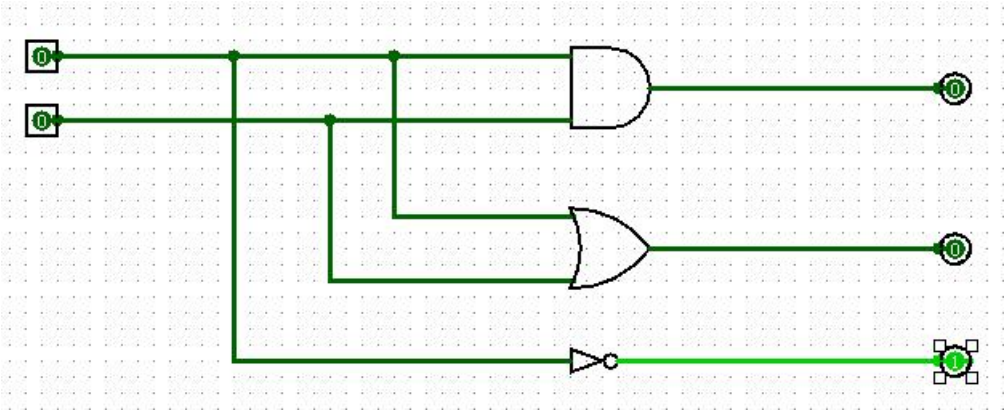
一、实验内容

本次实验分为三个部分：第一部分先用 logisim 画出基本与或非门实验电路原理图，验证逻辑，再使用 Verilog HDL 语言，采用三种不同描述方式设计与或非门电路，调用 modelsim 仿真测试各模块，最后综合下板验证；第二部分使用 Verilog HDL 语言，设计三态门电路，调用 modelsim 仿真测试，并进行下板验证；第三部分则是使用 Verilog HDL 语言，实现将输入的数据扩展为 32 位数据的功能，并调用 modelsim 进行仿真测试。

二、硬件逻辑图



三、模块建模

6.1_1 结构型描述

```
module logic_gates_1(iA,iB,oAnd,oOr,oNot);
    input iA,iB;
    output oAnd,oOr,oNot;
    and and_inst(oAnd,iA,iB);
    or or_inst(oOr,iA,iB);
    not not_inst(oNot,iA);
endmodule
```

功能描述：该模块利用结构型描述构建一个 Verilog 模块，用于实现基本的与或非门功能。

输出 oAnd：是输入 iA 和 iB 的与操作的结果。只有当 iA 和 iB 都为 1 时，oAnd 才为 1；否则，oAnd 为 0。

输出 oOr：是输入 iA 和 iB 的或操作的结果。只要 iA 或 iB（或两者都）为 1，oOr 就为 1；只有当 iA 和 iB 都为 0 时，oOr 才为 0。

输出 oNot：是输入 iA 的非操作的结果。即，如果 iA 为 1，则 oNot 为 0；如果 iA 为 0，则 oNot 为 1。

6.1_2 数据流型描述

```
module logic_gates_2(iA,iB,oAnd,oOr,oNot);
```

```
    input iA,iB;
    output oAnd,oOr,oNot;
    assign oAnd= iA & iB;
    assign oOr= iA | iB;
    assign oNot= ~iA;
endmodule
```

功能描述：该模块利用数据流型描述构建一个 Verilog 模块，用于实现基本的与或非门功能。

输出 oAnd：是输入 iA 和 iB 的与操作的结果。只有当 iA 和 iB 都为 1 时，oAnd 才为 1；否则，oAnd 为 0。

输出 oOr：是输入 iA 和 iB 的或操作的结果。只要 iA 或 iB（或两者都）为 1，oOr 就为 1；只有当 iA 和 iB 都为 0 时，oOr 才为 0。

输出 oNot：是输入 iA 的非操作的结果。即，如果 iA 为 1，则 oNot 为 0；如果 iA 为 0，则 oNot 为 1。

6.1_3 行为描述

```
module logic_gates_3(iA,iB,oAnd,oOr,oNot);
    input iA,iB;
    output   oAnd,oOr,oNot;
    reg oAnd,oOr,oNot;
    always @ (*)
    begin
        oAnd = iA & iB;
        oOr = iA | iB;
        oNot = ~ iA;
    end
endmodule
```

功能描述：该模块利用行为描述构建一个 Verilog 模块，用于实现基本的与或非门功能。

输出 oAnd：是输入 iA 和 iB 的与操作的结果。只有当 iA 和 iB 都为 1 时，oAnd 才为 1；否则，oAnd 为 0。

输出 oOr：是输入 iA 和 iB 的或操作的结果。只要 iA 或 iB（或两者都）为 1，oOr 就为 1；只有当 iA 和 iB 都为 0 时，oOr 才为 0。

输出 oNot：是输入 iA 的非操作的结果。即，如果 iA 为 1，则 oNot 为 0；如果 iA 为 0，则 oNot 为 1。

6.1_4 三态门

```
module three_state_gates(iA,iEna,oTri);
    input iA;
    input iEna;
    output oTri;
    assign oTri = (iEna == 1)? iA:'bz;
endmodule
```

功能描述：该模块实现了三态门的基本功能。

当 iEna 等于 1 时，oTri 被赋值为 iA 的值。

当 iEna 不等于 1（即 iEna 为 0）时，oTri 被置于高阻态。在这种状态下，oTri 不会对连接的电路产生任何驱动作用。

6.1_5 数据扩展实验

```
module extend # (parameter WIDTH = 16)(
    input [WIDTH-1:0] a,
    input sext,
    output [31:0]b
    );
    assign b = sext?{{(32-WIDTH){a[WIDTH-1]}},a} : {27'b0,a};
endmodule
```

功能描述：该模块实现将输入数据扩展到 36 位的功能。如果 sext 为 1，则执行符号扩展；如果 sext 为 0，会将其视为无符号数进行扩展。

四、测试模块建模

6.1_1 结构型描述

```
module logic_gates_tb;
    reg iA;
    reg iB;
    wire oAnd;
    wire oOr;
    wire oNot;

    initial
    begin
        iA=0;
        #40 iA=1;
        #40 iA=0;
        #40 iA=1;
        #40 iA=0;
     end
     initial
     begin
        iB=0;
        #40 iB=0;
        #40 iB=1;
        #40 iB=1;
        #40 iB=0;
     end
     logic_gates_1
     logic_gates_inst(.iA(iA),.iB(iB),.oAnd(oAnd),.oOr(oOr),.oNot(oNot));
```

```
endmodule
```

描述：通过模拟输入信号 iA 和 iB 的变化，并观察输出信号 oAnd、oOr 和 oNot 的响应来验证逻辑门模块的功能。

6.1_2 数据流型描述

```
module logic_gates_tb;
    reg iA;
    reg iB;
    wire oAnd;
    wire oOr;
    wire oNot;

    initial
    begin
        iA=0;
        #40 iA=1;
        #40 iA=0;
        #40 iA=1;
        #40 iA=0;
     end
     initial
     begin
        iB=0;
        #40 iB=0;
        #40 iB=1;
        #40 iB=1;
        #40 iB=0;
     end
     logic_gates_2
     logic_gates_inst(.iA(iA),.iB(iB),.oAnd(oAnd),.oOr(oOr),.oNot(oNot));
endmodule
```

描述：通过模拟输入信号 iA 和 iB 的变化，并观察输出信号 oAnd、oOr 和 oNot 的响应来验证逻辑门模块的功能。

6.1_3 行为描述

```
module logic_gates_tb;
    reg iA;
    reg iB;
    wire oAnd;
    wire oOr;
    wire oNot;

    initial
    begin
```

```
        iA=0;
        #40 iA=1;
        #40 iA=0;
        #40 iA=1;
        #40 iA=0;
    end
    initial
    begin
        iB=0;
        #40 iB=0;
        #40 iB=1;
        #40 iB=1;
        #40 iB=0;
    end
    logic_gates_3
    logic_gates_inst(.iA(iA),.iB(iB),.oAnd(oAnd),.oOr(oOr),.oNot(oNot));
endmodule
```

描述：通过模拟输入信号 iA 和 iB 的变化，并观察输出信号 oAnd、oOr 和 oNot 的响应来验证逻辑门模块的功能。

6.1_4 三态门

```
module three_state_gates_tb;
    reg iA;
    reg iEna;
    wire oTriState;
    three_state_gates uut(.iA(iA),.iEna(iEna),.oTri(oTriState));
    initial
    begin
        iA=0;
        #40 iA=1;
        #40 iA=0;
        #40 iA=1;
    end
    initial
    begin
        iEna = 1;
        #20 iEna=0;
        #40 iEna=1;
        #20 iEna=0;
    end
endmodule
```

描述：该模块为三态门对应设计程序的测试程序，给予激励信号，观察输入输出，对设计程序的正确性进行判断。

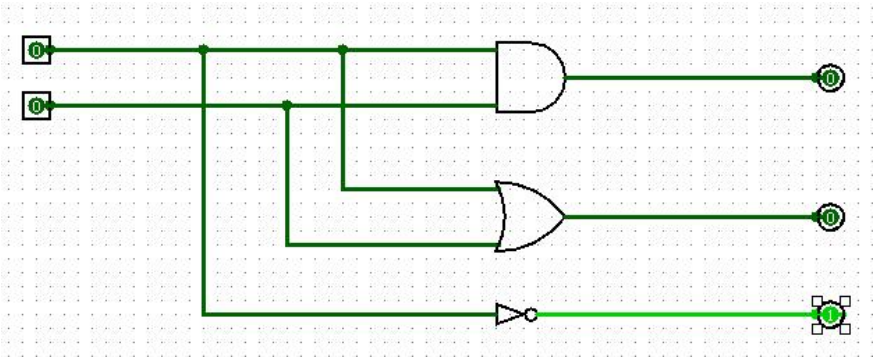
6.1_5 数据扩展实验

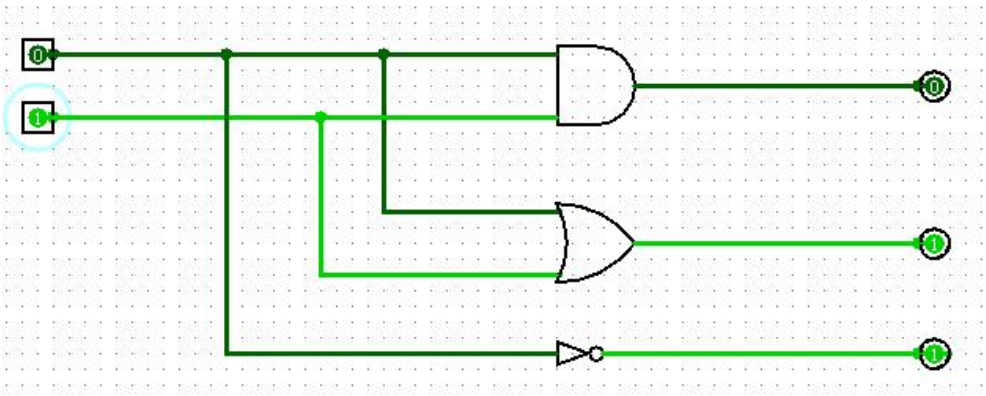
```
module extend_tb;
    reg [15:0] a;
    reg sext;
    wire [31:0] b;
    extend uut (.a(a),.sext(sext),.b(b));
    initial
    begin
    a = 0;
    sext = 0;
    #100;
    sext = 1;
    a = 16'h0000;
    #100;
    sext = 0;
    a = 16'h8000;
    #100;
    sext = 1;
    a = 16'h8000;
    #100;
    sext = 0;
    a = 16'hffff;
    #100;
    sext = 1;
    a = 16'hffff;
    #100;
    end
endmodule
```

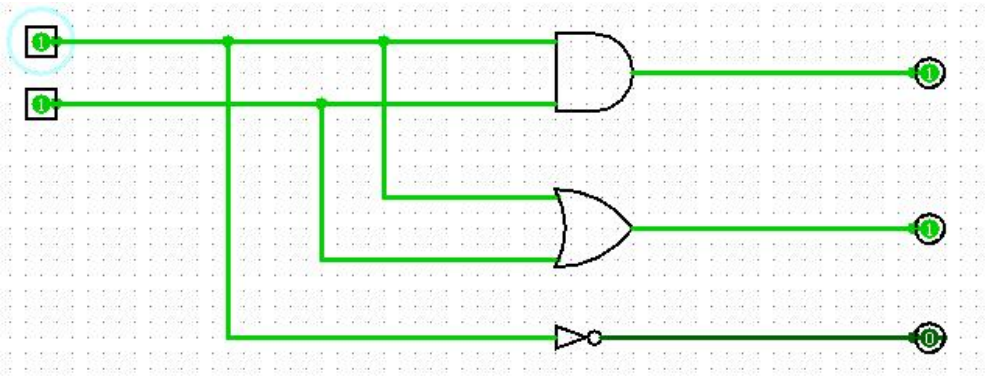
描述：该模块用于验证 extend 模块的功能。该模块根据 sext 信号的值来执行符号扩展或零扩展。程序中通过改变 a 和 sext 的值，并观察 b 的输出，来验证 extend 模块是否正确实现了这些功能。

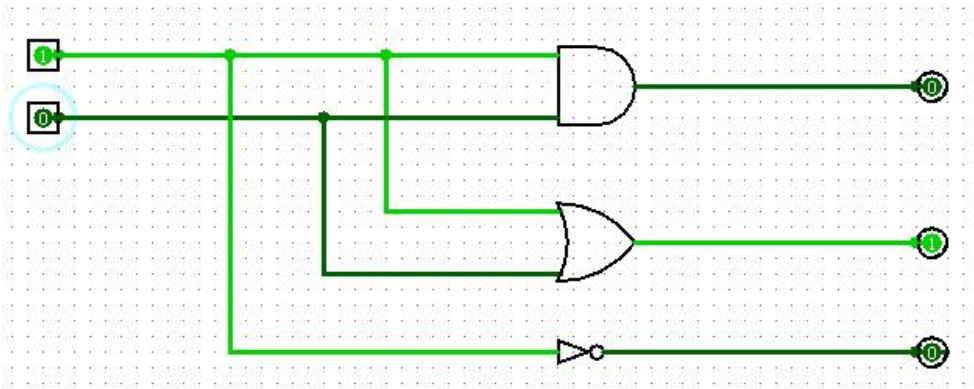
五、实验结果

6.1_1/6.1_2/6.1_3　logisim 逻辑验证图







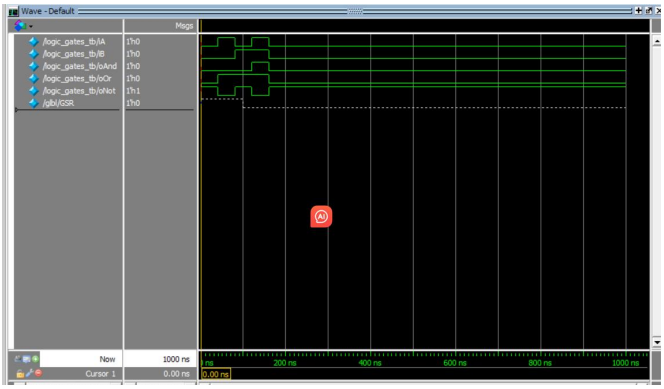


Truth Table

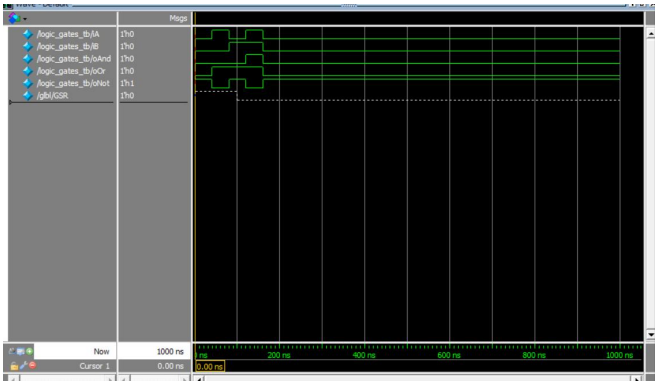
iA	iB	oAnd	oOr	oNot
0	0	0	0	1
0	1	0	1	1
1	1	1	1	0
1	0	0	1	0

通过真值表验证，可知 logisim 逻辑图设计正确。

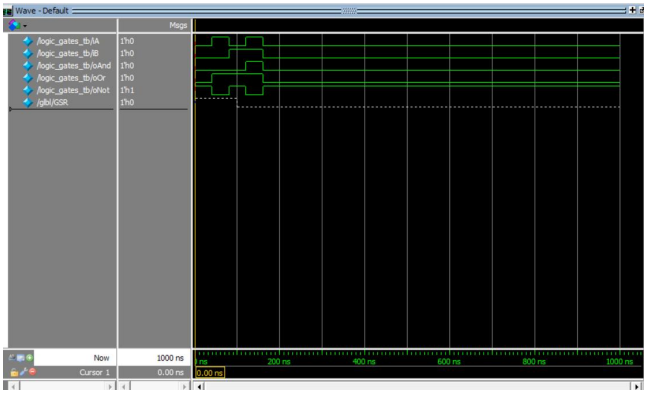
6.1_1 modelsim 仿真波形图



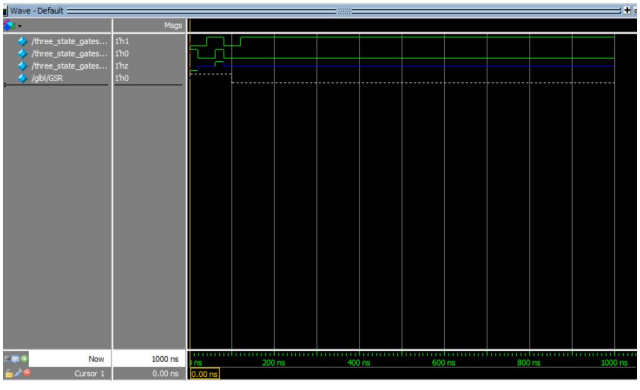
6.1_2 modelsim 仿真波形图



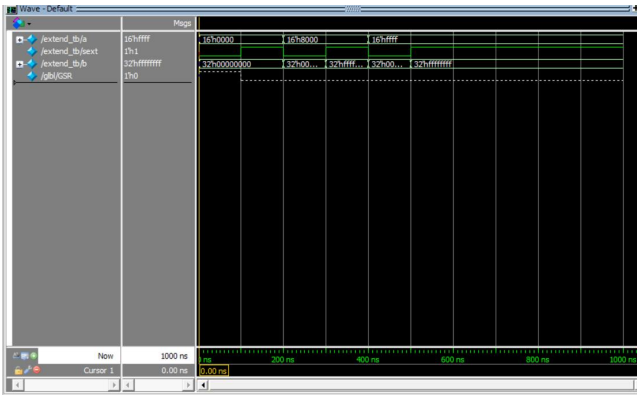
6.1_3 modelsim 仿真波形图



6.1_4 modelsim 仿真波形图

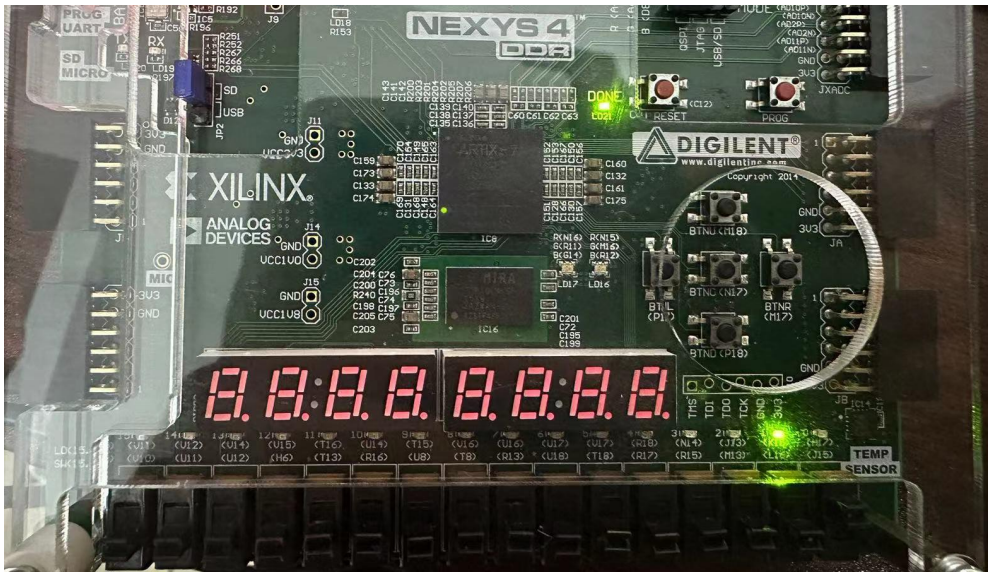


6.1_5 modelsim 仿真波形图

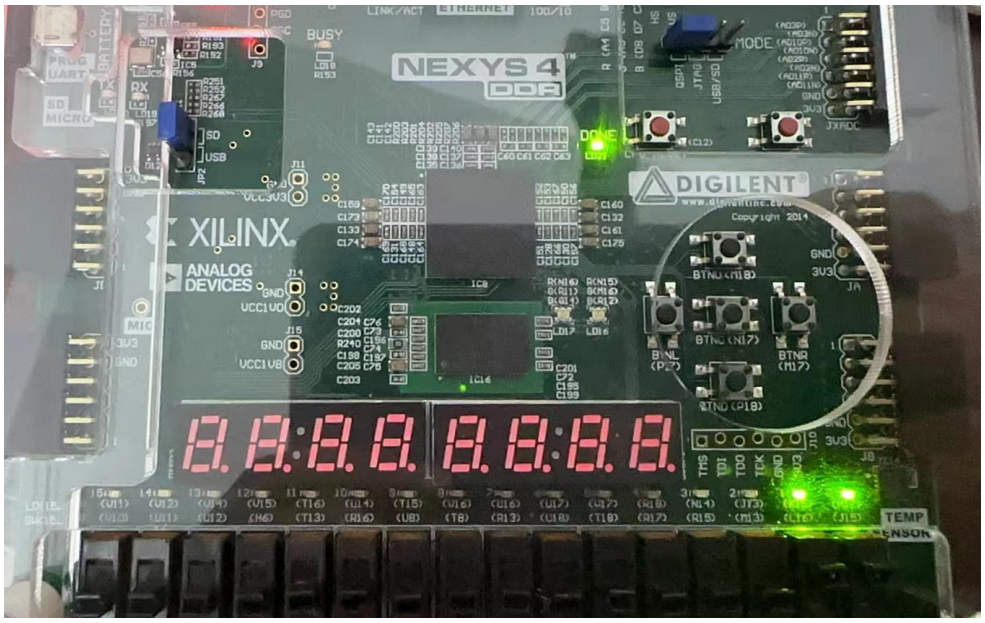


6.1_1 下板实验结果

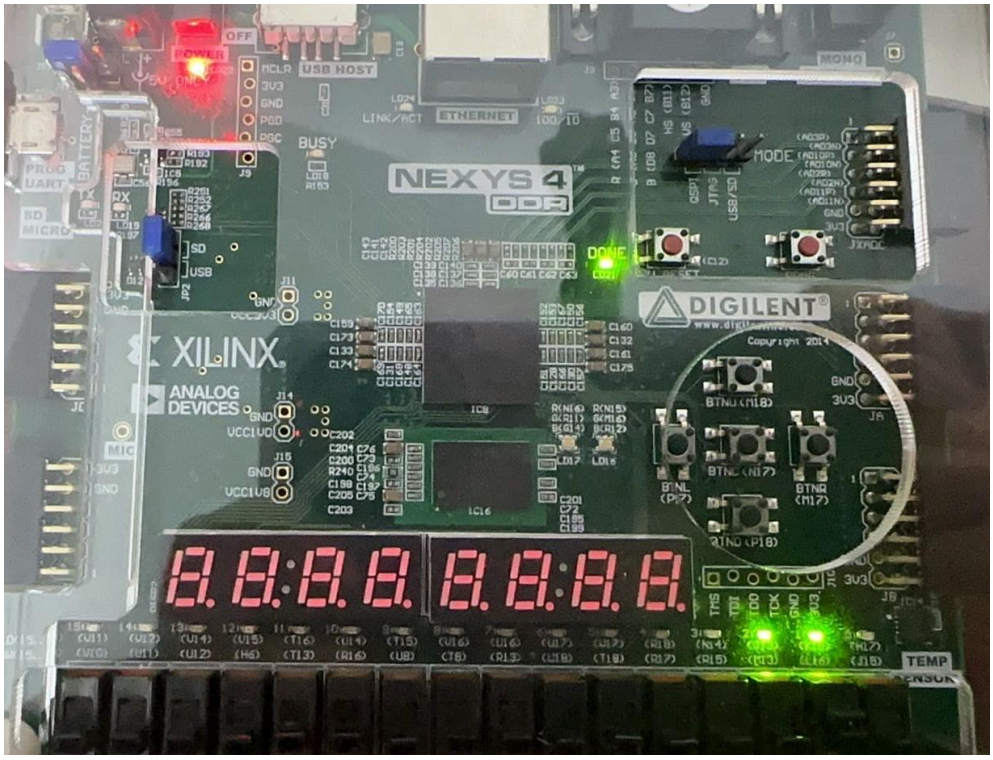
iA=1 iB=0



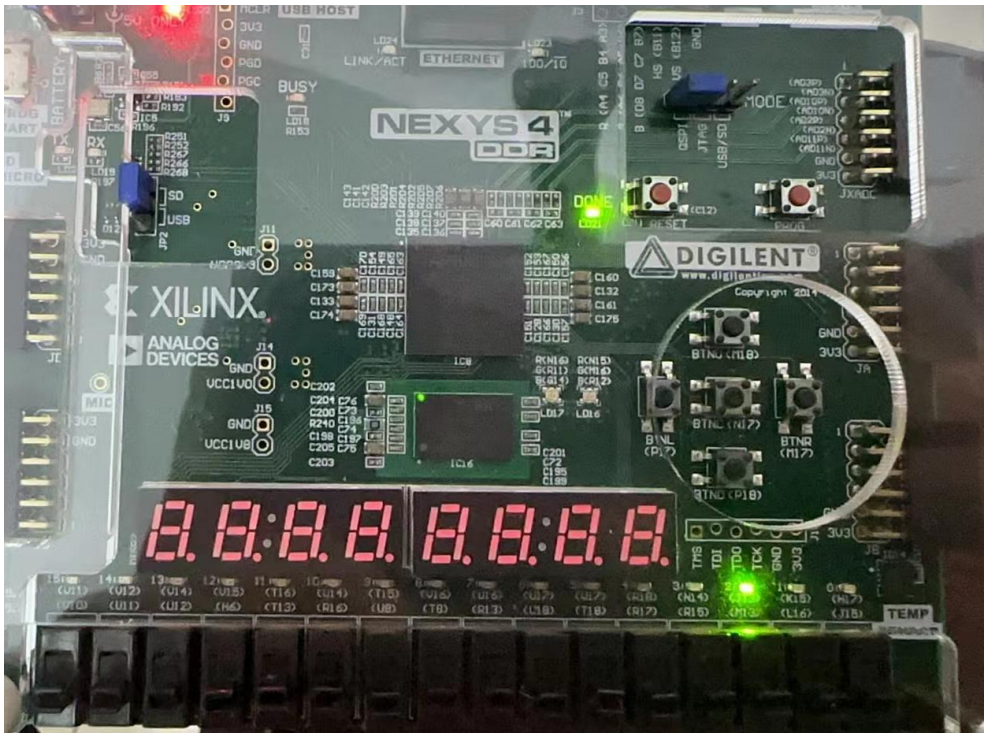
iA=1 iB=1



iA=0 iB=1



iA=0 iB=0



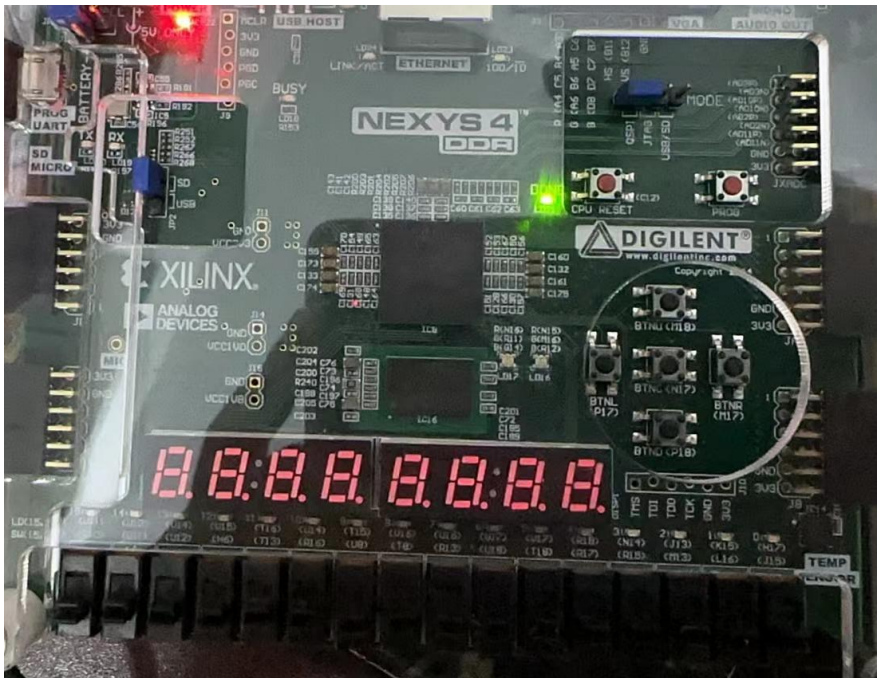
已验证 6.1_2 下板实验结果/6.1_3 下板实验结果与 6.1_1 下板实验结果相同

6.1_4 下板实验结果

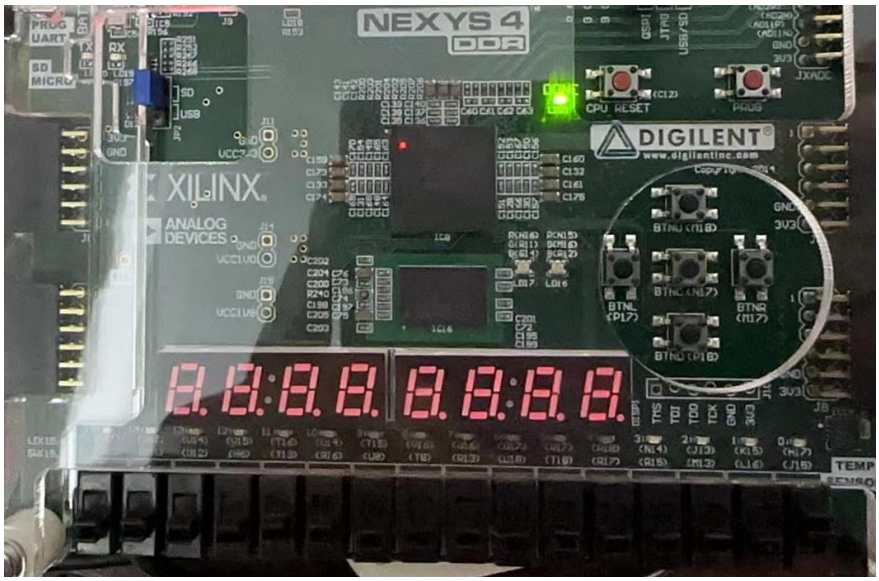
Truth Table

iA	iEna	oTri
0	0	z
1	0	z
0	1	0
1	1	1

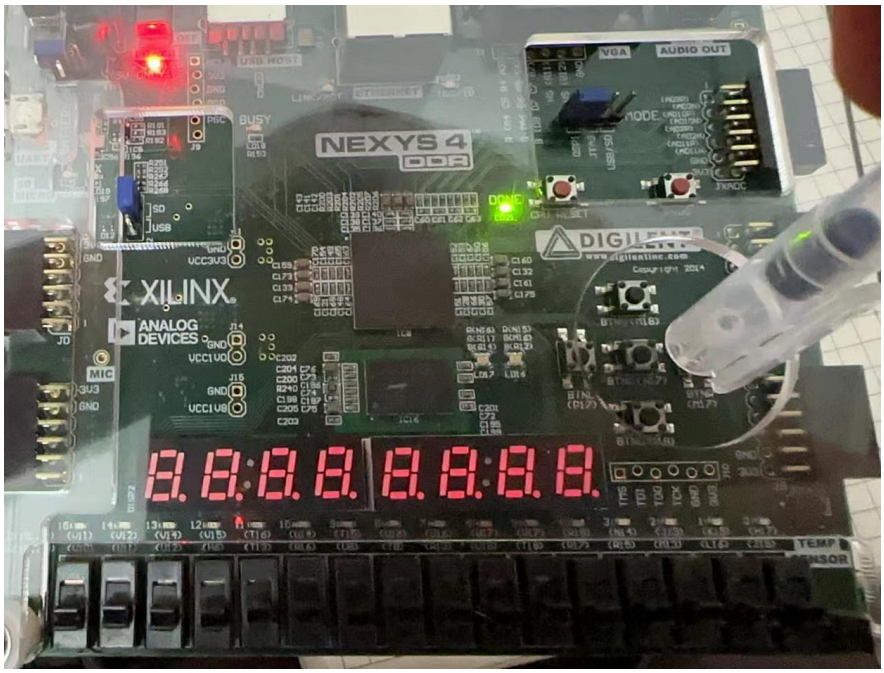
iA=0 iEna=0



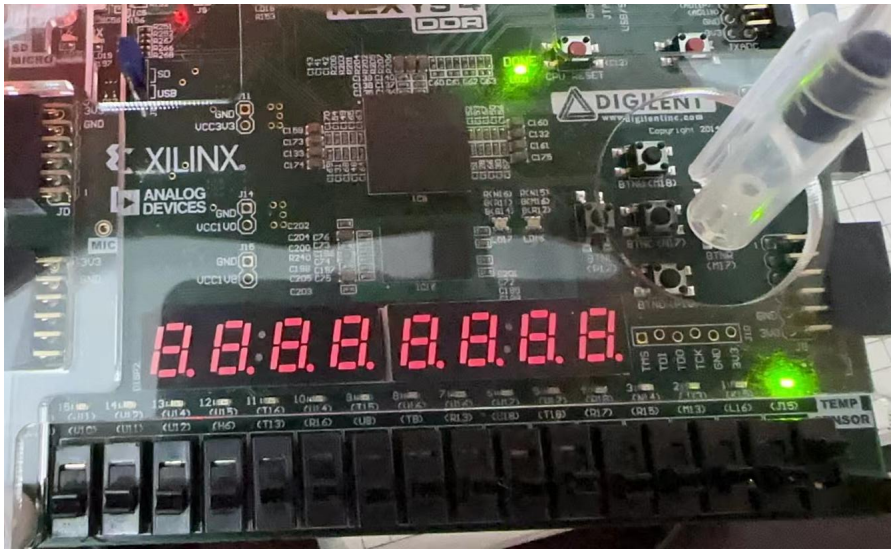
iA=1 iEna=0



iA=0 iEna=1


NEXYS 4
DDR
XILINX
ANALOG DEVICES
VGA
AUDIO OUT

iA=1 iEna=1


XILINX
ANALOG DEVICES