

# 同济大学计算机系

## 数字逻辑课程实验报告



学 号 2252707

姓 名 陈艺天

专 业 计算机科学与技术

授课老师 张冬冬

## 1. 4 位数据比较器

### 一、实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现 4 位比较器的设计和仿真。设计用来完成两组二进制数大小比较的逻辑电路。

### 二、模块建模

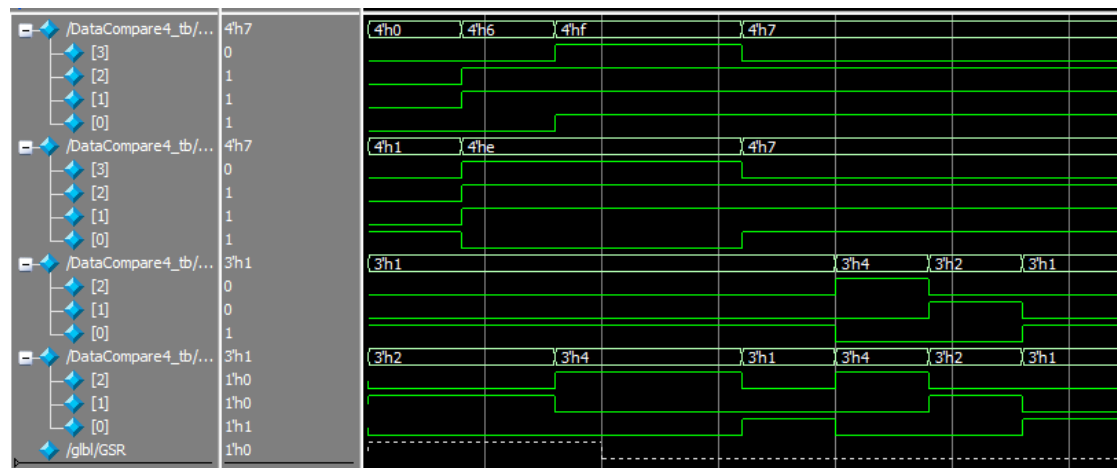
```
module DataCompare4(  
    input [3:0] iData_a,  
    input [3:0] iData_b,  
    input [2:0] iData,  
    output [2:0] oData  
);  
  
    assign oData = iData_a==iData_b?  
        iData:iData_a>iData_b?  
            3'b100:3'b010;  
endmodule
```

### 三、测试模块建模

```
module DataCompare4_tb;  
    reg [3:0] iData_a;  
    reg [3:0] iData_b;  
    reg [2:0] iData;  
    wire [2:0] oData;  
    DataCompare4 inst_datacmp(iData_a,iData_b,iData,oData);  
  
    initial begin  
        iData=3'b001;  
        iData_a=4'b0000;  
        #40 iData_a=4'b0110;  
        #40 iData_a=4'b1111;  
        #40 iData_a=4'b1111;  
        #40 iData_a=4'b0111;  
  
        #40 iData=3'b100;  
        #40 iData=3'b010;  
        #40 iData=3'b001;  
  
    end  
  
    initial begin  
        iData_b=4'b0001;  
        #40 iData_b=4'b1110;  
        #40 iData_b=4'b1110;  
        #40 iData_b=4'b1110;  
        #40 iData_b=4'b0111;  
  
    end  
end  
endmodule
```

# 四、实验结果

## (1) Modelsim 仿真图

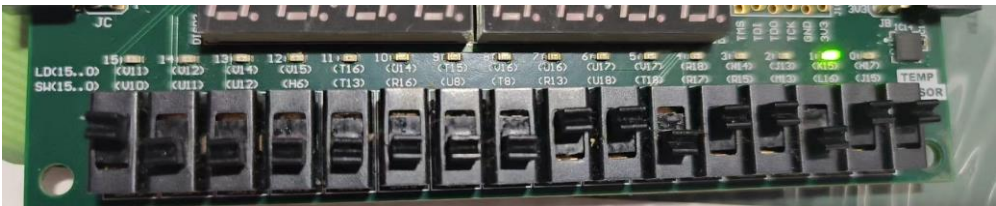


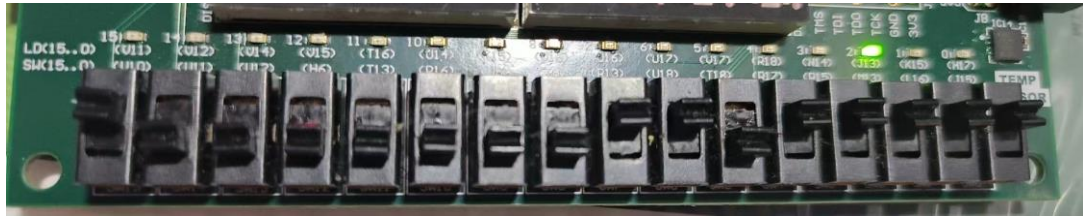
## (2) 下板验证结果

<1>以下三组为两组二进制数相等时，oData 取决于 iData



<2>以下两组为两组二进制数不相等时的结果。





## 2. 8 位数据比较器

### 一、 实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现 8 位比较器的设计和仿真。设计用来完成两组二进制数大小比较的逻辑电路。

### 二、 模块建模

```
module DataCompare4(
    input [3:0] iData_a,
    input [3:0] iData_b,
    input [2:0] iData,
    output [2:0] oData
);

    assign oData = iData_a==iData_b?
        iData:iData_a>iData_b?
        3'b100:3'b010;
endmodule
```

通过两片 DataCompare4 级联实现。

```
module DataCompare8(
    input [7:0] iData_a,
    input [7:0] iData_b,
    output [2:0] oData
);
    reg [2:0] iData_low=3'b001;
    wire [2:0] oData_tmp;
    DataCompare4 low4bits(iData_a[3:0], iData_b[3:0], iData_low, oData_tmp);
    DataCompare4 high4bits(iData_a[7:4], iData_b[7:4], oData_tmp, oData);
endmodule
```

### 三、 测试模块建模

```
module DataCompare8_tb;
    reg [7:0] iData_a;
    reg [7:0] iData_b;
    wire [2:0] oData;

    DataCompare8 inst(iData_a, iData_b, oData);
endmodule
```

```

initial begin
    iData_a=8'b11110000;
    #40 iData_a=8'b10100110;
    #40 iData_a=8'b00001111;
    #40 iData_a=8'b00001111;
    #40 iData_a=8'b10000111;
end

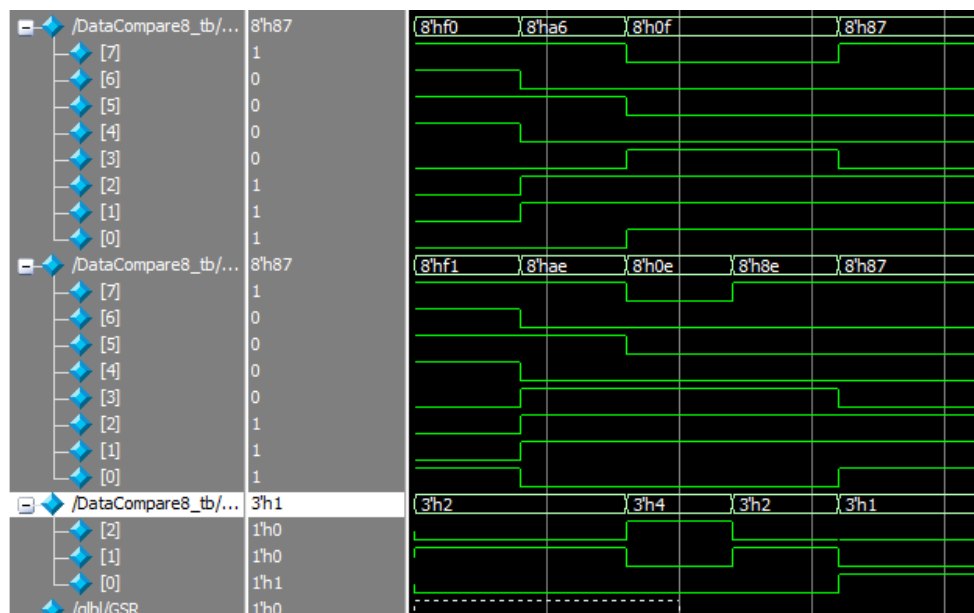
initial begin
    iData_b=8'b11110001;
    #40 iData_b=8'b10101110;
    #40 iData_b=8'b00001110;
    #40 iData_b=8'b10001110;
    #40 iData_b=8'b10000111;
end

endmodule

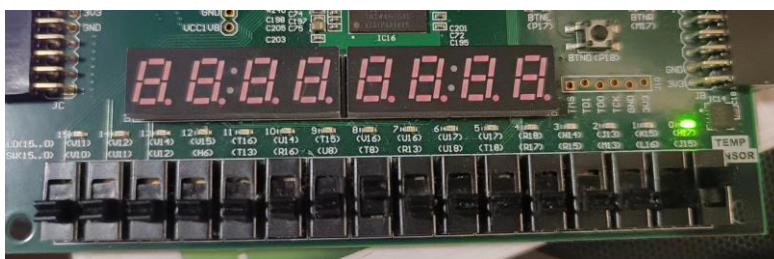
```

## 四、 实验结果

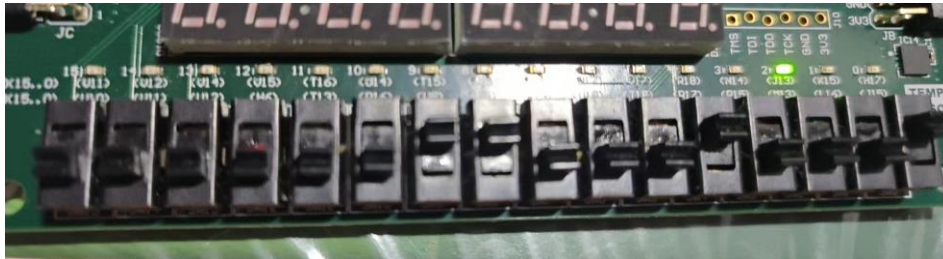
### (1) Modelsim 仿真图



### (2) 下板结果验证







### 3. 1 位加法器

#### 一、 实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现 1 位加法器的设计和仿真。

#### 二、 模块建模

```
module FA(
    input iA,
    input iB,
    input iC,
    output oS,
    output oC
);

    assign oS=iA^iB^iC;
    assign oC=(iA&iB)|(iA&iC)|(iB&iC);
endmodule
```

#### 三、 测试模块建模

```
module FA_tb;
    reg iA;
    reg iB;
    reg iC;
    wire oS;
    wire oC;
    FA fa_ins(iA,iB,iC,oS,oC);

    initial begin
        iA=0;
        #20 iA=1;
        #20 iA=0;
        #20 iA=1;

        #20 iA=0;
        #20 iA=1;
        #20 iA=0;
        #20 iA=1;
    end

    initial begin
        iB=0;
        #20 iB=0;
        #20 iB=1;
        #20 iB=1;

        #20 iB=0;
        #20 iB=0;
        #20 iB=1;
        #20 iB=1;
    end

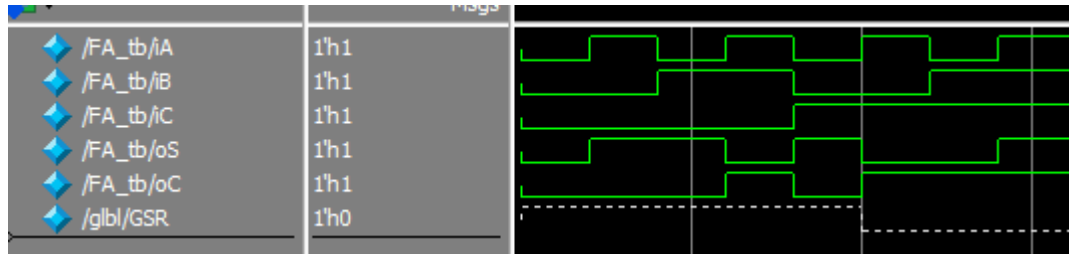
    initial begin
        iC=0;
        #20 iC=0;
        #20 iC=0;
        #20 iC=0;

        #20 iC=1;
        #20 iC=1;
        #20 iC=1;
        #20 iC=1;
    end

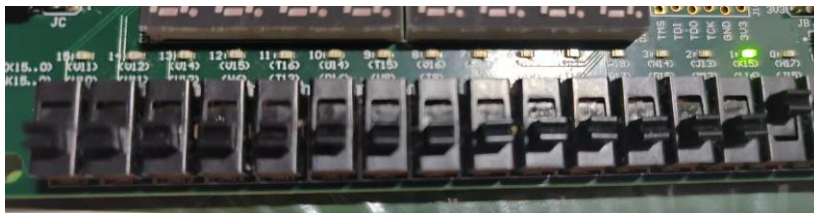
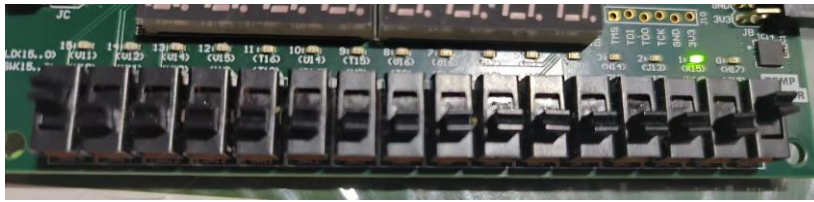
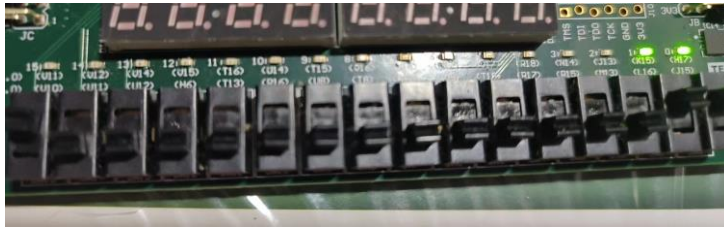
endmodule
```

## 四、实验结果

### (1) Modelsim 仿真图



## (2) 下板结果验证



## 4. 8 位加法器

### 一、 实验内容

在本次实验中，我们将使用 Verilog HDL 语言实现 8 位加法器的设计和仿真。

### 二、 模块建模

```
module FA(  
    input iA,  
    input iB,  
    input iC,  
    output oS,  
    output oC  
);  
  
    assign oS=iA^iB^iC;  
    assign oC=(iA&iB)|(iA&iC)|(iB&iC);  
  
endmodule
```

```
module Adder(  
    input [7:0] iData_a,  
    input [7:0] iData_b,  
    input iC,  
    output[7:0] oData,  
    output oData_C  
);  
    wire o1;  
    wire o2;  
    wire o3;  
    wire o4;  
    wire o5;  
    wire o6;  
    wire o7;  
  
    FA w0(iData_a[0], iData_b[0], iC, oData[0], o1);  
    FA w1(iData_a[1], iData_b[1], o1, oData[1], o2);  
    FA w2(iData_a[2], iData_b[2], o2, oData[2], o3);  
    FA w3(iData_a[3], iData_b[3], o3, oData[3], o4);  
    FA w4(iData_a[4], iData_b[4], o4, oData[4], o5);  
    FA w5(iData_a[5], iData_b[5], o5, oData[5], o6);  
    FA w6(iData_a[6], iData_b[6], o6, oData[6], o7);  
    FA w7(iData_a[7], iData_b[7], o7, oData[7], oData_C);  
  
endmodule
```



### 三、 测试模块建模

```
module Adder_tb;
    reg [7:0] iData_a;
    reg [7:0] iData_b;
    reg iC;
    wire [7:0] oData;
    wire oData_C;

    Adder add_inst(iData_a,iData_b,iC,oData,oData_C);

    initial begin
        iData_a=8'b00010001;
        #20 iData_a=8'b11111110;
        #20 iData_a=8'b00000001;

        #20 iData_a=8'b00010001;
        #20 iData_a=8'b11111110;
        #20 iData_a=8'b00000001;

    end

    initial begin
        iData_b=8'b00010001;
        #20 iData_b=8'b00000001;
        #20 iData_b=8'b11101111;

        #20 iData_b=8'b00010001;
        #20 iData_b=8'b00000001;
        #20 iData_b=8'b11101111;

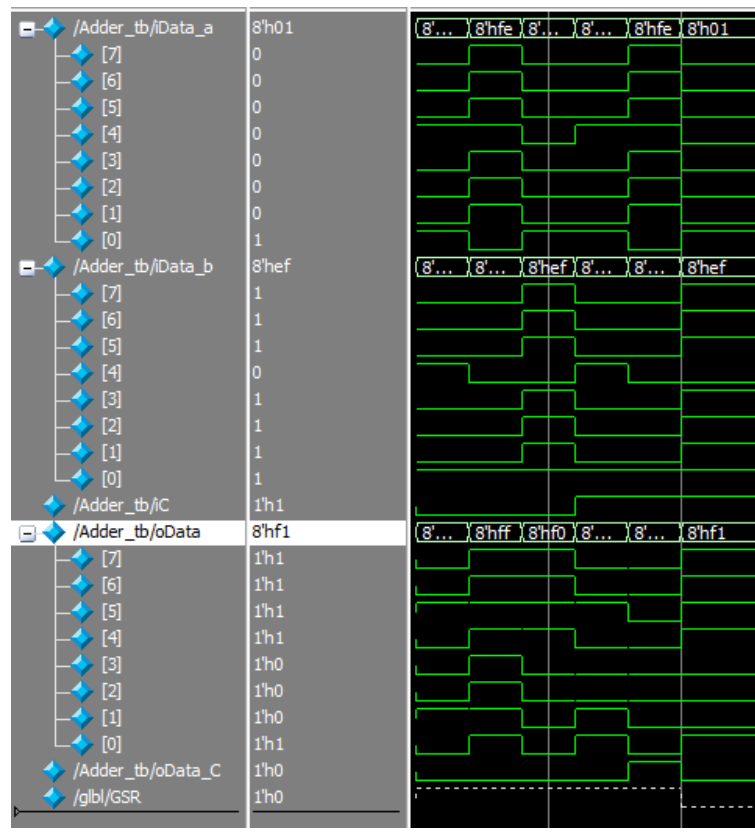
    end

    initial begin
        iC=0;
        #20 iC=0;
        #20 iC=0;
        #20 iC=1;
        #20 iC=1;
        #20 iC=1;

    end
endmodule
```

# 四、 实验结果

(1) Modelsim 仿真图



(2) 下板实验验证

