**同济大学计算机系**

**数字逻辑课程实验报告**

****

**学 号**

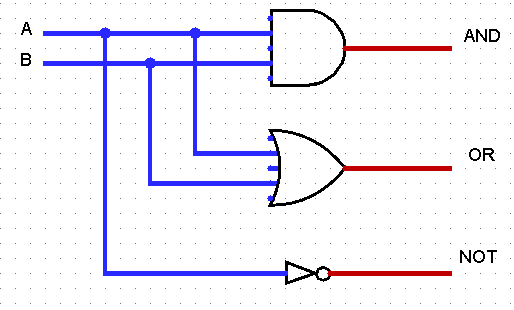
**姓 名**

**专 业**

**授课老师**

1. 实验内容
   * + 1. 使用三种不同的方式：门级、数据流级、行为级，来描述基本门电路。
       2. 三态门的建模。
       3. 对输入数据进行扩展，将16位输入转变为32位数据。
       4. 学习使用vivado，modelsim和开发板。
2. 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）



1. 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）

Logic\_gates\_1

module logic\_gates\_1(

input iA,

input iB,

output oAnd,

output oOr,

output oNot

);

and(oAnd, iA, iB);

or(oOr, iA, iB);

not(oNot, iA);

endmodule

Logic\_gates\_2

module logic\_gates\_2(

input iA,

input iB,

output oAnd,

output oOr,

output oNot

);

assign oAnd = iA & iB;

assign oOr = iA & iB;

assign oNot = ~iA;

endmodule

Logic\_gates\_3

module logic\_gates\_3(

input iA,

input iB,

output reg oAnd,

output reg oOr,

output reg oNot

);

always @ (\*)

begin

oAnd = iA & iB;

oOr = iA | iB;

oNot = ~iA;

end

endmodule

three\_state\_gates

module three\_state\_gates(

input iA,

input iEna,

output oTri

);

assign oTri = (iEna == 1) ? iA : 'bz;

endmodule

extend

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

1. 测试模块建模

（要求列写各建模模块的test bench模块代码）

Logic\_gates\_1

`timescale 1ns / 1ns

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(Not));

endmodule

Logic\_gates\_2

`timescale 1ns / 1ns

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(Not));

endmodule

Logic\_gates\_3

`timescale 1ns / 1ns

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(Not));

endmodule

Three\_state\_gates

`timescale 1ns / 1ns

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

extend

`timescale 1ns / 1ns

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;

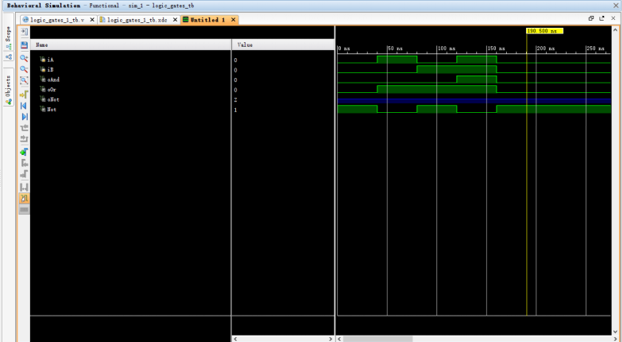
end

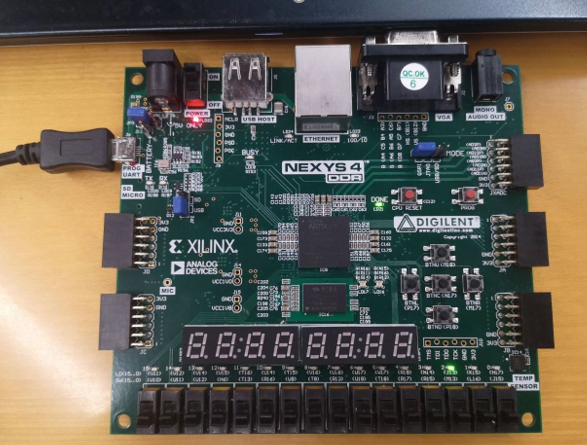
endmodule

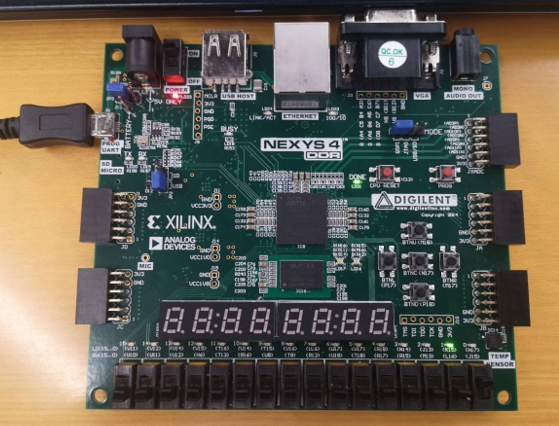
1. 实验结果

（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））

Logic\_gates\_1

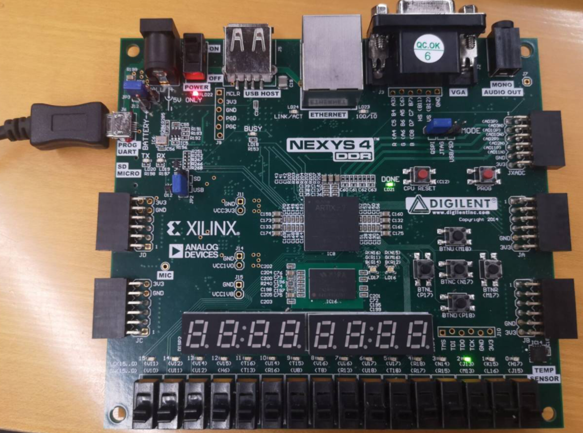


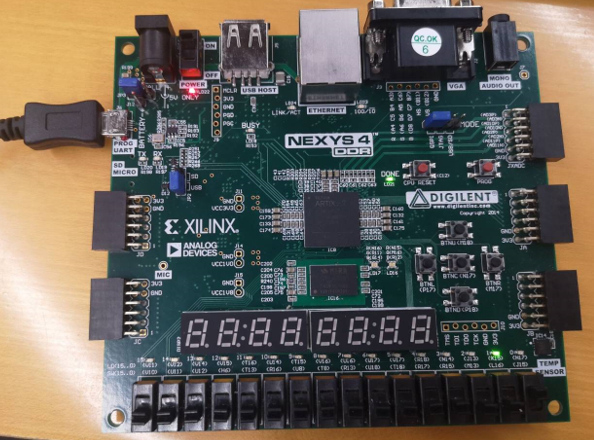




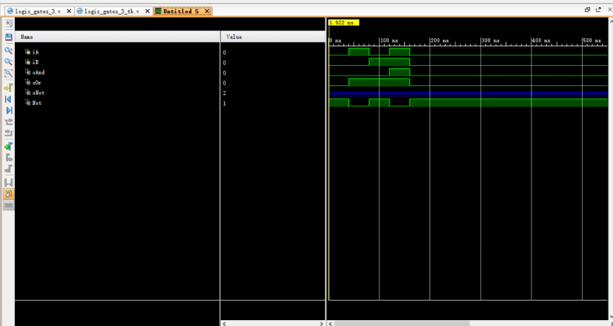
Logic\_gates\_2

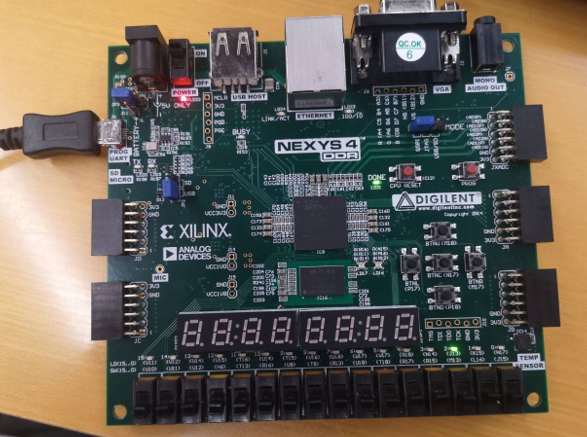


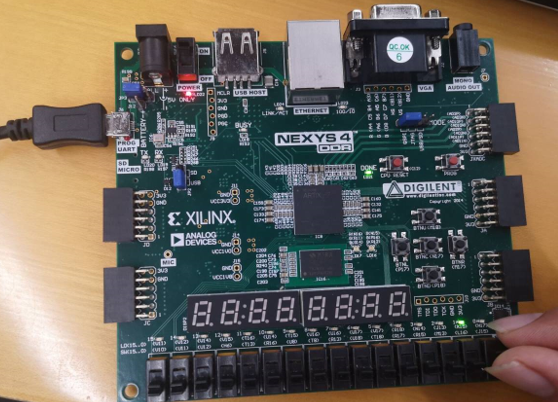




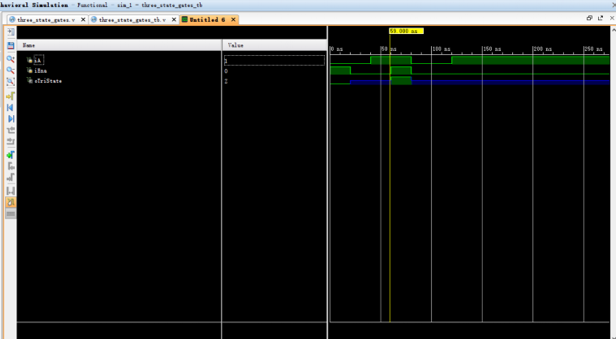
Logic\_gates\_3

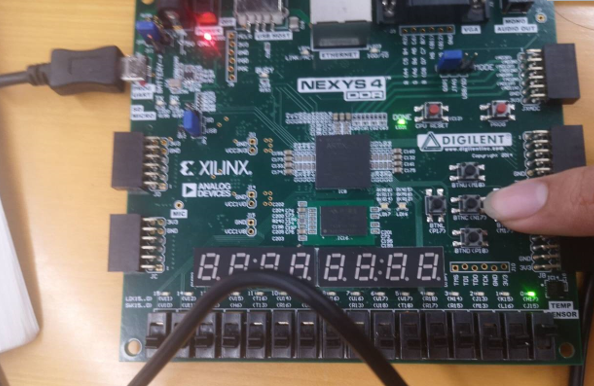






Three\_state\_gates





Extend

