**Experiment #6**

**Sum of Product**

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 sum of product using behavioral model and using 3 level hierarchy. 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:**

Two files were created using Verilog with names adder.v, nultiplier.v, register.v, second\_level.v, sumofproduct.v, sop.v, and sop\_tb.sv. adder.v for the behavioral description of an adder, nultiplier.v for the behavioral description of a multiplier, register.v for the behavioral description of a register with CLK, second\_level.v for the behavioral description using 2 register 2 multiplier and a adder, sumofproduct.v for the behavioral description using 2 second\_level instance and an adder, sop.v for the behavioral description of the sum of product circuit without hierarchy and sop\_tb.sv consist of the Verilog test bench.

sumofproduct.v sop.v and sop\_tb.sv 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. ‘simv’ 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\_IN, CLK, C0, C1, C2, C3-data inputs and OUT -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. Inputs and outputs are in decimal format.

Chart

Description automatically generated

Here we can observe that at each clock cycle the inputs of DATA\_in is transferred to R1, R1is transferred to R2, R2 is transferred to R3 and R3 is transferred to R4.

A picture containing graphical user interface

Description automatically generated

**A picture containing graphical user interface

Description automatically generated**

Here is the Dve representation of the last few outputs.

A picture containing graphical user interface

Description automatically generated

This is the output when error is forced and the program terminates due to the error occurred.

**Conclusion:**

So, from the above analysis we can determine that at a given time T the output will be determined as OUT=(DATA\_IN[T-1]\*C1+ DATA\_IN[T-2]\*C2)+ (DATA\_IN[T-3]\*C3+ DATA\_IN[T-4]\*C4) at every periodic cycle the value of the register s forwarded to the next register.

**Lab Question:**

Here I didn’t include reset in my design of the register. I just implemented a simple register having data input and a clock input. As we don’t need to reset the data in the register to check output some specific input.

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 \_\_\_\_04/03/2021\_\_\_\_