

ECE437 Lab 4 — Adventure Game 2 (System Verilog)

TA Yibiao Ying • Fall 2025

1. Lab 3 Review

- Revisit Lab 3 map and the dual FSM concept (Room & Sword).

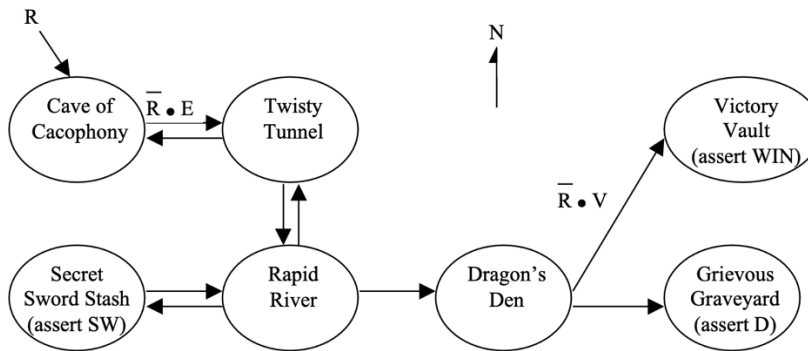


Figure 1. Partially Completed State Transition Diagram for Room FSM

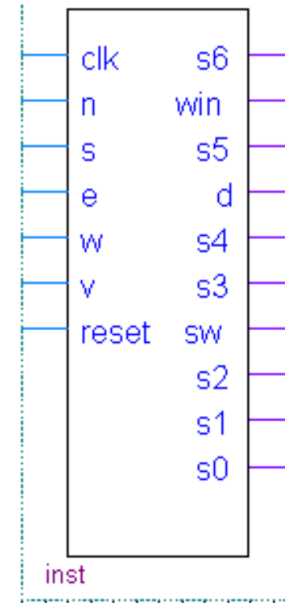


Figure 3. Symbol for Room FSM, showing its inputs and outputs

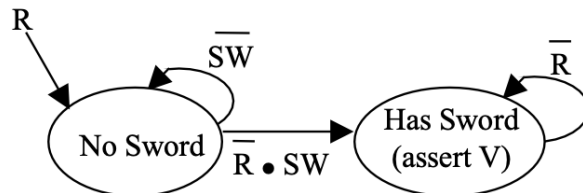


Figure 2. State Transition Diagram for Sword FSM

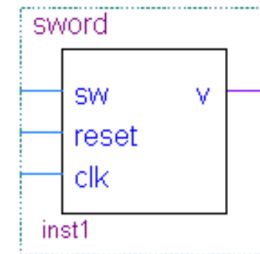


Figure 4. Symbol for Sword FSM, showing its inputs and outputs

2. Lab 4 Changes

- We are using **System Verilog** instead of block diagram this time
- Some minor changes to game design requirements

Changes to the game design requirements

1. The Dragon's Den is now in the **SOUTH-EAST direction** from the Rapid River in Lab 4 instead of the East direction as in Lab 3. This means that both South and East direction need to be asserted simultaneously to make the hero go from the Rapid River to the Dragon's Den.
2. The room states in the Room FSM need to persist, until a legal direction input (or reset) is asserted. Here legal direction means a direction that can directly lead to another room from the current room. For example, the only legal direction from Cave of Cacophony is East, while for Twisty Tunnel both West and South are legal directions.
3. Note that any illegal input should be ignored and should NOT cause any state transition.
4. Note that for the Dragon's Den you do not need to make the persistence assumption because by design the hero will automatically end up in either the Victory Vault or the Grievous Graveyard, depending on whether they have the Vorpall Sword.
5. Just like the Room FSM there is a persistence requirement for the Sword FSM. Once the Vorpall Sword output is asserted, it needs to persist until reset is asserted.

3. Lab 4 Approaches

- Modify your Boolean equations derived in lab3 to take care of the lab4 changes
- Implement a System Verilog hierarchy:
 - D-Flip Flop module
 - Room FSM module
 - Sword FSM module
 - Game module

1. D-Flip Flop module:

```
module d_ff (input logic d, clk,  
             output logic q);  
  
    always @ (posedge clk) begin  
        //your own module definition  
    End  
  
endmodule
```

2. Room FSM module

```
module room (input logic clk, n, s, e, w, v, reset,  
            output logic s6, win, s5, d, s4, s3, sw, s2, s1, s0);  
  
    //one instance of D-Flip Flop for each room state  
  
endmodule
```

3. Sword FSM module

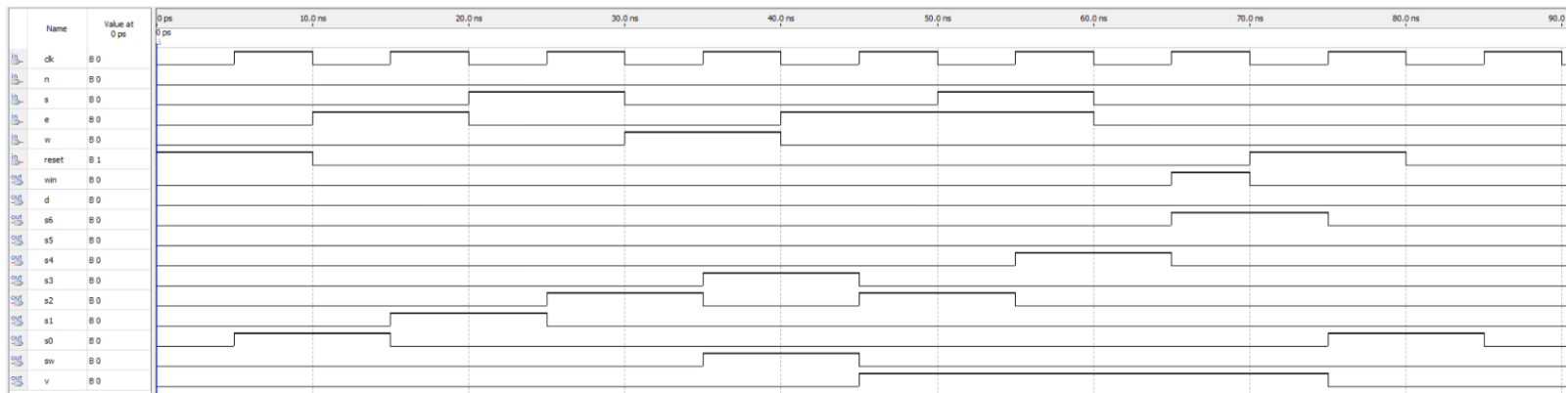
```
module sword (input logic sw, reset, clk,  
              output logic v);  
  
    //one instance of D-flip flop to store sword FSM state  
  
endmodule
```

4. Whole game module

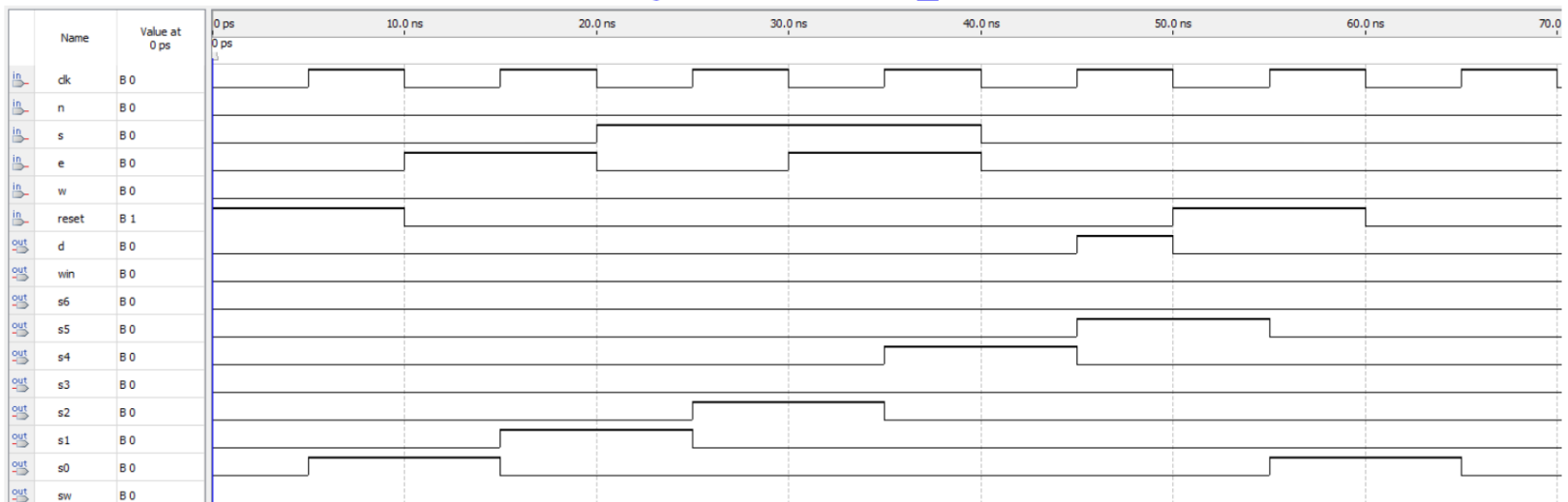
```
module game(input logic clk, n, s, e, w, reset,  
            output logic s6, win, s5, d, s4, s3, sw, s2, s1, s0, v);  
  
    //one instance of Room FSM module  
    //one instance of Sword FSM module  
  
endmodule
```

4. Lab 4 Requirements

- Simulation Requirements



win



lose

4. Lab 4 Requirements

- FPGA Demo (Terasic Board)

5. Lab Assessment 1 Reminder

in the week of 10/13/2025 and 10/20/2025

- **Due date-** Lab 4 in-lab report is due in the week of 10/06/2025, after completing the in-lab session for Lab 4.
- **Due date-** Lab 4 assessments will be performed over two weeks (TBA). Assessment slots will be assigned through a google sign-up sheet on a first-come-first-served basis.
- **Points assigned to mandatory assessment-** 20 points for Lab 4 assignment come from the mandatory assessment.

- **Contents of the mandatory assessment-**

Exact questions for the assessment will be revealed to you at your assigned slot. To give you an idea, here is an outline of what you might be asked. You will be required to make certain changes to your design in a rapid-fire manner. A few examples of such “changes”-

1. Changes in the definition of certain modules that you created.
2. Changes in the pin assignments so that input/output is assigned to a different hardware on the board.
3. Changes in the output requirements.
4. Or something else along these lines.

The idea is to ask the student to make a few random changes to their game design and see how they respond.