CMSC 313 Lecture 21

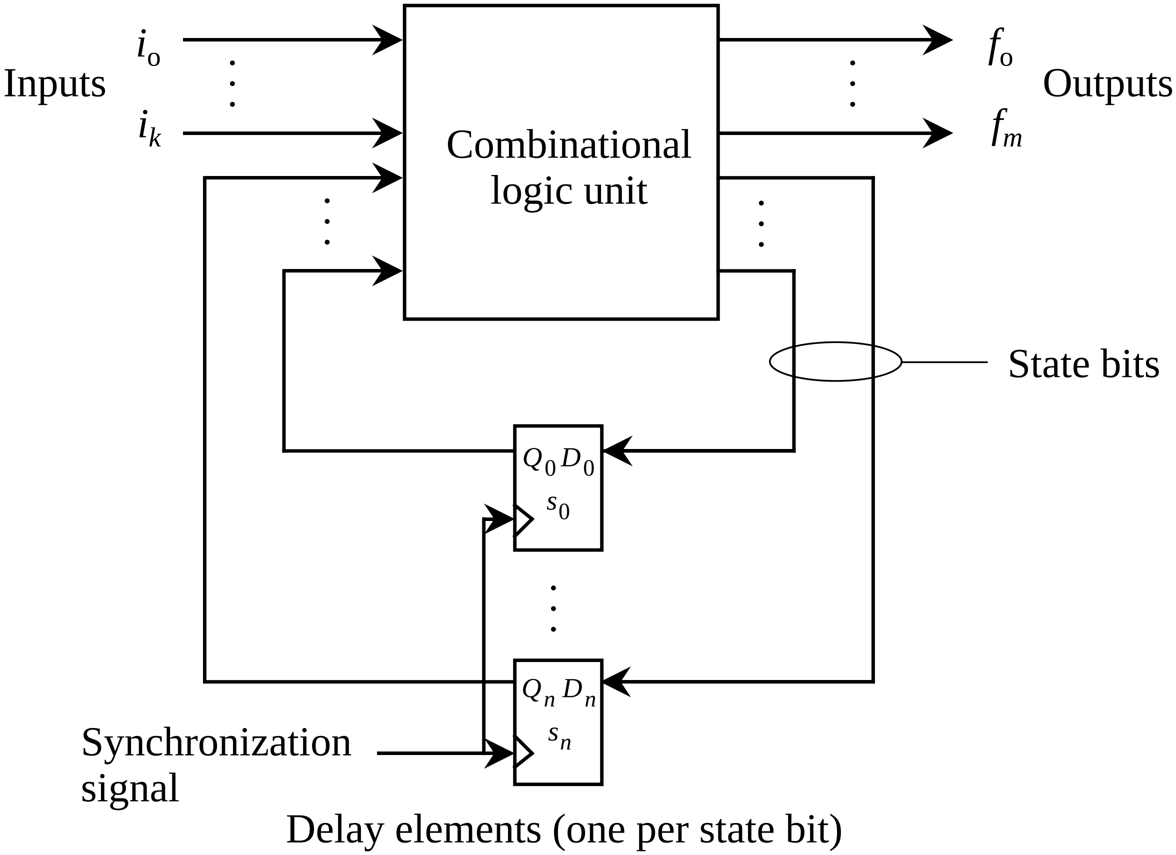
- Introduction to Sequential Logic
- Flip-Flops

Sequential Logic

- **The combinational logic circuits we have been studying so far have no memory. The outputs always follow the inputs.**
- **There is a need for circuits with memory, which behave differently depending upon their previous state*.***
- **An example is a vending machine, which must remember how many and what kinds of coins have been inserted. The machine should behave according to not only the current coin inserted, but also upon how many and what kinds of coins have been inserted previously.**
- **These are referred to as *finite state machines*, because they can have at most a finite number of states.**

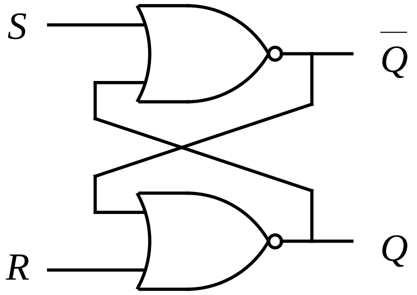
Classical Model of a Finite State Machine

- **An FSM is composed of a combinational logic unit and delay elements (called *flip-flops*) in a feedback path, which maintains state information.**

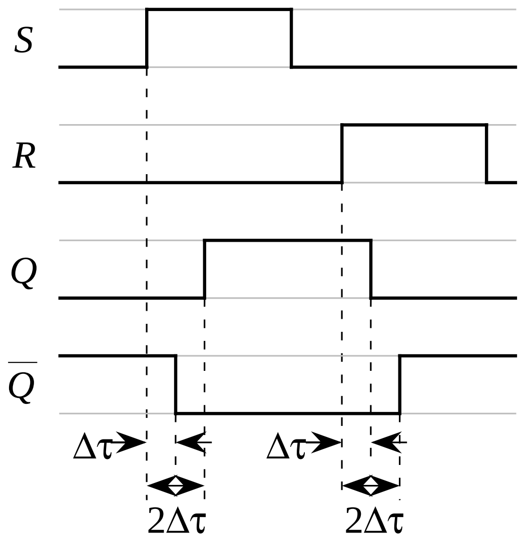


S-R Flip-Flop

- The S-R flip-flop is an active high (positive logic) device.

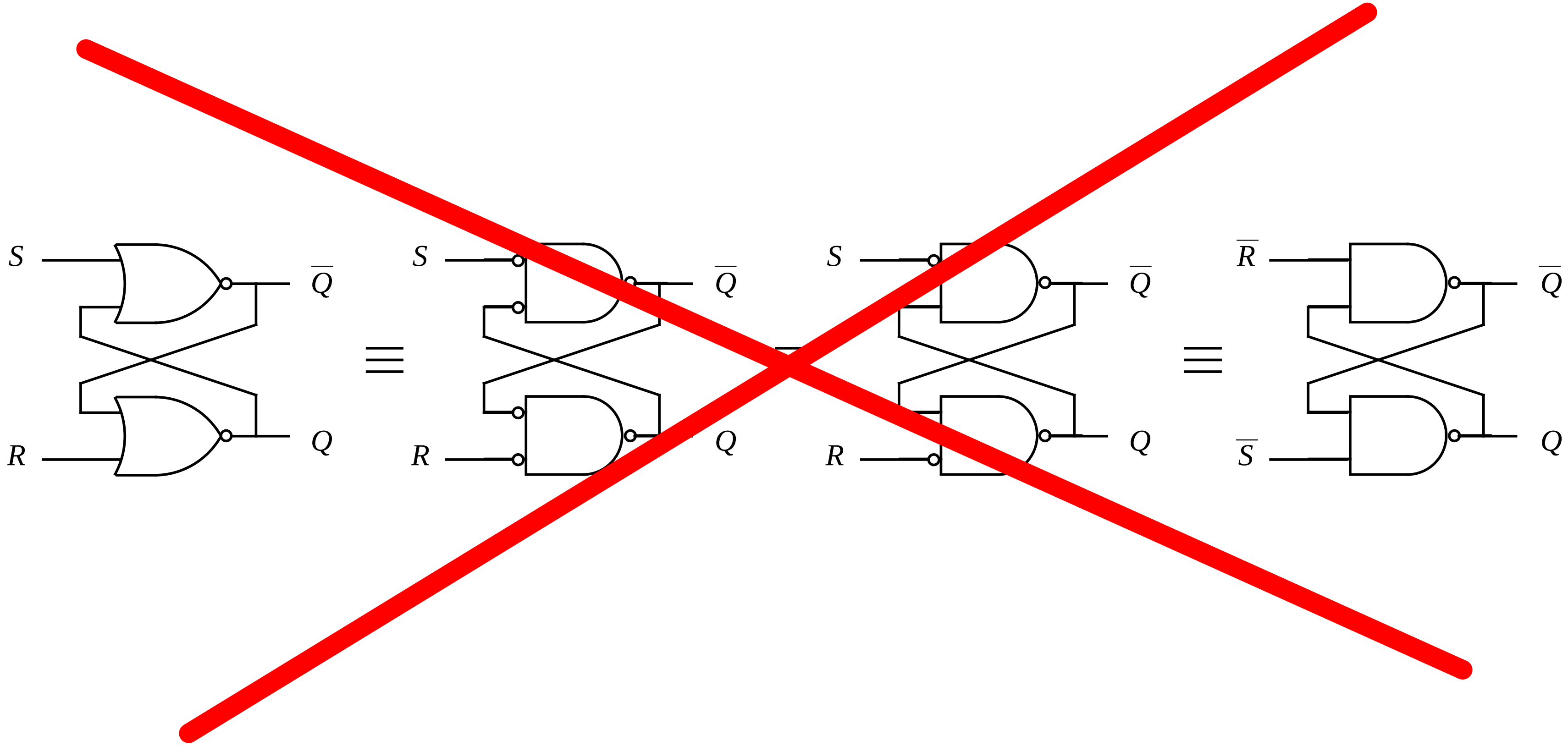


Q_t	S_t	R_t	Q_{i+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	(disallowed)
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	(disallowed)

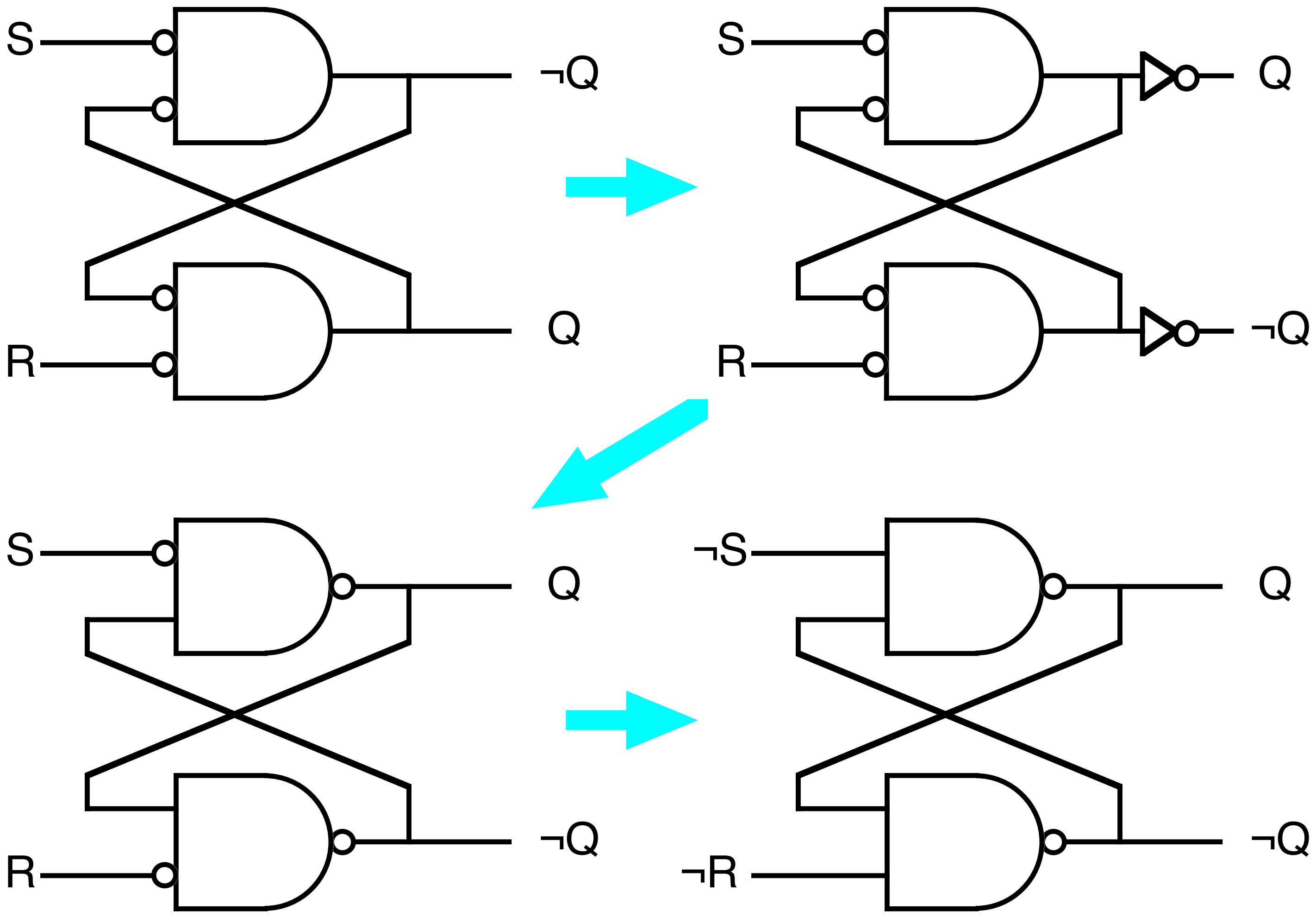


Timing Behavior

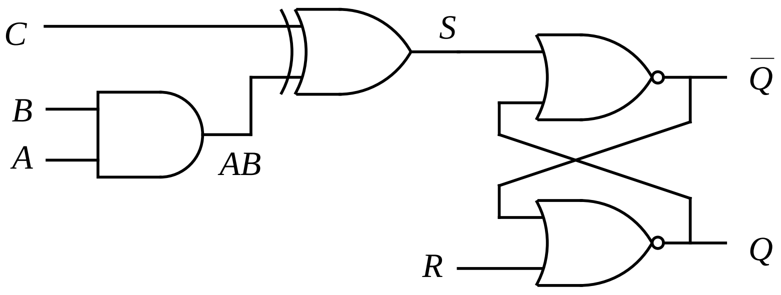
NAND Implementation of S-R Flip-Flop

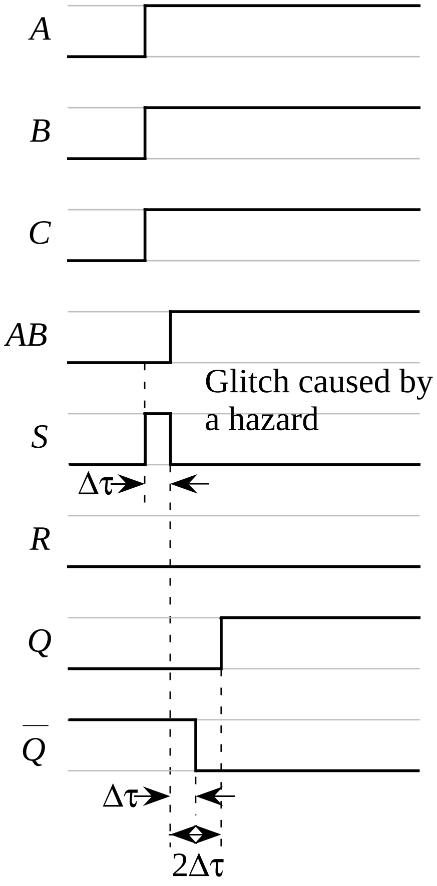


SR Latch using NAND GATES



A Hazard

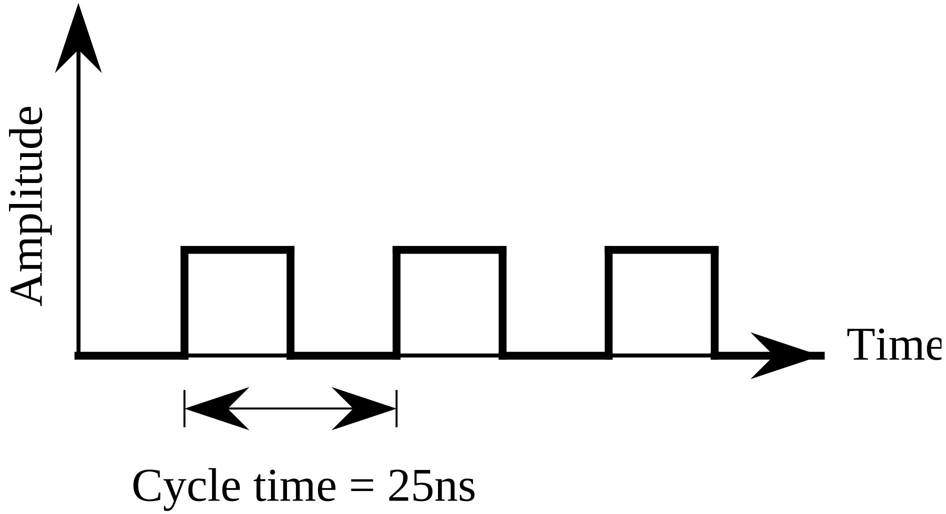




Timing Behavior

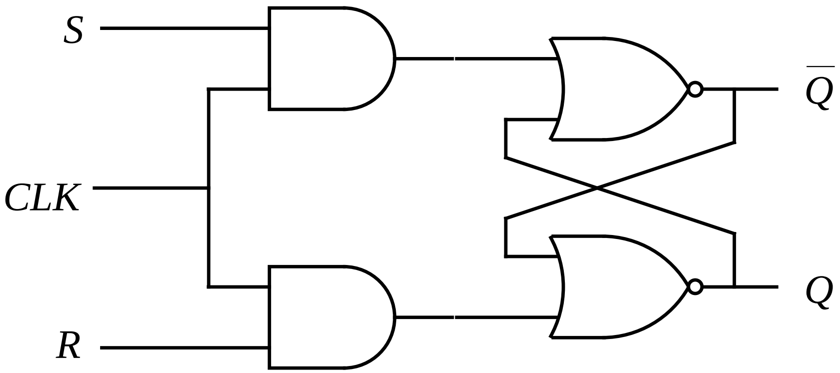
- It is desirable to be able to “turn off” the flip-flop so it does not respond to such hazards.

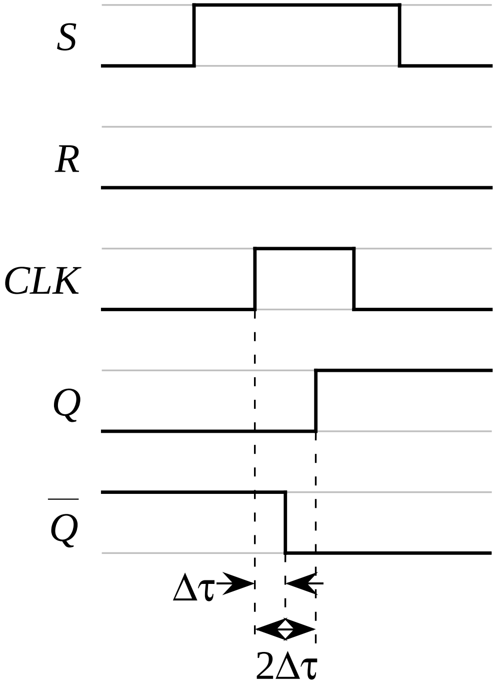
A Clock Waveform: The Clock Paces the System



- **In a positive logic system, the “action” happens when the clock is high, or positive. The low part of the clock cycle allows propagation between subcircuits, so their inputs settle at the correct value when the clock next goes high.**

Clocked S-R Flip-Flop



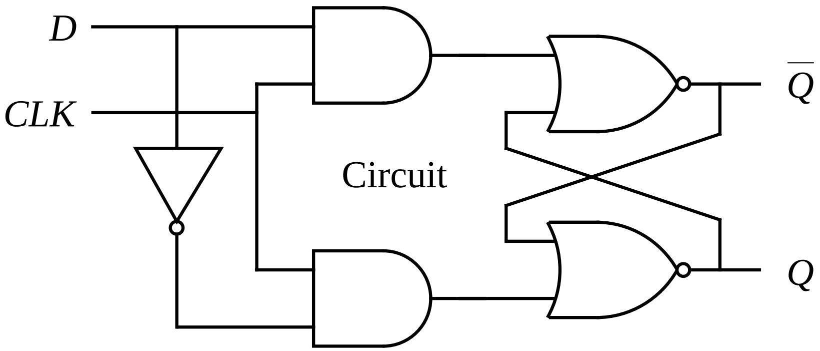


Timing Behavior

- **The clock signal, CLK, enables the S and R inputs to the flip-flop.**

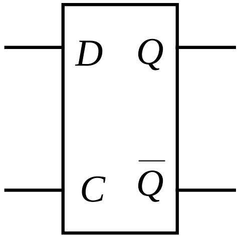
Clocked D Flip-Flop

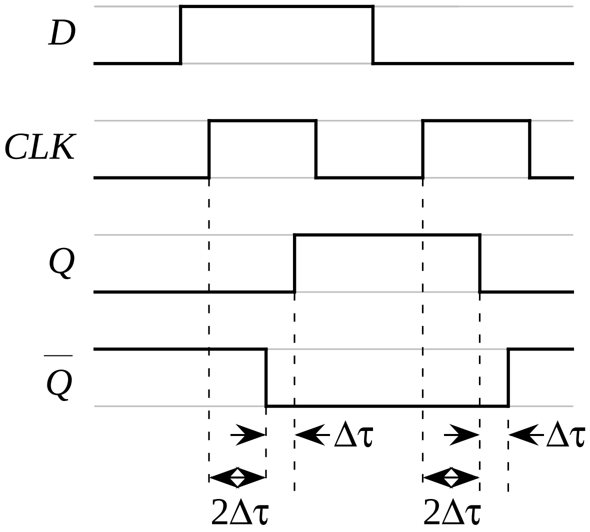
- The clocked D flip-flop, sometimes called a *latch*, has a potential problem: If D changes while the clock is high, the output will also change. The *Master-Slave* flip-flop (next slide) addresses this problem.



Circuit

Symbol

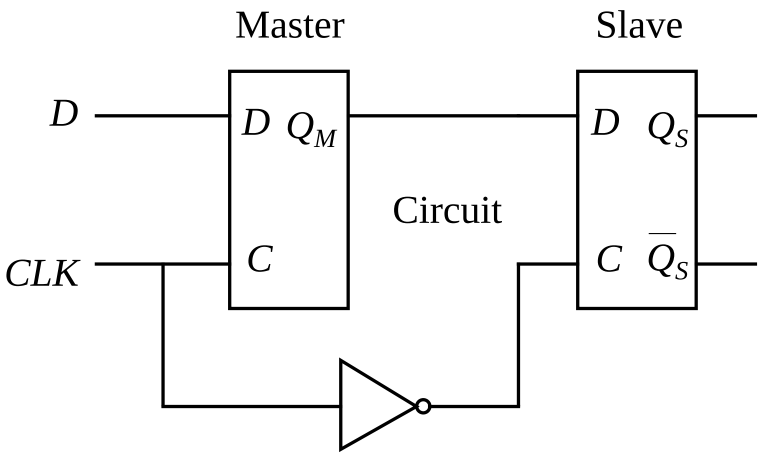




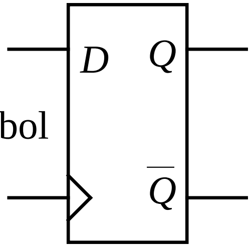
Timing Behavior

Master-Slave Flip-Flop

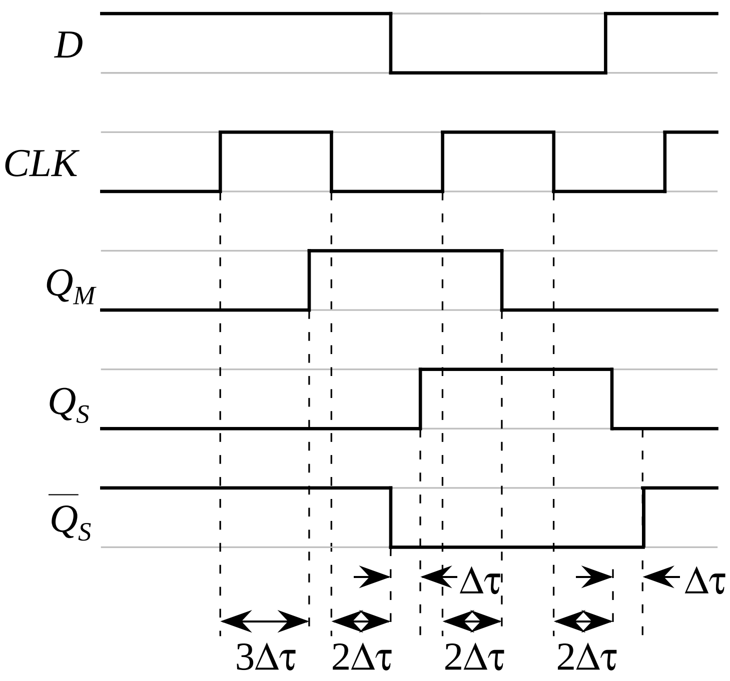
- **The rising edge of the clock loads new data into the master, while the slave continues to hold previous data. The falling edge of the clock loads the new master data into the slave.**



Circuit



Symbol



Timing Behavior

Latches vs Flip-Flops

- **Latch**
  - ◇ Output changes right after the input changes
  - ◇ No reference to clocking event
  - ◇ M&H's "SR flip-flop" if often called an SR latch in other texts.
- **Level-Sensitive Latch**
  - ◇ A latch that operates only when the clock is high or only when low
  - ◇ M&H's "clocked SR flip-flop" is a level sensitive latch
- **Flip-Flop**
  - ◇ Reserved for circuits that record the input only during clocking events
  - ◇ The output of the flip-flop does not change during this clocking event
  - ◇ M&H's "master-slave flip-flop" fits this definition

J-K Flip-Flops

- Allows both “set” and “reset” to be 1
- When both J and K are 1, the output toggles
- If the clock is high, “endless toggle” occurs
- Master-slave J-K flip-flops solve the endless toggle problem, but has the ones-catching problem
- Use edge-triggered flip-flops to eliminate the ones-catching problem.

Next Time

- Edge-triggered Flip Flops
- Introduction to Finite State Machines