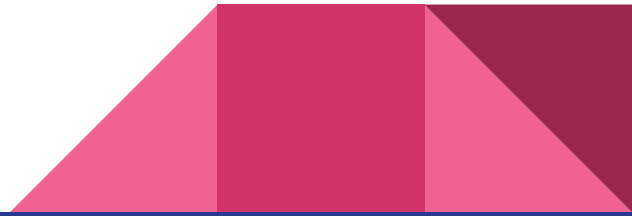
VHDL Stopwatch with Lap ROM Function

Ryan Paye
Nathan Fazecas
Jenna Grajewski

ECE 2700 TR(5:30 - 7:17 PM)
Daniel Llamocca

Objective

1. To make a Stopwatch correctly using both design aspects and VHDL coding methods taught in class. Mainly used with State Machines to aggregate elaborated design as a technique.
2. Streamline code/design so there is limited chance of unforeseen error during implementation and bitstream generation. Also, looks better with a lot less lines of code.

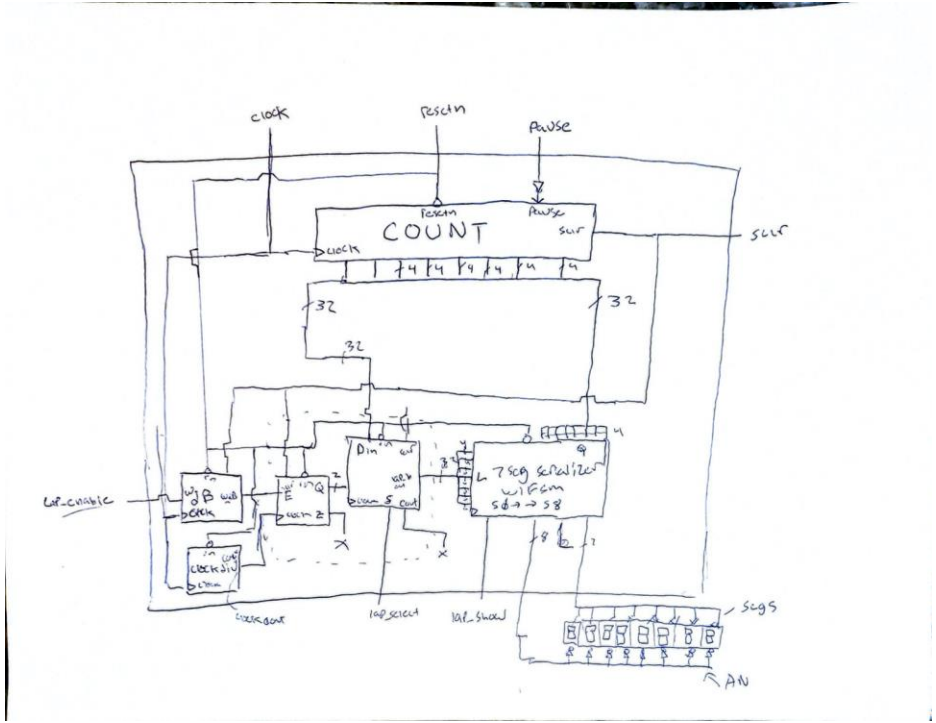


Design Pathway

Overall, stopwatch designed to pass bit count to an external 7segment display board.

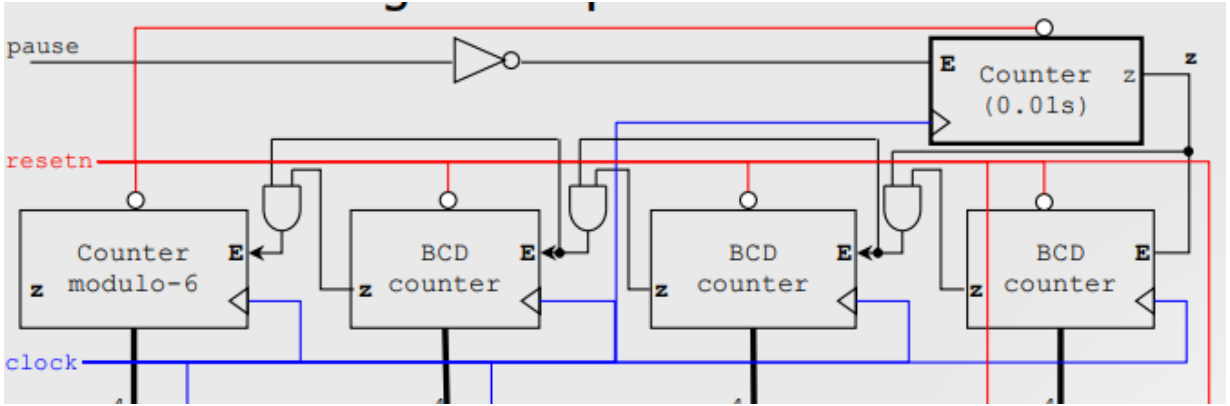
1. 6 BCD-Counters and 2 Modulo-6 Counters make up the most essential part of stop watch.
2. ROM functions as a LAP function w/input from a 2 bit Counter (Integer = 5) activated by a debounced push button.
3. Count and Lap time sent to a 7seg Serializer. 2-1 MUX with Lapshow select decides whether the serialized value passed to the BCD- 7seg converter is the lap or rolling time.

Schematic Design



Count/Counters

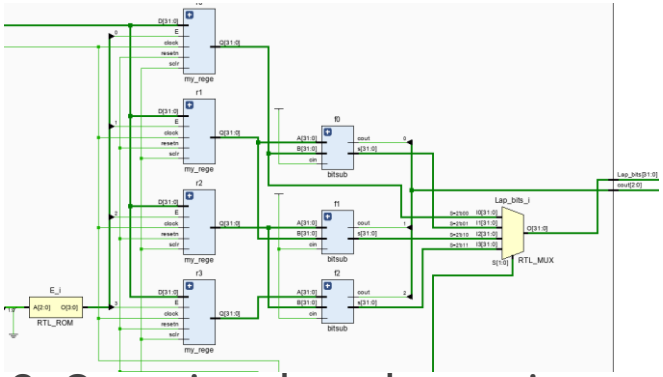
Credit - Dr. Llamocca



The count is given through a variety of 2 different by means of generic/port mapping. Enables activated by function E(i+1) = (z(i) and (E(i).)

1. 0.001 second counter with Integer = (10**5.) Multiply int with the period of clock 1/100 MHz (10 ns.) Not passed to 7segs.
2. 6 BCD counters Integer =10.
3. 2 Modulo-6 counters Integer =6.

Lap ROM

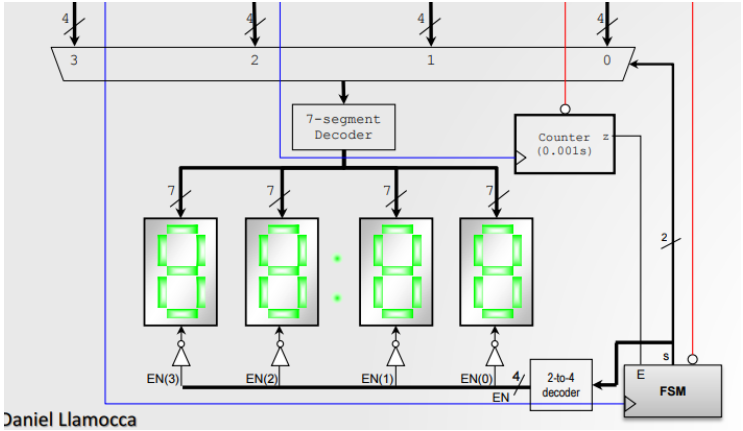


Lap Function used as a ROM similar to lap 3. Contains decoder, registers activated by decoder/enable output, and a MUX for writing memory. Push button increments the encoded bit on the decoder

1. Decoder takes in counter value from the push button activated counter (Q goes from 1- 4 in 2 bits.) Output enables 4 registers.
2. Registers pass running time to MUX. (laps2,3,4 Must be converted)
3. 3 32-bit subtractors give the true lap time to MUX

Serializer

Credit - Dr. Llamocca

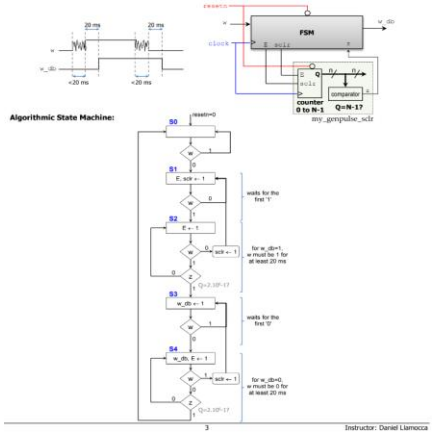


Common cathode on each BCD converter. Therefore design must send only 1 4-bit BCD value to the 7seg converter. Cannot tie display to multiple converters

1. FSM controls state which acts as multiplexor select
2. Still need to turn on displays. Send the state to a 3-8 decoder to activate display enables (active low)
3. Two 8-1 mux's used to select between lap and running time. All have the state values as inputs

Debouncer

Credit - Dr. Llamocca



Much mechanical bouncing with a push button. Noise comes from elasticity between metal contacts, and this repeatedly opens and closes circuit. Takes a few milliseconds to finish (much too slow for our 100 MHz clock.)

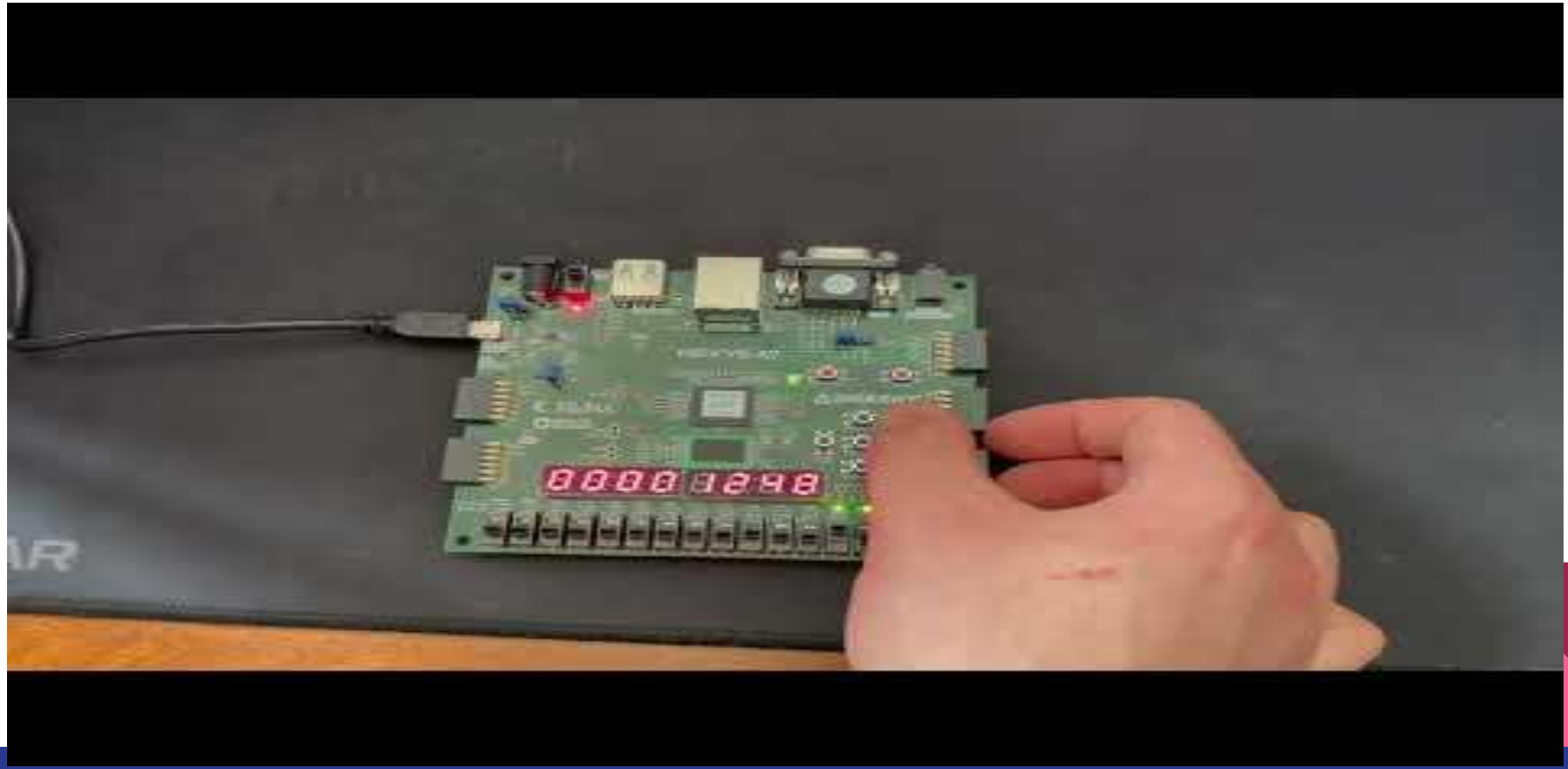
1. Function incorporates a state machine that cycles until the button stays at the same value consistently, must be at 20ms due to counter between each input.

Implementation/Connection of Code/ Tricks Used

1. Generic mapping: Much more concise to recall one class in TOP modules than having to recall multiple counter components of different integers.

1. For loops make port mapping large structures easier. Specifically used this in implementation of 32-bit add/sub.

Demonstration Video



Conclusion/Final Thoughts

This made us more knowledgeable of all the aspects of circuit design and VHDL. Formatting was much easier when using a bunch of these techniques and aspects. This was reinstated from the aspects of collaboration done within the group. In the future, it would be fun to make the stopwatch have more functionalities that involve more elaborate designs.