Name: Christian Okyere
Project 3: Reaction Timer
Date: 10/20/2023

Reaction Timer

The system is created to assess a person's response time and is managed by a state machine programmed in VHDL. The setup comprises a driver for a 7-segment display, as well as a top-level file in either VHDL for simulation or BDF for configuring the board.

Testing for the reaction time

1. The user engages with the circuit using three buttons: "reset" (rightmost button), "start" (middle button), and "react" (leftmost button).
2. The circuit operates in three primary states: "Idle," "Wait," and "Count."
    - In the "Idle" state, "00" is displayed on two digits, and the rightmost green LED (green LED 0) is illuminated.
    - Transition to the "Wait" state happens when the user presses the "start" button. The circuit then enters a waiting period.
    - Following the wait period, it switches to the "Count" state, activates green LED 9, and begins counting milliseconds.
3. The user can interrupt the counting by pressing the "react" button, causing it to pause.
    - If the user reacts before green LED 9 lights up, the counter displays "FF," and the system returns to the "Idle" state.

4. Pressing the "start" button resets the digit display back to "00."
5. The user's interactions and the circuit's responses are controlled by the mentioned buttons and the three main states: "Idle," "Wait," and "Count."

I have attached a video of the reaction showing how the reaction timer works.

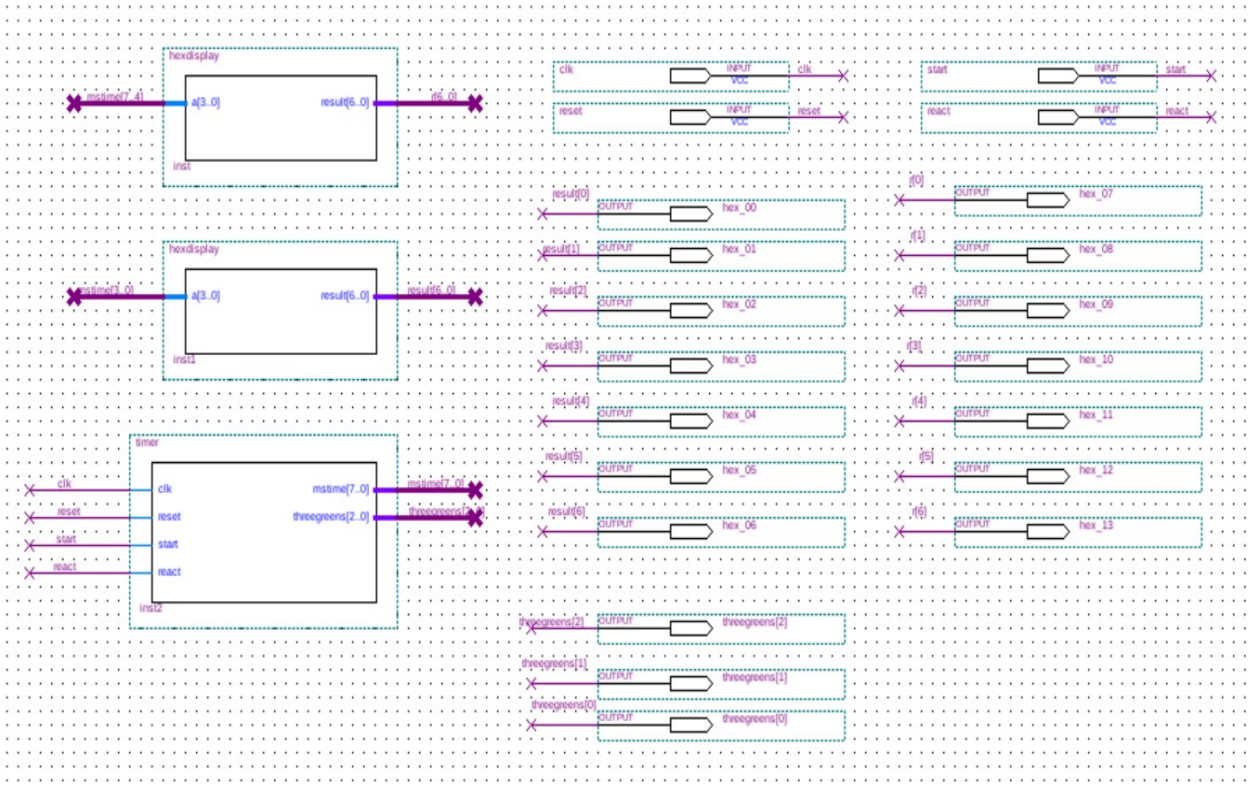

Figure 1: A figure showing the circuit diagram for the reaction.bdf

Extensions

As part of this extension, I introduced a random wait time feature. This involved the incorporation of a 4-bit register known as "randomDisplay," responsible for displaying a random value on the 7-segment display when the circuit is in the "sWait" state. The value for "randomDisplay" is derived from the feedback loop of the last 3 bits of the "randomTime" register, using an XOR logic operation.

Initially, the "randomTime" register is set to all '1's, and it represents the random wait time. During each clock cycle, the value of "randomTime" is updated by shifting the last 3 bits one position to the left, combining them with the result of the XOR operation between "randomTime(3)" and "randomTime(2)." This process generates a pseudo-random sequence of bits that is utilized to determine the value displayed on the 7-segment display while the circuit is in the wait state.

Consequently, the "randomDisplay" signal is refreshed with the last 4 bits of the "randomTime" register, ensuring a random display value during the waiting phase.

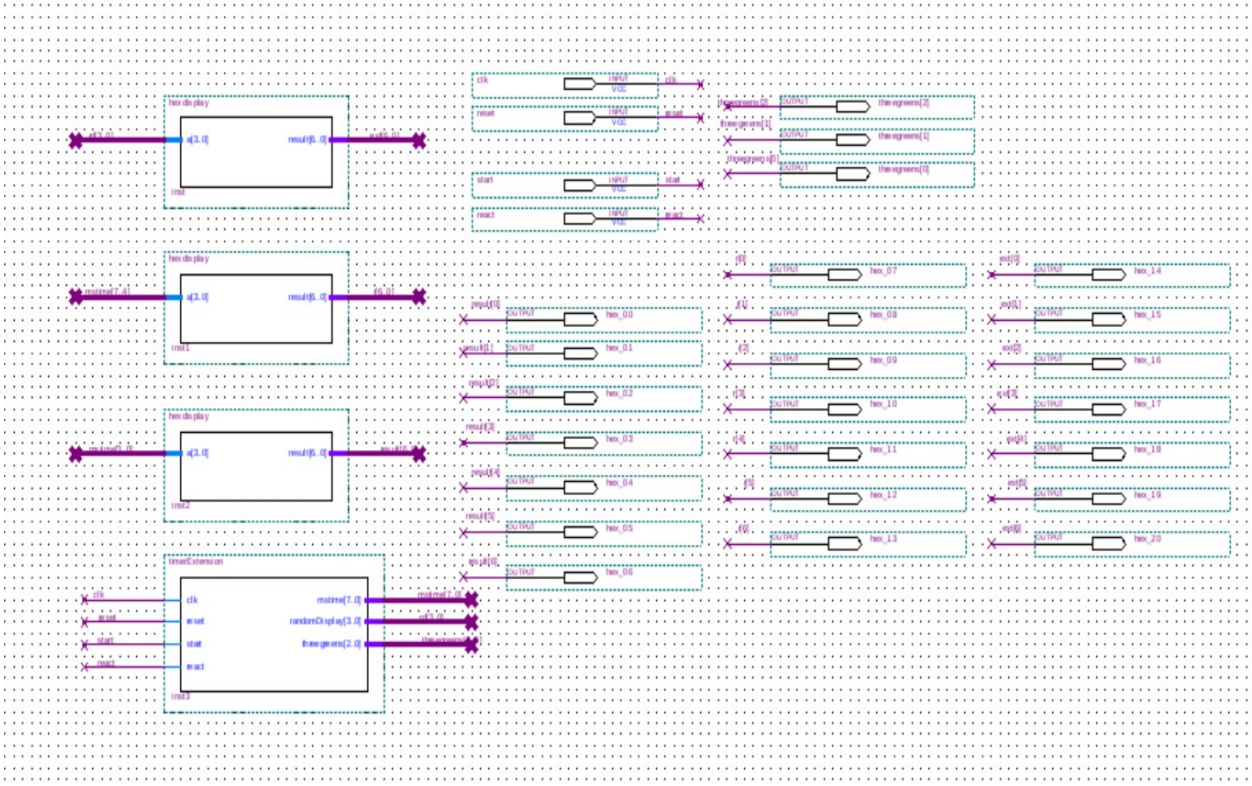

Figure 2: A figure showing the circuit diagram for the reactionExtension.bdf

Testing for the extensions

To evaluate the functionality of my extension, I included an additional hexadecimal display within the "reaction.bdf" file. I defined its outputs and set up the required pin connections. Subsequently, I conducted a board simulation. In the course of the simulation, I pressed the start button, causing the circuit to calculate and display the randomly generated wait time on the leftmost hexadecimal display unit. I have attached a video of the reactionExtension showing how the extension works.

Acknowledgements

Delanyo Nutakor, Alex Solano