

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： 谭骏飞

学生学号： PB20061276

完成时间： 2021.11.4

## 实验题目

### Verilog 硬件描述语言

#### 实验练习

题目 1. 阅读以下 **Verilog** 代码，找出其语法错误，并进行修改

```
module test(  
    input a,  
    output b);  
    if(a) b = 1'b0;  
    else b = 1'b1;  
endmodule
```

语法错误：涉及到有关条件判断的语句，应当将其用**always@\***语句括起来，从而**b**也应当定义为**reg**类型。正确代码如下。

```
module test(  
    input a,  
    output reg b  
);  
    always@(*)begin  
        if(a) b=1'b0;  
        else b=1'b1;  
    end  
endmodule //实验4-1
```

## 题目 2. 阅读以下 **Verilog** 代码，将空白部分补充完整

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

分析：由于是在定义模块，因此最后一行必定是"**endmodule**"; 在**always**语句中，将**a**通过阻塞式赋值的方式赋值给**b**，因此**b**的位宽应与**a**相同，也为**4**位，且由于是在**always**语句中赋值，**b**应当定义为**reg**类型，故前一行应为"**output reg [4:0] b**"。填完后代码如下。

```
module test(  
    input [4:0] a,  
    output reg [4:0] b  
);  
    always @(*)  
        b=a;  
endmodule //实验4-2
```

## 题目 3. 阅读以下 **Verilog** 代码，写出当 **a = 8'b0011\_0011**, **b = 8'b1111\_0000** 时各输出信号的值。

```
module test(  
    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
```

相关分析与输出结果如下。

```
a = 8'b0011_0011 //51(d)
b = 8'b1111_0000 //240(d)
c = 8'b0011_0000 //a与b
d = 8'b1111_0011 //a或b
e = 8'b1100_0011 //a异或b
f = 8'b1100_1100 // 按位求反
g = 8'b0011_0000 //g把a的后三位和b的后三位拼接起来
h = 8'b0000_0110 //a右移三位
i = 0 //把b的每一位都与起来
j = 8'b1111_0000 //取出a b中较大的一个
k = 51 - 240 = -189(d)
```

题目 4. 阅读以下 **Verilog** 代码，找出代码中的语法错误，并修改。

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

module test(
    input a,b,c,
    output o);
    reg temp;
    sub_test(.a(a),.b(b),temp);
    sub_test(temp,c,.c(o));
endmodule
```

语法错误：**sub\_test**中的**c**和**test**中的**temp**应当为**wire**类型；**test**模块中两个**sub——test**实例都没有实例化的名字；第二个**sub\_test**实例的端口信号在关联时不能位置关联和名称关联两种方式混用。修改后的代码如下。

```
module sub_test(
    input a,b,
    output c
);
```

```

    assign c = (a<b)?a:b;
endmodule //实验4-4

module test(
    input  a,b,c,
    output o
);
wire temp;
sub_test ins_name1(
    .a (a),
    .b (b),
    .c (temp)
);
sub_test ins_name2(//两个实例的名字不能一样
    .a(temp),
    .b(c),
    .c(o)
);

```

题目 5. 阅读以下 **Verilog** 代码，找出其中的语法错误，说明错误原因,并进行修改。

```

module sub_test(
input a,b);
output o;
    assign o = a + b;
endmodule
module test(
input a,b,
output c);
    always@(*)
    begin
        sub_test sub_test(a,b,c);
    end
endmodule

```

语法错误：**sub\_test**模块中端口**o**的定义应当放在模块名**sub\_test**后紧跟的小括号内；**a+b**的值有可能是两位，因此**o**应当定义为两位的**wire**值以便存储；**test**模块中实例化的名字不能和模块名一样；不能在**always**语句中进行实例化。修改后的代码如下。

```
module sub_test(  
    input a,b,  
    output [1:0]o  
);  
    assign o = a+b;  
endmodule //实验4-5  
  
module test(  
    input a,b,  
    output [1:0]c  
);  
    sub_test sub_test1(a,b,c);  
endmodule //实验4-5
```

## 总结与思考

本次实验让我进一步了解和熟悉了**verilog**语言中的模块及其实例化、端口定义与其在实例化时的关联、条件判断语句、**always**语句、各种运算符的功能等知识点，为以后利用**vivado**进行仿真打下基础，也利于我进一步理解数字电路的相关知识。