存储器、高速缓存、虚拟存储器作业

2022.5 .31

1. 存储器容量为 64MB，字长 64 位，体（bank）数 m = 8，分别用顺序方式和交叉方式进行组织。存储周期 T = 100ns，数据总线宽度为 64 位，总线周期 τ = 10ns。问顺序存储器和交叉存储器的带宽（单位：位/秒）各是多少？

解：信息总量：q=64 位×8=512 位

顺序存储器和交叉存储器读出 8 个字的时间分别是：

$t2 = m \cdot T = 8 \times 100ns = 8 \times 10^{-7}$ (s)

$t1 = T + (m-1)\tau = 100 + 7 \times 10 = 1.7 \times 10^{-7}$(s)

顺序存储器带宽是：

$W2 = q / t2 = 512 \div (8 \times 10^{-7}) = 64 \times 10^{7}$（位/S）

交叉存储器带宽是：

$W1 = q / t1 = 512 \div (1.7 \times 10^{-7}) = 301 \times 10^{7}$（位/S）

2. →用 64K×1 位的 DRAM 芯片构成 256K×8 位的存储器。要求：

（1）计算所需芯片数。

（2）若采用异步刷新方式，每单元刷新间隔不超过 2ms，则产生刷新信号的间隔是多少时间？

参考答案：

（1）256KB/64K×1 位=4×8=32 片。

（2）因为每个单元的刷新间隔为 2ms，所以，采用异步刷新时，在 2ms 内每行必须被刷新一次，且仅被刷新一次。因为 DRAM 芯片存储阵列为 64K=256×256，所以一共有 256 行。因此，存储器控制器必须每隔 2ms/256=7.8μs 产生一次刷新信号。

异步刷新、集中刷新、分散刷新

1.→直接映射（directed mapped）：

为了将内存数据块放置到高速缓存中，可以将内存地址分几部分看待：

Offset bits 是块内偏移，Index bits 用于标记该内存数据块在高速缓存中的组号（set number），剩下的 tag bits 是该数据块的标记。

如果高速缓存采用直接映射（direct mapped）采用写直达（write through）的更新策略，那么高速缓存中每一行包含的内容为：cache data block, tag, valid bit，不需要包含 dirty bit.

根据已知信息，请填写下表中的空格部分：

Address size (bits)	Cache size	Block size	Tag bits	Index bits	Offset bits	Bits per row
16	4KiB	4B	4	10	2	32+4+1
32	32KiB	16B	17	11	4	128+17+1
32	64KiB	16B	16	12	4	128+16+1
64	2048KiB	128B	43	14	7	1068

Way × Set× Block Size = Cache Size

2、组相联映射(set associative)：

假设某计算机的主存地址空间大小为 64MB，采用字节编址方式。其 cache 数据区容量为 4KB，采用 4 路组相联映射方式、LRU 替换和回写（write back）策略，块大小为 64B。请问：

（1）→主存地址字段如何划分？要求说明每个字段的含义、位数和在主存地址中的位置。

（2）该 cache 的总容量（不仅包括数据区容量）有多少位？

参考答案：

（1）cache 的划分为：$4KB=2^{12}B=2^4$ 组×2^2 行/组×2^6 字节/行，所以，cache 组号（组索引）占 4 位。主存地址划分为三个字段：高 16 位为标志字段（1 分）、中间 4 位为组号（1 分）、最低 6 位为块内地址

即主存空间划分为：$64MB=2^{26}B=2^{16}$ 组群×2^4 块/组群×2^6 字节/块

（2）cache 共有 64 行，每行中有 16 位标志、1 位有效位、1 位修改(dirty)位、2 位 LRU 位，以及数据 64B。故总容量为 64×(16+1+1+2+64×8)=34048 位。

Way × Set× Block Size = Cache Size

3、代码分析与高速缓存的性能：

一个二路组相联映射的高速缓存（2-way associative cache）容量为 128 bytes，每个高速缓存块大小为 32 字节（32 bytes per block）. long long 型数据的长度为 8 个字节（8 bytes）. 假定 table 数组的起始地址为 0x0. **以下代码的高速缓存失效率(miss rate)为多少？**

```
int i;
int j;
long long table[4][8];
for (j = 0; j < 8; j++)
    for (i = 0; i < 4; i++)
        {table[i][j] = i + j;}
```

tag 标记	组号	块内偏移
	1位	5位

Way

Set

Table[0][0-3] Table[0][0] Table[1][0]	Table[1][0-3] Table[2][0] Table[3][0]
Table[0][4-7]	Table[1][4-7]

假设高速缓存系统的属性如下，求 AMAT 是多少？

a) L1$ hits in 1 cycle (local miss rate 25%)
b) L2$ hits in 10 cycles (local miss rate 40%)
c) L3$ hits in 50 cycles (global miss rate 6%)
d) Main memory hits in 100 cycles (always hits)

Alternatively, we can calculate the global hit rates for each hierarchy:

- L1$: 0.75
- L2$: 0.25*0.6 = 0.15
- L3$: 0.94 – (0.75+0.15) = 0.04
- Main Memory: 1 – 0.75 – 0.15 – 0.04 = 0.06

And the following hit times:

- L1$: 1 cycle
- L2$: 1+10 = 11 cycles
- L3$: 1 + 10 + 50 = 61 cycles
- Main Memory: 1 + 10 + 50 + 100 = 161 cycles

Then, AMAT = 0.75*1 + 0.15*11 + 0.04*61 + 0.06 * 161 = 14.5 cycles.

1）如果页表地址寄存器中装入了新的值，TLB 会发生什么操作？

The valid bits of the TLB should all be set to 0. The page table entries in the TLB corresponded to the old page table, so none of them are valid once the page table address register points to a different page table.

TLB一般采用Write through策略，dirty位表示的是page被修改过，而不是TLB本身被修改过

Initial TLB

VPN	PPN	Valid	Dirty	LRU
0x01	0x11	1	1	0
0x00	0x00	0	0	7
0x10	0x13	1	1	1
0x20	0x12	1	0	5
0x00	0x00	0	0	7
0x11	0x14	1	0	4
0xac	0x15	1	1	2
0xff	0x16	1	0	3

设现在空闲的物理页是：0x17,0x18,0x19;

TLB
- 命中：当前行置0，比当前行计数器小的+1
- 未命中
  - 有空闲行：当前行置0，其余+1
  - 无空闲行：当前行置0，淘汰最大(7)，其余+1

Read 0x11f0: hit, LRUs: 1,7,2,5,7,0,3,4

Write 0x1301: miss, map VPN 0x13 to PPN 0x17, valid and dirty, LRUs: 2,0,3,6,7,1,4,5

Write 0x20ae: hit, dirty, LRUs: 3,1,4,0,7,2,5,6

Write 0x2332: miss, map VPN 0x23 to PPN 0x18, valid and dirty, LRUs: 4,2,5,1,0,3,6,7

Read 0x20ff: hit, LRUs: 4,2,5,0,1,3,6,7

Write 0x3415: miss and replace last entry, map VPN 0x34 to 0x19, dirty, LRUs, 5,3,6,1,2,4,7,0