南开大学

Nankai University

南　　开　　大　　学

计　算　机　学　院

并行程序设计期末实验报告

第四、五章习题

冯思程

年级：2021 级

专业：计算机科学与技术

指导教师：张金

2023 年 5 月 31 日

目录

一、 习题

(一) 题目

小冯同学最近学习完了计算机组成与设计：硬件软件接口这本教材的第四，五章，组成原理之神许诺小冯同学如果可以完全掌握这两章的知识可以奖励小冯同学一个愿望，小冯同学十分想获得这个愿望，但他对一些内容仍存在困惑，希望同学们可以为他解答困惑，帮助他获得神的奖励。

1. 第四章

单周期数据通路

- 1）请你解释边沿触发的时钟的含义，并说出 MIPS 核心子集包含哪些条指令
- 2）如图1所示，它是一个数据通路的实现图（没有包括跳转），在图中做出了一些提示性的标注，请你根据这个图从左到右说出数据通路指令执行的五个阶段，并说出 R 型指令的执行过程并画出对应的局部数据通路图。

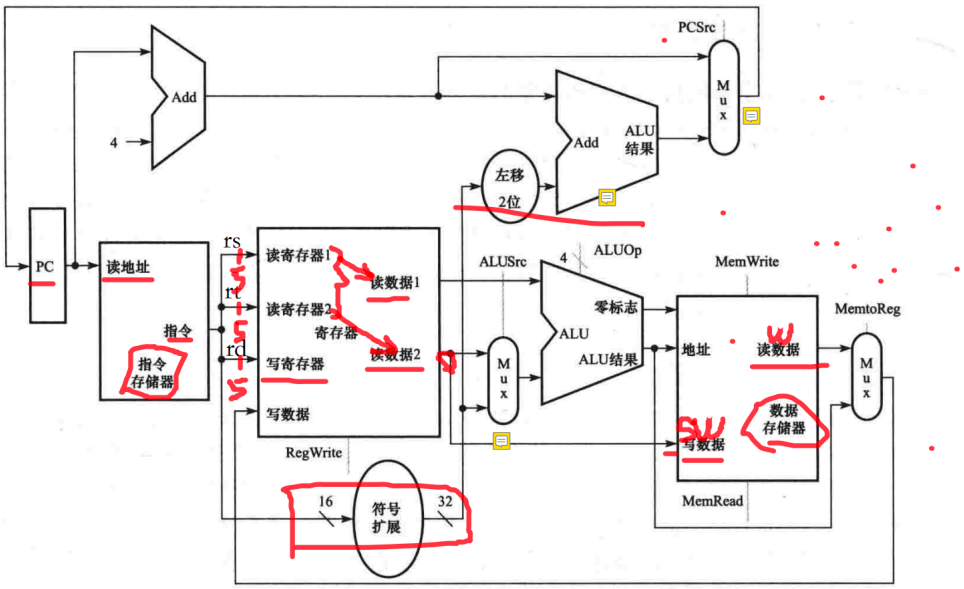

图 1: 数据通路实现图

- 3）请你总结 MIPS 核心子集涉及到的指令的指令周期阶段，以表格方式给出答案。

单周期控制单元

- 1）小冯同学遇到了难题，如果你可以为他解释一下 R 型指令（add $t2, $t3, $t6）的执行过程，过程要设计对控制信号的说明。

流水线数据通路与控制、流水线冒险

- 1）小冯同学对流水线与时钟周期有一些困惑，请你帮助他理解知识点，下面分别给出数据通路不同阶段的延迟和不同指令类型在程序中的比率：

IF	ID	EX	MEM	WB
150ps	300ps	250ps	100ps	200ps

表 1: 数据通路不同阶段延迟表

ALU	Lw	Sw	Beq
30%	10%	25%	35%

表 2: 不同指令类型比率表

- 问题 1: 非流水线处理器的时钟周期?
- 问题 2: 流水线处理器的时钟周期?
- 问题 3: 分别说明 lw 指令在非流水线处理器和流水线处理器的延迟时间。
- 问题 4: 现在小冯同学感觉流水线的时钟周期有些长，想对流水线级进行划分，划分规则是从 1 级变成 2 级，延迟各一半，现在可以挑选两个两个阶段进行划分，请你回答是哪两级，并回答划分之后的处理器时钟周期。
- 问题 5: 数据存储器的利用率? 数据存储器写端口的利用率? 寄存器堆的写端口的利用率?

- 2）
  - 问题 1: 给出连续执行的两条指令:
    sub $s2, $t3, $t6
    add $t2, $s2, $t5
    请回答这两条指令连续执行会发生什么并解释，以及要怎么解决。
  - 问题 2: 给出两条指令:
    lw $s2, 20($t6)
    add $t2, $s2, $t5
    请回答这两条指令连续执行会发生什么并解释，以及要怎么解决。
  - 问题 3: 解释一下结构冒险的含义。
  - 问题 4: 解决控制冒险有两种方法，请你分别进行解释。
  - 问题 5: 下面给出一个指令序列:
    8 sub $s2, $t3, $t6
    12 beq $s1, $t2, 6
    16 add $s3, $t5, $t7
    ∘∘∘∘∘
    40 lw $s2, 20($t1)
    假定流水线对分支不发生进行了优化，并且分支的执行提前到流水线的 ID 级，请你试着说明该指令序列在分支发生时候的执行情况。
- 3）判断题：结构冒险可以通过增加支持硬件解决。
- 4）判断题：控制冒险的解决方案可以选择不同预测方法。

异常、ARM 与 x86 处理器、指令级并行矩阵乘法

- 1）请你区分一下异常和中断。
- 2）分别说明一下 ARM 和 x86 处理器的流水线结构。
- 3）对推测和推测的技术进行简要说明。

2. 第五章

存储器技术概要

- 1）从顶层开始，对存储器层次结构进行说明，说明包括名字、技术、特点等。
- 2）对局部性原理进行说明。

高速缓存 cache

- 1）一个程序的访存次数是 20，其中直接在 cache 中找到数据有 12 次，已知命中时间是 1T，判断 cache 没有数据进而从主存中取回数据的时间是 101T。现在你回答命中率是多少？缺失时间是多少？访存阻塞周期数是多少？
- 2）一个直接映射的 cache，有 32KiB 的数据，一个块的大小是 8 个字，地址是 32 位，请问这个 cache 一共需要多少位。
- 3）在面对 cache 缺失处理时，一共有三种处理方法请你一一解释。
- 4）处理器时钟周期的时间为 2ns, 缺失代价是 100 个时钟周期，缺失率为每条指令 0.05 次缺失，cache 访问时间（包括命中判断）为 1 个时钟周期，假设读操作和写操作的缺失代价相同，并且忽略其他写阻塞，请计算 AMAT
- 5）一个 cache 有 1024 个块，块大小是 8 个字，地址为 32 位,，请你分别计算在四路组相联和全相联情况下，组数和标记位数。
- 6）假定我们的处理器基本的 CPI=1. 所有访问在一级 cache 中均命中，时钟频率为 5GHz。假设缺失代价的时间是 100ns。一级 cache 中每条指令缺失率为 4%。如果增加二级 cache, 命中或缺失访问的时间都是 10ns, 而且容量大到必须使访问主存的缺失率减少到 1%, 这时的处理器速率能提高多少？

虚拟存储器

- 1）假设虚拟内存有 4GB，物理内存有 1GB，请问他们分别有多少位；假设页是 4KB，请问页内偏移是多少位，虚拟页号是多少位，物理页号是多少位。并说明虚拟页号怎么映射到物理页号，并说明虚拟地址怎么映射到物理地址。
- 2）请你解释脏位和引用位的作用。
- 3）说明 TLB 中的内容，并说明带 TLB 的一次访问的整个过程。

可靠性与校验码

- 1）一批出场的磁盘的 MTTF 是 20w 小时，评价他们的可靠性怎么样，并请计算这批磁盘的 AFR。如果这批磁盘的 MTTR 是 200 小时，那么请你评价这批磁盘的可用性怎么样。
- 2）假定存在某个单字节数据 10010010，首先写出对应的汉明纠错码，然后把第 10 位取反，说明纠错码如何找到并纠正该错误。（默认为偶校验）

虚拟机、cache 控制器、分块加速、ARM 与 x86 存储器层次结构

- 1）解释虚拟机的含义，并说明什么可以实现不同操作系统间的资源隔离和控制。
- 2）简要说明分块加速的原理。
- 3）说明 ARM 和 x86 的存储器层次结构。

(二) 覆盖知识点

1. 第四章

1. 数字逻辑基础
2. MIPS 核心子集
3. 数据通路部件
4. ALU 控制单元
5. 主控制单元
6. 流水线数据通路与控制概述
7. 流水线数据通路
8. 流水线控制单元
9. 多种流水线冒险：结构冒险、数据冒险、控制冒险
10. 带控制的流水线数据通路
11. 异常
12. ARM 和 x86 处理器
13. 指令级并行
14. 矩阵乘法

2. 第五章

1. 存储器层次
2. 局部性原理
3. 访存性能
4. cache 基础
5. cache 改进
6. cache 相关计算
7. cache 一致性
8. 内存是磁盘的 cache

9. 页表、快表
10. 可靠性、可用性
11. 奇偶校验码
12. 汉明校验码
13. 虚拟机
14. cache 控制器
15. 分块加速
16. ARM 和 x86 存储器层次结构

(三) 设计思路

本次习题的设计思路是我在对两章内容进行划分整理后进行出题的，可以发现在出题的部分我已经将每道题划分了所属的知识点模块，这样不仅更加方便根据题来更有针对性的复习，也可以根据题目的所属知识点模块来进行题目的快速解答。同时，为了增加题目的趣味性，我为题目设置了一个情景，小冯同学在面对神的奖励的时候对同学们发出的援助请求。然后具体题目，我参照教材和上课所学习的知识结合进行题目的编写。以上就是我全部的设计思路。

(四) 答案与解析

1. 第四章

单周期数据通路

- 1）答：边沿触发的时钟的含义是只有在时钟信号发生变化的时候（上升沿或者下降沿），才能向状态单元写入数据。MIPS 核心子集包含的 9 条指令分别是：add、sub、and、or、slt、lw、sw、beq、j。
- 2）答：首先回答数据通路的五个基本阶段：IF 取指令、ID 译码和读寄存器、EX 运算、MEM 访存与分支、WB 写回。然后说出 R 型指令的执行过程：首先从 PC 取到要执行指令地址（R 型指令），然后利用这个地址到指令存储器的相应地址去读指令，R 型指令最多涉及三个寄存器号，分别为 rs、rt、rd，前两个是读寄存器，rd 是写寄存器，都是 5 位，然后将读到的两个数据从寄存器堆输出，然后需要进行相应的运算，所以连接到 ALU，然后将结果写回到寄存器堆，这里写回的地址也是上文中给出的 rd 寄存器号，图如下：

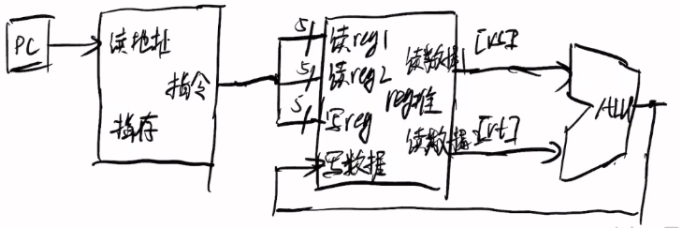

图 2: R 型指令局部数据通路图

- 3）答：表格如下

	IF取指令	ID译码与读reg	EX运算	MEM访存分支	WB写回
R型	根据PC 从存储器中 取出指令	取源操作数 rs和rt	执行运算	-	将ALU运算结果 写回rd
lw		取基址寄存器rs 偏移量符号扩展	计算内存地址	读数据存储器 取得数据	将访存数据 写回rt
sw				写数据存储器	-
beq		取rs和rt 字地址符号扩展 并左移两位	比较rs和rt 算分支目标地址	地址写回PC	-

图 3: 指令周期总结表

单周期控制单元

- 1）答：首先从 PC 取到要执行指令地址（add 指令），然后利用这个地址到指令存储器的相应地址去读指令，然后分别从寄存器堆读出寄存器 $t3 和 $t6 的数据，同时指令的 [31-26] 位输入到主控制单元，进行信号控制，首先会控制到 2 位的 ALUOp 信号（add 指令 ALUOp 信号是 10），然后再结合 [5-0] 位的功能码（add 指令是 100000），输出 ALU 控制信号（add 指令是 0010），这个信号会控制 ALU 执行加法运算，将刚才读出的数据进行相加运算，然后将 ALU 运算的结果写回到寄存器堆，这里的寄存器会根据 [15-11] 位来进行选择目标寄存器（$t2），这是 RegDst 信号和 RegWrite 信号有效，RegDst 信号有效的含义是写寄存器目标寄存器号来自 [15-11] 位，RegWrite 信号有效的含义是寄存器堆写使能信号有效。综上完成了一个 add 指令的执行过程。

流水线数据通路与控制、流水线冒险

- 1）
  - 问题 1: 答：所有阶段延迟求和：100+150+200+250+300=1000ps
  - 问题 2: 答：延迟最长的一个阶段：300ps
  - 问题 3: 答：lw 指令包括五个阶段，非流水线处理器的延迟时间：1000ps；流水线处理器的延迟时间：300×5=1500ps。
  - 问题 4: 答：挑选 ID 和 EX 阶段进行划分，第一步划分会挑选 ID 阶段（延迟时间最长），划分后第二步继续划分会同理挑选 EX 阶段，再划分后的处理器时钟周期是 200psWB 阶段，因为它是经过两次划分之后延迟时间最长的。
  - 问题 5: 答:涉及数据存储器的指令是 lw 和 sw,所以数据存储器的利用率是 10+25=35%；用到数据存储器写端口的只有 sw 指令，所以数据存储器写端口的利用率是 25%；用到寄存器堆写端口的指令是 ALU 和 lw，所以寄存器写端口的利用率是 30+10=40%。
- 2）
  - 问题 1: 答：这两条指令有数据相关，add 指令需要寄存器 $s2 中的数据，而 sub 指令需要完成第四级完成之后才行，对于 add 指令来说太迟了因为 add 第三级输入就需要，这里的解决方法就是使用旁路方法，连接从 sub 指令的 EX 操作到 add 指令的 EX 操作作为输入的旁路路径，替换掉 add 第二步从寄存器 $s2 读取的数据。

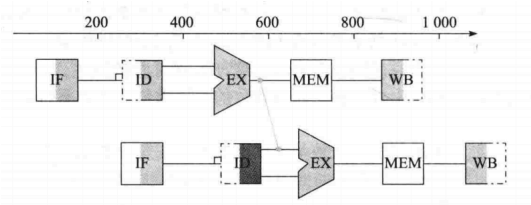

图 4: 旁路

– 问题 2: 答：这两条数据会遇到取数-使用型数据冒险，可以发现即使在添加旁路之后也无法正确执行，这时候我们需要流水线阻塞机制，在两条指令之间插入一行阻塞（即空指令 nop），从而再进行旁路连接指令得以正常执行。

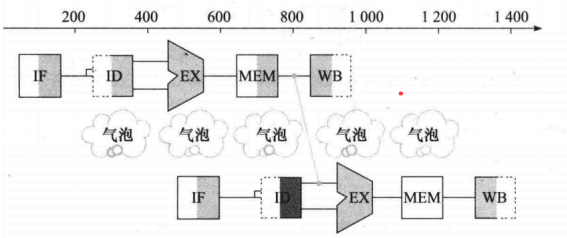

图 5: 阻塞

– 问题 3: 答：因缺乏硬件支持而导致指令不能在预定的时钟周期内执行的情况。

– 问题 4: 答：第一种方法是阻塞：是取分支指令后立即阻塞流水线，直到流水线确定分支指令的结果并知道下一条真正要执行的指令在哪为止，但是这种方法的效率很低。第二种方法是预测：一种简单的预测方法是总预测分支不发生，但是这种方法没那么有效；还有一种方法是预测一些分支发生而预测另一些分支不发生。另外还有一种技术是动态分支预测，通过查找指令的地址观察上一次执行该指令的时候分支是否发生，如果上次执行时候分支发生，就从上次分支发生的地方开始取新的指令。

– 问题 5: 答：在第三个时钟周期 ID 级确定分支发生，这个时候地址 40 被选中下一个 PC 地址，第四个时钟周期地址为 40 的指令被取回，然后后果是产生了一个阻塞（nop）。

- 3）答：正确
- 4）答：正确

异常、ARM 与 x86 处理器、指令级并行矩阵乘法

- 1）答：异常是源于指令本身，如算术溢出、请求系统调用、使用未定义的指令；中断是源于指令外部，如 I/O 请求、硬件故障（部分硬件故障也属于异常）
- 2）答：ARM Cortex-A8 采用三阶段、14 级的流水线结构；Intel Core i7 920 采用更复杂的流水线，通过将 x86 指令翻译成微操作来简化流水线实现。
- 3）答：推测是一种为了使依赖于被推测指令的其他指令可以执行，而允许编译器或处理器”猜测”指令结果的方法。技术包括静态多发射和动态多发射，静态多发射和动态多发射都是用于提高指令级并行性的技术，但它们在发射指令的方式和时机上有所不同。静态多发

射在取指令阶段就决定同时发射多条指令，而动态多发射在运行时根据实际情况动态地选择是否发射多条指令。这两种技术都可以提高处理器的并行执行能力，但需要考虑指令依赖关系、资源利用和处理器设计等因素来确定最佳的指令发射策略。

2. 第五章

存储器技术概要

- 1) 答: 在存储器层次结构中, 从顶端到底端: 快贵小到慢廉大。下面从顶层开始介绍: L1-L3: 这部分是高速缓存 cache，通常集成在 CPU 中，采用 SRAM 集成电路，由双稳态触发器制造;
  L4：内存，采用 DRAM 集成电路，硬件规模远远小于 SRAM。
  SRAM 和 DRAM 都是易失性存储器。
  L5：二级存储器或辅存，这部分以前通常用磁盘，现在更多的使用闪存。其中闪存是一种集成电路制造的电可擦除可编程只读存储器。磁盘和闪存都是非易失性存储器。

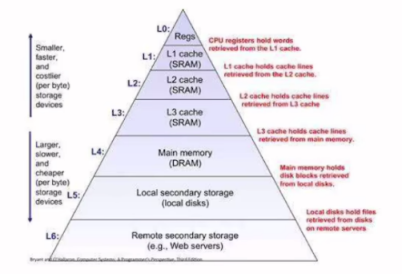

图 6: 存储器层次结构图

- 2）答：局部性原理分为时间局部性和空间局部性：时间局部性是指如果程序正在执行的语句位于循环体中，那么这条指令很可能不久之后再被访问；空间局部性是指如果在循环体或顺序指令流中，那么将要执行的指令往往地址相近。

高速缓存 cache

- 1)答:命中率是 12/20=60%;缺失时间是 101-1=100T;访存阻塞周期数是 20×0.4×100=800T。
- 2）答：数据一共有 2^{15} 字节，一个块有 8 个字是 32 字节，所以一共有 2^{10} 个块，每个块有 256 位的数据。另一方面一个块内有 32 字节，所以块内字节偏移是 5 位；一共有 2^{10} 个块，所以索引位是 10 位，所以标记位是 32-5-10=17 位，还有一位的有效位，综上总 cache 的位需要
  $2^{10} \times (256 + 17 + 1) = 2^{10} \times 274 = 274\text{Kib}$，说明 cache 一共需要 280576 位。
- 3）答：三种方法分别为：写直达、写缓冲、写回。
  写直达：将要写的数据同时写入主存和 cache 中。
  写缓冲：当一个数据在等待被写入主存的时候，先将它写入缓冲中。当把数据写入 cache 和写缓冲后，程序继续执行。当写主存操作完成后，写缓冲里的数据项也得以释放。
  写回：在发生写操作的时候，新值仅仅被写入 cache 块中。只有当修改过的块被替换时才需要写到较低层存储结构中。
- 4）答：AMAT= 命中时间 + 缺失率 × 缺失代价 =1+0.05×100=6T，即 12ns。

- 5）答:
  对于全相联，只有一个 1024 块的组，组数是 1，一个块有 8 个字就是 32 字节，所以块内字节偏移有 5 位，没有索引位，所以标记位是 32-5=27 位。
  对于四路相联，一共有 256 组，所以索引位是 8 位，块内字节偏移还是 5 位，所以标记位是 32-5-8=19 位。

- 6）答：缺失代价的时钟周期是 100ns/(1/5GHz)=500T，只有一级的 cache 的 CPI 是 1+4%×500=21，然后同理可以计算出二级 cache 的缺失代价是 50T，此时的有效 CPI 是 1+4%×50+1%×500=8，所以有二级 cache 的处理器性能是没有二级 cache 处理器性能的 21/8=2.625 倍。

虚拟存储器

- 1）答：虚拟地址 32 位，物理地址 30 位，页是 4KB，需要 12 位来表示页内偏移，所以虚拟页号是 32-12=20 位，物理页号是 30-12=18 位。虚拟页号通过页表映射到物理页号，物理地址由虚拟页号映射的物理页号与页内偏移拼接而成。

- 2）答：脏位是用来判断是否写入过，从而判断是否需要写回操作；引用位是用来近似实现 LRU 替换算法的位，操作系统回定期将页表中的引用位清零，访问过的页引用位再设置为 1，随后从引用位为 0 的页中选择一页进行替换。

- 3）答：TLB 有三个标志位：有效位、重写位、引用位，一个标记位（虚拟页号），对应的物理页地址。TLB 是一个全相联的 cache。
  下面根据下图进行完成流程的说明：一个虚拟地址首先访问 TLB 进行匹配，如果有效位为 1 而且找到了对应的虚拟页号，则说明对应的物理页地址是在内存中，直接访问即可。如果找不到则继续访问页表，如果与一个有效位为 0 的行记录匹配上，说明这是存在磁盘中的数据。

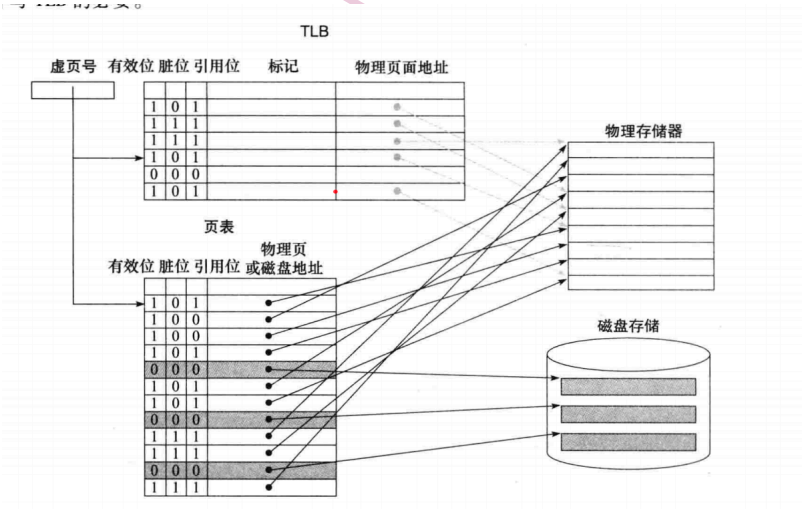

图 7: TLB 流程图

-

可靠性与校验码

- 1）答：20w 小时大约是 23 年，可靠性一般水平。AFR=365×24/200000=0.04272。可用性=MTTF/MTBF=MTTF/MTTF+MTTR=200000/(200000+200)=0.999001，可用性还是不错的。

- 2）答：首先将校验位空出来：___1__001__0010.
  位置 1 检查 1、3、5、7、9、11 位，偶校验，该校验位是 1
  位置 2 检查 2、3、6、7、10、11 位，偶校验，该校验位是 1
  位置 3 检查 4、5、6、7、12 位，偶校验，该校验位是 1
  位置 4 检查 8、9、10、11、12 位，偶校验，该校验位是 1
  所以汉明码是 111100110010。把第 10 位取反得到 111100110110。重新计算 4 位校验位，同理上文分别是 1010，然后将两次的校验位从右向左读取，分别是 1111 和 0101，进行异或运算，是 1010，代表 10，说明是第 10 位发生了错误，改正就可。

虚拟机、cache 控制器、分块加速、ARM 与 x86 存储器层次结构

- 1）答：虚拟机能够实现多个客户端在一个硬件主机上运行。由虚拟机监视器（VMM）实现不同操作系统间的资源隔离与控制。

- 2）答：通过使用软件方法对大型矩阵进行分块。让处理器每次运算一个小矩阵块，而不是二位数组中的一行或一列。可以显著降低 cache 缺失率，利用存储器层次结构，将矩阵乘法性能再次翻倍。

- 3）答：ARM Cortex-A8 采用两级 cache，Intel Core i7 Nehalem 采用三级 cache。其中 L1 cache 均为指令与数据分离的 cache，A8 的 L2 cache 使用统一 cache。Core i7 中的 L2 cache 每个核心统一，L3 cache 由多核共享。

二、 参考文献

1. 计算机组成与设计：硬件软件接口，David A.Patterson、John L.Hennessy，2015