```
Problem 3: [2 points] Drill problem
Filename: library.sv
AndrewID: xinyew
     `default_nettype none
  2
    module Decoder
       (input logic en,
        input logic [2:0] I, output logic [7:0] D);
  5
  6
  7
       always_comb begin
  8
         unique case (I)
  9
            3'b000: D = 8'b000000001;
            3'b001: D = 8'b00000010;
 10
            3'b010: D = 8'b00000100
 11
            3'b011: D = 8'b00001000;
 12
 13
            3'b100: D = 8'b00010000;
            3'b101: D = 8'b001000000;
 14
 15
            3'b110: D = 8'b010000000;
            3'b111: D = 8'b100000000;
 16
         endcase
 17
 18
         D = en ? D : 0;
 19
 20
 21 endmodule : Decoder
 22
 23 module BarrelShifter
 24
       (input logic [15:0] v,
 25
        input logic [3:0] by,
output logic [15:0] s);
 26
 27
       always_comb begin

s = by[3] ? v << 8 : v;

s = by[2] ? s << 4 : s;
 28
 29
 30
         s = by[1] ? s << 2 : s;
s = by[0] ? s << 1 : s;
 31
 32
 33
 34
 35 endmodule : BarrelShifter
 36
 37 module Multiplexer
       (input logic [7:0] I,
input logic [2:0] s,
 38
 39
        output logic Y);
 40
 41
 42
       always_comb
 43
         case (s)
 44
            3'b000: Y = I[0];
            3'b001: Y = I[1];
 45
            3'b010: Y = I[2];
3'b011: Y = I[3];
 46
 47
            3'b100: Y = I[4];
 48
            3'b101: Y = I[5];
 49
            3'b110: Y = I[6]
 50
 51
            3'b111: Y = I[7];
 52
         endcase
 53
 54 endmodule: Multiplexer
 55
 56 module Mux2to1
 57
       (input logic [6:0] IO,
        input logic [6:0] I1, input logic S,
 58
 59
        output logic [6:0] Y);
 60
 61
 62
       always_comb
 63
         case (S)
            0: Y = I0;
1: Y = I1;
 64
 65
 66
         endcase
 67
 68 endmodule : Mux2to1
```

Filename: library.sv

```
Page #: 2
```

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

```
Problem 3: [2 points] Drill problem
Filename: library_tests.sv
AndrewID: xinyew
    `default_nettype none
  2 module Decoder_test;
       logic en;
       logic [2:0] I;
logic [7:0]D;
  5
  6
  7
       Decoder DUT(.en(en),
  8
                      .I(I)
  9
                      .D(D));
 10
       initial begin
 11
         $monitor($time,,
   "en = %b, I = %b, D = %b", en, I, D);
 12
 13
 14
 15
         for (I = 3'b000; I < 3'b111; I++)
           #1;
 16
         en = 0;
 17
 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;
      logic [15:0] v;
logic [3:0] by;
logic [15:0] s;
 26
 27
 28
 29
 30
       BarrelShifter DUT(.v(v),
 31
                             .by(by),
 32
                             .s(s));
 33
 34
       initial begin
 35
         $monitor($time,,
   "v = %b, by = %b, s = %b", v, by, s);
 36
         v = 16'b010101010101010101;
 37
         for (by = 4'b0000; by < 4'b1111; by++)
 38
 39
           #1:
 40
         #1 $finish;
 41
       end
 42
 43 endmodule : BarrelShifter_test
 44
 45 module Multiplexer_test;
       logic [7:0] I;
logic [2:0] s;
 46
 47
       logic Ÿ;
 48
 49
 50
       Multiplexer DUT(.I(I),
 51
                          .s(s),
 52
                           .Y(Y));
 53
 54
       initial begin
         $monitor($time,,
"I = %b, s = %b, Y = %b", I, s, Y);
 55
 56
         I = 8'b01010101;
 57
         for (s = 3'b000; s < 3'b111; s++)
 58
 59
           #1;
 60
         #1 $finish;
 61
       end
 62
 63 endmodule : Multiplexer_test
 64
 65 module Mux2to1_test;
       logic [6:0] Ī0;
logic [6:0] I1;
logic S;
 66
 67
 68
       logic [6:0] Y;
```

```
Filename: library_tests.sv
                                                                                     Page #: 2
 70
 71
      Mux2to1 DUT(.IO(IO),
 72
                    .I1(I1),
                    .S(S),
.Y(Y));
 73
 74
 75
 76
      initial begin
         $monitor($time,,
   "I0 = %b, I1 = %b, S = %b, Y = %b", I0, I1, S, Y);
 77
 78
 79
         I0 = 7'b0000000;
         I1 = 7'b11111111;
 80
         S = 0;
 81
         #1 S = 1;
 82
 83
         #1 $finish;
 84
      end
 85
 86 endmodule : Mux2to1_test
87
88 module MagComparator_test;
      logic [7:0] A; logic [7:0] B;
 89
 90
 91
      logic AltB;
 92
      logic AgtB;
 93
      logic AeqB;
 94
      MagComparator DUT (.A(A),
 95
 96
                            .B(B),
 97
                            .AltB(AltB),
 98
                            .AgtB(AgtB)
 99
                            .AeqB(AeqB));
100
101
      initial begin
         $monitor($time,,
102
         "A = %b, B = %b, AltB = %b, AgtB = %b, AeqB = %b", A, B , AltB, AgtB, AeqB);
103
104
         A = 8'b00000000;
         B = 8'b111111111;
105
         #2 A = 8'b111111111;
106
         #2 B = 8'b000000000;
107
108
         #2 $finish;
109
110
111 endmodule : MagComparator_test
112
113 module Comparator_test;
114
      logic [3:0] A;
115
      logic [3:0] B;
116
      logic AeqB;
117
      MagComparator DUT (.A(A),
118
119
120
                            .AeqB(AeqB));
121
122
      initial begin
         $monitor($time,,
123
         "A = \%b, B = \%b, AeqB = \%b", A, B, AeqB);
124
         A = 8'b00000000;
125
126
         B = 8'b111111111
127
         #2 A = 8'b111111111;
128
         #2 $finish;
129
130
131 endmodule : Comparator_test
132
133 module Adder_test;
134
      logic [4:0] A;
      logic [4:0] B;
logic [4:0] sum;
135
136
137
      logic cin;
```

138

139

140

logic cout;

Adder DUT (.A(A),

```
Filename: library_tests.sv
                              .B(B),
.cin(cin),
.cout(cout),
141
142
143
144
                              .sum(sum));
145
          initial begin
  $monitor($time,,
146
147
             smonitor($time,,
"A = %b, B = %b, cin = %b, sum = %b, cout = %b",
A, B, cin, sum, cout);
A = 5'b000000;
B = 5'b000000;
cin = 1'b0;
#2 cin = 1'b1;
#2 A = 5'b00001;
#3 B = 5'b00001;
148
149
150
151
152
153
154
             #2 B = 5'b00001;
#2 A = 5'b11111;
155
156
157
                 cin = 1'b0;
              #2 cin = 1'b1;
158
159
              #2 $finish;
160
161
```

162 endmodule : Adder_test

Page #: 3

```
Problem 10: [10 points]
Filename: hw3prob10.sv
AndrewID: xinyew
    `default_nettype none
    module hw3prob10
       (input logic [7:0] a, b,
       output logic [7:0] sum,
  5
       output logic overflow);
  6
  7
      logic [6:0] mag_a, mag_b , max, min;
logic [6:0] mag_sum;
  8
  9
      logic sign_a, sign_b, sign_sum;
 10
      assign mag_a = a[6:0];
 11
      assign mag_b = b[6:0]
 12
      assign sign_a = a[7];
assign sign_b = b[7];
 13
 14
 15
 16
      always_comb begin
 17
        if (mag_a > mag_b) begin
           max = mag_a;
 18
           min = mag_b;
 19
 20
           sign_sum = sign_a;
 21
        end
 22
        else begin
           max = mag_b;
 23
           min = mag_a;
 24
 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
 37
        sum = {sign_sum, mag_sum};
 38
 39
      end
 40 endmodule : hw3prob10
 41
 42 module hw3prob10_test;
      logic [7:0] a, b, sum;
43
44
      logic overflow;
 45
 46
      hw3prob10 DUT (.a(a),
 47
                      .b(b),
 48
                      .sum(sum)
                      .overflow(overflow));
 49
 50
 51
      initial begin
        $monitor($time,,
   "a = %b, b = %b, sum = %b, overflow = %b",
 52
 53
           a, b, sum, overflow);
 54
        a = 8'b000000000;
 55
        b = 8'b111111111
 56
        #1 a = 8'b00000001;
 57
        #1 a = 8'b10000001
 58
        #1 a = 8'b111111111
 59
        #1 a = 8'b011111111
 60
 61
            b = 8'b10000000;
        #1 b = 8'b10000001;
 62
        #1 b = 8'b00000001;
 63
 64
        #1 b = 8'b111111111;
        #1 b = 8'b01111111;
 65
 66
        #1 $finish;
 67
 68 endmodule : hw3prob10_test
```