# CSE 100 – Lab 4: Reaction Timer Game

Name: Benjamin Gofman

Date: November 5, 2025

Lab Section: Tu, Thu, 3:20 - 5:10

## 1. Description

The goal of Lab 4 is to design and implement a reaction timer game using Verilog on the Basys3 FPGA board. The design integrates multiple hardware modules, including counters, random target generation via LFSR, a finite state machine (FSM), and a display system utilizing LEDs and seven-segment displays. The game randomly selects a target LED position, then the player must flick the switch to match the target number. The FSM manages game logic, determines wins or losses, and updates the score.

## 2. Design

The system design consists of several modules working in synchronization with a generated clock. The following subsections describe each module and its purpose.

### top_lab4.v

Top-level integration module connecting all submodules, clock dividers, FSM, and I/O interfaces.

### qsec_clks.v

Divides the 100 MHz clock to generate slower signals: digsel (7-seg refresh) and qsec (game timing).

### time_counter.v

Counts quarter-second ticks to track elapsed time in the game.

### edge_detector.v

Detects rising edges of button inputs to register single presses.

### lfsr.v

Generates pseudo-random numbers to select the target LED for the game.

### decoder.v

Converts a 4-bit binary input into a one-hot 16-bit output to drive the 16 LEDs corresponding to the board’s switches.

### fsm.v

Controls the state of the game: waiting for start, running timer, detecting matches, handling wins/losses.

### ring_counter.v

Drives anode lines in a rotating fashion to indicate the active seven-segment digit.

### selector.v

Selects one of four 4-bit segments from a 16-bit bus to multiplex digits for the seven-segment display.

### hex7seg.v

Encodes hexadecimal values into seven-segment display patterns.

### Finite State Machine (FSM)

The FSM controls the flow of the Quick Decode game. Its states include IDLE (waiting to start), SHOW (displaying the random target LED), COUNTING (ring counter active, awaiting a match), WIN (successful reaction), and LOSE (timeout). State transitions depend on button input, timer signals, and match detection, all synchronized to quarter-second clock pulses.

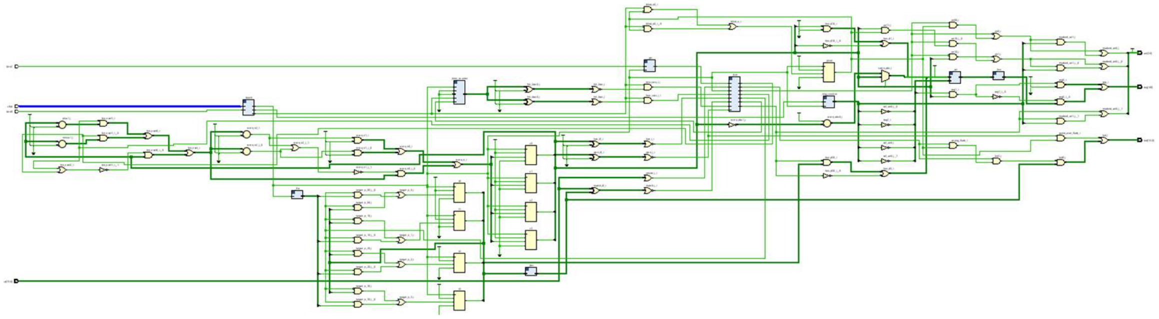

Figure 1. Original RTL ANALYSIS Schematic.

## 3. Testing and Simulation

Simulation was performed using the testbench (testTC.v) to validate the timing behavior and FSM transitions. Waveforms were examined to confirm correct LED rotation timing, target selection randomness, and scoring logic. Key test cases included: (1) pressing a button or switch, (2) missing the target, and (3) exceeding the timeout period.

## 4. Results

The Quick Decode game successfully demonstrates correct functionality on the board. Segments rotate smoothly, the target LED is displayed randomly, and the score increments or decrements appropriately. Timing accuracy was verified via simulation and confirmed in hardware testing on the Basys3 board.

## 5. Conclusion

In this lab, we built a Quick Decode game on an FPGA using a modular Verilog design. By combining FSM control, random number generation, and LED display logic, we created an interactive and responsive system. In the future, the game could be improved with adjustable difficulty levels and a more detailed scoring display.

## 6. Appendix

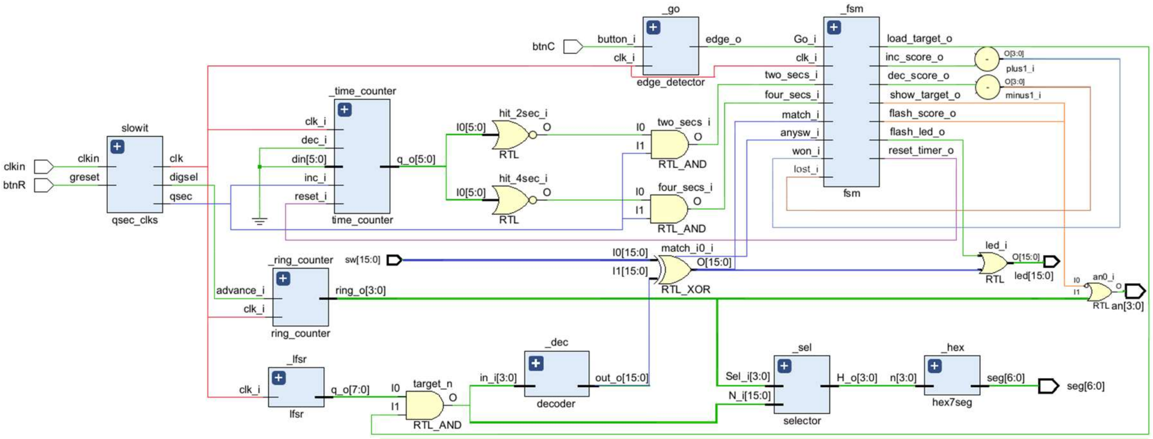

Figure 2. Custom design of RTL ANALYSIS.