

# 实验报告

计算机科学与技术

19307130296

孙若诗

## 缓存描述

1. 本实验实现了写分配、回写的四路缓存。
2. tag为26位, index为2位, offset为4位, 因此缓存中共有4个cache set, 每个cache set有4条cache line。每条cache line存储4条指令, 也即4字、16字节、128位, 地址3-2位表示指令在cache line中的偏移量。若考虑缓存性能, 设计需要进一步调整。
3. 替换策略为随机替换。出于硬件效率和便于检验正确性考虑, 采用伪随机数, 将tag的1-0、3-2、5-4、7-6位异或, 得到一个2位地址, 作为缓存不命中时替换的地址。
4. 每条cache line记录4个值: tag、dirty、valid、data。FETCH后, valid为1, dirty为0。FLUSH后, valid为0, dirty为0 (因为FLUSH一定是为了载入新的cache line)。修改data后, valid为0, dirty为1。
5. 缓存的状态为: IDLE、SEARCH、HIT、MISS、FETCH、FLUSH、READY, 分别表示空、寻找缓存、命中、不命中、读内存、写内存、读完毕。处理具体事件经过的状态如下:
6. 读hit: IDLE->SEARCH->HIT->READY->IDLE
7. 读miss: IDLE->SEARCH->MISS(->FLUSH)->HIT->FETCH->READY->IDLE
8. 写hit: IDLE->SEARCH->HIT->READY->IDLE
9. 写miss: IDLE->SEARCH->MISS(->FLUSH)->HIT->FETCH->READY->IDLE

## 其他

1. 在关联度为 $2^k$ 的缓存中实现 LRU 算法, 每个cache set要记录所有cache line的出现次序, 有 $2^k$ 个cache line, 出现次序也需要 $2^k$ 位表示, 共 $2^{2k}$ 位额外信息。
2. 缓存重置时, 在我的实现中不需要将数据重置, 但是需要把cache line状态相关量清零。因为dirty和valid都为0时一定会进入FETCH状态和内存取得同步, 所以不重置数据不会有影响。不重置则需要注意处理数据初始化问题, 重置则需要考虑reset时间不够的问题。
3. 相邻的流水线阶段 X 和 Y 间有一对握手信号 `valid` 和 `ready`。阶段 X 在阶段 Y 之前。`valid` 表示阶段 X 已经执行完成, 可以进入到阶段 Y。`ready` 表示阶段 Y 已经执行完成, 阶段 X 的数据可以进入。
  - 假设流水线只有最后一个阶段可能需要多个周期, 其余阶段只需要一个周期。最后一个阶段使 `ready` 为0, 则之前每个阶段的 `ready` 都相应地为0, 可阻塞第一阶段数据进入, 延时为最后一阶段所需的周期数。
  - 将最后一个阶段切分为多个阶段, 使每个阶段周期数较小, 可减少上一问中握手信号的延时。

## 实验现象

1. `lab3a` `verilator` 模块级测试

```
VCacheTop/vmain
./build/gcc+optimized/mycpu/VCacheTop/vmain
[OK] void (0ms)
[OK] reset (0ms)
[OK] fake load (55ms)
[OK] fake store (60ms)
[OK] naive (0ms)
[--] akarin~ (skipped)
[--] strobe (skipped)
[OK] ad hoc (0ms)
[OK] pipelined (0ms)
[OK] memory cell (0ms)
[OK] memory cell array (0ms)
[OK] cmp: word (14ms)
[OK] cmp: halfword (17ms)
[OK] cmp: byte (50ms)
[OK] cmp: random (131ms)
[OK] memset (345ms)
[OK] memcpy (392ms)
[OK] load/store repeat (195ms)
[OK] backward memset (454ms)
[OK] backward load/store (525ms)
[OK] random step (991ms)
[OK] random load/store (1125ms)
[OK] random block load/store (2094ms)
"std::sort": bingo!
[OK] std::sort (2790ms)
"std::stable_sort": bingo!
[OK] std::stable_sort (2299ms)
"heap sort": bingo!
[OK] heap sort (3314ms)
"binary search tree": bingo!
[OK] binary search tree (3718ms)
(info) 27 tests passed.
```

```
./build/gcc+optimized/mycpu/VCacheTop/vmain -p 0.5
[OK] void (0ms)
[OK] reset (0ms)
[OK] fake load (73ms)
[OK] fake store (68ms)
[OK] naive (0ms)
[--] akarin~ (skipped)
[--] strobe (skipped)
[OK] ad hoc (0ms)
[OK] pipelined (0ms)
[OK] memory cell (0ms)
[OK] memory cell array (0ms)
[OK] cmp: word (13ms)
[OK] cmp: halfword (29ms)
[OK] cmp: byte (43ms)
[OK] cmp: random (163ms)
[OK] memset (367ms)
[OK] memcpy (636ms)
[OK] load/store repeat (306ms)
[OK] backward memset (638ms)
[OK] backward load/store (835ms)
[OK] random step (1765ms)
[OK] random load/store (1714ms)
[OK] random block load/store (3017ms)
"std::sort": bingo!
[OK] std::sort (4188ms)
"std::stable_sort": bingo!
[OK] std::stable_sort (3912ms)
"heap sort": bingo!
[OK] heap sort (5994ms)
"binary search tree": bingo!
[OK] binary search tree (6167ms)
(info) 27 tests passed.
```

2. 上板

