同济大学计算机系

数字逻辑课程实验报告

学　　号 **2151140**

姓　　名 王谦

专　　业 信息安全

授课老师 郭玉臣

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、8 位比较器、无符号加法器和有符号加法器的设计和仿真。

二、硬件逻辑图

（实验步骤中要求用 logisim 画图的实验，在该部分给出 logisim 原理图，否则该部分在实验报告中不用写）

三、模块建模

（该部分要求对实验中建模的所有模块进行功能描述，并列出各模块建模的 verilog 代码）

1. 比较 iData_a 和 iData_b 的大小，大于则 oData 为 100；小于则 oData 为 010；相等则参考 iData，oData 与 iData 一致。

表 6.5.1 4 位二进制数比较器 74C85 真值表

比较输入				级联输入			输出		
a3b3	a2b2	a1b1	a0b0	a>b	a<b	a=b	A>B	A<B	A=B
a3>b3	×	×	×	×	×	×	1	0	0
a3<b3	×	×	×	×	×	×	0	1	0
a3=b3	a2>b2	×	×	×	×	×	1	0	0
a3=b3	a2<b2	×	×	×	×	×	0	1	0
a3=b3	a2=b2	a1>b1	×	×	×	×	1	0	0
a3=b3	a2=b2	a1<b1	×	×	×	×	0	1	0
a3=b3	a2=b2	a1=b1	a0>b0	×	×	×	1	0	0
a3=b3	a2=b2	a1=b1	a0<b0	×	×	×	0	1	0
a3=b3	a2=b2	a1=b1	a0=b0	1	0	0	1	0	0
a3=b3	a2=b2	a1=b1	a0=b0	0	1	0	0	1	0
a3=b3	a2=b2	a1=b1	a0=b0	0	0	1	0	0	1

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

always@(iData_a or iData_b or iData)
begin
    if(iData_a>iData_b) begin oData=3'b100; end
```

```
    if(iData_a<iData_b) begin oData=3'b010; end
    if(iData_a==iData_b) begin oData=iData; end
end
endmodule
```

2．类似于 1，但是没有了 1 中的 iData，转而将 iData_a 和 iData_b 扩充

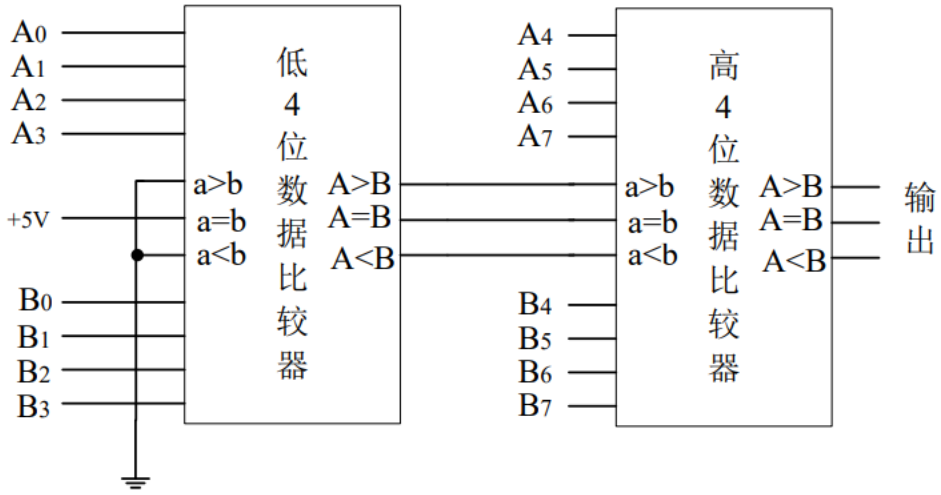
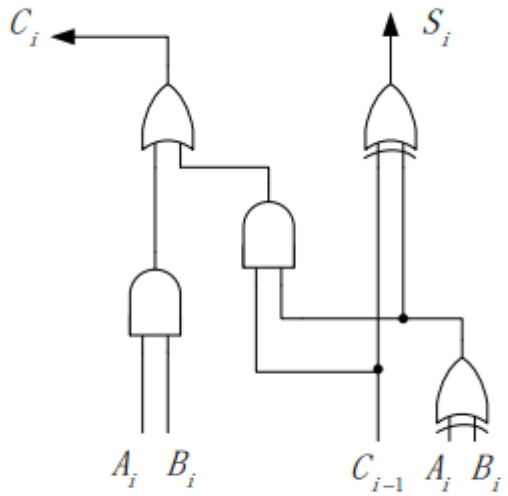


图 6.5.2 使用 2 片 74HC85 组成 8 位比较器

```
module DataCompare4(iData_a,iData_b,iData,oData);
input [3:0] iData_a,iData_b;
input [2:0] iData;
output reg [2:0] oData;
always@(iData_a or iData_b or iData)
begin
    if(iData_a>iData_b) begin oData=3'b100; end
    if(iData_a<iData_b) begin oData=3'b010; end
    if(iData_a==iData_b) begin oData=iData; end
end
endmodule

module DataCompare8(iData_a,iData_b,oData);
    input wire [7:0] iData_a; //  比较数
    input wire [7:0] iData_b; //  比较数
    output reg [2:0] oData; //  比较结果

    always @(iData_a or iData_b) //  行为描述

    begin
        if(iData_a == iData_b)
            begin oData = 1;end
        else if(iData_a < iData_b)
```

```
            begin oData = 2;end
        else if(iData_a > iData_b)
            begin oData = 4;end

    end

endmodule
```

3.

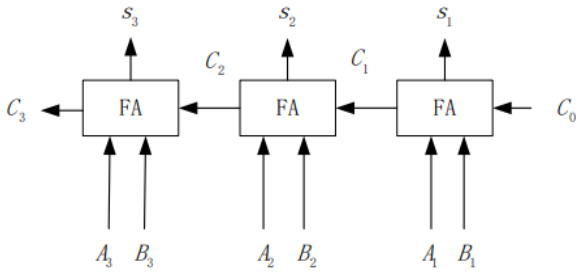


(b) 一位 FA 的逻辑图

利用位运算符简化关系如下所示

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

4.



借助 3 中的模块为基本单位，调用计算 8 次

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

module Adder(oData,oData_C,iData_a,iData_b,iC);
input[7:0]iData_a,iData_b;
input iC;
output[7:0]oData;
output oData_C;
wire [6 : 0] carry;

    FA m0(oData[0], carry[0], iData_a[0], iData_b[0], iC);
    FA m1(oData[1], carry[1], iData_a[1], iData_b[1], carry[0]);
    FA m2(oData[2], carry[2], iData_a[2], iData_b[2], carry[1]);
    FA m3(oData[3], carry[3], iData_a[3], iData_b[3], carry[2]);
    FA m4(oData[4], carry[4], iData_a[4], iData_b[4], carry[3]);
    FA m5(oData[5], carry[5], iData_a[5], iData_b[5], carry[4]);
    FA m6(oData[6], carry[6], iData_a[6], iData_b[6], carry[5]);
    FA m7(oData[7], oData_C, iData_a[7], iData_b[7], carry[6]);

endmodule
```

四、测试模块建模

（要求列写各建模模块的 test bench 模块代码）

1.

```
`timescale 1ns / 1ps

module DataCompare4_tb;
reg [3:0] iData_a,iData_b;
reg [2:0] iData;
wire [2:0] oData;
initial
begin
     iData=3'b100;iData_a=4'b0110;iData_b=4'b0111;//<>
    #50
    iData=3'b001;iData_a=4'b0101;iData_b=4'b1010;//<=
    #50
    iData=3'b010;iData_a=4'b1111;iData_b=4'b1101;//><
```

```
        #50
        iData=3'b001;iData_a=4'b1111;iData_b=4'b1110;//>=
        #50
        iData=3'b100;iData_a=4'b0100;iData_b=4'b0010;//>>
        #50
        iData=3'b001;iData_a=4'b0001;iData_b=4'b0001;//==
        #50
         iData=3'b010;iData_a=4'b0101;iData_b=4'b0101;//=<
         #50
         iData=3'b100;iData_a=4'b0110;iData_b=4'b0110;//=>
        #50
        $stop;
end
DataCompare4
DataCompare4_init(
.iData(iData),
.iData_a(iData_a),
.iData_b(iData_b),
.oData(oData)
);
endmodule
```

2.

```
`timescale 1ns / 1ps

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

        initial
        begin

                iData_a=8'b0000_0000;iData_b=8'b0000_0000;

                #50
                iData_a=8'b0010_0100;iData_b=8'b0001_0010;

                #50
                iData_a=8'b0000_1000;iData_b=8'b0001_0000;

                #50
                iData_a=8'b0001_0010;iData_b=8'b0001_0010;
                #50
```

```
        $stop;
        end

        DataCompare8
        DataCompare8_init(
                .oData(oData),
                .iData_a(iData_a),
                .iData_b(iData_b)
        );

Endmodule
```

3.

```
`timescale 1ns / 1ps

module FA_tb;
reg iA,iB,iC;
wire oS,oC;

initial
begin
iC=0;
iA=0;iB=0;
#50
iA=0;iB=1;
#50
iA=1;iB=1;
#50
iC=1;
iA=0;iB=0;
#50
iA=0;iB=1;
#50
iA=1;iB=0;
#50
iA=1;iB=1;
#50
$stop;
end

FA
FA_init(
.iA(iA),
.iB(iB),
```

```
.iC(iC),
.oS(oS),
.oC(oC)
);
Endmodule
```

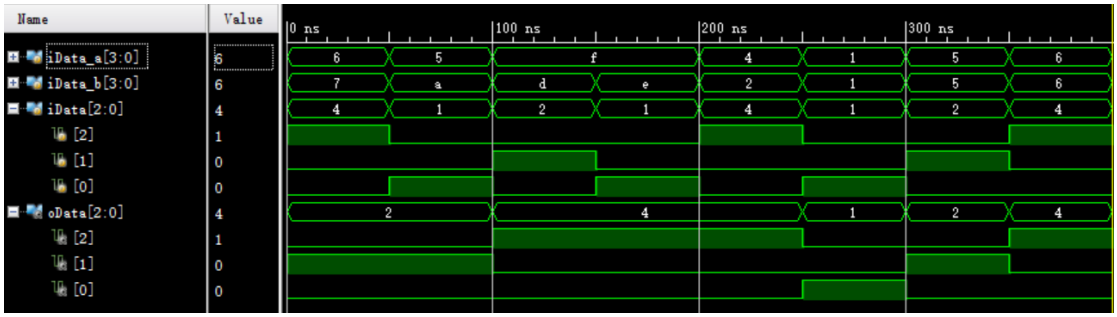
4.

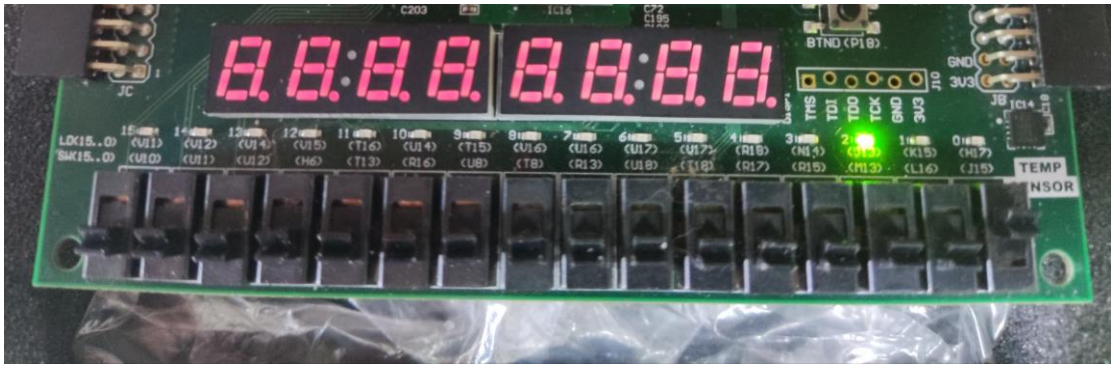
```
`timescale 1ns / 1ps

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

     initial
     begin
          iData_a = 8'b0000_0000; iData_b = 8'b0000_0000; iC = 1'b0; //0 + 0 + 0= 0
          #100 iData_a = 8'b0000_0001; iData_b = 8'b0000_0001; iC = 1'b1; //1 + 1 +
1 = 3
          #100 iData_a = 8'b0010_0000; iData_b = 8'b0010_0011; iC = 1'b1; //32 + 35
+ 1 = 68
          #100 iData_a = 8'b1111_1100; iData_b = 8'b0000_0011; iC = 1'b0; //252 + 3
+ 0 = 255
          #100 iData_a = 8'b1111_1100; iData_b = 8'b0000_0011; iC = 1'b1; //252 + 3
+ 1 = 256
          #100 iData_a = 8'b1111_1100; iData_b = 8'b0000_1000; iC = 1'b0; //252 + 8
+ 0 = 260 (溢出错误)
          #500 $stop;
     end

     Adder
m0(.oData(oData), .oData_C(oData_C), .iData_a(iData_a), .iData_b(iData_b), .iC(iC)
);

Endmodule
```

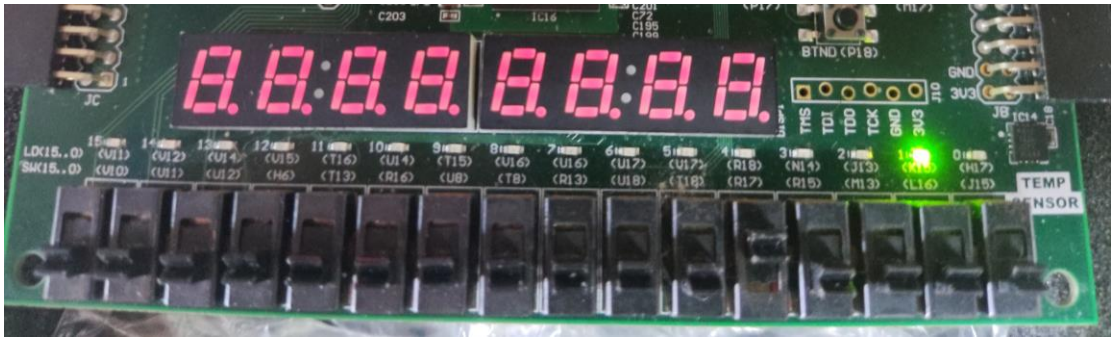
五、实验结果

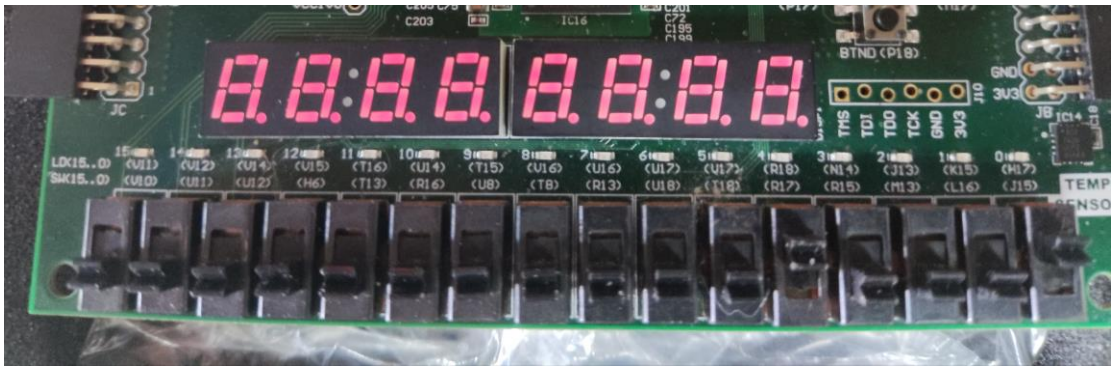
（该部分可截图说明，要求 logisim 逻辑验证图、modelsim 仿真波形图、以及下板后的实验结果贴图（实验步骤中没有下板要求的实验，不需要下板贴图））

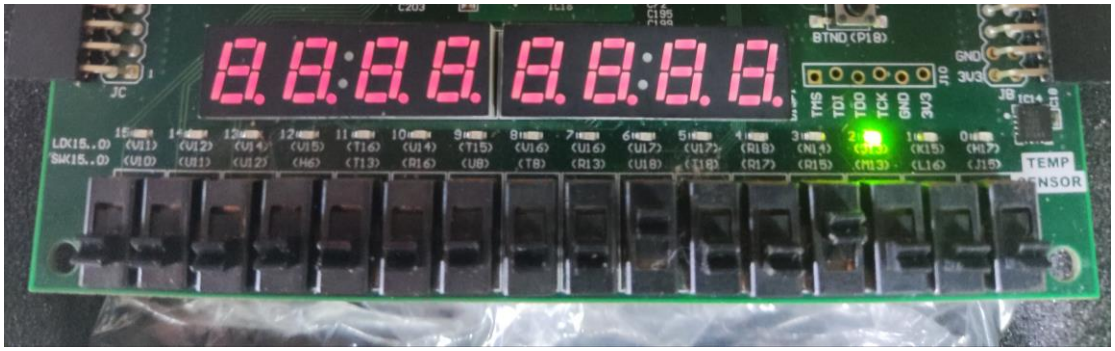
1.


Name
Value
0 ns
100 ns
200 ns
300 ns
iData_a[3:0]
iData_b[3:0]
iData[2:0]
[2]
[1]
[0]
oData[2:0]
[2]
[1]
[0]



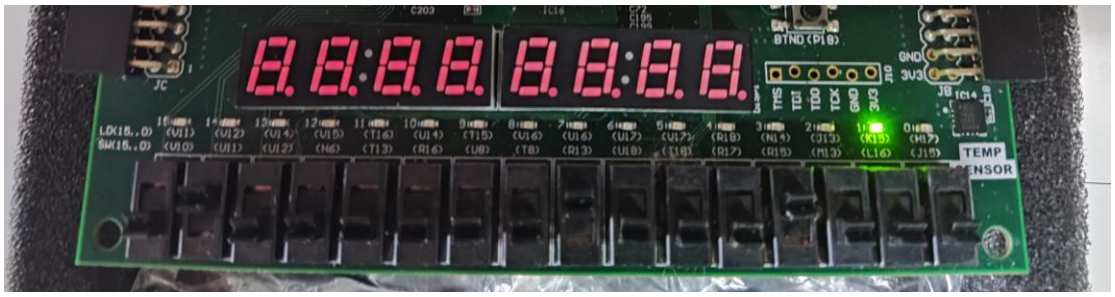






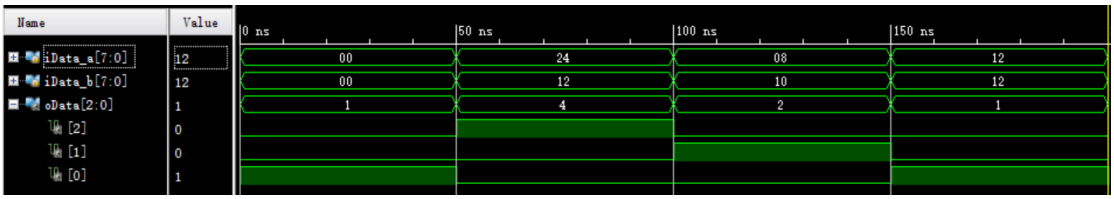

BTND (P18)
GND
3V3
JB
JC
TMS
TDI
TDO
TCK
GND
3V3
J10
TEMP
SENSOR


BTND (P18)
GND
3V3
JB
JC
TMS
TDI
TDO
TCK
GND
3V3
J10
TEMP
SENSOR


BTND (P18)
GND
3V3
JB
JC
TMS
TDI
TDO
TCK
GND
3V3
J10
TEMP
SENSOR

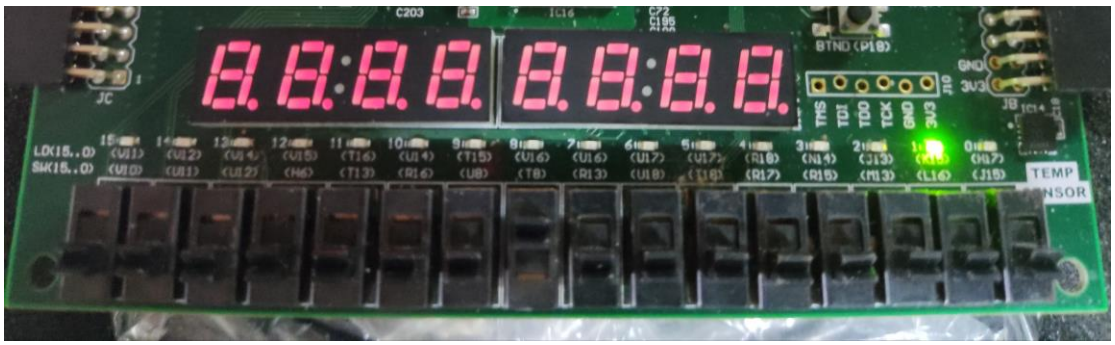

BTND (P18)
GND
3V3
JB
JC
TMS
TDI
TDO
TCK
GND
3V3
J10
TEMP
SENSOR

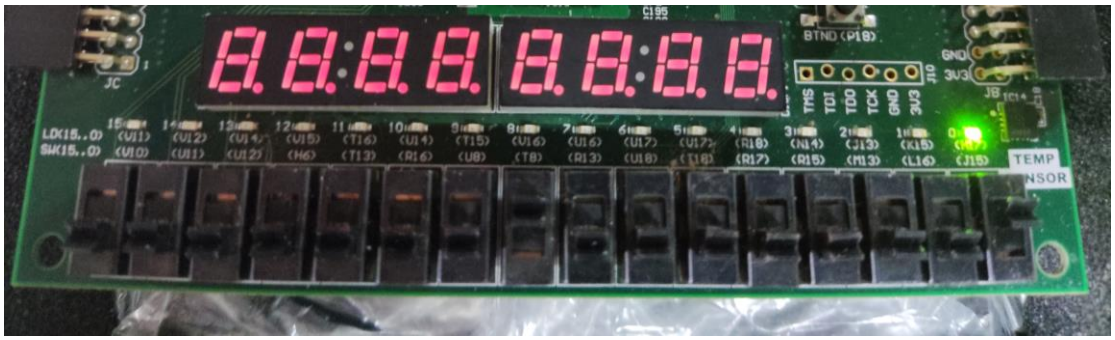
2.


Name
Value
0 ns
50 ns
100 ns
150 ns
iData_a[7:0]
12
00
24
08
12
iData_b[7:0]
12
00
12
10
12
oData[2:0]
1
1
4
2
1
[2]
0
[1]
0
[0]
1


BTND (P18)
JC
JB
TEMP


BTND (P18)
JC
JB
TEMP

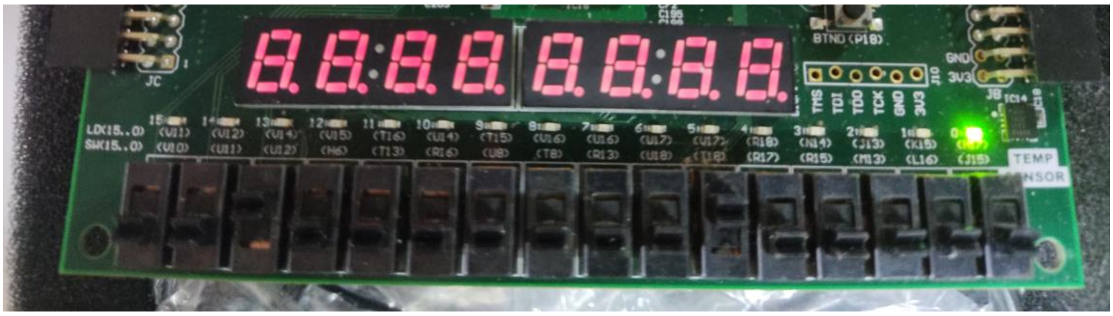

BTND (P18)
JC
JB
TEMP


BTND (P18)
JC
JB
TEMP

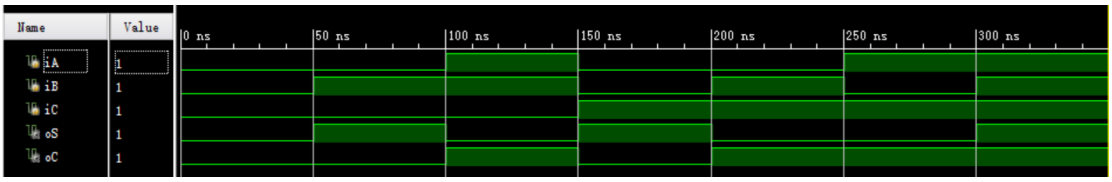

BTND (P18)
TMS TDI TDO TCK GND 3V3
J10
LD(15..0)
SW(15..0)
TEMP
JC

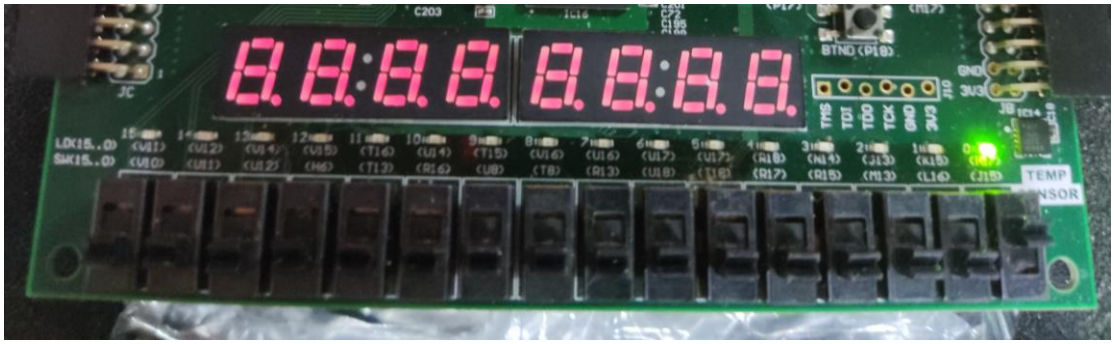

BTND (P18)
TMS TDI TDO TCK GND 3V3
LD(15..0)
SW(15..0)
TEMP
JC

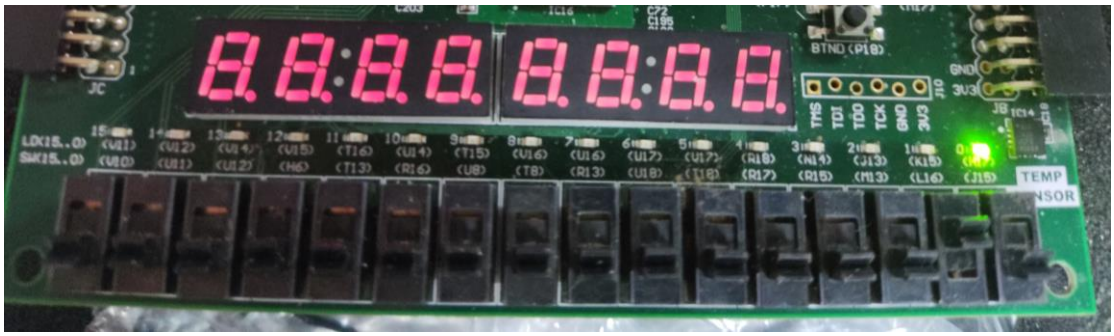

BTND (P18)
TMS TDI TDO TCK GND 3V3
LD(15..0)
SW(15..0)
TEMP
JC


BTND (P18)
TMS TDI TDO TCK GND 3V3
LD(15..0)
SW(15..0)
TEMP
JC

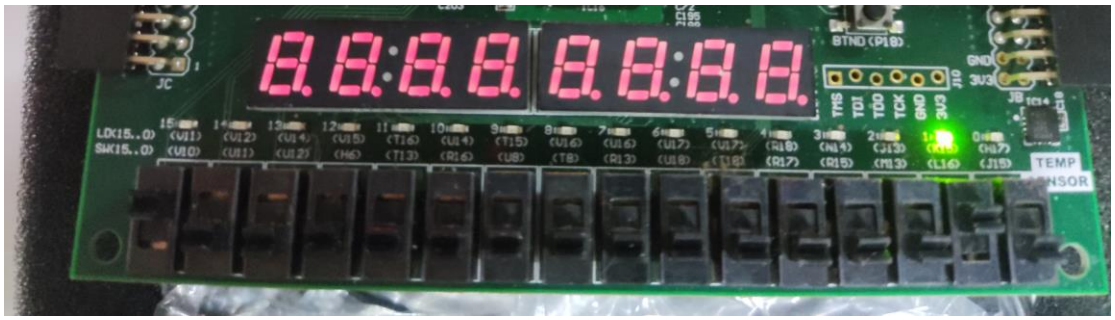
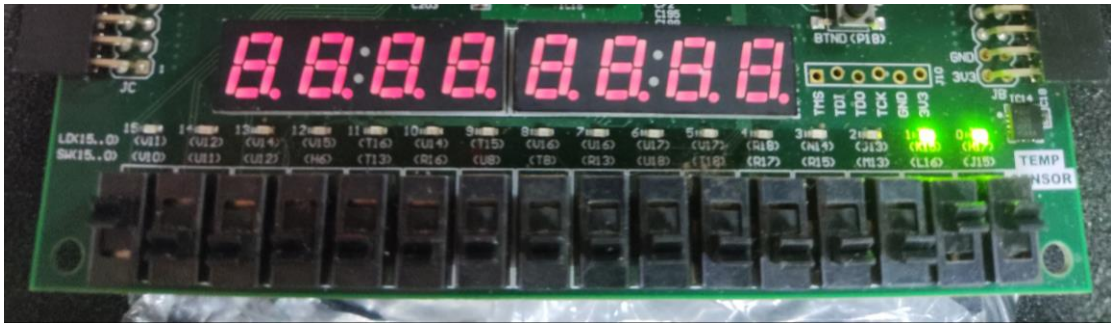
3.


Name
Value
0 ns
50 ns
100 ns
150 ns
200 ns
250 ns
300 ns
iA
iB
iC
oS
oC
1
1
1
1
1

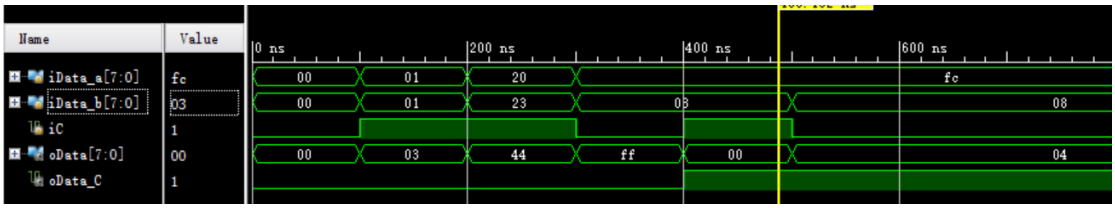




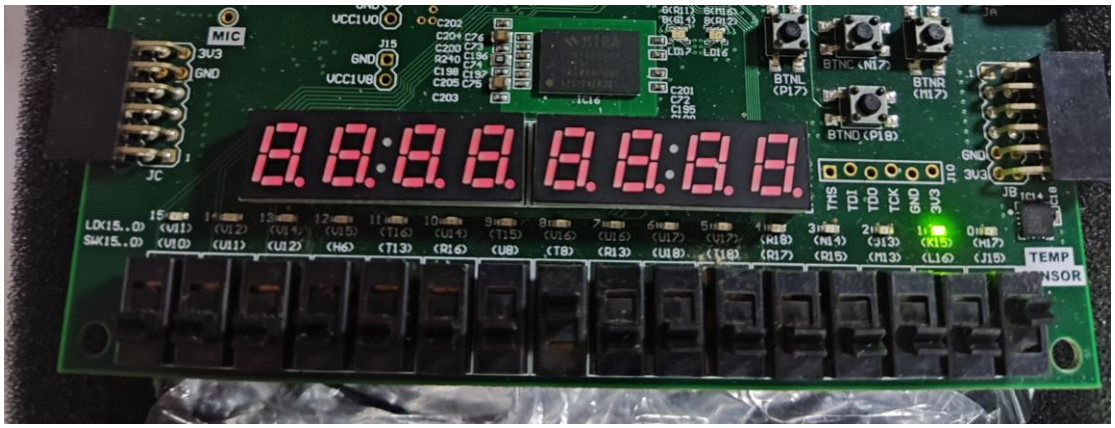


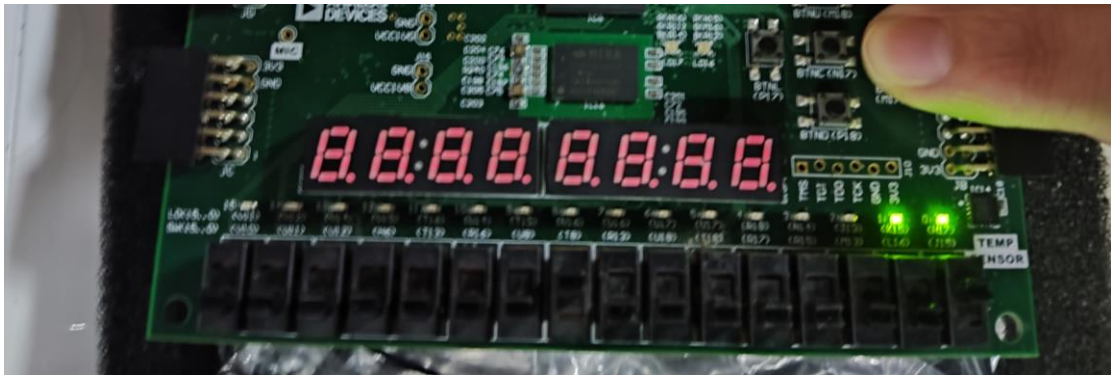
4.



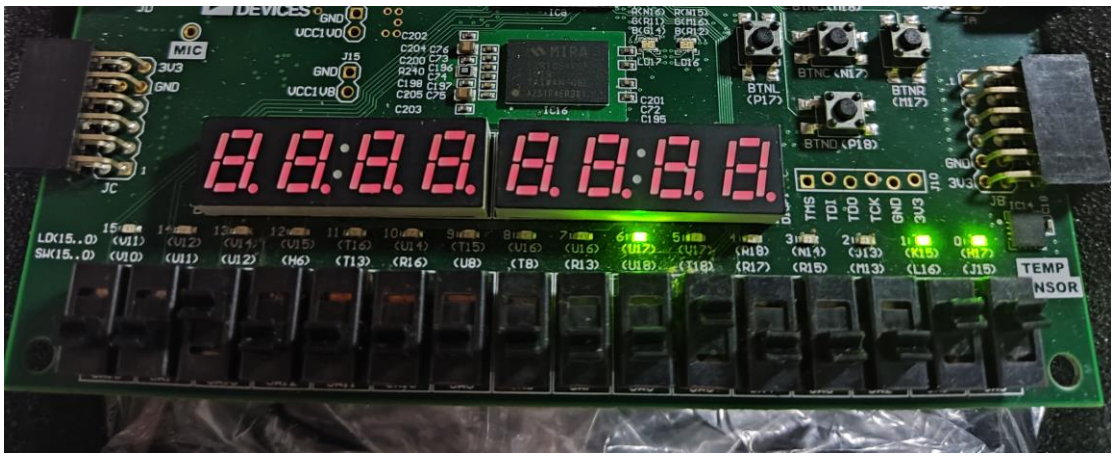
1+1+0=0



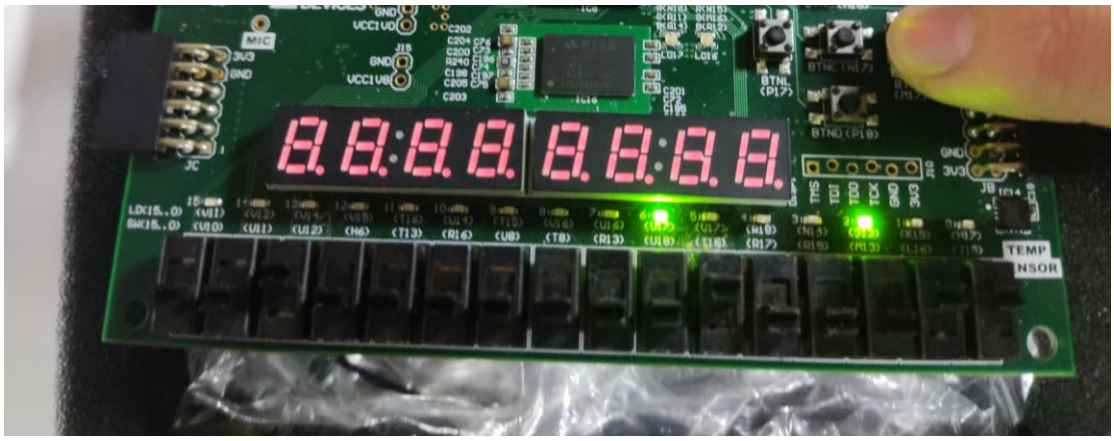
1+1+1=3



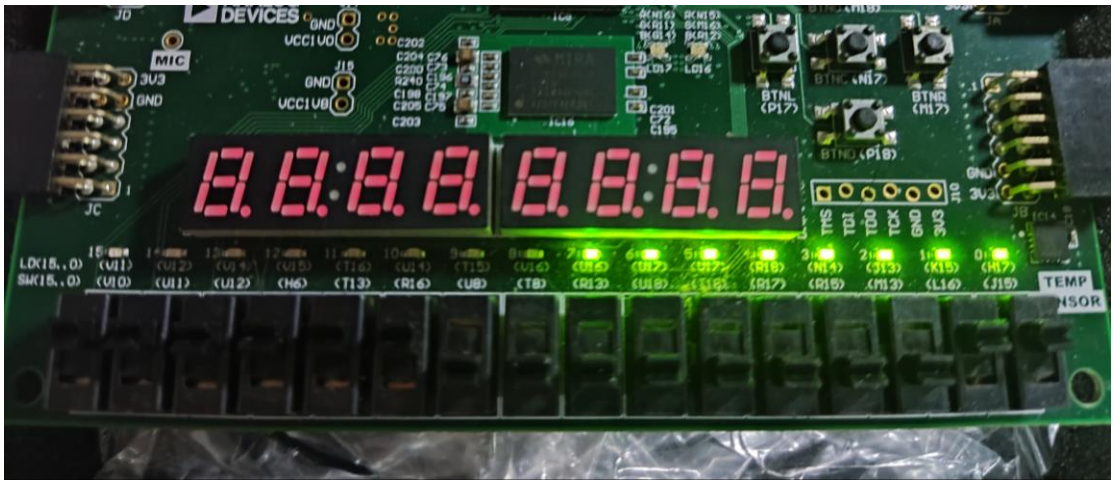
32 + 35 + 0 = 67



32 + 35 + 1 = 68



252 + 3 + 0 = 255



252 + 3 + 1 = 256