Last Lectures :-

- System design blow
- Basic Logic gates & HDL
- Combinational Logic block
  &
  ALU Design

Today's Lecture :-

- Sequential design elements :-
  - Latch
  - Flip-flops
  - Registers
  - Memory
  - Counter
- Timing Concept-

Need of Sequential Design :-

Example

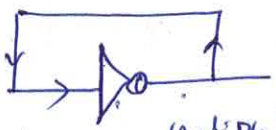
(1)

```
a[10] = {0,1,2,3,4,5,6,7,8,9};
for (i=0; i<=9; i++) {
    odd = odd + a[2*i+1];
    eve = eve + a[2*i];
}
```

(2) $f(n) = f(n-1) + f(n-2) + c$ recursion

② Memory element - Design :-

- Representation
- Feedback



input output

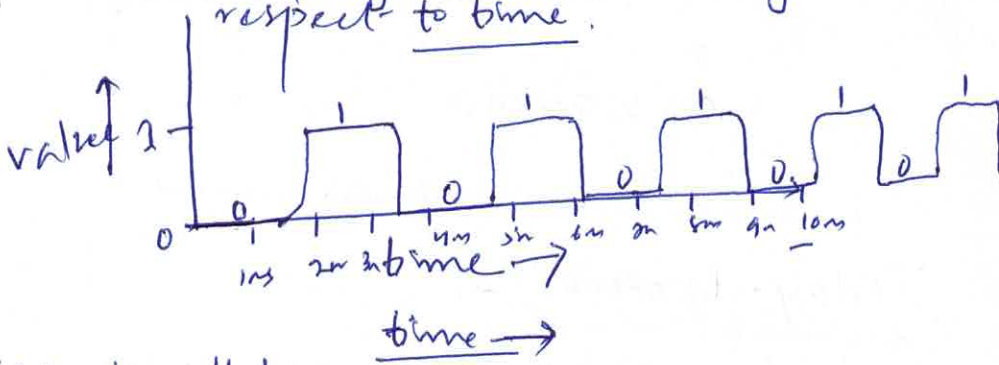
1 nsecond delay

0.5 ns

1.5 ns

Timing diagram :-

Mapping of functionality with respect to time.

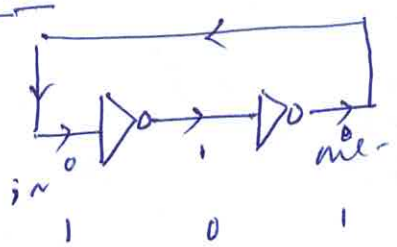


time →

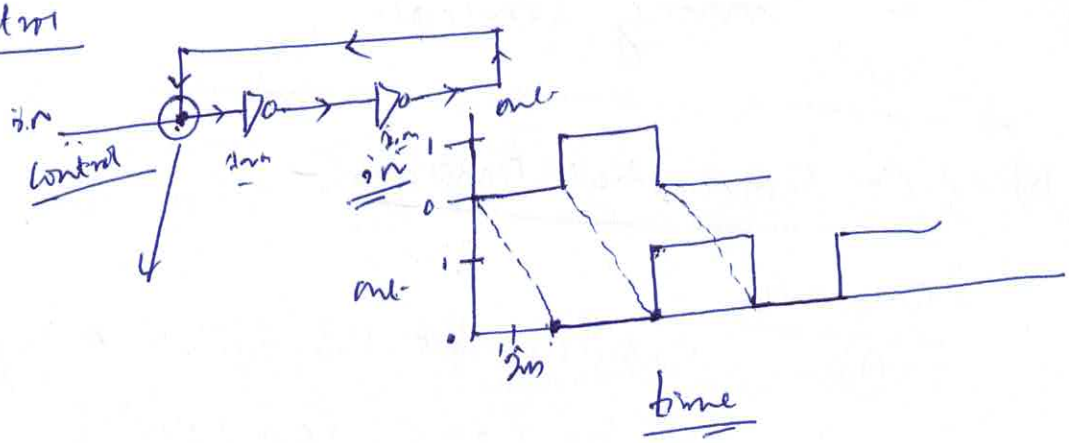
→ This kind of design is called

oscillator

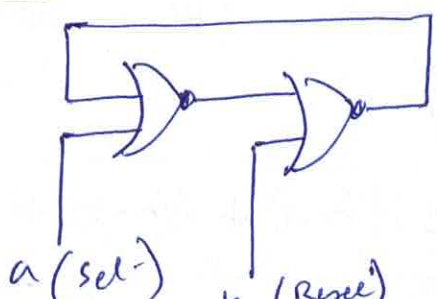
Two inverters :-



Adding control

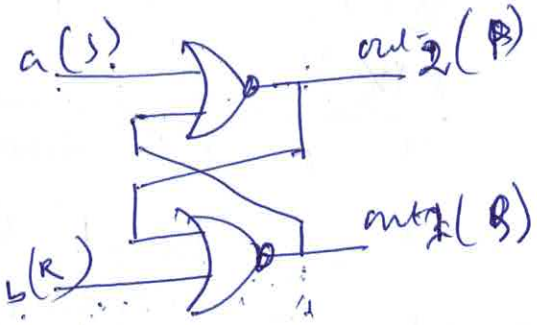


- Design with XNOR :-



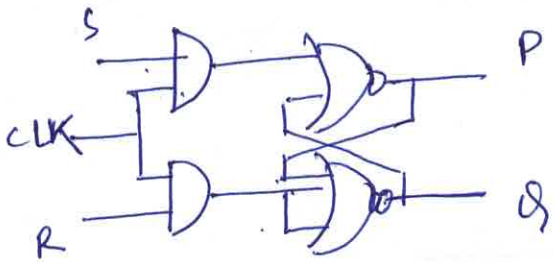
Check with OR gate, to see the limitations

Function table of NOR based feedback logic :-

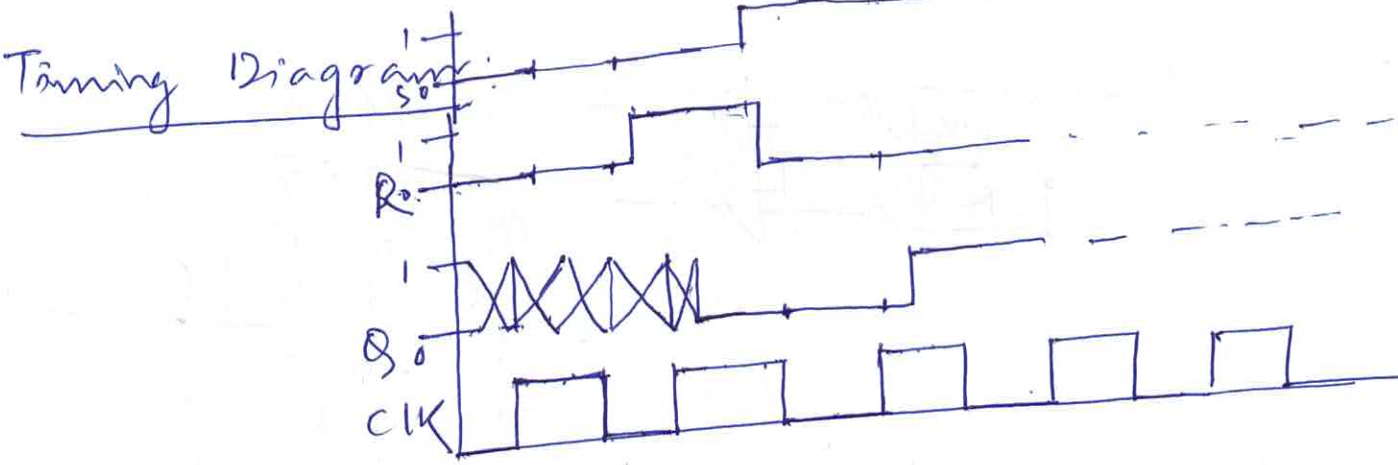


a	b	out 1 (Q)	out 2 (P)
0	0	previous output	
0	1	0	1
1	0	1	0
1	1	invalid state	

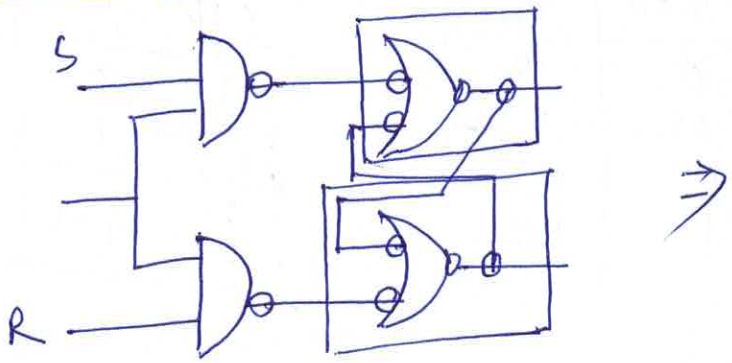
Clock controlled S-R Latch



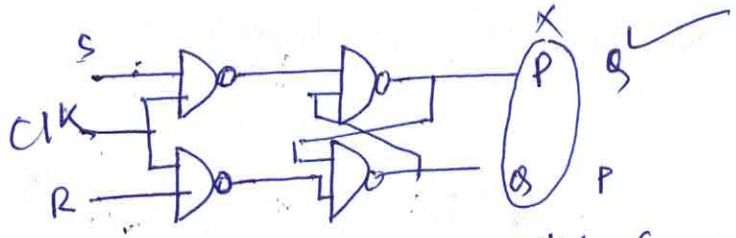
Timing Diagram :-



Nand gate based Design :-



(4)

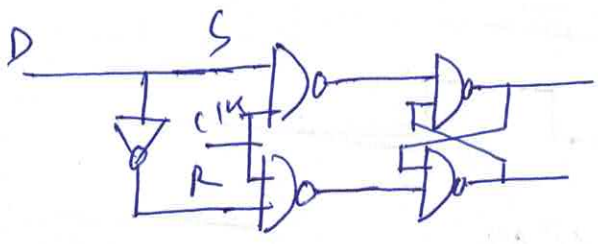


CLK = 0 → ~~previous~~ Memory State

CLK = 1, S R State

CLK	S	R	Q	P.X
1	0	0	Memory	
1	0	1	1	0 ~~X~~
1	1	0	0	1 ~~X~~
1	1	1	1	1 (invalid)

CLK	S	R	Q
1	0	0	
1	0	1	
1	1	0	
1	1	1	



S = D

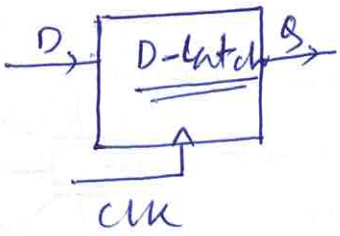
R = $\bar{D}$

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

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

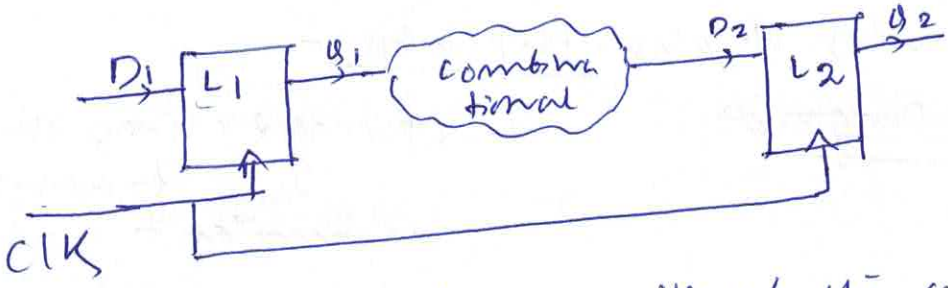
CLK	D	Q
1	0	0
1	1	1
0	0	previous value
0	1	

- Flip-flop Design:-



Limitation of latch -

- transparency.



D_2 gets overwritten by the current D_1

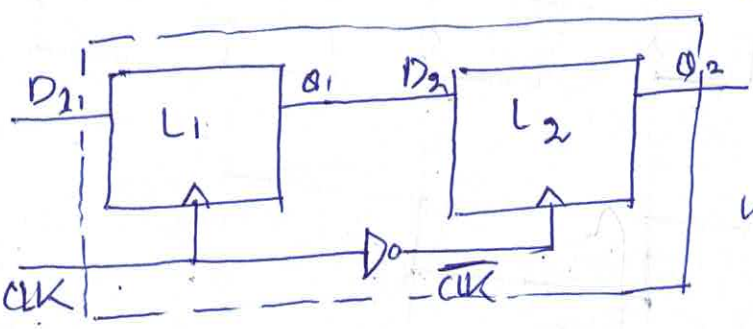
$$Q_2(t) = D_2(t-1)$$

$$Q_1(t) = D_1(t-1)$$

~~$Q_2(t) = f(Q_1(t), comb)$~~

$$Q_2(t) = f(Q_1(t), comb) \rightarrow \text{wrong}$$

- Master-Slave flip-flop Design:-



when L_1 is transparent. L_2 should be opaque

when L_1 is opaque L_2 should be transparent

Triggering of end-pod:-

- Negative edge trigger
- positive edge trigger

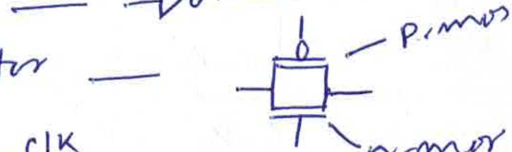
(6)

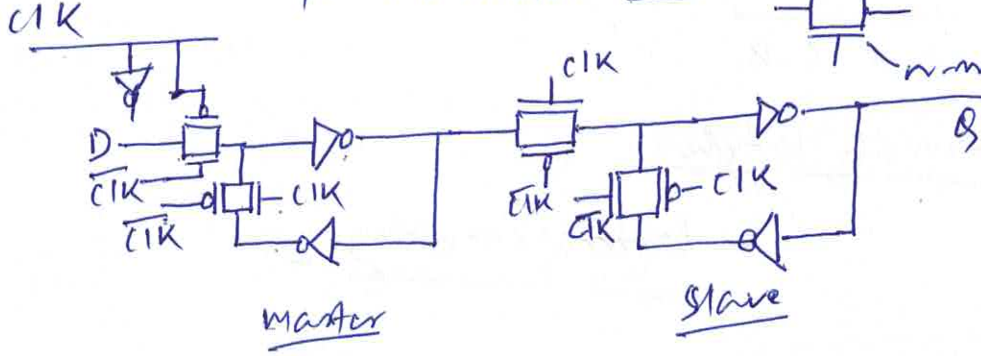
The modern flip flop Design

Component used -

— Inverter

— pass transistor



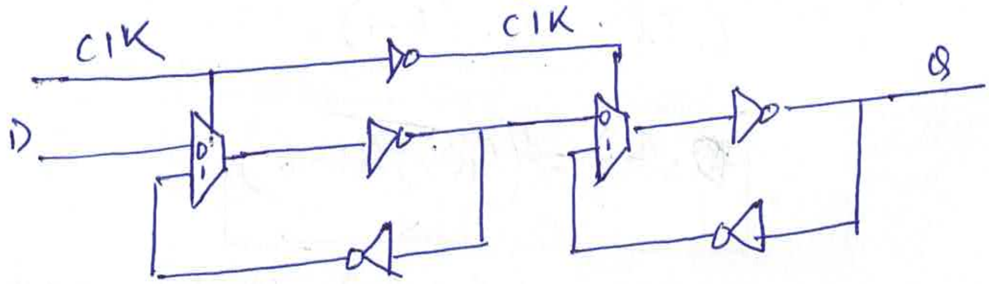


functionality of above flip-flop :-

~~Alternative Design~~

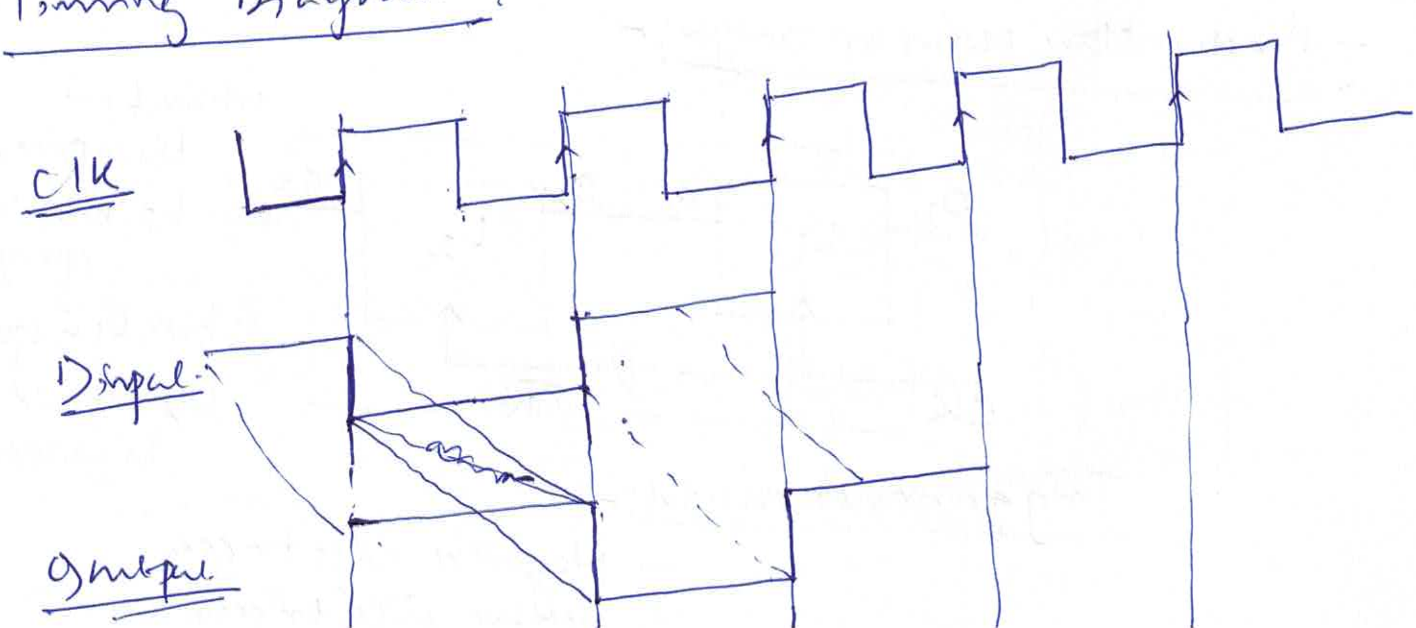
— opening & closing of pass transistor is similar to multiplexer

Alternative Design (Multiplexer based D-flip flop Design)

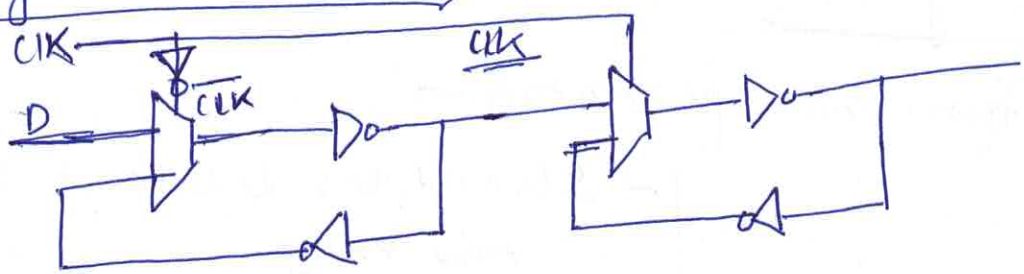


Timing Diagram :-

the flip-flop is positive edge triggered

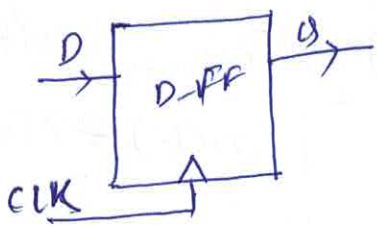


Negative edge triggered :-

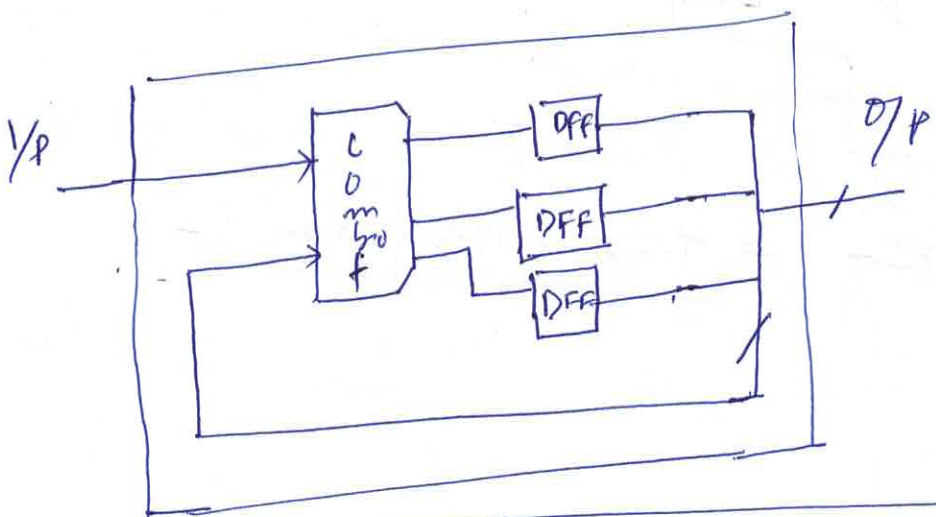


- change in output will be reflected in negative edge

The D-flipflop :-



Nature of Sequential Design :-



$$State(t) = f\left(State(t-1), input(t)\right)$$

1-bit Register :-

Functional specification —

- Should be able to load the new value
- input
- output

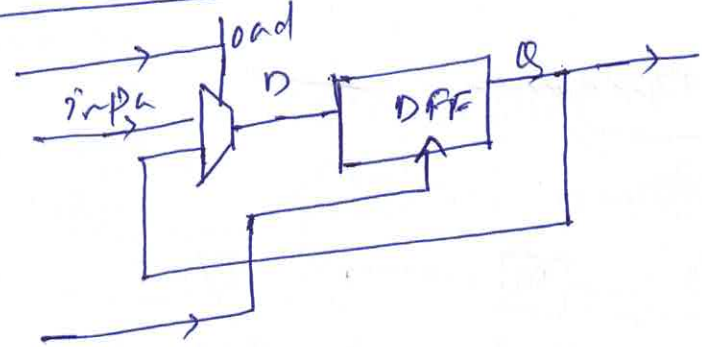
if load(t) then out(t+1) = in(t)

else out(t+1) = out(t)

if load(t-1) then out(t) = in(t-1)

else out(t) = out(t-1)

Design using DFF :-



~~HDL code~~ :- → Timing Diagram :-

→ HDL code :-