




logisim计组实验九 存储系统设计

原创

Treblez*  于 2020-05-24 15:11:01 发布  7137  收藏 119

分类专栏: [logisim与计算机组成](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/treblez/article/details/106231787>

版权



[logisim与计算机组成](#) 专栏收录该内容

14 篇文章 23 订阅

订阅专栏

原理: [点这里](#)

本章节均已通过测试~有问题请留言

电路文件已经托管至Github, 欢迎star: [点这里](#)

文章目录

字库电路

电路

MIPS RAM设计

原理和要求

电路图

测试

MIPS寄存器文件设计

电路

测试

Cache映射机制与逻辑实现

直接相连映射

电路和讲解

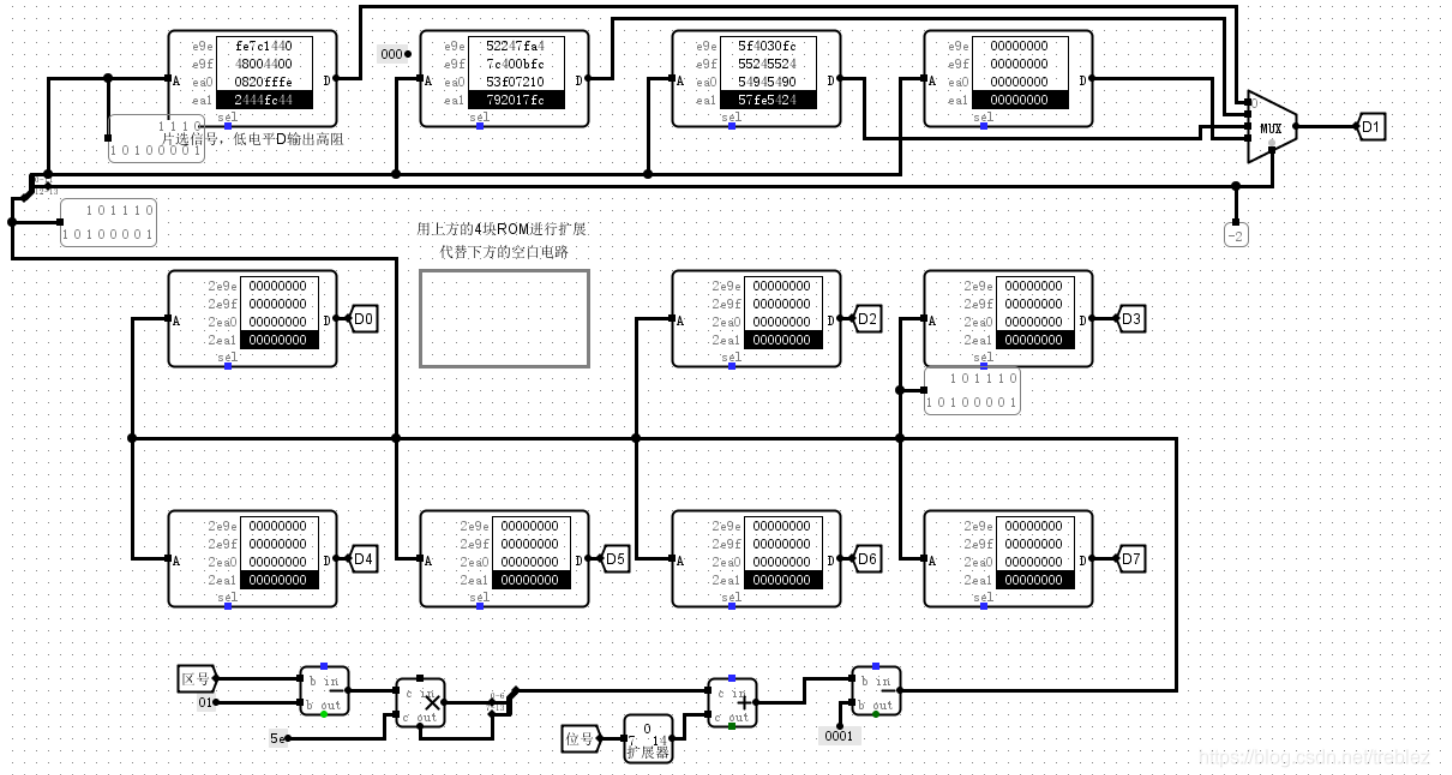
测试

字库电路

电路

字库文件在16点阵字库文件分拆中

电路图



然后注意字库是按顺序放入
我还傻傻写了个均分的程序...

MIPS RAM设计

原理和要求

实验任务：MIPS 存储器设计

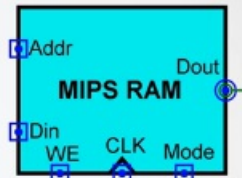
中国大学MOOC

设计支持MIPS Load/Store指令的存储器

- lb lh lw sb sh sw
- 字节/半字/字访问

封装规格

所需器件：多路选择器，译码器，三态门...



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

输入输出总线控制

中国大学MOOC



补充：

Addr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。

这里的难点是如何进行Din分割，和str信号的选取。

首先要理解片选信号和存储单元的关系。

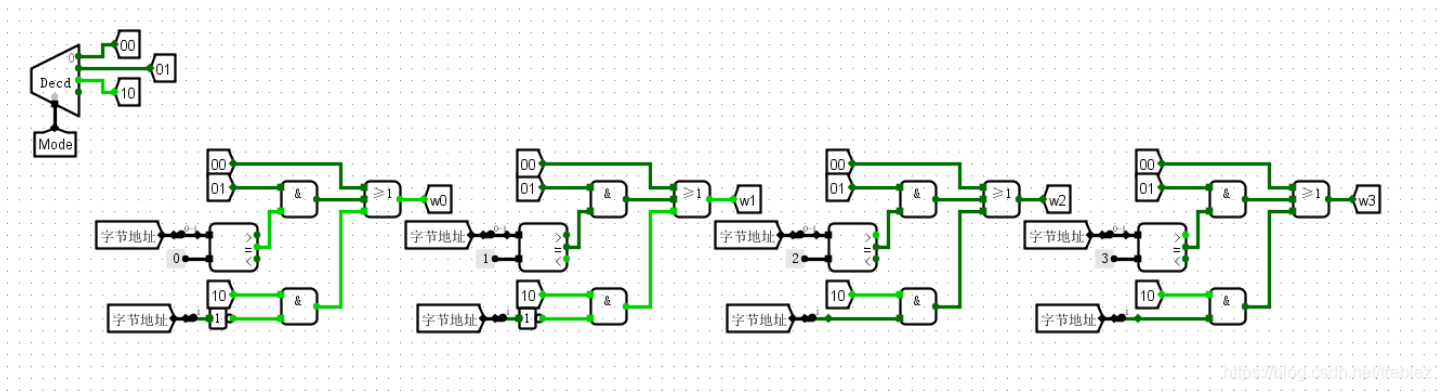
为了避免存储单元中的数据被覆盖，所以要根据Mode和低二位决定4个存储器哪一个需要去存储。

以最低位存储器的str信号来说，字访问的时候str必须高电平；1字节访问的时候当片选信号为0时进行访问；2字节访问的时候，片选信号为0时进行访问。

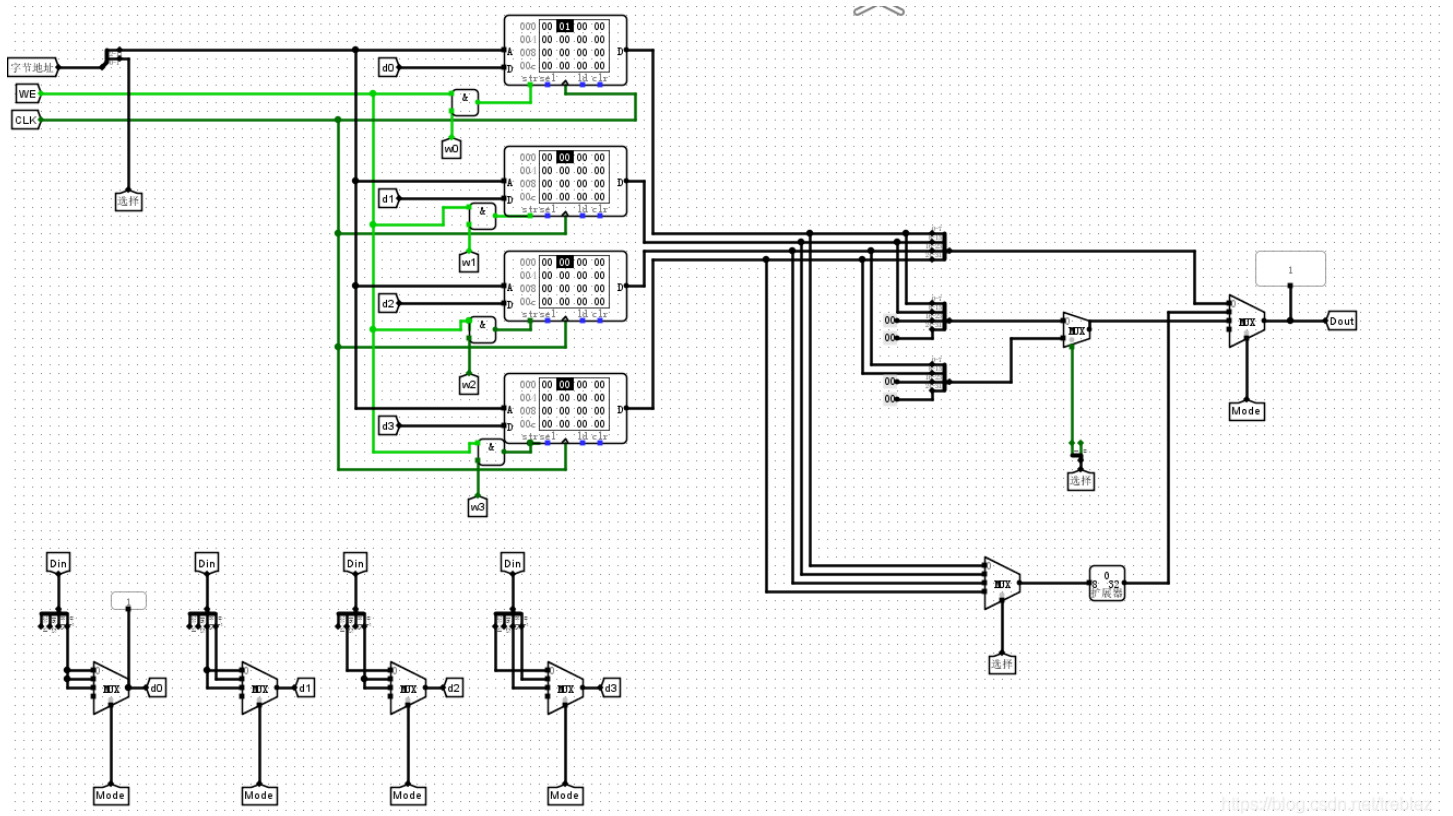
电路图

在这里挣扎了很长时间，教材和课程都过于不详细，朋友们帮忙去mooc上评论一下希望老师提供答案吧。也可能因为我太笨了

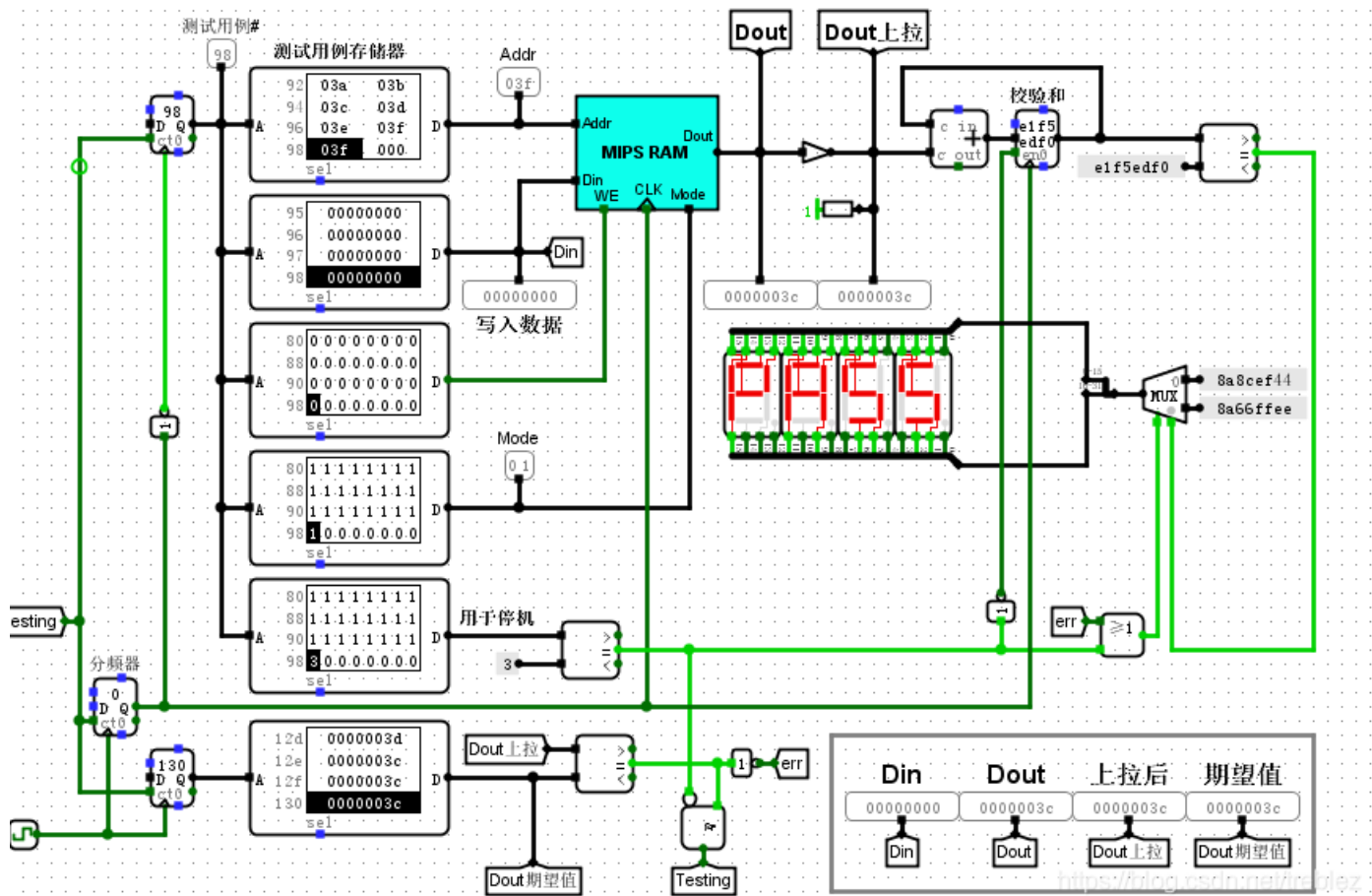
str信号区



其它区域

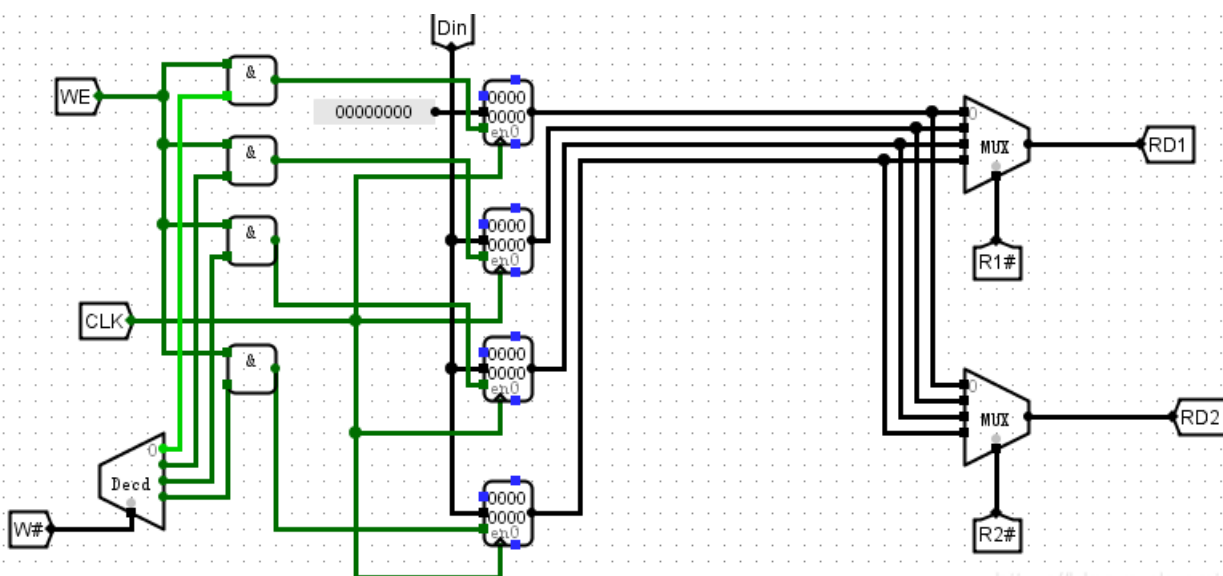


测试

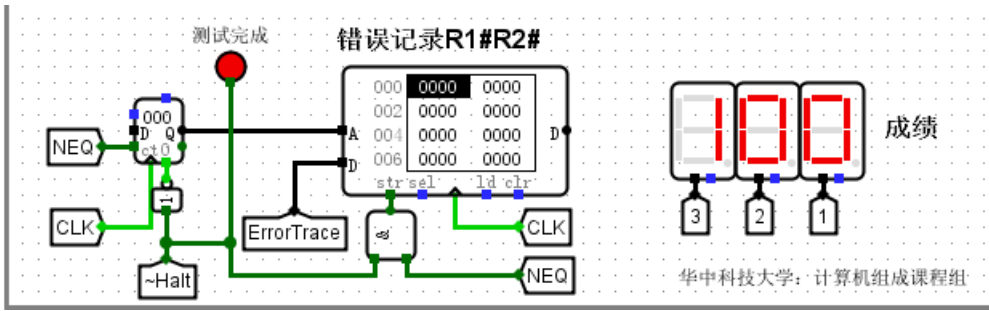


MIPS寄存器文件设计

电路



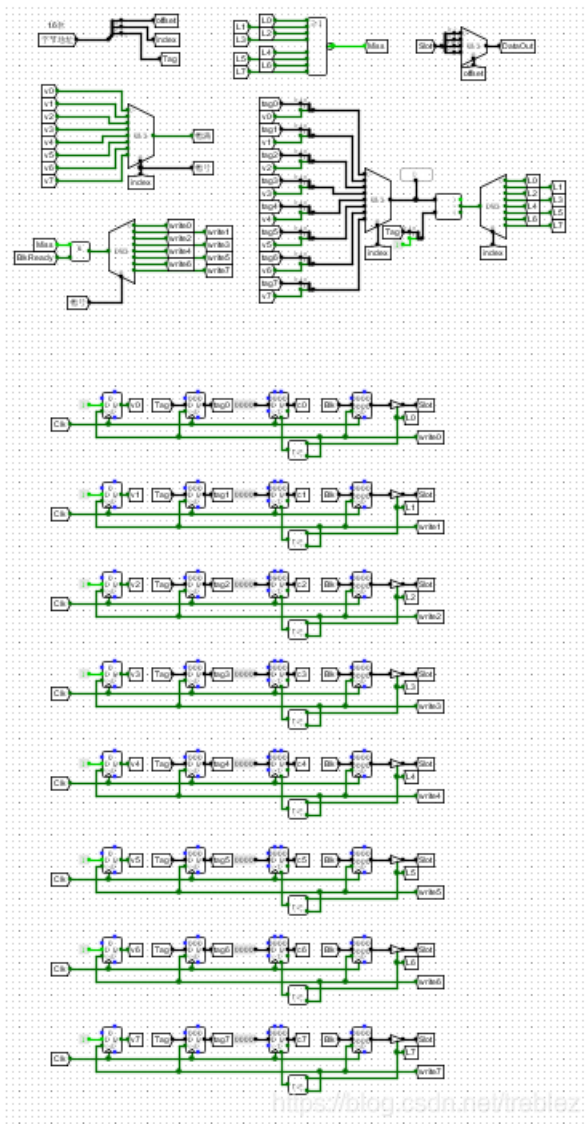
测试



Cache映射机制与逻辑实现

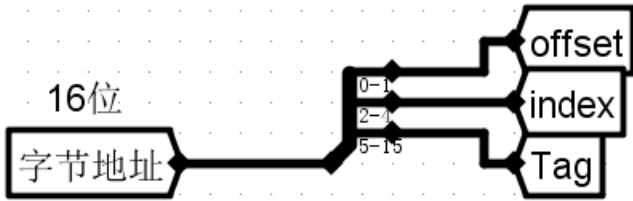
直接相连映射

- Blkready信号用于判断数据块是否准备好，当Blkready信号有效并且时钟到来时，Cache将块数据从BlkDin端口一次性载入对应的Cache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。



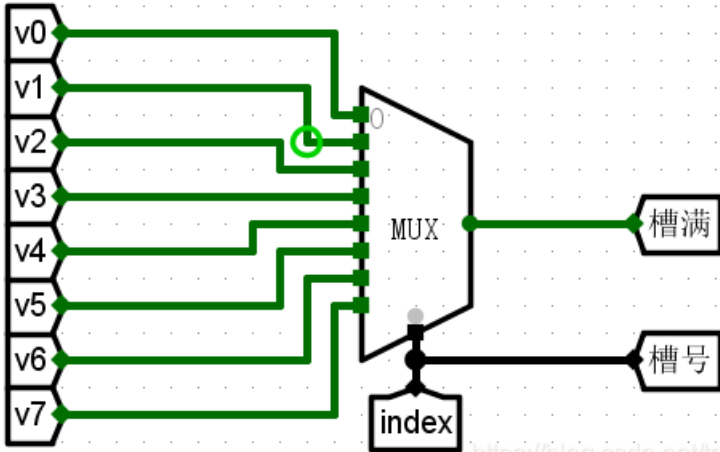
电路和讲解

这个电路实现起来比较麻烦，不过幸好Cache行是可以复用的。
首先，字节地址的分配是这样的：

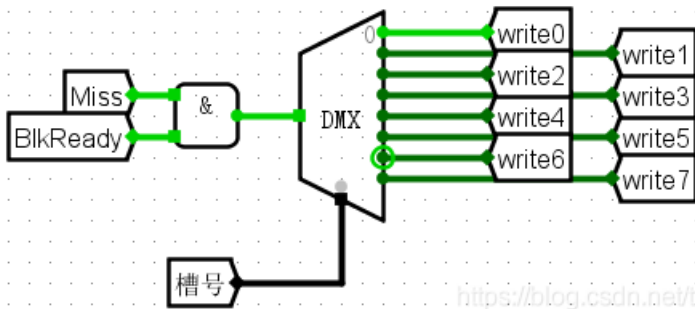


index就是槽号。

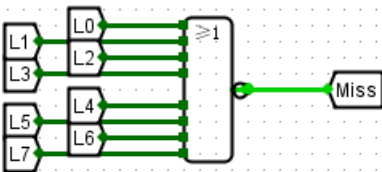
加入一个判断Cache行是否已被写入的多路选择器：



判断是否需要写入的解复用器，当数据块准备好并且Miss（没有命中）时需要进行写入：

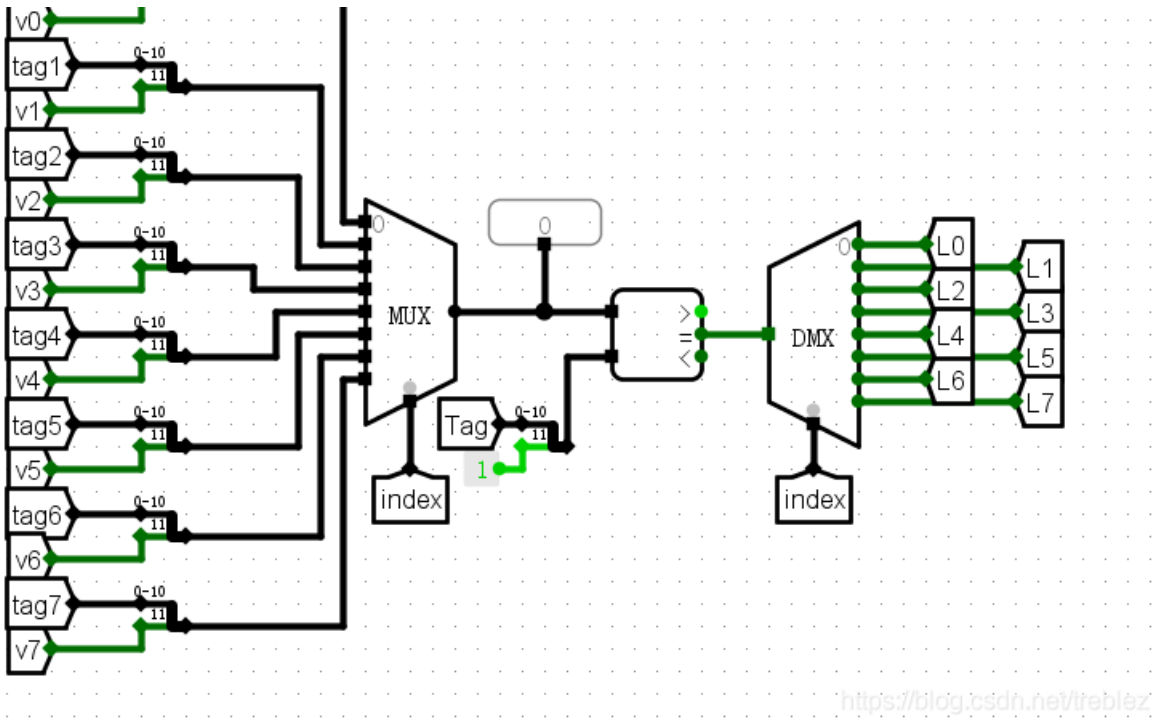


或非门，用于判断是否命中：

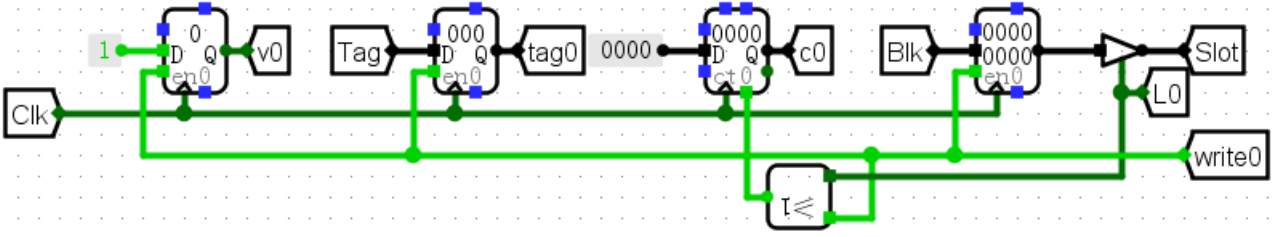


解复用器和比较器，找出相应的Cache行

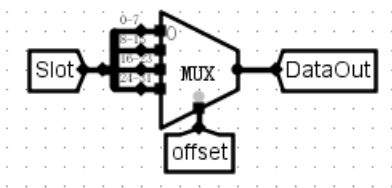




Cache行，使用了三个寄存器和一个计时器，使能端为1时进行数据读入。
 当命中时，L为1，三态门选通，进行数据的写出。

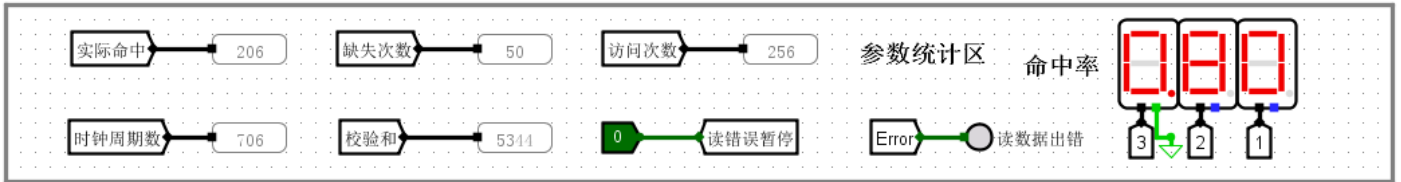


Cache行偏移，得到数据。

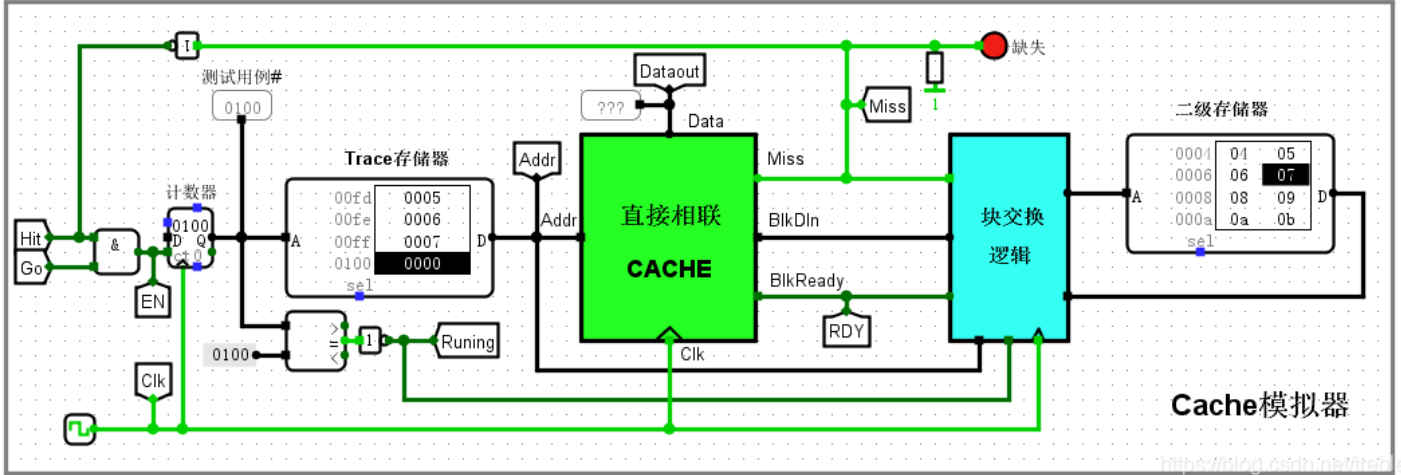


测试

命中率稳定在了0.8



电路功能: 测试Cache模块, ctrl+k启动时钟后, 会自动对cache模块进行读数据测试, 并将数据逐一读出进行校验和计算
ctrl+r复位, 错误暂停模式位为1时, 数据命中但读出数据不正确时会暂停系统
电路状态稳定后显示最终命中率



<https://blog.csdn.net/itef...>