

同济大学计算机系

数字逻辑课程实验报告



学 号 2350752

姓 名 田思宇

专 业 计算机科学与技术

授课老师 张冬冬

一、实验内容

深入了解比较器和加法器的原理。使用 Verilog HDL 语言实现 4 位比较器、8 位比较器、一位全加器和八位全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，并列出具体的模块建模的 verilog 代码）

实验一：

模块描述：本模块是四位的数据比较器，输入两个四位信号，以及对应的三位级联信号，输出是三位，最高位代表 $A > B$ ，第二位代表 $A < B$ ，最后一位代表 $A = B$ ，本模块采用行为及描述，使用了 if else 语句，从两个输入信号 A、B 的最高位开始比较，如果不同则输出对应结果，如果相同则比较下一位，为了代码易读，使用了 begin end 语句块，最后如果 $A = B$ ，那么就根据级联输入来判断 A、B 的关系

```
module DataCompare4(
input [3:0] iData_a, //四位输入 a
input [3:0] iData_b, //四位输入 b
input [2:0] iData,    //级联信号 a>b?a<b?a=b
output reg [2:0] oData    //三位输出 A>B、A<B、A=B
);
always@(*)
begin
if(iData_a[3]>iData_b[3])
oData=3'b100;
else if(iData_a[3]<iData_b[3])
oData=3'b010;
else if(iData_a[3]==iData_b[3])
begin
if(iData_a[2]>iData_b[2])
oData=3'b100;
else if(iData_a[2]<iData_b[2])
oData=3'b010;
else if(iData_a[2]==iData_b[2])
```

```

begin
if(iData_a[1]>iData_b[1])
oData=3'b100;
else if(iData_a[1]<iData_b[1])
oData=3'b010;
else if(iData_a[1]==iData_b[1])
begin
if(iData_a[0]>iData_b[0])
oData=3'b100;
else if(iData_a[0]<iData_b[0])
oData=3'b010;
else if(iData_a[0]==iData_b[0])
oData=iData;
end
end
end
end
endmodule

```

实验二：

模块描述：本模块是八位的数据比较器，输入两个把位信号，输出是三位，最高位代表 $A > B$ ，第二位代表 $A < B$ ，最后一位代表 $A = B$ ，本模块采用实例化方法，把第一个实验的四位的数据比较器进行实例化并且级联，最终实现了八位数据比较的功能，如果数据不同，输出对应比较结果、如果数据完全相同，则输出 $A = B$ 。也就是 001。

```

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

```

```

else if(iData_a[2]==iData_b[2])
begin
if(iData_a[1]>iData_b[1])
oData=3'b100;
else if(iData_a[1]<iData_b[1])
oData=3'b010;
else if(iData_a[1]==iData_b[1])
begin
if(iData_a[0]>iData_b[0])
oData=3'b100;
else if(iData_a[0]<iData_b[0])
oData=3'b010;
else if(iData_a[0]==iData_b[0])
oData=iData;
end
end
end
end
endmodule
module DataCompare8(
input [7:0] iData_a,  //????a
input [7:0] iData_b,  //????b
output [2:0] oData
);
wire [2:0] hind;
reg [2:0] low=3'b001; //????????
DataCompare4 uu1(iData_a[7:4],iData_b[7:4],hind,oData); //???
DataCompare4 uu2(iData_a[3:0],iData_b[3:0],low,hind); //???
endmodule

```

实验三：

模块描述：本模块实现的功能是一位二进制的全加器，首先有三个输入，分别是加数和被加数以及上一位的进位，两个输出，一个是本位输出，一个是进位输出。本模块采用实例化的方法进行描述，根据对应的门电路图，首先是 iA、iB 进行异或得到 T0，并将得到的结果与 iC（上一位的进位）异或，最终得到本位输出 oS。

然后是 iA、iB 与运算得到 T1，T0 和 iC 与运算得到 T2，T1 和 T2 再进行或运算得到进位输出 oC。

```

module FA(
input iA,
input iB,
input iC,
output oS,

```

```

output oC
);
xor (T0,iA,iB);
xor (oS,T0,iC);
and (T1,iA,iB);
and (T2,T0,iC);
or (oC,T1,T2);
endmodule

```

实验四：

模块描述： 本模块实现的功能是八位串行二进制的全加器，首先有三个输入，分别是两个八位的操作数 A、B，一位进位输入 iC。两个输出，一个是八位的输出，一个是一位进位输出。由于上一个模块已经编写好了一位全加器，本模块采用实例化的方法进行描述，实例化八个一位全加器并进行级联最终得到八位的全加器。

```

module FA(
input iA, //1 ??????
input iB, //1 ??????
input iC, //???????
output oS, //1 ???
output oC  //????????
);
xor (T0,iA,iB);//??????
xor (oS,T0,iC);//??????
and (T1,iA,iB);//??????
and (T2,T0,iC);//??????
or (oC,T1,T2);//c?????
endmodule

module Adder(
input [7:0] iData_a, //8 ??????
input [7:0] iData_b, //8 ??????
input iC,
output [7:0] oData,  //8???
output oData_C
);
FA u0(iData_a[0],iData_b[0],iC,oData[0],c0);
FA u1(iData_a[1],iData_b[1],c0,oData[1],c1);
FA u2(iData_a[2],iData_b[2],c1,oData[2],c2);
FA u3(iData_a[3],iData_b[3],c2,oData[3],c3);
FA u4(iData_a[4],iData_b[4],c3,oData[4],c4);
FA u5(iData_a[5],iData_b[5],c4,oData[5],c5);
FA u6(iData_a[6],iData_b[6],c5,oData[6],c6);
FA u7(iData_a[7],iData_b[7],c6,oData[7],oData_C);

```

```
endmodule
```

四、测试模块建模

（要求列写各建模模块的 test bench 模块代码）

实验一：

```
`timescale 1ns/1ns
module datacompare4_tb;
reg [3:0] a;
reg [3:0] b;
reg [2:0] idata;
wire [2:0] odata;
initial
begin
idata=3'b001;
a=3'b110;
b=3'b111;
#50
a=3'b100;
b=3'b011;
#50
a=3'b101;
b=3'b111;
#50
a=3'b100;
b=3'b100;
end
DataCompare4 uut(a,b,idata,odata);
endmodule
```

实验二：

```
`timescale 1ns/1ns
module datacompare8_tb;
reg [7:0] a;
reg [7:0] b;
wire [2:0] odata;
initial
begin
a=8'b10000001;
b=8'b10000011;
#50
a=8'b10000011;
```

```

b=8'b10000011;
#50
a=8'b11000001;
b=8'b10000011;
#50
a=8'b11100001;
b=8'b11110011;
end
DataCompare8 uut(a,b,odata);
endmodule

```

实验三：

```

`timescale 1ns/1ns
module adder1_tb;
reg ia,ib,ic;
wire os,oc;
initial
begin
ia=1'b0;
ib=1'b1;
ic=1'b1;
#50
ia=1'b1;
ib=1'b1;
#50
ia=1'b0;
ib=1'b0;
#50
ia=1'b1;
ib=1'b1;
ic=1'b1;
end
FA uut(ia,ib,ic,os,oc);
endmodule

```

实验四：

```

`timescale 1ns/1ns
module adder8_tb;
reg [7:0] idata_a,idata_b;
wire [7:0] odata;
wire oc;
reg ic;
initial
begin

```

```

idata_a=8'b00110011;
idata_b=8'b11001100;
ic=1'b0;
#50
idata_a=8'b00001011;
idata_b=8'b10001100;
#50
idata_a=8'b10001011;
idata_b=8'b10001100;
#50
idata_a=8'b00001011;
idata_b=8'b10001100;
ic=1'b1;
end
Adder uut(idata_a,idata_b,ic,odata,oc);
endmodule

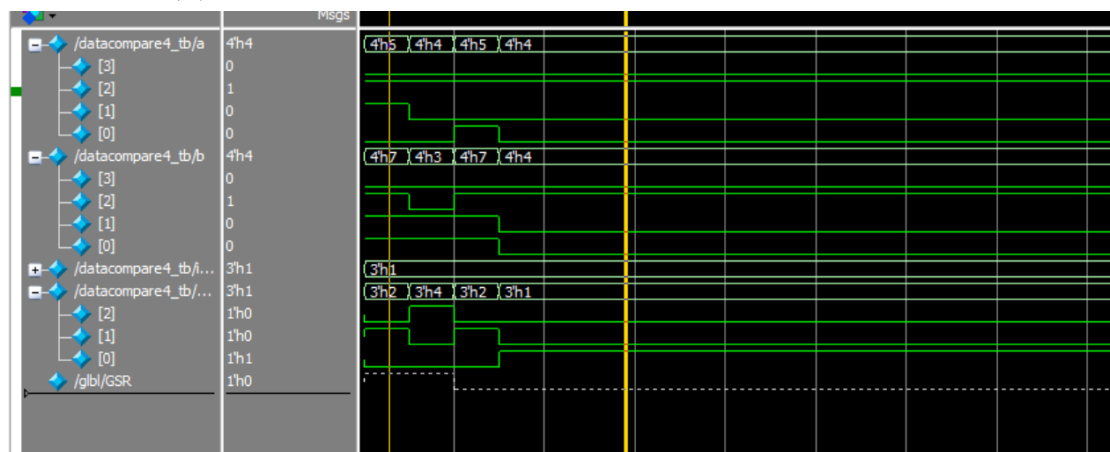
```

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））

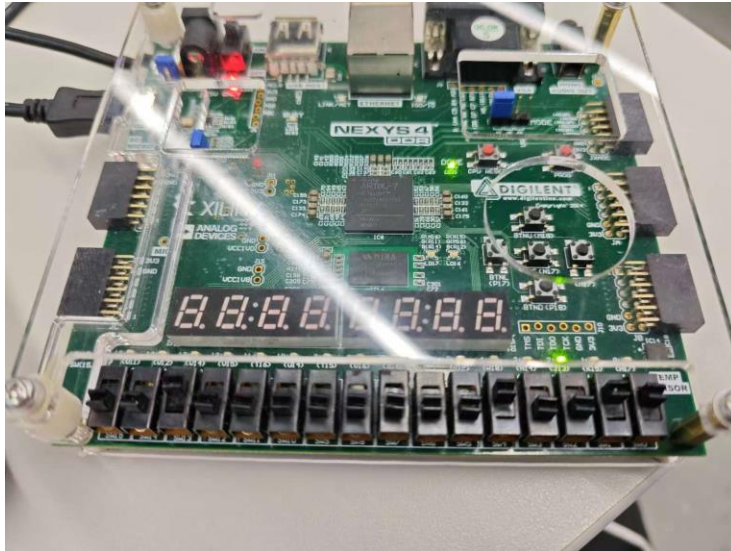
实验一：

modelsim 仿真波形图

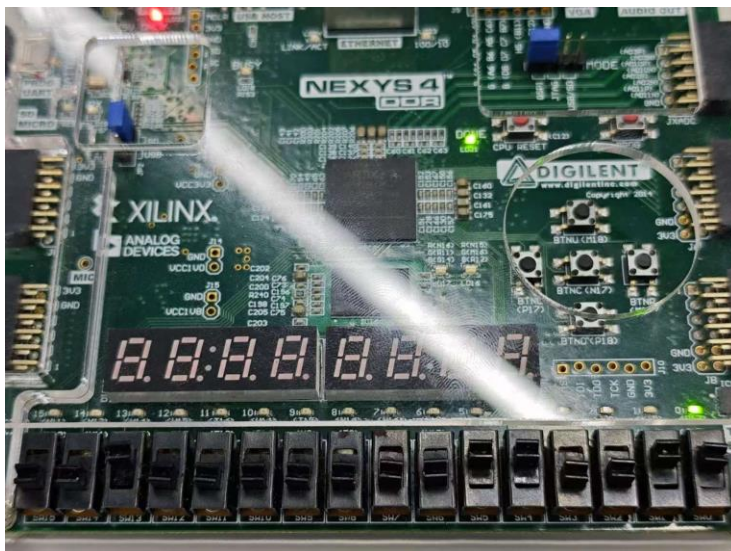


下板结果

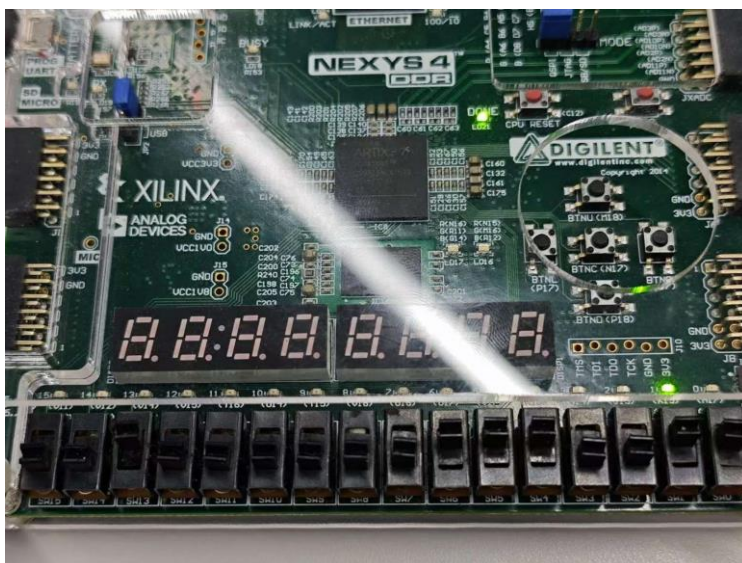
1、级联端输入是 001(等于)，A 是 0011，B 是 0001，输出位 100(A>B)。



2、级联端输入是 001(等于), A 是 0011, B 是 0011, 输出位 001($A=B$)。

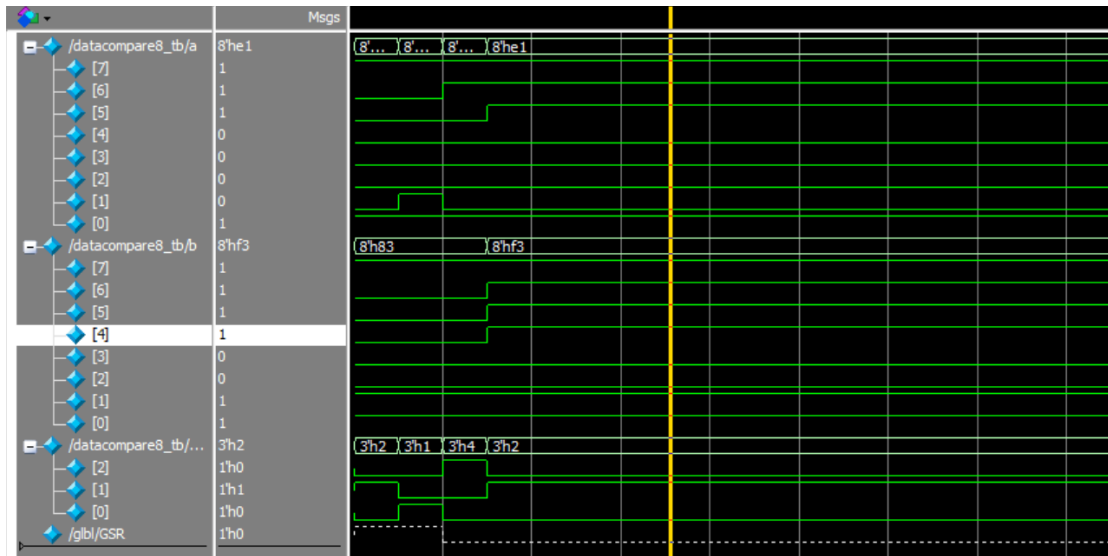


3、级联端输入是 001(等于), A 是 0011, B 是 0111, 输出位 010($A < B$)。



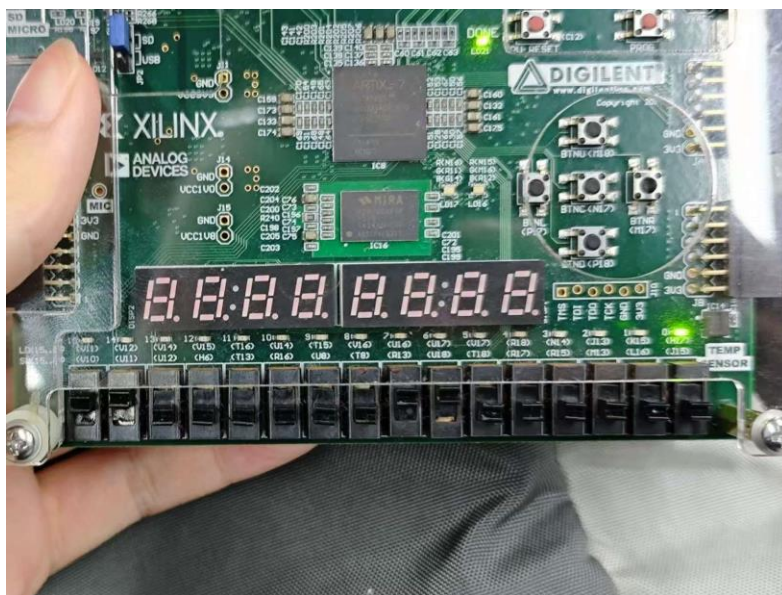
实验二：

modelsim 仿真波形图

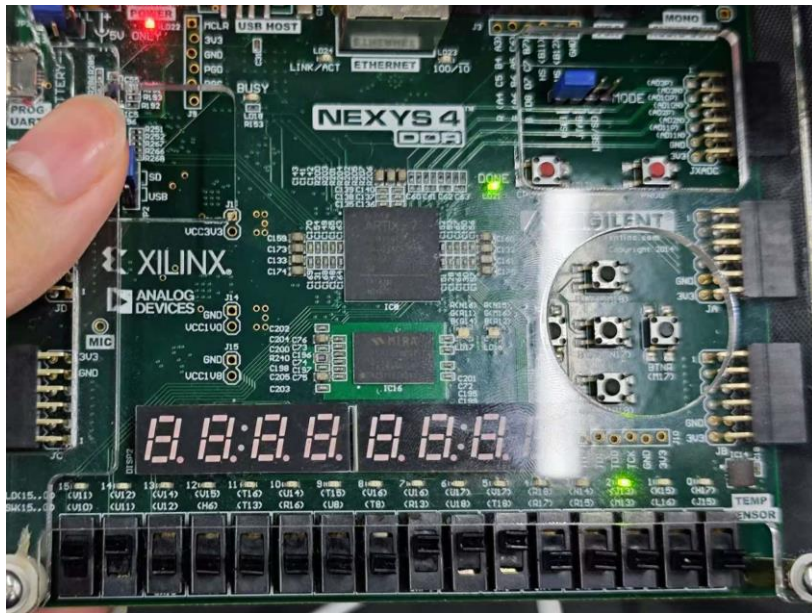


下板结果

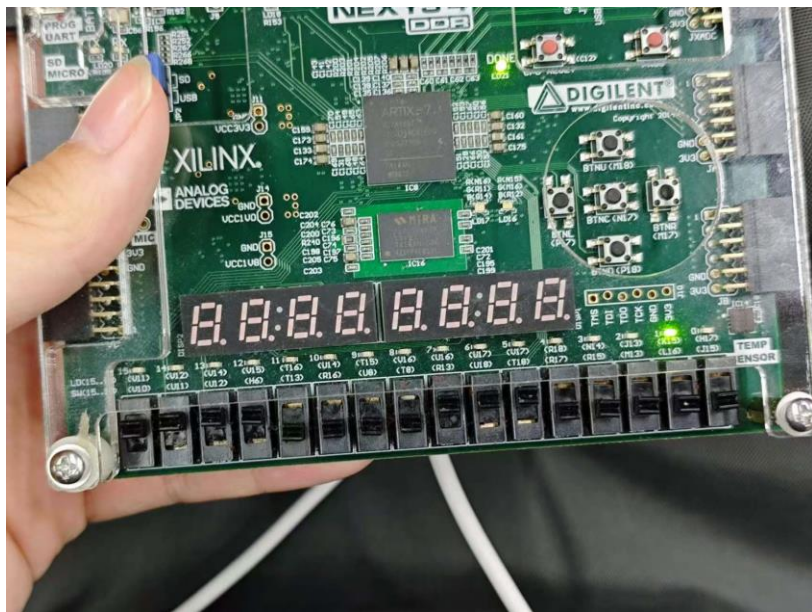
1、A=11000000,B=11000000,输出=001(A=B)



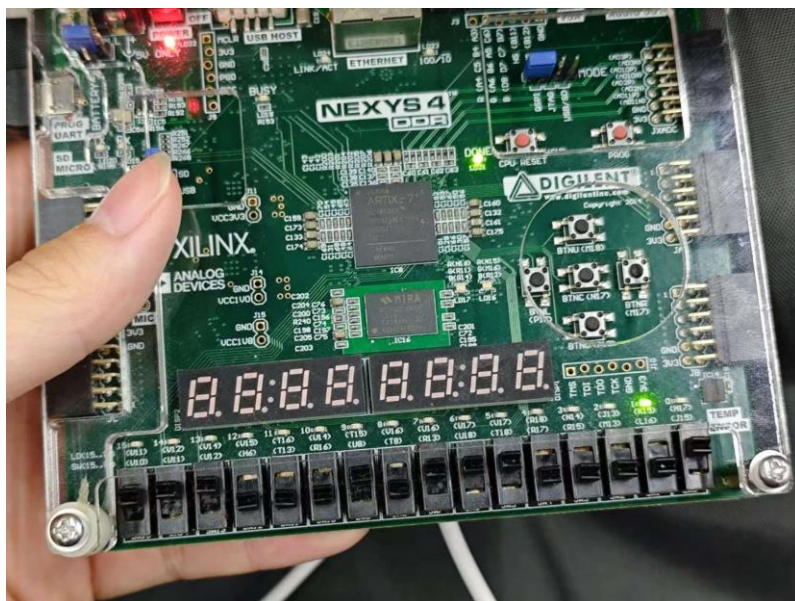
2、A=11100000,B=11000000,高四位不同,输出=100(A>B)



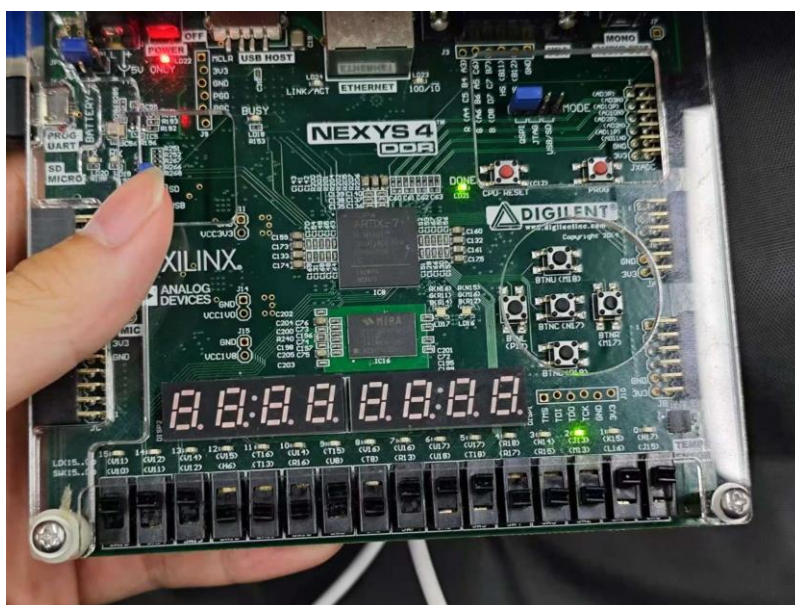
3、 $A=11100000, B=11110000$, 高四位不同, 输出=010($A < B$)



4、 $A=11100000, B=111100010$, 低四位不同, 输出=010($A < B$)

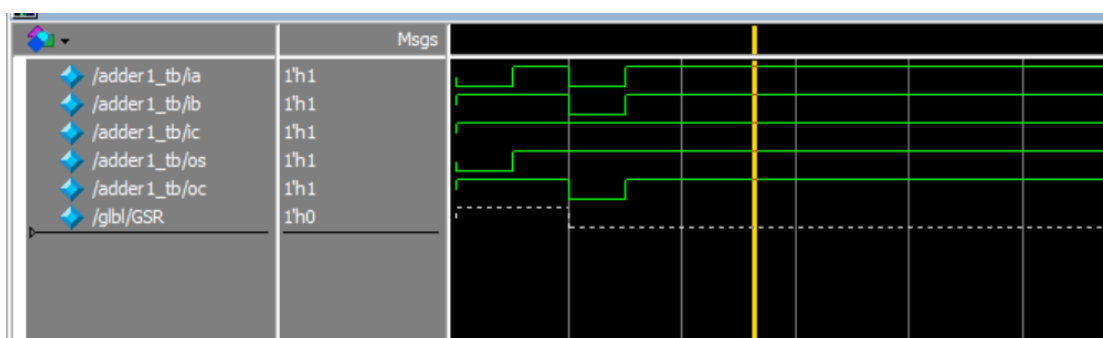


5、 $A=11100010, B=11000011$, 低四位不同, 输出=100($A>B$)



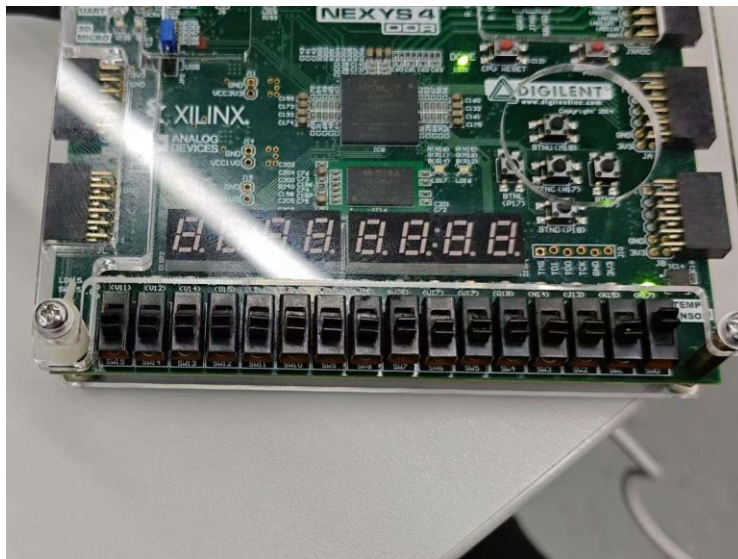
实验三:

modelsim 仿真波形图

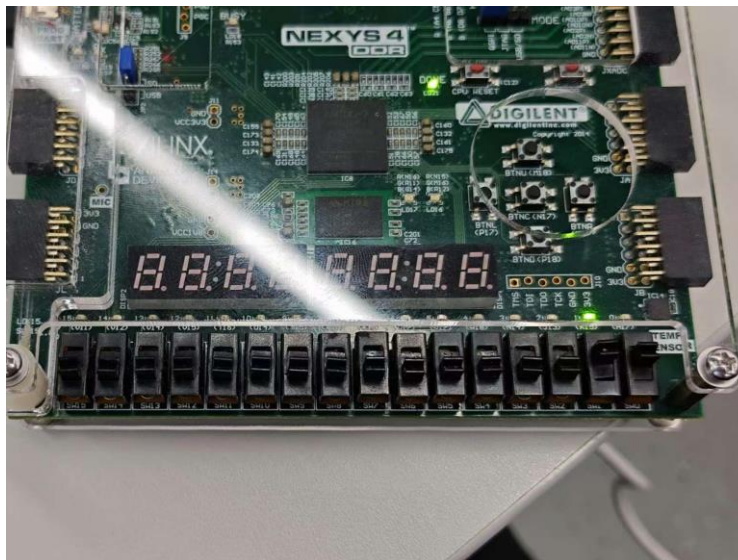


下板结果

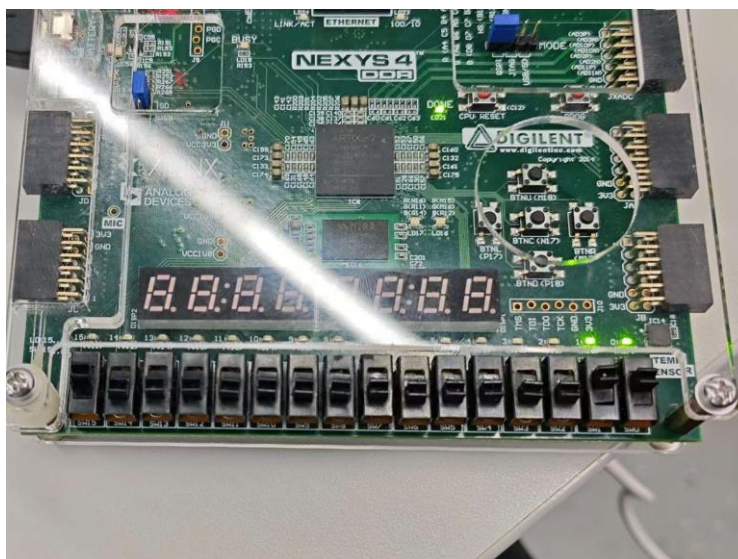
1、ia=1, ib=0, ic=0, os=1, oc=0



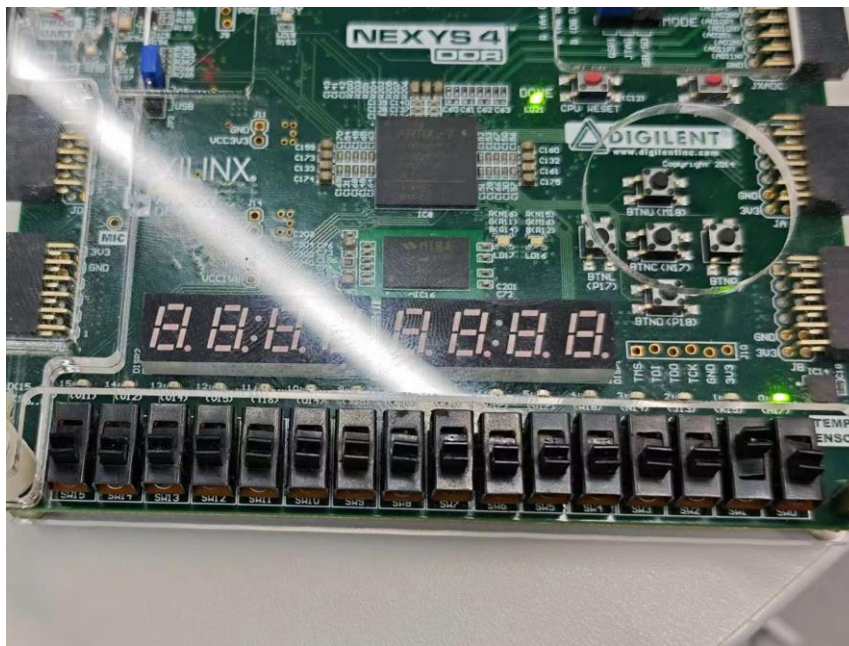
2、ia=1, ib=1, ic=0, os=0, oc=1



3、ia=1, ib=1, ic=1, os=1, oc=1

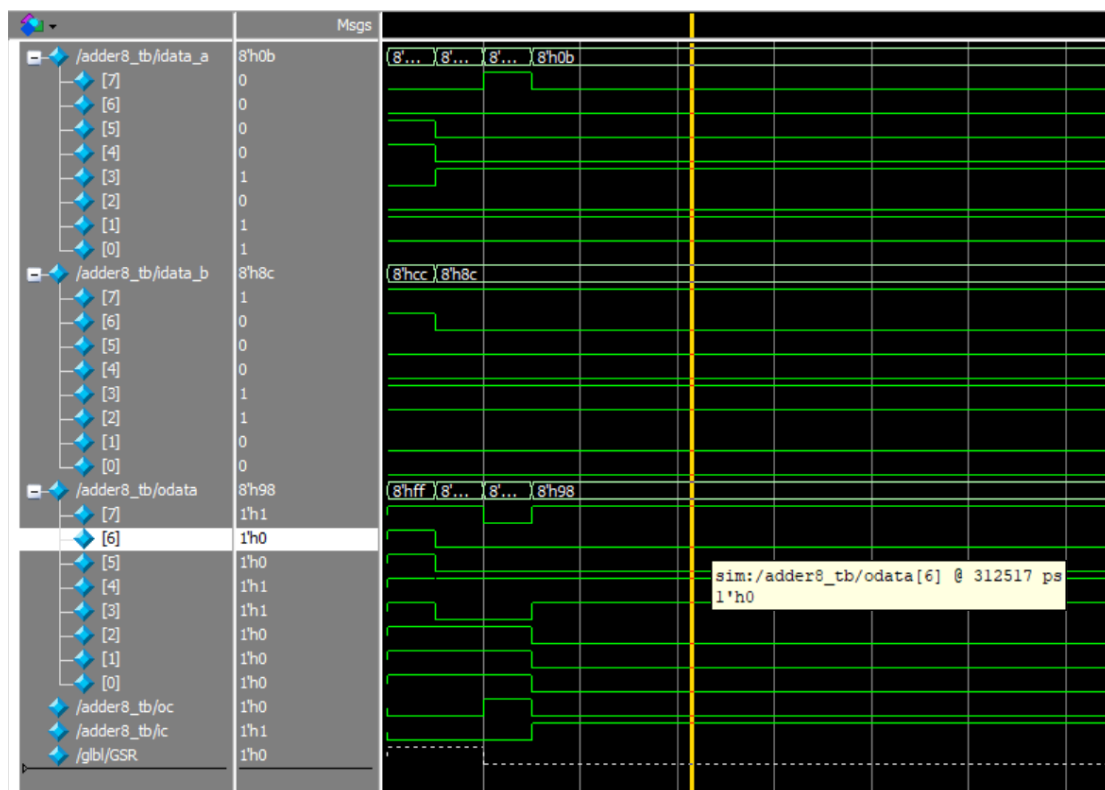


4、ia=0, ib=1, ic=0, os=1, oc=0



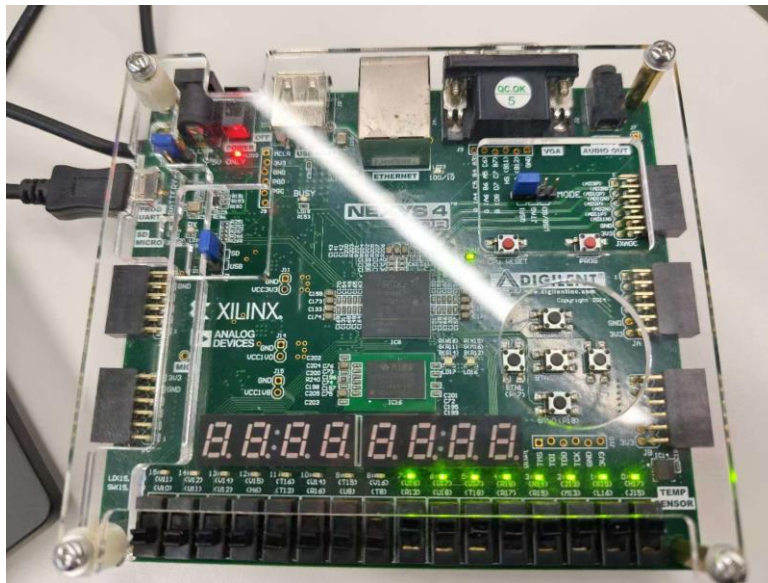
实验四：

modelsim 仿真波形图

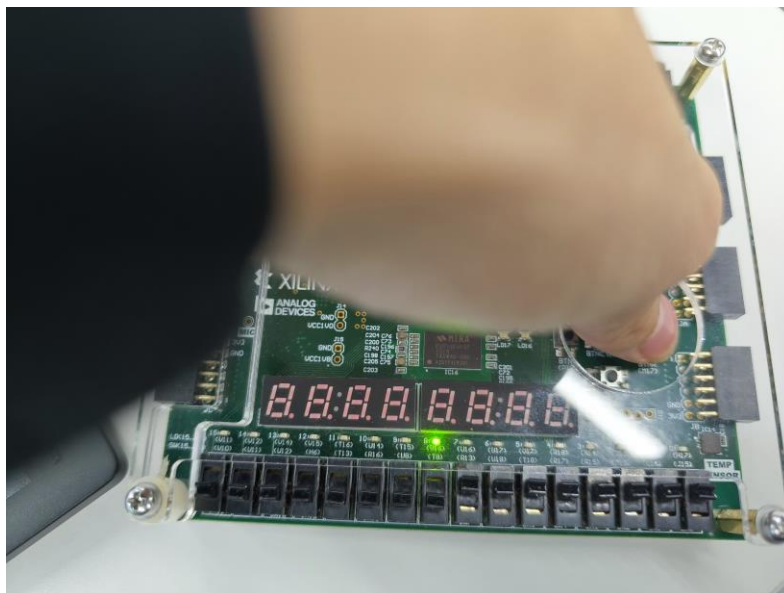


下板结果

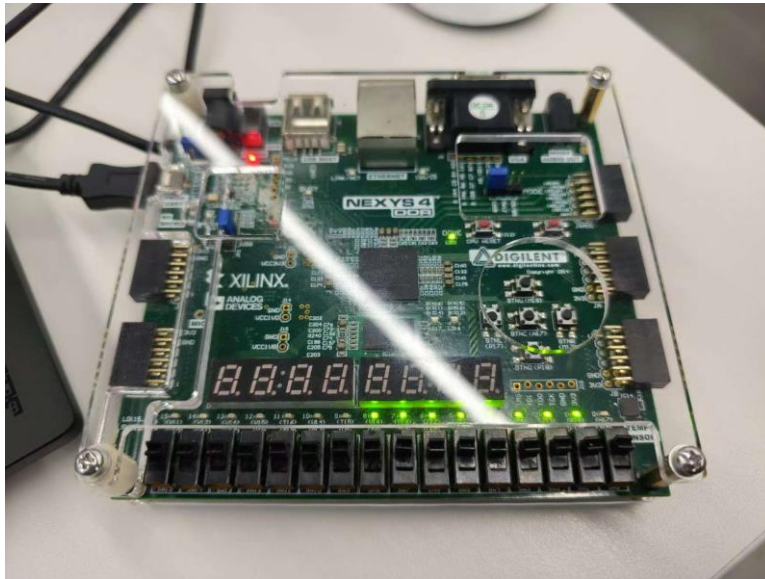
1、A=11111111, B=00000000, iC=0, oData=11111111, oC=0



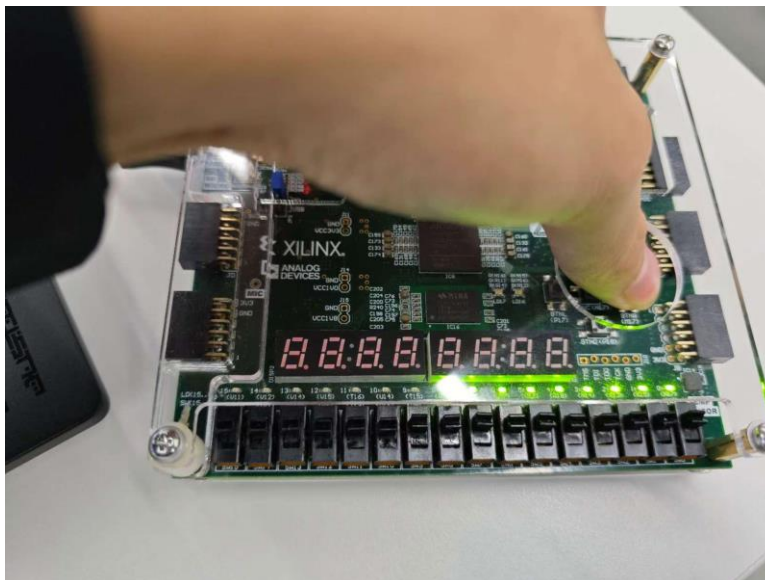
2、A=11111111, B=00000000, iC=1, oData=00000000, oC=1



3、A=11111111, B=11111111, iC=0, oData=11111110, oC=1



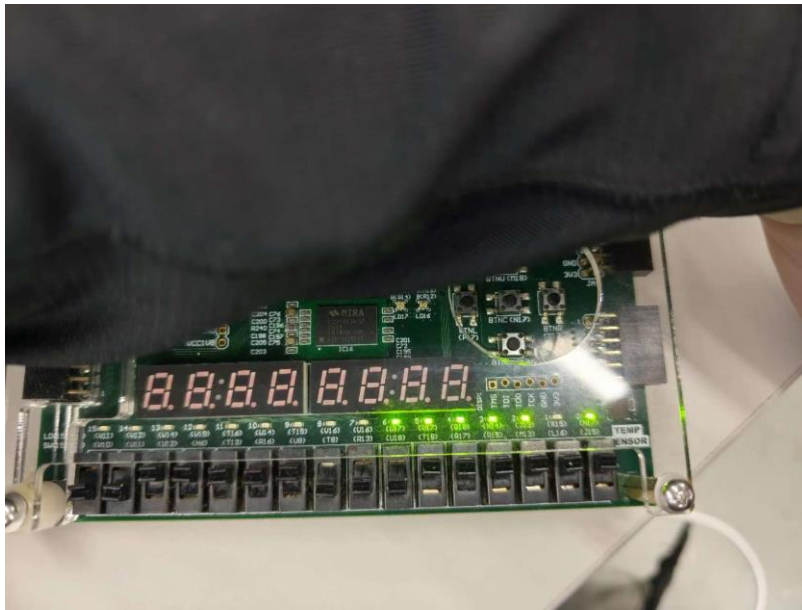
4、A=11111111, B=11111111, iC=1, oData=11111111, oC=1



5、A=10111101, B=00110110, iC=0, oData=11110011, oC=0



6、A=00111111, B=00111110, iC=0, oData=01111101, oC=0



7、A=00111111, B=00111110, iC=1, oData=01111110, oC=0

