

同济大学计算机系

数字逻辑课程综合实验报告



学 号 2352595

姓 名 张嘉麟

专 业 计算机科学与技术

授课老师 郭玉臣

一、实验内容

本实验主要包括两个部分：基本门电路设计与数据扩展模块的实现。通过 Verilog HDL 语言实现各模块，并进行功能验证。具体步骤包括：

- 1. 设计基本的与、或、非门电路。
- 2. 实现三态门电路，控制信号的传输。
- 3. 设计一个数据扩展模块，将输入数据扩展为 32 位，支持符号扩展与零扩展。

二、与或非、三态门系统总框图

输入模块：接收外部信号，处理并传递给逻辑运算模块。

逻辑运算模块：包括与、或、非电路。

模块功能描述表：

| 模块名 | 功能描述 |
|----------------|------------------------|
| logic_gates_1 | 实现与、或、非门的组合，使用结构化描述方式。 |
| logic_gates_2 | 实现与、或、非门的组合，使用数据流描述方式。 |
| logic_gates_3 | 实现与、或、非门的组合，使用行为描述方式。 |
| logic_gates_tb | 测试模块，产生不同的输入信号组合以验证逻辑。 |

测试用例表：

| 测试用例编号 | iA | iB | 预期输出 (oAnd, oOr, oNot) |
|--------|----|----|------------------------|
| TC1 | 0 | 0 | (0, 0, 1) |
| TC2 | 0 | 1 | (0, 1, 1) |
| TC3 | 1 | 0 | (0, 1, 0) |
| TC4 | 1 | 1 | (1, 1, 0) |

模块接口定义表：

| 模块名 | 输入信号 | 输出信号 |
|---------------|--------|-----------------|
| logic_gates_1 | iA, iB | oAnd, oOr, oNot |
| logic_gates_2 | iA, iB | oAnd, oOr, oNot |

| 模块名 | 输入信号 | 输出信号 |
|---------------|--------|-----------------|
| logic_gates_3 | iA, iB | oAnd, oOr, oNot |

•

状态转移真值表：

| 当前状态 (iA, iB) | 输出状态 (oAnd, oOr, oNot) |
|---------------|------------------------|
| (0, 0) | (0, 0, 1) |
| (0, 1) | (0, 1, 1) |
| (1, 0) | (0, 1, 0) |
| (1, 1) | (1, 1, 0) |

接口信号定义表：

| 信号名 | 类型 | 描述 |
|------|----|--------|
| iA | 输入 | 输入信号 A |
| iB | 输入 | 输入信号 B |
| oAnd | 输出 | 与运算结果 |
| oOr | 输出 | 或运算结果 |
| oNot | 输出 | 非运算结果 |

三、三态门系统设计

三态门实现

```

verilog 代码:
module logic_gates(iA, iB, oAnd, oOr, oNot);
input iA, iB;
output oAnd, oOr, oNot;
assign oAnd = iA & iB;
assign oOr = iA | iB;
assign oNot = ~iA;
Endmodule

```

模块功能描述表：

| 模块名 | 功能描述 |
|----------------------|---------------------------|
| three_state_gates_1 | 实现三态门的基本功能，使用结构化描述方式。 |
| three_state_gates_2 | 实现三态门的基本功能，使用数据流描述方式。 |
| three_state_gates_3 | 实现三态门的基本功能，使用行为描述方式。 |
| three_state_gates_tb | 测试模块，产生不同的输入信号组合以验证三态门逻辑。 |

测试用例表：

| 测试用例编号 | iA | iEna | 预期输出 (oTriState) |
|--------|----|------|------------------|
| TC1 | 0 | 0 | Z |
| TC2 | 0 | 1 | 0 |
| TC3 | 1 | 0 | Z |
| TC4 | 1 | 1 | 1 |

模块接口定义表：

| 模块名 | 输入信号 | 输出信号 |
|---------------------|----------|-----------|
| three_state_gates_1 | iA, iEna | oTriState |
| three_state_gates_2 | iA, iEna | oTriState |
| three_state_gates_3 | iA, iEna | oTriState |

状态转移真值表：

| 当前状态 (iA, iEna) | 输出状态 (oTriState) |
|-----------------|------------------|
| (0, 0) | Z |
| (0, 1) | 0 |
| (1, 0) | Z |
| (1, 1) | 1 |

接口信号定义表：

| 信号名 | 类型 | 描述 |
|-----|----|--------|
| iA | 输入 | 输入信号 A |

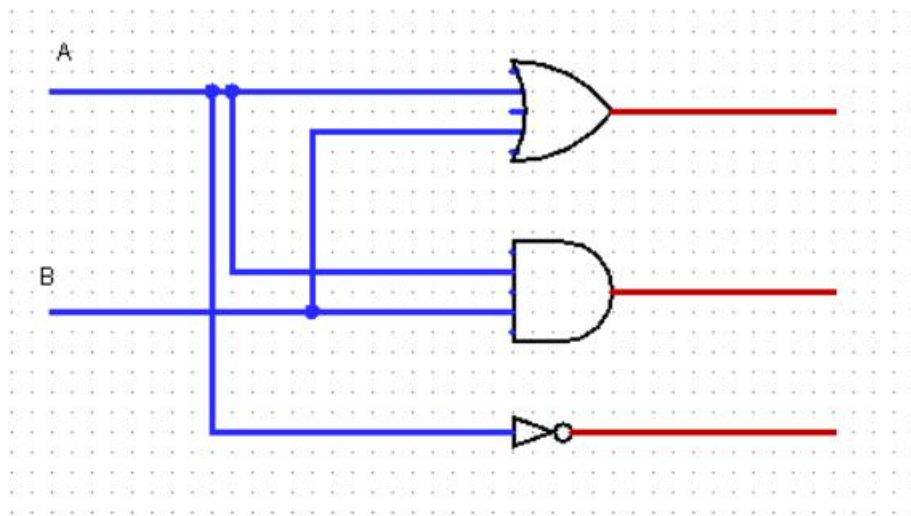
| 信号名 | 类型 | 描述 |
|-----------|----|--------|
| iEna | 输入 | 使能信号 |
| oTriState | 输出 | 三态输出结果 |

四、系统控制器设计

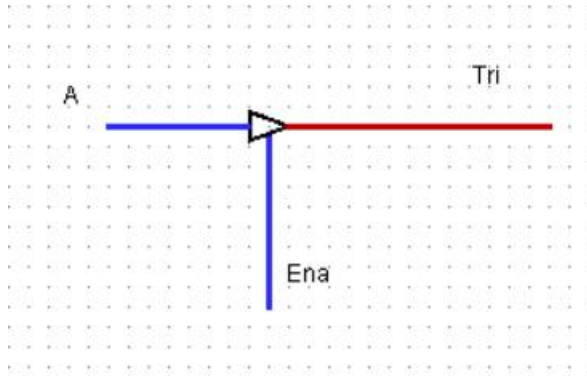
控制器设计的核心在于状态机的设计，包括以下步骤：

1. **状态转移真值表**：真值表记录每个状态下的输出与下一个状态，为后续逻辑表达式的推导提供依据。
2. **激励函数和逻辑表达式推导**：
 - **核心步骤**：从真值表推导出次态激励函数和控制命令的逻辑表达式。例如，使用卡诺图简化逻辑表达式。
3. **Logisim 逻辑方案图**：将逻辑表达式实现为电路图，通过 Logisim 工具验证逻辑正确性。

与或非门电路实验：



三态门实验：



五、测试模块建模

测试模块的设计至关重要，确保每个子系统功能正常：

- **Test Bench 描述：**

设计 test bench 时，定义输入信号并设置激励波形，观察输出结果是否符合预期。

核心步骤： 确保每个模块的 test bench 能够全面覆盖可能的输入情况，从而提高测试的有效性和可靠性。

与或非门实验 tb.v 文件核心代码：

```
module logic_gates_tb;
reg iA;
reg iB;
wire oAnd;
wire oOr;
wire oNot;
initial begin
    iB = 0; #40 iB = 1; #40 iB = 0;
end
initial begin
    iA = 0; #40 iA = 1; #40 iA = 0;
end
end
logic_gates logic_gates_inst(
    .iA(iA),
    .iB(iB),
    .oAnd(oAnd),
    .oOr(oOr),
    .oNot(oNot)
);
endmodule
```

三态门实验 tb.v 文件核心代码：

```
three_state_gates uut (  
  
    .iA(iA),  
  
    .iEna(iEna),  
  
    .oTri(oTriState)  
  
);
```

```
initial begin  
  
    // 初始化输入信号  
  
    iA = 0;  
  
    iEna = 1;  
  
    #20;  
  
    // 改变 iA 和 iEna  
  
    iA = 1;  
  
    iEna = 0;  
  
    #40;  
  
    // 改变 iA 和 iEna  
  
    iA = 0;  
  
    iEna = 1;  
  
    #40;  
  
    // 改变 iA 和 iEna  
  
    iA = 1;  
  
    iEna = 0;
```

```

#20;

// 结束仿真

$finish;

end

// 显示输出信号

always @(posedge iA or posedge iEna) begin

    $display("iA=%b, iEna=%b, oTriState=%b", iA, iEna, oTriState);

end

endmodule

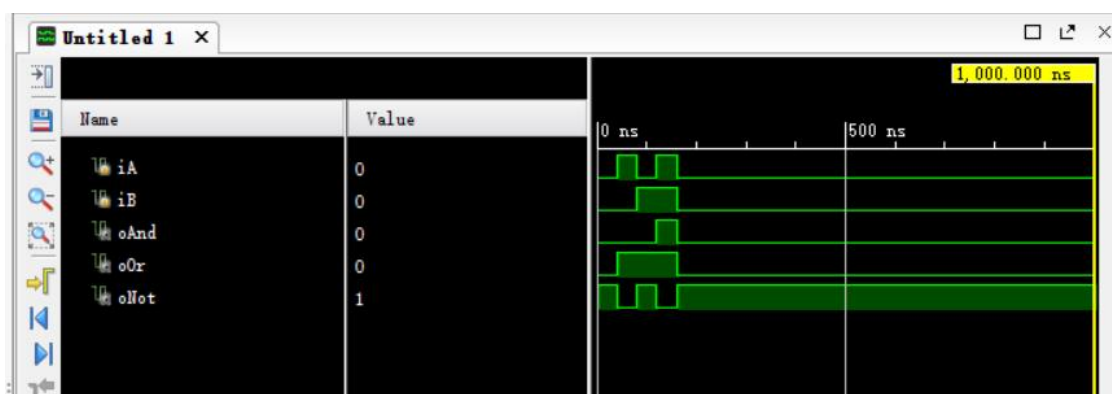
```

五、实验结果

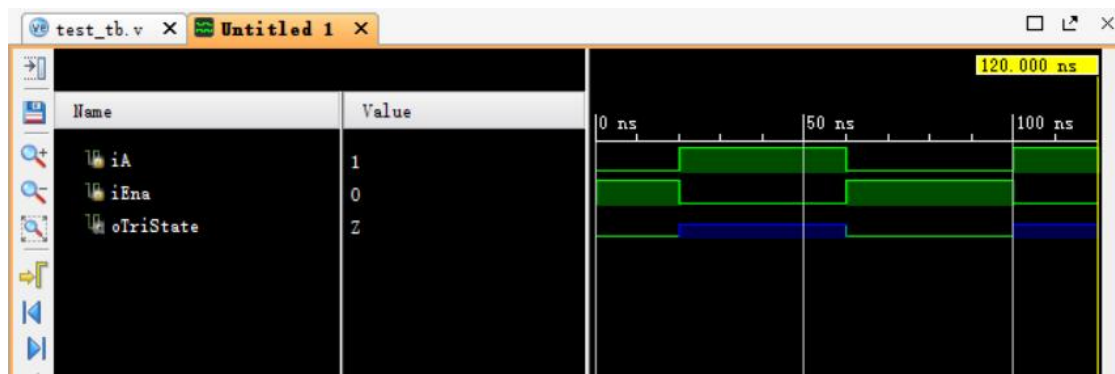
这一部分通过截图和数据展示实验验证的结果：

- **Logisim 逻辑验证图：**展示设计电路的逻辑实现，确认与预期逻辑一致。（见上文）
- **ModelSim 仿真波形图：**提供信号在时间域上的变化图，验证时序关系是否正确。

与或非门仿真图：

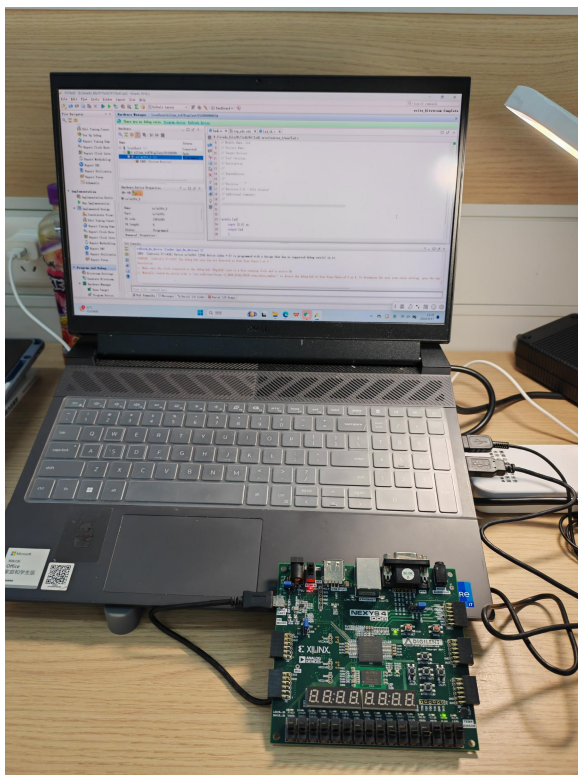


三态门仿真图：

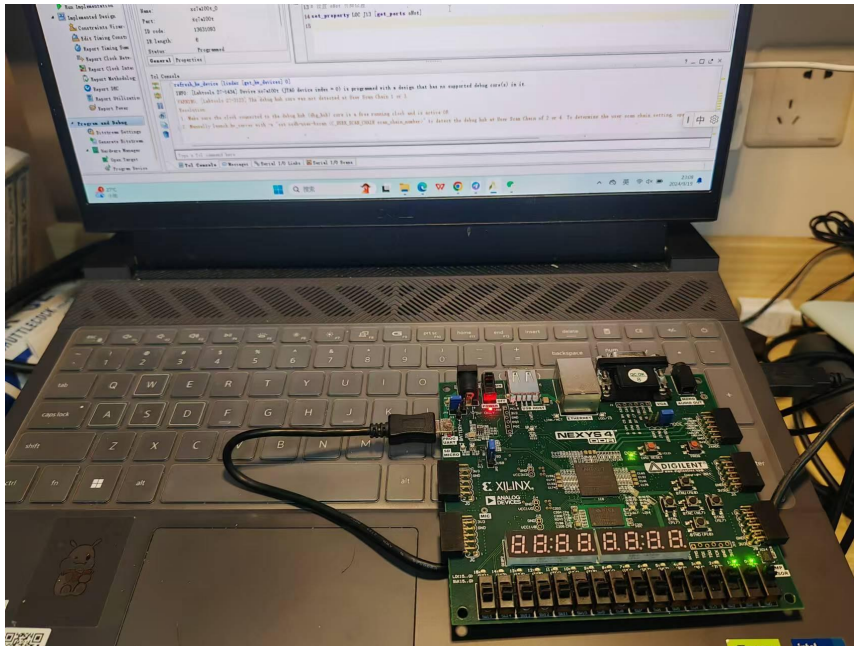


- **实际硬件测试结果：**通过开发板进行测试，记录测试过程中 LED 状态的变化，附测试照片：

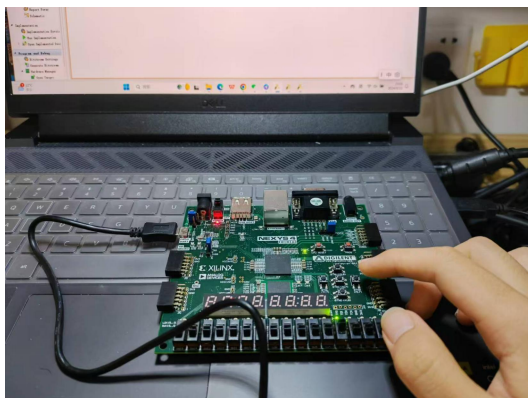
LED 小灯实验：



与或非门电路实验：



三态门实验：



（点击手指所指按键后亮灯）

核心步骤： 确保所有测试结果与设计预期相符，特别是在时序和逻辑正确性上，进行反复验证。

六、结论

在结论部分总结实验成果和经验：

- **成功之处：** 指出在逻辑设计、Verilog 编程以及仿真方面的收获，强调项目的整体成功。
- **不足之处：** 反思在模块连接和信号时序方面的潜在问题，建议改进设计流程。有几个地方尝试了更改小灯泡的对应关系，最后没有成功，交给 chatgpt 之后成功了。

七、心得体会及建议

结合实验经历，分享个人感悟：

- **收获：**深入理解数字电路的设计与实现过程，提升了 Verilog HDL 的编程能力。
- **建议：**在未来的实验中，可以增加对复杂电路的设计挑战，提升模块化设计和测试的质量。