



長安大學

数字逻辑实验报告

课程名称： 数字逻辑

实验名称： 基于数据流和行为级建模的组合逻辑电路

专业名称： 计算机科学与技术/卓越工程师

小组成员： 刘文越、王子卓、孙翔、康旭

指导教师： 马峻岩

一、实验目的

1. 掌握 Basys3 开发板 4 位七段 LED 数码管显示原理；
2. 基于数据流建模方式实现 4 位二进制到七段 LED 数码译码；
3. 基于行为级描述实现 4 位二进制到七段 LED 数码译码；
4. 掌握使用多个模块实现复杂的组合逻辑功能。

主要仪器和设备：计算机，Basys3 开发板。

二、实验要求

1. Verilog 实现 16 进制-数码管译码（数据流级）

（1）功能要求

- ① sw0、sw1、sw2、sw3 位 4 位二进制输入，sw0 为最低位，sw3 为最高位。
- ② 当 sw4-sw0 位设置后，四位数码管同时点亮并显示对应的 16 进制数。

（2）其他要求

① 使用数据流级建模方式实现；使用 Logism 的组合逻辑分析工具对数码管逻辑表达式进行化简。

- ② 要求撰写仿真程序，对代码进行仿真测试。
- ③ 将仿真后的 Verilog 代码进行综合与实现，并下载到 Basys3 上验证。

2. Verilog 实现 16 进制-数码管译码（行为级）

（1）功能要求

同“Verilog 实现 16 进制-数码管译码（数据流级）”。

（2）其他要求

- ① 使用行为级建模方式实现。
- ② 要求撰写仿真程序，对代码进行仿真测试。
- ③ 将仿真后的 Verilog 代码进行综合与实现，并下载到 Basys3 上验证。

3. Verilog 实现复用的数码管显示电路

（1）功能要求

① sw0~sw3，sw4~sw7，sw8~sw11，sw12~sw15 为四组四位二进制数，数字编号小的为低位，编号大的为高位。

② 当 btnU 按下时，sw0~sw3 的数值在第 0 个数码管中显示，其他 3 个数码管熄灭；当 btnL 按下时，sw4~sw7 的数值在第 1 个数码管中显示，其他 3 个数码管熄灭；

btnD 按下，sw8~sw11 的数值在第 2 个数码管中显示，其他 3 个数码管熄灭；btnR 按下，sw12~sw15 的数值在第 3 个数码管中显示，其他 3 个数码管熄灭。

③ 实例化 1 个 16 进制-数码管译码模块。

④ 实现 1 个 N 位宽 4 选一数据选择器，默认位宽为 4；通过 4 位 4 选一多路选择器实现四组 sw 到对十六进制-数码管译码器的复用。

⑤ 实现 4 线-2 线优先编码器（最大值输入优先），2 位编码器输出连接到④中 4 选一数据选择器的数据选择，btnU、btnL、btnD、btnR 分别连接编码的 0，1，2，3 输入端。

⑥ 实现一个 2 线-4 线译码器，低电平输出有效，译码器输出连接 Basys3 数码管的位选位，输入与⑤中的 4 线-2 线优先编码器输出相连。

(2) 其他要求

① 使用 Logisim 自带模块，在给定的 logisim_basys3.circ 上实现上述功能。

② 要求撰写仿真程序，对代码进行仿真测试。

③ 将仿真后的 Verilog 代码进行综合与实现，并下载到 Basys3 上验证。

三、相关外设接口及原理

1. Basys3 开发板拨码开关原理图及简介

拨码开关的电路如图 3-1 所示。当开关打到下档时，FPGA 对应引脚输入为低电平；当开关打到上档时，FPGA 对应引脚输入为低电平。

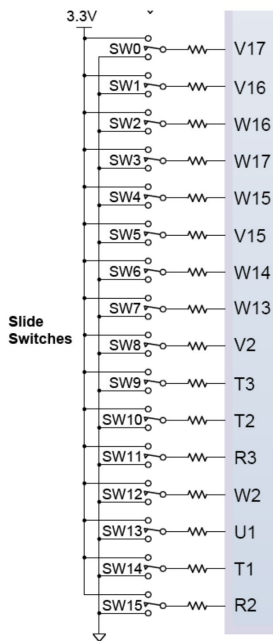


图 3-1 板拨码开关原理图

2. Basys3 开发板数码管电路原理图及简介

数码管显示部分的电路如图 3-2 所示。这里是四位带小数点的七段共阳数码管，当相应的输出脚为低电平时，该段位的 LED 点亮。位选位也是低电平选通。图 3-3 位数码管的十六进制数表示。表 3-1 为共阳极数码管十六进制显示真值表。

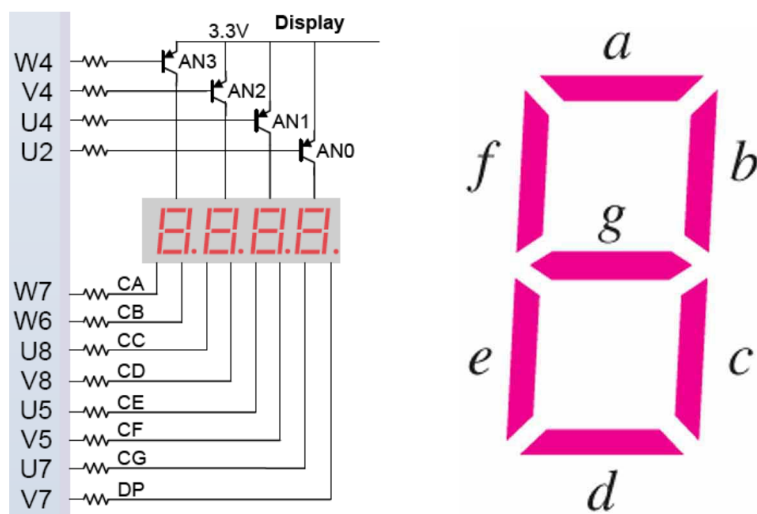


图 3-2 四位数码管电路原理图



图 3-3 数码管十六进制数表示

表 3-1 共阳极数码管十六进制显示真值表

X	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
B	1	1	0	0	0	0	0

C	0	1	1	0	0	0	1
D	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

3. Basys3 开发板按键电路原理图及简介

按键部分的电路如图 3-4 所示。Basys3 开发板上配有 5 个按键，当按键按下时 FPGA 相应输入脚为高电平，否则为低电平。

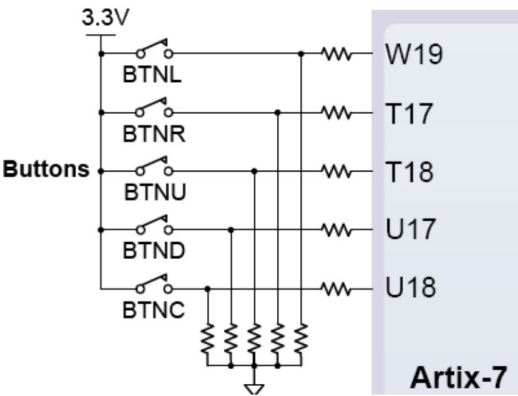


图 3-4 按键电路原理图

四、实验步骤及关键代码

1. Verilog 实现 16 进制-数码管译码（数据流级）

(1) 7 段数码管译码输入输出信号真值表

组合逻辑电路分析										
文件 编辑 工程 电路仿真 窗口 帮助										
输入				输出						
真值表				表达式						
最小项										
sw3	sw2	sw1	sw0	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0
1	0	1	0	0	0	0	1	0	0	0
1	0	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	0	0	0	0	1	0
1	1	1	0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	1	0	0	0

图 4-1 7 段数码管译码输入输出信号真值表

(2) Logisim 组合逻辑分析工具的卡诺图化简

- $a = \sim sw3 \sim sw2 \sim sw1 sw0 + \sim sw3 sw2 \sim sw1 \sim sw0 + sw3 \sim sw2 sw1 sw0 + sw3 sw2 \sim sw1 sw0$
- $b = \sim sw3 sw2 \sim sw1 sw0 + sw2 sw1 \sim sw0 + sw3 sw1 sw0 + sw3 sw2 \sim sw0$
- $c = \sim sw3 \sim sw2 sw1 \sim sw0 + sw3 sw2 \sim sw0 + sw3 sw2 sw1$
- $d = \sim sw3 \sim sw2 \sim sw1 sw0 + \sim sw3 sw2 \sim sw1 \sim sw0 + sw2 sw1 sw0 + sw3 \sim sw2 sw1 \sim sw0$
- $e = \sim sw3 sw0 + \sim sw2 \sim sw1 sw0 + \sim sw3 sw2 \sim sw1$
- $f = \sim sw3 \sim sw2 sw0 + \sim sw3 \sim sw2 sw1 + \sim sw3 sw1 sw0 + sw3 sw2 \sim sw1 sw0$
- $g = \sim sw3 \sim sw2 \sim sw1 + \sim sw3 sw2 sw1 sw0 + sw3 sw2 \sim sw1 \sim sw0$

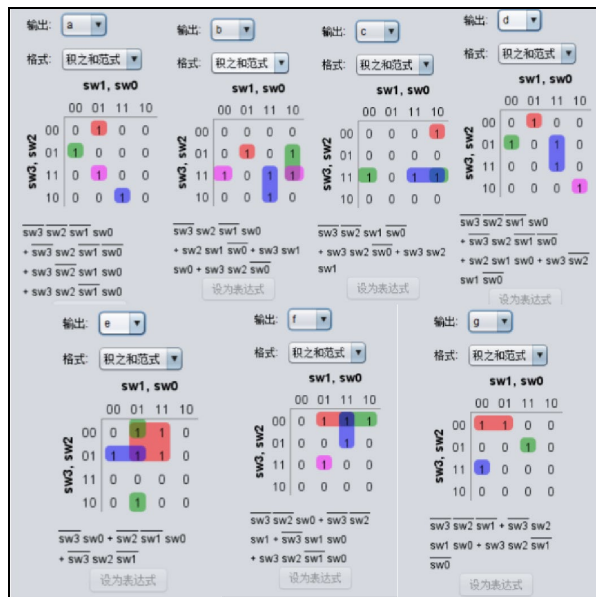


图 4-2 Logisim 组合逻辑分析卡诺图化简

(3) 主要模块代码

① 设计文件:

hex_7_Dataflow.v

```
`timescale 1ns / 1ps

module hex_7_Dataflow(
    input [3: 0] D,
    output [6: 0] seg
);

    assign seg[6] = (~D[3] & ~D[2] & ~D[1] & D[0]) | (~D[3] & D[2] & ~D[1]
& ~D[0]) | (D[3] & ~D[2] & D[1] & D[0]) | (D[3] & D[2] & ~D[1] & D[0]);
```

```

    assign seg[5] = (~D[3] & D[2] & ~D[1] & D[0]) | (D[2] & D[1] & ~D[0])
    | (D[3] & D[1] & D[0]) | (D[3] & D[2] & ~D[0]);
    assign seg[4] = (~D[3] & ~D[2] & D[1] & ~D[0]) | (D[3] & D[2] & ~D[0])
    | (D[3] & D[2] & D[1]);
    assign seg[3] = (~D[3] & ~D[2] & ~D[1] & D[0]) | (~D[3] & D[2] & ~D[1]
    & ~D[0]) | (D[2] & D[1] & D[0]) | (D[3] & ~D[2] & D[1] & ~D[0]);
    assign seg[2] = (~D[3] & D[0]) | (~D[2] & ~D[1] & D[0]) | (~D[3] & D[2]
    & ~D[1]);
    assign seg[1] = (~D[3] & ~D[2] & D[0]) | (~D[3] & ~D[2] & D[1]) | (~D[3]
    & D[1] & D[0]) | (D[3] & D[2] & ~D[1] & D[0]);
    assign seg[0] = (~D[3] & ~D[2] & ~D[1]) | (~D[3] & D[2] & D[1] & D[0])
    | (D[3] & D[2] & ~D[1] & ~D[0]);

endmodule

```

② 顶层文件:

Hex_7_Dataflow_top.v

```

`timescale 1ns / 1ps

module hex_7_Dataflow_top(
    input [3: 0] sw,
    output [0: 6] seg,
    output [3: 0] an,
    output dp
);

    assign an = 4'b0000;
    assign dp = 1;
    hex_7_Dataflow(.D(sw), .seg(seg));

endmodule

```

(4) 仿真测试代码

hex_7_Dataflow_simulation.v

```

`timescale 1ns / 1ps

module hex_7_Dataflow_simulation(
);
    reg [3: 0] in;
    wire [6: 0] out;

    hex_7_Dataflow h7d(.D(in), .seg(out));

```

```
initial begin
    #0 in = 0;
    #10 in = 1;
    #10 in = 2;
    #10 in = 3;
    #10 in = 4;
    #10 in = 5;
    #10 in = 6;
    #10 in = 7;
    #10 in = 8;
    #10 in = 9;
    #10 in = 10;
    #10 in = 11;
    #10 in = 12;
    #10 in = 13;
    #10 in = 14;
    #10 in = 15;
    #10 $finish;
end

endmodule
```

(5) 仿真波形图

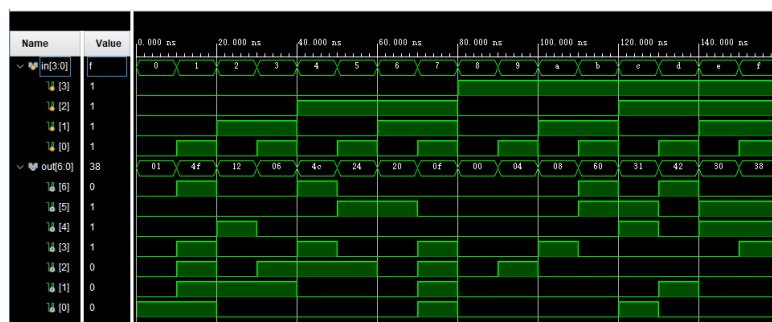


图 4-3 7 段数码管译码仿真波形图

(6) 开发板测试照片

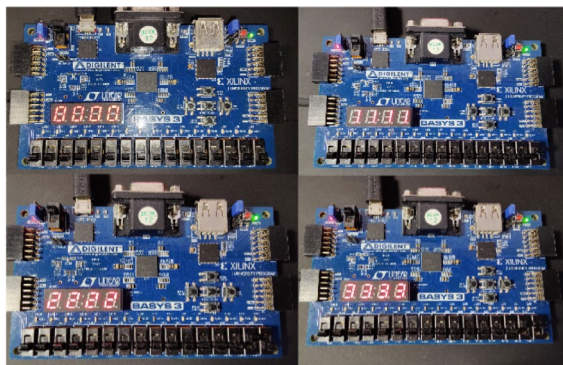


图 4-4 sw[3:0]为“0000”“0001”“0010”“0011”时开发板照片

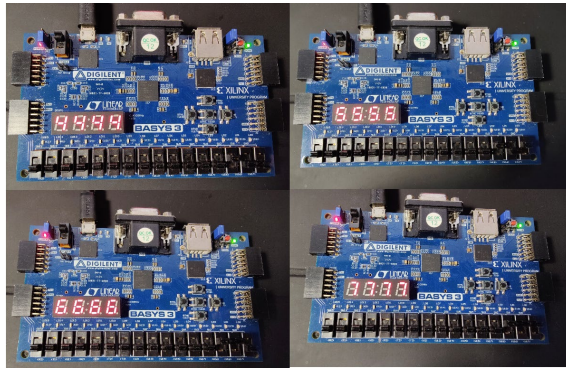


图 4-5 sw[3:0]为“0100”“0101”“0110”“0111”时开发板照片

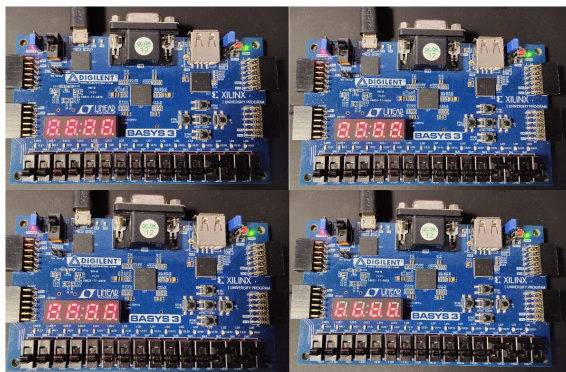


图 4-6 sw[3:0]为“1000”“1001”“1010”“1011”时开发板照片

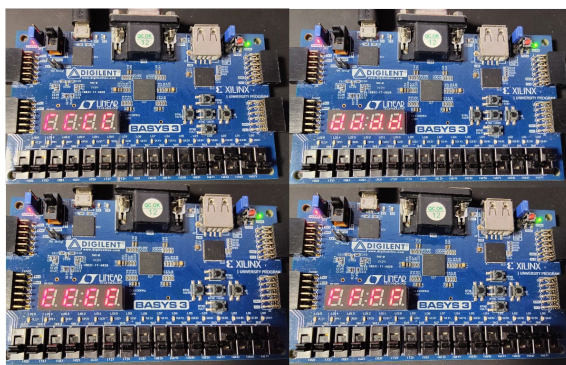


图 4-7 sw[3:0]为“1100”“1101”“1110”“1111”时开发板照片

2. Verilog 实现 16 进制-数码管译码（行为级）

（1）主要模块代码

① 设计文件：

hex_7_Behavioural.v

```
`timescale 1ns / 1ps

module hex_7_Behavioural(
    input [3: 0] D,
    output reg [6: 0] seg
);
    always @(*) begin
        case(D)
```

```

        0: seg = 7'b0000001;
        1: seg = 7'b1001111;
        2: seg = 7'b0010010;
        3: seg = 7'b0000110;
        4: seg = 7'b1001100;
        5: seg = 7'b0100100;
        6: seg = 7'b0100000;
        7: seg = 7'b0001111;
        8: seg = 7'b0000000;
        9: seg = 7'b0000100;
        10: seg = 7'b0001000;
        11: seg = 7'b1100000;
        12: seg = 7'b0110001;
        13: seg = 7'b1000010;
        14: seg = 7'b0110000;
        15: seg = 7'b0111000;
    endcase
end
endmodule

```

② 顶层文件:

hex_7_Behavioural_top.v

```

`timescale 1ns / 1ps

module hex_7_Dataflow_top(
    input [3: 0] sw,
    output [0: 6] seg,
    output [3: 0] an,
    output dp
);

    assign an = 4'b0000;
    assign dp = 1;
    hex_7_Dataflow(.D(sw), .seg(seg));

endmodule

```

(2) 仿真测试代码

hex_7_Behavioural_simulation.v

```

`timescale 1ns / 1ps

module hex_7_Behavioural_simulation(
);

```

```
reg [3: 0] in;
wire [6: 0] out;

hex_7_Behavioural h7d(.D(in), .seg(out));

initial begin
    #0 in = 0;
    #10 in = 1;
    #10 in = 2;
    #10 in = 3;
    #10 in = 4;
    #10 in = 5;
    #10 in = 6;
    #10 in = 7;
    #10 in = 8;
    #10 in = 9;
    #10 in = 10;
    #10 in = 11;
    #10 in = 12;
    #10 in = 13;
    #10 in = 14;
    #10 in = 15;
    #10 $finish;
end

endmodule
```

(3) 仿真波形图



图 4-8 7 段数码管译码器仿真波形图

(4) 开发板测试照片

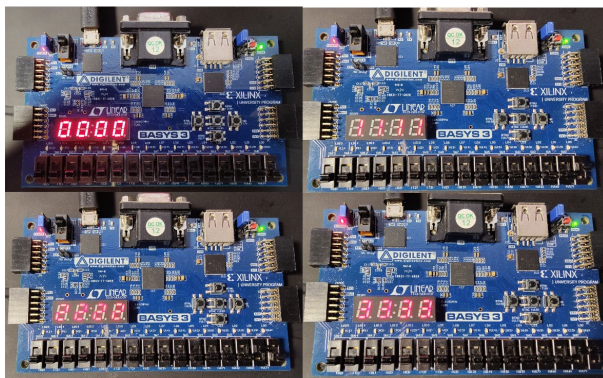


图 4-9 sw[3:0]为“0000”“0001”“0010”“0011”时开发板照片

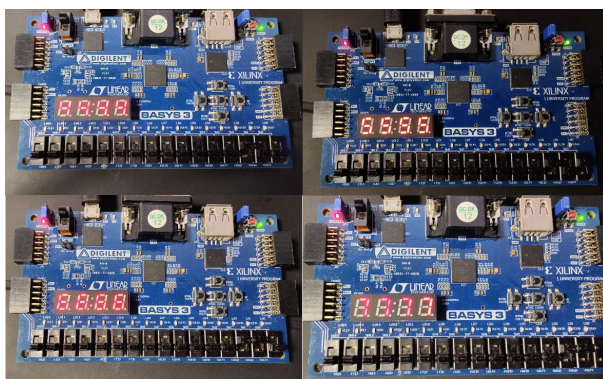


图 4-10 sw[3:0]为“0100”“0101”“0110”“0111”时开发板照片

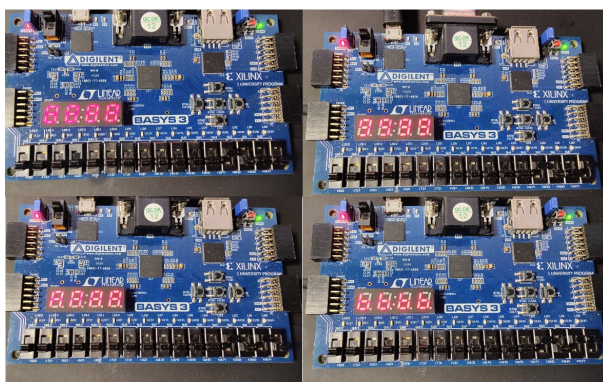


图 4-11 sw[3:0]为“1000”“1001”“1010”“1011”时开发板照片

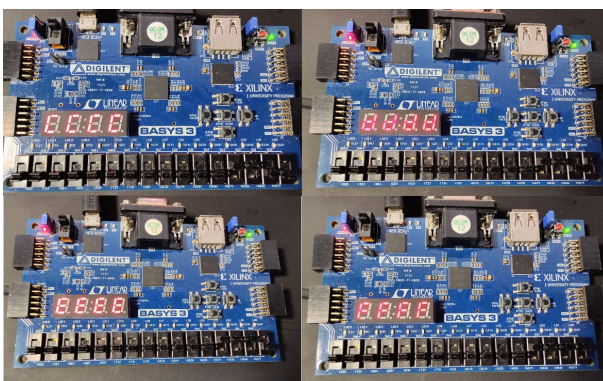


图 4-12 sw[3:0]为“1100”“1101”“1110”“1111”时开发板照片

3. Verilog 实现复用的数码管显示电路

(1) 绘制各模块连接的逻辑框图

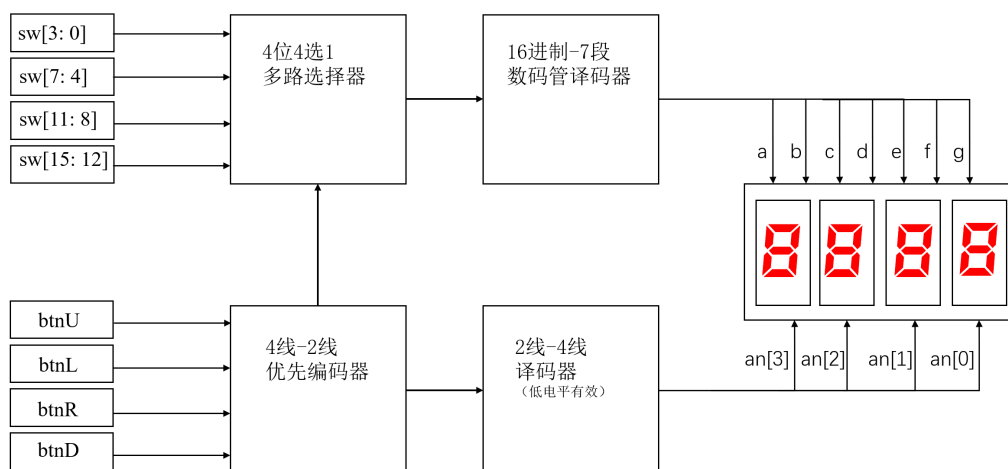


图 4-13 复用四选一 7 段数码管显示电路逻辑框图

(2) 主要模块代码

① 设计文件:

mux41_hex7.v

```
`timescale 1ns / 1ps

module mux41_hex7(
    input [15: 0] x,
    input [3: 0] btn,
    output [6: 0] seg,
    output [3: 0] an
);

    wire [1: 0] sel;
    wire [3: 0] y;
    wire inv;

    priority_encoder_42 pe42(.D(btn), .Y(sel), .inv(inv));
    decoder_24 d24(.D(sel), .inv(inv), .Y(an));
    mux41 m41(.SEL(sel), .D0(x[3: 0]), .D1(x[7: 4]), .D2(x[11: 8]), .D3(x[15: 12]), .Y(y));
    hex_7 h7(.D(y), .seg(seg));

endmodule
```

mux41.v

```
`timescale 1ns / 1ps
```



```

module mux41(SEL, D0, D1, D2, D3, Y);
    parameter n = 4;
    input [1: 0] SEL;
    input [n - 1: 0] D0, D1, D2, D3;
    output reg [n - 1: 0] Y;

    always @(*) begin
        case(SEL)
            2'b00: Y = D0;
            2'b01: Y = D1;
            2'b10: Y = D2;
            2'b11: Y = D3;
        endcase
    end
endmodule

```

hex_7.v

```

`timescale 1ns / 1ps

module hex_7(
    input [3: 0] D,
    output reg [6: 0] seg
);

    always @(*) begin
        case(D)
            0: seg = 7'b0000001;
            1: seg = 7'b1001111;
            2: seg = 7'b0010010;
            3: seg = 7'b0000110;
            4: seg = 7'b1001100;
            5: seg = 7'b0100100;
            6: seg = 7'b0100000;
            7: seg = 7'b0001111;
            8: seg = 7'b0000000;
            9: seg = 7'b0000100;
            10: seg = 7'b0001000;
            11: seg = 7'b1100000;
            12: seg = 7'b0110001;
            13: seg = 7'b1000010;
            14: seg = 7'b0110000;
            15: seg = 7'b0111000;
        endcase
    end
end

```

```
endmodule
```

priority_encoder_42.v

```
`timescale 1ns / 1ps

module priority_encoder_42(
    input [3: 0] D,
    output reg [1: 0] Y,
    output inv
);

    assign inv = D[0] | D[1] | D[2] | D[3];

    always @(*) begin
        casex(D)
            4'b1xxx: Y = 2'b11;
            4'bx1xx: Y = 2'b10;
            4'bxx1x: Y = 2'b01;
            4'bxxx1: Y = 2'b00;
        endcase
    end
endmodule
```

decoder_24.v

```
`timescale 1ns / 1ps

module decoder_24(
    input [1: 0] D,
    input inv,
    output reg [3: 0] Y
);

    always @(*) begin
        if (inv)
            case(D)
                2'b00: Y = 4'b1110;
                2'b01: Y = 4'b1101;
                2'b10: Y = 4'b1011;
                2'b11: Y = 4'b0111;
            endcase
        else
            Y = 4'b1111;
        end
    end
endmodule
```

② 顶层文件:

mux41_hex7_top.v

```
`timescale 1ns / 1ps

module mux41_hex7_top(
    input [15: 0] sw,
    input btnU, btnL, btnR, btnD,
    output [0: 6] seg,
    output [3: 0] an,
    output dp
);

    assign dp = 1;

    mux41_hex7(
        .x(sw),
        .btn({btnD, btnR, btnL, btnU}),
        .seg(seg),
        .an(an)
    );

endmodule
```

(3) 仿真波形图

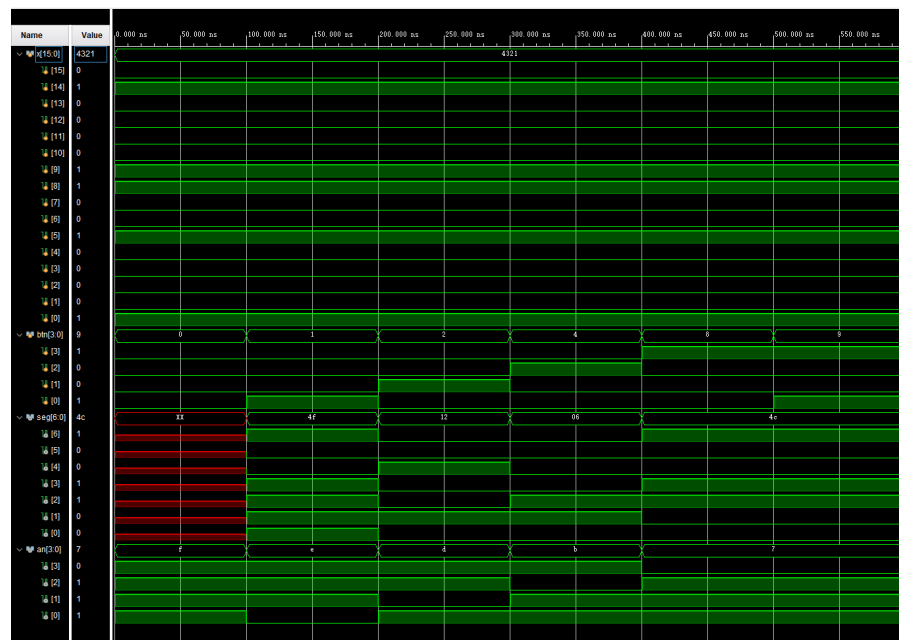


图 4-14 复用四选一 7 段数码管显示电路仿真波形图

(4) 开发板测试照片

开发板测试照片中 sw[3: 0]均为“0001”, sw[7: 4]均为“0010”, sw[11: 8]均为“0011”,

sw[15: 12]均为“0100”，以区别四个不同的输入。

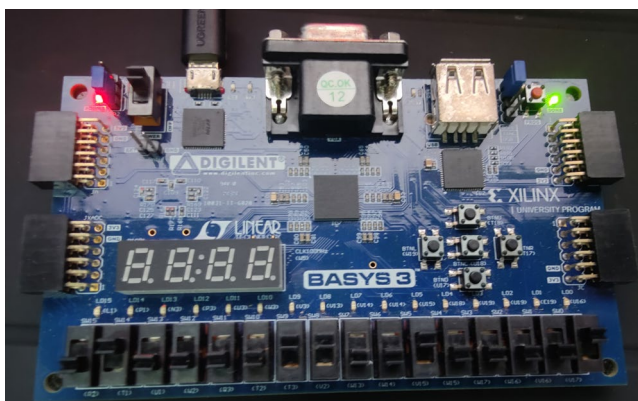


图 4-15 btnU、btnL、btnR、btnD 全为 0 时

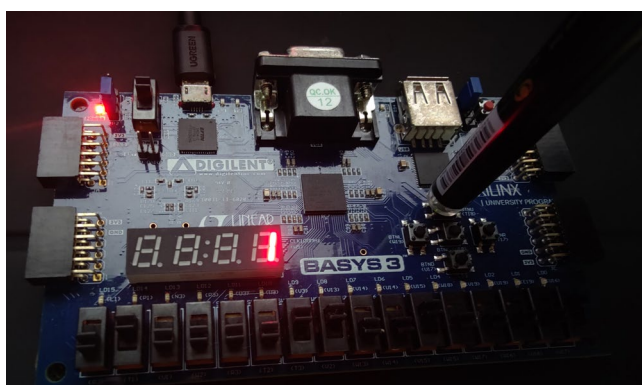


图 4-16 btnL、btnR、btnD 全为 0，btnU 为 1 时

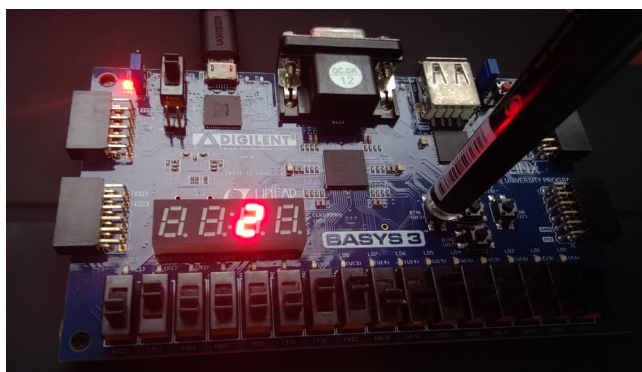


图 4-17 btnU、btnR、btnD 全为 0，btnL 为 1 时

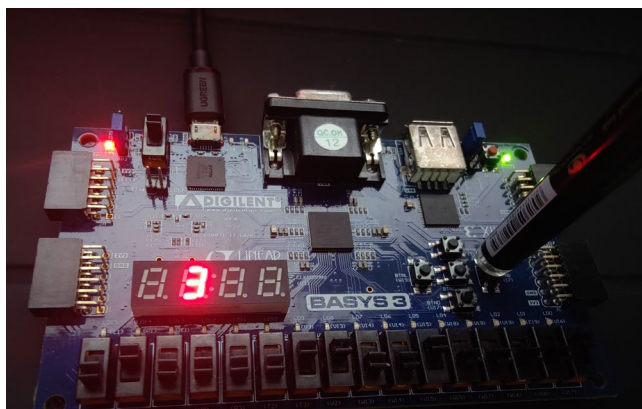


图 4-18 btnU、btnL、btnD 全为 0，btnR 为 1 时

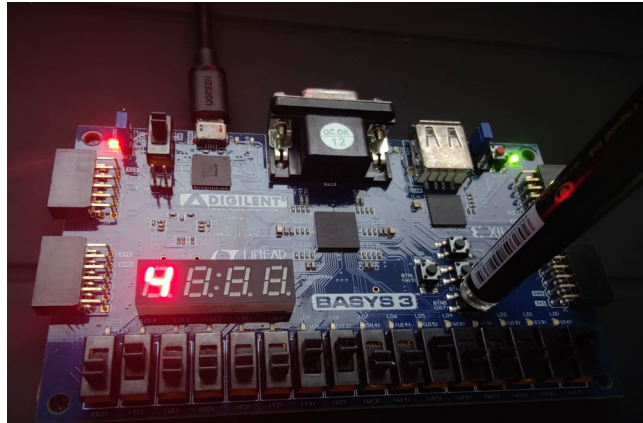


图 4-19 btnU、btnL、btnR 全为 0，btnD 为 1 时

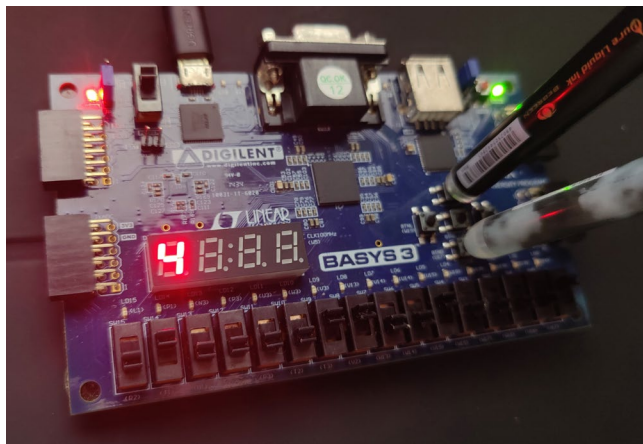


图 4-110 btnL、btnR 为 0，btnU、btnD 为 1 时