

1.4.4 实验思考

- (1) 零号寄存器的值恒零，具体实现是如何实现的，如何实现成本最优，MIPS 指令集中为什么要引入一个恒零的寄存器？
- (2) 本实验中利用译码器和解复用器均可实现 MIPS 寄存器文件中的写入控制，尝试用两种不同的方案实现。

1.5 Cache 硬件设计实验

1.5.1 实验目的

学生掌握 cache 实现的三个关键技术：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。

1.5.2 实验内容

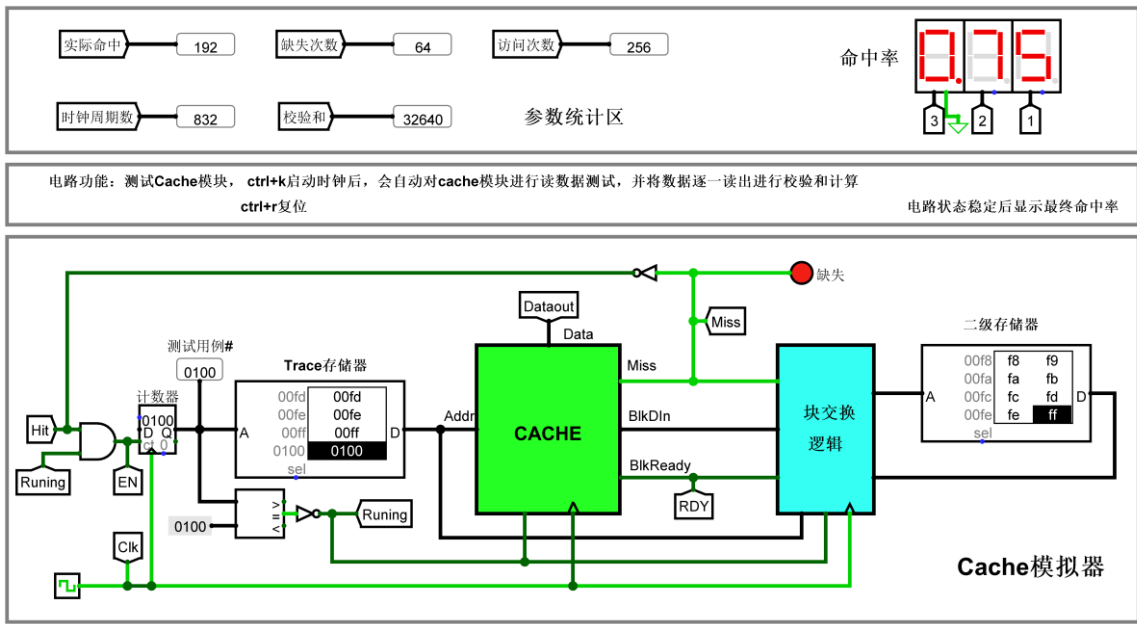


图 1.10 Cache 模块自动测试电路

图 1.10 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，此时所有存储访问的 cache 命中率将会在右上角 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。

实验要求：根据图 1.11 所示的 cache 模块输入输出引脚示意图，结合表 1.2 的引脚功能说明，实现只读 cache 模块，该 cache 模块共包括 8 个 cache 行，每个数据块包含包括 4 个字节共 32 位数据。

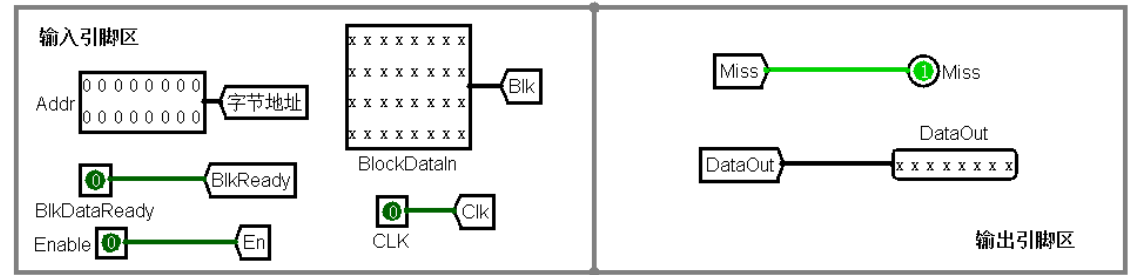


图 1.11 Cache 模块输入输出引脚

Cache 映射策略和调度策略共有如下几种组合：

- 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。

请在以上三个选项中人选一个进行电路实现，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？

表 1.2 芯片引脚与功能描述

引脚	输入/输出	位宽	功能描述
Addr	输入	16	主存地址
BlkDataIn	输入	32	块数据输入
BlkDataReady	输入	1	块数据准备就绪
Enable	输入	1	使能信号，1：工作；0：输出高阻态
CLK	输入	1	时钟输入
Miss	输出	1	1：数据缺失；0：数据命中
DataOut	输出	8	数据输出

实验提示：为简化电路绘图工作量，可以将 cache 行模块化后进行复制，具体可以参考图

1.12 的 cache 行参考设计，图中存储器件全部采用寄存器实现，valid 数据位通过三态门输出到 V0，标记位通过三态门输出到 Tag0，淘汰标志位通过三态门输出到 C0，三个三态门全部由组索引信号 Set0 进行控制，也就是说只有当前第 0 组被选中时对应数据才能输出，同一组中所有标志位标记位均同时输出到多路并发比较电路；数据块副本通过三态门输出到 SlotData，输出控制由组索引信号和行选中信号 L0 逻辑与后得到，也就是说只有当前组第 0 组被选中且第 0 行命中时才将数据副本输出。图中 R0 信号表示当前组被替换的信号，如果此信号为 1，且组索引信号 Set0 有效则所有寄存器使能端为 1，时钟到来新的有效位、标记位、淘汰标志位、数据块副本均载入新数据，实现新数据块的载入。

完成一个 cache 槽设计后，可直接复制该电路 8 份，分别修改其中信号的隧道标签使其构成有效的电路，然后利用外围电路实现整个 cache 的功能。

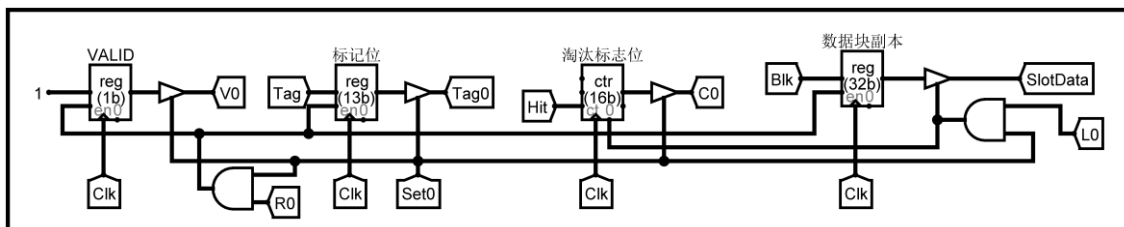


图 1.12 cache 行参考设计

1.5.3 实验思考

- 1) 不同地址映射方式在具体实现时硬件开销有哪些区别？
- 2) 如果采用软件实现 cache，那种映射方法更有效？LRU 算法采用什么数据结构实现更方便？软件实现中不可能实现全相联查找的并发查找机制，如何提升查找速度？