

Lab 4: Synchronous State Machine Design

Summary:

Design and implement synchronous state machine circuits and test them using a logic analyzer.

Learning Objectives:

- Experience state machine design procedure
- Experience using a PC-based logic analyzer

Resources and Supplies:

- *State Machine Introduction**
- *Logic Analyzer Tutorial**
- *Protoboard Guide**
- *IC Data Sheet**
- *Logic Probe Guide**
- Wire cutters
- IC's kit
- 331 protoboard
- Power supply
- Logic probe
- Safety glasses (required!)

All documents* are available on the class website

Important Reminders:

- Bring your SRB to lab.
- Pre-lab assignments must be completed before coming to the lab.

Background:

Synchronous state sequencer:

Please see the *State Machine Introduction* notes for a more thorough description of state machines and their implementation using flip flops. Figure 1 illustrates the states for a 2 bit sequencer. Synchronous state machines are typically implemented using flip-flop circuits. In this lab you will be asked to design and implement a state machine using the DM7476 J-K FF IC that can be reset to the initial 0,0 state when the momentary switch button on your SRB is pressed. This is called an asynchronous reset because it acts regardless of the clock state. You will use the PR and/or CLR inputs of the DM7476 and simple logic to realize the reset function.

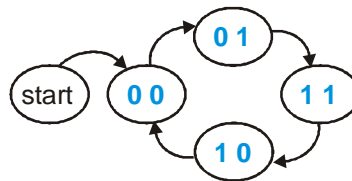


Figure 1. States in an example 2-bit synchronous state sequencer.

To construct the Figure 1 state sequencer using JK flip flops, you would follow the same steps as in the *State Machine Introduction* for an SR flip flop. First, you need the complete JK flip flop excitation table, which is copied below from the *State Machine Introduction*.

truth table				excitation table			
J	K	Q^{t+1}	action	$Q^t \rightarrow Q^{t+1}$	action	J	K
0	0	Q^t	hold	0 \rightarrow 0	hold (00) reset (01)	0	X
0	1	0	reset	0 \rightarrow 1	set (10) toggle (11)	1	X
1	0	1	set	1 \rightarrow 0	reset (01) toggle (11)	X	1
1	1	$\overline{Q^t}$	toggle	1 \rightarrow 1	hold (00) set (10)	X	0

Table 1. Truth table and excitation table for a JK flip flop.

The following steps provide an example of how the state sequencer in Figure 1 could be designed using JF flip flops.

step 1: next-state excitation table

Table 2 shows each *current* state of the Figure 1 sequencer and its associated *next* state that defines the sequencing of the state machine. Using the JK excitation table (Table 1), the required input values (J and K) for each state can then be determined, as shown in Table 2. **It is imperative that you understand how Table 2 was generated because you will be repeating this in lab for a more complex sequencer.**

current state		next state		bit 1 inputs		bit 0 inputs	
Q_1^t	Q_0^t	Q_1^{t+1}	Q_0^{t+1}	J_1	K_1	J_0	K_0
0	0	0	1	0	X	1	X
0	1	1	1	1	X	X	0
1	0	0	0	X	1	0	X
1	1	1	0	X	0	X	1

Table 2. Next-state excitation table for Fig. 1 sequencer.

step 2: Input truth tables

The data in Table 2 can be rearranged to show the truth table for each bit based on the current-state output values. Table 3 shows the truth tables for all inputs, uniquely defining the inputs needed to match the sequence in Figure 1. **It is important that you understand how Table 3 was generated because you will be repeating this in lab for a more complex sequencer.**

Q_1^t	Q_0^t	J_1	Q_1^t	Q_0^t	K_1	Q_1^t	Q_0^t	J_0	Q_1^t	Q_0^t	K_0
0	0	0	0	0	X	0	0	1	0	0	X
0	1	1	0	1	X	0	1	X	0	1	0
1	0	X	1	0	1	1	0	0	1	0	X
1	1	X	1	1	0	1	1	X	1	1	1

Table 3. Input truth tables based on current-state outputs of Fig. 1 sequencer.

step 3: Minimize logic expression for all inputs

Table 3 defines each flip flop input as a function of the 2-bit flip flop outputs (Q_1 - Q_0). We must next minimize this expression to define the simplest logic circuit possible for each input. The K-maps in describe the relationship for each input and can be used to determine the minimized logic expression shown below each K-map. **Again, you must be able to perform this step during lab for a more complex sequencer.**

		J_1	
		Q_0	
		0	1
Q_1	0	0	1
	1	X	X

		K_1	
		Q_0	
		0	1
Q_1	0	X	X
	1	1	0

		J_0	
		Q_0	
		0	1
Q_1	0	1	X
	1	0	X

		K_0	
		Q_0	
		0	1
Q_1	0	X	0
	1	X	1

$J_1 = \underline{Q_0}$, $K_1 = \overline{\underline{Q_0}}$, $J_0 = \overline{\underline{Q_1}}$, $K_0 = \underline{Q_1}$

step 4: Circuit implementation

Based on the equations from step 3, you should now be able to sketch the schematic of the Figure 1 2-bit sequencer using the DM7476 J-K FF IC and, if necessary, basic logic gates. This is an exercise in the Pre-lab.

State machine -unused states:

Figure 2 shows the state model of a specific 3-bit sequence generator that operates only in the 001, 010, 110, 011 states. Any of the remaining states (000, 100, 101, 111) should return to 110 so the sequencer can continue to operate within 4 states only.

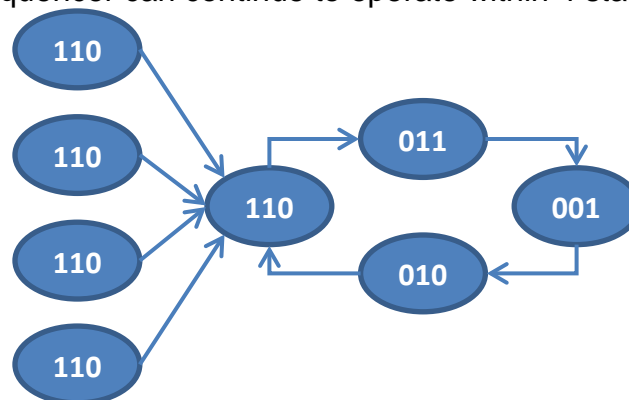


Figure 2. 3-bit state sequencer.

Logic Analyzer:

A logic analyzer is a very useful tool in digital hardware design that facilitates observation of circuit output waveforms. It displays signal logic states and permits observation and recording signal values over time. In this lab you will use the PC-based *NCI GoLogic* logic analyzer to monitor the signals in a simple counter (sequencer) application. A tutorial provided with this lab will walk you through the proper connections and setup.

Pre-lab Assignment:

- Read this entire lab assignment so you know what to expect in the lab.

- Complete the steps described in the Pre-lab sheet near the end of this document. Each student must complete his/her own pre-lab before coming to the lab and hand it in to the lab TA at the beginning of the lab.

Laboratory Assignment:

This lab consists of two parts. A check-off sheet is included at the end of this lab document to record your design work and results. After you successfully finish each part of the lab, show the TA your results and ask him to sign the check-off sheet.

- Print the check off sheet.

Part 1: Two bit sequence generator.

In this part you will implement the Figure 1 2-bit state sequencer JK flip-flops based on your pre-lab design. Remember to include the wire diagram from pre-lab in your report.

Implementation:

1. Using your wiring diagram from pre-lab, place the necessary ICs on the protoboard and wire your complete circuit. Be sure to connect the power and ground of each IC.
2. Connect the 1 Hz clock generator from the SRB to the clock input of the circuit. It may also be useful to connect this SRB output to an LED on the SRB.
3. Connect the reset signal to the momentary switch on the SRB.
4. Connect the circuit's two-bit output to SRB LED inputs making sure you know which LED belongs to each output bit.

Inspection& IC Testing Precautions

Follow the circuit inspection and testing precautions detailed in Labs 2 and 3. Please refer back to those documents if you have any concerns.

Testing:

The 2-bit output of the 2 bit sequencer should be connected to two LEDs through the DB connector. You will test the circuit using the 1 Hz clock generator.

5. Without connecting the power supply to your protoboard or SRB, turn it on to set the proper voltage (approximately 8V). Turn it off, connect it to your protoboard and SRB, and then turn on the power supply to power up your circuits.
6. Use the logic probe to test the values of the power supply pins and the circuit inputs (J, K, clock, etc.). Note on the check-off sheet the 2-bit output value (Q from JK FFs) and note if the value is changing or constant. If it is not changing, check your connections and your design and repeat until the outputs are changing.
7. Press the reset button (SRB momentary switch) and record a comment on your observations.
8. Record the sequence as the clock triggers the circuit. Remember you can press the reset to start the sequence over.
9. If the sequence you observe on the LEDs is not correct, check your design and repeat experiments until you get the correct sequence. Consult the TA if necessary.

Demonstration:

10. Follow the step-by-step instructions in the *Logic Analyzer Tutorial* to practice using the PC-based logic analyzer. Ask the TA if you have any questions about the use of the logic analyzer before continuing.
11. Connect the outputs of your 2-bit sequencer to the NCI Logic Analyzer and observe the proper state sequence. Print the output waveforms, one copy for each student. Write a descriptive label/title on each plot and attach these to your lab report.
12. When you are satisfied that the circuit is operating correctly, ask the TA to check a demonstration of your functioning sequencer. Ask the TA to check off Part 1 on your lab check-off sheet.
13. Do not continue until the TA has checked your Part 1 experiment. Turn off the power supply. Remove the components and wiring from your circuit in Part 1.

Part 2: State machine design challenge.

In part 2 you will implement the 3-bit state sequencer shown in Figure 2 based on your design in the pre-lab.

Implementation:

1. Use the *Wire Diagram Template* to create the wiring diagram of the 3-bit state sequencer circuit. Use any number of DM7476 to develop the circuit. Be sure to label the circuit in your diagram and include the part # of the chips used. In addition to the DM7476 you can use any other ICs. Include the wire diagram in your report.
2. Using your wiring diagram from step 1, place the necessary ICs on the protoboard and wire your complete circuit. Be sure to connect the power and ground of each IC.
3. Connect the 1 Hz clock generator from the SRB to the clock input of the circuit. It may also be useful to connect this SRB output to an LED on the SRB.
4. Connect the reset signal to the momentary switch on the SRB.
5. Connect the circuit's three-bit output to SRB LED inputs making sure you know which LED belongs to each output bit.

Inspection& IC Testing Precautions

Thoroughly inspect your circuit before proceeding.

Testing:

The 3-bit output of the circuit should be connected to three LEDs through the DB connector. You will now test the circuit using a 1 Hz clock generator.

6. Turn on the power supply to provide power to your chips and to the SRB.
7. Use the logic probe to test the values of the power supply pins and the circuit inputs (J, K, clock, etc.). Note the 3-bit output value (Q from JK FFs) and note if the value is changing or constant. If it is not changing, check your connections and your design and repeat until the outputs are changing.

8. Press the reset button to insure it is working as designed. Note what state your circuit goes to following the reset state. Note if this is the correct state.
9. Record the sequence as the clock triggers the circuit. Remember you can use the reset to start the sequence over.
10. If the sequence you observe on the LEDs is not correct, check your design and repeat experiments until you get the correct sequence. Consult the TA if necessary.

Demonstration:

11. Connect the outputs of your 3-bit sequencer to the NCI Logic Analyzer and observe the proper state sequence. Print the output waveforms: one copy for each student. Write a descriptive label/title on each plot and attach these to your lab report.
12. When you are satisfied that the circuit is operating correctly, ask the TA to check a demonstration of your functioning sequencer. Ask the TA to check off Part 2 on your lab check-off sheet.

Final Tasks and Notes

- Turn off the power supply and anything else that you might have turned on.
- Remove all wires and ICs from the protoboard, returning them to their storage locations.
- Return the protoboard, IC's kit and all tools you used to the lab closet.
- Clean up your lab bench area, removing any trimming to the trash can and any parts/components back to their proper location.

Discussion Points

As explained in the *Lab Report Guide*, you should address these discussion points in a designated section of your report.

1. How many flip flops (bits) are needed for a sequencer for 4 states? How many are needed for 8 states? Can you write a general expression relating the number of flip flops, n , to the number of sequencer states, m ?
2. Explain why the states 000, 100, 101 and 111 in Part 2 must be mapped into a sequential state even though they are not in the desired sequence. What could happen if this was not done?
3. If we wanted to implement a synchronous (rather than asynchronous) reset, where the reset only occurs at a clock transition, how could this be accomplished? Briefly describe the required reset circuitry.
4. For both parts 1 and 2, explain how to interpret the logic analyzer waveforms to verify that your circuits worked properly.

Remember to include your wiring diagrams and logic analyzer plots as attachments to your report.

PRE-LAB 4

Due: At the beginning of lab.

Student Name: _____ **Lab. Section (time):** _____

Make sure you read the lab document before you start the pre-lab, especially the Background section.

1. Read the *Logic Analyzer Tutorial* posted on the class website. Note any questions you might want to ask the TA regarding use of the logic analyzer.
2. Read the *State Machine Introduction* and review again the example JK flip flop implementation of the Figure 1 state sequencer in the Background section. Below, record the logic expressions that define each of the J & K inputs to both of the flip flops (J_1, K_1, J_0, K_0) as a function of current-state outputs ($Q_1, Q_0, Q'_1, Q'_0, \dots$).

$$J_1 = \underline{\hspace{2cm}}, K_1 = \underline{\hspace{2cm}}, J_0 = \underline{\hspace{2cm}}, K_0 = \underline{\hspace{2cm}}$$

3. Based on the equations in #2, draw a schematic of the complete 2-state sequencer using the DM7476 J-K FF IC and any necessary basic logic gates.
4. Use the *Wire Diagram Template* to create the wiring diagram of the 2-state sequencer.

<pre-lab continues on next page>

5. In Part 2 of the lab, you will design and implement the JK 3-bit state sequencer shown in Figure 2. Based on the 2-bit example (from Background and earlier pre-lab questions), complete the following steps to design the 3-bit sequencer.

a. How many JK flip-flops are needed for a 3-bit sequencer? _____

b. Complete the present state/next state table below for the Figure 2 sequencer where $Q_2^n Q_1^n Q_0^n$ are the current states and $Q_2^{n+1} Q_1^{n+1} Q_0^{n+1}$ are the next states.

Q_2^n	Q_1^n	Q_0^n	Q_2^{n+1}	Q_1^{n+1}	Q_0^{n+1}	J_2	K_2	J_1	K_1	J_0	K_0

c. Complete the K-map for each flip flop input, clearly indicating the input on each map.

		Q_1, Q_0			
		0,0	0,1	1,1	1,0
Q_2	0				
	1				

		Q_1, Q_0			
		0,0	0,1	1,1	1,0
Q_2	0				
	1				

		Q_1, Q_0			
		0,0	0,1	1,1	1,0
Q_2	0				
	1				

		Q_1, Q_0			
		0,0	0,1	1,1	1,0
Q_2	0				
	1				

		Q_1, Q_0			
		0,0	0,1	1,1	1,0
Q_2	0				
	1				

		Q_1, Q_0			
		0,0	0,1	1,1	1,0
Q_2	0				
	1				

d. Based on the K-maps above, determine the minimized Boolean expressions for all inputs of the 3 flip flops.

$J_2=$ _____, $K_2=$ _____, $J_1=$ _____, $K_1=$ _____, $J_0=$ _____, $K_0=$ _____

e. On a separate sheet (or on back), draw the schematic of the circuit 3-bit sequencer from Figure 2 using the DM7476 J-K FF IC and any necessary basic logic gates.

LAB 4 CHECK-OFF SHEET

Student Name: _____ **Lab. Section (time):** _____

Complete this sheet as you complete the lab. Remember to have the TA check off each section of the assignment. This sheet must be included in your lab report.

Part 1: Two bit sequence generator

Step 6. Record the observed initial value (Q_1, Q_0). _____

Is the value changing with the clock? _____

Step 7. Comment on using reset input.

Step 8. Record the observed sequence

Does it agree with the correct sequence? _____

Part 1: TA sign off

Part 1: Two bit sequence generator displayed on logic analyzer Initial _____

Part 2: State machine design challenge

Step 7. Record the observed initial value (Q_2, Q_1, Q_0). _____

Is the value changing with the clock? _____

Step 8. What state follows a reset? _____ Is this correct? _____

Step 9. Record the observed sequence

Does it agree with the correct sequence? _____

Part 2: TA sign off

Part 2: State machine design challenge Initial _____