```
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

```
1  //`default_nettype none
2
3  module Zorgian
4    (input logic a, b, c, d, e, f,
5     output logic valid, vowel);
6
7      logic a_not, b_not, c_not, d_not, e_not, f_not;
8      logic nt1, nt2, nt2a, nt2b, nt3a, nt3b, nt3,
9          nt4, nt4a, nt4b, nt4c, nt4d, nt5, nt5a, nt5b, nt6, nt7,
10         nt8, nt9, nt10, nt11, nt12,
11         nt13, nt14, nt15, nt16,
12         nt17, nt18;
13     logic at1, at2, at3, at4, at5;
14
15     logic ft1, ft2, ft3;
16     logic ft4, ft5;
17
18     // inverters
19     not (a_not, a),
20         (b_not, b),
21         (c_not, c),
22         (d_not, d),
23         (e_not, e),
24         (f_not, f);
25
26     // (d' + e' + f')
27     nor (nt1, d_not, e_not, f_not);
28
29     // (a + b + c' + d + f)
30     nor (nt2a, a, b, c_not, d),
31         (nt2b, nt2a, nt2a),
32         (nt2, f, nt2b);
33
34     // (a + c + d + e + f')
35     nor (nt3a, a, c, d, e),
36         (nt3b, nt3a, nt3a), //0
37         (nt3, f_not, nt3b); //
38
39     // (a' + b' + c' + d + e + f')
40     nor (nt4a, a_not, b_not, c_not),
41         (nt4b, nt4a, nt4a),
42         (nt4c, d, e, f_not),
43         (nt4d, nt4c, nt4c),
44         (nt4, nt4d, nt4b);
45
46     // (a+b'+c+e+f)
47     nor (nt5a, a, b_not, c, e),
48         (nt5b, nt5a, nt5a),
49         (nt5, f, nt5b);
50
51     // (b' + c' + d' + e')
52     nor (nt6, b_not, c_not, d_not, e_not);
53
54     // (a + c' + e' + f')
55     nor (nt7, a, c_not, e_not, f_not);
56
57     // (a' + c + e' + f)
58     nor (nt8, a_not, c, e_not, f);
59
60     // (a' + b + e' + f)
61     nor (nt9, a_not, b, e_not, f);
62
63     // (b + c + d' + e')
64     nor (nt10, b, c, d_not, e_not);
65
66     // (a + b' + e' + f')
67     nor (nt11, a, b_not, e_not, f_not);
68
69     nor (at1, nt1, nt2, nt3, nt4),
```

```
70          (at2, nt5, nt6, nt7, nt8),
71          (at3, nt9, nt10, nt11);
72
73      // valid output
74      nor (ft1, at1, at1);
75      nor (ft2, at2, at2);
76      nor (ft3, at3, at3);
77      nor (valid, ft1, ft2, ft3);
78
79      // (e + f)
80      nor (nt12, e, f);
81
82      // d' + e' + f
83      nor (nt13, d_not, e_not, f);
84
85      // b + c' + d + e'
86      nor (nt14, b, c_not, d, e_not);
87
88      // a + f'
89      nor (nt15, a, f_not);
90
91      // a' + d' + f'
92      nor (nt16, a_not, d_not,f_not);
93
94      // b' + c' + d + f'
95      nor (nt17, b_not, c_not, d, f_not);
96
97      // a' + c + d
98      nor (nt18, a_not, c, d);
99
100     // vowel output
101     nor (at4, nt12, nt13, nt14),
102         (at5, nt15, nt16, nt17, nt18);
103     nor (ft4, at4, at4);
104     nor (ft5, at5, at5);
105     nor (vowel, ft4, ft5);
106
107 endmodule : Zorgian
108
109 module Zorgian_test
110     (output logic a, b, c, d, e, f,
111      input logic valid, vowel);
112
113     initial begin
114         $monitor( "a = %b, b = %b, c = %b, d = %b \n",
115                   a, b, c, d,
116                   "e = %b, f = %b, valid = %b, vowel = %b \n",
117                   e, f, valid, vowel);
118
119         // if d e f = 1 0 1, should always be valid but not vowel
120
121              d = 1; e = 0; f = 1;
122
123         #10 a = 0; b = 0; c = 0;
124
125         #10 if (!(valid == 1 && vowel == 0))
126                  $display("FAIL: def = 101, abc = 000");
127
128         #10 a = 1; b = 1; c = 1;
129
130         #10 if (!(valid == 1 && vowel == 0))
131                 $display("FAIL: def = 101, abc = 111");
132
133         #10 a = 1; b = 0; c = 1;
134
135         #10 if (!(valid == 1 && vowel == 0))
136                 $display("FAIL: def = 101, abc = 101");
137
138         // if d e f = 0 1 0 and valid then its a vowel
139
140         #10 a = 0; b = 0; c = 0; d = 0; e = 1; f = 0;
```

```
141
142         #10 if (!(valid == 1 && vowel == 1))
143                 $display("FAIL: def = 010, abc = 000");
144
145         #10 a = 0; b = 1; c = 1; // valid and vowel
146
147         #10 if (!(valid == 1 && vowel == 1))
148                 $display("FAIL: def = 010, abc = 011");
149
150         #10 a = 0; b = 1; c = 0; // valid and vowel
151
152         #10 if (!(valid == 1 && vowel == 1))
153                 $display("FAIL: def = 010, abc = 010");
154
155         // vowel u
156
157         #10 a = 1; b = 0; c = 1; d = 0; e = 0; f = 1; // valid and vowel
158
159         // rest of letters that are not in def 101 or 010
160         #10 $display("Valids and not vowels below");
161         #10 a = 0; b = 0; c = 0; d = 0; e = 0; f = 0; // valid and not vowel
162         #10 a = 0; b = 1; c = 1; d = 0; e = 0; f = 0; // valid and not vowel
163         #10 a = 1; b = 0; c = 1; d = 0; e = 0; f = 0; // valid and not vowel
164         #10 a = 1; b = 1; c = 0; d = 0; e = 0; f = 0; // valid and not vowel
165         #10 a = 1; b = 1; c = 1; d = 0; e = 0; f = 0; // valid and not vowel
166         #10 a = 0; b = 0; c = 1; d = 0; e = 0; f = 1; // valid and not vowel
167         #10 a = 0; b = 1; c = 1; d = 0; e = 0; f = 1; // valid and not vowel
168         #10 a = 1; b = 1; c = 0; d = 0; e = 0; f = 1; // valid and not vowel
169         #10 a = 0; b = 0; c = 0; d = 0; e = 1; f = 1; // valid and not vowel
170         #10 a = 1; b = 0; c = 1; d = 0; e = 1; f = 1; // valid and not vowel
171         #10 a = 1; b = 1; c = 0; d = 0; e = 1; f = 1; // valid and not vowel
172         #10 a = 1; b = 1; c = 1; d = 0; e = 1; f = 1; // valid and not vowel
173         #10 a = 0; b = 0; c = 1; d = 1; e = 1; f = 0; // valid and not vowel
174         #10 a = 0; b = 1; c = 0; d = 1; e = 1; f = 0; // valid and not vowel
175         #10
176
177         // invalids
178         $display("Invalids");
179         #10 a = 0; b = 0; c = 0; d = 0; e = 0; f = 1;
180         #10 a = 1; b = 0; c = 1; d = 1; e = 1; f = 0;
181         #10 a = 0; b = 1; c = 1; d = 0; e = 1; f = 1;
182         #10 $finish;
183
184     end
185
186 endmodule : Zorgian_test
187
188 module top();
189     logic valid, vowel, a, b, c, d, e ,f;
190
191     Zorgian     DUT(.a, .b, .c, .d, .e, .f, .valid, .vowel);
192
193     Zorgian_test T(.a, .b, .c, .d, .e, .f, .valid, .vowel);
194
195 endmodule : top
```

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

```
1  //`default_nettype none
2
3  module lab1SOP (
4      input logic A, B, C, D, E, F,
5      output logic valid, output logic vowel);
6
7      logic nA, nB, nC, nD, nE, nF;
8      logic na1, na2, na3, na4, na5, na6;
9      logic na7, na8, na9, na10, na11;
10
11     logic na12;
12     logic na122;
13
14     logic na13;
15     logic na132;
16
17     logic na14;
18     logic na142;
19
20     logic na15, na16, na17;
21     logic na152, na162, na172;
22
23     logic na18, na19, na20;
24     logic na21, na22, na23;
25     logic na212, na222, na232;
26
27     // inverters
28     not (nA, A);
29     not (nB, B);
30     not (nC, C);
31     not (nD, D);
32     not (nE, E);
33     not (nF, F);
34
35     // DE'
36     nand (na1, D, nE);
37
38     // AC'E'
39     nand (na2, A, nC, nE);
40
41     // BCD'F'
42     nand (na3, B, C, nD, nF);
43
44     // AD'EF
45     nand (na4, A, nD, E, F);
46
47     // A'CD'E'F
48     nand (na12, nA, C, nD, nE);
49     nand (na122, na12, na12);
50     nand (na5, na122, F);
51
52     // AB'D'E'
53     nand (na6, A, nB, nD, nE);
54
55     // B'C'D'F'
56     nand (na7, nB, nC, nD, nF);
57
58     // A'B'CDF'
59     nand (na13, nA, nB, C, D);
60     nand (na132, na13, na13);
61     nand (na8, na132, nF);
62
63     // B'C'D'E
64     nand (na9, nB, nC, nD, E);
65
66     // A'BC'EF'
67     nand (na14, nA, B, nC, E);
68     nand (na142, na14, na14);
69     nand (na10, na142, nF);
```

```
70
71      // B'C'E'F'
72      nand (na11, nB, nC, nE, nF);
73
74      // valid output
75      nand (na15, na1, na2, na3, na4);
76      nand (na16, na5, na6, na7, na8);
77      nand (na17, na9, na10, na11);
78
79      nand (na152, na15, na15);
80      nand (na162, na16, na16);
81      nand (na172, na17, na17);
82
83      nand (valid, na152, na162, na172);
84
85      // vowels
86      // A'C'D'EF'
87      nand (na21, nA, nC, nD, E);
88      nand (na212, na21, na21);
89      nand (na18, na212, nF);
90
91      // BCD'EF'
92      nand (na22, B, C, nD, E);
93      nand (na222, na22, na22);
94      nand (na19, na222, nF);
95
96      // AB'D'E'F
97      nand (na23, A, nB, nD, nE);
98      nand (na232, na23, na23);
99      nand (na20, na232, F);
100
101     // vowel output
102     nand (vowel, na18, na19, na20);
103
104 endmodule : lab1SOP
105
106 module lab1SOP_test
107     (output logic A, B, C, D, E, F,
108      input logic valid, vowel);
109
110     initial begin
111         $monitor( "A = %b, B = %b, C = %b, D = %b \n",
112                   A, B, C, D,
113                   "E = %b, F = %b, valid = %b, vowel = %b \n",
114                   E, F, valid, vowel);
115
116         // if D E F = 1 0 1, should always be valid but not vowel
117
118             D = 1; E = 0; F = 1;
119
120         #10 A = 0; B = 0; C = 0;
121
122         #10 if (!(valid == 1 && vowel == 0))
123                  $display("FAIL: def = 101, abc = 000");
124
125         #10 A = 1; B = 1; C = 1;
126
127         #10 if (!(valid == 1 && vowel == 0))
128                 $display("FAIL: def = 101, abc = 111");
129
130         #10 A = 1; B = 0; C = 1;
131
132         #10 if (!(valid == 1 && vowel == 0))
133                 $display("FAIL: def = 101, abc = 101");
134
135         // if D E F = 0 1 0 and valid then its a vowel
136
137         #10 A = 0; B = 0; C = 0; D = 0; E = 1; F = 0;
138
139         #10 if (!(valid == 1 && vowel == 1))
140                 $display("FAIL: def = 010, abc = 000");
```

```
141
142         #10 A = 0; B = 1; C = 1; // valid and vowel
143
144         #10 if (!(valid == 1 && vowel == 1))
145                 $display("FAIL: def = 010, abc = 011");
146
147         #10 A = 0; B = 1; C = 0; // valid and vowel
148         #10
149
150         #10 if (!(valid == 1 && vowel == 1))
151                 $display("FAIL: def = 010, abc = 010");
152
153         // vowel u
154
155         #10 A = 1; B = 0; C = 1; D = 0; E = 0; F = 1; // valid and vowel
156
157         // rest of letters that are not in def 101 or 010
158         #10 $display("Valids and not vowels below");
159         #10 A = 0; B = 0; C = 0; D = 0; E = 0; F = 0; // valid and not vowel
160         #10 A = 0; B = 1; C = 1; D = 0; E = 0; F = 0; // valid and not vowel
161         #10 A = 1; B = 0; C = 1; D = 0; E = 0; F = 0; // valid and not vowel
162         #10 A = 1; B = 1; C = 0; D = 0; E = 0; F = 0; // valid and not vowel
163         #10 A = 1; B = 1; C = 1; D = 0; E = 0; F = 0; // valid and not vowel
164         #10 A = 0; B = 0; C = 1; D = 0; E = 0; F = 1; // valid and not vowel
165         #10 A = 0; B = 1; C = 1; D = 0; E = 0; F = 1; // valid and not vowel
166         #10 A = 1; B = 1; C = 0; D = 0; E = 0; F = 1; // valid and not vowel
167         #10 A = 0; B = 0; C = 0; D = 0; E = 1; F = 1; // valid and not vowel
168         #10 A = 1; B = 0; C = 1; D = 0; E = 1; F = 1; // valid and not vowel
169         #10 A = 1; B = 1; C = 0; D = 0; E = 1; F = 1; // valid and not vowel
170         #10 A = 1; B = 1; C = 1; D = 0; E = 1; F = 1; // valid and not vowel
171         #10 A = 0; B = 0; C = 1; D = 1; E = 1; F = 0; // valid and not vowel
172         #10 A = 0; B = 1; C = 0; D = 1; E = 1; F = 0; // valid and not vowel
173         #10
174
175         // invalids
176         $display("Invalids");
177         #10 A = 0; B = 0; C = 0; D = 0; E = 0; F = 1;
178         #10 A = 1; B = 0; C = 1; D = 1; E = 1; F = 0;
179         #10 A = 0; B = 1; C = 1; D = 0; E = 1; F = 1;
180         #10 $finish;
181
182     end
183
184 endmodule : lab1SOP_test
185
186 module top();
187     logic valid, vowel, A, B, C, D, E ,F;
188
189     lab1SOP     DUT(.A, .B, .C, .D, .E, .F, .valid, .vowel);
190
191     lab1SOP_test T(.A, .B, .C, .D, .E, .F, .valid, .vowel);
192
193 endmodule : top
194
```