

Problem 3: [2 points] Drill problem
Filename: library.sv
AndrewID: xinyew

```
1 `default_nettype none
2 module Decoder
3   (input logic en,
4    input logic [2:0] I,
5    output logic [7:0] D);
6
7   always_comb begin
8     unique case (I)
9       3'b000: D = 8'b00000001;
10      3'b001: D = 8'b00000010;
11      3'b010: D = 8'b00000100;
12      3'b011: D = 8'b00001000;
13      3'b100: D = 8'b00010000;
14      3'b101: D = 8'b00100000;
15      3'b110: D = 8'b01000000;
16      3'b111: D = 8'b10000000;
17    endcase
18    D = en ? D : 0;
19  end
20
21 endmodule : Decoder
22
23 module BarrelShifter
24   (input logic [15:0] v,
25    input logic [3:0] by,
26    output logic [15:0] s);
27
28   always_comb begin
29     s = by[3] ? v << 8 : v;
30     s = by[2] ? s << 4 : s;
31     s = by[1] ? s << 2 : s;
32     s = by[0] ? s << 1 : s;
33   end
34
35 endmodule : BarrelShifter
36
37 module Multiplexer
38   (input logic [7:0] I,
39    input logic [2:0] s,
40    output logic Y);
41
42   always_comb
43     case (s)
44       3'b000: Y = I[0];
45       3'b001: Y = I[1];
46       3'b010: Y = I[2];
47       3'b011: Y = I[3];
48       3'b100: Y = I[4];
49       3'b101: Y = I[5];
50       3'b110: Y = I[6];
51       3'b111: Y = I[7];
52     endcase
53
54 endmodule: Multiplexer
55
56 module Mux2to1
57   (input logic [6:0] I0,
58    input logic [6:0] I1,
59    input logic S,
60    output logic [6:0] Y);
61
62   always_comb
63     case (S)
64       0: Y = I0;
65       1: Y = I1;
66     endcase
67
68 endmodule : Mux2to1
69
```

```
70 module MagComparator
71   (input logic [7:0] A,
72    input logic [7:0] B,
73    output logic AltB,
74    output logic AeqB,
75    output logic AgtB);
76
77   assign AltB = A < B;
78   assign AgtB = A > B;
79   assign AeqB = A == B;
80
81 endmodule : MagComparator
82
83 module Comparator
84   (input logic [3:0] A,
85    input logic [3:0] B,
86    output logic AeqB);
87
88   assign AeqB = A == B;
89
90 endmodule : Comparator
91
92 module Adder
93   (input logic [4:0] A,
94    input logic [4:0] B,
95    input logic cin,
96    output logic cout,
97    output logic [4:0] sum);
98
99   assign {cout, sum} = A + B + cin;
100 endmodule : Adder
```

Problem 3: [2 points] Drill problem
Filename: library_tests.sv
AndrewID: xinyew

```
1 `default_nettype none
2 module Decoder_test;
3     logic en;
4     logic [2:0] I;
5     logic [7:0] D;
6
7     Decoder DUT(.en(en),
8                 .I(I),
9                 .D(D));
10
11     initial begin
12         $monitor($time,,
13                 "en = %b, I = %b, D = %b", en, I, D);
14         en = 1;
15         for (I = 3'b000; I < 3'b111; I++)
16             #1;
17         en = 0;
18         for (I = 3'b000; I < 3'b111; I++)
19             #1;
20         #1 $finish;
21     end
22
23 endmodule: Decoder_test
24
25 module BarrelShifter_test;
26     logic [15:0] v;
27     logic [3:0] by;
28     logic [15:0] s;
29
30     BarrelShifter DUT(.v(v),
31                       .by(by),
32                       .s(s));
33
34     initial begin
35         $monitor($time,,
36                 "v = %b, by = %b, s = %b", v, by, s);
37         v = 16'b0101010101010101;
38         for (by = 4'b0000; by < 4'b1111; by++)
39             #1;
40         #1 $finish;
41     end
42
43 endmodule : BarrelShifter_test
44
45 module Multiplexer_test;
46     logic [7:0] I;
47     logic [2:0] s;
48     logic Y;
49
50     Multiplexer DUT(.I(I),
51                     .s(s),
52                     .Y(Y));
53
54     initial begin
55         $monitor($time,,
56                 "I = %b, s = %b, Y = %b", I, s, Y);
57         I = 8'b01010101;
58         for (s = 3'b000; s < 3'b111; s++)
59             #1;
60         #1 $finish;
61     end
62
63 endmodule : Multiplexer_test
64
65 module Mux2to1_test;
66     logic [6:0] I0;
67     logic [6:0] I1;
68     logic S;
69     logic [6:0] Y;
```

```

70
71     Mux2to1 DUT(.I0(I0),
72                 .I1(I1),
73                 .S(S),
74                 .Y(Y));
75
76     initial begin
77         $monitor($time,,
78                 "I0 = %b, I1 = %b, S = %b, Y = %b", I0, I1, S, Y);
79         I0 = 7'b00000000;
80         I1 = 7'b11111111;
81         S = 0;
82         #1 S = 1;
83         #1 $finish;
84     end
85
86 endmodule : Mux2to1_test
87
88 module MagComparator_test;
89     logic [7:0] A;
90     logic [7:0] B;
91     logic AltB;
92     logic AgtB;
93     logic AeqB;
94
95     MagComparator DUT (.A(A),
96                       .B(B),
97                       .AltB(AltB),
98                       .AgtB(AgtB),
99                       .AeqB(AeqB));
100
101     initial begin
102         $monitor($time,,
103                 "A = %b, B = %b, AltB = %b, AgtB = %b, AeqB = %b", A, B , AltB, AgtB, AeqB);
104         A = 8'b00000000;
105         B = 8'b11111111;
106         #2 A = 8'b11111111;
107         #2 B = 8'b00000000;
108         #2 $finish;
109     end
110
111 endmodule : MagComparator_test
112
113 module Comparator_test;
114     logic [3:0] A;
115     logic [3:0] B;
116     logic AeqB;
117
118     MagComparator DUT (.A(A),
119                       .B(B),
120                       .AeqB(AeqB));
121
122     initial begin
123         $monitor($time,,
124                 "A = %b, B = %b, AeqB = %b", A, B, AeqB);
125         A = 8'b00000000;
126         B = 8'b11111111;
127         #2 A = 8'b11111111;
128         #2 $finish;
129     end
130
131 endmodule : Comparator_test
132
133 module Adder_test;
134     logic [4:0] A;
135     logic [4:0] B;
136     logic [4:0] sum;
137     logic cin;
138     logic cout;
139
140     Adder DUT (.A(A),

```

```
141         .B(B),
142         .cin(cin),
143         .cout(cout),
144         .sum(sum));
145
146     initial begin
147         $monitor($time,,
148             "A = %b, B = %b, cin = %b, sum = %b, cout = %b",
149             A, B, cin, sum, cout);
150         A = 5'b000000;
151         B = 5'b000000;
152         cin = 1'b0;
153         #2 cin = 1'b1;
154         #2 A = 5'b000001;
155         #2 B = 5'b000001;
156         #2 A = 5'b111111;
157         cin = 1'b0;
158         #2 cin = 1'b1;
159         #2 $finish;
160     end
161
162 endmodule : Adder_test
```

Problem 10: [10 points]
Filename: hw3prob10.sv
AndrewID: xinyew

```
1 `default_nettype none
2 module hw3prob10
3     (input logic [7:0] a, b,
4      output logic [7:0] sum,
5      output logic overflow);
6
7     logic [6:0] mag_a, mag_b , max, min;
8     logic [6:0] mag_sum;
9     logic sign_a, sign_b, sign_sum;
10
11    assign mag_a = a[6:0] ;
12    assign mag_b = b[6:0] ;
13    assign sign_a = a[7];
14    assign sign_b = b[7] ;
15
16    always_comb begin
17        if (mag_a > mag_b) begin
18            max = mag_a;
19            min = mag_b;
20            sign_sum = sign_a;
21        end
22        else begin
23            max = mag_b;
24            min = mag_a;
25            sign_sum = sign_b;
26        end
27
28        if (sign_a==sign_b) begin
29            {overflow, mag_sum} = max + min;
30        end
31        else begin
32            mag_sum = max - min;
33            overflow = 0;
34        end
35
36        sum = {sign_sum, mag_sum};
37    end
38 endmodule : hw3prob10
39
40 module hw3prob10_test;
41     logic [7:0] a, b, sum;
42     logic overflow;
43
44     hw3prob10 DUT (.a(a),
45                   .b(b),
46                   .sum(sum),
47                   .overflow(overflow));
48
49     initial begin
50         $monitor($time,,
51                 "a = %b, b = %b, sum = %b, overflow = %b",
52                 a, b, sum, overflow);
53         a = 8'b00000000;
54         b = 8'b11111111;
55         #1 a = 8'b00000001;
56         #1 a = 8'b10000001;
57         #1 a = 8'b11111111;
58         #1 a = 8'b01111111;
59         b = 8'b10000000;
60         #1 b = 8'b10000001;
61         #1 b = 8'b00000001;
62         #1 b = 8'b11111111;
63         #1 b = 8'b01111111;
64         #1 $finish;
65     end
66 endmodule : hw3prob10_test
```