**Experiment #5**

**Reloadable 8-bit up counter**

Laboratory Report

Prepared for: ECE 526/L – Digital Design with Verilog and System Verilog Lab

Prepared by:

Jaymish Patel

**Introduction:**

The goal of this experiment is to create a module of counter using behavioral model which functions as reloadable 8-bit up counter. Here instead of describing the actual hardware implementation of the circuit, the behavior of the input and output is used to create the model.

**Methodology:**

Four files were created using Verilog with names aasd.v for asynchronous assert, synchronous de-assert function of reset synchronizer, counter.v for reloadable 8-bit up counter, top\_counter.v modules a top-level design unit, and counter\_tb.v consist of the Verilog test bench.

top\_counter.v and counter\_tb.v programs were simulated using Synopsis VCS after completing the development of the two programs. VCS program was compiled using these 3 steps:

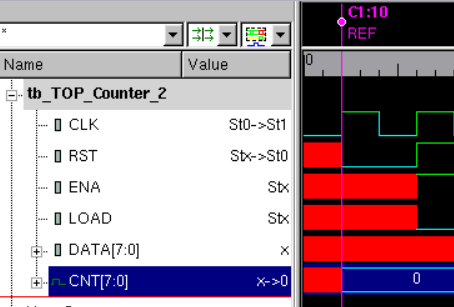
1. The 2 files were compiled using ‘vcs’ command.
2. ‘smv’ command was used to run the simulation.
3. ‘dve’ command was used to display the waveform of the simulation

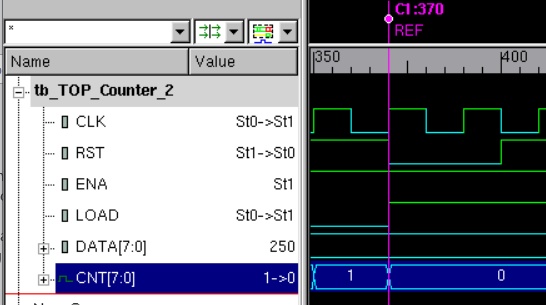
A waveform view of top-level signals consisting of DATA-data input RST, CLK, ENA, LOAD input ports and CNT-data output has been created using the DVE graphical interface.

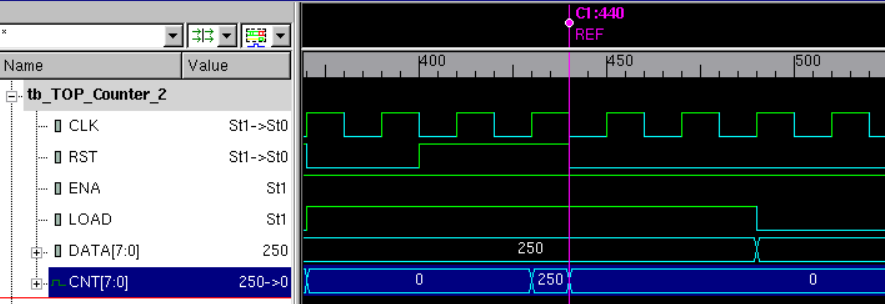
**Analysis of Results:**

Analyzing the waveform produced from the simulation, it is possible to consider some instances of output signal effects with respect to certain conditions of input signals. The result of analysis is summarized here. LOAD, CLK, ENA, RST are in binary and DATA & CNT in decimal for observation.

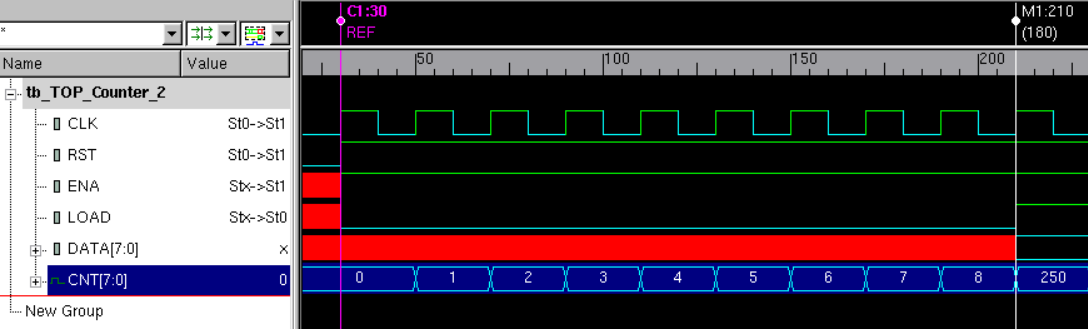
* *Demonstrates an asynchronous reset* - At 10ns RST =0, we get output as CNT=0 regardless of CLK,LOAD and ENA. Similar output is observed at 370ns and at 440ns due to RST=0.



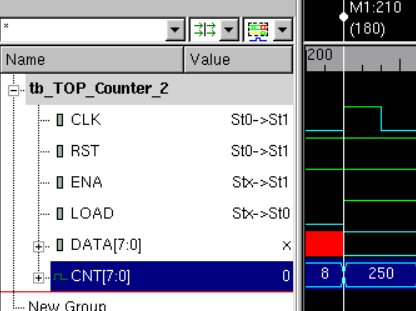




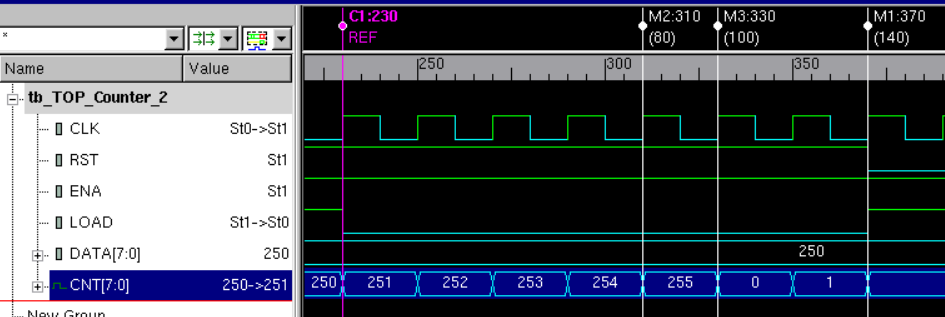
* *The counter initiates incrementing after reset is released* - At 30ns to 200ns RST=1, ENA= 1, LOAD=0 and at every positive edge of the clock we get CNT incremented by 1 upto 8



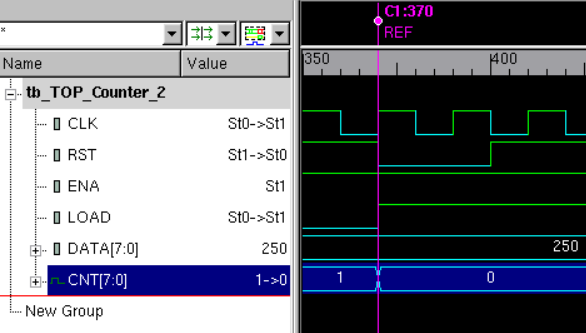
* *After it reaches a count of 8, parallel load 250 decimal is introduced* - Now at 210ns DATA=250 and LOAD=1,ENA=1,RST=1 so at positive edge of the clock DATA is propagated to the output. So CNT=250.



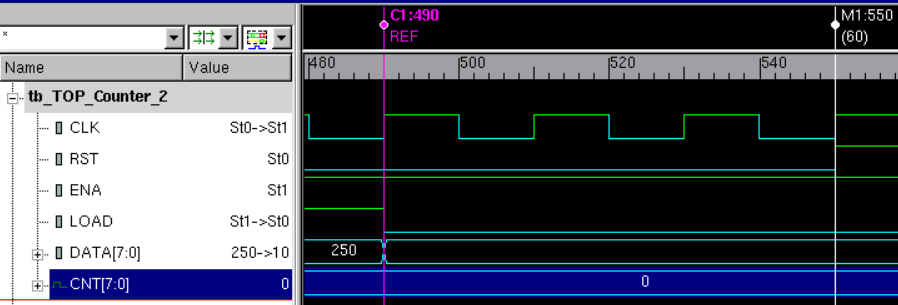
* *It will count from 250 until the counter rolls over (i.e. returns to zero) and then starts counting back up* - At 230ns LOAD=0, RST=1 and ENA=1 so the counter starts increamenting from 250 until 255 at 310ns. At 330ns the counter wants to increment but it reaches to its maximum limit so it starts counting from 0.



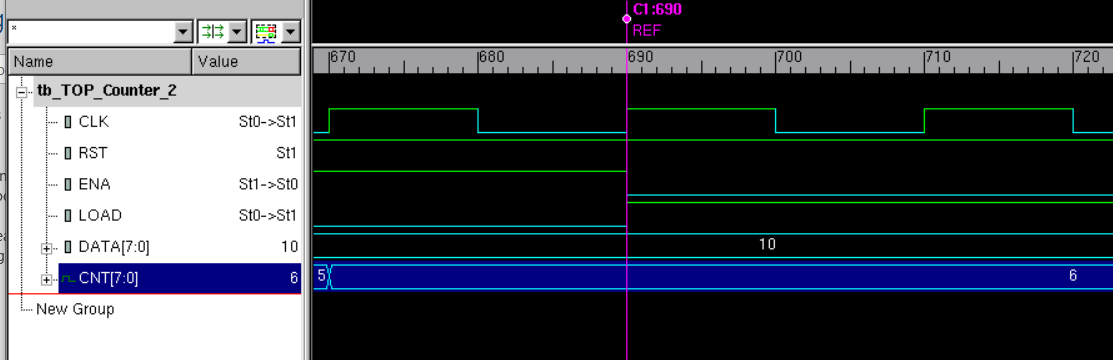
* *Reset overrides load* - At 370ns, LOAD=1 but RST=0 so the CNT gets reset overriding LOAD



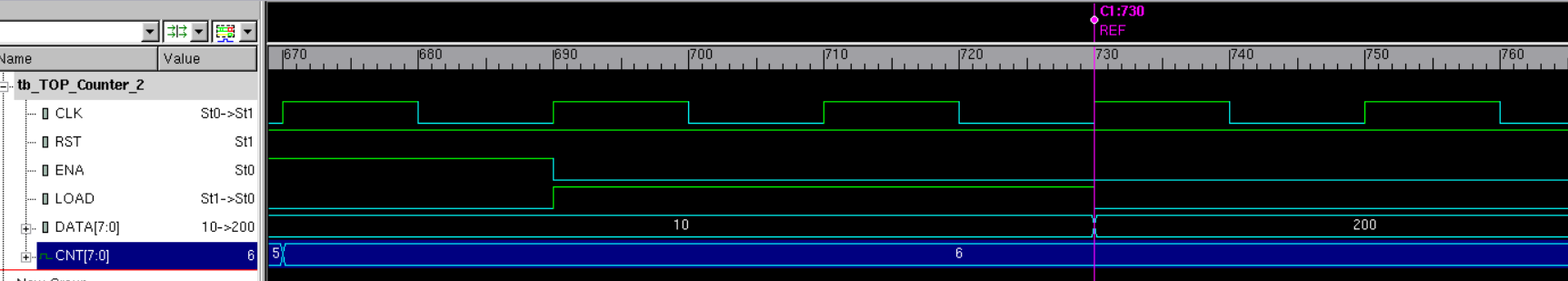
* *Reset overrides enable* - At 490ns LOAD=0, ENA=1 so increment should takes place but RST=0 so there will be no increment as reset overrides increment.



* *Enable low for Load* - At 690ns LOAD=1 and RST=1 so the DATA must be propagated to CNT but as ENA=0, DATA is not propagated to the CNT



* *Enable low for increment* - At 730ns LOAD=0 and RST=1 so the CNT must be incremented but as ENA=0, CNT is not incremented.



**Conclusion:**

So, from the above analysis we can determine that the reloadable 8-bit up counter when Instantiate with AASD reset it resets at RST=0 regardless of clock or any output, when RST=1 and ENA=1 but LOAD=0 the counter is incremented, the counter rolls over when it reaches its limit while incrementing, when RST=1 and ENA=1 but LOAD=1 the parallel load is propagated to CNT.

**Extra Question:**

Currently when RST=0, regardless of LOAD, ENA and CLK, it resets the output as it is asynchronous. If the RST is asynchronous then the output will reset only at the positive edge. So it the RST=0 it will wait for the positive edge of the CLK to reset the output.

I hereby attest that this lab report is entirely my own work. I have not copied either code or text from anyone, nor have I allowed or will I allow anyone to copy my work.

Name (printed) \_\_\_\_\_Jaymish Raju Patel\_\_\_\_\_\_\_\_\_

Name (signed) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_03/06/2021\_\_\_\_