CS221 : Digital Design Scribe

Date: 03/10/2013

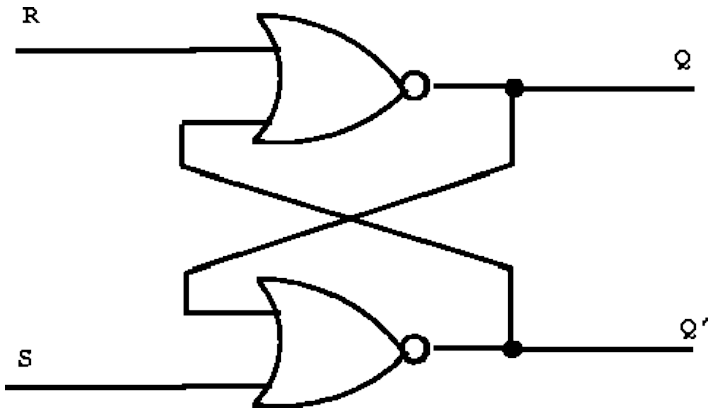
By:-

Shreyansh Sharma(120123034)

Shubham Goyal(120123035)

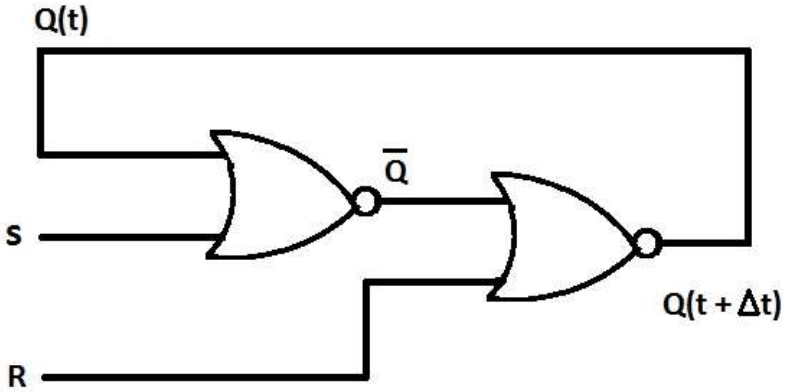
Ujjwal Kumar Kansal(120123044)

Cross Coupled NOR-Gate (RS Flip-Flop)



R	S	Q	Q'
0	1	1	0
1	0	0	1
0	0	No Change	No Change
1	1	Invalid Input	Invalid Input

Cross-Coupled NOR Gates

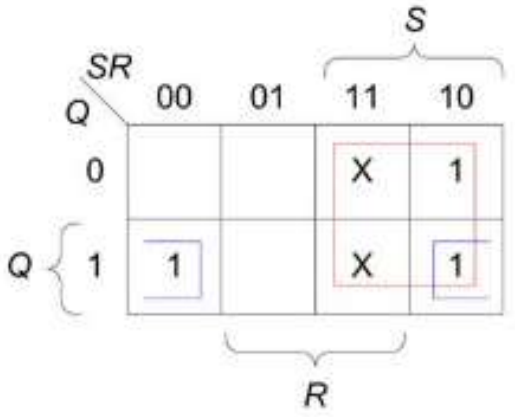


R(t)	S(t)	Q(t)	Q(t+Dt)
0	1	0	1
1	0	1	0
0	0		Q(t)
1	1	Invalid Input	Invalid Input

Cross-Coupled NOR Gates-Alternative

$$Q(t+\Delta t) = S(t) + R'(t)\, Q(t)$$

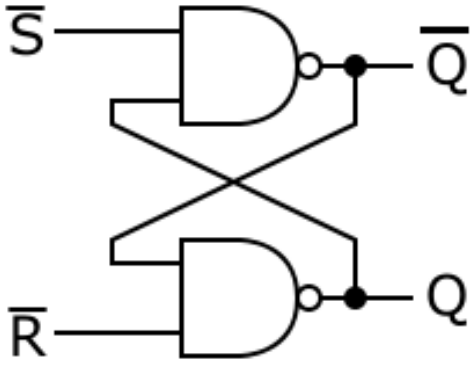
Derivation by K-Map



Q(t+Dt) = S + R'Q and RS=0

1. **X is the Don't Care condition used because R = 1 & S = 1 is an invalid input.**
2. **Empty Space stands for 0's.**
3. **R, S & Q stands for R(t), S(t) & Q(t) respectively.**

RS Flip-Flop using NAND Gates

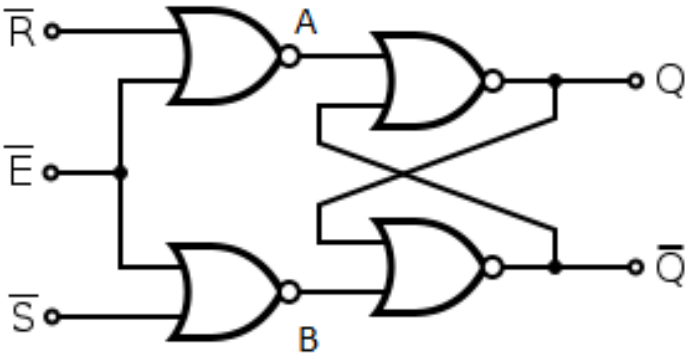


S	R	S'	R'	Q(n+1)
0	1	1	0	1
1	0	0	1	0
0	0	1	1	Q(n) (Hold)
1	1	0	0	Not Allowed

RS Flip-Flop using NAND Gate

NAND-Gate works as an inverter when anyone of the input for it is 1. This can be explained from the fact that AB = A if B = 1 and complement will be A', therefore on the whole it works as an inverter if one input (in this case B) is 1.

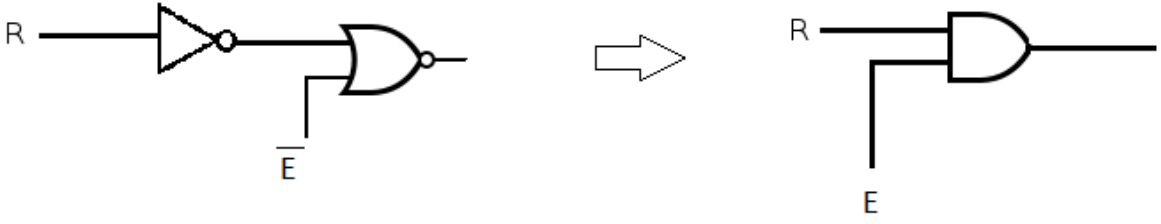
If we want to choose a particular value to be stored in memory element then following circuit is used.



1. **If E = 1, E’ = 0, then output at gates A and B will always be R and S respectively, therefore final output depends on R and S.**

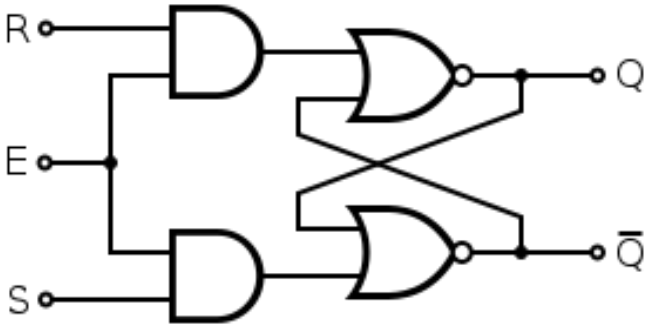
2. **If E = 0, E’ = 1, then output at gates A and B will always be E = 0,irrespective of R and S, therefore final output is independent of R and S and value in the memory element will be hold.**

Two Gates NOT and NOR can be replace by a single AND gate by using the following transformation.



AND-Gate Equivalent of the NOR-Gate component

After applying above transformation to the circuit



E = Clock

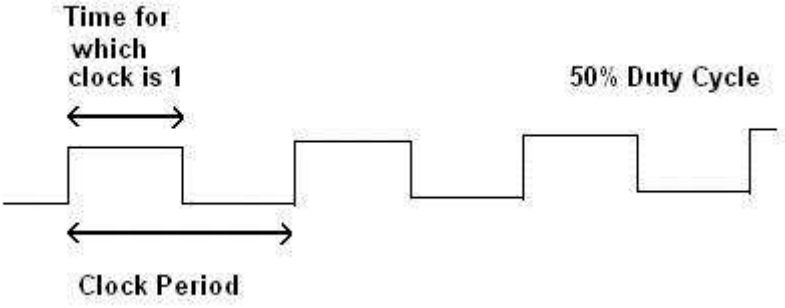
Clock = 1, E = 1, Output depends upon R and S

Clock = 0, E = 0, Output is independent of R and S, and therefore values hold.

Duty Cycle

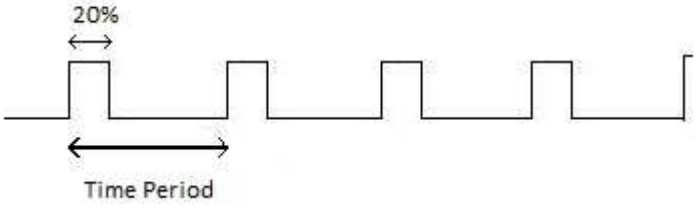
Duty Cycle is defined as the percentage of time when RS Flip-Flop needs to be stable. Hence, it can be calculated as the percent of time that an entity spends in an active state as a fraction of the total time under consideration. Duty Cycle is kept minimum so that circuit can have maximum time to compute the next value when CLK=0.

Duty Cycle = (Duration for which circuit is active / Period of the function)



Waveform for 50% duty cycle

Similarly, 20% Duty Cycle can be represented as

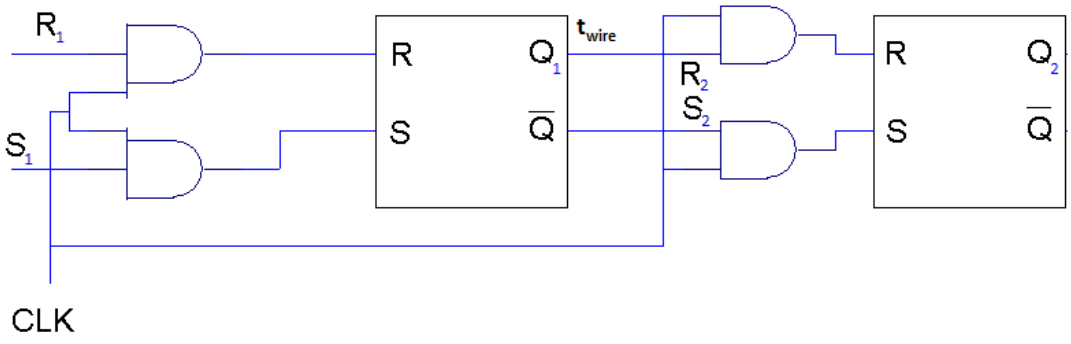


Waveform for 20% duty cycle

Time Delay

The propagation delay from starting to end caused by several components in the circuit i.e. gates, wires etc. is called time delay (t_d).

Connection of two RS latches



t_{wire} = time delay due to connecting wire between two flip-flop

Consider,

1. **Initially, CLK = 0, suppose stored values in both latches at Q_1 and Q_2 be a and b respectively i.e. $Q_1 = a$ and $Q_2 = b$.**

2. **When value of clock gets high i.e. CLK = 1, both flip-flops opens. Let us suppose input at R_1 be c i.e. $R_1 = c$.**

3. **The value at Q_1 and Q_2 changes accordingly which let us suppose are d and e respectively i.e. values become $Q_1 = d$ and $Q_2 = e$.**

4. **Now again when clock becomes low i.e. CLK = 0. If, both latches closes at the same time then circuit will always generate correct output. But it may happen that sometimes first latch closes sometime before second latch which may cause output at Q_1 to enter second latch and produces wrong output.**

5. Such a situation can be encountered if we have a large t_{wire} delay then output at Q_1 takes some time to reach second latch and we will get correct output.

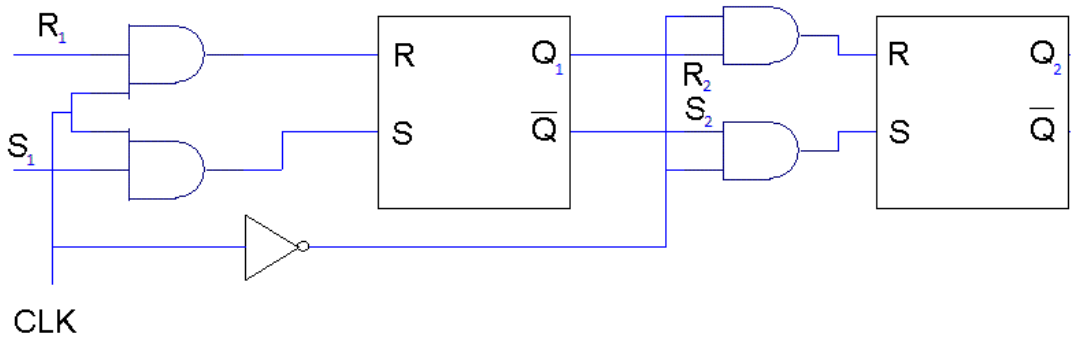
6. Therefore,

$$t_{wire} > t_d >> D$$

t_d = time delay of the circuit
D = Duty Cycle

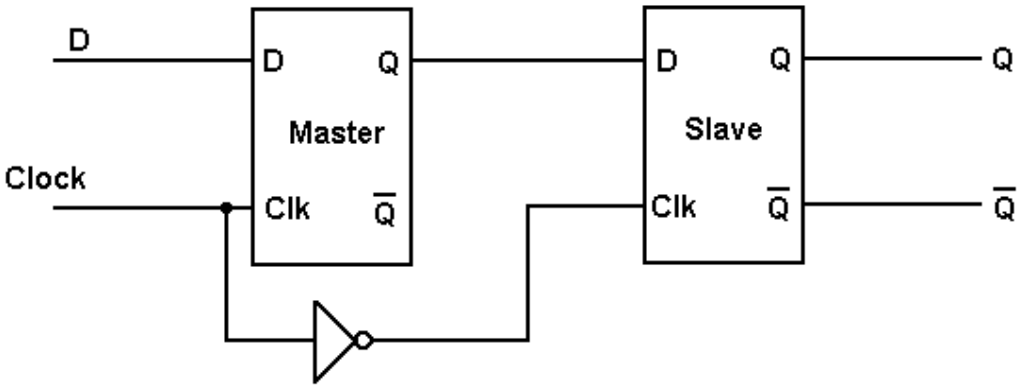
Permanent Solution

The above solution in temporary solution because it depends on t_{wire} which means we may have to alter the length of the wire to large lengths.



But, if we apply an inverter before clock input of second latch then either one of the both latches works at a time and hence the above problem gets solved.

Master-Slave Latch



A Master-Slave flip-flop is built with two latches – a Master latch followed by a Slave latch.

The Master Latch responds directly to the inputs while the CLK is HIGH – during this time the Slave is disconnected from the Master and stays in the same state.

When CLK is LOW, the Master Latch is disconnected from the input and its output drives the Slave Latch. During this time, the Master will not change state.