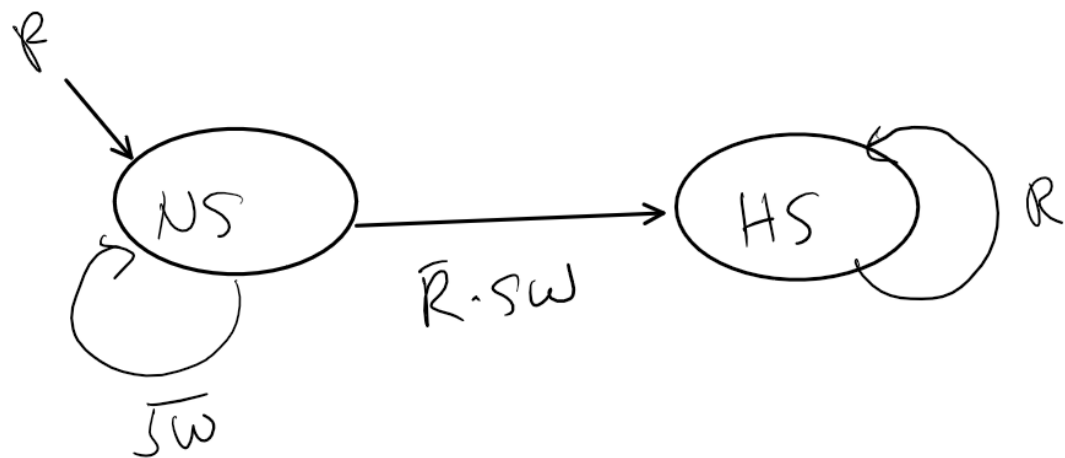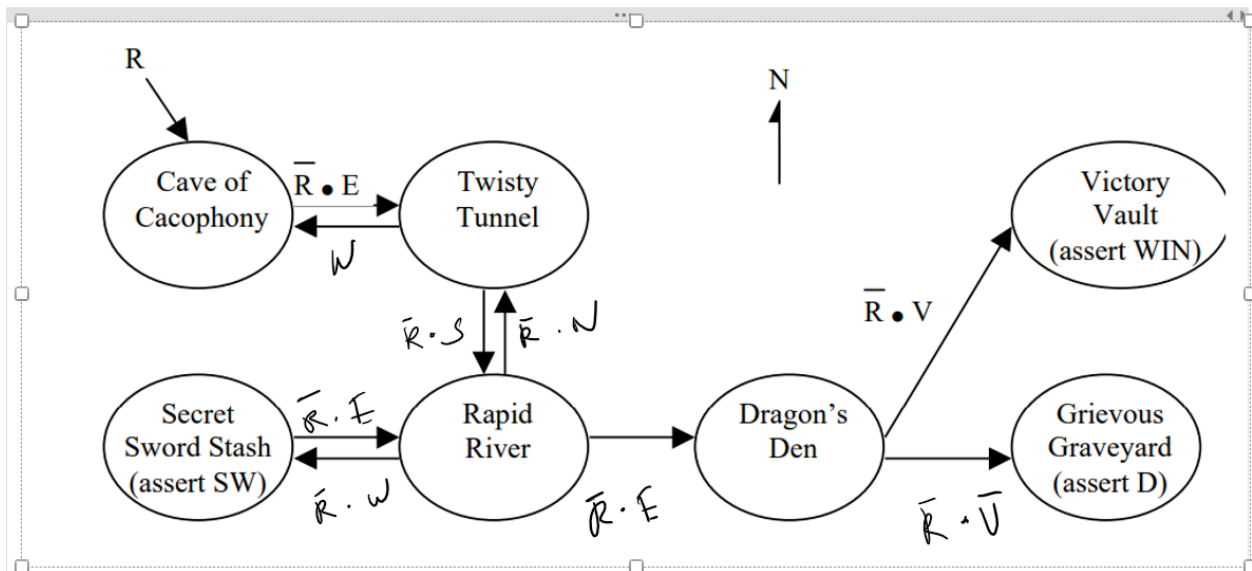# Paul Kim

# Lab 03 Final Report

# Friday section

I have spent about total 2 hours reading the "DSD lab03.pdf" lab instruction. I've tried to understand and grasp the idea fully.

Then I've first tried to write my own table, write my equations, and draw the schematic drawings. Then I've compared my answers to hint pdf. It really helped me a lot. Otherwise I might have much more hours to debug my design. Also whenever I was stuck on some problems on software or the lab itself during Thursday 2-4pm, I went to Mr. Feng's office hours 2-4pm Thursday while I was working downstairs on the computer lab. All these procedure took me about 5 hours.

Waveform simulation took me about 2 hours because my Quartus 'prime lite' on my pc was giving me node error. I've tried to figure out in my pc for awhile by remaking the project etc etc. But eventually I've gave up and decided to use the computer lab computer and rebuilt the project and I eventually got the result.

So This lab took me 2+5+2=9 hours total

R

N

Cave of
Cacophony

$\overline{R} \cdot E$

Twisty
Tunnel

W

Victory
Vault
(assert WIN)

$\overline{R} \cdot V$

$\overline{R} \cdot S$   $\overline{R} \cdot N$

Secret
Sword Stash
(assert SW)

$\overline{R} \cdot E$

Rapid
River

$\overline{R} \cdot W$

Dragon's
Den

$\overline{R} \cdot E$

Grievous
Graveyard
(assert D)

$\overline{R} \cdot \overline{V}$

R

NS

$\overline{SW}$

$\overline{R} \cdot SW$

HS

R

Room FSM

state transition table

| inputs | | | | | | | output |
| state | N | S | W | E | R | V | state |
|---|---|---|---|---|---|---|---|
| X | x | x | x | x | 1 | x | CC |
| CC | x | x | x | 1 | 0 | x | TT |
| TT | x | x | 1 | x | 0 | x | CC |
| TT | x | 1 | x | x | 0 | x | RR |
| RR | x | x | 1 | x | 0 | x | SS |
| RR | x | x | x | 1 | 0 | x | DD |
| RR | 1 | x | x | x | 0 | x | TT |
| SS | x | x | x | 1 | 0 | x | RR |
| DD | x | x | x | x | 0 | 1 | VV |
| DD | x | x | x | x | 0 | 0 | GG |

output table

| current state | sw | d | win |
|---|---|---|---|
| CC | 0 | 0 | 0 |
| TT | 0 | 0 | 0 |
| RR | 0 | 0 | 0 |
| SS | 1 | 0 | 0 |
| DD | 0 | 0 | 0 |
| VV | 0 | 0 | 1 |
| GG | 0 | 1 | 0 |

# Sword FSM

## state transition table

| input | | | output |
| current state | sw | R | next state |
|---|---|---|---|
| x | x | 1 | NS |
| NS | 0 | 0 | NS |
| x | 1 | 0 | HS |
| HS | x | 0 | HS |

## output table

| input | output |
| current state | V |
|---|---|
| NS | 0 |
| HS | 1 |

**State encodings**

| | Room | | |
|---|---|---|---|
| | | CC | 0000001 |
| | | TT | 0000010 |
| | | RR | 0000100 |
| | | SS | 0001000 |
| | | DD | 0010000 |
| | | GG | 0100000 |
| | | VV | 1000000 |
| | | | |
| | Sword | | |
| | | NS | 0 |
| | | HS | 1 |

**Room FSM**

binary state transition table

| inputs | | | | | | | output |
|---|---|---|---|---|---|---|---|
| state | N | S | W | E | R | V | state |
| X | x | x | x | x | 1 | x | 0000001 |
| 0000001 | x | x | x | 1 | 0 | x | 0000010 |
| 0000010 | x | x | 1 | x | 0 | x | 0000001 |
| 0000010 | x | 1 | x | x | 0 | x | 0000100 |
| 0000100 | x | x | 1 | x | 0 | x | 0001000 |
| 0000100 | x | x | x | 1 | 0 | x | 0010000 |
| 0000100 | 1 | x | x | x | 0 | x | 0000010 |
| 0001000 | x | x | x | 1 | 0 | x | 0000100 |
| 0010000 | x | x | x | x | 0 | 1 | 1000000 |
| 0010000 | x | x | x | x | 0 | 0 | 0100000 |

output table

| current state | sw | d | win |
|---|---|---|---|
| 0000001 | 0 | 0 | 0 |
| 0000010 | 0 | 0 | 0 |
| 0000100 | 0 | 0 | 0 |
| 0001000 | 1 | 0 | 0 |
| 0010000 | 0 | 0 | 0 |
| 0100000 | 0 | 0 | 1 |
| 1000000 | 0 | 1 | 0 |

**Sword FSM**

binary state transition table

| input | | | output |
|---|---|---|---|
| current state | sw | R | next state |
| x | x | 1 | 0 |
| 0 | 0 | 0 | 0 |
| x | 1 | 0 | 1 |
| 1 | x | 0 | 1 |

output table

| input | output | |
|---|---|---|
| current state | V | |
| 0 | 0 | |
| 1 | 1 | |

Sword.bdf
$$V' = (sw * \bar{R})+(V*\bar{R})$$
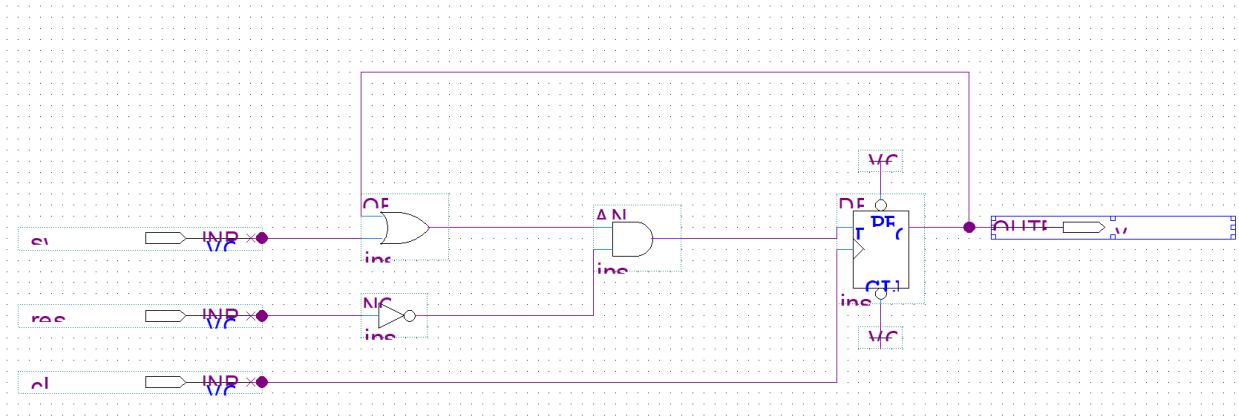
Room.bdf

Room FSM boolean equation

$$S_1' = R + S_2 W \overline{R}$$

$$S_2' = S_1 E \overline{R} + S_3 N \overline{R}$$

$$S_3' = S_2 S \overline{R} + S_4 E \overline{R}$$

$$S_4' = S_3 W \overline{R}$$

$$S_5' = S_3 E \overline{R}$$

$$S_6' = S_5 \overline{R} \overline{V}$$
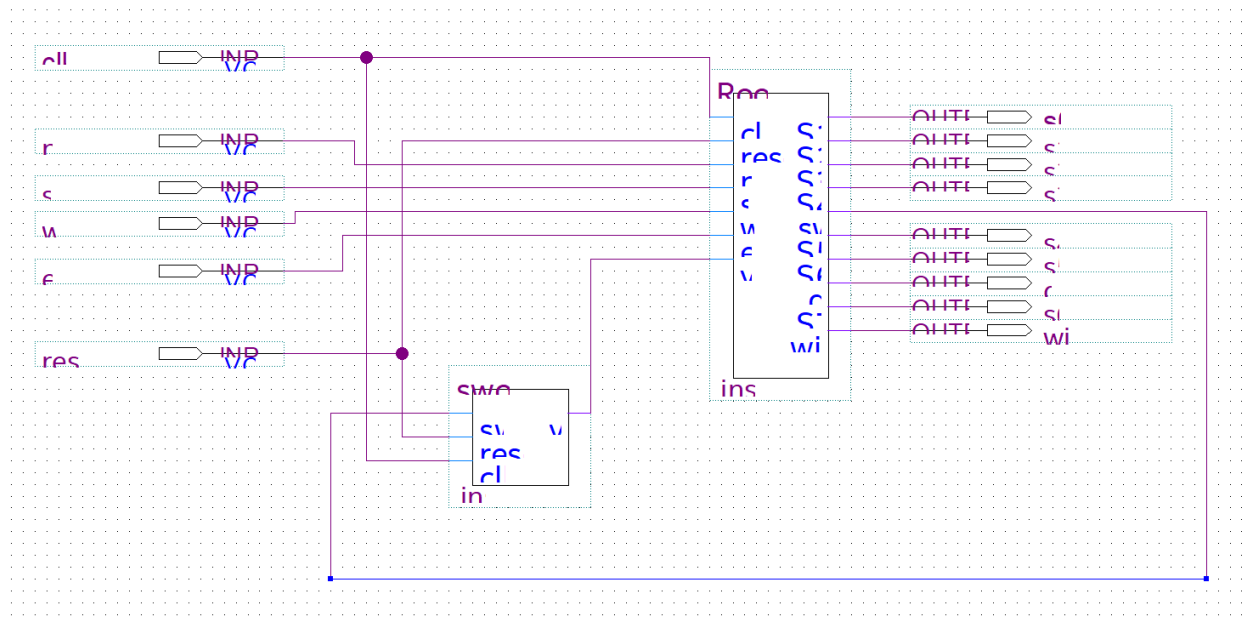
$$S_7' = S_5 \overline{R} V$$

$$Sw = S_4$$

$$D = S_6$$

$$Wm = S_7$$

Lab03_PK.bdf

Losing game

| | Name | Value at 0 ps | |
|---|---|---|---|
| in | clk | B 0 | |
| in | n | B 0 | |
| in | s | B 0 | |
| in | e | B 0 | |
| in | w | B 0 | |
| in | reset | B 1 | |
| out | win | B 0 | |
| out | d | B 0 | |
| out | s6 | B 0 | |
| out | s5 | B 0 | |
| out | s4 | B 0 | |
| out | s3 | B 0 | |
| out | s2 | B 0 | |
| out | s1 | B 0 | |
| out | s0 | B 0 | |
| out | sw | B 0 | |
| out | v | B 0 | |

winning game
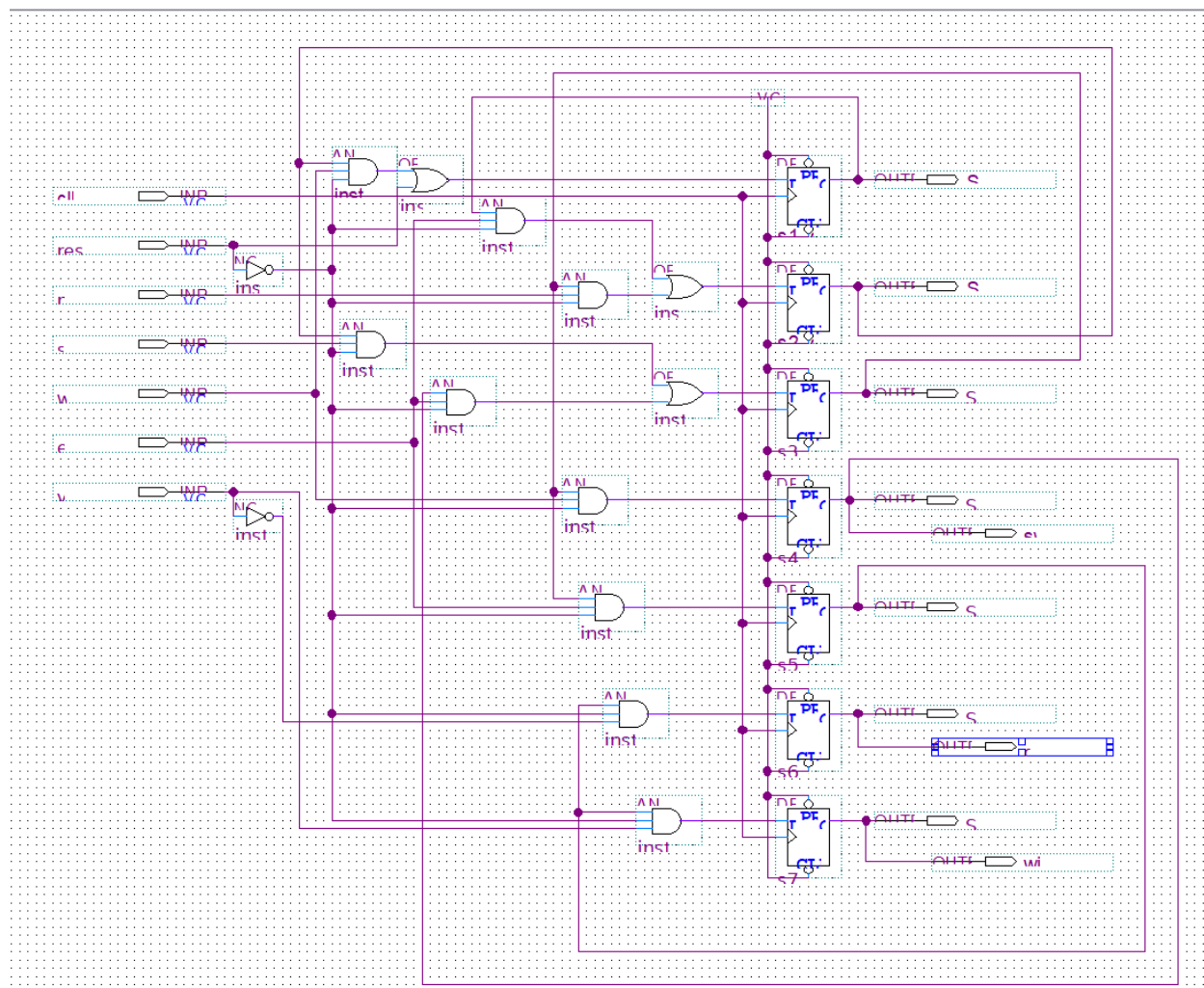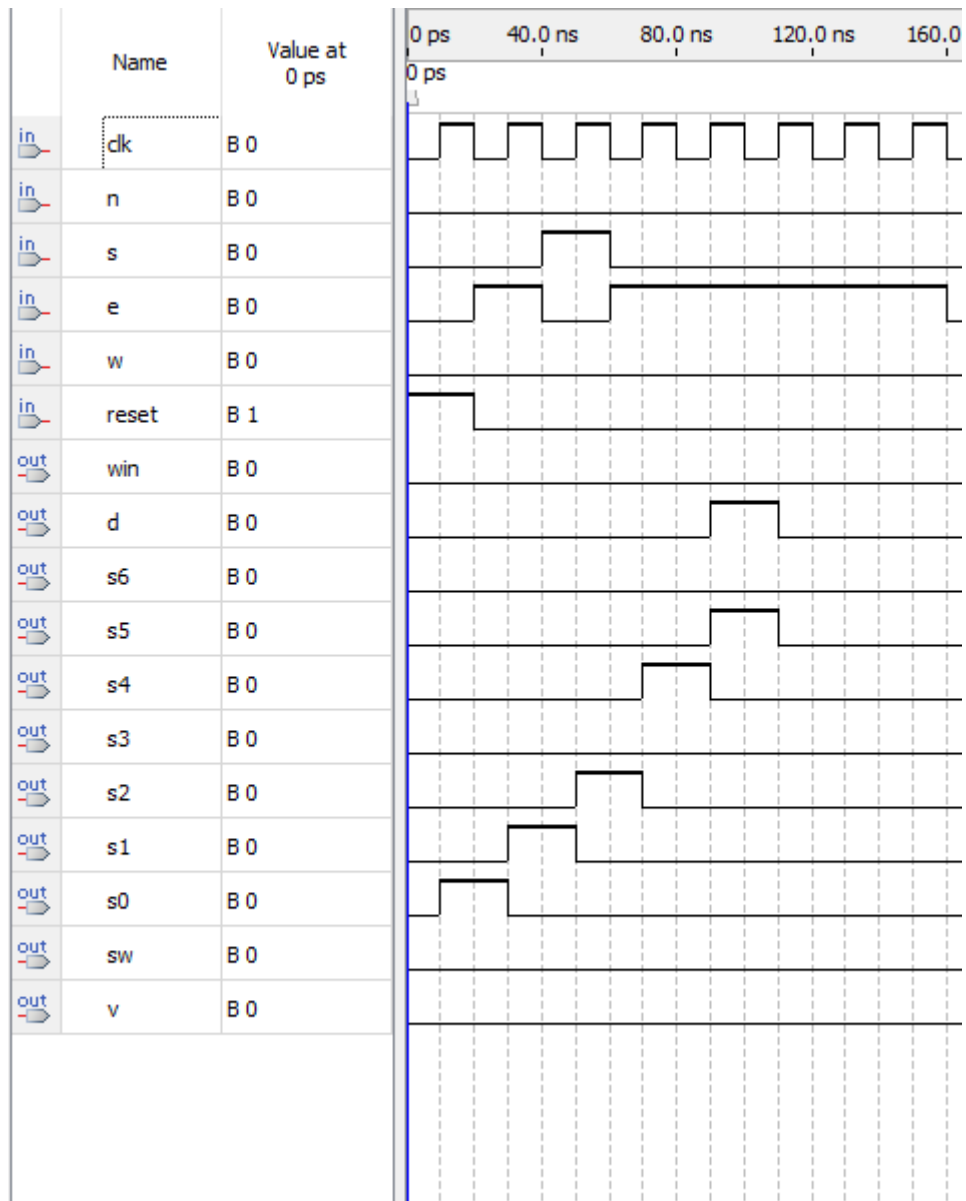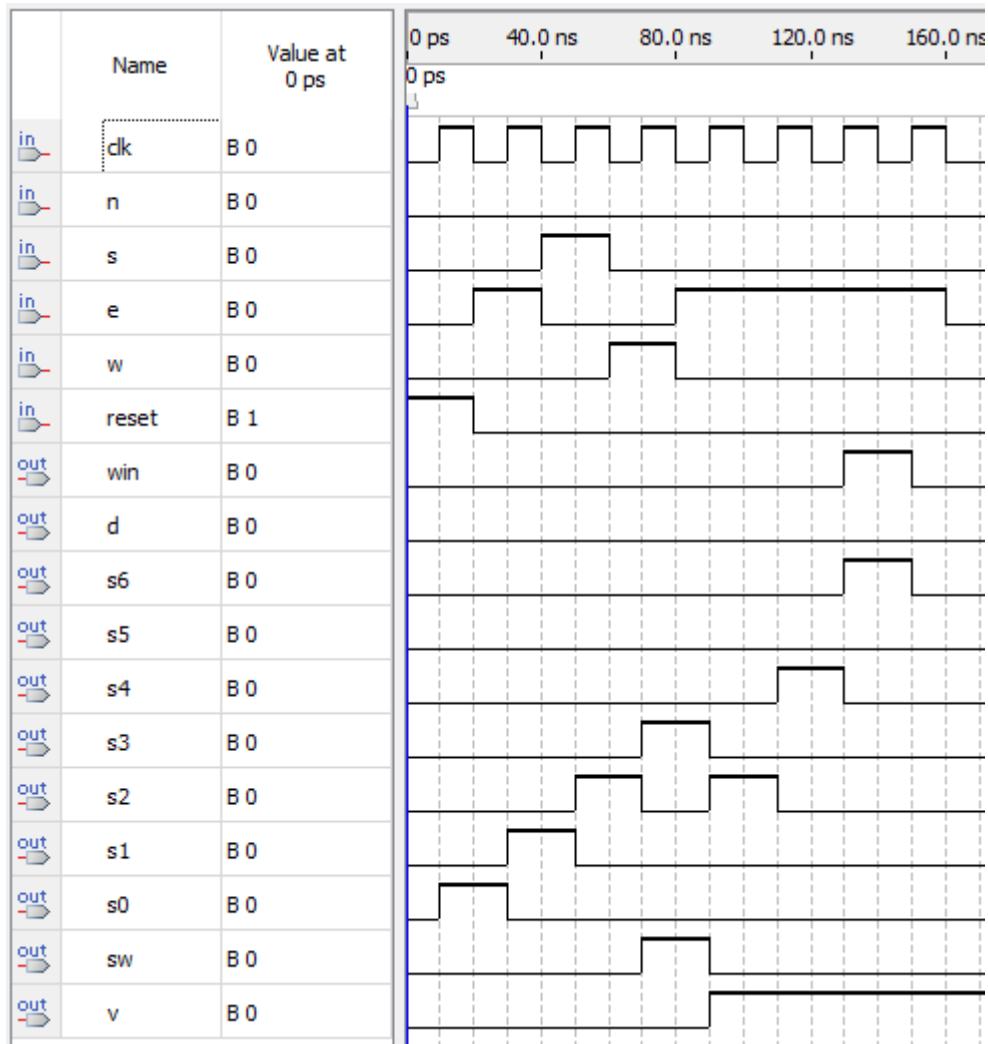
| Name | Value at 0 ps | | 0 ps | 40.0 ns | 80.0 ns | 120.0 ns | 160.0 ns |
|------|---------------|---|------|---------|---------|----------|----------|

| | tatu | From | To | nment l | Value | Enabled | Entity |
|---|---|---|---|---|---|---|---|
| 1 | ✔ ... | | in s | Locati... | PIN_AB27 | Yes | |
| 2 | ✔ ... | | in reset | Locati... | PIN_AB28 | Yes | |
| 3 | ✔ ... | | in e | Locati... | PIN_AC27 | Yes | |
| 4 | ✔ ... | | in n | Locati... | PIN_AC28 | Yes | |
| 5 | ✔ ... | | in w | Locati... | PIN_AD27 | Yes | |
| 6 | ✔ ... | | out s5 | Locati... | PIN_E18 | Yes | |
| 7 | ✔ ... | | out s2 | Locati... | PIN_E19 | Yes | |
| 8 | ✔ ... | | out s4 | Locati... | PIN_F18 | Yes | |
| 9 | ✔ ... | | out s1 | Locati... | PIN_F19 | Yes | |
| 10 | ✔ ... | | out s3 | Locati... | PIN_F21 | Yes | |
| 11 | ✔ ... | | out s0 | Locati... | PIN_G19 | Yes | |
| 12 | ✔ ... | | out win | Locati... | PIN_H19 | Yes | |
| 13 | ✔ ... | | out d | Locati... | PIN_J17 | Yes | |
| 14 | ✔ ... | | out s6 | Locati... | PIN_J19 | Yes | |
| 15 | ✔ ... | | in clk | Locati... | PIN_M23 | Yes | |
| 16 | | <<...>> | <<...>> | <<ne... | | | |

My LED signals and switches assignment values
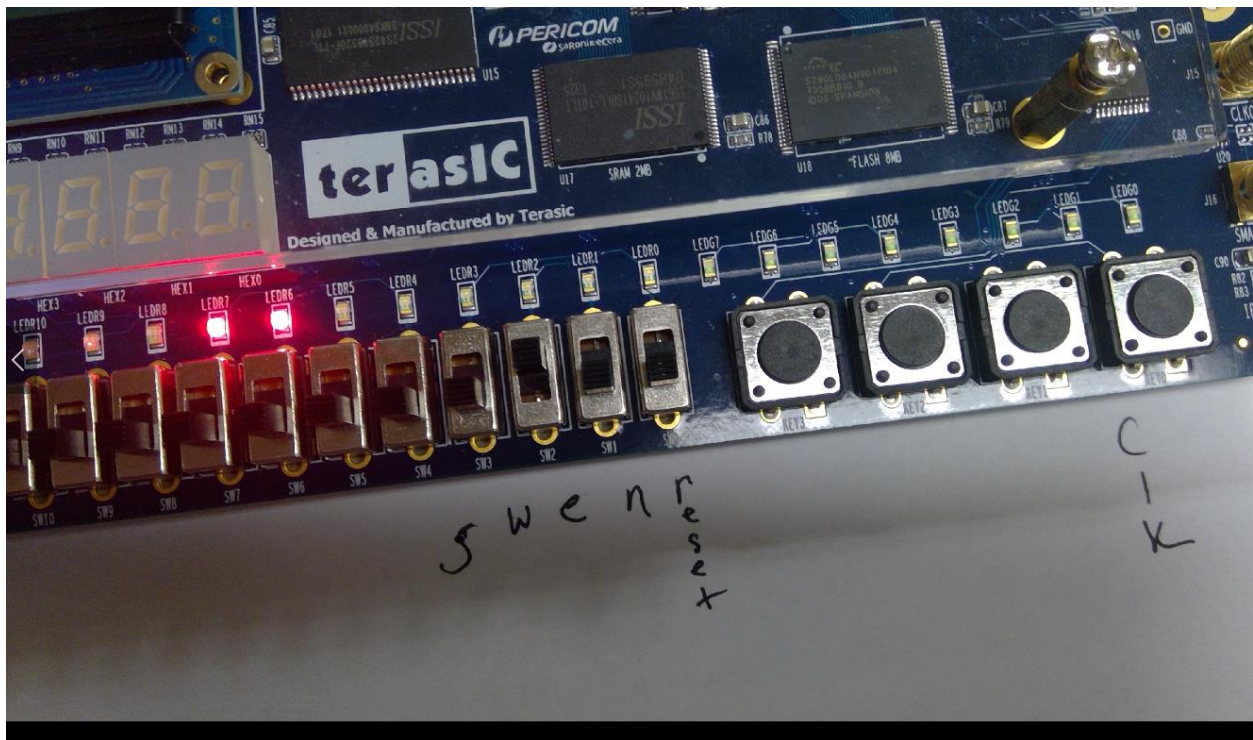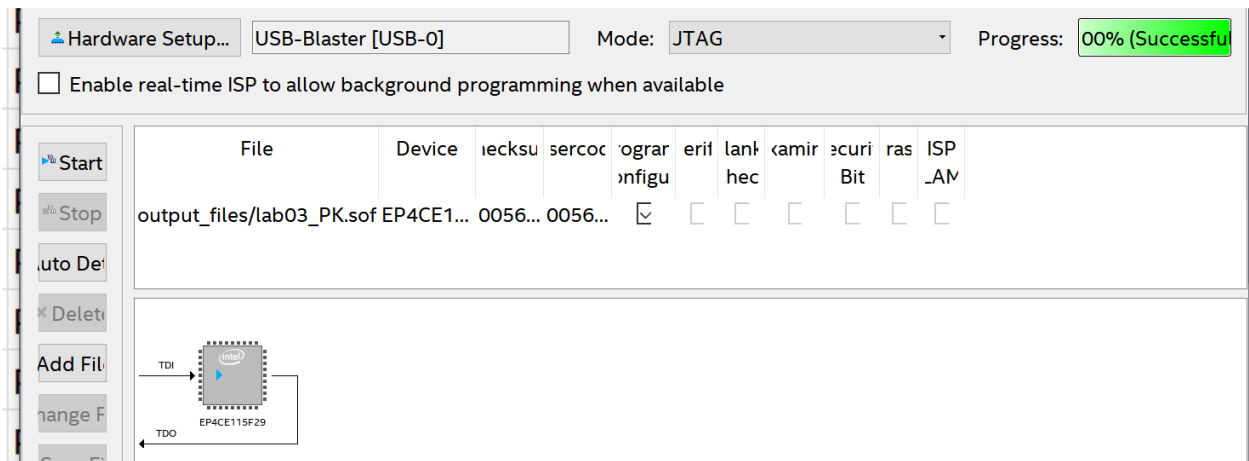
In this lab, we've created a simple game on the board. A player has seven rooms that he can go through and the player can only wins if he goes to specific room with an object that can be only acquired by going through another specific room.

First, we had to design by making room FSM and another separate FSM for sword. Then we had to make state transition table for both room and sword. And turn those tables into binary form. From there, output boolen equations were able to acquired. Then using those equation, we implemented schematic drawing and made hierarchical design by creating each individual Room and Sword FSM's. And the top entity lab3_PK.bdf combined and connected all the inputs(directions, reset, clk), room and sword.