

同济大学计算机系

数字逻辑课程实验报告



学 号 2252941

姓 名 杨瑞灵

专 业 计算机科学与技术

授课老师 张冬冬

一、实验内容

(1)6.3_13-8 译码

使用 Verilog 语言实现 3-8 译码器并用 logicsim 验证逻辑，modelsim 波形仿真，以及下板验证。功能：输入二进制代码，输出一组高低电平信号其包含 3 个编码输入，8 个输出和两个使能 输入端，在使用时使能端必须满足 $G1=1,G2=0$

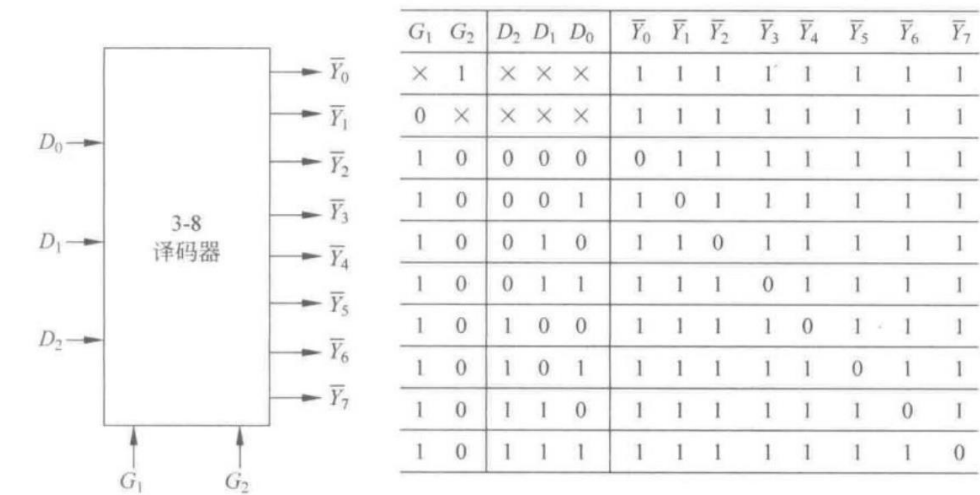


图 6.11 3-8 译码器及其真值表

(2)6.3_2 七段数码管译码驱动器

图 6.3.2 为所要建模的七段数码管译码驱动原理图，它由译码驱动器和荧光数码管组成。荧光数 码管是分段式半导体显示器件，7 个发光二极管组成 7 个发光段，发光二极管可以将电能转换成光能， 从而发出清晰悦目的光线。本实验采用的是共阳极电路，故译码器的输出 a~g 分别加到 7 个阴极上。只 有在阴极上呈低电平的二极管导通发光，显示 0~9 中相应的十进制数字。表 6.3.1 所示为七段数码管 译码驱动器逻辑功能真值表，四个输入和七个输出以及对应显示的字符。

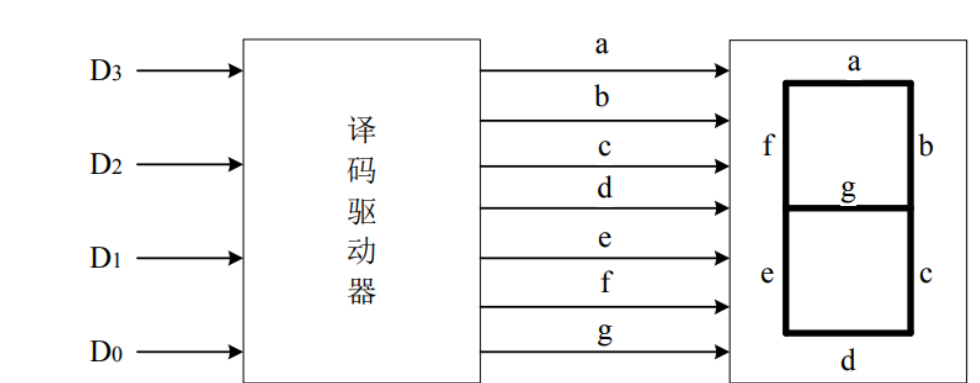


图 6.3.2 七段数码管译码驱动原理图

(3)6.3_3 普通 8-3 编码器

用来完成编码工作的电路称为编码器。它可以实现对一组输入信号的二进制编码。图 6.3.3 为所 要建模的普通 8-3 编码器及其真值表。它有 8 个输入以及 3 个输出，真值表中每行 只有一个输入电平 有效（高电平为 1，低电平为 0）。

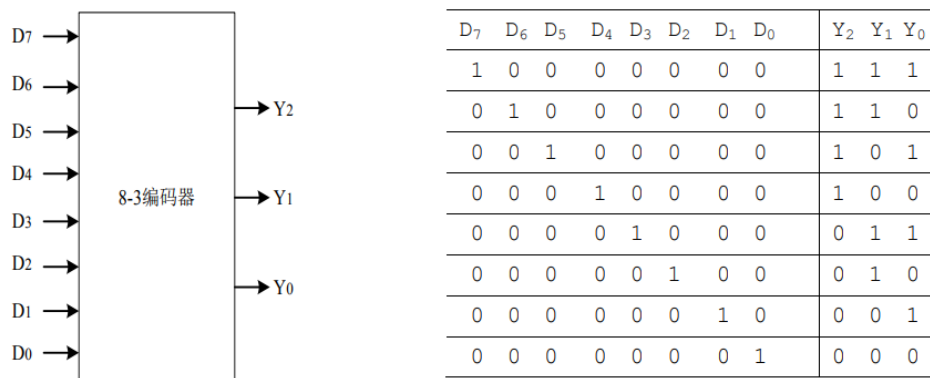
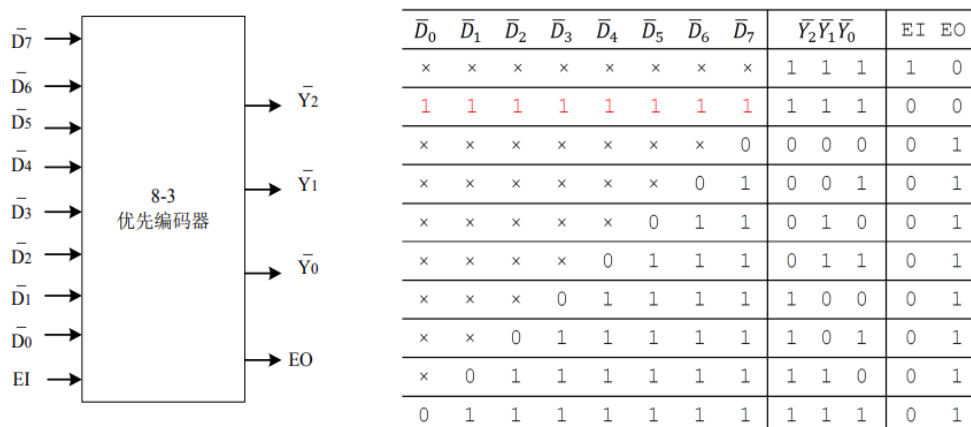


图 6.3.3 普通 8-3 编码器及其真值表

(4)6.3_4 具有优先级的 8-3 编码器

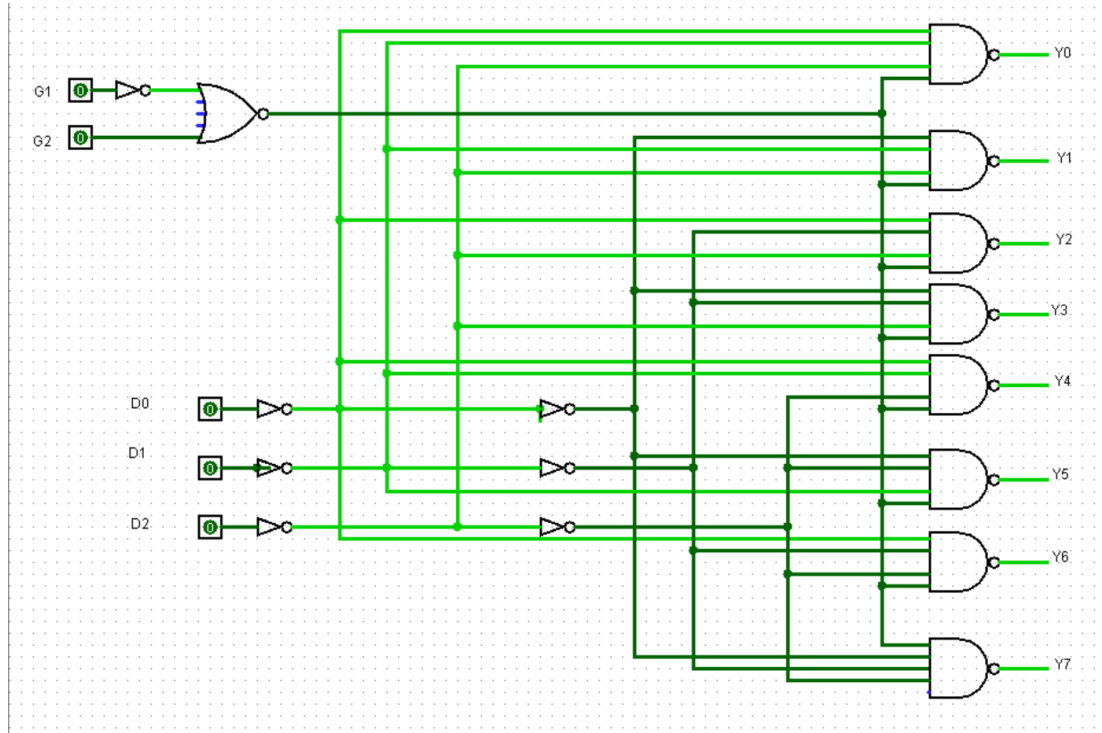
普通编码器对输入线是有限制的，即在任意一时刻所有输入线中只允许一个输入线信号有效，否则 编码器将发生混乱。为解决这一问题可以采用具有优先级的编码器。图 6.3.4 为所要建模的具有优先 级的 8-3 编码器及其真值表，它有八个输入端、三个输出端、一个选通输入端 EI 以及一个扩展输出端 EO。从真值表可以看出，输入输出的有效信号是低电平，在输入中，角标越大，优先级越高。 图 6.3.4 具有优先级的 8-3 编码器及其真值表



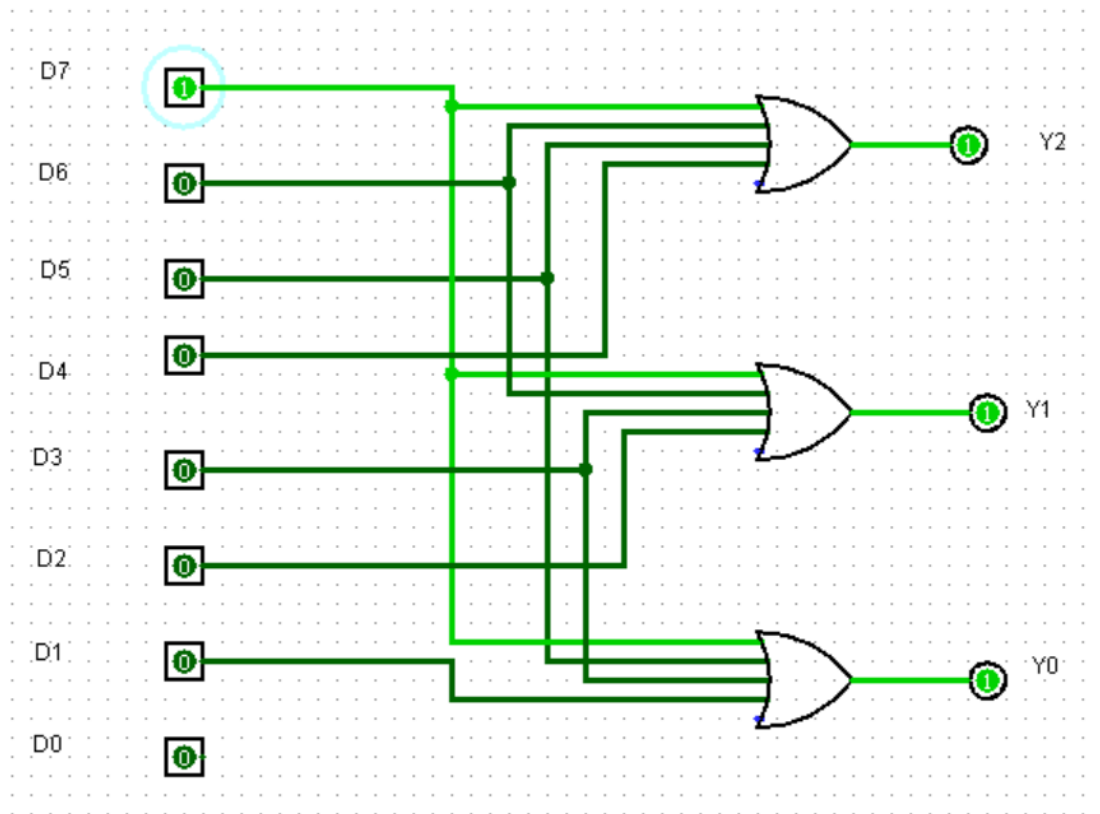
二、硬件逻辑图

(实验步骤中要求用 logisim 画图的实验, 在该部分给出 logisim 原理图, 否则该部分在实验报告中不用写)

(1)6.3_1



(2)6.3_3



三、模块建模

(1) 6.3_1

```

module decoder(
    input  [2:0] iData,
    input  [1:0] iEna,//iEna[1]:G1      iEna[0]:G2
    output [7:0] oData
);
    assign oData[0]=~iEna[1] | iEna[0] | iData[2] | iData[1] | iData[0];
    assign oData[1]=~iEna[1] | iEna[0] | iData[2] | iData[1] | ~iData[0];
    assign oData[2]=~iEna[1] | iEna[0] | iData[2] | ~iData[1] | iData[0];
    assign oData[3]=~iEna[1] | iEna[0] | iData[2] | ~iData[1] | ~iData[0];
    assign oData[4]=~iEna[1] | iEna[0] | ~iData[2] | iData[1] | iData[0];
    assign oData[5]=~iEna[1] | iEna[0] | ~iData[2] | iData[1] | ~iData[0];
    assign oData[6]=~iEna[1] | iEna[0] | ~iData[2] | ~iData[1] | iData[0];
    assign oData[7]=~iEna[1] | iEna[0] | ~iData[2] | ~iData[1] | ~iData[0];
endmodule

```

(2) 6.2_2

```

module display7(
    input [3:0] iData,

```

```

output [6:0] oData
);
reg [6:0] oData_t;
always @(*)
begin
case(iData)
4'b0000: oData_t=7'b1000000;
4'b0001: oData_t=7'b1111001;
4'b0010: oData_t=7'b0100100;
4'b0011: oData_t=7'b0110000;
4'b0100: oData_t=7'b0011001;
4'b0101: oData_t=7'b0010010;
4'b0110: oData_t=7'b0000010;
4'b0111: oData_t=7'b1111000;
4'b1000: oData_t=7'b0000000;
4'b1001: oData_t=7'b0010000;
default;;
endcase
end
assign oData = oData_t;
endmodule

```

(3)6.3_3

```

module encoder83(
input [7:0] iData,
output reg [2:0] oData
);
always @(*)
begin
if (iData[7]) oData = 3'b111;
else if (iData[6]) oData = 3'b110;
else if (iData[5]) oData = 3'b101;
else if (iData[4]) oData = 3'b100;
else if (iData[3]) oData = 3'b011;
else if (iData[2]) oData = 3'b010;
else if (iData[1]) oData = 3'b001;
else if (iData[0]) oData = 3'b000;
end
endmodule

```

(4)6.3_4

```

module encoder83_Pri(
input [7:0] iData,
input iEI,

```

```

output reg [2:0] oData,
output reg oEO
);

always @(*)
begin

    if (iEI)
        begin
            oEO = 1;
            oData = 3'b111;
        end
    else if (iData == 8'b11111111) oEO = 0;
    else
        begin
            oEO = 1;
            if (!iData[7]) oData = 3'b000;
            else if (!iData[6]) oData = 3'b001;
            else if (!iData[5]) oData = 3'b010;
            else if (!iData[4]) oData = 3'b011;
            else if (!iData[3]) oData = 3'b100;
            else if (!iData[2]) oData = 3'b101;
            else if (!iData[1]) oData = 3'b110;
            else if (!iData[0]) oData = 3'b111;
        end
    end
end
endmodule

```

四、测试模块建模

（要求列写各建模模块的 test bench 模块代码）

(1)6.3_1

```

module decoder_tb();
reg [2:0]iData;
reg [1:0] iEna;
wire [7:0]oData;
decoder de(iData,iEna,oData);
initial
begin
iEna<=2'b1;
#50 iEna<=2'b0x;
#50 iEna<=2'b10;
iData<=3'b000;

```

```

#50 iData<=3'b001;
#50 iData<=3'b010;
#50 iData<=3'b011;
#50 iData<=3'b100;
#50 iData<=3'b101;
#50 iData<=3'b110;
#50 iData<=3'b111;
end
endmodule

```

(2)6.3_2

```

module display7_tb();
reg [3:0] iData;
wire [6:0] oData;
display7 dp(.iData(iData),.oData(oData));
initial
begin
iData<='d0;
#50 iData<='d1;
#50 iData<='d2;
#50 iData<='d3;
#50 iData<='d4;
#50 iData<='d5;
#50 iData<='d6;
#50 iData<='d7;
#50 iData<='d8;
#50 iData<='d9;
end

endmodule

```

(3)6.3_3

```

module encoder83_tb();
reg [7:0] iData;
wire [2:0] oData;
encoder83 ec(.iData(iData),.oData(oData));
initial
begin
iData = 8'b10000000;
#50 iData = 8'b01000000;
#50 iData = 8'b00100000;
#50 iData = 8'b00010000;
#50 iData = 8'b00001000;

```



```
#50 iData = 8'b000000100;
#50 iData = 8'b000000010;
#50 iData = 8'b000000001;
end
```

```
endmodule
```

(4)6.3_4

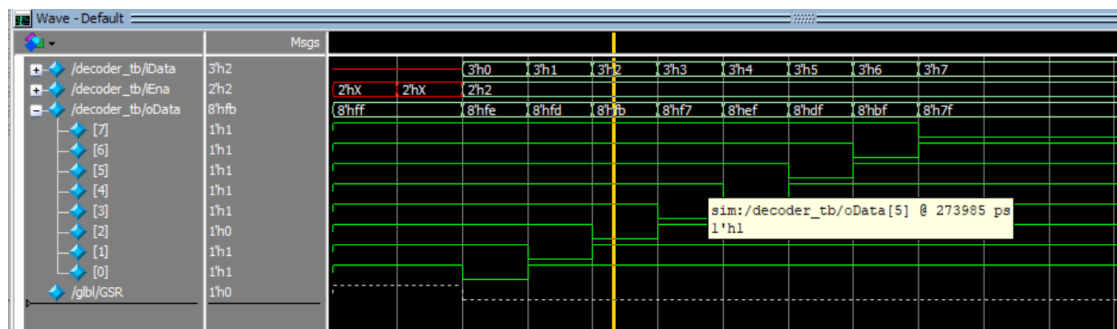
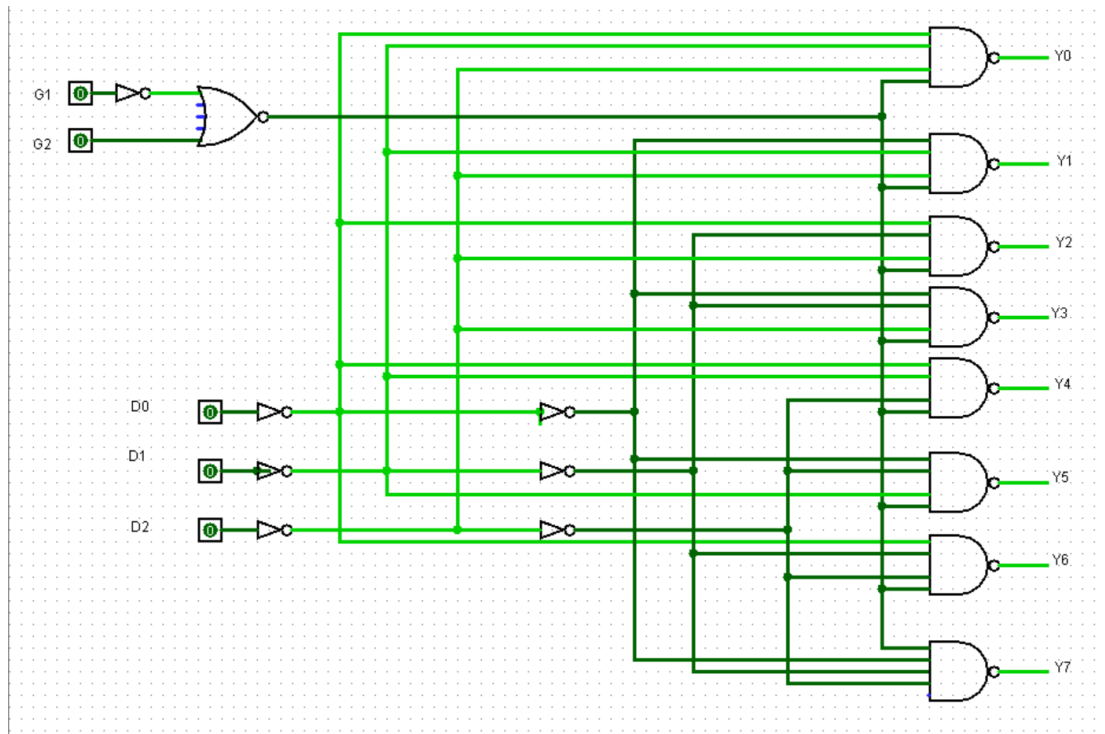
```
module encoder83_pri_tb();
reg [7:0] iData;
reg iEI;
wire [2:0] oData;
wire oEO;
encoder83_Pri ecp(.iData(iData), .iEI(iEI), .oData(oData), .oEO(oEO));
initial
begin
iEI <= 1;
#50 iEI<=0;
    iData<=8'b11111111;
#50 iData<=8'b0xxxxxxx;
#50 iData<=8'b10xxxxxx;
#50 iData<=8'b110xxxxx;
#50 iData<=8'b1110xxxx;
#50 iData<=8'b11110xxx;
#50 iData<=8'b111110xx;
#50 iData<=8'b1111110x;
#50 iData<=8'b11111110;
end
endmodule
```

五、实验结果

(1)6.3_1

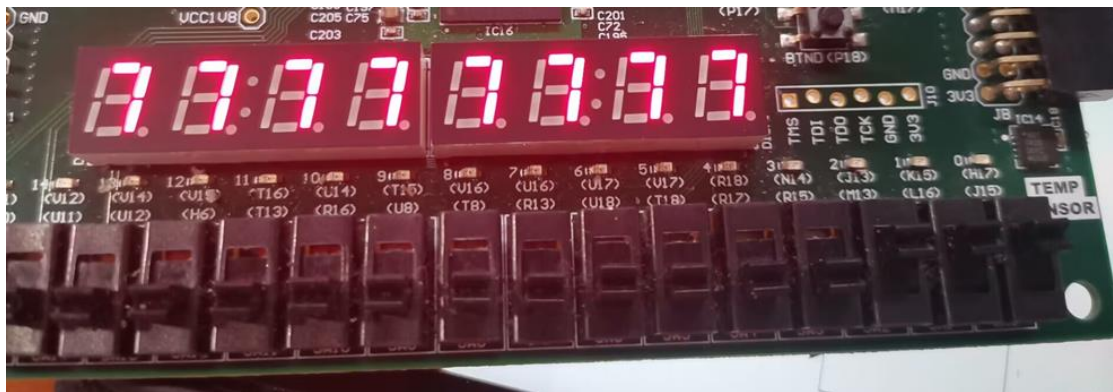
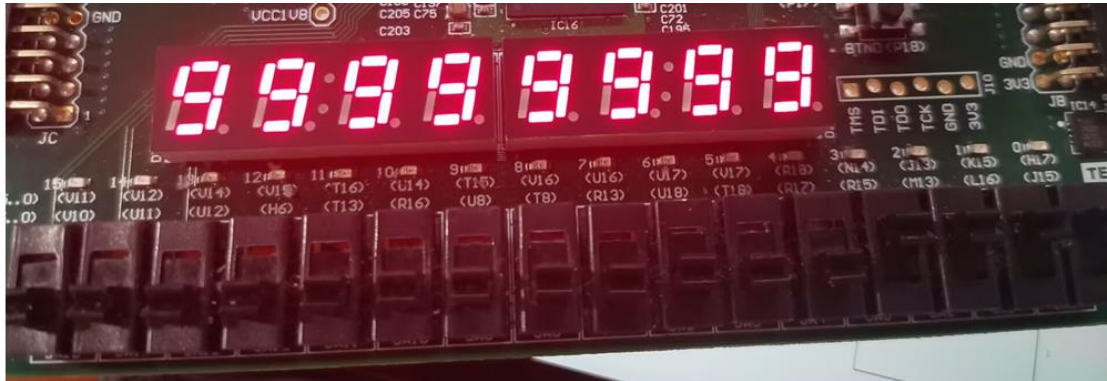
- XDC 文件配置：

变量	iData[0]~[2]	iEna[0]~[1]	oData[0]~[7]
N4 板上的管脚	SW0~2 (J15、L16、M13)	SW14~15 (V10、U11)	LD0~7 (H17、K15、J13、N14、 R18、V17、U17、U16)

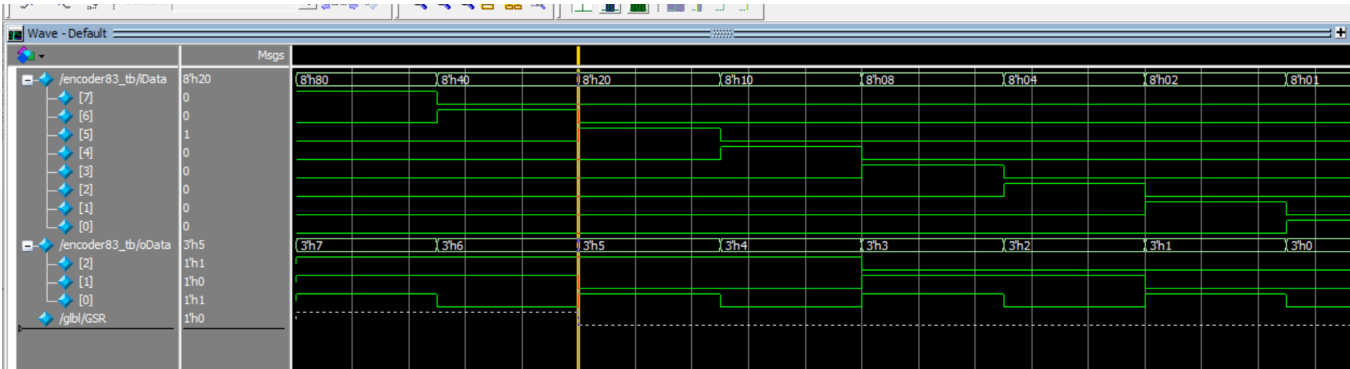
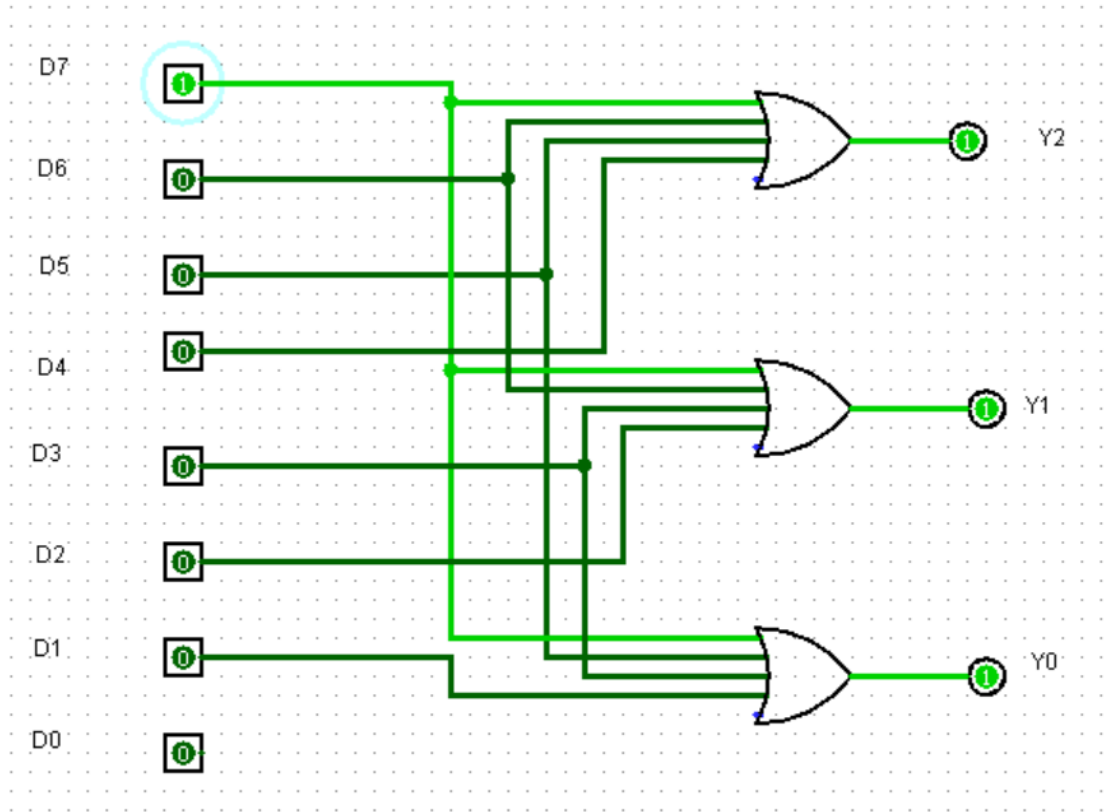


● XDC 文件配置:

变量	iData[0]~[3]	oData[0]~[6]
N4 板上的管脚	SW0~3 (J15、L16、M13、R15)	CA (T10)、CB (R10)、CC (K16)、CD (K13)、CE (P15)、CF (T11)、CG (L18)

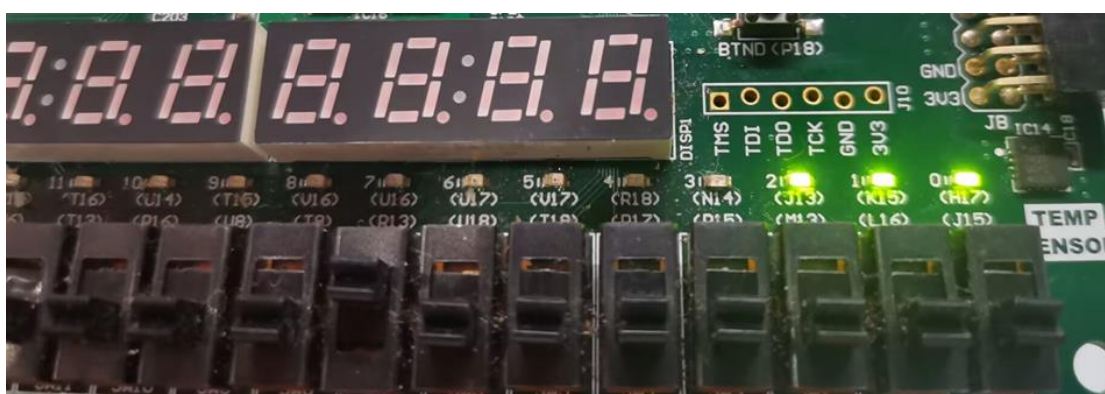
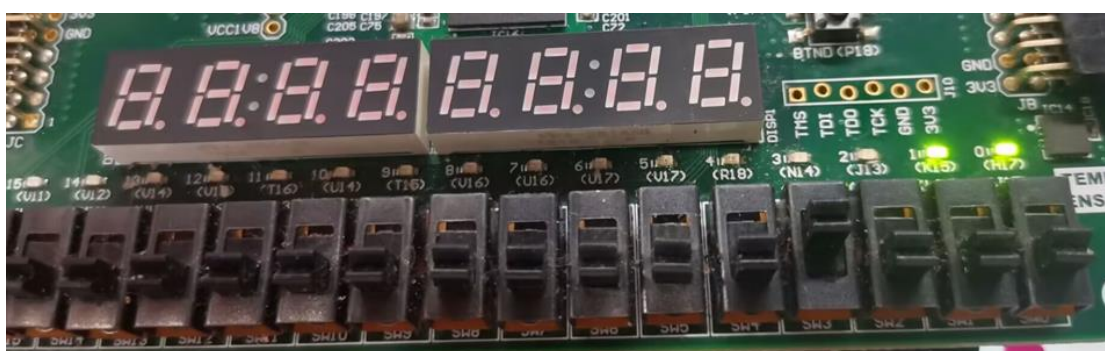
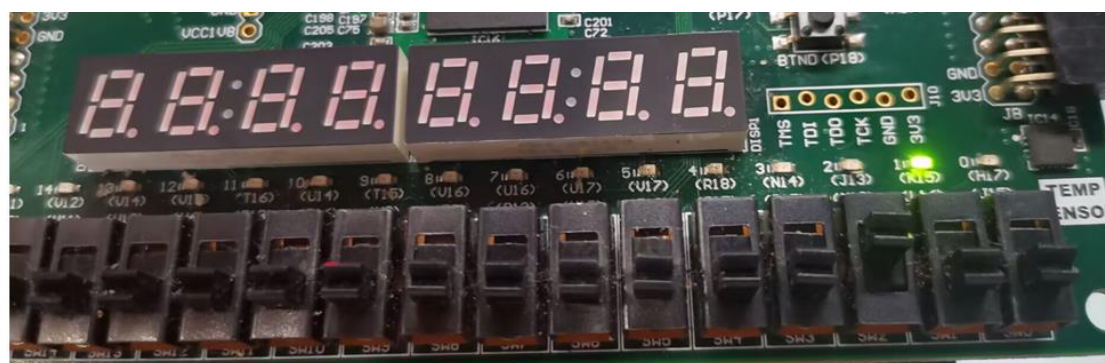


(3)6.3_3



● XDC 文件配置:

变量	iData[0]~[7]	oData[0]~[2]
N4 板上的管脚	SW0~7 (J15、L16、M13、R15、R17、T18、U18、R13)	LD0~2 (H17、K15、J13)



(4)6.3_4



● XDC 文件配置:

变量	iData[0]~[7]	oData[0]~[2]	iEI	oEO
N4 板上的 管脚	SW0~7 (J15、L16、M13、R15、 R17、T18、U18、R13)	LD0~2 (H17、K15、J13)	SW15 (V10)	LD15 (V11)

