CLO215 HW-Assignment 2

- Report by Aastha Rajani (2021cs10093) and Piya Bundela (2021cs10118)

In this assignment we have been asked to implement a stopwatch and display it on the basys board. We have to control this stopwatch using four switches namely start, pause, continue, and reset.

The inbuilt frequency of the clock in the basys board is 100 MHz, whereas we have to display the minutes, seconds and every tenth of a second (10 Hz), so we have implemented a clock of 10Hz(clk_out) in module named timing_circuit. We have then defined four process in the module named num_new.

1. This process takes clk_out as an input and increments the value of tenth of second(in1) at every rising_edge of this clock. It also builds a new clock(o1) whose rising edge occurs after every 10 tenth of seconds.
2. This process takes o1 as an input and increments the unit value of seconds(in2) at every rising_edge of this clock. It also builds a new clock(o2) whose rising_edge occurs after every 10 units of seconds.
3. This process takes o2 as an input and increments the tens value of seconds(in3) at every rising_edge of this clock. It also builds a new clock(o3) whose rising_edge occurs after every 6 tens of seconds.
4. This process takes o3 as an input and increments the value of minute(in4) at every rising_edge of this clock. It also builds a new clock(out_wave) whose rising_edge occurs after every 10 minutes.
   We have then defined a new module int_to_bin which converts a given integer to its binary representation.
   These four bits i.e., in1, in2, in3, in4 are converted to their respective four_bit binary representations. These sixteen bits go as an input to the main module (which is the 1st assignment). This returns 7 outputs and 4 anode pins which correspondingly results in the display of numbers in the basys board.

Coming to switches, we have been asked to handle transitions from 0 to 1 only, so we have defined another module named switch

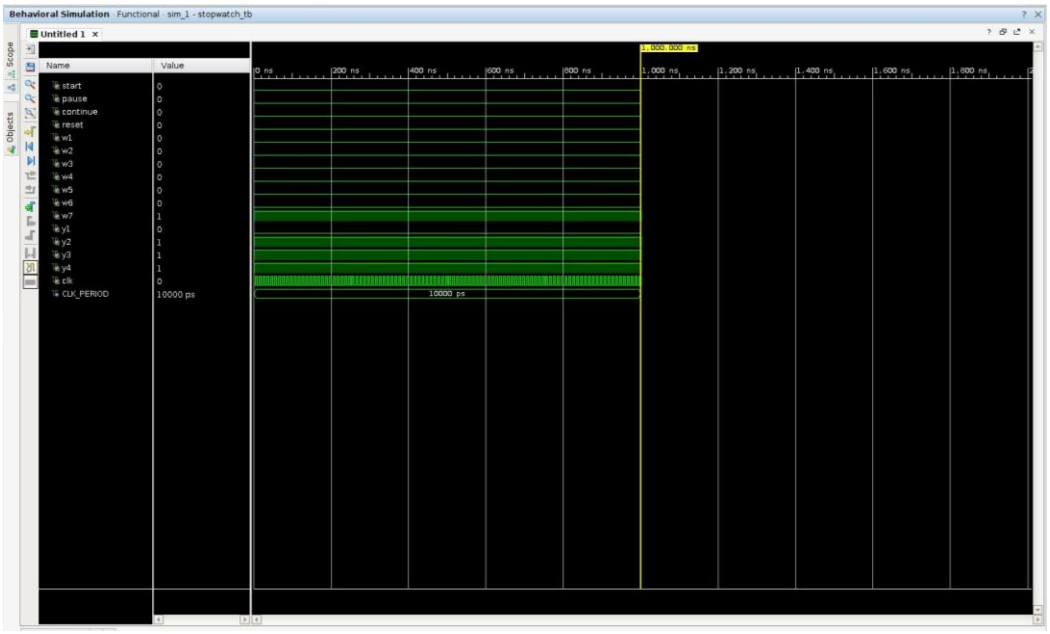
It takes for inputs start, pause, continue, and reset and outputs enable_watch and reset_watch.

We have declared four signals start_prev, pause_prev, continue_prev, reset_prev which basically stores the previous values of start, pause, continue and reset in order to record the transitions from 0 to1.

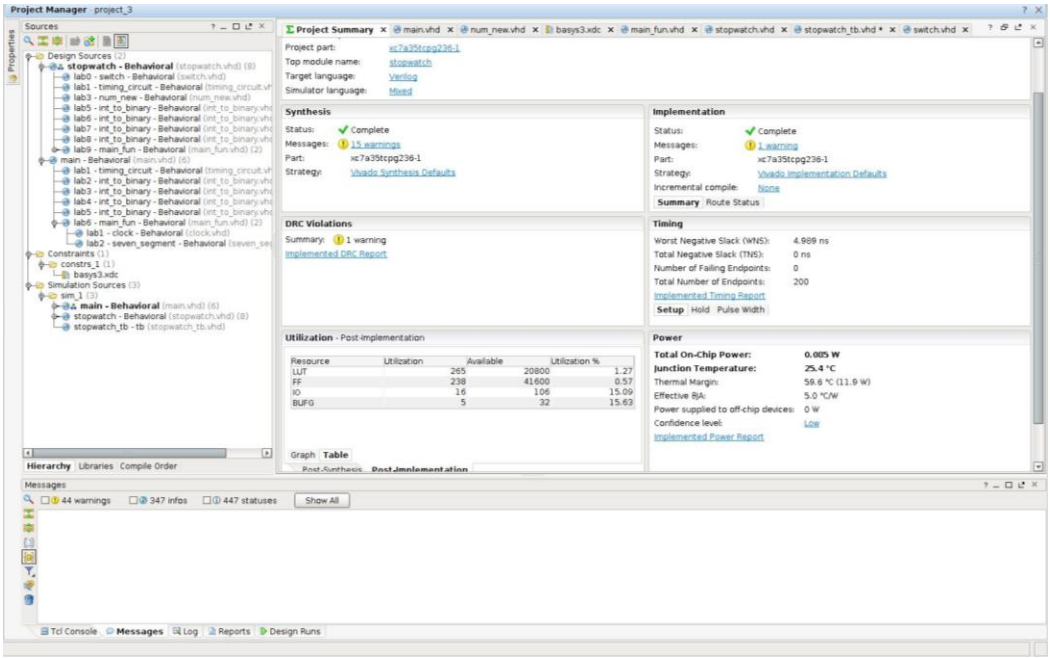
1. If start_prev is 0 and start is 1, we set the value of enable_watch to be 1 and reset_watch to be 0.
2. If pause_prev is 0 and pause is 1, we set the value enable_watch to be 0 and reset_watch to be 0.
3. If continue_prev is 0 and continue is 1, we set the value enable_watch to be 1 and reset_watch to be 0.
4. If reset_prev is 0 and reset is 1, we set the value enable_watch to be 0 and reset_watch to be 1.

If the reset_watch is 1 we set all the digits to be 0, else we continue with the processes mentioned above.

Simulation(test-bench)



Synthesis report



```
+---Muxes :
          7 Input      2 Bit        Muxes := 2
         10 Input      1 Bit        Muxes := 1
          3 Input      1 Bit        Muxes := 3
          9 Input      1 Bit        Muxes := 1
Module clock
Detailed RTL Component Info :
+---Adders :
          2 Input     32 Bit       Adders := 2
+---Registers :
                     32 Bit    Registers := 2
                      1 Bit    Registers := 2
+---Muxes :
          2 Input     32 Bit        Muxes := 2
          2 Input      1 Bit        Muxes := 6
---------------------------------------------------------------------------------
Finished RTL Hierarchical Component Statistics
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Part Resource Summary
---------------------------------------------------------------------------------
Part Resources:
DSPs: 90 (col length:60)
BRAMs: 100 (col length: RAMB18 60 RAMB36 30)
---------------------------------------------------------------------------------
Finished Part Resource Summary
---------------------------------------------------------------------------------
---------------------------------------------------------------------------------
Start Cross Boundary and Area Optimization
---------------------------------------------------------------------------------
INFO: [Synth 8-5545] ROM "count" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "tmp" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "count1" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "O20" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "count2" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "O30" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "count3" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "O40" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "count4" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "O50" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "p_0_out" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "p_0_out0" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "p_0_out" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "p_0_out" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "p_0_out0" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "p_0_out" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
INFO: [Synth 8-5545] ROM "p_0_out" won't be mapped to RAM because address size (32) is larger than maximum supported(25)
```