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

```
1  `default_nettype none
2
3  /*
4  This module is the chipInterface that connects the Zorgian module's input
5  and output signals to FPGA pins. This module is necessary when synthesizing
6  the circuit into FPGA.
7  */
8  module ChipInterface
9      (output logic [17:0] LEDR,
10      input logic [5:0] SW);
11
12      Zorgian Z(.a(SW[5]), .b(SW[4]), .c(SW[3]), .d(SW[2]), .e(SW[1]), .f(SW[0]),
13              .valid(LEDR[17]), .vowel(LEDR[16]));
14
15 endmodule: ChipInterface
```

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

```
`default_nettype none
/*
Filename: lab1.sv
Author: Yutark Kim, Phil Du
AndrewID: yutarkk, yuchangd
This file contains module: valid_POS, vowel_POS, valid_SOP, vowel_SOP,
Test_SOP and Test_POS and Zorgian.

Zorgian sound can be represented as a 3-bit logic:
click-000, pop-001, hiss-010, shriek-011, whistle-101, bang-110, gargle-111.
Combination of two sounds will compose 6-bit input logic.

This file descirbes whether the letter from combinations of two sounds is
valid or vowel.
If it is a vowel letter, valid=1, vowel=1.
If it is non-vowel letter: valid=1, vowel=0.
If it is not valid, valid=0,vowel=0.
*/


/*
This module describes valid output signals using only using NOR and NOT Gates.
For non-input values, 1-input NOR gates are used to replace the NOT Gates.
Total 26 NOR gates were used.
*/
module valid_POS
    (input logic a, b, c, d, e, f,
    output logic valid);
    logic a_not, b_not, c_not, d_not, e_not, f_not;
    logic v1, v2, v3, v4, v5, v6, v7, v7_not, v7_final, v8, v8_not,
            v8_final, v9, v9_not, v9_final, v10, v10_not,v10_final, v11,
            v1to4, v5to8, v9to11, v1to4_not, v5to8_not, v9to11_not, v_not;

    not n1(a_not, a),
        n2(b_not, b),
        n3(c_not, c),
        n4(d_not, d),
        n5(e_not, e),
        n6(f_not, f);
    nor no1(v1, a_not, c, e_not, f),
        no2(v2, a, b_not, e_not, f_not),
        no3(v3, a, c_not, e_not, f_not),
        no4(v4, b, c, d_not, e_not),
        no5(v5, a_not, b, e_not, f),
        no6(v6, a_not, d_not, e_not),
        no7(v7, a, b_not, c, d),
        no8(v7_not, v7),
        no9(v7_final, v7_not, e),
        no10(v8, a, c, d, e),
        no11(v8_not, v8),
        no12(v8_final, v8_not, f_not),
        no13(v9, a, b, c_not, d),
        no14(v9_not, v9),
        no15(v9_final, v9_not, f),
        no16(v10, a_not, b_not, c_not, d),
        no17(v10_not, v10),
        no18(v10_final, v10_not, e, f_not),
        no19(v11, b_not, c_not, d_not, e),
        no20(v1to4, v1, v2, v3, v4),
        no21(v5to8, v5, v6, v7_final, v8_final),
        no22(v9to11, v9_final, v10_final, v11),
        no23(v1to4_not, v1to4),
        no24(v5to8_not, v5to8),
        no25(v9to11_not, v9to11),
        no26(valid, v1to4_not, v5to8_not, v9to11_not);

endmodule: valid_POS

/*
```

```
This module describes vowel output signals using only using NOR and NOT Gates.
For non-input values, 1-input NOR gates are used to replace the NOT Gates.
Total 8 NOR gates were used.
*/
module vowel_POS
    (input logic a, b, c, d, e, f,
    output logic vowel);

    logic a_not, b_not, c_not, d_not, e_not, f_not;
    logic v1, v2, v3, v4, v5, v6, v6_not;

    not n1(a_not, a),
        n2(b_not, b),
        n3(c_not, c),
        n4(d_not, d),
        n5(e_not, e),
        n6(f_not, f);
    nor no1(v1, b_not, f_not),
        no2(v2, e, f),
        no3(v3, a, f_not),
        no4(v4, a_not, c),
        no5(v5, b, c_not, e_not),
        no6(v6, d, v1, v2, v3),
        no7(v6_not, v6),
        no8(vowel, v6_not, v4, v5);

endmodule: vowel_POS

/*
This module contains selective test cases to test
valid_POS and vowel_POS module's behavior.
*/
module test_POS ();
   logic a, b, c, d, e, f, valid, vowel;
   valid_POS DUT1(.*);
   vowel_POS DUT2(.*);

   initial begin
     $monitor($time,, "Input: %d%d%d%d%d%d valid: %d vowel: %d",
              a,b,c,d,e,f,valid,vowel);
     {a,b,c,d,e,f} = 6'b000000;
     //cases of valid is 1 and vowel is 1
     #10 {a,b,c,d,e,f} = 6'b000010;
     $display("Expecting valid=1, vowel=1");
     #10 {a,b,c,d,e,f} = 6'b011010;
     #10 {a,b,c,d,e,f} = 6'b101001;
     #10 {a,b,c,d,e,f} = 6'b111010;
     //cases of valid is 1 and vowel is 0
     #10 {a,b,c,d,e,f} = 6'b000011;
     $display("Expecting valid=1, vowel=0");
     #10 {a,b,c,d,e,f} = 6'b001110;
     #10 {a,b,c,d,e,f} = 6'b101101;
     #10 {a,b,c,d,e,f} = 6'b111011;

     //cases of valid is 0
     #10 {a,b,c,d,e,f} = 6'b000001;
     $display("Expecting valid=0, vowel=0");
     #10 {a,b,c,d,e,f} = 6'b001000;
     #10 {a,b,c,d,e,f} = 6'b101010;

     #10 $display("Test Cases for CheckOff");
     #10 {a,b,c,d,e,f} = 6'b010010;
     #10 {a,b,c,d,e,f} = 6'b011011;
     #10 {a,b,c,d,e,f} = 6'b110101;
     #10 {a,b,c,d,e,f} = 6'b111111;
     #10 $finish;
   end
endmodule: test_POS

/*
This module describes valid output signals using only using NOR and NOT Gates.
```

```
141 For non-input values, 1-input NAND gates are used to replace the NOT Gates.
142 Total 24 NAND gates were used.
143 */
144 module valid_SOP
145     (input logic a, b, c, d, e, f,
146     output logic valid);
147     logic a_not, b_not, c_not, d_not, e_not, f_not;
148     logic v1, v2, v3, v3_not, v4, v5, v6, v7, v7_not, v8, v9, v10, v10_not,
149             v11, v12, v13, v14, v15, v15_not, v16, v16_not, v17, v17_not;
150 
151     not n1(a_not, a),
152         n2(b_not, b),
153         n3(c_not, c),
154         n4(d_not, d),
155         n5(e_not, e),
156         n6(f_not, f);
157     nand na1(v1, b, c, d_not, f_not),
158          na2(v2, d, e_not),
159          na3(v3, a_not, b_not, c, d),
160          na4(v3_not, v3),
161          na5(v4, v3_not, f_not),
162          na6(v5, a, b_not, e_not),
163          na7(v6, a_not, c, e_not, f),
164          na8(v7, a, c, d_not, e),
165          na9(v7_not, v7),
166          na10(v8, v7_not, f),
167          na11(v9, b_not, c_not, d_not, e),
168          na12(v10, a_not, b, c_not, e),
169          na13(v10_not, v10),
170          na14(v11, v10_not, f_not),
171          na15(v12, a, c_not, e_not),
172          na16(v13, b_not, c_not, e_not, f_not),
173          na17(v14, a, c_not, d_not, f),
174          na18(v15, v1, v2, v4, v5),
175          na19(v15_not, v15),
176          na20(v16, v6, v8, v9, v11),
177          na21(v16_not, v16),
178          na22(v17, v12, v13, v14),
179          na23(v17_not, v17),
180          na24T(valid, v15_not, v16_not, v17_not);
181 
182 endmodule: valid_SOP
183 
184 /*
185 This module describes valid output signals using only using NAND and NOT Gates.
186 For non-input values, 1-input NAND gates are used to replace the NOT Gates.
187 Total 10 NAND gates were used.
188 */
189 module vowel_SOP
190     (input logic a, b, c, d, e, f,
191     output logic vowel);
192 
193     logic a_not, b_not, c_not, d_not, e_not, f_not;
194     logic v1, v1_not, v2, v5, v5_not, v6, v7, v7_not, v8;
195 
196     not n1(a_not, a),
197         n2(b_not, b),
198         n3(c_not, c),
199         n4(d_not, d),
200         n5(e_not, e),
201         n6(f_not, f);
202 
203     nand na1(v1, a_not, c_not, d_not, e),
204          na2(v1_not, v1),
205          na3(v2, v1_not, f_not),
206          na4(v5, b, c, d_not, e),
207          na5(v5_not, v5),
208          na6(v6, v5_not, f_not),
209          na7(v7, a, b_not, d_not, e_not),
210          na8(v7_not, v7),
211          na9(v8, v7_not, f),
```

```
212          na10(vowel, v2, v6, v8);
213
214 endmodule: vowel_SOP
215
216 /*
217 This module contains selective test cases to test
218 valid_SOP and vowel_SOP module's behavior.
219 */
220 module test_SOP();
221    logic a, b, c, d, e, f, valid, vowel;
222    valid_POS DUT1(.*);
223    vowel_POS DUT2(.*);
224
225    initial begin
226      $monitor($time,, "Input: %d%d%d%d%d%d valid: %d vowel: %d",
227               a,b,c,d,e,f,valid,vowel);
228      {a,b,c,d,e,f} = 6'b000000;
229      //cases of valid is 1 and vowel is 1
230      #10 {a,b,c,d,e,f} = 6'b000010;
231      $display("Expecting valid=1, vowel=1");
232      #10 {a,b,c,d,e,f} = 6'b011010;
233      #10 {a,b,c,d,e,f} = 6'b101001;
234      #10 {a,b,c,d,e,f} = 6'b111010;
235      //cases of valid is 1 and vowel is 0
236      #10 {a,b,c,d,e,f} = 6'b000011;
237      $display("Expecting valid=1, vowel=0");
238      #10 {a,b,c,d,e,f} = 6'b001110;
239      #10 {a,b,c,d,e,f} = 6'b101101;
240      #10 {a,b,c,d,e,f} = 6'b111011;
241
242      //cases of valid is 0
243      #10 {a,b,c,d,e,f} = 6'b000001;
244      $display("Expecting valid=0, vowel=0");
245      #10 {a,b,c,d,e,f} = 6'b001000;
246      #10 {a,b,c,d,e,f} = 6'b101010;
247
248      #10 $display("Test Cases for CheckOff");
249      #10 {a,b,c,d,e,f} = 6'b010010;
250      #10 {a,b,c,d,e,f} = 6'b011011;
251      #10 {a,b,c,d,e,f} = 6'b110101;
252      #10 {a,b,c,d,e,f} = 6'b111111;
253      #10 $finish;
254    end
255 endmodule: test_SOP
256
257 /*
258 This module is the module that connects to valid_POS and vowel_POS
259 and to ChipInterface module in chipInterface.sv.
260 */
261 module Zorgian(input logic a, b, c, d, e, f,
262                output logic valid, vowel);
263
264     valid_POS DUT1(.*);
265     vowel_POS DUT2(.*);
266 endmodule: Zorgian
```