

# 存储

---

## 存储层级

---

### 定义

1. 判断：内存结构中耗时最多的是最上层（L1-cache）

### CPI 计算

1. Instruction cache miss rate is 2%; Data cache miss rate is 4%; CPI without any memory stalls is 2; Miss penalty is 100 cycles; The frequency of all loads and stores in gcc is 36%  
Q: How faster a processor would run with a perfect cache?
2. 200 MHz. 采用 mixed cache. ALU/Logic 35% Load 30% Store 15% Branch 20% cache hit rate 98%, 内存读写 100ns. branch 采用 predict-not-taken. 假设跳转在 3rd 阶段决定。branch not taken 的比例为 40%, load 后有 50% 的比例会有 load-use hazard. 求 CPI.
3. Assuming a base CPI of 1.0 without any memory stalls, what is the total CPI for P1 with the addition of an L2 cache? Assume that P1 clock cycle is 0.66ns, L1 miss rate is 8%, L1 hit time is 0.66ns, L2 miss rate is 95%, L2 hit time is 5.94ns, main memory need 70.62ns, 36% instructions need memory access.

# 优化

---

## 位数计算

1. By convention, a cache is named according to the amount of data it contains (i.e., a 4 KB cache can hold 4 KB of data); however, caches also require SRAM to store metadata such as tags, valid bits, dirty bits. Assume that the caches are byte addressable, and that addresses and words are 32 bits. Calculate the total number of bits required to implement a 16 KB direct-mapped cache with 4-word blocks.
2. A memory and cache subsystem uses byte-addressing, 32-bit address. Each row of the table below indicates a kind of cache, the "Cache feature" column of table describes the total size of cache data(not containing tag and valid bit), cache kind, cache block size, A 32-bit memory address 0x22339AB contains 3 fields: tag, index, byte offset, some of these field's size(bit number of this field) or value should be filled into table blank below, You should use Hex-decimal number to fill into column "Index value (Hex)", Total cache size containing data block, tag and valid bit should also be filled into column "Total Size,kB"(kB means unit is KB-1024 byte, not byte), only number(not expression) is allowed to be filled into this column.

cache feature	index value	index size,bits	tag size,bits	total size ,KB
64KB data,direct-mapped,8 bytes block	0x0735	13	16	81
512KB data,direct-mapped,64 bytes block				

<b>cache feature</b>	<b>index value</b>	<b>index size,bits</b>	<b>tag size,bits</b>	<b>total size ,KB</b>
512KB data,2-Way set associative,64 bytes block				
1024KB data,8-Way set associative,32 bytes block				

2. By convention, a cache is named according to the amount of data it contains (i.e., a 4 KB cache can hold 4 KB of data); however, caches also require SRAM to store metadata such as tags, valid bits, dirty bits. Assume that the caches are byte addressable, and that addresses and words are 32 bits. Calculate the total number of bits required to implement a 16 KB direct-mapped cache with 4-word blocks.

## 优化效果

1. 提高block size可以显著增加
  - A. cache miss B. Compulsory miss rate C. hit time D. miss penalty
2. 提高 cache 的组相联度可以提高(improve)
  - A. capacity miss
  - B. hit time
  - C. conflict miss
  - D. compulsory miss

## cache 访问

### 过程模拟

1. Assume we have a direct-mapped byte-addressed cache with capacity 32B and block size of 8B.
  1. Of the 32 bits in each address, which bits do we use to find the tag, index, and offset of the cache?
  2. Classify each of the following byte memory accesses as a cache hit, cache miss, or cache miss with replacement

<b>address</b>	<b>tag</b>	<b>index</b>	<b>offset</b>	<b>Hit/Miss/Replace</b>
0x00000004				
0x00000005				
0x00000068				
0x000000c8				

address	tag	index	offset	Hit/Miss/Replace	
0x00000068					
0x000000dd					
0x00000045					
0x000000cf					
0x000000f3					
0x000000db					
0x0000009c					
0x00000157					
0x00000fe9					

## 虚拟地址

---

### 页表大小计算

1. Virtual address 48 bit, DRAM 512GiB, SRAM 256GiB, PageSize 4 KiB, PageEntry 4 bytes 求  
5并行进程需要多大的PageTable

### 过程模拟

1. Virtual memory uses a page table to track the mapping of virtual addresses to physical addresses. This exercise shows how this table must be updated as addresses are accessed. The following data constitute a stream of virtual byte addresses as seen on a system. Assume 4 KiB pages, a four entry fully associative TLB, and true LRU replacement. If pages must be brought in from disk, increment the next largest page number.

Decimal	4669	2227	13916	34587	48870	12608	49225
hex	0x123d	0x08b3	0x365c	0x871b	0xbbee6	0x3140	0xc049

## TLB

Valid	Tag	Physical Page Number	Time Since Last Access
1	0xb	12	4
1	0x7	4	1
1	0x3	6	3
0	0x4	9	7

## Page table

Index	Valid	Physical Page or in Disk
0	1	5
1	0	Disk
2	0	Disk
3	1	6
4	1	9
5	1	11
6	0	Disk
7	1	4
8	0	Disk
9	0	Disk
a	1	3
b	1	12

- For each access shown above, list whether the access is a hit or miss in the TLB, whether the access is a hit or miss in the page table, whether the access is a page fault, the updated state of the TLB.
- Discuss why a CPU must have a TLB for high performance. How would virtual memory accesses be handled if there were no TLB?

## 综合应用

1. It assumes that the address of main memory is 32-bit and is addressed by byte. 8-way set-associative mapping is used between instruction cache and data cache and main memory, and write through strategy is used. The data capacity of the cache is 32KB and the block size is 64B

1. How many bits is the tag of each cache line? Does it contain dirty bits?

2. Here is a C code:

```
for (k = 0; k < 1024; k++)
    s[k] = 2 * s[k];
```

If both array s and variable k are int, int accounts for 4B, variable k is allocated in registers, and the start address of array s in main memory is 0x008000C0, what are the number of data cache misses accessing array s during the execution of this program segment?

3. If the first access operation of the CPU is to read the instruction in the main memory unit 0x0001003, briefly explain the process of accessing the instruction from the cache, including the cache miss handling process.