This document is created by Sweety Pinjani. All rights are reserved by the original author.
DO NOT DISTRIBUTE

DAY 1 - 111 DAYS VERIFICATION CHALLENGE

Topic: Flip Flop & Latches

Skill: Digital Electronics

DAY 1 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.

©2024 SwitiSpeaksOfficial. All rights reserved.

Author – Sweety Pinjani

III day Verification

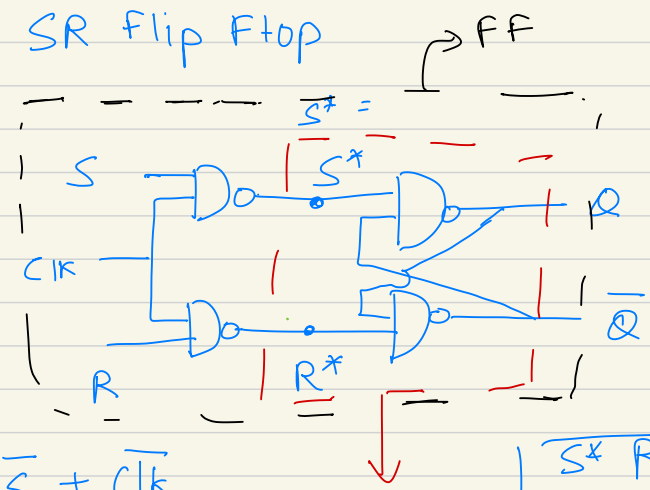
31 Aug.

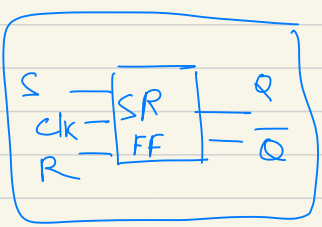
(Credits: Swaety Pinjani)

Day 1

Q1) Explain functioning of JK & SR Flip Flop.

SR Flip Flop





$S^* = \overline{S} + \overline{Clk}$

$R^* = \overline{R} + \overline{Clk}$

Latch

Latch

S^*	R^*	Q	$\overline{Q}$
0	0	Not used	
0	1	1	0
1	0	0	1
1	1	Memory	

SR FF Truth table

Clk	S	R	Q	$\overline{Q}$
0	X	X	1	0
1	0	0	memory	
1	0	1	0	1
1	1	0	1	0
1	1	1	Not used	

NAND T.T

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

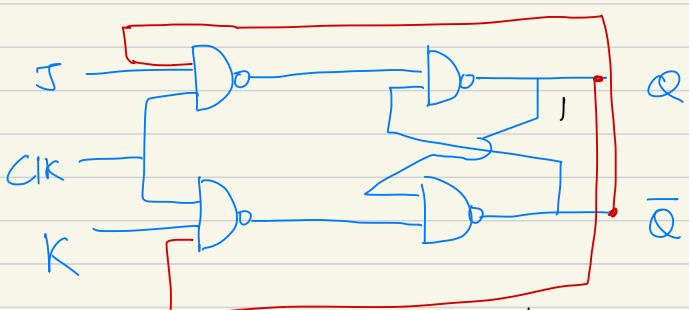
In the above flip flop we use the control clk $\overline{SN}$ additional to the latch (SR) to make it a flip flop; which controls o/p.

Latch → memory unit (reqd for sequential logic)

* on adding the control $\overline{SN}$ it becomes a FF.

JK Flip flop

JK flip flop is just an extension of SR flip flop but with the '1' o/p case working.



Clk	J	K	Q	$\overline{Q}$
0	x	x	Q_n (Memory)	
1	0	0	Q_n (Memory)	
1	0	1	0	1
1	1	0	1	0
1	1	1	$\overline{Q}_n$	

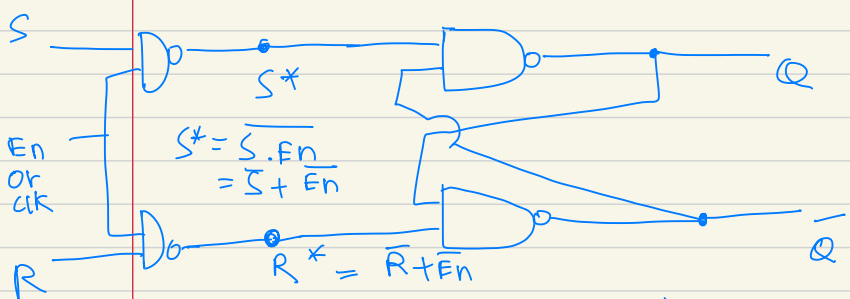
SR latch

S	R	Q	$\overline{Q}$
0	0	Not used	
0	1	1	0
1	0	0	1
1	1	Memory	

Q2) Difference between latch & Flip flop?

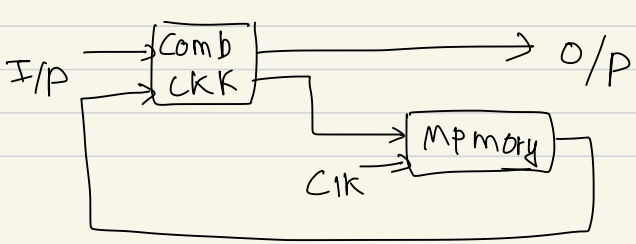
The **control input** to the SR latch or Flip Flop decides whether it would be a latch or Flip flop.

In order to control a latch or FF, control S/N can be given (or the bits can flip otherwise).

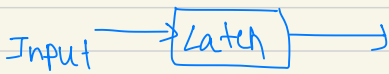


The way in we give the control signal determines whether it is a latch or F/F. The only thing that determines if it is a latch or F/F is the control input. If the control i/p is edge triggered then it is FF; otherwise if it is level sensitive then it is latch.

When Control S/N is enable then whenever it is in high state then it is active.

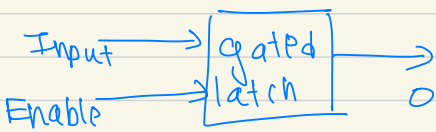


Latches are level sensitive & only when the



(transparent latch)

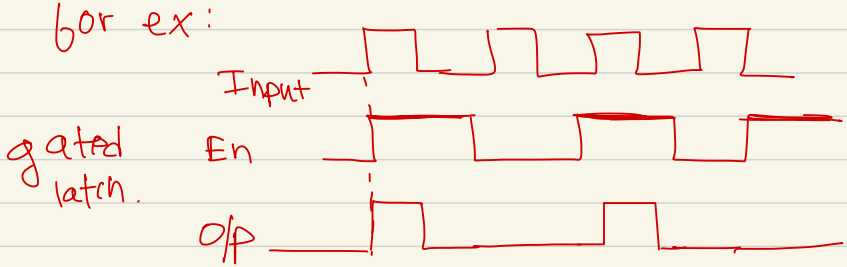
: Immediately responds to I/P change.



(Gated latch).

gated latch responds to the input only when the enable S/N is high (for +ve level triggered).

for ex:



Flip flop: It is edge sensitive.
∴ only on +ve edge trigger does the values change (So whatever value is there at the +ve edge is transferred to o/p when +ve edge).

3) Why are latches faster than FF.

Latches are faster ∵ they respond to the input change continuously as soon as the i/p change o/p changes whereas in flip flop this happens only on edges.

4) Use of FF & Latch

Flip flops are used for:

① data storage & Syncronizatn.

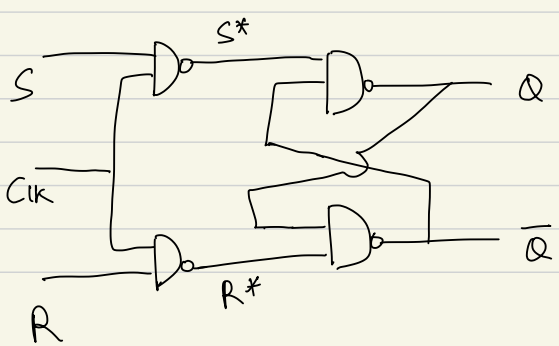
② Timing Control.

Latches are used for

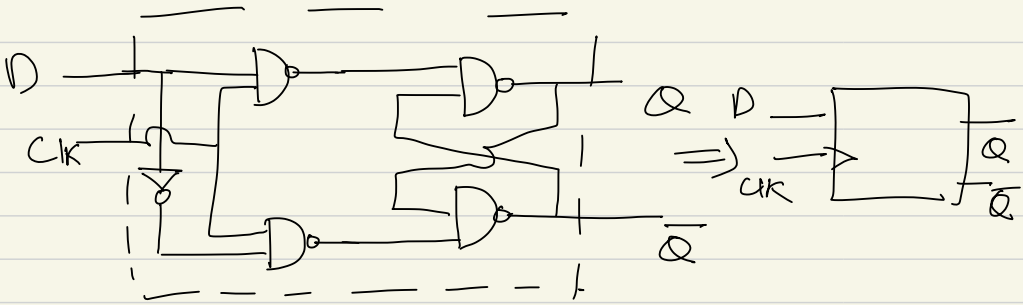
① data storage.

② Building blocks for flip flops.

5) Why is gated SR flip flop called asynchronous latch.



6) Implement D-flip flop using NAND



Clk	S	R	Q	$\overline{Q}$
0	x	x	memory	
1	0	0	memory	
1	0	1		
1	1	0		
1	1	1		

⇓

Clk	D	Q	$\overline{Q}$
0	x	memory	
1	0	0	1
1	1	1	0

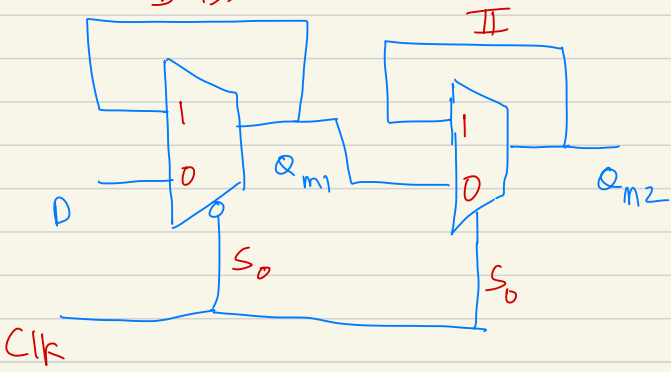
7) Design D-FF using 2:1 mux.

(we are using a combinational ckt to build sequential ckt). ⇒ For this we usually go for master-slave appr.

S_0	I_0	I_1	X	
0	0	X	0	$\{I_0\}$
0	1	X	1	
1	X	0	0	$\{I_1\}$
1	X	1	1	

Construct a D-FF (using 2 muxes).

I ⇒ when clk i/p = 1 (retains value.



S_0	D	Out
0	X	npm
1	0	0
1	1	1

The o/p should just send the data forward in case I is getting i/p(D); otherwise if I is storing then II should also store the value.