5.3.3 Cache和主存之间如何映射？

刘 芳 副教授
国防科学技术大学计算机学院

5.3.3 Cache和主存之间的映射方式

Cache映射解决什么问题？

- 将要访问的局部主存数据取到Cache中，应该放到Cache的何处？
- Cache行比主存块少，多个主存块会映射到同一个Cache行中，如何建立Cache地址与主存地址的对应关系？

如何进行映射？

- 把主存划分成大小相等的主存块(Block)
- Cache中存放一个主存块的对应单位称为行(line) 或槽(Slot)或项(Entry)或块(Block)
- 主存块和Cache行可按三种方式进行映射：直接、全相联、组相联

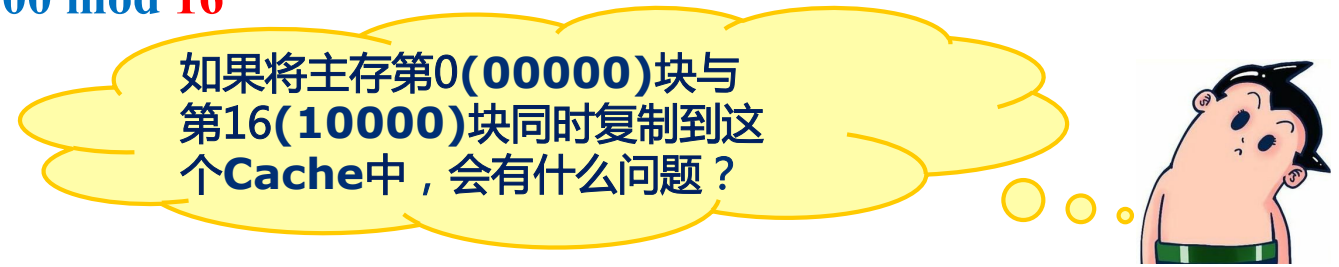
直接映射

把主存的每一块映射到一个固定的Cache行中。即每个主存地址对应于高速缓存中唯一的地址，也称模映射

映射关系为：Cache行号 =主存块号 mod Cache行数

例： 假定Cache共有16行，主存中的第100块，应该映射到Cache的哪个位置？

求解：4 = 100 mod 16



由于它们都映射到Cache第0行，即使Cache中其它行空闲，也有一个主存块不能写入Cache，这样就会产生频繁的 Cache替换，称之为Cache抖动

直接映射

把主存的每一块映射到一个固定的Cache行中。即每个主存地址对应于高速缓存中唯一的地址，也称模映射

映射关系为：Cache行号 =主存块号 mod Cache行数

例： 假定Cache共有16行，主存中的第100块，应该映射到Cache的哪个位置？

求解：4 = 100 mod 16

特点

- 易实现，命中时间短
- 淘汰 / 替换策略简单
- 不灵活，Cache存储空间得不到充分利用，命中率低

直接映射

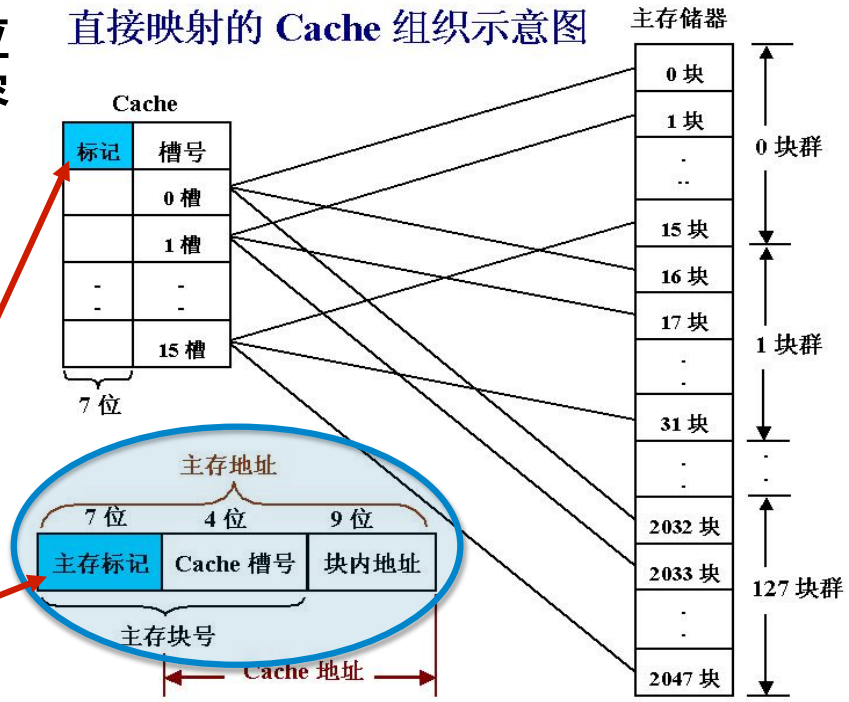
若数据在主存和Cache之间按块传送单位为512字节。Cache大小为8KB，主存容量为1MB

Cache：8KB= 2^{13}B =16槽×512B/槽

主存大小：1024KB=2^{20}B=2^{11}块 × 512B/ 块

=2^{7}×2^{4}×512B/ 块

- Cache标记(tag)指出对应槽取自哪个主存块群
- 主存tag指出对应地址位于哪个块群



直接映射

若数据在主存和Cache之间按块传送单位为512字节。Cache大小为8KB，主存容量为1MB

Cache：8KB= 2^{13}B =16槽×512B/槽

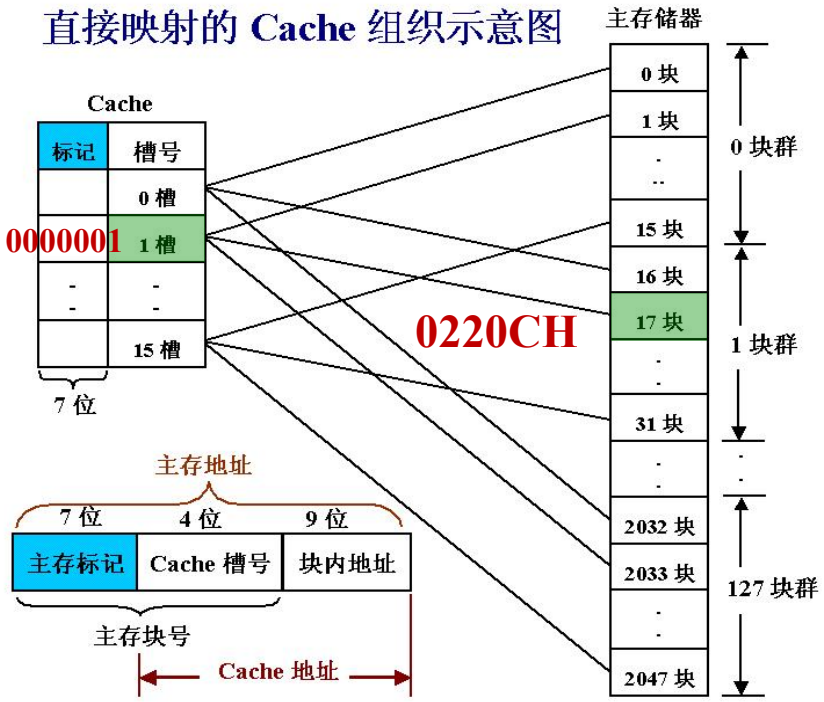
主存大小：1024KB=2^{20}B=2^{11}块 × 512B/ 块

=2^{7}×2^{4}×512B/ 块

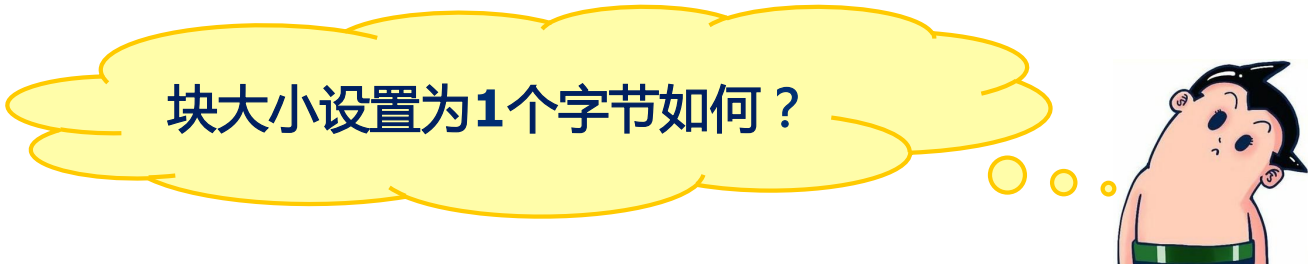
例：若Cache为空，如何对0220CH单元进行访问？

0000 001 0001 0 0000 1100B

第1块群中0001块(第17块)的第12号单元！



5.3.3 Cache和主存之间的映射方式 —— 直接映射



利用时间局部性：某字节不久又可能被使用

没有空间局部性：虽然某字节的邻近字节不久可能被访问，但没有被调到Cache(因为每次只调入一个字节！)

冲突概率增大，块小使映射到同一个Cache行的主存块几率增加

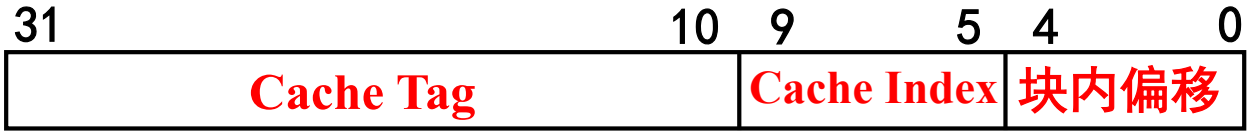
增大块的大小，从而利用空间局部性！

直接映射(Direct Mapped Cache)

例：一个直接映射Cache：容量为1 KB；块大小为32Byte

2^N byte cache,
此例中N=10

Block Size=2^M,
此例中M=5



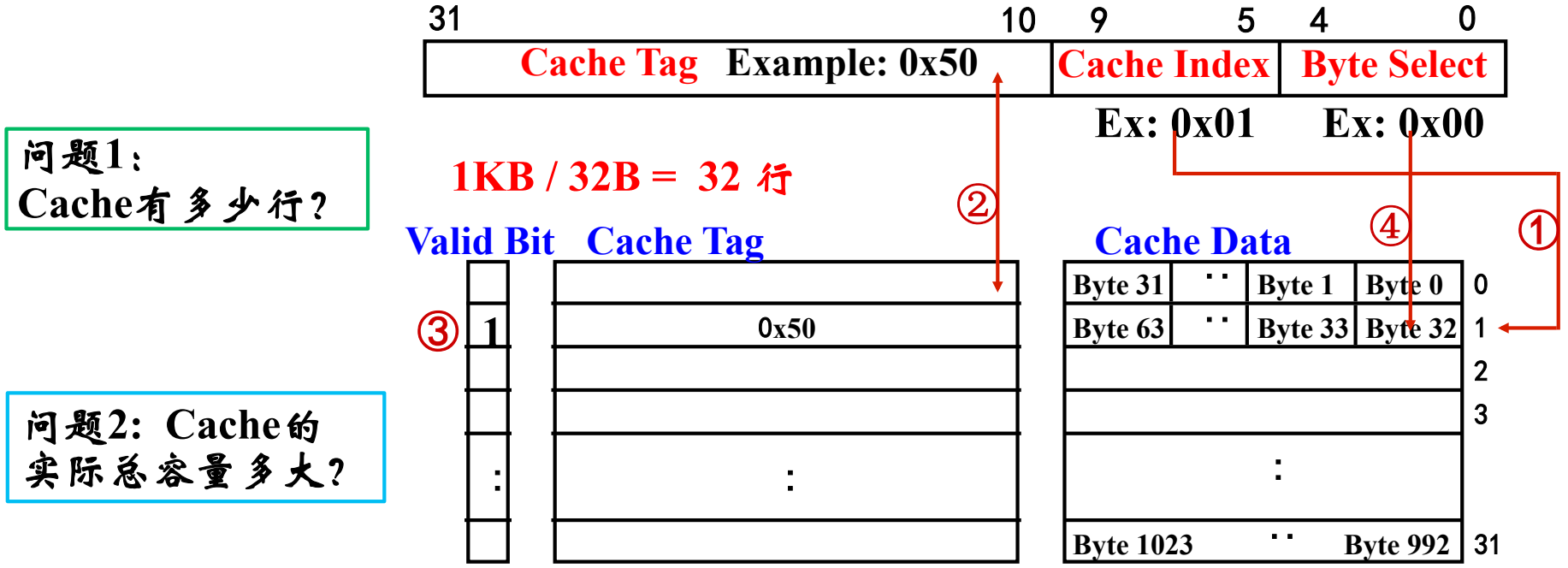
最高的 (32 - N) 地址位： Cache标记域

中间(N-M)地址位：Cache索引

最低的M地址位：块内偏移

直接映射(Direct Mapped Cache)

例：一个直接映射Cache：容量为1 KB；块大小为32Byte



容量： 32× (1+ 22)+1K×8=8928bits=1116B，数据占： 1024B / 1116B = 91.76%

全相联映射

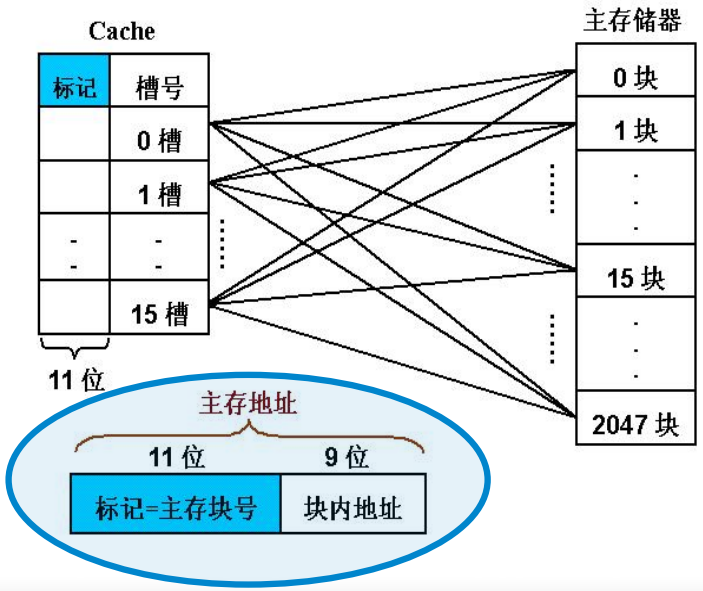
主存块可装到Cache任一行/槽中，称为全相联映射

如果数据在主存和Cache之间块传送单位为512B

Cache大小：8KB=2^{13}B=16槽× 512B/ 槽

主存大小：1MB=2^{20}B=2^{11}块 × 512B/ 块

全相联映射的 Cache 组织示意图



全相联映射

主存块可装到Cache任一行/槽中，称为全相联映射

如果数据在主存和Cache之间块传送单位为512B

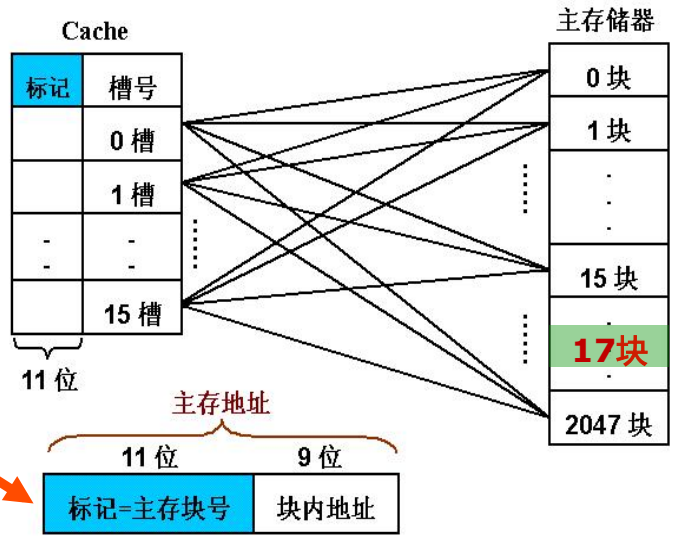
Cache大小：8KB=2^{13}B=16槽× 512B/ 槽

主存大小：1MB=2^{20}B=2^{11}块 × 512B/ 块

- Cache标记(tag)指出对应槽取自哪个主存块
- 主存tag指出对应地址位于哪个主存块

两个标记相等，说明要找的地址在对应槽中。比较所有标记都不等，则失靶

全相联映射的 Cache 组织示意图



全相联映射

主存块可装到Cache任一槽Slot中，称为全相联映射

如果数据在主存和Cache之间块传送单位为512B

Cache大小：8KB=2^{13}B=16槽× 512B/ 槽

主存大小：1MB=2^{20}B=2^{11}块 × 512B/ 块

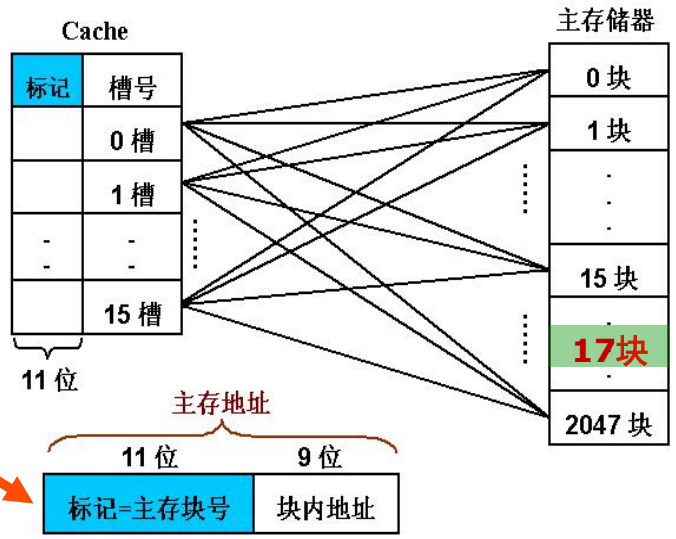
- Cache标记(tag)指出对应槽取自哪个主存块
- 主存tag指出对应地址位于哪个主存块

例：CPU如何访问0220CH单元？

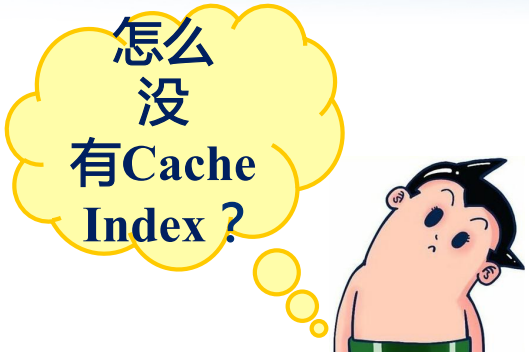
000 0001 0001 0 0000 1100B

第17块的第12个单元！可映射到任意cache槽

全相联映射的 Cache 组织示意图



5.3.3 Cache和主存之间的映射方式 —— 全相联映射



因为同时比较所有Cache行的tag标记

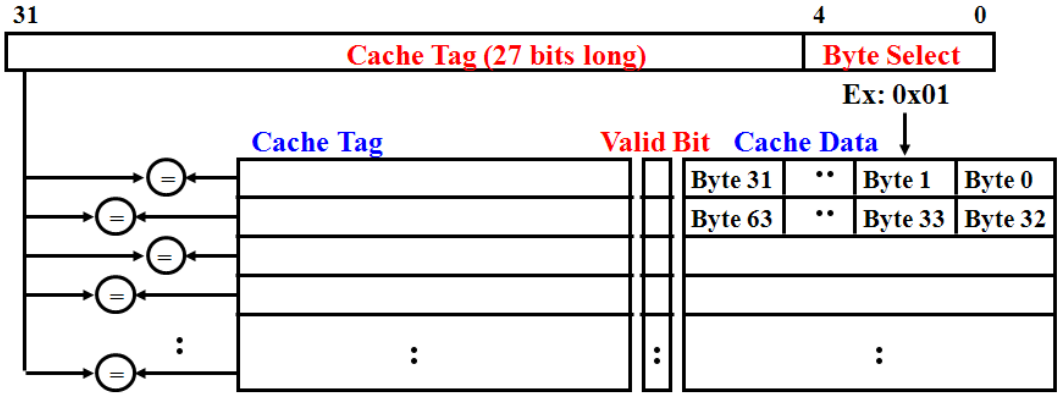
- 块冲突概率低：只要有空闲Cache块，都不会发生冲突
- 实现复杂(比较逻辑的硬件代价大)、速度慢

例：若主存地址32位，Cache块大小32B 。比较器位数多长？

$32B=2^5B$

$32 - 5 = 27$

需要N=27bit的比较器



组相联映射

将Cache所有行分组，把主存块映射到Cache固定组的任一行中。即：组间模映射、组内全映射

映射关系为：

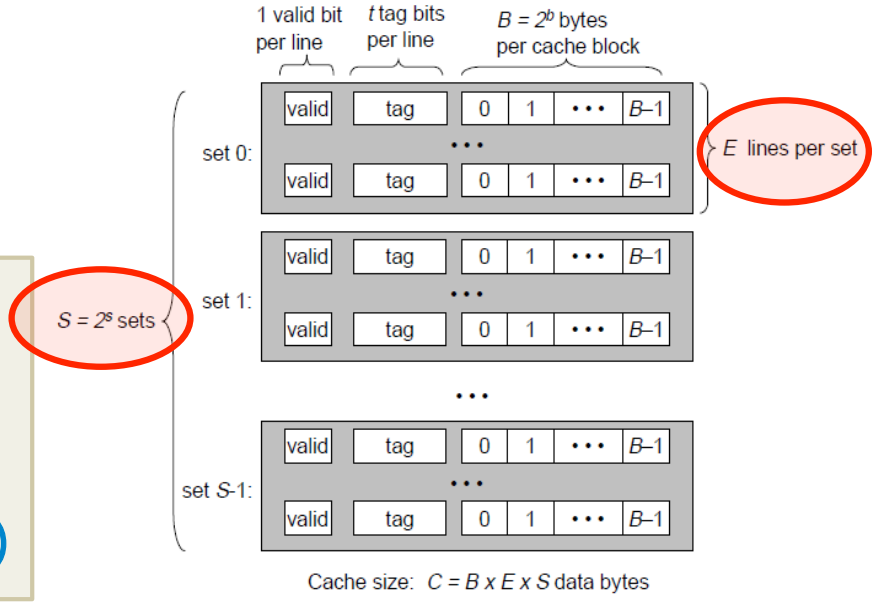
Cache组号=主存块号 mod Cache组数

例：若Cache划分为：

8K字=8组×2行/组×512字节/行

100 mod 8 = 4

(主存第100块映射到Cache组4中的任意一行)



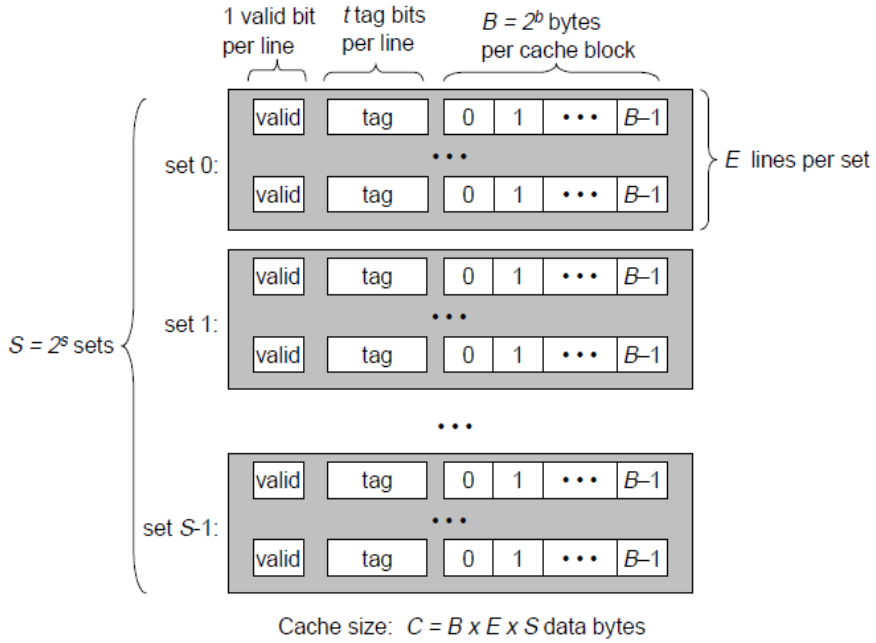
组相联映射

特点

•结合直接映射和全相联映射的优点。

当Cache的组数为1时，则为全相联映射；当每组只有一行时，则为直接映射

•每组两个行（2路组相联）较常用。在较大容量的L2 Cahce和L3 Cahce中使用4路以上



5.3.3 Cache和主存之间的映射方式 —— 组相联

如果数据在主存和Cache之间块传送单位为512B

Cache大小：8KB=2^{13}B=16槽×512B/槽

=8组×2槽/组×512B/槽

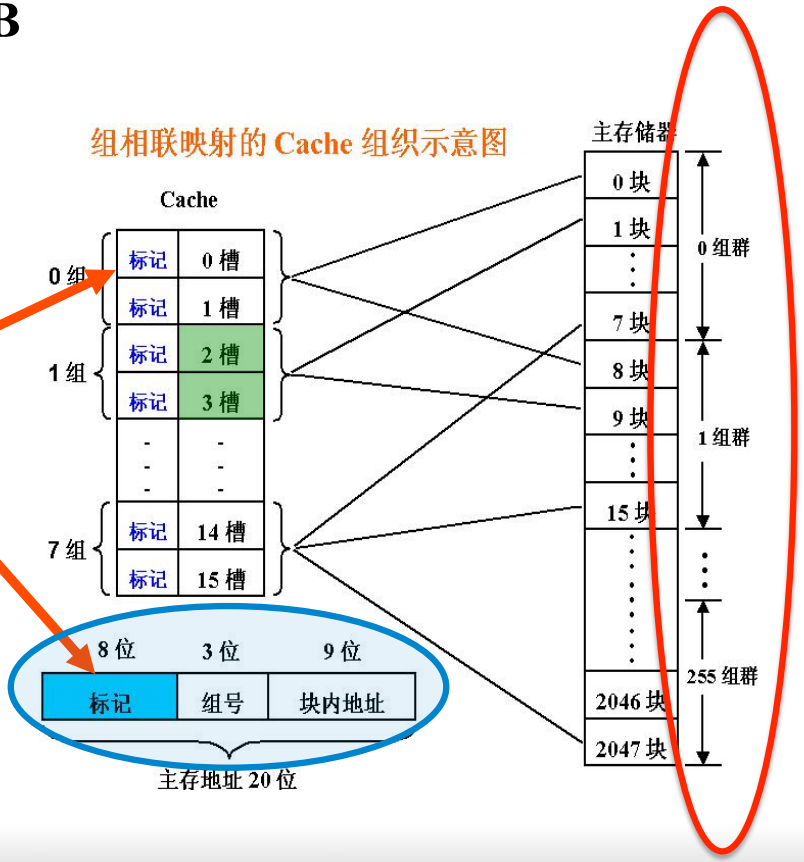
主存大小：1MB=2^{20}B=2^{11}块 × 512B/ 块

=2^{8}组× 8块/组× 512B/ 块

- Cache标记(tag)指出对应槽取自哪个主存组群
- 主存tag指出对应地址位于哪个主存组群中

将主存地址标记和对应Cache组中每个Cache标记比较：

两个标记相等时，说明要找的地址在对应槽中



5.3.3 Cache和主存之间的映射方式 —— 组相联

如果数据在主存和Cache之间块传送单位为512B

Cache大小：8KB=2^{13}B=16槽×512B/槽

=8组×2槽/组×512B/槽

主存大小：1MB=2^{20}B=2^{11}块 × 512B/ 块

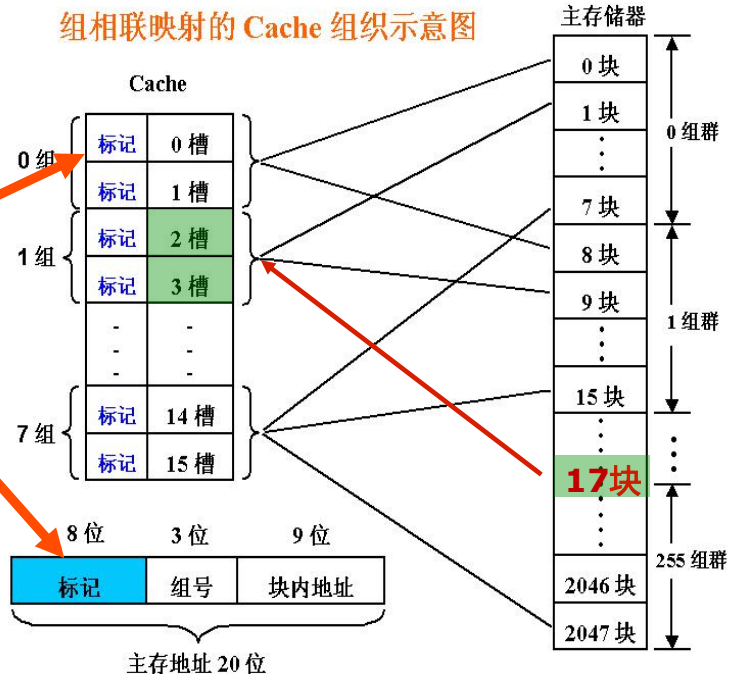
=2^{8}组×8块/组× 512B/ 块

- Cache标记(tag)指出对应槽取自哪个主存组群
- 主存tag指出对应地址位于哪个主存组群中

例：CPU如何对0220CH单元进行访问？

0000 0010 001 0 0000 1100B

第2组群中001块(第17块)的第12个单元



5.3.3 Cache和主存之间的映射方式

- 高速缓存的缺失率和关联度

三种映射方式

- 直接映射：唯一映射(只有一个可能的位置)
- 全相联映射：任意映射(每个位置都可能)
- N-路组相联映射：N-路映射(有N个可能的位置)

什么是关联度？

主存块映射到Cache时，可能存放的位置个数

关联度示例

One-way set associative
(direct mapped)

Block	Tag	Data
0		
1		
2		
3		
4		
5		
6		
7		

关联度为多少？ 1

Two-way set associative

Set	Tag	Data	Tag	Data
0				
1				
2				
3				

关联度为多少？ 2

Four-way set associative

Set	Tag	Data	Tag	Data	Tag	Data	Tag	Data
0								
1								

关联度为多少？ 4

Eight-way set associative (fully associative)

Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data	Tag	Data

关联度为多少？ 8

5.3.3 Cache和主存之间的映射方式

三种映射方式的关联度

主存块映射到Cache时，可能存放的位置个数

- ◆关联度最低？直接映射（关联度为1）
- ◆关联度最高？全相联映射（为Cache行数）
- ◆关联度居中？N-路组相联映射（关联度为N）

关联度和misss rate什么关系？

直观结论（Cache大小和块大小一定的情况下)

- ◆提高关联度通常能够降低缺失率(miss rate)；
- ◆提高关联度通常会增加命中时间

5.3.3 Cache和主存之间的映射方式

若三个大小相等的Cache，分别采用三种不同的映射策略，均有四行，每行一个字，按以下主存块地址顺序访问：

0——>8——>0——>6——>8

右边三种情况各对应哪种Cache?

Cache1：直接映射

Cache2：2路组相联

Cache3：全相联

相联度高，缺失率低

Address of memory block accessed	Hit or miss	Contents of cache blocks after reference			
		0	1	2	3
0	miss	Memory[0]			
8	miss	Memory[8]			
0	miss	Memory[0]			
6	miss	Memory[0]		Memory[6]	
8	miss	Memory[8]		Memory[6]	

1

Cache块号=主存块号 mod 4

Address of memory block accessed	Hit or miss	Contents of cache blocks after reference			
		Set 0	Set 0	Set 1	Set 1
0	miss	Memory[0]			
8	miss	Memory[0]	Memory[8]		
0	hit	Memory[0]	Memory[8]		
6	miss	Memory[0]	Memory[6]		
8	miss	Memory[8]	Memory[6]		

2

Cache组号=主存块号 mod 2

Address of memory block accessed	Hit or miss	Contents of cache blocks after reference			
		Block 0	Block 1	Block 2	Block 3
0	miss	Memory[0]			
8	miss	Memory[0]	Memory[8]		
0	hit	Memory[0]	Memory[8]		
6	miss	Memory[0]	Memory[8]	Memory[6]	
8	hit	Memory[0]	Memory[8]	Memory[6]	

3

分析 ：Cache缺失带来的损失到底多大？

例1：假定执行某程序时，指令Cache的缺失率为2%，数据Cache的缺失率为4%。若一个处理器在没有任何存储器阻塞时的CPI为2，miss penalty为100个时钟周期。如果用SPECint2000来衡量，若有一个从不产生缺失的理想Cache，那么机器速度能快多少？

分析：

指令的缺失损失时钟数为：I×2%×100=2.0×I

SPECint2000的访存指令(Load和Store)频度为：36%，所以数据的缺失损失时钟数为：I×36%×4%×100=1.44×I

指令和数据总的缺失损失时钟数为：2×I+1.44×I=3.44I，也即：

平均每条指令要有3.44个时钟周期处在存储器阻塞状态。即每一条指令都需要3.44个时钟周期去停下来到内存中取相应的数据

因此，由于存储器阻塞而使得CPI数增大到2+3.44=5.44. 故：

$$\frac{\text{CPU time with stalls}}{\text{CPU time with perfect cache}} = \frac{\text{I}\times\text{CPIstall}\times\text{Clock cycle}}{\text{I}\times\text{CPIperfect}\times\text{Clock cycle}} = \frac{5.44}{2}$$

带有理想Cache的机器性能更快

分析 ：处理器速度提高而存储器不变时的情况

例2：假定上例中CPI减为1，时钟频率不变，则：

因为存储器阻塞而使得CPI数增大到1+3.44=4.44。故：

$$\frac{\text{CPU time with stalls}}{\text{CPU time with perfect cache}} = \frac{I \times \text{CPIstall} \times \text{Clock cycle}}{I \times \text{CPIperfect} \times \text{Clock cycle}} = \frac{4.44}{1}$$

可知：存储器阻塞所花时间占整个执行时间的比例：

3.44 / 5.44=63% ——》上升到 3.44 / 4.44=77%

结论：CPI越小，Cache缺失的影响越大

分析 ：处理器速度提高而存储器不变时的情况

例3：若例1中时钟频率加倍，CPI不变，则：若主存速度不改变，即绝对时间不变。所以，miss损失为200个时钟周期。

每条指令发生的总cache缺失时钟数为：(2%×200)+36%×(4%×200)=6.88

故：存储器阻塞使得CPI数增大到2+6.88=8.88

结论：CPU时钟频率越高，Cache缺失损失就相对越大

$$\frac{\text{时钟快的机器的性能}}{\text{时钟慢的机器的性能}} = \frac{I \times CPI_{\text{stall of slow}} \times \text{Clock cycle}}{I \times CPI_{\text{stall of fast}} \times \text{Clock cycle}/2} = \frac{5.44}{8.88/2} = 1.23$$

处理器性能越高，高速缓存的性能就越重要！