

中国科学技术大学计算机学院

《数字电路实验》报告



实验题目：Logisim 入门

学生姓名：林宸昊

学生学号：PB20000034

完成日期：2021.10.16

【实验题目】 Logisim 入门

【实验目的】

- 1、 若个人 pc 上进行实验则目的在于能够自行搭建 Logisim 实验环境，若在虚拟机上进行实验则目的在于能够创建，登录并熟练操作虚拟机（虚拟机已预配置 Logisim 实验环境）；
- 2、 熟悉 Logisim 的各种基础器件（包括 led 点阵与晶体管）与基本操作；
- 3、 能够使用 Logisim 搭建基本电路以及简单的组合逻辑电路并进行仿真；
- 4、 能够对电路进行封装并利用封装电路设计数据选择器

【实验环境】

EDA 工具：Logisim 仿真工具

实验设备：个人 PC 一台（Windows 操作系统）

实验平台：vlab.ustc.edu.cn 虚拟机 vm3377

实验地点：电三楼 406 机房

【实验过程】

Step 1: 登录虚拟机

Step1.1: 下载 Tigervnc 以便直接登录并便于使用虚拟机

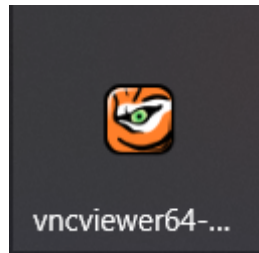
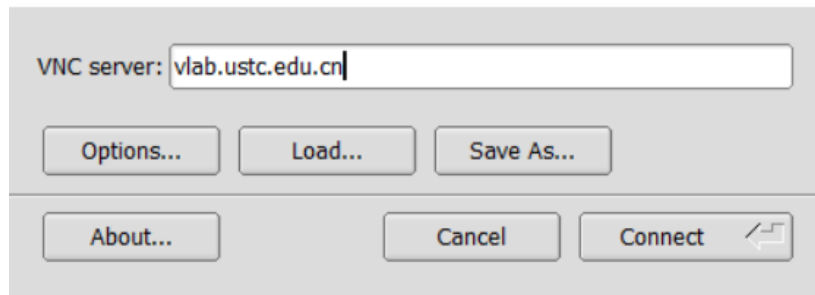


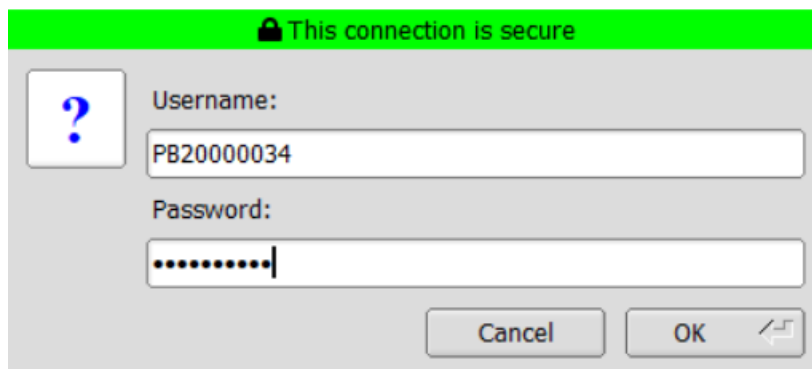
图 1.1:TigerVnc

Step1.2: 连接服务器并输入学号密码登录

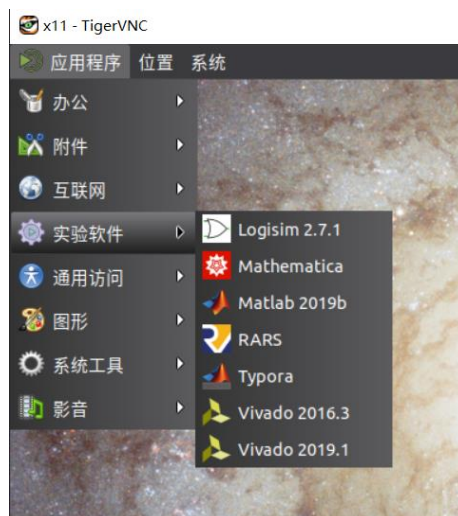
VNC Viewer: Connection Details

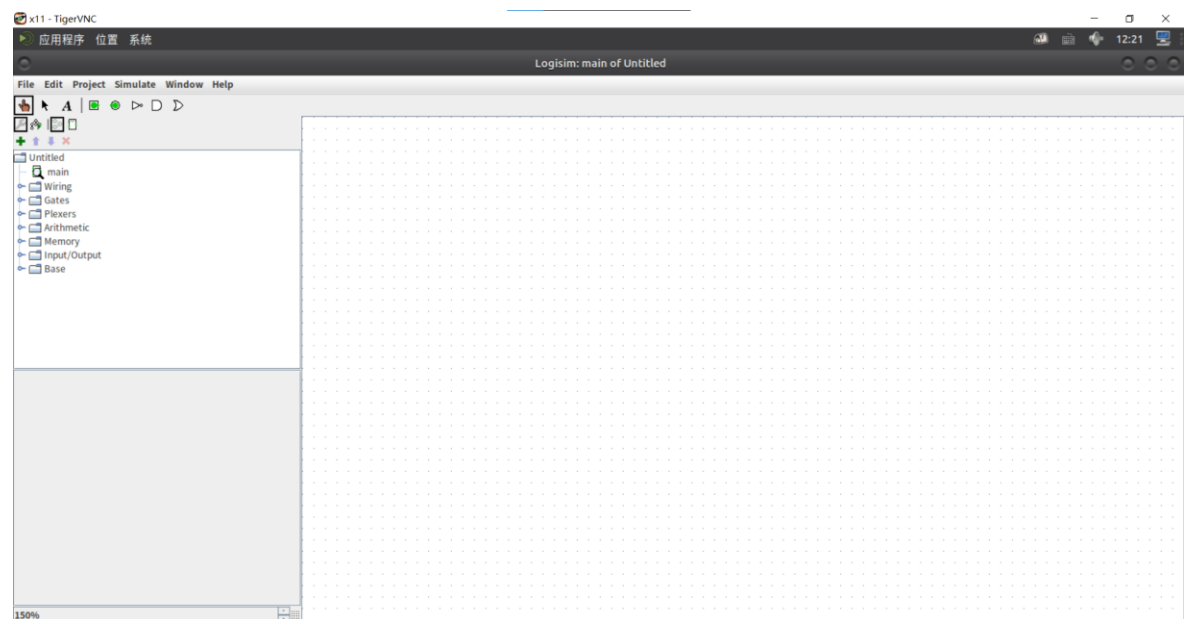


VNC authentication



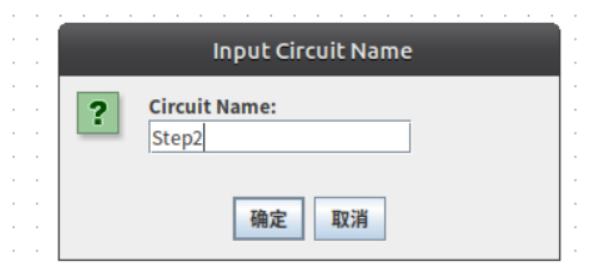
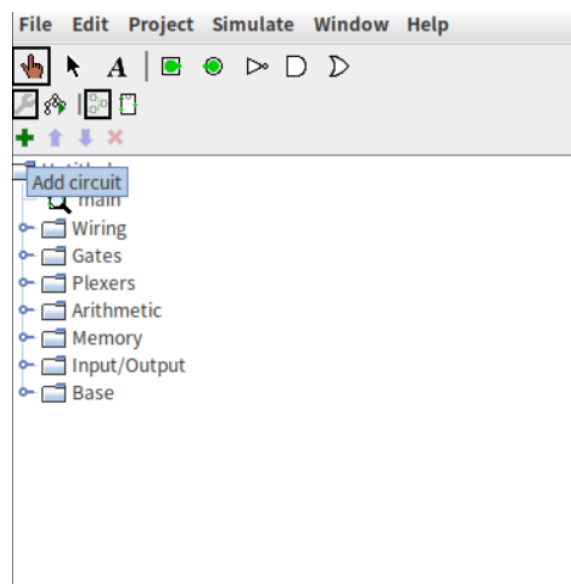
Step1.3: 打开 Logisim 进入实验界面





Step2: 熟悉 Logisim 界面

Step2.1: 添加新电路并命名



Circuit: Step2	
Circuit Name	Step2
Shared Label	
Shared Label Facing	East
Shared Label Font	SansSerif Plain 12

可以通过左侧属性表来更改预设值以及重命名。

Step2.2: 熟悉工具栏工具

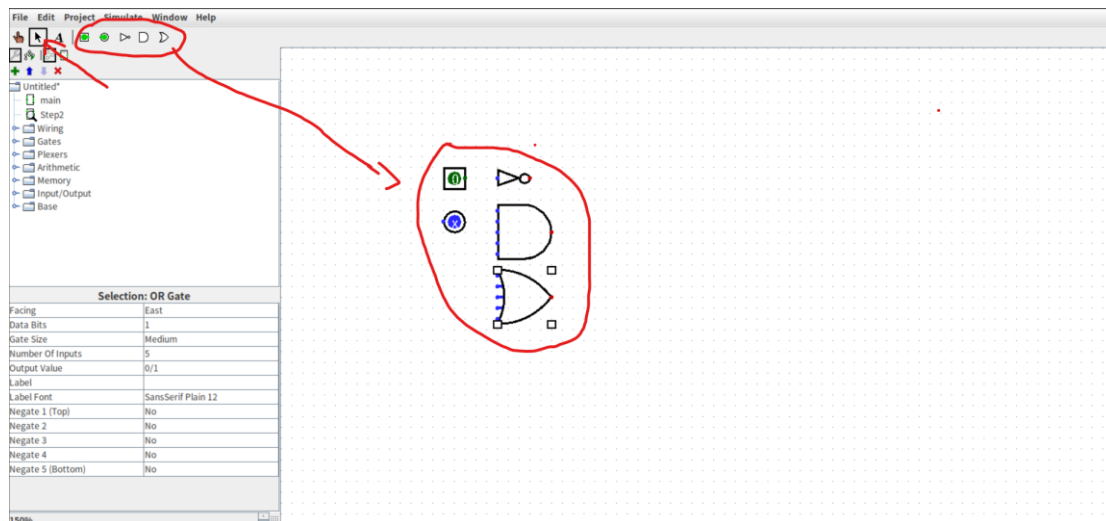


图 2.2.1: 通过箭头点选放置五个基本器件

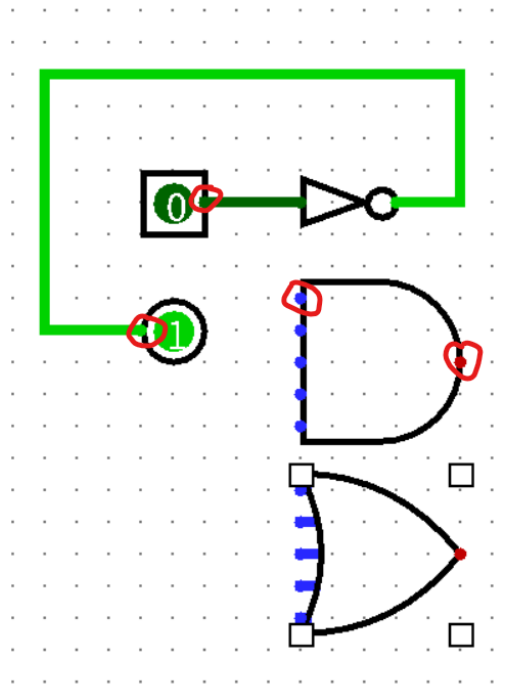


图 2.2.2：用箭头工具按住红色圆圈并进行拖拽以绘制线路

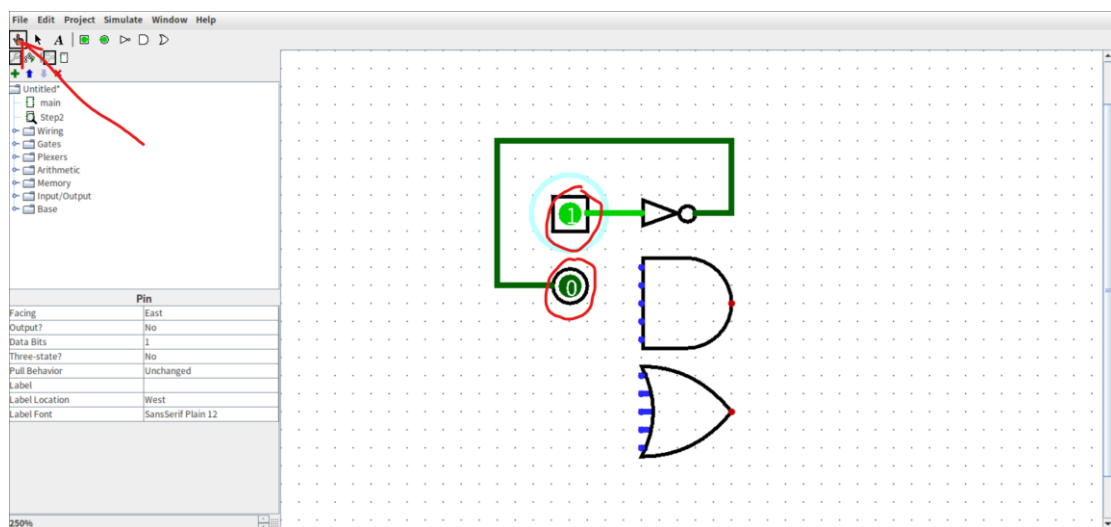


图 2.2.3：通过手形工具点选输入以改变输入值

注意，只有当上方菜单栏 Simulate 下的 Simulate Enabled 选项被打开时才能进行仿真（即此时改变输入输出才会变化）。

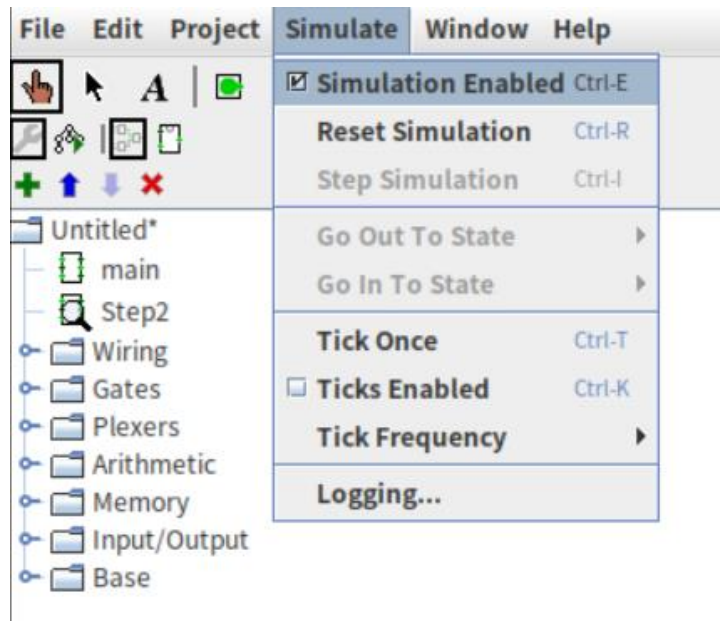


图 2.2.4: 勾选 Simulated Enabled

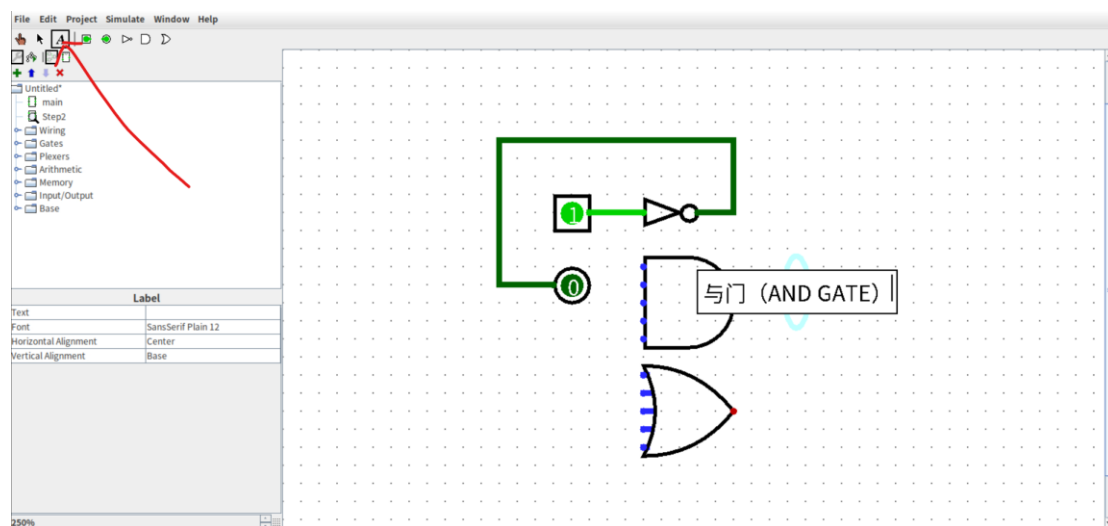


图 2.2.5: 使用文字工具为电路添加文字描述

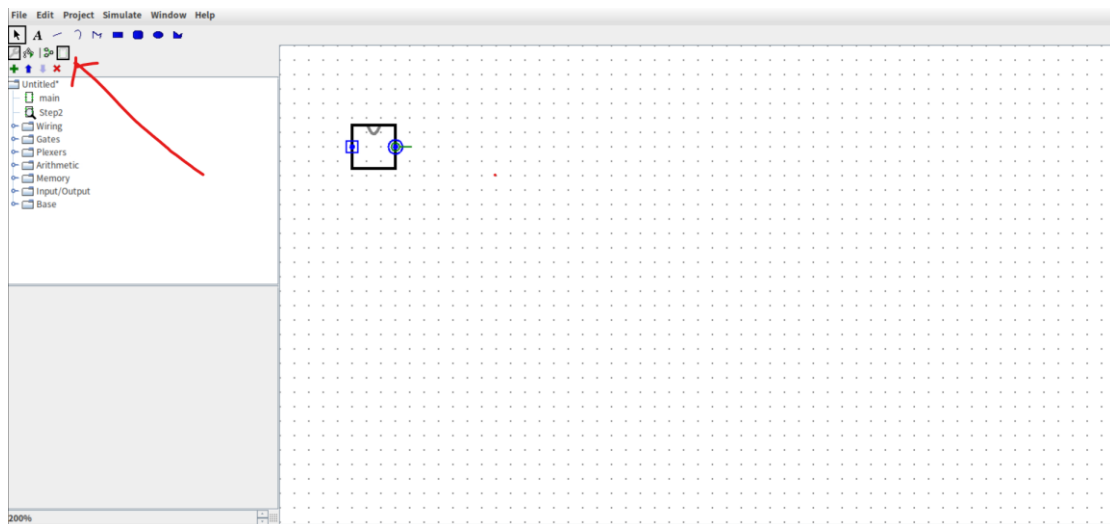


图 2.2.6：对电路封装

Step 3：通过搭建部分基本电路来熟悉 Logisim 基本操作

Step3.1：LED 开关电路

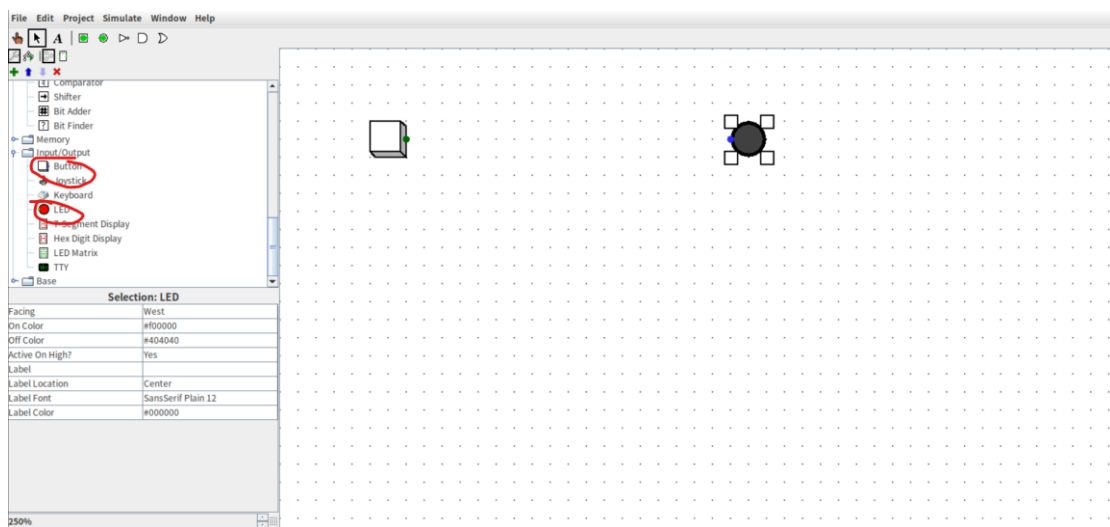


图 3.1.1：在 Input/Output 中选择 Button 和 Led 组件添加到画布上

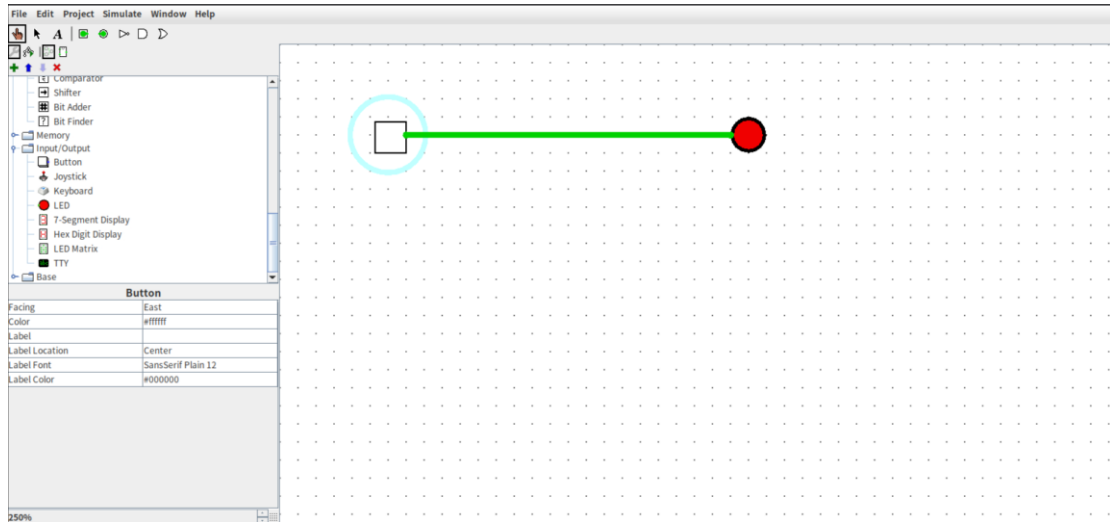


图 3.1.2：使用手形工具按下按钮点亮 Led 灯

Step3.2：管脚输入与输出

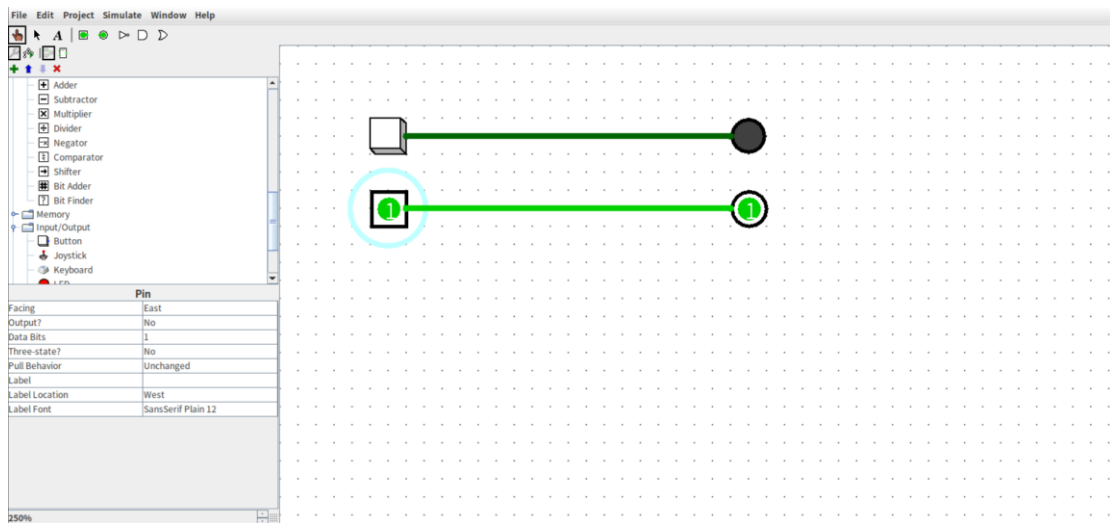


图 3.2.1：输入管脚与输出管脚

Step3.3：探针（获取输入作不同格式输出）

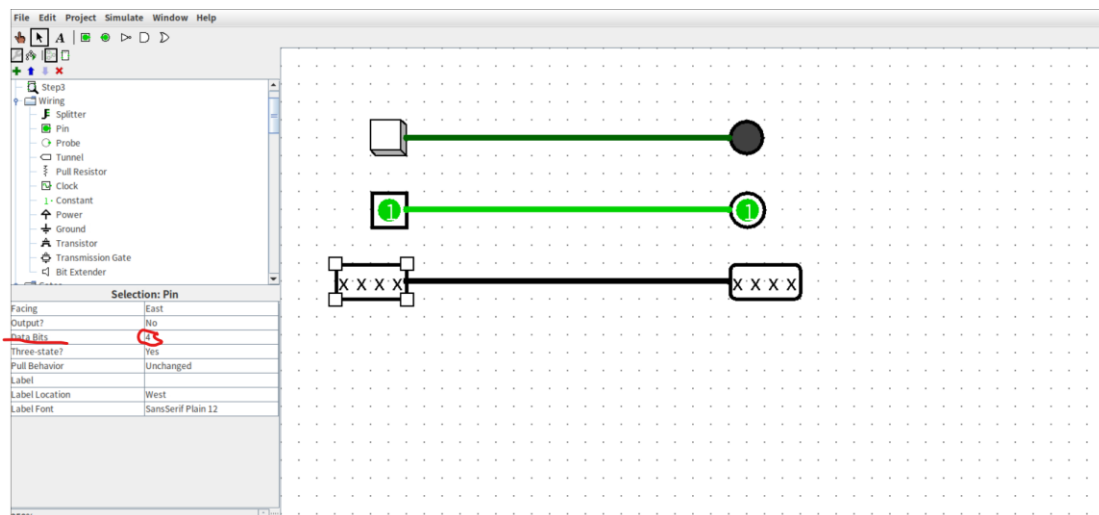


图 3.3.1：将常规管脚输入位宽改为 4 位

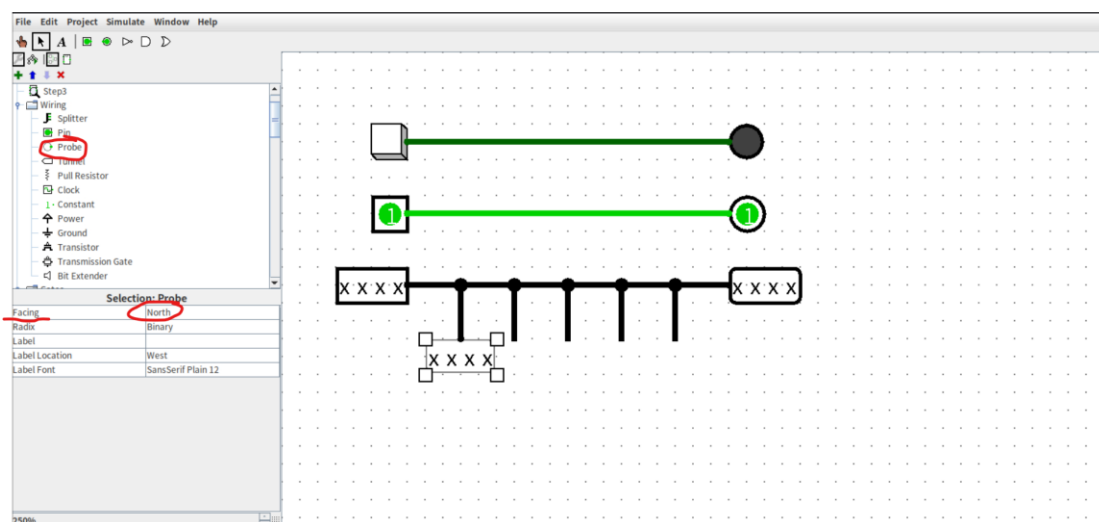


图 3.3.2：添加探针并将更改朝向方便接线

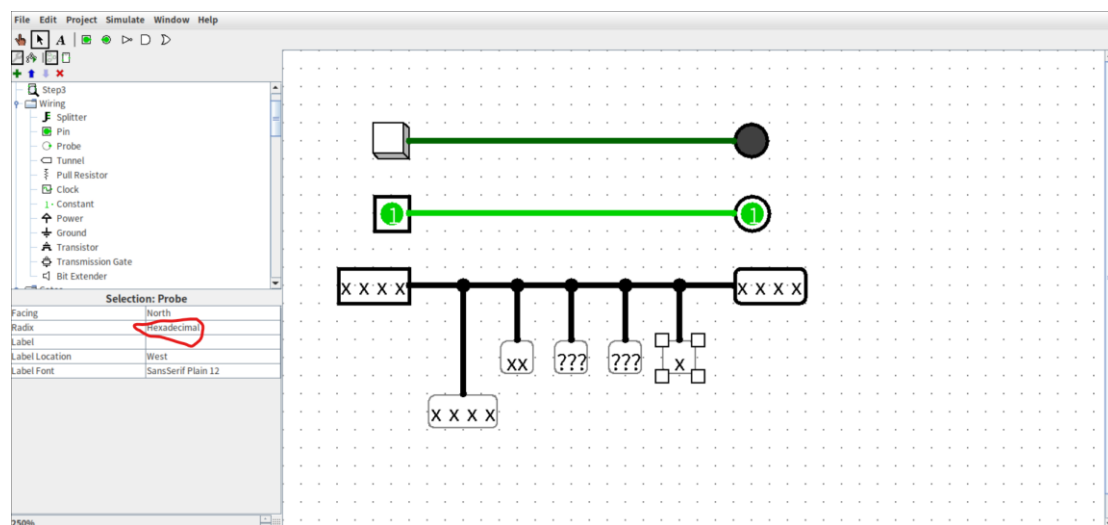


图 3.3.3：更改不同探针的相应进制以获取不同输出

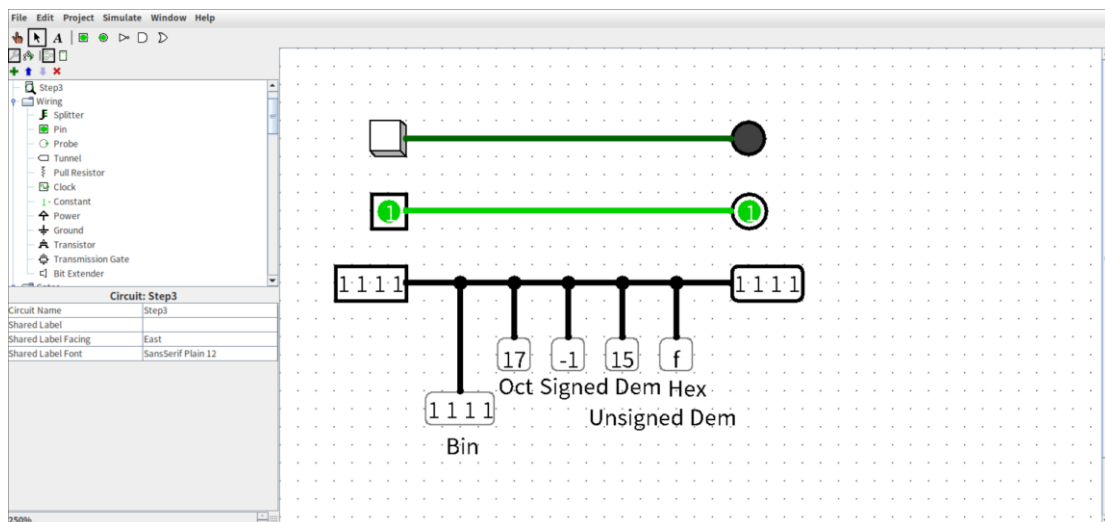


图 3.3.4：最后输入以获得输出

Step3.4：分线器（将多位输入按指定位宽分别输出）

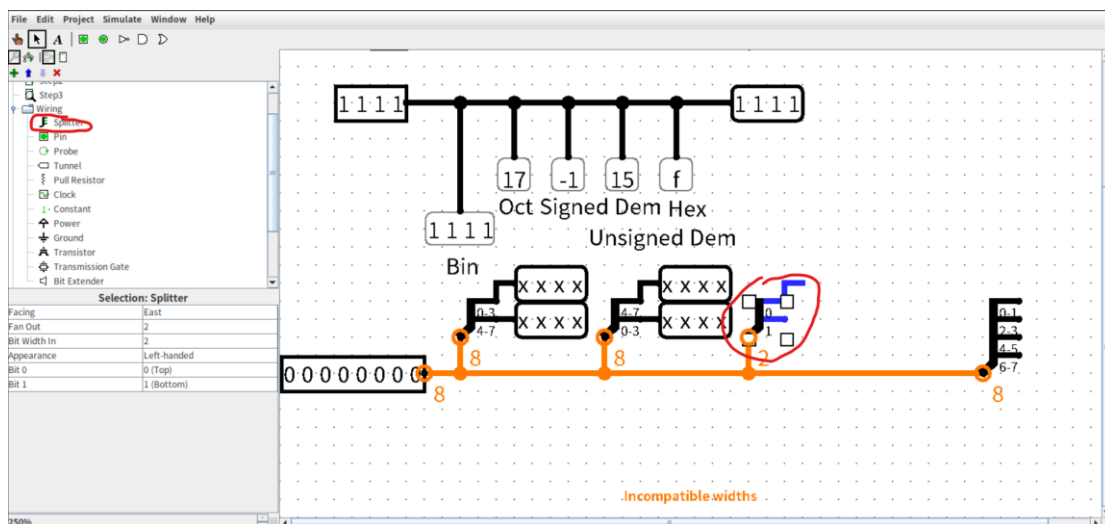


图 3.4.1：添加分线器

Selection: Splitter	
Facing	East
Fan Out	2
Bit Width In	8
Appearance	8
Bit 0	9
Bit 1	10
Bit 2	11
Bit 3	12
Bit 4	13
Bit 5	14
Bit 6	15
Bit 7	1 (Bottom)

Selection: Splitter	
Facing	East
Fan Out	2
Bit Width In	8
Appearance	Left-handed
Bit 0	0 (Top)
Bit 1	0 (Top)
Bit 2	0 (Top)
Bit 3	0 (Top)
Bit 4	None
Bit 5	0 (Top)
Bit 6	1 (Bottom)
Bit 7	1 (Bottom)

图 3.4.2：在左侧属性栏更改分线器属性

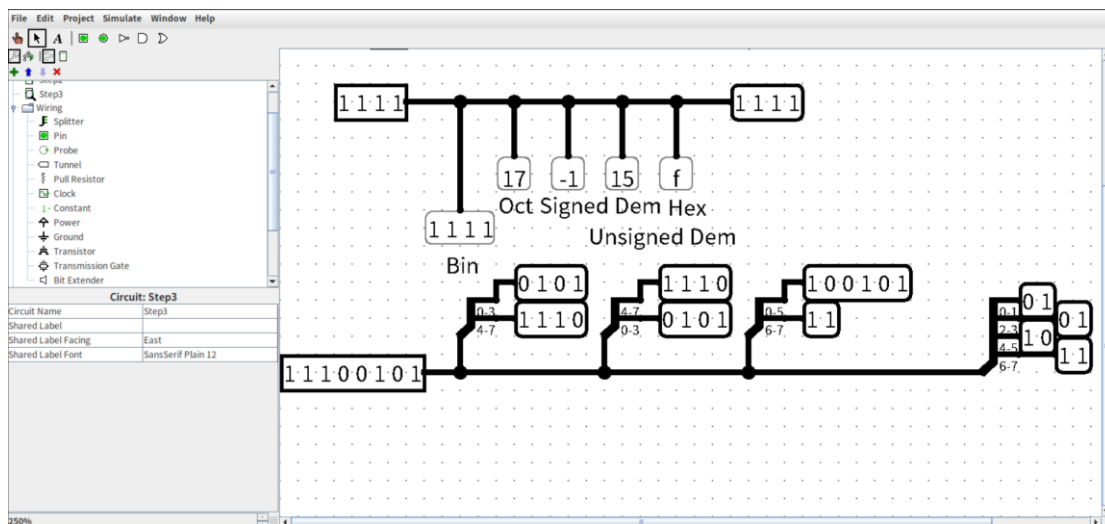


图 3.4.3: 最后输入

Step3.5: 简单组合逻辑电路的搭建

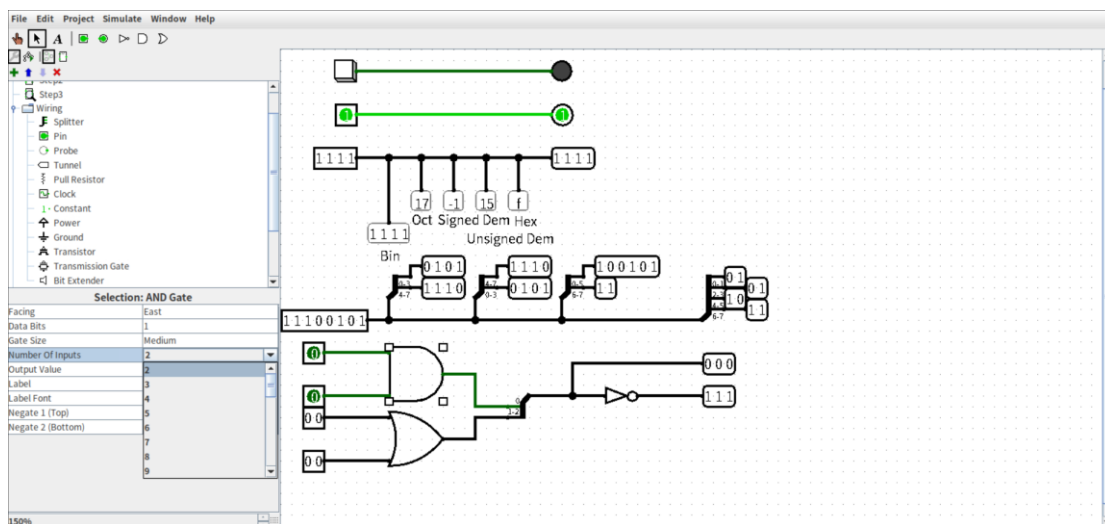


图 3.5.1: 放置与（或、非）门并更改其输入门输及其接受数据位宽

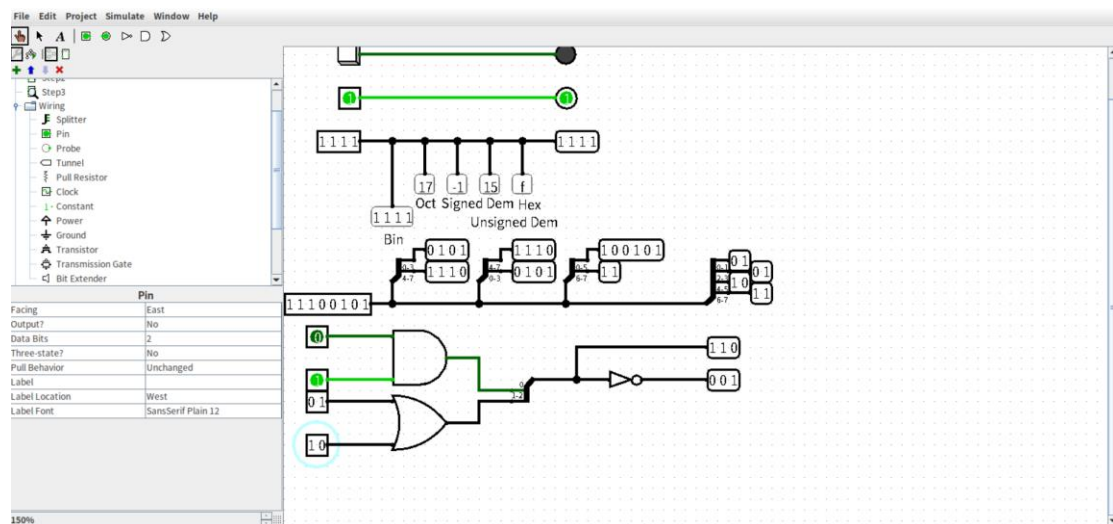
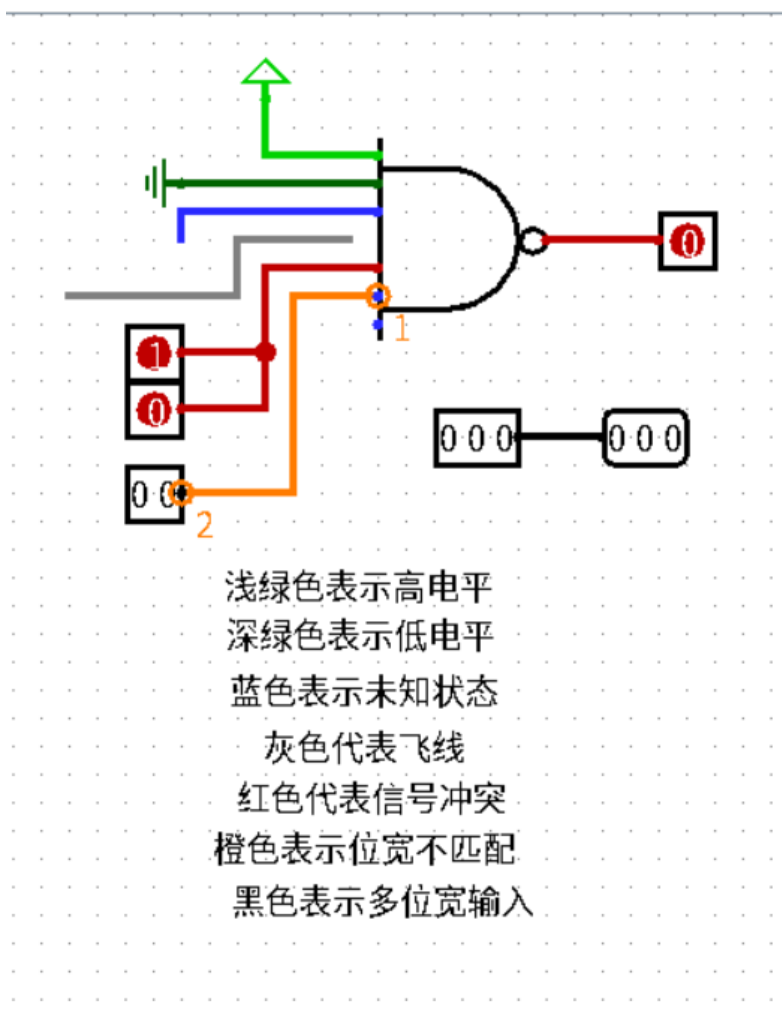


图 3.5.2：最后输入

Step3.6：一些常见线路意义



Step 4: 模块封装

Step4.1: 通过上述基本操作先搭建一个半加器

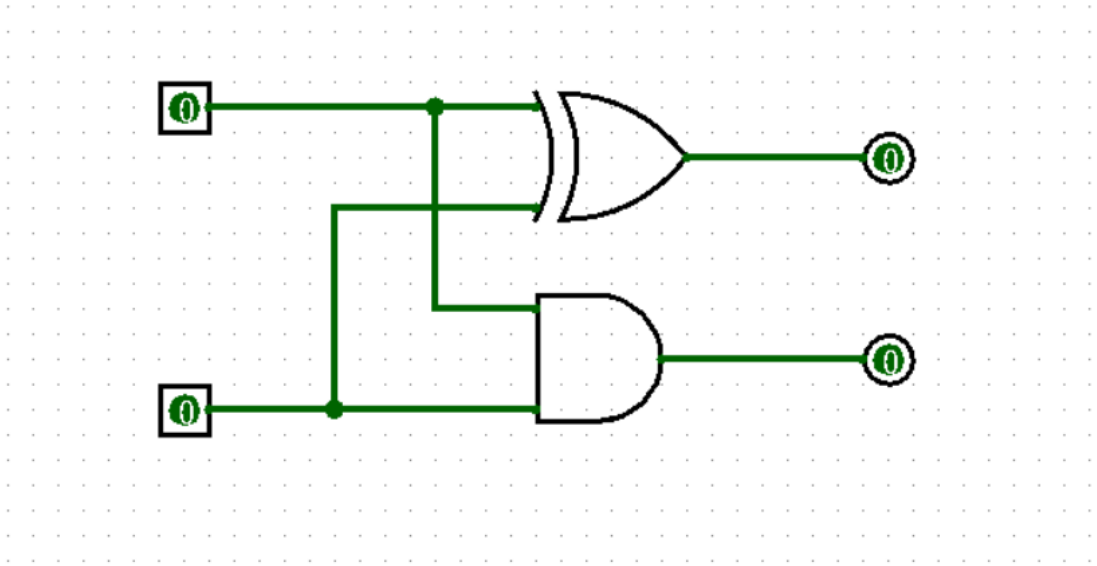


图 4.1.1: 搭建半加器

Step4.2: 进行封装

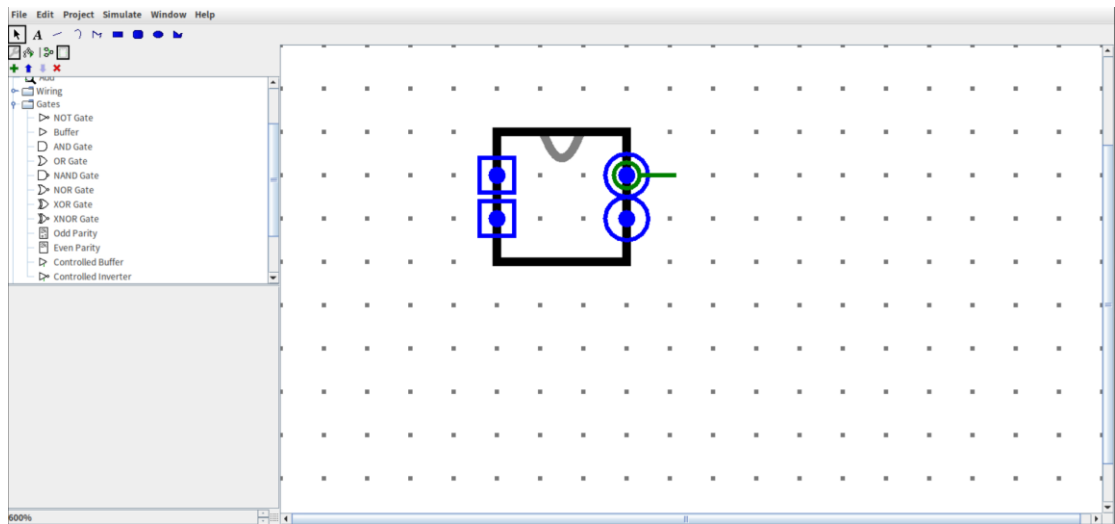


图 4.2.1: 封装

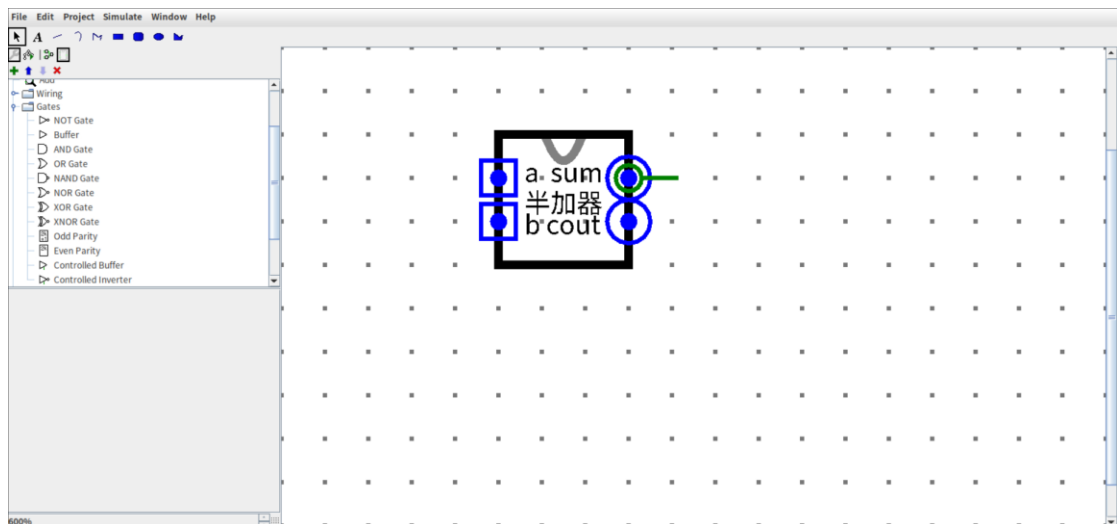


图 4.2.2：添加备注

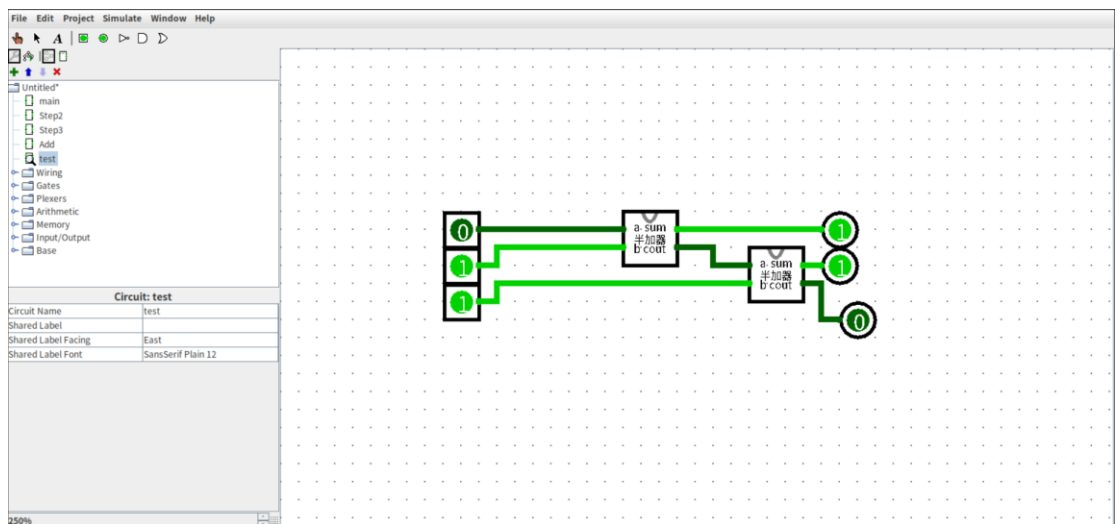


图 4.2.3：在其他电路中使用封装好的电路

【实验练习】

题目 1. LED 点阵显示目标文字

Step1.1: 生成 16&16 高有效 LED 点阵并使用常量作为输入

注意，引入常量输入后要在属性栏中更改其位宽以适配 16*16LED 点阵的输入

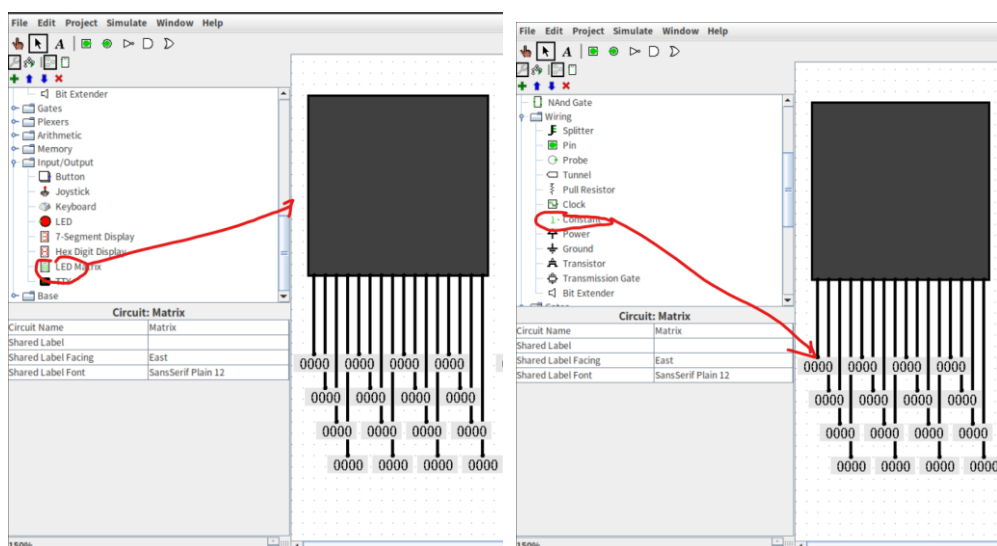


图 1.1.1: 生成点阵及常量输入

Step1.2: 根据显示确定常量的赋值

如序列 0001000100010001, 其代表从上至下第四、八、十二、十六盏 led 灯接受高电平为有效, 灯亮, 其余灯灭, 对应的 16 进制常量则为 1111, 由此可以对应出所有列的常量输入。

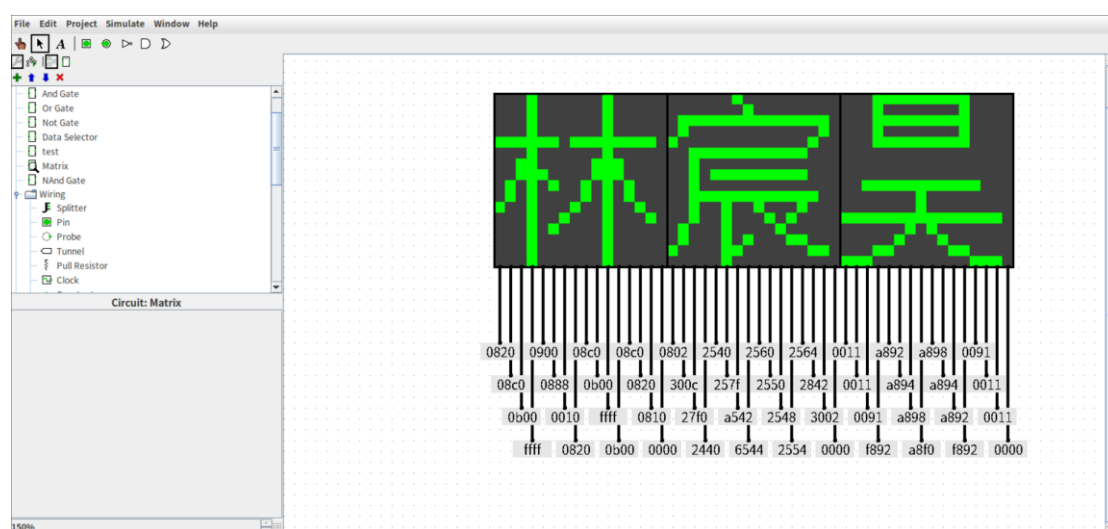


图 1.2.1: 最终显示结果

题目 2. 七段数码管显示学号

Step2.1: 数码管原理 (了解各输入管脚对应控制哪一段数码管)

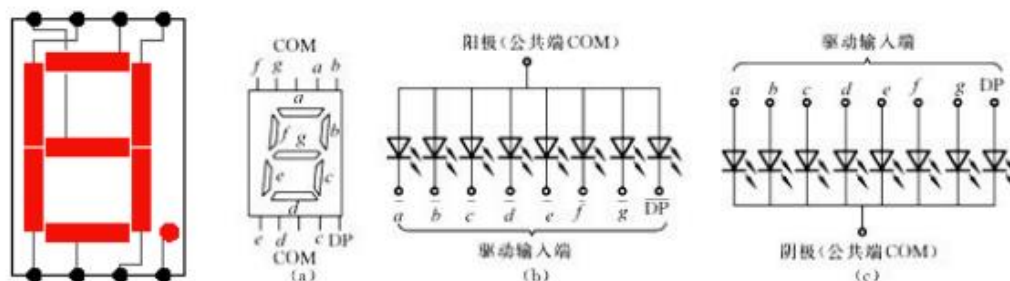


图 2.1：七段数码管原理

Step2.2: 生成七段数码管并引入常量作为输入管脚

注意，若不使用常量，则在复制或下一次打开时其所含值将被清 0，即重置，故使用常量更为保险与有效。

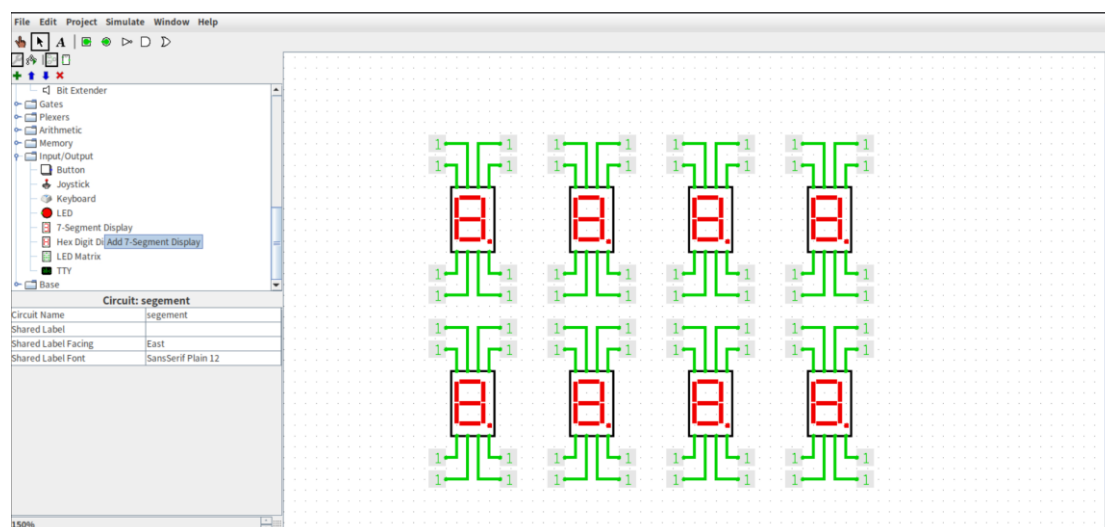


图 2.2.1：七段数码管

Step2.3: 根据学号对常量赋值

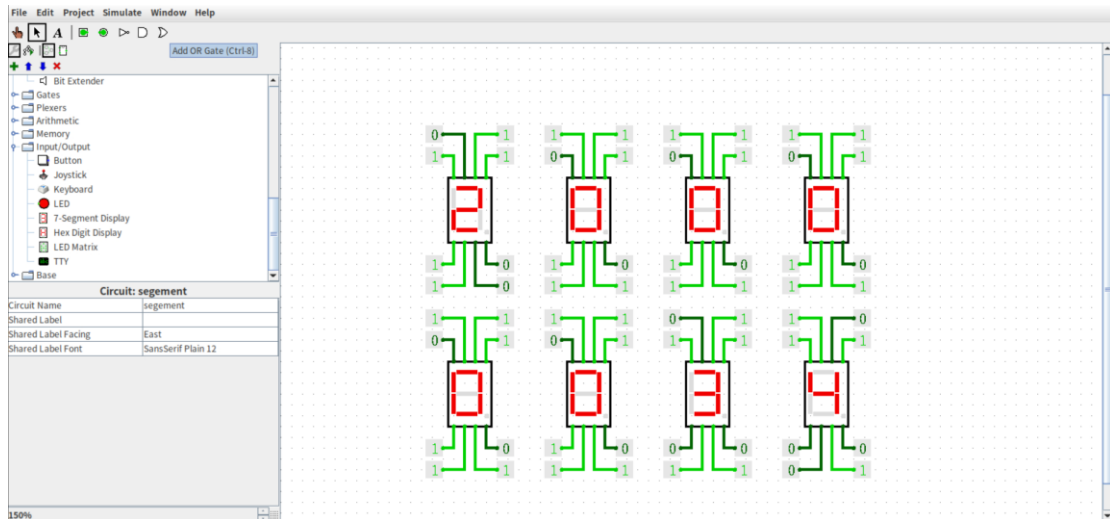
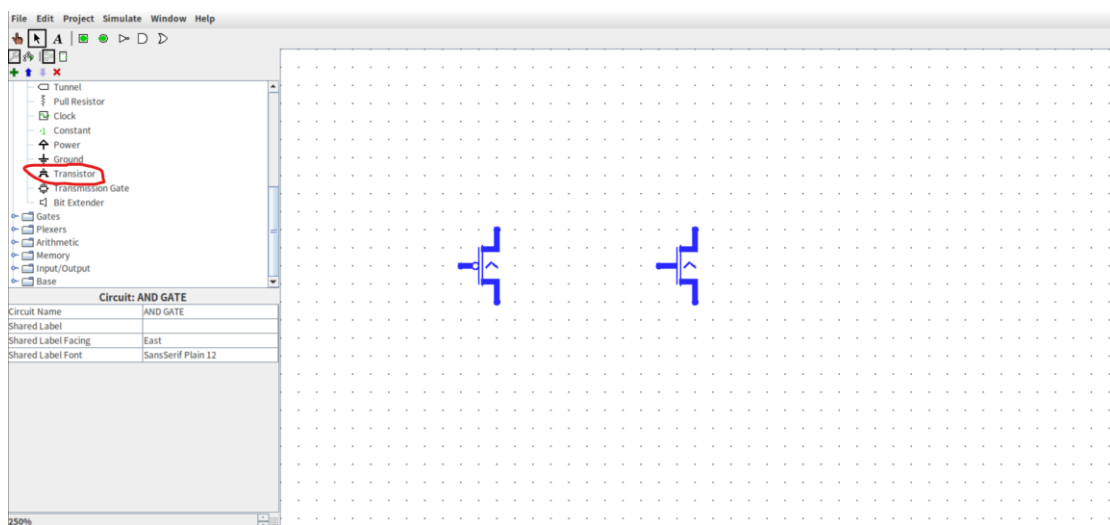


图 2.3.1：数码管显示学号

题目 3. MOS 晶体管搭建三个基本逻辑门

Step3.1：了解 P 型与 N 型晶体管



如图，左边为 PMOS 晶体管，右边则为 NMOS 晶体管。可以发现，PMOS 型有一个代表“非”的小圆圈，即 PMOS 晶体管是低电平有效，而 NMOS 则为高电平有效。

晶体管上下为源极漏极，接地或接电源，左侧为栅极，接输入。以 PMOS 晶体管为例，若左侧输入为低电平，则栅极打开，电流从漏极流向源极。晶体管中的箭头指向

代表所接电源的流向，不可反向流动。在实际搭建中切忌接反。

Step3.2: 搭建三种逻辑门

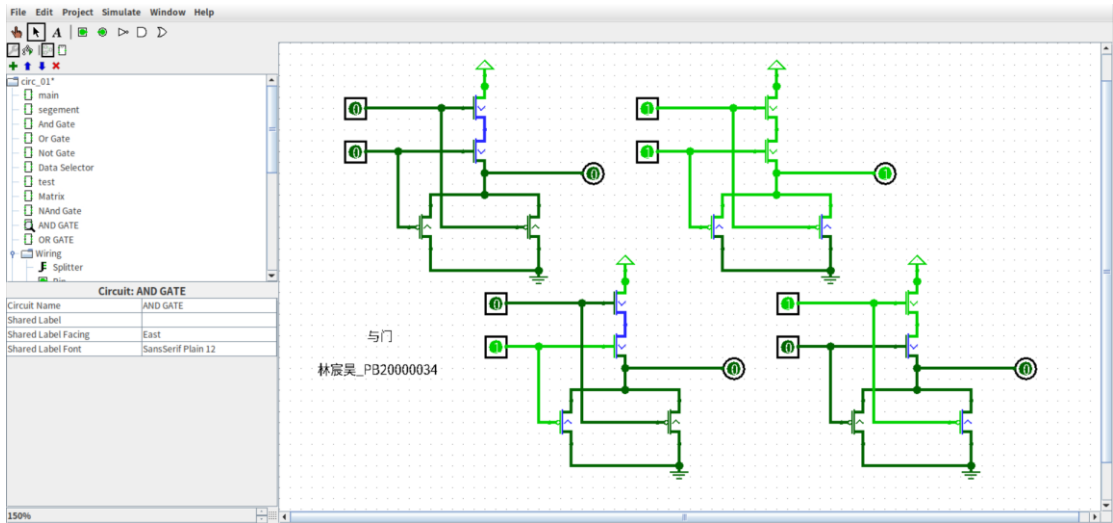


图 3.2.1: 与门电路

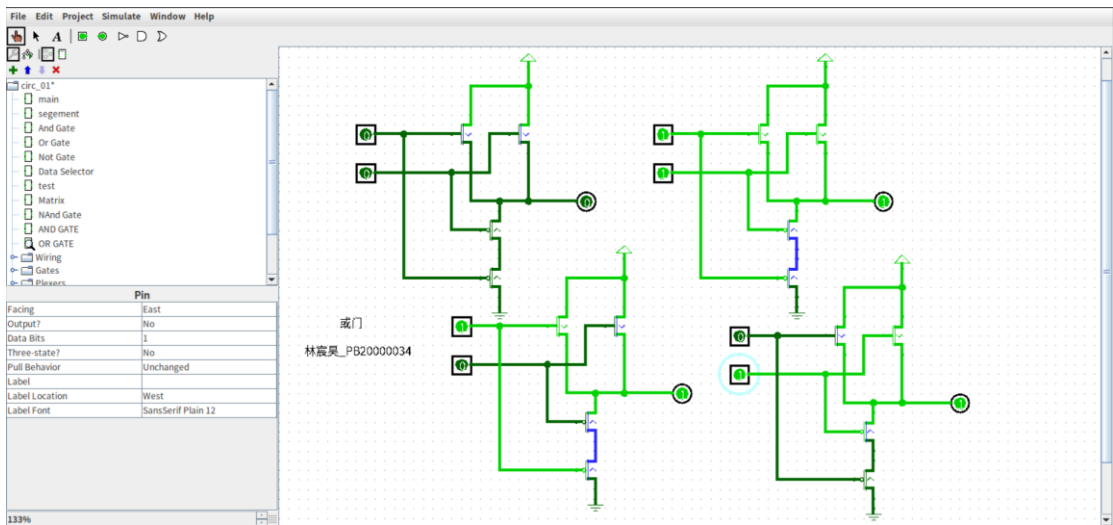


图 3.2.2: 或门电路

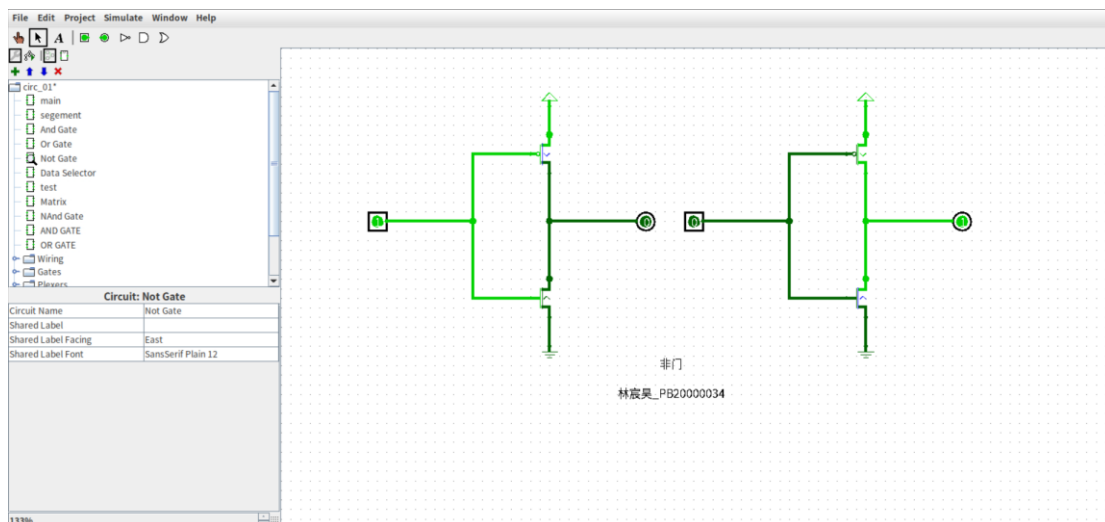


图 3.2.3: 非门电路

EXTRA: 根据 ICS 教科书所说, PMOS 晶体管最好不要接地, 以防止发生短路。故利用相同的 PMOS 与 NMOS, 还有另外的搭建方法, 即使 NMOS 接地。

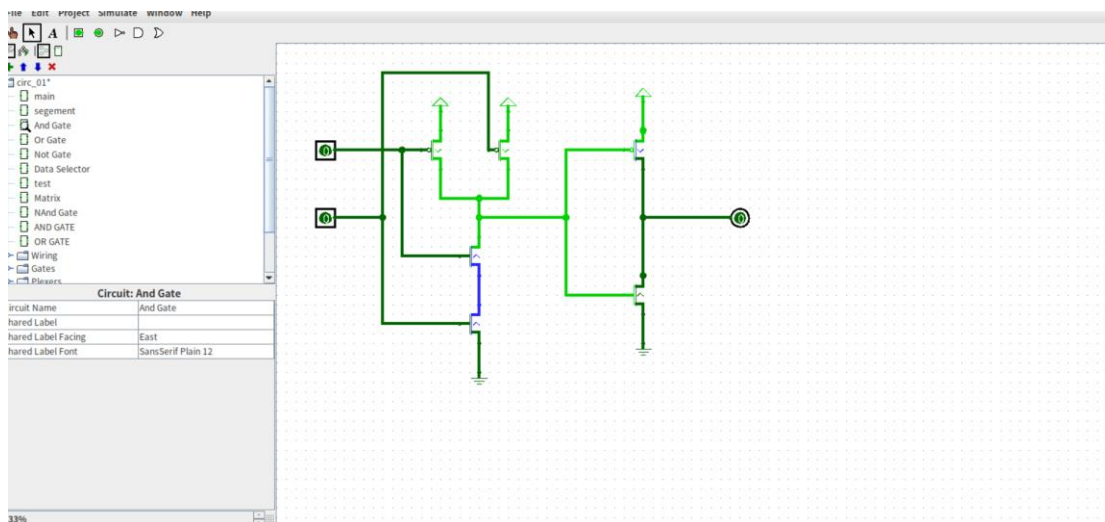


图 ex.1: 与门 (另)

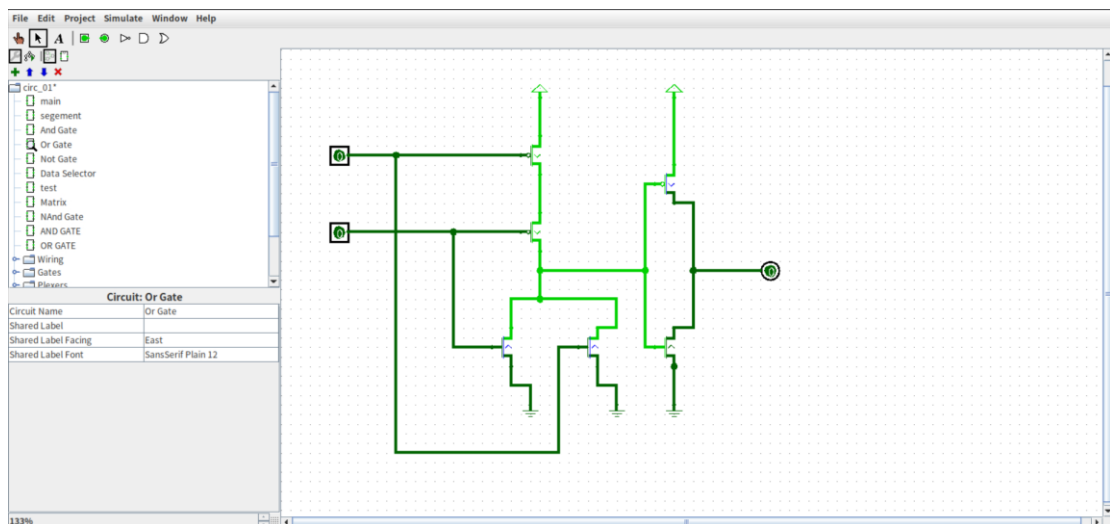
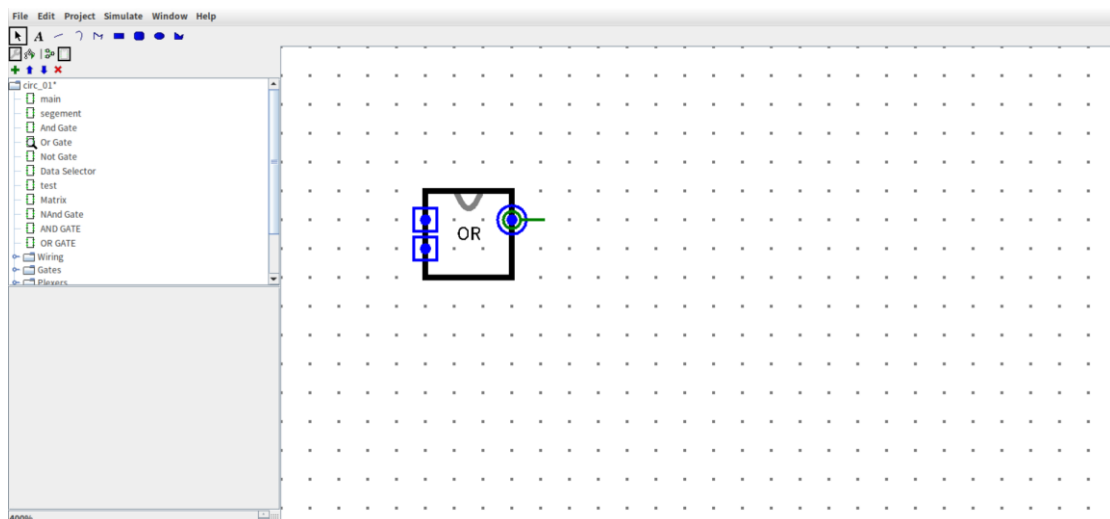
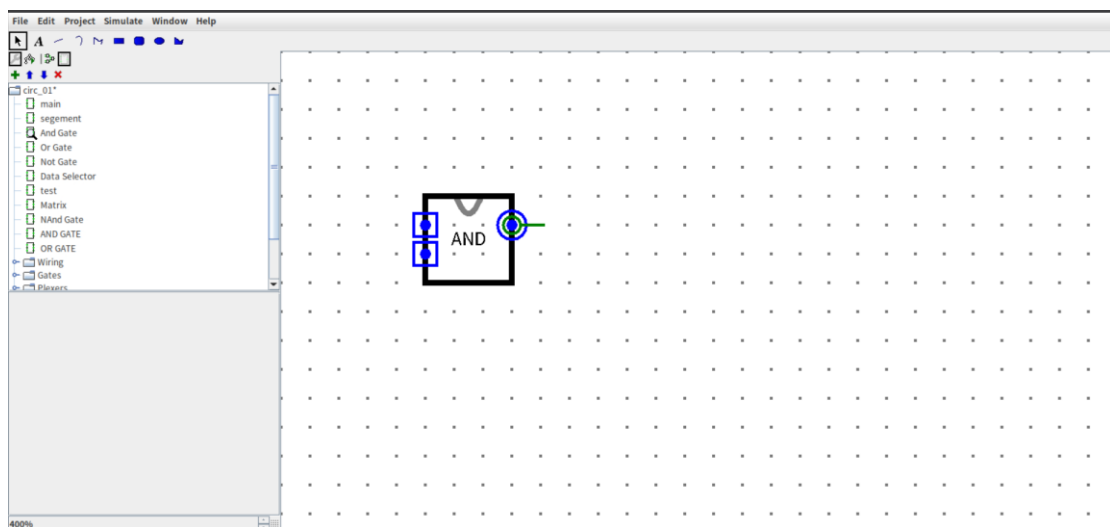


图 ex.2: 或门 (另)

题目 4. 设计选择器

Step4.1: 封装电路



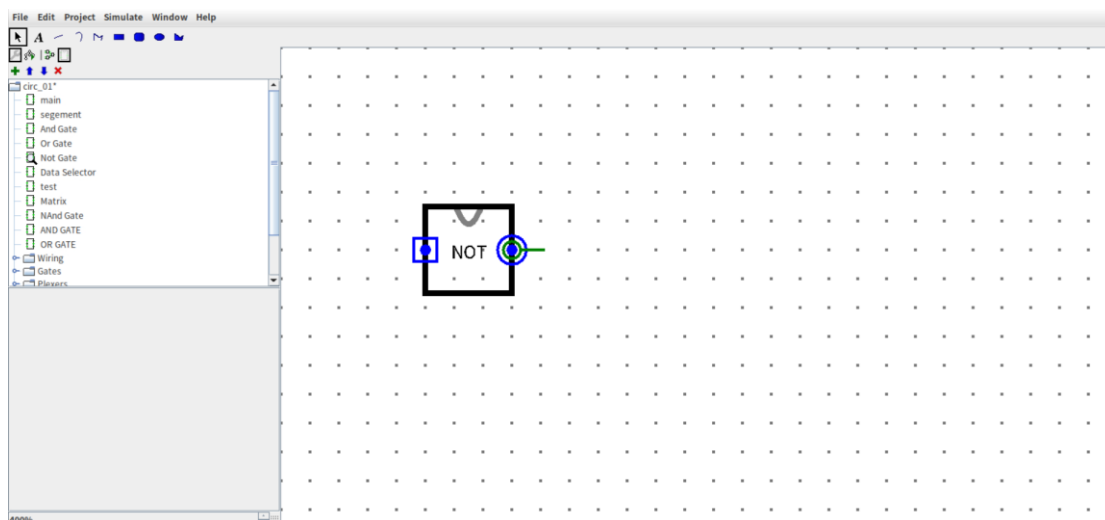
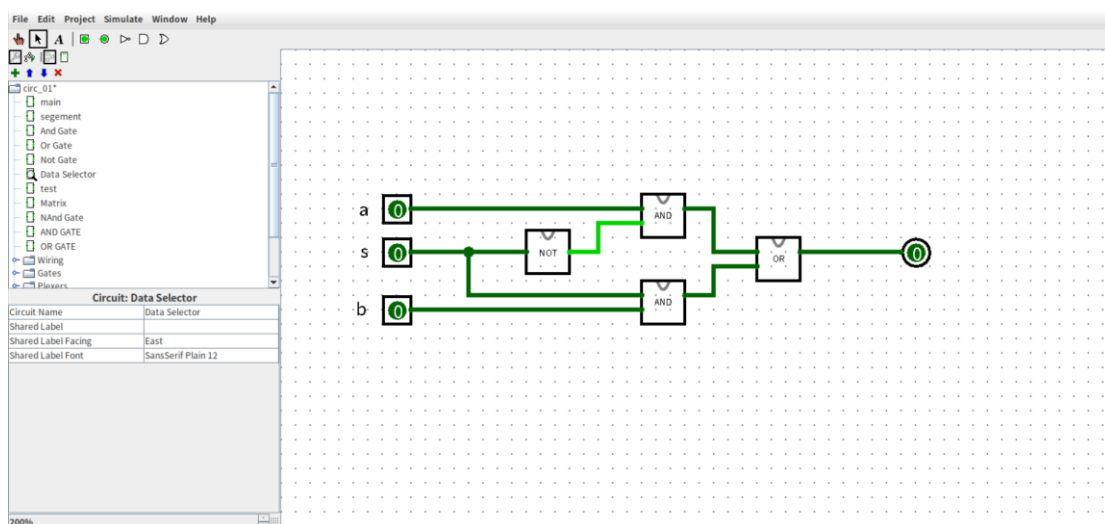


图 4.1.1: 封装后的与或非门电路

Step4.2: 设计 1bit 位宽 2 选 1 数据选择器



如图，a,b 为待选数据，分别享有一个与门用来输入。s 为选择输入，当 s 为 1 时，其与 a 共同连接的与门输出被固定为 0，不再变化，因此此时代表选择的是 b，为 0 时同理。最后通过或门输出，防止另一端与门的输出对最终结果造成影响。

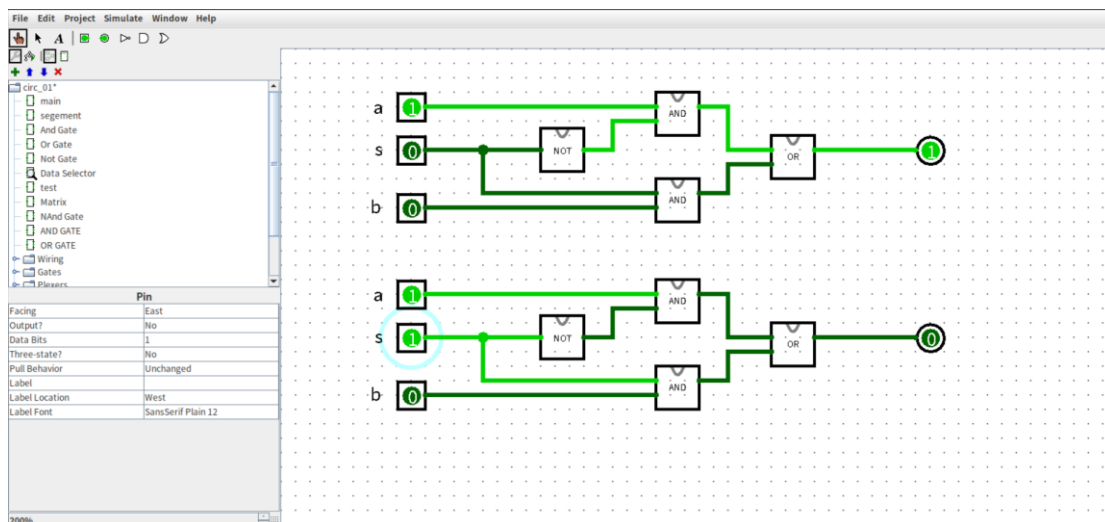


图 4.2.1：选择器工作示意图

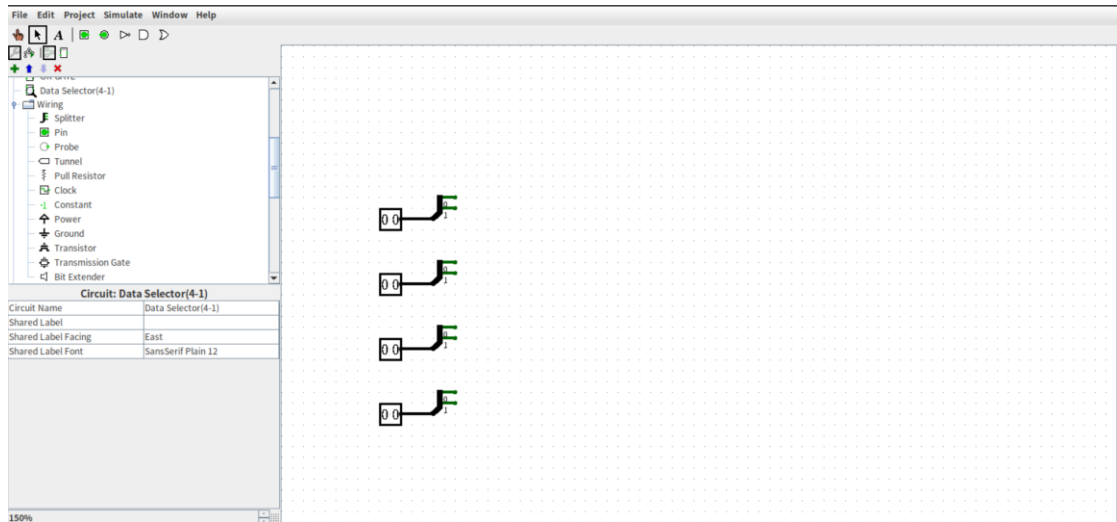
容易得出，最终需要的基本门数量为 4，其中与门 2 个，或门 1 个，非门 1 个。

Step4.3：设计 2bit 位宽 4 选 1 数据选择器

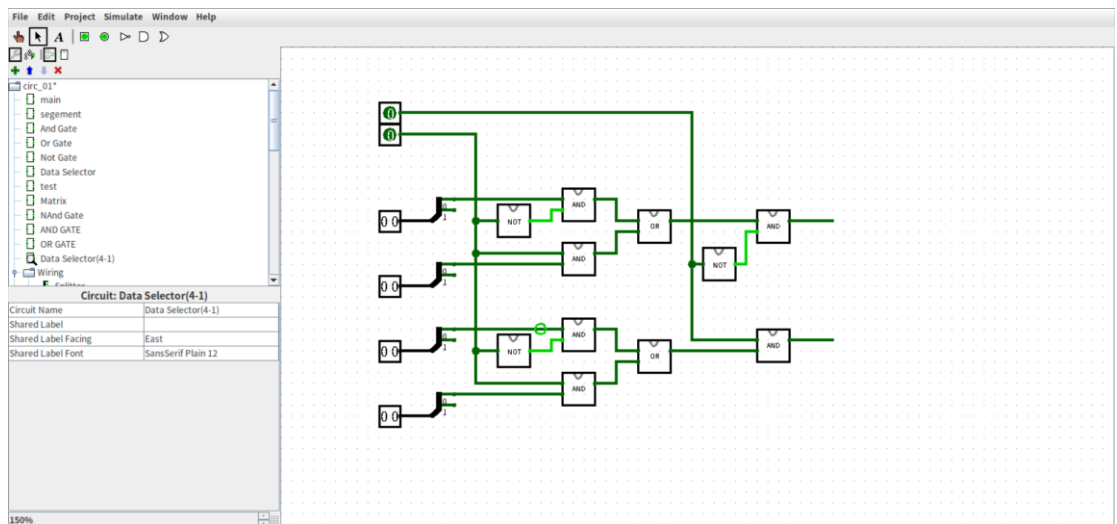
正常情况下，设计 1bit 位宽 4 选 1 数据选择器需要两个 2 选 1 来构成 4 选 1。由此我们可以对 2bit 位宽的数据的不同位分别使用选择器进行选择，最终再综合输出。

如对第一位进行选择，可以使用两个 2 选 1 数据选择器，先选出“2 名候选者”，然后再通过一个数据选择器选出最终“获胜者”。故综合来看，使用 6 个 2 选 1 数据选择器即可，所需的基本门数量为 24，其中与门 12 个，或门 6 个，非门 6 个。具体步骤如下：

EX1：使用分线器获得不同位的输入

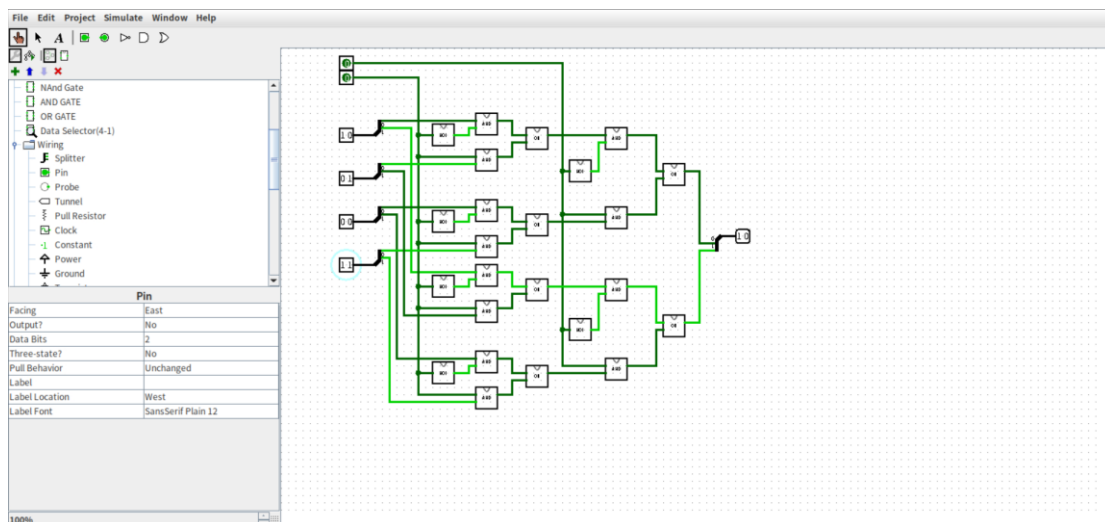


EX2：对每组输入的第一位进行选择



如图所示，目前选择的是第一组数据，则剩余第二位的选择也保持第一位即可。

EX3：对剩余第二位进行选择，同时与第一位的选择方式保持一致



最终得到如图所示的 2bit 位宽 4 选 1 数据选择器。

【总结与思考】

1. 实验收获

其一，收获了一台虚拟机，并且通过 vlab 上的使用文档能够在 windows 上使用 wsl 模拟 linux 系统，收获一定 linux 相关的知识，相信未来一定会很多次用到；

其二，能够较熟练的使用 logisim 的基本工具并搭建一些简单的组合逻辑电路（尤其是 2bit 位宽 4 选 1 数据选择器的设计与搭建对我的操作能力有很大提升）。在一开始我认为输入是固定的只需要通过改变选择输入来获得四种输出，之后发现错误之后才想到使用分线器并通过两次选择来最终得到结果；

其三，在实验报告书写过程中掌握了截图技巧对实验步骤的重新拆解（为了截图）对操作进一步熟练，同时已经把 2bit 位宽数据选择器的设计方法刻进了 DNA，可喜可贺（悲）。

2. 实验难易

总的来说如果不是我脑抽应该评定为中下。当然最多的时间被

我花在在个人 pc 上模拟 linux 系统进行的配置以及对各种各样的报错生无可恋。事实上看着自己画出来具有对称美感（一丢丢）并且整洁的逻辑图时还是很舒畅的（至少相比大物实验舒畅一万倍）。也许任何实验都有这样的成就感。

3. 实验任务量

不多不少。但是实验报告花了我很久很久（也许是第一次写也许是被助教吓到了所以每一步都截了图）。除去这一项任务还是很轻松的。

4. 改进建议

就熟悉基本操作而言，实验步骤第三步的各种线路搭建以及各种颜色线路所带变含义也许更为重要。相比之下，题目 1 只要知道了如何得到点阵并且如何引入常量管脚，剩下的就是与实验无关的重复劳作，而且对不同的人重复劳作的时间还不统一，第 2 题目亦是如此。也许可以放大第三步，作为一个题目，而把原来的第一题第二题作为展示或者说趣味题。

理论 4 学分 72 学时，实验 1 学分 40 学时，模数你坏事做尽