Lab Code [5 points]
Filename: chipInterface.sv
AndrewID: xinyew

```
`default_nettype none
module chipInterface
  (input  logic SW  [17:0],
   output logic LEDR[17:0]);

  multiplexer DUT (.a(SW[0]),
                   .b(SW[1]),
                   .sel(SW[7]),
                   .f(LEDR[15]));

endmodule: chipInterface
```

Lab Code [5 points]
Filename: vcs_test.sv
AndrewID: xinyew

```
module testBench();

  logic w1,w2,w3,w4,w5,w6,w7;

  comb_nands   mydesign(w1,w2,w3,w4,w5,w6,w7);
  test_module tester(w1,w2,w3,w4,w5,w6,w7);

endmodule: testBench

module comb_nands(
  input  logic A,B,C,D,E,F,
  output logic G);  //We're implementing: AB + EF' + D'F'

  logic D_not, F_not;
  logic f1, f2, f3;

  not  d_inv(D_not,D),
       f_int(F_not,F);
  nand g1(f1,A,B),
       g2(f2,E,F_not),
       g3(f3,D_not,F_not),
       g4(G,f1,f2,f3);

endmodule: comb_nands

module test_module(
  output logic test_a, test_b, test_c, test_d, test_e, test_f,
  input  logic test_g);

  logic [6:0] tb;

  assign {test_a, test_b, test_c, test_d, test_e, test_f} = tb;

  initial begin
    $monitor($time,,,"ABCDEF=%d, G=%b", tb, test_g);

    for(tb=0; tb < 10; tb = tb + 1)
      #10;

    #10 $finish;

  end

endmodule: test_module
```