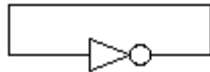


Appendix B.4: Sequential Circuits

The output of combinational circuits depends on the input to the circuit, therefore combinational circuits have no memory.

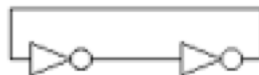
In order for the circuit to have memory, its output must depend on both the inputs to the circuit and the circuit's previous state.

An example of a sequential circuit is an inverter with feedback:



The output repeatedly oscillates between zero and one, so this is not a stable circuit.

Here's an example of a stable sequential circuit:



For a circuit to have memory it must have the following characteristics:

1. The circuit must have two stable states.
2. There must be a way to read the state of the device.
3. There must be a way to set the state of the device at least once.

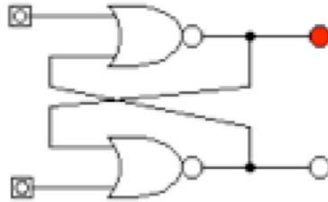
Two types of sequential circuits are:

1. Latches: circuit that does not have a clock input. State changes occur in response to changes in input.
2. Flip-flops: circuit that does have a clock input. State changes occur in response to a clock pulse.



S-R Latch

Consider the following:



Q1: If the output of one NOR gate is true, what is the output of the other gate?

Q2: What happens if the upper input (S) is turned to true?

Q3: What happens if the upper input (S) is turned back to false? Why?

An S-R Latch is also called a Set-Reset latch. An input to S sets the latch. An input to R resets the latch. In either case the output is stable when the input is turned off.

The state (output) is only changed when the complimentary input is set or turned on. This circuit is bistable because it has two stable states.

Q4: Having both S and R turned on is not allowed. Why?

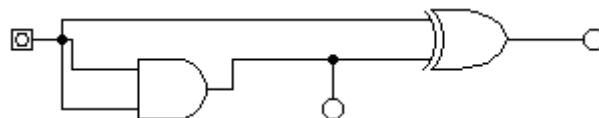
The S-R Latch can be described using a characteristic table:

Characteristic Table		
Current Inputs	Current State	Next State
SR	Q_n	Q_{n+1}
00	0	
00	1	
01	0	
01	1	
10	0	
10	1	
11	0	
11	1	

Hazards

The switching time of a transistor is a few nanoseconds, but there is a slight delay between the switching of the input and the changes happening in the output. This is called a gate delay.

Consider the following circuit:



Q5: What is the output if the input is 1? What about 0?

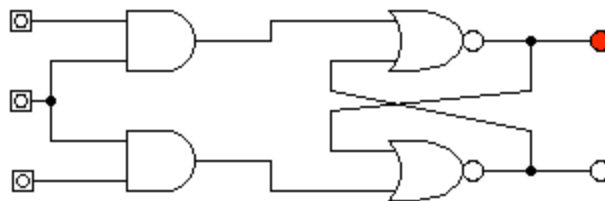
Q6: Now, consider what happens when the input changes given a slight gate delay. What is the problem?

A circumstance where timing delays could cause an error is called a hazard.

Q7: What would happen if the output of the above circuit were inputted into an S-R latch? This is called a glitch.

Clocked S-R Latch (Flip-flop)

To avoid glitches, we want to design memory circuits to only accept input when ordered to do so. The way to achieve this is to interpose a couple of AND gates with a control signal.



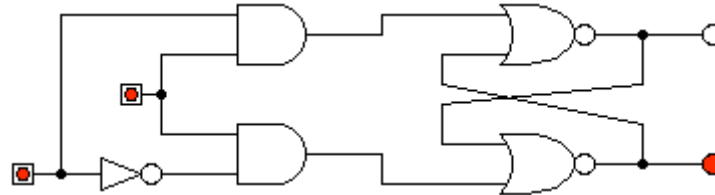
Q8: What happens when the control signal is off and we set S? How about if we set R?

Q9: What is the output if the control signal is on and we set S? How about if we set R?

Q10: Does clocking help with the problems we get when we set both S and R to 1?

Clocked D Flip-flop

We really just want to store one bit of information. Setting and resetting the flip-flop is added complexity. What we need is one input (D) and one output (Q). When the clock signal is on, we want to store in Q is whatever is in D.

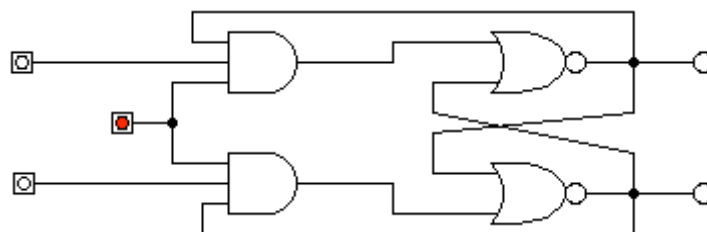


Q11: This flip-flop only works if it is clocked. Why?

Characteristic Table	
D	Q_{n+1}
0	
1	

J-K Flip-flops

Another way to fix the problem (race condition) with S-R flip-flops is to use a J-K flip-flop that toggles the output when both S and R are set.



Characteristic Table		
J	K	Q_{n+1}
0	0	Q _n
0	1	0
1	0	1
1	1	(Q _n)'