**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：\_\_\_\_\_\_\_\_徐奥\_\_\_\_\_\_\_\_

学生学号：\_\_\_\_\_PB20061343\_\_\_\_\_

完成日期：\_\_2021年11月4日\_\_

计算机实验教学中心制

2020年09月

【实验题目】

Verilog硬件描述语言

【实验目的】

1. 掌握 Verilog HDL 常用语法
2. 能够熟练阅读并理解 Verilog 代码
3. 能够设计较复杂的数字功能电路
4. 能够将 Verilog 代码与实际硬件相对应

【实验环境】

vlab.ustc.edu.cn

【实验过程】

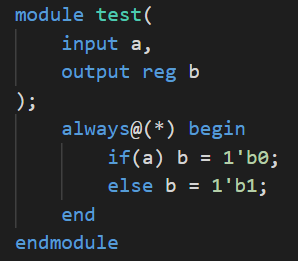
1. Verilog关键字。包括：module/endmodule、input、output、wire、reg、 assign、always、initial、begin/end、posedge、negedge、if、else、 case、endcase等。
2. Verilog代码基本结构。每个模块都是由关键字 module 开头，由 endmodule 结束。每个模块都应该有一个唯一的模块名，模块名不能使用Verilog 语法的关键字。模块名后面的括号内是对输入输出信号的定义。模块主体部分只能出现四类语句：内部信号定义、模块实例化、assign 语句、always 语句，每类语句的数量不受限制。
3. Verilog 数据及类型。4中基本值，常量。
4. Verilog 操作符：算术运算符、关系运算符、逻辑操作符、归约操作符、条件操作符、移位操作符、拼接操作符
5. Verilog 表达式
6. 模块调用

【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

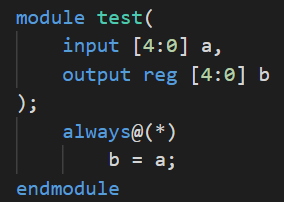
语法错误包括：

1. if、else 用于实现带有优先级的条件分支， 一般出现在 always 语句的过程语句部分，而不能直接在模块内部单独出现。所以应将if、else语句放在always中。
2. 在always语句中的赋值操作对象，应设置为reg型，而非默认的wire型。

修改为：

**图1**

**题目2.** 阅读以下 Verilog 代码，将空白部分补充完整



**图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

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 = 8'b0000\_0000

j = 8'b1111\_0000

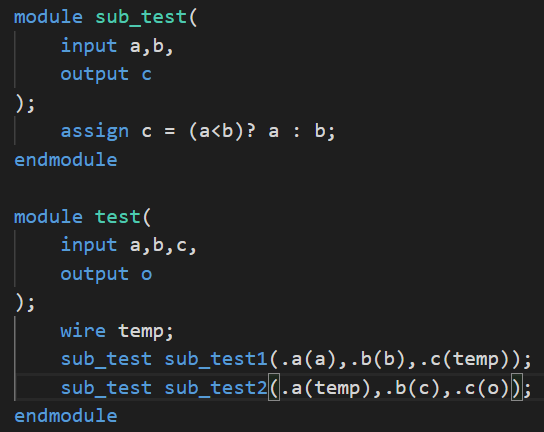
k = 8'b0100\_0011

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

错误：

1. assign赋值是对线网型赋值，不能对reg类型赋值
2. 端口信号可以通过位置或名称进行关联，但两种关联方式不能混用。

修改为：



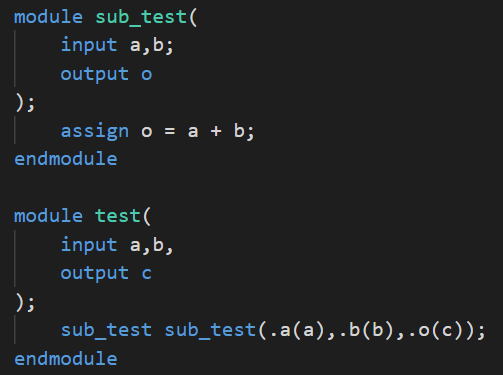
**图3**

**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。

错误：

1. 模块实例化不可以在always里，Verilog是硬件设计语言，模块实例化在往电路板上安装芯片（module）。如果模块实例化放在 always 里面了，就相当于“有条件地安装一个芯片”，这显然不是一个正常的硬件逻辑。
2. 模块名后面的括号内应该包含输入输出信号的定义，不应该把output放在括号外。

修改为：



**图4**

【总结与思考】

1. 本次实验系统学习了verilog语言，对于它的使用有了更深的了解，为之后的实验打下基础。
2. 在复习过程中，对于实验文档的细节知识需要认真学习，其中就包括：模块实例化：实例化的输出端只能接 wire 类型信号
3. 本次实验内容量适中，难度适中，但对于细节知识的考查十分细致。