Sequential Circuits

Objective

- In this lesson, you will learn about:

    1. Sequential Circuits, Synchronous Sequential Circuits and Memory Elements.
    2. Clocked RS, D, JK, & T latches with their analysis.
    3. Characteristic and excitation behavior of these latches.

Introduction

- This is an introductory lesson on sequential logic circuits.
- The general block diagram of a combinational circuit is shown in Figure 1.
- A Combinational logic circuit consists of input variables (X), logic gates (Combinational Circuit), and output variables (Z).
-

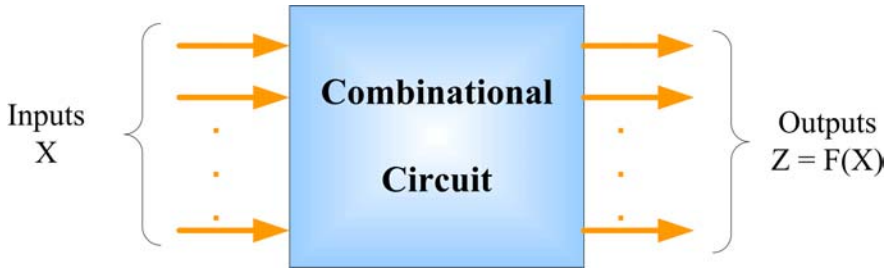


Figure 1: General Block Diagram of a Combinational Circuit

- Unlike combinational circuits, sequential circuits include **memory elements** (See Figure 2).
- The memory elements are circuits capable of storing binary information.
- The binary information stored in these memory elements at any given time defines the **state** of the sequential circuit at that time.
- The outputs, **Z**, of a sequential circuit depends both on the present inputs, **X**, and the present state **Y** (i.e., information stored in the memory elements).
- The next state of the memory elements also depends on the inputs **X** and the present state **Y**.

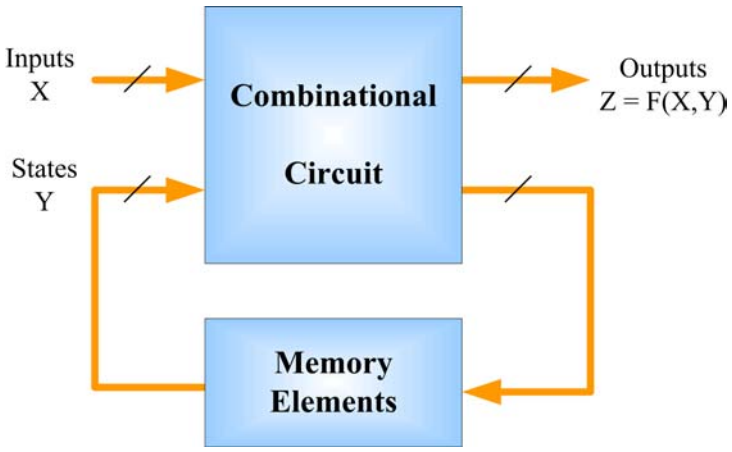


Figure 2: General Block Diagram of a Sequential Circuit

Sequential Adder

- To best understand sequential circuits, let's re-visit a known iterative circuit, a 4-bit combinational ripple carry adder (See Figure 3).

- The combinational circuit of a 4-bit ripple carry adder comprises 4-full adders. The inputs to the circuit are a single-bit *carry-in* (C_{IN}) & two 4-bit numbers A & B. This circuit produces a 4-bit sum S & a single-bit *carry-out* (C_{OUT}).

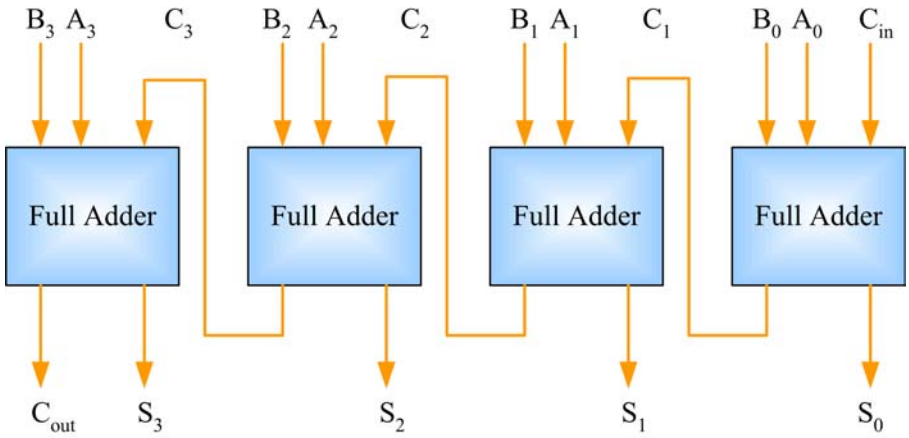


Figure 3: 4-bit Ripple-Carry Adder

- We can notice that all 4-bits of the sum are not computed at the same instance of time. The 1st stage produces the LSB of the sum, S_0, and an intermediate carry C_0 using C_{IN} and the LSB of A & B (A_0, B_0).

- The 2nd stage, using the intermediate carry C_0 along with A_1 and B_1, produces the 2nd bit of the sum, S_1. In this way, the intermediate carry propagates through the stages of the adder & each stage, on the arrival of this carry, produces its corresponding bit of final sum S.

- We observe that only one stage is *active* during the computation of the sum. Based on this observation, we can make an *n*-bit adder using only one stage

full-adder as shown in Figure 4.

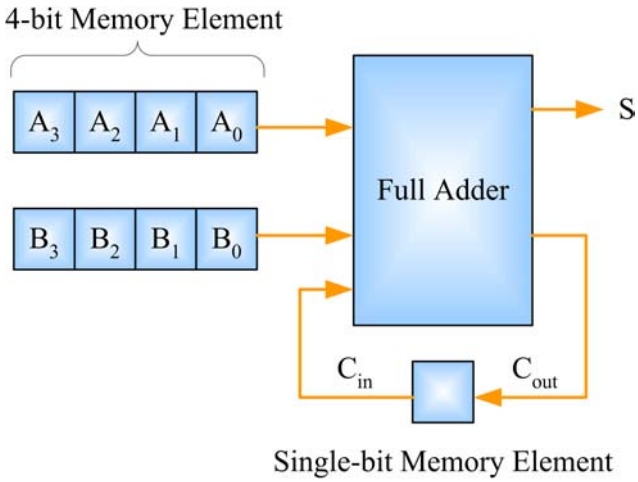


Figure 4: 4-bit Sequential Adder

- However, we need a single-bit memory element to temporarily store the value of the intermediate carry.
- Two 4-bit memory elements are used to store bit-vectors A and B while a single-bit memory element is used to store the intermediate carry.
- As we have only one full adder, it will take four instances of time to add the corresponding bits of A and B.
- We notice here that the sequential adder has one memory element, which stores the state of the circuit as carry. These states define the condition of having a carry or no carry.
- In other words, to define 2-states (0 and 1) in a sequential circuit, we require 1 memory element. In general, for an *n*-state circuit we require $\lceil \log_2 n \rceil$ memory elements.
- We also notice that to move from one state to another, we need a periodic signal, which we called the **Clock**, to synchronize the activity.

Synchronous & Asynchronous Sequential Circuits

- There are two main types of sequential circuits. Their classification depends on the timing of their signals.
- **Synchronous sequential circuits** are systems whose behaviors can be defined from the knowledge of their signals at discrete instants of time.
- While the behavior of **asynchronous sequential circuits** depends upon the order in which their input signals change at any instant of time.

- **Synchronous** sequential logic systems must employ signals that affect the memory elements only at discrete instants of time.
- To achieve this goal, a timing device called a master-clock generator is used to generate a periodic train of Clock pulses.
- These clock pulses are distributed throughout the system in such a way that memory elements are affected only with arrival of the Clock pulse.

Memory Elements

- A basic memory element, as shown in Figure 5 (a), is the **latch**.
- A latch is a circuit capable of storing one bit of information.
- The latch circuit consists of two inverters; with the output of one connected to the input of the other.
- The latch circuit has two outputs, one for the stored value (**Q**) and one for its complement (**Q'**).
- Figure 5 (b) shows the same latch circuit re-drawn to illustrate the two complementary outputs.
- The problem with the latch formed by **NOT** gates is that we can't change the stored value. For example, if the output of inverter B has logic 1, then it will be latched forever; and there is no way to change this value.

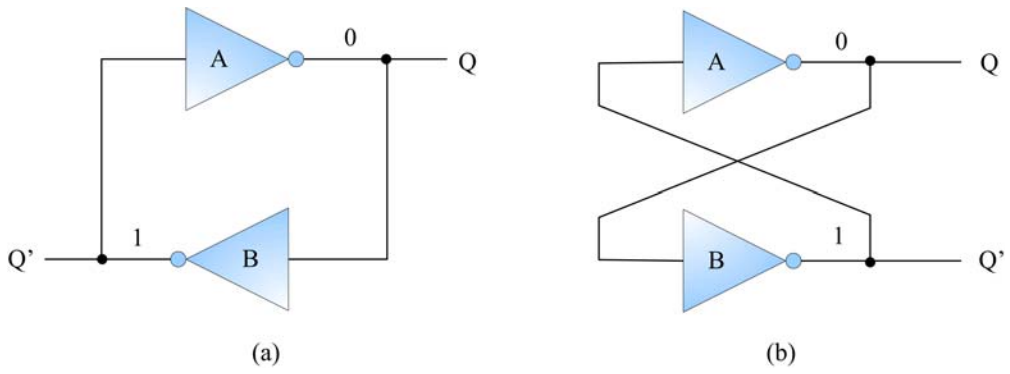


Figure 5: Simple Latch

SR Latch

- Recall that a NOT gate can alternatively be expressed using **NAND** and **NOR** gates as shown in Figure 6 (a).
- Using NOR gates, we can obtain the latch circuit shown in Figure 6 (b).
- This latch has two outputs, **Q** and **Q'**, and two inputs **S** and **R**.
- This type of latches is sometimes called a cross-coupled SR latch or simply SR latch.

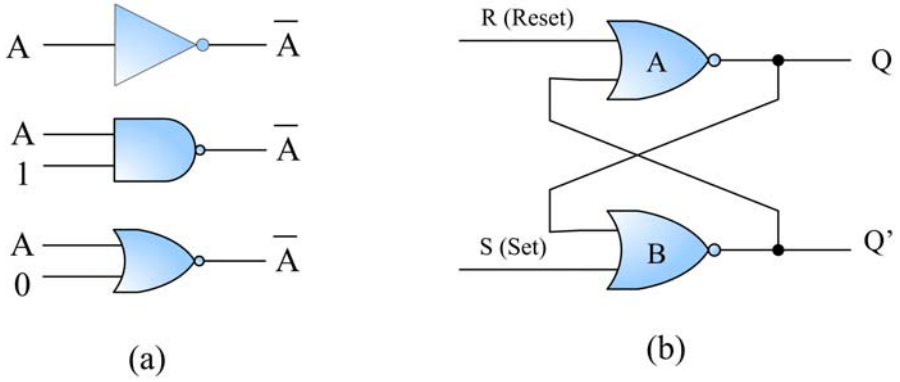


Figure 6: (a) Alternative forms of NOT gate (b) Basic SR latch with NOR gates

Table 1: Functional Table of the Basic SR Latch with NOR Gates

S	R	Q	Q'	
1	0	1	0	Set State
0	0	1	0	
0	1	0	1	Reset State
0	0	0	1	
1	1	0	0	Undefined

- The SR latch has two main states: **set** and **reset** (See Table 1).
- When output **Q=1** and **Q'=0**, the latch is said to be in the **set state**; and when **Q=0** and **Q'=1**, it is in the **reset state**.
- When the input **S=0** and **R=0**, the SR latch remains in its current state (i.e. set or reset). In this case, the values of Q and Q' are latched forever.
- When the SR latch is in the set state, we can change the state to the reset state by making R=1.
- Similarly, the state of the SR latch can be changed from reset to set by making S=1.

- If a 1 is applied to both inputs of the SR latch, both outputs go to 0.
- This produces an undefined state, because it violates the requirement that the outputs be complement of each other.
- It also results in an indeterminate next state when both inputs return to 0 simultaneously as shown in the figure.
- In normal operation, these problems are avoided by making sure that 1's are not applied to both inputs simultaneously.

SR Latch with NAND Gates

- The SR latch with two cross-coupled **NAND** gates is shown in Figure 7.
- It operates with both inputs normally at 1, unless the state of the latch has to be changed (See Table 2).
- With both inputs at 1, applying 0 to the S input causes the output Q to go to 1 (i.e. **set state**).
- In the same way, applying 0 to the R input causes the output Q to go to 0 (i.e. **reset state**).
- The condition that **undefined** for this NAND latch is when both inputs are equal to 0 at the same time, which causes both outputs Q and Q' to go to 1.

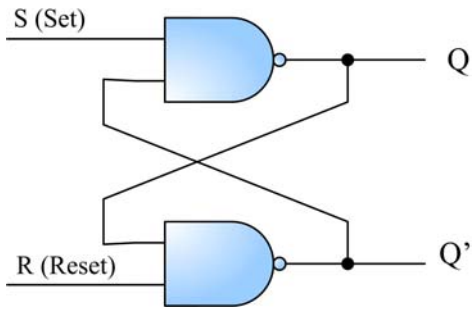


Figure 7: Basic SR LATCH with NAND Gates

Table 2: Functional Table of the Basic SR Latch with NAND Gates

S	R	Q	Q'	
0	1	1	0	Set State
1	1	1	0	
1	0	0	1	Reset State
1	1	0	1	
0	0	0	0	Undefined

Clocked SR Latch

- The operation of the basic SR latch can be modified by providing an additional control input (**clock**) that determines when the state of the latch can be changed.
- An SR latch with a control input **C** is shown in Figure 8.
- It consists of the basic SR latch with two additional AND gates.
- The control input C acts as an **enable** signal to the latch (See Table 3).
- When **C=0**, the S and R inputs have no effect on the latch, so the latch will remain in the same state regardless of the values of S and R.
- When **C=1**, the S and R inputs will have the same effect as in the basic SR latch.

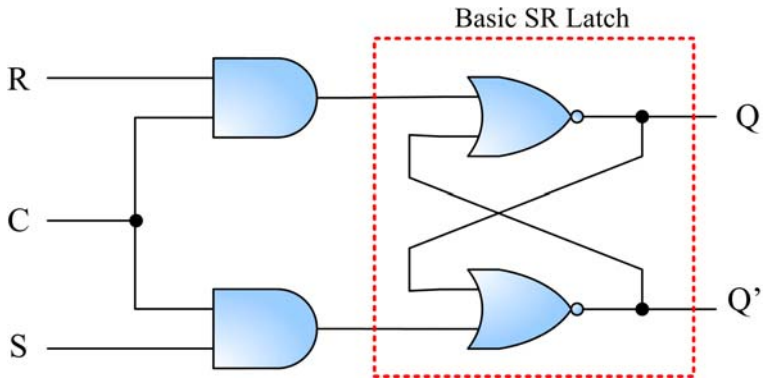


Figure 8: Clocked SR Latch

Table 3: Functional Table of Clocked SR Latch

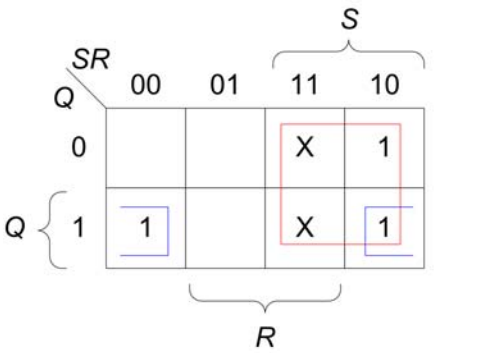
C	S	R	Next State of Q
0	X	X	No Change
1	0	0	No Change
1	0	0	Q = 0; Reset State
1	1	0	Q = 0; Set State
1	1	1	Undefined

Characteristic Table of the SR Latch

- The **characteristic** (behavior) of the sequential circuit defines its logical property by specifying the next states when the inputs and the present states are known. The characteristic of the RS latch is shown in Table 4.
- The characteristic table can also be represented algebraically using what is known as a **characteristic equation**.
- The characteristic equation is derived using the K-Map as shown in Figure 9.
- X's mark the two indeterminate states in the map in Figure 9, since their inputs are never allowed (Recall "Don't Cares").
- Note that the condition S.R = 0 must also be included as both S and R cannot simultaneously be 1.
- The characteristic equations are used in the analysis of sequential circuits.

Table 4: Characteristic Table of SR Latch

Q(t)	S	R	Q(t + 1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	Indeterminate
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	Indeterminate



$Q(t+1) = S + R'Q \qquad SR = 0$

Figure 9: Characteristic Equation of the SR Latch

Excitation Table of the SR Latch

- During the design process we usually know the transition from present state to next state and wish to find the latch input conditions that will cause the required transition.

- For this reason, we need a table that lists the required inputs for a given change of state. Such a table is called an *excitation table*, and it specifies the excitation behavior of the sequential circuits. These are used in the synthesis (design) of sequential circuits, which we shall see later.

- The excitation of the SR latch is given in Table 5.

Table 5: Excitaion table of the SR latch

Q(t)	Q(t+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

Clocked D-Latch

- One way to eliminate the undesirable undefined state in the SR latch is to ensure that the inputs S and R are never equal to 1 at the same time.

- This is done in the D latch shown in Figure 10.

- This latch has only two inputs **D** (**Data**) and **C** (**Clock**). Note that D is applied directly to the set input S, and its complement is applied to the reset input R.

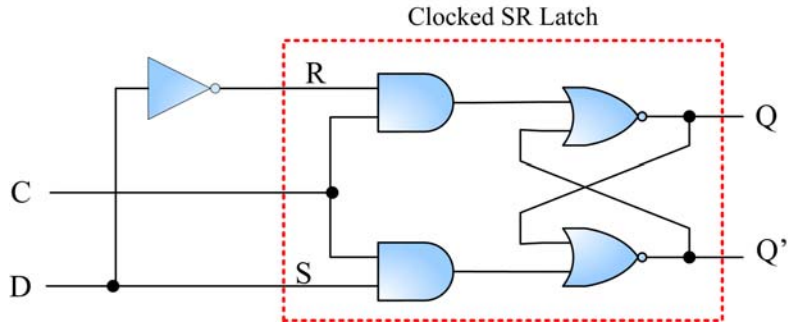


Figure 10: Clocked D Latch

- As long as the clock input **C = 0**, the SR latch has both inputs equal to 0 and it can't change its state regardless of the value of D (See Table 6).

- When C is 1, the latch is placed in the set or reset state based on the value of D.

- If **D = 1**, the **Q** output goes to **1**.
- If **D = 0**, the **Q** output goes to **0**.

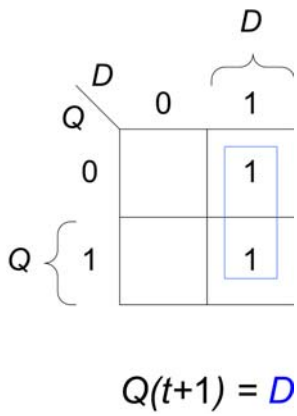
- The characteristic table and the characteristic equation of a D latch are illustrated in Table 7 and Figure 11 respectively.

Table 6: Functional Table of the D-Latch

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

Table 7: Characteristic Table of the D-Latch

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



$Q(t+1) = D$

Figure 11: Characteristic Equation of the D-Latch

Clocked JK-Latch

- The clocked JK latch is shown in Figure 12. Note the feedback path from the outputs Q and Q' to the AND gates at the input.
- JK latch is an improvement over the SR latch in the sense that it does not have any indeterminate states.
- Inputs J and K behave like S and R of the SR latch. J and K set and clear the state of the latch, respectively.

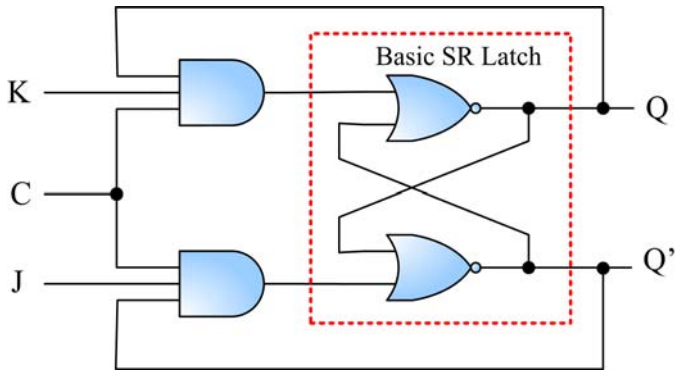


Figure 12: Clocked JK-Latch

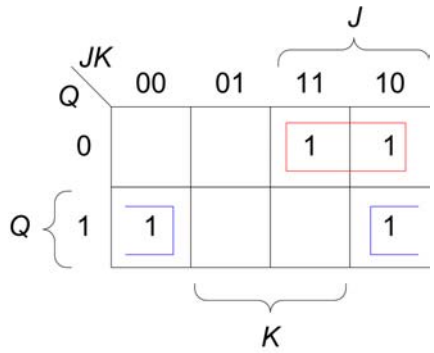
- The functional table of the clocked JK-Latch is illustrated in Table 8.
- If both J and K are made high (recall that both S and R cannot be made high at the same time) then the latch switches to its complement state, that is, if Q=1 then it switches to Q=0, and vice versa.
- Output Q is ANDed with K and C inputs so that the latch is cleared during a clock pulse only if Q was previously 1.
- Similarly, Q' is ANDed with J and C inputs so that the latch is set with a clock pulse only if Q' was previously 1.
- The JK latch behaves exactly like the SR latch, except when both J and K are 1.
- Characteristic table and characteristic equation of the JK-Latch are shown in Table 8 and Figure 13 respectively.

Table 8: Functional Table of the Clocked JK-Latch

C	J	K	Next State of Q
0	X	X	Q *(No Change)*
1	0	0	Q *(No Change)*
1	0	1	0 *(Reset State)*
1	1	0	1 *(Set State)*
1	1	1	Q' *(Complement)*

Table 9: Characteristic Table of the JK-Latch

Q(t)	J	K	Q(t + 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



$$Q(t+1) = JQ' + K'Q$$

Figure 13: Characteristic Equation of the JK-Latch

- Excitation table of the JK-Latch are illustrated in Table 9.
-
- When both states, present and the next one are to be 0, then the J input must remain at 0 and the K input can be either 0 or 1 (i.e., X).

- Similarly, when both present state and the next state are 1, the K input must remain at 0 while J input can be 0 or 1 (i.e., X).

- If the latch is to have a transition from the 0-state to 1-state, J must be equal to 1 since the J input sets the latch. However, input K may be either 0 or 1.

- Similarly, for a 1-to-0 transition, K must be set to 1 and J can be either 0 or a 1.

Table 10: Excitaion Table of the JK-Latch

Q(t)	Q(t+1)	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Clocked T-Latch

- The T latch is a single-input version of the JK latch. It is obtained by tying both the inputs J and K together as shown in Figure 14. The name comes from the ability of the latch to "toggle" or change the state.

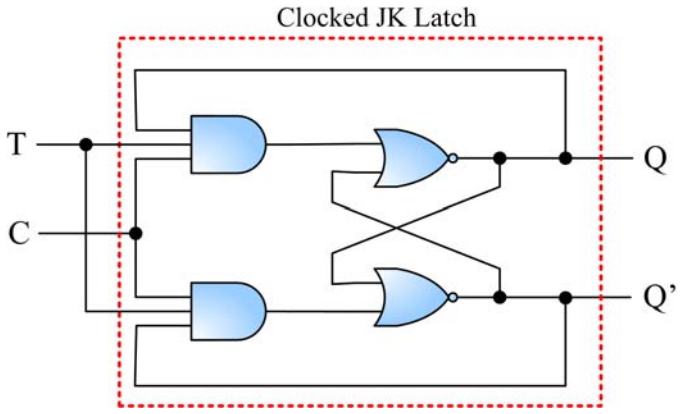


Figure 14: Clocked T-Latch

- Observe that when T=1, regardless of the present state, the latch toggles or changes to the complement state when the clock pulse occurs (See Table 11).

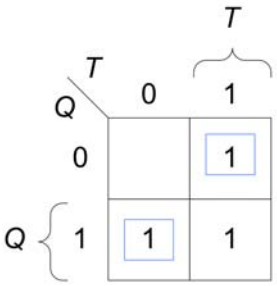
Table 11: Functional Table of the Clocked T-Latch

C	T	Next State of Q
0	X	*No Change*
1	0	*No Change*
1	1	*Q'*

- The toggling effect can be seen more clearly in the characteristic behavior of the T-Latch (See Table 12 and Figure 15). Notice that when T = 0, the state of the latch remains unchanged.

Table 12: Characteristic Table of the T-Latch

Q(t)	T	Q(t + 1)
0	0	0
0	1	1
1	0	1
1	1	0



$$Q(t+1) = TQ' + T'Q$$

Figure 15: Characteristic Equation of the T-Latch

- The excitation table of the T-Latch is illustrated in Table 12.

- Note that when the state of the latch must remain the same, the requirement is that T = 0. When the state of the latch has to be complemented, T must equal 1, as summarized in the excitation table.

Table 13: Excitation Table of the T-Latch

Q(t)	Q(t + 1)	T
0	0	0
0	1	1
1	0	1
1	1	0

Problem with the Level Triggered JK and T latches

- In JK latch, with J = 1 and K = 1 the state of the latch toggles. However, if the clock signal remains at 1 (while J = K = 1), the output will go in repeated transitions; this is an undesirable oscillating effect. And when clock goes to 0, output will be latched to an unknown state.

- To avoid this undesirable operation the clock pulse must have pulse duration, which is shorter than the propagation delay of the signal through the latch. This however is not at all acceptable since the operation of the circuit will then depend on the width of the clock pulse and/or the delay through the latch.

- For this reason, JK latches are never constructed as discussed above. The restriction on the pulse width can be eliminated with a master-slave or edge-triggered construction described in the next lesson. The same reasoning applies to the T latch.