Northwestern Polytechnic University

EE461 Verilog-HDL
Lab Assignment #4

Due day: 12/6/2016

Instruction:

1. Print your program for each question in "word" file and paste running results in screen shot on it. Printing results or electrical copies from monitor can't be accepted.
2. Please follow the code style rule like programs on handout.
3. Overdue homework submission could not be accepted.
4. Takes academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)

1. Generate 4 logic synthesized schematics from the examples on "Examples of Logic Synthesis Scripting" tutorial.

2. Design a detector to detect if all bits are 0s or all 1s for an 8-bits input. If all bits are 0s, one of two outputs, "zeroflag" is 1. If all 1s, the other of two outputs, "oneflag" is 1. After that, write a synthesis script to create gate level netlist (gate level module) and schematic.

3. Design 4-bits 2's complement number converter. And synthesize it to gate level and schematic.

4. Design 4-1 mux in continuous assign. Write a testbench to assign x & z to select-bits to observe what you are going to get.

5. Write 2 logic synthesis scripts to convert them to schematics and compare what the difference is.

```
a. always @(a, b, c) begin
        d = a + b;
        e = d + c;
 end

 b. always @(a, b, c) begin
        e = d + c;
        d = a + b;
 end
```

6. When designing a 5-to-1 mux by "case" structure, inferred latch will be generated if design isn't in proper way after logic synthesis. Please fix it and compare new design hardware schematic with original one.

```
module mux(a,b,c,d,e,sel,out);
       input a,b,c,d,e;
      input[2:0] sel;
      output out;
      reg     out;
```

```
       always @(a, b, c, d, e, sel) begin
            case(sel)
               3'b000: out=a;
               3'b001: out=b;
               3'b010: out=c;
               3'b011: out=d;
               3'b100: out=e;
             endcase
       end
endmodule
```