

PRELAB 2: THỰC HIỆN CÁC IC CHỨC NĂNG CƠ BẢN TRÊN FPGA

Họ và tên: Lâm Thành Phát	Lớp TN: L21
MSSV: 2111974	Ngày: 15/10/2022

A. PRELAB

I. MỤC TIÊU

- Nắm được cách sử dụng kit thí nghiệm, phần mềm lập trình.
- Nắm được cách khảo sát và thiết kế hệ mạch đếm sử dụng các IC chức năng cơ bản.
- Nắm được quy trình mô tả phần cứng trên FPGA.

II. CHUẨN BỊ

- Để chuẩn bị tốt cho bài thí nghiệm, sinh viên **PHẢI** đọc trước phần Phụ lục 1 và hoàn thành các bước của Sample lab trong Phụ lục 2, nộp kèm với Prelab 2 trước khi vào lớp.
- Sinh viên **phải** hoàn thành và nộp Prelab 2 trước khi vào lớp.

Sinh viên xem phần phụ lục và Kit DE2 Manual để hiểu cách sử dụng Kit DE2, nối dây, các thiết bị ngoại vi, cách sử dụng phần mềm Quartus để mô phỏng, tổng hợp mạch. Sinh viên tham khảo các tài liệu để hiểu cách viết thiết kế phần cứng bằng ngôn ngữ SystemVerilog.

Với Lab 2 sinh viên sẽ sử dụng các ngoại vi LEDR, LEDG, và SW. Dưới đây là header nối dây căn bản của các ngoại vi này khi sinh viên tổng hợp project trên Intel Quartus. (Lưu ý import file DE2_pin_list/assignment).

Lưu ý, top-level của project phải đặt tên là tên của file header (nối pin), ví dụ là “lab2tn1_wrapper”, với module “lab2tn1” là module chứa đoạn mã mô tả thiết kế của thí nghiệm 1.

Giả sử, với thí nghiệm viết thiết kế phần cứng bằng ngôn ngữ SystemVerilog cho hàm $f(x, y, z) = \underline{xyz}$ là thí nghiệm 1 của lab 2, ta có đoạn mã SystemVerilog như sau:

```
lab2tn1.sv
1  module lab2tn1 (
2      input x,y,z,
3      output f
4  );
5
6  assign f = !x & y & z
7
8  endmodule
```

Hình 2.1: module “lab2tn1” thực hiện hàm $f(x,y,z) = \underline{xyz}$

Vậy đoạn mã của top level “lab2tn1_wrapper” như sau:

```
lab2tn1_wrapper.sv
1  module lab2tn1_wrapper (
2      input [2:0] SW,
3      output [2:0] LEDR,
4      output [0:0] LEDG
5  );
6
7  lab2tn1 ilab2tn1 (SW[2],SW[1],SW[0],LEDG);
8
9  assign LEDR = SW;
10
11 endmodule
```

Hình 2.2: Top level “lab2tn1_wrapper” thực hiện hàm $f(x,y,z) = \underline{xyz}$

Như vậy ta đã hoàn thành phần viết đoạn mã của chương trình và project Quartus, sinh viên thực hiện mô phỏng, nạp Kit như đã được hướng dẫn tại Lab 0.

Với các bài thí nghiệm yêu cầu sử dụng IC 74LS151/74LS138, sinh viên viết đoạn mã mô tả IC này và tiến hành gọi module như trên.

BÀI CHUẨN BỊ 1

Mục tiêu: Thực hiện khảo sát hoạt động của hàm Boolean trên Kit DE2

Yêu cầu:

- Sinh viên thực hiện khảo sát hoạt động của hàm $f(x, y, z) = \underline{xyz} + \underline{xyz} + xy$
- Sinh viên viết đoạn mã SystemVerilog thực hiện hoạt động của hàm $f(x, y, z) = \underline{xyz} + \underline{xyz} + xy$ như **hướng dẫn** từ lab 0 với sơ đồ gán chân như sau:
 - Chân x, y, z gán tới SW2, SW1, SW0 và LEDR2, LEDR1, LEDR0 tương ứng
 - Chân f gán tới LEDG0
- Sinh viên tiến hành biên dịch đoạn chương trình trên. Sau đó, sinh viên tiến mô phỏng dạng sóng ngõ ra trong tất cả các trường hợp của ngõ vào. Chụp hình dạng sóng ngõ ra.

File design.sv:

```
// Code your design here
module lab2_tn1(
    input a,b,c,
    output out
);

    assign out = (!a&b&c)|(!a&!b&c)|(a&b);

endmodule
```

File testbench.sv:

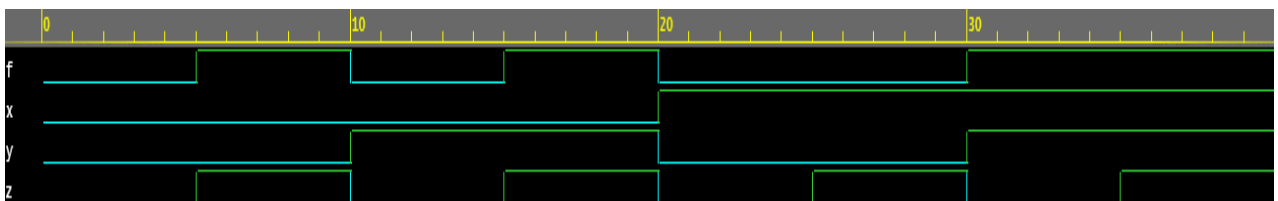
```
lab2_tn1 > test_bench.sv
1 // Code your testbench here
2 // or browse Examples
3 `timescale 1ns/1ns
4 module testbench;
5     reg x,y,z;
6     wire f;
7
8     lab2_tn1 dut(.a(x),.b(y),.c(z),.out(f));
9
10    initial begin
11        $dumpfile("dump.vcd");
12        $dumpvars(1);
13        x = 0;
14        y = 0;
15        z = 0;
16        #5
17        x = 0;
18        y = 0;
19        z = 1;
20        #5
21        x = 0;
22        y = 1;
23        z = 0;
24        #5
25        x = 0;
26        y = 1;
27        z = 1;
28        #5
29        x = 1;
30        y = 0;
31        z = 0;
```

```

28      #5
29      x = 1;
30      y = 0;
31      z = 0;
32      #5
33      x = 1;
34      y = 0;
35      z = 1;
36      #5
37      x = 1;
38      y = 1;
39      z = 0;
40      #5
41      x = 1;
42      y = 1;
43      z = 1;
44      #5
45      x = 0;
46      y = 0;
47      z = 0;
48      #5
49      $finish;
50  end
51  endmodule

```

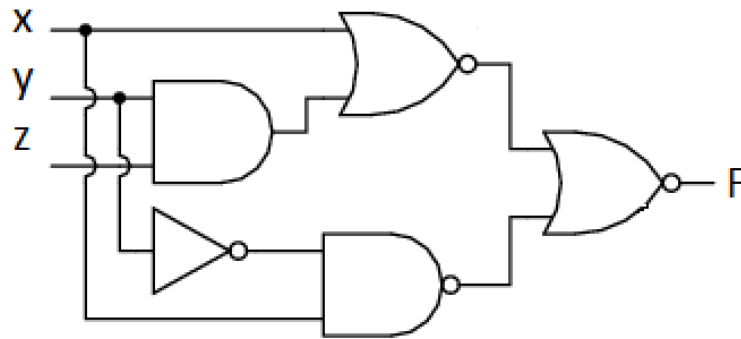
Hình dạng sóng ngõ ra:



BÀI CHUẨN BỊ 2

Mục tiêu: Hiểu được cách khảo sát hàm boolean sử dụng kit DE2

Yêu cầu: Sinh viên xem mạch được cho bởi **Hình 2.3**



Hình 2.3

- Sinh viên viết đoạn mã SystemVerilog thực hiện hoạt động của hàm F trên **Hình 2.3** như **hướng dẫn** từ lab 0 với sơ đồ gán chân như sau:
 - Chân x, y, z gán tới SW2, SW1, SW0 và LEDR2, LEDR1, LEDR0 tương ứng
 - Chân f gán tới LEDG0
- Sinh viên tiến hành biên dịch đoạn chương trình trên. Sau đó, sinh viên tiến mô phỏng dạng sóng ngõ ra trong tất cả các trường hợp của ngõ vào. Chụp hình dạng sóng ngõ ra.

Hàm số đề bài: $f(x,y,z) = \overline{\overline{xy} + x + yz}$

File design.sv:

```
lab2_tn2 > ≡ design.sv
1  // Code your design here
2  module lab2_tn2(
3      input a,b,c,
4      output out
5  );
6      assign out = (!((a&!b))|(!a|(b&c)));
7
8  endmodule
```

File testbench.sv:

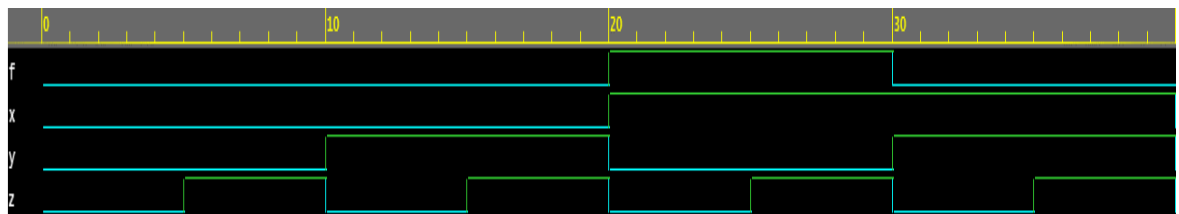
```
lab2_tn2 > testbench.sv
1 // Code your testbench here
2 // or browse Examples
3 `timescale 1ns/1ns
4 module testbench;
5     reg x,y,z;
6     wire f;
7
8     lab2_tn2 dut(.a(x),.b(y),.c(z),.out(f));
9
10    initial begin
11        $dumpfile("dump.vcd");
12        $dumpvars(1);
13        x = 0;
14        y = 0;
15        z = 0;
16        #5
17        x = 0;
18        y = 0;
19        z = 1;
20        #5
21        x = 0;
22        y = 1;
23        z = 0;
24        #5
25        x = 0;
26        y = 1;
27        z = 1;
28        #5
29        x = 1;
30        y = 0;
31        z = 0;
```

```

32      #5
33      x = 1;
34      y = 0;
35      z = 1;
36      #5
37      x = 1;
38      y = 1;
39      z = 0;
40      #5
41      x = 1;
42      y = 1;
43      z = 1;
44      #5
45      x = 0;
46      y = 0;
47      z = 0;
48      #5
49      $finish;
50  end
51  endmodule

```

Hình dạng sóng ngõ ra:



BÀI CHUẨN BỊ 3

Mục tiêu: Nắm được cách thức thiết kế các IC chức năng bằng ngôn ngữ SystemVerilog và ứng dụng trên kit DE2

Yêu cầu:

- Cho hàm boolean $f(x, y, z) = \sum(1, 2, 4, 7)$
- Sinh viên sử dụng IC 74LS151 và các cổng logic cần thiết để thiết kế mạch thực hiện hàm boolean đã cho.

- Sinh viên viết đoạn mã SystemVerilog thực hiện hoạt động của hàm $f(x, y, z) = \sum(1, 2, 4, 7)$ như **hướng dẫn** từ Lab 0 với sơ đồ gán chân như sau:
 - Chân x, y, z gán tới SW2, SW1, SW0 và LEDR2, LEDR1, LEDR0 tương ứng
 - Chân f gán tới LEDG0
 - Trong đó, gọi IC 74LS151 như một module con
- Sinh viên tiến hành biên dịch đoạn chương trình trên. Sau đó, sinh viên tiến mô phỏng dạng sóng ngõ ra trong tất cả các trường hợp của ngõ vào. Chụp hình dạng sóng ngõ ra.

File design.sv:

```
LAB_2 > lab2_tn3 > design.sv
1 // Code your design here
2 module Mux(d0,d1,d2,d3,d4,d5,d6,d7,x,y,z,out);
3 input d0,d1,d2,d3,d4,d5,d6,d7;
4 input x,y,z;
5 output reg out;
6
7 logic [2:0] sel;
8 assign sel[0] = z;
9 assign sel[1] = y;
10 assign sel[2] = x;
11 always@(sel)
12 begin
13 case(sel)
14 3'b000:out=d0;
15 3'b001:out=d1;
16 3'b010:out=d2;
17 3'b011:out=d3;
18 3'b100:out=d4;
19 3'b101:out=d5;
20 3'b110:out=d6;
21 3'b111:out=d7;
22 endcase
23 end
24 endmodule
25
```

File testbench.sv:

LAB_2 > lab2_tn3 > testbench.sv

```
1 // Code your testbench here
2 // or browse Examples
3 `timescale 1ns/1ns
4 module TestModule;
5 // Inputs
6 reg d0;
7 reg d1;
8 reg d2;
9 reg d3;
10 reg d4;
11 reg d5;
12 reg d6;
13 reg d7;
14 reg x,y,z;
15
16 // Outputs
17 wire f;
```

```

19
20 Muiltplexer uut (
21     .d0(d0),
22     .d1(d1),
23     .d2(d2),
24     .d3(d3),
25     .d4(d4),
26     .d5(d5),
27     .d6(d6),
28     .d7(d7),
29     .x(x),
30     .y(y),
31     .z(z),
32     .out(f)
33 );
34
35 initial begin
36     $dumpfile("dump.vcd");
37     $dumpvars(1);

```

```

64
65     #10;
66     d0 = 0;d1 = 1;d2 = 1;d3 = 0;d4 = 1;d5 = 0;d6 = 0;d7 = 1;
67     x=1;y=1;z=1;
68     #10
69     $finish;
70 end
71 endmodule

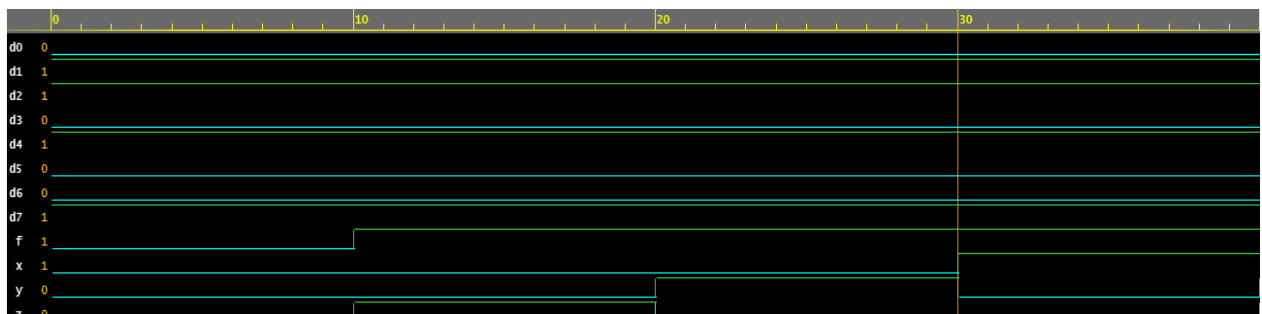
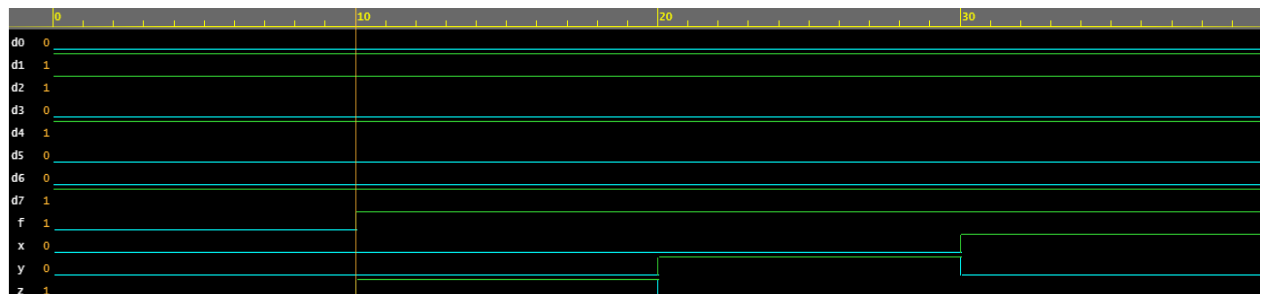
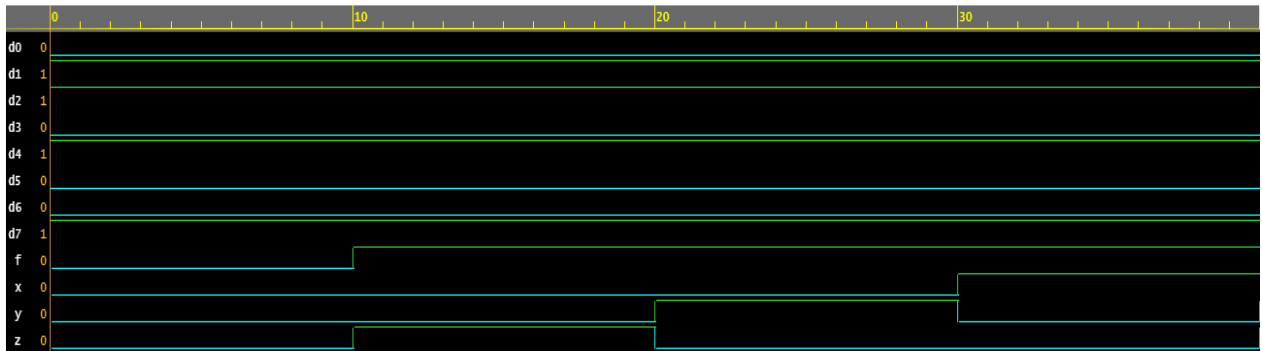
```

```

38     d0 = 0; d1 = 1; d2 = 1; d3 = 0; d4 = 1; d5 = 0; d6 = 0; d7 = 1;
39     x=0;y=0;z=0;
40
41     #10;
42     d0 = 0; d1 = 1; d2 = 1; d3 = 0; d4 = 1; d5 = 0; d6 = 0; d7 = 1;
43     x=0;y=0;z=1;
44
45     #10;
46     d0 = 0; d1 = 1; d2 = 1; d3 = 0; d4 = 1; d5 = 0; d6 = 0; d7 = 1;
47     x=0;y=1;z=0;
48
49     #10;
50     d0 = 0; d1 = 1; d2 = 1; d3 = 0; d4 = 1; d5 = 0; d6 = 0; d7 = 1;
51     x=0;y=1;z=1;
52
53     #10;
54     d0 = 0; d1 = 1; d2 = 1; d3 = 0; d4 = 1; d5 = 0; d6 = 0; d7 = 1;
55     x=1;y=0;z=0;
56
57     #10;
58     d0 = 0;d1 = 1;d2 = 1;d3 = 0;d4 = 1;d5 = 0;d6 = 0;d7 = 1;
59     x=1;y=0;z=1;
60
61     #10;
62     d0 = 0;d1 = 1;d2 = 1;d3 = 0;d4 = 1;d5 = 0;d6 = 0;d7 = 1;
63     x=1;y=1;z=0;
64
65     #10;
66     d0 = 0;d1 = 1;d2 = 1;d3 = 0;d4 = 1;d5 = 0;d6 = 0;d7 = 1;

```

Hiện thị dạng sóng ngõ ra:



BÀI CHUẨN BỊ 4

Mục tiêu: Nắm được cách thức thiết kế các IC chức năng bằng ngôn ngữ SystemVerilog và ứng dụng trên kit DE2

Yêu cầu:

- Cho hàm boolean $f(x, y, z, w) = \sum(2, 5, 7, 9, 12, 13)$
- Sinh viên sử dụng IC 74LS151 và các cổng logic cần thiết để thiết kế mạch thực hiện hàm boolean đã cho.

- Sinh viên viết đoạn mã SystemVerilog thực hiện hoạt động của hàm $f(x, y, z, w)$ như **hướng dẫn** từ Prelab với sơ đồ gán chân như sau:
 - Chân x, y, z, w gán tới SW3, SW2, SW1, SW0 và LEDR3, LEDR2, LEDR1, LEDR0 tương ứng
 - Chân f gán tới LEDG0
 - Trong đó, gọi IC 74LS151 như một module con
- Sinh viên tiến hành biên dịch đoạn chương trình trên. Sau đó, sinh viên tiến mô phỏng dạng sóng ngõ ra trong tất cả các trường hợp của ngõ vào. Chụp hình dạng sóng ngõ ra.

$$f(x, y, z, w) = (m_1 + m_6)\overline{w} + (m_2 + m_3 + m_4 + m_5)w$$

$$\Rightarrow D_1 = D_6 = \overline{w}; D_2 = D_3 = D_4 = D_5 = w$$

File design.sv:

LAB_2 > lab2_tn4 > ≡ design.sv

```
1  // Code your design here
2  module Muiltplexer(d0,d1,d2,d3,d4,d5,d6,d7,x,y,z,w,out);
3  input d0,d1,d2,d3,d4,d5,d6,d7;
4  input x,y,z,w;
5      output reg out;
6
7      logic [2:0] sel;
8      assign sel[0] = z;
9      assign sel[1] = y;
10     assign sel[2] = x;
11
12     assign d1 = !w;
13     assign d6 = !w;
14     assign d2 = w;
15     assign d3 = w;
16     assign d4 = w;
17     assign d5 = w;
18 always@(sel)
19 begin
20 case(sel)
21 3'b000:out=d0;
22 3'b001:out=d1;
23 3'b010:out=d2;
24 3'b011:out=d3;
25 3'b100:out=d4;
26 3'b101:out=d5;
27 3'b110:out=d6;
28 3'b111:out=d7;
29 endcase
30 end
31 endmodule
```

File testbench.sv:

LAB_2 > lab2_tn4 > testbench.sv

```
1  // Code your testbench here
2  // or browse Examples
3  `timescale 1ns/1ns
4  module TestModule;
5  // Inputs
6  reg d0;
7  reg d1;
8  reg d2;
9  reg d3;
10 reg d4;
11 reg d5;
12 reg d6;
13 reg d7;
14 reg x,y,z,w;
15 // Outputs
16 wire f;
17
```

```
19 Muiltplexer uut (
20   .d0(d0),
21   .d1(d1),
22   .d2(d2),
23   .d3(d3),
24   .d4(d4),
25   .d5(d5),
26   .d6(d6),
27   .d7(d7),
28   .x(x),
29   .y(y),
30   .z(z),
31   .w(w),
32   .out(f)
33 );
```

```

50  #10;
51
52  d0 = 0;
53  d1 = 0;
54  d2 = 1;
55  d3 = 1;
56  d4 = 1;
57  d5 = 1;
58  d6 = 0;
59  d7 = 0;
60  x=0;
61  y=1;
62  z=0;
63  w=1;
64  #10;
65  d0 = 0;
66  d1 = 0;
67  d2 = 1;
68  d3 = 1;
69  d4 = 1;
70  d5 = 1;
71  d6 = 0;
72  d7 = 1;
73  x=0;
74  y=1;
75  z=1;
76  w=1;
77  #10;
78  d0 = 0;
79  d1 = 0;

```

```

80  d2 = 1;
81  d3 = 1;
82  d4 = 1;
83  d5 = 1;
84  d6 = 0;
85  d7 = 1;
86  x=1;
87  y=0;
88  z=0;
89  w=1;
90  #10;
91  d0 = 0;
92  d1 = 1;
93  d2 = 0;
94  d3 = 0;
95  d4 = 0;
96  d5 = 0;
97  d6 = 1;
98  d7 = 1;
99  x=1;
100 y=1;
101 z=0;
102 w=0;
103 #10
104 $finish;
105 end
106 endmodule

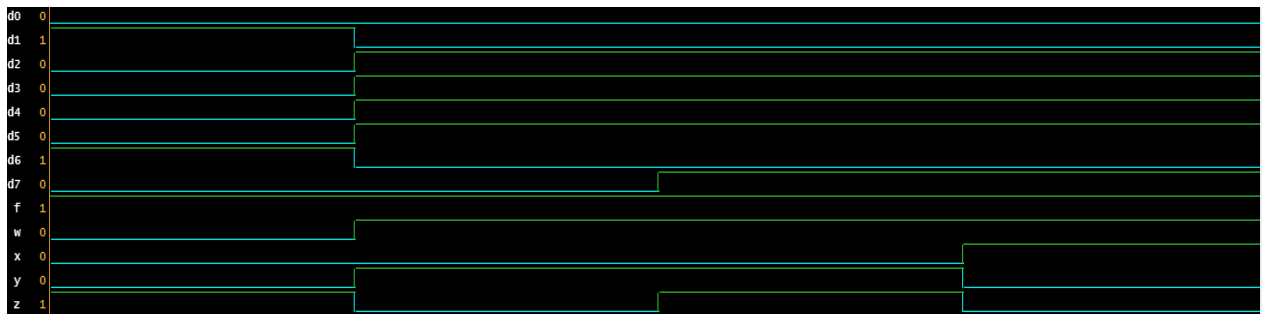
```

```

35  initial begin
36      $dumpfile("dump.vcd");
37      $dumpvars(1);
38      d0 = 0;
39      d1 = 1;
40      d2 = 0;
41      d3 = 0;
42      d4 = 0;
43      d5 = 0;
44      d6 = 1;
45      d7 = 0;
46      x=0;
47      y=0;
48      z=1;
49      w=0;

```


Hiển thị dạng sóng ngõ ra :



BÀI CHUẨN BỊ 5

Mục tiêu: Nắm được cách thức thiết kế các IC chức năng bằng ngôn ngữ SystemVerilog và ứng dụng trên kit DE2

Yêu cầu:

- Cho hàm boolean $f(x, y, z) = \sum(0, 2, 5, 7)$

- Sinh viên sử dụng IC 74LS138 và các cổng logic cần thiết để thiết kế mạch thực hiện hàm boolean đã cho.
- Sinh viên viết đoạn mã SystemVerilog thực hiện hoạt động của hàm $f(x, y, z)$ như **hướng dẫn** từ Prelab với sơ đồ gán chân như sau:
 - Chân x, y, z gán tới SW2, SW1, SW0 và LEDR2, LEDR1, LEDR0 tương ứng
 - Chân f gán tới LEDG0
 - Trong đó, gọi IC 74LS138 như một module con
- Sinh viên tiến hành biên dịch đoạn chương trình trên. Sau đó, sinh viên tiến mô phỏng dạng sóng ngõ ra trong tất cả các trường hợp của ngõ vào. Chụp hình dạng sóng ngõ ra.

File design.sv:

```

LAB_2 > lab2_tn5 > design sv
1  module ic138(
2      input logic g,
3      input logic g2a,g1a,
4      input logic x,y,z,
5      output logic [7:0] y_o,
6      output logic y_out);
7      reg [2:0] temp;
8      assign combine = {x,y,z};
9      assign temp ={g,g1a,g2a};
10     always_comb
11     begin
12         if (temp == 3'b100)
13         begin
14             case (combine)
15                 3'b111: y_o[7] = 1'b0;
16                 3'b110: y_o[6] = 1'b0;
17                 3'b101: y_o[5] = 1'b0;
18                 3'b100: y_o[4] = 1'b0;
19                 3'b011: y_o[3] = 1'b0;
20                 3'b010: y_o[2] = 1'b0;
21                 3'b001: y_o[1] = 1'b0;
22                 3'b000: y_o[0] = 1'b0;
23             default: y_o = 1'b0;
24         endcase
25         end
26     else
27         y_o = 1'b0;
28     end
29     assign y_out = ~(y_o[0]&y_o[2]&y_o[5]&y_o[7]);
30 endmodule

```

File testbench.sv:

```

`timescale 1ns/1ns
module testbench;
    reg g;
    reg g2a;
    reg g1a;
    reg x,y,z;
    wire y_o;
    wire y_out;
    ic138 dut(.g(g),.g2a(g2a),.g1a(g1a),.x(x),
              .y(y),.z(z),.y_o(y_o),.y_out(y_out));

```

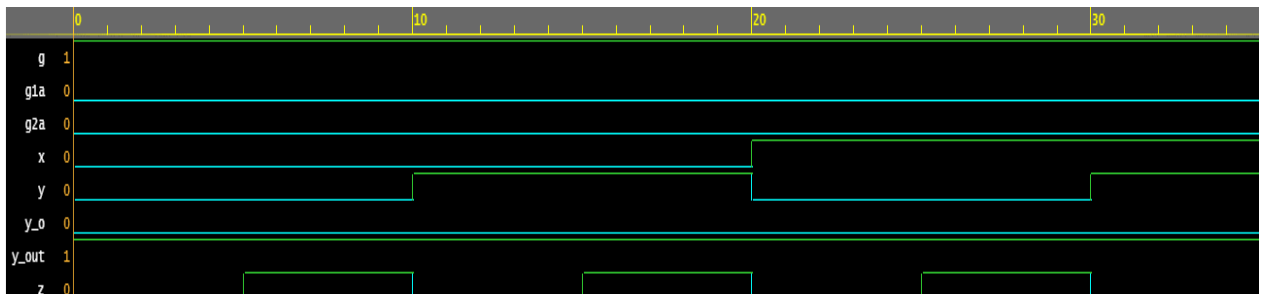
```

initial begin
    $dumpfile("dump.vcd");
    $dumpvars(1);
    g = 1; g2a = 0; g1a = 0;
    x = 0; y = 0; z = 0;
    #5
    g = 1; g2a = 0; g1a = 0;
    x = 0; y = 0; z = 1;
    #5
    g = 1; g2a = 0; g1a = 0;
    x = 0; y = 1; z = 0;
    #5
    g = 1; g2a = 0; g1a = 0;
    x = 0; y = 1; z = 1;
    #5
    g = 1; g2a = 0; g1a = 0;
    x = 1; y = 0; z = 0;
    #5
    g = 1; g2a = 0; g1a = 0;
    x = 1; y = 0; z = 1;
    #5
    g = 1; g2a = 0; g1a = 0;
    x = 1; y = 1; z = 0;
    #5
    g = 1; g2a = 0; g1a = 0;
    x = 1; y = 1; z = 1;

    #5
    $finish;
end
endmodule

```

Hiện thị dạng sóng đầu ra:



BÀI CHUẨN BỊ 6

Mục tiêu: Nắm được cách thức thiết kế các IC chức năng bằng ngôn ngữ SystemVerilog và ứng dụng trên kit DE2

Yêu cầu:

- Cho hàm boolean $f(x, y, z, w) = \sum(1, 3, 5, 7, 10, 12, 14)$
- Sinh viên sử dụng IC 74LS138 và các cổng logic cần thiết để thiết kế mạch thực hiện hàm boolean đã cho.
- Sinh viên viết đoạn mã SystemVerilog thực hiện hoạt động của hàm $f(x, y, z, w)$ như **hướng dẫn** từ Prelab với sơ đồ gán chân như sau:
 - Chân x, y, z, w gán tới SW3, SW2, SW1, SW0 và LEDR3, LEDR2, LEDR1, LEDR0 tương ứng
 - Chân f gán tới LEDG0
 - Trong đó, gọi IC 74LS138 như một module con
- Sinh viên tiến hành biên dịch đoạn chương trình trên. Sau đó, sinh viên tiến mô phỏng dạng sóng ngõ ra trong tất cả các trường hợp của ngõ vào. Chụp hình dạng sóng ngõ ra.

$$\begin{aligned} f(x, y, z, w) &= m_0(x, y, z)w + m_1(x, y, z)w + m_2(x, y, z)w + m_3(x, y, z)w + m_5(x, y, z)\bar{w} + m_7(x, y, z)\bar{w} \\ &= w(m_0 + m_1 + m_2 + m_3) + \bar{w}(m_5 + m_7) \end{aligned}$$

File design.sv:

LAB_2 > lab2_tn6 > design.sv

```
1  module ic138(  
2      input logic g,  
3      input logic g2a,g1a,  
4      input logic x,y,z,w,  
5      output logic [7:0] y_o,  
6      output logic y_out);  
7      reg [2:0] temp;  
8      assign combine = {x,y,z};  
9      assign temp = {g,g1a,g2a};  
10     assign y_0 = 8'b11111111;  
11     always_comb  
12     begin  
13         if (temp == 3'b100)  
14         begin  
15             case (combine)  
16                 3'b111: y_o[7] = 1'b0;  
17                 3'b110: y_o[6] = 1'b0;  
18                 3'b101: y_o[5] = 1'b0;  
19                 3'b100: y_o[4] = 1'b0;  
20                 3'b011: y_o[3] = 1'b0;  
21                 3'b010: y_o[2] = 1'b0;  
22                 3'b001: y_o[1] = 1'b0;  
23                 3'b000: y_o[0] = 1'b0;  
24                 default: y_o = 1'b0;  
25             endcase  
26         end  
27         else  
28             y_o = 1'b0;  
29         end  
30         assign y_out = (w&y_o[1])|(w&y_o[2])|(w&y_o[3])|(!w&y_o[5])|(!w&y_o[7]);  
31     endmodule
```

File testbench:

```
`timescale 1ns/1ns  
module testbench;  
    reg g;  
    reg g2a;  
    reg g1a;  
    reg x,y,z,w;  
    wire y_o;  
    wire y_out;  
    ic138 dut(.g(g),.g2a(g2a),.g1a(g1a),.x(x),  
              .y(y),.z(z),.w(w),.y_o(y_o),.y_out(y_out));  
    initial begin  
        $dumpfile("dump.vcd");  
        $dumpvars(1);  
        g = 1; g2a = 0; g1a = 0;
```

```

x = 0; y = 0; z = 0; w = 0;
#5
g = 1; g2a = 0; g1a = 0;
x = 0; y = 0; z = 0; w = 1;
#5
g = 1; g2a = 0; g1a = 0;
x = 0; y = 0; z = 1; w = 0;
#5
g = 1; g2a = 0; g1a = 0;
x = 0; y = 0; z = 1; w = 1;
#5
g = 1; g2a = 0; g1a = 0;
x = 0; y = 1; z = 0; w = 0;
#5
g = 1; g2a = 0; g1a = 0;
x = 0; y = 1; z = 0; w = 1;
#5
g = 1; g2a = 0; g1a = 0;
x = 0; y = 1; z = 1; w = 0;
#5
g = 1; g2a = 0; g1a = 0;
x = 0; y = 1; z = 1; w = 1;
#5
g = 1; g2a = 0; g1a = 0;
x = 1; y = 0; z = 0; w = 0;
#5
g = 1; g2a = 0; g1a = 0;
x = 1; y = 0; z = 0; w = 1;
#5
g = 1; g2a = 0; g1a = 0;
x = 1; y = 0; z = 1; w = 0;
#5
g = 1; g2a = 0; g1a = 0;
x = 1; y = 0; z = 1; w = 1;
#5
g = 1; g2a = 0; g1a = 0;
x = 1; y = 1; z = 0; w = 0;
#5
g = 1; g2a = 0; g1a = 0;
x = 1; y = 1; z = 0; w = 1;
#5
g = 1; g2a = 0; g1a = 0;
x = 1; y = 1; z = 1; w = 0;
#5
g = 1; g2a = 0; g1a = 0;
x = 1; y = 1; z = 1; w = 1;
#5
$finish;
end

```

```
endmodule
```

Hình ảnh dạng sóng ngõ ra:

