

电 工 电 子 实 验 报 告

课程名称： 电工电子实验（二）

实验名称： Verilog HDL组合逻辑电路设计

&Verilog HDL时序逻辑电路设计

学 院： 通信与信息工程学院

班 级： B210111

学 号： B21011125

姓 名： 徐秋旸

指导教师： 林宏

学 期： 2022-2023 学年第 二 学期

电工电子实验教学中心

**Verilog HDL组合逻辑电路设计**

**一、实验目的**

1.使用ISE软件完成组合逻辑设计的输入并仿真。

2.掌握Testbench中组合逻辑测试文件的写法。

3.下载并测试实现的逻辑功能。

**二、主要仪器设备及软件**

硬件：DGDZ-5型实验箱，可编程器件XC3S50ANTQG144

软件：ISE Design Suite 14.7

**三、实验原理**

Verilog HDL是用来设计数字和计算机系统的新技术，在业界广泛使用。通过使用集成开发环境，设计人员可以在常见的Windows或其他图形化系统中进行设计、仿真、验证。

由于C语言在Verilog HDL设计之初已经在许多领域得到广泛应用，因此Verilog HDL的设计初衷是成为一种基本语法与C语言相近的硬件描述语言。但是，Verilog HDL作为一种与普通计算机编程语言不同的硬件描述语言，还具有一些独特的语言要素，如向量形式的线网和寄存器、过程中的非阻塞赋值等。总体而言，具备C语言基础的设计人员能够快速掌握Verilog HDL。

随着大规模可编程逻辑器件和电子设计自动化平台在数字设计系统中的应用，以及硬件描述语言的出现，设计者可以使用硬件描述语言描述自己的设计，借助电子设计自动化平台进行综合、优化、布局布线以及可编程逻辑器件的适配和下载。

基于Verilog HDL的数字电路基本描述方法包括门级结构描述、数据流描述和行为描述，具体而言就是根据电路功能抽象出端口，根据功能抽象采用相应方法进行描述，实现功能电路的建模。

**四、实验内容和实验结果**

1.用Verilog HDL设计一个8选1数据选择器，完成设计模块、Testbench，并保存仿真波形。要求Testbench能够覆盖所有的输入组合、下载到FPGA、完成硬件调测并实现电路功能。

①8选1数据选择器功能表如下：

表1 8选1数据选择器功能表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 使能输入 | 输入 | | | 输出 |
|  |  |  |  | *Y* |
| 1 | Φ | Φ | Φ | 0 |
| 0 | 0 | 0 | 0 |  |
| 0 | 0 | 0 | 1 |  |
| 0 | 0 | 1 | 0 |  |
| 0 | 0 | 1 | 1 |  |
| 0 | 1 | 0 | 0 |  |
| 0 | 1 | 0 | 1 |  |
| 0 | 1 | 1 | 0 |  |
| 0 | 1 | 1 | 1 |  |

②设计代码如下：

module sel(EN, sel, D, Y);

input EN;

input [2:0] sel;

input [7:0] D;

output Y;

reg Y;

always @ (EN or sel or D)

if(!EN)

case(sel)

3'b000:Y=D[0];

3'b001:Y=D[1];

3'b010:Y=D[2];

3'b011:Y=D[3];

3'b100:Y=D[4];

3'b101:Y=D[5];

3'b110:Y=D[6];

3'b111:Y=D[7];

endcase

else Y=0;

endmodule

③激励代码如下：

module tb\_sel;

reg EN;

reg [2:0] sel;

reg [7:0] D;

wire Y;

sel uut (

.EN(EN),

.sel(sel),

.D(D),

.Y(Y)

);

initial begin

EN=1;

sel=3'b000;

D=8'b10101100;

fork

repeat(12.5) #80 EN=~EN;

repeat(100) #10 sel=sel+1;

join

end

endmodule

④行为仿真图如下：

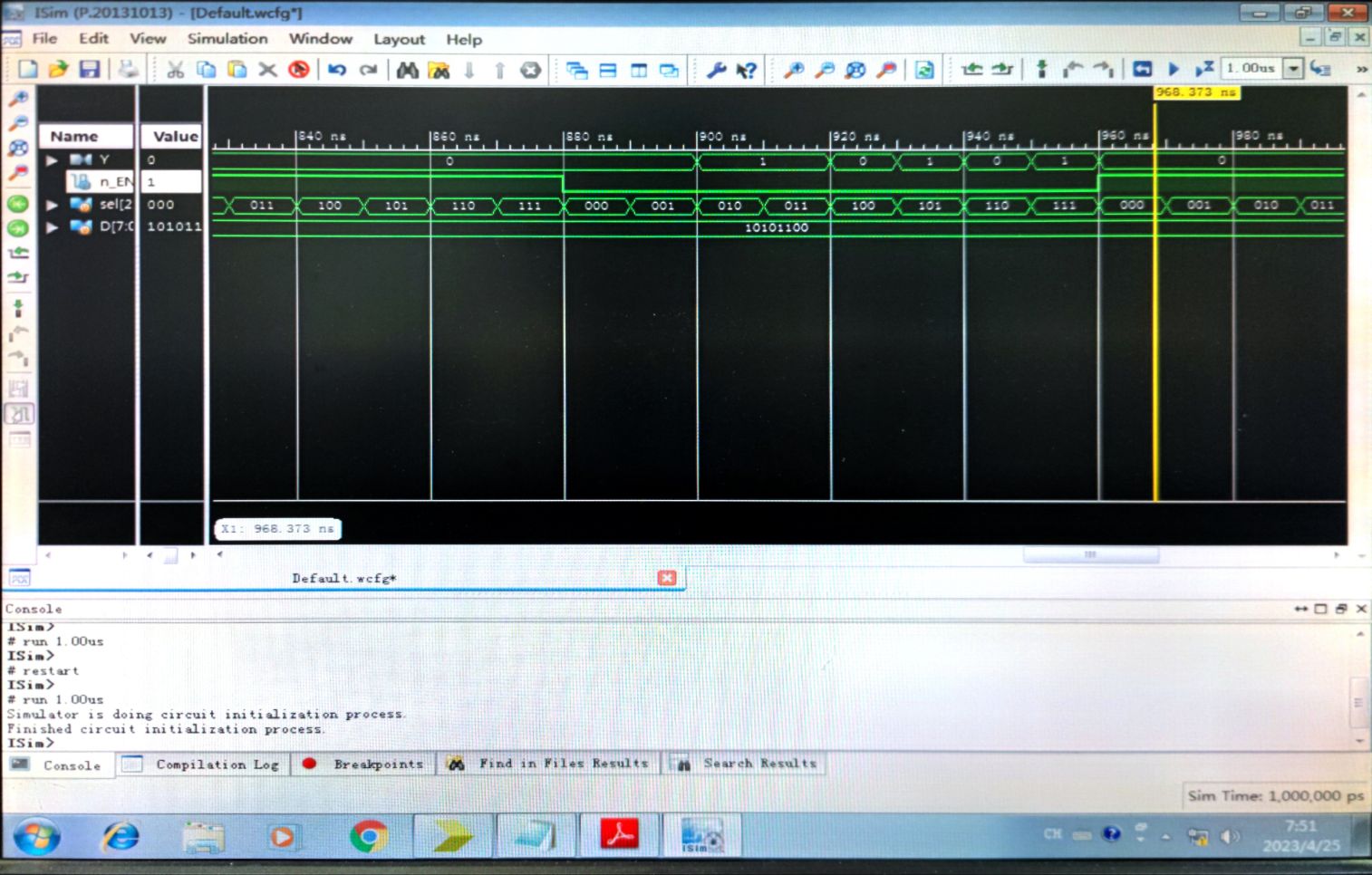


图4 8选1数据选择器行为仿真图

由上图可以看出，当使能端为高电平时，输出恒为低电平；当使能端为低电平时，输出依次为预置好的D0~D7端。设计代码满足功能表，能够完成8选1数据选择器的功能；激励代码能够测试所有真值表中的状态。

⑤下载、硬件测试

通过静态测试的方法，按照真值表逐行进行测试，测试结果满足功能表，设计能够完成8选1数据选择器的功能。

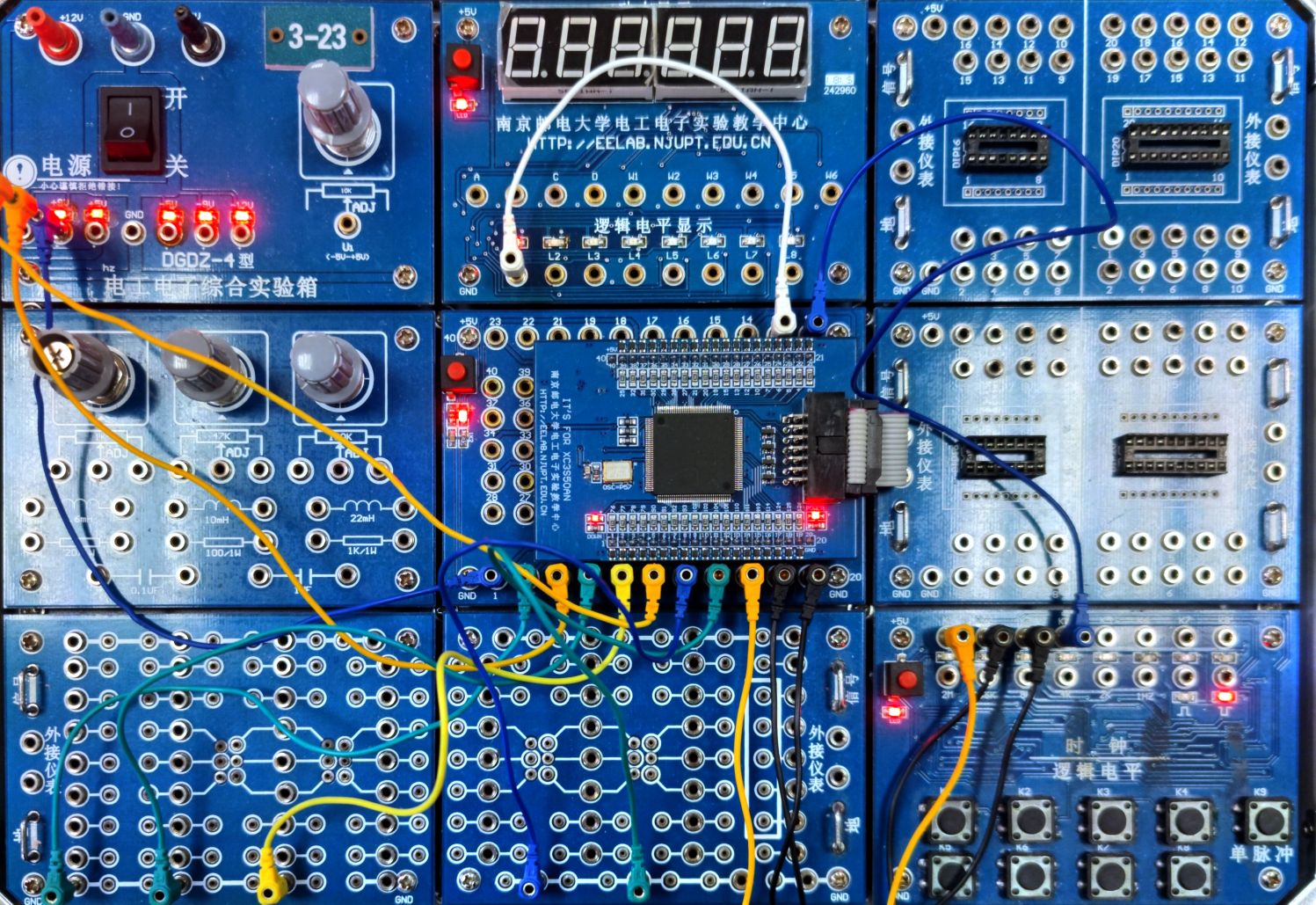


图5 8选1数据选择器烧录测试

2.用Verilog HDL设计一个3-8线译码器，完成设计模块、Testbench，并保存仿真波形。要求Testbench能够覆盖所有的输入组合、下载到FPGA、完成硬件调测并实现电路功能。

①3-8线译码器功能表如下：

表2 典型3-8线译码器功能表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 输入 | | | | | | 输出 | | | | | | | |
| G1 | G2A | G2B | C | B | A | Y7 | Y6 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |
| 0 | Φ | Φ | Φ | Φ | Φ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Φ | 1 | Φ | Φ | Φ | Φ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Φ | Φ | 1 | Φ | Φ | Φ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

②设计代码如下：

module decoder(en, in, out);

input [2:0] in;

input en;

output [7:0] out;

always @ (en or in)

if(en)

case(in)

3'b000:out=8'b11111110;

3'b001:out=8'b11111101;

3'b010:out=8'b11111011;

3'b011:out=8'b11110111;

3'b100:out=8'b11101111;

3'b101:out=8'b11011111;

3'b110:out=8'b10111111;

3'b111:out=8'b01111111;

default:out=8'b11111111;

endcase

else out=8'b11111111;

endmodule

③激励代码如下：

module tb\_decoder;

reg en;

reg [2:0] in;

wire [8:1] out;

decoder uut (

.en(en),

.in(in),

.out(out)

);

initial begin

en=3'b000;

in=3'b000;

fork

repeat(12.5) #80 en=en+1;

repeat(100) #10 in=in+1;

join

end

endmodule

④行为仿真图如下：

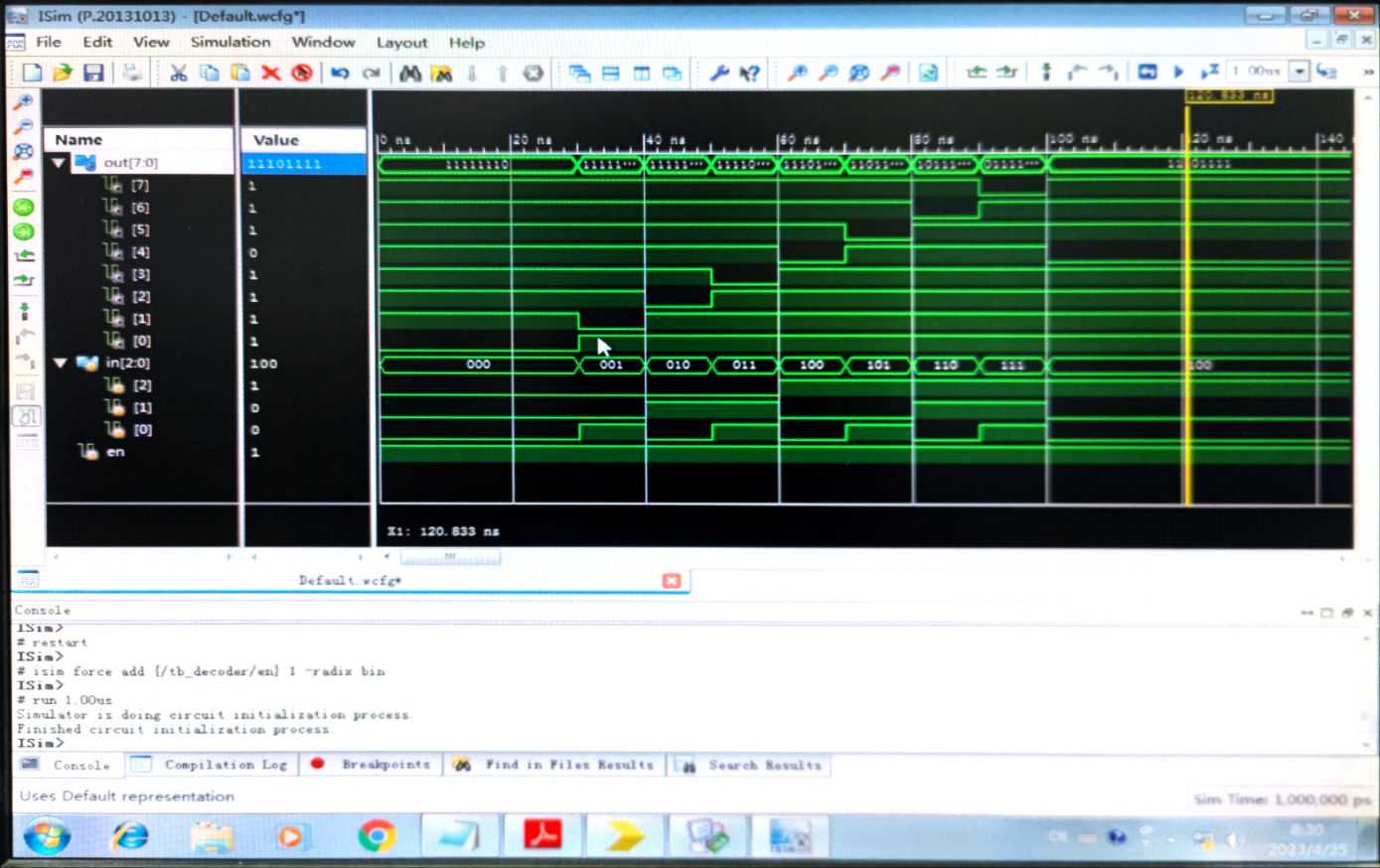


图6 3-8线译码器行为仿真图

由上图可以看出，当使能端无效时，输出恒为高电平；当使能端有效时，Y0~Y7依次输出低电平。设计代码满足功能表，能够完成3-8线译码器的功能；激励代码能够测试所有真值表中的状态。

⑤下载、硬件测试

通过静态测试的方法，按照真值表逐行进行测试，测试结果满足功能表，设计能够完成3-8线译码器的功能。

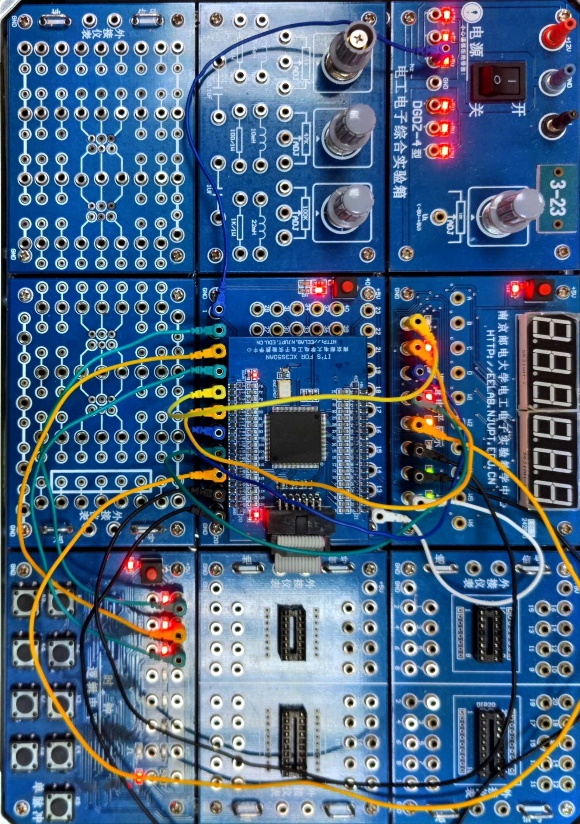


图7 3-8线译码器烧录测试

**六、结果分析**

所设计的三段代码以及三段激励代码能够完成所需的8选1数据选择器、3-8线译码器的功能及行为仿真，下载测试后结果也满足真值表。

**七、实验小结**

1.实现同一元件的代码可以有很多种写法，但所实现的功能应该相同。

2.Verilog HDL语言具有其自己的语法规则，在编写时要遵守语法规则，并进行测试修改，直到没有错误为止。

3.为了测试到功能表的所有状态，在编写激励文件时要注意各个信号之间周期的关系。