



**《计算机组成原理》**

**课程实验报告**

姓名： 陈澄

学院：信息学院

系：软件工程

专业：软件工程

学号：32420212202930

2023年4月6日

**第3次实验 存储系统**

1. **实验环境**

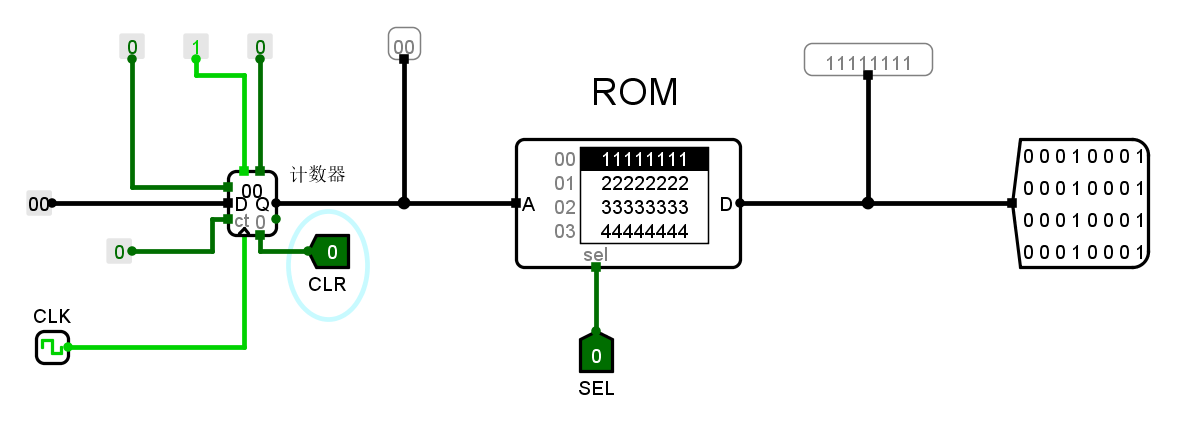
**Windows系统下运行Logisim软件（需安装JDK）。**

1. **实验内容**
   1. **课堂完成部分（验证实验的内容）**
2. **请按照实验课件规定的内容，逐个完成实验。**
3. **采用屏幕拷贝的方式记录实验结果，黏贴到实验报告中，并给出相关的文字说明。**

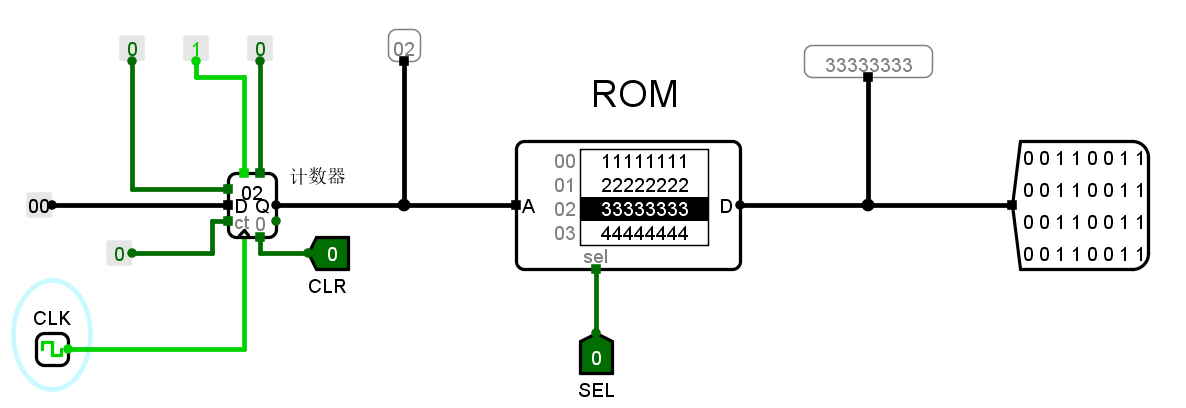
一、存储器组件验证实验

1、ROM存储器组件电路

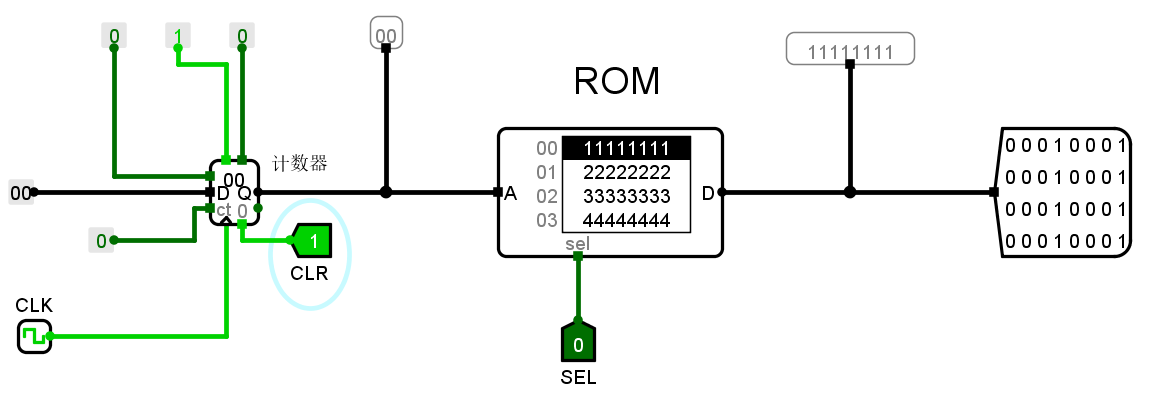
向ROM中装入内容，具体内容在“ROM内容.txt”文件中



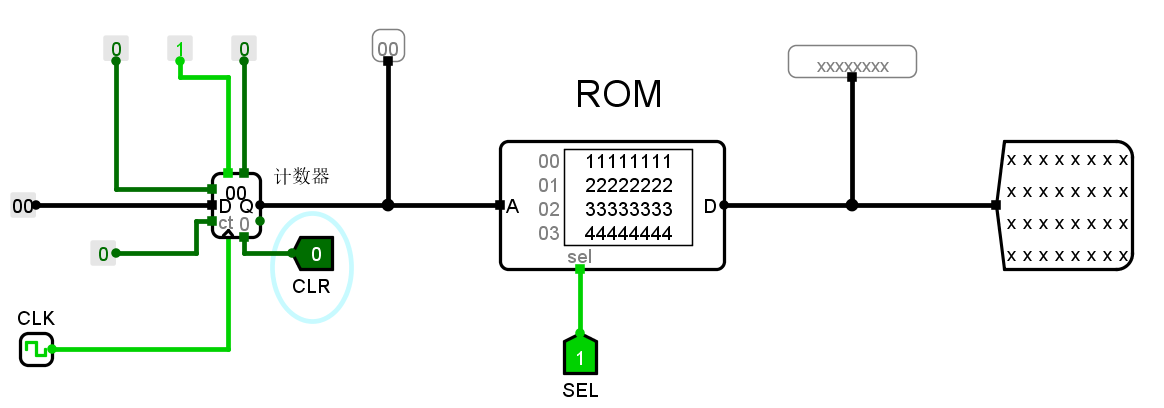
置CLR=0、SEL=0；然后点击CLK，则改变地址A的值，观察对应的输出数据D



置CLR=1，让地址A变为0；再置CLR=0，然后点击CLK，则重新开始

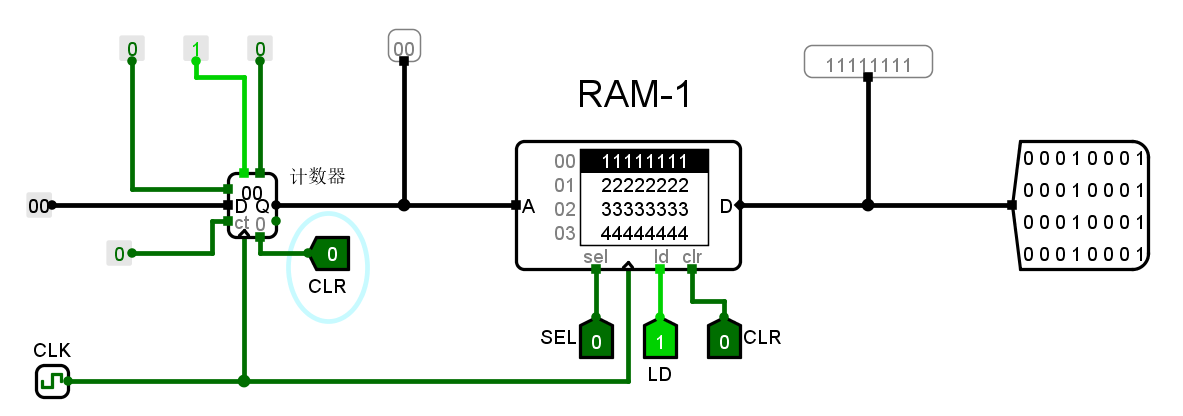


置SEL=1，则ROM存储器不工作

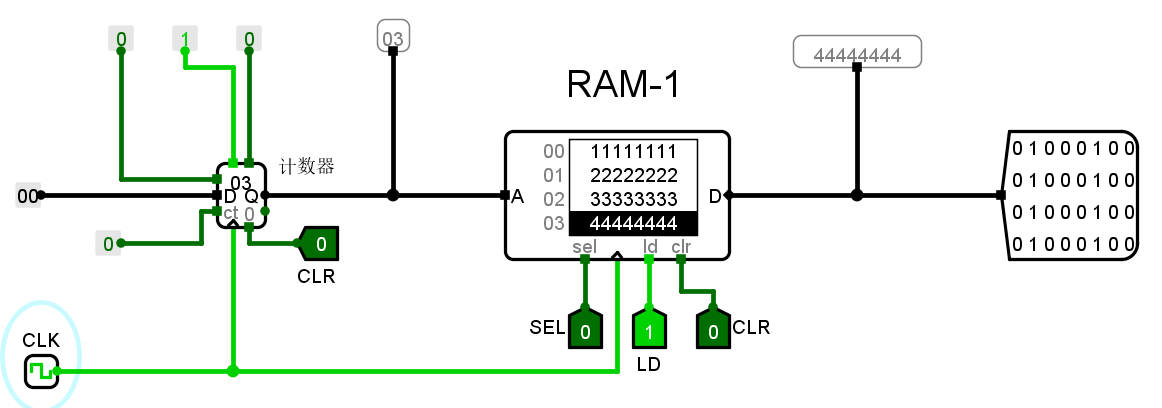


2、RAM存储器组件电路（同步模式）

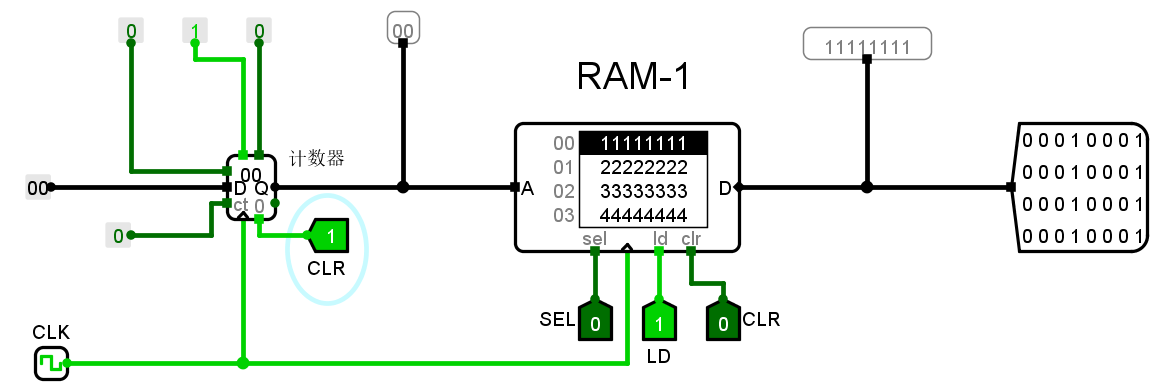
向ROM中装入内容，具体内容在“ROM内容.txt”文件中



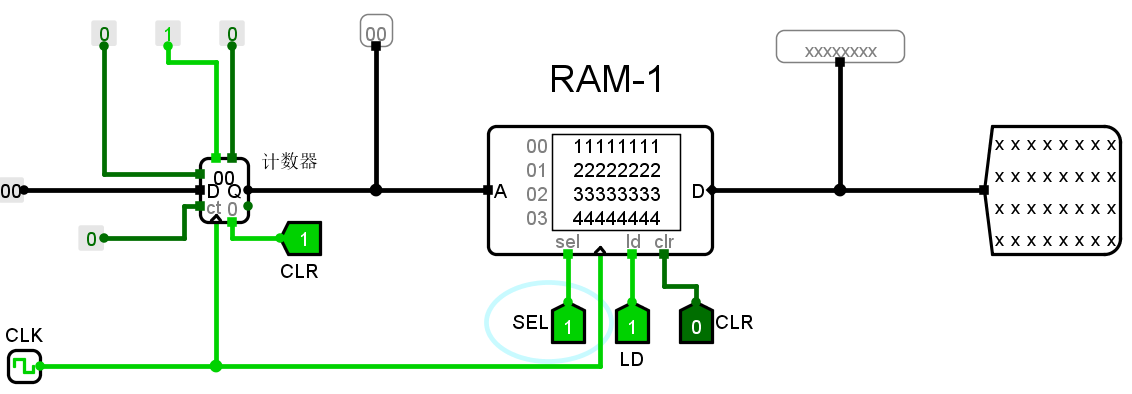
置CLR=0、SEL=0；然后点击CLK，则改变地址A的值，观察对应的输出数据D



置CLR=1，让地址A变为0；再置CLR=0，然后点击CLK，则重新开始

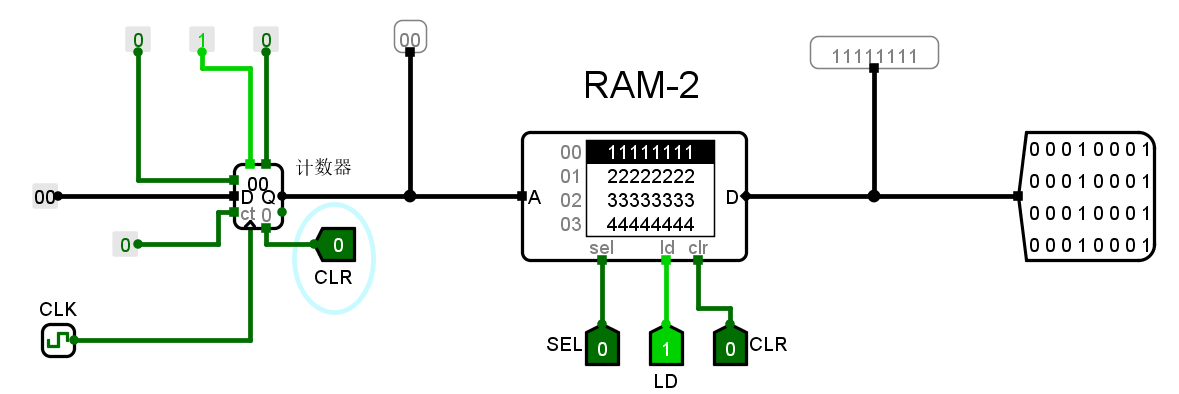


置SEL=1，则ROM存储器不工作

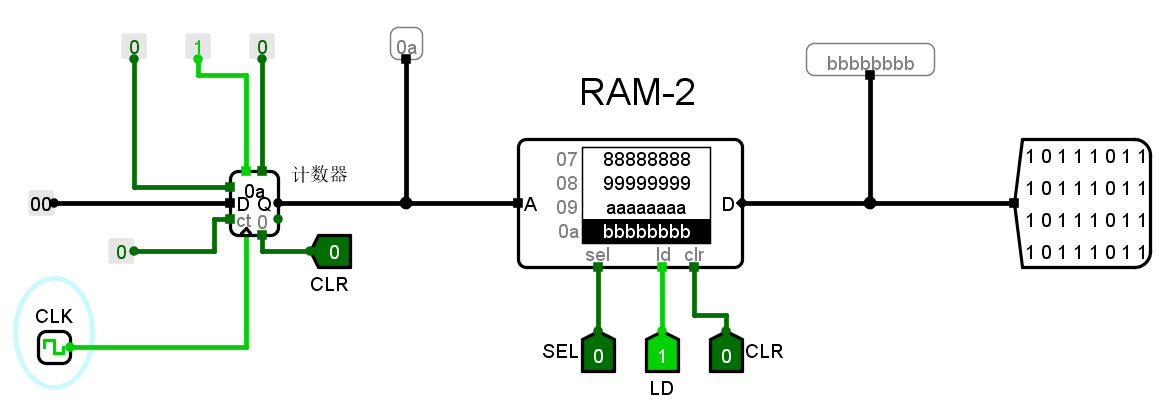


3、RAM存储器组件电路（异步模式）

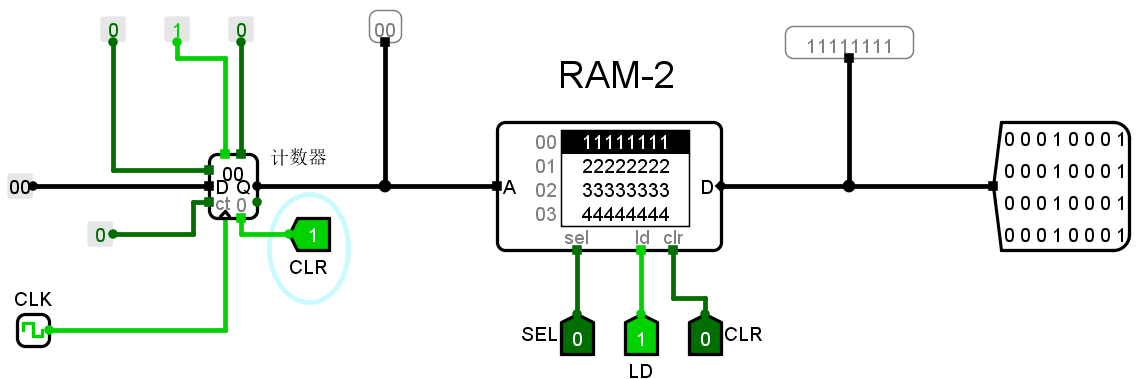
向ROM中装入内容，具体内容在“ROM内容.txt”文件中



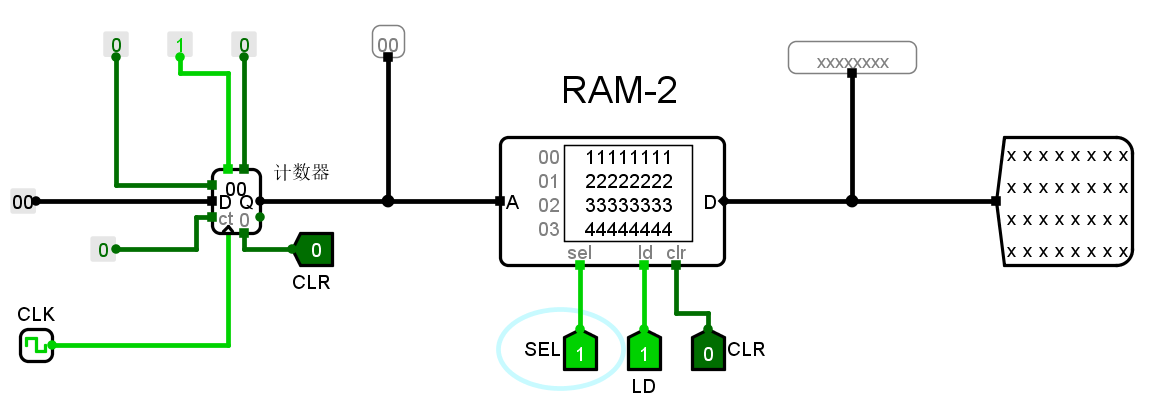
置CLR=0、SEL=0；然后点击CLK，则改变地址A的值，观察对应的输出数据D



置CLR=1，让地址A变为0；再置CLR=0，然后点击CLK，则重新开始

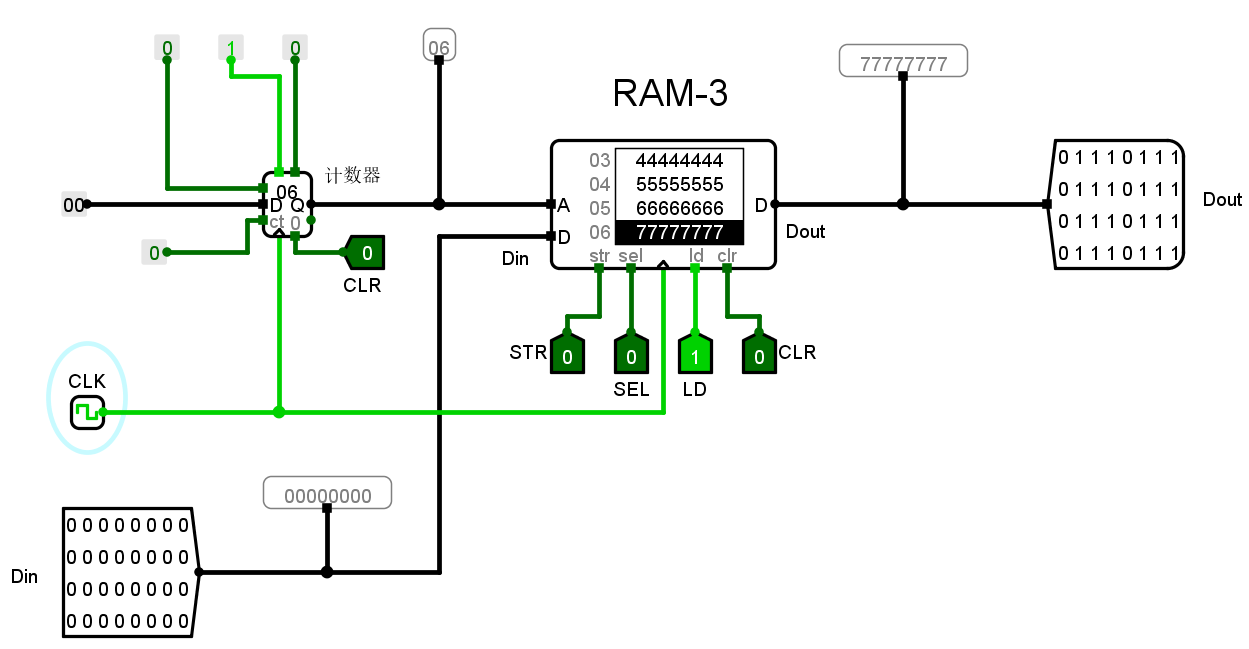


置SEL=1，则ROM存储器不工作

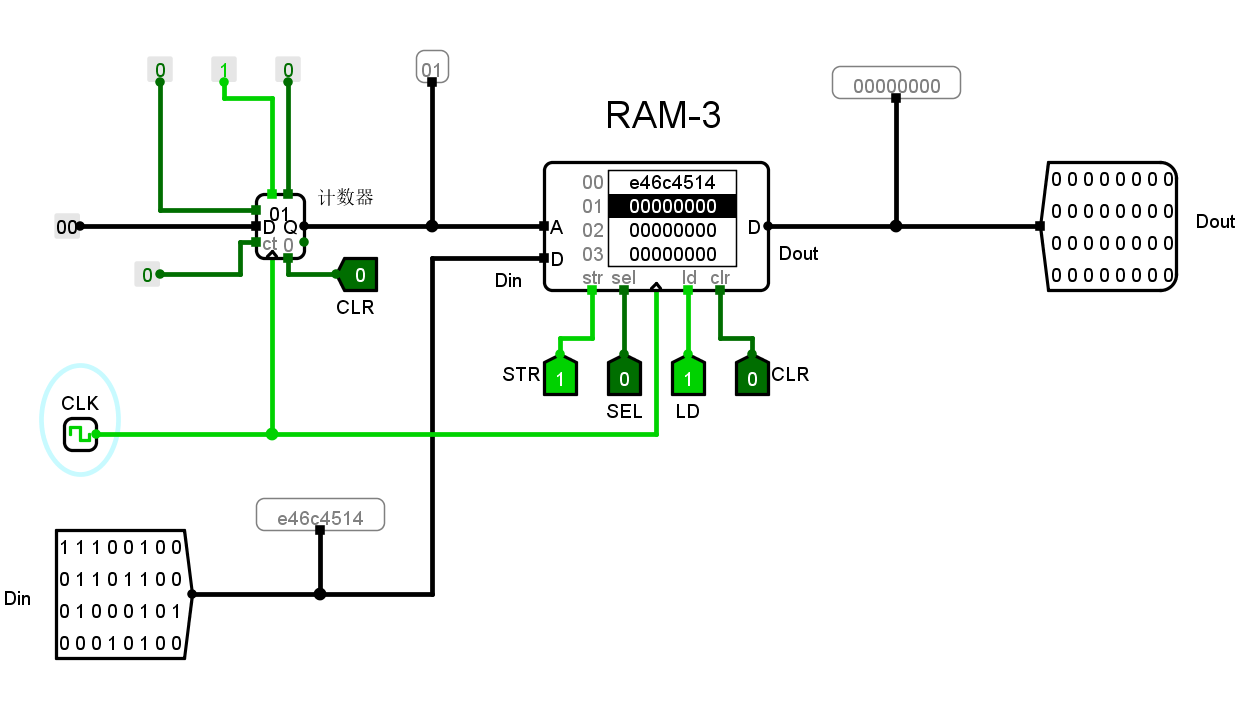


4、RAM存储器组件电路（分离模式）

读操作



写操作

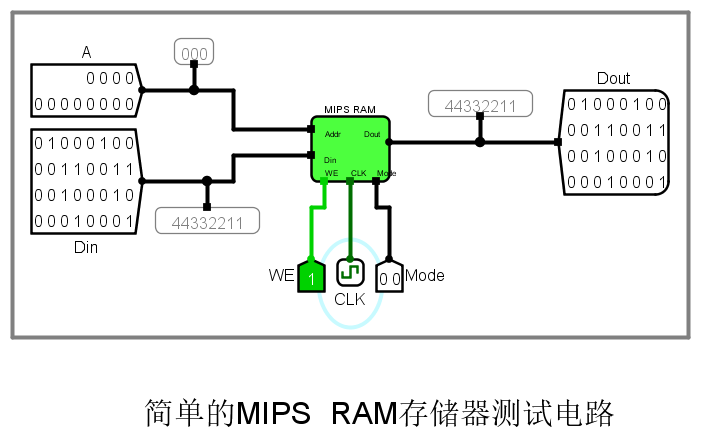


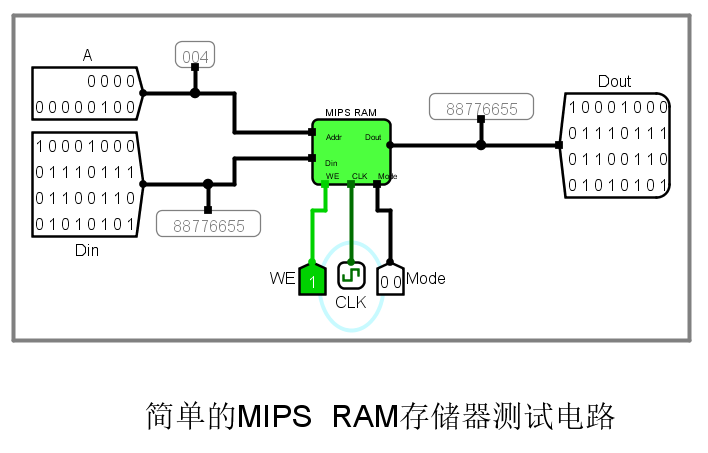
二、MIPS RAM 存储器验证实验

1、MIPS RAM 存储器电路

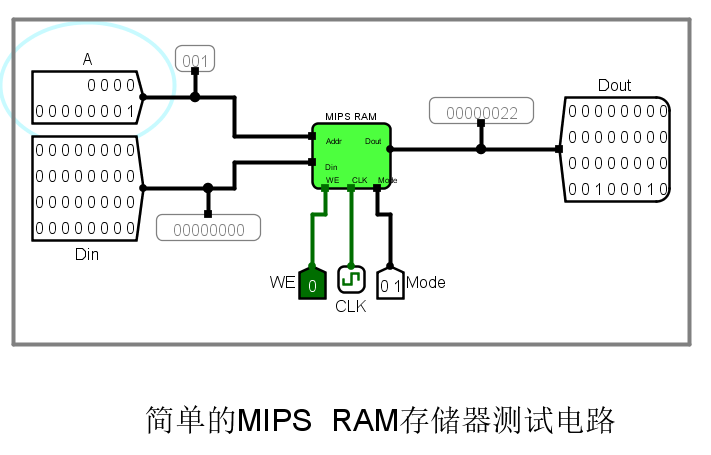
2、简单的MIPS RAM 存储器测试电路

第1步：将存储单元0～7，分别写入11～88（均为十六进制，下同）

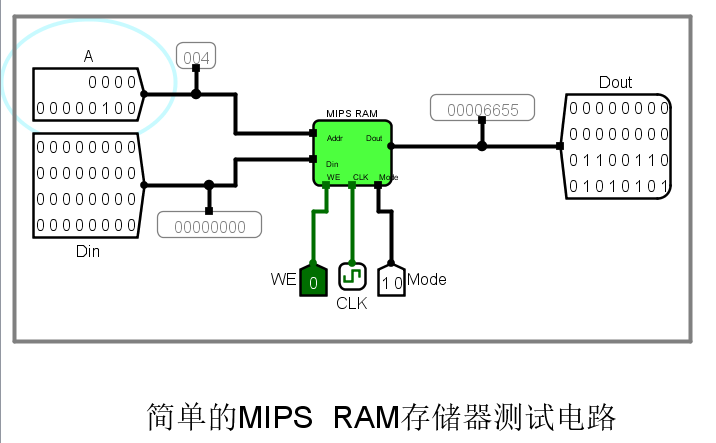




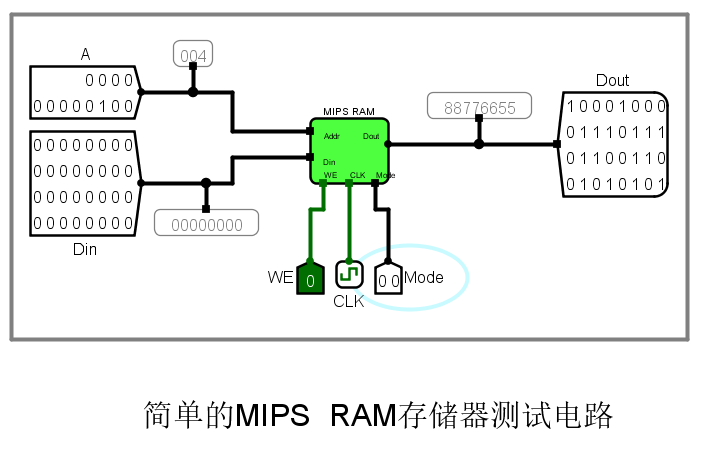
第2步：按字节访问方式读



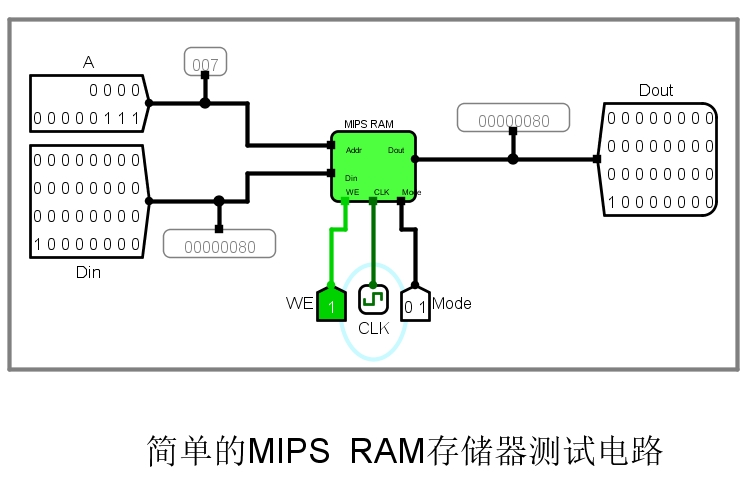
第3步：按半节访问方式读



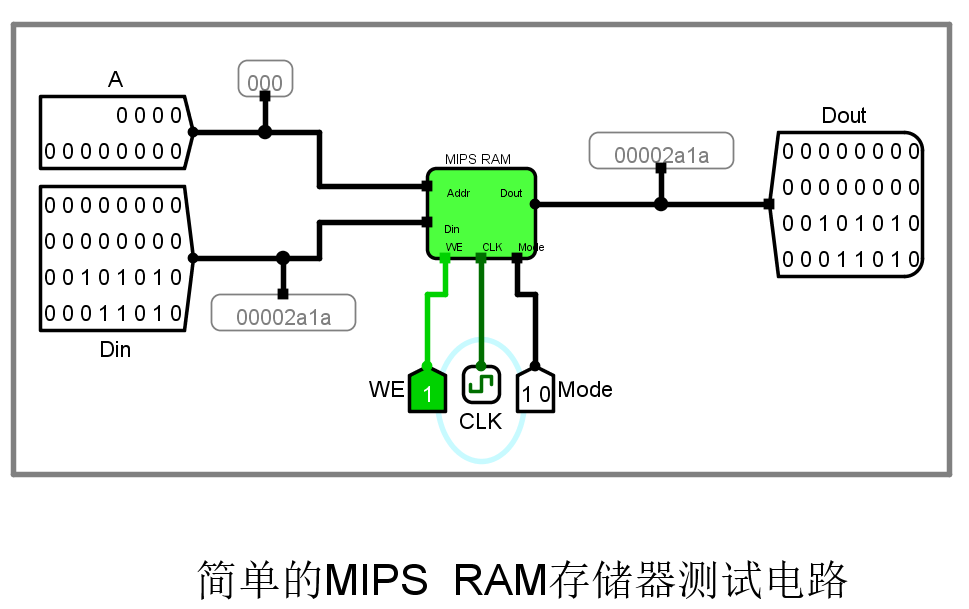
第4步：按字访问方式读



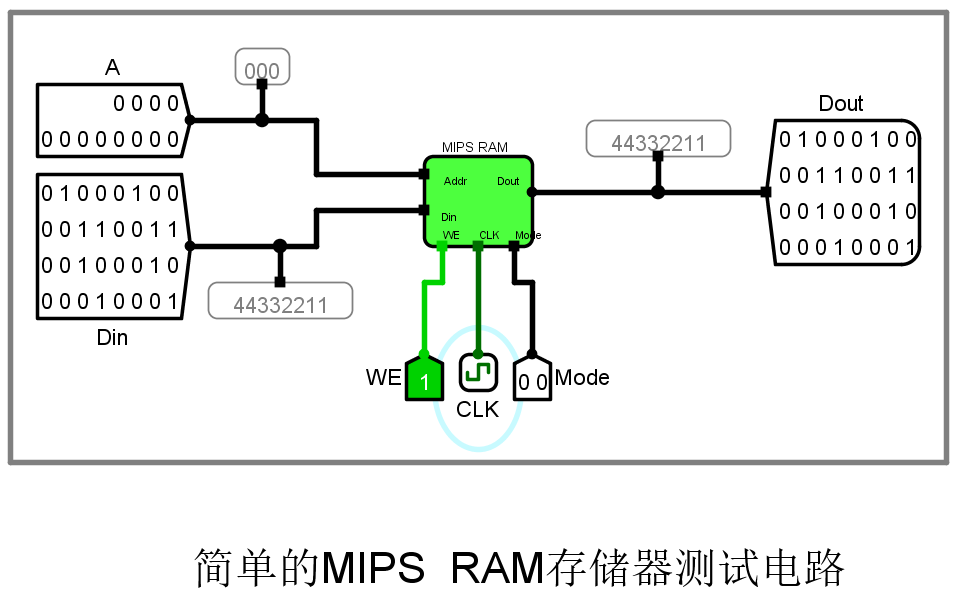
第5步：按字节访问方式写



第6步：按半字访问方式写

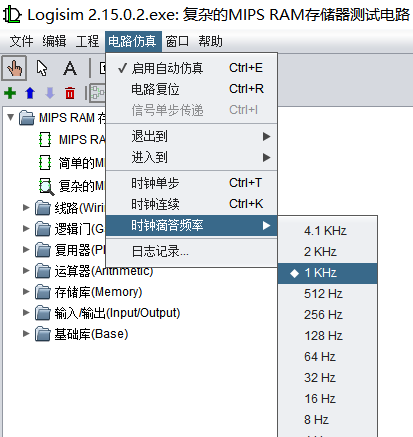


第7步：按字访问方式写



3、复杂的MIPS RAM 存储器测试电路

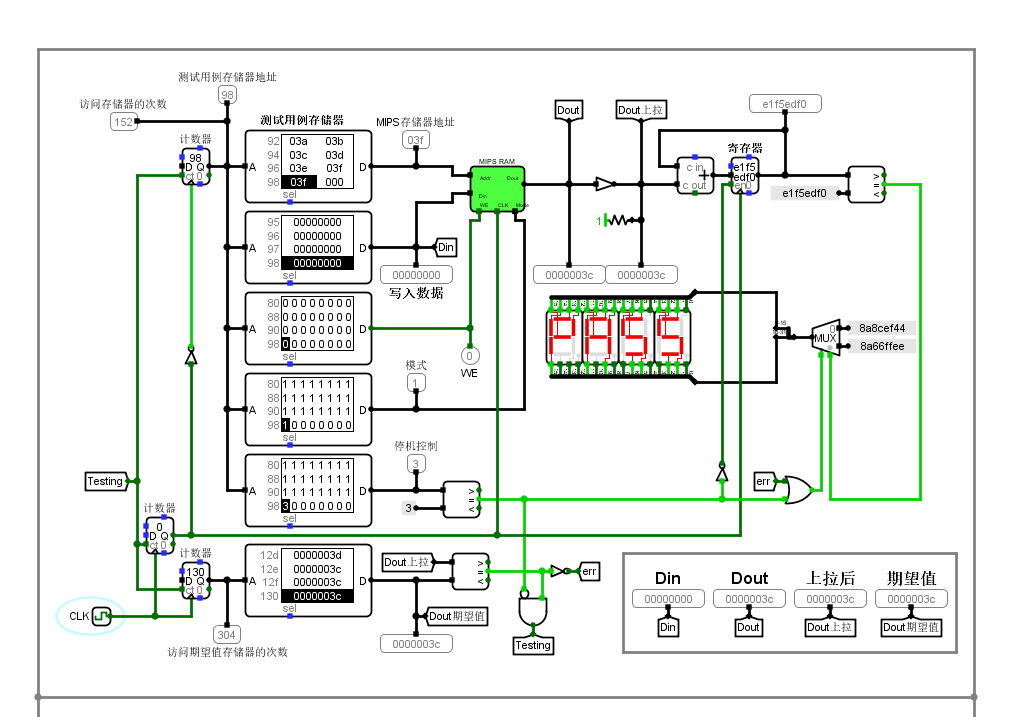
设置时钟频率=1KHz



按Ctrl+R，使电路复位

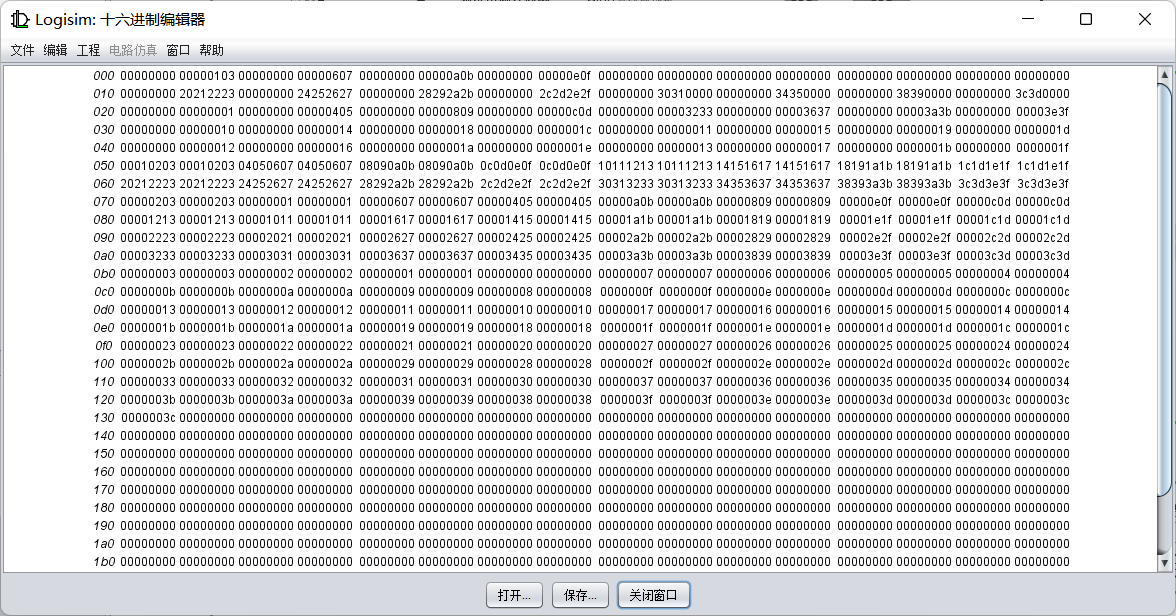
按Ctrl+K，启动时钟

如果显示“PASS”，表示测试通过，如果显示“FAIL”，表示测试失败

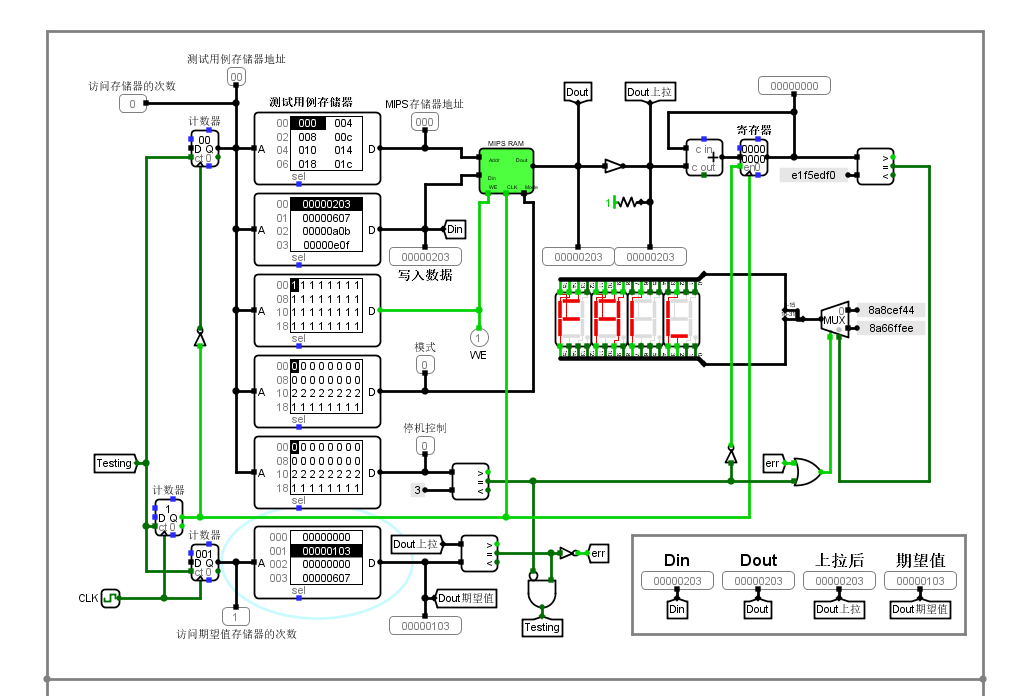


再按Ctrl+K，则停止时钟

修改“期望值”存储器的内容

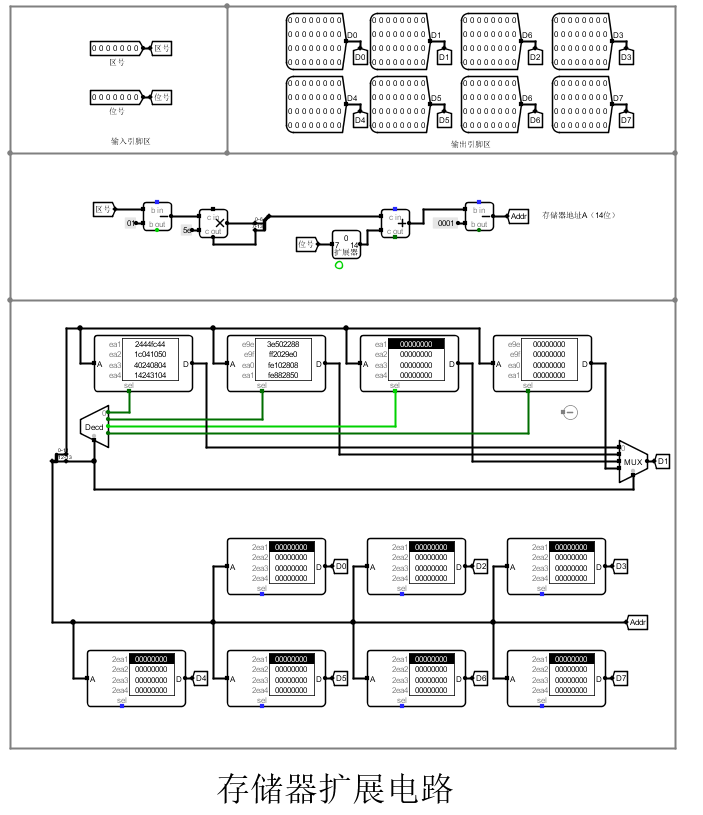


再按Ctrl+K，启动时钟，运行结束后，将显示“FAIL”

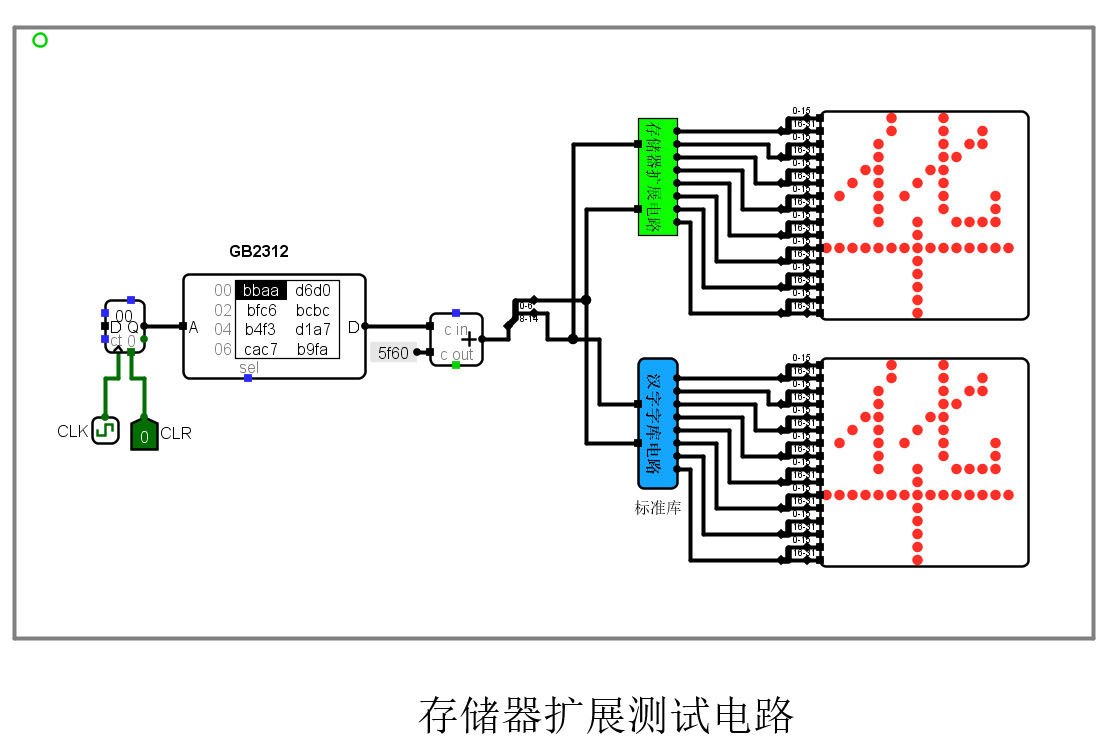


三、存储器扩展设计实验

1、存储器扩展电路（设计实验）



1. 存储器扩展测试电路



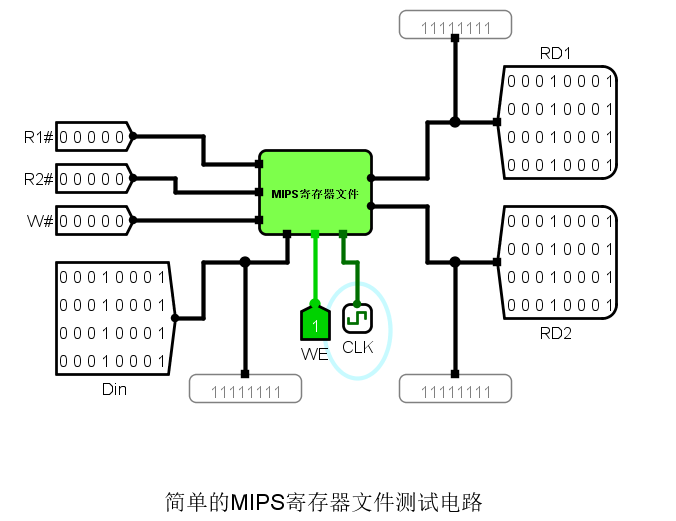
四、MIPS 寄存器文件验证实验

1、MIPS 寄存器文件电路

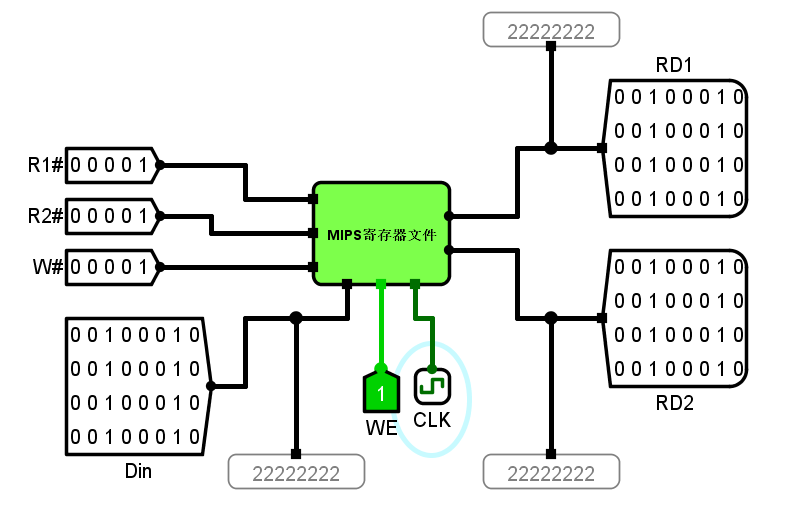
2、简单的MIPS 寄存器文件测试电路

第1步：向寄存器0～寄存器3，分别写入11111111、22222222、33333333、44444444（均为十六进制，下同）

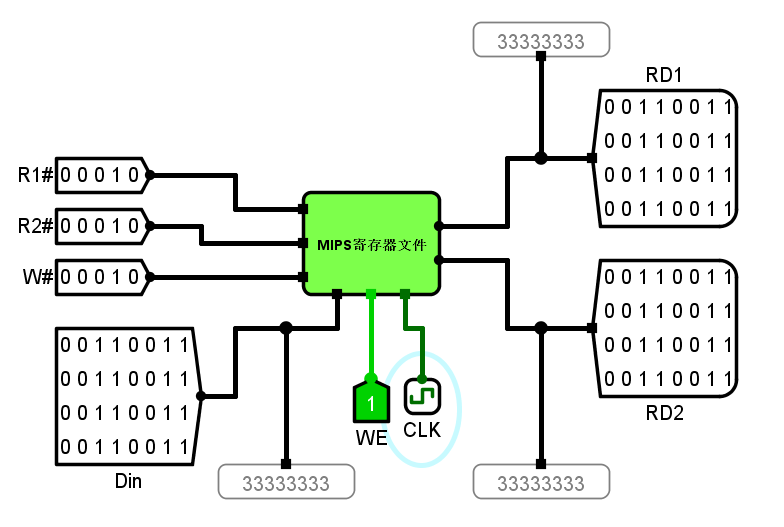
置W#=00000，Din=11111111H，WE=1，按时钟CLK按钮2次



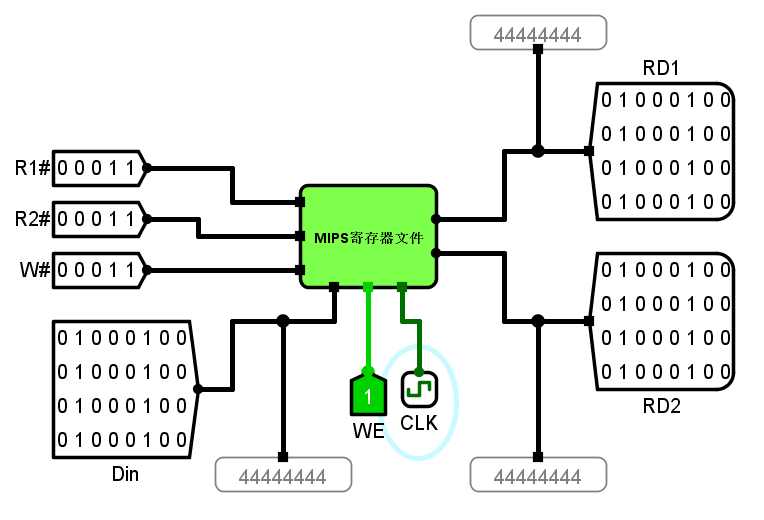
置W#=00001，Din=22222222H，WE=1，按时钟CLK按钮2次



置W#=00010，Din=33333333H，WE=1，按时钟CLK按钮2次



置W#=00011，Din=44444444H，WE=1，按时钟CLK按钮2次

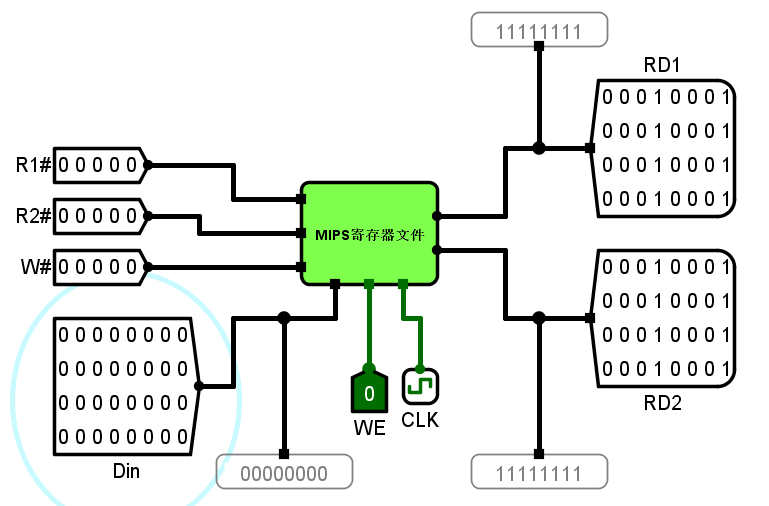


第2步：从RD1口读寄存器0～寄存器3

第3步：从RD2口读寄存器0～寄存器3

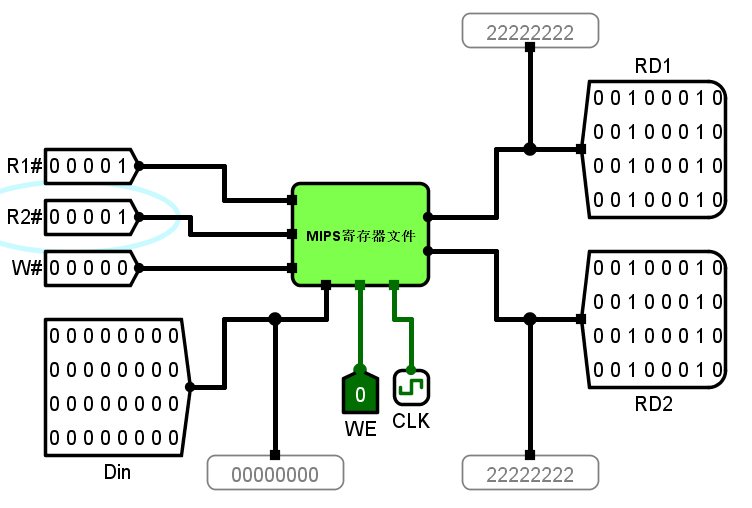
置R1#=00000，WE=0，观察RD1是否=11111111H？是

置R2#=00000，WE=0，观察RD2是否=11111111H？是



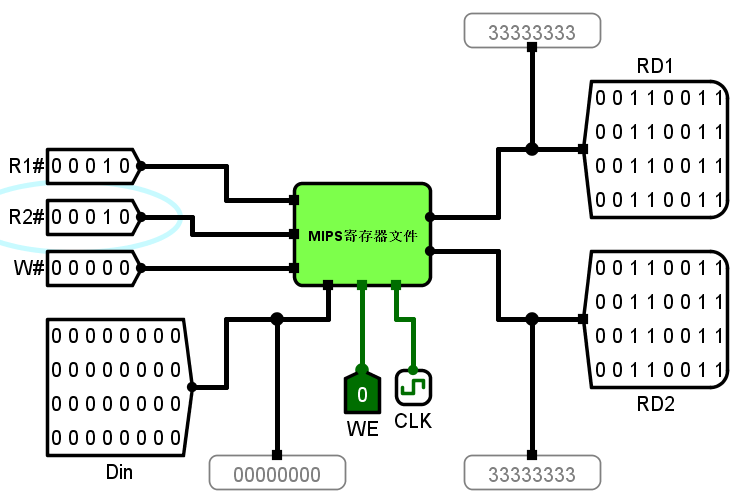
置R1#=00001，WE=0，观察RD1是否=22222222H？是

置R2#=00001，WE=0，观察RD2是否=22222222H？是



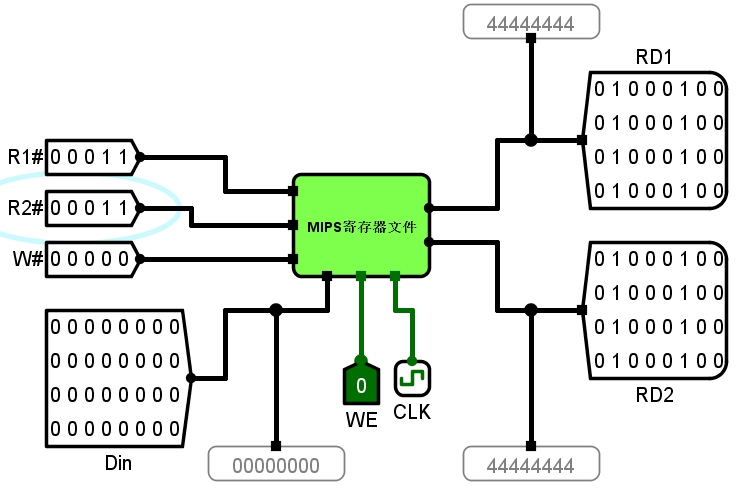
置R1#=00010，WE=0，观察RD1是否=33333333H？是

置R2#=00010，WE=0，观察RD2是否=33333333H？是



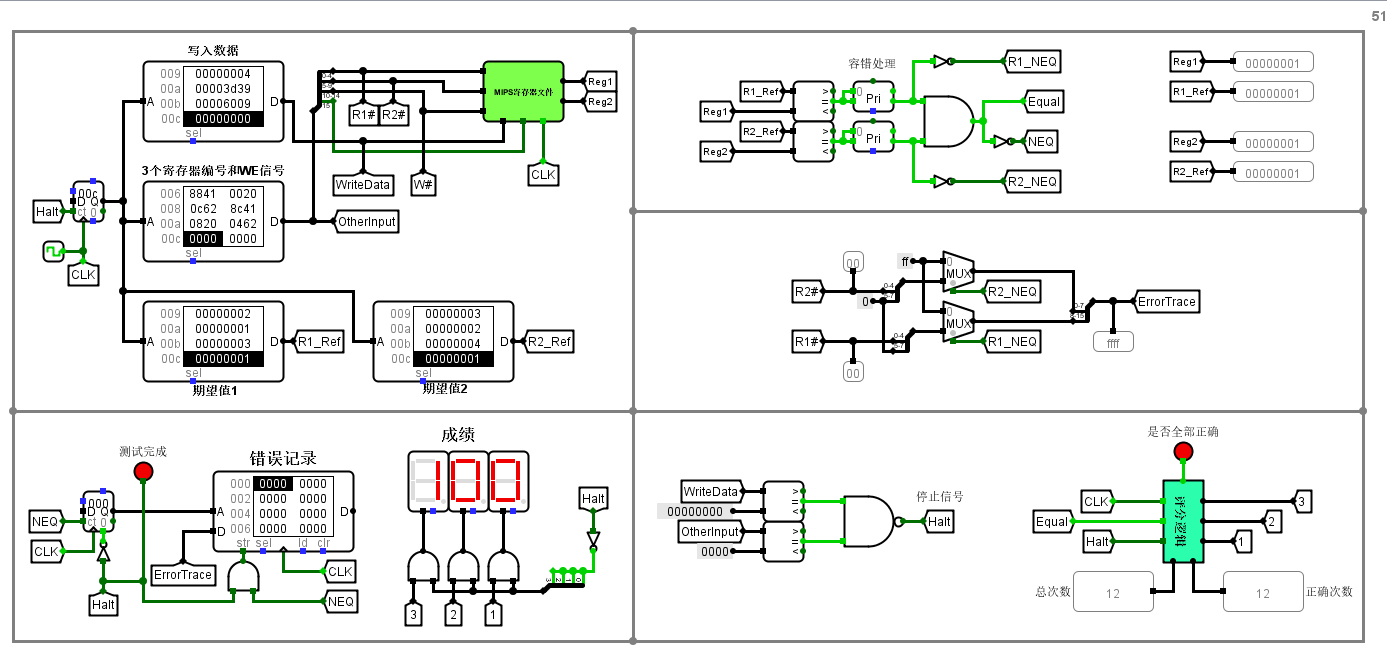
置R1#=00011，WE=0，观察RD1是否=44444444H？是

置R2#=00011，WE=0，观察RD2是否=44444444H？是



1. 复杂的MIPS 寄存器文件测试电路

测试步骤：先按Ctrl+R，使电路复位；然后按Ctrl+K，启动时钟；如果显示“100”，则表示测试通过

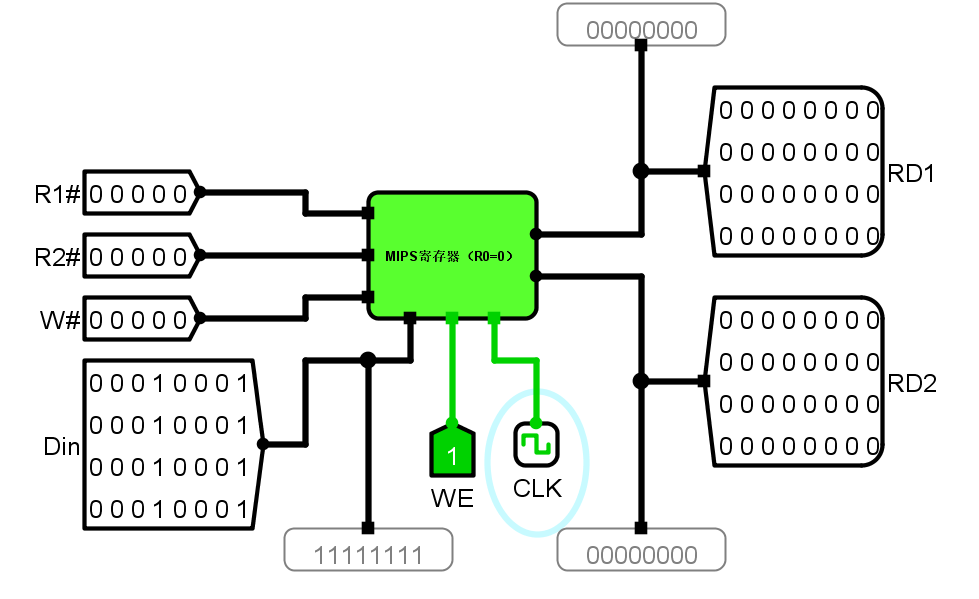


4、MIPS 寄存器文件电路（R0=0）

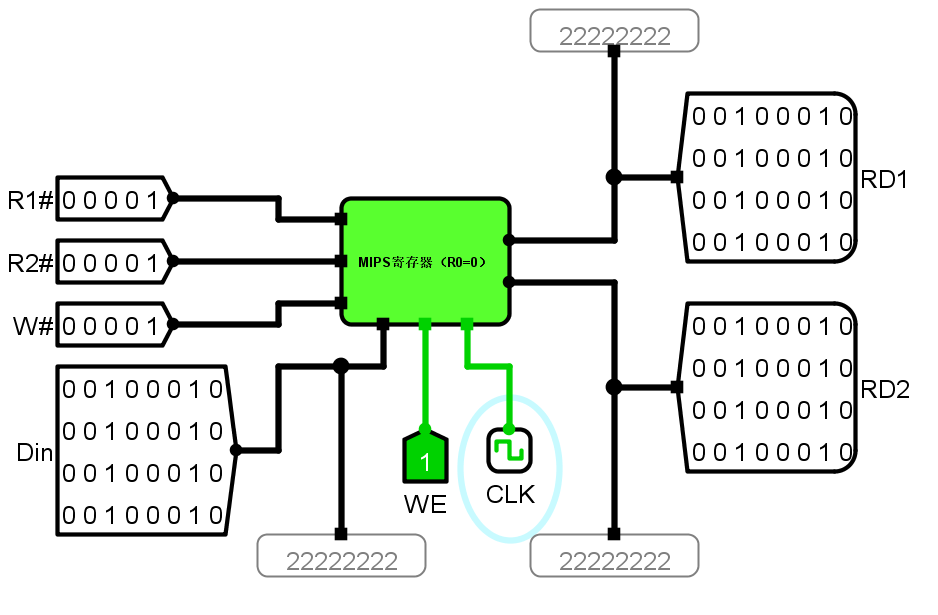
5、简单的MIPS 寄存器文件测试电路（R0=0）

第1步：向寄存器0～寄存器3，分别写入11111111、22222222、33333333、44444444（均为十六进制，下同）

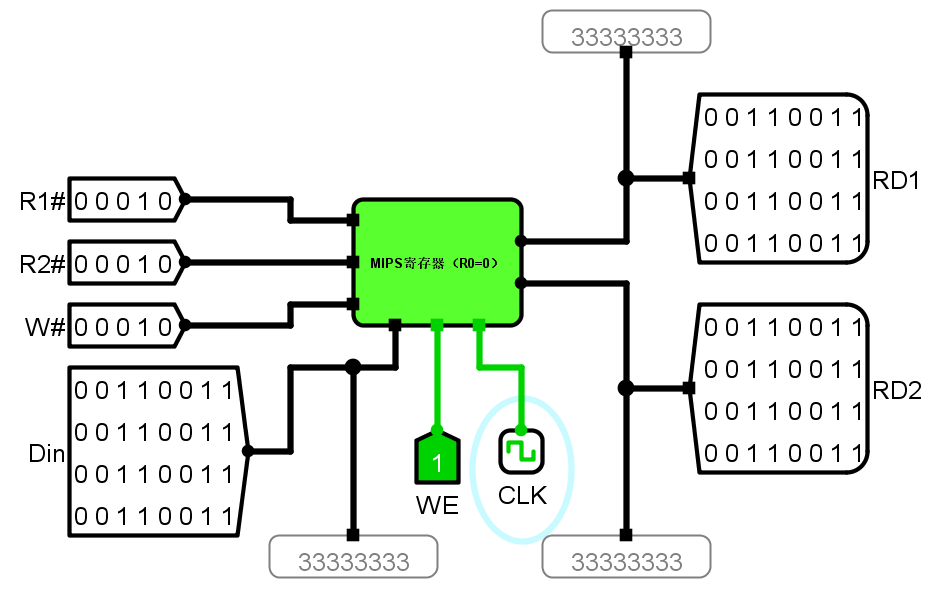
置W#=00000，Din=11111111H，WE=1，按时钟CLK按钮2次



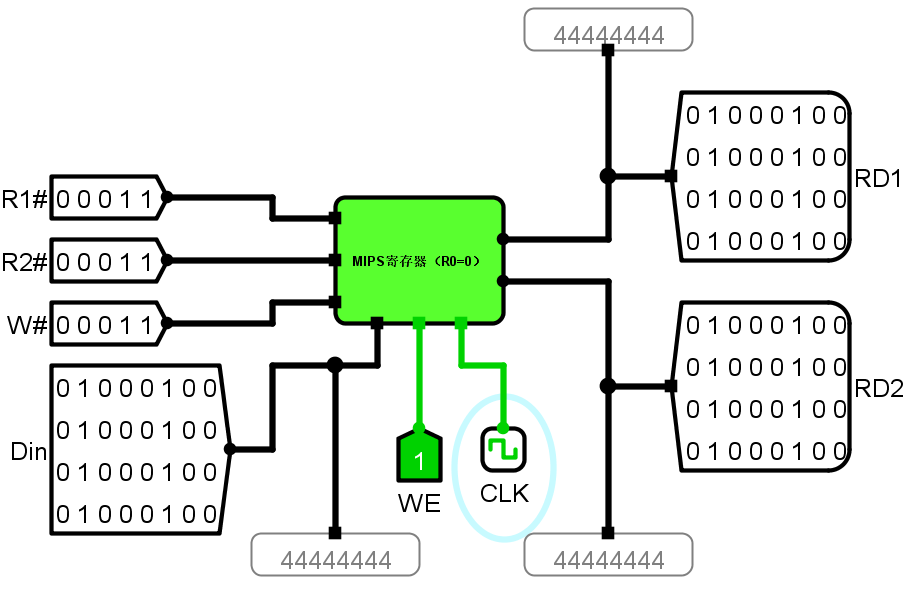
置W#=00001，Din=22222222H，WE=1，按时钟CLK按钮2次



置W#=00010，Din=33333333H，WE=1，按时钟CLK按钮2次



置W#=00011，Din=44444444H，WE=1，按时钟CLK按钮2次

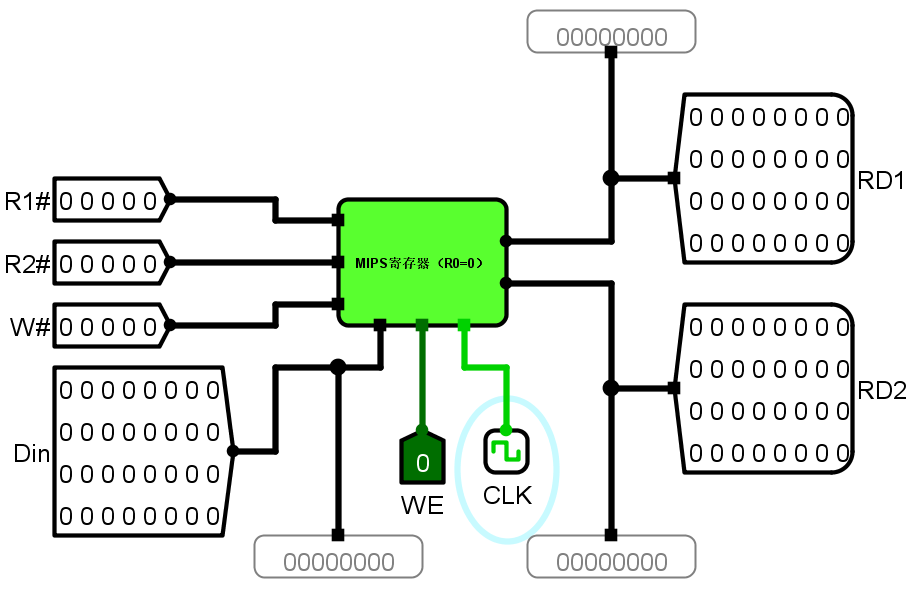


第2步：从RD1口读寄存器0～寄存器3

第3步：从RD2口读寄存器0～寄存器3

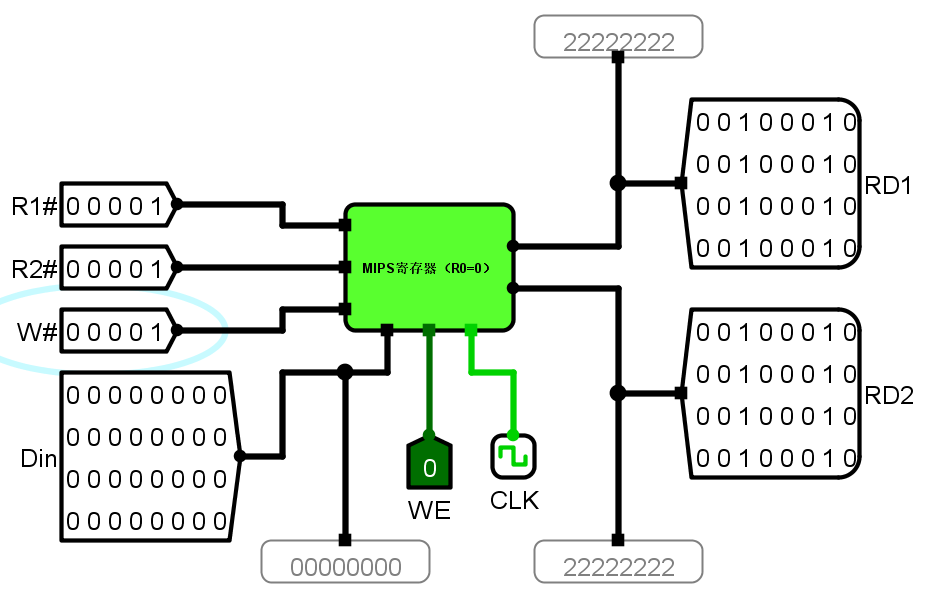
置R1#=00000，WE=0，观察RD1是否=00000000H？（虽然前面向R0中写入了11111111H，但是这里读出的R0仍然为0）是

置R2#=00000，WE=0，观察RD2是否=00000000H？（虽然前面向R0中写入了11111111H，但是这里读出的R0仍然为0）是



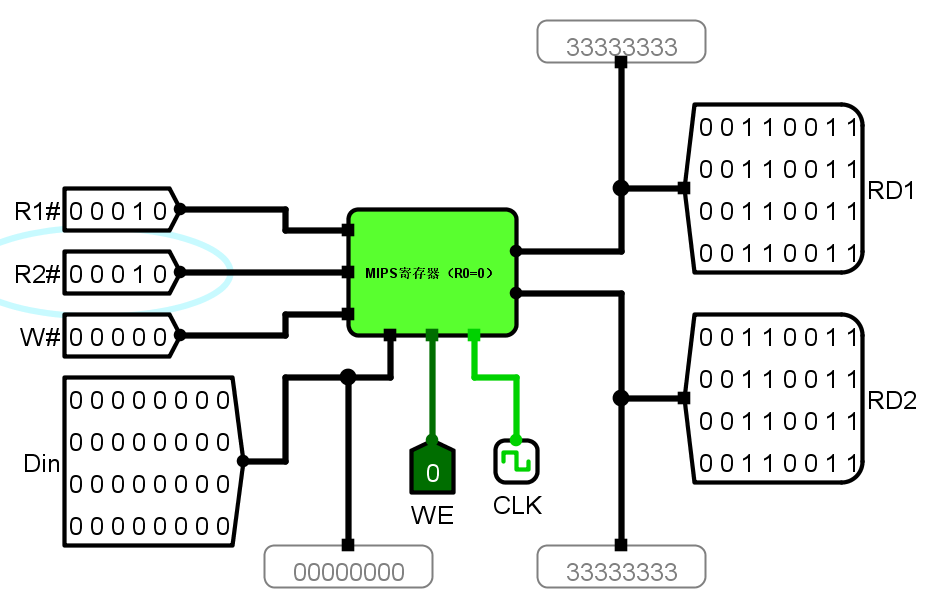
置R1#=00001，WE=0，观察RD1是否=22222222H？是

置R2#=00001，WE=0，观察RD2是否=22222222H？是



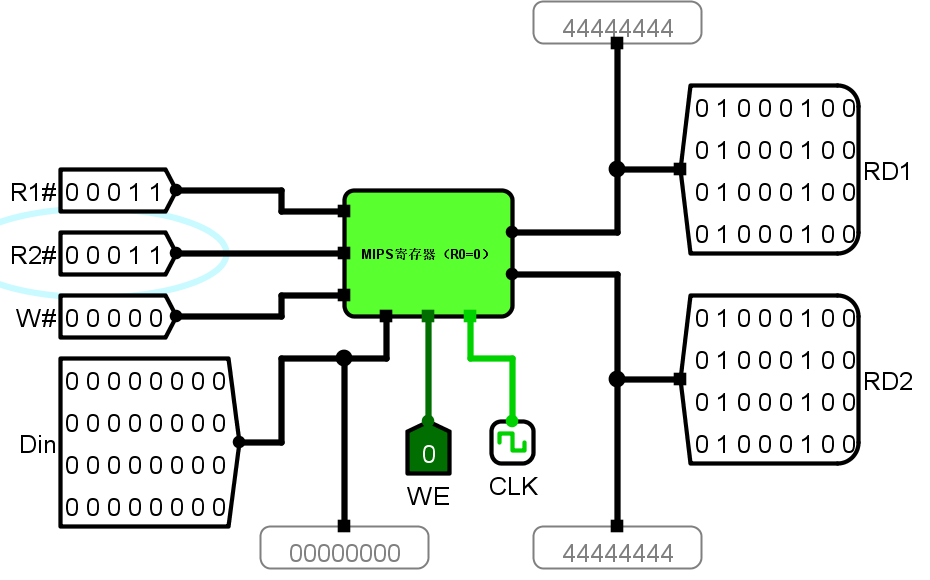
置R1#=00010，WE=0，观察RD1是否=33333333H？是

置R2#=00010，WE=0，观察RD2是否=33333333H？是



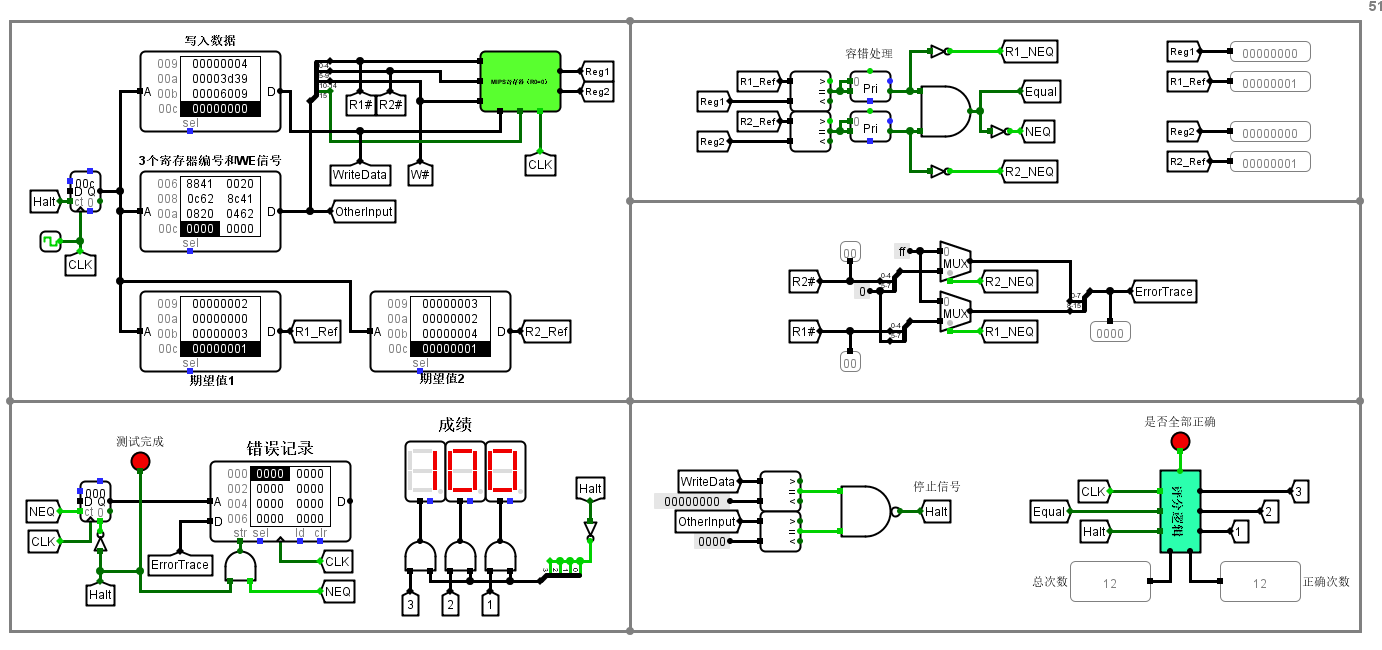
置R1#=00011，WE=0，观察RD1是否=44444444H？是

置R2#=00011，WE=0，观察RD2是否=44444444H？是

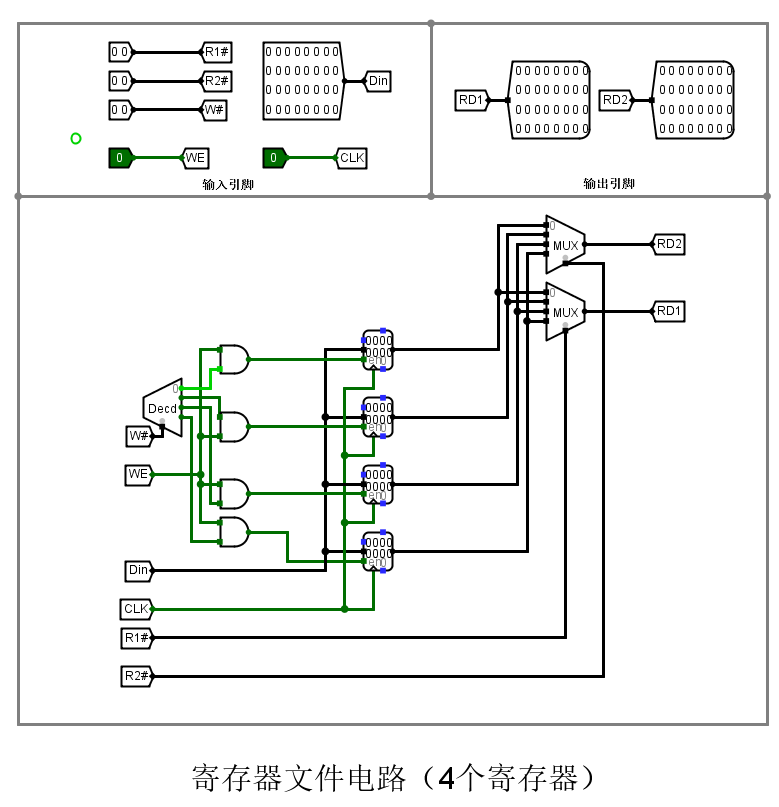


1. 复杂的MIPS 寄存器文件测试电路（R0=0）

测试步骤：先按Ctrl+R，使电路复位；然后按Ctrl+K，启动时钟；如果显示“100”，表示测试通过

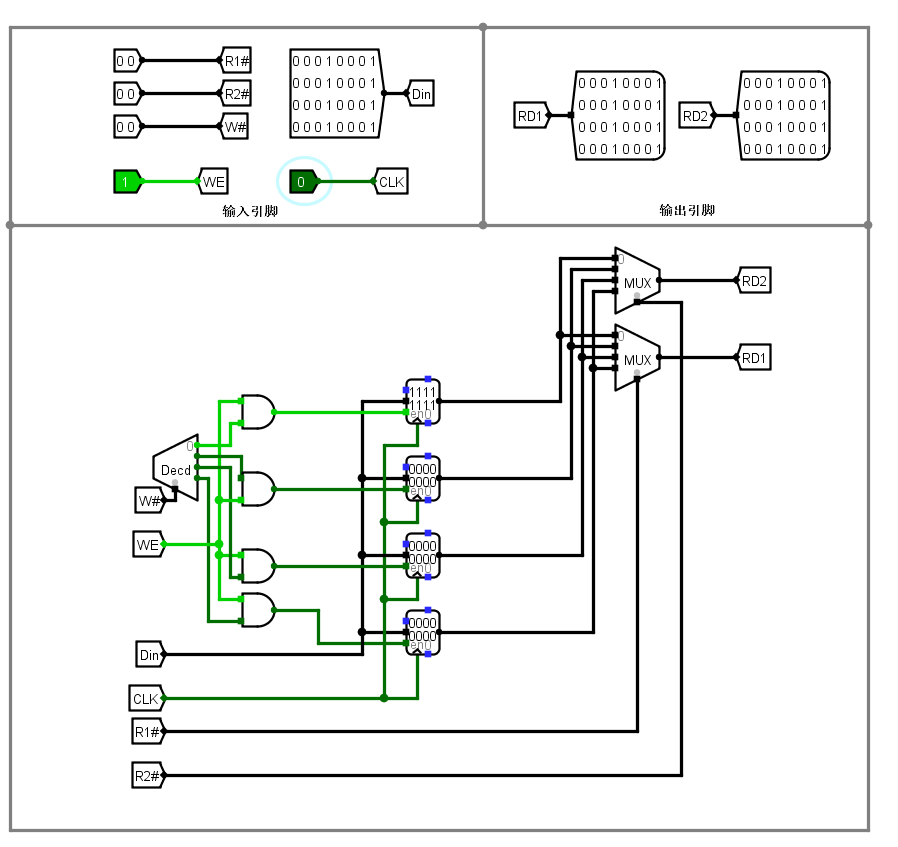


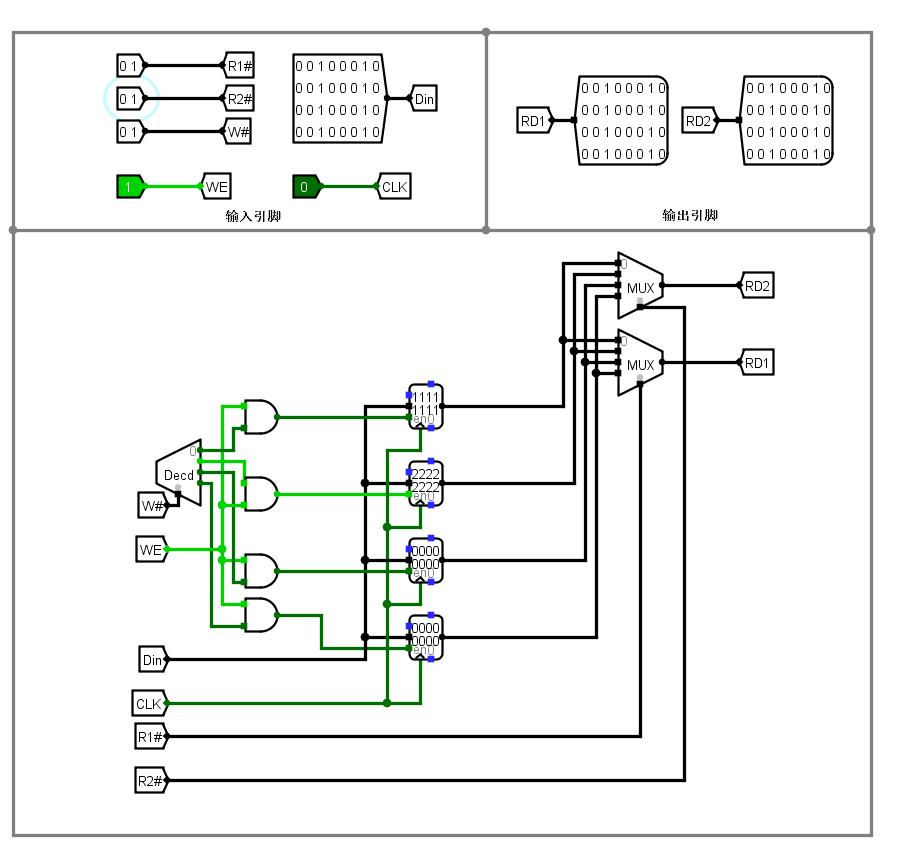
1. 寄存器文件设计实验
2. 寄存器文件电路（4个寄存器）（设计实验）



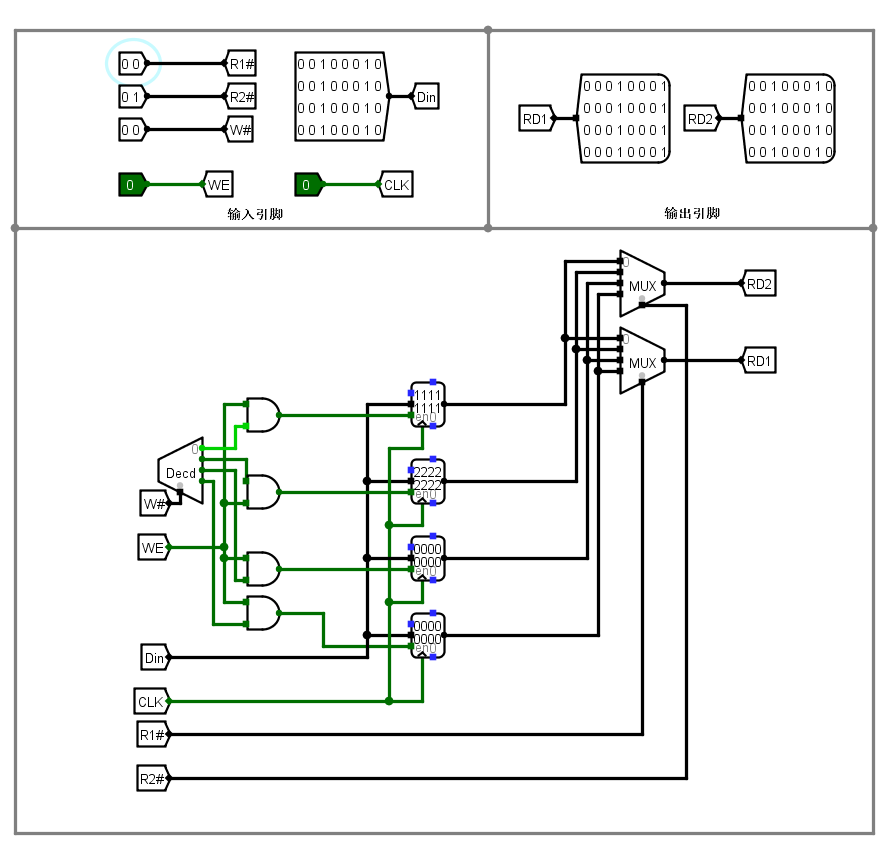
验证：

1. 向不同的寄存器中写入内容（向R0写入11111111，向R1写入22222222）

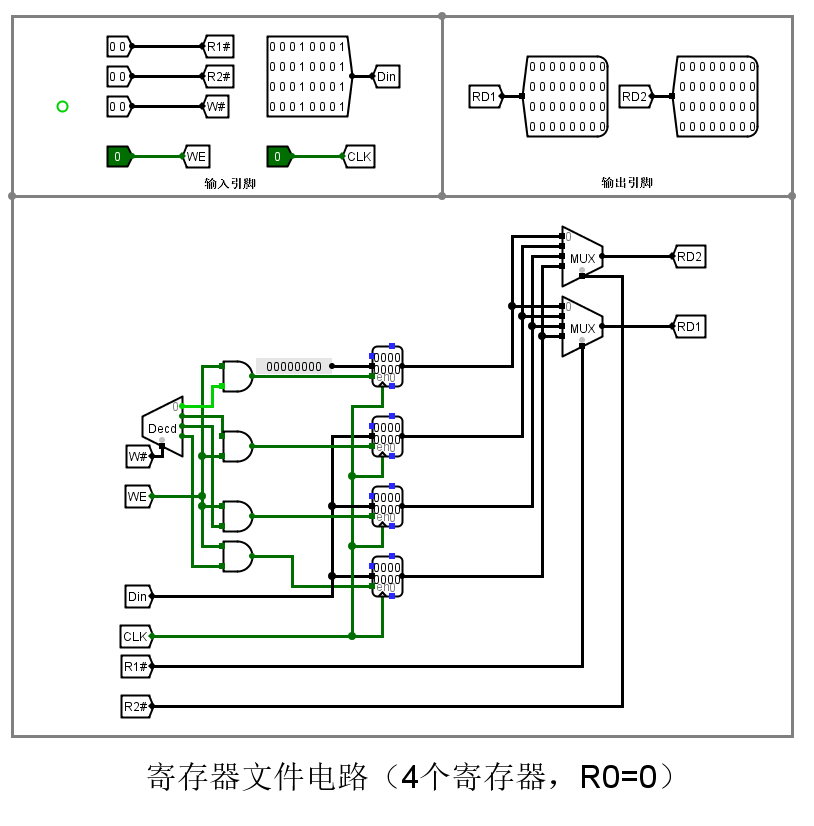




1. 分别从两个不同的端口读出第①步写入的内容（RD1读取R0,RD2读取R1）

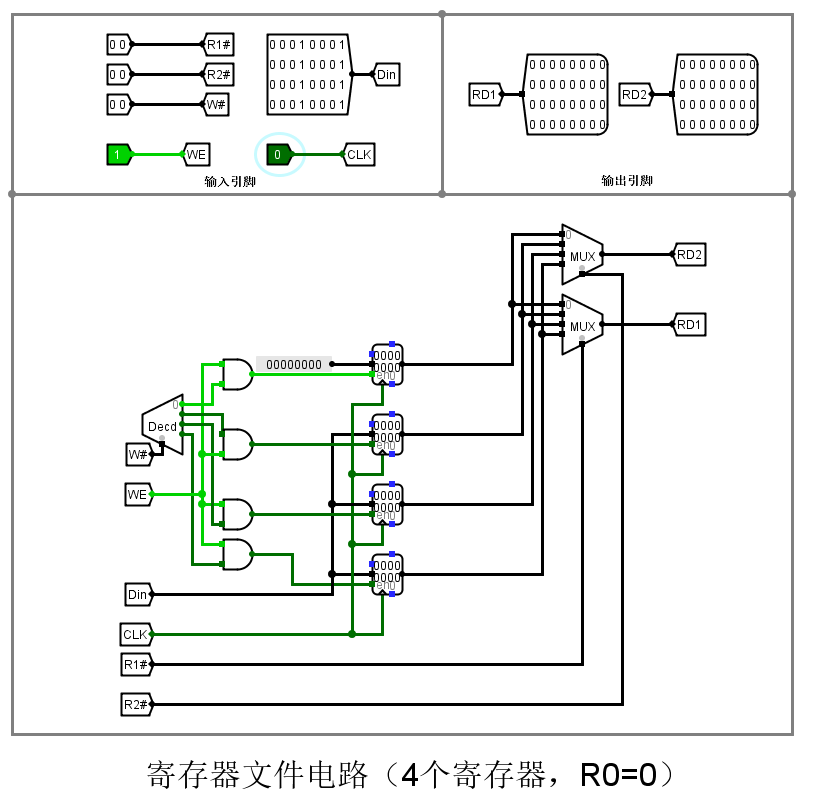


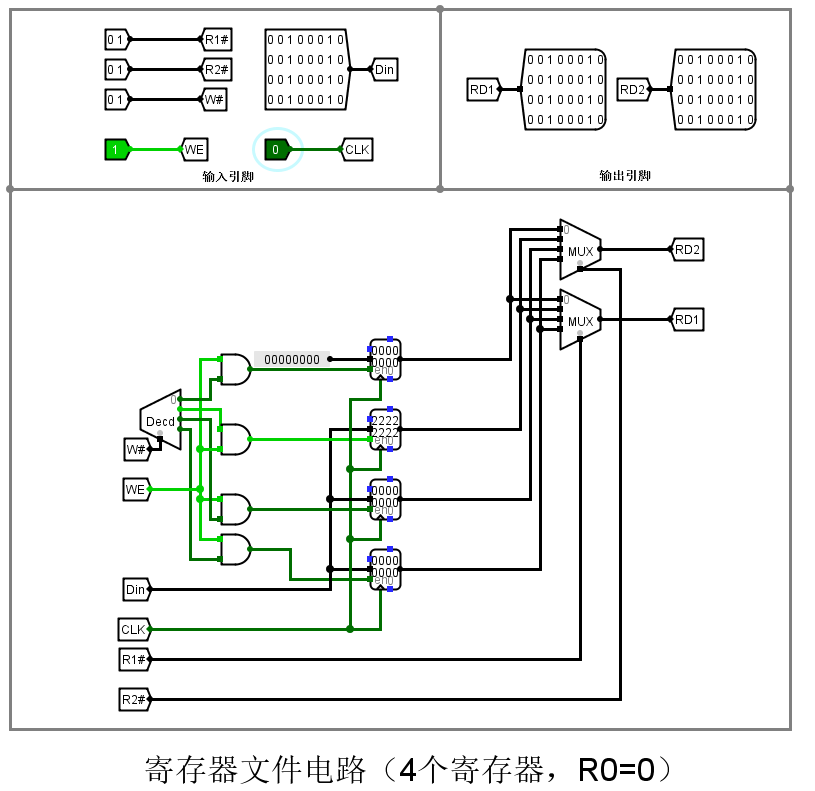
1. 寄存器文件电路（4个寄存器，R0=0）（设计实验）



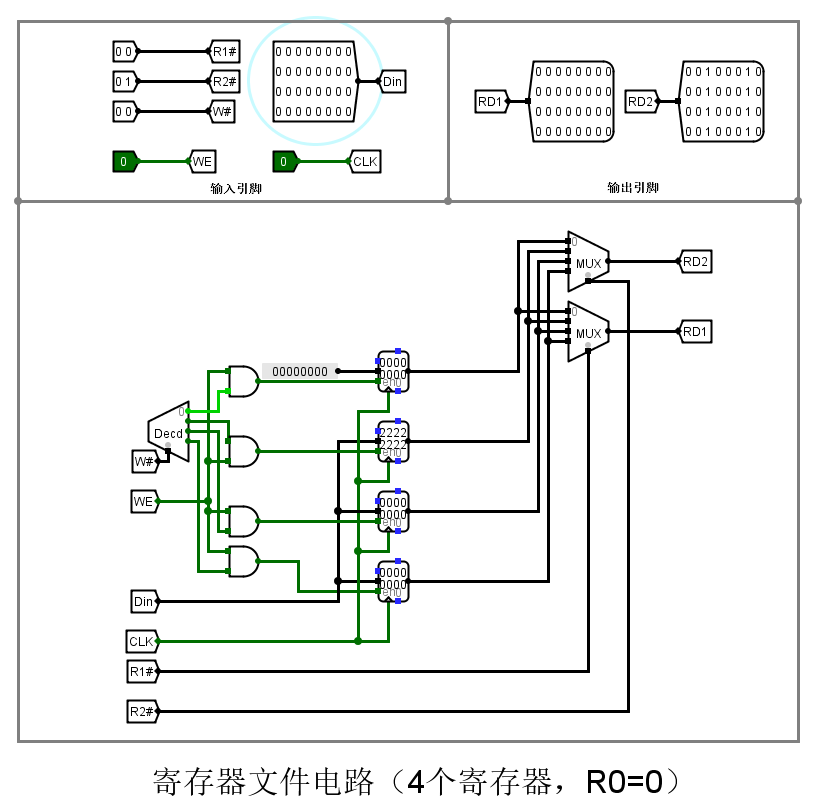
验证：

①向不同的寄存器中写入内容（向R0写入11111111，向R1写入22222222）





②分别从两个不同的端口读出第①步写入的内容（RD1读取R0,RD2读取R1）



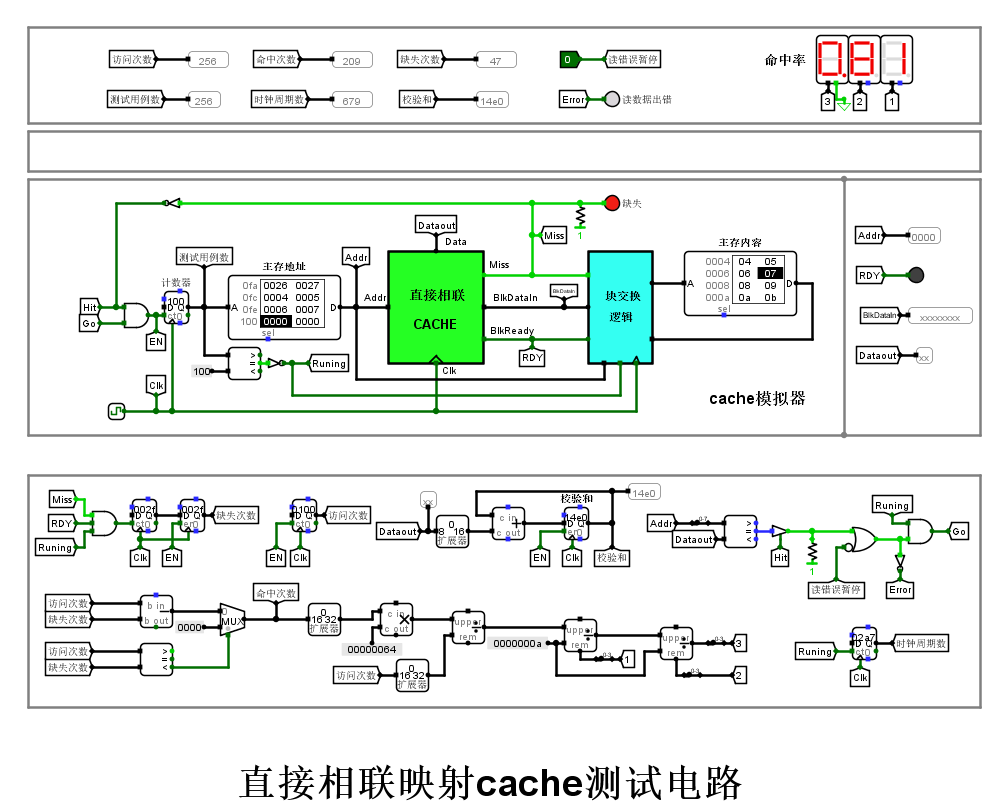
1. 验证R0是否恒等于0？是

六、cache 验证实验

1、直接相联映射方式 cache 电路

2、直接相联映射方式 cache 测试电路

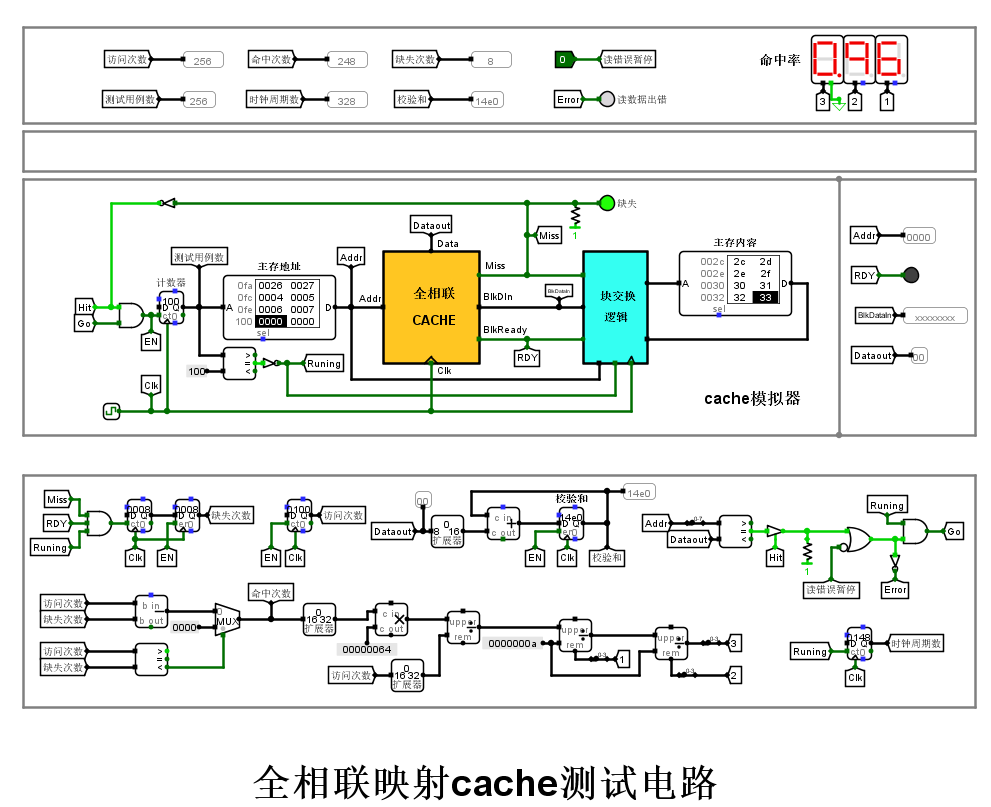
测试步骤：先按Ctrl+R，使电路复位；然后按Ctrl+K，启动时钟；等到命中率数字不变，表示表示测试完成



3、全相联映射方式 cache 电路

4、全相联映射方式 cache 测试电路

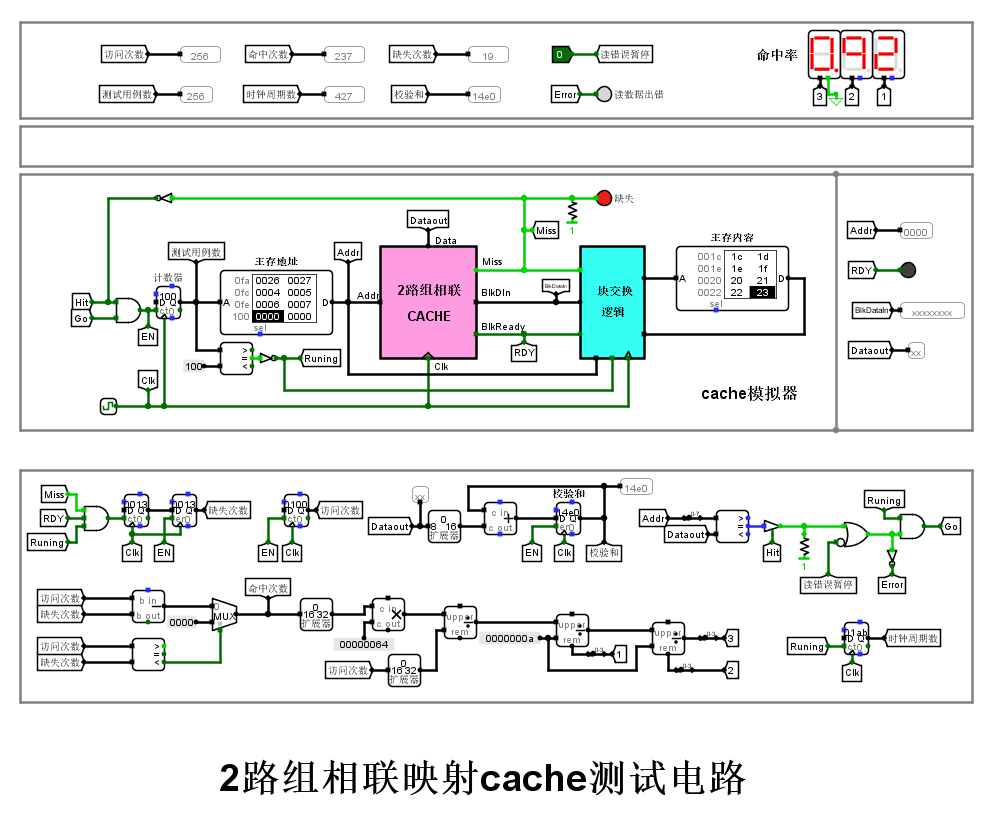
测试步骤：先按Ctrl+R，使电路复位；然后按Ctrl+K，启动时钟；等到命中率数字不变，表示表示测试完成



5、2路组相联映射方式 cache 电路

6、2路组相联映射方式 cache 测试电路

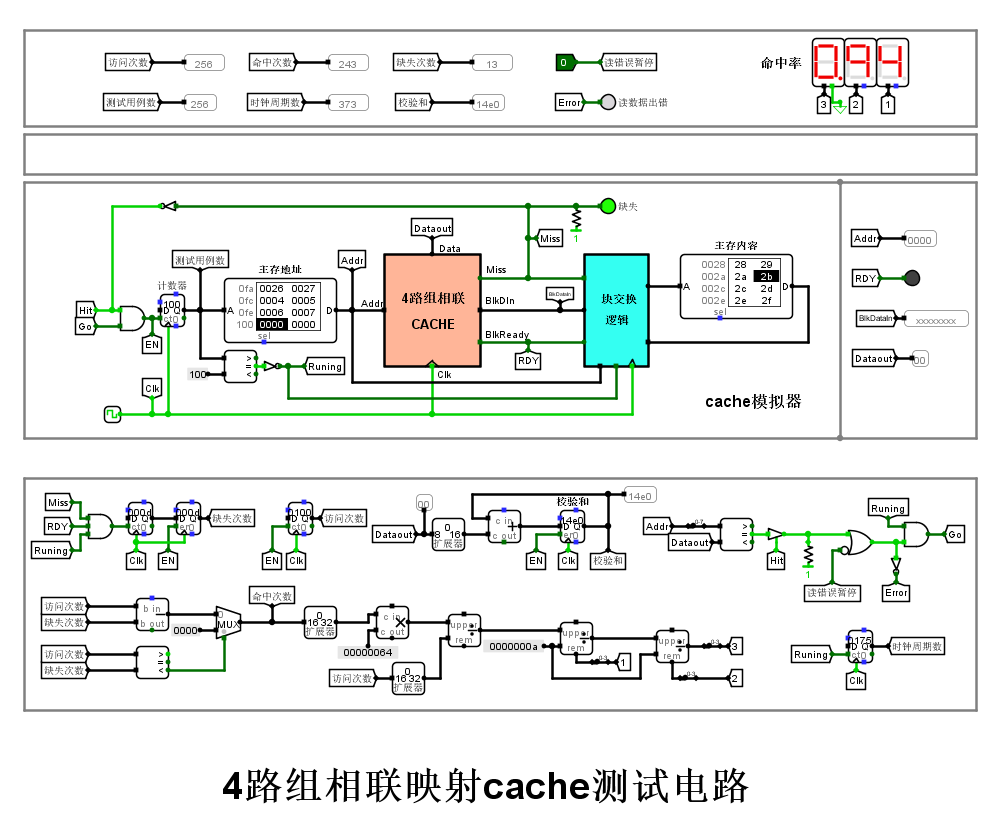
测试步骤：先按Ctrl+R，使电路复位；然后按Ctrl+K，启动时钟；等到命中率数字不变，表示表示测试完成



7、4路组相联映射方式 cache 电路

8、4路组相联映射方式 cache 测试电路

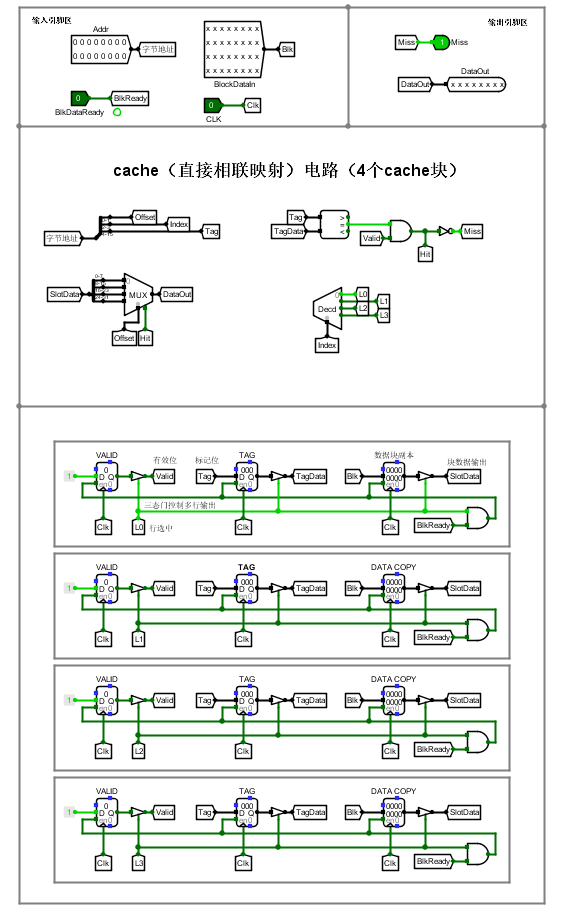
测试步骤：先按Ctrl+R，使电路复位；然后按Ctrl+K，启动时钟；等到命中率数字不变，表示表示测试完成



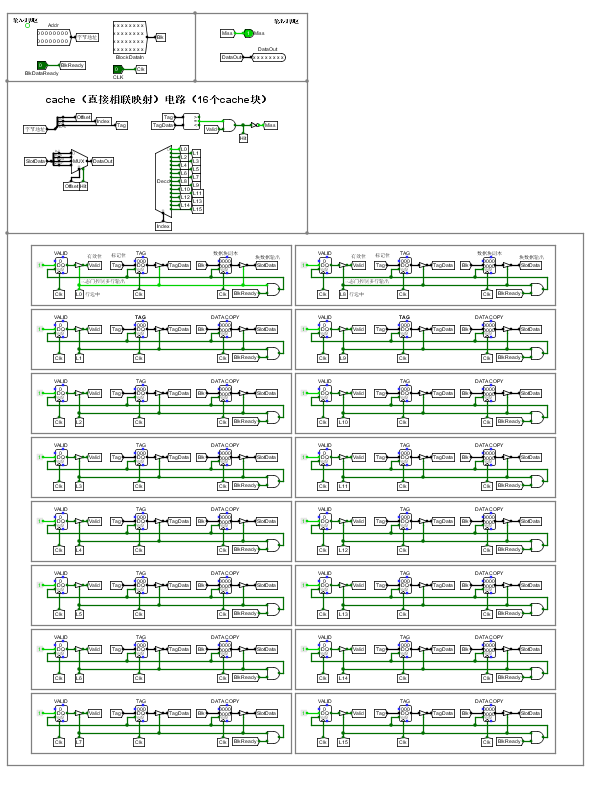
七、cache 设计实验

请同学们从以下4组中任选1组，设计相应的电路（挑战性实验）

1. 直接相联映射方式 cache 电路（4个cache块）



1. 直接相联映射方式 cache 电路（16个cache块）

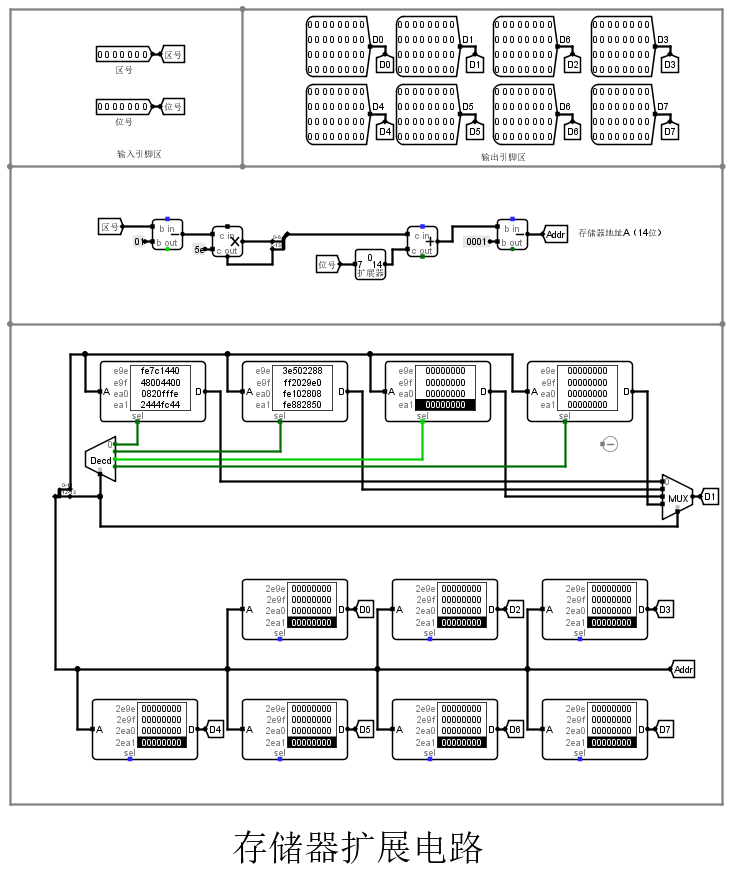


* 1. **课后完成部分（设计实验的内容）**

1. **存储器扩展电路（设计实验）——文件名：存储器扩展电路.circ**

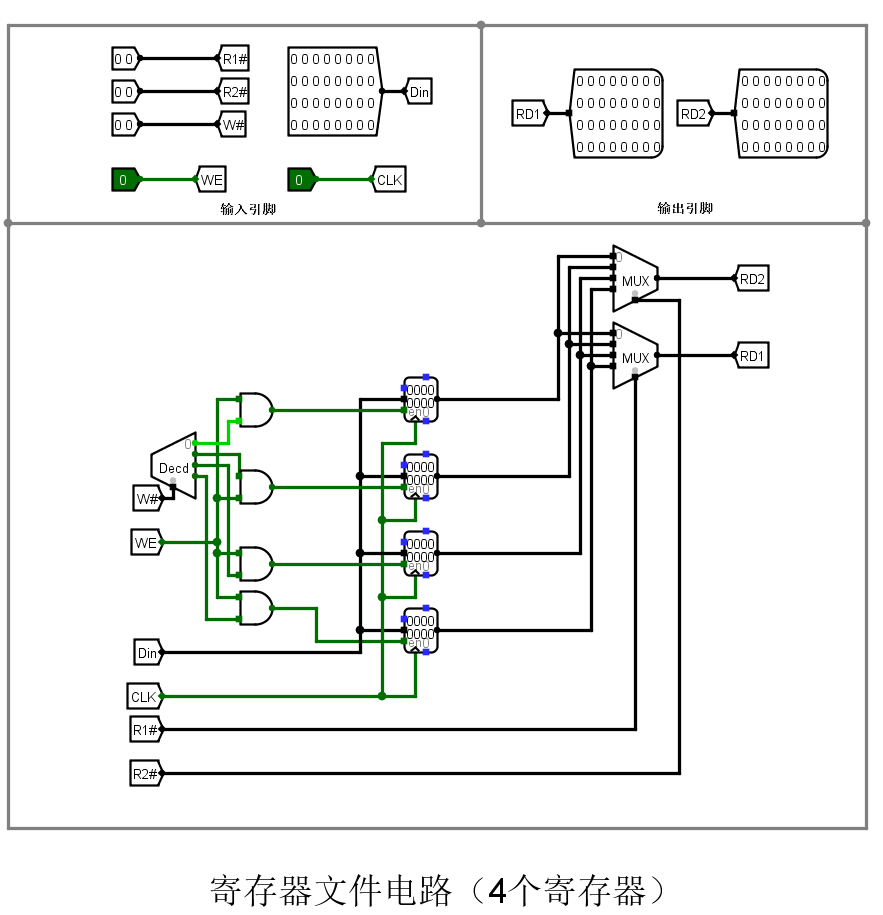
**实现思路：用4片4k\*32为的储存器采用字拓展拓展为等效1片16k\*32位储存器；再将该储存器与剩下7个16k\*32位储存器采用位拓展拓展为等效1片16k\*256位储存器。**

**其中4片4k\*32位的字拓展需要一个多路选择器以及一个译码器，用14位地址线的高2位进行选择不同的储存器。**



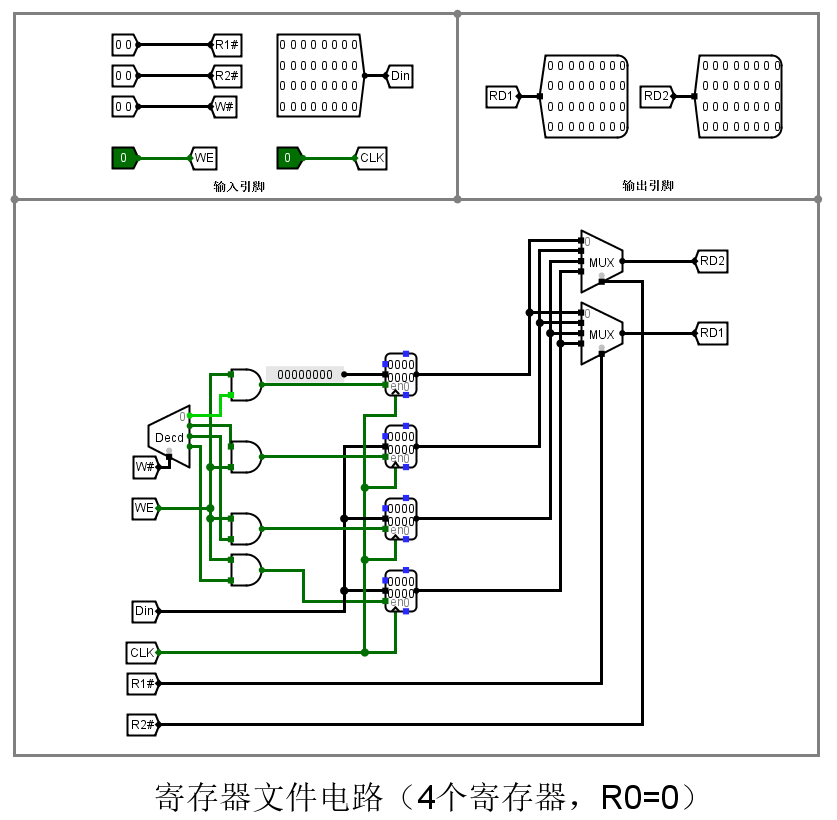
1. **寄存器文件电路（4个寄存器）（设计实验） ——文件名：寄存器文件电路（4个寄存器）.circ**

**实现思路：用译码器实现2位的W#选择4个寄存器进行输入，将WE的信号与W#的相与以实现写模式的开关。Din与四个寄存器输入端相连即可。R1#与R2#通过多路选择器以从四个寄存器中选择一个读出。**



1. **寄存器文件电路（4个寄存器，R0=0）（设计实验） ——文件名：寄存器文件电路（4个寄存器，R0=0）.circ**

**实现思路：将上个电路的第一个寄存器的输入端与常量0相连，因此实际上该寄存器可以输入，只不过输入的值恒为0。其余不变。**



1. **cache 设计实验（挑战性实验，从4组中任选1组） ——文件名：cache 设计实验.circ**

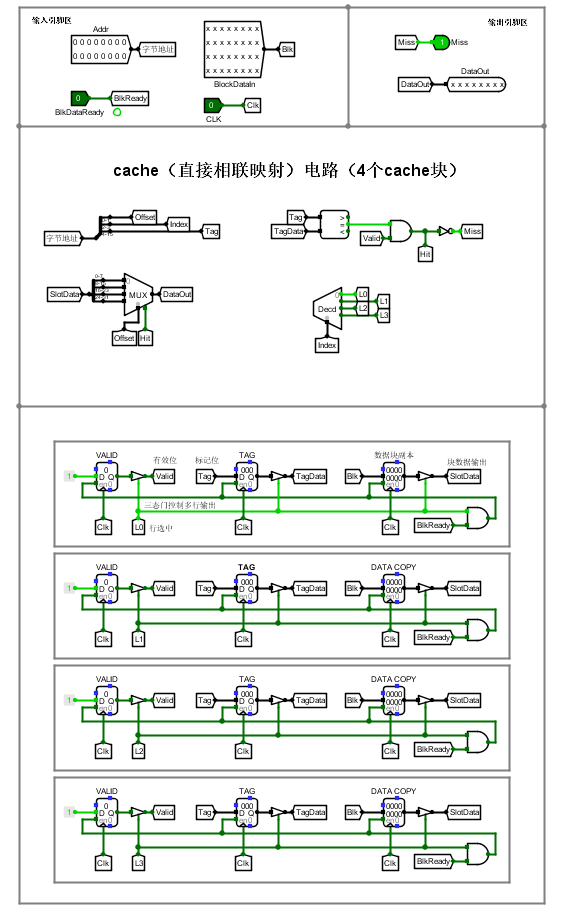
**实现思路：**

**1.4块：将8个cache电路的行索引index从3位改为2位，区地址tag改为12位，其余电路随之更改。**

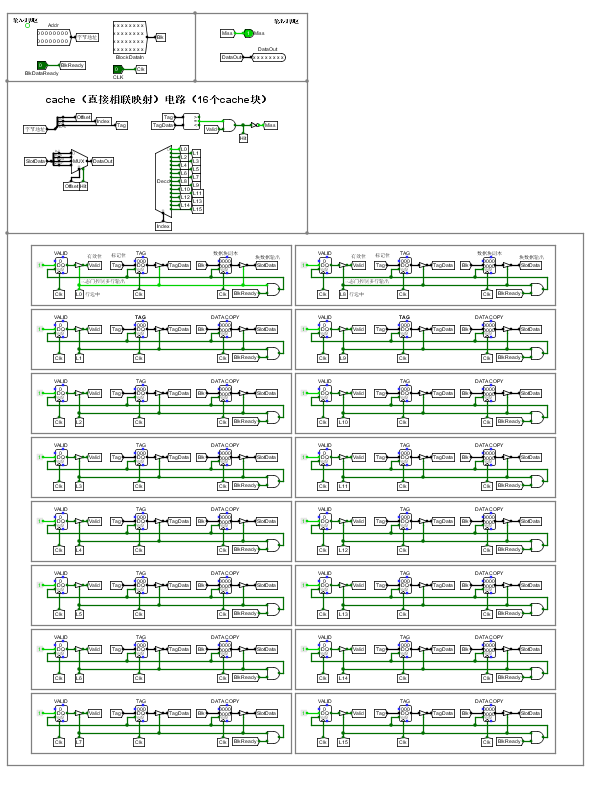
**2.16块：将8个cache电路的行索引index从3位改为4位，区地址tag改为10位，其余电路随之更改。**

**选择第一组：**

**1、直接相联映射方式 cache 电路（4个cache块）**



**2、直接相联映射方式 cache 电路（16个cache块）**



**请将上述设计实验（挑战性实验）的运行结果，以屏幕拷贝形式保存，黏贴在实验报告中，并给出相关的文字说明。**