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：王章瀚

学生学号：PB18111697

完成日期：2019/10/25

计算机实验教学中心制

2019 年 09 月

1 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

2 实验环境

PC 一台
Windows 或 Linux 操作系统
Java 运行环境（jre）
Logisim 仿真工具
vlab.ustc.edu.cn (jre 和 Logisim 工具都可在此网站获取)

3 实验过程

3.1 Verilog 关键字

初学者常用的关键字有，module/endmodule、input、output、wire、reg、assign、always、initial、begin/end、posedge、negedge、if、else、case、endcase。

其中，本人较为不熟悉的是 case/endcase 语句，故总结如下：具有相同优先级的多路条件分支，两个关键字必须成对出现。一般出现在 always 的过程语句部分，而不能在模块内部单独出现。例如：

```
1   module test(
2       input wire a,b,clk,
3       output reg o);
4
5   always@( posedge clk )
6   case({a,b})
7       2'b00: o <= 1'b0;2'b01: o <= 1'b0;
8       2'b10: o <= 1'b0;
9       2'b11: o <= 1'b1;
10  endcase
11  endmodule
12
```

3.2 Verilog 基本结构

Verilog 基本结构可以总结为如下：

```
module 模块名（
    输入端口定义 //输入端口只能是 wire 类型
    输出端口定义 //输入信号可根据需要定义成 wire 或 reg 类型
    ）；
    内部线信号定义 //内部信号可根据需要定义成 wire 或 reg 类型
    模块实例化 //实例化的输出端只能接 wire 类型信号
    assign 连续赋值语句
    always 过程语句
endmodule

```

3.3 Verilog 数据及类型

Verilog 有四种基本的值：0, 1, x, z。

Verilog 有三种常量：整数，实数，字符串。其中整数例如 6'b101100。可选的有二进制 (b/B)、八进制 (o/O)、十进制 (d/D) 和十六进制 (h/H) Verilog 有两种常用数据类型：wire（线网类型）和 reg（寄存器类型）关于两种数据类型的使用只需要遵循以下规则即可：凡是通过 assign 语句赋值的信号（一定是组合逻辑赋值信号），都应定义成 wire 类型，凡是在 always 语句中赋值的信号（可能是组合逻辑赋值信号、也可能是时序逻辑赋值信号），都应定义成 reg 类型。

3.4 Verilog 操作符

```
算数运算符： +、-、*、/、%
关系运算符：>、<、>=、<=、==、!=
逻辑操作符：&&、||、!、~、&、|、^ 、~^
归约操作符：&、~&、|、~|、^、~^
条件操作符：？：//三目运算符
移位操作符：<<、>>
拼接操作符：{}

```

3.5 Verilog 表达式

表达式可以是以下类型：常数、参数、线网、寄存器、位选择、部分选择、存储器单元、函数调用。

表达式都有一个值，因此可以将其赋给 wire 或 reg 类型的信号，也可以用在逻辑判断语句（如 if、case）中。如 assign a = 表达式 1、always@（*）if(表达式)... else ...。

3.6 模块调用

模块实例语句的形式为：模块名 实例化名（端口关联）。

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。

```
add add_inst1(a,b,s,carry1);//通过位置关联
add add_inst2(.a(s),.b(cin),.sum(sum),.cout(cout));//通过名称关联

```

3.7 代码实例

```
8bit 位宽 4 选 1 选择器，纯组合逻辑
module mux_4to1( //8bit 位宽的 4 选 1 选择器
    input [7:0] a,b,c,d,
    input [1:0] sel,
    output reg [7:0] o); //always 语句内赋值的信号都应定义成 reg 类型

always@(*) //always 语句内实现组合逻辑
begin
    case(sel)
        2'b00: o = a; //组合逻辑使用“=”进行赋值
        2'b01: o = b;
        2'b10: o = c;
        2'b11: o = d;
        default: o = 8'h0;//用 case 语句实现组合逻辑时一定要有 default
    endcase
end
endmodule
================================
1~10 循环计数的计数器
module cnt_1to10(
    input clk,rst_n,
    output reg [3:0] cnt);
```

```
23
24      always@(posedge clk or negedge rst_n)
25      //时序控制条件为时钟上升沿和复位的下降沿
26      begin
27          if(!rst_n) //复位信号优先级最高，应是第一个判断的条件
28              cnt <= 4'h1;
29          else if(cnt>=10)
30              cnt <= 4'h1;
31          else
32              cnt <= cnt + 4'h1;
33      end
34      endmodule
35
```

4 实验练习

4.1 题目 1.

修改结果如下：

```
1       module test(
2           input a,
3           output reg b); //因为b需要在always内赋值，须为reg类型
4
5       // if-else语句不能直接在模块内使用，
6       // 应在always语句的过程语句部分内使用
7       always @(*)
8       begin
9           if(a) b = 1'b0;
10          else b = 1'b1;
11      end
12      endmodule
13
```

4.2 题目 2.

```
1       module test(
2           input [4:0] a,
3           output reg[4:0] b);
```

```
4
5       always@(*)
6           b = a;
7       endmodule
8
```

4.3 题目 3.

列表如下：

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

4.4 题目 4.

```
1       module sub_test(
2           input a,b,output wire c); //将reg修改为wire，因为通过assign赋值
3
4       assign c = (a<b)? a : b;
5       endmodule
6
7       module test(
8           input a,b,c,
9           output o);
10
11      reg temp;
12      sub_test(a, b, temp); //两种传参方法不能同时使用
13      sub_test(temp, c, o); //两种传参方法不能同时使用
14      endmodule
15
```

4.5 题目 5.

```
1   module sub_test(
2       input a,b;
3       output o); //output的声明应该也要在括号内
4
5   assign o = a + b;
6   endmodule
7
8
9   module test(
10      input a,b,
11      output c); //在always内赋值，应是reg类型
12
13  //always里不应该使用模块的例化
14  sub_test sub_test(a, b, c);
15  endmodule
16
```

5 总结与思考

5.1 本次实验的收获

在本次实验中，我系统了解了 Verilog 这个语言，也发现了一些以往的误区，收获颇丰。

5.2 评价本次实验的难易程度

本次实验内容难度适中。

5.3 评价本次实验的任务量

本次实验任务量合理。

5.4 为本次实验提供改进建议

暂无建议。