

数电实验4

【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

【实验步骤】

- Verilog关键字
- Verilog代码基本结构
- Verilog数据及类型
- Verilog操作符
- Verilog表达式
- 模块调用

【实验题目】

T1

```
1  module tedt(  
2      input a,  
3      output reg b  
4  );  
5      always@(*)  
6          begin  
7              if(a) b=1'b0;  
8              else b=1'b1;  
9          end  
10 endmodule
```

T2

```
1 module test(  
2     input [4:0] a,  
3     output reg[4:0] b  
4 );  
5     always@(*)  
6         b=a;  
7 endmodule
```

T3

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 = b = 8'b1111_0000

k = 8'b0100_0011

T4

```
1 module sub_test(  
2     input a,b,  
3     output c  
4 );  
5     assign c=(a>b)?b:a; //reg类型不能再次使用assign赋值  
6 endmodule
```

```
1 module test(  
2     input a,b,c,  
3     output o);  
4     wire temp; //c是wire型, temp也应该是wire型  
5     //模块调用要有实例化名, 且实例化位置关联与名字关联应统一  
6     sub_test sub1(.a(a), .b(b), .c(temp));  
7     sub_test sub2(temp,c,o);  
8 endmodule
```

T5

```
1 module sub_test(  
2     input a,b,  
3     output reg o  
4 ); //sub_test右边的括号位置应该位于output o后  
5     always@(*) //always语句没办法直接模块调用  
6         o=a+b;  
7 endmodule  
8  
9 module test(  
10    input a,b,  
11    output c  
12 );  
13    sub_test sub_test1(a,b,c); //实例化名字不要和模块名相同
```

【总结与思考】

收获：更加深入了解使用了Verilog

难易程度：本次实验较简单

任务量：比较少