UNITED STATES AIR FORCE ACADEMY

ECE 281

Lesson 17 Notes

Objectives:

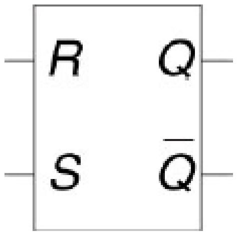
- Recognize and explain the difference between combinational and sequential logic
- Be able to demonstrate how to build a flip-flop from smaller parts (D-latch, SR latch)
- Show the difference between a latch and flip-flop by graphing their respective outputs given a timing diagram of their inputs

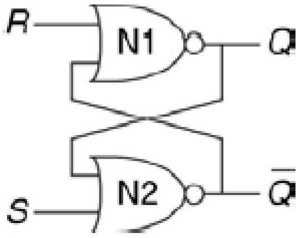
Up to this point we have been discussing primarily combinational logic, associated components and logic. We will now transition to start discussing sequential logic and associated designs.

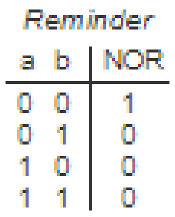
Combinational logic: dependent on

Sequential logic: dependent on

SR Latch: (Set / Reset Latch): Simplest circuit for storing a bit. From zyBooks, an SR latch stores one bit with an input s to set the latch to 1, in input r to reset the latch to 0, and with the stored bit appearing on output Q. Hence you can think of S as “set” and R as “reset”. Clearly, the timing or history of inputs now becomes important.







Reminder

a	b	NOR
0	0	1
0	1	0
1	0	0
1	1	0

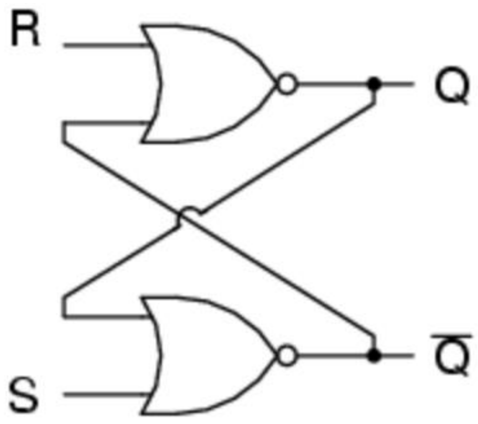
S =

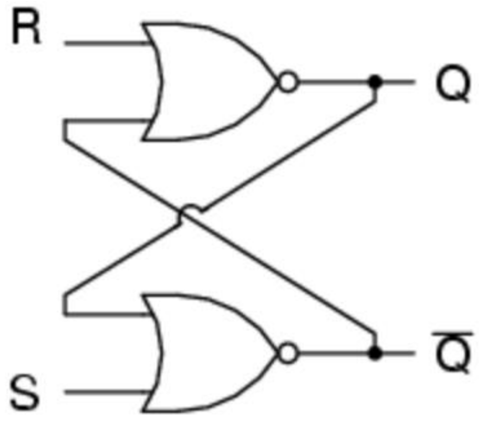
R =

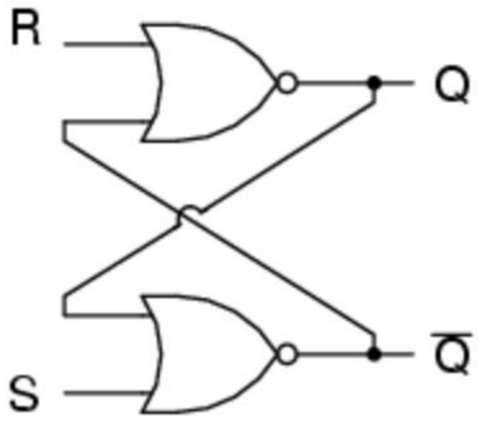
Q =

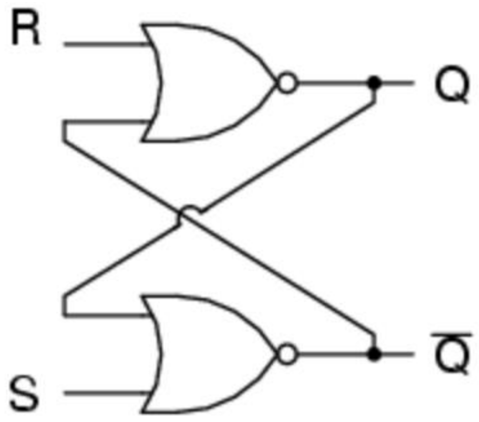
SR Latch Examples:

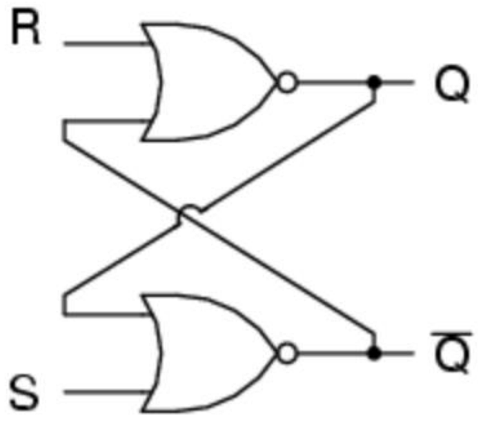
S	R	Q	$\overline{Q}$
1	0		
0	0		
0	1		
0	0		
1	1		







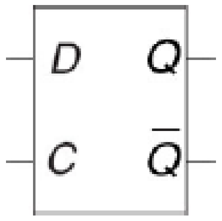


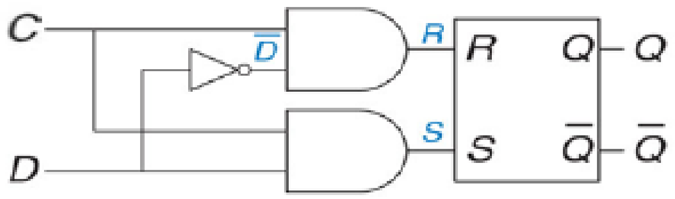


Key problem with SR Latch:

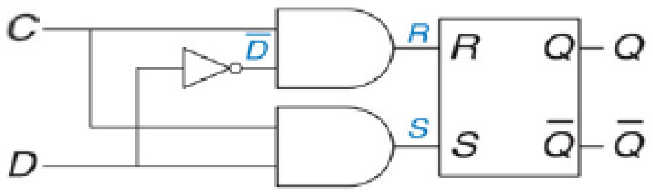
1.

D Latch: (Data / Enable Latch): Think of this as an expansion of the SR latch. From zyBooks, stores one bit, with an input D having the "data" bit to be stored, an input C or E to "enable" storing the bit and an output Q. This design allows for overcoming the problem previously discussed with the SR Latch of "uncertainty" in the S=1, R=1 case.

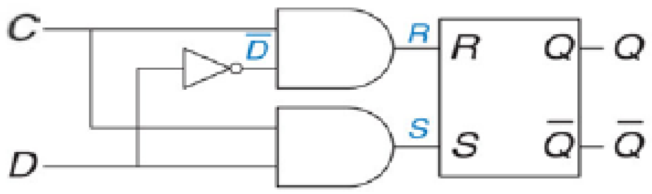




C =
D =
Q =



C	D	Q	$\overline{Q}$
1	0		
1	1		
0	X		

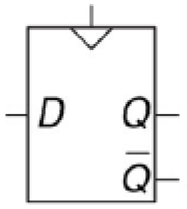


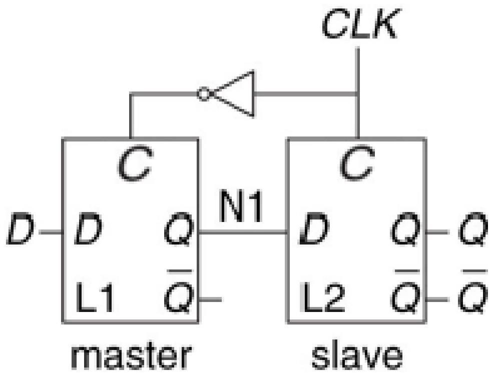
With the SR Latch, Q changes:

With the D Latch, Q changes:

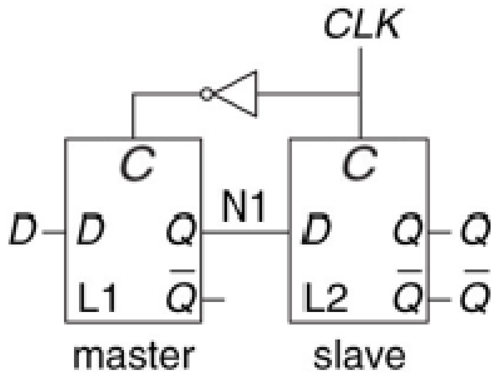
Terminology: You may hear the terms *Opaque* and *Transparent*. The D Latch is **Opaque** when **C is low**, because D does not flow through. Likewise, the D Latch is **Transparent** when **C is high**, because D flows through.

D Flip Flop: From zyBooks, a latch and flip-flop both store a bit. However, while the latch is *level sensitive* and stores a new bit when the Enable or Control is high, the flip-flop is *edge triggered* and only stores a new bit at the instant of a clock input's rising edge.





CLK =
D =
Q =



CLK	D	Q	$\overline{Q}$
0	0		
0	1		
1	0		
1	1		
↑	0		
↑	1		