

中国科学技术大学计算机学院  
《数字电路实验》报告



实验题目: verilog 硬件描述语言  
学生姓名: 牛庆源  
学生学号: PB21111733  
完成日期: 2022. 11. 3

## 【实验题目】

阅读一些 verilog 代码并改错或者填写空白

## 【实验目的】

1. 了解 verilog 的基本语法，关键字，以及模块的调用之类的。

## 【实验环境】

Vscode

## 【实验练习】

### 题目一

```
module test(  
    input a,  
    output reg b);  
always @ (*)  
begin  
    if(a) b = 1'b0; // 组合逻辑赋值用等号  
    else b = 1'b1;  
end  
endmodule  
  
// if else 语句用在 always 语句的过程语句部分，不能在模块内部独立出现  
// always 中赋值的变量用 reg 类型声明
```

### 题目二

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

### 题目三

```
module test(  
    input a,  
    output reg b);  
always @ (*)  
begin  
    b = a;  
end  
endmodule
```

```

    input [7:0] a,b,
    output [7:0] c,d,e,f,g,h,i,j,k);

    assign c = a & b;
    assign d = a | b;
    assign e = a ^ b;
    assign f = ~a;
    assign g = {a[3:0], b[3:0]};
    assign h = a >> 3;
    assign i = &b;
    assign j = (a > b)? a : b;
    assign k = a - b;
endmodule
// 在输入 a = 8'b0011_0011,
// 且输入 b = 8'b1111_0000 时
// 输出为 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'b0000_0000
// j = 8'b1111_0000
// k = 8'b0100_0011

```

## 题目四

```

module small_test(
input a, b,
output c);
assign c = (a < b)? a : b;
endmodule

module test(
input a, b, c,
output o);
wire temp;
small_test small_test1(.a(a),.b(b),.c(temp));
small_test small_test2(.a(temp),.b(c),.c(o));
endmodule
// 模块实现的是比较两个一位二进制数的大小并输出的功能，模块名为 sub_test 显然不合，改为 small_test
// small_test 两次调用应该赋予不同的模块名
// 模块调用时候通过名称关联和通过位置关联不能混用

```

```
// assign 连续赋值语句中的变量用 wire 声明
// 模块中 c 是 wire 型变量，第一次使用模块时 temp 为 reg 型，所以应该改为 wire 型
```

## 题目五

```
module add_test(
input a,b,
output o);
assign o = a + b;
endmodule

module test(
input a,b,
output c);
add_test add_test(a,b,c);
endmodule
// 首先模块是实现两个 1 位二进制数的加法，但模块名字为 sub_test 显然不和，改为
add_test
// 模块调用时候若选择通过位置关联则需要知道模块声明有什么变量
// 这里只有 input a,b，没有 output，需要把 output o 放在模块的括号内声明
// always 中调用的模块内部使用 assign 时，output 为 wire 则 always 报错，为 reg
则 assign 报错，删除 always 直接例化即可。
```

### 【总结与思考】

1. 了解了 assign, always 等 verilog 关键字；debug 能力得到提高；对 wire 和 reg 类型更加敏感；熟悉了基本的模块调用方法。