**存**

**储**

**系**

**统**

**实**

**验**

**报**

**告**

**书**

**学 院： 人工智能与自动化学院**

**班 级： 人工智能2004班**

**姓 名： 陈乃睿**

**学 号： U202012593**

**实验时间： 2022年6月10日**

目录

[一、 实验名称 2](#_Toc8408)

[二、 实验目的 2](#_Toc6823)

[1. 存储器拓展试验： 2](#_Toc17940)

[2. MIPS RAM设计实验： 2](#_Toc26925)

[3. MIPS寄存器文件实验： 3](#_Toc32758)

[4. Cache硬件设计实验： 3](#_Toc10956)

[三、 实验内容 3](#_Toc20072)

[1. 存储器拓展实验： 3](#_Toc25131)

[2. MIPS RAM设计实验： 3](#_Toc10832)

[3. MIPS寄存器文件实验： 3](#_Toc2567)

[4. Cache硬件设计实验： 3](#_Toc15135)

[四、 实验电路与设计思路 4](#_Toc3346)

[(1) 输入数据拆分部分： 7](#_Toc9659)

[(2) 输入信号判断部分： 7](#_Toc13736)

[(3) 主体部分： 7](#_Toc10201)

[(4) 输出筛选部分 7](#_Toc7247)

[五、 实验结果 9](#_Toc2816)

[1. 存储器扩展实验： 9](#_Toc4897)

[2. MIPS RAM实验： 10](#_Toc5326)

[3. MIPS寄存器文件实验： 11](#_Toc15756)

[4. Cache硬件设计实验: 12](#_Toc1587)

[六、 实验总结 13](#_Toc16141)

# 实验名称

存储系统实验

1. 存储器拓展实验
2. MIPS RAM设计实验
3. MIPS寄存器文件实验
4. Cache硬件设计实验

# 实验目的

1. 存储器拓展试验：

理解存储系统进行位扩展、字扩展的基本原理，能利用相关原理解决实验中汉字库的存储扩展问题，并且能够使用正确的字库数据填充。

1. MIPS RAM设计实验：

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

1. MIPS寄存器文件实验：

了解MIPS寄存器文件基本概念，进一步熟悉多路选择器、译码器、解复用器等Logisim组件的使用，并利用相关组件构建MIPS寄存器文件。

1. Cache硬件设计实验：

掌握Cache实现的3个关键技术：数据查找、地址映射、替换算法。熟悉译码器、多路选择器、寄存器的使用，能根据不同的映射策略在Logisim平台中用数字逻辑电路实现Cache机制。

# 实验内容

1. 存储器拓展实验：

在第一章汉字编码实验中实现了汉字字符的32\*32点阵显示，一个汉字编码的显示需要1024位的点阵信息。为了实现汉字自型码在组合逻辑电路中的直接显示，本实验在Logisim平台利用32个32位ROM组件按位扩展的方式，构造了位宽为1024的存储系统，用于储存汉字字库，将所有汉字字形码存储在该系统中，并且利用汉字区位码进行索引，给出一个区位码，可一次性地取出1024位字形码进行显示。

现有如下ROM组件，4片4KB\*32位ROM，7位16位KB\*32位ROM，请在Logisim平台中构建GB3212汉字编码的16\*16位点阵汉字字库，电路输入为汉字区号和位号，电路输出为8\*32位（16\*16=256位点阵信息）。

1. MIPS RAM设计实验：

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3中访问模式，实验要求利用4个4KB\*8位的RAM组件进行扩展，设计完成技能按照8位，也能按照16位，还能按照32位进行读写访问的32位存储器。

1. MIPS寄存器文件实验：

利用Logisim平台构建一个MIPS寄存器文件，内部包含32个32位寄存器。（采用分线器将5位寄存器编号低两位引出，实际只使用了两位编号，所以最终只需实现4个寄存器）

1. Cache硬件设计实验：

Cache测试用给出了一个在Logisim中设计完成的Cache系统自动测试电路，为简化实验设计，这里所有Cache模块均为只读Cache（类似指令Cache）

，无写入机制。电路左侧计数器与存储器部分会在时钟驱动下逐一生成地址访问序列给Cache模块。计数器模块的使能端受命中信号驱动，缺失时使能端无效，计数器不计数，等待系统将待请求数据所在块从二级存储器中调度到Cache后才能继续计数。Cache与二级存储器之间通过块交换逻辑实现数据块交换，由于二级存储器相比Cache慢很多，所以一次块交换需要多个时钟周期才能完成，Cache模块判断数据块准备好的逻辑是BlkReady信号有效，该信号有效且时钟信号到来时，Cache将块数据从BlkDin端口一次性载入到对应Cache行缓冲区中，此时Cache数据命中，直接输出请求数据，解锁计数器使能端，继续访问下一个地址。

自动测试电路会逐一取出trace存储器中的主存地址去访问存储系统，并逐一将数据从Cache模块取出送校验和计算电路计算校验和，当计数器值为256时会停止电路运行，此时存储访问的Cache命中率将会在右上角的LED数码管显示。本次实验的主要任务就是设计该电路的核心模块Cache子电路。

Cache映射策略和调度策略共有如下3中组合：

1. 全相联映射（8路组相联），LRU调度。
2. 直接相联映射（1路组相联）。
3. 组相联映射（2路组相联，4路组相联二选一），LRU调度。

请在以上3种组合中任选一组进行电路实现，并在最终的测试电路中进行命中率测试，最终校验和应为32640。

# 实验电路与设计思路

1. 存储器拓展实验：

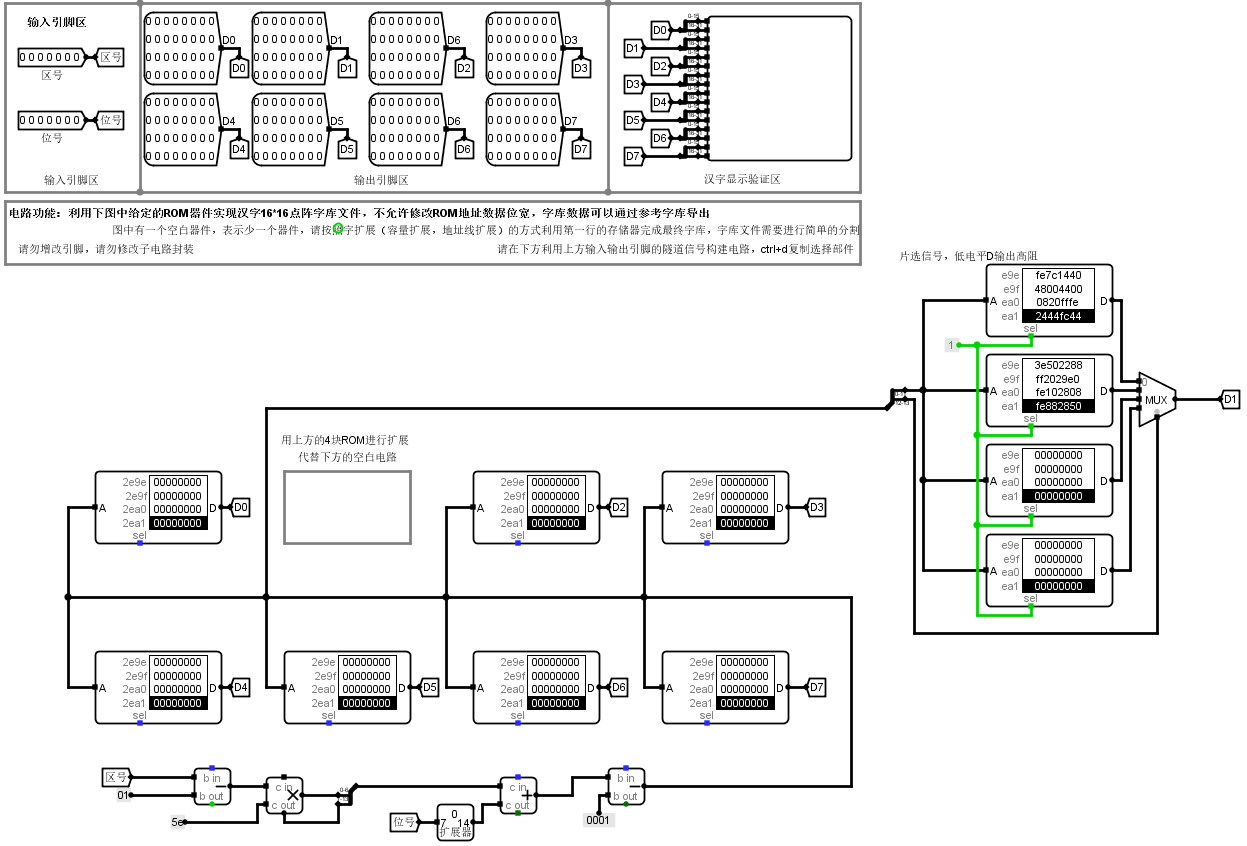


图 1

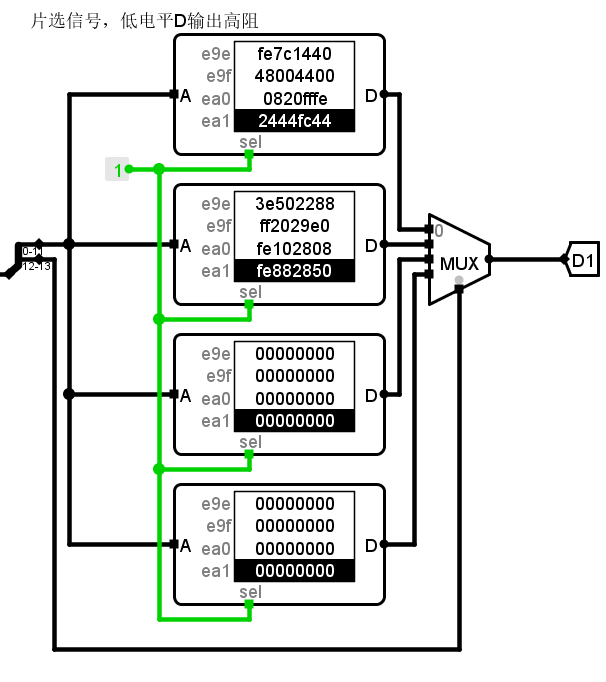


图 2 字扩展放大

图 3

该电路应由8块16KB\*32位ROM组成，而本实验只有7块16KB\*32位ROM以及4块4KB\*32位ROM，故：应对这4块ROM进行字扩展。由题目要求知：具体哪一个芯片工作由高两位决定，故：最高两位决定由哪块ROM输出信号进入D1,后12位决定该ROM输出信号的地址（由图2所示，其中sel引脚表示当该引脚信号为0时，禁用该ROM）。

然后是数据分配问题。文件中已经给出所有汉字的字库文件，只需导入即可；而对于这4个ROM，应注意每个ROM能存储的数据大小为4KB=4096B,即：按照顺序，每个ROM只能存储4096B个字库文件。将所有字库文件存入后即可完成。

1. MIPS RAM实验

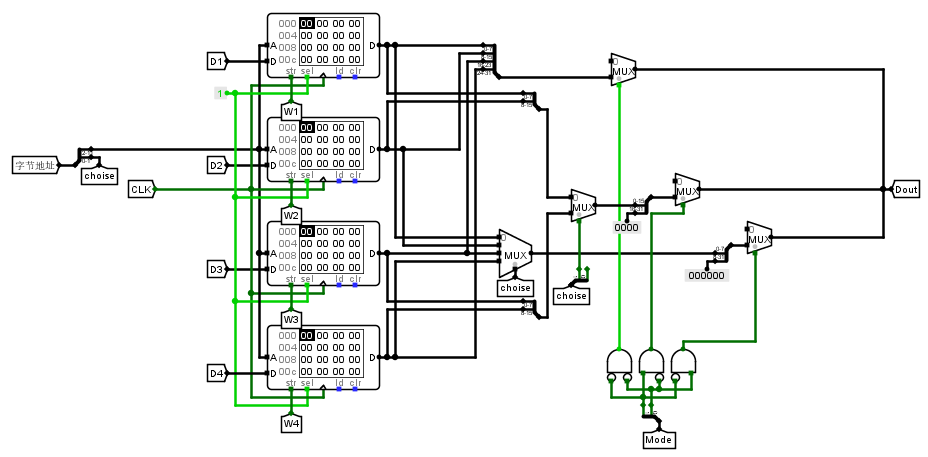


图 4

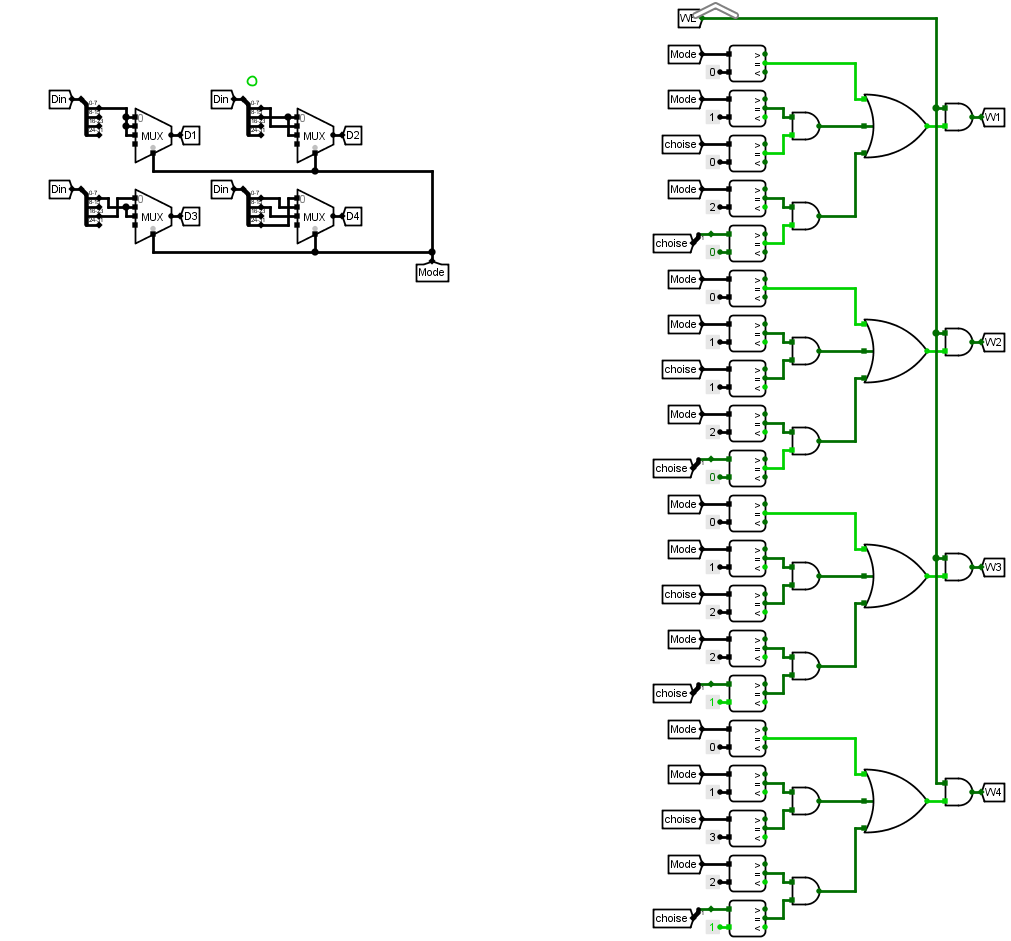


图 5

该实验的核心是存储器的位扩展，因此图中的4个RAM应为并联结构。该实验电路主要分为4个部分：主体部分（图4左）、输出筛选部分（图4右）、输入数据拆分部分（图5左）、输入信号判断部分（图5右）。

1. 输入数据拆分部分：

按照Mode的数值，将输入数据按大小进行拆分。当Mode=00时，输入为字，此时输入数据（32位）应按顺序平均分配到各个RAM中；Mode=01时，输入为字节，此时输入数据只取最低8位，再按照输入信号判断部分的信号输入到相应RAM中；Mode=10时，输入为半字，此时输入数据只取低16位，再按照输入信号判断部分的信号输入到相应的RAM中（平均分配到0、1号RAM中或2、3号RAM中）。

1. 输入信号判断部分：

按照Mode的数值以及地址低两位的数值，对每个RAM是否该进行存储进行判断。当Mode=00时，此时每一个RAM都应该存储一部分数据；当Mode=01时，只需一个RAM存储一个字节，Addr最低两位的数值就是应该存储该字节的RAM的编号；当Mode=10时，需要两个RAM共同存储半个字，Addr倒数第二位为0时，存入0、1号RAM中，当Addr倒数第二位为1时，存入2、3号RAM中。

1. 主体部分：

无论Mode为多少，数据存储的地址都只看前10位。故：可将地址分为两部分，前面10位用于查找位置，后面两位用于查找RAM。通过上述两部分以及时钟信号的作用，即可将正确的数据存入RAM中。

1. 输出筛选部分

输出时按照Mode的数值输出结果。当Mode=00时，将4个RAM中对应的地址集合起来输出即可；当Mode=01时，用多路选择器选择该输出哪个RAM的数据；当Mode=10时，将0、1号RAM的数据用集线器集合，2、3号RAM的数据用集线器集合，再通过多路选择器选择该输出哪个数据即可。

1. MIPS寄存器文件实验

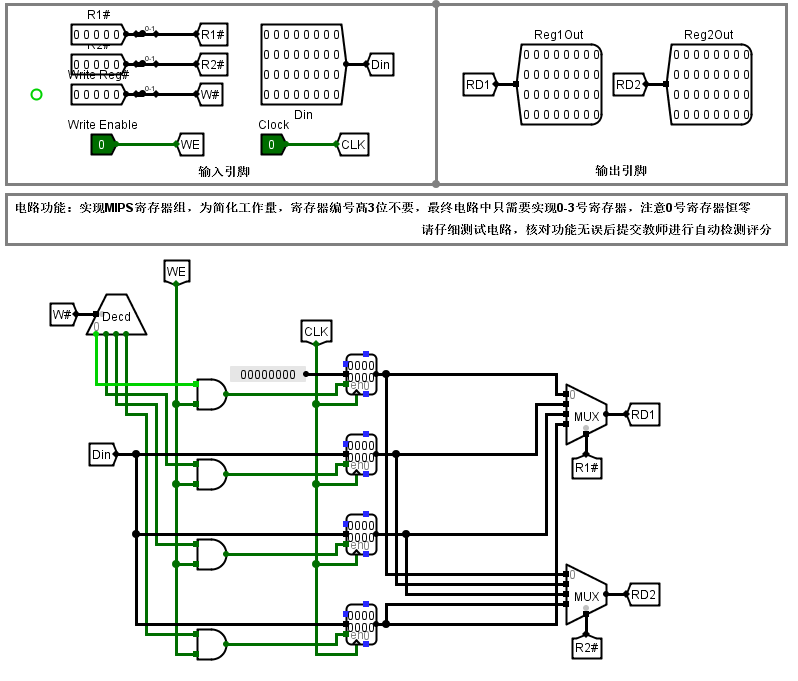


图 6

由题目可知，0号寄存器的输入数据恒为0，其他寄存器输入数据由Din决定。W#决定哪个寄存器被激发，WE决定寄存器该不该被激发。当以上两个条件均满足时，数据会进入寄存器。R1#和R2#分别代表应将哪个寄存器中的数据输出至RD1/RD2中。

1. Cache硬件设计实验：（注：由于大小原因，原电路中有8个Cache行，由于设计相似，仅截图两行）

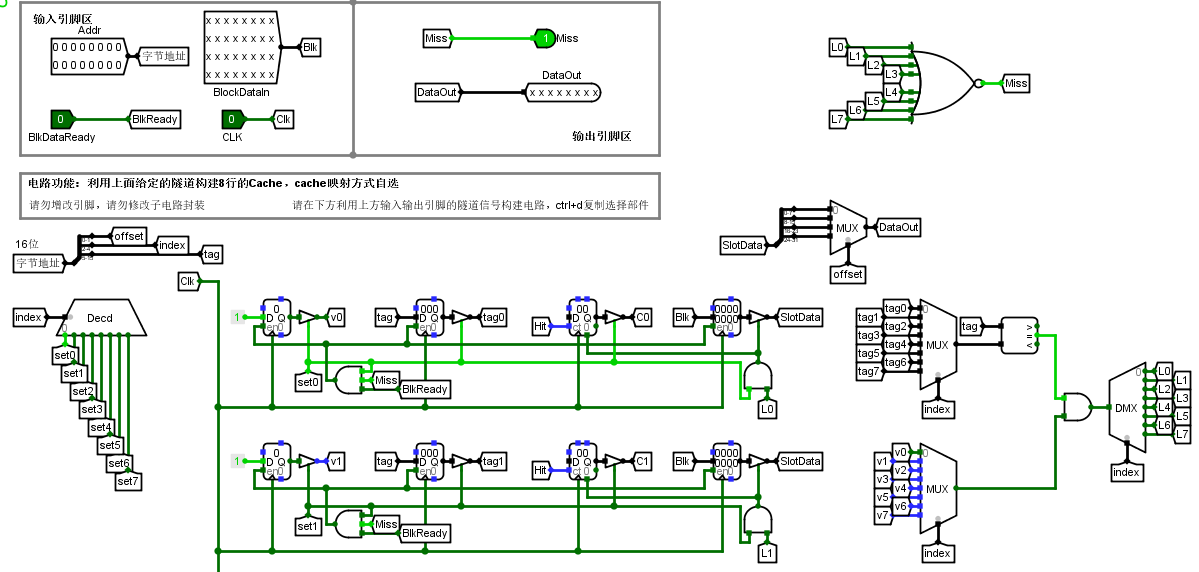


图 7

我选择的是直接相联映射线路。首先按照Cache的规则，将字节地址分为三个部分：tag、index、offset。按照书上T70图3.24练好大概的电路图。首先，将index用译码器译码，查找哪行Cache有操作信号（set）。以第0行为例，以set0=1为前提，当输入信号（BlkDataReady）准备好且该行Cache未命中（Miss）时，可以将数据写入该行。当该行中存储的tag0=tag且标志位v0=1时，标志信号L0=1，此时可将存入32位寄存器的数据Blk输出。

当L0~L7中有一个信号存在时，Cache成功命中，否则未命中（Miss）。当Cache成功命中时，用offset（块内偏移）来输出正确的数值即可。

# 实验结果

1. 存储器扩展实验：

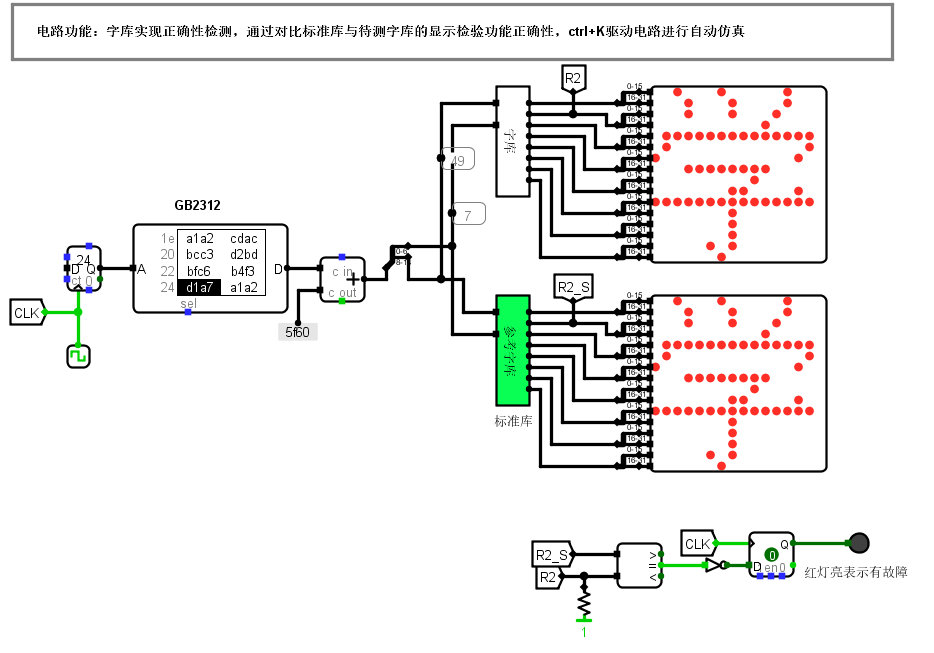


图 8

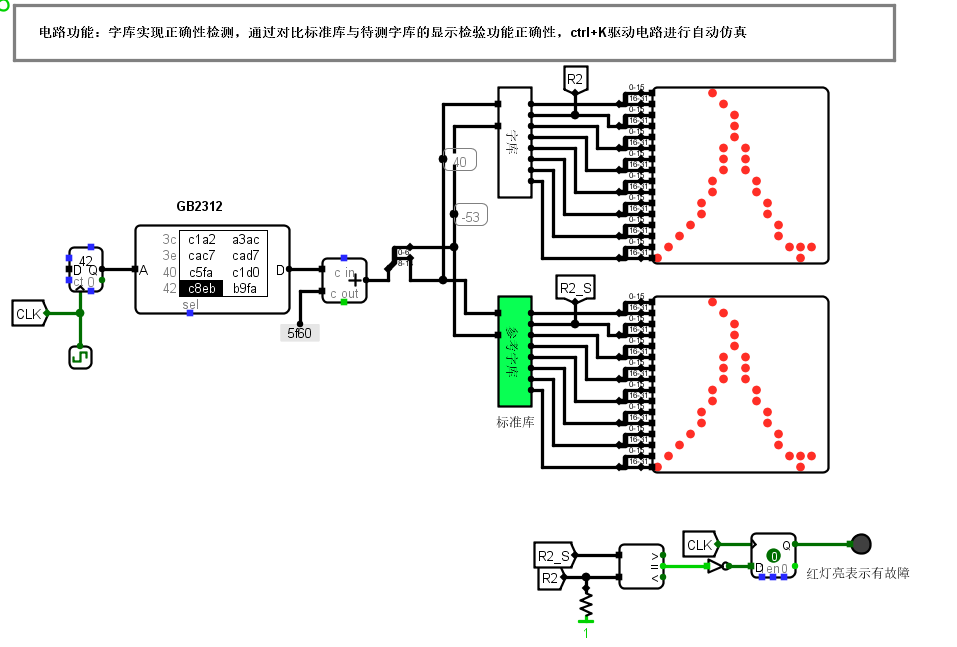


图 9

1. MIPS RAM实验：

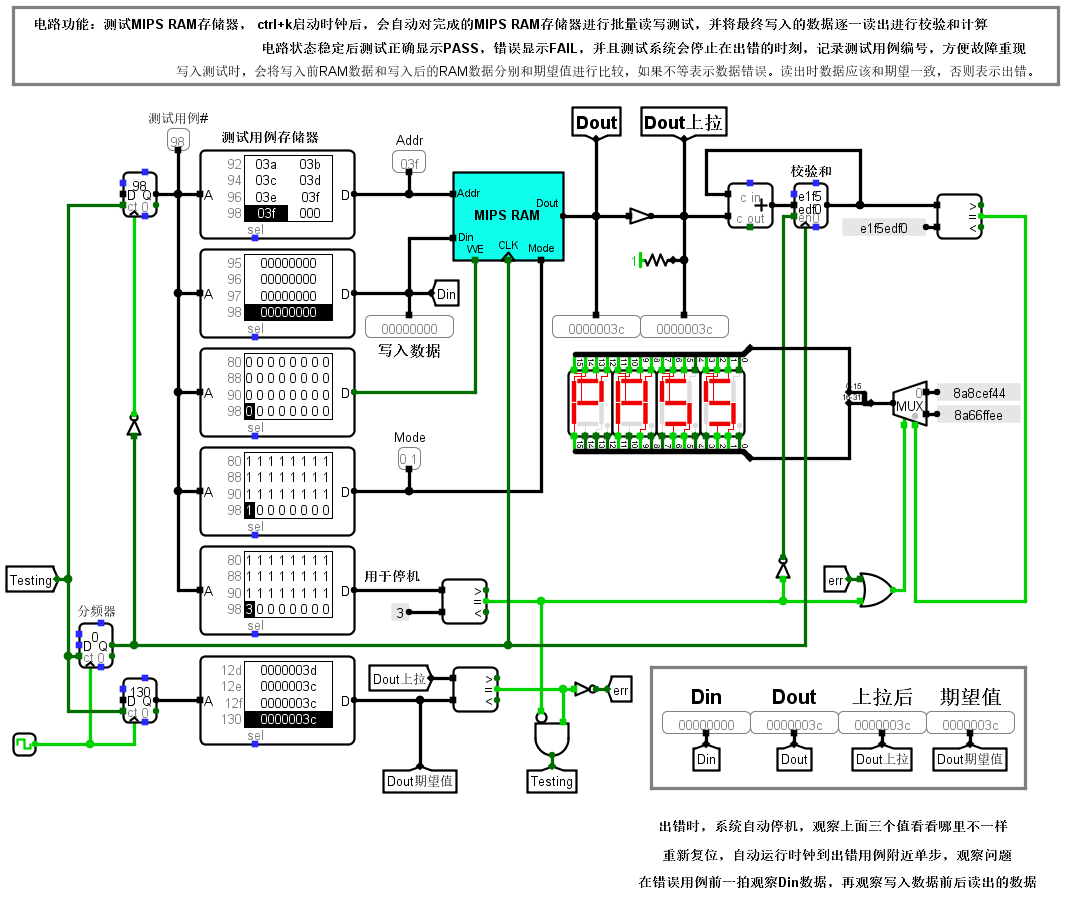


图 10

1. MIPS寄存器文件实验：

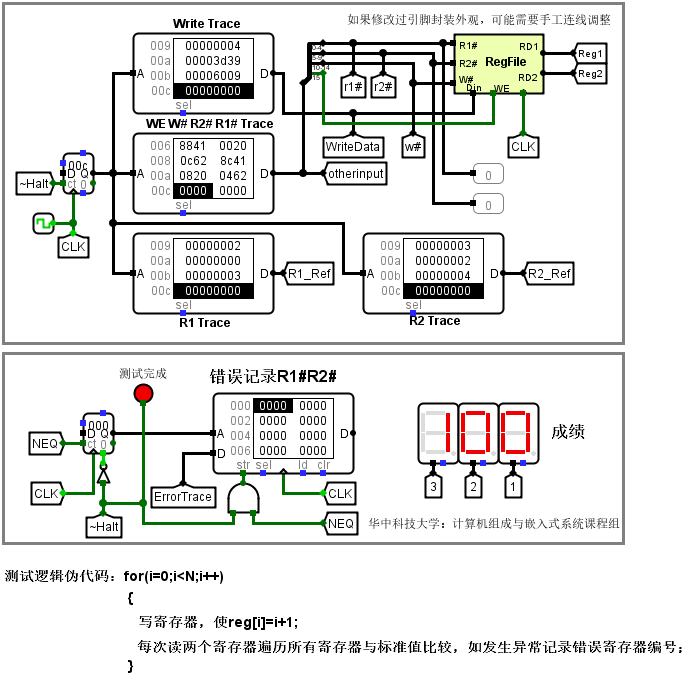


图 11

1. Cache硬件设计实验:

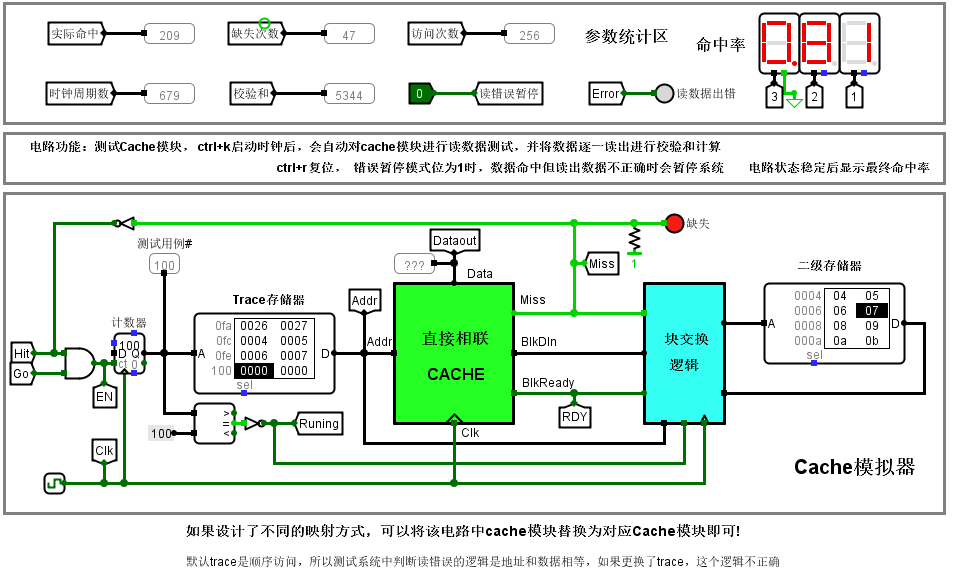


图 12

# 实验总结

本次实验具有一定的挑战性，要求对于Cache的工作原理较为熟悉。在存储器扩展实验的设计中，我不知道要将汉字库中的数据手动导入到ROM中，在网上寻找解决方法时才得以解决。然而，对于D1，我最开始的导入方式是将对应的汉字库平均分为4份导入，但是显示一直不正确。后来才发现原来汉字库的数据并没有ROM能储存的那么多，必须按照实验的要求计算后分块导入。第二次出问题是在MIPS RAM的连线中，刚开始的时候并没有完全理解四个RAM并联的意义，将一个字导入到了同一个RAM中导致出错，后来查看教材才得以解惑。第三次出问题是在Cache（直接相联）中，我连接的Cache行是按照书上给的参考设计来的。然而这个参考设计中使用了三态门，这就导致当三态门不导通时，有几个数据会被悬空，导致后续的线路无法正常运转。最后通过查看书上的电路，重新仅使用一个比较器，解决了该问题。总之，这次实验又是一次挑战满满的实验。