江西理工大学
Jiangxi University of Science and Technology
信息工程学院
School of information engineering

Dr Ata Jahangir Moshayedi

Digital System Design

Clip Lecture series

Spring_2020

Prof Associate , School of information engineering Jiangxi university of science and technology, China
EMAIL: ajm@jxust.edu.cn

Jiangxi University of Science and Technology

FF_review

Introduction

- Sequential circuits are circuits that implement two concepts: memory and time. **Latches** are the basic circuits that implement these two concepts. A latch has two inputs (set and reset) and one output. Often you will see latch circuits drawn with an output and its inverse. A **flip-flop** is a specific kind of latch that has two conditions of stability, is enabled for a short time, and can be edge-triggered. In this lesson we will start by looking at a simple circuit that is able to remember it's state (memory) and then we will learn about different latch improvements that allow for better functionality.

The SR-Latch

- An **SR Latch** is the simplest circuit that implements memory and is known as a **set-reset** latch. The following table is the **state table** of a this type of sequential circuit that can memorize its state. A state table can be defined as a one that relates all possible combinations of the inputs AND the current states with corresponding output values.

R	S	Q
0	0	Memorize the current state
0	1	1
1	0	0
1	1	X

- Notice the fourth combination of input which is 'R=1 S=1' in the state table. The output (Q) for this combination is called a **don't care** (X) condition. Figure 1 shows a circuit that implements this state table. Take some time to review it and try the different input combinations using Logisim or any other simulation software you are comfortable with.

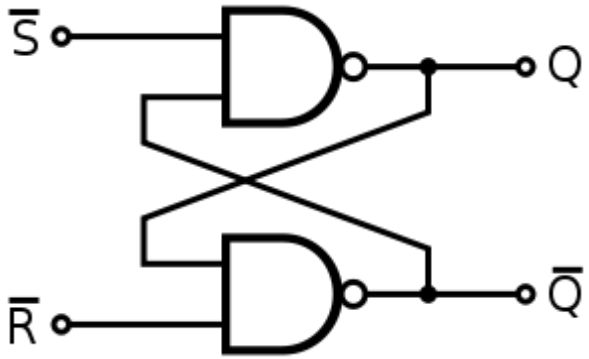


Figure 1: Common drawing the SR latch.

- This SR latch implements memory or state. We still need to implement the concept of time, however. This can be done by **gating** the latch using a square signal. Electronically, this means adding a square signal to the inputs of the SR latch as shown in Figure 2. We will use Q' for the inverse of Q.

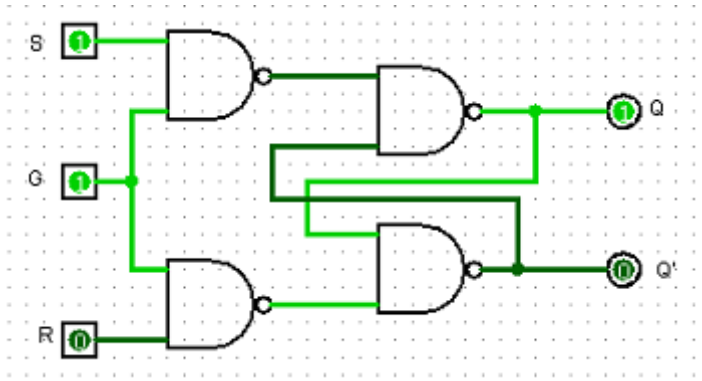


Figure 2: Gated SR latch.

- In Figure 2, the input G, sometimes called E to denote **Enable**, 'isolates' the latch when its value is 0. By isolation, we mean that no matter how the inputs (R and S) change, the outputs do not. In other words, the state is memorized. The input (G) is added to allow the latch to implement the 'time' concept. This is done by allowing only one input change (on both R and S inputs) during a '1' period on G before the latter changes back to '0'. If we want to change the inputs, we can then do this when G changes back to '1'. Note that this is an additional condition that allows the latch to implement the memory concept. The following is the state table of the gated SR-latch:

G	S	R	Q
0	X	X	Q_{-1}
1	0	0	Q_{-1}
1	0	1	0
1	1	0	1
1	1	1	Not allowed

- Figure 3 shows the logic symbol of the SR flip-flop. Notice that the gate input is denoted as E for Enable and it has the same functionality as G discussed above.

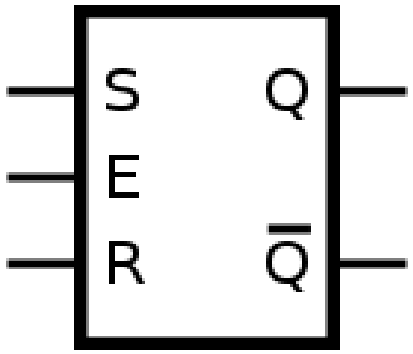


Figure 3: Gated SR-latch.

When the square signal is a square wave we call it a **clock** and we call the latch a **clocked** latch.

The D-Latch

- The SR-latch that we saw above implements the two required aspects of sequential circuits: memory and time. We still need to be careful however not to input S=1 and R=1 as this will put the circuit in an **unallowed state**. In order to overcome this issue, we can insert what is called an **inverter** between the S and R inputs. Using the gate input (G) will allow us to compensate for the S=0 R-0 memory state in the SR-latch. This results in a latch called the D-gated latch, shown in Figure 4.

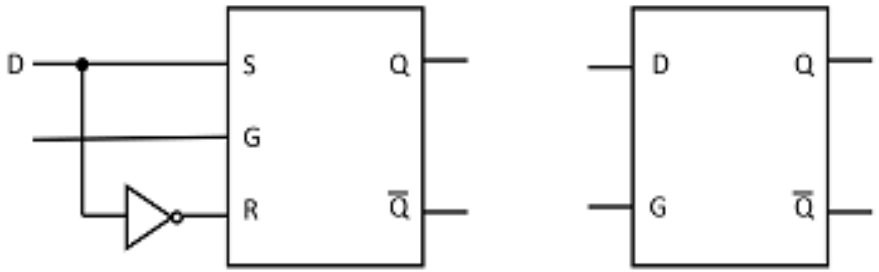


Figure 4: The D-gated latch built using an RS-latch and the symbol of a D-gated latch

The D-latch 'copies' its input to its output. Its main use is to isolate two parts of a system while the latch is not enabled.

The JK-Latch

- While the D-latch solves an unallowed state when the S=1 and R=1, we can still use this input combination to produce a new state transfer which is the inversion of the current state, and is very useful when designing sequential circuits. This can be done by using the circuit depicted in Figure 5, which is called a JK-gated latch.

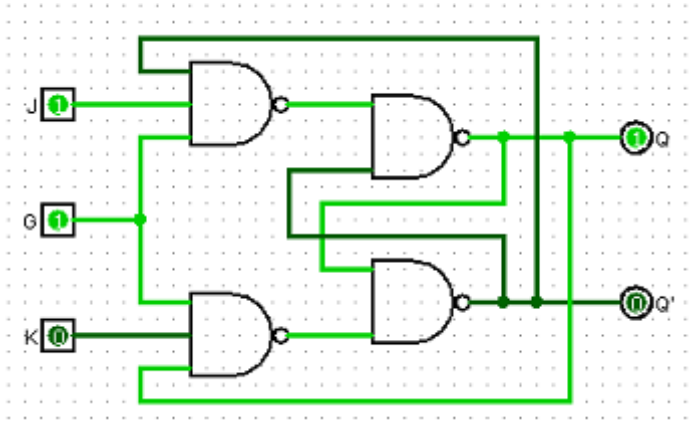


Figure 5: JK-gated latch.

- The state table of the JK-latch is as follows:

G	J	K	Q
0	X	X	Q_{-1}
1	0	0	Q_{-1}
1	0	1	0
1	1	0	1
1	1	1	Q'_{-1}

The T-Latch

- Now that both inputs (S=0, R=0 and S=1, R=1) do not lead to unallowed states, we can use another type of gated latch called the T-latch. The T-latch is a JK-latch where both data inputs J and K are connected as shown in Figure 6.

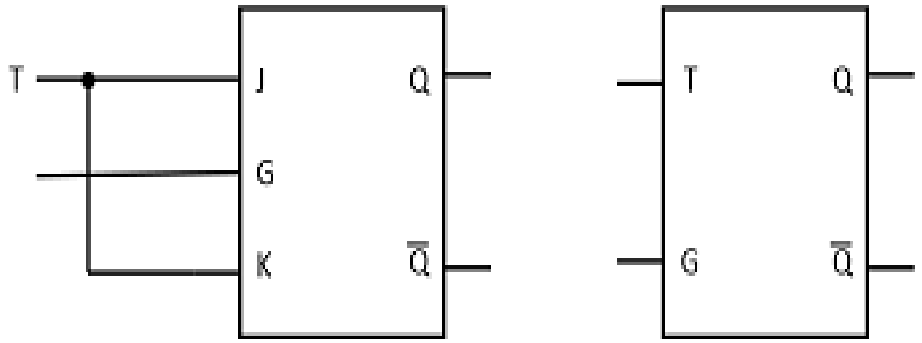


Figure 6: T-gated latch.

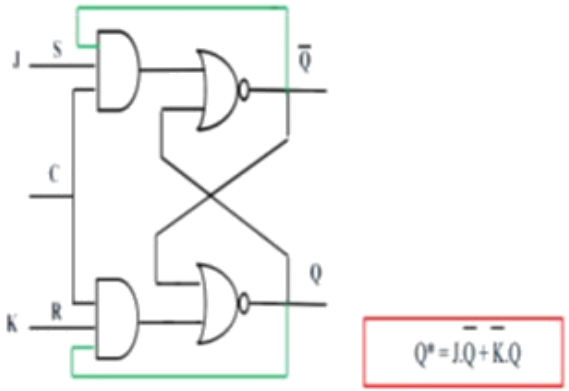
The main function of a T-latch is to divide frequency.

All Flip-Flop Characteristic Tables

Table 5.1
Flip-Flop Characteristic Tables

JK Flip-Flop

J	K	Q(t + 1)	
0	0	$Q(t)$	No change
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Complement

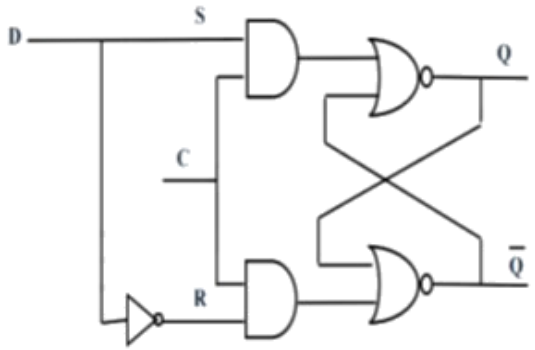


$Q^* = J\bar{Q} + \bar{K}Q$

D Flip-Flop

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

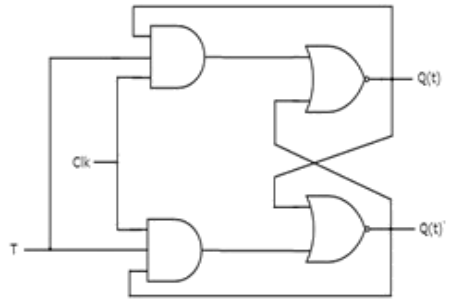
$Q^* = D$



(b) Graphic Symbol

T Flip-Flop

T	Q(t + 1)	
0	$Q(t)$	No change
1	$Q'(t)$	Complement



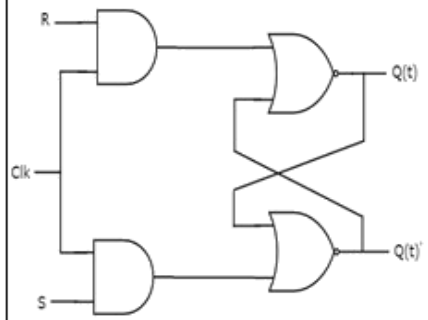
$Q(t+1) = T \oplus Q(t)$

SR FF

S	R	Q*
0	0	Q
0	1	0
1	0	1
1	1	X

$Q^* = S + \bar{R}.Q$

$S.R = 0$



Refrence

- http://osp.mans.edu.eg/cs212/Seq_circuits_analysis.htm