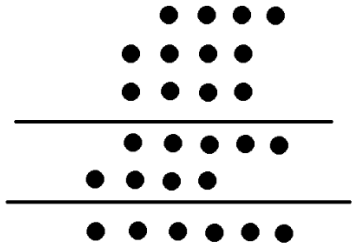
EEC 281 - Homework/Project #1

Problem 1

a) 6 bits, 4.2 format

b)



c) -7.5

d) 0.25

e) 10.75

f)

Time: 0 | a: 0001 | b: 0000 | c: 0000 | d: 000001 Y
Time: 100 | a: 0000 | b: 0000 | c: 0000 | d: 000000 Y
Time: 200 | a: 0000 | b: 1111 | c: 1000 | d: 100010 Y
Time: 300 | a: 0011 | b: 0011 | c: 0011 | d: 001111 Y
Time: 400 | a: 0100 | b: 0110 | c: 0111 | d: 011110 Y
Time: 500 | a: 0101 | b: 0011 | c: 0111 | d: 011001 Y
Time: 600 | a: 0111 | b: 1010 | c: 1100 | d: 111011 Y
Time: 700 | a: 1111 | b: 0111 | c: 0111 | d: 101011 Y
Time: 800 | a: 1111 | b: 1111 | c: 1111 | d: 111111 Y
Time: 900 | a: 1000 | b: 1001 | c: 1010 | d: 111010 Y
Time: 1000 | a: 1100 | b: 1101 | c: 1110 | d: 111110 Y
Time: 1100 | a: 1010 | b: 1011 | c: 1100 | d: 111100 Y
Time: 1200 | a: 0110 | b: 0101 | c: 0011 | d: 010110 Y
Time: 1300 | a: 0010 | b: 0001 | c: 0000 | d: 000100 Y
Time: 1400 | a: 1110 | b: 1101 | c: 1011 | d: 111010 Y

Functionality: Fully Functional

Code:

prob1.v

```
`timescale 10ps/1ps


module threetwo (
    a,
    b,
    c,
    e,
    f
```

```
);
    input  a;
    input  b;
    input  c;
    output e;
    output f;
    assign e = a ^ b ^ c;
    assign f = (a & b) | (b & c) | (a & c);

endmodule

module threetwo4(
    a,
    b,
    c,
    e,
    f
    );

    // input interface
    input [3:0]     a;
    input [3:0]     b;
    input [3:0]     c;
    // output interface
    output [4:0]    e;
    output [4:0]    f;

    reg [3:0]      _b;

    //----- main

    threetwo num00 (.a(a[0]),  .b(0),  .c(0),  .e(e[0]),  .f(f[0]));
    threetwo num01 (.a(a[1]),  .b(_b[0]),  .c(c[0]),  .e(e[1]),  .f(f[1]));
    threetwo num02 (.a(a[2]),  .b(_b[1]),  .c(c[1]),  .e(e[2]),  .f(f[2]));
    threetwo num03 (.a(a[3]),  .b(_b[2]),  .c(c[2]),  .e(e[3]),  .f(f[3]));
    threetwo num04 (.a(0),  .b(_b[3]),  .c(c[3]),  .e(e[4]),  .f(f[4]));

    always @(*) begin
        if (b[3] == 1'b1) begin
            _b = ~{1'b0, b[2:0]} + 1'b1;
        end
        else begin
            _b = b;
        end
```

```
        // $write("b: %b ", b);
        // $write("_b: %b\n", _b);
    end
endmodule

module prob1(
    a,
    b,
    c,
    d
    );

    // input interface
    input [3:0]     a;
    input [3:0]     b;
    input [3:0]     c;
    // output interface
    output [5:0]    d;

    wire [4:0]       e;
    wire [4:0]       f;
    //----- main
    threetwo4 _threetwo4(
        .a(a),
        .b(b),
        .c(c),
        .e(e),
        .f(f)
    );
    assign d = e + {f, 1'b0};
endmodule
```

prob1_tbench.vt

```
`timescale 10ps/1ps

module prob1_tbench;


    reg [3:0]   a;
    reg [3:0]   b;
    reg [3:0]   c;

    wire [5:0]  d;
```

```
    prob1 _prob1 (
        .a(a),
        .b(b),
        .c(c),
        .d(d)
    );


    initial begin

        a = 4'b0001; b = 4'b0000; c = 4'b0000;
        #10 a = 4'b0000; b = 4'b0000; c = 4'b0000;
        #10 a = 4'b0000; b = 4'b1111; c = 4'b1000;
        #10 a = 4'b0011; b = 4'b0011; c = 4'b0011;
        #10 a = 4'b0100; b = 4'b0110; c = 4'b0111;
        #10 a = 4'b0101; b = 4'b0011; c = 4'b0111;
        #10 a = 4'b0111; b = 4'b1010; c = 4'b1100;
        #10 a = 4'b1111; b = 4'b0111; c = 4'b0111;
        #10 a = 4'b1111; b = 4'b1111; c = 4'b1111;
        #10 a = 4'b1000; b = 4'b1001; c = 4'b1010;
        #10 a = 4'b1100; b = 4'b1101; c = 4'b1110;
        #10 a = 4'b1010; b = 4'b1011; c = 4'b1100;
        #10 a = 4'b0110; b = 4'b0101; c = 4'b0011;
        #10 a = 4'b0010; b = 4'b0001; c = 4'b0000;
        #10 a = 4'b1110; b = 4'b1101; c = 4'b1011;
        #10 $finish;
    end


    initial begin
        $monitor("Time: %0t | a: %b | b: %b | c: %b | d: %b",
                 $time, a, b, c, d);
    end

endmodule
```

Problem 2

a) 11 bits, 6.5 format

b) -32

c) -0.125

d) 28

e)

Time: 0 | mantissa: 0100 | exp: 100 | out: 00000000100 Y

Time: 100 | mantissa: 1000 | exp: 011 | out: 10000000000 Y

Time: 200 | mantissa: 0111 | exp: 011 | out: 01110000000 Y

Time: 300 | mantissa: 1010 | exp: 111 | out: 11111010000 Y
Time: 400 | mantissa: 0100 | exp: 101 | out: 00000001000 Y
Time: 500 | mantissa: 1001 | exp: 010 | out: 11001000000 Y
Time: 600 | mantissa: 0110 | exp: 110 | out: 00000011000 Y
Time: 700 | mantissa: 0101 | exp: 100 | out: 00000000101 Y
Time: 800 | mantissa: 0101 | exp: 000 | out: 00001010000 Y
Time: 900 | mantissa: 0101 | exp: 011 | out: 01010000000 Y
Time: 1000 | mantissa: 1011 | exp: 100 | out: 11111111011 Y
Time: 1100 | mantissa: 1011 | exp: 000 | out: 11110110000 Y
Time: 1200 | mantissa: 1011 | exp: 011 | out: 10110000000 Y
Time: 1300 | mantissa: 1100 | exp: 110 | out: 11111110000 Y
Time: 1400 | mantissa: 1100 | exp: 011 | out: 11000000000 Y

Functionality: Fully Functional

Code:
prob2.v

```
`timescale 10ps/1ps

module prob2(
    mantissa,
    exp,
    out
    );

    input [3:0]     mantissa;
    input [2:0]     exp;

    output reg [10:0]    out;
    always @(*) begin
        if (exp[2] == 2'b0)begin
            out = {{3{mantissa[3]}}, mantissa, {4'b0000}} << exp;
        end else begin
            out = $signed({{3{mantissa[3]}}, mantissa, {4'b0000}}) >>> (-exp);
            // $write("-exp: %h\n", -exp);
        end
    end

endmodule
```

prob2_tbench.vt

```
`timescale 10ps/1ps

module prob2_tbench;
```

```
    reg [3:0]     mantissa;
    reg [2:0]     exp;

    wire [10:0]    out;

    prob2 _prob2(
        .mantissa(mantissa),
        .exp(exp),
        .out(out)
    );
    initial begin

        mantissa = 4'b0100; exp = 3'b100;       //minimum attainable positive (non-
zero) value

        #10 mantissa = 4'b1000; exp = 3'b011;   //minimum attainable negative value
        #10 mantissa = 4'b0111; exp = 3'b011;   //maximum attainable positive value
        #10 mantissa = 4'b1010; exp = 3'b111;
        #10 mantissa = 4'b0100; exp = 3'b101;
        #10 mantissa = 4'b1001; exp = 3'b010;
        #10 mantissa = 4'b0110; exp = 3'b110;
        #10 mantissa = 4'b0101; exp = 3'b100;
        #10 mantissa = 4'b0101; exp = 3'b000;
        #10 mantissa = 4'b0101; exp = 3'b011;
        #10 mantissa = 4'b1011; exp = 3'b100;
        #10 mantissa = 4'b1011; exp = 3'b000;
        #10 mantissa = 4'b1011; exp = 3'b011;
        #10 mantissa = 4'b1100; exp = 3'b110;
        #10 mantissa = 4'b1100; exp = 3'b011;
        #10 $finish;
    end


    initial begin
        $monitor("Time: %0t | mantissa: %b | exp: %b | out: %b",
                $time, mantissa, exp, out);
    end

endmodule
```

Problem 3

a) 3 bits, exponent can be from -4 to 2.

b) mantissa: -8 exponent: -4

c) mantissa: 2 exponent: 1

d) mantissa: 3.5 exponent:2
e)
Time: 0 | in: 0001000 | mantissa: 0100 | exp: 000 Y
Time: 100 | in: 0000001 | mantissa: 0100 | exp: 101 Y
Time: 200 | in: 0100000 | mantissa: 0100 | exp: 010 Y
Time: 300 | in: 1111111 | mantissa: 1000 | exp: 100 Y
Time: 400 | in: 0010000 | mantissa: 0100 | exp: 001 Y
Time: 500 | in: 0111000 | mantissa: 0111 | exp: 010 Y
Time: 600 | in: 0111111 | mantissa: 0111 | exp: 010 Y
Time: 700 | in: 0001010 | mantissa: 0101 | exp: 000 Y
Time: 800 | in: 1000011 | mantissa: 1000 | exp: 010 Y
Time: 900 | in: 0000101 | mantissa: 0101 | exp: 111 Y
Time: 1000 | in: 0000010 | mantissa: 0100 | exp: 110 Y
Time: 1100 | in: 1111110 | mantissa: 1000 | exp: 101 Y
Time: 1200 | in: 1111100 | mantissa: 1000 | exp: 110 Y
Time: 1300 | in: 1111000 | mantissa: 1000 | exp: 111 Y
Time: 1400 | in: 1110000 | mantissa: 1000 | exp: 000 Y

Functionality: Fully Functional

Code:
prob3.v

```
`timescale 10ps/1ps

module prob3(
    in,
    mantissa,
    exp
    );

    input [6:0]         in;

    output reg [3:0]    mantissa;
    output reg [2:0]    exp;
    reg [9:0]           tmp;
    integer i;

    always @(*) begin
        i = 9;
        // $write("in: %d\n", in);
        tmp = in <<< 3;
        // $write("in: %d\n", in);
        while(i > 0 && tmp[i - 1] == tmp[i])
            i = i - 1;
```

```
        exp = i - 7;
        mantissa = tmp[i -: 4];
        // $write("i: %d\n", i);
    end

endmodule
```

prob3_tbench.vt

```
`timescale 10ps/1ps

module prob3_tbench;

    wire [3:0]     mantissa;
    wire [2:0]     exp;

    reg [6:0]       in;

    prob3 _prob3(
        .in(in),
        .mantissa(mantissa),
        .exp(exp)
    );
    initial begin
        in = 7'b0001000;
        #10 in = 7'b0000001;
        #10 in = 7'b0100000;
        #10 in = 7'b1111111;
        #10 in = 7'b0010000;
        #10 in = 7'b0111000;
        #10 in = 7'b0111111;
        #10 in = 7'b0001010;
        #10 in = 7'b1000011;
        #10 in = 7'b0000101;
        #10 in = 7'b0000010;
        #10 in = 7'b1111110;
        #10 in = 7'b1111100;
        #10 in = 7'b1111000;
        #10 in = 7'b1110000;
        #10 $finish;
    end


    initial begin
        $monitor("Time: %0t | in: %b | mantissa: %b | exp: %b",
                $time, in, mantissa, exp);
```

```
    end

endmodule
```

Problem 4

a)

inputs					outputs		
				c i	c o	c 1	s
a	b	c	d				
0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	1
0	0	0	1	0	0	0	1
0	0	0	1	1	0	1	0
0	0	1	0	0	0	0	1
0	0	1	0	1	0	1	0
0	0	1	1	0			0
0	0	1	1	1			1
0	1	0	0	0	0	0	1
0	1	0	0	1	0	1	0
0	1	0	1	0			0
0	1	0	1	1			1
0	1	1	0	0			0
0	1	1	0	1			1
0	1	1	1	0			1
0	1	1	1	1	1	1	0
1	0	0	0	0	0	0	1
1	0	0	0	1	0	1	0
1	0	0	1	0			0
1	0	0	1	1			1
1	0	1	0	0			0
1	0	1	0	1			1
1	0	1	1	0			1
1	0	1	1	1	1	1	0
1	1	0	0	0			0
1	1	0	0	1			1
1	1	0	1	0			1
1	1	0	1	1	1	1	0
1	1	1	0	0			1
1	1	1	0	1	1	1	0
1	1	1	1	0	1	1	0
1	1	1	1	1	1	1	1

Problem 5

a	b	c	d	ci	co	c1	s	
0	0	0	0	0	0	0	0	Y
0	0	0	0	1	0	0	1	Y
0	0	0	1	0	0	0	1	Y
0	0	0	1	1	0	1	0	Y
0	0	1	0	0	0	0	1	Y
0	0	1	0	1	0	1	0	Y
0	0	1	1	0	0	1	0	Y
0	0	1	1	1	0	1	1	Y
0	1	0	0	0	0	0	1	Y
0	1	0	0	1	0	1	0	Y
0	1	0	1	0	0	1	0	Y
0	1	0	1	1	0	1	1	Y
0	1	1	0	0	1	0	0	Y
0	1	1	0	1	1	0	1	Y
0	1	1	1	0	1	0	1	Y
0	1	1	1	1	1	1	0	Y
1	0	0	0	0	0	0	1	Y
1	0	0	0	1	0	1	0	Y
1	0	0	1	0	0	1	0	Y
1	0	0	1	1	0	1	1	Y
1	0	1	0	0	1	0	0	Y
1	0	1	0	1	1	0	1	Y
1	0	1	1	0	1	0	1	Y
1	0	1	1	1	1	1	0	Y
1	1	0	0	0	1	0	0	Y
1	1	0	0	1	1	0	1	Y
1	1	0	1	0	1	0	1	Y
1	1	0	1	1	1	1	0	Y
1	1	1	0	0	1	0	1	Y
1	1	1	0	1	1	1	0	Y
1	1	1	1	0	1	1	0	Y
1	1	1	1	1	1	1	1	Y

Functionality: Fully Functional

Code:
prob5.v

```
`timescale 10ps/1ps

module fourtwo(
    a,
    b,
    c,
```

```
    d,
    ci,
    co,
    c1,
    s
    );

    input   a;
    input   b;
    input   c;
    input   d;
    input   ci;

    output  co;
    output  c1;
    output  s;

    wire    co1;
    wire    s1;
    FA fa1(
        .a(a),
        .b(b),
        .c(c),
        .co(co),
        .s(s1)
    );
    FA fa2(
        .a(s1),
        .b(d),
        .c(ci),
        .co(c1),
        .s(s)
    );
endmodule

module FA(
    a,
    b,
    c,
    co,
    s
);
    input   a;
    input   b;
```

```
    input   c;

    output  co;
    output  s;

    assign s = a ^ b ^ c;
    assign co = (a & b) | (a & c) | (b & c);
endmodule
```

prob5_tbench.vt

```
`timescale 10ps/1ps

module prob5_tbench;

    reg     a;
    reg     b;
    reg     c;
    reg     d;
    reg     ci;

    wire    c1;
    wire    co;
    wire    s;

    reg [5:0]   i;

    fourtwo _fourtwo(
        .a(a),
        .b(b),
        .c(c),
        .d(d),
        .ci(ci),
        .co(co),
        .c1(c1),
        .s(s)
    );
    initial begin
        for(i = 0;i <= 5'b11111; i = i + 1)begin
            a = i[4];
            b = i[3];
            c = i[2];
            d = i[1];
            ci = i[0];
            #10;
```

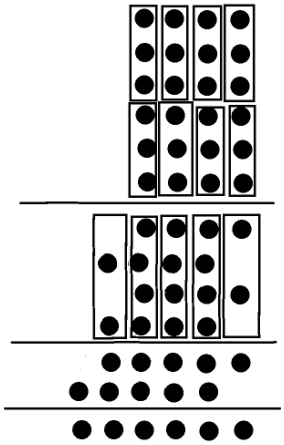
```
        end
        $finish;
    end


    initial begin
        $write("a \t | b \t | c \t | d \t | ci \t || co \t | c1 \t | s\n");
        $monitor(" %b \t | %b \t | %b \t | %b \t | %b \t || %b \t | %b \t | %b",
                 a, b, c, d, ci, co, c1, s);
    end

endmodule
```

Problem 6

a)



b)

a	b	c	d	e	f || g	
0000	0000	0000	0000	0000	0000 || 000000	Y
0001	0000	0000	0000	0000	0000 || 000001	Y
0000	0001	0000	0000	0000	0000 || 000001	Y
0000	0000	0001	0000	0000	0000 || 000001	Y
0000	0000	0000	0001	0000	0000 || 000001	Y
0000	0000	0000	0000	0001	0000 || 000001	Y
0000	0000	0000	0000	0000	0001 || 000001	Y
1111	0000	0000	0000	0000	0000 || 111111	Y
0000	1111	0000	0000	0000	0000 || 111111	Y
0000	0000	1111	0000	0000	0000 || 111111	Y
0000	0000	0000	1111	0000	0000 || 111111	Y
0000	0000	0000	0000	1111	0000 || 111111	Y
0000	0000	0000	0000	0000	1111 || 111111	Y
0111	0000	0000	0000	0000	0000 || 000111	Y
0000	0111	0000	0000	0000	0000 || 000111	Y

0000 | 0000 | 0111 | 0000 | 0000 | 0000 || 000111 Y
0000 | 0000 | 0000 | 0111 | 0000 | 0000 || 000111 Y
0000 | 0000 | 0000 | 0000 | 0111 | 0000 || 000111 Y
0000 | 0000 | 0000 | 0000 | 0000 | 0111 || 000111 Y
1000 | 0000 | 0000 | 0000 | 0000 | 0000 || 111000 Y
0000 | 1000 | 0000 | 0000 | 0000 | 0000 || 111000 Y
0000 | 0000 | 1000 | 0000 | 0000 | 0000 || 111000 Y
0000 | 0000 | 0000 | 1000 | 0000 | 0000 || 111000 Y
0000 | 0000 | 0000 | 0000 | 1000 | 0000 || 111000 Y
0000 | 0000 | 0000 | 0000 | 0000 | 1000 || 111000 Y
0001 | 0001 | 0001 | 0001 | 0001 | 0001 || 000110 Y
1111 | 1111 | 1111 | 1111 | 1111 | 1111 || 111010 Y
0001 | 0010 | 0011 | 0100 | 0101 | 0110 || 010101 Y
0111 | 0100 | 0101 | 0101 | 0101 | 0101 || 011111 Y
1001 | 1011 | 1011 | 1011 | 1011 | 1011 || 100000 Y

Functionality: Fully Functional

Code:
prob6.v

```
`timescale 10ps/1ps

module prob6(
    a,
    b,
    c,
    d,
    e,
    f,
    g
);
    input [3:0]     a;
    input [3:0]     b;
    input [3:0]     c;
    input [3:0]     d;
    input [3:0]     e;
    input [3:0]     f;

    output [5:0]    g;

    wire [4:0]      e1, f1, e2, f2;
    wire [5:0]      e3, f3;
    threetwo4_new _threetwo4_new_1(
        .a(a),
```

```
        .b(b),
        .c(c),
        .e(e1),
        .f(f1)
    );

    threetwo4_new _threetwo4_new_2(
        .a(d),
        .b(e),
        .c(f),
        .e(e2),
        .f(f2)
    );

    fourtwo4 _fourtwo4(
        .a(e1),
        .b(f1),
        .c(e2),
        .d(f2),
        .e(e3),
        .f(f3)
    );
    assign g = e3 + f3;
endmodule

module fourtwo4(
    a,
    b,
    c,
    d,
    e,
    f
);
    // input interface
    input  [4:0]        a;
    input  [4:0]        b;
    input  [4:0]        c;
    input  [4:0]        d;

    // output interface
    output [5:0]        e;
    output [5:0]        f;

    wire  [5:0]        co;      // intermediate "sideways" carry wires
```

```
    assign f[0] = 0;
    fourtwo num00
(.a(a[0]),  .b(b[0]),  .c(c[0]),  .d(d[0]),  .ci(0),   .co(co[0]),  .c1(f[1]),  .s(e
[0]));
    fourtwo num01
(.a(a[1]),  .b(b[1]),  .c(c[1]),  .d(d[1]),  .ci(co[0]),  .co(co[1]),  .c1(f[2]),  .
s(e[1]));
    fourtwo num02
(.a(a[2]),  .b(b[2]),  .c(c[2]),  .d(d[2]),  .ci(co[1]),  .co(co[2]),  .c1(f[3]),  .
s(e[2]));
    fourtwo num03
(.a(a[3]),  .b(b[3]),  .c(c[3]),  .d(d[3]),  .ci(co[2]),  .co(co[3]),  .c1(f[4]),  .
s(e[3]));
    fourtwo num04
(.a(a[4]),  .b(b[4]),  .c(c[4]),  .d(d[4]),  .ci(co[3]),  .co(co[4]),  .c1(f[5]),  .
s(e[4]));
    fourtwo num05
(.a(a[4]==1),  .b(b[4]==1),  .c(c[4]==1),  .d(d[4]==1),  .ci(co[4]),  .co(co[5]),  .
c1(),  .s(e[5]));
endmodule
module threetwo4_new(
    a,
    b,
    c,
    e,
    f
);
    // input interface
    input  [3:0]       a;
    input  [3:0]       b;
    input  [3:0]       c;

    // output interface
    output [4:0]       e;
    output [4:0]       f;

    assign f[0] = 0;
    threetwo num00 (.a(a[0]),  .b(b[0]),  .c(c[0]),  .e(e[0]),  .f(f[1]));
    threetwo num01 (.a(a[1]),  .b(b[1]),  .c(c[1]),  .e(e[1]),  .f(f[2]));
    threetwo num02 (.a(a[2]),  .b(b[2]),  .c(c[2]),  .e(e[2]),  .f(f[3]));
    threetwo num03 (.a(a[3]),  .b(b[3]),  .c(c[3]),  .e(e[3]),  .f(f[4]));
    threetwo num04 (.a(a[3] == 1),  .b(b[3] == 1),  .c(c[3] ==
1),  .e(e[4]),  .f());
```

```
endmodule
```

prob6_tbench.vt

```
`timescale 10ps/1ps

module prob6_tbench;

    reg [3:0]   a;
    reg [3:0]   b;
    reg [3:0]   c;
    reg [3:0]   d;
    reg [3:0]   e;
    reg [3:0]   f;

    wire [5:0]  g;


    prob6 _prob6(
        .a(a),
        .b(b),
        .c(c),
        .d(d),
        .e(e),
        .f(f),
        .g(g)
    );

    initial begin
        a = 0; b = 0; c = 0; d = 0; e = 0; f = 0;
        #10 a = 1; b = 0; c = 0; d = 0; e = 0; f = 0;
        #10 a = 0; b = 1; c = 0; d = 0; e = 0; f = 0;
        #10 a = 0; b = 0; c = 1; d = 0; e = 0; f = 0;
        #10 a = 0; b = 0; c = 0; d = 1; e = 0; f = 0;
        #10 a = 0; b = 0; c = 0; d = 0; e = 1; f = 0;
        #10 a = 0; b = 0; c = 0; d = 0; e = 0; f = 1;
        #10 a = -1; b = 0; c = 0; d = 0; e = 0; f = 0;
        #10 a = 0; b = -1; c = 0; d = 0; e = 0; f = 0;
        #10 a = 0; b = 0; c = -1; d = 0; e = 0; f = 0;
        #10 a = 0; b = 0; c = 0; d = -1; e = 0; f = 0;
        #10 a = 0; b = 0; c = 0; d = 0; e = -1; f = 0;
        #10 a = 0; b = 0; c = 0; d = 0; e = 0; f = -1;
        #10 a = 7; b = 0; c = 0; d = 0; e = 0; f = 0;
        #10 a = 0; b = 7; c = 0; d = 0; e = 0; f = 0;
        #10 a = 0; b = 0; c = 7; d = 0; e = 0; f = 0;
```

```
        #10 a = 0; b = 0; c = 0; d = 7; e = 0; f = 0;
        #10 a = 0; b = 0; c = 0; d = 0; e = 7; f = 0;
        #10 a = 0; b = 0; c = 0; d = 0; e = 0; f = 7;
        #10 a = -8; b = 0; c = 0; d = 0; e = 0; f = 0;
        #10 a = 0; b = -8; c = 0; d = 0; e = 0; f = 0;
        #10 a = 0; b = 0; c = -8; d = 0; e = 0; f = 0;
        #10 a = 0; b = 0; c = 0; d = -8; e = 0; f = 0;
        #10 a = 0; b = 0; c = 0; d = 0; e = -8; f = 0;
        #10 a = 0; b = 0; c = 0; d = 0; e = 0; f = -8;
        #10 a = 1; b = 1; c = 1; d = 1; e = 1; f = 1;
        #10 a = -1; b = -1; c = -1; d = -1; e = -1; f = -1;
        #10 a = 1; b = 2; c = 3; d = 4; e = 5; f = 6;
        #10 a = 7; b = 4; c = 5; d = 5; e = 5; f = 5;
        #10 a = -7; b = -5; c = -5; d = -5; e = -5; f = -5;
        #10 $finish;
    end


    initial begin
        $write("a | b | c | d | e | f || g\n");
        $monitor(" %b | %b | %b | %b | %b | %b || %b",
                 a, b, c, d, e, f, g);
    end

endmodule
```