EET 3030 Embedded Controllers
Laboratory Report

Lab 6
Stop Watch Design Using VHDL
Using Altera Quartus II v 15.0

Presented
By

Spencer Rosenvall
W01351365

Fall 2018

Electronics Engineering Technology
Department of Engineering Technology
College of Engineering, Applied Science and Technology
Weber State University

Introduction and Overview

Design and simulate stop watch according to instructions from lab 6 for EET3030. This consists of coding inputs, and outputs and testing with FPGA.

Circuit is expected to change behave correspondingly to 4 bit adder, respectively.

Equipment and Parts

Laptop, Quartus II software, FPGA.

Theory

Stop watch counts down from 9 seconds, key 1 resets the stop watch, key 0 pauses and resumes the stop watch.

7 Segment Displays

Flow operation, CLK rising edge theory.

Design Calculations

1. Write the code for stop watch, use keys0-1 for reset and pause, use 7 seg to indicate seconds.
2. Compile the code.
3. Test Code on FPGA.
4. Verify circuit functionality and decimal representations.

Schematic Diagram of the Circuits

```
CLK_50: in std_logic;
KEY: in std_logic_vector(1 downto 0); --start/stop, reset
HEX2: out std_logic_vector(6 downto 0)  -- 7 segment displays for seconds
```

INPUTS: CLK, KEY1, KEY0 OUTPUTS: HEX2

Logic / Code

```
IF(rising_edge(CLK_50)) THEN
   -- s 7-Segment display (6543210)
   IF s = 9    THEN
      HEX2  <= "0010000";  -- '9'
   ELSIF s = 8 THEN
      HEX2  <= "0000000";  -- '8'
   ELSIF s = 7 THEN
      HEX2  <= "1111000";  -- '7'
   ELSIF s = 6 THEN
      HEX2  <= "0000010";  -- '6'
   ELSIF s = 5 THEN
      HEX2  <= "0010010";  -- '5'
   ELSIF s = 4 THEN
      HEX2  <= "0011001";  -- '4'
   ELSIF s = 3 THEN
      HEX2  <= "0110000";  -- '3'
   ELSIF s = 2 THEN
      HEX2  <= "0100100";  -- '2'
   ELSIF s = 1 THEN
      HEX2  <= "1111001";  -- '1'
   ELSIF s <= 0 THEN -- Handle when stopwatch hits 0
      HEX2  <= "1000000";  -- '0'
   END IF;
```

7 Segment Display representations of Time on stop watch.

```
-- key pressed event start/stop
IF KEY(0) = '0' THEN
   IF stop = 0 THEN
      stop := 1;
   ELSE stop := 0;
   END IF;
END IF;

-- reset, running, stopped
IF KEY(1) = '0' THEN
   s := 9;
ELSIF KEY(1) = '1' THEN
   IF stop = 1 THEN
      s := s;
   ELSE
      track := track + 1;
      IF track = 25000000 THEN
         s := s - 1;
         track := 0;
      END IF;
   END IF;
END IF;
```

Logic for key events and tracking time elapsed.

Results

4 Bit Adder: The Stopwatch worked great. It would correctly pause and resume, as well as reset. Operation was as expected.

Conclusion

I was really stuck on this because I was trying to make a while loop like you would do in C programming. The solution which finally worked for me after being stuck on analysis and synthesis forever, was to treat the clock cycle as a while loop and have all my logic and conditions checked in there. I also learned more about what a clock is and that is has two edges. My program checks for the rising edge. I also learned about Quartus and its chain files. Save those when it asks.