同济大学计算机系

数字逻辑课程实验报告

学　　号 1652286

姓　　名 李辉

专　　业 计算机

授课老师 张冬冬

二、实验内容

1. 实现 4 位数据比较器和 8 位数据比较器（以 4 位比较器为基础）的设计； 2. 实现 1 位加法器和 8 位加法器的设计；

3.实验过程包括 Verilog HDL 语言描述、Modelsim 仿真和下板验证。硬件逻辑图

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)6.5_1 DataCompare4 四位比较器

```
`timescale 1ns / 1ns
module DataCompare4(
    input [3:0] iData_a,
    input [3:0] iData_b,
    input [2:0] iData,
    output reg[2:0] oData
    );
    always @(*)
    if(iData_a>iData_b)
    oData=3'b100;
    else if(iData_a<iData_b)
    oData=3'b001;
    else if(iData_a==iData_b)
    begin
    if(iData==3'b100)
    oData=3'b100;
    else if(iData==3'b010)
    oData=3'b010;
    else if(iData==3'b001)
    oData=3'b001;
    else
    oData=3'b000;
    end
endmodule
```

(2)6.5_2 DataCompare8 八位比较器

```
`timescale 1ns / 1ns
module DataCompare8(
    input [7:0] iData_a,
    input [7:0] iData_b,
    output [2:0] oData
    );
    wire [2:0]fff;
    DataCompare8_low
ddd(.iData_a(iData_a[3:0]),.iData_b(iData_b[
3:0]),.oData(fff));
    DataCompare8_high
ggg(.iData_a(iData_a[7:4]),.iData_b(iData_b[
7:4]),.iData(fff),.oData(oData));
    endmodule

    module DataCompare8_low(
        input [3:0] iData_a,
        input [3:0] iData_b,
        output reg[2:0] oData
        );
        always @(*)
            if(iData_a>iData_b)
            oData=3'b100;
            else if(iData_a<iData_b)
            oData=3'b001;
            else if(iData_a==iData_b)
            oData=3'b010;
        endmodule

    module DataCompare8_high(
                input [3:0] iData_a,
                input [3:0] iData_b,
                input [2:0] iData,
                output reg[2:0] oData
                );
                always @(*)
                    if(iData_a>iData_b)
```

```
                    oData=3'b100;
                    else if(iData_a<iData_b)
                    oData=3'b001;
                    else
if(iData_a==iData_b)
                    begin
                    if(iData==3'b100)
                    oData=3'b100;
                    else if(iData==3'b010)
                    oData=3'b010;
                    else if(iData==3'b001)
                    oData=3'b001;
                    else
                    oData=3'b000;
                    end
                endmodule
```

(3)6.5_3　FA　8-3 编码器　一位加法器

```
`timescale 1ns / 1ns
module FA(
    input iA,
    input iB,
    input iC,
    output reg oS,
    output reg oC
    );
    always@(*)
    begin
    oC=(iA&iB)|(iA&iC)|(iB&iC);
    if(iC)
    oS=!(iA^iB);
    else
    oS=(iA^iB);
    end
endmodule
```

(4)6.5_4　Adder　八位加法器

```
 `timescale 1ns / 1ns
module Adder(
    input [7:0] iData_a,
    input [7:0] iData_b,
    input iC,
    output reg [8:0] oData,
    output reg oData_C
    );
    always @(*)
    if(iC==0)
    begin
    oData=iData_a+iData_b;

    if(oData[8]==1)
    oData_C=1;
    else
    oData_C=0;
    end

    else
if(iC==1&&iData_a[7]==0&&iData_b[7]==0)
    begin
    oData=iData_a+iData_b;
    if(oData[7]==1)
    oData_C=1;
    else
    oData_C=0;
    end

    else
if(iC==1&&iData_a[7]==1&&iData_b[7]==0)
    begin
    if(iData_a[6:0]>iData_b[6:0])
    begin
    oData[7:0]=iData_a[6:0]-iData_b[6:0];
    oData[8]=1;
    oData_C=0;
    end
    else if(iData_a[6:0]<iData_b[6:0])
    begin
    oData[7:0]=iData_b[6:0]-iData_a[6:0];
    oData[8]=0;
    oData_C=0;
    end
    else if(iData_a[6:0]==iData_b[6:0])
    begin
    oData=0;
    oData_C=0;
    end
    end
    else
if(iC==1&&iData_a[7]==0&&iData_b[7]==1)
```

```
        begin
        if(iData_a[6:0]<iData_b[6:0])
          begin
          oData[7:0]=iData_b[6:0]-iData_a[6:0];
        oData[8]=1;
        oData_C=0;
        end
        else if(iData_a[6:0]>iData_b[6:0])
        begin
        oData[7:0]=iData_a[6:0]-iData_b[6:0];
        oData[8]=0;
        oData_C=0;
        end
        else if(iData_a[6:0]==iData_b[6:0])
        begin
        oData=0;
        oData_C=0;
        end
        end
        else
if(iC==1&&iData_a[7]==1&&iData_b[7]==1)

        begin
        oData[7:0]=iData_a[6:0]+iData_b[6:0];
        oData[8]=1;
        if(oData[7]==1)
        oData_C=1;
        else oData_C=0;
        end
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) 6.5_1 DataCompare4 四位比较器

```
`timescale 1ns / 1ns
module DataCompare4_tb();
reg [3:0] iData_a;
reg [3:0] iData_b;
reg [2:0] iData;
wire [2:0] oData;
DataCompare4
uut(.iData_a(iData_a),.iData_b(iData_b),.iDat
a(iData),.oData(oData));
initial
begin
#50
iData_a=4'b1010;
iData_b=4'b0110;
iData=3'b010;
#50
iData_a=4'b0010;
iData_b=4'b1110;
iData=3'b010;
#50
iData_a=4'b1010;
iData_b=4'b1110;
iData=3'b010;
#50
iData_a=4'b1110;
iData_b=4'b1010;
iData=3'b010;
#50
iData_a=4'b1100;
iData_b=4'b1110;
iData=3'b010;
#50
iData_a=4'b1110;
iData_b=4'b1101;
iData=3'b010;
#50
iData_a=4'b1100;
iData_b=4'b1101;
iData=3'b010;
#50
iData_a=4'b1101;
iData_b=4'b1100;
iData=3'b010;
#50
iData_a=4'b1101;
iData_b=4'b1101;
iData=3'b100;
#50
iData_a=4'b1101;
iData_b=4'b1101;
iData=3'b010;
#50
```

```
iData_a=4'b1101;
iData_b=4'b1101;
iData=3'b001;
#50
iData_a=4'b1101;
iData_b=4'b1101;
iData=3'b101;
end
endmodule
```

(2)6.5_2　DataCompare8　八位比较器

```
`timescale 1ns / 1ns
module DataCompare8_tb();
reg [7:0] iData_a;
reg [7:0] iData_b;
wire [2:0] oData;
DataCompare8
uut(.iData_a(iData_a),.iData_b(iData_b),.oDat
a(oData));
initial
begin
#50
iData_a=8'b10101011;
iData_b=8'b01111101;
#50
iData_a=8'b00101011;
iData_b=8'b11111101;
#50
iData_a=8'b10101011;
iData_b=8'b11111101;
#50
iData_a=8'b11101011;
iData_b=8'b10111101;
#50
iData_a=8'b10101011;
iData_b=8'b10011101;
#50
iData_a=8'b10001011;
iData_b=8'b10111101;
#50
iData_a=8'b10101011;
iData_b=8'b10111101;
#50
iData_a=8'b10111011;
iData_b=8'b10101101;
#50
iData_a=8'b10110011;
iData_b=8'b10111101;
#50
iData_a=8'b10111011;
iData_b=8'b10110101;
#50
iData_a=8'b10110011;
iData_b=8'b10110101;
#50
iData_a=8'b10110111;
iData_b=8'b10110001;
#50
iData_a=8'b10110111;
iData_b=8'b10110101;
#50
iData_a=8'b10110101;
iData_b=8'b10110111;
#50
iData_a=8'b10110100;
iData_b=8'b10110101;
#50
iData_a=8'b10110101;
iData_b=8'b10110100;
end
endmodule
```

(3)6.5_3　FA　一位加法器

```
`timescale 1ns / 1ns
module FA_tb();
reg a;
reg b;
reg c;
wire s;
wire out;
FA uut(.iA(a),.iB(b),.iC(c),.oS(s),.oC(out));
initial
begin
a=0;b=0;c=0;
#10
a=0;b=0;c=1;
#10
a=0;b=1;c=0;
#10
a=0;b=1;c=1;
```

```
#10
a=1;b=0;c=0;
#10
a=1;b=0;c=1;
#10
a=1;b=1;c=0;
#10
a=1;b=1;c=1;
end
endmodule
```

(4)6.5_4　Adder 八位加法器

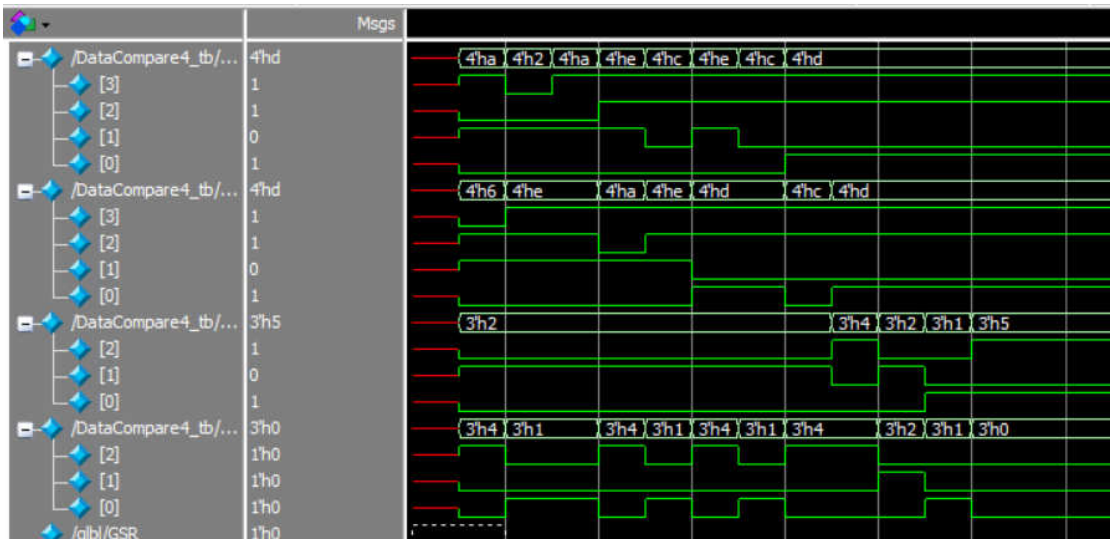
```
`timescale 1ns / 1ns
module Adder_tb();
reg iC;
reg [7:0] iData_a;
reg [7:0] iData_b;
wire [8:0] oData;
wire oData_C;

Adder uut(.iC(iC),
.iData_a(iData_a),
.iData_b(iData_b),
.oData(oData),
.oData_C(oData_C));
initial
begin
#50
iC=0;
  iData_a=8'b00000001;
  iData_b=8'b00000001;
  #50
  iC=0;
  iData_a=8'b11111111;
  iData_b=8'b11111111;
  #50
  iC=0;
  iData_a=8'b10001001;
  iData_b=8'b10001101;
  #50
  iC=0;
  iData_a=8'b00000001;
  iData_b=8'b00000001;
  #50
  iC=1;
  iData_a=8'b00000001;
  iData_b=8'b00000001;
  #50
  iC=1;
  iData_a=8'b10000001;
  iData_b=8'b00000001;
  #50
  iC=1;
  iData_a=8'b10000001;
  iData_b=8'b00000011;
  #50 iC=1;
  iData_a=8'b10000011;
  iData_b=8'b00000001;
  #50 iC=1;
  iData_a=8'b00000001;
  iData_b=8'b10000011;
  #50
  iC=1;
  iData_a=8'b00000011;
  iData_b=8'b10000001;
  #50
  iC=1;
  iData_a=8'b10000011;
  iData_b=8'b10000001;
  #50 iC=1;
  iData_a=8'b11111111;
  iData_b=8'b11111111;
end
Endmodule
```

五、实验结果

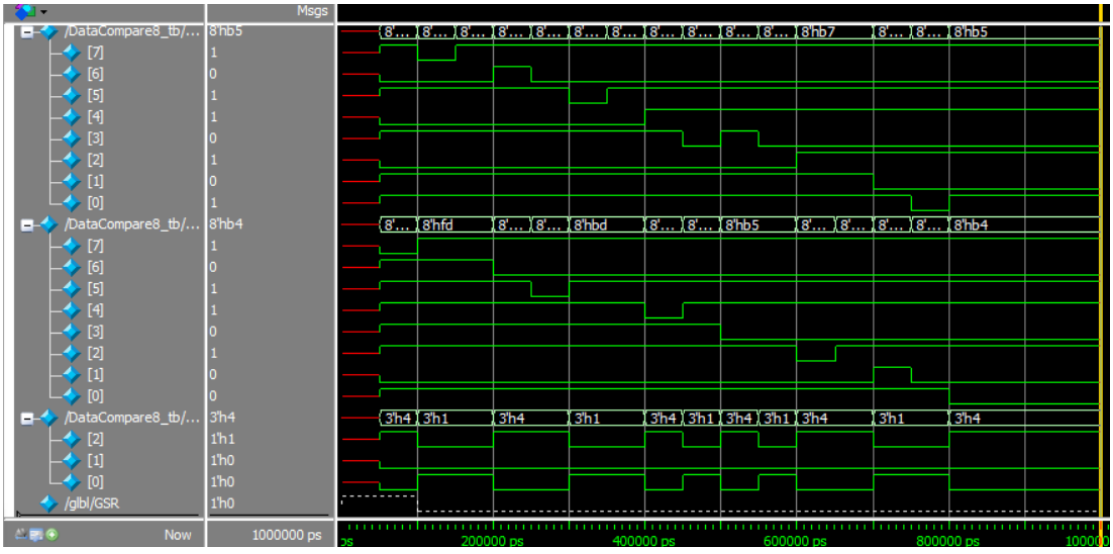
（该部分可截图说明，要求 logisim 逻辑验证图、modelsim 仿真波形图、以及下板后的实验结果贴图（实验步骤中没有下板要求的实验，不需要下板贴图））

Modelsim 仿真波形图

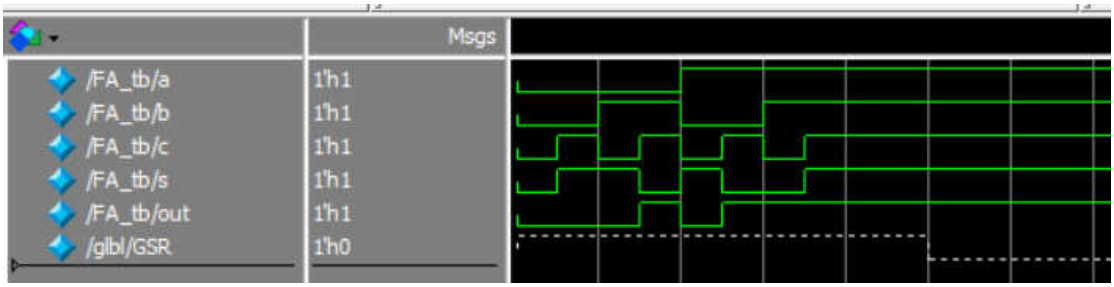
(一)6.5_1 Datacompare4 4 位比较器



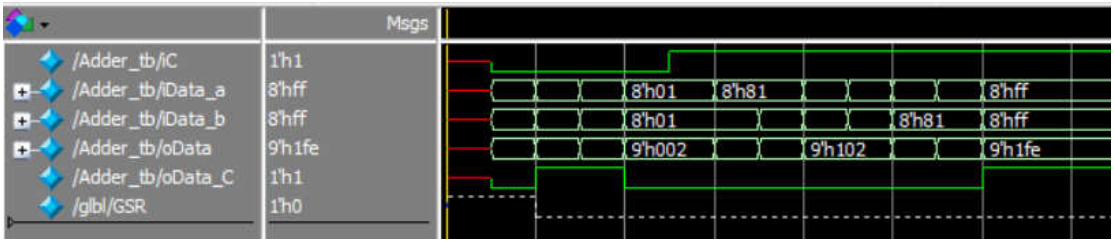
(二)6.5_2 DataCompare8 八位比较器



(三)6.5_3 FA 一位加法器

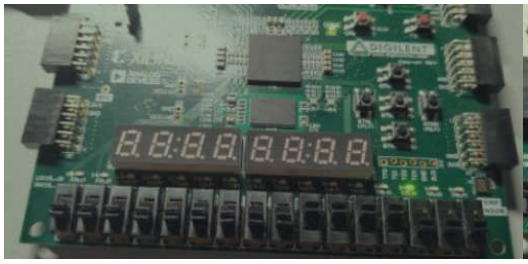
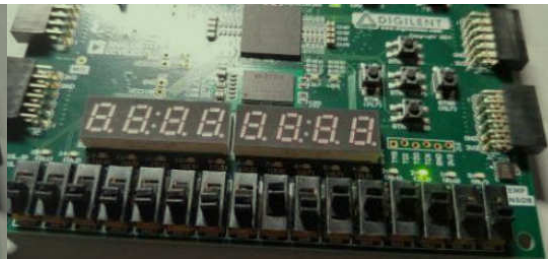
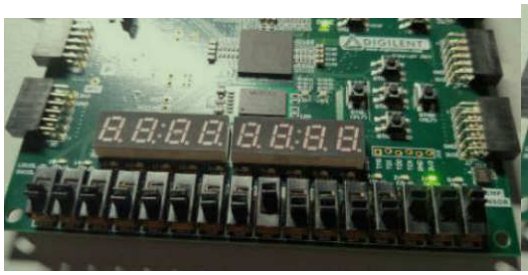
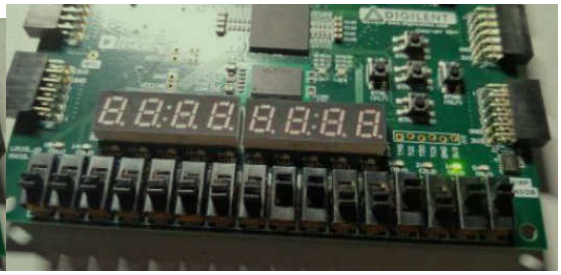
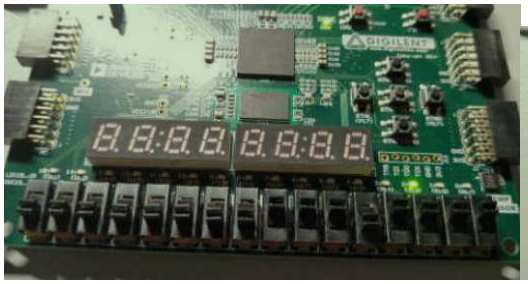
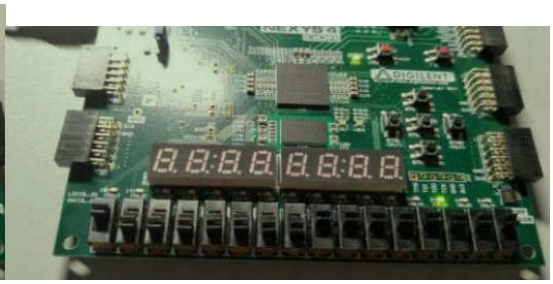


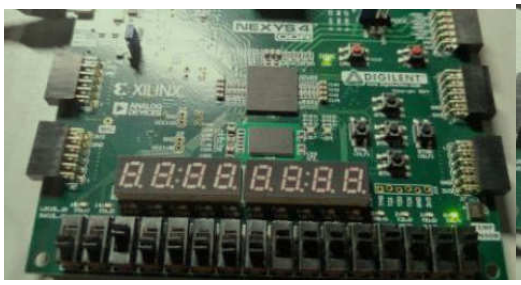
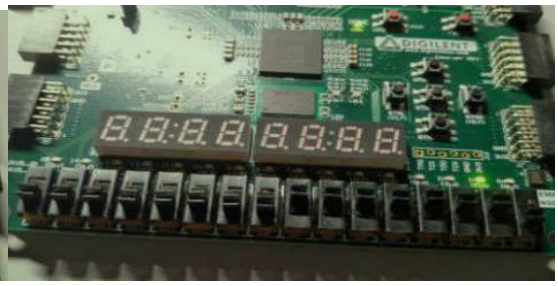
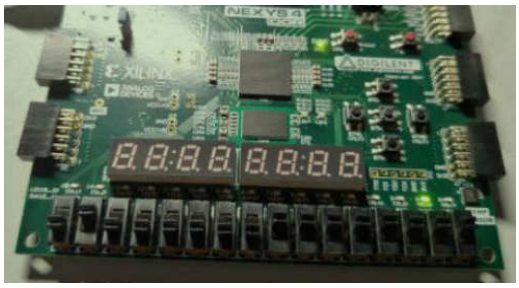
(四)6.5_4　Adder 八位加法器



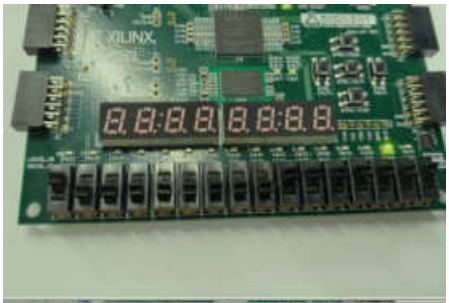
下板后贴图

(一)6.5_1　DataCompare4　四位比较器

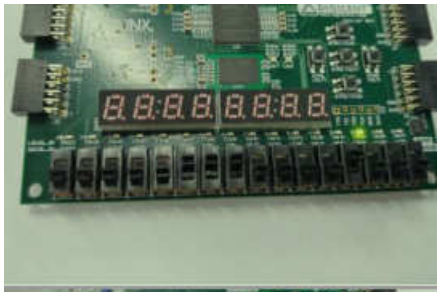
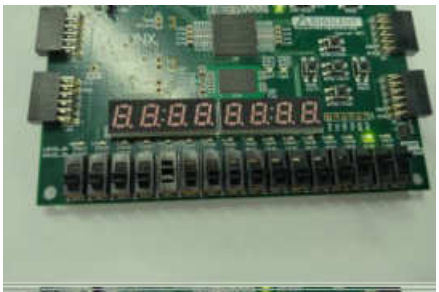




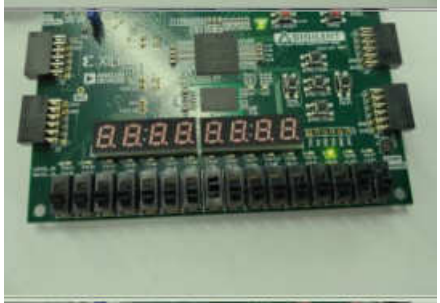
(二)6.5_2　DataCompare8　八位比较器

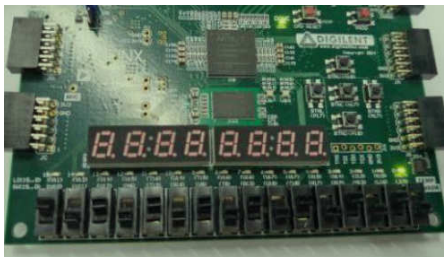
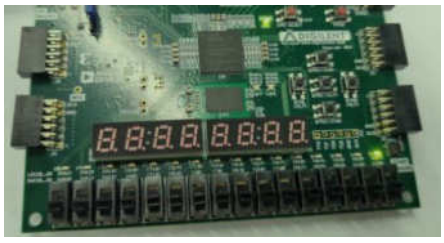




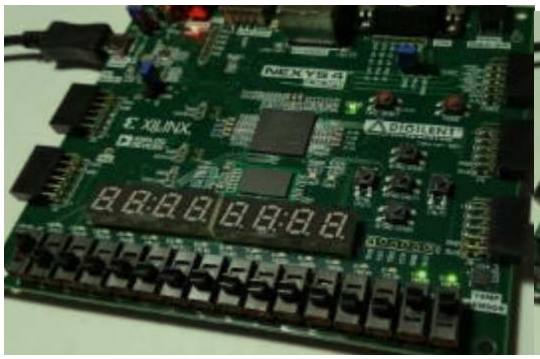
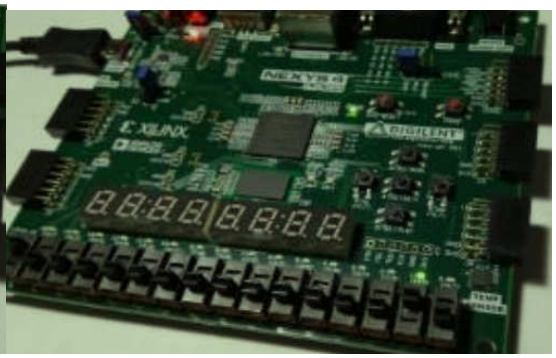
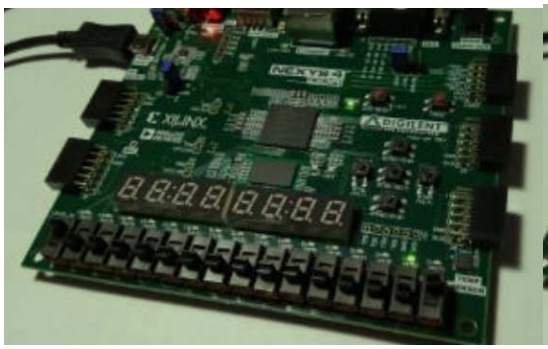
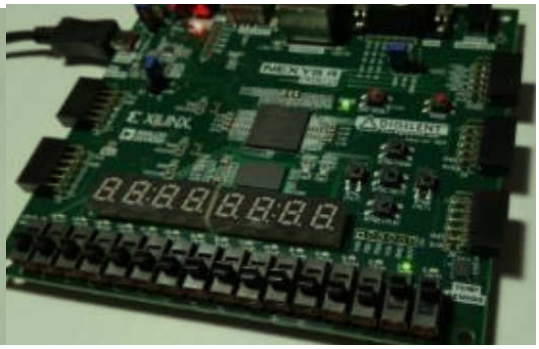
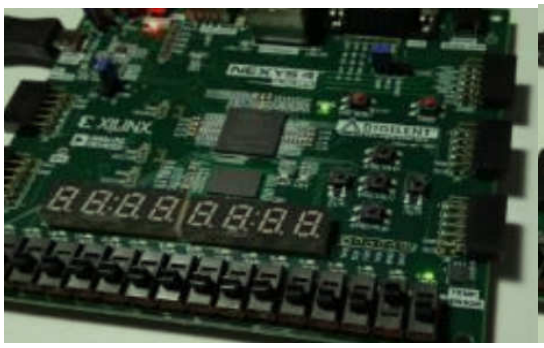
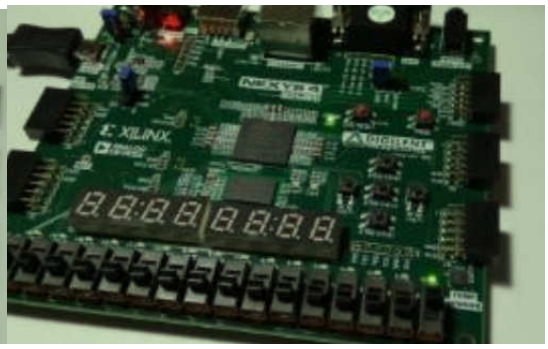
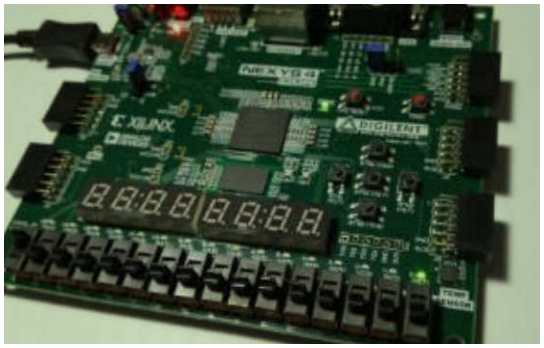





(三)6.5_3　FA　一位加法器

(四)6.5_4　Adder 八位加法器

结果比较多，不一一列出