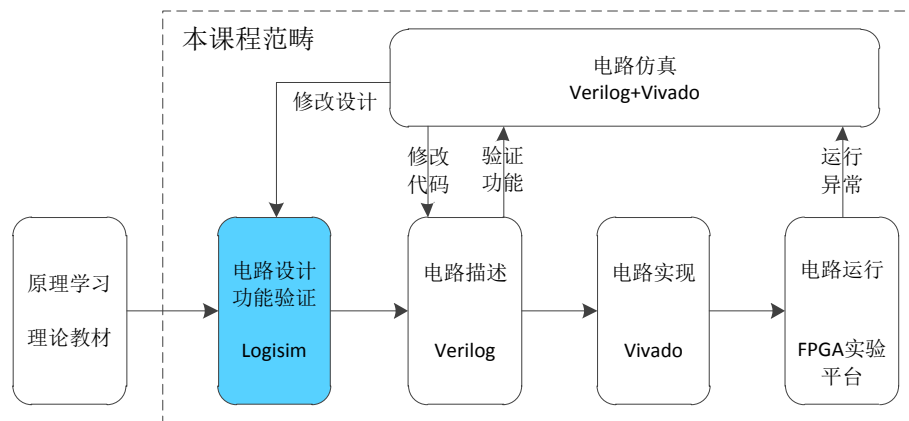


实验 01 Logisim 入门

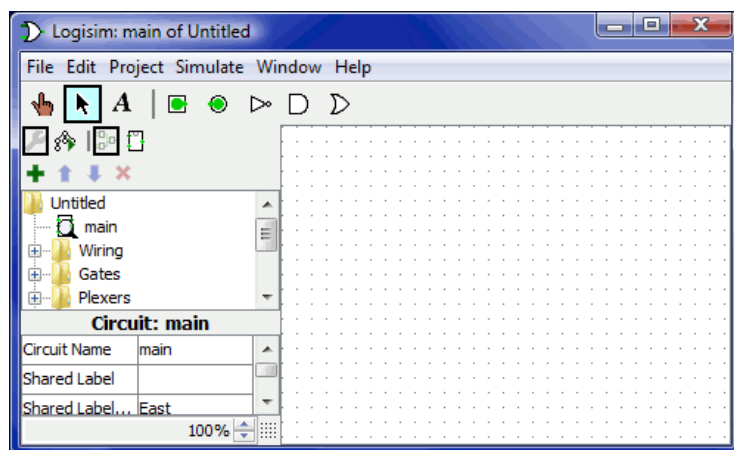
简介

本次实验我们将学习与原理图设计、功能验证相关的工具和知识。
包括 Logisim 工具的使用和基本逻辑门电路的相关知识。



Logisim 是一种用于设计和仿真数字逻辑电路的教学仿真工具。
操作简单，显示直观，通过简单的鼠标拖拽连线即可完成数字电路设计和仿真，子电路封装功能更可以方便用户构建更大规模的数字电路。

Logisim 基于 Java 环境，在安装了 Java 运行环境的电脑上都可以正常使用。本次实验我们主要学习 Logisim 的一些基本操作。



实验目的

能够自行搭建 Logisim 实验环境

熟悉 Logisim 的各种基础器件和基本操作

能够使用 Logisim 搭建组合逻辑电路并进行仿真

能够使用封装子电路并进行电路设计

实验环境

PC 一台：Windows 或 Linux 操作系统/Java 运行环境（jre）

Logisim 仿真工具

vlab.ustc.edu.cn

实验步骤

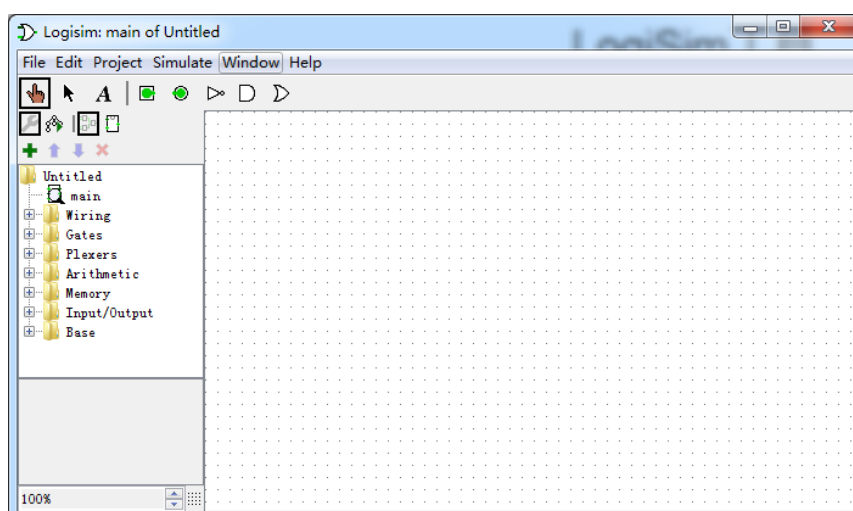
Step1：获取 Logisim 实验环境

在 VLAB 平台下新建一个支持图形界面的 ubuntu 环境，上面已经预装了 Logisim，实验室机房内电脑加载“硬件实验”模板后也可直接获取 Logisim 环境。

如想在自己的电脑上安装 Logisim，可在 vlab.ustc.edu.cn 网站下载与用户操作系统相匹配的 jre 软件和 logisim，下载连接为：













- Java 8 运行环境：[Windows 64 位](#) / [Mac OS X 10.7.3+](#)
- Logisim (v2.7.1, 需要 Java): [Windows](#) / [macOS 10.12+](#) / [Java JAR](#)
- RealVNC 客户端 (v6.19.715): [Windows](#) / [macOS 10.12+](#)
- Vivado Design Suite - HLx Editions 2019.1: [全平台 \(tar.gz 压缩包, 大小 21 GB\)](#)
- Wireshark (v3.0.3): [Windows 64 位](#) / [macOS 10.12+](#)

下载完毕后，首先安装与操作系统对应的 Java 运行环境，然后便可双击 Logisim 可执行文件，启动 Logisim 工具。

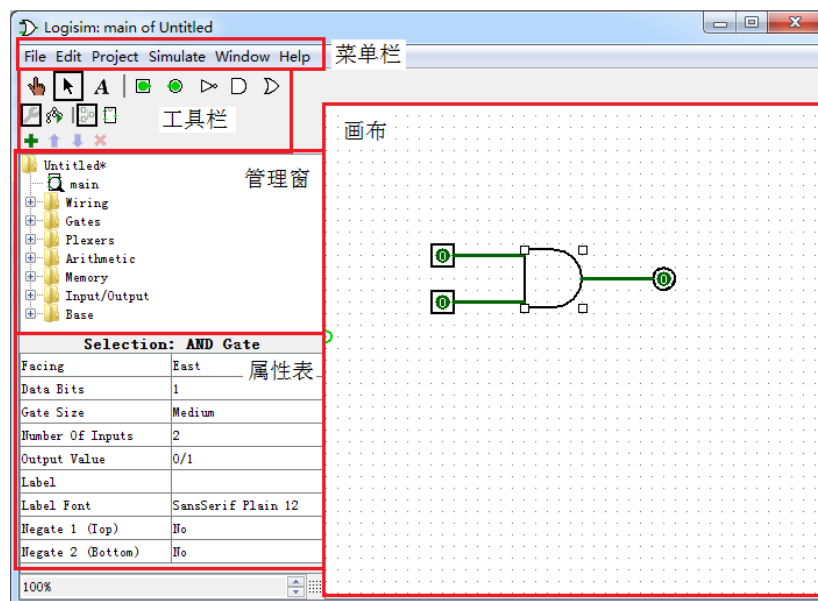


step2: 熟悉 Logisim 界面

Logisim 主界面包括 5 大部分：菜单栏、工具栏、管理窗、属性表、画布。画布区域是用户绘制电路的窗口。管理窗口提供所有的基本组件，以文件夹目录形式显示，其中第一个目录为用户目录，用户所设计的电路都显示在这一级目录下，用户可在该文件夹目录上单击鼠标右键，选择“Add Circuit”添加新的电路。用户设计的电路可以封装成一个模块在同一工程的其它电路中使用。属性表为当前选中组件的基本属性，用户可以根据需要修改其属性参数。菜单栏中除打开关闭文件等基本操作外，其它功能我们可以暂不了解，等需要用的时候再深入研究。

工具栏中有四大类组件：第一类   ；第二类     ；第三类  ；第四类  。第一类中手形工具用于改变电路选定组件的值，箭头工具用于编辑组件或者添加电路，文本工具（字母 A）用于在电路中添加文字描述。第二类是几种常用基本电路组件的快捷方式，这几种组件也可以在管理窗口内找到。第三类用于切换管理窗的显示列表，扳手工具显示工程电路和库文件，

树状结构显示仿真电路的层次结构，一般在时序仿真时用到。第四类用于切换查看电路结构和封装。

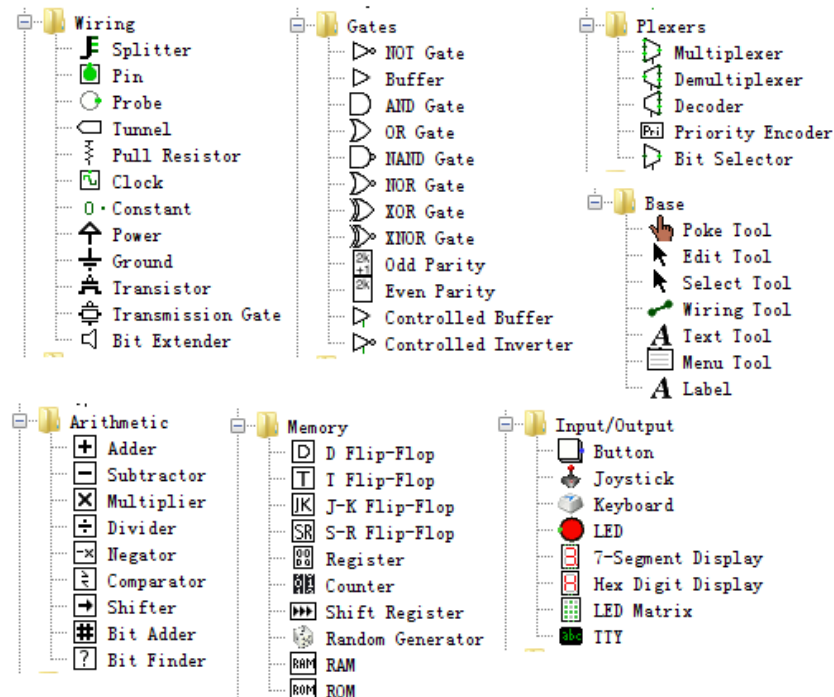
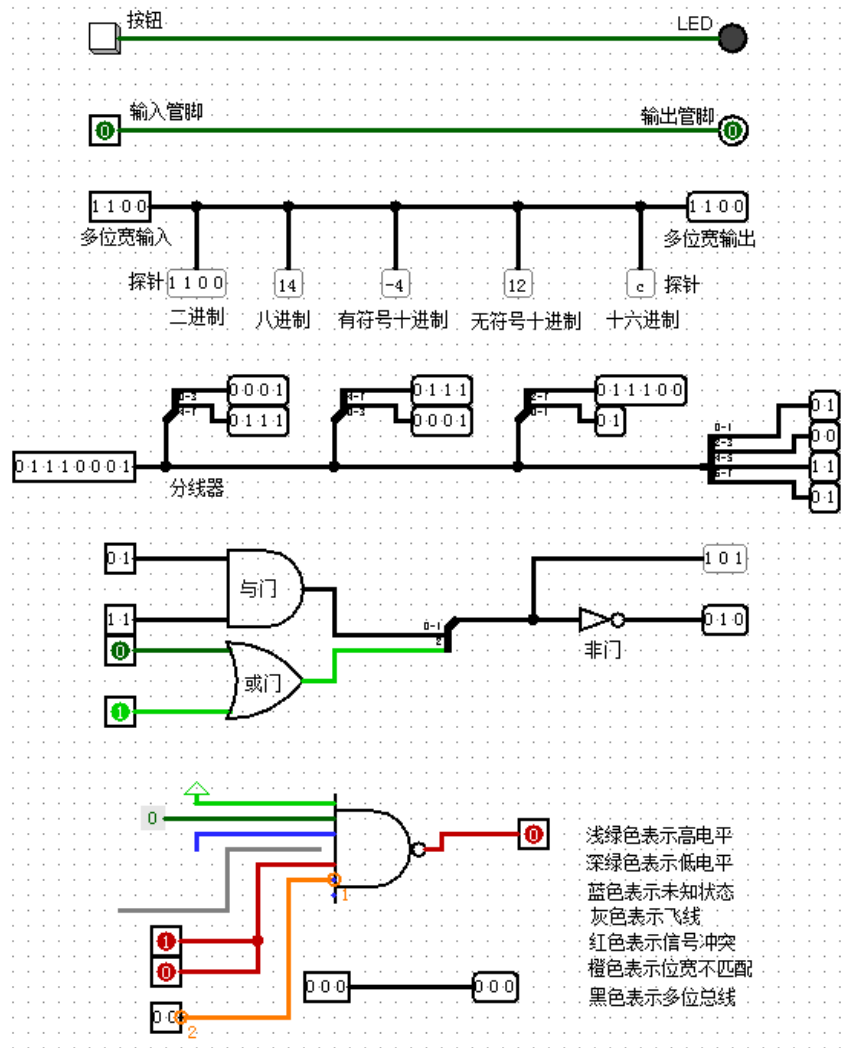


Step3: 熟悉 Logisim 基本操作

打开 Logisim，尝试在画布区域完成如下所示的电路，体会各种组件的使用。

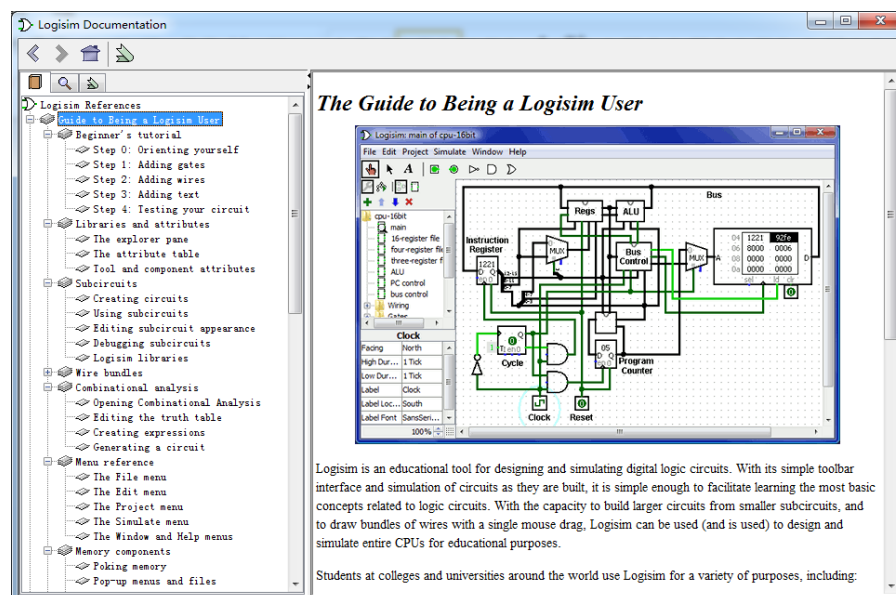
通过练习，用户可以了解到：按钮、LED、输入管脚、输出管脚、多位宽信号、探针、分线器、基本逻辑门等各类组件，以及不同颜色的线缆所代表的含义。

Logisim所支持的组件都可以在管理窗内的各个文件夹里面找到。每个组件都有其对应的参数，用户可根据需要进行修改。



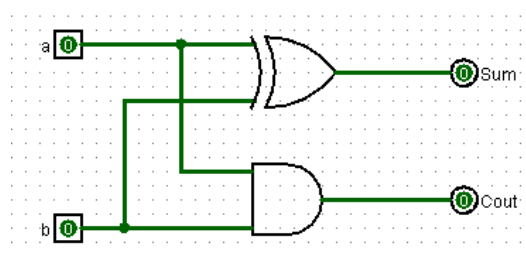
关于各组件的具体功能及使用方法,用户可自行体验,如有问题,可查阅菜单栏“Help”下的“Tutorial”和“User’s Guide”等文档,里面对 Logisim 的使用方法、功能特性以及各组件的使用都有非常详尽的说明。

对于数字电路的初学者来说,可以先学习 Logisim 中组合逻辑相关的组件,时序逻辑相关的组件可在具备了相关的知识储备后进行,可能会相对轻松一些。





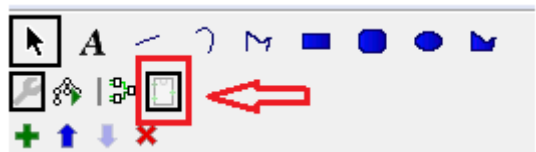
Step4: 模块封装

在 Logisim 软件中,新建一个新的电路命名为“Add”,并绘制电路结构,完成半加器的设计,如下图所示。

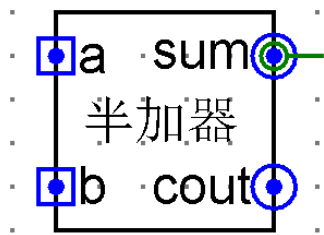


在此电路处于打开状态时（画布区域显示该电路结构，且管理窗

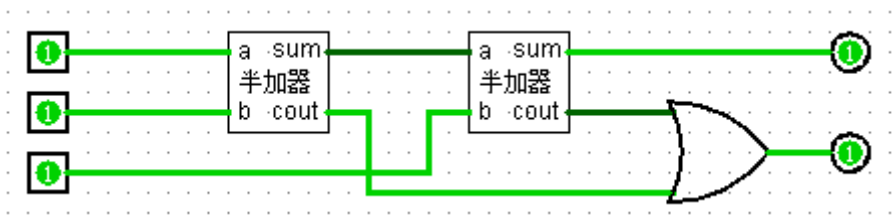
口中该电路图标上有一个放大镜标志： Add ），点击工具栏中的编辑电路封装图标，进入电路封装编辑页面。



修改电路封装样式，并对管脚添加注释。



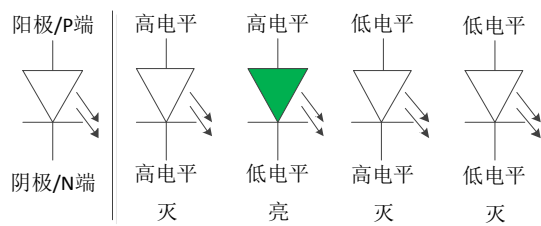
此时，电路封装编辑结束，可在其它电路文件中使用该模块。



注意，在硬件电路中，模块调用可以嵌套（A 调用 B，B 又调用 C），但不能循环（A 调用 B，B 又调用 A）或递归（模块调用自身）调用。

实验练习

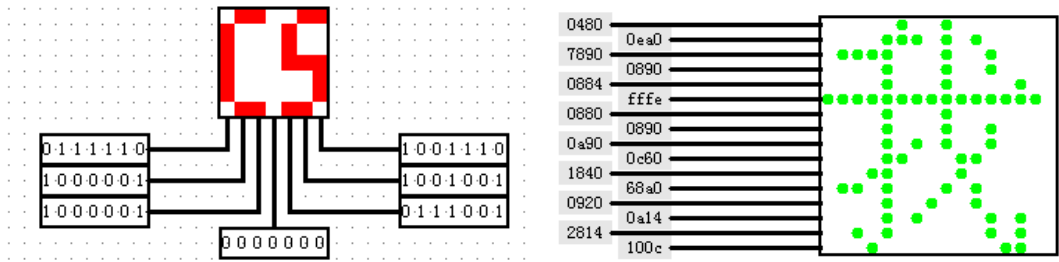
题目 1. 发光二极管（又称 LED）顾名思义是一种会发光的二极管，有 P（阳极）、N（阴极）两个端口，当且只当阳极为高电平，阴极为低电平时才会发光。



LED 点阵是由多个 LED 构成的一个阵列，Logisim 中的 LED 点阵有高

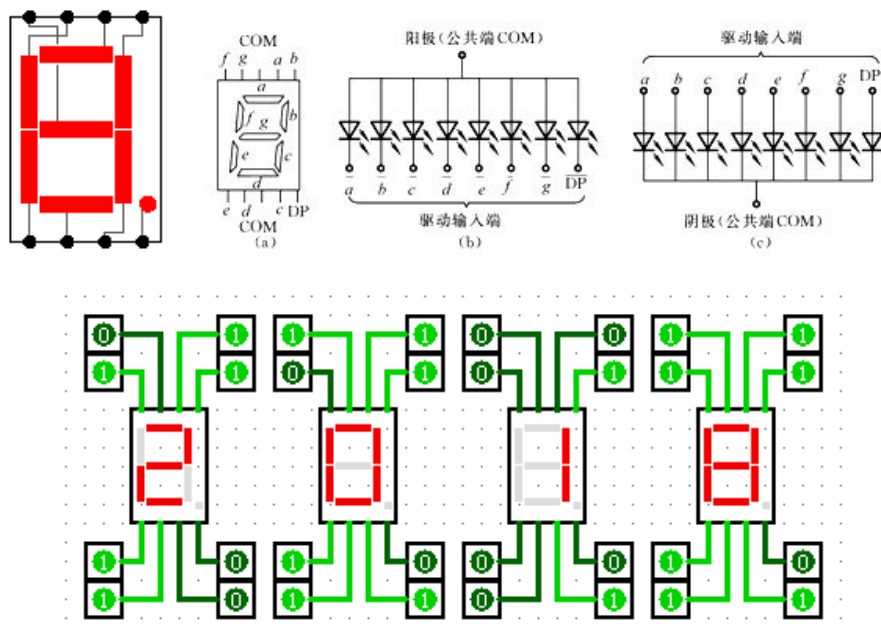
有效和低有效两种（通过组件参数进行设置），以高有效点阵为例，可以认为是把所有 LED 的阴极统一接地，每一个控制位控制一个 LED 的阳极，当控制位为高电平时，对应的 LED 点亮，为低电平时则熄灭。

如下图所示，左侧用一个 7*7 的 LED 点阵通过引脚输入的方式显示出了 CS 的字样，右侧用 16*16 的点阵通过十六进制常量赋值的方式显示了一个汉字，请使用合适分辨率的 LED 点阵显示出自己的姓名。提示：Logisim 中复制电路时，其输入引脚初值会被清空，用常量对 LED 点阵赋值效率会更高。

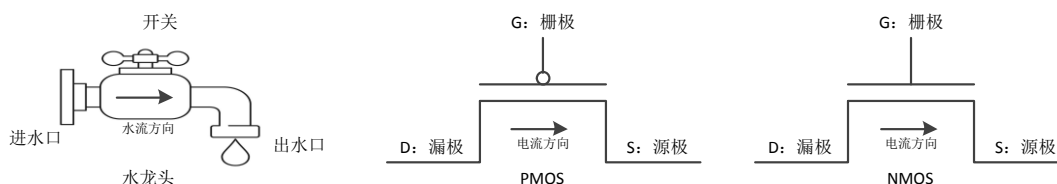


题目 2. 七段数码管实际上是用 7 个发光二极管（LED）拼成的数字形状，再加上右下角的小点，总共需要 8 个 LED，通过控制 LED 的亮灭组合，便能显示出不同的数字（或者字符）。根据其公共端的不同，分为共阳极和共阴极两种，以共阴极为例，8 个 LED 的 N 端（阴极）连接在一起，当阴极为高电平的时候，无论阳极电平高低，所有的 LED 都不亮，当阴极为低电平的时候，阳极为高电平的 LED 点亮，阳极为低电平的 LED 不亮。在 Logisim 工具中，共阳极数码管的公共端默认接了高电平，共阴极数码管的公共端默认接了低电平，用户无法控制，只需控制驱动输入端的信号即可。

如图所示，用 4 个共阴极七段数码管显示出了“2018”的字样，请用若干个共阴极七段数码管显示出自己的学号。



题目 3. MOS 全称为 Metal Oxide Semiconductor，即金属氧化物半导体，顾名思义是一种包含了金属和氧化物的半导体器件，因其行为特性与施加在其上的电场有关，又称 MOS 场效应管，根据其导电载流子的不同，可分为 P 型和 N 型两种，简称为 PMOS 管和 NMOS 管，两种 MOS 管都包含三个端口，分为为 G（栅极）、D（漏极）、S（源极），通过在 G（栅极）上施加电压，便可控制 D（漏极）到 S（源极）的通断。其行为特性可通过与我们常见的水龙头对比加以了解，如下图所示。

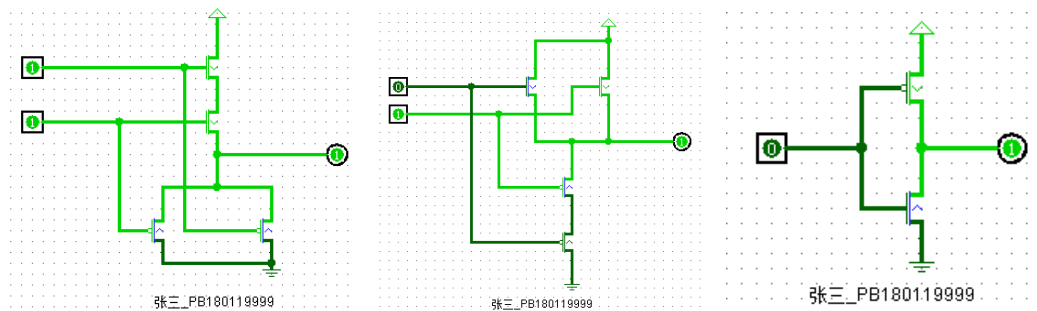


当开关（栅极）打开的时候，水流（电流）便从进水口（漏极）流向出水口（源极），两端处于单向导通状态，也基本不存在水压（电压）差。开关（栅极）关闭的时候，则没有水流（电流），两端处于夹断（断路）状态。导通状态时，水流（电流）只能从进水口（漏极）

流向出水口（源极），而不能反向流动。

PMOS 和 NMOS 在行为特性上的区别则主要表现在栅极有效电平的不同，PMOS 栅极为低电平时导通，高电平时截止，NMOS 则相反，因此在 PMOS 符号的栅极上有一个小圆圈，以表示低电平有效。

如下图所示，是用晶体管搭出来的三个逻辑门，试分析其行为特性，判定各自为哪种逻辑门。有兴趣的同学还可以用同样的方式搭建与非、或非门。请在每张截图下标注本人的姓名及学号。设计电路时请注意场效应管的类型，以及场效应管中箭头的方向（应该是从电源或地指向输出端）



题目 4. 将前面设计的单 bit 与门、或门、非门进行封装，并使用自己搭建的三种基本门电路设计一个 1bit 位宽的二选一选择器，统计各种基本门的数量。如设计一个 2bit 位宽的四选一选择器，三种基本门各需要多少个？

总结与思考

1. 请总结本次实验的收获
2. 请评价本次实验的难易程度
3. 请评价本次实验的任务量
4. 请为本次实验提供改进建议