

计算机组成原理

作业 3

李祥泽

2018011331

1

False.

引入虚拟存储是为了给所有程序提供相同的地址空间; 方便操作系统, 连接器, 装载器等的设计; 给程序员提供比实际主存大得多的内存空间.

2

False.

使用高速缓存是为了提供和 CPU 周期相当的快速内存读写.

3

True.

4

D.

5

B.

6

D.

7

C.

8

C.

9

C.

10

C.

11

B.

12

容量缺失, 冲突缺失

13

2 TB, 1250 GB.

14

91%, 27.2ns

15

(1)

TLBI 2位, TLBT 6位, VPN 8位, PPN 6位, VPO 6位, PPO 6位, CT 6位, CI 4位, CO 2位.

VA 各位拆分如下

1	1 1	0
2	3 2 1 0 9 8 7 6 5 4 3 2 1 0	
3	VPN VPO	
4	TLBT TLBI	

PA 各位拆分如下

1	1 1	0
2	1 0 9 8 7 6 5 4 3 2 1 0	
3	PPN PPO	
4	CT CI CO	

(2)

	VA	0x027C	0x03A9	0x0040	0x03D7
TLB 命中	F	F	F	T	
缺页	F	F	T	F	
PPN	0x17	0x11	-	0x0D	
Cache 命中	F	F	-	T	
字节偏移	0	1	0	3	
返回的缓存字节	-	-	-	0x1D	

16

这样拆分是为了更好地配合程序运行的空间局部性, 使地址临近的内存可以同时被装入缓存.

举例说明如下.

假设地址为 12 位. 缓存每行 16B, 即偏移占低 4 位; 缓存 16 行, 使用直接映射, 索引 4 位; 剩余 4 位为标记.

假定有以下程序,

```
1 int8_t a[128]; // at address 0x100 ~ 0x17F
2 int8_t b[128]; // at address 0x180 ~ 0x1FF
3
4 void foo()
5 {
6     for (int i = 0; i < 128; i++)
7         b[i] = a[i];
8 }
```

访存序列是 0x100, 0x180, 0x101, 0x181, ..., 0x17F, 0x1FF. 初始状态缓存全都无效.

如果索引占中间 4 位

0x100 Miss, 0x10_ 的 16B 装入 Index 0;

0x180 Miss, 0x18_ 的 16B 装入 Index 8;

0x101 Hit @ Index 0;

0x181 Hit @ Index 8;

...

0x110 Miss, 0x11_ 装入 Index 1;

0x190 Miss, 0x19_ 装入 Index 9;

0x111 Hit @ Index 1;

0x191 Hit @ Index 9;

...

以此类推, 共访存 256 次, 32 次缺失 (均为必然缺失).

如果索引占高 4 位

0x100 Miss, 0x10_ 装入 Index 1 (Tag 0);

0x180 Miss, 0x18_ 装入 Index 1 (Tag 8), 替换出 0x10_ (Tag 0);

0x101 Miss, 0x10_ 装入 Index 1 (Tag 0), 替换出 0x18_ (Tag 8);

...

以此类推, 共访存 256 次, 缺失 256 次, 其中只有 1 次必然缺失 (0x100 那一次), 其余都是冲突缺失.

17

(1)

VPN 21位, PPN 17位;

Cache Tag 12位, Index 10位, Offset 6位.

(2)

Cache Tag 22位, Index 2位, Offset 4位.

A 的访存顺序是 0x420, 0x4A0, 0x424, 0x4A4, 0x428, 0x4A8, 0x42C, 0x4AC;

B 的访存顺序是 0x420, 0x424, 0x428, 0x42C, 0x4A0, 0x4A4, 0x4A8, 0x4AC. (每个地址访存两次: 一次读, 一次写)

Cache A

	T	V	W0	W1	W2	W3	T	V	W4	W5	W6	W7
0	-	0	-	-	-	-	-	0	-	-	-	-
1	-	0	-	-	-	-	-	0	-	-	-	-
2	0x10	1	a[0]+1	a[1]+1	a[2]+1	a[3]+1	0x12	1	a[32]+1	a[33]+1	a[34]+1	a[35]+1
3	-	0	-	-	-	-	-	0	-	-	-	-

#1 必然缺失, 将 0x42_ 的 16B 装入 Index 2 的其中一组; #2 命中; #3 必然缺失, 将 0x4A_ 的 16B 装入 Index 2 的另外一组 (因这一组此时 V=0); 其余次访存均命中. 命中率 $14/16 = 0.875$.

Cache B

	T	V	W0	W1	W2	W3	T	V	W4	W5	W6	W7
0	-	0	-	-	-	-	-	0	-	-	-	-
1	-	0	-	-	-	-	-	0	-	-	-	-
2	0x10	1	a[0]+1	a[1]+1	a[2]+1	a[3]+1	0x12	1	a[32]+1	a[33]+1	a[34]+1	a[35]+1
3	-	0	-	-	-	-	-	0	-	-	-	-

#1 必然缺失, #2 ~ #8 命中; #9 必然缺失, #10 ~ #16 命中. 命中率 $14/16 = 0.875$.

在本题中, 由于缓存使用了组相联, 因此两块可以同时被读入缓存, 即使按 A 那样违背局部性的方法访存, 也不会造成额外的冲突缺失. 如果是直接映射, A 的命中率将为 0.