TutorTube: Sequential Logic Analysis

Spring 2021

Introduction

Hello and welcome to TutorTube, where The Learning Center's Lead Tutors help you understand challenging course concepts with easy to understand videos. My name is Jeff, Lead Tutor for Electrical Engineering. In today's video, we will explore analyzing sequential circuits. Let's get started!

Sequential Logic Overview

Sequential logic is an extension of the combinational logic we've seen so far. The advantage of sequential logic is that it presents a memory element to the circuit. This means that the next output will depend on the previous outputs. This will make our circuits more powerful and able to achieve more complex functions.

We have two devices that we can use for sequential logic: latches and flip-flops. The primary difference between the two is that latches are asynchronous, and flip-flops are synchronous. This means a latch will respond immediately to a change in input if the clock signal is high. This also means that the output of the latch is not guaranteed to be periodic. A flip-flop will only respond to a change in input on the rising or falling edge of the clock signal, depending on if the clock input is active high or low. A clock signal is a periodic square wave that is used to control the pacing of the circuit. If it is not specified, you can assume the flip-flop responds to inputs on the rising edge of the clock.

Types of Latches/Flip-Flops

Next, we're going to discuss each type of Latch or Flip-Flop. Throughout this section I will likely only refer to each type of device as just a flip-flop but know that a latch or flip-flop with the same name, exhibit the same pattern of behavior, the only difference is how the device reacts to clock signals.

The SR Flip-Flop is the most primitive type of memory device. SR stands for "Set Reset" and describes the behavior of the device. It has two variable inputs, a clock input, and two outputs, which are really just the same output, but one is a complement of the other. To show how it behaves, we use an excitation table, which is the sequential version of a regular truth table. The latch has 3 primary functions: Hold, Set, and Reset. When S and R are both low, the output, which we will refer to as the state of the device, is the same from whatever state was before. If S is high and R is low, the output becomes high. If R is high and S is low, then the output is reset back to low. Both the Set and Reset operations change the output to 1 and 0 respectively regardless of what the previous state was. The combination of S and R being high is considered forbidden because it will cause the two outputs to become the same due to the design of the device, which causes problems since they are supposed to be complements of each other.

S	R	$Q_{present}$	Q_{next}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

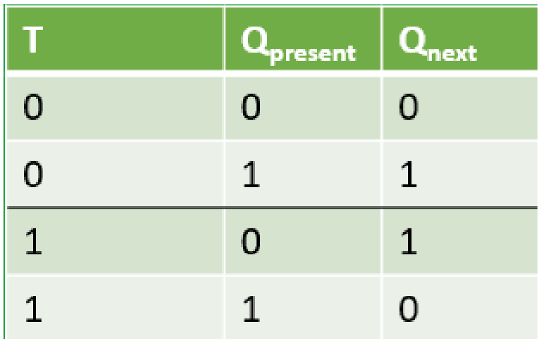
Table 1 - SR Flip-Flop Excitation Table

The D Flip-Flop is a simpler device, one that has just a single input besides the clock. Its name means "Delay" and it acts like a buffer that will hold an input until the proper time. When it receives a 0, it will output a 0, and when it receives a 1 it will output a 1, regardless of the state prior. D Flip-Flops are great for slowing down a circuit with feedback loops from racing on themselves.

D	$Q_{present}$	Q_{next}
0	0	0
0	1	0
1	0	1
1	1	1

Table 2 - D Flip-Flop Excitation Table

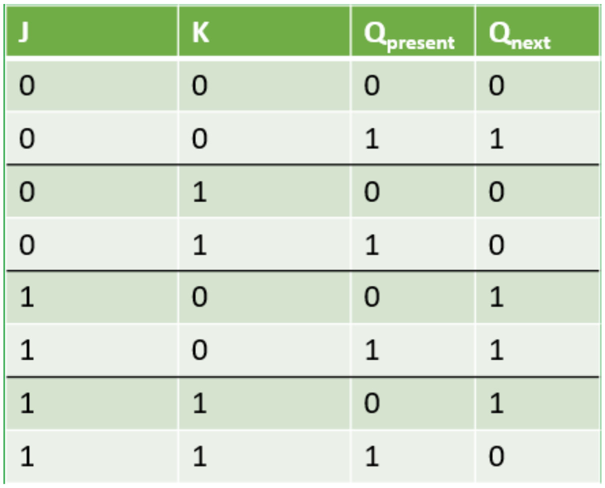
The T Flip-Flop looks like the D Flip-Flop but works slightly differently. The T stands for "Toggle," and has two functions: Hold and Toggle. If the input, T, is low, then the output remains the same as whatever it was before, just like the hold function from the SR Flip-Flop. If the T input is high, then the output is opposite from whatever it was before. If the output was high, then it becomes low and vice versa.



T	$Q_{present}$	Q_{next}
0	0	0
0	1	1
1	0	1
1	1	0

Table 3 - T Flip-Flop Excitation Table

The last flip-flop you'll see is the JK Flip-Flop. It's named after the creator, Jack Kilby, who sought to solve the racing issue from the SR Flip-Flop. It's sometimes referred to as the universal flip-flop because it can mimic every other flip-flop, we've seen either inherently, or with some clever connection configurations. Its excitation table is the same as the SR Flip-Flop except it replaces the forbidden state from before with a toggle function, which works that same way as the T Flip-Flop.



J	K	$Q_{present}$	Q_{next}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Table 4 - JK Flip-Flop Excitation Table

Representing Sequential Logic

So how do we visualize and represent sequential logic? We use two things: state diagrams and state tables. They allow us to convey how a system will respond to an input when at a certain state. Each circle on the diagram represents that state, and the arrows describe all the possible paths between states. Every arrow is labeled with the input and output that is required and what will result from each path.

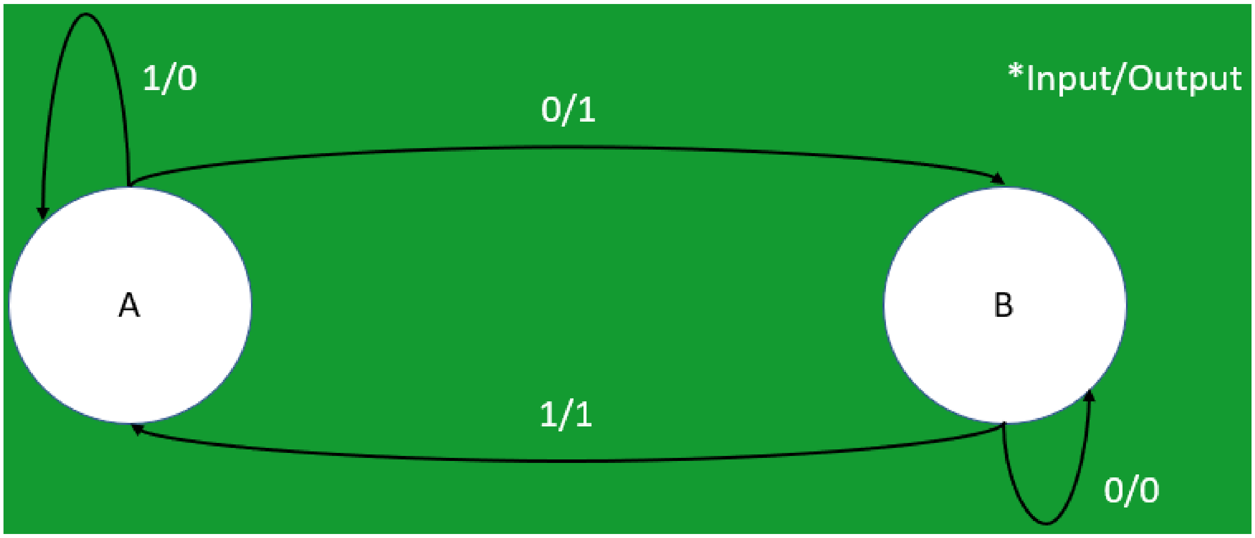


Figure 1 - Example State Diagram

The table describes the same thing, where the present state is treated like an input. For any present state and a certain input, you can determine what the next state will be, and what the output of the system is.

Input	Present State	Next State	Output
0	A	B	1
1	A	A	0
0	B	B	0
1	B	A	1

Table 5 - Example State Table

Example 1

Let's look at an example of analyzing a sequential circuit. Given the circuit below, we're asked to come up with the state diagram and state table that will describe this circuit. Let's start by coming up with the Boolean equations for the output of the system and the inputs to each D flip-flop.

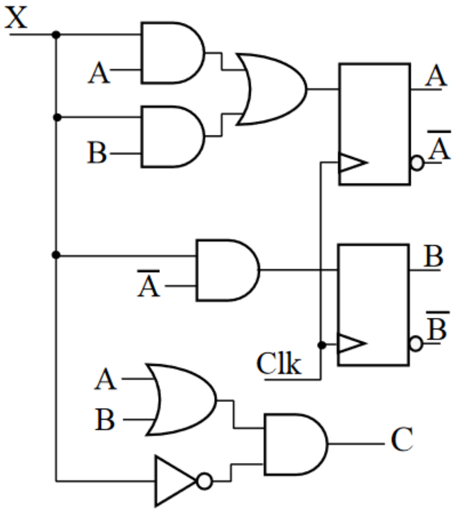


Figure 2 - Example 1 Circuit

- $C = (A+B)X'$
- $D_A = AX + BX = A_{next}$
- $D_B = XA' = B_{next}$

Now we need to be familiar with how a D Flip-Flop behaves. Remember that the output will be set to whatever the input is. So, the next state will be equivalent to those equations that describe the input.

Now with these equations let's fill out our state table. The state table is similar to any other truth table. The Input and present states make up your input side, and your next state and output make up the output side. We'll fill in the left half just to get every possible combination. Now to get each output, we just evaluate each expression from before by plugging in each input.

Input	Present State		Next State		Output
X	A	B	A_{next}	B_{next}	C
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	0	1
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	1	0	0
1	1	1	1	0	0

Table 7 - Example 1 State Table

With the state table done, we can now create the state diagram. We start by drawing circles for each number of states. If you're ever unsure, 2^n where n is the number of flip-flops will tell you how many states you'll have. Each state is labelled with the corresponding present state combination, listed as A and then B. Then we connect an arrow from each present state to its corresponding next state. Finally, we label each arrow with the corresponding input and output separated by a slash. There we go. We have a state diagram that will describe that circuit.

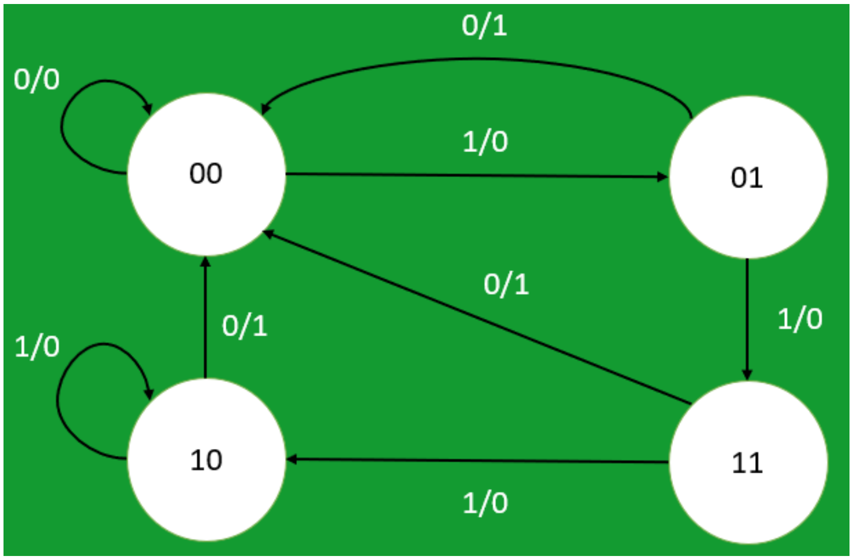


Figure 3 - Example 1 State Diagram

Example 2

Let's do the sample type of problem again, but with a different circuit: this time with JK Flip-Flops. The steps will be the same. We'll start by coming up with the logic equations of the inputs to each flip-flop. We'll also assume that the states will also behave like the output of the system. We do need to know how the JK Flip-Flop behaves, which we know changes depending on what the inputs are.

- J_A = B
- K_A = BX'
- J_B = X'
- K_B = AX' + A'X

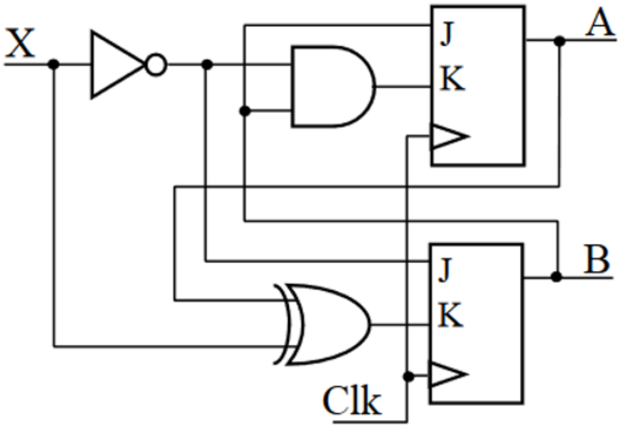


Figure 4 - Example 2 Circuit

We'll start with the state table, which is going to be large but don't worry, it's perfectly manageable. Consider A, B, and input to be the true "inputs" to the table, and the JK Flip-Flop inputs to be transitional. I also have the excitation table of the JK Flip-Flop and the Boolean equations from before on the side to help us remember. Feel free to pause at any point if you need more time to read the table. Let's start with the A flip-flop, whose equations just depend on B and X. Next, we'll fill out the B flip-flop using its equations. Finally, we'll determine the next state of each flip-flop based on what the inputs are. For example, the first inputs are 00, and according to the table, that represents the hold function, so whatever A is, that's what A will continue to be. A was 0, so A_{next} will be 0. Again, pause if you need time to check that each stage of the table makes sense to you because there is a lot of information there.

Input	Current State		JK_A Inputs		JK_B Inputs		Next State	
X	A	B	J_A	K_A	J_B	K_B	A+	B+
0	0	0	0	0	1	0	0	1
0	0	1	1	1	1	0	1	1
0	1	0	0	0	1	1	1	1
0	1	1	1	1	1	1	0	0
1	0	0	0	0	0	1	0	0
1	0	1	1	0	0	1	1	0
1	1	0	0	0	0	0	1	0
1	1	1	1	0	0	0	1	1

Table 8 - Example 2 State Table

Now let's create the state diagram. I've copied the table over from before, but just the columns for the input, present state, and next state. I have two flip-flops like before, so there will be 4 states. From the table, I can place my paths to and from each state. Finally, we can label the paths, and this time we will just write the input, since the system output is just the state value.

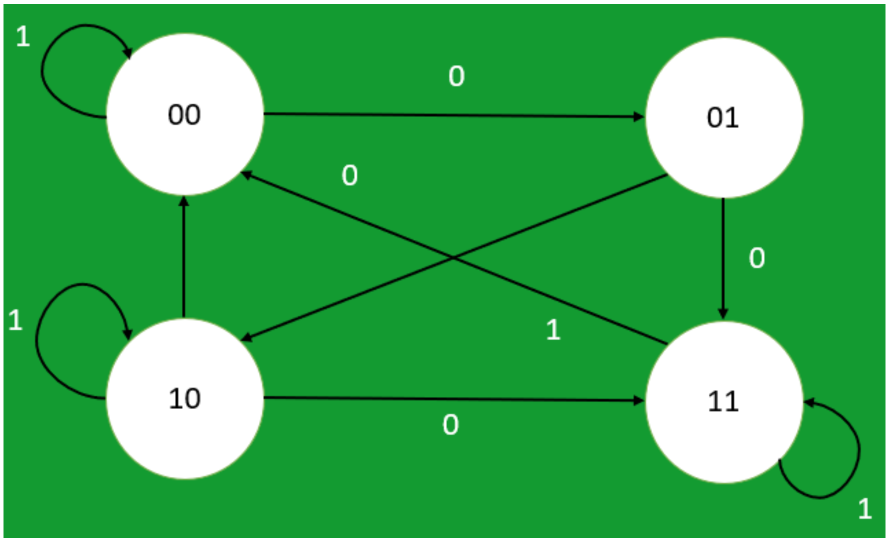


Figure 5 - Example 2 State Diagram

This strategy will work to analyze most circuits, the only piece that you have to remember is the behavior of each flip-flop type.

Our example problems today came from Professor Stroud at Auburn University.

Outro

Thank you for watching TutorTube! I hope you enjoyed this video. Please subscribe to our channel for more exciting videos. Check out the links in the description below for more information about The Learning Center and follow us on social media. See you next time!

References

Stroud, C. E. (n.d.). *Sequential Logic Analysis* [PDF]. Auburn University.

http://www.eng.auburn.edu/~strouce/class/elec2200/elec2200-11.pdf