

数字电路与数字系统实验报告

实验一：双控开关和表决器

院系： 人工智能学院

姓名： 方盛俊

学号： 201300035

班级： 人工智能 20 级 2 班

邮箱： 201300035@smail.nju.edu.cn

时间： 2021 年 7 月 5 日

目录	1
----	---

目录

一、 实验目的	2
二、 实验原理 (背景知识)	2
1. 晶体管	2
2. 逻辑表达式	3
3. Logisim 的基础使用	3
三、 实验环境/器材	5
四、 实验内容	5
1. 利用晶体管构建或门	5
a) 实验原理	5
b) 实验步骤	6
c) 仿真验证	8
d) 实验结果	9
2. 双控开关	9
a) 实验原理	9
b) 实验步骤	10
c) 仿真验证	10
d) 实验结果	11
3. 多数表决器	12
a) 实验原理	12
b) 实验步骤	12
c) 仿真验证	13
d) 实验结果	14
五、 实验中遇到的问题和解决的办法	14
1. 本地仿真模拟与平台上运行不一致	14

一、 实验目的

1. 熟悉 Logisim 软件的基本使用方法
2. 掌握使用晶体管实现基本逻辑部件的方法
3. 利用基础元器件库设计简单数字电路
4. 了解子电路的设计和应用
5. 掌握分线器、隧道、探针等组件的使用方法

二、 实验原理 (背景知识)

1. 晶体管

我们电路设计中使用的晶体管，即基于金属氧化物半导体场效应晶体管的 **CMOS** (Complementary Metal-Oxide Semiconductor)。

MOS 晶体管三极：

- 栅极 gate
- 源极 source
- 漏极 drain

MOS 晶体管可被模型化为一种 3 端子压控电阻导体，将电压加到一个端子，来控制其他两个端子间的电阻。

MOS 晶体管分为：

- N 沟道型 NMOS，N 型杂质有磷或者锑
- P 沟道型 PMOS，P 型杂质有硼或者镓

栅极和源极之间电压 V_{gs} 控制源极和漏极间电阻 R_{ds} 的大小，进而可以组合出不同的基本电路器件。

2. 逻辑表达式

在实际构建组合电路之前，我们常常先把真值表写出，例如异或门：

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

进而根据真值表推导出规范的逻辑表达式，常常是最简化的积之和表达式，如异或的积之和表达式：

$$F = \overline{A}B + A\overline{B}$$

进而根据推导出来的逻辑表达式，选取需要的基本逻辑门，构建该电路。

3. Logisim 的基础使用

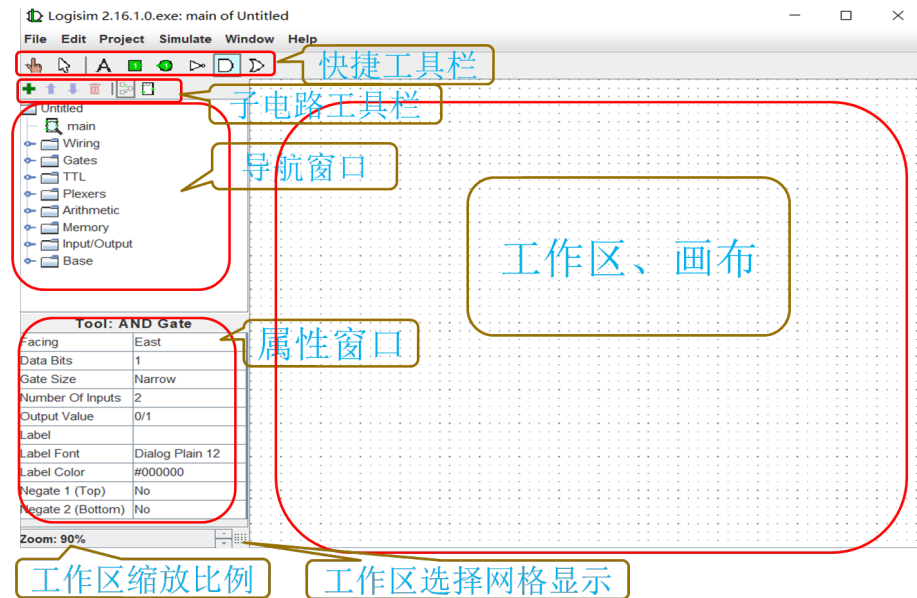
Logisim 是 Carl Burch 开发的一个设计和仿真数字逻辑电路的图形化教学工具。非常便于学习数字逻辑电路设计的基本概念，能够从简单的子电路分层构建较复杂的数字电路。

2001 年 4 月-2011 年 5 月，版本 V2.7.1，基于 Java 开发，跨平台运行，按照 GNU 开源协议授权使用。Windows 平台使用 EXE 文件，MAC 或 Linux 平台使用 JAR 文件。

在 SourceForge 上有多个 Logisim 项目，其中意大利语版本 2017 年开

始修改，在 Java 10 及以上版本上重新编译，修改 bug，添加新组件。最新版本：V2.16.1.0 2020-04-18

电路文件类型为 circ，格式为 xml，所有子电路都是以这样的标记出现，`<circuit name="xxx"></circuit>`，包含电路外观。



在查阅相关文档之后，发现有一个命令可以模仿头歌平台的运行，本地进行样例测试。

即命令：

```
java -jar logisim.jar file.circ -tty table
```

使用这个命令，就可以较为方便地先在本地进行样例测试，以防提交过程中过多的错误尝试。

三、 实验环境/器材

Logisim-ITA V2.16.1.2

<https://sourceforge.net/projects/logisimit/>

头歌线上评测平台

<https://www.educoder.net/classrooms/10924/>

四、 实验内容

1. 利用晶体管构建或门

a) 实验原理

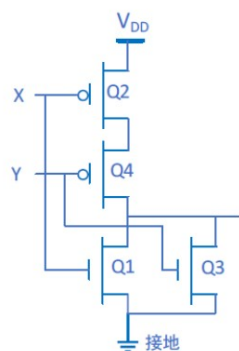
二极管、三极管、MOS 管都是基础的电器元件，其中 MOS 管（即晶体管）是现代数字电路的基础。

与门、或门、非门等门电路是数字电路中常用的基础门电路。本次实验的内容是通过利用晶体管构建实现基本逻辑运算的门电路，这里选择的是或门电路。

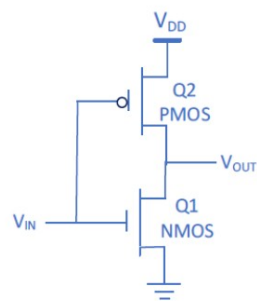
根据数字电路原理，或门是由或非门级联一个反相器构成。即公式：

$$OR(x, y) = NOT(NOR(x, y))$$

或非门和反向器的原理如图：



或非门



反相器

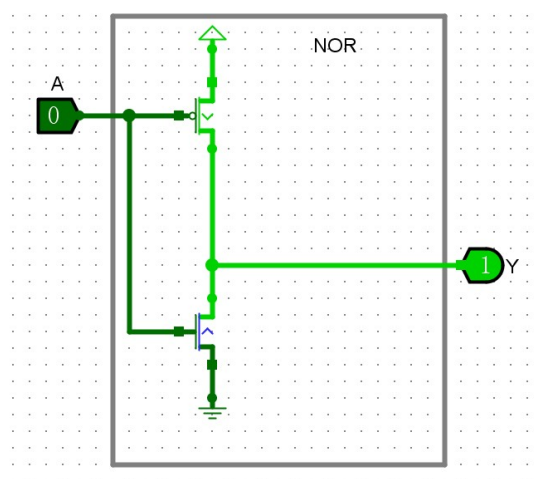
b) 实验步骤

1) 构建非门

需要电路部件：1 对晶体管、1 个输入引脚、1 个输出引脚、1 个电源、1 个接地线。

放入一个晶体管 P-Type，朝向为 South；再放入一个晶体管 N-Type，朝向为 North。

按照原理图，将非门的电路连接好，如图：

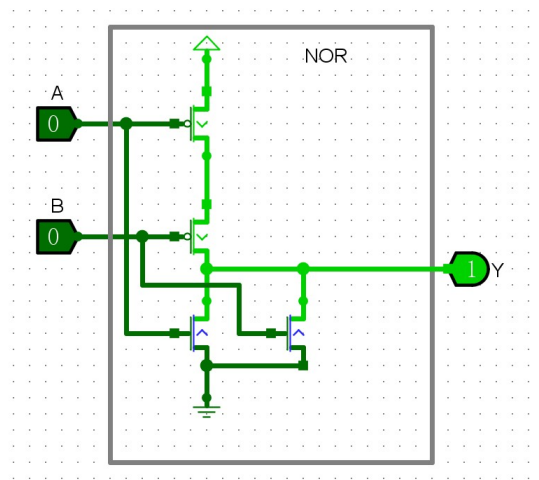


2) 构建或非门

需要电路部件：2 对晶体管、2 个输入引脚、1 个输出引脚、1 个电源、1 个接地线。

放入两个晶体管 P-Type，朝向为 South；再放入两个晶体管 N-Type，朝向为 North。

按照原理图，将或非门的电路连接好，如图：

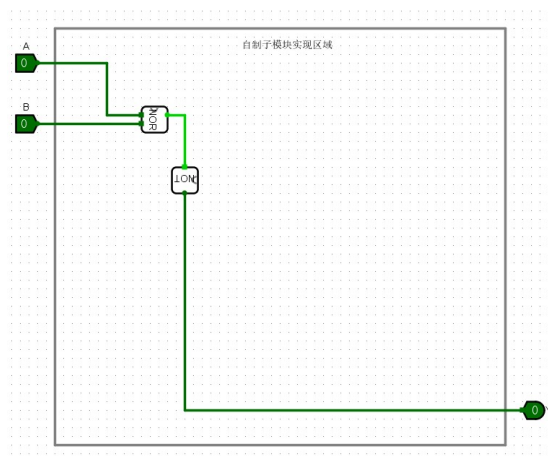


3) 利用非门和或非门构建或门

根据公式

$$OR(x, y) = NOT(NOR(x, y))$$

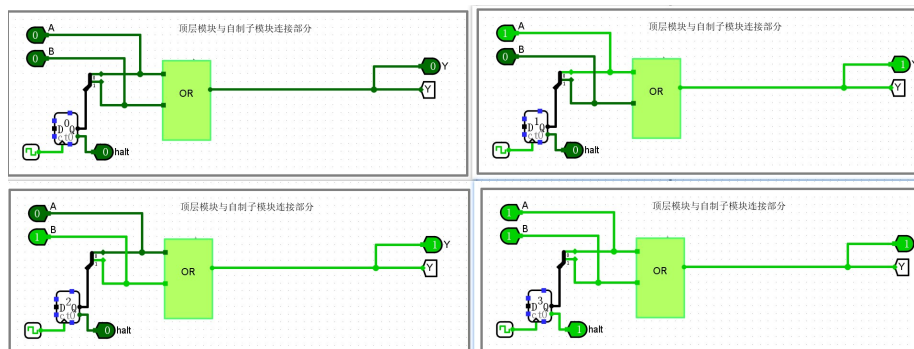
将非门和或非门连接：



一个或门电路就搭好了。

c) 仿真验证

我们先在本地 Logisim 软件进仿真验证，按下快捷键 **Ctrl + K**, Logisim 便开始了仿真。



可见，电路的输出符合描述。

再在本地运行命令：

```
java -jar logisim.jar exp1/exp1-1.circ -tty table
```

便可得到电路模拟输出：

```
(base) PS D:\Project\Circ> java -jar logisim.jar exp1\exp1-1.circ -tty table
Although an Internet connection should be available, the system couldn't connect to the URL requested by the auto-updater
If the error persist, please contact the software maintainer
-- AUTO-UPDATE ABORTED --
0      0      0
1      0      1
0      1      1
1      1      1
```

也符合预期，说明电路在本地仿真验证通过了。

d) 实验结果

提交到头歌平台，获得输出如下：



实验结果一致。

2. 双控开关

a) 实验原理

双控开关是生活中常见的开关，双控开关是指两个拨动开关能够同时控制一盏灯。

例如，在卧室中可以通过门口或床头的开关来开关卧室的灯，这样开关灯时就不用跑来跑去了。

实际上，双控开关的电路原理就是异或电路，两个开关的状态作为输入，灯亮与否作为输出的异或电路。

根据数字电路原理，异或电路可以由一个公式表示：

$$F = A \oplus B = \overline{A}B + A\overline{B}$$

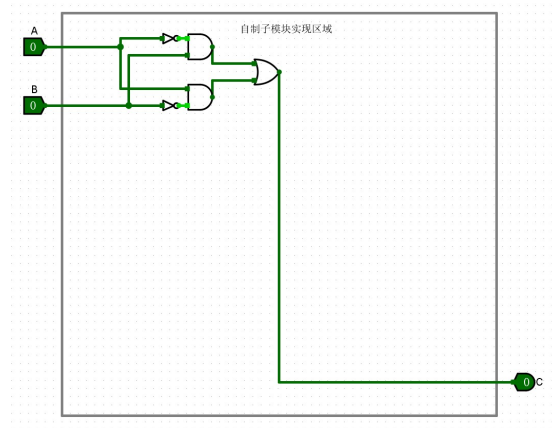
b) 实验步骤

1) 放入电路原件

分析公式可知，我们需要电路部件：2 个输入引脚，1 个输出引脚，2 个非门，2 个 2 输入与门、1 个 2 输入或门。

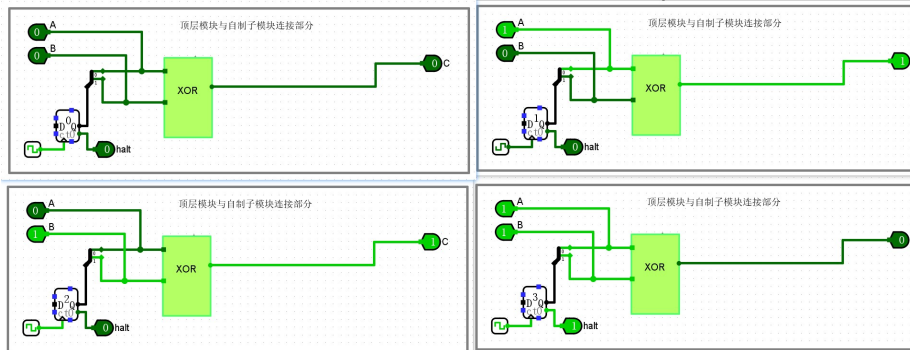
2) 连接异或电路

按照公式，两个输入分别取反之后作为两个与门的输入，两个与门的输出再作为或门的输入，最后将或门输出导向输出引脚，如图：



c) 仿真验证

我们先在本地 Logisim 软件进行仿真验证，按下快捷键 **Ctrl + K**, Logisim 便开始了仿真。



可见，电路的输出符合描述。

再在本地运行命令：

```
java -jar logisim.jar exp1/exp1-2.circ -tty table
```

便可得到电路模拟输出：

```
(base) PS D:\Project\Circ> java -jar logisim.jar exp1/exp1-2.circ -tty table
0      0      0
1      0      1
0      1      1
1      1      0
```

也符合预期，说明电路在本地仿真验证通过了。

d) 实验结果

提交到头歌平台，获得输出如下：

1/1 全部通过					
测试集1			消耗内存317.33MB 代码执行时长: 3.74秒		
预期输出			实际输出		
0	0	0	0	0	0
1	0	1	1	0	1
0	1	1	0	1	1
1	1	0	1	1	0

实验结果一致。

3. 多数表决器

a) 实验原理

三路多数表决器，顾名思义，就是三个人进行投票，两票及以上为同意，零票或一票为不同意。

用电路来描述，就是有两路及以上输入为 1 时，输出才为 1，否则为 0，用真值表描述如下：

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

用公式来表达也很简单：

$$F = A \cdot B + B \cdot C + A \cdot C$$

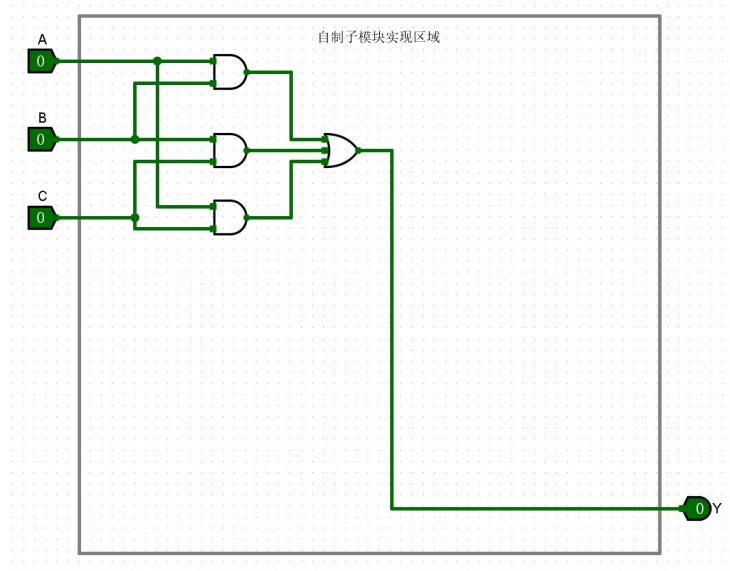
b) 实验步骤

1) 放入电路原件

分析公式可知，我们需要电路部件：3 个输入引脚，1 个输出引脚，3 个 2 输入与门、1 个 3 输入或门。

2) 连接异或电路

连接电路如图：



c) 仿真验证

在本地运行命令

```
java -jar logisim.jar exp1/exp1-3.circ -tty table
```

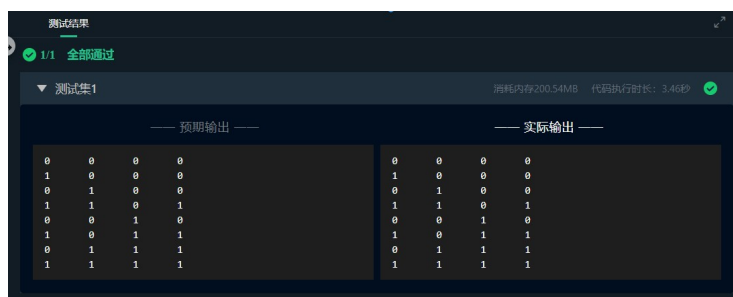
便可得到电路模拟输出：

```
(base) PS D:\Project\Circ> java -jar logisim.jar exp1/exp1-3.circ -tty table
Although an Internet connection should be available, the system couldn't connect
r
If the error persist, please contact the software maintainer
-- AUTO-UPDATE ABORTED --
0      0      0      0
1      0      0      0
0      1      0      0
1      1      0      1
0      0      1      0
1      0      1      1
0      1      1      1
1      1      1      1
```

也符合预期，说明电路在本地仿真验证通过了。

d) 实验结果

提交到头歌平台，获得输出如下：



预期输出				实际输出			
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0
1	1	0	1	1	1	0	1
0	0	1	0	0	0	1	0
1	0	1	1	1	0	1	1
0	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1

实验结果一致。

五、 实验中遇到的问题和解决的办法

1. 本地仿真模拟与平台上运行不一致

在本地运行相关的电路代码的时候，有时候会碰到本地和平台上输出结果不一致。

如多数表决器，在本地运行的时候输出如下：

```
(base) PS D:\Project\Circ> java -jar logisim.jar exp1-3.circ -tty table
Although an Internet connection should be available, the system couldn't co
r
If the error persist, please contact the software maintainer
-- AUTO-UPDATE ABORTED --
0      0      0      0
1      0      0      0
0      1      0      0
1      1      0      1
0      0      1      0
1      0      1      1
0      1      1      1
1      1      1      1
```

而在平台上运行时却成了这样：

—— 预期输出 ——				—— 实际输出 ——			
0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0
0	1	0	0	0	1	0	0
1	1	0	1	1	1	0	0
0	0	1	0	0	0	1	0
1	0	1	1	1	0	1	0
0	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1

说明平台上的运行环境可能和本地运行有一定差异。

这个问题难以独自解决，于是请教了老师，最后通过修改平台的评判代码的方式，解决了这个问题。