

同济大学计算机系

数字逻辑课程实验报告



学 号 2154312

姓 名 郑博远

专 业 计算机科学与技术

授课老师 郭玉臣

一、实验内容

(1) 3-8 译码器实验：译码器是能够实现译码功能的组合逻辑电路。它的输入是一组二进制代码，输出是一组高低电平信号。每输入一组不同的代码，只有一个输出呈现有效状态。本实验中 D0、D1、D2 三个输入为输入信号， $\overline{Y0} \sim \overline{Y7}$ 为输出信号，G1、G2 为使能输入端；

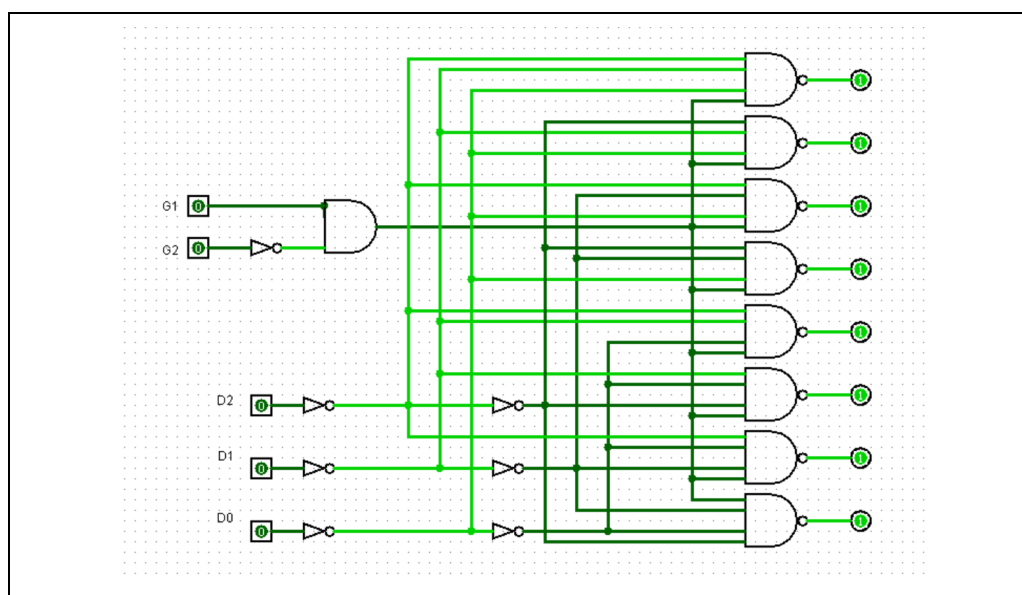
(2) 七段数码管译码驱动器实验：七段数码管译码驱动器由译码驱动器和荧光数码管组成。后者由 7 个发光二极管组成 7 个发光段，能将电能转换为光能，显示出对应的十进制数字；

(3) 普通 8-3 编码器：编码器用于完成编码功能，能够实现与译码器相反的“一对多”译码。通过 D7~D0 的 8 个输入，能对应转换为 Y2~Y0 的 3 个输出；

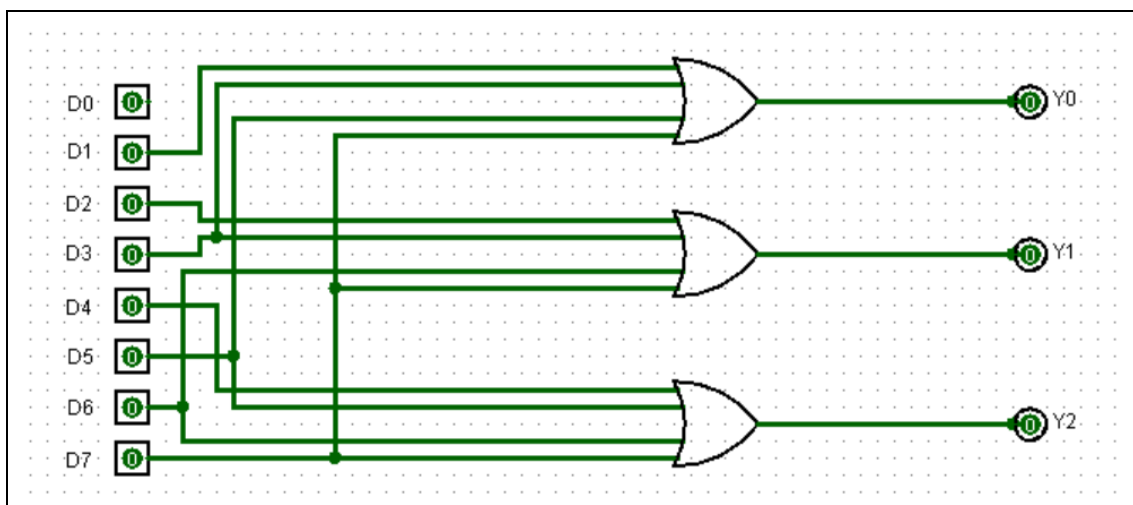
(4) 具有优先级的 8-3 编码器：普通编码器在任意时刻所以输入线中只允许有一个输入线信号有效，否则将发生混乱。具有优先级的 8-3 译码器则能够根据优先级来选择对应的输出。

二、硬件逻辑图

(1) 3-8 译码器实验:



(2) 数据分配器实验:



三、模块建模

(1) 3-8 译码器实验:

iData 为模块的输入，iEna 代表两个使能输入端，oData 为输出。以下是 Verilog 代码:

```
module decoder(  
    input [2:0] iData,  
    input [1:0] iEna,  
    output [7:0] oData  
);  
  
    assign oData = (iEna == 2'b10 ? ~(8'b00000001 <<  
        iData) : 8'b11111111);  
  
endmodule
```

(2) 七段数码管译码驱动器实验:

iData 为模块的输入；oData 为模块的输出，分别对应显示数字的 7 根发光二极管。Verilog 代码如下:

```

module display7(
    input [3:0] iData,
    output reg [6:0] oData
);

always @(*)
begin
    case(iData)
        4'b0000 : oData = 7'b1000000;
        4'b0001 : oData = 7'b1111001;
        4'b0010 : oData = 7'b0100100;
        4'b0011 : oData = 7'b0110000;
        4'b0100 : oData = 7'b0011001;
        4'b0101 : oData = 7'b0010010;
        4'b0110 : oData = 7'b0000010;
        4'b0111 : oData = 7'b1111000;
        4'b1000 : oData = 7'b0000000;
        4'b1001 : oData = 7'b0010000;
    endcase
end

endmodule

```

(3) 普通 8-3 编码器实验:

iData 与 oData 分别为模块的输入与输出。当不同的 iData 输入时, oData 中有相应的编码输出。以下是本小题的 Verilog 代码:

```

module encoder83(
    input [7:0] iData,
    output reg [2:0] oData
);

    always@ (*)
        case(iData)
            8'b10000000 : oData = 3'b111;
            8'b01000000 : oData = 3'b110;
            8'b00100000 : oData = 3'b101;
            8'b00010000 : oData = 3'b100;
            8'b00001000 : oData = 3'b011;
            8'b00000100 : oData = 3'b010;
            8'b00000010 : oData = 3'b001;
            8'b00000001 : oData = 3'b000;
        endcase

endmodule

```

(4) 具有优先级的 8-3 编码器实验:

```

module encoder83_Pri(
    input [7:0] iData,
    input iEI,
    output reg [2:0] oData,
    output reg oEO
);

    always@ (*)
        begin

```

```

        if(iEI == 1)
            {oData, oEO} = 4'b1110;
        else
            begin
                casex(iData)
                    8'b11111111 : {oData, oEO} = 4'b1110;
                    8'b0xxxxxxx : {oData, oEO} = 4'b0001;
                    8'b10xxxxxx : {oData, oEO} = 4'b0011;
                    8'b110xxxxx : {oData, oEO} = 4'b0101;
                    8'b1110xxxx : {oData, oEO} = 4'b0111;
                    8'b11110xxx : {oData, oEO} = 4'b1001;
                    8'b111110xx : {oData, oEO} = 4'b1011;
                    8'b1111110x : {oData, oEO} = 4'b1101;
                    8'b11111110 : {oData, oEO} = 4'b1111;
                endcase
            end
        end
    end

endmodule

```

四、测试模块建模

(1) 3-8 译码器实验:

```

`timescale 1ns / 1ps

module decoder_tb;
    reg [3 : 0] iData;
    reg [2 : 0] iEna;
    wire [7 : 0] oData;

```

```

        initial
        begin
            for(iEna = 0; iEna <= 2'b11; iEna = iEna +
2'b01)
                for(iData = 0; iData <= 3'b111; iData =
iData + 3'b001)
                    begin
                        #20;
                    end
                end
            end
        decoder
        decoder_instc(iData[2 : 0], iEna[1 : 0], oData);
    endmodule

```

(2) 七段数码管译码驱动器实验:

```

`timescale 1ns / 1ps

module display7_tb;
    reg [3:0] iData;
    wire[6:0] oData;

    initial
    begin
        for(iData = 0; iData <= 4'b1001; iData = iData +
4'b0001)
            #20;
        end

        display7 display7_inst(iData, oData);
    endmodule

```

(3) 普通 8-3 编码器实验:

```

`timescale 1ns / 1ps

module encoder83_tb;
    reg [8:0] iData;
    wire [2:0] oData;

    initial
    begin
        for(iData = 8'b00000001; iData <= 8'b10000000;
iData = iData << 1)
            begin
                #20;
            end
        end

        encoder83 encoder83_inst(iData[7 : 0], oData);

    endmodule

```

(4) 具有优先级的 8-3 编码器传输实验:

```

`timescale 1ns / 1ps

module encoder83_Pri_tb;
    reg [8:0] iData;
    reg iEI;
    wire [2:0] oData;
    wire oEO;

    initial
    begin
        iEI = 0;
        for(iData = 8'b01111111; iData <= 8'b11111111;
iData = iData + 8'b00000001)
            #20;
        end

        encoder83_Pri encoder83_Pri_inst(iData, iEI, oData,
oEO);

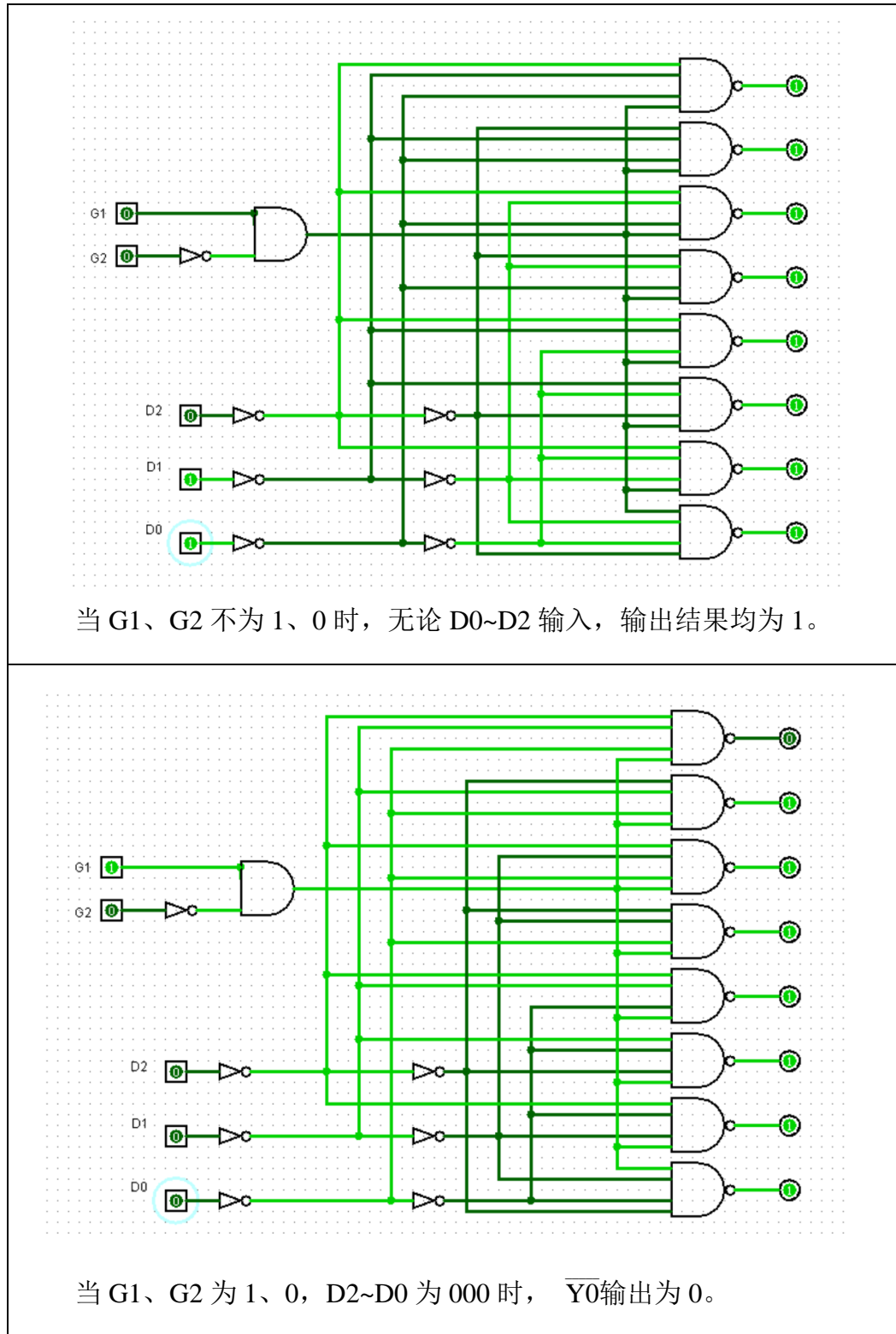
    endmodule

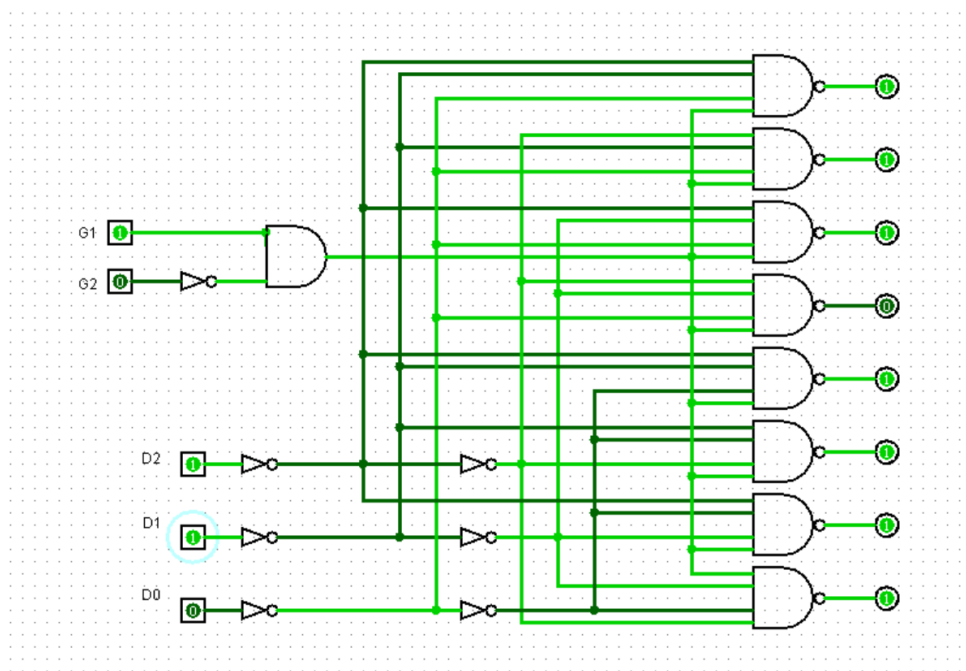
```


五、实验结果

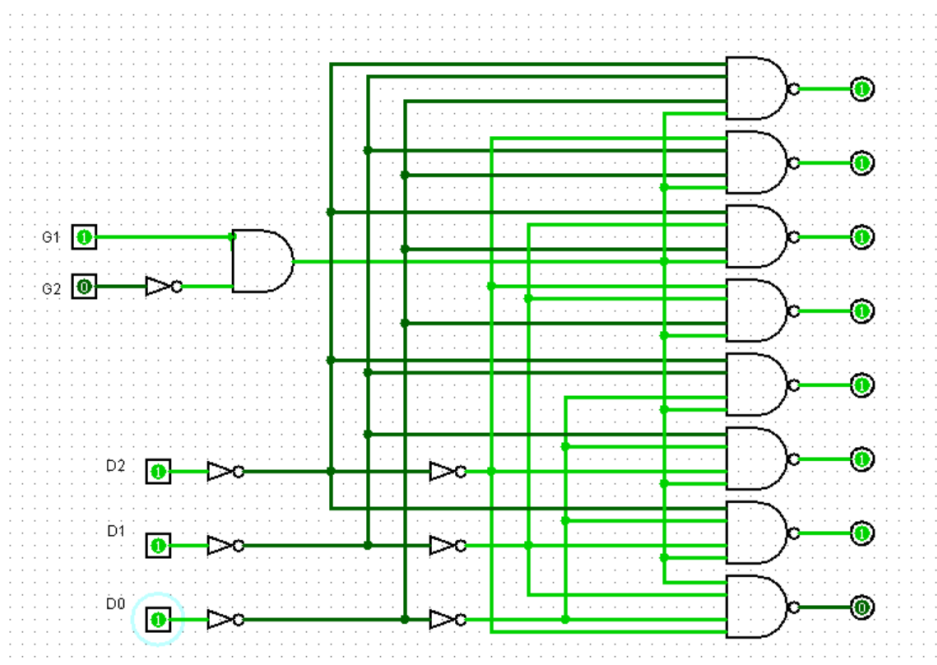
(1) 3-8 译码器电路实验:

a) logisim 逻辑验证图



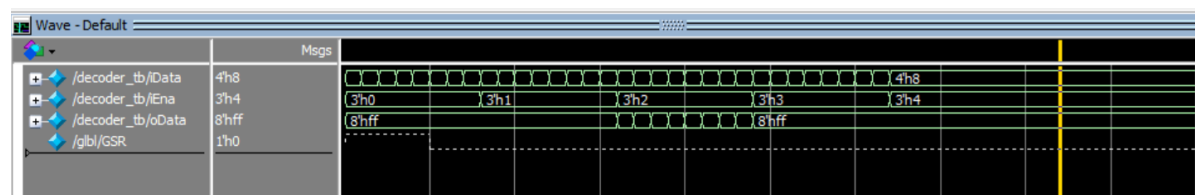


当 G1、G2 为 1、0，D2~D0 为 110 时， $\overline{Y3}$ 输出为 0。



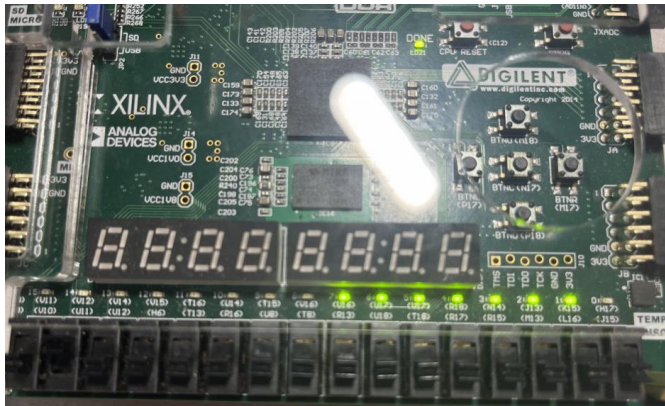
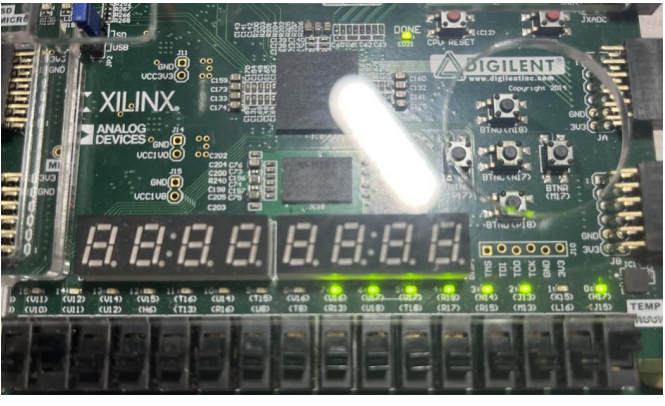
当 G1、G2 为 1、0，D2~D0 为 111 时， $\overline{Y7}$ 输出为 0。

b) modelsim 仿真波形图

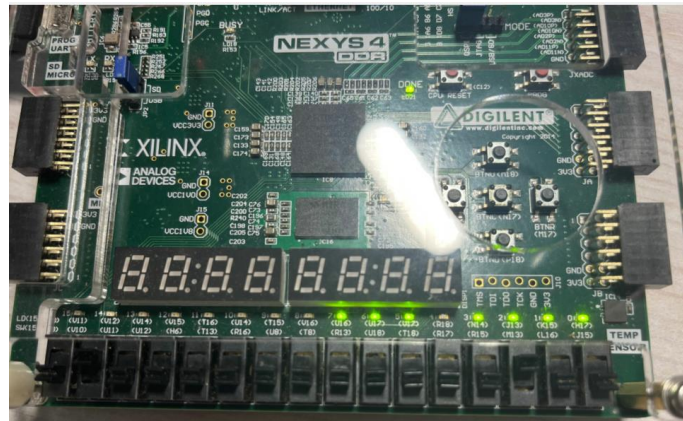


c) 下板实验结果图

V10 打开或 V11 关闭的情况下，译码器不工作，所有灯均亮起；下面测试 V10 关闭且 V11 打开的情况：

<p>M13、L16、J15 均关闭， 除 H17 外所有灯均亮起</p>	
<p>M13、L16 关闭， J15 打开， 除 K15 外所有灯均亮起</p>	

M13 打开，
L16、J15 关闭，
除 R17 外所有灯均亮起

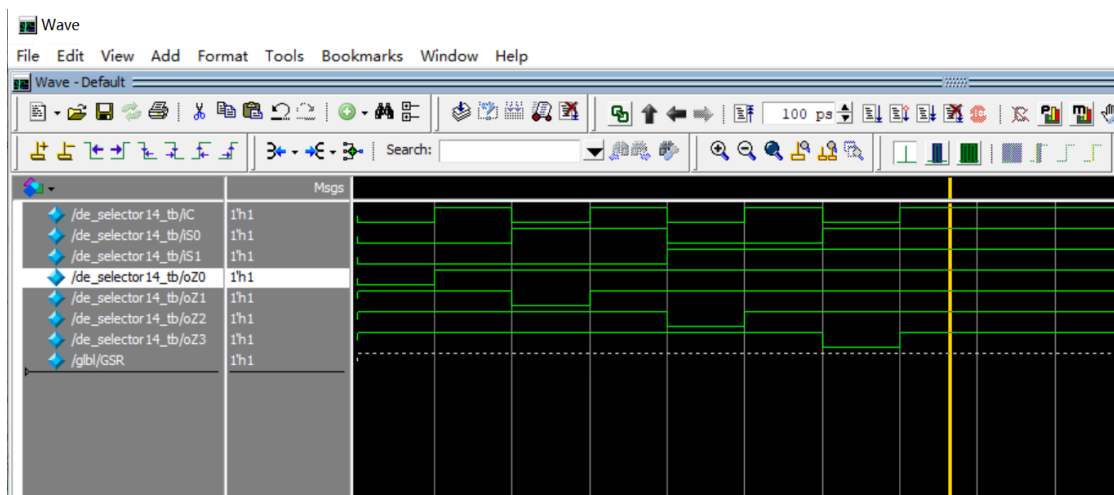


M13、L16 打开，
J15 关闭，
除 U17 外所有灯均亮起



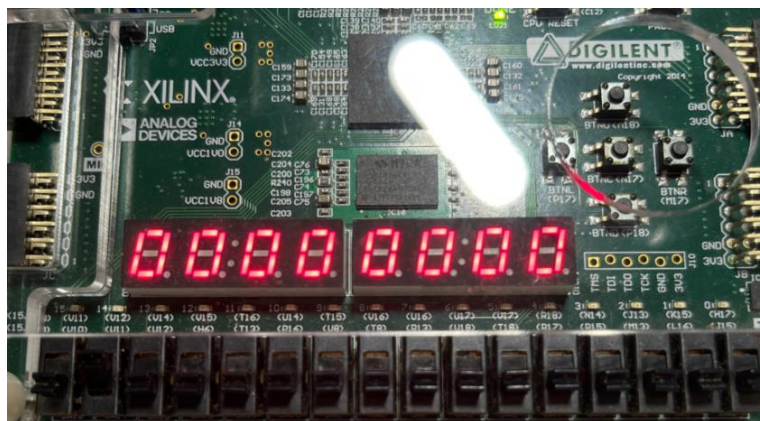
(2) 七段数码管译码驱动器实验：

a) modelsim 仿真波形图

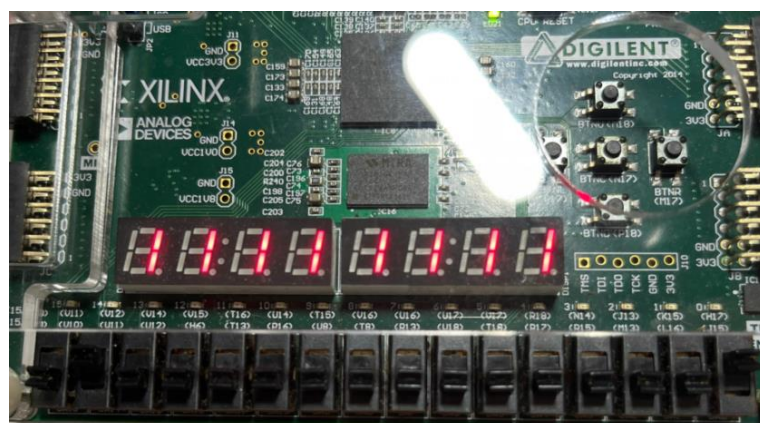


b) 下板实验结果图

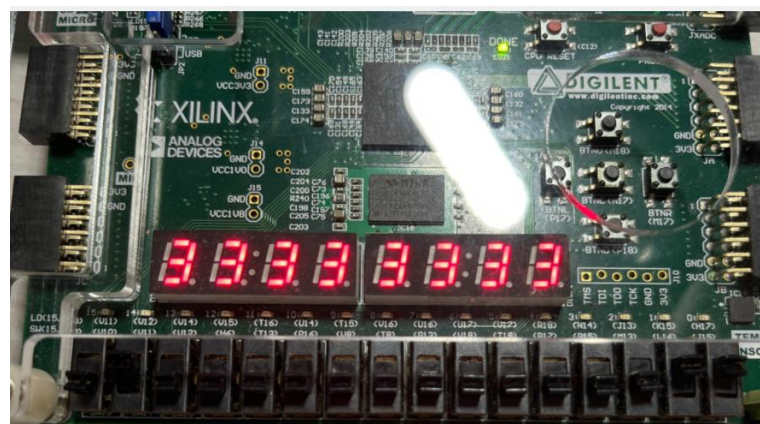
关闭所有开关，
显示数字 0



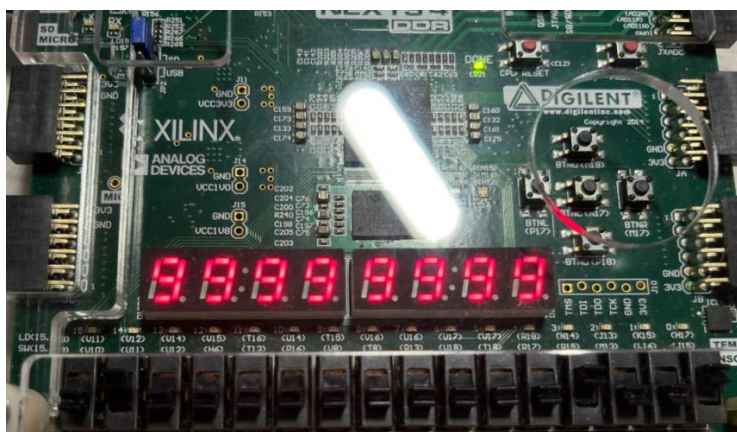
仅打开 J15，
显示数字 1



打开 L16、J15，
显示数字 3

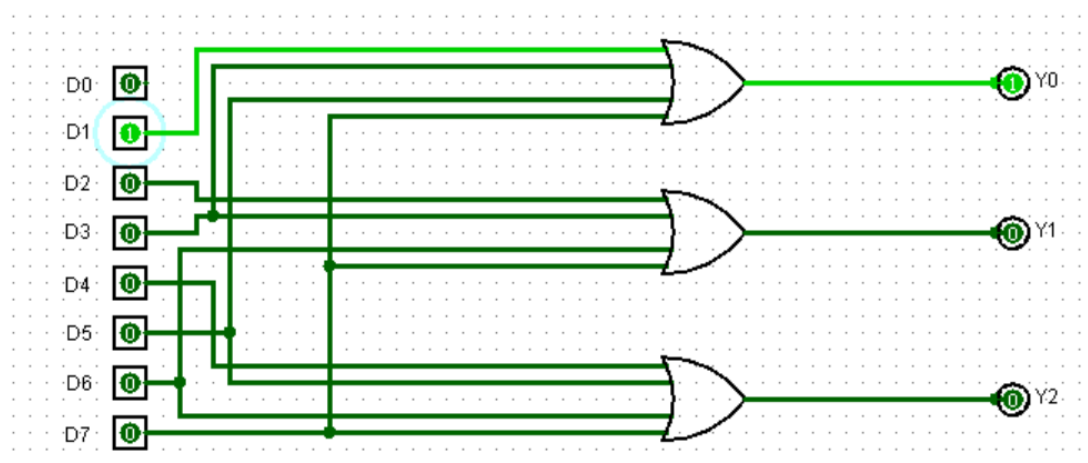


打开 R15、J15，
显示数字 9

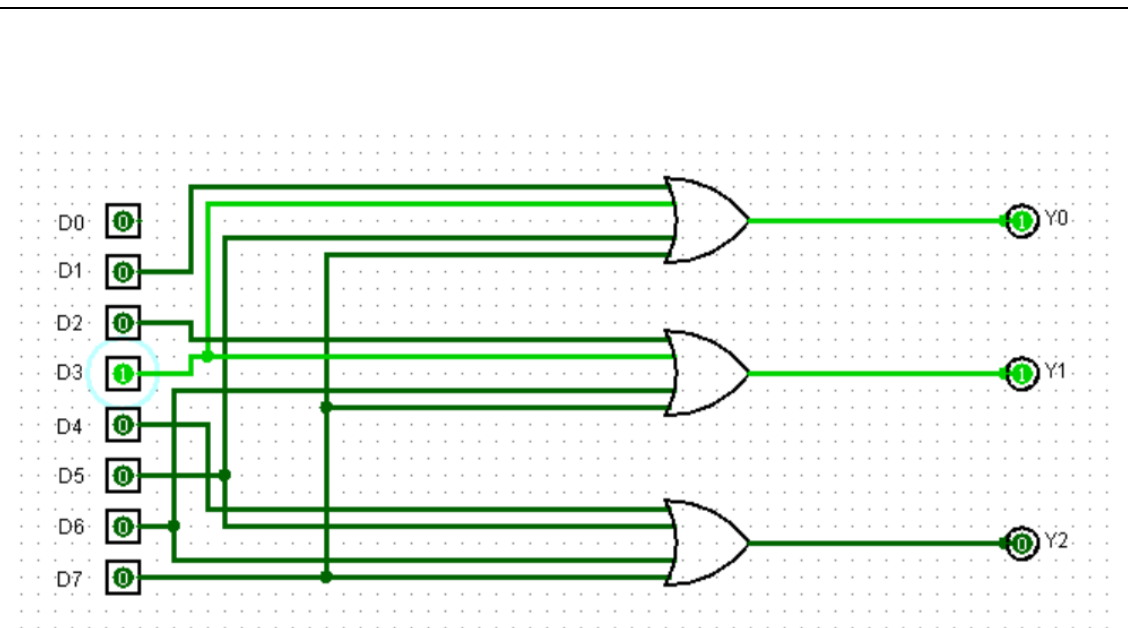


(3) 普通 8-3 选择器实验：

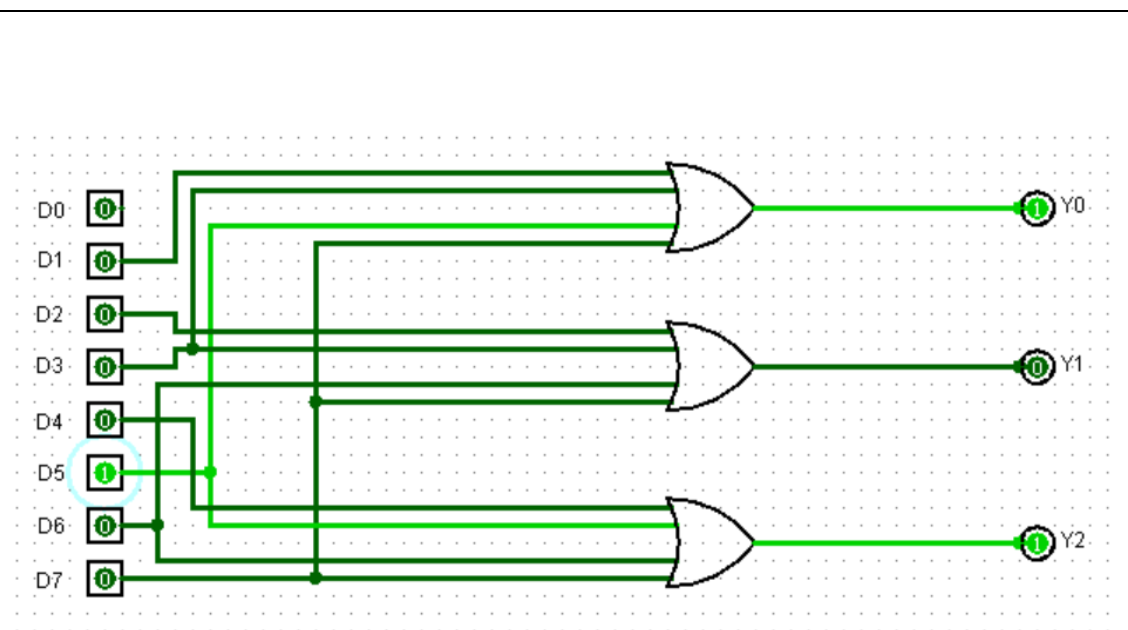
a) logisim 逻辑验证图



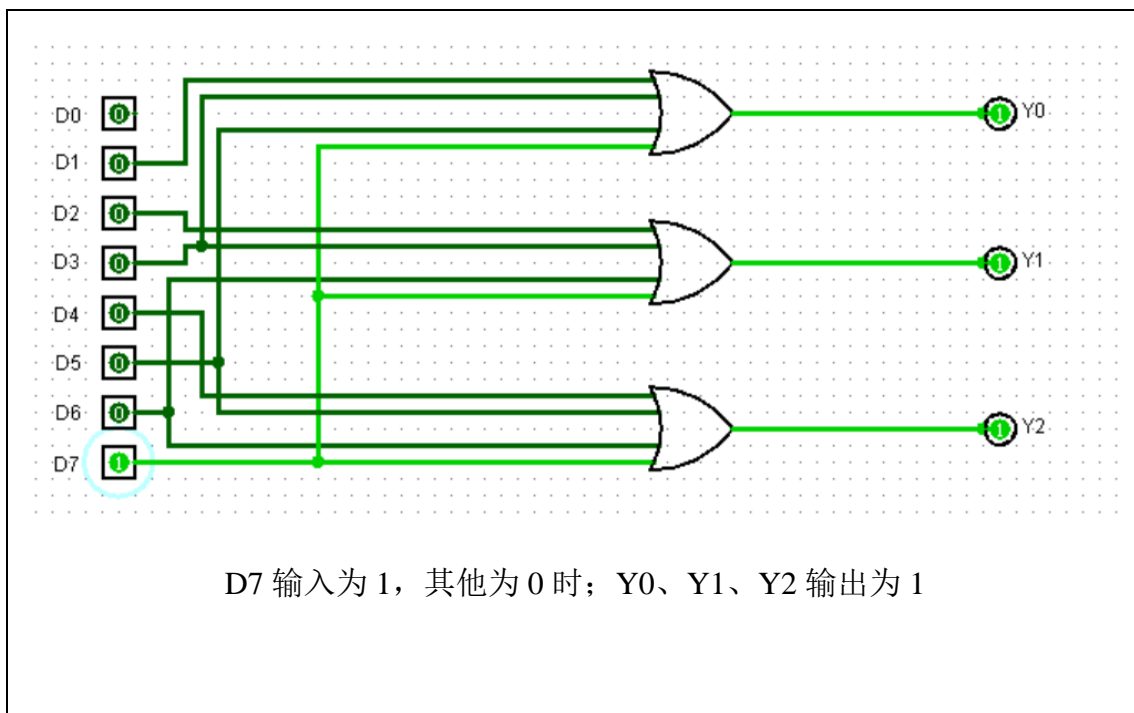
D1 输入为 1，其他为 0 时；Y1、Y2 输出为 0，Y0 为 0



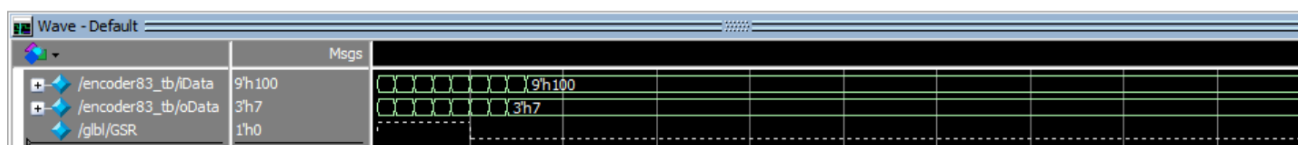
D3 输入为 1，其他为 0 时；Y0、Y1 输出为 1，Y2 为 0



D5 输入为 1，其他为 0 时；Y1、Y2 输出为 1，Y0 为 0



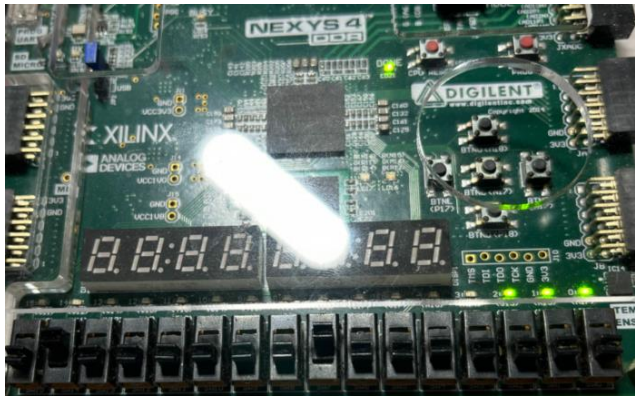


b) modelsim 波形仿真图



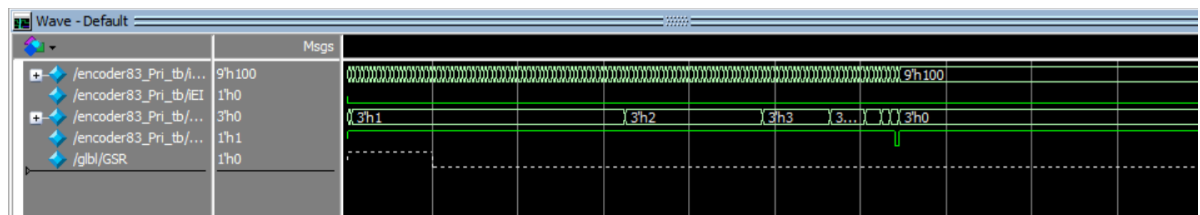
c) 下板实验结果图



<p>R15 打开，其他开关关闭</p> <p>H17、K15 亮起</p>	
<p>U18 打开，其他开关关闭</p> <p>K15、J13 亮起</p>	
<p>R13 打开，其他开关关闭</p> <p>H17、K15、J13 亮起</p>	

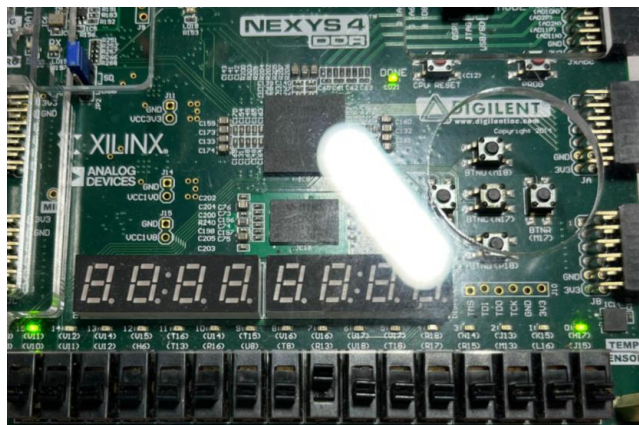
(4) 具有优先级的 8-3 选择器实验：

a) modelsim 波形仿真图

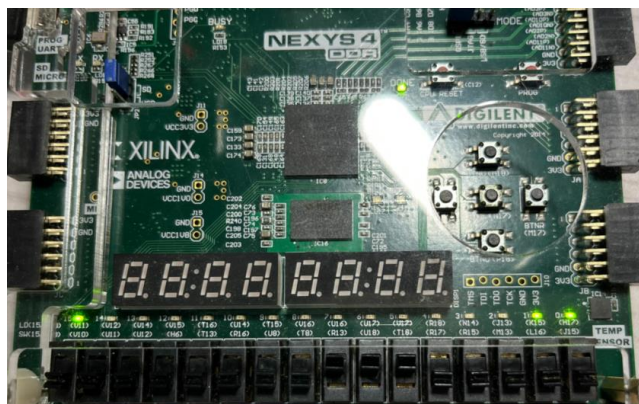


b) 下板实验结果图

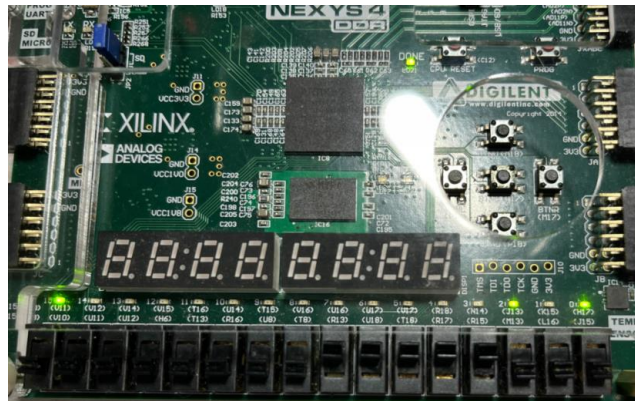
V10 关闭,
R13 打开, 其余开关关闭;
J15 亮起, V11 亮起



V10 关闭,
R13、V18、T18 打开;
K15、H17 亮起, V11 亮起



V10 关闭,
R13、V18、T18、R17、R15
打开;
J13、H17 亮起, V11 亮起



V10 关闭,
R13、V18、T18、R17、R15、
M13、L16 打开;
J13、K15、H17 亮起, V11
亮起

