Introduction to Sequential Systems: The R-S Latch

Sequential Logic

The values output by a combinational logic circuit (which is all that we have dealt with in EC262 up until now) depend only on the current input values. Previous input values do not affect the current output values; only the current input values matter.

Sequential logic circuits are fundamentally different. The output values still depend, of course, on the current input values. But the output values of a sequential logic circuit also depend on *previous* input values. In other words, the same inputs may result in different output values, depending on the past history of inputs.

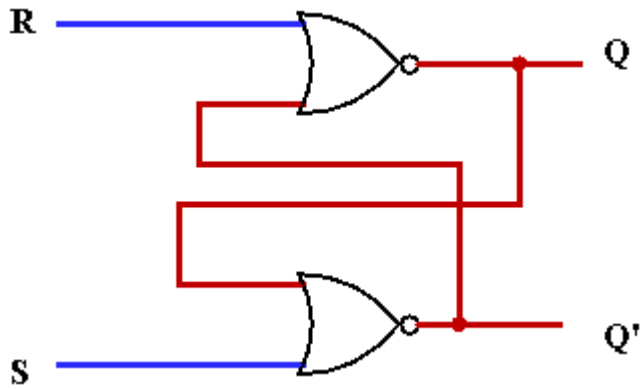
Another way of saying this is that sequential circuits _____.

For a circuit's output to be dependent on the past, what must be the case?

Note that this is in stark contrast to a combinational logic circuit, which was defined as a circuit made up of basic logic gates with the proviso that the output of a gate may not eventually feed back to that same gate.

The R-S Latch

Consider the circuit shown below, called an *R-S* latch.



Suppose that $R = 1$ and $S = 0$. Determine the outputs Q and Q'.

The input R stands for

Let's say we now set S to 1 and R to 0. Determine the outputs Q and Q'.

The input S stands for

Let's say that at this point (S to 1 and R to 0) we set S to 0. What happens?

Suppose I am bored and toggle the value of S between 0 and 1 2014 times. What will be the value of Q on toggle number 983?

What if I want to

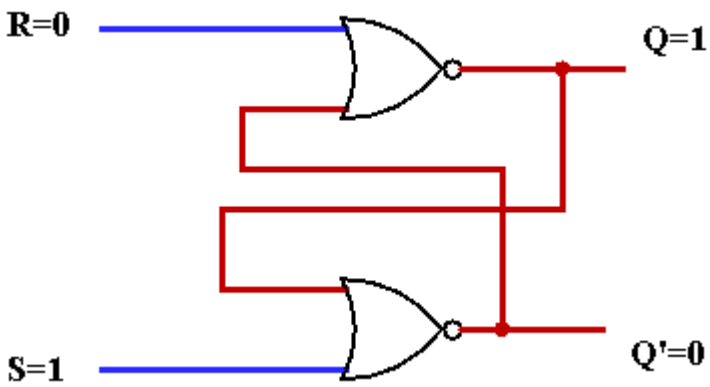
Summary:

Placing a 1 on *S* sets
Placing a 1 on *R* sets

We say that the value stored by this device—the output *Q* of the *R*-*S* latch—has two states: $Q = 0$ and $Q = 1$. The operation of the inputs causes the output *Q* to change its state from 0 to 1, or change its state from 1 to 0, or remain in its current state (0 or 1).

Do you see a problem with the *R*-*S* latch?

Let's look at the example on the prior page, where we had the stable configuration $S = 1$, $R = 0$, $Q = 1$, $Q' = 0$.



Now, set _____. What happens? This

situation is mildly distasteful because

Can you think of a potential use for this circuit under the operation described above?

Would this circuit under the operation described above be useful as a memory device?

How can we prevent this situation from arising?

Write the truth table for the R-S latch, disregarding the output Q' in your truth table.

State Tables

The truth table above is not along the lines of those examined before…

For sequential circuits let's modify the truth table to show the "current inputs", the "current state" and the state that results (i.e., the next state).

Stated another way, let's consider the inputs to be the values of R and S at $t = 0$ and the value of the current state Q at $t = 0$. The output will be Q after the circuit delays are complete. To differentiate between the value of Q after circuit delays are complete and the value of Q at $t = 0$, we will denote the former as Q^*.

In other words, consider Q to be the value of the output at $t = 0$, and Q^* to denote the value of the output at $t + \Delta$, where Δ is the propagation delay of the circuit.

This modified truth table will be termed a *state table*.

Write the state table for the R-S latch:

State Diagrams

We will find it very useful to represent the operation of sequential circuits using *state diagrams*. The state diagram consists of nodes, representing the states of the circuit, and arcs, which are directed arrows pointing from a beginning state to an ending state.

The states of a circuit represent _____ .

The arcs are labeled with input combinations that cause the transition from one state to the next. (Note that the ending state might be the same as the beginning state.)

So, for the *R-S* latch, let's consider the "state" to be the output Q since that output holds the stored value.

How many states do we have?

So, our state diagram starts as:

Since the $R = 1$, $S = 1$ input combination is not allowed, how many possible input combinations are there to consider for each state?

So, each of these states will have _____.

Let's look at the state $Q = 0$. Suppose we have $R = 0$, $S = 1$. What happens?

This is shown as:

Still considering the state $Q = 0$, suppose we have $R = 0$, $S = 0$. What happens?

This is shown as:

Complete the state transition diagram for the R-S latch.

Clocks

We saw when looking at the R-S latch that an instability can arise if we allow R = 1 and S =1, and then drive R and S to 0 at the same time. This instability is avoided by not permitting the state R =1 and S =1 to occur in the first place.

This type of instability problem can occur in other ways. If a number of R-S latches are interconnected together, it is important that the various elements switch their states at the same time, in a coordinated and controlled fashion. This can be accomplished by distributing a periodic *enable signal* throughout the system, with each memory device only permitted to switch states when enabled. This single enable signal, which periodically activates, thereby enabling all of the devices at the same instant, ensuring that all state shifts are synchronized, is called

A system of sequential elements that all change their states at the same time is termed a

In a synchronous system, all sequential elements can change state only once per periodic clock signal, and no elements can alter their states between clock signals.

Clock signals, such as those shown below, are characterized by two values.

From A. Marcovitz, *Introduction to Logic Design*, McGraw Hill, 2010

- The _____, which is the
  The

- (i.e., 1 for active high, 0 for active low).

The signals above have the

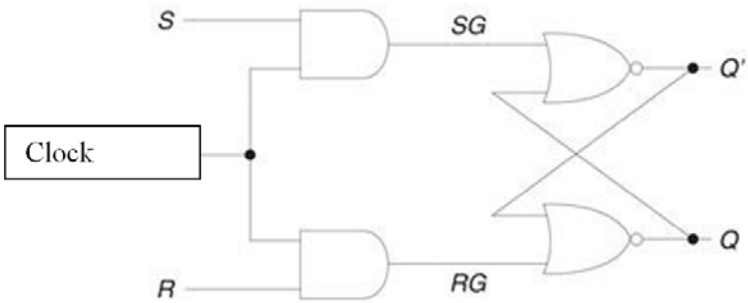
Do you think that we want clocks to have a high frequency or a low frequency (or it doesn't matter)?

Most good computers have a clock rate of around

Do you think that we want clocks to have a high duty cycle or a low duty cycle (or it doesn't matter)?

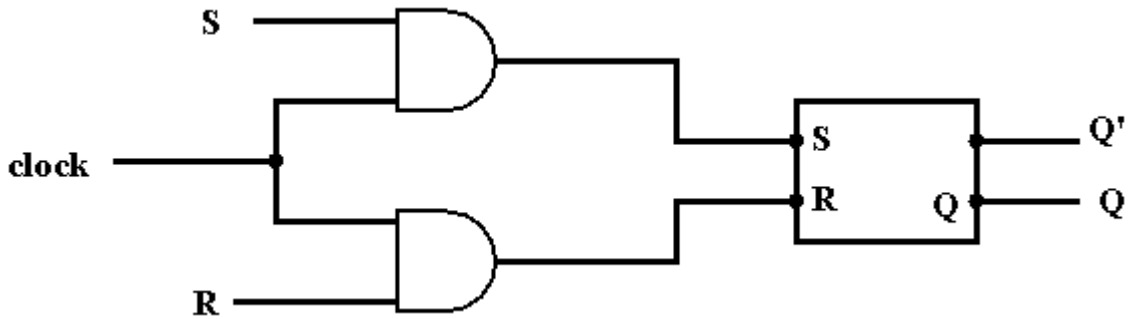
The R-S Latch Under the Control of a Clock

Explain the operation of the circuit below:



From A. Marcovitz, *Introduction to Logic Design*, McGraw Hill, 2010

We must worry about the stability of *S* and *R* only during the time that

Suppose we have that terrible situation of setting *S* and *R* both to 1, and then simultaneously shifting them to 0. What happens?

As a shorthand, we will represent the above picture of the clocked R-S latch as shown below: