Lab Code [10 points]
Filename: chipInterface.sv
AndrewID: eqian

```
module ChipInterface
    (output logic [15:0] LD,
     input logic [15:0] SW);

    lab1SOP Z(.A(SW[5]), .B(SW[4]), .C(SW[3]),
              .D(SW[2]), .E(SW[1]), .F(SW[0]),
              .valid(LD[15]), .vowel(LD[14]));

endmodule: ChipInterface
```

Lab Code [10 points]
Filename: lab1.sv
AndrewID: eqian

```
//`default_nettype none

module Zorgian
  (input logic a, b, c, d, e, f,
   output logic valid, vowel);

    logic a_not, b_not, c_not, d_not, e_not, f_not;
    logic nt1, nt2, nt2a, nt2b, nt3a, nt3b, nt3,
        nt4, nt4a, nt4b, nt4c, nt4d, nt5, nt5a, nt5b, nt6, nt7,
        nt8, nt9, nt10, nt11, nt12,
        nt13, nt14, nt15, nt16,
        nt17, nt18;
    logic at1, at2, at3, at4, at5;

    logic ft1, ft2, ft3;
    logic ft4, ft5;

    // inverters
    not (a_not, a),
        (b_not, b),
        (c_not, c),
        (d_not, d),
        (e_not, e),
        (f_not, f);

    // (d' + e' + f')
    nor (nt1, d_not, e_not, f_not);

    // (a + b + c' + d + f)
    nor (nt2a, a, b, c_not, d),
        (nt2b, nt2a, nt2a),
        (nt2, f, nt2b);

    // (a + c + d + e + f')
    nor (nt3a, a, c, d, e),
        (nt3b, nt3a, nt3a), //0
        (nt3, f_not, nt3b); //

    // (a' + b' + c' + d + e + f')
    nor (nt4a, a_not, b_not, c_not),
        (nt4b, nt4a, nt4a),
        (nt4c, d, e, f_not),
        (nt4d, nt4c, nt4c),
        (nt4, nt4d, nt4b);

    // (a+b'+c+e+f)
    nor (nt5a, a, b_not, c, e),
        (nt5b, nt5a, nt5a),
        (nt5, f, nt5b);

    // (b' + c' + d' + e')
    nor (nt6, b_not, c_not, d_not, e_not);

    // (a + c' + e' + f')
    nor (nt7, a, c_not, e_not, f_not);

    // (a' + c + e' + f)
    nor (nt8, a_not, c, e_not, f);

    // (a' + b + e' + f)
    nor (nt9, a_not, b, e_not, f);

    // (b + c + d' + e')
    nor (nt10, b, c, d_not, e_not);

    // (a + b' + e' + f')
    nor (nt11, a, b_not, e_not, f_not);

    nor (at1, nt1, nt2, nt3, nt4),
        (at2, nt5, nt6, nt7, nt8),
        (at3, nt9, nt10, nt11);

    // valid output
    nor (ft1, at1, at1);
    nor (ft2, at2, at2);
    nor (ft3, at3, at3);
    nor (valid, ft1, ft2, ft3);

    // (e + f)
    nor (nt12, e, f);

    // d' + e' + f
    nor (nt13, d_not, e_not, f);

    // b + c' + d + e'
    nor (nt14, b, c_not, d, e_not);

    // a + f'
    nor (nt15, a, f_not);

    // a' + d' + f'
    nor (nt16, a_not, d_not,f_not);

    // b' + c' + d + f'
    nor (nt17, b_not, c_not, d, f_not);

    // a' + c + d
    nor (nt18, a_not, c, d);

    // vowel output
    nor (at4, nt12, nt13, nt14),
        (at5, nt15, nt16, nt17, nt18);
    nor (ft4, at4, at4);
    nor (ft5, at5, at5);
    nor (vowel, ft4, ft5);

endmodule : Zorgian

module Zorgian_test
    (output logic a, b, c, d, e, f,
     input logic valid, vowel);

    initial begin
        $monitor( "a = %b, b = %b, c = %b, d = %b \n",
                   a, b, c, d,
                   "e = %b, f = %b, valid = %b, vowel = %b \n",
                   e, f, valid, vowel);

        // if d e f = 1 0 1, should always be valid but not vowel

             d = 1; e = 0; f = 1;

        #10 a = 0; b = 0; c = 0;

        #10 if (!(valid == 1 && vowel == 0))
                 $display("FAIL: def = 101, abc = 000");

        #10 a = 1; b = 1; c = 1;

        #10 if (!(valid == 1 && vowel == 0))
                $display("FAIL: def = 101, abc = 111");

        #10 a = 1; b = 0; c = 1;

        #10 if (!(valid == 1 && vowel == 0))
                $display("FAIL: def = 101, abc = 101");

        // if d e f = 0 1 0 and valid then its a vowel

        #10 a = 0; b = 0; c = 0; d = 0; e = 1; f = 0;

        #10 if (!(valid == 1 && vowel == 1))
                $display("FAIL: def = 010, abc = 000");

        #10 a = 0; b = 1; c = 1; // valid and vowel

        #10 if (!(valid == 1 && vowel == 1))
                $display("FAIL: def = 010, abc = 011");

        #10 a = 0; b = 1; c = 0; // valid and vowel

        #10 if (!(valid == 1 && vowel == 1))
                $display("FAIL: def = 010, abc = 010");

        // vowel u

        #10 a = 1; b = 0; c = 1; d = 0; e = 0; f = 1; // valid and vowel

        // rest of letters that are not in def 101 or 010
        #10 $display("Valids and not vowels below");
        #10 a = 0; b = 0; c = 0; d = 0; e = 0; f = 0; // valid and not vowel
        #10 a = 0; b = 1; c = 1; d = 0; e = 0; f = 0; // valid and not vowel
        #10 a = 1; b = 0; c = 1; d = 0; e = 0; f = 0; // valid and not vowel
        #10 a = 1; b = 1; c = 0; d = 0; e = 0; f = 0; // valid and not vowel
        #10 a = 1; b = 1; c = 1; d = 0; e = 0; f = 0; // valid and not vowel
        #10 a = 0; b = 0; c = 1; d = 0; e = 0; f = 1; // valid and not vowel
        #10 a = 0; b = 1; c = 1; d = 0; e = 0; f = 1; // valid and not vowel
        #10 a = 1; b = 1; c = 0; d = 0; e = 0; f = 1; // valid and not vowel
        #10 a = 0; b = 0; c = 0; d = 0; e = 1; f = 1; // valid and not vowel
        #10 a = 1; b = 0; c = 1; d = 0; e = 1; f = 1; // valid and not vowel
        #10 a = 1; b = 1; c = 0; d = 0; e = 1; f = 1; // valid and not vowel
        #10 a = 1; b = 1; c = 1; d = 0; e = 1; f = 1; // valid and not vowel
        #10 a = 0; b = 0; c = 1; d = 1; e = 1; f = 0; // valid and not vowel
        #10 a = 0; b = 1; c = 0; d = 1; e = 1; f = 0; // valid and not vowel
        #10

        // invalids
        $display("Invalids");
        #10 a = 0; b = 0; c = 0; d = 0; e = 0; f = 1;
        #10 a = 1; b = 0; c = 1; d = 1; e = 1; f = 0;
        #10 a = 0; b = 1; c = 1; d = 0; e = 1; f = 1;
        #10 $finish;

    end

endmodule : Zorgian_test

module top();
    logic valid, vowel, a, b, c, d, e ,f;

    Zorgian     DUT(.a, .b, .c, .d, .e, .f, .valid, .vowel);

    Zorgian_test T(.a, .b, .c, .d, .e, .f, .valid, .vowel);

endmodule : top
```

Lab Code [10 points]
Filename: lab1SOP.sv
AndrewID: eqian

```
//`default_nettype none

module lab1SOP (
    input logic A, B, C, D, E, F,
    output logic valid, output logic vowel);

    logic nA, nB, nC, nD, nE, nF;
    logic na1, na2, na3, na4, na5, na6;
    logic na7, na8, na9, na10, na11;

    logic na12;
    logic na122;

    logic na13;
    logic na132;

    logic na14;
    logic na142;

    logic na15, na16, na17;
    logic na152, na162, na172;

    logic na18, na19, na20;
    logic na21, na22, na23;
    logic na212, na222, na232;

    // inverters
    not (nA, A);
    not (nB, B);
    not (nC, C);
    not (nD, D);
    not (nE, E);
    not (nF, F);

    // DE'
    nand (na1, D, nE);

    // AC'E'
    nand (na2, A, nC, nE);

    // BCD'F'
    nand (na3, B, C, nD, nF);

    // AD'EF
    nand (na4, A, nD, E, F);

    // A'CD'E'F
    nand (na12, nA, C, nD, nE);
    nand (na122, na12, na12);
    nand (na5, na122, F);

    // AB'D'E'
    nand (na6, A, nB, nD, nE);

    // B'C'D'F'
    nand (na7, nB, nC, nD, nF);

    // A'B'CDF'
    nand (na13, nA, nB, C, D);
    nand (na132, na13, na13);
    nand (na8, na132, nF);

    // B'C'D'E
    nand (na9, nB, nC, nD, E);

    // A'BC'EF'
    nand (na14, nA, B, nC, E);
    nand (na142, na14, na14);
    nand (na10, na142, nF);

    // B'C'E'F'
    nand (na11, nB, nC, nE, nF);

    // valid output
    nand (na15, na1, na2, na3, na4);
    nand (na16, na5, na6, na7, na8);
    nand (na17, na9, na10, na11);

    nand (na152, na15, na15);
    nand (na162, na16, na16);
    nand (na172, na17, na17);

    nand (valid, na152, na162, na172);

    // vowels
    // A'C'D'EF'
    nand (na21, nA, nC, nD, E);
    nand (na212, na21, na21);
    nand (na18, na212, nF);

    // BCD'EF'
    nand (na22, B, C, nD, E);
    nand (na222, na22, na22);
    nand (na19, na222, nF);

    // AB'D'E'F
    nand (na23, A, nB, nD, nE);
    nand (na232, na23, na23);
    nand (na20, na232, F);

    // vowel output
    nand (vowel, na18, na19, na20);

endmodule : lab1SOP

module lab1SOP_test
    (output logic A, B, C, D, E, F,
     input logic valid, vowel);

    initial begin
        $monitor( "A = %b, B = %b, C = %b, D = %b \n",
                   A, B, C, D,
                   "E = %b, F = %b, valid = %b, vowel = %b \n",
                   E, F, valid, vowel);

        // if D E F = 1 0 1, should always be valid but not vowel

            D = 1; E = 0; F = 1;

        #10 A = 0; B = 0; C = 0;

        #10 if (!(valid == 1 && vowel == 0))
                 $display("FAIL: def = 101, abc = 000");

        #10 A = 1; B = 1; C = 1;

        #10 if (!(valid == 1 && vowel == 0))
                $display("FAIL: def = 101, abc = 111");

        #10 A = 1; B = 0; C = 1;

        #10 if (!(valid == 1 && vowel == 0))
                $display("FAIL: def = 101, abc = 101");

        // if D E F = 0 1 0 and valid then its a vowel

        #10 A = 0; B = 0; C = 0; D = 0; E = 1; F = 0;

        #10 if (!(valid == 1 && vowel == 1))
                $display("FAIL: def = 010, abc = 000");

        #10 A = 0; B = 1; C = 1; // valid and vowel

        #10 if (!(valid == 1 && vowel == 1))
                $display("FAIL: def = 010, abc = 011");

        #10 A = 0; B = 1; C = 0; // valid and vowel
        #10

        #10 if (!(valid == 1 && vowel == 1))
                $display("FAIL: def = 010, abc = 010");

        // vowel u

        #10 A = 1; B = 0; C = 1; D = 0; E = 0; F = 1; // valid and vowel

        // rest of letters that are not in def 101 or 010
        #10 $display("Valids and not vowels below");
        #10 A = 0; B = 0; C = 0; D = 0; E = 0; F = 0; // valid and not vowel
        #10 A = 0; B = 1; C = 1; D = 0; E = 0; F = 0; // valid and not vowel
        #10 A = 1; B = 0; C = 1; D = 0; E = 0; F = 0; // valid and not vowel
        #10 A = 1; B = 1; C = 0; D = 0; E = 0; F = 0; // valid and not vowel
        #10 A = 1; B = 1; C = 1; D = 0; E = 0; F = 0; // valid and not vowel
        #10 A = 0; B = 0; C = 1; D = 0; E = 0; F = 1; // valid and not vowel
        #10 A = 0; B = 1; C = 1; D = 0; E = 0; F = 1; // valid and not vowel
        #10 A = 1; B = 1; C = 0; D = 0; E = 0; F = 1; // valid and not vowel
        #10 A = 0; B = 0; C = 0; D = 0; E = 1; F = 1; // valid and not vowel
        #10 A = 1; B = 0; C = 1; D = 0; E = 1; F = 1; // valid and not vowel
        #10 A = 1; B = 1; C = 0; D = 0; E = 1; F = 1; // valid and not vowel
        #10 A = 1; B = 1; C = 1; D = 0; E = 1; F = 1; // valid and not vowel
        #10 A = 0; B = 0; C = 1; D = 1; E = 1; F = 0; // valid and not vowel
        #10 A = 0; B = 1; C = 0; D = 1; E = 1; F = 0; // valid and not vowel
        #10

        // invalids
        $display("Invalids");
        #10 A = 0; B = 0; C = 0; D = 0; E = 0; F = 1;
        #10 A = 1; B = 0; C = 1; D = 1; E = 1; F = 0;
        #10 A = 0; B = 1; C = 1; D = 0; E = 1; F = 1;
        #10 $finish;

    end

endmodule : lab1SOP_test

module top();
    logic valid, vowel, A, B, C, D, E ,F;

    lab1SOP     DUT(.A, .B, .C, .D, .E, .F, .valid, .vowel);

    lab1SOP_test T(.A, .B, .C, .D, .E, .F, .valid, .vowel);

endmodule : top

```