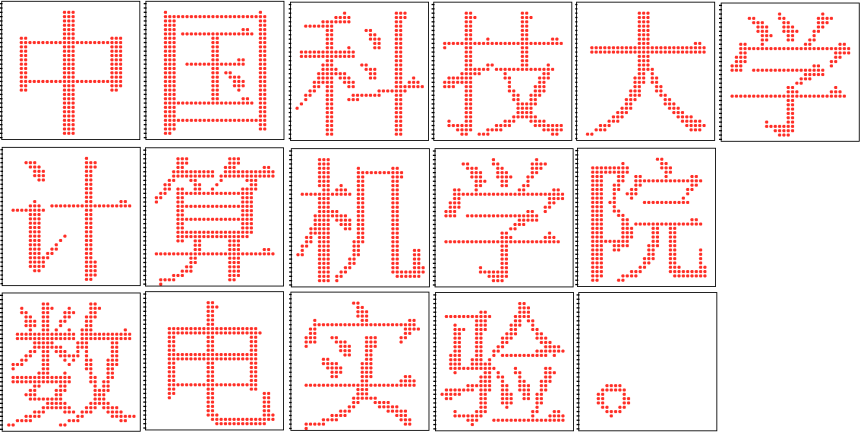
**实验17**

# **汉字点阵**

# **实验介绍**

本次实验中，我们将介绍如何在 Logisim 中进行汉字的显示。

[](https://soc.ustc.edu.cn/Digital/lab8/figs/logisim/logisim.png)

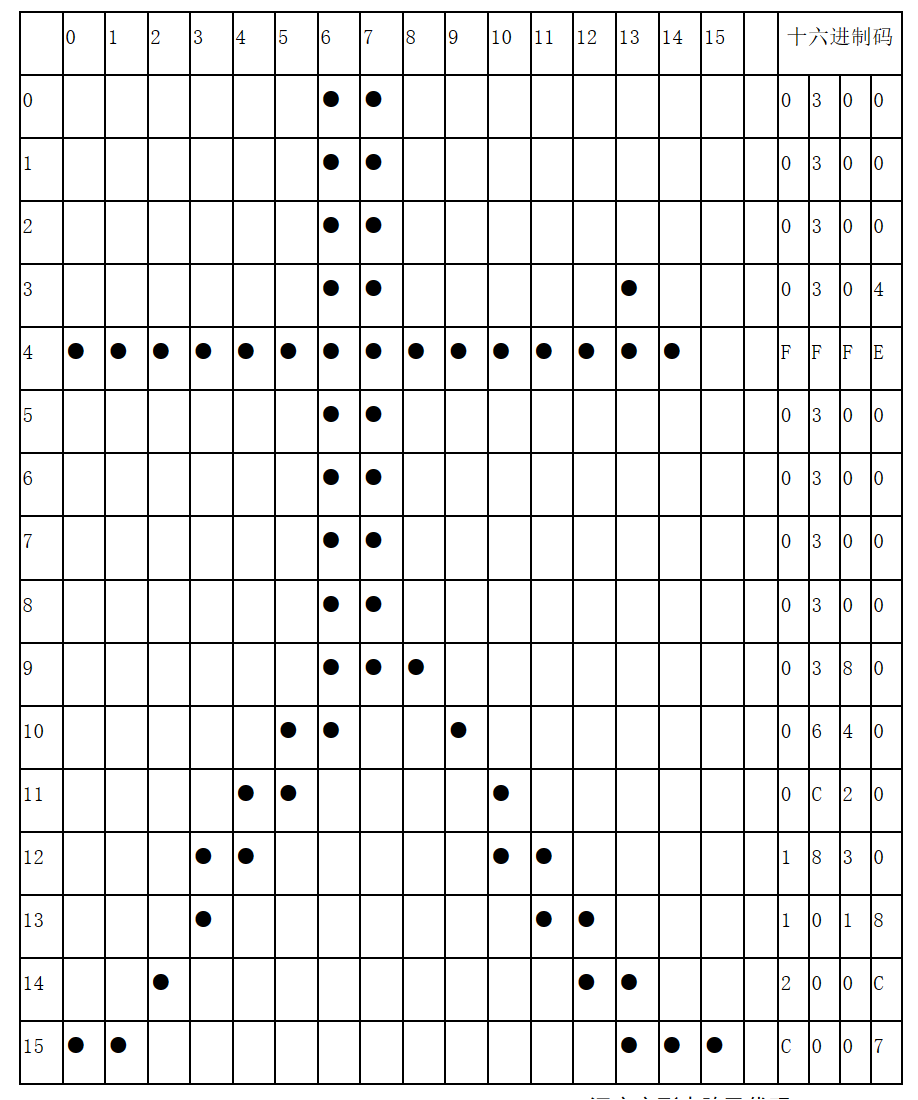
# **实验内容**

## 1 显示原理

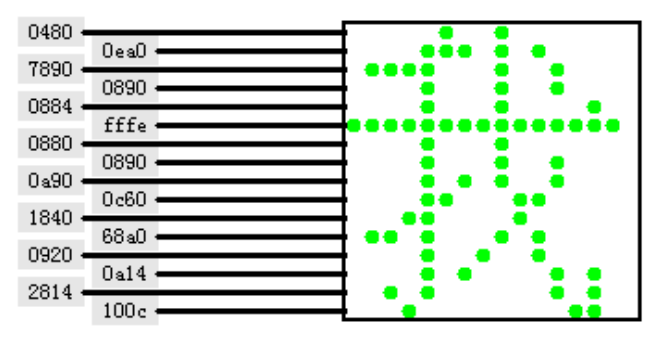
我们提到，普通的 LED 灯包括亮起和熄灭两个状态，如果输入的信号为低电平，则 LED 等保持熄灭；如果输入的信号为高电平，则 LED 保持亮起。七段数码管可以视作由 7 个条形 LED 灯组成，根据其亮起 / 熄灭的状态不同显示不同的字符。

LED 显示屏采用了类似的原理：一块显示屏上有 r×c 个 LED 灯，以矩形的方式排列。根据不同位置上 LED 灯的亮起与熄灭不同，我们可以显示不同的图案，其中就包括了汉字。

为了规范汉字显示时的字形，我们采用字形码作为显示方案。字形码是点阵代码的一种。为了将汉字在显示器或打印机上输出，我们把汉字视作一个个图形符号，并将其设计成点阵图，就得到了相应的点阵代码（字形码）。无论汉字的笔画有多少，我们都可以将其写在同样大小的方块中。

[](https://soc.ustc.edu.cn/Digital/lab8/figs/logisim/example_word.png)

上图为「大」字的字形点阵及编码，你可以自行验证：对于点阵的每一行，右侧的 4 位十六进制数即对应着左侧的 16 位「二进制数」（有点为 1，无点为 0）。

[](https://soc.ustc.edu.cn/Digital/lab8/figs/logisim/logisim_me.png)

我们在 Logisim 中可以使用 LED Matrix 组件，设置其大小为 16×16，以同样的方式显示中文字符。由于 Logisim 中复制电路时，输入引脚的状态会被清空，故我们建议大家使用常量（Constant，在 Wiring 目录中）而不是输入端口对 LED 点阵进行赋值。

由于 Logisim 中复制电路时，输入引脚的状态会被清空，故我们建议大家使用常量（Constant，在 Wiring 目录中）而不是输入端口对 LED 点阵进行赋值。

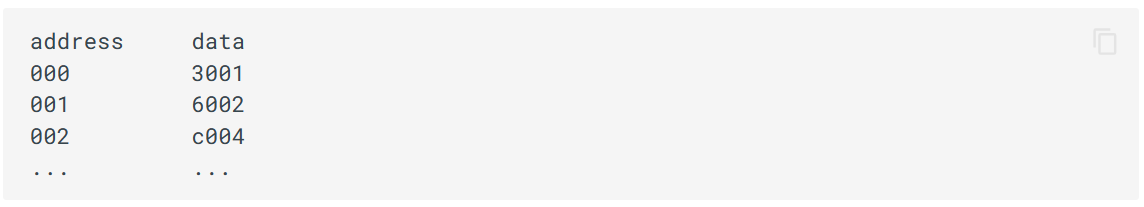
那么，如何得到汉字的字形码呢？你可以访问[这个网站](https://www.zhetao.com/fontarray.html)（https://www.zhetao.com/fontarray.html）获取一定的帮助。

## 2 滚动显示

实际生活中，我们见到的 LED 显示屏往往具有滚动显示的功能，用较少的 LED 屏幕显示一段较长的文本。现在，我们也可以在 Logisim 中实现类似的功能。

现在，每个 LED 屏幕上显示的内容是动态的，因此我们不能使用常量作为数据的输入，而是选择使用只读存储器 ROM 的数据端口作为输入。这样，我们只需要给出确定的地址 address，ROM 便会给出对应的数据 data 显示在 LED 屏幕上了。

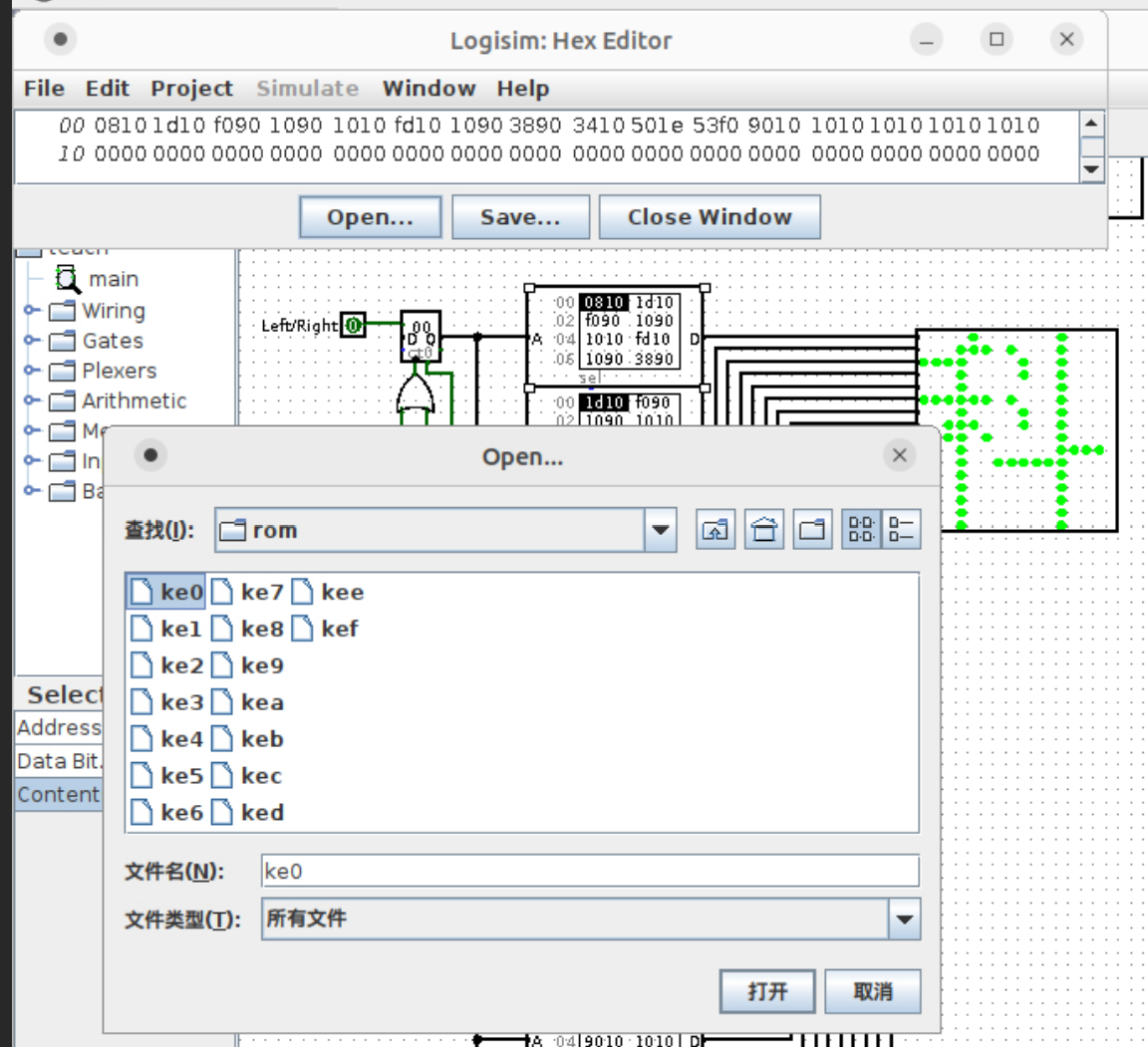
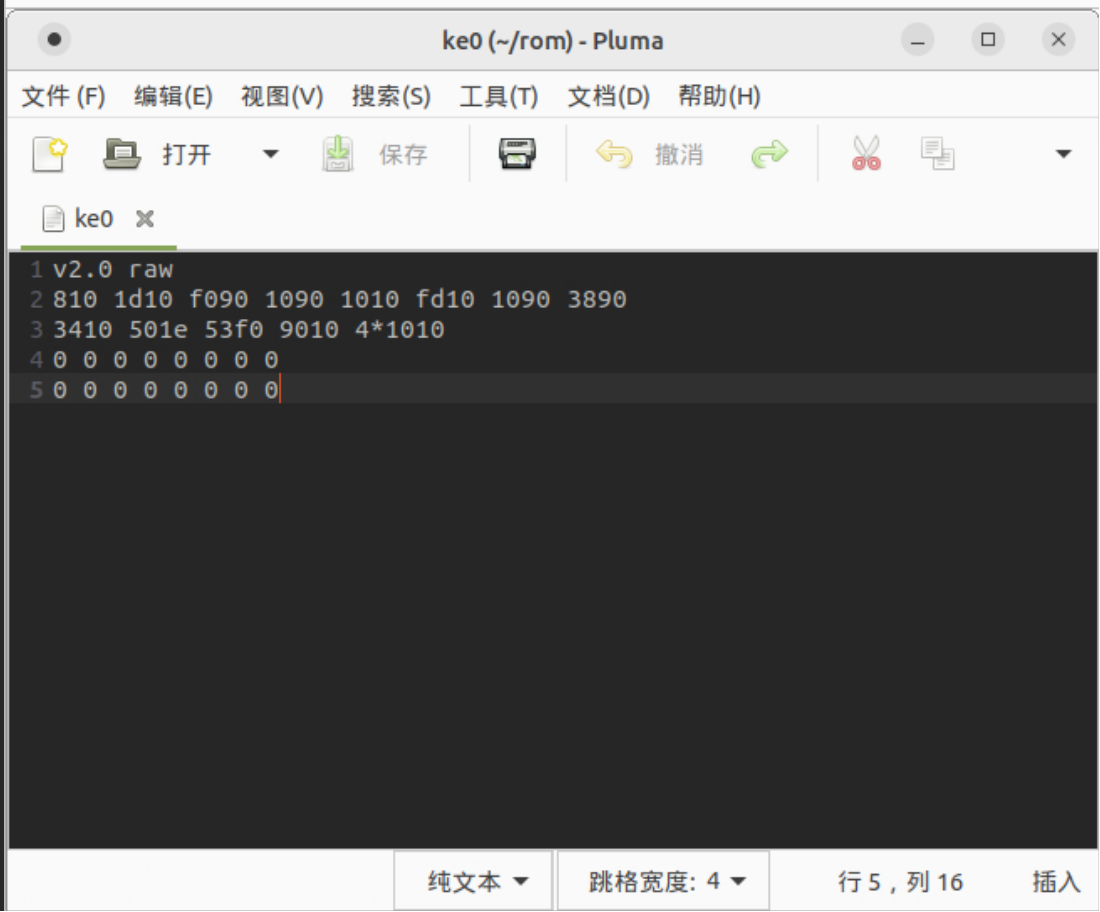
滚动显示的原理也十分直观：穷举每一时刻应当显示的字符即可。假定我们只在水平方向上进行移动操作，对于 LED 显示屏的某一行，某时刻其显示的数值为 0x3001。如果向左移动，则下一时刻显示的数值就是 0x6002，再下一时刻显示的数值就是 0xc004。因此，我们可以用一个计数器存储地址，并在 ROM 中写入如下的内容：



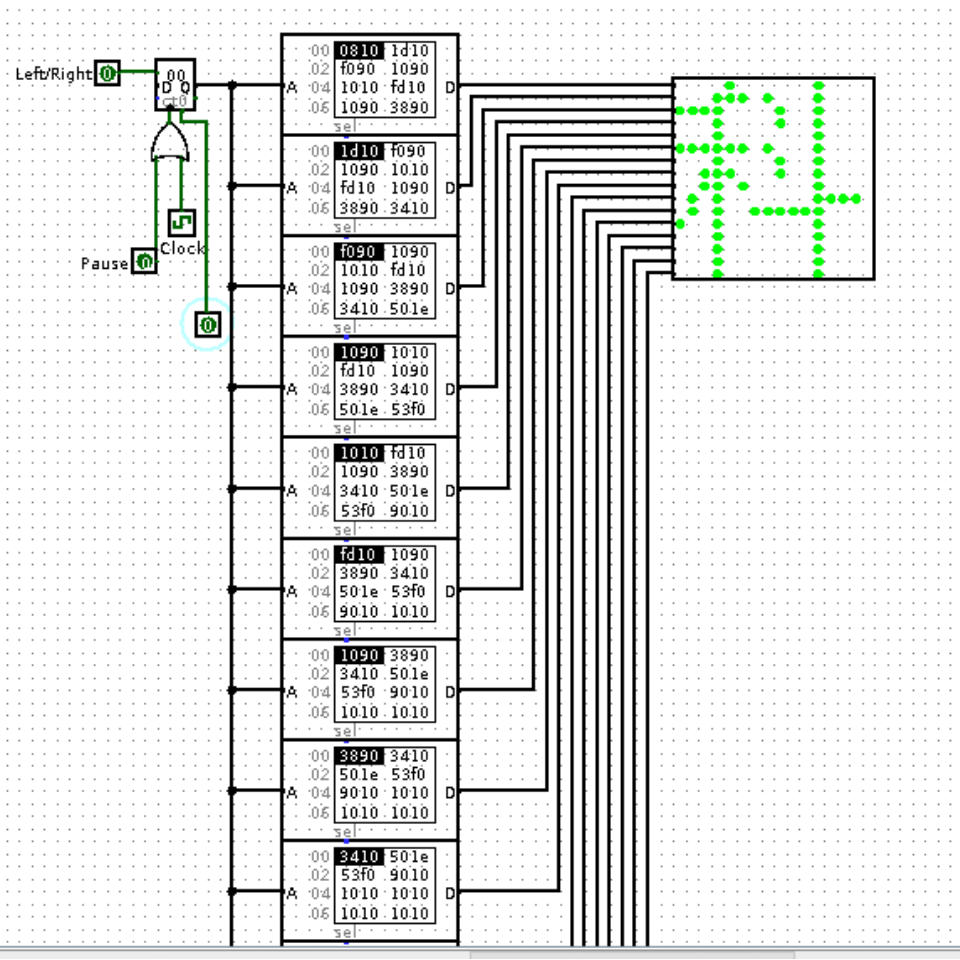
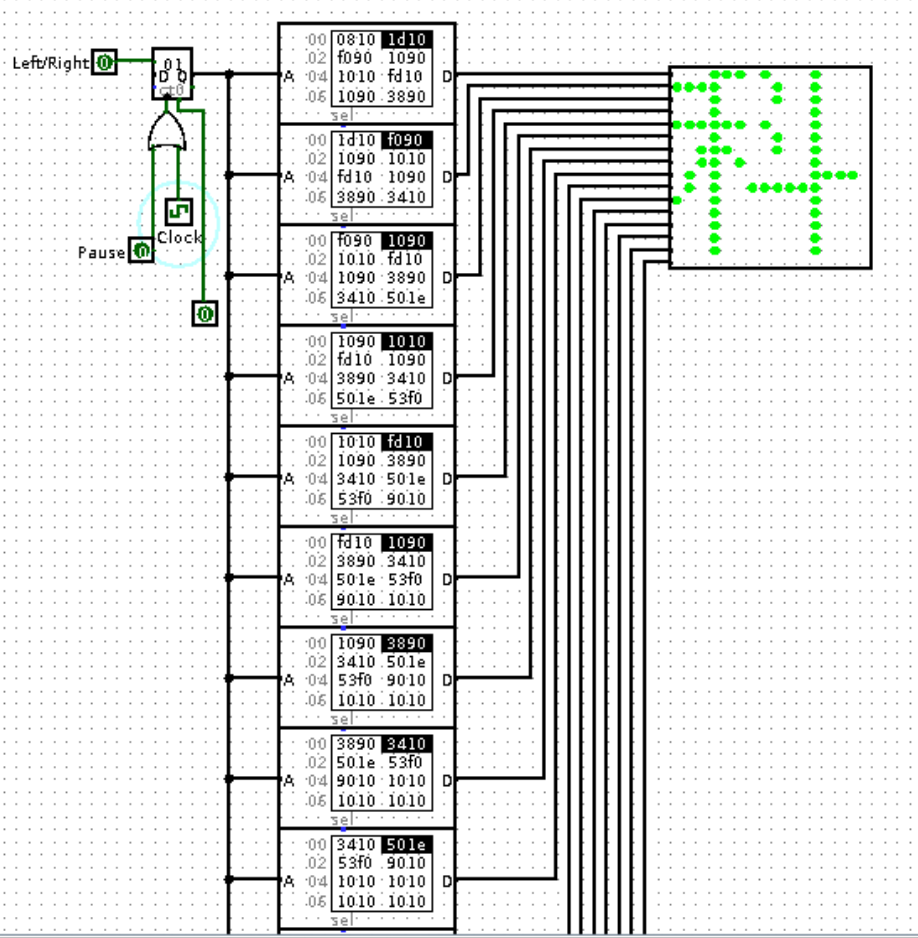
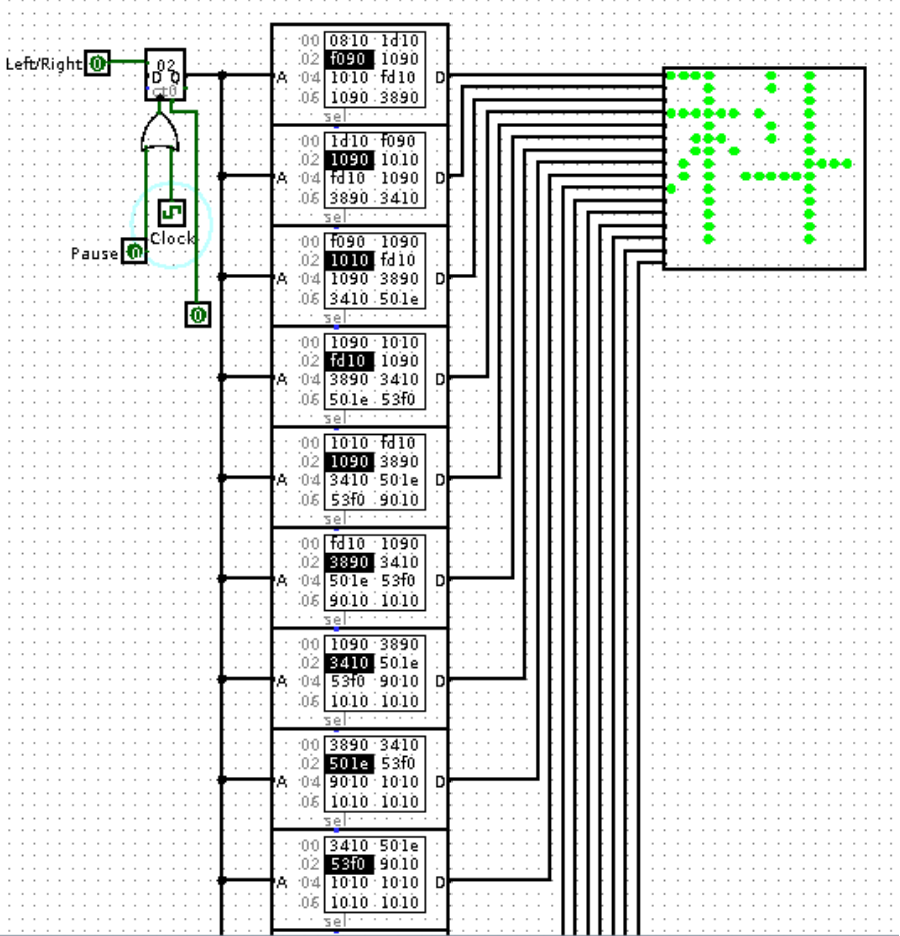
每个时钟周期，如果 address 增加，则 LED 显示屏会呈现向左移动的效果；如果 address 减小，则 LED 显示屏会呈现向右移动的效果。如果我们调整 address 变化的频率，则 LED 显示屏移动的速度也会发生变化。

上面展示了显示屏的某一行由 ROM 控制的流程。你可以使用 16 个 ROM 控制显示屏的每一行。

ROM 组件位于 Memory 目录下，可定制的内容包括存储单元数目、存储单元位宽以及内部存储的数值。你可以直接点击对应的存储单元，在键盘上输入该单元存储的数值；也可以使用特定的文件进行批量导入。文件开头必须为 v2.0 raw，后面的十六进制字形码可写作任意行，顺序正确即可。

[](https://soc.ustc.edu.cn/Digital/lab8/figs/logisim/rom_1.png)[](https://soc.ustc.edu.cn/Digital/lab8/figs/logisim/rom_2.png)

下面是对于「科」字的向上滚动显示。你可以参考其实现的思路完成自己的设计。

[](https://soc.ustc.edu.cn/Digital/lab8/figs/logisim/ke_1.png)[](https://soc.ustc.edu.cn/Digital/lab8/figs/logisim/ke_2.png)[](https://soc.ustc.edu.cn/Digital/lab8/figs/logisim/ke_3.png)

那么，如何实现循环显示呢？只需要让存储单元「循环」存储即可。

# 思考与练习

1. 请参考教程，基于 Logisim 中 Input/Output 目录下的 LED Matrix 组件搭建一个基础的汉字循环显示电路。要求显示的字符为「姓名 汉字点阵」（「 和 」 为中文引号，无需显示；姓名需要替换成你的名字）。

除了以上内容，你还需要额外添加一部分扩展功能，包括：

1. 实现 LED 横向循环显示的效果。要求 LED 显示屏需要以一定的速率向左或向右循环显示指定的字符，同一时刻显示的字符应不少于三个。
2. 在横向移动的基础上增加对移动方向的控制功能。你需要设计一个位宽为 1 的输入 lr，当其为高电平时，字符向右循环移动；否则字符向左循环移动。
3. 在横向移动的基础上增加暂停功能。你需要设计一个位宽为 1 的输入 pause，当其为高电平时，字符暂停移动（不是复位）；否则继续循环移动。
4. 在横向移动的基础上增加对移动速度的控制功能。你需要设计一个特定的输入 speed，要求移动范围至少可以在 0.5 倍、1 倍、2 倍、4 倍四个档位上调整。
5. 实现 LED 纵向循环显示的效果。要求 LED 显示屏需要以一定的速率向上或向下循环显示指定的字符，同一时刻显示的字符应不少于三个。

注意：前四项内容需要在一个 .circ 项目中的一个电路内实现。最后一项可以在 .circ 项目中新建一个电路。你可以自由使用 Logisim 中的任何组件。