Digital Design Assignment 1

Name: Ahmed Nour

ID in sheet: 16

Q1)

q1_1.v

Generate Simulate

Q1 > q1_1.v > muxed

```
1  module muxed (a ,b ,c ,d ,e ,f ,sel, out ,out_bar);
2
3  input a ,b ,c ,d ,e ,f ,sel;
4  output out ,out_bar;
5
6  assign out = (sel == 0)? (a & b & c) : (d ~^ e ~^ f);
7  assign out_bar = ~out ;
8
9  endmodule
```

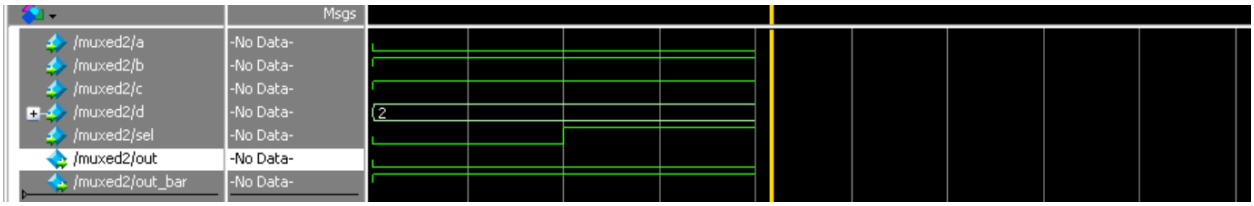
Q2)

q2_1.v

Generate Simulate

Q2 > q2_1.v > muxed2

```
1  module muxed2 (d, a, b, c, sel, out ,out_bar);
2
3  input [2:0] d;
4  input a, b, c, sel;
5  output reg out , out_bar;
6
7  always @(*) begin
8  casex (sel)
9  1'b0 : out = d[2] | (d[1] & d[0]);
10 1'b1 : out = a ~^ (b ~^ c);
11 endcase
12 out_bar = ~out;
13 end
14
15 endmodule
```



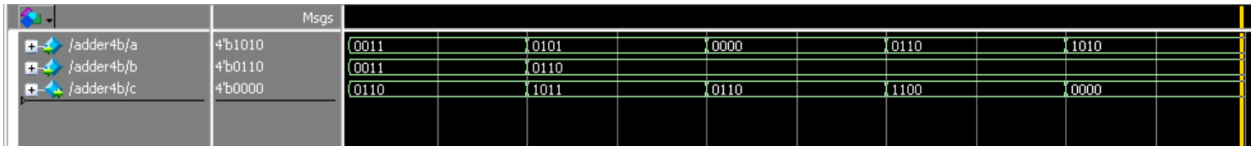
Q3)

q3_1.v

Ass1 > Q3 > q3_1.v > adder4b

Generate Simulate

```
1  module adder4b (a, b, c);
2
3  input  [3:0] a , b;
4  output [3:0] c;
5
6  assign c = a + b;
7
8  endmodule
```



Note: the last test case has 1 in ignored carrier.

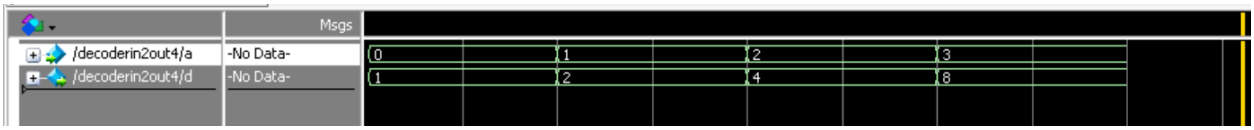
Q4)

q4_1.v

Generate Simulate

Ass1 > Q4 > q4_1.v > ...

```
1  module decoderin2out4 (a, d);
2
3  input  [1:0] a;
4  output [3:0] d;
5
6  assign d = (a == 2'b00) ? (4'b0001) :
7             (a == 2'b01) ? (4'b0010) :
8             (a == 2'b10) ? (4'b0100) :
9             (a == 2'b11) ? (4'b1000) :
10            (4'b0000);
11
12 endmodule
```



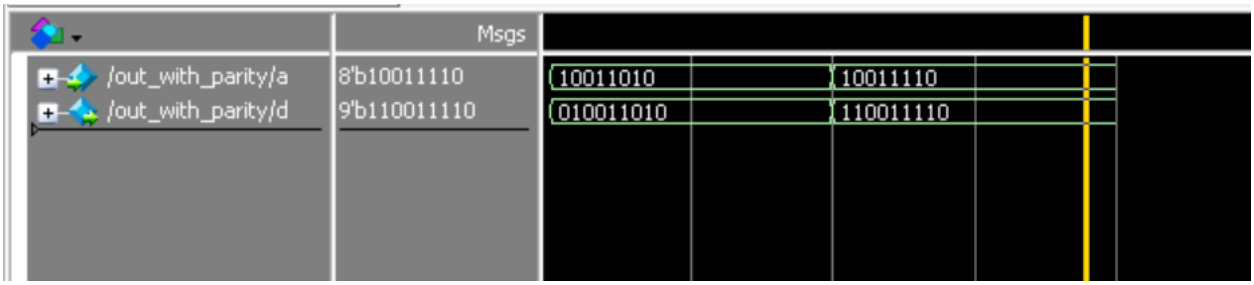
Q5)

q5_1.v

Generate Simulate

Ass1 > Q5 > q5_1.v > ...

```
1  module out_with_parity (a, d);
2
3  input  [7:0] a;
4  output [8:0] d;
5
6  assign d = {(^a) , a};
7
8  endmodule
9
```



Q6)

```v
module Nbit_ALU (in0, in1, opcode, out);
parameter N1 = 4;

input [N1 -1 :0] in0, in1;
input [1:0] opcode;

output reg [N1 -1 :0] out;

always @(*) begin
    case (opcode)
        2'b00: out = in0 + in1;
        2'b01: out = in0 | in1;
        2'b10: out = in0 - in1;
        2'b11: out = in0 ^ in1;
    endcase
end
endmodule
```

```v
module sev_seg_dec (in0, enable, a, b, c, d, e, f, g);
parameter N2 = 4;
input [N2 -1 :0] in0;
input enable;
output reg a, b, c, d, e, f, g;

wire in1;
always @(*) begin
if (enable) begin
    case (in0)
        4'h0: {a,b,c,d,e,f,g} = 7'b1111110;
        4'h1: {a,b,c,d,e,f,g} = 7'b0110000;
        4'h2: {a,b,c,d,e,f,g} = 7'b1101101;
        4'h3: {a,b,c,d,e,f,g} = 7'b1111001;
        4'h4: {a,b,c,d,e,f,g} = 7'b0110011;
        4'h5: {a,b,c,d,e,f,g} = 7'b1011011;
        4'h6: {a,b,c,d,e,f,g} = 7'b1011111;
        4'h7: {a,b,c,d,e,f,g} = 7'b1110000;
        4'h8: {a,b,c,d,e,f,g} = 7'b1111111;
        4'h9: {a,b,c,d,e,f,g} = 7'b1111011;
        4'hA: {a,b,c,d,e,f,g} = 7'b1110111;
        4'hB: {a,b,c,d,e,f,g} = 7'b0011111;
        4'hC: {a,b,c,d,e,f,g} = 7'b1001110;
        4'hD: {a,b,c,d,e,f,g} = 7'b0111101;
        4'hE: {a,b,c,d,e,f,g} = 7'b1001111;
        4'hF: {a,b,c,d,e,f,g} = 7'b1000111;
        default: {a,b,c,d,e,f,g} = 7'b0000000;
    endcase
end else begin {a,b,c,d,e,f,g} = 7'b0000000; end
end
endmodule
```

```v
module executor (Ain, Bin, opcode, enable, ALUout, a, b, c, d, e, f, g);
parameter N = 4;

input [N-1 :0] Ain, Bin;
input [1:0] opcode;
input enable;

output [N-1 :0] ALUout;
output a, b, c, d, e, f, g;

Nbit_ALU alu1 (Ain, Bin, opcode, ALUout);
sev_seg_dec sdec1 (ALUout, enable, a, b, c, d, e, f, g);

endmodule
```