University of Jordan
Computer Engineering Department
CPE439: Computer Design Lab

Experiment 1: Introduction to Verilogger Pro

Objective:
The objective of this experiment is to introduce the student to the environment of the Verilog simulator, and write a simple program to simulate the operation of AND and NAND gates.

The VeriLogger Pro Environment:
When you start the VeriLogger Pro program, you will notice that there are four windows. The upper left is the project window; in this window you select the HDL source files to be simulated. The upper right window enables the programmer to add a free parameter. The lower left window is the place where you will see the timing diagram that shows the waveforms of the signals monitored throughout the simulation. The lower right window is the place where the contents of the log file can be seen, and the errors of compilation are displayed.

How to write a program that describes the operation of AND and NAND gates? Perform the following steps:

1. Open a new project file by selecting "New HDL Project" from the Project menu. Name the project "AND_project.hpj". The name is given when you select "**Save HDL Project As**…" from the Project menu.
2. Open a new source file by selecting "New HDL File" from the Editor menu. A new window should appear in which you should copy the following Verilog code.

```
// This module describes 2-input NAND gate behaviorally
module NAND (out, in1, in2);
 input in1, in2;
 output out;

 assign #2 out = ~ (in1 & in2);

endmodule
```

3. Save this new HDL file as "NAND.v" by selecting "**Save HDL File As…"** from the Editor menu.
4. Add NAND.v to your HDL project by selecting the project window, right click in the workspace of this window, and select "Add HDL File(s)…".
5. Similar to Steps 2 through 4, add to your project a new file named AND.v that contains following code.

```
// This module describes 2-input AND gate structurally
 module AND (out, in1, in2);
    input  in1, in2;
    output  out;
    wire w1;

    NAND NAND1( w1,in1,in2);
    NAND NAND2 (out,w1,w1);

endmodule
```

6. Now you need to test your AND and NAND modules and verify that they operate properly. Similar to Steps 2 through 4, add to your project a new file named test.v that contains following code.

```
module test;
reg in1,in2;         //declaring in1 and in2 as registers for inputs
wire andout;         //declaring  andout as wire for output


AND  n1(andout,in1,in2); //Creating an instance of the module AND

initial begin: stop_at         //This initial statement selects
   #250; $finish;              //an appropriate simulation period
end                            //We choose it here to be 250 time units


initial begin :init
in1=0;
in2=0; //Initially set in1 and in2 to zero

// The $display statement prints the sentence between quatations in the
// log file. It Operates in the same way the printf function does
// in the C language.

$display("*** Eng. Muath AlHiari ***");
$display("Time    in1   in2    andout");

// The monitor statement prints the values of the different
// parameters whenever a change in the value of one of
// them or more occurs.

$monitor("%0d       %b      %b       %b",$time,in1,in2,andout);

end

always begin             // We use this always construct to
#10 in1 = ~in1;          // continuously vary the values of the input
end                      // registers in1 and in2, in order to have
                         // a simulation whose output continuously changes.

always begin
#20 in2 = ~in2;
end

endmodule
```

7. After you have added the required files start the program simulation by clicking on the green arrow in the center of the Tool bar. The results should appear in the log file and the waveforms should appear in the timing diagram.