Digital Logic- ECSE 222

Describing Sequential Circuits in VHDL

LAB 2 REPORT

GROUP 38

Mustafain Ali Khan, ID: 260770776

Syed Hadi Zia Rizvi, ID: 260775855

A description of the counter and clock divider circuits. Explain why these two circuits are considered as sequential designs.

Counter:

The counter circuit is a 4-bit up counter which is positive edge triggered i.e. the counter value changes at the positive edge of the clock cycle. As required, the circuit also has an enable and reset signal. The reset is asynchronous and is active low, i.e. the count is set to 0 as soon as reset is 0. Counter works normally when reset is 1(high). When the enable signal is 0(low), the counter value does not change and it counts only when enable is 1(high).

Clock divider:

The clock divider circuit generates an output signal (en_out) that is high (‘1’) every 10 milliseconds. The clock divider is implemented with a down counter that counts down from (T-1) to 0, where T is the number of clock cycles before a high output signal is asserted/generated. Since the FPGA board has a clock signal with a frequency of 50Mhz, we calculate T by first calculating the time period of the clock which is 1/50Mhz = 20ns. Now if we divide 1oms by 20ns, we get a value of 500,000 for T. This means that the en_out signal should be 1 after 500,000 cycles of the clock. Therefore, the down counter counts from 499,999 to 0. The down counter is positive edge triggered and as a result the count decreases by 1 at each positive edge of the clock. When the counter reaches a value of 0, the output signal (en_out) becomes high (‘1’) and the count is reset to 499,999, this occurs after 10 ms. The reset and enable signals for the clock divider work similar to those in the counter. When enable is high, the counter works normally whereas when enable is low, the counter stops decreasing i.e. it holds its value. When reset is low, the counter value resets to 499,999 and on the contrary if reset is high, the counter works as usual.

Sequential circuits remember previous inputs and their output at any given time depends not only on the inputs at that time but also on the previous inputs. Sequential circuits use memory elements to remember the current state. As we are designing

circuits which require that they remember their previous inputs, these circuits are considered sequential. The counter is a sequential circuit because it needs to remember the previous count value in order to increment it. When the count reaches a value of “1111” and it is incremented by 1, it is reset to “0000”, for this to be possible the circuit needs to remember the value of the count.
The clock divider is also a sequential circuit because similar to the counter, the down counter in the clock divider needs to remember what the previous count value to decrement it. It resets the count from 0 to 499,999 when current value of count is 0 and its decremented by 1 i.e. it needs to know the current/previous count value.

Explain why even though we could build a clock divider using an up-counter it is easier to build the divider using a down-counter.

It is cheaper to implement the clock divider because we can use a single NOR gate to get a HIGH output when count is 0 (“0000000000000000”). Instead if an up-counter is used, a HIGH signal would have to be generated when the count is 499,999 which would require multiple AND gates to get an output of one.

A discussion of how the counter and clock divider circuits were tested, showing representative simulation plots. How do you know that these circuits work correctly?

Testing the counter:

To test the counter circuit a test bench was written, and simulation was done in ModelSim. A clock signal is applied and different combinations for reset and enable are used to ensure that the output can be checked for all possible input sequences. The simulation waveform is shown below in fig. 1.

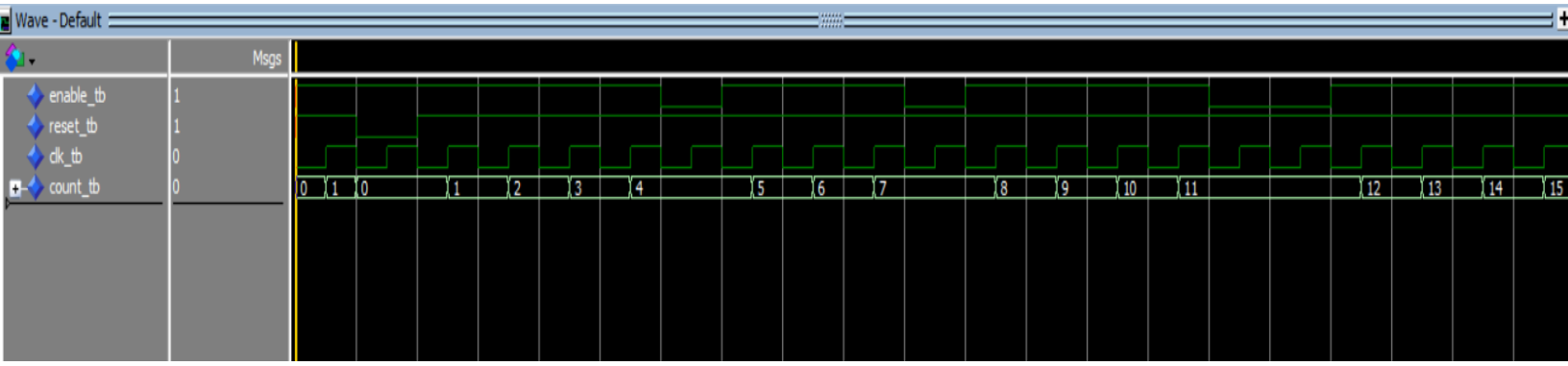


Figure 1. Counter Circuit Simulation

The working of the counter can be seen in the simulation. It can be seen in the simulation that the change in the value of the counter is positive edge clock triggered. When the enable is high, the counter increments its value for e.g when count is 2, enable is high and reset is high, the counter increments to 1. When enable is low, the counter holds its value (e.g when count reaches for 4, it stays 4 for 1 clock period when enable is low). When the reset is low, the value of the counter resets back to zero, this can be seen on the waveform when the count is reset to 0 when it was previously 1.

Testing the clock divider:

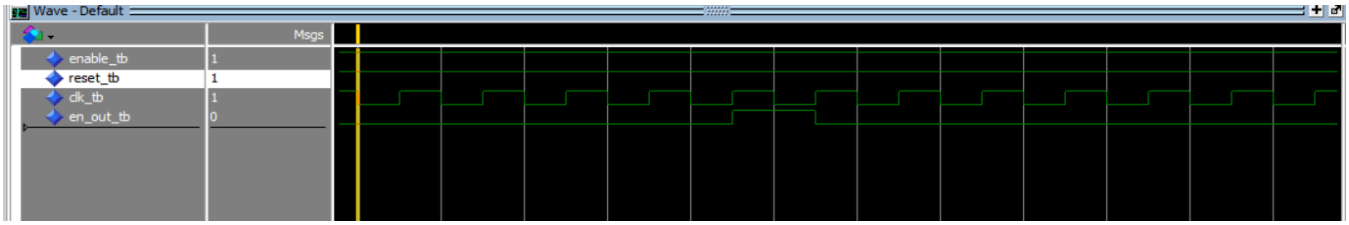


Figure 2. Clock Divider Circuit Simulation

Test bench was written for the clock divider circuit as well in order to check whether the circuit functions as required. Since the output signal must be HIGH after 500,000 clock cycles i.e 10 ms, it wasn't possible to show the entire 500,000 cycles of the clock on a single waveform but the fig.2 which is zoomed in, shows that en_out is high(this happens after 10ms). The simulation also shows that when reset is low, the output is low. Therefore, according to the simulation the clock divider works correctly.

A description of the stopwatch circuit. Explain why you created six instances of the counter circuit in your design and why?

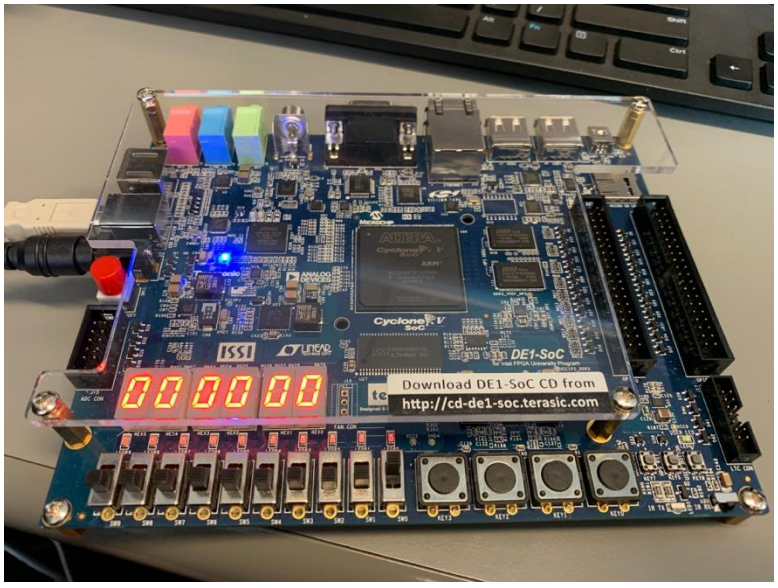
The stopwatch circuit has six instances of the 7 segment decoder(g38_hex_decoder) which are used to display the time for the stopwatch, each instance of the decoder is used for each HEX (HEX0-5) display on the FPGA board. Two decoders are used to display centi-seconds, two for seconds and the remaining two for minutes.

There is an instantiation of six counter circuits in which four counter circuits count from 0 to 9 and two counters count from 0 to 5. These values of counters are set by using the counter circuit clock and the reset signal. When HEX0 value changes from 9 to 0, it gives clock signal to HEX1, and similarly when HEX1 value change from

9 to 0 it gives clock signal to HEX2, this repeats for HEX2 and HEX3. When HEX3 value reaches 5, HEX3 resets to zero on the next clock cycle and gives a clock signal to Hex4, Hex4 gives clock signal to Hex5 when changes value from 9 to 0, same as HEX3, HEX5 also resets to 0 when its value is 5.

Testing the stopwatch:

The designed stopwatch can be tested as follows. We initially start by pressing the reset button which resets the value of the stopwatch to zero.



We then press the start button, and the stopwatch starts counting until the stop button is pressed. The following picture shows that it appropriately displays the time in centi-seconds, seconds and minutes.

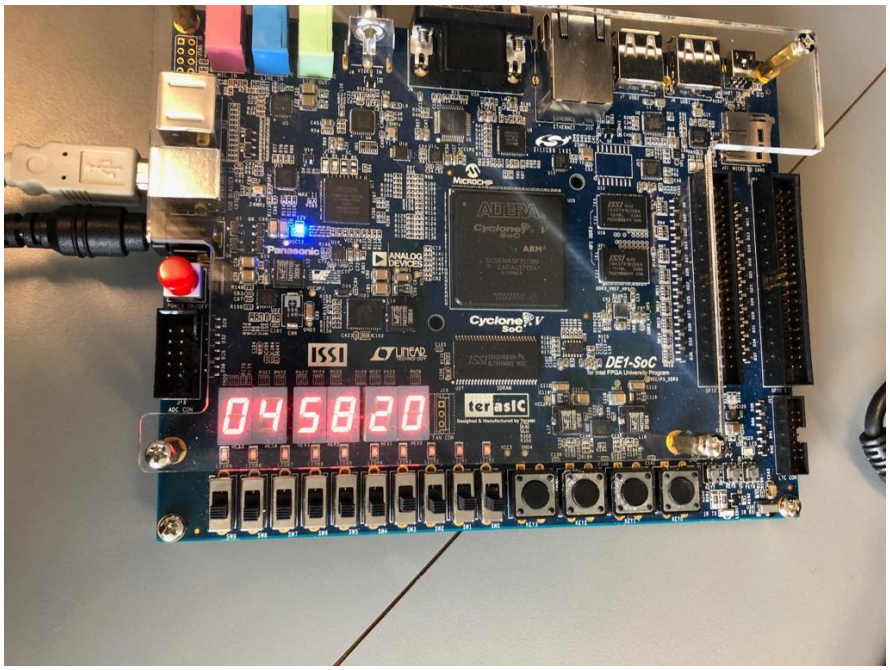


Figure Stopwatch working when start pressed

When the stop button is pressed, the value of the seven segments stops changing and the holds. If the start button is pressed again, the stopwatch will continue counting. If reset button is pressed, the value of the stopwatch resets back to zero.

A summary of the FPGA resource utilization (from the Compilation Report's Flow Summary) and the RTL schematic diagram for the stopwatch circuit. Clearly specify which part of your code maps to which part of the schematic diagram.

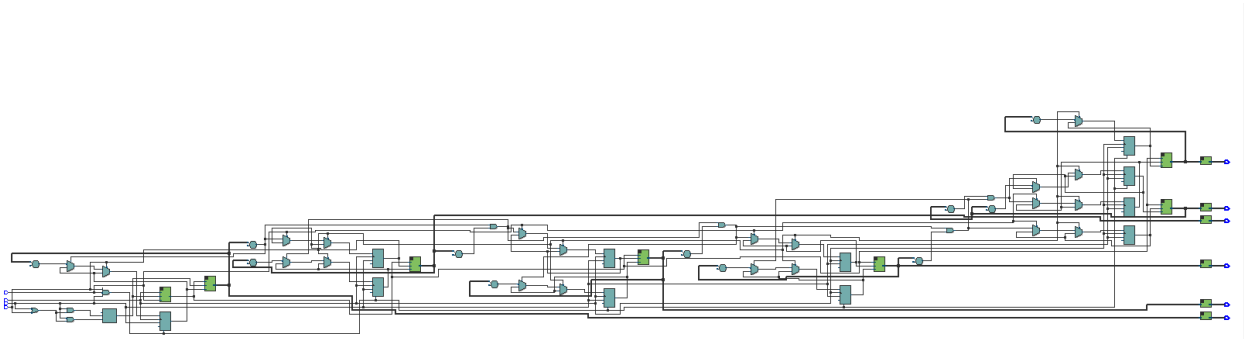
Flow Summary

<<Filter>>

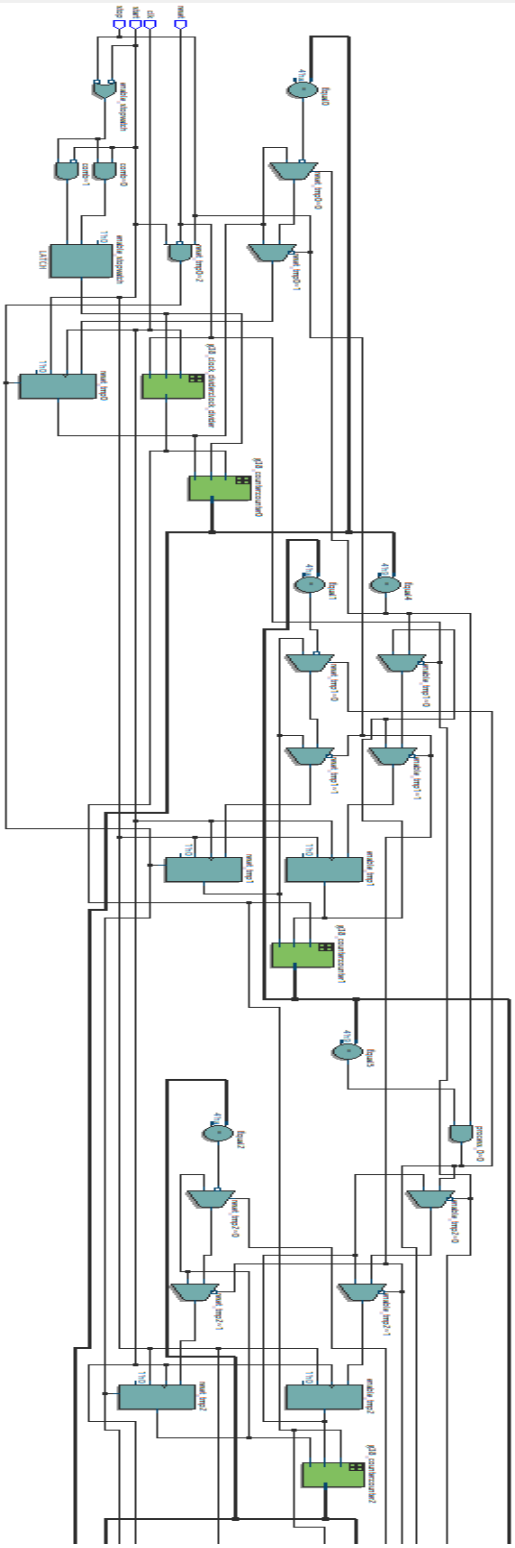
Flow Status	Successful - Fri Apr 12 21:16:51 2019
Quartus Prime Version	18.0.0 Build 614 04/24/2018 SJ Lite Edition
Revision Name	g38_lab1
Top-level Entity Name	g38_stopwatch
Family	Cyclone V
Device	5CSEMA5F31C6
Timing Models	Final
Logic utilization (in ALMs)	65 / 32,070 (< 1 %)
Total registers	54
Total pins	46 / 457 (10 %)
Total virtual pins	0
Total block memory bits	0 / 4,065,280 (0 %)
Total DSP Blocks	0 / 87 (0 %)
Total HSSI RX PCSs	0
Total HSSI PMA RX Deserializers	0
Total HSSI TX PCSs	0
Total HSSI PMA TX Serializers	0
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 4 (0 %)

Compilation summary

RTL Design

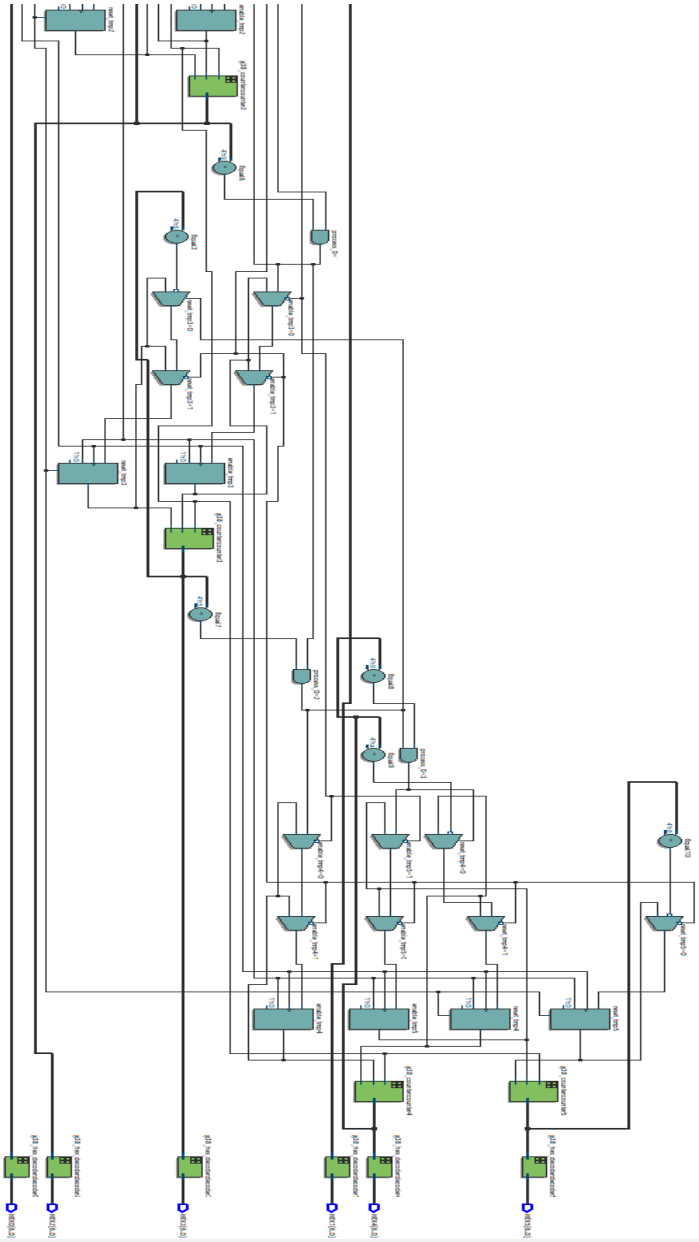


Rtl diagram 1



Part 1 of 2 of RTL schematic

Part 2 of 2 of RTL schematic



The RTL schematic shows that the stopwatch circuit consists of 6 instances of both the 7 segment decoder and the counter. It also has an instance as of the clock divider and has 3 more input signals namely start, stop and reset along with a clock signal. Part 1 of t he RTL shows the instance of the clock divider(clock_divider)

which generates an en_out signal every 10ms. This en_out signal is used as the clock input for all 6 instances of the counters(namely counter0, counter1, counter2, counter3, counter4, and counter5). The RTL schematic shows that the count output of each counter is used by an instance of the decoder to display the time on the board.
-decoder0 uses the count output of counter0.

-decoder1 uses the count output of counter1

-decoder2 uses the count output of counter2

-decoder3 uses the count output of counter3

-decoder4 uses the count output of counter4

--decoder5 uses the count output of counter5

The RTL schematic shows the 6 instances of the counters that work according to the logic defined above in this lab report.

Furthermore, the RTL schematic shows signals for the 6 temporary reset signals, 6 temporary enable signals and 5 temporary count signals(one for each counter except counter0) that are used for the stopwatch VHDL code. The multiple instances of the multiplexers allow the functionality of the start, stop and reset of the stopwatch(multiplexers implement the control flow in the vhdl code). The RTL schematic also shows the use of Data latches which are memory elements used to store the state of the pushbuttons(start,stop,reset).