

华中科技大学

计算机组成原理实验报告

实验名称：存储系统实验

班 级：人工智能 2204 班

学 号：U202214123

姓 名：陈博

报告日期：2024 年 5 月 23 日

目录

1. 实验内容及目的.....	3
1.1 存储器扩展实验.....	3
1.2 MIPS RAM 设计实验.....	3
1.3 MIPS 寄存器文件设计实验.....	3
1.4 Cache 硬件设计实验.....	3
2. 实验电路、设计思路及结果.....	4
2.1 存储器扩展实验.....	4
2.2 MIPS RAM 设计实验.....	5
2.3 MIPS 寄存器文件设计实验.....	7
2.4 Cache 硬件设计实验.....	9
3. 实验中遇到的问题.....	11
3.1 存储器扩展实验.....	11
3.2 MIPS RAM 设计实验.....	11
3.3 MIPS 寄存器文件设计实验.....	11
4.4 Cache 硬件设计实验.....	11
4. 实验总结.....	12

1. 实验内容及目的

1.1 存储器扩展实验

理解存储系统进行位扩展、字扩展的基本原理，包括位扩展（数据总线扩展、字长扩展）、字扩展（地址总线扩展、字数扩展）、字位同时扩展（综合扩展）。能利用相关原理解决实验中汉字字库的存储扩展问题，包括能设计汉字字库存储扩展电路、能使用正确的字库数据填充。

1.2 MIPS RAM 设计实验

理解主存储器地址基本概念，包括内存访问地址都是字节地址、字节/半字/字访问。理解存储位扩展基本思想，能设计同时支持字节、半字、字访问的存储子系统。

1.3 MIPS 寄存器文件设计实验

了解 MIPS 寄存器文件基本概念，包括寄存器文件（寄存器堆）：通用寄存器的集合、MIPS32 指令集支持 32 个通用寄存器（32 位）、通过对应寄存器编码访问、熟悉多路选择器、译码器、解复用器等组件、能设计 MIPS 寄存器文件电路。

1.4 Cache 硬件设计实验

掌握 Cache 实现的关键技术，包括数据查找（如何快速判断数据是否在 Cache 中）、地址映射（主存中的数据块应如何放置在 Cache 中）、全相联、直接相连、组相联、替换算法（选择什么样的 Cache 数据块进行替换或淘汰）、熟悉译码器、多路选择器、寄存器的使用、能设计实现 Cache 机制。

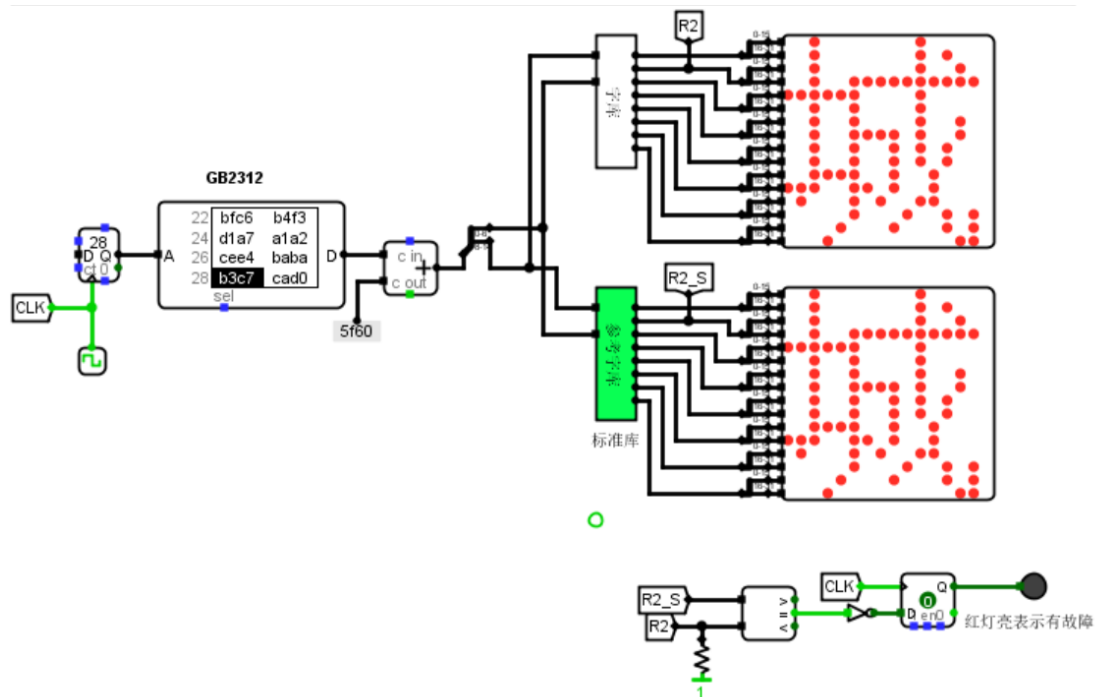


图 2 存储器扩展实验结果

2.2 MIPS RAM 设计实验

本次实验的核心要求是利用 4 个 4 K*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图 3 所示。

引脚	输入/输出	位宽	功能描述
Addr	输入	12	字节地址（字访问半字访问时应硬件强制对齐）
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
WE	输入	1	写使能，高电平有效
CLK	输入	1	时钟信号，上跳沿有效
Mode	输入	2	访问模式 00：字访问，01：字节访问，10：2字节访问
Dout	输出	32	输出数据，不同访问模式有效数据均存放在最低位，高位忽略

图 3 MIPS RAM 实验的引脚定义要求

对于这个实验，先考虑输入，题目给出的 ROM 的地址位宽位 10 位，所以地址取高 10 位。Str 信号由各 ROM 的 CSi 此信号给出，表示仅当写模式下，该 ROM 需要参与写时才有效。sel 端置 1。

然后是数据的取值方面，如果 Mode 为 00 时表示字访问，即写入 32 位数据，

4 片 ROM 都被选中，分别对应第 0，1，2，3 个字节；Mode 为 01 表示单字节访问，即写入 8 位数据，这时只选中一片 ROM，而且写入的是 Din 的最低字节；Mode 位 10 时表示半字访问，即写入 16 位数据。这里要选中 2 片 ROM，由于要对齐所以只能选高两位或低两位。

最后确定各个 CSi，先使用一个解码器，根据 Mode 的取值来决定要进行的访问方式，然后在三种访问方式下设计如下的电路来确定 CSi 的取值，地址在字访问时忽略最低两位；半字访问时忽略最低位，倒数第二位为片选；字节访问时，第二位进行片选。实验电路图如图 4 所示，实验结果如图 5 所示，显示出了 PASS。

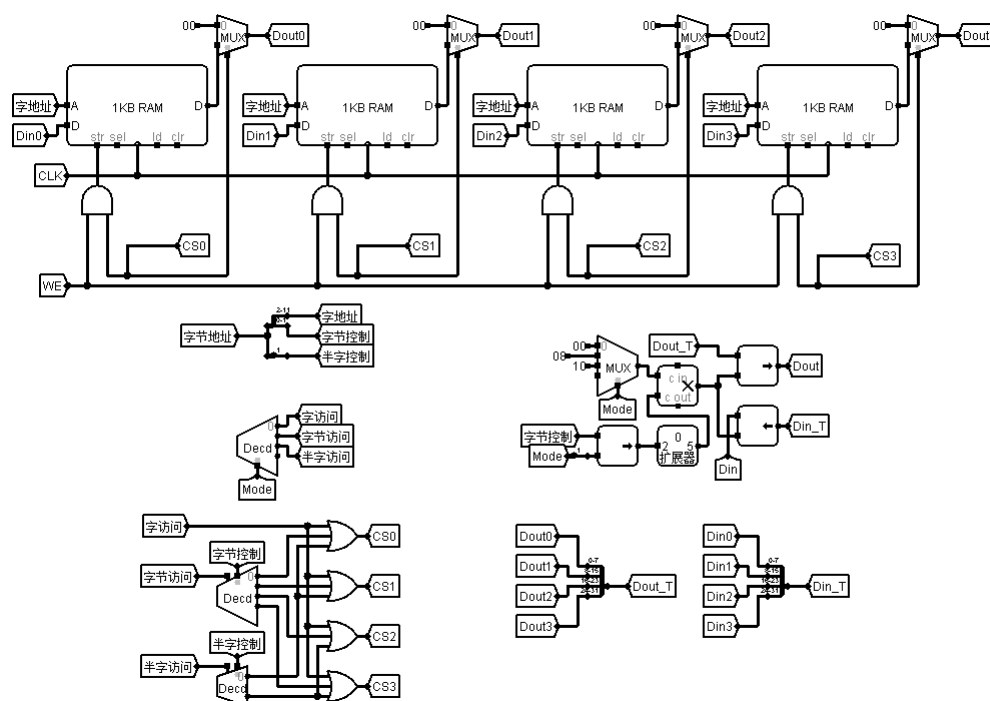


图 4 MIPS RAM 实验电路图

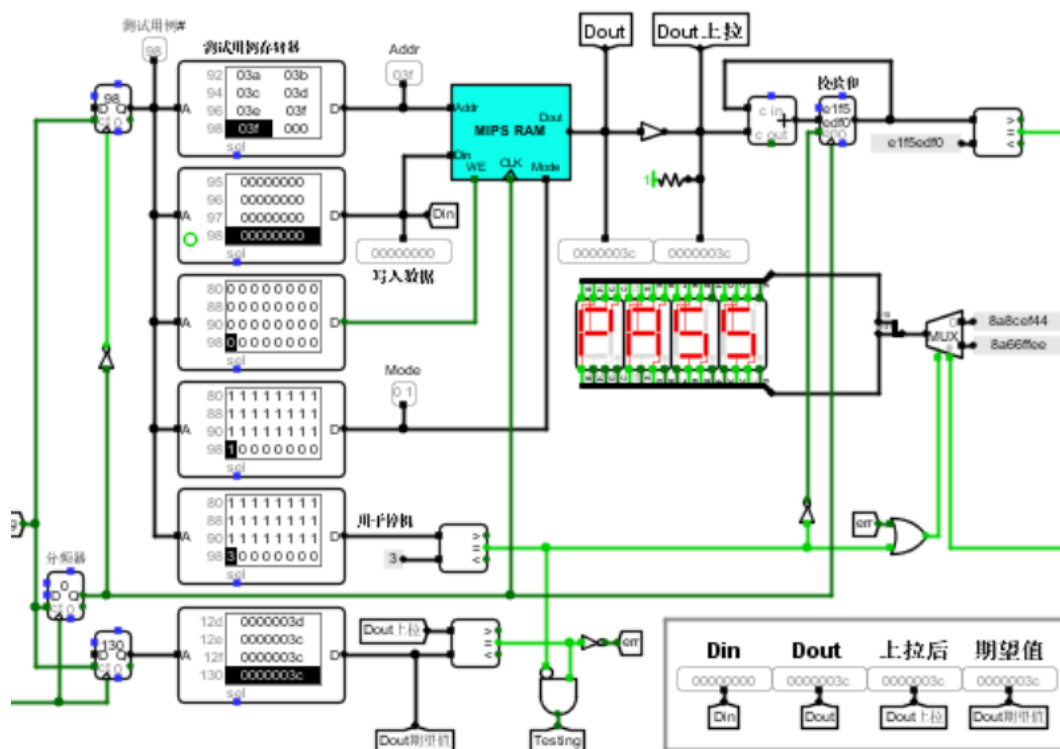


图 4 MIPS RAM 实验结果

2.3 MIPS 寄存器文件设计实验

实验的要求如图 5 所示。

引脚	输入输出	位宽	功能描述
R1#	输入	5	第1个读寄存器的编号
R2#	输入	5	第2个读寄存器的编号
W#	输入	5	写入寄存器编号
Din	输入	32	写入数据
WE	输入	1	写使能信号，为1时在CLK上跳沿，将Din数据写入W#寄存器
CLK	输入	1	时钟信号，上跳沿有效
RD1	输出	32	R1#寄存器的值，MIPS寄存器文件中0号寄存器的值恒零
RD2	输出	32	R2#寄存器的值，MIPS寄存器文件中0号寄存器的值恒零

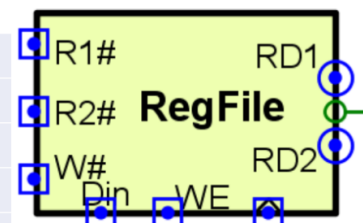


图 5 MIPS 寄存器文件设计实验要求

首先，将 **W#** 进行译码，产生每一个寄存器对应的写信号。然后，对于每个寄存器的使能，由两个部分组成，一个是写入寄存器编号 **Wi**，一个是写使能。

二者同时满足，且在上升沿的条件下才能将数据写入寄存器。最后的输出，由 R1#和 R2#进行数据选择，输出到 RD1 和 RD2 即可。实验电路图如图 6 所示，实验结果如图 7 所示。

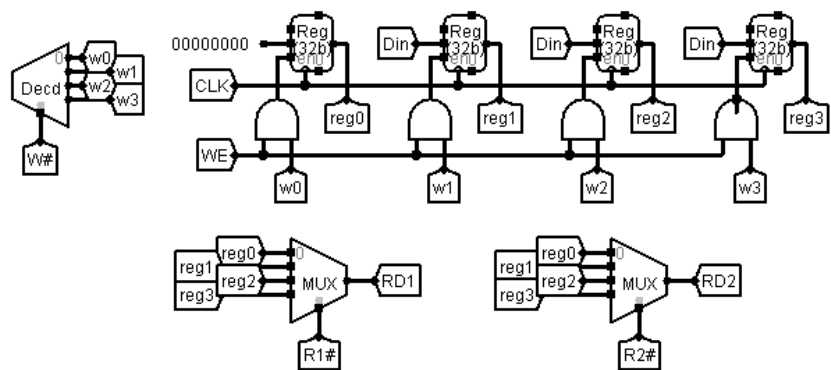


图 6 MIPS 寄存器文件设计实验电路图

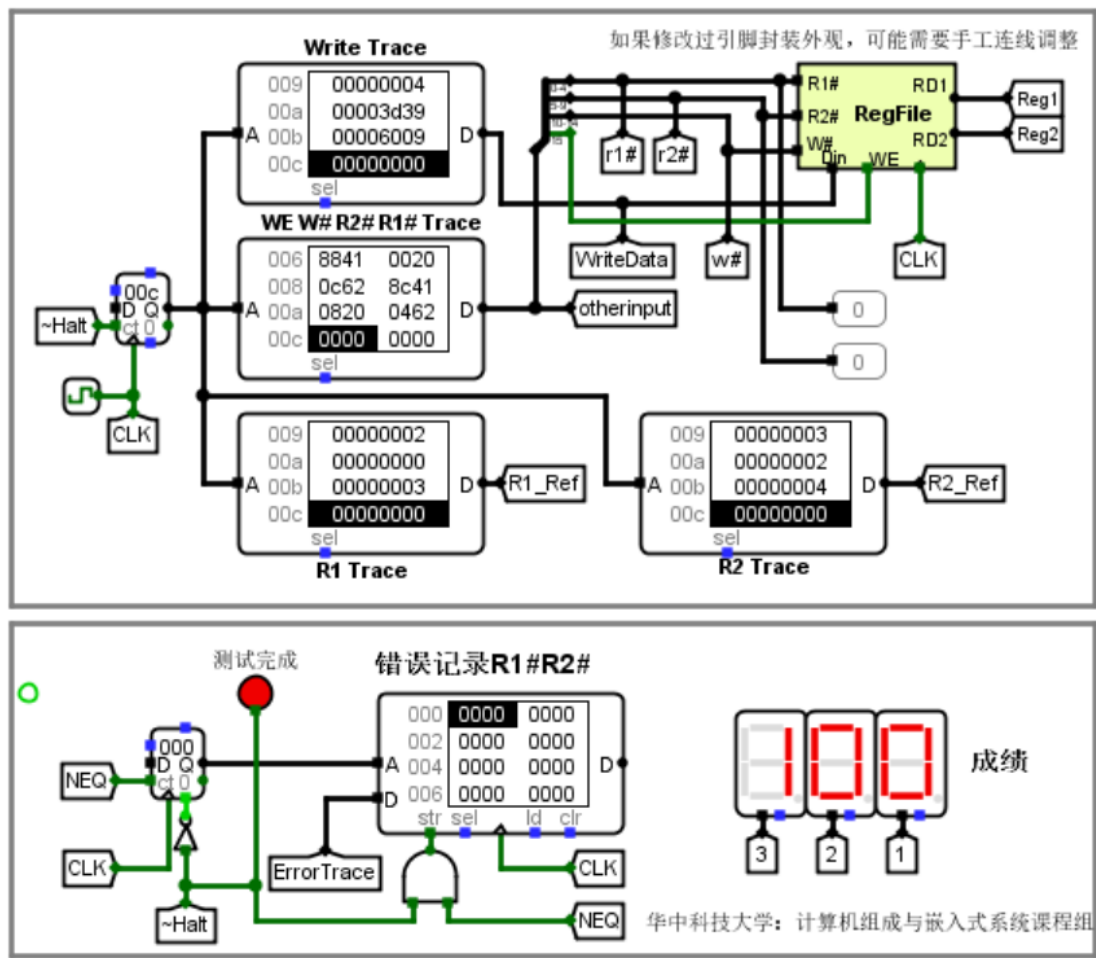


图 7 MIPS 寄存器文件设计实验结果

2.4 Cache 硬件设计实验

本次实验我选择用直接相连映射。实验的具体思路跟直接相连的原理类似，如图 8 所示。

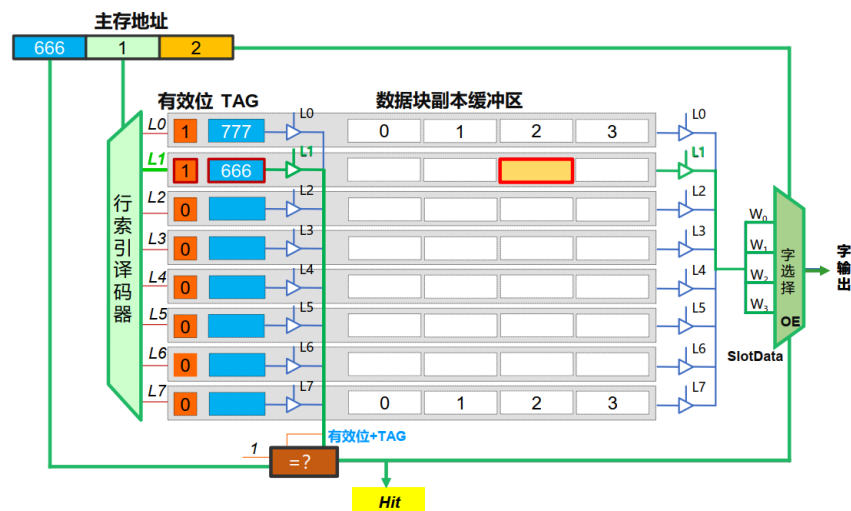


图 8 直接相连映射的原理

Cache 有 n 行，主存的每个区就有 n 行，对应的 index 就有 $\log n$ 位。由于每一个主存地址被映射到单独的某一行，所以我们只需要一个比较器来比较 tag 信号是否相同来判断是否命中。命中后根据块内偏移直接读出数据，而未命中的情况下根据替换算法进行更新。

具体的电路思路跟直接相连映射的原理一样。分命中和不命中，命中就直接读出数据，不命中就更新 cache 当前行的 tag，有效位，以及数据块。具体的电路图如图 9 所示，实验结果如图 10 所示。

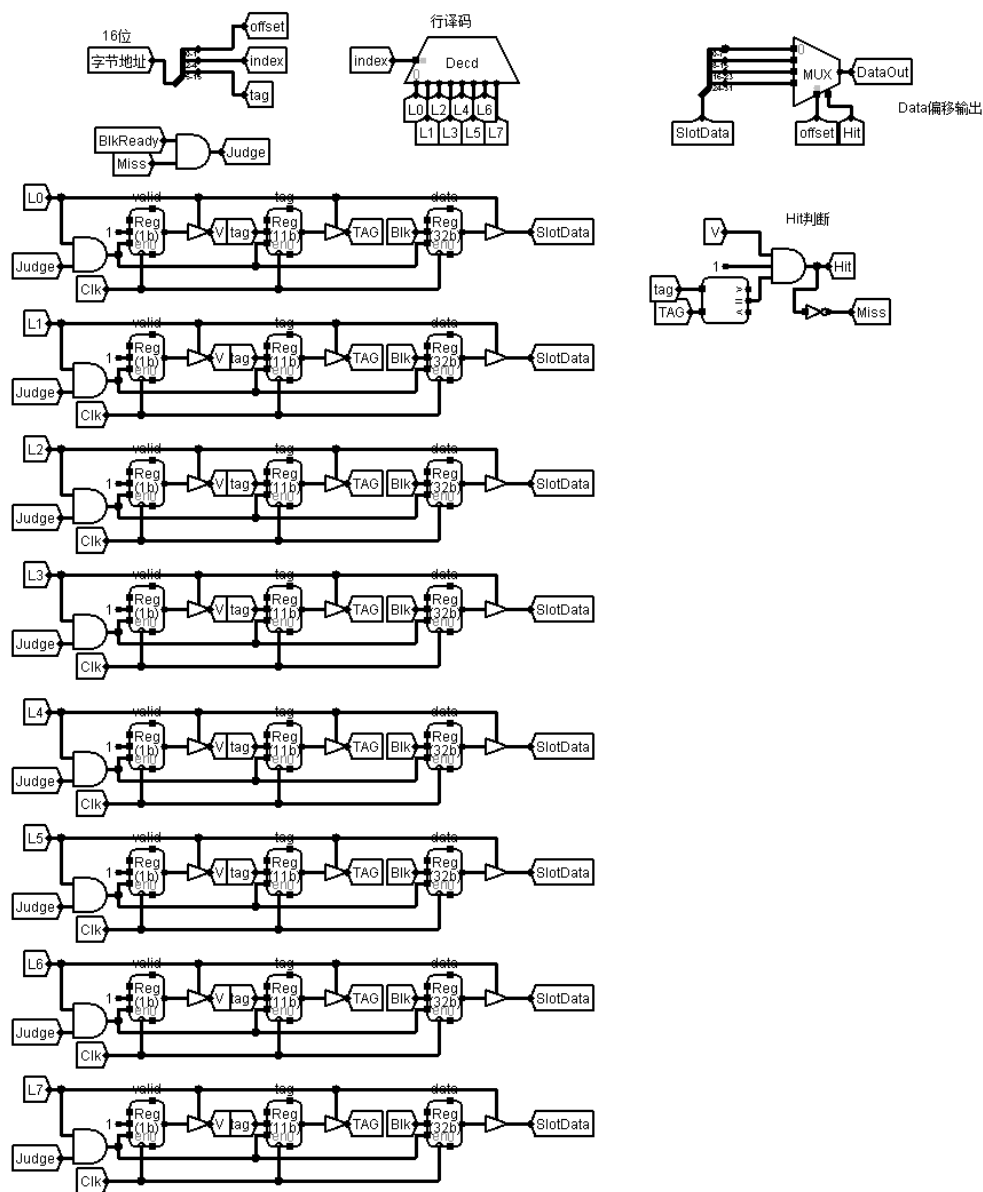


图 9 直接相连映射实验电路图

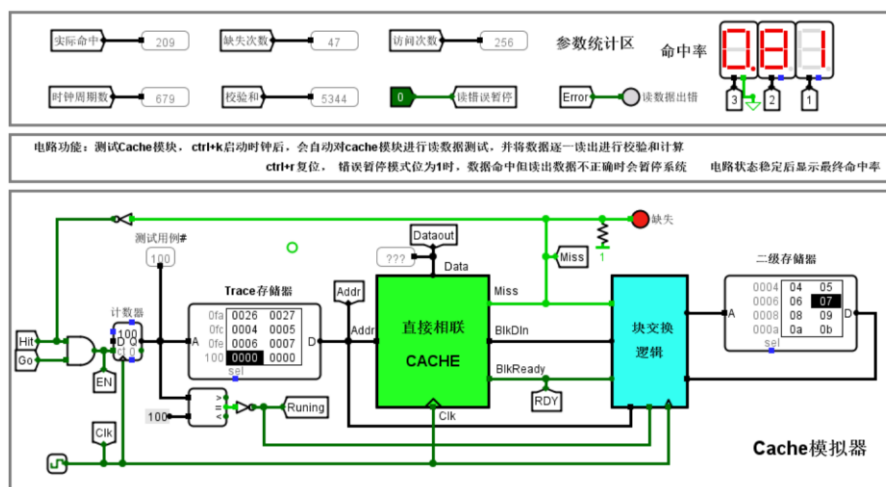


图 10 直接相连映射实验结果

3. 实验中遇到的问题

3.1 存储器扩展实验

一开始的时候，我线路连好了，就直接开始测试了，然后出不来汉字，才意识到字库的数据没有导进去，然后我就去导入了数据，但是导入的数据不是 4K 个，复制的位数搞错了，然后出来的汉字不正确，红灯也亮了。通过询问同学，我知道了正确的数据导入方式，这样才正确。

3.2 MIPS RAM 设计实验

这个实验就是一开始字节，半字，字访问的概念有点模糊，不过通过查阅网上的一些资料，发现它就是一些位数上的控制，没有那么的复杂，然后设计好每个 RAM 的使能信号，就很顺利的完成了。

3.3 MIPS 寄存器文件设计实验

这个实验比较简单，但是我没有注意到要求，就是 0 号寄存器恒为 0，我接的是 Din。得到的分数是 66，如图 11 所示。然后改正后就没问题了。

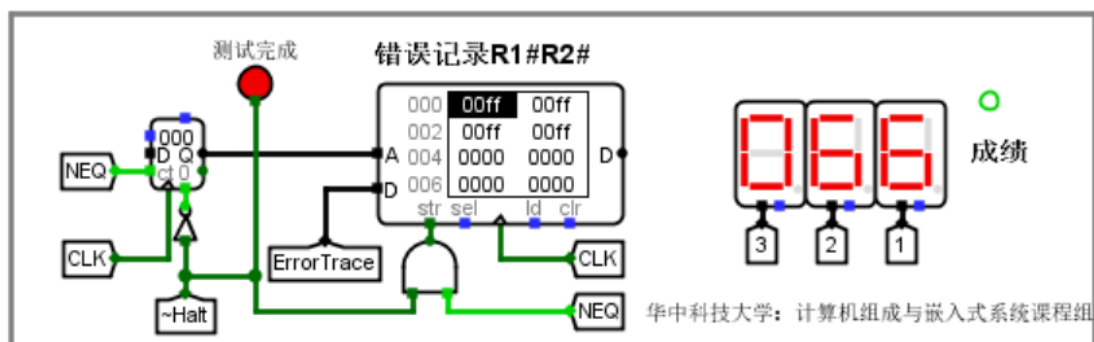


图 11 MIPS RAM 实验遇到的问题

4.4 Cache 硬件设计实验

这个实验比较复杂，虽说选择了不需要替换算法的直接相连映射，但还是有些问题。比如一开始的时候，我直接将 Judge 信号接到了使能端，如图 12 所示。这样是不对的，因为只有当输入的地址选到了这一行，才有使能可谈。然后还有些小问题，比如最后输出数据选择器那里忘记了接 Hit 使能，如图 13 所

示，改正后就没问题了。

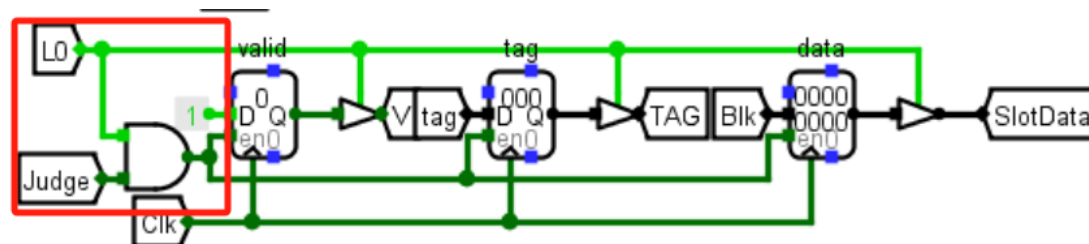


图 12 直接相连映射实验遇到的问题 1

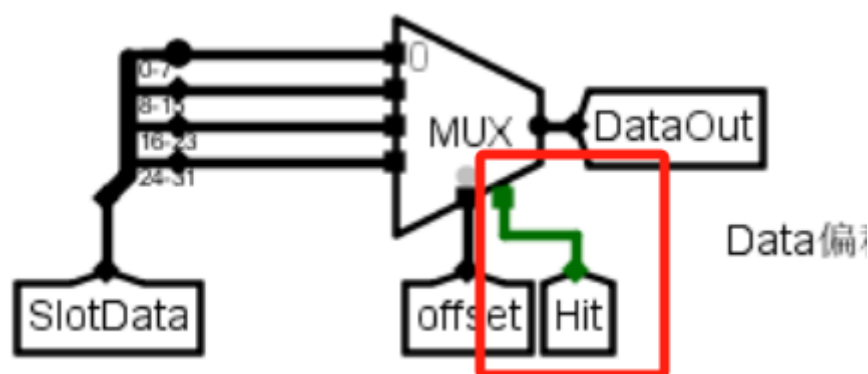


图 12 直接相连映射实验遇到的问题 2

4. 实验总结

本次实验较前几次实验比较困难，一方面是线路复杂了，其次还有通过测试结果不好判断自己的错误，这就更需要细节上的完善。通过这次实验，让我意识到设计好使能端的组合逻辑电路是非常重要的，往往就是这一部分出错。同时，这次试验帮我复习了存储系统，对 cache 的原理有了进一步的了解。