1. **实验目的**

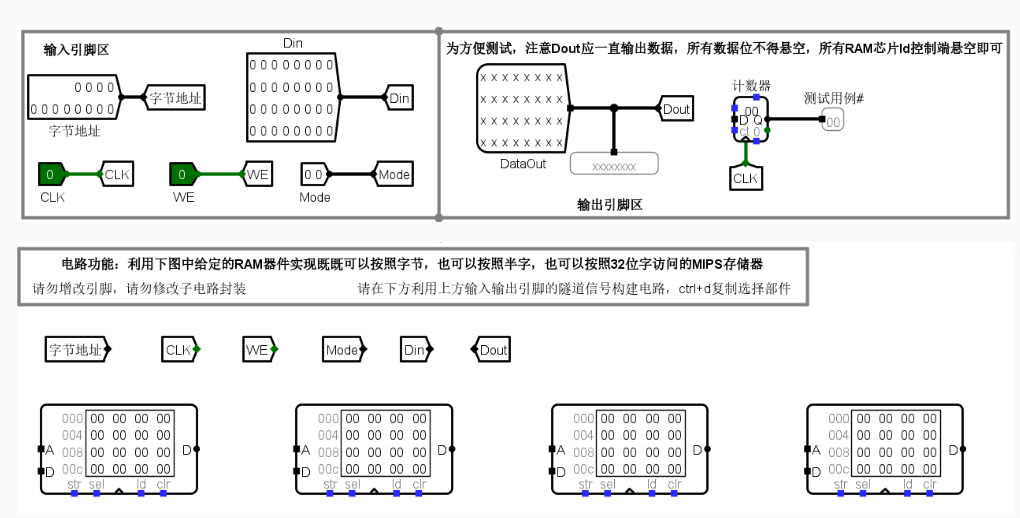
学生理解主存地址基本概念，理解存储位扩展基本思想，并能利用相关原理构建能同时支持字节、半字、字访问的存储子系统。

1. **实验内容**

Logisim 中 RAM 组件只能提供固定的地址位宽，数据输出也只能提供固定的数据位宽，访问时无法同时支持字节/半字/字三种访问模式，实验要求利用4个8位的 RAM 组件进行扩展，设计完成既能按照8位、也能按16位、也能按照32位进行读写访问的32位存储器，最终存储器引脚定义如下图。

**电路框架**

alu.circ

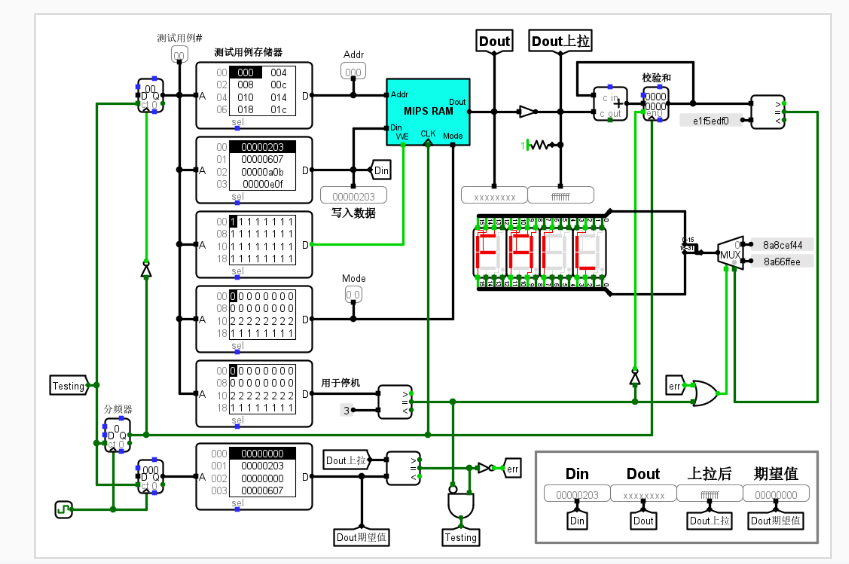
**电路引脚**

| **信号** | **输入/输出** | **位宽** | **功能描述** |
| --- | --- | --- | --- |
| Addr | 输入 | 12 | 字节地址输入（字访问时忽略最低两位，半字访问时忽略最低位，倒数第二位片选，字节访问时，低两位进行片选） |
| Din | 输入 | 32 | 写入数据 （不同访问模式有效数据均存放在最低位，高位忽略 |
| Mode | 输入 | 2 | 访问模式控制位（00 表示字访问，01 表示 1 字节访问，10 表示 2 字节访问） |
| WE | 输入 | 1 | 写使能，1 表示写入，0 表示读出 |
| Dout | 输出 | 32 | 读出数据 （不同访问模式有效数据均存放在最低位，高位补零）； |

****

**电路测试**

完成存储扩展设计后可以在自动测试电路中进行测试，电路通过会显示 Pass ，未能通过为停在出错位置，并显示 Fail ，如下图所示：

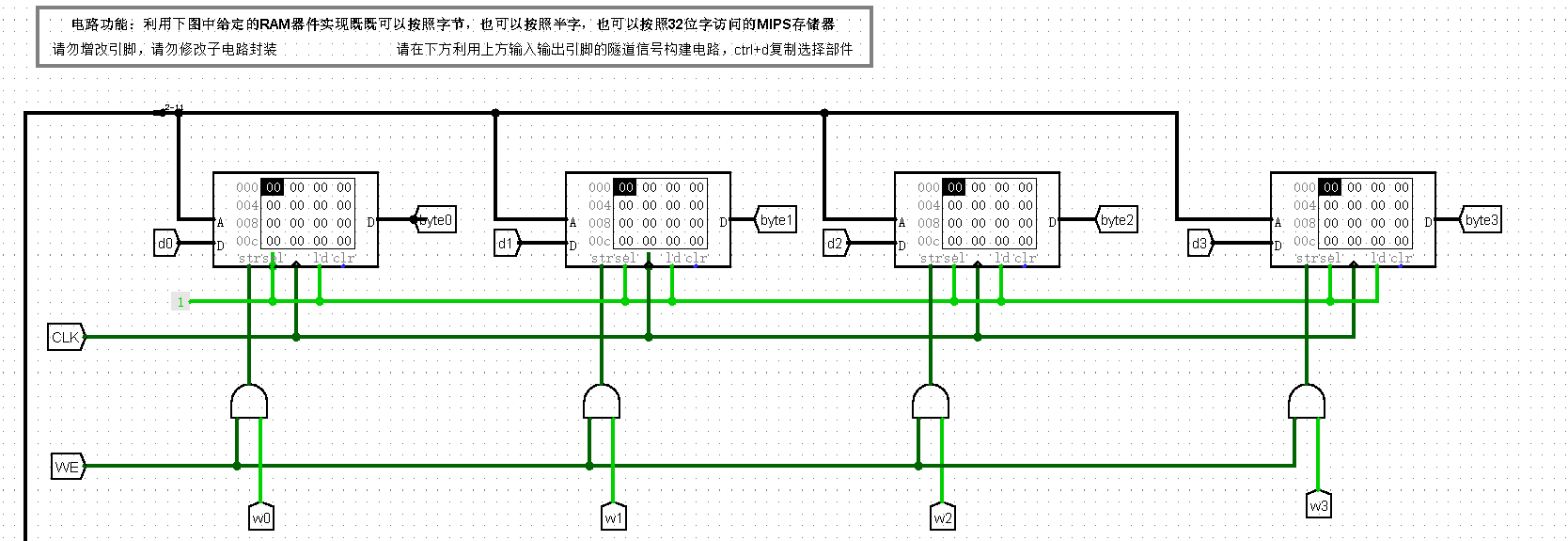


1. **电路设计**

**寄存器**

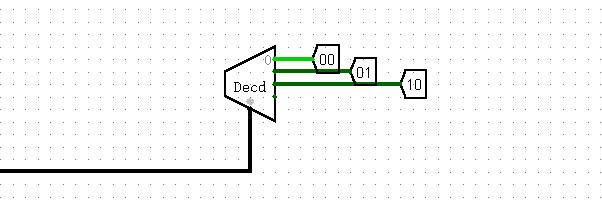
寄存器位宽为8，而最大单位字32，所以需要4片相同寄存器级联，分别为Byte1，2，3，4。将12位字节地址分出最低两位用来片选，接入寄存器。将d0，1，2，3作为数据输入。

构建此部分电路如下：（左上边为下方引出的字节地址线）



**设计电路将模式信号转换为位宽为1的信号**

采用译码器实现，如下：



**读取逻辑电路：**

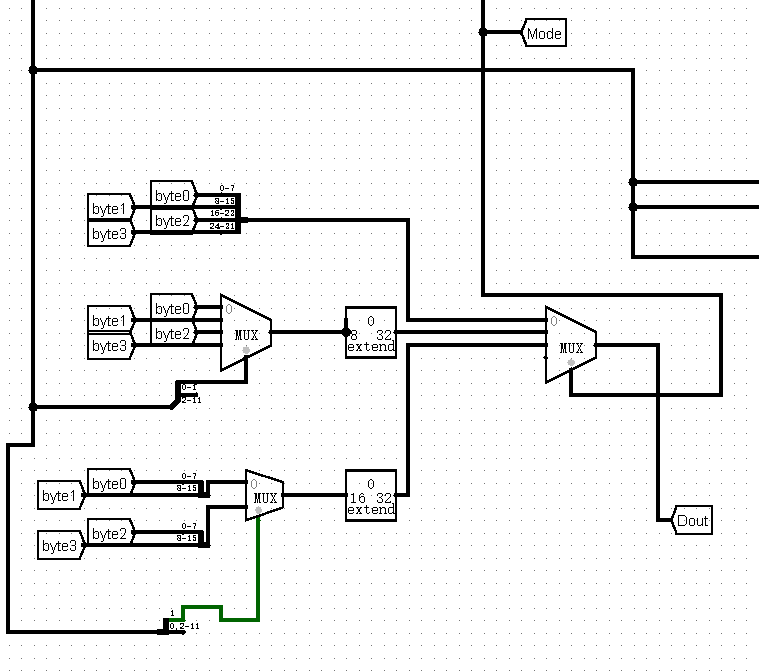
将ld信号设置为1，即恒为真，保持数据输出。

当需完成字读出时，需要输出全部数据。

当需完成半字读出时，需要字节地址的倒数第二位决定读出半字内容。

当需完成字节读出时，需要字节地址的倒数两位决定读出字节内容。

由此，搭建读取功能电路部分如下：



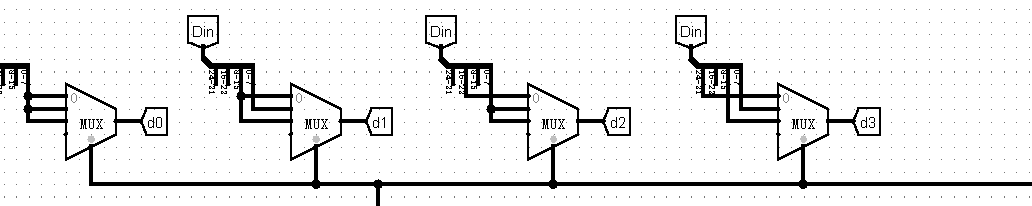
**写入逻辑电路：**

当mode==00时(字写入)，使0对应->7，1对应->15，2对应->23，3对应->31,从各个端口分别读取数据

当mode==01时(字节写入)，由写命令的片选信号挑选目的写入寄存器，将d0d1d2d3全部输入0-7位数据，

当mode==01时(半字写入)，会写入Din的0-15位数据，因为需要实现对齐，所以对于01，23号组合，较低编号都存放低位数据，较高编号存放高位数据，所以使d0，2输入0-7位数据，d1，3输入8-15位数据。

电路如下：



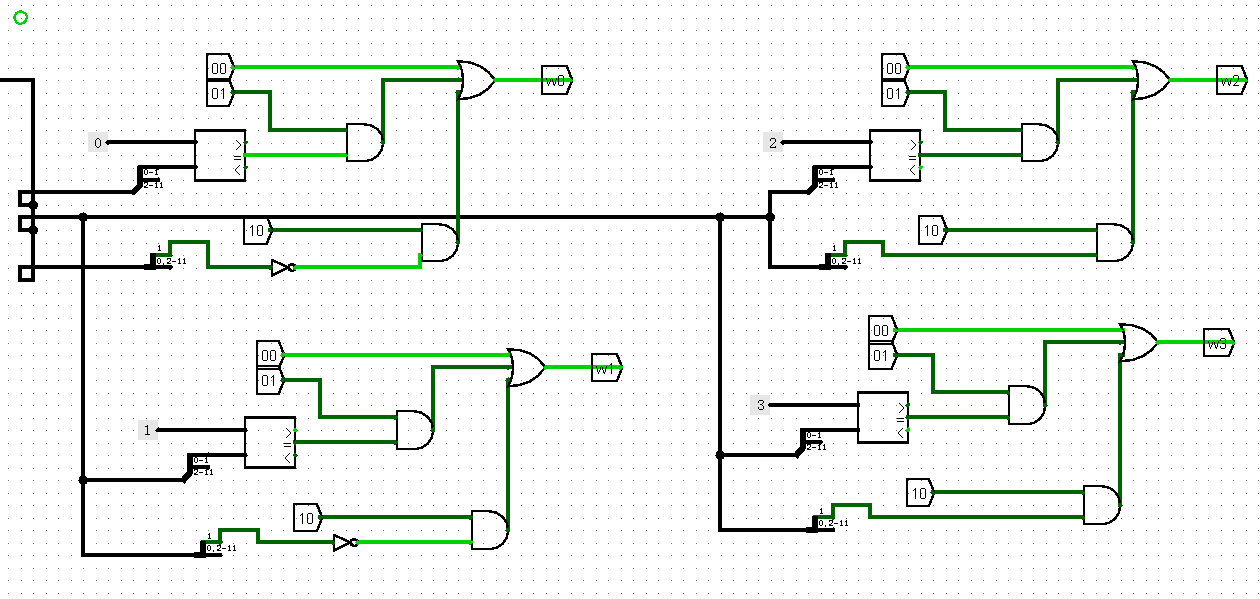
**写入片选逻辑电路：**

mode = 00时，所有片选信号都为1

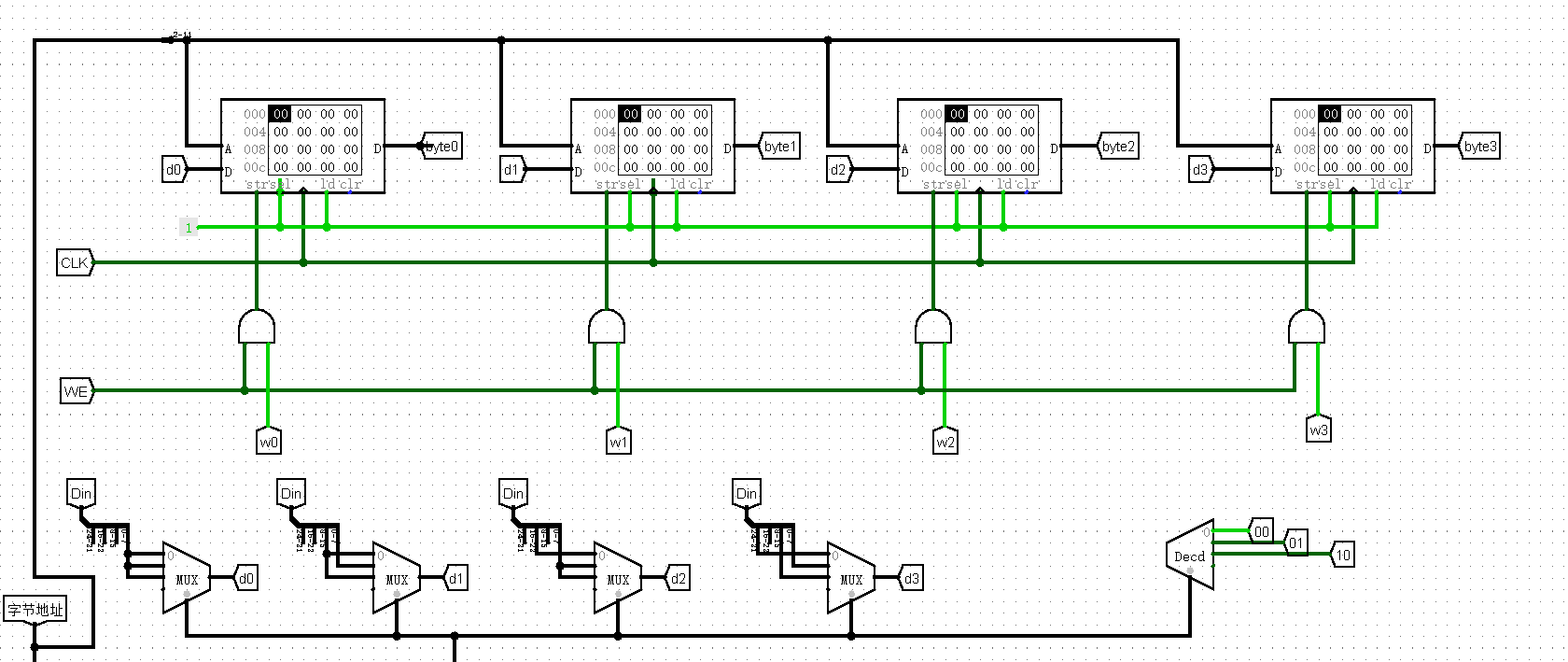
mode = 01时，字节地址最低两位和寄存器编号相对应的片选信号为1

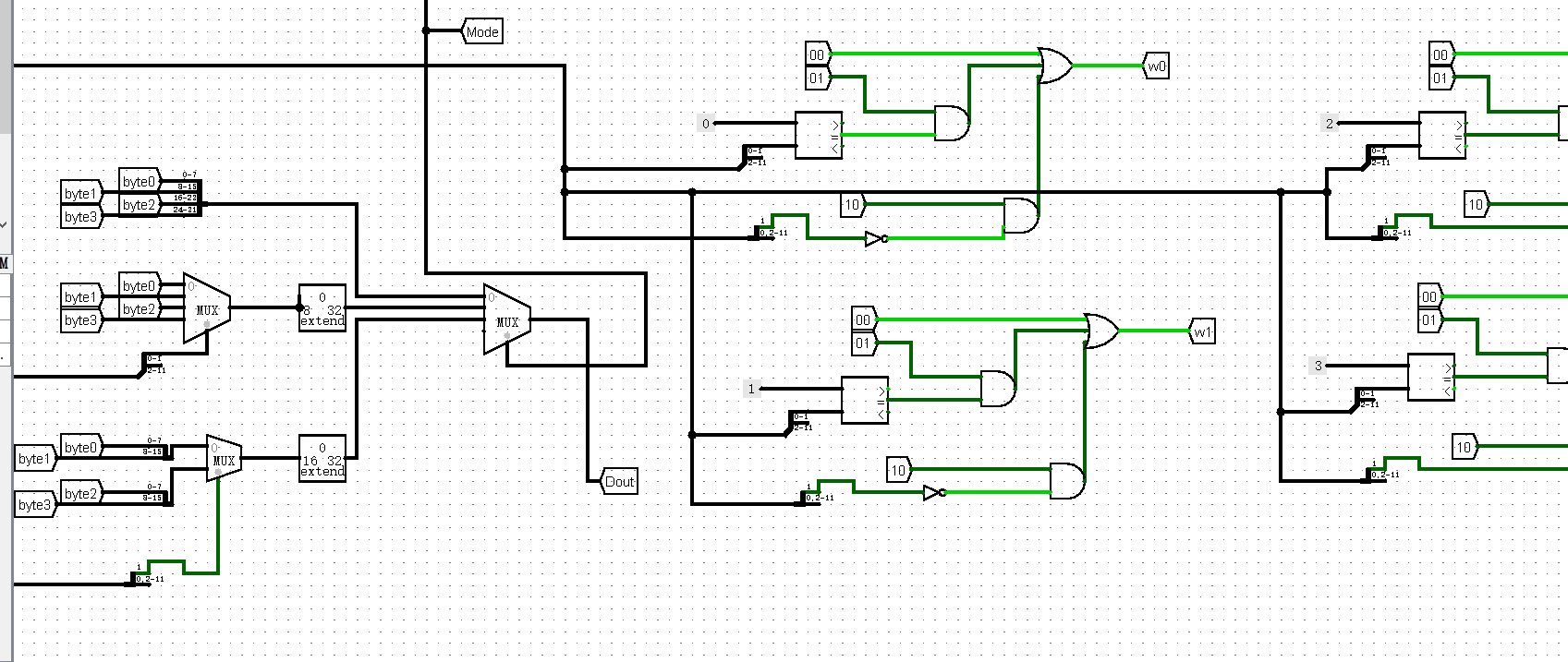
mode = 10时，字节地址的倒数第二低的一位来决定选择那两片。如果位0，选择01，选择23

设计电路如下：



**整体逻辑电路图：**

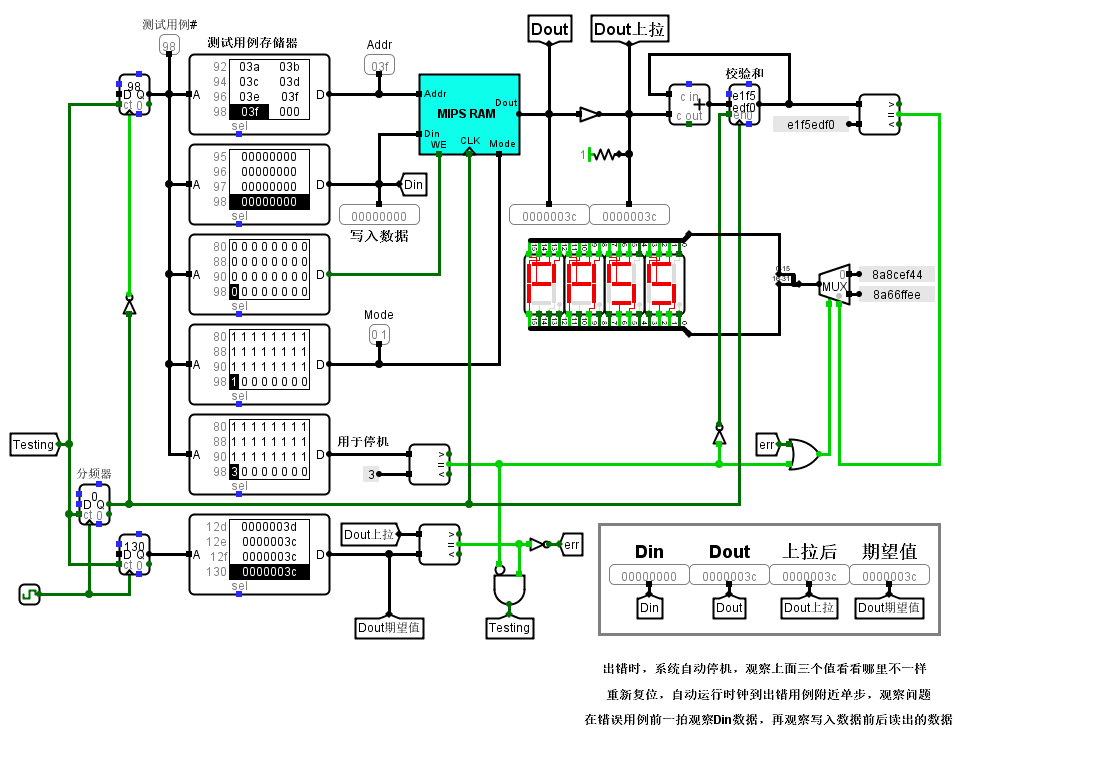




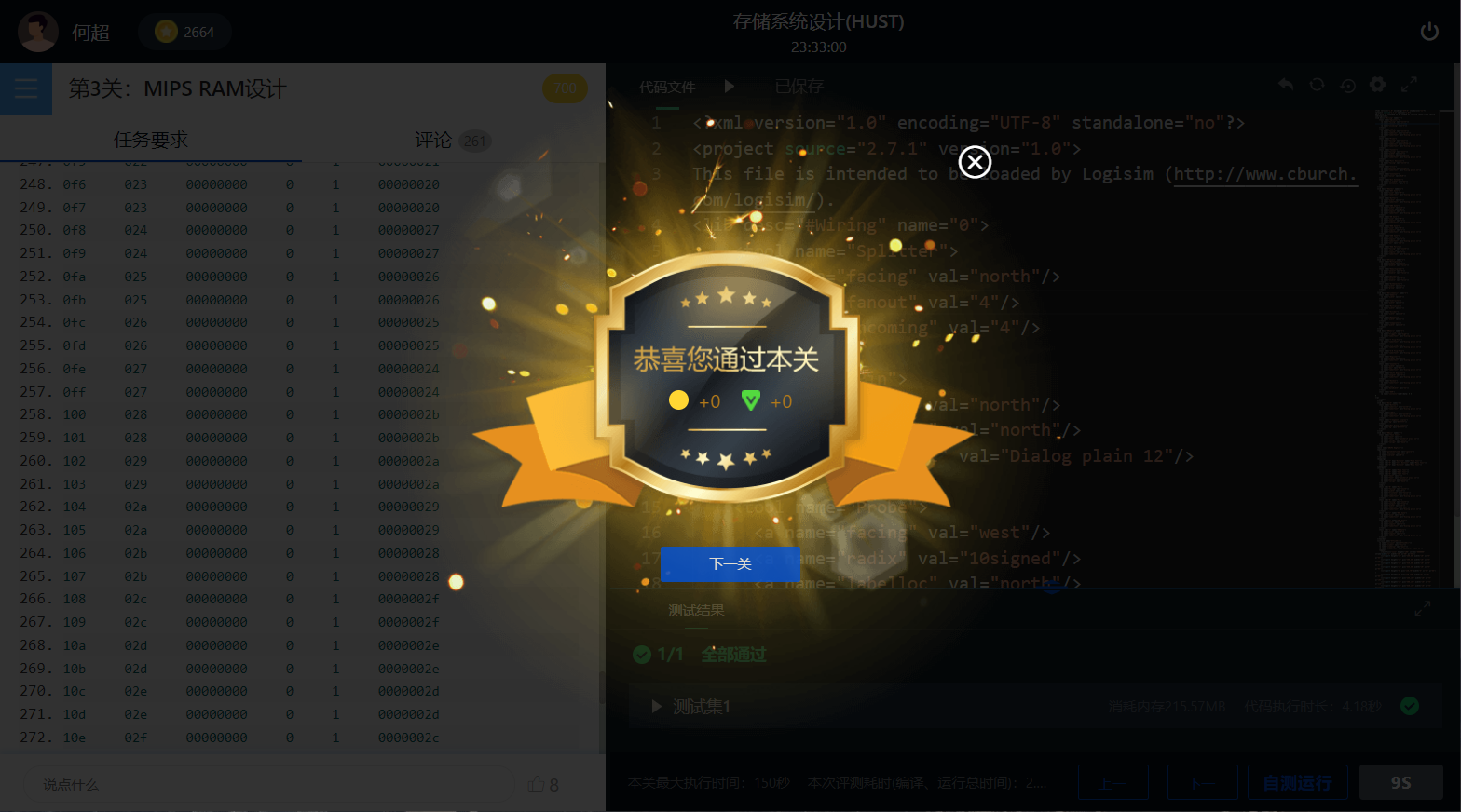
1. **实验结果**

**使用TEST电路按CTRL+K测试：**

**结果如下：**

****

**提交代码至平台：**

****

1. **实验总结/体会**

本次实验使我更加理解了存储位扩展基本原理，本次实验主要要求构建能同时支持字节、半字、字访问的存储子系统。本次实验电路规模较大，虽然电路通过了测试，但是没有做到模块化设计，直接观察自己设计的电路较难理解。但本次实验使用的部件较为简单基础，经过了本次实验，我收获了很多RAM设计相关基础知识，对Logisim有了更为深入的操作经验。在实验电路设计时，需要利用互联网查询陌生知识点，熟悉原理，以便于后续设计更为复杂的实验电路。