EE2174 Lab 11

Finite State Machines

Introduction

In this lab, you will examine the role of blocking vs. non-blocking statements in Verilog HDL and create finite state machines using the Quartus State Machine wizard. Finite state machines are widely used in designing compilers, protocols and aid in the study of computation.

1 Blocking vs. Non-Blocking Statements

In this part of the lab, you will examine the difference between blocking and non-blocking statements. Open a new project and create a module based on Listing 1. Compile the code and open the "**RTL View**". Save a screenshot of this viewer to submit with the lab.

```
11  module Lab10Part1(
12  input clk,
13  input [7:0] A,
14  output reg [7:0] out1,
15  output reg [7:0] out2);
16
17      // Non-blocking approach
18      always@(posedge clk)
19      begin
20
21          out1 <= A;
22          out2 <= out1;
23
24      end
25
26  endmodule
```

Listing 1: Non-blocking module

Next, change the statements in the always statement to be blocking. Recompile and reopen the "**RTL View**". Save a screenshot of this viewer to submit with the lab.

Signoff

Show your TA both RTL Views. **Save a screenshot of both viewers and submit them with this lab.**

2 Finite State Machine

In this part of the lab, you will create a finite state machine and physically implement it on the DE10-Lite.

2.1 Quartus State Machine Wizard

Create a new Quartus project and add a new empty Verilog HDL file to it named "**Lab11FSM.v**". This file will be used later to house the Verilog generated by the State Machine Wizard.

Next, add a new "**State Machine File**". Use Figure 1 to navigate the State Machine Menu Strip.

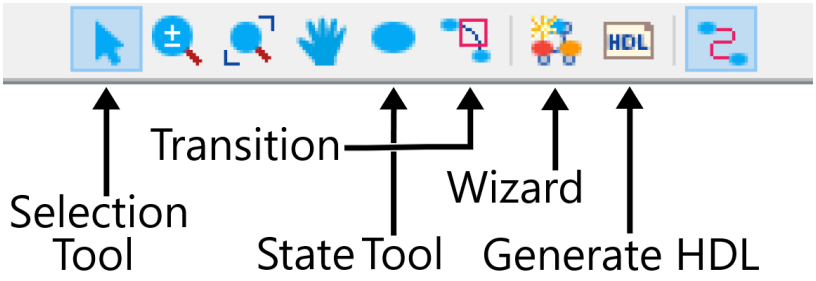

Figure 1: State Machine Menu Strip

Select the "State Tool" and add the necessary states to build the state machine in Figure 2. Select the "Selection Tool" to select each state; change the name of the states to "A-E". Next, use the "Transition Tool" to add transitions between each state.

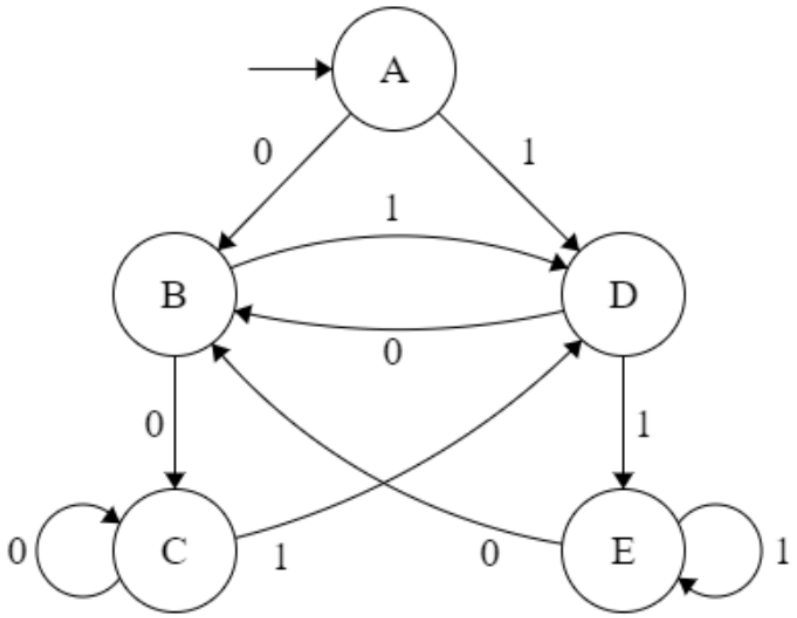

Figure 2: State Machine to implement in Part 2.1.

Once the state machine in Figure 2 has been constructed, select the State Machine Wizard from the menu strip. Select “Edit an existing state machine design” and make the following updates:

- Under the “Inputs” tab, add one input called “**transition**”.
- Under the “Outputs” tab, add one 4-bit output; this value will be used to drive the hex display decoder. This output name must match what is used in the “Actions” tab for the output port.
- Under the “Transitions” tab, add the logic required to transition between states. See Figure 3 for a few transitions filled out. Complete the remaining transitions.
- Under the “Actions” tab, add the logic for the output values. Two example states have been given in Figure 4; follow the mapping to have state A have an output of 1, state B an output of 2, and so on.
- Select “Apply” and exit the wizard.

General | Inputs | Outputs | States | Transitions | Actions

Source State	Destination State	Transition (In Verilog or VHDL 'OTHERS')
A	D	transition==1
D	E	transition==1
E	E	transition==1
C	C	transition==0

Figure 3: Transition table.

General | Inputs | Outputs | States | Transitions | Actions

Output Port	Output Value	In State	Additional Conditions
value[3:0]	1	A	
value[3:0]	2	B	

Figure 4: Action tab filled in mapping states with output values.

Now, select the “Generate HDL File” button and generate in Verilog HDL. Copy the contents of the generated file to the empty Verilog HDL file you initially created.

Instantiate this generated module from a main module using SW[0] as the transition bit, KEY[0] as the clock, and KEY[1] as reset. Remember to invert the keys in the instantiation as they are active low. Display the state machine's current state on HEX0 by adding 0x9 to the state machine value before sending it to a seven segment driver.

Now, import the pin assignments and compile the program. Examine the “**RTL Viewer**” and “**State Machine Viewer**”. You can access the “**State Machine Viewer**” by selecting “**Tools**” -> “**Netlist Viewers**” -> “**State Machine Viewer**”. Save a screenshot of both viewers, then program the DE10-Lite.

Signoff

Show your TA the RTL view of the state machine and the State Machine viewer. **Save a screenshot of both viewers and submit them with this lab.** Next, demonstrate to your TA that KEY[0] changes the state of the machine based on the input of SW[0]. You can reset the state machine by holding down the reset key and pulsing the clock key.

2.2 Modifying the State Machine in Verilog

Create a new Verilog file named "**Lab11FSM2.v**". Copy the generated state machine Verilog HDL code from Part 2.1 into this file. Next, modify this code to add state F as shown in Figure 5. Note that the 1 transition out of B no longer goes to D, but goes to F. Do not use the wizard to change the state machine.

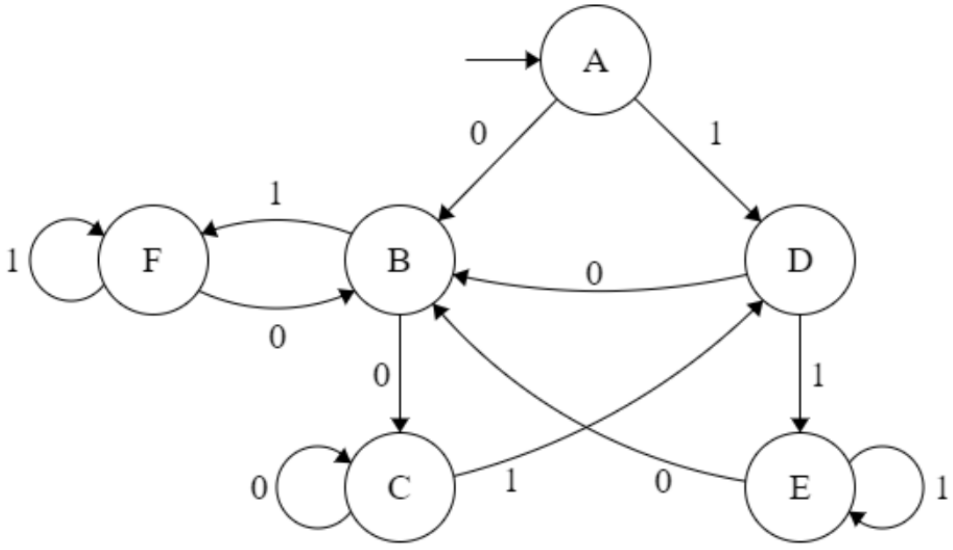

Figure 5: Updated state machine to implement in Part 2.2.

Recommended method for completion:

- Add a new state parameter.
- Duplicate a state.
- Set the next states and the output value for this state.
- Update the next state from B.

Recompile and open the "**RTL Viewer**" and "**State Machine Viewer**". Program the test the DE10-Lite.

Signoff

Show your TA the RTL view of the state machine and the State Machine viewer. **Save a screenshot of both viewers and submit them with this lab.** Next, demonstrate to your TA that KEY[0] changes the state of the machine based on the input of SW[0]; have your state machine enter state F.

3 Returning the DE10-Lite

In this part of the lab, you will return your department-issued DE10-Lite. Your board must be returned by the end of week 14; if you do not return the DE10-Lite, your lab grade will be filed as X (conditional) until it is returned to the department. See the Lab Policies for more details.

To return the DE10-Lite, bring the board and USB cable in its box to your TA. The TA will upload a test program to the board to ensure all peripherals are still working correctly. If you are unable to return your DE10-Lite, or it fails the peripheral test, the ECE Lab Supervisor will contact you to resolve the problem.

If you did not bring your DE10-Lite with you to this lab, visit office hours during the week to return the board. If you are unable to come to office hours, contact your TA to set up an additional time to drop off your board.

4 Submission

List of screenshots that must be submitted with the submission sheet in one combined pdf, and list of code files that must be submitted. All code required to run the project should be submitted.

Single PDF

- Submission Sheet
- Part 1 RTL Viewer (Non-Blocking)
- Part 1 RTL Viewer (Blocking)
- Part 2.1 State Machine View
- Part 2.1 RTL Viewer
- Part 2.2 State Machine View
- Part 2.2 RTL Viewer

Code

- Lab11Part1.v (Blocking vs. Non-Blocking)
- Lab11FSM.v (Generated Finite State Machine Verilog HDL)
- Lab11Part2.v (Finite State Machine Main Module)
- Lab11FSM2.v (Updated Finite State Machine)
- hex7seg.v (Seven Segment Display Controller)

Reminder: Your department-issued DE10-Lite is due at the end of the semester. Failure to return the board will result in your lab grade being filed as X (conditional).