityagi digital 2023

Day : 20

Topic : Bolean Algebra, number System & digital Codes.

Sol^n ① How many types of number System are there? describe each in detail.

∴ Based on the base value & the number of digits, number System are many types: The four Common types of Number System are

① Decimal number System [Base is 10]
② Binary number System [Base is 2]
③ Octal number System [Base is 8]
④ Hexadecimal number System [Base is 16

① Decimal number System :- A number system with a base value of 10 because it uses ten digits from 0 to 9. eg: 1457

eg: $(1\times10^3) + (4\times10^2) + (5\times10^1) + (7\times10^0)$

$(1\times1000) + (4\times100) + (5\times10) + (7) \Rightarrow 1457.$

② Binary number System :-

A base 2 number System is also known as Binary number System wherein, only two Binary digit exist. ie 0 and 1. the usual base-2 is a radix of 2.

eg: $(14)_{10} = (1110)_2$

③ Octal number System :-

The base is 8 & it used number from 0 to 7

• Octal number are comonlly used in Computer applications.

eg: $(215)_8 = 2\times8^2 + 1\times8^1 + 5\times8^0 \Rightarrow 128 + 8 + 5 \Rightarrow (141)_{10}$

④ Hexadecimal number System :

number are written or represent with base 16. in this first represent just like decimal ie 0 to 9 then the numbers are represent using alphabet from A to F. SEAM ← eg:

eg:

Date
Page

Sol^n ② what are the fundamental properties of Boolean algebra?

① Communicative : The order of the operands in a Boolean expression does not affect the Result.

eg:- $A+B = B+A$
$A \cdot B = B \cdot A$

② Associativity :

The way Boolean expression are grouped does not affect the result :

eg :- • $A+(B+C) = (A+B)+C$
• $A \cdot (B \cdot C) = (A \cdot B) \cdot C$

③ distributivity :-

The AND and OR Operation distribute over each other

eg:- $A \cdot (B+C) = AB+AC$

④ Complement :- Every Boolean expression has a complement which is opposite of the expression.

eg:
$A+A' = 1$
$A \cdot A' = 0$

⑤ Inversion :- $A = (A')'$

⑥ Demorgan's Theorem : $(a+b)' = a' \cdot b'$
$(A \cdot B)' = A'+B'$

Sol^n ③ State the disadvantage of a 8-4-2-1 Code?

① Requires more number of bits for representation.
② makes arithmatic operation more complex.
③ difficult to convert to other binary codes
④ Inefficent in terms of spaces.

Sol^n (4) Explain Below Codes :-

(i) BCD code (ii) Gray Code

(1) BCD Code :-

→ Binary coded decimal

→ weighted code & 4 bit code means each decimal digit with represent with 4 bit.

Decimal	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
⋮	⋮
15	1111

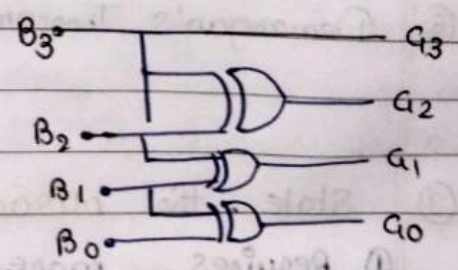
invalid BCD code or don't care

(2) (ii) Binary to Gray code (Gray code):-

i.) Unweighted code. & also called unit distance code

Binary 1 0 1 1

Gray 1 1 1 0

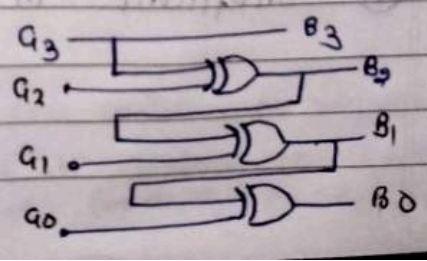


ii.) Gray to Binary:

Gray 1 1 1 0 ← XOR operation

Binar 1 0 1 1

(MSB) LSB



Sol^n (5) what are the application of :-

(1) weighted code :

(1) it is used in digital circuits to represent and data manipula during Arithmatic operations.

(2) it is also used to detect & correct errors in digital data transmission

(3) One-hot code

(2) Non-weighted code : APPLICATION

1.) Error detection and correction.

2.) Code division multiple access (CDMA)

3.) Gray code & Excess-3 Code.

(3) Excess - 3 Code : Application

(1) Arithmatic operations

(2) Error correction and detection

(3) Digital signal processing

(4) BCD code : Application

(1) digital clocks (2) Aerospace System (3) Communication System

(5) Gray code : APPLICATION

(1) Digital television (2) Robotic (3) optical communication Syst

(6) ASCII code : APPLICATION :

(1) Communicaton Protocols (2) Programming langu (3) Web Browsers.

(7) Octal number System : APPLICATION

(1) Computer Science (2) Assemply language (3) Electronics

(8) Hexadecimal number System : Application

(1) Web development (2) machine code (3) Assembly language

DAY 21 – 100 DAYS VERIFICATION CHALLENGE

Topic: Optimization techniques

Homework: Optimization techniques form the backbone of designing chips. Let' learn about it.

DAY 21 CHALLENGE:

1. State the De Morgan's Theorem?
2. What are the 4 methods to reduce a Boolean expression?
3. What is meant by K-Map or Karnaugh Map? What are the limitations of the Karnaugh Map?
4. What are the advantages and disadvantages of the K-Map Method?
5. What is the Quine-McCluskey method?
6. Which code is called a minimum change code and why?
7. What are degenerate forms in two-level logic implementation?
8. What are logic optimization techniques?
9. How do you optimize a logic circuit?
10. Explain below concepts:
    i. Duality Theorem
    ii. Minterm and Maxterm
    iii. Consensus Theorem

Copyright @Sweety Pinjani

Aditiyagi_digital_2023

DAY : 21

Topic : Optimization Technique

Sol^n ① State the Demorgan's Theorem?

De Morgans laws :

∴ The Complement of the union of two sets is the intersection of their Complements.

eg: $(x+y)' = x' \cdot y'$

∴ The Complement of the intersection of two sets is the union of their Complements.

eg: $(x \cdot y)' = x' + y'$

Sol^n ② what are the 4 methods to reduce a Boolean expression?

① Boolean Algebra
② Karnaugh maps (K-map)
③ Quine - mccluskey Algorithm (Tabular method)
④ Exercises

Sol^n ③ What is meant by K-map or Karnaught map? what are limition of the Karnaugh map?

Kmap is a table like representation but it provides more information than the truth Table. we will the grid of K map with 0's and ones then solve it creating group. Although this approach may applied any no. it is used Speed up to six variable.

K map key Terms

minterms and maxterms

minterms : its known as the product term. In the minterm, each uncomplemented term is indicated by '1' & each Complemented term is '0'

eg :- 011 ⇒ ABC

- maxterms :- it is know as the Sum term, In maxterm, each uncomplemented term is indicated by 0 & each complemented is indicated by '1'.

  eg: 1 0 0 → $\bar{A} + B + C$

⇒ Sum of Product (SOP Form): In SOP or Sum of Product, form variables are ANDed (i.e multiplied) & used to interept output with logic '1' combination.

→ In SOP form each term is identified as a minterm.

eg:- $ABC' + A'BC + ABC$

110 011 111

⇒ Product of Sum (POS Form): In pos or product of Sum, from variables are ORed (ie added) & applied to represent o/p with logic '0' combination.

→ In POS form every single term is known as the max term. eg:- $(A+B'+C)\ (\bar{A}+B+C)\ (A+B+C)$

010 101 000

:- Limitations of Kmaps :-

① The most significant limitation of the Kmap is that it only efficient when the Boolean expression has less number of variable. it becomes quite complicated with the higher number of variable in the logical expression.

② The Simplification of a Boolean function having more than or equal to five variables using K-map is quite complex.

③ its difficult to make judgements about which combination from the minimum expression.

Sol^n ④ what are the advantage and disadvantage of the K-map method ?

Advantage K-map	disadvantage K-map
① Reduced expression without Boolean theorem	① Become tedious for 5 variable K-map
② minimized number of logic gates used.	② It is not suitable for Computer reduction

Sol^n ⑤ what is the Quine - mc cluskey method ?

→ Quine - mccluskey method is also known as the Tabulation method is used to minimize the Boolean functions.

→ It simplified Boolean expression in to the simplified form using prime implicants.

→ This method is Convient to simplyfy boolean expression with more than 4 input variables.

⊙ Terminologies used in Quine mccluskey methods are: implicant, prime implicant and essential prime implicant

- Implicant : implicant can be defined as a collection of one's for minterms.

→ Prime implicant, prime implicant can be defined as a collection of min possible group's of one's for minterm.

→ Essential prime implicant, it is a group that covers as a minimum single minterm that can not be enclosed with other applicants. So this technique utilizes decimal to Binary representation.

Sol^n ⑥ what/which code is called minimum change code and why?

→ Gray code is the minimum change code category of coding in which, The two consecutive values changes by only a single bit.

→ It is also termed as reflected binary code or cyclic code.

→ Gray codes are less error-prone because they only change in one-bit position at a time, hence they are considered as the minimum error code.

→ Gray codes are mostly used to facilitate error correction in digital communication

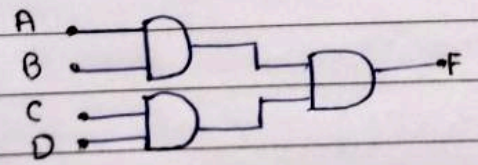
Sol^n ⑦ what are degenerate forms in two-level logic implementation?

∴ if the output of two level logic realization can be obtained by using single logic gate, then it is called as degenerative form. obviously the number of inputs of single logic gate increase.

→ Degenerate logic gate combinations are those which are degenerate and result in a single operation effectively.

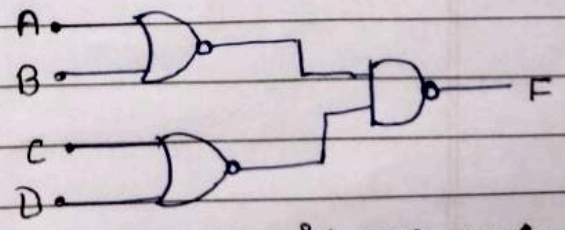
eg: AND-AND, OR-OR, AND-NAND, OR-NOR, NAND-OR, NOR-AND NOR-NAND, NAND-NOR combinations are degenerate because they will result in a single operation effectively

eg:- AND-AND logic



∴ here F will be : A·B·C·D (a single level logic that can performed with a 4-input AND gate only)

:- NOR-NAND logic

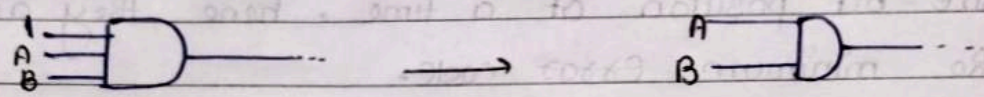


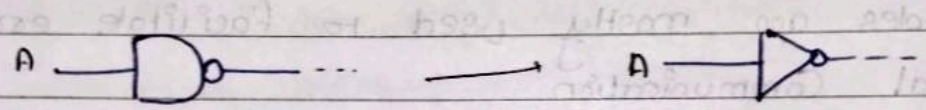
∴ here F will give you:- $[(A+B)' \cdot (C+D)']$ == $(A+B+C+D)$ (using de-morgan's law) which is nothing but a single level 4 input OR gate.

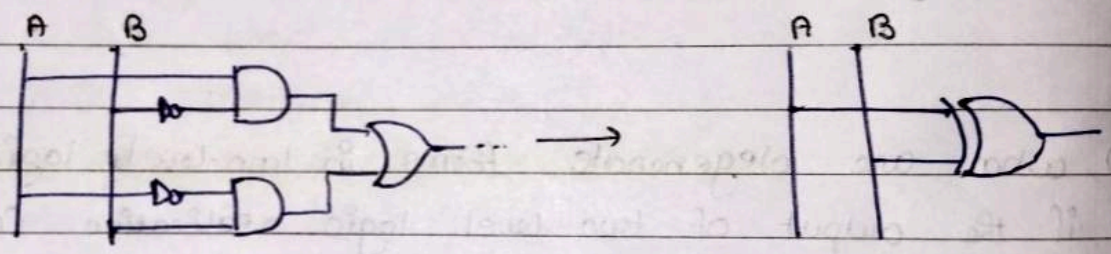
Soln ⑧ what are logic optimization techniques?

- logic optimization is a process of finding an equivalent Representation of the specific logic circuit under one or more specified constraints.
- logic optimization reduce the program size and execution time.

Before optimization after optimization

① 

② 

③ 

Soln ⑨ How do you optimize a logic circuit?

∴ The purpose of logic optimization is to enhance the simulation efficiency.

logic optimization is divied into various catagories

- Based on circuit representation
- Based on circuit characteristics.
- Based on type of execution

Sol^n (10) Explain Below Concepts:
① duality Theorem ② minterm & maxterm ③ Consensus Theorem

① **Duality Theorem:**

The duality Theorem states that the dual of the Boolean function is obtained by interchanging the logical AND operator with the logical OR operator & zeros with ones.

→ for every Boolean function, there will be corresponding dual function

→ Dual expression is also used to convert positive logic to negative logic & vice-versa

i.) variable are not complemented, $A \cdot B \rightarrow A + B$

ii.) ones are interchanged with zero & vice-versa, $A + 1 = 1 \rightarrow A \cdot 0 = 0$

• de morgan's theorem, works depending on the concept of Duality.

② **minterm & maxterm:-** Boolean function can be represented in two forms: ① POS (Product of Sum)
② SOP (Sum of Product)

∴ Sop representation is the sum of minterm where minterm is the function variable combination for which the o/p is high (1).

∴ In POS representation, a product of maxterm.

eg: POS :- $(A+B) \cdot (C+D)$ eg: SOP : $AB + A'B'$

③ **Consensus Theorem:** The redundancy Theorem is used as a Boolean algebra trick in Digital Electronics.

It is also known as Consensus Theorem.

$$AB + A'C + BC = AB + A'C$$

∴ The Consensus or resolvent of the terms AB & A'C is BC.

Its the conjunction of all the unique literals of the terms. excluding the literal that appears unnegated in one term & negated in other.

Condition for applying Reducdency Theorem: ① Each variable repeated twice ② Three variable must present in eq^n. A, B & C are used as variable ③ one variable must present in the complemented form.

DAY 22 – 100 DAYS VERIFICATION CHALLENGE

Topic: Static Timing Analysis

Homework: Let's deep dive into STA concepts.

Understand STA numerical online. Here is one link where you can find some solved STA numerical:
https://www.vlsiuniverse.com/sta-solved-problems-vlsi-interview-2021/

DAY 22 CHALLENGE:

1. What do you mean by clock skew and clock jitter?
2. What are the different types of skews used in VLSI?
3. What is slack in VLSI? What is negative slack & positive slack?
4. Explain:
    i. Setup Time
    ii. Hold Time
    iii. Rise Time
    iv. Fall Time
5. How can you avoid setup & hold time violations?
6. What are ideal characteristics of a clock during STA?
7. What is the difference between time borrowing & time stealing?
8. What is timing path? What are start & end points?
9. Explain following concepts:
    i. Launch Edge
    ii. Capture Edge
    iii. Reset Assertion
    iv. Reset De-assertion
    v. Critical path
    vi. False path
    vii. Multicycle path

Copyright @Swetty Pinjani

Artiyogi_digital2023

DAY : 35

Topic : STATIC Timing Analysis

Sol^n ① what do you mean by clock Skew and clock jitter?

Clock skew refers to the variation in arrival times of clock signals at different signals paths in a circuit introduce different delays to the same clock signal.

- The deterministic difference in clock arrival times at each flip-flop.
- Caused mainly by imperfect balancing of clock tree.
- can be deliberately introduced using delay Blocks in order to time-borrow.

Clock jitter :-

Clock jitter represents the fluctuations in the timing of clock edges. it is the amount of variation in the periodicity of a clock signal.

- jitter is caused by which high frequency enviormental variation, particular power supply noise.

- The random difference in clock arrival times at each flip-flop.
- Caused by on die process, Vdd, temprature variation crosstalk.

Date
Page

INTERVIEW QUESTION :-

④ what is Setup time and Hold time

Setup time :-

setup time is amount of time required for the input to a FF to be stable befor a clock edge.

OR minimum time Interval for which the input signal must be stable (unchanging)

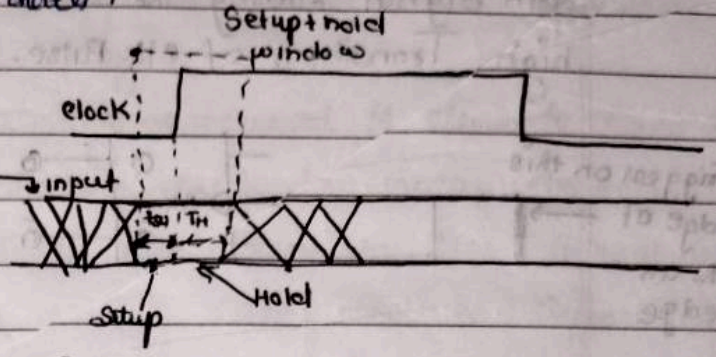
OR

Setup time is defined as the minimum amount of time before the clock active edge that the data must be stable for it to be latched correctly. in other words, each FF (or any sequential element in general) needs some time for the data to remain stable before the clock edge arrives.

→ Setup time signifies maximum delay constraints.

Hold time :-

Hold time is defined as the minimum amount of time after the clock's active edge during which data must be stable. similar to set-up time, each sequential element needs some time for data to remain stable after clock edge arrives to reliably capture data.



→ in figure showing : that the input data cannot change with in setup/hold window for a FF.

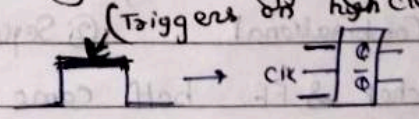
→ hold time is for minimum delay constraints.

⊙ A Hold time violation is when a input signal change too quickly, after the clock's active transition.

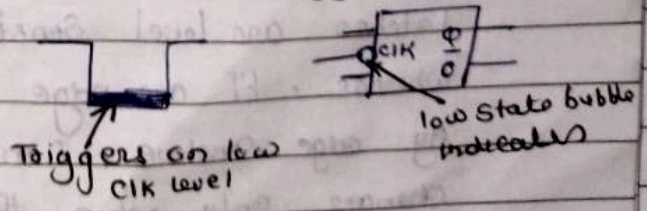
- A setup time violation is when a signal arrives too late, and misses the time when it should advance.

:- CLOCK PULSE TRIGGERING OF FLIP-FLOP

① High level Triggering



② Low level Triggering



• Flip-flops are two types

1. Positive edge triggered
2. Negative edge triggered.

Sol^n ② what are different types of Skews used in VLSI?

There are two types of skew used in VLSI.

① Local skew ② Global skew

① local skew:

local skew is the difference in the arrival of clock signal at the clock pin of reflected flops.

② Global skew:

Global skew is the difference in the arrival of clock signal at the clock pin of non-related flops.

Global skew = longest clock path - Shortest path.

∴ another two types are skew ① positive skew ② Negative skew

① Positive skew:

if capture clock comes late than launch clock then it is called positive skew.

∴ it can lead to hold violation but improves setup-time

② Negative skew:-

it capture clock comes early than launch clock. it can lead to setup violation but improves hold time.

Sol^n ③ what is slack in VLSI? what is negative slack & positive slack?

∴ Slack is the difference btw required time and arrival time.

min slack = Arrival time - minimum expected time ⇒ Positive slack or setup slack.

max slack = max expected time - Arrival time ⇒ Negative slack or Hold slack

∴ if the value are negative then there is a violation. if setup slack is -ve then its setup violation and if Hold slack -ve then its hold violation.

Sol^n ④ Rise time :-

Rise time is the time taken to force the value of a signal to high from low logic level, x, z.

FALL time :- Fall time is the time taken to force the value of signal to low('0') from all other logic 1, x, Z.

Sol^n ⑤ How can you avoid setup and hold time violation?

i. Improve the setup time constraint of the capture flip-flop when there is a setup violation on any path in the design. the capture flop can be reduced with a flop that has a small setup time window so that the path can accommodate large data path delay.

ii. Reduce clock to a delay at launch flop.

iii) Using a faster cell for launch flip-flop.

iv decreasing the logic delay.

Avoid hold violation:

i) Increase the clock a delay of launch flip flop.

ii) Icrease logic delay

iii) Improving the hold time constraint of launch flip-flop

iv. Use a slower cell for launch flip-flop.

Sol^n ⑥ what are ideal characteristic of a clock during STA?

(i) The clock should be free of glitches.

(ii) clock jitter should be avoided & duty cycle should be s

(iii) The period of the clock should be accurately determined and correct phase relationship btw two different clocks of interest should be created.

(v) The worst-case duty-cycle should be applied when moving data between clock edge.

Sol^n ⑦ what is the difference Between time Borrowing & time stealing?

Time Borrowing	Time stealing
① Time Borrowing applies to latch - based design.	① whereas time stealing applies to flip-flops based design.
② In the time Borrowing concept there is no interference with the clock phase. The latch uses the previous cycle slack automatically throug the pipeline	② time stealing, steal the time from the next stage and it results in less time to next stage.

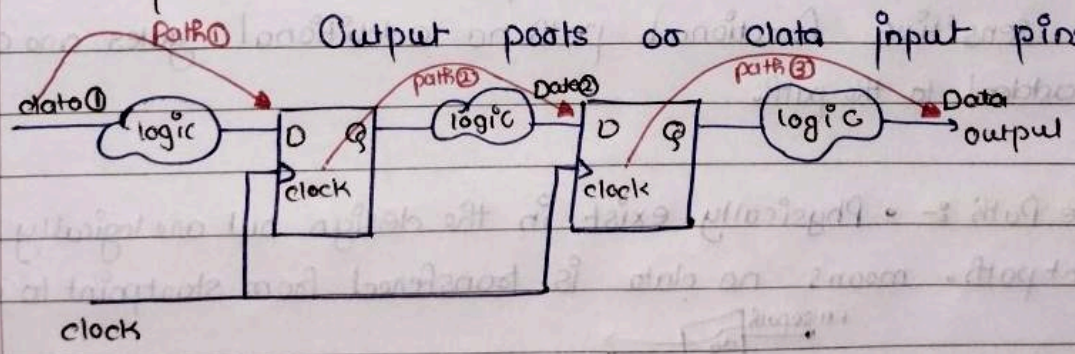
Sol^n ⑧ what is timing path? what are start point & end points?

∴ A Timing path is a ~~poi~~ point to point path in a design which can propagate data from one flip-flop to another.

→ Each path has a start point and an end point.

- Start point :
  input ports or clock pins of flip-flops.

- End point :
  Output ports or data input pins of flip-flops.



∴ Timing path are grouped into path groups By the clocks controlling their endpoints.

Sol^n ⑨ Explain Following Concepts :-

① **Launch Edge :-**

Launch edge is the active edge of the clock at which data is launched at flipflop output. The launch edge is the edge of the launch clock, which triggers the data in to initial flop.

② **Capture Edge :** The capture edge is the specific edge of a clock signal at which the data is captured by a flip-flop or latch at the ending point of a timing path

③ **Reset DeAssertion :-**

Resetdeassertion is the release of reset mode when the reset signal is logically false. during asynchronous reset, de-assertion may cause metastability coz it may be the case that some flip-flops come out of reset befor other

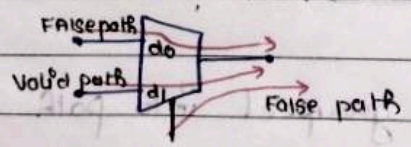
④ **Reset Assertion :-** Reset assertion in STA is when the reset signal is logically true. This means that the reset signal is active.

⑤ **critical path :-** • The path which creates longest delay & also called worst path / late path / max. path.

• Timing sensitive functional path no additional gates are allowed to be added to the path.

⑥ **False Path :-** • Physically exist in the design but are logically inactive. incorrect path. means no data is transfered from startpoint to endpoint.

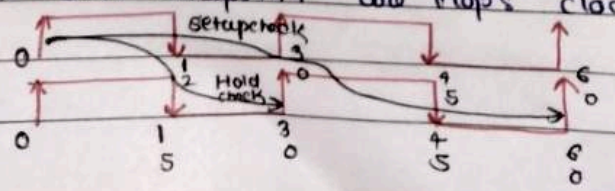
eg :-



⑦ **multicycle path :-**

∴ Timing path that is designed to take more than one clock cycle for the data to propogate from the start point to end point.

• start point and endpoint are flops clocked by the same clock.



DAY 23 – 100 DAYS VERIFICATION CHALLENGE

Topic: Memory elements

Homework: Let's learn about different memory elements.

DAY 23 CHALLENGE:

1. Difference between RAM and ROM.
2. How is data written in ROM?
3. Explain different types of ROM Memory
4. Why does Dynamic ROM need to be refreshed multiple times per second?
5. What is the size of ROM for the n-bit full adder?
6. Explain different types of RAM Memory.
7. What is the difference between static RAM and dynamic RAM?
8. What is EPROM? What is its application?
9. Difference between EPROM and EEPROM.
10. What are the essential features of a memory element?
11. What are Shift registers? What are the types of Shift registers?
12. How many 32K * 1 RAM chips are needed to provide a memory capacity of 256 K-bytes?
    (Hint: No. of chips needed = Memory capacity/RAM chip's capacity)

Copyright © Sweety Pinjani

Aditya digital. 2023

DAY: 23

Topic: Memory Elements

Sol^n ① Difference Between RAM and ROM.

RAM	ROM
① RAM (Random Access memory) is costly if Compared to ROM.	① ROM is more inexpensive when Compared to RAM.
② Data stored in RAM can be modified, erased, retrived and altered and read.	② data stored in ROM can only be read. It cannot be revised or cancelled.
③ The capacity of RAM is higher when Compared to ROM.	③ ROM has a lesser capacity when Compared to ROM.
④ Data of RAM is volatile, that is it will survive as long as there is no break power. It can said RAM is temporary memory of Computer	④ Data available in Read-only memory is non-volatile, that its permanant and will remain Constant even then when there is disturbance in the power supply. It can said ROM is permanant memory of the Computer.

Sol^n ② How is data written in ROM?

∴ Data is written onto the chip by the manufacturer. just like with any other IC transistors are placed in such a way as to contains the data. This chip written to during manufacturing and cannot be erased or re-written afterwards.

Sol^n ③ Explain different types of Rom memory?

∴ Types of Rom:

① PROM (Programmable Read-only memory)
② EPROM (erasable programmable read-only memory)
③ EEROM (electrically errasable programmable Rom)
④ mask ROM (MROM)

① PROM: (Programmable Read-only memory)

∴ PROM is a type of Rom that can be ~~implemented~~ amended only once by a user. The user buys an empty PROM and inputs the required data using a PROM program.

- it consists of the small fuses inside which are burnt open during programming. its possible to program this memory only once and is not errasable.

② EPROM: (Erasable & Programmaby Read-only memory)

∴ it is possible to errase EPROM by exposing it to ultraviolet light for a period of up to 40 minutes. normally, an EPROM eraser peforms this operation. An electrical charge is stored in an isulated gate region while programmy.

→ it can be reprogrammed. To erase data from it, expose to ultraviolet light. To reprogram it, erase all the previous data.

③ EEPROM: The data can be erased by applying an electric field, with no need for ultra violet light. we can erase only portions of the chip.

④ MROM [MASK ROM]:

mask Rom is a kind of Read-only memory that is masked off at the time of production. Like other types of Rom, mask Rom cannot enable the user to change the data stored in it. if it can, the process would be difficult or slow.

Sol^n ④ why does Dynamic Ram need to be refreshed multiple times per Second?

∴ it must be refreshed many times per second to store information otherwise, it will lose what it is holding.

→ it is most common type of memory in use today.

→ inside a dynamic RAM chip, each memory cell hold one bit of information and is made up of two parts: ① a transistor and a capacitor.

∴ These are, extremly Small transistor and Capacitor so that millions of them can fit on a Single memory chip.

Sol^n ⑤ what is the size of RAM for the n-bit full adder?

∴ The size of RAM for n-bit full adder can be calculated based on the number of input Combinations it need to handle.

So the size of the RAM for n-bit full adder is 2^N bits. and RAM Stores the Truth Table for the full adder.

Sol^n ⑥ Explain different types of RAM memory?

→ Types of RAM :-

① DRAM ② SRAM

① SRAM (Static RAM):

∴ SRAM stand for static Random Access memory. it is a type of Semiconductor which is widely used in Computing devices & microprocessors.

→ The word static implies that the memory holds its Contents as long as electricity is being supplied & the data is dumped when the power gets down because of its volatile nature.

→ SRAM chips are equiped with a matric of 6-transistor & no capacitors.

• Transistor are not dependent on power to prevent leaking e. hence SRAM need not to be refreshed at a regular interval.

② DRAM [Dynamic RAM]:

→ unlike SRAM, DRAM need to be constantally reffreshed so data should be maintained. This is achieved by placing the memory on a refresh circuit that rewrites the content several hundred times every second.

→ DRAM is installed for most system memory as its cheap & small.

→ All DRAM consists of memory cells, which constitute a single capacitor & a single transistor.

:→ Few other types of RAM are:

- SDRAM [Synchronous Dynamic RAM],

SDRAM is a type of DRAM & works in syn with the CPU clock, which implies it wait for the clock signal before acknowledge the data input.

- mostly appled in computer memory, video game etc.

- SDR RAM [Single Data Rate Synchronous Dynamic RAM],

- The 'Single data rate' symbolises how the memory processes. it can process one read & one write instruction per clock cycle.

- popularly used in computer memory, video game consoles etc.

- DDR SDRAM [Double Data Rate Synchronous Dynamic RAM],

- DDR SDRAM works similar to SDR SDRAM just twice faster than it.

- DDR SDRAM can process two read & two write instruction per clock cycle. mosly used in computer memory. The other upgraded versions of DDR SDRAM are DDR2, DDR3 & DDR4.

- GDDR SDRAM [Graphics Double data Rate Synchronous Dynamic RAM]

- its variety of DDR SDRAM & its specifically designed for video graphic cards.

- Flash memory:

flash memory is a sort of non-volatile storage that holds all data. after power off also. mostly used in digital camra, tablets, phones

Sol^n ⑦ what is difference Between static RAM and dynamic RAM?

Static RAM [SRAM]	Dynamic RAM
① uses a flip flop to store data.	① uses a capacitor and a transistor to store data.
② maintain stored data as long as power is supplied.	② Data must be constantly refreshed to maintain its state.
③ High power consumption as it needs a constant power supply.	③ low power consumption as refreshing consumes less power.
④ more expensive due to its complex design and high power consumption.	④ less expensive due to its simpler design and low power consumption.

Sol^n ⑧ what is EPROM? what is application?

EPROM (Erasable programmable Read only memory) is a type of non-volatile memory that can be proggramed and reprogrammed using special equipment. it is called "Read only" because it cannot be written to or modified by a computer's central processing unit (CPU). However, it can be erased using ultraviolet light. and then reprogrammed using a device called a PROM programmer or EP Programmer.

Application :-

→ EPROP chips are used in variety of application, including computer system, industrial control system and consumer electronics.

→ EPROM are also used in other type of device, such as tv Remote controls, printer controllers and automotive engine control system. The are often used in these type of appications because they offer a convienint way to store & update program code without having to physical replace the chip.

(9) Difference Between EPROM and EEPROM.

EPROM	EEPROM
① Read only.	② Read only.
② Can be erased by exposing to ultraviolet light.	② Can be erased electrically, without removing the chip from the CKT.
③ must be removed from the CKT and programmed with a special device.	③ Can be programmed while in the circuit.
④ it is an older type of technology.	④ it is a Comparatively modern version as Compared to EPROM.

Sol^n (10) what are essential features of a memory elements?

The most essential feature of memory is the ability to encode store and retrive information. or memory element is a fundamental block in digital electronics and Computer system that can Store and retain Binary information (0 and 1). The essential feature of memory element include:

- **Storage capacity:** memory element should able to store binary data typically in the form of bits.
- Read operation:

The ability to read the stored data is a fundamental feature of a memory element. it should provide an output that represents the value of the stored data when read operation performed.

- write operation:

A memory element should support write operation which allows you to change the stored data to a new binary value.

- Speed: memory elements should have a certain speed both reading and writing operations.
- Cost
- Volatility
- Stability and Reliability

Sol^n (11) what are shift Registers? what are the types of shift Register.

→ Shift Register are Sequential logic ckt, capable of storage and transfer of data or a group of flip-flops which is used to store multiple bits of data & data moved from one flip-flop to another flip-flops is known as shift Register.

→ if the register is capable of shifting bits either towards right hand side or towards left hand side or know as shift Register.

→ An 'N' bit shift Register contain N Flip-flops.

The shift Register is classified in to following types:

- Serial in Serial out [SISO]
- Serial in parallel out [SIPO]
- parallel in Serial out [PISO]
- parallel in parallel out [PIPO]
- Bidirectional shift Register
- Universal shift Register

Sol^n (12) How many 32K * 1 RAM chips are needed to provided a memory capacity of 256 K-bytes?

(HINT : No. of chips needed = memory capacity / RAM chips capacity)

Given:

RAM chip's capacity = 32K x1

= 32 x1024 x1 bits

memory capacity = 256K bytes

⇒ 256 x1024 x8 bits

$$\text{No. of chips needed} = \frac{\text{memory capacity}}{\text{RAM chips' capacity}}$$

$$\text{No. of chips needed} = \frac{\cancel{256}^{8} \times \cancel{1024} \times 8}{\cancel{32} \times \cancel{1024} \times 1}$$

$$\boxed{\text{No. of chips} = 64 \text{ chips}}$$ Ans

information. An n bit register can store n bit information.

A register in which data can be shifted called Shift Reg

A Register is able to shift the information either from left to right or from left to right is called shift register.

→ Shift Register can perform different operations :-

1.> Serial input - Parallel output
2.> Serial input - Serial output
3.> Parallel input - Parallel output
4.> Parallel output - Serial output

→ Universal Shift Register :-

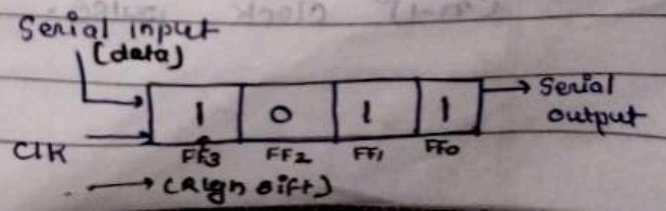
A Register which able to shift the information from left to right or from right to left and which can perform all four operations is called universal shift register.

→ APPLICATION OF SHIFT REGISTER :-

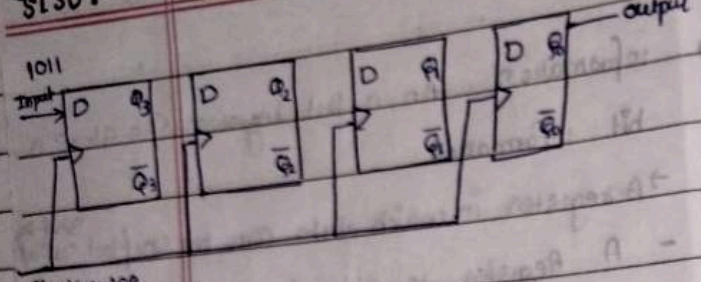
depending on application the register are two types

1.> Shift Register
2.> Storage Register

Ⓐ SISO (Serial Input in Serial out) :-



SISO :→ we can shif data two mode ① Righ Sift ② Left Shift



eg →

initial data 1011

Q_3	Q_2	Q_1	Q_0	CLK
0	0	0	0	0
1	0	0	0	1
1	1	0	0	2
0	1	1	0	3
1	0	1	1	4

∴ after 2n-1 clock we will get desired output.

→ for Serial in Register the n bit data Storage requires n clock pulse.

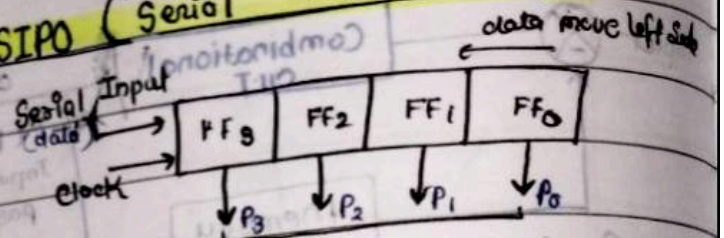
→ In SISO Register to store n bit data is required n clock pulse.

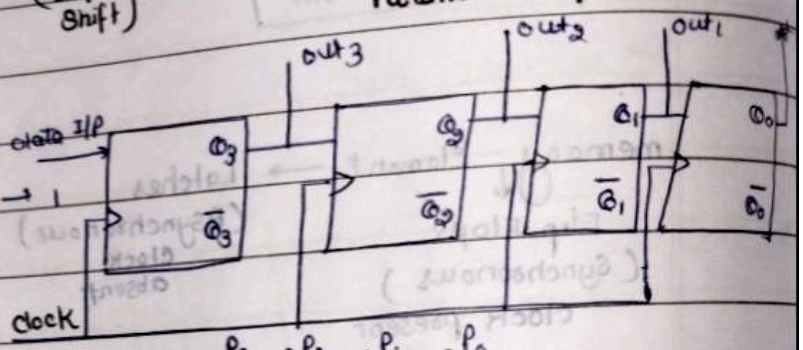
→ SISO register used to provide n clock pulse delay to I/P data.

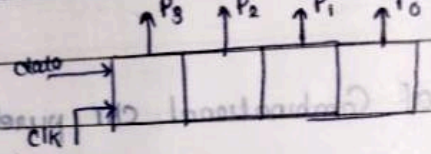
$\boxed{\text{delay} = nT_{CLK}}$

→ To provide n bit data serially out it requires (n-1) clock pulse.

Ⓑ SIPO (Serial in Parallel out):-







→ In SIPO register to provide n bit data serially in it requires n clock pulse and provide parallel o/p it requires 0 clk pulse required

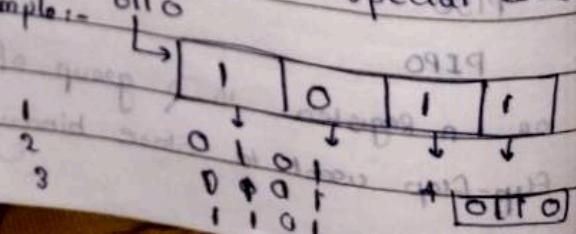
→ It's used to Serial to Parallel Converter.

→ SIPO is used to serial to Parallel Converter.

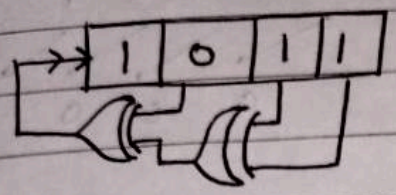
→ SIPO is used to Convert Temporial Code to Special Code.

→ Slow to fast Converter (Serial = temporal, Parallel = Special Code)

Example :- 0110



Example



initially loaded 1011 its clk pulse applied Continously after how many clk pulse again the data becomes 1010

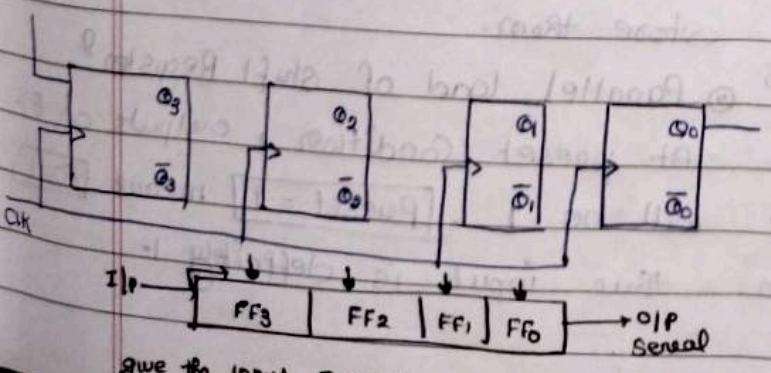
Sol

O/P of 3 variable XOR is 1 if no. of 1's is odd

CLK	Q_3	Q_2	Q_1	Q_0
1	0	1	0	1
2	0	0	1	0
3	1	0	0	1
4	1	1	0	0
5	1	1	1	0
6	0	1	1	1
7	0	0	1	1
8	0	0	0	1
9	1	0	0	0
10	0	1	0	0
11	1	0	1	0

after 7 clock

ⓒ PISO (Parallel in ~~Parallel~~ Serial out) :-



give the input parallel each FF

Control = 0 ,	Parallel in
Control = 1 ,	Serial out

→ in PISO register to provide Parallel in it require 1 clock pulse and to Provide Serial out (n-1) clock pulse.

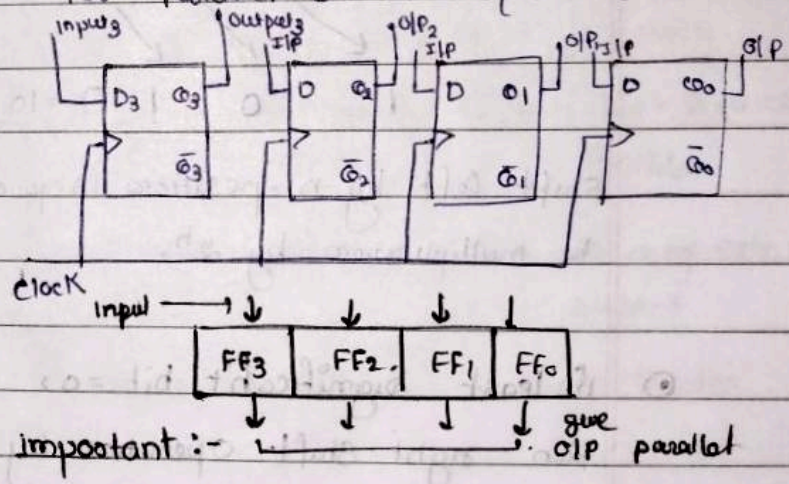
→ PISO is also used to Convert Special Code to temporial Code.

Ⓓ PIPO (Parallel in Parallel out) :-

→ PIPO is used to as Storage Register

→ for Parallel in it requires 1 clock Pulse.

→ for Parallel out it requires 0 clock Pulse



important :-

	Input	output
SIPO	n	0
SISO	n	n-1
PISO	1	n-1
PIPO	1	0

→ each shift left register operation provide multiplication by 2.

→ If n shift left operation performed then data is multiplied by 2^n.

eg

Q_3	Q_2	Q_1	Q_0	
0	1	0	1	= 5
0	0	0	0	= 2 (instead of 0.5)

→ Each shift right operation performed then data is divided by 2.

→ If n shif right operation performed then data is divided by 2^n.

- Left shift operation is nothing but multiplied by 2.

eg:

Q_3	Q_2	Q_1	Q_0	
0	1	0	1	= 5
1	0	1	0	= 10

shift left by n-position is equivalent to multiplication by 2^n.

- If least significant bit = 0, then right shift operation by one position is same as division by 2

eg:

Q_3	Q_2	Q_1	Q_0	
1	0	1	0	= 10
0	1	0	1	= 5

- If LSB = 1, then right shift operation gives integer division by 2.

eg:-

Application of Shift Register:-

1. Serial to parallel Conversion (its also called spatial to temporal Code Conversion)
2. Parallel to Serial Conversion (its also called temporal to Spatial Code Conversion)
3. Sequence generator
4. multiplication and Division
5. Ring Counter and Twisted ring Counter
6. Digital delay Line (SISO) (Serial input and Serial output operation)
7. To Storage data temporary
8. Serial data transmission

NOTE:-

→ A register is made up of flip-flop And each flip-flop store 1 bit of data

- Thus the shift Register has the capability to store one bit and if another bit is to store, in such a situation it delete the previous data and store them.

- Parallel load of shift Register?

- At preset Condition, outputs of FF will be 1. $\boxed{\text{Preset} = 1}$ means $\boxed{Q=1}$ thus input is deffinitely 1.

DAY 24 – 100 DAYS VERIFICATION CHALLENGE

Topic: Digital Electronics Miscellaneous

Homework: Focusing on some important interview questions on Digital Electronics.

DAY 24 CHALLENGE:

1. What is a sequence generator circuit? Application of sequence generator?
2. What is a transmission gate in the digital circuit?
3. What is a pass transistor logic circuit? Advantages of using pass transistor logic.
4. Why are most interrupts active low?
5. Define Pair, Quad, and Octet?
6. Define Fan-in and Fan-out?
7. What is power dissipation?
8. What is metastability? What are its effects?
9. What is an Arbiter? Explain it's operation.

Copyright @Sweety Pinjani

Arti tyagi digital 2023

DAY : 24

Topic : Digital Electronic Miscellaneous

Sol^n ① what is Sequence generator & circuit? application of sequence generator?

- A Sequence generator is one kind of digital logic circuit.
- The main function of this is to generate a set of outputs.
- A special kind of sequence generator is a Binary Counter.

eg: design sequence generator first we find no of ff we have generate sequence of 10110...

No of flip flop can be identify by

$$L \leq 2^n - 1$$

∴ where, L = length of sequence

n = No of flip-flop (minimum)

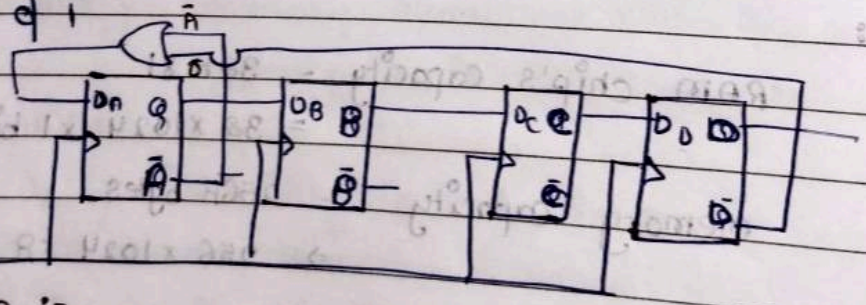
∴ here $L = 5$ $\quad 5 \leq 2^n - 1$ $\quad 5+1 = 2^n$

$\boxed{n = 3}$ minimum 3 flip flop required

A	B	C	D	output (feedback)
1	0	1	1	0
0	1	0	1	1
1	0	1	0	1
1	1	0	1	0
0	1	1	0	1

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

$\boxed{F = \bar{A} + \bar{D}}$



- Application :-
- Counter - Based Systems.
- Sequence generator are often used for memory addressing
- State machine
- Data transmision & Encoding
- microprocessor & micro controller Systems.

Sol^n ② what is a transmission gate in the digital circuit?

∴ Transmission gate represent another class of logic circuits.

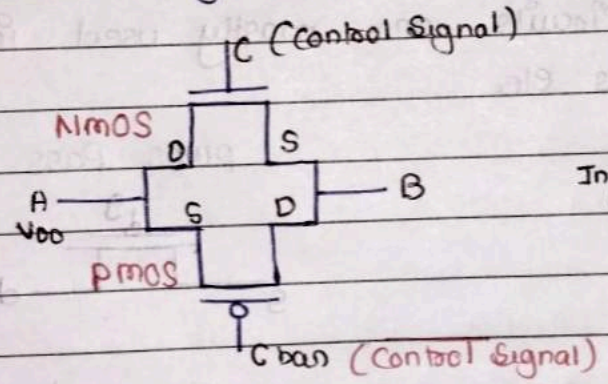
→ Transmission gate consist of a pmos and nmos connected in parallel.

→ Gate voltage applied to these gate is complementary of each other (C and Cbar)

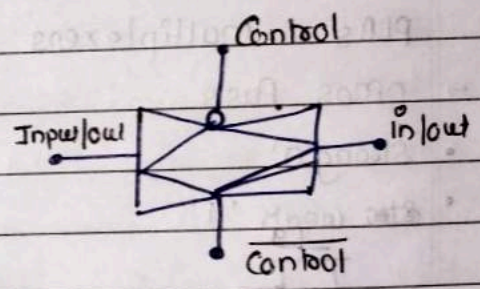
→ Transmission gates act as Bi-directional switch between two node A & node B controlled by signal C.

→ Gate of NMOS is connected to C & gate of pmos is connected to Cbar. when control signal C is high i.e VDD both transistor are on and provides a low resistance path Between A & B.

→ on the otherhand, when C is low, both transistor are turned off & provide high impedance path Between A & B.



Transmission gate



Graphic Symbol

Sol^n ③ what is power dissipation?

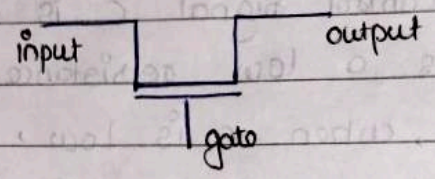
- Power dissipation is the supplied power required to operate the desired logic function.
- This parameter does not include the Power delivered from another gate.
- for ex: an ic with four gate will require, from its power supply, four times the power dissipated in each gate.
- Period time is the electrical energy used by the logic ckt. it is expressed in milliwatts or nanowatts.

Power dissipation = Supply vtg * mean current taken from Supply.

Sol^n ③ what is a pass transistor logic circuit ? Advantages of using pass transistor logic.

∴ Pass transistor logic (PTL) is a type of digital logic design that uses pass transistor (also know as transmission gates or pass gates).

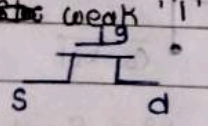
- PTL uses a nmos or pmos transistor to transfer charge from input node to output node, under the control of Gate voltage.
- The output remains in High impedance state when gate voltage is zero



- Pass transistor circuits are mostly used in design of ROMs, PLAs, multiplexers etc.

→ nmos Pass

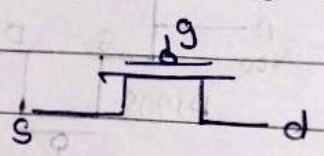
- Strong '0'
- weak '1'



when g=1 input '0'
O/P strong '0'

∴ g=1 input 1
O/P weak '1'.

pmos Pass



Strong '1'
Weak '0'

when g=0 input '0'
O/P weak '0'

g=0 input '1'
O/P strong 1

- Advantage :-
- The advantages of high functionality with few pass transistor & of small I/P capacitance in the CPL style & less wiring overhead make it better choice
- Thus CPL reduce the circuit delay, & less number of Components reduces the area of layout up to 57% approx & Power Consumption of the CKT.

Sol^n ④ why are most interrupts Active low?

∴ if we consider level of a module. active low means the capacitor in the output terminal gets changed or discharged based on low to high & high to low transition respectively.

- when it goes from high to low it depends on the pull down register that pulls it down & it is relatively easy for the output capacitance to discharge rather than changing. hence people prefer using active low signals.

Sol^n ⑤ Define pair, Quad and octet?

- pair :- Two adjacent cell of Karnaugh map (Kmap) is called as pair. it cancels one variable in a K-map.

- Quad :- A pair of four adjacent pairs in a K-map is called a quad. it is cancel two variable in a K-map specification.

- Octet : A pair of eight adjecent pair in a K-map is called an octet. it cancels four variable in a Kmap simplification

Sol^n ⑥ Define fan-in and fan-out?

- fan-in :-

The fan-in of the gate means that the number of Inputs that are connected to the gate without the degradation of the voltage level of the system.

- Fan-out : The fan-out is the maximum number of same inputs of the same IC family that a gate can drive maintaining its output levels with the specified limits.

Sol^n ⑧ what is metastablity ? what are its effects ?

∴ whenever there is Setup and hold time violations in a flip flop it enters a state where it is output is unpredictable. This state of unpredictable output is known as metastable state.

- The effect of metastability can be problematic for the proper operation of digital systems. if metastablity state persist for a long, it can lead to incorrect data being latched.
- Circuit does not meet timing Constraints.
- The output will have non-deterministic behavior.
- The output of the clocked pass gate does not change properly.

Sol^n ⑨ what is an Arbiter ? Explain it's operation ?

∴ Arbiters are electronic devices that allocate access to shared resources.

- Typical shared resources are buses, memories & mutipliers.
- Arbiter circuit may be synchronous or asynchronous.

There are two main arbitration

① Static priority - based on input port number.

② Round Robin - based on last user.

DAY 25 – 100 DAYS VERIFICATION CHALLENGE

Topic: MOSFET, BJT, CMOS

Homework: Interview questions on MOSFET, BJT, CMOS

DAY 25 CHALLENGE:

1. Why do the present VLSI circuits use MOSFETs instead of BJTs?
2. What are the various regions of operation of MOSFET? How can we use these regions?
3. What do you understand by the threshold voltage?
4. What does "the channel is pinched off" mean?
5. What are the key differences between the TTL chips and CMOS chips?
6. What is the most significant advantage of the CMOS chips over the TTL chips?
7. What do you understand by Channel-length Modulation?
8. What is the depletion region in VLSI?
9. What are the various factors that can affect the threshold voltage?
10. What is the reason behind the number of gate inputs to CMOS gates usually limited to four?

Copyright @Sweety Pinjani

Arti_tyagi_digital_2023

DAY : 25

Topic : MOSFET, BJT, CMOS

Sol^n ① Why do the present VLSI circuits use MOSFETs instead of BJTs?

∴ The advantage of using MOSFETs over BJTs is that power dissipation and leakage currents are minimized in the circuits, so they have been used instead.

→ in present VLSI circuits use MOSFETs instead BJTs due to the following reasons.

- MOSFET are small in size and occupy a minimal silicon area on IC chips as compared to BJTs.
- MOSFET are relatively simple in terms of manufacturing as compared to BJTs.
- The package density of MOSFETs is high.
- MOSFET are unipolar devices that control voltage. BJT are polar devices used to direct current.
- BJT is a 3 terminal device which are emitter, collector and Base while MOSFET is a four terminal device mainly source, gate, drain and substrate (also known as Body) and can be usefull for maintaing low leakage currents.

Sol^n ② What are the various regions of operation of MOSFET? How can we use these regions?

∴ MOSFETs operate in three regions :

① The cut-off region ② The triode region ③ The Saturation Region

∴ The cut-off region and the triode region are used to operate as a switch & the saturation region is used to operate as an amplifier.

- cut-off region: when $V_{GS} < V_t$, no channel is induced & The MOSFET will be cut-off region. No current flow.
- Triode region: when $V_{GS} \geq V_t$, a channel will be induced & current starts flowing. if $V_{DS} > 0$, MOSFET will be triode region as long as $V_{DS} < V_{GS} - V_t$.
- Saturation region: when $V_{GS} \geq V_t$, and $V_{DS} \geq V_{GS} - V_t$, the channel will be saturation mode. where the current saturates.

Sol^n - (3) what do you understand by the threshold voltage?

∴ The term "Threshold voltage" refer to a specific voltage that, depending on the device, cause a particular phenomenon to occur.

- The minimum voltage required to ON the transistor is called Threshold Voltage. It is observed as positive value for NMOS transistor and negative value for PMOS transistor.
- After the threshold voltage, a large quantity of current to flow.

Sol^n - (4) what does "The channel is pinched off" mean?

∴ For a MOSFET when VGS is greater than Vt, (means when the voltage btw gate and Source (VGS) is greater than the threshold voltage) The channel is induced, As we increase, VDS current starts following from Drain to Source till the voltage btw gate and channel at the drain end become Vt, ie: VGS − VDS = Vt, The channel depth at Drain end decrese almost to zero. at this stage, the channel is said to be pinched off. in this condition, MOSFET enters the saturation region.

Sol^n - (5) what are the key difference btw The TTL chips & CMOS chips?

TTL chips	CMOS chips
(1) Transistor-Transistor logic or TTL, uses bipolar junction transistor	(1) Complementary metal oxide semiconductor (CMOS) used field effect transistor
(2) The density of TTL is comparatively less.	(2) There is a higher density of logic gates in CMOS.
(3) TTL chips operate slower than CMOS	(3) work at a faster speed.
(4) TTL chips consume more energy. A single gate in TTL consume mV.	(4) It consume less power. one single CMOS chip utilize 10nW of Power.
(5) TTL chips generally used in computer.	(5) CMOS chips are used in mobile.

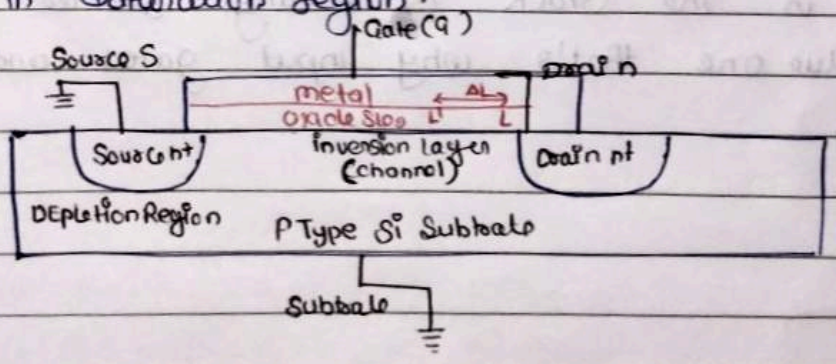
Sol^n ⑥ what is the most significant advantage of the cmos chips over the TTL chips?

∴ The main advantage of cmos as compared to TTL is low consumption & better noise margins.

- cmos chip over the TTL chip is that the cmos chip has higher density of logic gate within the same material that improve its efficancy.
- cmos chips consume less power than the power consumed by the TTL chips.

Sol^n ⑦ what do you understand by channel-length modulation?

- In saturation region of working with MOSFET, There is channel length modulation with MOSFET.
- In that channel length will change with respect to drain voltage of MOSFET.
- channel length will decrese with respect to increse in drain voltage in saturation region.



Sol^n ⑧ what is the depletion region in VLSI?

∴ when we apply a positive voltage across Gates, it cause the free holes (positive charge) to be repelled from the substrate region under the channel region. when these holes are pushed down the substrate, they create a carrier depletion region.

Sol^n (9) what are the various factor that can affect the threshold voltage?

Threshold voltage depends on the following parameter

- Gate material
- Gate insulator material
- Channel doping
- Gate insulator thickness
- Voltage btw Source & Subtrate
- Temprature
- Length of channel, threshold vtg decrese as the channel length decrese

Sol^n (10) what is the reason behind the number of gate input to cmos gates usually limited to four?

∴ The number of gate inputs to cmos gates usually limited to four because a higher number of stacks make the gate slower. In NOR & NAND gates, the number of gates present in the stack is usually like the number of input plus one that's why input gates are restricted four