

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：徐亦昶

学生学号：PB20000156

完成日期：2021.10.21

计算机实验教学中心制

2020年09月

## 【实验题目】

Logisim 入门

## 【实验目的】

1. 能够自行搭建 Logisim 实验环境
2. 熟悉 Logisim 的各种基础器件和基本操作
3. 能够使用 Logisim 搭建组合逻辑电路并进行仿真
4. 能够使用封装子电路并进行电路设计

## 【实验环境】

Linux 2.6.32、Logism 2.7.1

## 【实验过程】

由于配置环境和熟悉界面是进行实验的前提条件，这里直接从Step3开始写起。

Step 3:

- 1、常见器件及使用方式

1. wire: 按钮，使用时将其在画布上连接即可，一根导线最多可以手动控制拐一个弯，使用时软件会自动控制其走向。
2. pin: 分为输入和输出接口，对应属性选项卡中的output。可以设置Data Bits来改变端口的位宽，同时应改变facing使其朝向自然。
3. probe: 探针，可以自动检测线上的数据，并以设置好的方法表示出来。
4. splitter: 分线器，Fan Out表示分成几部分，Bit Width表示输入位宽，后面的每个Bit可取0到Fan Out-1个值，表示相应的位被分到那一部分。
5. Gates: 常见逻辑门。
6. Button: 按钮。
7. LED/LED Matrix: LED灯或点阵。
8. Constant: 用于给输入端口赋常量值，程序重新打开时不会变。
9. Power/Ground: 高电平或接地。
10. Poke Tool: 给端口赋值。

## 2、实验结果

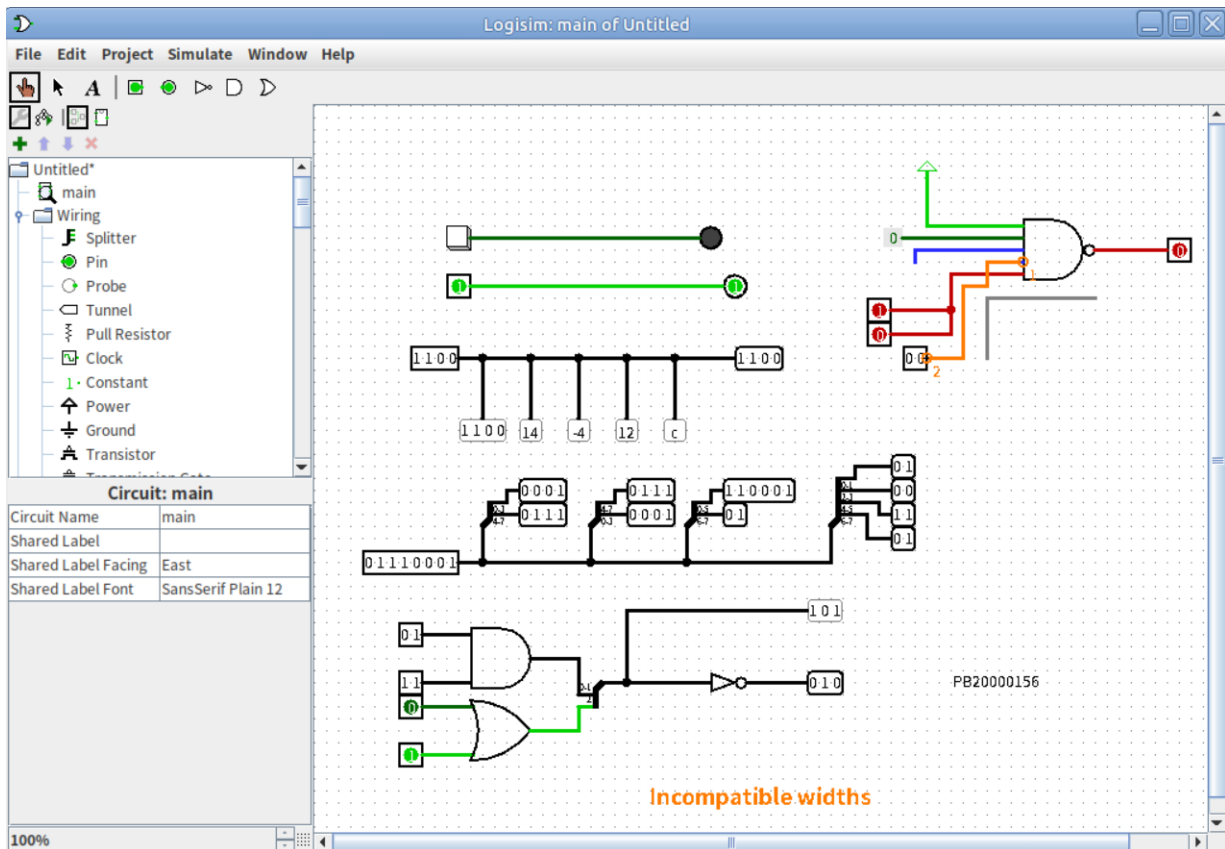


图 1: 界面截图

Step 4:

按照实验文档逐步搭建电路，每步截图如下：

### 【实验练习】

#### 第一题 LED点阵

- 1、在左侧的组件中找到LED Matrix,调整行数和列数分别为16，使用16个Constant给每个引脚赋值，将Constant组件的Data Bits设为16，

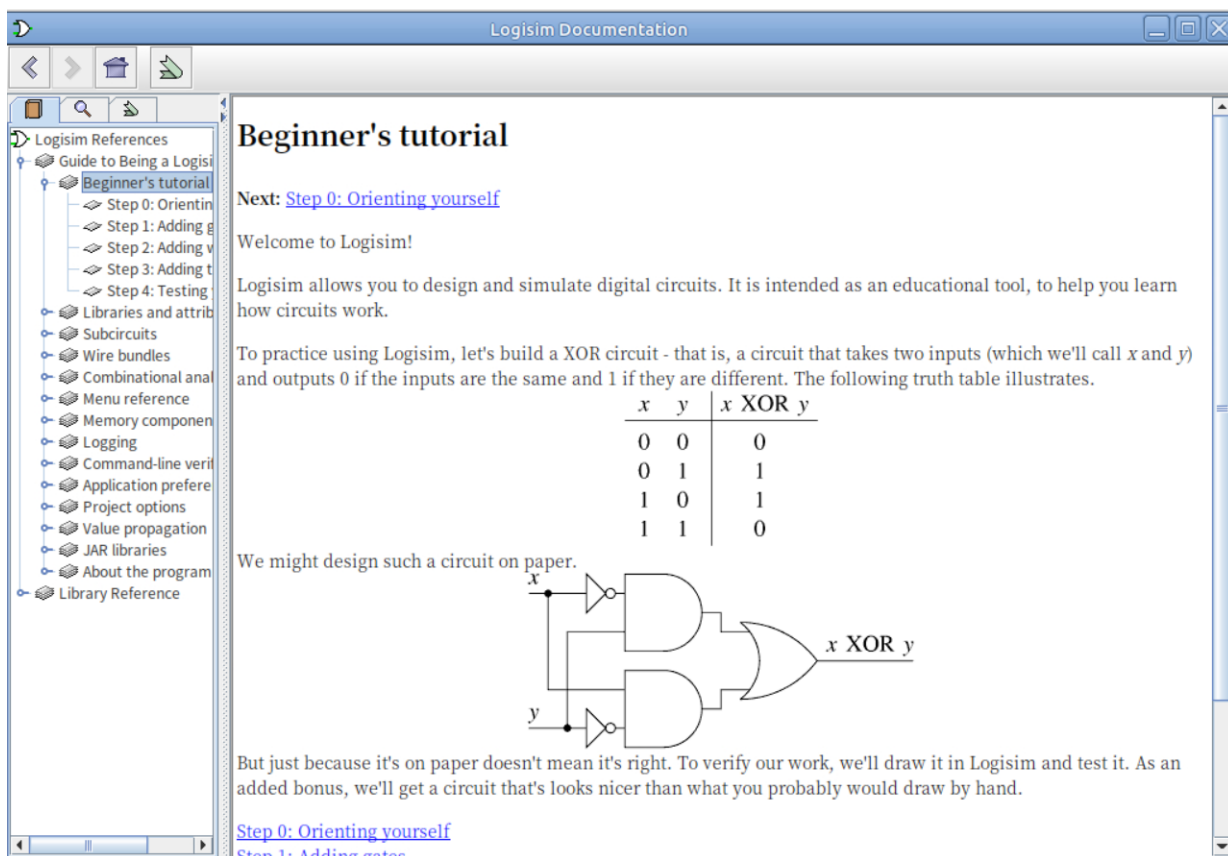


图 2: 使用文档

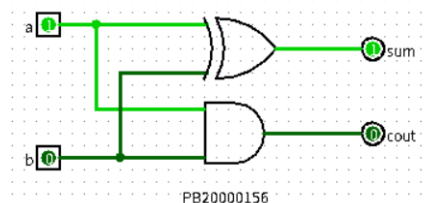


图 3: Step4 设计半加器

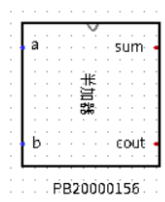


图 4: Step4 电路封装

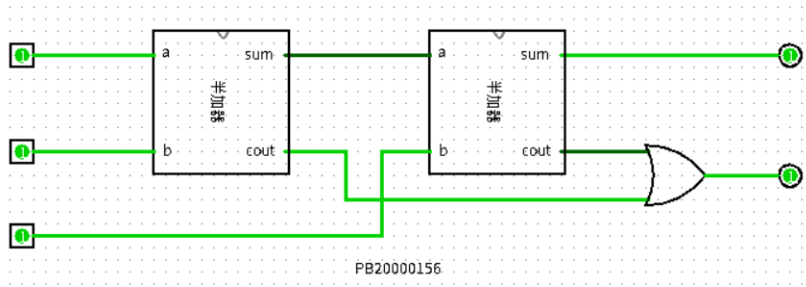


图 5: Step4 全加器

并拖放到合适的位置排成一列。

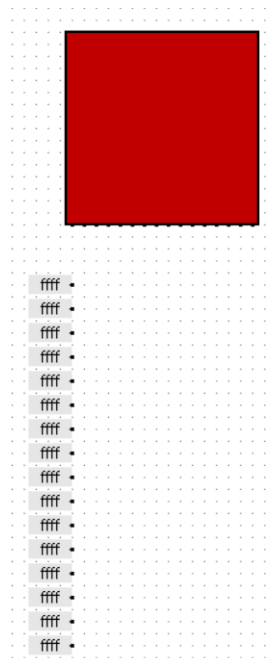


图 6: 第一题第一步

2、连线

3、将该电路复制成三份，并从左到右依次摆放。

4、使用js脚本将汉字分解为点阵

注意后面的二进制矩阵应该按列转换成16进制，因此需要写一个程序，

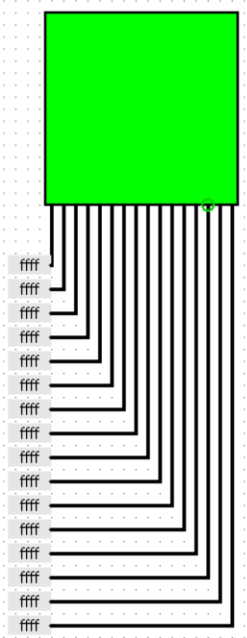


图 7: 第一题第二步

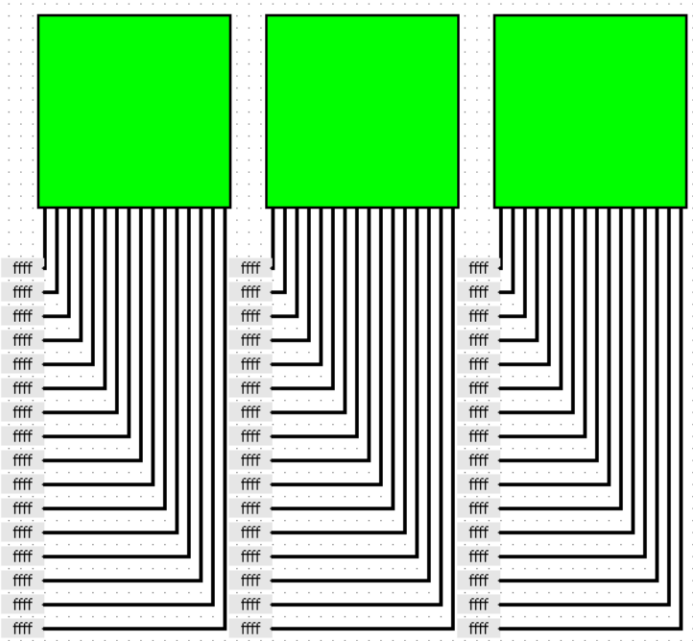


图 8: 第一题第二步

将矩阵转置后输出。

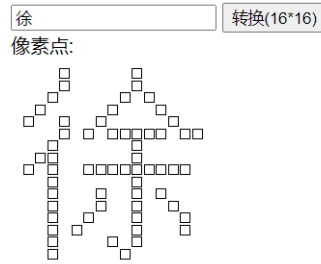


图 9: 汉字点阵

2进制:

```

0001000001000000
0001000001000000
0010000010100000
0100000100010000
1001001000001000
0001010111110110
0010000001000000
0110000001000000
1010011111111100
0010000001000000
0010001001010000
0010001001001000
0010010001000100
0010100001000100
0010000101000000
0010000010000000

```

图 10: 二进制矩阵

```

1  #include<iostream>

2  using namespace std;

3  int main()

4  {

5      char mat[16][16];

6      for(int i=0;i<16;i++) cin>>mat[i];

7      system("cls");

```



```

8         for(int i=0;i<16;i++)
9
10            {
11
12                for(int j=0;j<16;j++)
13
14                    cout<<mat[j][i];
15
16                cout<<endl;
17
18            }
19
20        return 0;
21    }

```

再使用Python脚本将每一行的二进制转换成16进制：

```

1  a=[]
2
3  for i in range(0,16):
4
5      a.append(input())
6
7  for i in range(0,16):
8
9      print(hex(int(a[i],2)))

```

最后，给LED点阵赋值：

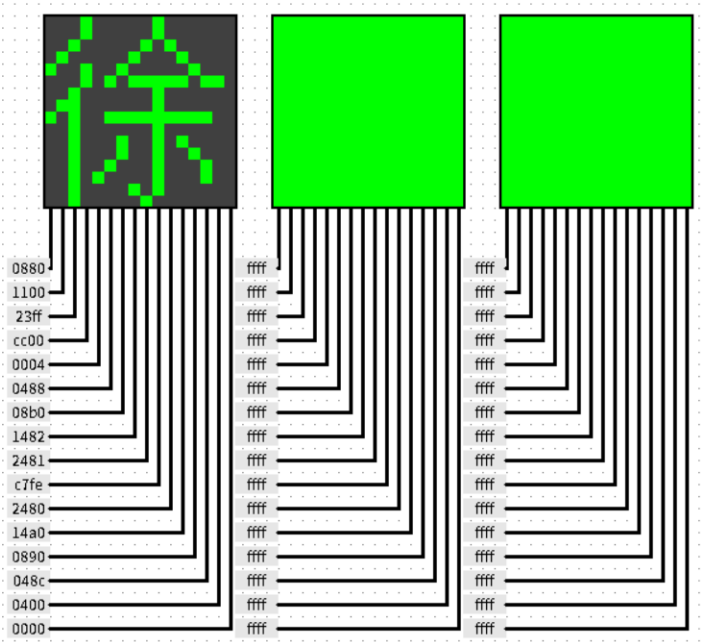


图 11: 第一题 LED点阵赋值

最终结果:

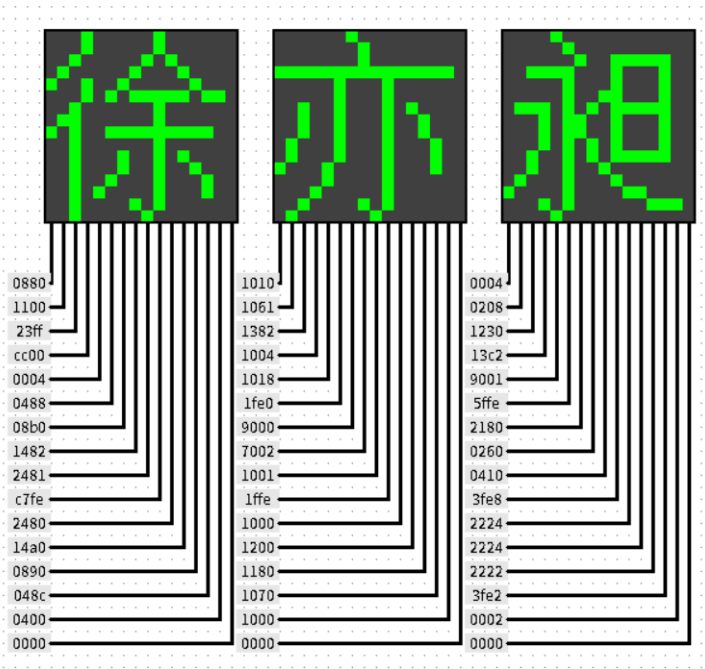


图 12: 第一题最终结果

第二题 七段数码管

- 1、在左侧的组件中找到7-Segment Display，布置相应的Pin为输入端口并进行布线。

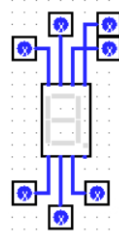


图 13: 第二题设置单个数码管

- 2、将数码管复制，排成一排。

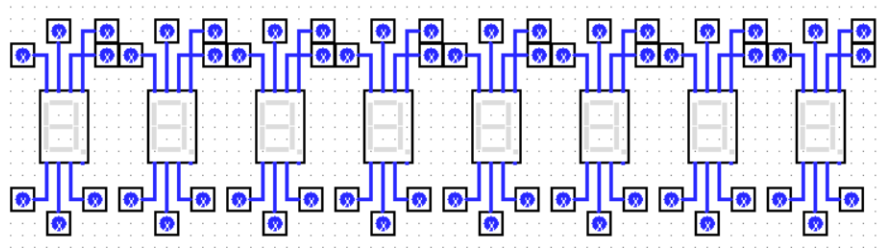


图 14: 第二题复制数码管

- 3、将阴极接地，形成共阴数码管。

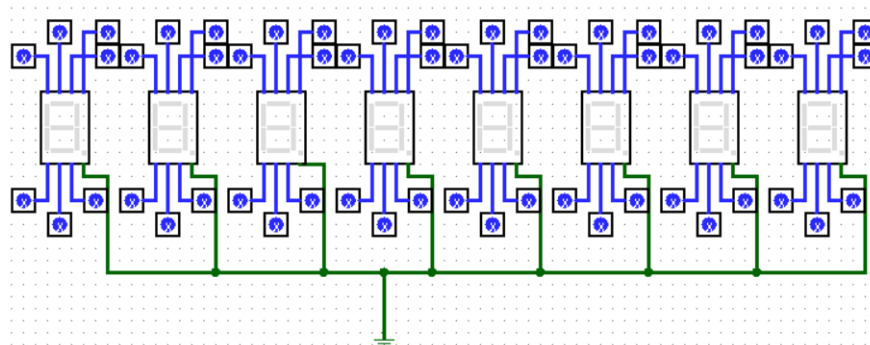


图 15: 第二题阴极接地

4、开启poke模式，依次点亮每一段。

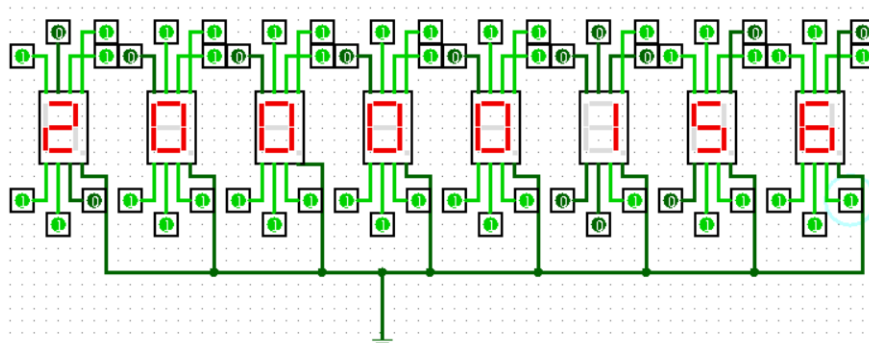


图 16: 第二题最终结果

### 第三题 MOS

1、以与门为例，第一步先将晶体管和端口排列好，可通过Type调整晶体管的类型(P型或N型)，以及Facing调节晶体管朝向。

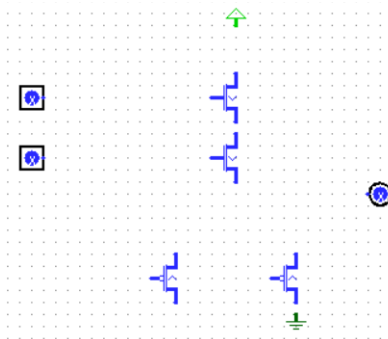


图 17: 第三题摆放组件

2、连线。

类似搭建其余两种门，最终结果如下：

### 第四题 数据选择器

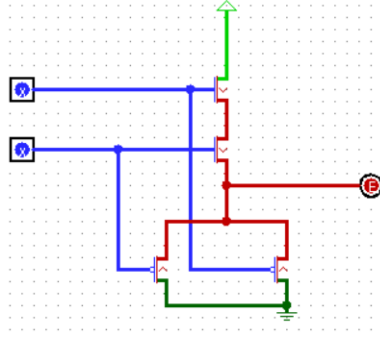


图 18: 第三题连线

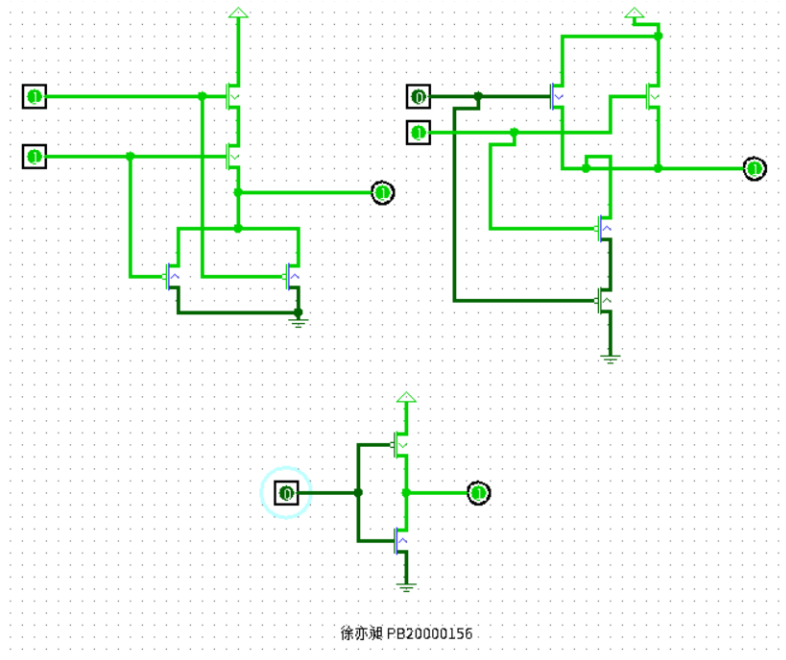


图 19: 第三题最终结果

1、添加电路，分别将上题中的三种门封装。

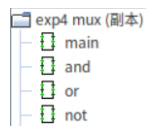


图 20: 第四题封装

2、设计二选一数据选择器，用到两个与门、一个非门、一个或门。

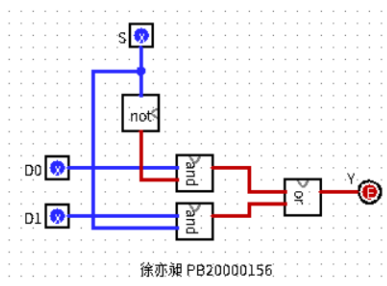


图 21: 第四题二选一数字选择器

3、将位宽改为2。

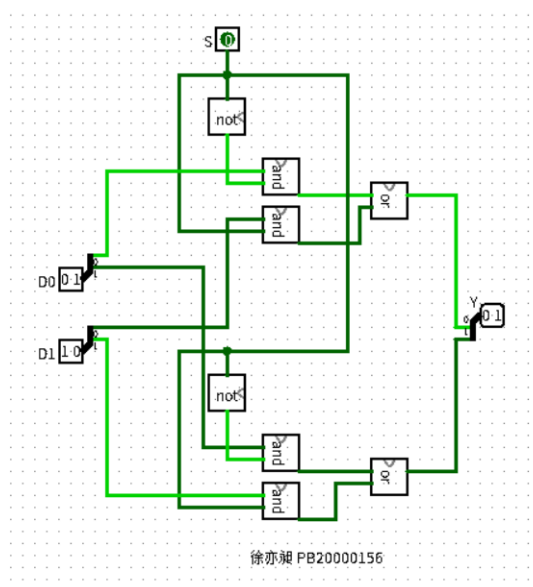


图 22: 第四题位宽改为2

4、新建电路，取名为mux，将上述二选一选择器放入并封装。最后的文件系统如图所示：

5、设计四选一选择器，用到六个与门、三个非门、三个或门。

### 【总结与思考】

1、通过本次实验，掌握了Logisim的基本用法，如电路的创建、模拟、

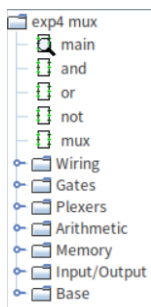


图 23: 第四题文件系统

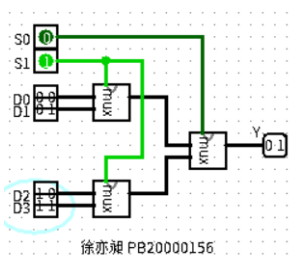


图 24: 第四题四选一选择器

封装等；通过具体的实践，对电路的设计有了更深的认识。

2、难度适中，可能不容易上手，但一旦上手可以很快完成。

3、任务量较大，比如前面的步骤4完全不必设计这么多电路，很多可以在后面的实验中学到。

4、建议以“学会为止”设计实验，比如练习题第一题让LED点阵显示自己的姓即可。