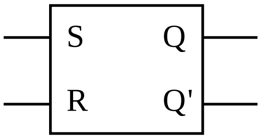
ECE 204 - FLIP-FLOPS AND LATCHES - INVESTIGATION 17
INTRODUCTION TO LATCHES

FALL 2003 **A.P. FELZER**

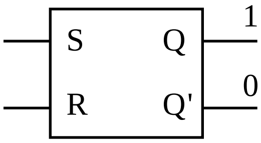
To do "well" on this investigation you must not only get the right answers but must also do neat, complete and concise writeups that make obvious what each problem is, how you're solving the problem and what your answer is. You also need to include drawings of all circuits as well as appropriate graphs and tables.

Up to now we've been analyzing and designing combinational logic circuits - logic circuits characterized by the fact that their present outputs depend uniquely on their present inputs. These circuits are indispenable. But most digital systems like digital watches, digital phones, digital computers, digital traffic light controllers and so on also require memory elements - digital circuits that can store and retrieve data in the form of 1's and 0's. These memory elements are critical because they enable us to build systems with outputs that depend not only on present inputs but also on what has happened in the past. The objective of this Investigation is to introduce digital memory circuits with an introduction to SR latches.

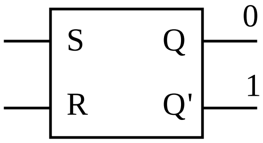
1. SR latches as follows



are the basic building blocks of the digital memory circuits we're going to be introducing in this and the next Investigation. When an SR latch is storing a 1 then its Q=1 and when storing a 0 its Q=0 as follows

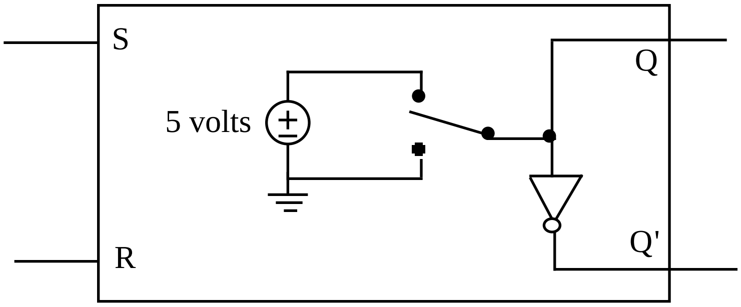


Storing 1



Storing 0

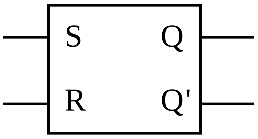
Given the following equivalent circuit of an SR latch



draw the circuit when the SR latch is

a. Storing a 0
b. Storing a 1

2. The output Q of an SR latch as follows



responds to its inputs S and R as follows

(1) If SR=10 then Q=1 and the latch is storing a 1. We call this **setting** the Latch.

(2) If SR=10 → SR=00 then the latch will remain set with Q=1

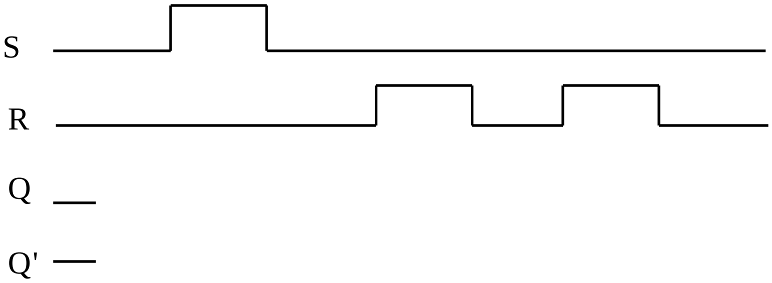
(3) If SR=01 then Q=0 and the latch is storing a 0. We call this **resetting** or **clearing** the Latch.

(4) If SR=01 → SR=00 then the latch will remain reset with Q=0

Memorize these four properties of SR latches.

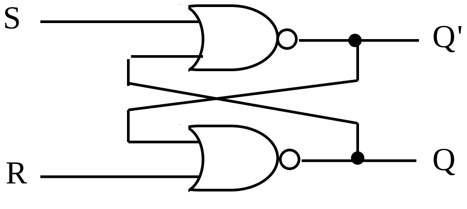
a. The *key feature* of SR latches is that if they're set to Q=1 they'll "remember" to stay set to 1 when SR=10 → SR=00. And if they're reset to Q=0 they'll "remember" to stay reset to 0 when SR=01 → SR=00. How does this behavior differ from that of combinational circuits like the ones we've studied in previous Investigations. Illustrate with an example.

b. Complete the following timing diagram for an SR latch



c. We call the value of Q at any given time the **state** of the latch. **Memorize** what we mean by the state of a latch. Then find the state of a latch with S=0 and R=1.

3. From our description of SR Latches in Problem (2) we know that Q can be either 0 or 1 when SR=00 depending on whether it was set or reset before both inputs became 0. So somehow or other the latch circuitry must take into account its state - its value of Q - when it responds to the input SR=00. The following NOR gate circuit

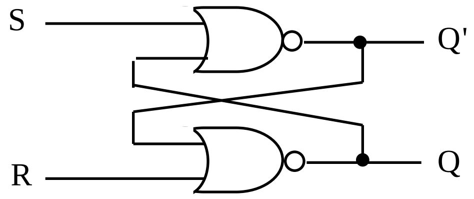


is able to do this because of the **feedback** from the output back to the input - because of the fact that both Q and Q' are "brought back" and connected to the inputs of the NOR gates. **Memorize** the definition of feedback.

The objective of this problem is to verify that this NOR gate circuit really is an SR latch. Be sure to note that Q and Q' are in opposite locations from those on the chip diagram.

a. Write the equations for Q and Q'
b. Make use of your equations in part (a) to show that Q=1 when SR=10. Hint - First find Q' and then Q
c. Make use of your equations in part (a) to show that Q stays equal to 1 in part (a) when SR=10 → SR=00. Hint - first show that S → 0 does not affect Q' and therefore does not affect Q
d. Make use of your equations in part (a) to show that Q=0 when SR=01
e. Make use of your equations in part (a) to show that Q stays equal to 0 in part (a) when SR=01 → SR=00.

4. Up to now we've been avoiding the input SR=11. The reason is that the response of our latch as follows



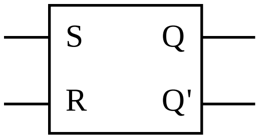
when SR=11 → SR=00 depends on which NOR gate responds first

a. First confirm that when both SR=11 then Q=0
b. Then draw a timing diagram to illustrate the fact that Q will stay equal to 0 when SR=11 → SR=00 if S goes to zero faster than R does.
c. And then draw a timing diagram to illustrate the fact that Q will change to 1 when SR=11 → SR=00 if R goes to zero faster than S does.
d. Make use of your results in parts (b) and (c) to explain why it's not a good idea to allows SR=11 in an SR latch

5. From the results of the last problem we see that when SR=11 then the operation of an SR latch is at best problematic. We can't really be sure what will happen to Q in a real circuit when

SR=11 → SR=00

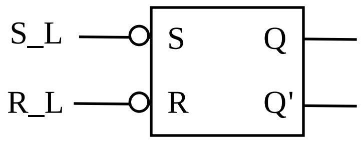
As a matter of fact an SR Latch with SR=11 may actually get into what's called a **metastable** state and start oscillating. So we simply "forbid" our SR latches to have have the input SR=11. We do this by either using them in situations where it's physically impossible for the input to make the transition SR=11 → SR=00 or we use them in situations where we add circuitry to prevent SR=11. So when we write out a **characteristic table** for a latch with active high inputs as follows



S	**R**	**Q**
0	0	No Change
0	1	0
1	0	1
1	1	Not Allowed

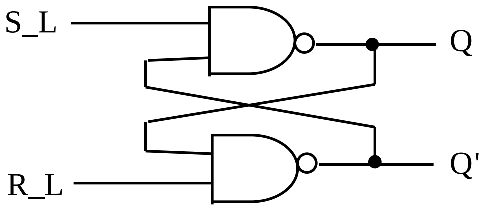
we write in "Not Allowed" for the state SR=11

a. Write out the characteristic table for an SR latch with active low inputs as follows



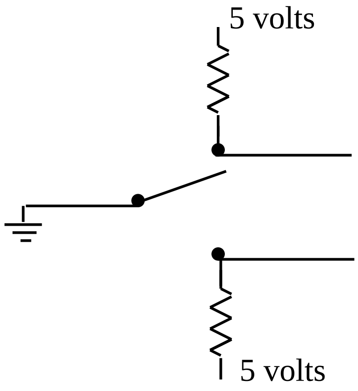
b. What's the relationship between the Not Allowed inputs of active high and active low SR latches

6. The objective of this problem is to analyze an active low SR latch built from NAND gates as follows

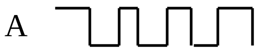


   a. Write the equations for Q and Q'
   b. Make use of your equations in part (a) to calculate the characteristic table
   c. Make use of your characteristic table to verify that this circuit really is an active-low SR latch

7. One particularly useful application of SR latches is in the elimination of "contact bounce" in a circuit like the following

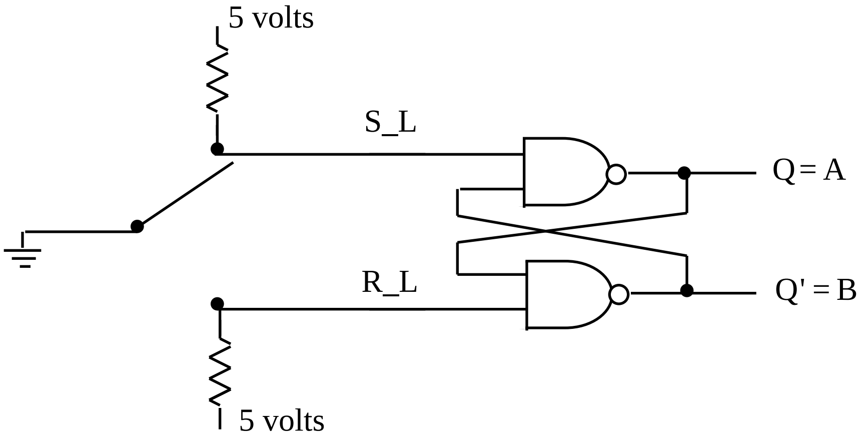


Whenever we flip a switch like the one in the above circuit the contact will make and break at least a few times before settling down as illustrated in the following timing diagram

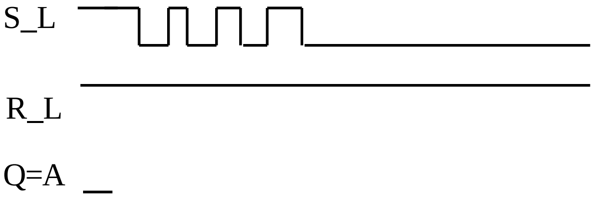


.

These bounces can cause a big problem if for example something is supposed to happen in the circuit every time A → 0 or B → 0. One convenient way to avoid the contact bounce problem is to insert an SR latch with active-low inputs made from NAND gates as follows



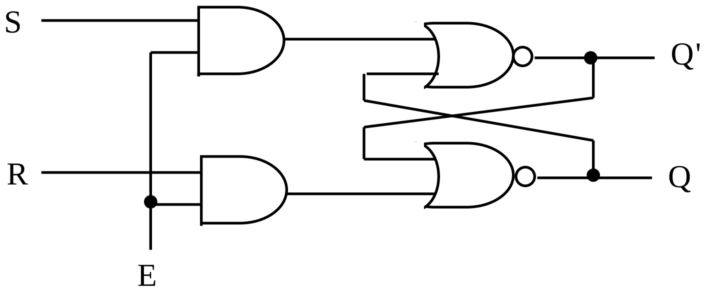
a. First explain why we don't have to worry about the inputs to the active-low SR latch equaling the "not allowed" values S_L=R_L=0 in this circuit
b. Complete the following timing diagram



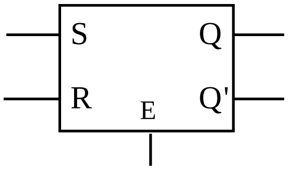
c. Now explain how the SR latch prevents contact bounces from affecting Q=A

8. SR latches like the ones we've been working with are great in applications like switch debouncers but when we try to use them in more sophisticated applications with changing system inputs we find that the states of the SR latches can unintentionally change before their inputs have a chance to settle down to stable values.

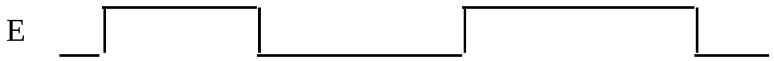
   One way to eliminate this problem is to add enable circuitry as follows



to give us an SR latch with a logic diagram as follows



The idea is to make E a signal like the following



with E=0 while we're waiting for the signals in the circuit to settle down. And E=1 when we want the latch to respond to the new stable values of S and R. We say that the latch is **transparent** when E=1. As a result we say our SR latch is **level sensitive** because its response depends on the level of the enable E. **Memorize** these terms for latches. Then complete the following timing diagram