

يوسف أحمد محمد ابراهيم

Assignment 2
Extra

Circuit_1 :

- module:

```
1  module circuit_1#(parameter USE_GRAY = 1)
2      |     |     |     (input [2:0]a, output reg [6:0]out);
3
4      always @(*) begin
5          if (USE_GRAY == 0)
6              case (a)
7                  3'd1 : out = 7'b0000001;
8                  3'd2 : out = 7'b0000010;
9                  3'd3 : out = 7'b0000100;
10                 3'd4 : out = 7'b0001000;
11                 3'd5 : out = 7'b0010000;
12                 3'd6 : out = 7'b0100000;
13                 3'd7 : out = 7'b1000000;
14                 default : out = 7'b0000000;
15             endcase
16         else
17             case (a)
18                 3'd1 : out = 7'b0000001;
19                 3'd2 : out = 7'b0000011;
20                 3'd3 : out = 7'b0000010;
21                 3'd4 : out = 7'b0000110;
22                 3'd5 : out = 7'b0000111;
23                 3'd6 : out = 7'b0000101;
24                 3'd7 : out = 7'b0000100;
25                 default : out = 7'b0000000;
26             endcase
27         end
28     endmodule
```

- Testbench1:

```

1  module circuit1_tp1();
2      reg [2:0]a_tb;
3      reg [6:0]out_expected;
4      wire [6:0] out_dut;
5      circuit_1 #(1) DUT(a_tb, out_dut);
6      integer i;
7      initial begin
8          for(i = 0; i < 8; i = i+1) begin
9              a_tb = i;
10             case (a_tb)
11                 3'd1 : out_expected = 7'b0000001;
12                 3'd2 : out_expected = 7'b0000011;
13                 3'd3 : out_expected = 7'b0000010;
14                 3'd4 : out_expected = 7'b0000110;
15                 3'd5 : out_expected = 7'b0000111;
16                 3'd6 : out_expected = 7'b0000101;
17                 3'd7 : out_expected = 7'b0000100;
18                 default : out_expected = 7'b0000000;
19             endcase
20             #10;
21             if (out_dut!= out_expected)
22                 $stop;
23         end
24     end
25 endmodule
26

```

- Output1 (80ns) “*i is not important*”

| | | |
|------------------------|--------------|---|
| /circuit1_tp1/a_tb | 3'h7 | 0 1 2 3 4 5 6 7 |
| /circuit1_tp1/out_e... | 7'h04 | 00 01 03 02 06 07 05 04 |
| /circuit1_tp1/out_dut | 7'h04 | 00 01 03 02 06 07 05 04 |
| /circuit1_tp1/i | 32'h00000008 | 00... 00... 00... 00... 00... 00... 00... 00... |

- Testbench2:

```

28  module circuit1_tp2();
29      reg [2:0]a_tb;
30      reg [6:0]out_expected;
31      wire [6:0] out_dut;
32      circuit_1 #(0) DUT(a_tb, out_dut);
33      integer i;
34  initial begin
35      for(i = 0; i < 8; i = i+1) begin
36          a_tb = i;
37          case(a_tb)
38              3'd1 : out_expected = 7'b0000001;
39              3'd2 : out_expected = 7'b0000010;
40              3'd3 : out_expected = 7'b0000100;
41              3'd4 : out_expected = 7'b0001000;
42              3'd5 : out_expected = 7'b0010000;
43              3'd6 : out_expected = 7'b0100000;
44              3'd7 : out_expected = 7'b1000000;
45              default : out_expected = 7'b0000000;
46          endcase
47          #10;
48          if (out_dut != out_expected)
49              $stop;
50      end
51  end
52 endmodule

```

- Output2 (80ns) “i is not important”

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|------------------------|--------------|---|-------|-------|-------|-------|-------|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|-------|-------|-------|-------|-------|
| /circuit1_tp2/a_tb | 3'h7 | <table border="1"> <tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr> <tr><td>00</td><td>01</td><td>02</td><td>04</td><td>08</td><td>10</td><td>20</td><td>40</td></tr> <tr><td>00</td><td>01</td><td>02</td><td>04</td><td>08</td><td>10</td><td>20</td><td>40</td></tr> <tr><td>00...</td><td>00...</td><td>00...</td><td>00...</td><td>00...</td><td>00...</td><td>00...</td><td>00...</td></tr> </table> | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 00 | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 00 | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 00... | 00... | 00... | 00... | 00... | 00... | 00... | 00... |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 | 01 | 02 | 04 | 08 | 10 | 20 | 40 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00 | 01 | 02 | 04 | 08 | 10 | 20 | 40 | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 00... | 00... | 00... | 00... | 00... | 00... | 00... | 00... | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| /circuit1_tp2/out_e... | 7'h40 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| /circuit1_tp2/out_dut | 7'h40 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| /circuit1_tp2/i | 32'h00000008 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Circuit_2 :

- module:

```
1  module circuit_2(input d, s0, s1, output reg [3:0]y);
2      always @(*) begin
3          y = 4'b0000;
4          case ({s1, s0})
5              1 : y[1] = d;
6              2 : y[2] = d;
7              3 : y[3] = d;
8              default : y[0] = d;
9      endcase
10     end
11 endmodule
```

- Testbench:

```

1  module circuit2_tp();
2      reg s0_tb, s1_tb;
3      wire [3:0]y_dut;
4      reg [3:0] y_expected;
5      localparam d = 1;
6
7      circuit_2 dut(d, s0_tb, s1_tb, y_dut);
8
9      integer i;
10     initial begin
11         for (i = 0; i < 4; i = i + 1) begin
12             {s1_tb, s0_tb} = i[1:0];
13             y_expected = 4'b0;
14             y_expected[i] = d;
15             #10;
16             if(y_dut != y_expected)
17                 $stop;
18         end
19     end
20 endmodule

```

- Output1 (40ns) “*i is not important*”

