



# 本科实验报告

课程名称: 计算机组成原理实验

课程编号: 08060038

学生姓名: 邝庆璇

学号: 2016051598

学院: 信息科学与技术学院

系: 计算机系

专业: 计算机科学与技术

指导教师: 王传胜 梁倬骞

教师单位: 暨南大学计算机系

开课时间: 2019~2020 学年度 第 1 期

暨南大学教务处

2019 年 12 月 05 日

## 数字逻辑实验 课程实验项目目录

学生姓名：邝庆璇 学号：2016051598

| 序号 | 实验项目编号     | 实验项目名称       | *实验项目<br>类型 | 成绩 | 指导教师        |
|----|------------|--------------|-------------|----|-------------|
| 1  | 0806003801 | 验证逻辑门        | 验证          |    | 王传胜 梁<br>倬骞 |
| 2  | 0806003802 | 基本逻辑门电路之间的转换 | 验证          |    | 王传胜 梁<br>倬骞 |
| 3  | 0806003803 | 超前进位全加器      | 验证          |    | 王传胜 梁<br>倬骞 |
| 4  | 0806003804 | 译码器          | 验证          |    | 王传胜 梁<br>倬骞 |
| 5  | 0806003805 | 数字比较器        | 验证          |    | 王传胜 梁<br>倬骞 |
| 6  | 0806003806 | 七段数码管        | 验证          |    | 王传胜 梁<br>倬骞 |
| 7  | 0712000807 | 计数器          | 验证          |    | 王传胜 梁<br>倬骞 |
| 8  | 0712000808 | 寄存器          | 验证          |    | 王传胜 梁<br>倬骞 |
| 9  | 0712000809 | 时序逻辑电路综合设计   | 验证          |    | 王传胜 梁<br>倬骞 |
| 12 |            |              |             |    |             |
| 13 |            |              |             |    |             |
| 14 |            |              |             |    |             |
| 15 |            |              |             |    |             |
| 16 |            |              |             |    |             |
| 17 |            |              |             |    |             |
| 18 |            |              |             |    |             |

\*实验项目类型：演示性、验证性、验证性、设计性实验。

\*此表由学生按顺序填写。

# 暨南大学本科实验报告专用纸

课程名称 数字逻辑实验 成绩评定                       
实验项目名称 Verilog 门级建模 指导教师 王传胜 梁倬骞  
实验项目编号 0806003801 实验项目类型 验证 实验地点 N126  
学生姓名 邝庆璇 学号 2016051598  
学院 信息科学技术 系 计算机科学 专业 计算机科学与技术  
实验时间 2019 年 9 月 30 日 上 午

## 一、实验目的

- 1、学会使用 Verilog HDL 进行门级建模；
- 2、初步学会怎么用 Verilog HDL 编写仿真程序；
- 3、初步学会怎么使用 Vivado 软件；
- 4、初步学会验证基本逻辑门的逻辑功能的方法；
- 5、初步学会验证怎么使用 EG0-1 实验板。

## 二、实验内容

编写 Verilog 程序，验证 3 输入的与门、与非门、或门、或非门、异或门、同或门

## 三、实验程序

```
expl.v
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/09/30 11:28:50
// Design Name:
// Module Name: exp-1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
```

```
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////
```

```
module exp1(
    input in1,
    input in2,
    input in3,
    output out1,
    output out2,
    //output out3,
    output out4,
    output out5,
    output out6,
    output out7
);
    and funcAnd(out1, in1, in2, in3);
    or funcOr(out2, in1, in2, in3);
    xor funcXor(out4, in1, in2, in3);
    nand funcNand(out5, in1, in2, in3);
    nor funcNor(out6, in1, in2, in3);
    xnor funcXnor(out7, in1, in2, in3);

endmodule
```

#### 四、仿真程序

```
sim_1.v
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/09/30 11:57:26
// Design Name:
// Module Name: sim_1
// Project Name:
// Target Devices:
```

```
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////
```

```
module sim_1(
    output out1,
    output out2,
    //output out3,
    output out4,
    output out5,
    output out6,
    output out7
);
    reg[2:0] in;
    initial
    begin
        in = 3'b00;
    end
    always #10
    begin
        in = in + 1;
    end
    exp1 exp1(in[2],in[1], in[0], out1, out2, out4, out5, out6, out7);

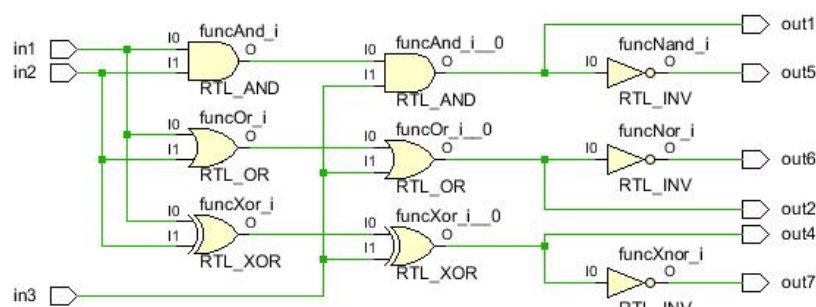
endmodule
```

## 五、仿真结果

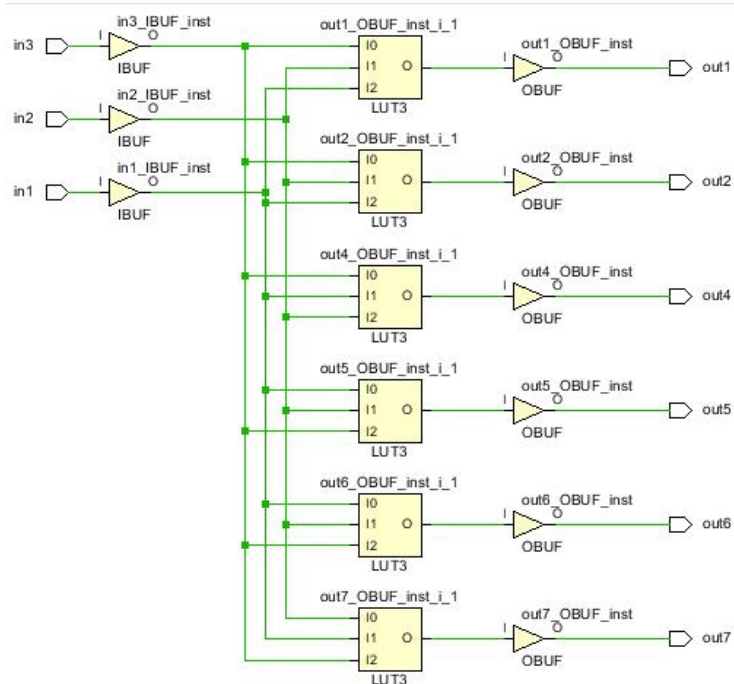


## 六、系统网表

### RTL Analysis



### Synthesis



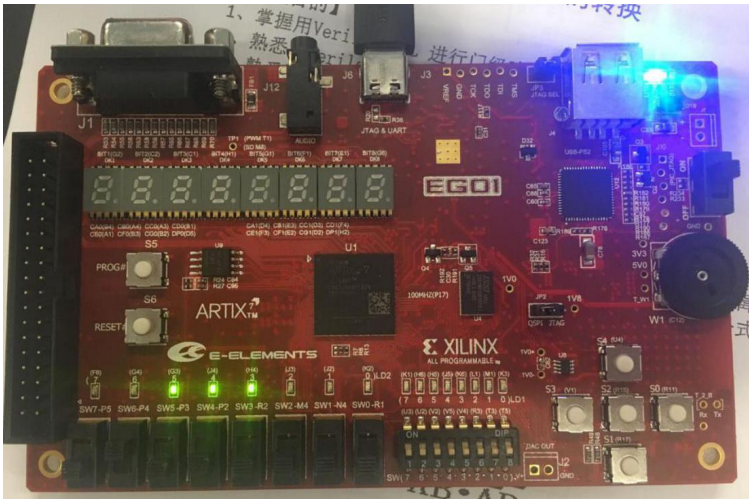
七、系统约束

| Name             | Direction | Neg Diff Pair | Package Pin | Fixed                               | Bank | I/O Std    | Vcco  |
|------------------|-----------|---------------|-------------|-------------------------------------|------|------------|-------|
| All ports (9)    |           |               |             |                                     |      |            |       |
| Scalar ports (9) |           |               |             |                                     |      |            |       |
| in1              | IN        |               | P5          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| in2              | IN        |               | P4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| in3              | IN        |               | P3          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| out1             | OUT       |               | G4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| out2             | OUT       |               | G3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| out4             | OUT       |               | J4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| out5             | OUT       |               | H4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| out6             | OUT       |               | J3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| out7             | OUT       |               | J2          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |

八、实验结果

示例：

|      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|
| 输入   | In1  |      | In2  |      | In3  |      |
| 输入的值 | 0    |      | 2    |      | 0    |      |
| 输出   | out1 | out2 | out4 | out5 | out6 | out7 |
| 逻辑运算 | And  | Or   | Xor  | Nand | Nor  | xnor |
| 输出的值 | 0    | 1    | 1    | 1    | 0    | 0    |



九、实验体会

此次试验属于初步的实践，学习了门级建模，熟悉了 vivado 软件的使用及 verilog 编程的语法，有效地为数字逻辑试验和学习作准备。感悟：刚开始使用时，使用软件进行建模的流程相对繁琐，但熟悉之后会比较顺利，毕竟初期的实验难度不大。

# 暨南大学本科实验报告专用纸

课程名称 数字逻辑实验 成绩评定             
实验项目名称 基本逻辑门电路之间的转换 指导教师 王传胜 梁倬骞  
实验项目编号 0806003802 实验项目类型 验证 实验地点 N126  
学生姓名 邝庆璇 学号 2016051598  
学院 信息科学技术 系 计算机科学 专业 计算机科学与技术  
实验时间 2019 年 10 月 14 日 上 午

## 一、实验目的

掌握用 Verilog HDL 进行门级建模；  
熟悉用 Verilog HDL 写仿真程序；  
熟悉使用 Vivado 软件；  
学会验证基本逻辑门电路之间的转换方法；  
熟悉使用 EGO-1 实验板。

## 二、实验内容

用与非门（NAND）来实现异或（XOR）运算

## 三、实验程序

```
expl.v
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/10/14 12:11:56
// Design Name:
// Module Name: exp_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
```



```
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////
```

```
module exp_1(
    input in1,
    input in2,
    output out
);

    nand (temp1, in1, in2);
    nand (temp2, in1, temp1);
    nand (temp3, in2, temp1);
    nand (out, temp2, temp3);

endmodule
```

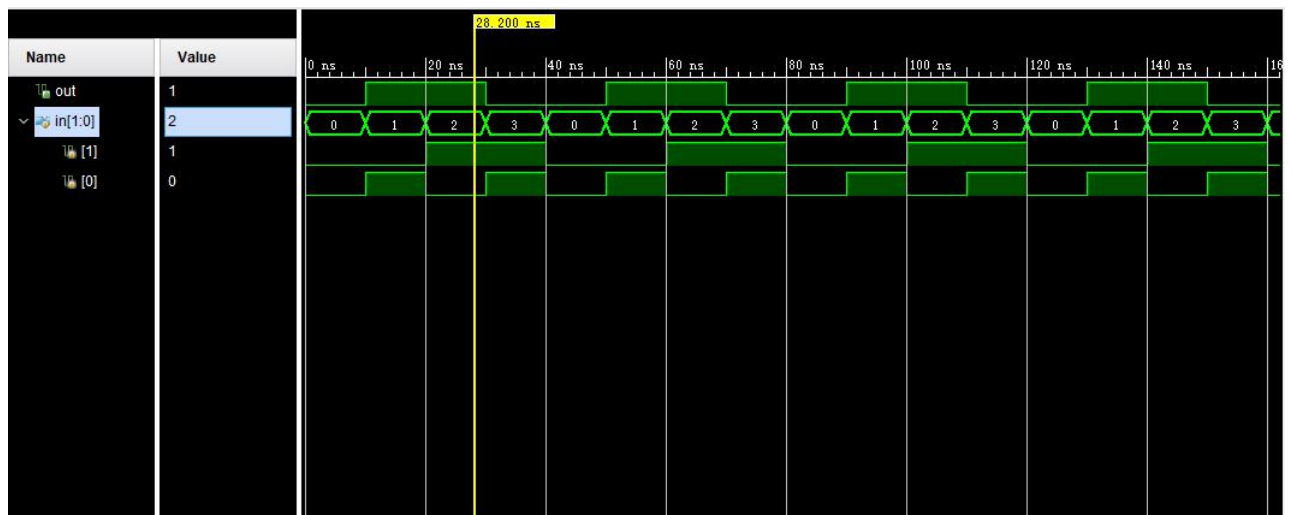
#### 四、仿真程序

```
sim_1.v
`timescale 1ns / 1ps
////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/10/14 12:22:02
// Design Name:
// Module Name: sim_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
```

```
// Additional Comments:
//
////////////////////////////////////////////////////////////////
```

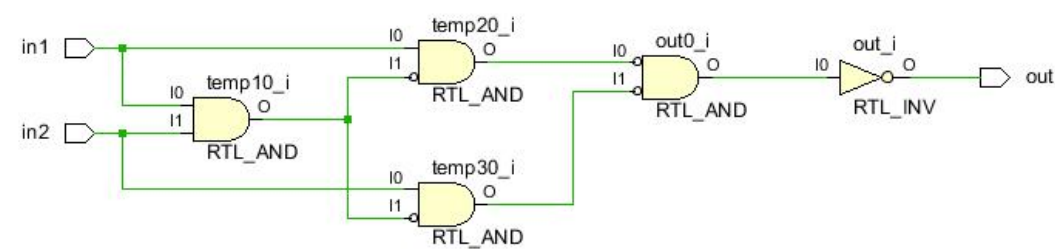
```
module sim_1(
    output out
);
    reg[1:0] in;
    initial
    begin
        in = 3'b00;
    end
    always #10
    begin
        in = in + 1;
    end
    exp_1 exp_1(in[1], in[0], out);
Endmodule
```

## 五、仿真结果

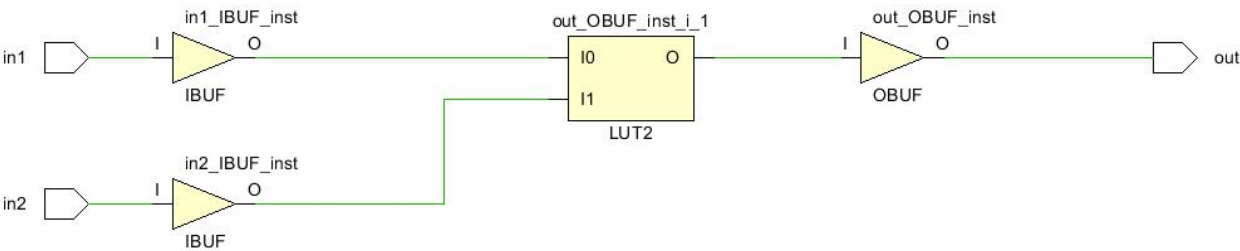


六、系统网表

RTL Analysis



Synthesis



七、系统约束

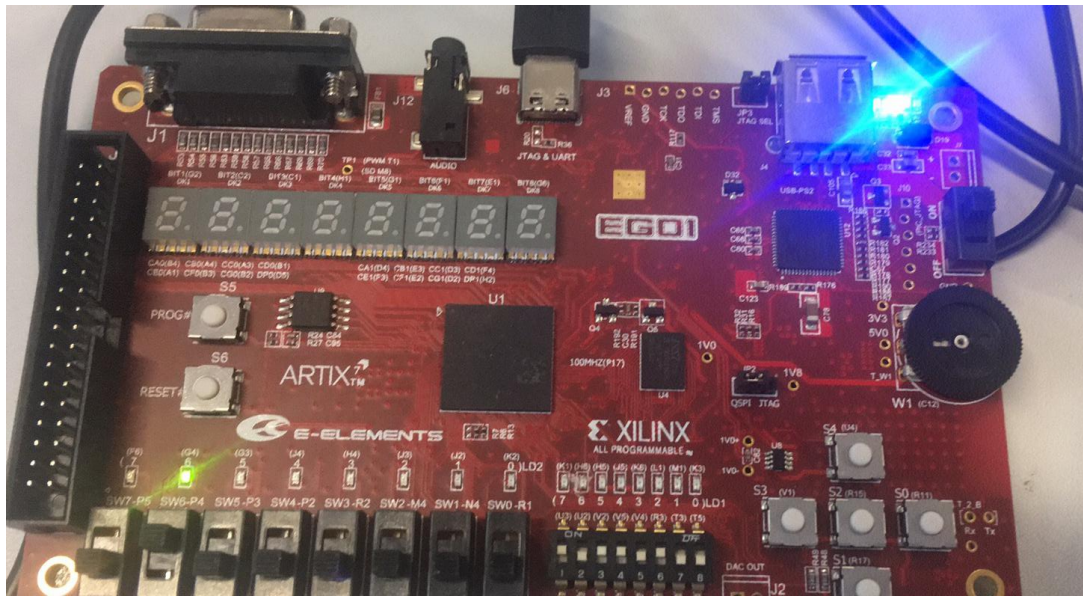
| Name             | Direction | Neg Diff Pair | Package Pin | Fixed                               | Bank | I/O Std    | Vcco  |
|------------------|-----------|---------------|-------------|-------------------------------------|------|------------|-------|
| All ports (3)    |           |               |             |                                     |      |            |       |
| Scalar ports (3) |           |               |             |                                     |      |            |       |
| in1              | IN        |               | P5          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| in2              | IN        |               | P4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| out              | OUT       |               | G4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |

八、实验结果

示例：

|      |     |     |
|------|-----|-----|
| 输入   | In1 | In2 |
| 输入端口 | P5  | P4  |
| 输入的值 | 0   | 1   |
| 输出   | out |     |
| 输出端口 | G4  |     |
| 输出的值 | 1   |     |

(示例照片见下页)



## 九、实验体会

通过此次试验，进一步地联系了门级建模，也更加熟悉 vivado 的编程及使用流程（包括模拟 simulation，综合 synthesis，根据硬件设备指定 I/O Ports，生产比特流 generate bitstream，到最后把程序放到设备里面），还有对每一步结果分析。

# 暨南大学本科实验报告专用纸

课程名称 数字逻辑实验 成绩评定                       
实验项目名称 超前进位加法器 指导教师 王传胜 梁倬骞  
实验项目编号 0806003803 实验项目类型 验证 实验地点 N126  
学生姓名 邝庆璇 学号 2016051598  
学院 信息科学技术 系 计算机科学 专业 计算机科学与技术  
实验时间 2019 年 10 月 21 日 上 午

## 一、实验目的

学会用 Verilog HDL 进行数据流建模；  
掌握用 Verilog HDL 写仿真程序；  
掌握使用 Vivado 软件；  
学会超前进位全加器的原理；  
掌握使用 EGO-1 实验板。

## 二、实验内容

用 Verilog HDL 实现 3 位的超前进位加法器。

\*注意：必须使用超前进位实现！！！不能  $\{c4, sum\} = a + b + c0$

## 三、实验程序

expl.v

`timescale 1ns / 1ps

//

// Company:

// Engineer:

//

// Create Date: 2019/10/21 12:43:16

// Design Name:

// Module Name: exp\_1

// Project Name:

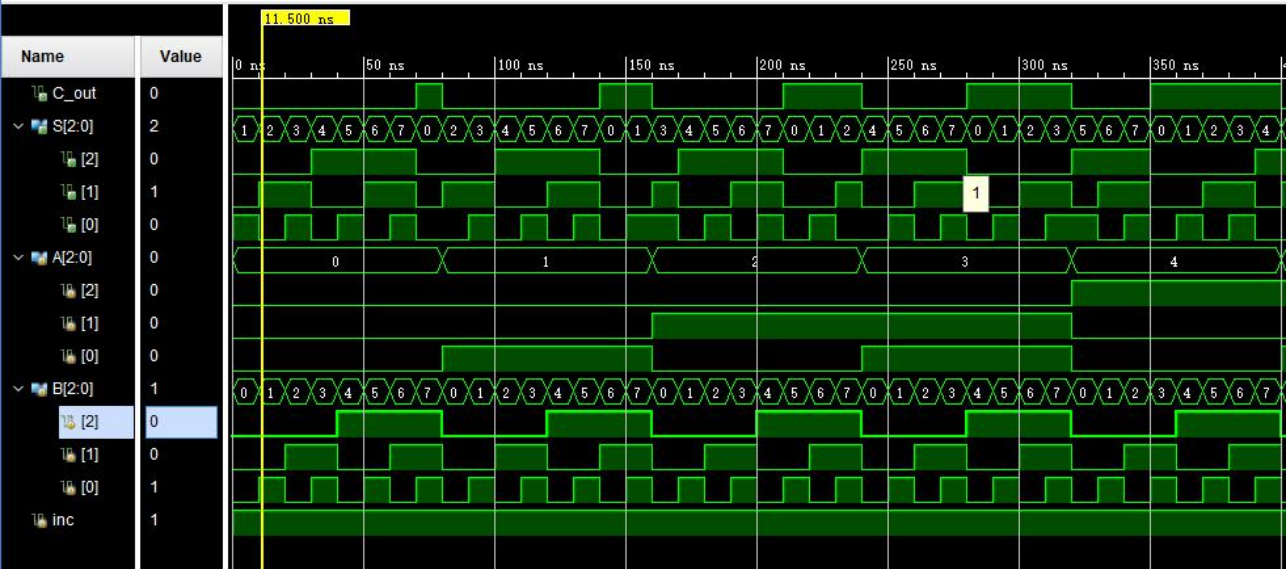


```
// Create Date: 2019/10/21 12:58:48
// Design Name:
// Module Name: sim_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////
```

```
module sim_1(
    output C_out,
    output [2:0] S
);
    reg [5:0] in;
    reg [2:0] A;
    reg [2:0] B;
    reg inc;

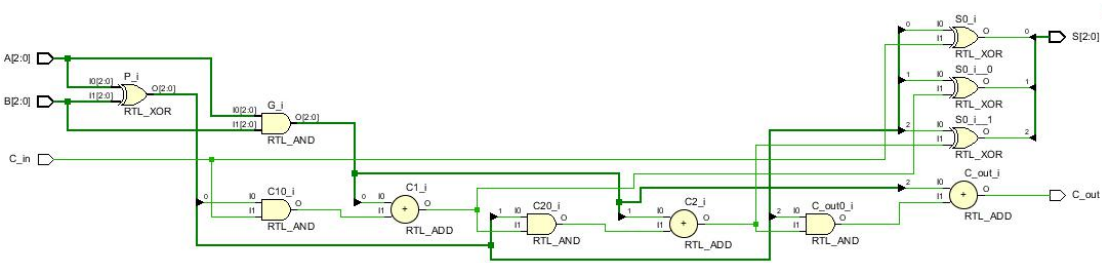
    initial
    begin
        in = 6'b000000;
        assign A = in[5:3];
        assign B = in[2:0];
        assign inc = 1;
    end
    always #10
    begin
        in = in + 1;
        inc = inc + 1;
    end
    exp_1 exp_1(A, B, inc, C_out, S);
endmodule
```

五、仿真结果

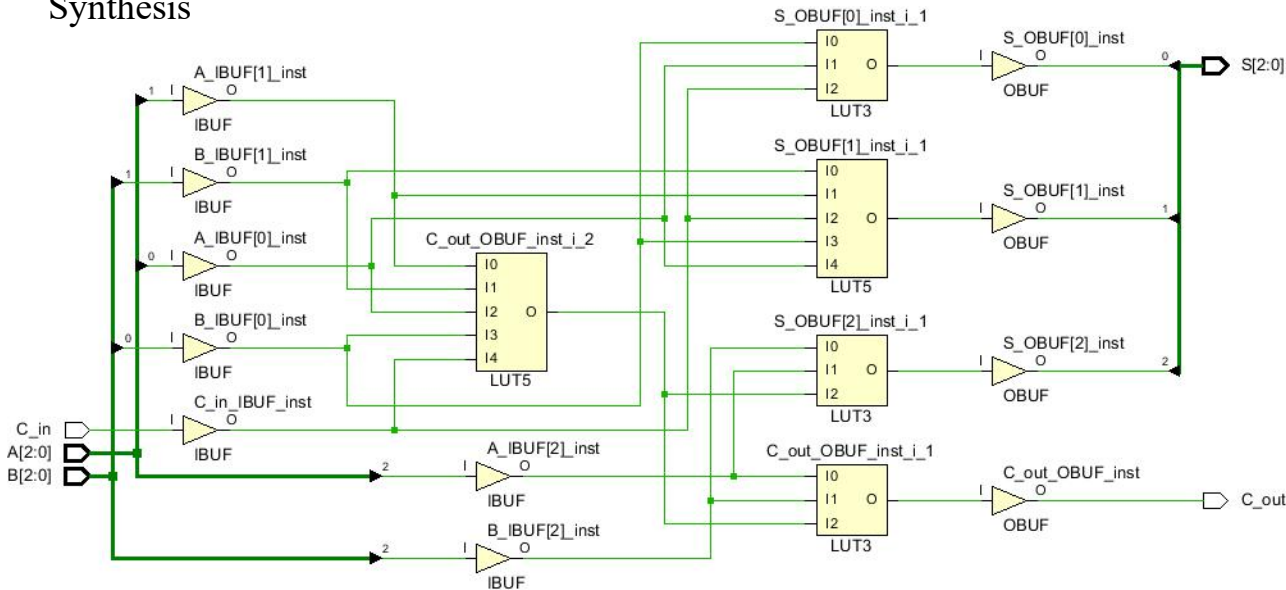


六、系统网表

RTL Analysis



Synthesis





七、系统约束

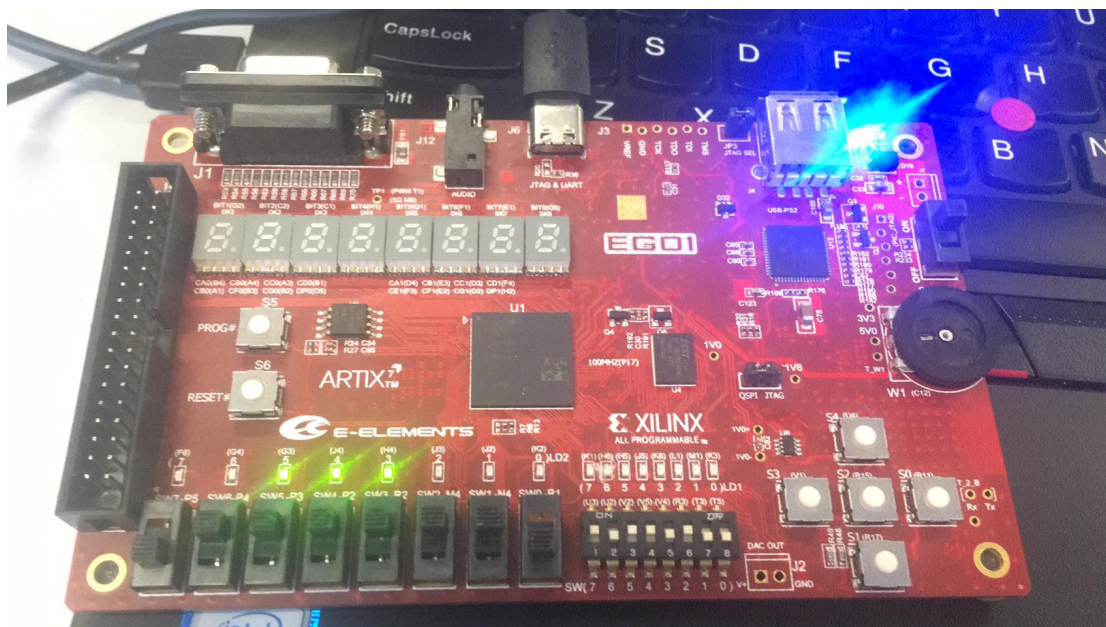
| Name             | Direction | Neg Diff Pair | Package Pin | Fixed                               | Bank | I/O Std    | Vcco  |
|------------------|-----------|---------------|-------------|-------------------------------------|------|------------|-------|
| All ports (11)   |           |               |             |                                     |      |            |       |
| A (3)            | IN        |               |             | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| A[2]             | IN        |               | P5          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| A[1]             | IN        |               | P4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| A[0]             | IN        |               | P3          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| B (3)            | IN        |               |             | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| B[2]             | IN        |               | P2          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| B[1]             | IN        |               | R2          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| B[0]             | IN        |               | M4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| S (3)            | OUT       |               |             | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| S[2]             | OUT       |               | G4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| S[1]             | OUT       |               | G3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| S[0]             | OUT       |               | J4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| Scalar ports (2) |           |               |             |                                     |      |            |       |
| C_in             | IN        |               | N4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| C_out            | OUT       |               | H4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |

八、实验结果

示例 (3+7+1=11=8+3) :

|      |              |              |      |
|------|--------------|--------------|------|
| 输入   | A (A2 A1 A0) | B (B2 B1 B0) | C_in |
| 输入端口 | P5 P4 P3     | P2 R2 M4     | N4   |
| 输入的值 | 0 1 1        | 1 1 1        | 1    |
| 输出   | S (S2 S1 S0) | C_out        |      |
| 输出端口 | G4 G3 J4     | H4           |      |
| 输出的值 | 0 1 1        | 1            |      |

(结果照片见下一页)



## 九、实验体会

通过此次试验，不但学习并练习了 Verilog HDL 进行数据流建模，熟悉了 Verilog 的各种操作符，而且还复习了超前进位全加器的原理。诚然，在编码过程当中，仍存在对 Verilog 语法及 Vivado 软件使用不熟练，导致走了一点弯路（比如没意识到实验程序 exp1.v 不能用 while, for 等语句，仿真文件里才能用），但在请教老师同学后迅速解决了问题，并完成了实验。

# 暨南大学本科实验报告专用纸

课程名称 数字逻辑实验 成绩评定 \_\_\_\_\_  
实验项目名称 译码器 指导教师 王传胜 梁倬骞  
实验项目编号 0806003804 实验项目类型 验证 实验地点 N126  
学生姓名 邝庆璇 学号 2016051598  
学院 信息科学技术 系 计算机科学 专业 计算机科学与技术  
实验时间 2019 年 10 月 28 日 上 午

## 一、实验目的

掌握用 Verilog HDL 进行数据流建模；  
深入理解译码器的原理；  
学会用 Verilog HDL 实现 74LS138；

## 二、实验内容

用 Verilog HDL 语言实现 74LS138 的功能。

## 三、实验程序

```
expl.v
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/10/28 12:13:17
// Design Name:
// Module Name: exp_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
```

```
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////
```

```
module exp_1(
    input [2:0] G,
    input [2:0] A,
    output [7:0] Y
);

    wire E_tmp = (G[0] & ~(G[1]) & ~(G[2]));
    wire A0_not = ~A[0];
    wire A1_not = ~A[1];
    wire A2_not = ~A[2];

    assign Y[0] = ~(A0_not & A1_not & A2_not & E_tmp);
    assign Y[1] = ~(A[0] & A1_not & A2_not & E_tmp);
    assign Y[2] = ~(A0_not & A[1] & A2_not & E_tmp);
    assign Y[3] = ~(A[0] & A[1] & A2_not & E_tmp);
    assign Y[4] = ~(A0_not & A1_not & A[2] & E_tmp);
    assign Y[5] = ~(A[0] & A1_not & A[2] & E_tmp);
    assign Y[6] = ~(A0_not & A[1] & A[2] & E_tmp);
    assign Y[7] = ~(A[0] & A[1] & A[2] & E_tmp);

endmodule
```

#### 四、仿真程序

```
sim_1.v
`timescale 1ns / 1ps
////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/10/28 12:16:20
// Design Name:
// Module Name: sim_1
// Project Name:
```

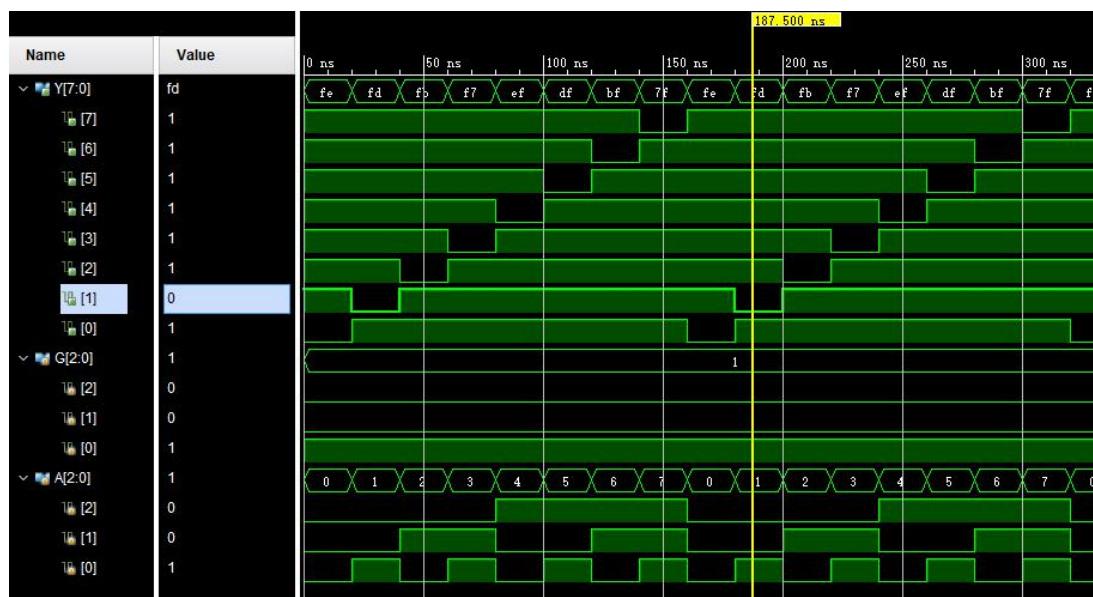
```
// Target Devices:
// Tool Versions:
// Description:
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////
```

```
module sim_1(
    output [7:0] Y
);

    reg [2:0] G;
    reg [2:0] A;

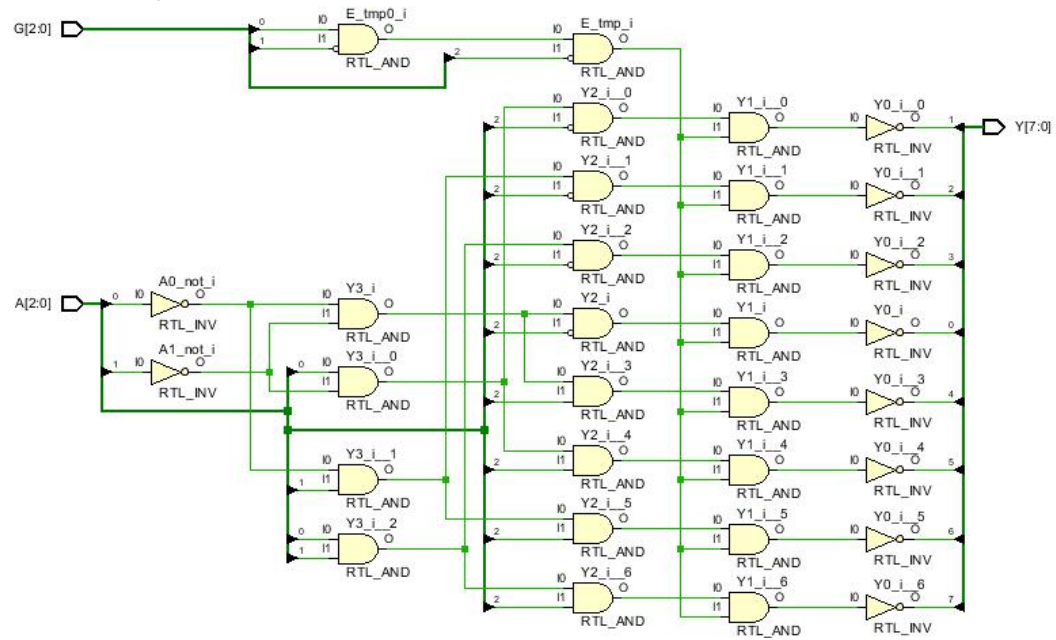
    initial
    begin
        assign A = 3'b000;
        assign G = 3'b001;
    end
    always #20
    begin
        assign A = A + 1;
    end
    exp_1 exp_1(G, A, Y);
Endmodule
```

## 五、仿真结果

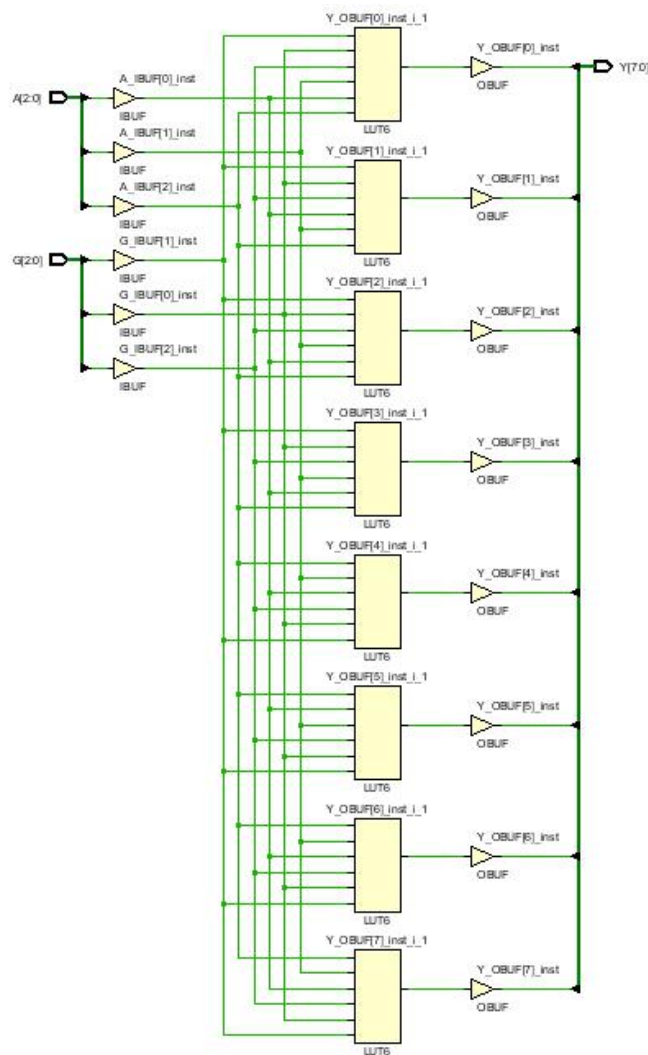


## 六、系统网表

### RTL Analysis



### Synthesis



七、系统约束

| Name             | Direction | Neg ... | Package Pin | Fixed                               | Bank | I/O Std    | Vcco  | Vref |
|------------------|-----------|---------|-------------|-------------------------------------|------|------------|-------|------|
| All ports (14)   |           |         |             |                                     |      |            |       |      |
| A (3)            |           |         |             |                                     |      |            |       |      |
| A[2]             | IN        |         | P5          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |      |
| A[1]             | IN        |         | P4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |      |
| A[0]             | IN        |         | P3          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |      |
| G (3)            |           |         |             |                                     |      |            |       |      |
| G[2]             | IN        |         | P2          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |      |
| G[1]             | IN        |         | R2          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |      |
| G[0]             | IN        |         | M4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |      |
| Scalar ports (0) |           |         |             |                                     |      |            |       |      |
| Y (8)            |           |         |             |                                     |      |            |       |      |
| Y[7]             | OUT       |         | F6          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      |
| Y[6]             | OUT       |         | G4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      |
| Y[5]             | OUT       |         | G3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      |
| Y[4]             | OUT       |         | J4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      |
| Y[3]             | OUT       |         | H4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      |
| Y[2]             | OUT       |         | J3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      |
| Y[1]             | OUT       |         | J2          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      |
| Y[0]             | OUT       |         | K2          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      |

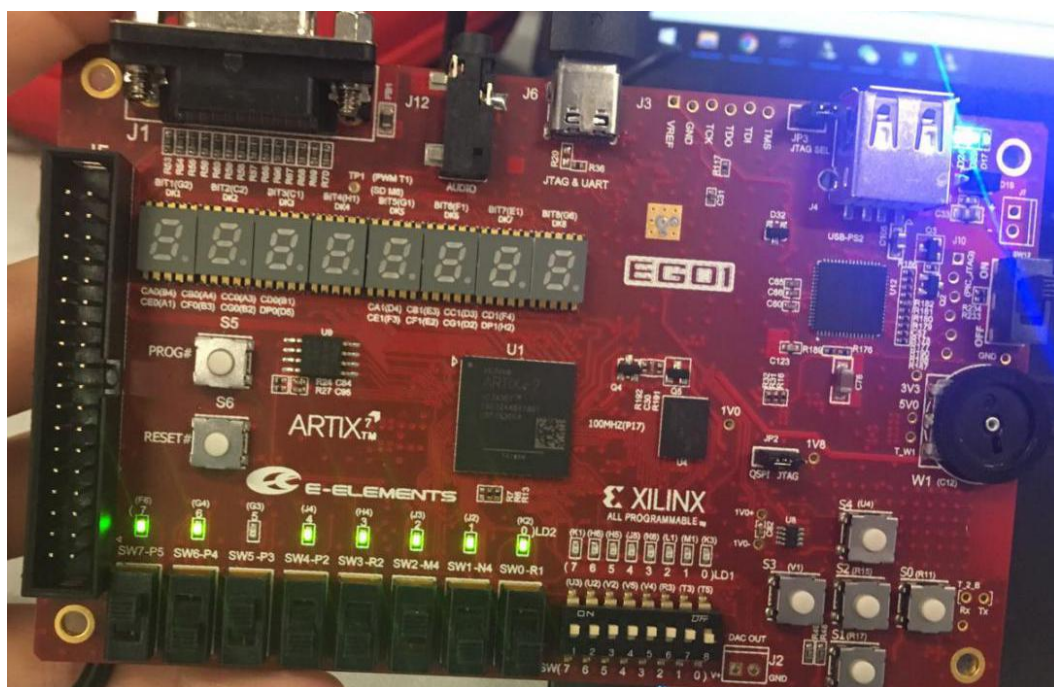
八、实验结果

示例（A=101, G=001）：

|      |                             |  |  |  |              |  |  |  |
|------|-----------------------------|--|--|--|--------------|--|--|--|
| 输入   | A (A2 A1 A0)                |  |  |  | G (G3 G2 G1) |  |  |  |
| 输入端口 | P5 P4 P3                    |  |  |  | P2 R2 M4     |  |  |  |
| 输入的值 | 1 0 1                       |  |  |  | 0 0 1        |  |  |  |
| 输出   | Y (Y7 Y6 Y5 Y4 Y3 Y2 Y1 Y0) |  |  |  |              |  |  |  |
| 输出端口 | F6 G4 G3 J4 H4 J3 J2 K2     |  |  |  |              |  |  |  |
| 输出的值 | 1 1 0 1 1 1 1 1             |  |  |  |              |  |  |  |

（照片见下页）





## 九、实验体会

在做之前的经验积累铺垫下，此次实验完成得比较顺利。但因为一开始的时候是照着图而不是真值表来编码，而图的线路比较复杂（看起来），所以很容易乱；然后改成看真值表，就容易找得到规律了。另外，要注意细心一点儿。



# 暨南大学本科实验报告专用纸

课程名称 数字逻辑实验 成绩评定             
实验项目名称 数字比较器 指导教师 王传胜 梁倬骞  
实验项目编号 0806003805 实验项目类型 验证 实验地点 N126  
学生姓名 邝庆璇 学号 2016051598  
学院 信息科学技术 系 计算机科学 专业 计算机科学与技术  
实验时间 2019 年 11 月 04 日 上 午

## 一、实验目的

掌握 Verilog HDL 进行数据流建模；  
学会使用层次建模的方式来设计整个系统；  
学会用 Verilog HDL 实现数字比较器

## 二、实验内容

用 Verilog HDL 数据流模式实现由级联 2 位数值比较器

## 三、实验程序

```
expl.v
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/11/04 11:28:52
// Design Name:
// Module Name: exp_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
```

```

// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

module exp_1(
    input [1:0] A,
    input [1:0] B,
    input [2:0] I, // I[2]: A>B; I[1]: A<B; I[0]: A==B
    output [2:0] F // F[2]: A>B; F[1]: A<B; F[0]: A==B
);

    // A > B
    assign F[2] = ((A[1] > B[1]) || (A[1] == B[1] && A[0] > B[0]) || (A[1] == B[1] &&
A[0] == B[0] && I[2] > 0));
    // A < B
    assign F[1] = ((A[1] < B[1]) || (A[1] == B[1] && A[0] < B[0]) || (A[1] == B[1] &&
A[0] == B[0] && I[1] > 0));
    // A == B
    assign F[0] = (A[1] == B[1] && A[0] == B[0] && I[0] > 0);

endmodule

```

#### 四、仿真程序

```

sim_1.v
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/11/04 11:37:29
// Design Name:
// Module Name: sim_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//

```

```
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////
```

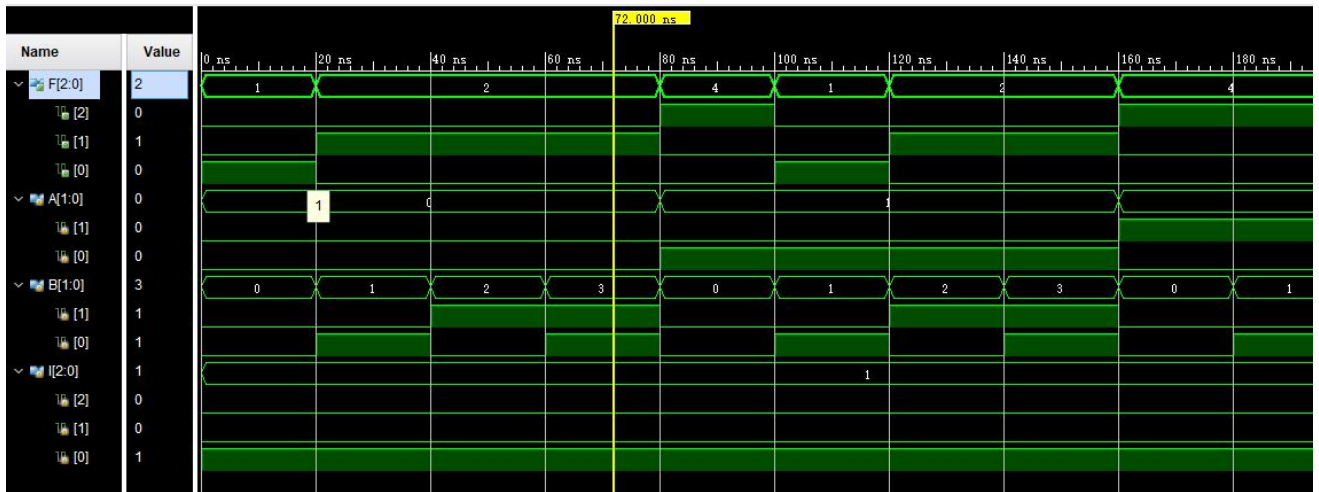
```
module sim_1(
    output [2:0] F
);

    reg [1:0] A;
    reg [1:0] B;
    reg [3:0] AB;
    reg [2:0] I;

    initial
    begin
        AB = 4'b0000;
        A = AB[3:2];
        B = AB[1:0];
        I = 3'b001;
    end
    always #20
    begin
        AB = AB + 1;
        A = AB[3:2];
        B = AB[1:0];
    end
    exp_1 exp_1(A, B, I, F);

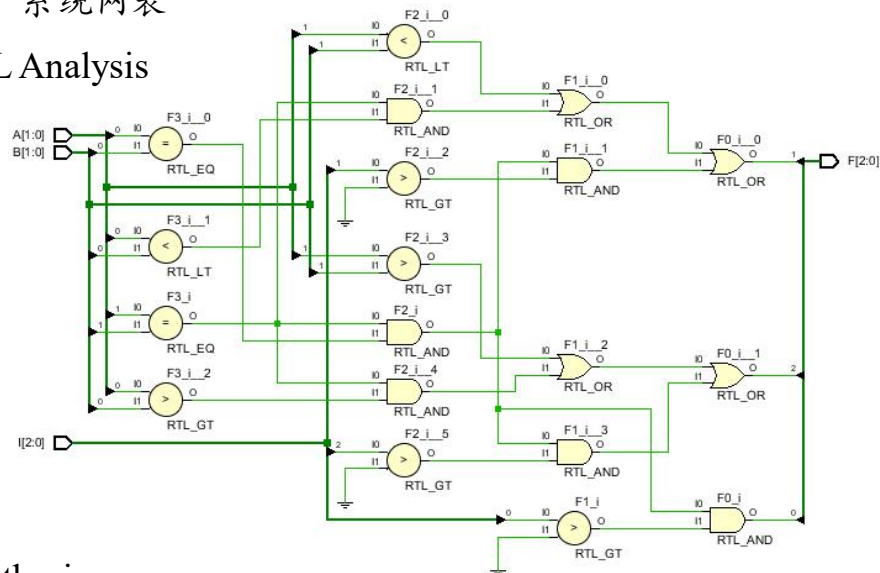
endmodule
```

## 五、仿真结果

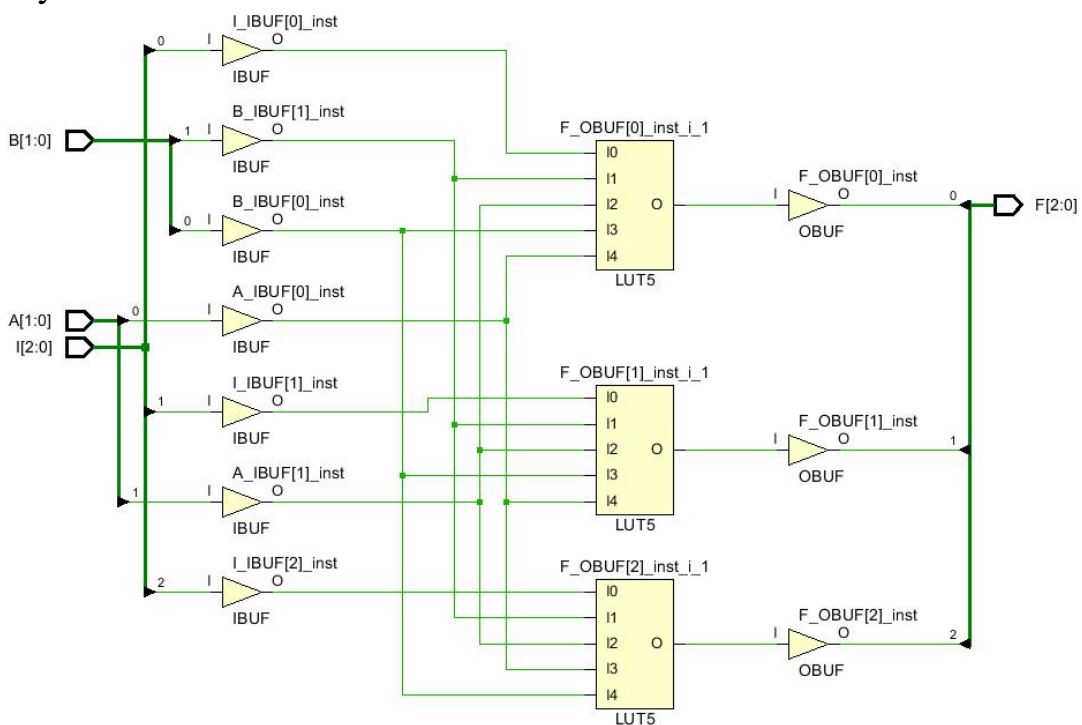


## 六、系统网表

### RTL Analysis



### Synthesis



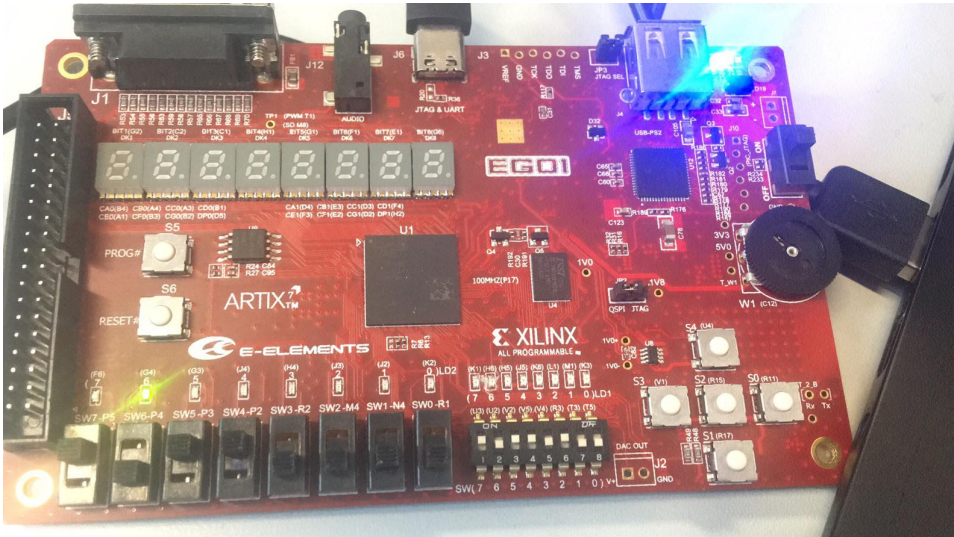
七、系统约束

| Name           | Direction | N... | Package Pin | Fixed                               | Bank | I/O Std    | Vcco  |
|----------------|-----------|------|-------------|-------------------------------------|------|------------|-------|
| All ports (10) |           |      |             |                                     |      |            |       |
| A (2)          | IN        |      |             | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| A[1]           | IN        |      | P5          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| A[0]           | IN        |      | P4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| B (2)          | IN        |      |             | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| B[1]           | IN        |      | P3          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| B[0]           | IN        |      | P2          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| F (3)          | OUT       |      |             | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| F[2]           | OUT       |      | F6          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| F[1]           | OUT       |      | G4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| F[0]           | OUT       |      | G3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| I (3)          | IN        |      |             | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| I[2]           | IN        |      | R2          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| I[1]           | IN        |      | M4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |
| I[0]           | IN        |      | N4          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |

八、实验结果

示例（A=10, B=11, I=100）：

|      |              |          |     |
|------|--------------|----------|-----|
| 输入   | A (A1 A0)    | B(B1 b0) | I   |
| 输入的值 | 1 0          | 11       | 100 |
| 输出   | F (F2 F2 F0) |          |     |
| 输出的值 | 010          |          |     |



## 九、实验体会

在做之前的经验积累铺垫下，此次实验完成得比较顺利。但因为一开始的时候是照着图而不是真值表来编码，而图的线路比较复杂（看起来），所以很容易乱；然后改成看真值表，就容易找得到规律了。另外，要注意细心一点儿。

# 暨南大学本科实验报告专用纸

课程名称 数字逻辑实验 成绩评定             
实验项目名称 七段数码管 指导教师 王传胜 梁倬骞  
实验项目编号 0806003806 实验项目类型 验证 实验地点 N126  
学生姓名 邝庆璇 学号 2016051598  
学院 信息科学技术 系 计算机科学 专业 计算机科学与技术  
实验时间 2019 年 11 月 11 日 上 午

## 一、实验目的

掌握用 Verilog HDL 进行数据流建模；  
学会在 EGO-1 实验板上使用七段数码管；  
学会用 Verilog HDL 实现用七段数码管实现 16 进制数；

## 二、实验内容

编写 Verilog 程序，使得在七段数码管中显示 0~9，A~F 这 16 进制数。

## 三、实验程序

```
expl.v
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/11/11 13:11:15
// Design Name:
// Module Name: exp_11
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
```

```
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////
```

```
module exp_11(
    input [3:0] in_num,
    input [3:0] in_chip,
    output reg [6:0] out_num,
    output [3:0] out_chip
);
    assign out_chip = in_chip;

    always @(in_num)
        case (in_num)
            4'b0000 : out_num = 7'b11111110;
            4'b0001 : out_num = 7'b01100000;
            4'b0010 : out_num = 7'b1101101;
            4'b0011 : out_num = 7'b1111001;
            4'b0100 : out_num = 7'b0110011;
            4'b0101 : out_num = 7'b1011011;
            4'b0110 : out_num = 7'b1011111;
            4'b0111 : out_num = 7'b1110000;
            4'b1000 : out_num = 7'b1111111;
            4'b1001 : out_num = 7'b1111011;
            4'b1010 : out_num = 7'b1110111;
            4'b1011 : out_num = 7'b0011111;
            4'b1100 : out_num = 7'b1001110;
            4'b1101 : out_num = 7'b0111101;
            4'b1110 : out_num = 7'b1001111;
            4'b1111 : out_num = 7'b1000111;
            default : begin
                out_num = 'bx;
                $display("not match");
            end
        endcase
endmodule
```



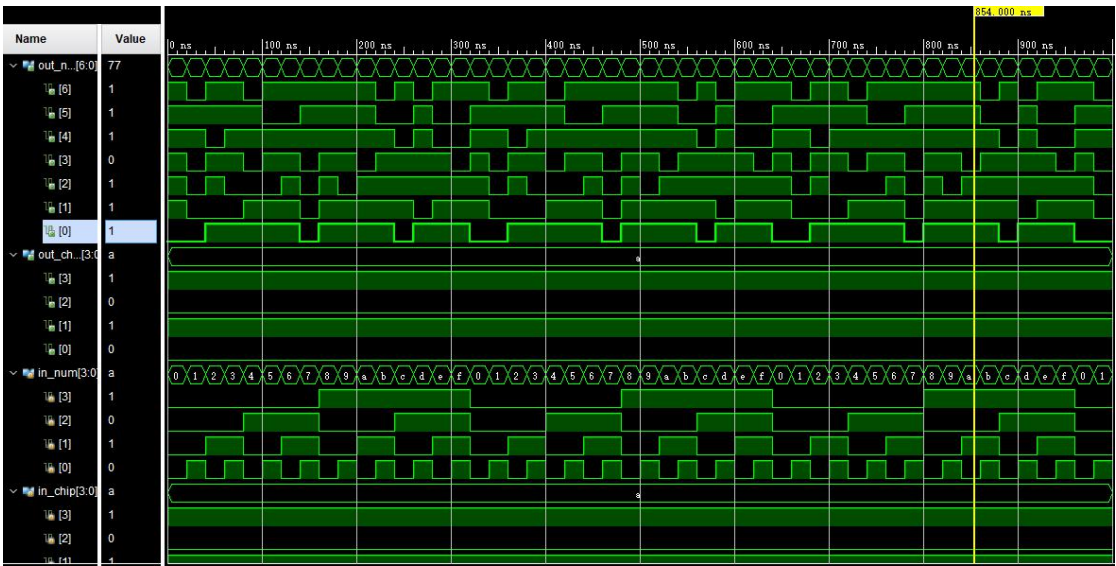
#### 四、仿真程序

```
sim_1.v
endmodule
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/11/11 13:14:02
// Design Name:
// Module Name: sim_11
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

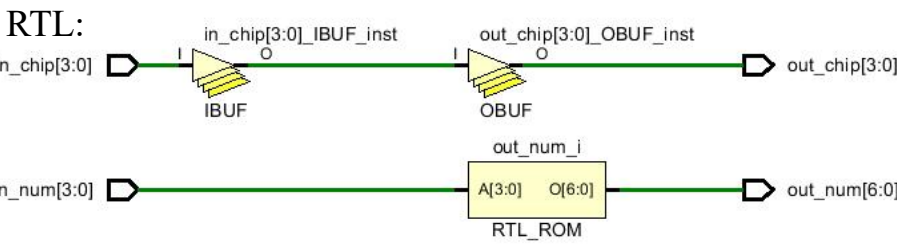
module sim_11(
    output [6:0] out_num,
    output [3:0] out_chip
);

    reg [3:0] in_num;
    reg [3:0] in_chip;
    initial
    begin
        in_chip = 4'b1010;
        in_num = 4'b0000;
    end
    always #20
    begin
        in_num = in_num + 1;
    end
    exp_11 exp_11(in_num, in_chip, out_num, out_chip);
endmodule
```

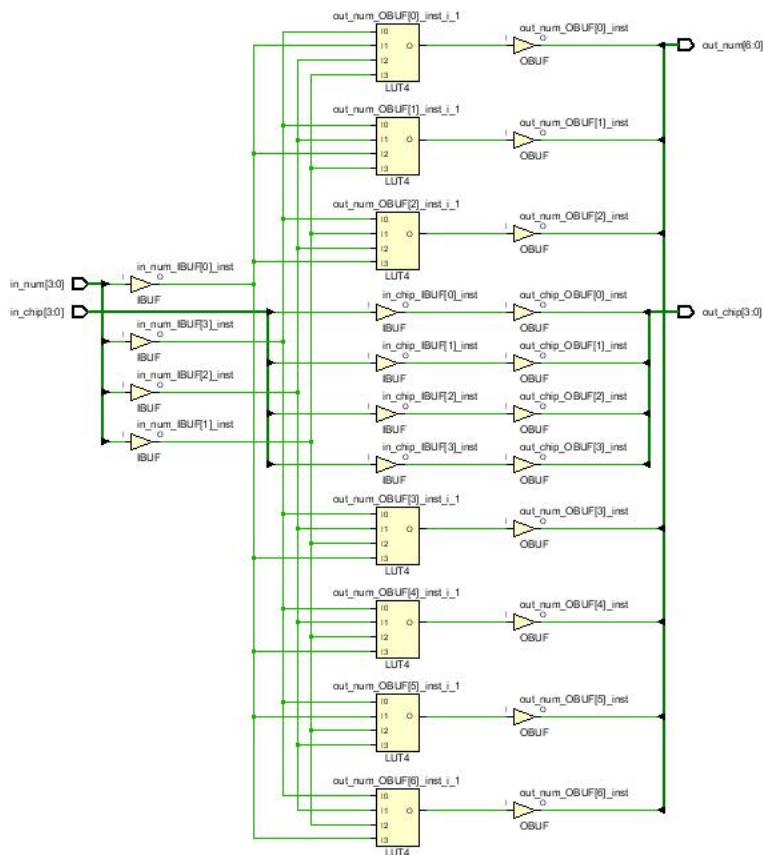
五、仿真结果



六、系统网表



Synthesis:



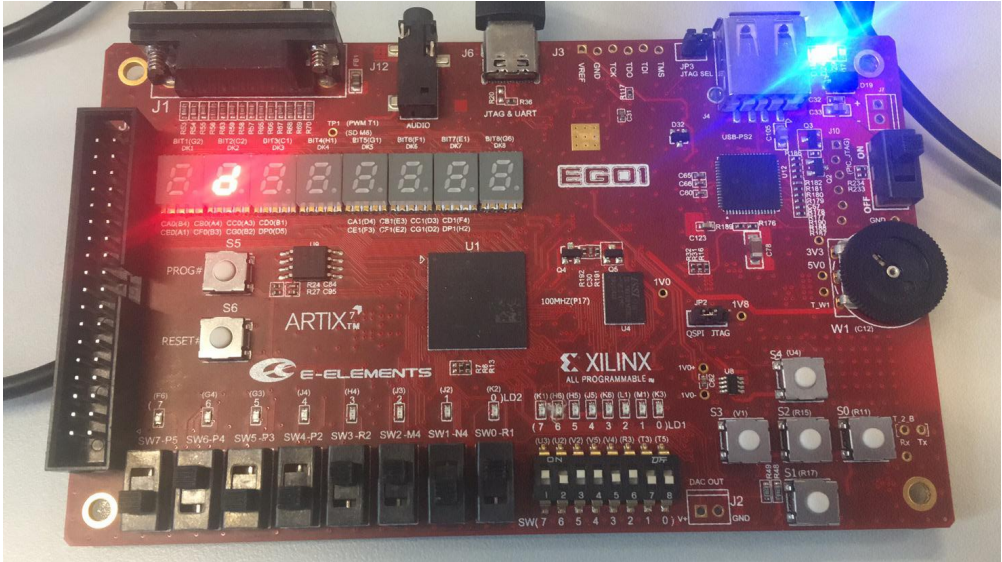
七、系统约束

| Name           | Direction | Neg Diff Pair | Package Pin | Fixed | Bank | I/O Std   | Vcco  | Vref | Drive Strength |
|----------------|-----------|---------------|-------------|-------|------|-----------|-------|------|----------------|
| All ports (19) |           |               |             |       |      |           |       |      |                |
| in_chip (4)    |           |               |             |       |      |           |       |      |                |
| in_chip[3]     | IN        |               | P5          |       | 34   | LVCMOS33* | 3.300 |      |                |
| in_chip[2]     | IN        |               | P4          |       | 34   | LVCMOS33* | 3.300 |      |                |
| in_chip[1]     | IN        |               | P3          |       | 34   | LVCMOS33* | 3.300 |      |                |
| in_chip[0]     | IN        |               | P2          |       | 34   | LVCMOS33* | 3.300 |      |                |
| in_num (4)     |           |               |             |       |      |           |       |      |                |
| in_num[3]      | IN        |               | R2          |       | 34   | LVCMOS33* | 3.300 |      |                |
| in_num[2]      | IN        |               | M4          |       | 34   | LVCMOS33* | 3.300 |      |                |
| in_num[1]      | IN        |               | N4          |       | 34   | LVCMOS33* | 3.300 |      |                |
| in_num[0]      | IN        |               | R1          |       | 34   | LVCMOS33* | 3.300 |      |                |
| out_chip (4)   |           |               |             |       |      |           |       |      |                |
| out_chip[3]    | OUT       |               | G2          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_chip[2]    | OUT       |               | C2          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_chip[1]    | OUT       |               | C1          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_chip[0]    | OUT       |               | H1          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_num (7)    |           |               |             |       |      |           |       |      |                |
| out_num[6]     | OUT       |               | B4          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_num[5]     | OUT       |               | A4          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_num[4]     | OUT       |               | A3          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_num[3]     | OUT       |               | B1          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_num[2]     | OUT       |               | A1          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_num[1]     | OUT       |               | B3          |       | 35   | LVCMOS33* | 3.300 | 12   |                |
| out_num[0]     | OUT       |               | B2          |       | 35   | LVCMOS33* | 3.300 | 12   |                |

八、实验结果

示例（A=101, G=001）：

|      |              |         |
|------|--------------|---------|
| 输入   | in_chip （片选） | in_num  |
| 输入的值 | 0100         | 1101    |
| 输出   | out_chip     | out_num |
| 输出的值 | 0100         | D       |



## 九、实验体会

在做之前的经验积累铺垫下，此次实验完成得比较顺利。但因为一开始的时候是照着图而不是真值表来编码，而图的线路比较复杂（看起来），所以很容易乱；然后改成看真值表，就容易找得到规律了。另外，要注意细心一点儿。

# 暨南大学本科实验报告专用纸

课程名称 数字逻辑实验 成绩评定             
实验项目名称 计数器 指导教师 王传胜 梁倬骞  
实验项目编号 0806003807 实验项目类型 验证 实验地点 N126  
学生姓名 邝庆璇 学号 2016051598  
学院 信息科学技术 系 计算机科学 专业 计算机科学与技术  
实验时间 2019 年 11 月 17 日 上 午

## 一、实验目的

学会使用 Verilog HDL 进行行为建模；  
学会的 Verilog HDL 行为建模来实现计数器；  
学会用 Verilog HDL 行为建模来实现分频器；

## 二、实验内容

用 Verilog HDL 实现 2 位的十六进制计数器，并用七段数码管来显示十六进制数 00~FF。

## 三、实验程序

```
top.v
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/12/03 23:51:03
// Design Name:
// Module Name: top
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
```



```

module div(
    input clk,
    output clk48,
    output clk190
);

    reg[24:0] q = 25'b0;
    always @(posedge clk) begin
        q <= q + 1;
    end

    assign clk190 = q[18];
    assign clk48 = q[24];
//    assign clk190 = clk;
//    assign clk48 = clk;
endmodule

```

## counter.v

```

`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/12/03 23:46:46
// Design Name:
// Module Name: counter
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

module counter(
    input clk48,
    input clr,

```

```

        output reg [7:0] data
    );
    reg isFirst = 1;

    always @(posedge clk48 or posedge clr) begin
        if (clr == 1)
            data <= 0;
        else begin
            if (isFirst) begin
                data <= 0;
                isFirst = 0;
            end
            data <= data + 1;
        end
    end
endmodule

```

## seg\_2.v

```

`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/12/03 23:29:37
// Design Name:
// Module Name: seg_2
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

module seg_2(
    input [7:0] data,
    input clk190,
    input clr,

```



```

output reg [6:0] a2g,
output reg [1:0] an
);

reg s = 0; // 位
reg [3:0] digit;
reg isFirst = 1;

always @(s) begin
    case(s)
        0: digit = data[3:0];
        1: digit = data[7:4];
        default: digit = data[3:0];
    endcase
    case(digit)
        4'b0000: a2g = 7'b11111110;
        4'b0001: a2g = 7'b01100000;
        4'b0010: a2g = 7'b1101101;
        4'b0011: a2g = 7'b1111001;
        4'b0100: a2g = 7'b0110011;
        4'b0101: a2g = 7'b1011011;
        4'b0110: a2g = 7'b1011111;
        4'b0111: a2g = 7'b1110000;
        4'b1000: a2g = 7'b1111111;
        4'b1001: a2g = 7'b1111011;
        4'b1010: a2g = 7'b1110111;
        4'b1011: a2g = 7'b0011111;
        4'b1100: a2g = 7'b1001110;
        4'b1101: a2g = 7'b0111101;
        4'b1110: a2g = 7'b1001111;
        4'b1111: a2g = 7'b1000111;
        default;;
    endcase
    an = 2'b00;
    an[s] = 1;
end

// 1 位计数器
always @(posedge clk190 or posedge clr) begin
    if (clr) s <= 0;
    else    s <= s + 1;
end
endmodule

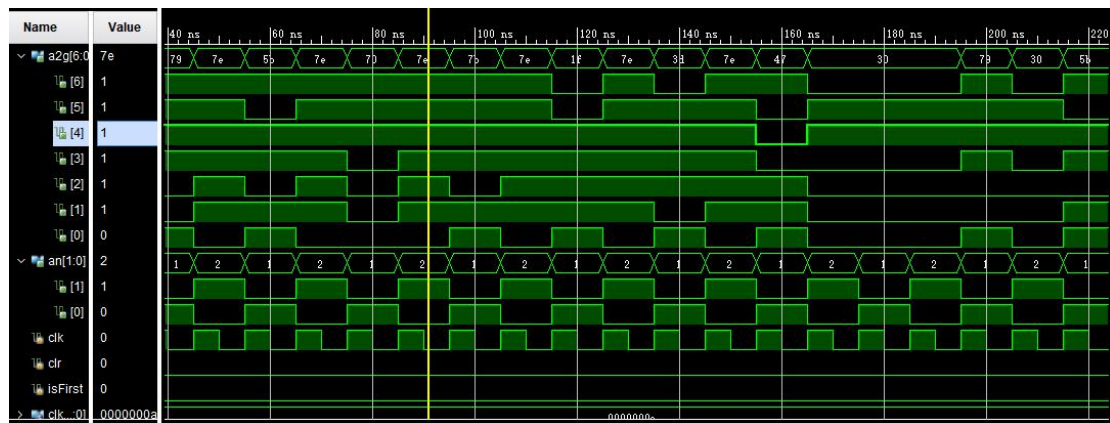
```

#### 四、仿真程序

```
sim_1.v
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/12/03 20:05:06
// Design Name:
// Module Name: sim_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

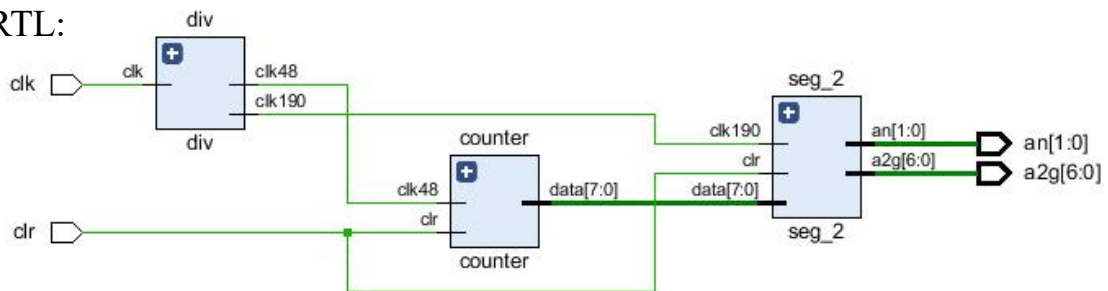
module sim_1(
    output [6:0] a2g,
    output [1:0] an
);
    reg clk, clr;
    reg isFirst = 1;
    parameter clk_period = 10;
    initial begin
        clk = 0;
        clr = 1;
        forever begin
            #(clk_period / 2) clk = ~clk;
            if (isFirst == 1) begin
                isFirst = 0;
                clr = 0;
            end
        end
    end
    top top(.clk(clk), .clr(clr), .a2g(a2g), .an(an));
endmodule
```

## 五、仿真结果

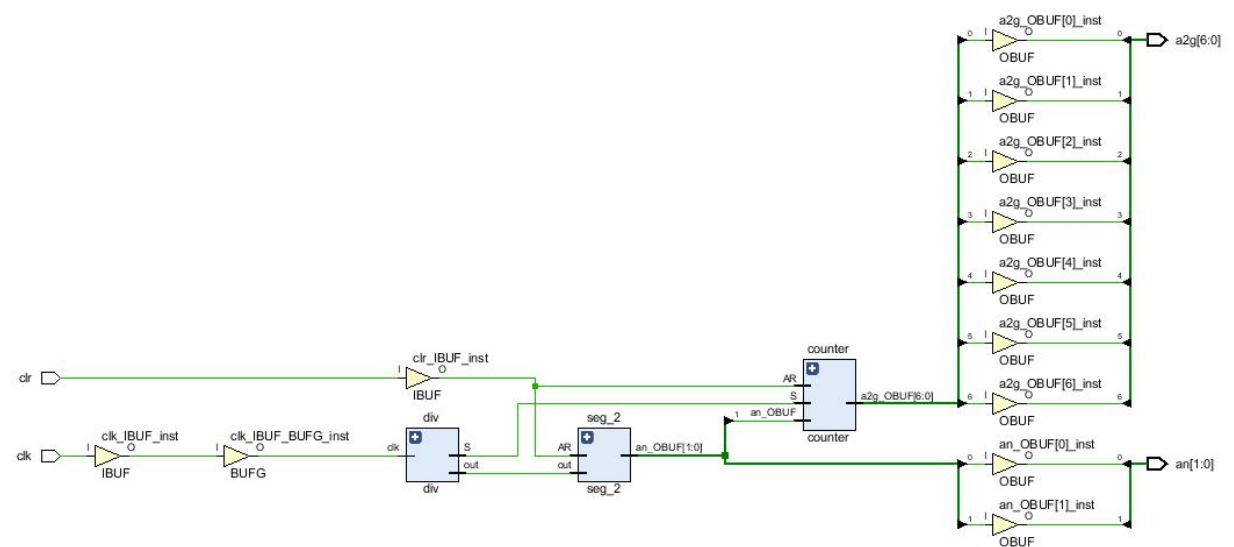


## 六、系统网表

RTL:



Synthesis:

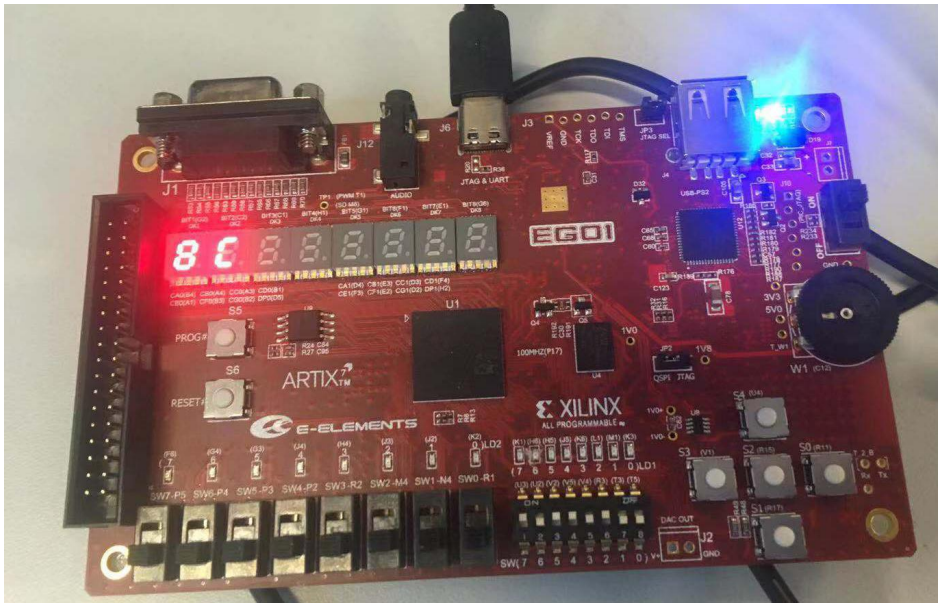


七、系统约束

| Name             | Direction | Neg Diff Pair | Package Pin | Fixed                               | Bank | I/O Std    | Vcco  | Vref | Drive Strength |
|------------------|-----------|---------------|-------------|-------------------------------------|------|------------|-------|------|----------------|
| All ports (11)   |           |               |             |                                     |      |            |       |      |                |
| a2g (7)          | OUT       |               |             | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      | 12             |
| a2g[6]           | OUT       |               | B4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      | 12             |
| a2g[5]           | OUT       |               | A4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      | 12             |
| a2g[4]           | OUT       |               | A3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      | 12             |
| a2g[3]           | OUT       |               | B1          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      | 12             |
| a2g[2]           | OUT       |               | A1          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      | 12             |
| a2g[1]           | OUT       |               | B3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      | 12             |
| a2g[0]           | OUT       |               | B2          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |      | 12             |
| an (2)           | OUT       |               |             | <input checked="" type="checkbox"/> | 35   | LVC MOS25* | 2.500 |      | 12             |
| an[1]            | OUT       |               | G2          | <input checked="" type="checkbox"/> | 35   | LVC MOS25* | 2.500 |      | 12             |
| an[0]            | OUT       |               | C2          | <input checked="" type="checkbox"/> | 35   | LVC MOS25* | 2.500 |      | 12             |
| Scalar ports (2) |           |               |             |                                     |      |            |       |      |                |
| clk              | IN        |               | P17         | <input checked="" type="checkbox"/> | 14   | LVC MOS33* | 3.300 |      |                |
| clr              | IN        |               | P5          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |      |                |

八、实验结果

当 clr=0 时，就会按照设定的频率连续地进行累加。实验结果照片，就是累加到 8C 的情况：



九、实验体会

主要的难点在于需要更加习惯于时序逻辑的设计、以及 verilog 里多模块的编程；并且发现一些语法上概念的还需要熟练，在编程过程中有所耽搁，但也的确学习了。

# 暨南大学本科实验报告专用纸

课程名称 数字逻辑实验 成绩评定                     

实验项目名称 移位寄存器 指导教师 王传胜 梁倬骞

实验项目编号 0806003808 实验项目类型 验证 实验地点 N126

学生姓名 邝庆璇 学号 2016051598

学院 信息科学技术 系 计算机科学 专业 计算机科学与技术

实验时间 2019 年 11 月 28 日 上 午

## 一、实验目的

熟练使用 Verilog HDL 进行行为建模；

学会 Verilog HDL 行为建模来实现移位寄存器；

## 二、实验内容

先用 Verilog HDL 实现 74LS194 的功能，再利用 74LS194 实现 8 位数据双向移位寄存器。

## 三、实验程序

dataShift8.v

`timescale 1ns / 1ps

//

// Company:

// Engineer:

//

// Create Date: 2019/12/01 15:05:17

// Design Name:

// Module Name: dataShift8

// Project Name:

// Target Devices:

// Tool Versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

```

// Additional Comments:
//
/////////////////////////////////////////////////////////////////

module dataShift8(
    input notCR,
    input clk,
    input [1:0] g,
    input [1:0] ds,
    input [7:0] DATA,
    output [7:0] Q
);

    wire clk_out;
    clkdiv clkdiv(.clk_in(clk), .clk(clk_out));

    wire [1:0] dsH, dsL; // 高 4 位/低 4 位的 ds
    assign dsH = {Q[3], ds[1]};
    assign dsL = {ds[0], Q[4]};
    shift_194
    shiftH(.notCR(notCR), .clk(clk_out), .s(g), .ds(dsH), .data(DATA[7:4]), .q(Q[7:4]));
    shift_194
    shiftL(.notCR(notCR), .clk(clk_out), .s(g), .ds(dsL), .data(DATA[3:0]), .q(Q[3:0]));

endmodule

```

clkdiv.v

```

`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/11/25 13:34:41
// Design Name:
// Module Name: clkdiv
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created

```

```
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

    module clkdiv(
        input clk_in,
        output clk
    );
        reg [24:0] q = 25'b0;
        always @(posedge clk_in)
            begin
                q = q + 1;
            end
        assign clk = q[24];
//    assign clk = clk_in;
endmodule
```

## Shift\_194.v

```
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/12/01 14:21:14
// Design Name:
// Module Name: shift_194
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
/////////////////////////////////////////////////////////////////

module shift_194(
    input notCR,
    input clk,
```

```

input [1:0] s, // 移位方向 (高左移低右移)
input [1:0] ds, // 所移入的数字 (左移高右移低)
input [3:0] data,
output reg [3:0] q
);

always @(posedge clk or posedge notCR)
begin
    if (notCR != 1)
        q = 4'b0000;
    else
        begin
            case (s)
                2'b00:;
                2'b01: q = {ds[0], q[3:1]}; // right
                2'b10: q = {q[2:0], ds[1]}; // left
                2'b11: q = data;
                default: q = 4'b0000;
            endcase
        end
    end
endmodule

```

#### 四、仿真程序

```

sim_1.v
`timescale 1ns / 1ps
/////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/12/01 16:50:08
// Design Name:
// Module Name: sim_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created

```



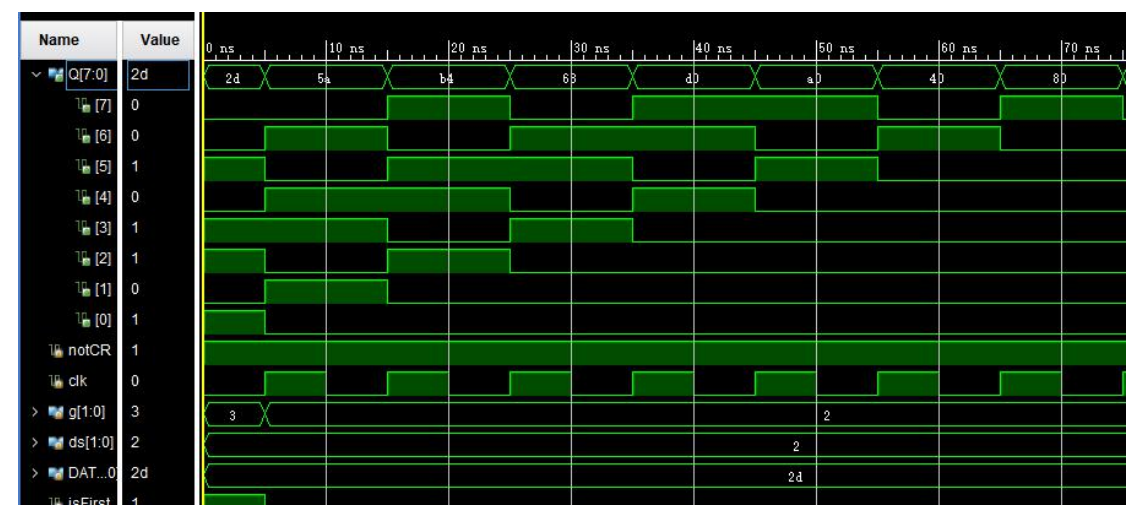
```
// Additional Comments:
```

```
//
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
module sim_1(  
    output [7:0] Q  
);  
    reg notCR;  
    reg clk;  
    reg [1:0] g;  
    reg [1:0] ds;  
    reg [7:0] DATA;  
    reg isFirst;  
    parameter clk_period = 10;  
  
    initial  
    begin  
        clk = 1'b0;  
        notCR = 1'b1;  
        g = 2'b11; // 先送数  
        ds = 2'b10;  
        DATA = 8'b00101101;  
        isFirst = 1'b1;  
        forever  
        begin  
            #(clk_period / 2) clk = ~clk;  
            if (isFirst != 0)  
            begin  
                isFirst = 1'b0;  
                g = 2'b10; // 左移  
            end  
        end  
    end  
end  
dataShift8 dataShift8(.notCR(notCR), .clk(clk), .g(g), .ds(ds), .DATA(DATA), .Q(Q));  
  
endmodule
```

五、仿真结果



六、系统网表

RTL



Synthesis

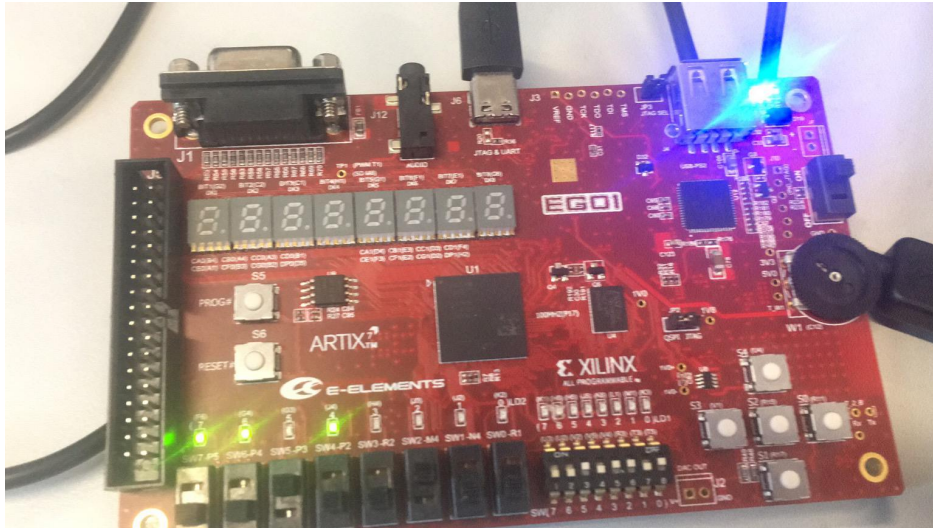


七、系统约束

|                  |     |     |                                     |    |            |       |    |      |      |           |  |
|------------------|-----|-----|-------------------------------------|----|------------|-------|----|------|------|-----------|--|
| All ports (22)   |     |     |                                     |    |            |       |    |      |      |           |  |
| DATA (8)         |     |     |                                     |    |            |       |    |      |      |           |  |
| DATA[7]          | IN  |     | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| DATA[6]          | IN  | U3  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| DATA[5]          | IN  | U2  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| DATA[4]          | IN  | V2  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| DATA[3]          | IN  | V5  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| DATA[2]          | IN  | V4  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| DATA[1]          | IN  | R3  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| DATA[0]          | IN  | T3  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| ds (2)           | IN  | T5  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| ds[1]            | IN  |     | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| ds[0]            | IN  | P5  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| g (2)            | IN  | P4  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| g[1]             | IN  |     | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| g[0]             | IN  | P3  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| Q (8)            | OUT | P2  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| Q[7]             | OUT |     | <input checked="" type="checkbox"/> | 35 | LVC MOS33* | 3.300 | 12 | SLOW | NONE | FP_VTT_50 |  |
| Q[6]             | OUT | F6  | <input checked="" type="checkbox"/> | 35 | LVC MOS33* | 3.300 | 12 | SLOW | NONE | FP_VTT_50 |  |
| Q[5]             | OUT | G4  | <input checked="" type="checkbox"/> | 35 | LVC MOS33* | 3.300 | 12 | SLOW | NONE | FP_VTT_50 |  |
| Q[4]             | OUT | G3  | <input checked="" type="checkbox"/> | 35 | LVC MOS33* | 3.300 | 12 | SLOW | NONE | FP_VTT_50 |  |
| Q[3]             | OUT | J4  | <input checked="" type="checkbox"/> | 35 | LVC MOS33* | 3.300 | 12 | SLOW | NONE | FP_VTT_50 |  |
| Q[2]             | OUT | H4  | <input checked="" type="checkbox"/> | 35 | LVC MOS33* | 3.300 | 12 | SLOW | NONE | FP_VTT_50 |  |
| Q[1]             | OUT | J3  | <input checked="" type="checkbox"/> | 35 | LVC MOS33* | 3.300 | 12 | SLOW | NONE | FP_VTT_50 |  |
| Q[0]             | OUT | K2  | <input checked="" type="checkbox"/> | 35 | LVC MOS33* | 3.300 | 12 | SLOW | NONE | FP_VTT_50 |  |
| Scalar ports (2) |     |     |                                     |    |            |       |    |      |      |           |  |
| clk              | IN  | P17 | <input checked="" type="checkbox"/> | 14 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |
| notCR            | IN  | M4  | <input checked="" type="checkbox"/> | 34 | LVC MOS33* | 3.300 |    |      | NONE | NONE      |  |

## 八、实验结果

比如，把 00101101 进行左移，循环左移入 0，右移到第 4 步，变成 11010000:



## 九、实验体会

主要的难点在于需要更加习惯于时序逻辑的设计、以及 verilog 里多模块的编程；并且发现一些语法上概念的还需要熟练，在编程过程中有所耽搁，但也的确学习了。

# 暨南大学本科实验报告专用纸

课程名称 数字逻辑实验 成绩评定                     

实验名称 时序逻辑电路综合设计（交通信号灯） 指导教师 王传胜 梁倬骞

实验项目编号 0806003808 实验项目类型 验证 实验地点 N126

学生姓名 邝庆璇 学号 2016051598

学院 信息科学技术 系 计算机科学 专业 计算机科学与技术

实验时间 2019 年 12 月 01 日 上 午

## 一、实验目的

理解有限状态机 FSM 的原理；

学会 Verilog HDL 来实现有限状态机 FSM

熟练利用 FSM 解决问题

## 二、实验内容

实现交通信号灯。我们要为一个十字路口交通信号灯设计程序。  
该十字路口分为南北方向和东西方向，每个方向都有红、黄、绿三种颜色信号灯。下图给出了信号灯的状态表：

| 状态 | 南北信号灯 | 东西信号灯 | 持续（秒） |
|----|-------|-------|-------|
| 0  | 绿     | 红     | 6     |
| 1  | 黄     | 红     | 1     |
| 2  | 红     | 绿     | 6     |
| 3  | 红     | 黄     | 1     |
| 0  | 绿     | 红     | 6     |

## 三、实验程序

top.v

```
`timescale 1ns / 1ps
```

```
////////////////////////////////////////////////////////////////
```

```
// Company:
```

```
// Engineer:
```

```
//
```

```
// Create Date: 2019/12/03 17:05:25
```

```
// Design Name:
// Module Name: top
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////
```

```
module top(
    input clk,
    input clr,
    output [5:0] lights
);

    wire clk_3;
    clk_div clkdiv(clk, clr, clk_3);
    traffic traffic(clk_3, clr, lights);
endmodule
```

#### clk\_div.v

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/12/03 17:00:01
// Design Name:
// Module Name: clk_div
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
```

```
//
////////////////////////////////////////////////////////////////

module clk_div(
    input clk,
    input clr,
    output clk_3
);

    reg [24:0] q;
    always @(posedge clk or posedge clr) begin
        if (clr == 1) q <= 0;
        else q <= q + 1;
    end

    //    assign clk_3 = q[24];
    assign clk_3 = clk;
endmodule
```

## traffic.v

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/12/02 17:09:54
// Design Name:
// Module Name: traffic
// Project Name:
// Target Devices:
// Tool Versions:
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////

module traffic(
    input clk_3,
    input clr,
```

```
output reg [5:0] lights
);
```

```
reg [1:0] state;
reg [4:0] count;
```

```
// 4 种状态
```

```
parameter S0 = 2'b00, S1 = 2'b01, S2 = 2'b10, S3 = 2'b11;
```

```
// 3HZ, 则 SEC6: 6S, 18 个时钟周期; SEC1: 1s, 3 个时钟周期
```

```
parameter SEC6 = 5'b10001, SEC1 = 5'b00010;
```

```
// 高 3 位: 南北方向灯色 (高到低每位分别代表绿黄红); 低三位: 西东
```

```
parameter LRR = 6'b001001, LRY = 6'b001010, LRG = 6'b001100,
```

```
LYR = 6'b010001, LYY = 6'b010010, LYG = 6'b010100,
```

```
LGR = 6'b100001, LGY = 6'b100010, LGG = 6'b100100;
```

```
// set status
```

```
always @(posedge clk_3 or posedge clr) begin
```

```
    if (clr == 1) begin
```

```
        state <= S0;
```

```
        count <= 0;
```

```
    end
```

```
    else begin
```

```
        case (state)
```

```
            S0:
```

```
                if (count < SEC6) begin
```

```
                    state <= S0;
```

```
                    count <= count + 1;
```

```
                end
```

```
            else begin
```

```
                state <= S1;
```

```
                count <= 0;
```

```
            end
```

```
            S1:
```

```
                if (count < SEC1) begin
```

```
                    state <= S1;
```

```
                    count <= count + 1;
```

```
                end
```

```
            else begin
```

```
                state <= S2;
```

```
                count <= 0;
```

```
            end
```

```
            S2:
```

```
                if (count < SEC6) begin
```

```
                    state <= S2;
```

```
                    count <= count + 1;
```

```

        end
        else begin
            state <= S3;
            count <= 0;
        end
    S3:
        if (count < SEC1) begin
            state <= S3;
            count <= count + 1;
        end
        else begin
            state <= S0;
            count <= 0;
        end
    default:
        state <= S0;
    endcase
end
end

// set lights according to current state
always @(*) begin
    case(state)
        S0: lights = LGR;
        S1: lights = LYR;
        S2: lights = LRG;
        S3: lights = LRY;
        default: lights = LRR;
    endcase
end
endmodule

```

#### 四、仿真程序

```

sim_1.v
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 2019/12/03 17:09:59
// Design Name:

```

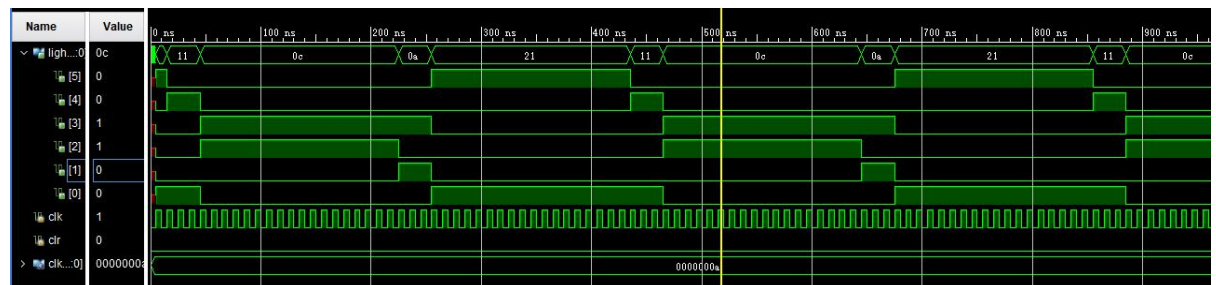


```
// Module Name: sim_1
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////
```

```
module sim_1(
    output [5:0] lights
);
    reg clk, clr;
    parameter clk_period = 10;

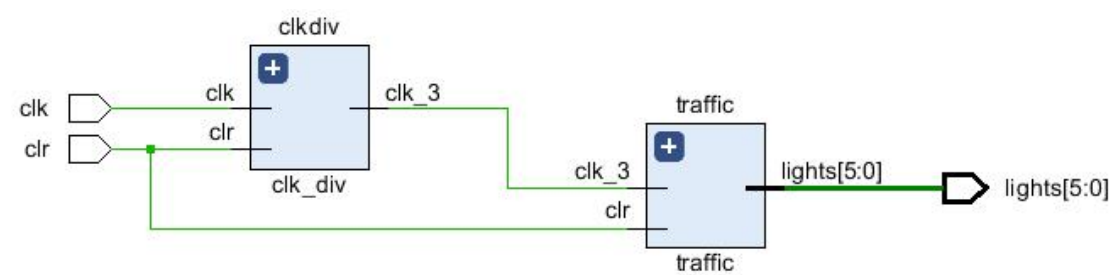
    initial begin
        clr = 1'b0;
        clk = 1'b0;
        forever
            #(clk_period / 2) clk = ~clk;
    end
    top top(clk, clr, lights);
endmodule
```

## 五、仿真结果

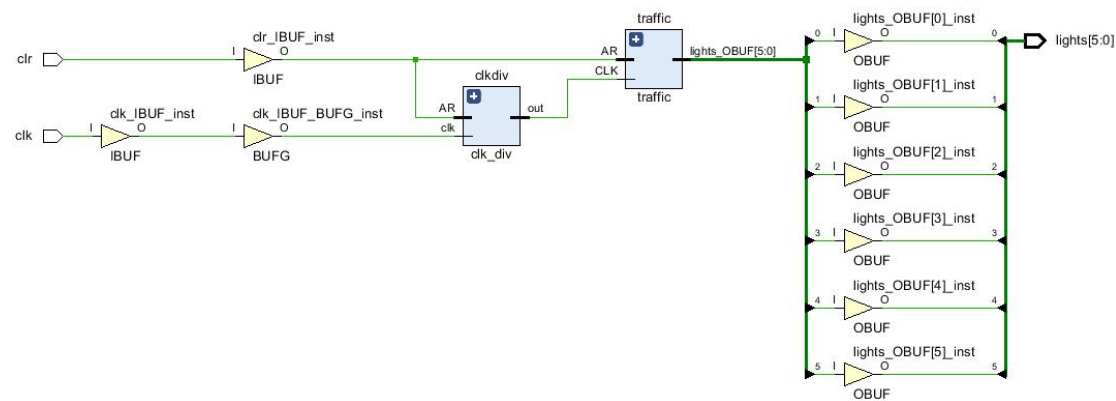


六、系统网表

RTL:



Synthesis:

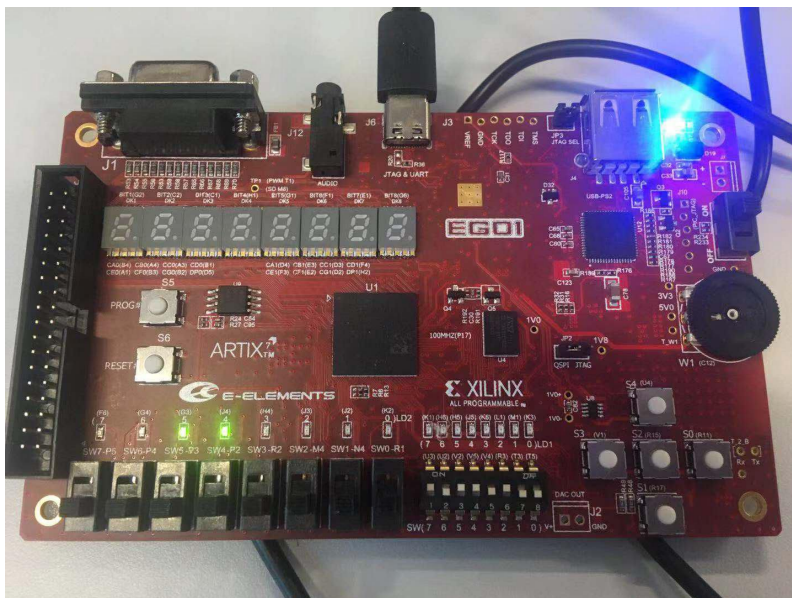


七、系统约束

| Name             | Direction | Neg Diff Pair | Package Pin | Fixed                               | Bank | I/O Std    | Vcco  |
|------------------|-----------|---------------|-------------|-------------------------------------|------|------------|-------|
| All ports (8)    |           |               |             |                                     |      |            |       |
| lights (6)       |           |               |             |                                     |      |            |       |
| lights[5]        | OUT       |               | F6          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| lights[4]        | OUT       |               | G4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| lights[3]        | OUT       |               | G3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| lights[2]        | OUT       |               | J4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| lights[1]        | OUT       |               | H4          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| lights[0]        | OUT       |               | J3          | <input checked="" type="checkbox"/> | 35   | LVC MOS33* | 3.300 |
| Scalar ports (2) |           |               |             |                                     |      |            |       |
| clk              | IN        |               | P17         | <input checked="" type="checkbox"/> | 14   | LVC MOS33* | 3.300 |
| clr              | IN        |               | P5          | <input checked="" type="checkbox"/> | 34   | LVC MOS33* | 3.300 |

## 八、实验结果

按照预期，按照 6s 或 1s 的时间间隔轮流着 4 个状态对应的 4 灯的情况：LGR, LYR, LRG, LRY。比如，以下就是 LRG 状态，表示南北为红(R. Red)，东西为绿(G, Green)



## 九、实验体会

经过前面的若干铺垫，对于时序逻辑已经比较熟悉，因此这次实验总体还算顺利。P.S.: 这算是本实验课最后一个实验了，一个学期下来，对 Verilog HDL 数字逻辑的编程有了较好的学习，收获颇丰。感谢老师的帮助与指点！有缘再会！