CPE462 - VHDL: Simulation and Synthesis - Fall'11

Prof. Nuno Alves (nalves@wne.edu), College of Engineering

WESTERN NEW ENGLAND UNIVERSITY | WNE

Homework Assignment #16

Due Date: Wednesday, November 30 2011

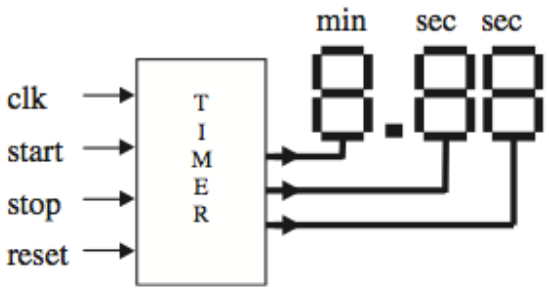
You may work in pairs for this assignment.

Question - Implement in our FPGA boards a timer capable of running from 0min:00sec to 9min:59sec.

- The circuit must have only two switches, one which must perform the start/stop and the other which performs the circuit reset.
- Look at the output wave-form so you can see the expected outcomes.
- The outputs must be seven segment display (SSD) coded.
- The input CLK is the reliable 50MHz clock signal.
- It is a requirement for this assignment to use a FSM to address the SSD time division multiplexing issues of our FPGA board.
- Make sure the circuit is synthesizable in hardware; for example do **NOT** initialize variables!
- You must use the following entity:

```
ENTITY timerCircuit IS
PORT (clk_50MHz, start_stop,reset : IN STD_LOGIC;
       min, sec1, sec2 : OUT STD_LOGIC_VECTOR (6 DOWNTO 0));
END timerCircuit;
```

- Use the the switch0 as the *reset* switch and the switch1 as the *start_stop* switch.
- Use three of the seven segment displays as the timer output.



Last updated on November 21, 2011 10:57 AM

Some useful information #1 - Modulus operation

Given two positive numbers, a (the dividend) and n (the divisor), a modulo n (abbreviated as a mod n) can be thought of as the remainder, on division of a by n. For instance, the expression "5 mod 4" would evaluate to 1 because 5 divided by 4 leaves a remainder of 1, while "9 mod 3" would evaluate to 0 because the division of 9 by 3 leaves a remainder of 0; there is nothing to subtract from 9 after multiplying 3 times 3.

VHDL has a modulus operation (MOD). Assuming signal *a* is an integer, MOD works as follows:

a MOD 2

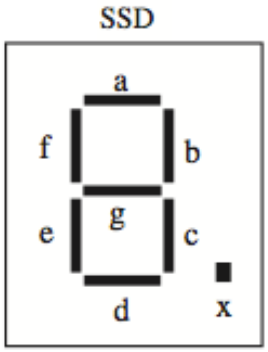
In this particular example 2 is the divisor.

Some useful information #2 - Seven Segment Displays SSD

A seven segment display is just a set of 7 LEDs, which becomes a 7 bit output. To write a coded zero into a seven segment display, all individual LEDS must be on, except LED g. So if the output signal was defined as:

signal temporary_bus : std_logic_vector (6 downto 0);

Then, the operation *temporary_bus <= "1111110";* will make sure a zero would appear on the display. Using the same rationale *temporary_bus <= "0110000";* will write a one.



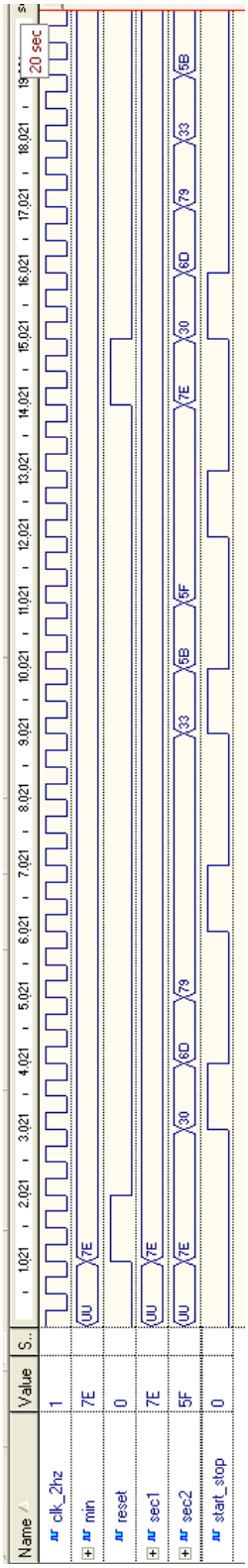
Input: "xabcdefg"

For your convenience, here is the full list of SSD coded digits.

```
0 ..... "1111110";     --7E
1 ..... "0110000";     --30
2 ..... "1101101";     --6D
3 ..... "1111001";     --79
4 ..... "0110011";     --33
5 ..... "1011011";     --5B
6 ..... "1011111";     --5F
7 ..... "1110000";     --70
8 ..... "1111111";     --7F
9 ..... "1111011";     --7B
```

Using the provided test-bench (with a clock signal of 2Hz instead of the 50MHz provided by the Spartan3 board) and running it for 20 seconds

test-bench is @ http://www.nunoalves.com/classes/fall11-cpe462/hw/q1.txt



Start timer (starts counting from zero)
Reset was pressed

Stop timer

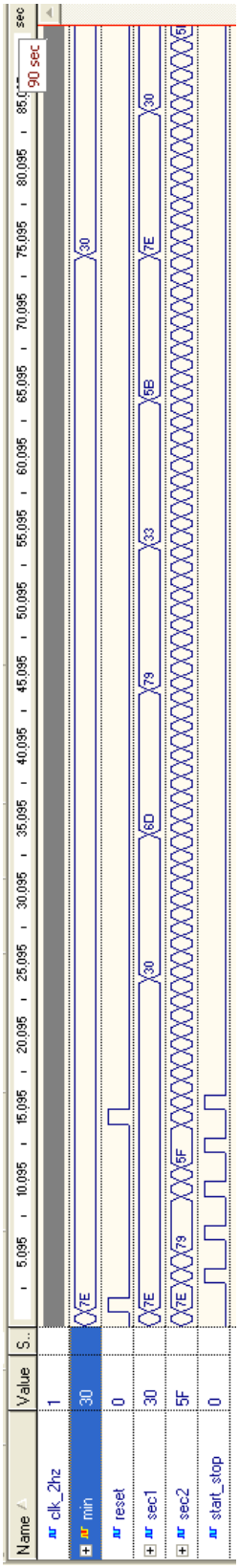
Re-start timer (reset was not issued so counting resumes)

Stop timer

Start timer

Reset was pressed so everything goes to zero

Using the provided test-bench and running it for 90 seconds

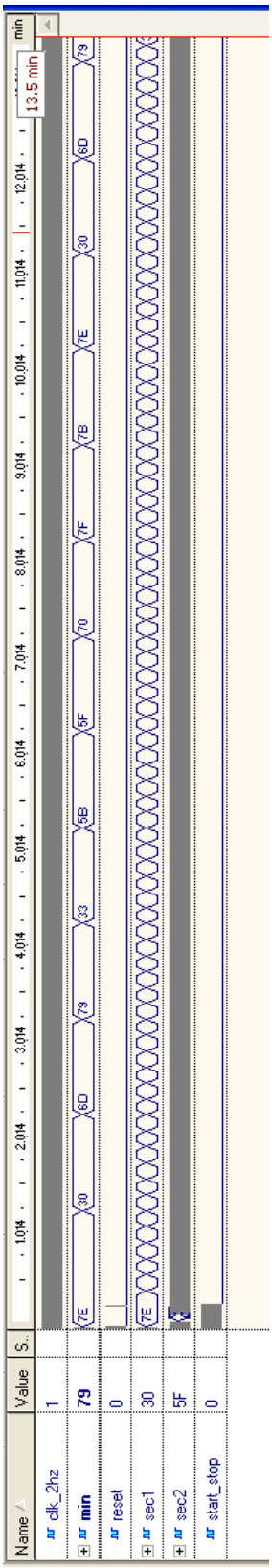


1 minute has passed!

50 seconds have passed

10 seconds have passed

Using the provided test-bench and running it for 810 seconds



After 9:59 minutes, the timer will reset to zero.