CSE 2105
Digital Logic Design

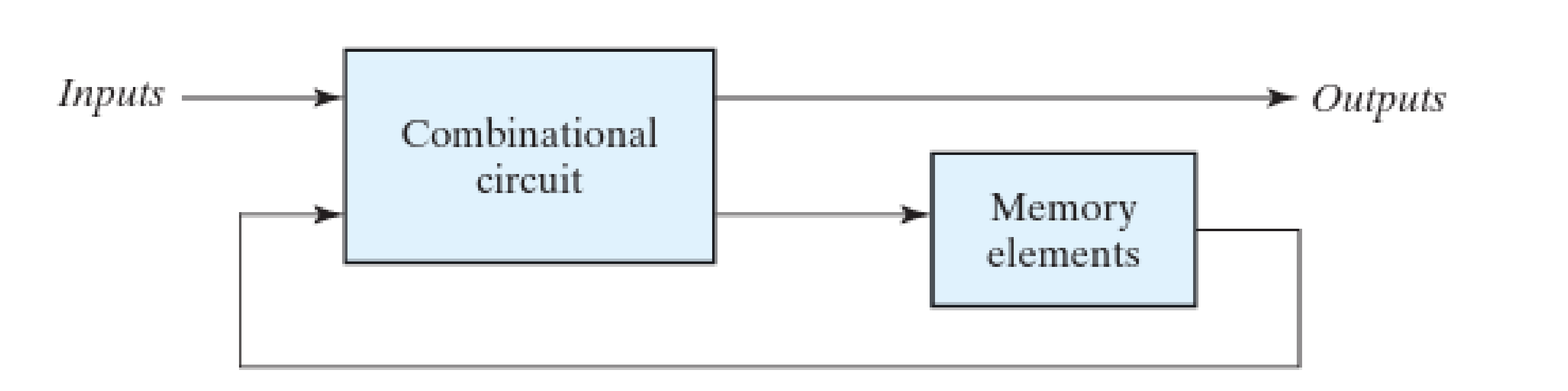
LOGIC OF LATCHES

Sequential Logic Circuit

- A sequential logic circuit consists of a combinational circuit to which storage elements are connected to form a feedback path.
- The storage elements are devices capable of storing binary information.
- The sequential circuit receives binary information from external inputs that, together with the present state of the storage elements, determine the binary value of the outputs. These external inputs also determine the condition for changing the state in the storage elements..
- Most digital systems consist of both combinational circuits and memory elements.

Sequential Logic Circuit

- The following figure shows a block diagram of a general digital system that combines combinational logic circuit with memory devices. The combinational portion accepts logic signals from external inputs and from the outputs of the memory elements. The combinational circuit operates on these inputs to produce various outputs, some of which are to be stored in the memory elements.
- The block diagram demonstrates that the outputs in a sequential circuit are a function not only of the inputs, but also of the present state of the storage elements.



Differences between Combinational & Sequential Logic Circuits

Combinational Logic Circuits

1. The output depends entirely on the present input.
2. No feedback is present between the input and output.
3. Exhibits a faster speed
4. Comparatively easier to design
5. Memory element doesn't exist
6. Example – Demultiplexer, Multiplexer, Decoder, Encoder, etc.

Sequential Logic Circuits

1. The output depends on both present input and past output values
2. Feedback exists between the input and output
3. Exhibits a comparatively slower speed
4. Comparatively tougher to design
5. Memory element is present
6. Example – Counter, Registers etc.

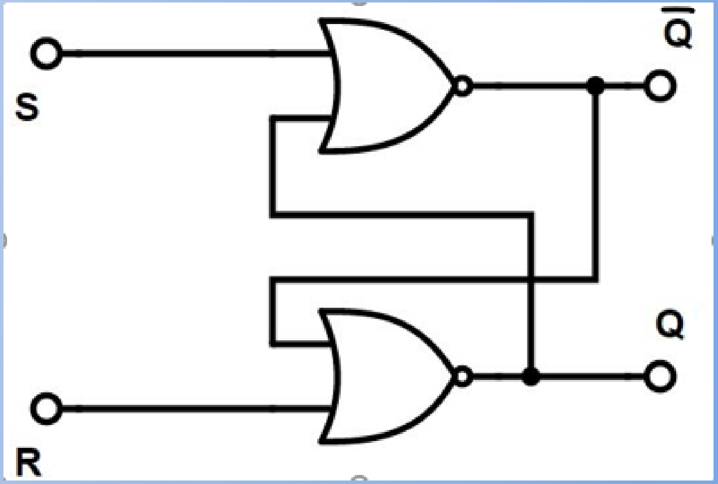
Storage element

- A storage element in a digital circuit can maintain a binary state indefinitely (as long as power is delivered to the circuit), until directed by an input signal to switch states.
- The major differences among various types of storage elements are in the number of inputs they possess and in the manner in which the inputs affect the binary state.
- Storage elements that operate with signal levels (rather than signal transitions) are referred to as latches
- Those controlled by a clock transition are flip-flops.
- Latches are said to be level sensitive devices; flip-flops are edge-sensitive devices.
- Latches and flip flops are related because latches are the basic circuits from which all flip-flops are constructed.

SR (Set-Reset) Latch with NOR gates

The SR latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates, and two inputs labeled S for set and R for reset. The latch has two useful states.

- When output Q=1 and Q’=0, the latch is said to be in the set state.
- When Q=0 and Q’=1, it is in the reset state. Outputs Q and Q’ are normally the complement of each other.
- However, when both inputs are equal to 1 at the same time, a condition in which both outputs are equal to 0 (rather than be mutually complementary) occurs.
- Consequently, in practical applications, setting both inputs to 1 is forbidden.
- Under normal conditions, both inputs of the latch remain at 0 unless the state has to be changed.

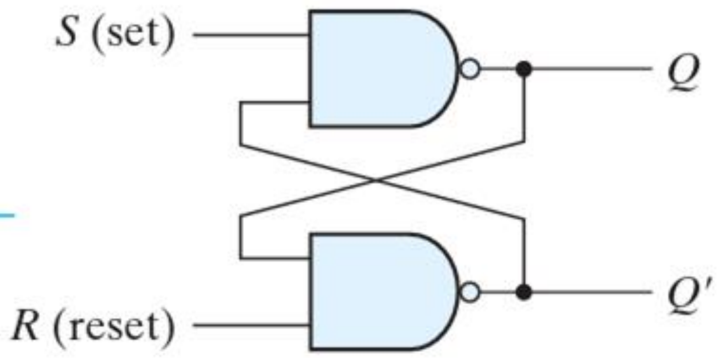


S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	(forbidden)

(b) Function table

SR (Set-Reset) Latch with NAND gates

- The SR latch with two cross-coupled NAND gates is shown in the figure.
- It operates with both inputs normally at 1, unless the state of the latch has to be changed. The application of 0 to the S input causes output Q to go to 1, putting the latch in the set state.
- When the S input goes back to 1, the circuit remains in the set state. After both inputs go back to 1, we are allowed to change the state of the latch by placing a 0 in the R input.
- This action causes the circuit to go to the reset state and stay there even after both inputs return to 1.
- The condition that is forbidden for the NAND latch is both inputs being equal to 0 at the same time, which should be avoided.



(a) Logic diagram

S	R	Q	Q'	
1	0	0	1	
1	1	0	1	(after $S = 1, R = 0$)
0	1	1	0	
1	1	1	0	(after $S = 0, R = 1$)
0	0	1	1	(forbidden)

(b) Function table

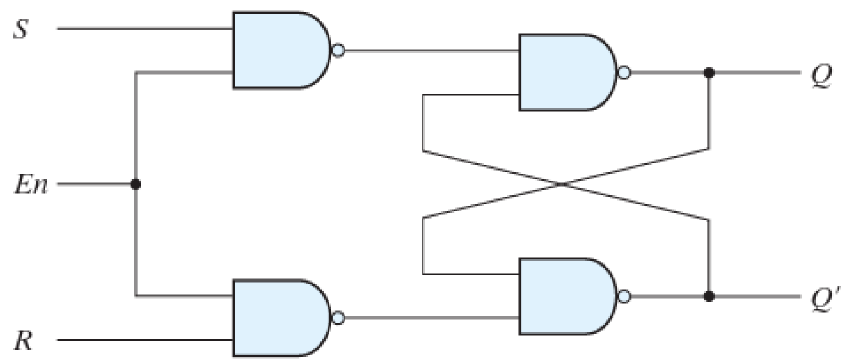
Set-Reset Latch with NAND gates

- In comparing the NAND with the NOR latch, note that the input signals for the NAND require the complement of those values used for the NOR latch.
- Because the truth table of NAND latch is complimentary to the one with NOR gates, it is sometimes referred to as an S’R’ latch.
- The operation of the basic SR latch can be modified by providing an additional input signal, En that controls when the state of the latch can be changed by determining whether S or R (or S and R) can affect the circuit.

Set-Reset Latch with NAND gates and Enable input

An SR latch with a control input is shown in the figure. It consists of the basic SR latch and two additional NAND gates. The control input En acts as an enable signal for the other two inputs.

- When the enable input goes to 1, information from the S or R input is allowed to affect the latch. When, S=1, R=0, and En=1 (active-high enabled), then the latch is in Set state (Q=1, Q'=0).
- To change to the reset state, the inputs must be S=0, R=1, and En=1. In either case, when En returns to 0, the circuit holds the memory.
- The control input disables the circuit by applying 0 to En, so that the state of the output does not change regardless of the values of S and R. Moreover, when En=1 and both the S and R inputs are equal to 0, the state of the circuit does not change.
- An indeterminate condition occurs when all three inputs are equal to 1.



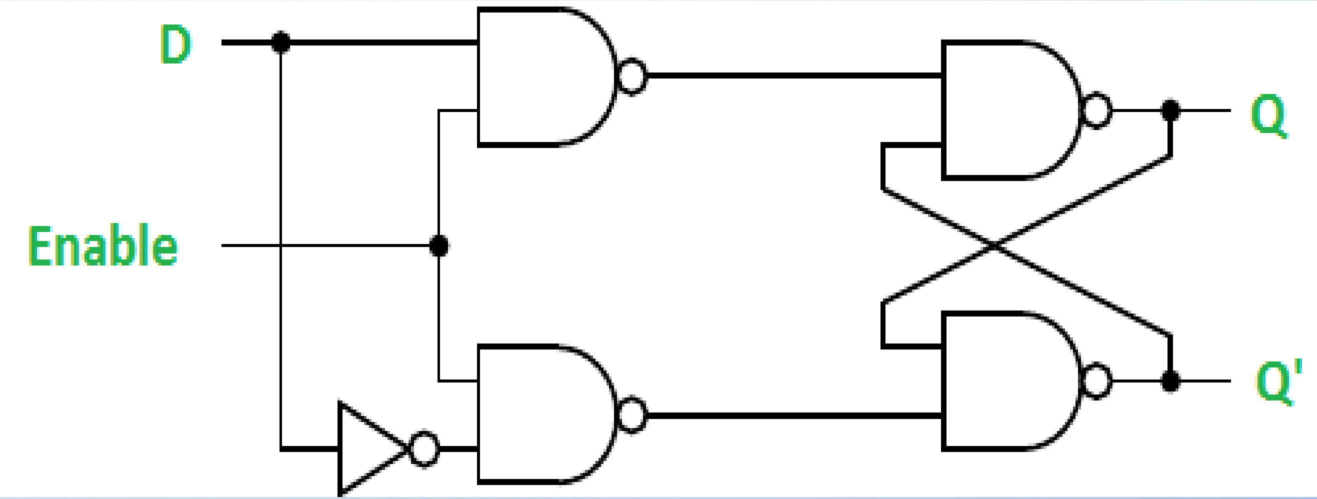
(a) Logic diagram

En	S	R	Q_n	Q_{n+1}	State
0	X	X	0 1	0 1	$Q_{n+1} = Q_n$ Hold/ No Change
1	0	0	0 1	0 1	$Q_{n+1} = Q_n$ Hold/ No Change
1	1	0	0 1	0 0	$Q_{n+1} = 1$ Set State
1	0	1	0 1	1 1	$Q_{n+1} = 0$ Reset State
1	1	1	0 1	0 or 1	Ambiguous/ Invalid Since $\overline{Q_{n+1}} = \overline{Q_n}$

(b) Functional Table

D (Data) Latch

- The 'D Latch' is a logic circuit most frequently used for storing data in digital systems. It is based on the S-R latch, but it doesn't have an 'undefined', 'indeterminate' or 'ambiguous' or 'invalid' state problem.
- A 'D latch' can store a bit value, either 1 or 0. When its Enable pin is HIGH, the value on the D pin will be stored on the Q output. It builds upon the design of the S-R latch, with a few added logic gates.



En	D	Next State of Q
0	X	No Change
1	0	Q = 0; Reset State
1	1	Q = 1; Set State

D (Data) Latch

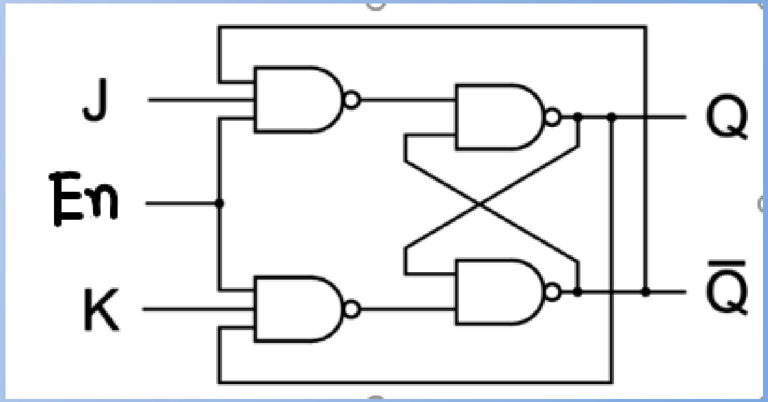
- The NOT gate on the input makes sure the S and the R inputs are always opposites, to avoid the invalid state of both being 1. The enable input, En (Enable) lets you control when you want to change the output to whatever is on the D input.
- This means that the output Q can only change when the enable signal is 1. If it's 0, the output is unaffected by any changes on D.

JK (Jack-Kilby) Latch with NAND gates

J-K latch is a modified version of an S-R latch with no ‘invalid’ or ‘illegal’ output state.

The first two-input NAND gates have been replaced with 3-input NAND gates, and the third input of each gate receives feedback from the Q and Q’ outputs.

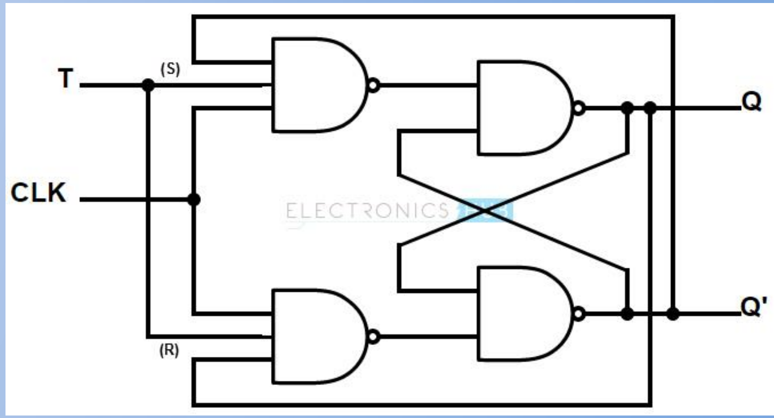
- J-K latch acts in a similar way as S-R latch for first 3 conditions. When both J and K inputs are 1, however, something unique happens.
- Let, previously, the latch was in Set state (Q=1, Q’=0).
- On the next clock pulse, the outputs will switch (‘toggle’) from set (Q=1 and Q’= 0) to reset (Q=0 and Q’=1). The end result is that the S-R latch’s ‘invalid’ state is eliminated and we get a useful feature as a bonus: the ability to toggle between the two output states.



En	J	K	Q_n	Q_{n+1}	State
0	X	X	0 1	0 1	$Q_{n+1} = Q_n$ Hold/ No Change
1	0	0	0 1	0 1	$Q_{n+1} = Q_n$ Hold/ No Change
1	1	0	0 1	0 0	$Q_{n+1} = 1$ Set State
1	0	1	0 1	1 1	$Q_{n+1} = 0$ Reset State
1	1	1	0 1	1 0	Toggle

T (Toggle) Latch with NAND gates

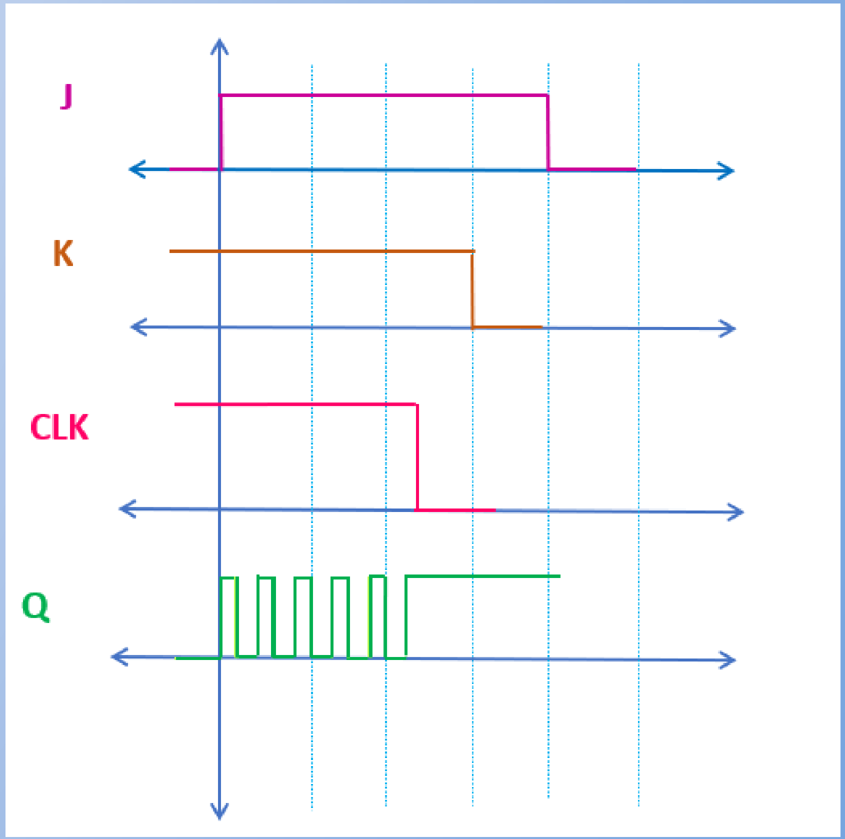
- A T latch is a digital circuit that can also store one bit of information. It has two inputs: a toggle input (T) and an Enable input (En).
- The T latch can be formed whenever the JK latch inputs are shorted. The function of T Latch will be like this: when the input of the latch is low, the latch will hold the previous output and when the input is high, the output will toggle.



Inputs		Outputs		Comments
E	T	Q_{n+1}	$\overline{Q}_{n+1}$	
1	0	Q_n	$\overline{Q}_n$	No change
1	1	$\overline{Q}_n$	Q_n	Toggle

Race Around Condition

- From the truth table of J-K latch, we know that when J, K and Enable are equal to 1, toggling takes place.
- Therefore, whenever Enable is equal to 1 there are consecutive toggling. This condition is called as Race around condition.
- To put it in words, “For JK flip-flop if J, K and Enable are equal to 1 the state of flip-flop keeps on toggling which leads to uncertainty in determining the output of the flip-flop.” This problem is called Race around condition. This condition also exists in T flip-flop since T flip-flop also has toggling options.



Solution of Racing Around Condition

There are three methods to eliminate race around condition as described below:

- **Increasing the delay of flip-flop**

The propagation delay should be made greater than the duration of the clock pulse (T). But it is not a good solution as increasing the delay will decrease the speed of the system.

- **Use of edge-triggered flip-flop**

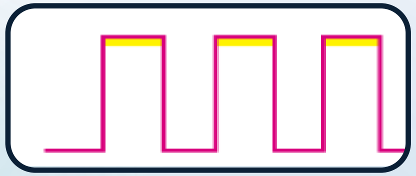
Instead of using level triggering , we can use edge triggering so that the latch gets triggered only once in a clock cycle.

- **Use of master-slave JK flip-flop**

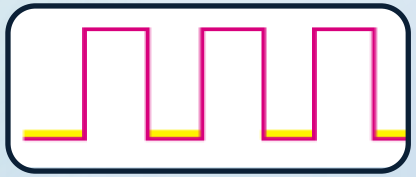
In master-slave JK flip-flop, one acts as a master flip flop and the other acts as a slave flip flop. In this master-slave flip flop, the outputs of the master JK flip flop are connected to the inputs of the slave JK flip flop. The outputs of the slave flip flop are fed back to the inputs of the master JK flip flop.

Triggering Types

Level Triggering

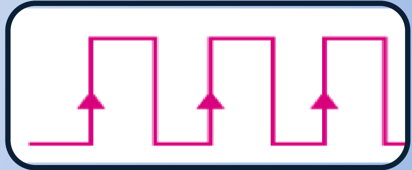


Positive Level Triggering

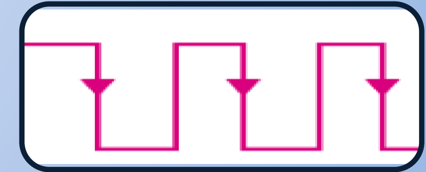


Negative Level Triggering

Edge Triggering



Positive Edge Triggering



Negative Edge Triggering