

# 数字电路实验4

---

徐亦昶

PB20000156

2021.11.18

## 实验题目

Verilog 硬件描述语言

## 实验目的

掌握 Verilog HDL 常用语法

能够熟练阅读并理解 Verilog 代码

能够设计较复杂的数字功能电路

能够将 Verilog 代码与实际硬件相对应

## 实验环境

vlab.ustc.edu.cn

verilog.ustc.edu.cn

## 实验练习

### 题目1

把代码改成下面这样：

```
`timescale 1ns / 1ps
module test(
  input a,
  output reg b); //b设为reg
always@(*) //if放入always语句
begin
  if(a) b <= 1'b0; //always中使用<=赋值
  else b <= 1'b1;
end
endmodule
```

### 题目2

```
module test(
  input[4:0] a,
  output reg [4:0]b);
always@(*)
```

```
b<=a;  
endmodule
```

### 题目3

```
c=8'b0011_0000  
d=8'b1111_0011  
e=8'b1100_0011  
f=8'b1100_1100  
g=8'b0011_0000  
h=8'b0000_0110  
i=1'b0  
j=8'b1111_0000  
k=8'b0100_0011
```

### 题目4

将代码更改成下面的样子：

```
module sub_test(  
    input a,b,  
    output c); //不需要设为reg型  
assign c=(a<b)?a:b;  
endmodule  
module test(  
    input a,b,c,  
    output o);  
wire temp; //使用wire  
sub_test(.a(a),.b(b),.c(temp)); //接口调用方式不能混用  
sum_test(tmp,c,o);  
endmodule
```

### 题目5

```
module sub_test(  
    input a,b,  
    output o); //将右括号放到正确的位置  
assign o=a+b;  
endmodule  
module test(  
    input a,b.  
    output reg c); //在always中赋值，需要设为reg类型  
always@(*)  
begin  
    sub_test sub_test(a,b,c);  
end  
endmodule
```

## 总结与思考

- 1、通过本次实验，学会了Verilog的基本语法，同时可以阅读他人的Verilog代码，并进行静态查错。
- 2、难易程度和任务量适中。
- 3、建议开发verilog在线编译平台，以方便地测试Verilog代码的正确性。