

本设计的验证文件包括两个机器代码文件，分别完成冒泡排序和跳转测试+跑马灯+冒泡排序操作，验证文件链接

<https://github.com/lixuf/MIPS32->

[logisim/tree/main/mips%E4%BA%94%E7%BA%A7%E6%B5%81%E6%B0%B4%E7%BA%BF%E5%85%B7%E4%BD%93%E8%AF%B4%E6%98%8E%E6%96%87%E4%BB%B6/%E9%AA%8C%E8%AF%81%E6%96%87%E4%BB%B6](https://github.com/lixuf/MIPS32-logisim/tree/main/mips%E4%BA%94%E7%BA%A7%E6%B5%81%E6%B0%B4%E7%BA%BF%E5%85%B7%E4%BD%93%E8%AF%B4%E6%98%8E%E6%96%87%E4%BB%B6/%E9%AA%8C%E8%AF%81%E6%96%87%E4%BB%B6)

只需要点击主文件中的指令存储器，打开具体实现页面，右键 ROM，导入机器代码文件即可。sort.hex 执行完毕后，数据存储器状态如图 1、图 2、图 3 和图 4 所示，其中图 1 中存储器存储低字节，图 2 中存储器存储次低字节，图 3 中存储器存储次高字节，图 4 中存储器存储高字节。

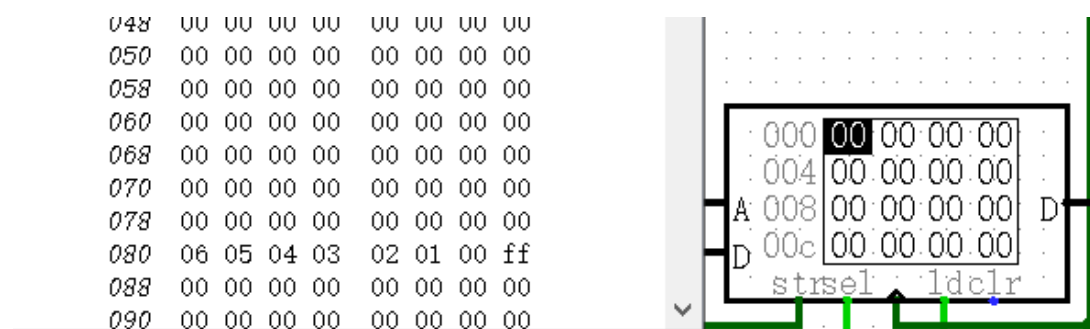


图 1



图 2

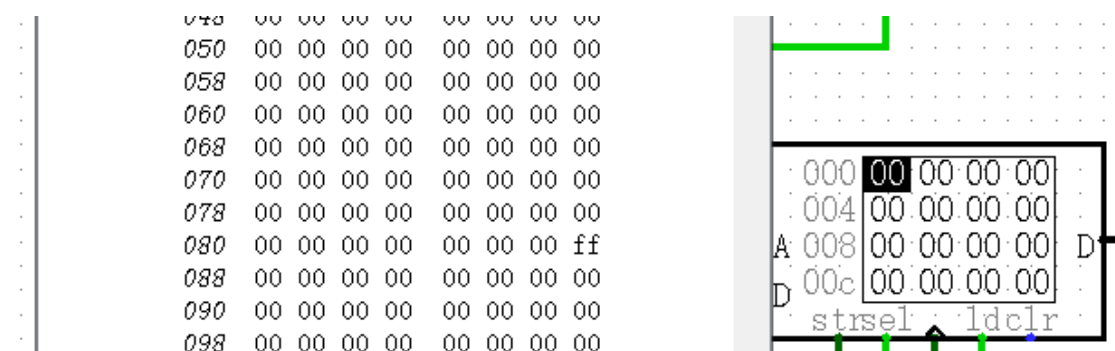


图 3

```

050 00 00 00 00 00 00 00 00
058 00 00 00 00 00 00 00 00
060 00 00 00 00 00 00 00 00
068 00 00 00 00 00 00 00 00
070 00 00 00 00 00 00 00 00
078 00 00 00 00 00 00 00 00
080 00 00 00 00 00 00 00 ff
088 00 00 00 00 00 00 00 00
090 00 00 00 00 00 00 00 00
098 00 00 00 00 00 00 00 00

```

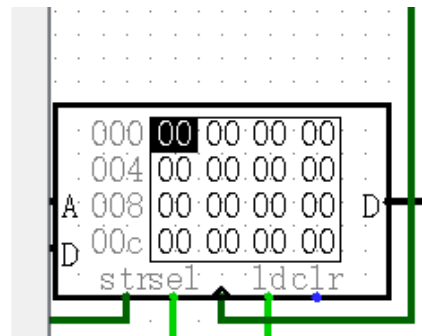


图 4

benchmark.hex 执行的一开始是各种跳转指令，如图 5 所示，之后是一段跑马灯程序，如图 6 所示，最后是一段冒泡排序算法，其结果如图 7 所示。

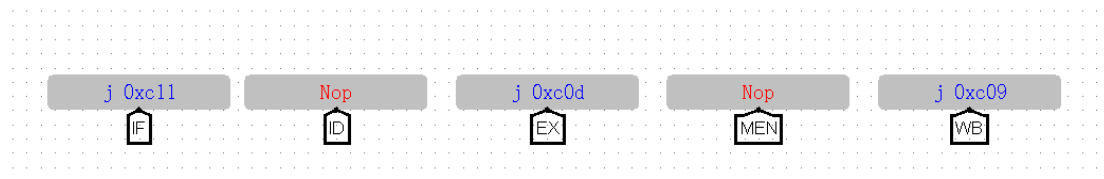


图 5

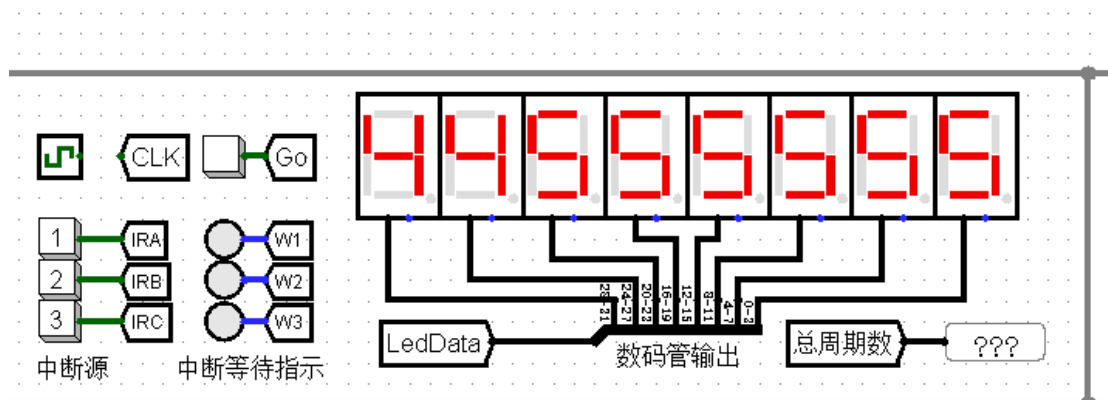


图 6

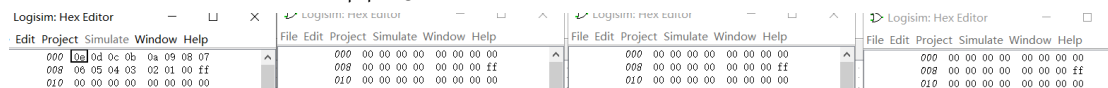


图 7 从左到右为低为、次低位、次高位和高位