

# cache 实验报告

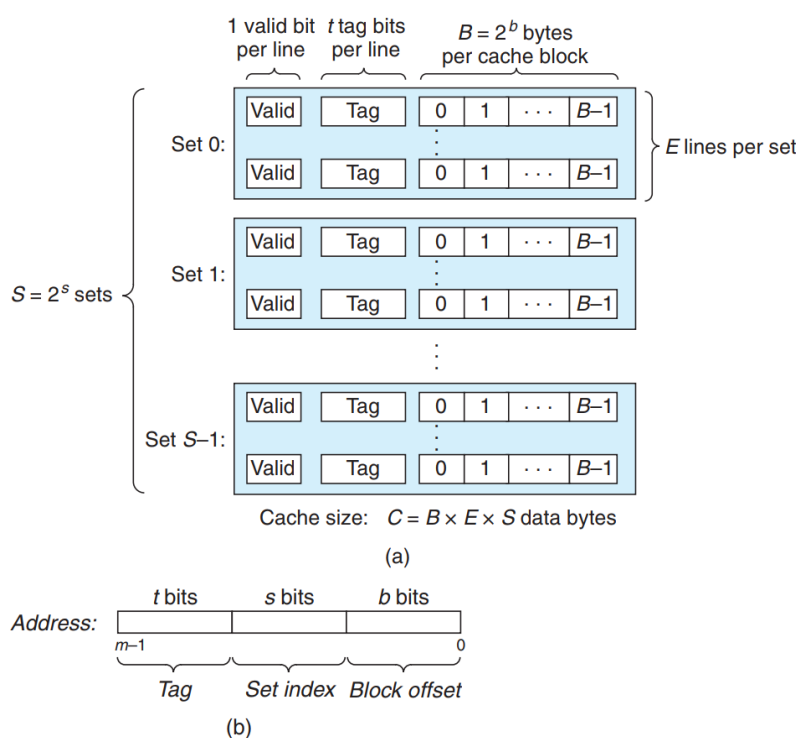
## 一、设计原理

高速缓存(cache)用于加速CPU和主存之间的数据传递。Cache的结构如下：（图片来源：拔尖班课程讲义）

Figure 6.25

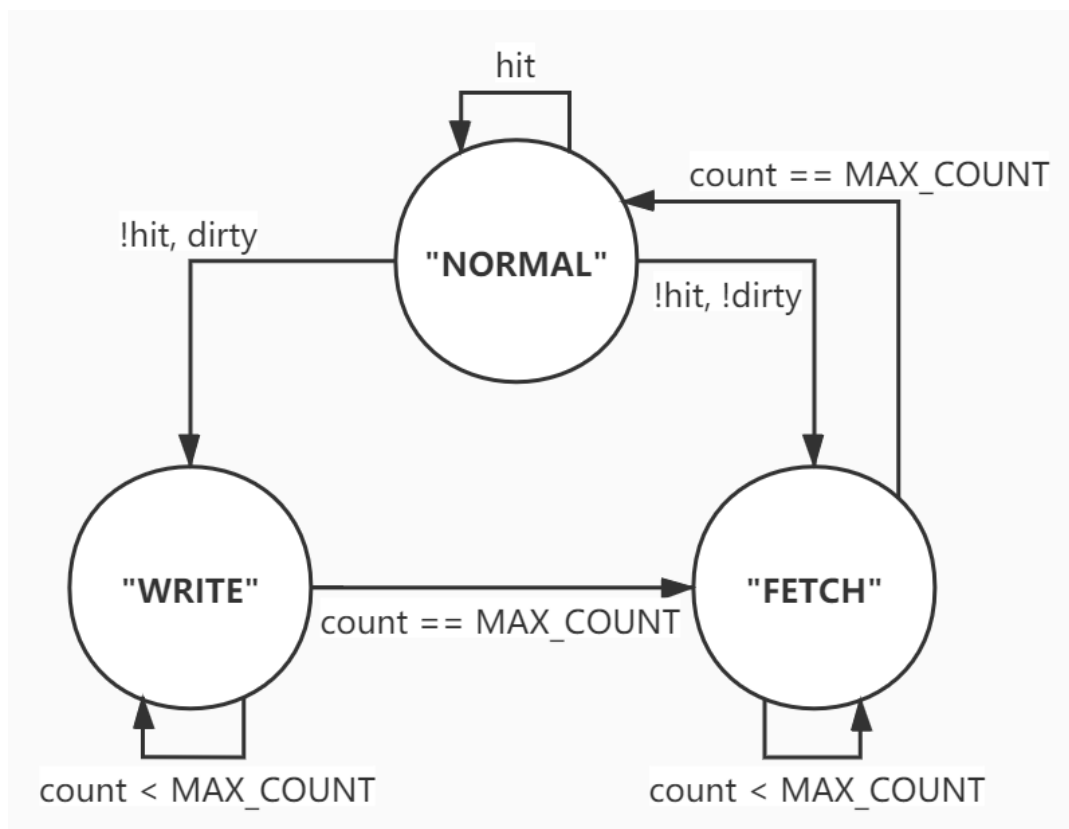
### General organization of cache ( $S, E, B, m$ ).

(a) A cache is an array of sets. Each set contains one or more lines. Each line contains a valid bit, some tag bits, and a block of data. (b) The cache organization induces a partition of the  $m$  address bits into  $t$  tag bits,  $s$  set index bits, and  $b$  block offset bits.



32位访存地址按位分为三个部分：标签(tag)、组索引(set index)、块位移(block offset)。我们定义每部分的宽度分别为CACHE\_T、CACHE\_S、CACHE\_B（在cache.svh中定义）。另外还定义相联度为常量CACHE\_E，其含义是每组包含CACHE\_E个缓存行(line)。

如果CACHE\_B为4且CACHE\_S为2，则每个cache行存储16个字节或4字，且整个cache分为4个组。



Cache的工作原理如上图所示。给定32位地址时，高速缓存首先读取该地址中的组索引(set index)字段 `i`，然后检查第*i*组 `set[i]` 是否存在标记(tag)匹配的行(line)。如果存在则缓存命中(cache hit)，否则缓存未命中(cache miss)，我们需要从内存中载入一个块(block)到缓存行中。

在现实世界中，一台32位计算机上的内存在一个时钟周期内最多可以提供32位数据。因此，若从内存中读取数据块，则需要大量的时钟周期才能获取所有数据。

在高速缓存未命中的情况下，如果所选集合中充满了有效(valid)行，并且我们选择了要替换的脏(dirty)行，则应首先将数据写回(writeback)到内存中，然后将新数据加载到该行中。选择要替换的行的方式有多种，如最久未使用(LRU)、随机替换(RANDOM)、先进先出(FIFO)；为简单起见，本人选用随机替换策略。

这一工作过程可以由上图所示的有限状态机(FSM)来表示。实际设计Verilog程序时，也根据该状态机来进行。

## 二、仿真测试

我们采用自动化的测试代码（代码见项目中的cpu\_tb.sv）和拔尖课程提供的11笔测资进行测试。这11笔测资分别是：

ad hoc：课本7.6.3节提供的测资，覆盖课本上实现过的所有指令（R类指令、addi、sw、lw、j），无循环和函数调用。

bisection：二分查找，考察循环（循环过程用到bne、beq、j指令）和移位指令sll、sra。

bubble sort：冒泡排序，嵌套循环。

en & clear：见下方问题讨论部分。

factorial：分解质因数，考察递归（函数调用过程用到jr、jal指令）。

i-type：针对立即数类型(i-type)指令进行考察，主要考察立即数类型指令的实现是否严格正确。特别地，考察addi是否是零扩展而andi、ori、slli是否是符号位扩展。

mutual recursion: 背景是编译原理文法分析过程, 双重递归 (即两个函数互相递归调用)。

gcd: 求最大公约数, 考察循环。

quick multiply: 快速乘, 考察位移和循环。

testjr: 是一笔直接考察jr指令的测资, 主要考察jr指令出现数据冒险时能否正确处理。

random write: 随机写入数据, 考察缓存是否正确工作。

(具体测资文件, 请见提交的项目目录 /benchtest 文件夹下的内容)

我的带cache的MIPS流水线处理器能通过上面所有的测资, 测试结果如下:

```
SIMULATION - Behavioral Simulation - Functional - sim_1 - cpu_tb

Tcl Console x Messages Log Find in Files

===== Test: en & clear =====
WARNING: file ../../../../benchtest/en & clear/en & clear.data could not be opened
===== In init =====
13 instructions in total
===== In runtime checker =====
successfully pass runtime checker
[OK] en & clear

===== Test: i-type =====
WARNING: file ../../../../benchtest/i-type/i-type.data could not be opened
===== In init =====
18 instructions in total
===== In runtime checker =====
successfully pass runtime checker
[OK] i-type

===== Test: random write =====
WARNING: file ../../../../benchtest/random write/random write.data could not be opened
===== In init =====
17 instructions in total
===== In runtime checker =====
successfully pass runtime checker
[OK] random write

[Done]

CPI = 1.993527

$finish called at time : 406960 ns : File "C:/Users/jasha/Desktop/MIPS_Pipeline_with_Cache/t
```