**计算机组成原理**

**--实验报告**

**专业班级： 人工智能2204**

**学 号： U202215150**

**姓 名： 王子旭**

**实验名称： 存储器实验**

**实验时间： 2024年5月23日**

## 1、实验目的

1.1、存储器扩展实验

完成storage.circ中字库电路，并使用字库测试电路进行测试。

1.2、MIPS RAM设计实验

完成storage.circ中MIPS RAM，并使用MIPS RAM测试电路进行测试。

1.3、MIPS 寄存器文件实验

完成storage.circ中MIPS Regfile，并使用MIPS Regfile测试电路进行测试。

1.4、Cache硬件设计实验

完成storage.circ中Cache电路，并使用Cache自动测试电路进行测试。Cache电路可选择直接相联、全相联、2路组相联和4路组相联中任意一种实现。

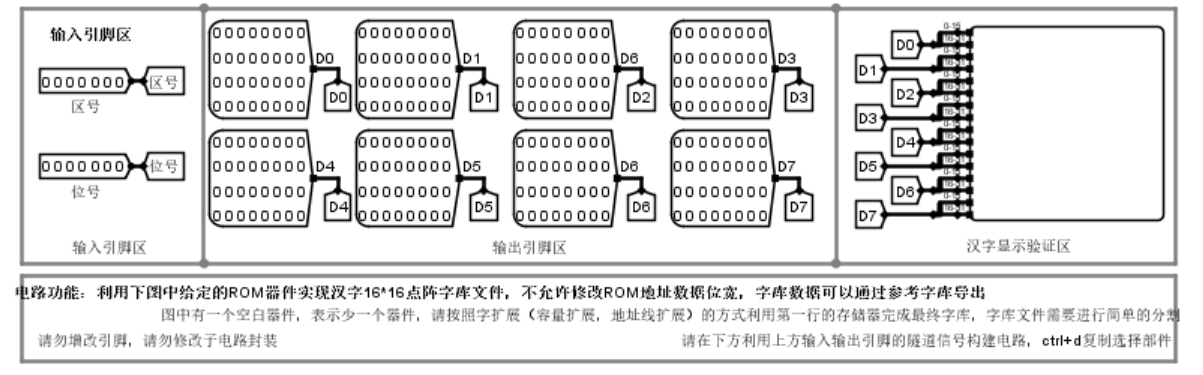
## 2、 实验内容

### 2.1、 汉字字库存储芯片扩展实验

**实验目的：**掌握存储扩展基本原理。

**实验内容：**设计字库文件，利用指定规格存储器进行存储器字扩展。

**实验要求：**现有如下 ROM 部件，4 个 4K\*32 位 ROM，7 个 16K\*32 位 ROM，请构建 GB2312 16\*16 点阵字库存储器电路，电路输入为汉字区号和位号，由于 16\*16 点阵的字模码需要 256 位点阵信息才能显示一个汉字，所以电路输出为 8\*32 位（256 位点阵信息），实验电路 输入输出引脚如下图：



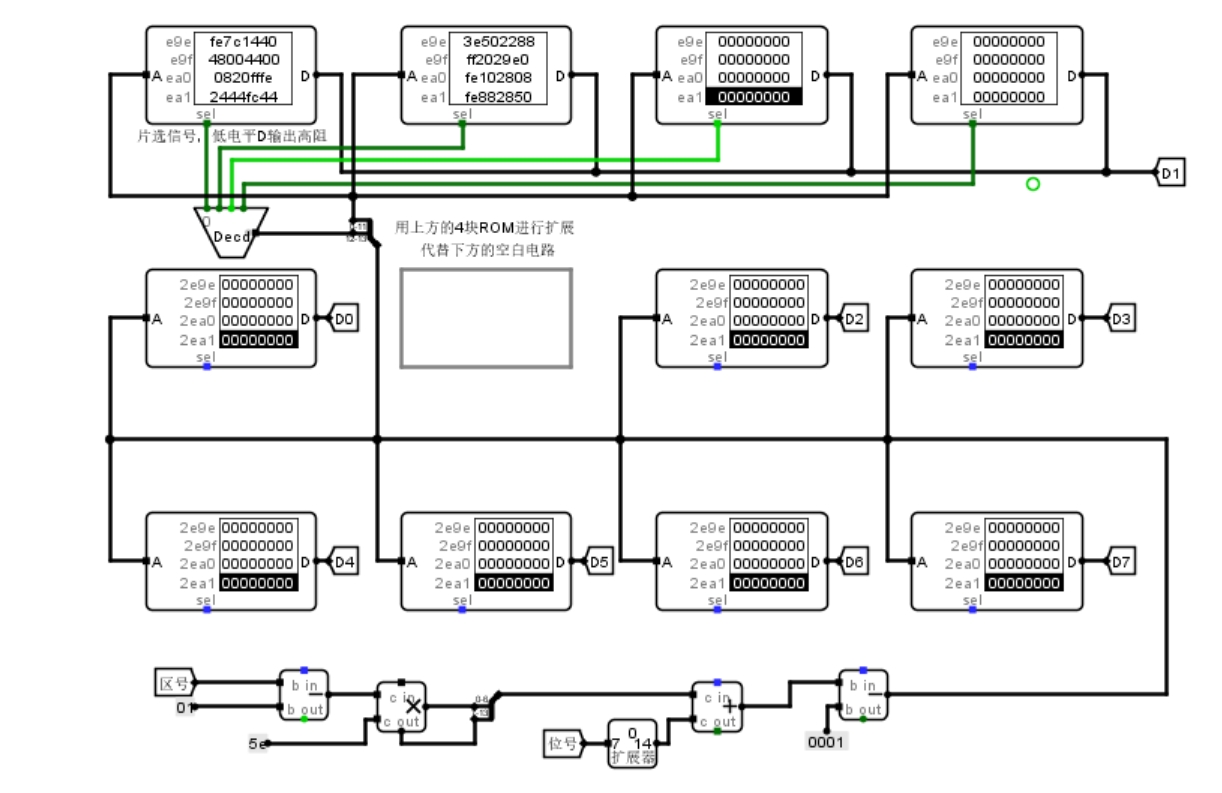
### 2.1.1、 原理

①实验的核心是位扩展；

②地址高2位用来选择小容量ROM：将地址高2位通入解码器，得到片选信号，将片选信号连接到小容量ROM的使能端，便实现了片选功能；

③剩下的地址位连接到小容量ROM的地址端，实现信息的读取。

#### 2.1.2、 电路



#### 2.1.3、 测试

在“字库测试”模块测试电路的正确性，如图：

图示, 示意图

描述已自动生成

### 2.2、 MIPS RAM 设计

**实验目的：**熟练掌握存储扩展基本原理，进一步熟悉片选机制。

**实验内容：**计算机中主存储器通常既能按照字节访问也能按照半字访问，还能按照字进行访问，如MIPS指令中的LB/SB 指令（Load/Store byte）、LH/SH 指令（Load/Store Half），LW/SW 指令（Load/Stire Word）。X86 指令中 mov eax/ax/ah,[200]，而 logisim 中 RAM 存储器只能按 照一种模式访问，为此本实验要求设计完成既能按照 8 位，也能按 16 位，也能按 32 位进行 读写访问的 32 位存储器，最终存储器规格如下：

①字节地址12位（字访问时，忽略低两位，半字访问，忽略最低位，倒数第二位片选，字节访问时，低两位进行片选）

②数据线宽度32位；

③访问 Mode 位: 2 位：00 表示字访问，01 表示 1 字节访问，10 表示 2 字节访问；

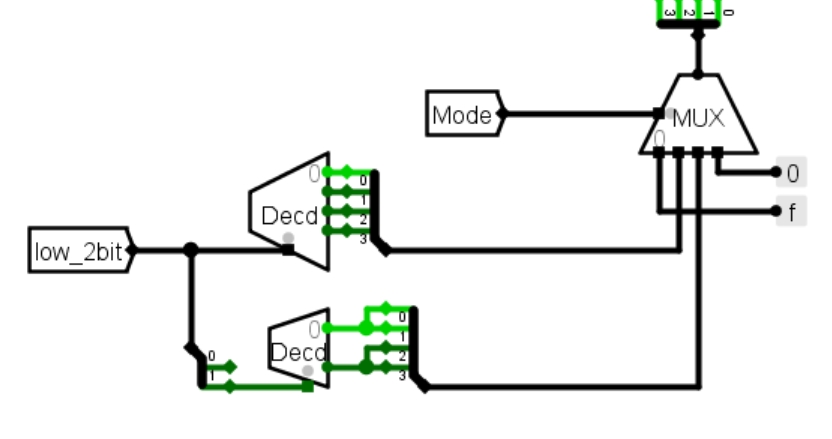
④WE： 写使能，1 表示写入，0 表示读出；

⑤Din： 32 位，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）；

⑥Dout：32 位，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）

#### 2.2.1、 原理

（1）通过Mode得到字访问，字节访问，半字访问的信号。



（2）在不同的模式下，结合字节地址，得到四个寄存器的选中信号：

①在字访问模式下，四个寄存器均被选中；

②在半字访问模式下，有两个寄存器被选中，结合字节地址，便可以使得被选中的寄存器对应的选中信号为1；

③在字节访问模式下，有1个寄存器被选中，结合字节地址，便可以使得被选中的寄存器对应的选中信号为1；

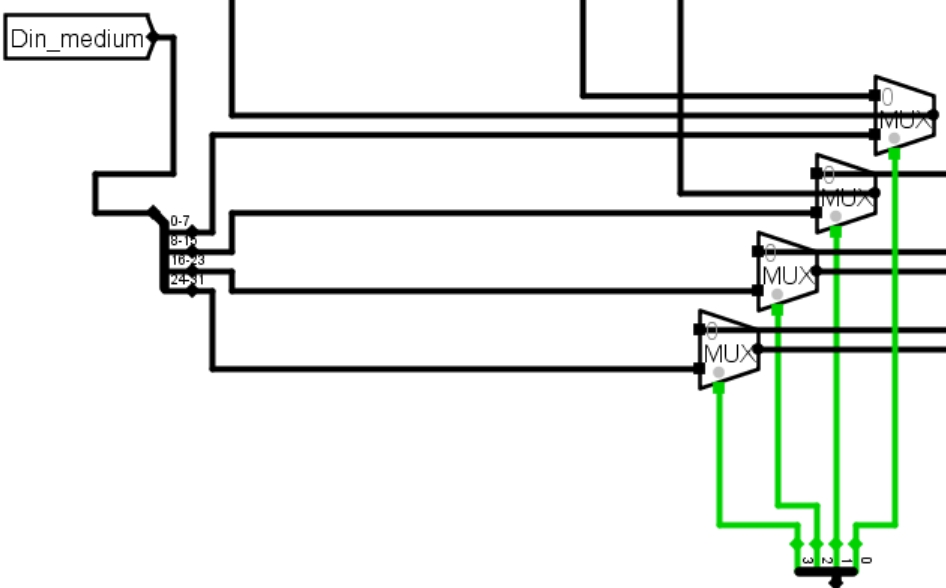
（3）在写入时，在不同模式下，寄存器得到的输入数据的高低位不同：

①在字访问模式下，D0得到输入数据的低八位，D1得到输入数据的8-15位，D2得到输入数据的16-23位，D3得到输入数据的高八位；

②在半字访问模式下，D0得到输入数据的低八位，D1得到输入数据的8-15位，D2得到输入数据的低八位，D3得到输入数据的8-15位；

③在字节访问模式下，D0得到输入数据的低八位，D1得到输入数据的低八位，D2得到输入数据的低八位，D3得到输入数据的低八位；

如图：

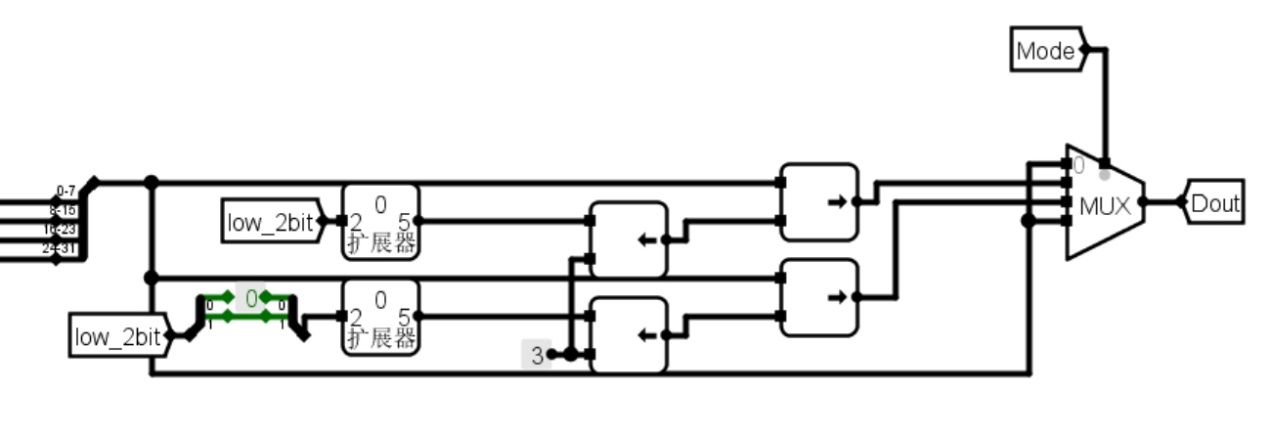


（4）在读模式下，不同的模式下结合字节地址会得到不同的寄存器的数据：

①在字访问模式下，输出数据为四个寄存器输出数据的组合；

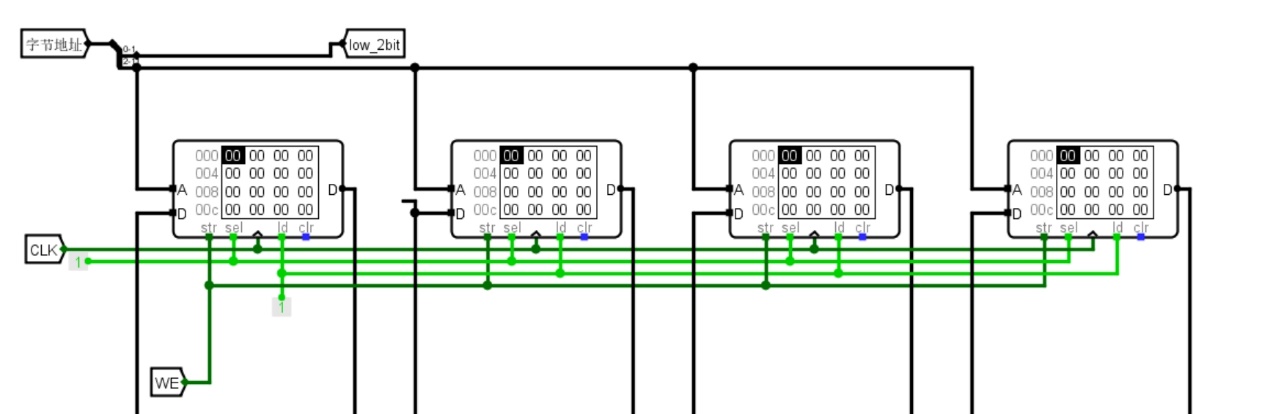
②在半字访问模式下，输出数据为4个寄存器中的2个寄存器数据的组合，结合字节地址决定是01组合还是23组合；

③在字节访问模式下，输出数据为4个寄存器中的1个寄存器数据的组合，结合字节地址决定是哪一个寄存器中的值；如图：



（5）当写使能信号和写选中信号均为1时，选中的寄存器可以被写入数据

如图：



#### 2.2.2、 电路

图示

描述已自动生成

#### 2.2.3、 测试

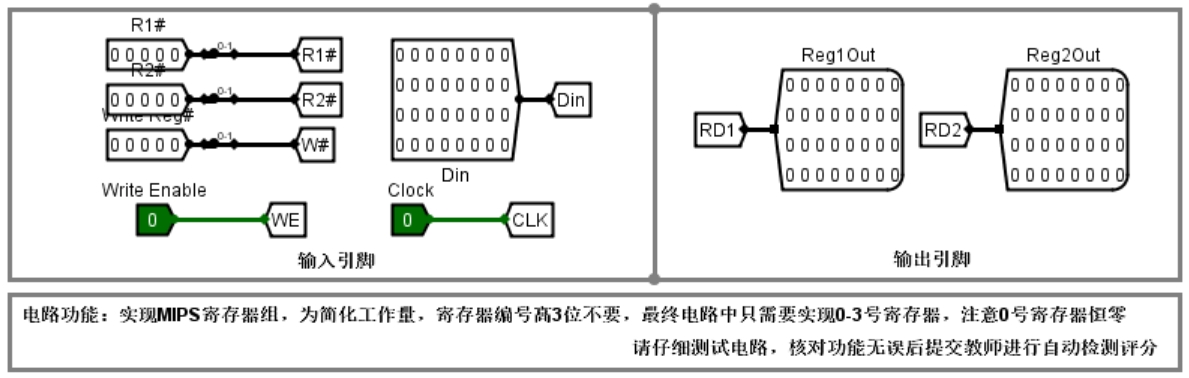
图示, 示意图

描述已自动生成

### 2.3、 MIPS 寄存器文件设计

**实验目的：**为 MIPS CPU 构造核心功能部件，进一步熟悉多路选择器，译码器，解复 用器等 Logisim 部件的使用。

**实验内容：**设计完成满足如下规格要求的 MIPS 通用寄存器组。



#### 2.3.1、 原理

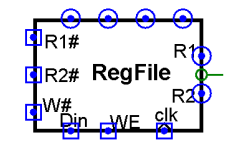
1) 利用 logisim 平台构建一个 MIPS 寄存器组，内部包含 32 个 32 位寄存器，其具体功能如下，具体封装文件为 regfile.circ.

图形用户界面, 文本, 应用程序

描述已自动生成

2) 为减少实验中画图工作量，实验工程文件中对 5 位寄存器地址进行了简化，具体见引脚示意图，最终只需实现4个寄存器，0 号寄存器功能仍然是恒零。后续实验中如需要使用 32 个寄存器的MIPS寄存器文件组，将提供标准组件。

3) 注意时钟信号和电平信号不要混连，时钟仅仅触发状态改变。



#### 2.3.2、 电路

图示

描述已自动生成

#### 2.3.3、 测试

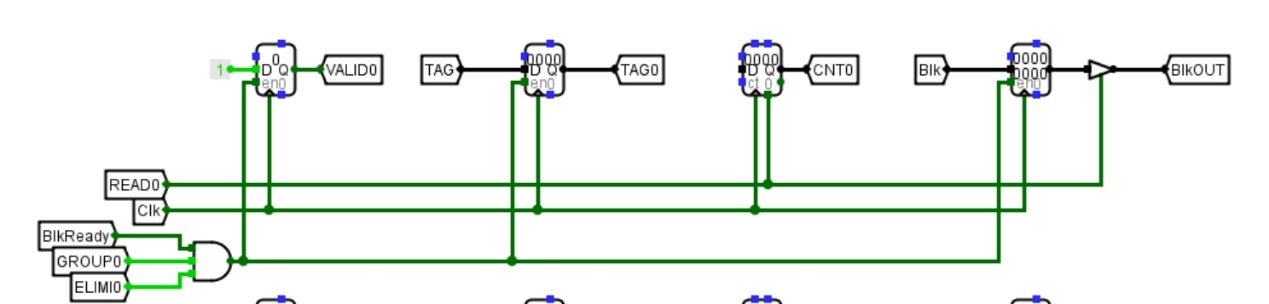
图示

描述已自动生成

### 2.4、 Cache四路组相连电路实现

#### 2.4.1、 原理

（1）Cache行封装如下：



（2）将字节地址分出偏移、行号、区地址：

①0-1位为偏移；②2-4位为行号；③5-15位为区地址。

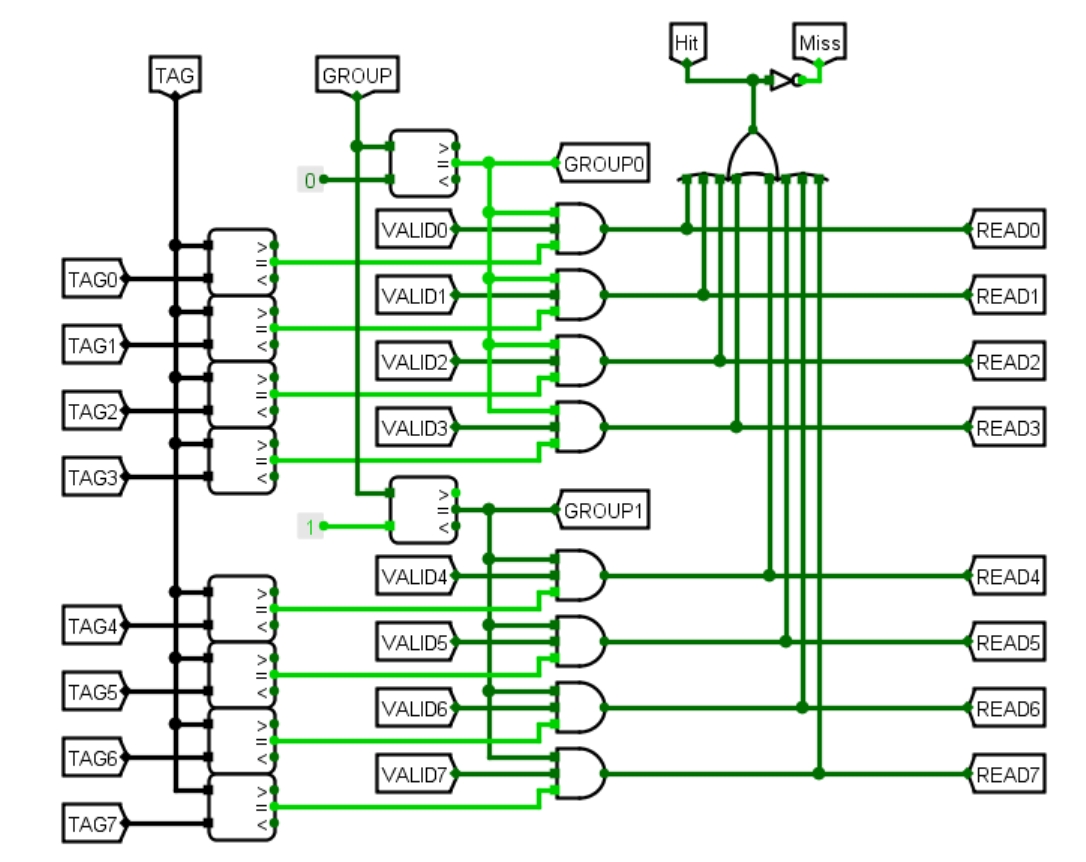
如图 ：

图示

描述已自动生成

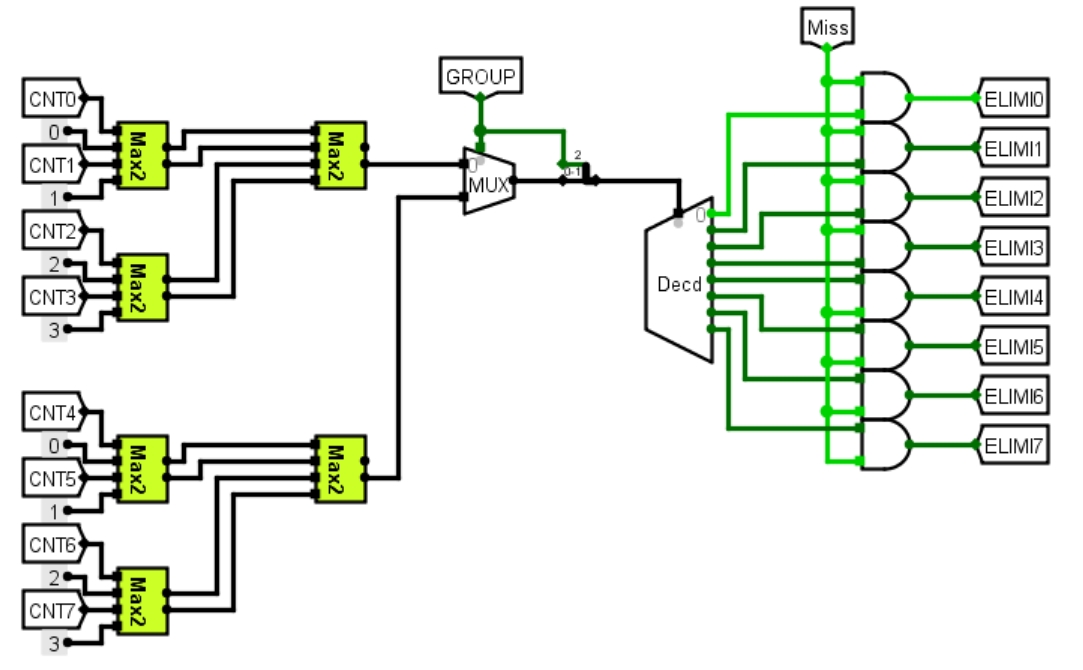
（3）得到读信号：

首先根据组索引确定所在组号（第0组或第1组），然后根据tag四路并发比较，判断数据所在行，当Cache中对应行valid为1且其中存放数据的tag标记与我们想要查找的数据块的地址相同，则得到该行的读信号。否则所有行的读信号均为0，如下图：。

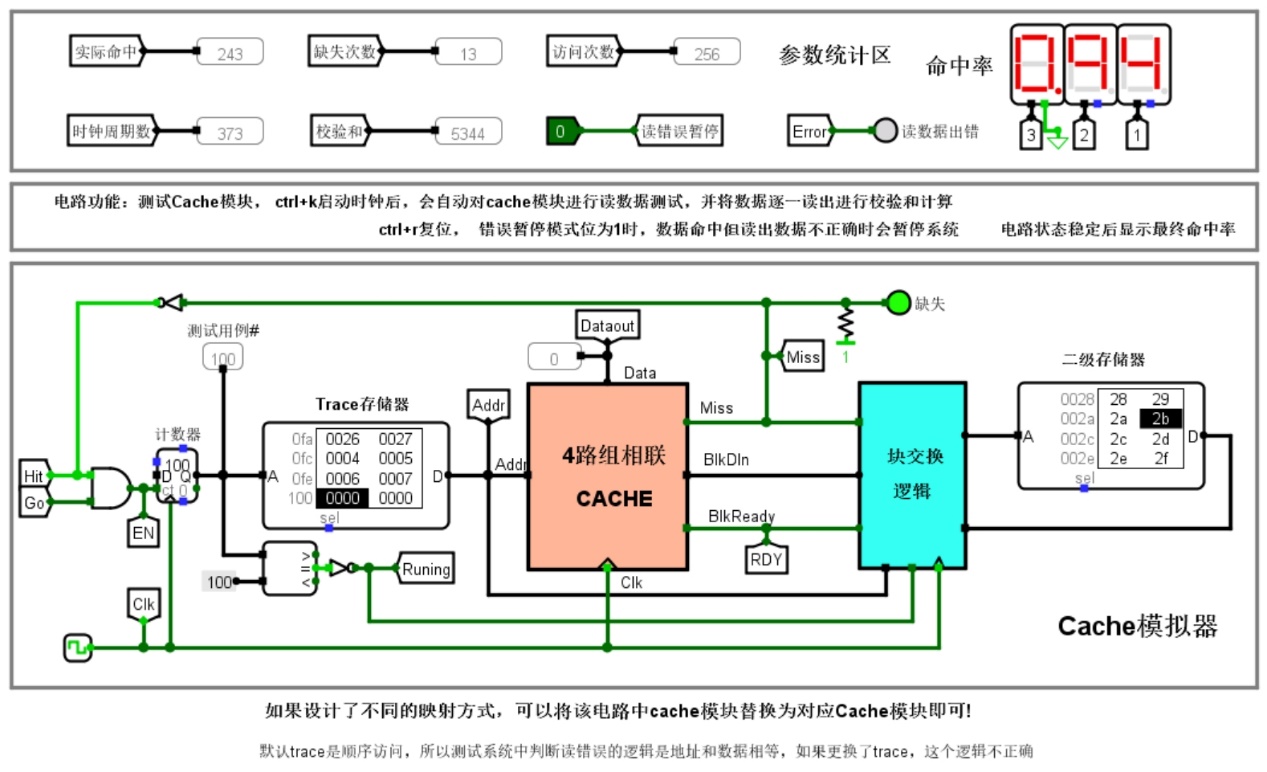


（4）得到miss信号：当没有选中信号为1，即未命中时，miss信号为1，表明未命中。如上图。

（5）数据淘汰算法采取近期最少使用算法，添加一个标记位CNT,记录cache行未被使用的次数，每个时钟所有CNT加1，命中的cache行CNT清零，故使用计数器控制CNT,将读信号READ接到计数器清零信号处。执行淘汰算法时 ，通过归并比较器，找到CNT最大的cache行对应的序号，通过译码器选通对应的淘汰信号ELIMI，实现LRU算法功能，电路图如下：

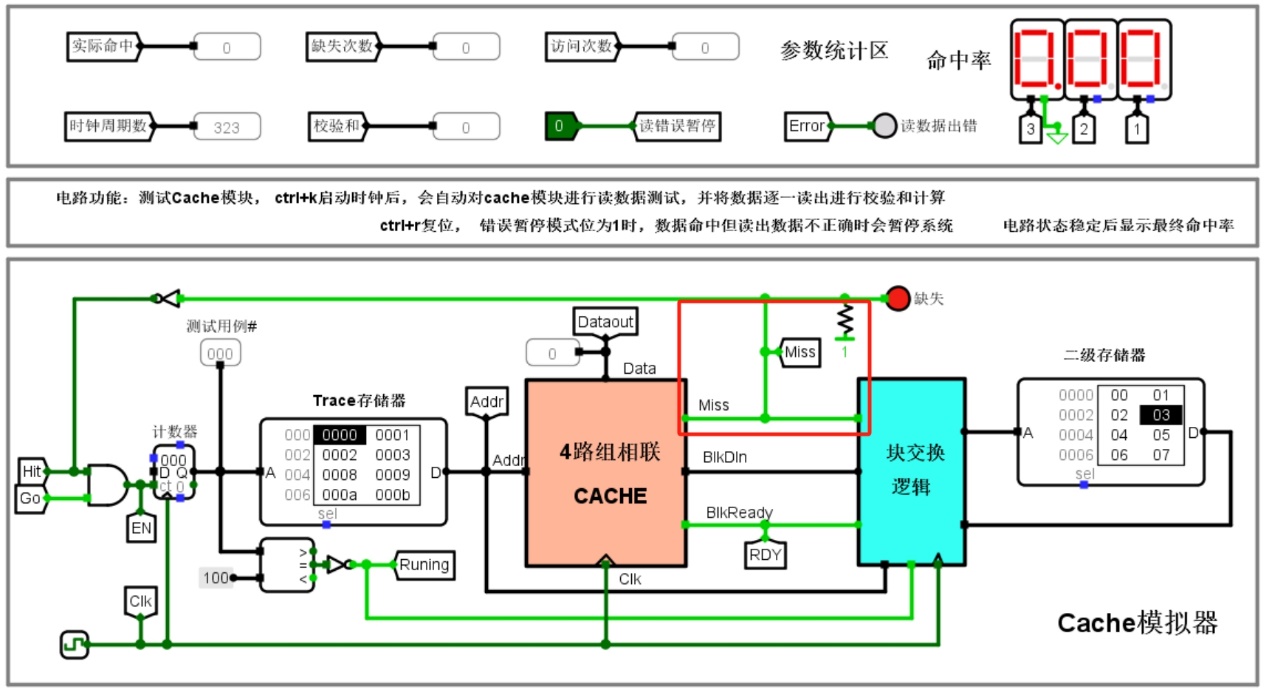


#### 2.4.2、 测试



## 3、遇到的问题及解决办法

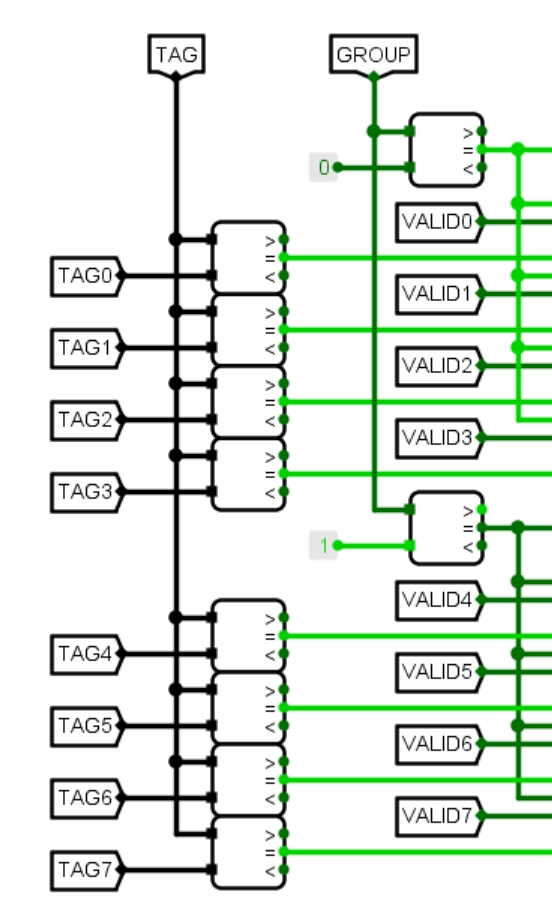
#### 3.1、 问题一



可见，所有时钟周期，Miss信号一直处于高电平，按正常逻辑，在前四个周期cache未命中，块交换逻辑将从二级存储器中读取四个数据，在BlkReady信号有效后，将数据写入cache，之后cache命中后Miss信号应置为低电平，但图中所示的Hit和Miss信号无法改变。

首先怀疑是cache电路设计问题，经检查，发现电路逻辑无误，单步调试发现每个周期BlkReady信号也正常，其他信号也正常变化，突然发现cache中隧道标签误写为HIT和MISS，导致这两个信号无法正常变化，以至于所有周期Miss信号一直拉高，出现问题

#### 3.2、 问题二



在实现组内tag比较时，使用了多余的比较器，实际上可以先将TAG送入一个比较器，之后再和每一行的tag比较，从而实现并发比较的功能，减少比较器的冗余。