

Homework Assignment 4 of Computer Architecture

College of Computer Science, Zhejiang University

Submission Deadline: Jan 10, 2020

(100 points)

1. (18 points) Consider the following description of a memory hierarchy.

Virtual address wide = 45 bits, Memory physical address wide = 38 bits, Page size = 4KB.

Cache capacity = 8KB. Block size = 32Byte. It is a write-back 2-way associative cache.

- How many bits are there in the fields of tag, index and block offset of the physical memory address.
- Draw a graph to show a cache line (including tag, data, and some other control bits) in the cache.
- Draw a graph to show if it is implemented in the way of virtually indexed and physically tagged cache.
- Please describe the access procedure to the memory hierarchy in (c) that when a CPU address (virtual address) is given to access the cache.

基本信息:

虚拟地址 45bits、物理地址 38bits、page size 4KB

cache 大小 8KB、cache block size 32Byte、write back 策略、2 路组相连

Solution:

a 对于物理地址-给出 tag、index、block offset 字段

由于 block-size=32Byte, 因此 **block offset = 5bit**

cache 能装的 block 数 = $8KB / 32B = 256$ 个

两路组相连, 因此 set 个数为 128 个, 于是 **index = 7**

所以 **tag = 38 - 5 - 7 = 26**

最终结果: tag: 26bit, index: 7bit; block offset: 5bit

b 画出 cache line 的格式

由于这里采用 write back 策略, 因此需要 dirty bit 来标记与内存的差别。因此这里的 control bit 为 valid bit 和 dirty bit。

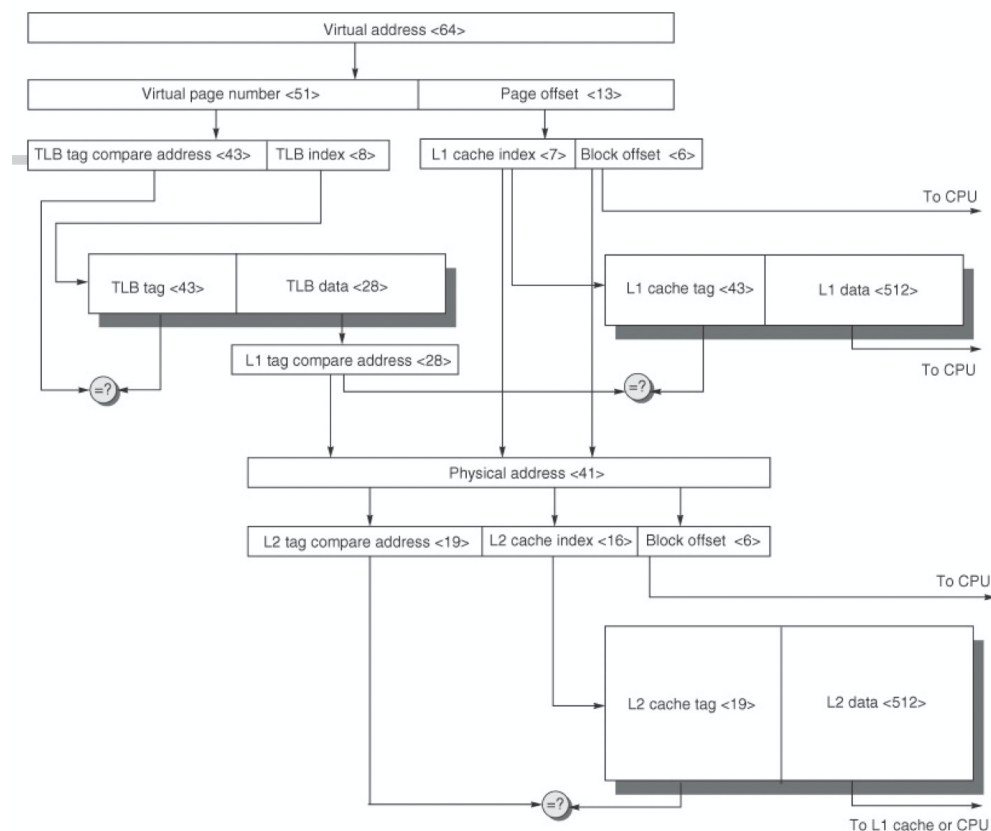
下图假设为一个 word 为 8 个 byte, 于是 data block 中有 4 个 word

Dirty	Valid	Tag(26bit)	Data block(256bit)			
			Word1 (64bit)	Word2	Word3	Word4

c 画出 virtually indexed and physically tagged cache 的图

事实上没有区别，因为 **cache 仍然是 physically tagged**，和上图的结构是一样的。区别在于访问方式不同，也就是 **index** 的时候，这个 **index** 的值是 **virtual address** 的，但对应的 **tag** 是 **physical** 的。

下面给出具体的访问示意图(类似课程资料中的)



d 描述 virtually indexed and physically tagged cache 的内存访问顺序

如上图所示。首先得到 **virtual address**，并行的访问 **TLB** 与 **L1 cache**。通过 **TLB** 转换可以得到物理的 **Page number**，在合理设计的情况下，物理的 **Page number** 即等于物理 **tag**，将其与 **L1** 上的物理 **tag** 进行比较，如果相同，说明 **hit**，将数据传送到 **CPU** 即可。如果不相同，说明 **miss**，但此时也已经得到了物理地址，因此可以继续访问下一级存储(一般情况下为 **L2 cache**-这要看具体设计)。

上面描述中没有考虑到 **TLB miss** 的情况。在这种情况下，只能产生 **stall**，从内存中访问 **page table**，而后进行地址转换，进一步重复上述过程。

2. (22 points) Assume that we have two machines A and B. The only difference between A and B lies in their cache hierarchies:

Machine A: 64 KB level-one data cache with a 8 ns access time and a miss rate of 8%

Machine B: 8 KB level-one data cache with a 2 ns access time and a miss rate of 15%, and a 1 MB level-two cache with a 20 ns access time and a miss rate of 10%.

Assume that both machines have an I-cache miss rate of 0%, a main memory access time of 50 ns , and all the bus transfer time could be ignored. Which machine will have a better performance in memory access (AMAT)? Why?

题目描述:

机器 A 和 B 的唯一区别是他们的 **cache** 层级不同。

机器 A: 64KB 一级 data cache, 8ns 的访问时间以及 8% 的 miss rate。

机器 B: 8KB 一级 data cache, 2ns 的访问时间以及 15% 的 miss rate。同时有 1MB 的二级 cache, 20ns 的访问时间以及 10% 的 miss rate。

问题: 假设指令 **miss** 率为 0, 主存访问时间为 50ns, 所有的总线传输时间可以忽略。计算 **AMAT**, 分析并比较

其性能。

Solution:

公式 $AMAT = hit\ time + miss\ rate * miss\ penalty$

对于二级 cache, $AMAT = hit\ time_{L1} + miss\ rate_{L1} * (hit\ time_{L2} + miss\ rate_{L2} * miss\ penalty_{L2})$

根据以上公式计算得

$AMAT_A = 8ns + 8\% * 50ns = 12ns$

$AMAT_B = 2ns + 15\% * (20ns + 10\% * 50ns) = 5.75ns$

根据 AMAT 指标, 有二级 cache 的机器 B 有更好的性能, 因为他的平均内存访问时更少。

3. (20 points) Suppose you own a computer which has the following properties:

- the pipeline can accept a new instruction every cycle
- the cache can provide data every cycle (i.e. no penalty for cache hits)
- the instruction cache miss rate is 2.5%
- the data cache miss rate is 3.5%
- 30% of instructions are memory instructions
- the cache miss penalty is 80 cycles.

Now you want to purchase a new computer. you can either

- purchase a machine with a processor and cache that is twice as fast as your current one (memory speed is the same as the old machine, though), or
- purchase a machine with a processor and cache that is the same speed as your old machine but in which the cache is twice as large.

Assume the cache miss rate will drop by 40% with this larger cache (although this is generally not true in the real world).

Which computer are you best off purchasing? Explain in detail, showing the relative performance of each choice.

题目描述: 假设计算机有如下属性

- 流水线每个周期可以接收一条指令 (ideal CPI = 1)
- cache hit 没有额外时间, 或者说在一个周期内完成
- 指令 cache miss 率是 2.5%
- 数据 cache miss 率是 3.5%
- 30% 的指令是内存访问指令
- miss penalty 是 80 个周期

问题:

你希望购买一台新电脑, 你的选择有, 购买的机器的处理器和 cache 的速度是当前的两倍, 但内存访问速度不变。或者购买的机器的处理器和 cache 速度和原来相同, 但 cache 大小是原来的两倍。对于第二种, 假设 cache miss 率会下降 40%。计算比较相关性能, 详细分析你的选择。

Solution:

假设原来的时钟周期为 CCT

Metrics1: CPU time

对于第一种

CPU 时间 = $1 * 0.5CCT$ (ideal 时间) + $0.3 * 3.5\% * 80CCT$ (data miss penalty-内存访问时间) + $1 * 2.5\% * 80CCT$ (instruction miss penalty-内存访问时间) = $0.5 + 0.84 + 2 = 3.34CCT$

cache 的访问命中就已经报过在 ideal 时间中, 这是因为 cache 命中的时候就占一个流水线周期, 仍然保持流水线执行, 没有 stall。

注: 这里采用直接根据时钟周期时间计算。另一种方式是通过 CPI, 而后计算时钟周期。由于时钟频率 double 了, 但是内存访问的时间不改变, 因此内存产生的 stall 个数为原来的两倍, 即 $80 * 2 = 160$ 个 cycles。类似的方法计算得到 CPI, 而后计算周期时间, 结果与上面是一致的。

对于第二种

$$\text{CPU 时间} = 1\text{CCT}(\text{ideal 时间}) + 0.3 * 3.5\% * 0.6 * 80\text{CCT} + 1 * 2.5\% * 0.6 * 80\text{CCT} = 0.3 + 0.504 + 1.2 = 2.004\text{CCT}$$

根据 CPU 时间，显然选择第二种，他的 CPU 时间更短。

Metrics2: AMAT

采用课程例题解法中 AMAT 的计算方式，仍然是直接计算时间-time

- Suppose a processor executes at
 - Clock Rate = 200 MHz (5 ns per cycle), Ideal (no misses) CPI = 1.1
 - 50% arith/logic, 30% ld/st, 20% control
- Suppose that 10% of memory operations get 50 cycle miss penalty
- Suppose that 1% of instructions get same miss penalty
- Calculate the AMAT and real CPI.

• Answer: CPI = ideal CPI + average stalls per instruction
= $1.1(\text{cycles/ins}) + [0.30(\text{DataMops/ins}) \times 0.10(\text{miss/DataMop}) \times 50(\text{cycle/miss}) + 1(\text{InstMop/ins}) \times 0.01(\text{miss/InstMop}) \times 50(\text{cycle/miss})]$
= $(1.1 + 1.5 + .5) \text{ cycle/ins} = 3.1$
• 58% of the time the proc is stalled waiting for memory!
• $\text{AMAT} = (1/1.3) \times [1.1 + 0.01 \times 50] + (0.3/1.3) \times [1.1 + 0.1 \times 50]$
• = 2.54

对于第一种

$$\begin{aligned} & 1/1.3 * (1 * 0.5\text{CCT} + 2.5\% * 80\text{CCT}) + 0.3/1.3 * (1 * 0.5\text{CCT} + 3.5\% * 80\text{CCT}) \\ &= 1/1.3 * 2.5 \text{ CCT} + 0.3/1.3 * 3.3 \text{ CCT} \\ &= 1.9231 + 0.7615 \text{ CCT} = 2.685 \text{ CCT} \end{aligned}$$

对于第二种

$$\begin{aligned} & 1/1.3 * (1\text{CCT} + 2.5\% * 0.6 * 80\text{CCT}) + 0.3/1.3 * (1\text{CCT} + 3.5\% * 0.6 * 80\text{CCT}) \\ &= 1/1.3 * 2.2 \text{ CCT} + 0.3/1.3 * 2.68 \text{ CCT} \\ &= 1.6923 \text{ CCT} + 0.6185 \text{ CCT} = 2.311\text{CCT} \end{aligned}$$

对于 AMAT，仍然是第二个更优秀。

综上所述，对于性能评价的两个指标 CPU time 和 AMAT，均为第二种 CPU 更优秀，主要原因在于内存访问时间占比大，因此减少 miss rate 能较好的提高 CPU 性能。

4. (40points) You are building a system around a processor with in-order execution that runs at 1.1 GHz and has a CPI of 0.7 excluding memory accesses. The only instructions that read or write data from memory are loads (20% of all instructions) and stores (5% of all instructions).

The memory system for this computer is composed of a split L1 cache that imposes no penalty on hits. Both the I-cache and D-cache are direct mapped and hold 32KB each. The I-cache has a 2% miss rate and 32-byte blocks, and the D-cache is write through with a 5% miss rate and 16-byte blocks. There is a write buffer on the D-cache that eliminates stalls for 95% of all writes.

The 512KB write-back, unified L2 cache has 64-byte blocks and an access time of 15ns. It is connected to the L1 cache by a 128-bit data bus that runs at 266MHz and can transfer one 128-bit word per bus cycle. Of all memory references sent to the L2 cache in this system, 80% are satisfied without going to main memory. Also 50% of all blocks replaced are dirty.

The 128-bit-wide main memory has an access latency of 60ns, after which any number of bus words may be transferred at the rate of one per cycle on the 128-bit-wide 133 MHz main memory bus.

- (8points) What is the average memory access time for instruction accesses ?
- (8points) What is the average memory access time for data reads ?
- (8points) What is the average memory access time for data writes ?
- (8points) What is the overall CPI, including memory accesses ?
- (8points) You are considering replacing the 1.1GHz CPU with one that runs at 2.1GHz, but is otherwise identical. How much faster does the system run with a faster processor ? Assume the L1 cache still has no hit penalty, and that the speed of the L2 cache, main memory, and buses remains the same in absolute terms (e.g. the L2 cache has a 15ns access time and a 266MHz bus connecting it to the CPU and L1 cache).

你要构造一个顺序执行的 CPU 系统，主频为 1.1GHz，CPI 为 0.7，不包括内存访问(即 ideal CPI)。唯一的访问内存的指令为 load(占比 20%)和 store(占比 5%)

内存系统由 split L1 cache，对于 hit 情况没有额外的时间(即不产生 stall)。Instruction cache 和 data cache 是 direct mapped，大小都为 32KB。Instruction cache 的 miss rate 为 2%，block-size 为 32 byte，data cache 采用 write through 策略，miss rate 为 5%，block-size 为 16byte。data cache 有 write buffer，从而能够减小 95%的 write stall。

512KB 大小的，采用 write-back 策略的，unified L2 cache，block-size 为 64 byte，访问时间为 15ns。他和 L1 cache 通过 128-bit 宽的总线连接，频率为 266MHz，并且每个 bus cycle 可以传输 128-bit word(这里 word 大小为 128bit-16 字节)。对于所有的 L2 cache 内存访问，80%能够满足而不需要到内存(也就是 miss rate 为 20%)。同时替换的 block 中有 50%是 dirty 的。

128-bit 宽的内存访问延迟为 60ns。内存总线的频率为 133MHz。后面有说：**after which any number of bus words may be transferred at the rate of one per cycle on the 128-bit-wide 133MHz main memory bus.**意思似乎是，第一次访问的时候需要花费 60ns，而后连续的内存 word 访问，只需要花费总线传输时间即可。

Solution:

此题信息量庞大，阅读和理解难度较大。下面我重新分析一遍数据，再进行具体计算。

CPU: 时钟频率 1.1GHz，对应时钟周期时间 $1/1.1\text{GHz} = 0.909\text{ns}$ 。ideal CPI = 0.7

程序指令类型组成：75%的非内存访问，20%的 load for read，5%的 store for write

L1 cache: split, hit time=0, direct mapped。

L1, I-cache, miss rate = 2%, block-size = 32 Bytes (2 个 128bit-word, 2 个 128-bit bus cycle 填充)。L1-L2 总线频率=266MHz，对应的时钟周期为 $1/266\text{MHz} = 3.759\text{ns}$ 。

L1, D-cache, miss rate = 5%, block-size = 16 Bytes (1 个 128bit-word, 1 个 128-bit bus cycle 填充)，write-through + write buffer 从而 5%的 write miss，但是 write miss 的时候不分配，直接写内存。

L2 cache: unified, write-back，与之配合的是 write-allocate，即 write miss 的时候先从内存填充到 L2 cache, miss rate = 20%，替换的时候 50%的概率是 dirty-需要写回内存，另外的不需要写回内存，直接从内存读取替换。block-size = 64 Bytes (4 个 128bit-word, 4 个 128-bit bus cycle)。L2-Memory 总线频率为 133MHz，对应时间为 $1/133\text{MHz} = 7.519\text{ns}$ 。access time = 15ns

内存，访问时间为 60ns。

采用 bottom-up 的方式计算可以直接由题目得到的一些时间参数

L2-miss penalty(无 write-back 的情况下) = 内存访问延迟+传输时间 = $60\text{ns} + 4 * 7.519\text{ns} \approx 90\text{ns}$

由于 L2 采用 write back 策略, 因此内存访问的时候会发生替换, 题目中给出 dirty 的概率为 50%, 因此替换发生写回的概率为 50%, 写回时间也会 90ns。

从而实际上的 L2-miss penalty = $1.5 * 90\text{ns} = 135\text{ns}$ 。

注: 这里的计算方式, 内存延迟只算了一次, 这是因为考虑到题目里面说的那句话, 可能他是采用“Interleaved bank memory 的方式——类似于流水线”, 具体的我也不清楚。

///L2-miss penalty for read 也是 135ns

///L2-miss penalty for write 也是 135ns, 因为 read miss 的情况必然会从内存读取到 L2-cache。而 write miss 由于 write-back 配合的是 write-allocate, 因此也会从内存读取到 L2-cache。当然也都会发生 dirty block 的替换。

L1-Instruction-miss penalty = L2 的 AMAT = L2 hit time + L2 miss rate * L2-miss penalty

= L2 access time + 传输时间 + L2 miss rate * L2-miss penalty

= $15\text{ns} + 3.759\text{ns} * 2 + 20\% * 135\text{ns} = 15\text{ns} + 7.518\text{ns} + 27\text{ns} = 49.52\text{ns}$

L1-Data-read-miss-penalty = L2 的 AMAT = L2 hit time + L2 miss rate * L2-miss penalty

= L2 access time + 传输时间 + L2 miss rate * L2-miss penalty

= $15\text{ns} + 3.759 * 1 + 20\% * 135\text{ns} = 45.759\text{ns}$

a 计算指令访问的 AMAT

指令访问 AMAT = L1 hit time(=0) + L1 指令 miss rate * L1-instruction-miss penalty = $2\% * 49.52\text{ns} = 0.9904\text{ns}$

b 计算数据读取的 AMAT

数据读取 AMAT = L1 hit time(=0) + L1 数据 read miss rate * L1-data-read-miss penalty = $5\% * 45.759\text{ns} = 2.288\text{ns}$

注意: 数据 miss rate 为 5% 而不是 2%

c 计算数据写入的 AMAT

由于 L1 D-cache 采用了 write-through 策略, 因此 write 会进行写下一级存储操作(应该是下一级-这里是 L2 cache, 不过也可以写内存-主要是看设计模式-总线怎么接的), 采用了 write buffer 的情况, 减少了 95% 的 write stalls。于是这里产生的 stall 时间应该为 L1-L2 总线传输时间 = $3.759\text{ns} * 1 \text{ L1-block} * 5\% =$

$3.759\text{ns} * 0.05 = 0.188\text{ns}$ 。从而在 L1 D-cache write hit 的情况下, L1 D-cache 的 write 时间即为 0.188ns(L1 cache 此题本身没有 hit time)。而在 L1 D-cache write miss 的情况下, 由于通常 write through 配合的是 write no-allocate, 结果则是直接写 L2 的数据, 这时考虑的就是 L2 的访问时间了。

综上所述: 数据写入的 AMAT = $0.188\text{ns} + 5\% * 45.759\text{ns} = 0.188\text{ns} + 2.288\text{ns} = 2.476\text{ns}$

///有的题目表述说对于 95% 的 write stall 被 eliminated 了, 因此 writes stall 相当于是 0, 这里我仍然考虑进去了。

d 计算总体的 CPI-包含内存访问

对于 CPI 而言

$$\begin{aligned}\text{overall CPI} &= \text{ideal CPI} + \text{instruction stall} + \text{data read stall} + \text{data write stall} \\ &= 0.7 + 0.9904/0.909(\text{instruction stall}) + 0.2 * 2.288/0.909(\text{data read stall}) + 0.05 * \\ &\quad 2.476/0.909(\text{data write stall}) = 0.7 + 1.089 + 0.503 + 0.136 = 2.428 \text{ CPI}\end{aligned}$$

这里采用了 a、b、c 三题的计算结果。

e 使用 2.1GHz 的 CPU 替换 1.1GHz 的 CPU，其他方面相同。对于这个更快的 CPU，计算整个系统的加速比。假设 L1 cache 仍然没有 hit penalty，并且 L2cache、内存、总线的速度的绝对值保持不变。

此题与作业第三题相同。仅仅更换 CPU，使用更高频率的 CPU，这也就导致了更低的时钟周期。

此题考虑 CPU 时间更加合理(大多数时候也都是如此)。

内存访问时间等都不改变。采用第三题做法

新的 CPU time = idea CPI * 新的时钟周期 + 原来的 stall cycle * 老的时钟周期

$$= 0.7 * 0.476 + 1.728 * 0.909\text{ns} = 0.333 + 1.571 = 2.0\text{ns}$$

$$\text{因此 speedup} = (2.428 * 0.909\text{ns}) / 2\text{ns} = 1.1$$

总结：memory 相关题目概念容易理解，但又容易混淆。题目有时也会表述不清，或者难以理解，

不过更多的还是因为自己一知半解，在透彻理解的情况下解题也不会很困难。更加深入的理解需

要有硬件实际操作的把握，自己还需多多锻炼。