

Homework 3: Light the Candles

October 13, 2017

Sabbir Ahmed

1 Background

For this assignment, a game was created to be played on the development board. The goal of the game is for the player to light all the candles, represented by the 8 LEDs on the board just above the switches, while they are being pseudo-randomly turned off.

The game starts off with the player pressing the RESET button (a.k.a. BTN_SOUTH) just below the rotatory knob.

The player is placed initially at a position 0.

The player may light the candles by jumping on them. This is done using the BTN_NORTH button on the board just above the rotary knob. A player may jump at the current position by setting the switches to zero and pressing BTN_NORTH. Any desired horizontal movement in the jump to another position is provided by the four switches which indicate a value from 7 to -8, in binary two's complement.

Once a candle is lit, the player may jump again jump on the same location (using 0) or to another location using the delta provided by the switches.

However, the candles that were lit will continually go out at random and must be relit. To be specific, a candle may at random go out at the same time the player jumps. Secondly, note that there is no direct indicator to where the player is; so it is up to the player to keep track.

2 Implementation

Three discrete modules were used to implement the game: `igniter`, `extinguisher` and `candle_controller`. These submodules were connected and debounced using a top level module that may be visualized with the schematic diagram configured as a block diagram in Figure 1.

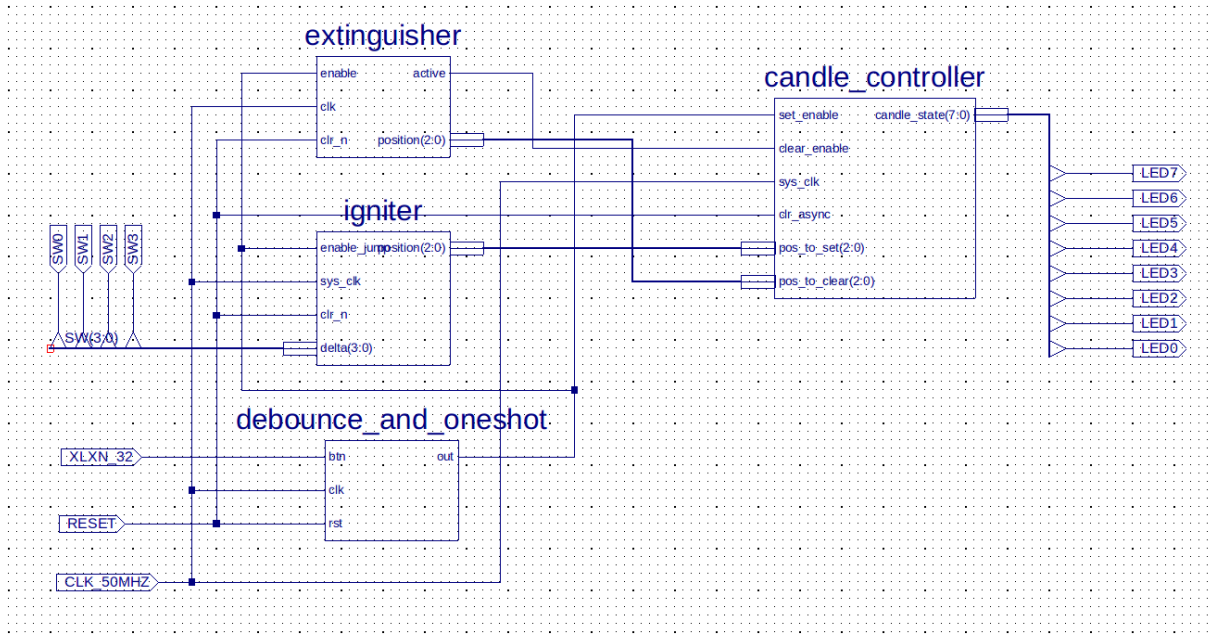


Figure 1: Schematic of the Implementation of the Game

2.1 igniter

The **igniter** submodule allowed the player to interface with the LEDs. It accepted the 4 switches as the input, representing them as the 4 bit `delta` of the current position of the user. The submodule also handled `BTN_NORTH`, acting as the enable.

A testbench has been included with the project to demonstrate usage of the submodule and its effects on the position of the player in `igniter_tb`. A sample waveform output has been provided:

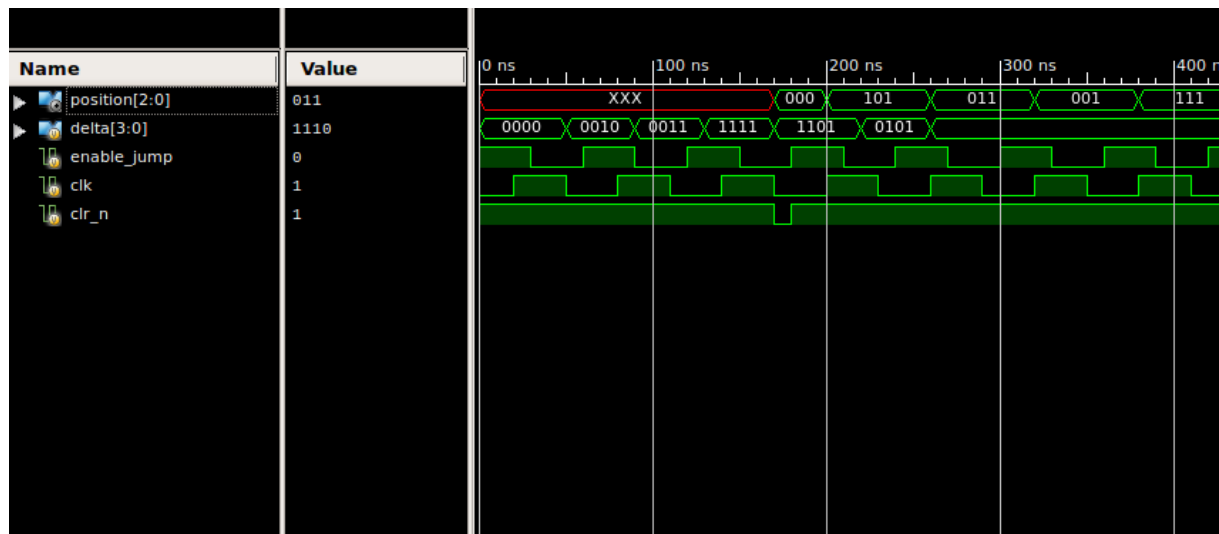


Figure 2: Waveform of the Testbench of igniter

2.2 extinguisher

The `extinguisher` submodule turns the LEDs off using a pseudo-random index generator. It consists of an internal counter, running independently of the rest of the submodule on each clock cycle. The 4 bit counter iterates through all its 16 states, and outputs an active signal when the module is enabled. The active signal outputs high when the counter is less than 8 and assigns the value to the index of the LED to be turned off.

The 50 MHz clock speed from the development board cycles through the counter very rapidly:

$$50 \text{ MHz} = \frac{1}{50 \times 10^6} \text{ s} = 20 \text{ ns} \times 16 \text{ states} = \frac{1 \text{ counter loop}}{320 \text{ ns}}$$

This speed allows the player to perceive the outputs of the submodule to be random.

A testbench has been included with the project to demonstrate usage of the submodule and its effects on the position of the player in `extinguisher_tb`. A sample waveform output has been provided:

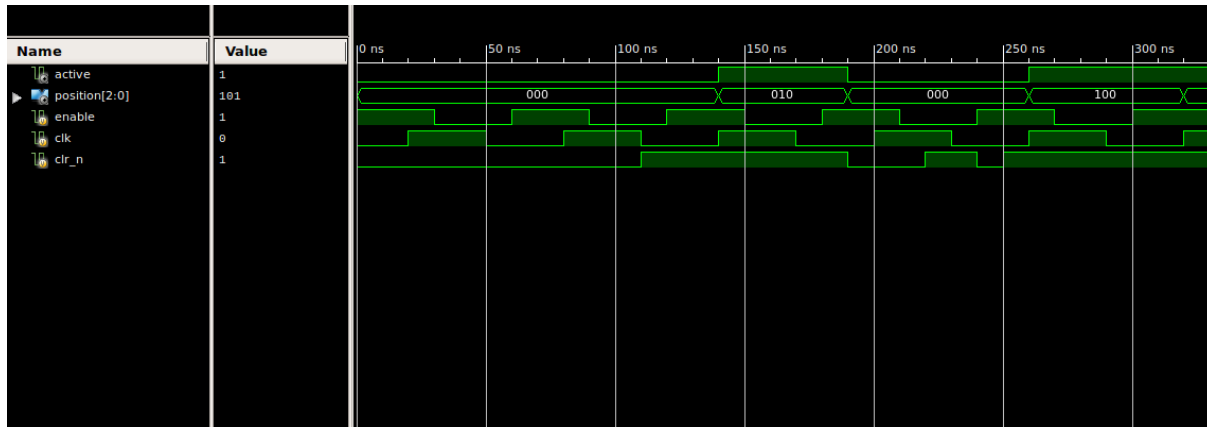


Figure 3: Waveform of the Testbench of extinguisher

2.3 candle_controller

The `candle_controller` submodule directly handles the states of the LED outputs. It utilizes the outputs from the other two submodules to determine the states of the individual LEDs.

A testbench has been included with the project to demonstrate usage of the submodule and its effects on the position of the player in `candle_controller_tb`. A sample waveform output has been provided:

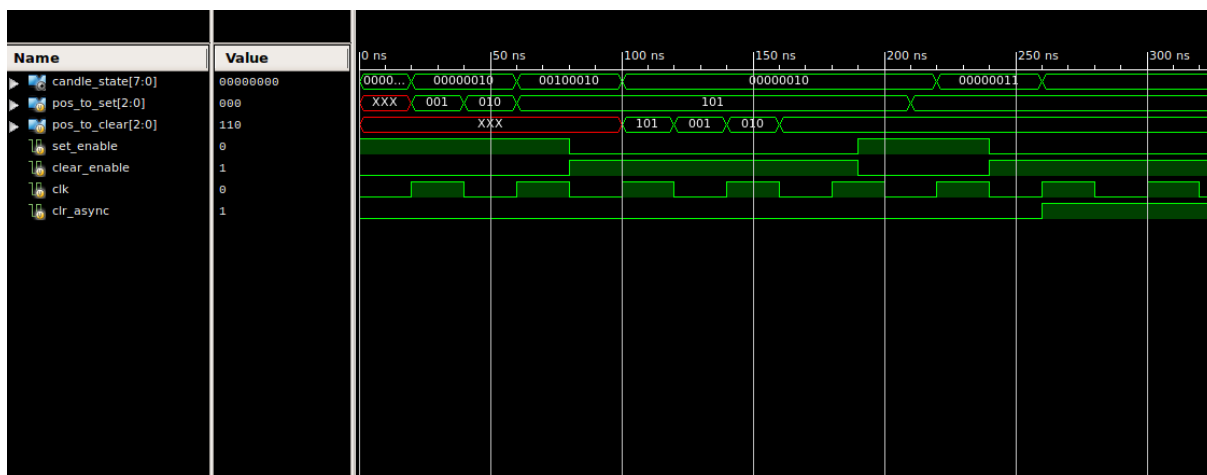


Figure 4: Waveform of the Testbench of candle_controller

2.4 Top Level Module

The top level module, `top.v`, connects all the submodules with wires. A testbench has also been included with the project to demonstrate the implementation of the game in `top_tb`. A sample waveform output has been provided:

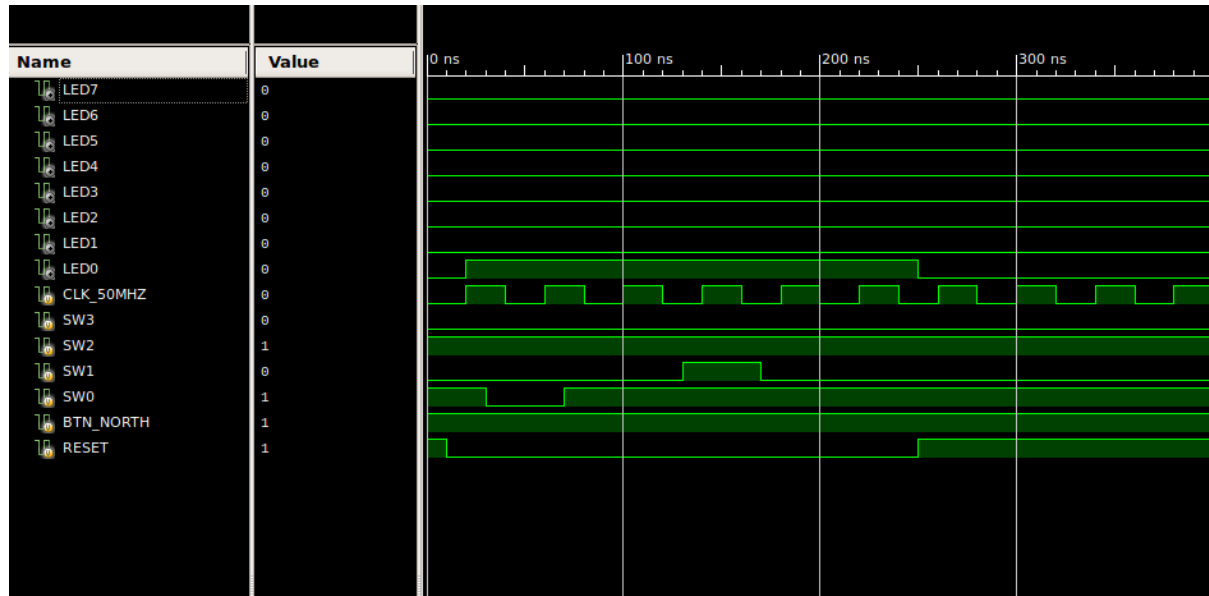


Figure 5: Waveform of the Testbench of top

3 Troubleshooting

Several software and hardware issues were emphasized during the timeline of this assignment. Software glitches include the Xilinx ISE not being able to interface the development board at random times after successful synthesis. Hardware issues included the board not being able to properly process the switches as inputs in the initial stages of the assignment timeline.

4 Results

All the testbenches generated the expected outputs, and the project was successfully synthesized. The game was implemented on the board to an extent. The LEDs appear to be frozen in state when the BTN_NORTH enable was pushed. The RESET button appear to work, but the inputs of the switches could not be validated because of the LEDs resisting to change their states.