Lab 4: Synchronous State Machine Design

Summary:
Design and implement synchronous state machine circuits and test them with simulations in Cadence Virtuoso.

Learning Objectives:

- Practice designing and simulating digital circuits including flip flops
- Experience state machine design procedure

Resources and Supplies:

- Engineering PC with PuTTY and Xming
- *Lab4A.txt & Lab4B.txt**
- *State Machine Introduction**

All documents* are available on the class website

Important Note Spring 2013:
We have made a change to the 331 *Virtuoso* library. Before using *Virtuoso* again, you must re-run the setup script below in your ECE331/virtuoso directory to install the new library.

source $SOFT/cadence-auto

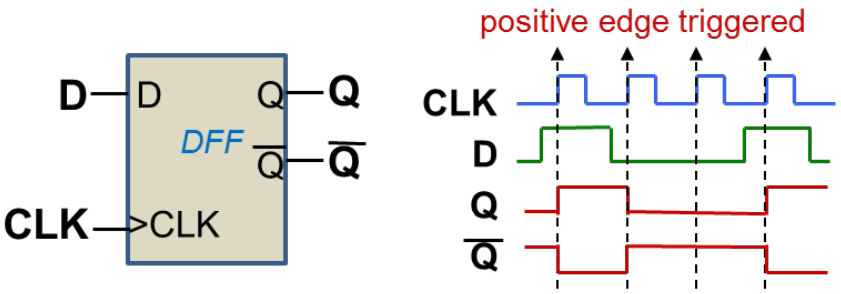
Pre-lab Assignment:
Each student must complete his/her own pre-lab before coming to the lab. Pre-lab check-off sheets must be turned in to the TA before starting the lab assignment.

1. Read the **Background** sections of this lab.
2. Print the Pre-lab 4 Check-off Sheet near the end of this document and perform the required tasks. Show the completed sheet to the TA at the beginning of your lab.
3. Read through the entire **Laboratory Assignment** section so you know what to expect in the lab.

Background:

D-type Flip Flop

Flip flops (FF) are data storage elements for synchronous circuits that have circuit blocks triggered simultaneously and periodically by a clock signal. FFs are used to store the logic state of a signal until the next rising (or falling) edge of the clock signal when the output is reevaluated and stored again. FFs are controlled by a clock signal and one or two data input signals, and they typically have differential (inverted) outputs *Q* and *Q'*. There are several variations of FFs including JK and SR type FFs. However, the most common FF used in digital integrated circuits is the D-type FF (DFF). In a DFF, the value of input D is transferred to the output Q at the rising (or falling) clock edge and held there until the next rising (or falling) edge. The standard schematic symbol for a DFF is shown below along with a timing diagram for a *positive edge triggered* DFF. In this lab you will use a DFF to create state machines. The DFF cell provided in your *Virtuoso* cell library is a positive edge triggered circuit. Refer to ECE331 Handout 0: 230 Review for additional background information on flip flops.



Synchronous state sequencer:

Unless you are very familiar with state machine design, you first need to carefully read the *State Machine Introduction* document for a thorough description of the implementation of state machines using flip flops. Once you are familiar with the state machine design process, return here for information on implementing state machines using DFFs.

Figure 1 illustrates the state transitions for an example 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 D-type flip flops (DFF).

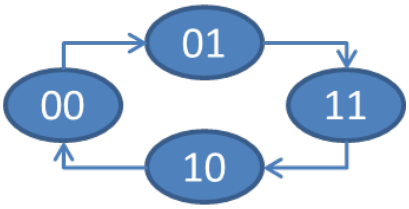


Figure 1. States in an example 2-bit synchronous state sequencer.

To construct the Figure 1 state sequencer using DFFs, you would follow the same steps as shown in the *State Machine Introduction* document for an SR flip flop. However, because the DFF is functionally different than the SR or JK flip flops, it has a different truth table and excitation table resulting in a different state machine implementation. Table 1 shows the truth table and excitation table for a DFF. Because the DFF only has one data input, it has fewer functions and its excitation table is less complex than the SR or JK. This feature generally results in more complex "glue logic" needed to implement a state sequencer.

truth table		
D	Q^{t+1}	action
0	0	hold
0	1	set
1	0	reset
1	1	hold

excitation table		
$Q^t \rightarrow Q^{t+1}$	action	D
0 → 0	hold	0
0 → 1	set	1
1 → 0	reset	0
1 → 1	hold	1

Table 1. Truth table and excitation table for a DFF.

Table 1 shows the operation of the DFF is really quite simple; the next state output, Q^{t+1}, is always the same as the input D at the time of the clock edge trigger.

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

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 DFF excitation table (Table 1), the required input values for each state can then be determined, as shown in Table 2. As you should expect for a DFF implementation, the required inputs are simply the next state output values. **It is imperative that you understand how Table 2 was generated because you will be repeating this in the pre-lab for a more complex sequencer**.

current state		next state		Bit 1 input	Bit 0 input
Q_1^t	Q_0^t	Q_1^{t+1}	Q_0^{t+1}	D_1	D_0
0	0	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	1	1	0	1	0

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 input bit (D_1, D_0) 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 the pre-lab for a more complex sequencer**.

Q_1^t	Q_0^t	D_1
0	0	0
0	1	1
1	0	0
1	1	1

Q_1^t	Q_0^t	D_0
0	0	1
0	1	1
1	0	0
1	1	0

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 state 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 in the pre-lab for a more complex sequencer.**

D_1

Q_1 \ Q_0	**0**	**1**
0	0	1
1	0	1

D_0

Q_1 \ Q_0	**0**	**1**
0	1	1
1	0	0

$D_1 = \_\_\_Q_0\_\_\_$, $D_0 = \_\_\_\_\overline{Q_1}\_\_\_\_$

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 DFF and, as necessary, basic logic gates. This task will be completed as a pre-lab exercise. Note that the DFF provides Q and Q' outputs, so you do not need an INV gate to complement the Q output.

State machine: non-sequenced 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 non-sequenced states (000, 100, 101, 111) should return to 110 so the sequencer can continue to operate within the four sequenced states.

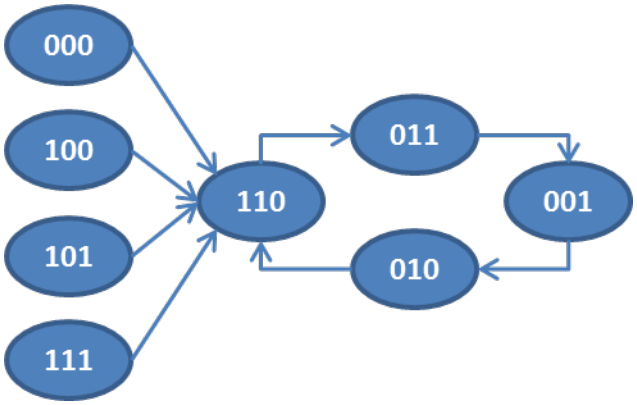


Figure 2. 3-bit state sequencer.

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/her to sign the check-off sheet.

- Print the check off sheet.
- Ask the TA to check your Pre-lab 4 worksheet to ensure you have the correct design and plans for the state sequencers. Fix any mistakes and ask the TA to sign your pre-lab to indicate you have completed the assignment. Keep the pre-lab worksheet after the TA signs it and then attach it to your lab report.

Important Note Spring 2013:

We have made a change to the 331 *Virtuoso* library. Before using *Virtuoso* again, you must re-run the setup script below in your ECE331/virtuoso directory to install the new library.

source $SOFT/cadence-auto

Part 1: Two bit sequence generator.

In this part you will implement the Figure 1 2-bit state sequencer using D flip-flops based on your pre-lab design.

Implementation in Cadence Virtuoso:

1. Start *Virtuoso* and create a new schematic named **2bitState** in your project library.
2. Using your design from pre-lab, instantiate the necessary gates and DFFs to implement your design. Make all necessary wire connections.

   To make your schematic less cluttered, you can press the 'L' key to open the label window. This allows you to label a wire. Any wires assigned to the same label as another wire, or an input pin, will act as if the wires were connected. This will prevent you from having wires spanning the entire schematic and cluttering up the schematic.
3. After you have completed your schematic check and save the schematic and make sure there are no warnings or errors.

Demonstration:

4. Start the ADE-L simulator from the launch menu. For the 2bit sequence, use the stimulus file named *Lab4A.txt* that you can download from the ECE 331 website.
5. Set the run time for **5ms**.
6. Select the CLK, Q0, and Q1 as the outputs to be grapghed.
7. Run the simulation, review your results and answer the questions on the check off sheet.
8. Save/print your 2bitState schematic and simulation plots to include in your report.
9. Ask the TA to check your results and initial your check off sheet.

Part 2: State machine design challenge.

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

Design:

1. In the Lab 4 Check-off Sheet, sketch a schematic showing all the circuitry needed to implement the 3-bit sequencer from Figure 2. This circuit should be based on your pre-lab preparations and should use DFF's and any necessary basic logic gates (included the 3-input AND, as noted in the pre-lab). As with the 2-bit sequencer, the only external input to this circuit should be a CLK signal.

Implementation:

2. Create a new schematic named **3bitState**.
3. Instantiate the necessary gates and DFFs and add wires to implement your pre-lab design for the 3bit state sequencer.

   Remember that you can label wires in order to keep your schematic clear and make it easier to view.
4. Check and save the schematic and make sure there are no errors or warnings.

Demonstration:

5. Start the ADE L simulator from the Launch menu in the schematic window. Use the file *Lab4B.txt* as the stimulus file which you can download from the ECE331 website.

   To test that your state machine will correctly go to state 110 from any of the "non-sequence" states, the stimulus file imposes an initial state that is outside the sequence states.

6. Set the run time for **5ms**. Select CLK, Q0, Q1, and Q2 to be graphed.
7. Run the simulation, review your results and answer the questions on the check off sheet.
8. Save/print your 3bitState schematic and simulation plots to include in your report.
9. Ask the TA to check your results and sign your check off sheet.

Wrap Up

Clean up your lab bench before you exit the lab.

Remember, your Lab 4 report will be due in lab next week. Each student must submit his/her own lab report. Include your check-off sheet and waveform printouts with your report. You may want to review the Discussion Points below and talk to the TA about anything that is not clear to you.

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. How could the 2-bit sequencer be modified to implement an asynchronous reset, where an additional reset signal was used to force the state machine into a given initial state (e.g., 00)? Briefly describe the required reset circuitry.
4. Other than a state machine, briefly describe a useful digital circuit that requires flip flops (rather than standard logic gates). Do you think you have the experience need to implement such a circuit in *Virtuoso*? If not, what knowledge do you feel you lack?

Remember to include your results graphs 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 4 Background before you start the pre-lab. After completing the pre-lab, please review the entire Lab 4 assignment so you will know what to expect when you come to lab.

1. To ensure you are familiar with the basic steps of implementing a state machine, read the *State Machine Introduction* document.

2. Review the example D flip flop implementation of the Figure 1 state sequencer in the Background section. Below, record the logic expression that define each of the sequencer inputs (D_1, D_0) as a function of current-state outputs (Q_1, Q_0).

D_1 = _____, D_0 = _____

3. Based on your equations from #2, draw a schematic of the complete 2-state sequencer using two DFF cells and any necessary basic logic gates. Note, the only external input to this circuit should be a CLK signal. You can draw your schematic below or attach separately.

<pre-lab continues on next page>

In Part 2 of the lab, you will design and implement the 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 D flip-flops are needed for a 3-bit sequencer? _____

b. Complete the next-state excitation 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}	D_2	D_1	D_0

c. Complete the K-map for each flip flop input. Note each K-map is assigned to a specific combination of D and Q_2.

D_2 $Q_2 = 0$

Q_1 \ Q_0	**0**	**1**
0		
1		

D_1 $Q_2 = 0$

Q_1 \ Q_0	**0**	**1**
0		
1		

D_0 $Q_2 = 0$

Q_1 \ Q_0	**0**	**1**
0		
1		

D_2 $Q_2 = 1$

Q_1 \ Q_0	**0**	**1**
0		
1		

D_1 $Q_2 = 1$

Q_1 \ Q_0	**0**	**1**
0		
1		

D_0 $Q_2 = 1$

Q_1 \ Q_0	**0**	**1**
0		
1		

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

D_2=_____, D_1=_____, D_0=_____,

Note: A 3-input AND gate will be provided to simplify your design. Choose a logic function for each input that can be implemented with only the 3-input AND and the basic logic gates used in Labs 1-3. You will be asked to design the schematic to implement these input functions during Part 2 of the in-lab assignment.

TA pre-lab sign off *Initial*_____

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 7. Is the state value changing with the clock? _____

Record the observed state sequence (e.g., 0 0 → 1 1 → …)

_____

Does it agree with the correct state sequence? _____

Part 1: TA sign off Initial_____

Part 2: ***State machine design challenge***

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

Is the state value changing with the clock? _____

Step 7. What state follows the initial non-sequenced state? _____

Is this correct? _____

Step 7. Record the observed state sequence (e.g., 101 → 110 → …)

_____

Does it agree with the desired sequence? _____

Part 2: TA sign off Initial_____