1 Project 4: Verilog Code Development using Cadence Xcelium – Run Tutorial

10 points

In previous projects, we learned how to use the MARS tool to develop and simulate MIPS assembly code. Next, we will use the Verilog HDL (Hardware Description Language) to develop and simulate various MIPS components.

We will use the Cadence Xcelium tool on the Lyle Unix servers. Please go to my web site http://lyle.smu.edu/~manikas/CAD_tool_info.html for information on how to do the following steps:

1. Obtain a Lyle Unix account, if you do not already have one: https://www.smu.edu/OIT/Services/genuse
2. Set up an X-Windows emulator on your computer

After you have set up your computer to run Xcelium, do the following to complete the project:

1. Please download the following Verilog files:
   - **S23_MIPS_ALU_basic.v** - a Verilog module for a simple MIPS arithmetic/logic unit (ALU).
   - **S23_MIPS_ALU_basic_tb.v** - the testbench for testing the ALU
2. This ALU performs the following 2 functions:
   - Function code 1 to implement bitwise OR (A or B)
   - Function code 7: set if A < B
3. Run the Verilog code using Xcelium:
   - Use the command: **xmverilog S23_MIPS_ALU_basic.v S23_MIPS_ALU_basic_tb.v**
   - Note that a copy of the testbench output will be stored in the **xmverilog.log** file by Xcelium
   - Please submit the resulting log file, edited with your name in the file.

2 SOLUTION

2.1 Result

After you run the program code, you should get the following results in your **xmverilog.log** file:

```
 0 Zero=1 ALUctl= 1 A=00000000 B=00000000 ALUOut=00000000
 2 Zero=0 ALUctl= 1 A=0000000c B=00000005 ALUOut=0000000d
 4 Zero=0 ALUctl= 1 A=0000000f B=00000006 ALUOut=0000000f
 6 Zero=1 ALUctl= 7 A=0000000f B=00000006 ALUOut=00000000
 8 Zero=0 ALUctl= 7 A=00000001 B=00000006 ALUOut=00000001
Simulation complete via $finish(1) at time 108 NS + 0
./S23_MIPS_ALU_basic_tb.v:34          #finishtime $finish;
xcelium> exit
```