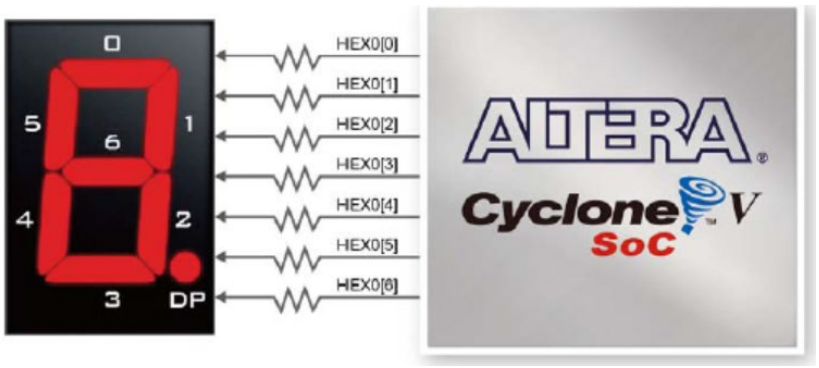
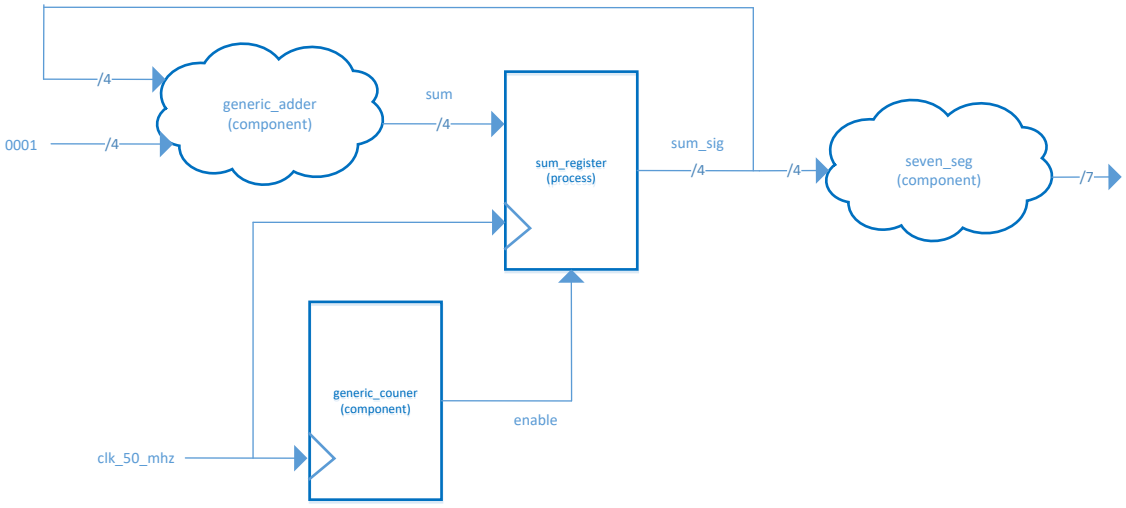
Laboratory 3: Seven Segment Display Counter

1 INTRODUCTION

In the previous lab we created a 4 bit adder via both behavioral and hierarchical techniques. In this lab we will put the adder to use as the main math engine of the design. The goal of this lab is to create a simple seven segment display counter that continuously cycles from 0 to 9 at an update rate of 1 Hz. This lab will give us the visualization piece of the puzzle that is needed to see the output of the custom processor. In order to target the seven segment display on the DE1-SoC board one must first determine which pins on the Cyclone V map to the desired LEDs. For this lab we will be using the right-most seven segment display. One can find the exact pinout by trolling through the DE1-SoC user's manual which is posted online. Apart from coding up the binary coded decimal conversion code, one needs to use the generic counter component to create the necessary 1 Hz timing update. The block diagram below shows the parts of the design that are all instantiated via the top module. It is important to understand that both registers are clocked on the 50 MHz edge however the sum_register process uses a low rate single-clock-pulse-wide enable signal that is generated from the generic_counter component.





The concepts of synchronicity and hierarchy are explored in this lab along with other tricks such as creating a custom radix in Modelsim and Quartus compilation via scripting. Version control is not required for this lab.

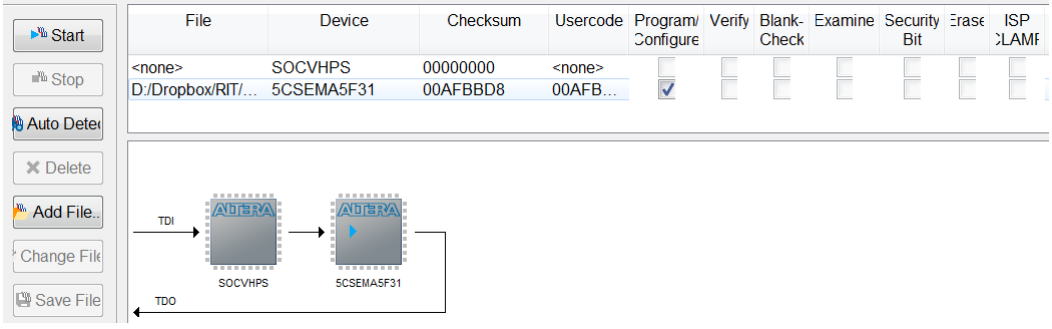
2 BLINK SIMULATION

- ☐ Unzip and simulate blink.zip by clicking on simulate.bat in the /sim/script folder. Make sure you have the below folder structure as sometimes unzipping a file adds another folder layer.
    - Lab3
        - blink
            - src
            - sim
                - src
                - script
- ☐ Receive a sign off but be ready to answer the below questions.
    - What is different between the blink simulation and the generic_counter simulation shown in class?

3 BASELINE HARDWARE [BLINK LED]

- ☐ Click on the compile.bat file in the /hw folder to compile the .sof file.
- ☐ Plug in your DE1-SoC board power and the USB blaster cable.
- ☐ Open up the Quartus Programming software and click 'auto detect'.
- ☐ Make sure to select the correct chip version.

- ☐ Add your file which is the .sof file located in the /hw/output_files/ directory and make sure the HPS row is on top



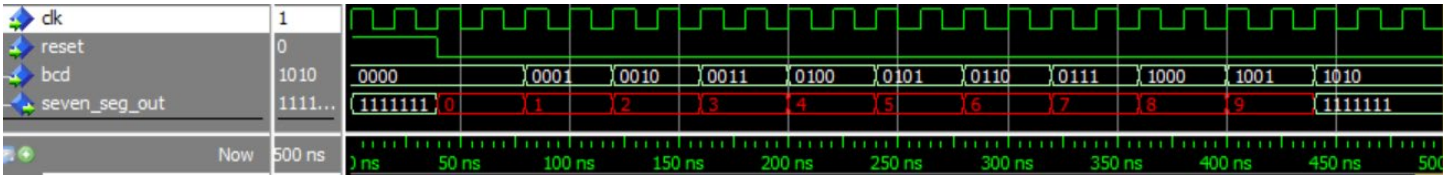
- ☐ Click 'Start'
- ☐ Try modifying the generic parameter to make the LED blink faster.
- ☐ Receive a signoff.

4 Seven Segment Binary Coded Decimal [BCD] Simulation

- Create a new folder under lab 3 and code up a BCD simulation that iterates through all possible 4 bit inputs.
  - Use the posted seven_seg_tb.vhd file if you like as your test bench.
  - BCD effectively converts the 4 bit input signal into the 7 necessary pins to turn on the seven seg display.
- Copy the below code into the wave.do file in order to create a custom radix in Modlsim.

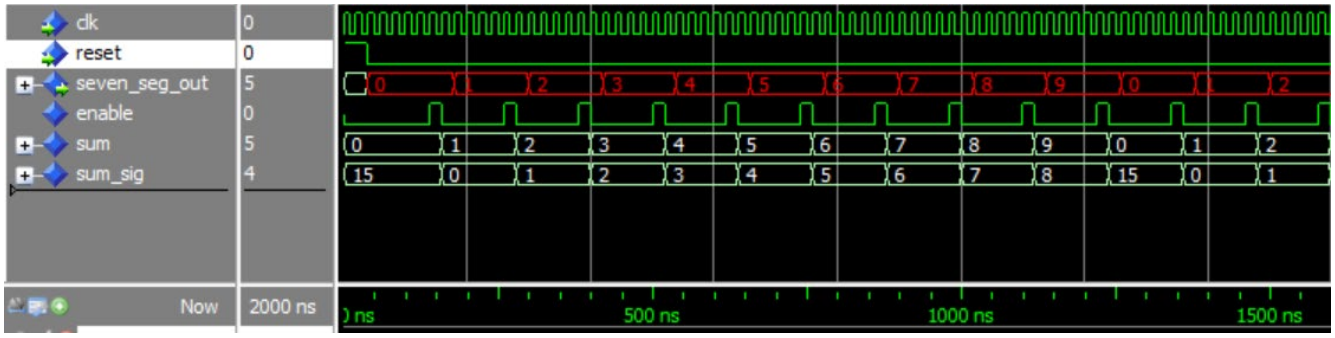
```
radix define States {
   "7'b1000000" "0" -color "red",
   "7'b1111001" "1" -color "red",
   "7'b0100100" "2" -color "red",
   "7'b0110000" "3" -color "red",
   "7'b0011001" "4" -color "red",
   "7'b0010010" "5" -color "red",
   "7'b0000010" "6" -color "red",
   "7'b1111000" "7" -color "red",
   "7'b0000000" "8" -color "red",
   "7'b0011000" "9" -color "red",
   -default default
}
```

- After running the sim, select a signal and right click on it, and set its radix to “States”. Then save the wave.do file and run the simulation again.
- Match the below waveform and receive a signoff.



5 Counter [Simulation]

- Create a counter simulation according to the block diagram in the introduction.
  - Make use of the posted generic_counter.vhd, generic_adder_beh.vhd
- Match the below waveform and receive a signoff.



6 COUNTER [HARDWARE TARGET]

- ☐ Modify the counter to update at 1 Hz and run it on HW
- ☐ Receive a signoff.

7 DELIVERABLES

To receive full credit for this lab one must hand in the below items no later than 168 hrs [7 days] after the start of one's lab session. Signoffs can be obtained after the due date as long as the time stamp of the code is from before the deadline.

- ☐ Hard copy of this document.
- ☐ Hard copy of all src files [no test benches needed][no tabs and print from notepad++ with 'show symbols' on].

8 SIGNOFFS

Category	Initials	Date	Points
Blink Simulation			/10
Baseline Hardware			/10
BCD Simulation			/20
Counter Simulation			/20
Counter Hardware Target			/20
Header/Comments/Tabs			/10
Deliverable			/10
Final Grade			/100