Flip-Flops

Objectives

- The objectives of this lesson are to study:

  1. Latches versus Flip-Flops
  2. Master-Slave Flip-Flops
  3. Timing Analysis of Master-Slave Flip-Flops
  4. Different Types of Master-Slave Flip-Flops
  5. Propagation Delay

Problem with Latches

- A latch is a level sensitive device.
- Because of this the state of the latch may keep changing in circuits with feedback as long as the clock pulse remains active.
- Thus, instead of having output change once in a clock cycle, the output may change a number of times resulting in latching of unwanted input to the output.
- Due to this uncertainty, latches can not be reliably used as storage elements.

Solution to this Problem

- To overcome this problem of undesired toggling, we need to have a mechanism in which we have higher degree of control on the output of the memory element when the clock pulse changes.
- This is achieved by introducing a special clock-edge detection logic, such that the state of the memory element is switched by a momentary change in the clock pulse (i.e. an edge).
- This is effective because the clock changes only once during a clock period.
- Such a memory element is "edge-sensitive", i.e., it changes its state at the rising or falling edge of a clock.
- Edge-sensitive memory elements are called Flip-Flops.
- Figure 1 shows the standard graphic symbols for positive and negative edge triggered Flip-Flops.

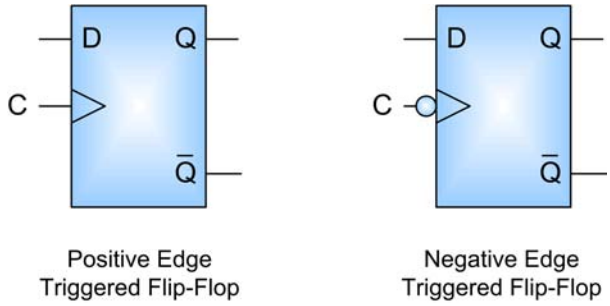

Figure 1: Graphic Symbols of Edge-Triggered Flip-Flops

Master-Slave Flip-Flops

- The simplest way to build a flip-flop is by using two latches in a 'Master-Slave' configuration as shown in Figure 2.
- In this configuration, one latch serves as the *master* receiving the external inputs and the other as a *slave,* which takes its inputs from the master.
- When the clock pulse goes high, information at S and R inputs is transmitted to master.
- The slave flip-flop however remains isolated since its control input C is 0.
- Now when the clock pulse returns to '0', the master gets disabled and blocks the external inputs to get to its outputs whereas slave gets enabled and passes the latched information to its outputs.

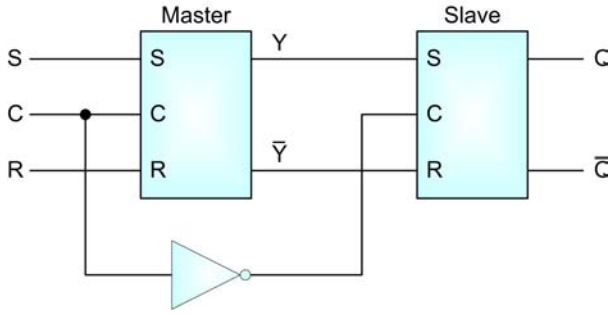

Figure 2: Block diagram of SR Master-Slave Flip-Flop

Timing Analysis of Master-Slave Flip-Flop

- Now let's view the operation of the master-slave flip-flop by analyzing its timing wave forms (See Figure 3).
- Consider a master-slave flip-flop in the clear state (i.e. Y=0 and Q=0) prior to the occurrence of a pulse.
- The inputs S=1 and R=0 are applied. So when the clock goes high, the output of the master latch will change to the set state, while the slave latch remains disabled.

- When the clock returns to 0, the master latch is disabled and the slave latch is enabled.
- Thus, the data at the slave's input when the clock was high gets latched at the slave's output.

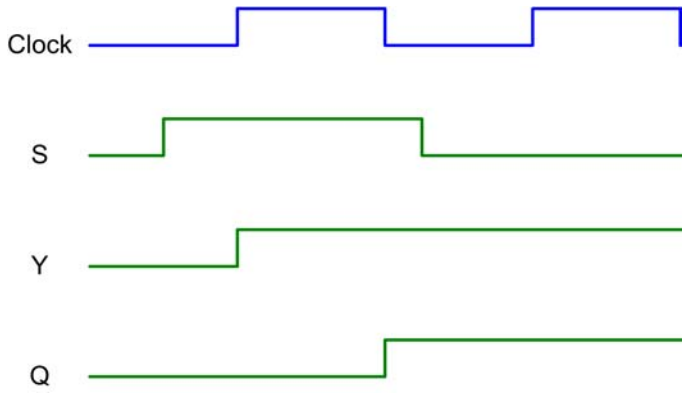

Figure 3: Timing wave form of SR Master-Slave Flip-Flop

Different Types of Master-Slave Flip-Flops

Master-Slave JK-FF

- The SR flip-flop can be modified to a JK flip-flop to eliminate the undesirable condition that leads to undefined outputs and indeterminate behavior.
- A Master-Slave JK Flip-Flop is shown in the Figure 4.
- Here, the output gets complemented when both J and K inputs are high.

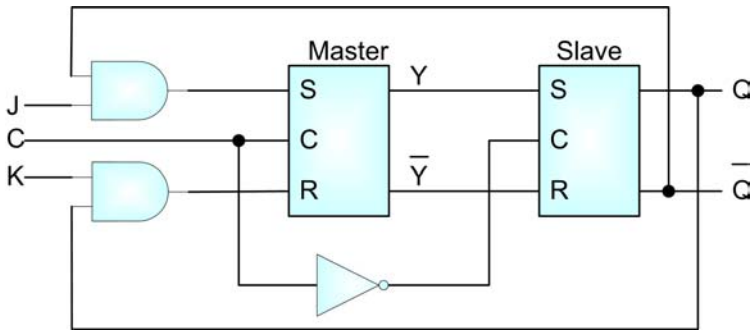

Figure 4: JK Master-Slave Flip-Flop

D-Type Positive-Edge-Triggered FF

- The logic diagram of a positive edge triggered D-type flip-flop is shown in the Figure 5.
- This flip-flop takes exactly the form of a master-slave flip-flop, with the master a D latch and the slave an SR latch. Also, an inverter is added to the clock input of the master latch.

- Because the master latch is a D latch, the flip-flop exhibits edge-triggered rather than master-slave (pulse-triggered) behavior.

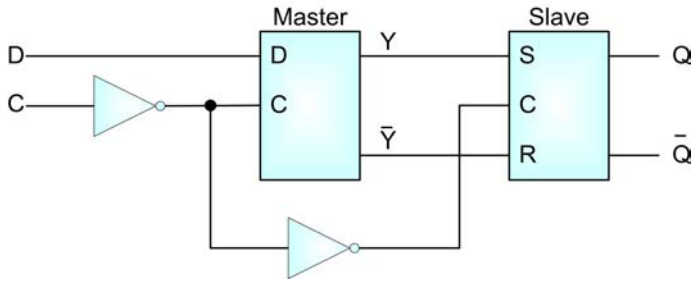

Figure 5: D-Type Positive-Edge-Triggered FF

Propagation Delay

- In digital logic, every gate has got some finite amount of delay because of which the change in the output is not instantaneous to the change in the input.
- In simple terms, the times it takes for an input to appear at the output is called the propagation delay.
- In Figure 6, $\mathbf{t_{PHL}}$, describes the time it takes for an input to cause the output to change from logic-level-high to logic-level-low.
- Similarly, $\mathbf{t_{PLH}}$, refers to the delay associated when an input change causes the output to change from logic-level-low to logic-level-high.
- The overall delay is average of these two delays.

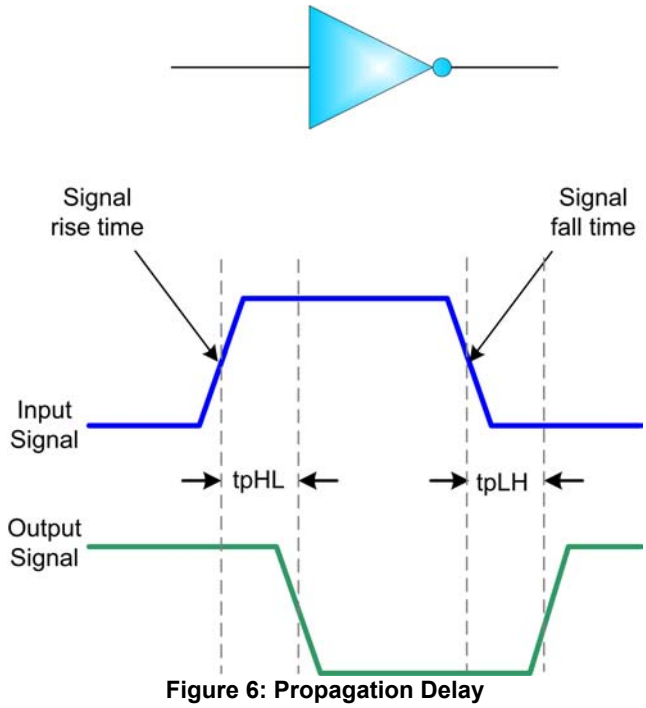

Figure 6: Propagation Delay

Setup and Hold Times

- For correct operation of logic gates we need to satisfy some timing constrains regarding application of inputs and collecting of their outputs.
- **Setup time** (T_s) refers to a constant duration for which the inputs must be held prior to the arrival of the clock transition (See Figure 7).
- Once the inputs are properly set, it must be kept for some time for their proper reading-in by the gate once the transition signal is triggered.
- **Hold time** (T_h) refers to the duration for which the inputs must not change after the arrival of the transition (See Figure 7).
- If the setup and hold times are violated, a gate may produce an unknown logic signal at its output. This condition is called as **meta-stability**.

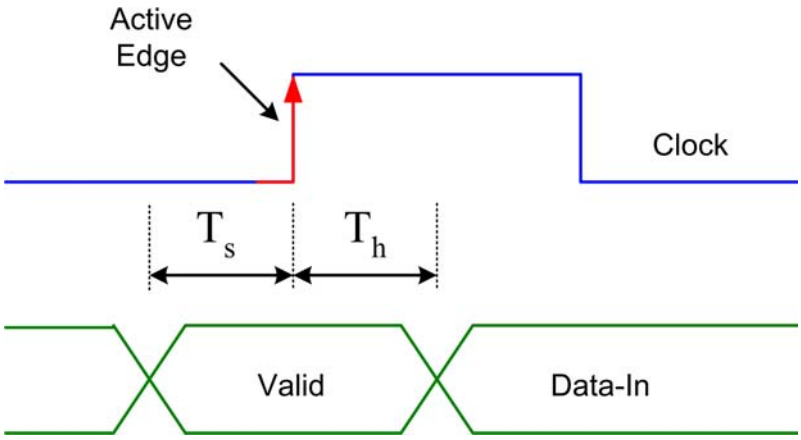

Figure 7: Setup and Hold Times

Propagation Delay

- To set or clear flip-flops asynchronously (i.e., without the use of clock and inputs) some flip-flops have direct inputs usually called **direct preset** or **direct clear**.
- These inputs are needed to bring the flip-flops to a known initial state prior to the normal clocked operation.
- A direct preset input, sets the output of a flip-flop to some known value, asynchronously, for example logic-1 or logic-0.
- A direct clear switch clears or resets all the flip-flops to logic value-0.
- Figure 8 shows the graphical symbol of a negative-edge-triggered JK-flip-flop with a direct clear.

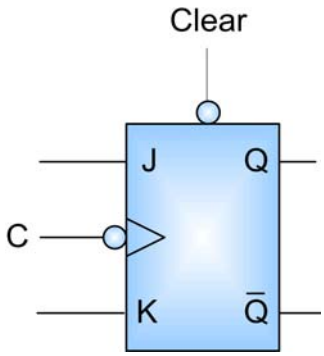

Figure 8: Negative-edge-triggered JK Flip-Flop with Asynchronous Clear