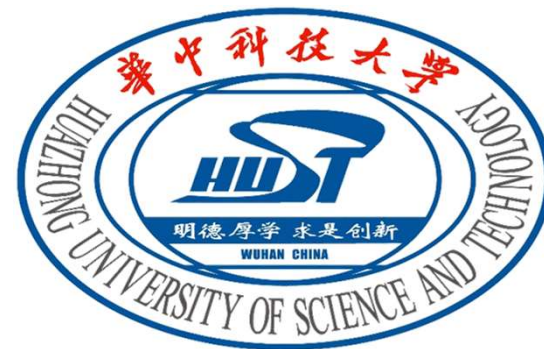


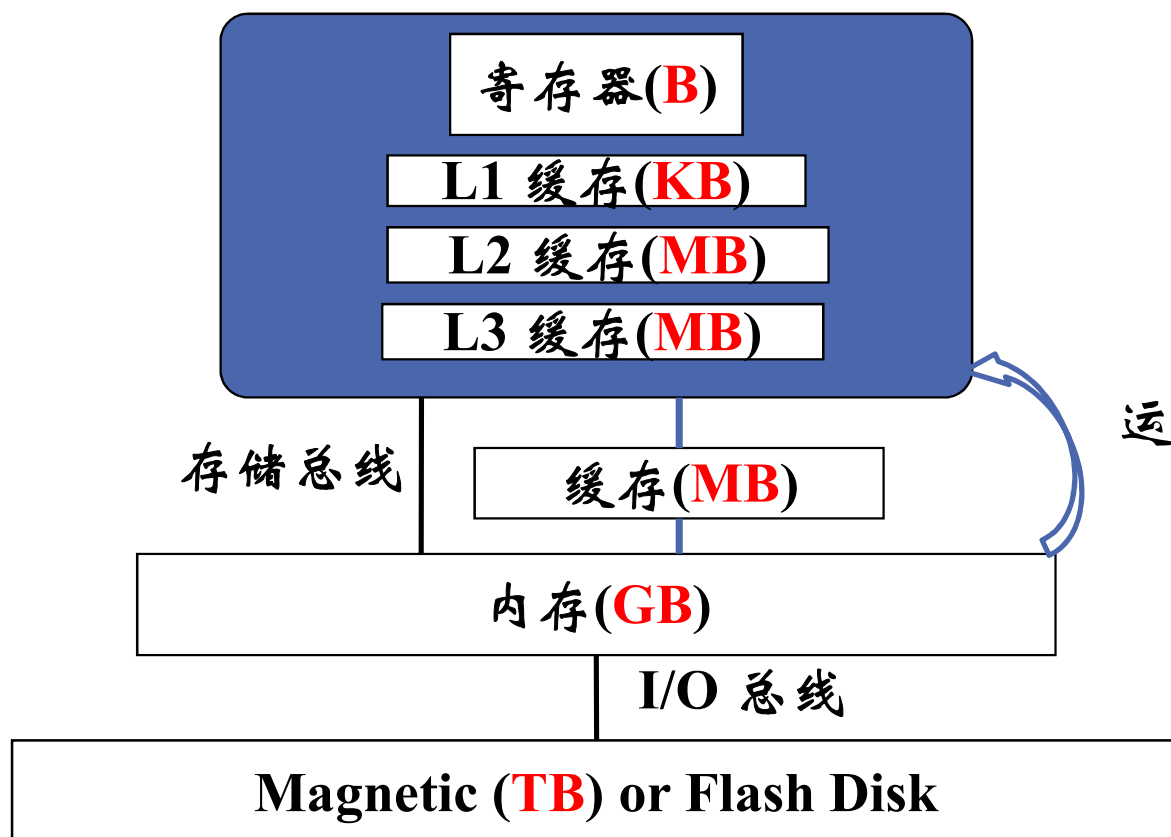
微机原理与接口技术

高速缓存映射机制

华中科技大学 左冬红



回顾



程序运行过程中CPU给出内存地址，如何确定给定内存地址的数据或指令是否在缓存中？

一次读取一片(小)连续区域的代码和数据到缓存

术语

命中

程序运行过程中，CPU给出的内存地址中的数据或指令在高速缓存中

命中率

CPU任一时刻从缓存中可靠获取数据的几率

未命中

程序运行过程中，CPU给出的内存地址数据或指令不在高速缓存中

未命中率

$1 - \text{命中率}$

CPU访问数据、指令时延

命中

访问时延即为高速缓存的响应时延

未命中

一次读取一片(小)连续区域的代码和数据到缓存

访问时延即为从内存中读取一片连续区域数据的响应时延加上高速缓存的响应时延

惩罚时间

从内存中读取一片连续区域数据的响应时延

一片(小)连续区域定义为缓存的一行

平均访问时延

CPU访问数据的缓存命中率为 p

从缓存获取一个数据所需时间为 t_1

缓存的行大小为 n_c

缓存与内存之间的总线宽度为 b

从内存中获取一个数据所需时间为 t_2

CPU访问数据的平均时延:

$$t_{av} = t_1 + (1 - p) \times \frac{n_c}{b} \times t_2$$

命中 未命中

Cache映射机制

0				
4				
8				
12				
16				
20				
24				
28				

cache

建立某种查找机制

查什么？

建立地址映射机制

程序运行过程中，CPU给出的**内存地址**中的数据或指令是否在高速缓存中

0				
4				
8				
12				
16				
20				
24				
28				
32				
36				
40				
44				
48				
52				
56				
60				
64				
68				
72				
76				
80				
84				
88				
92				

内存

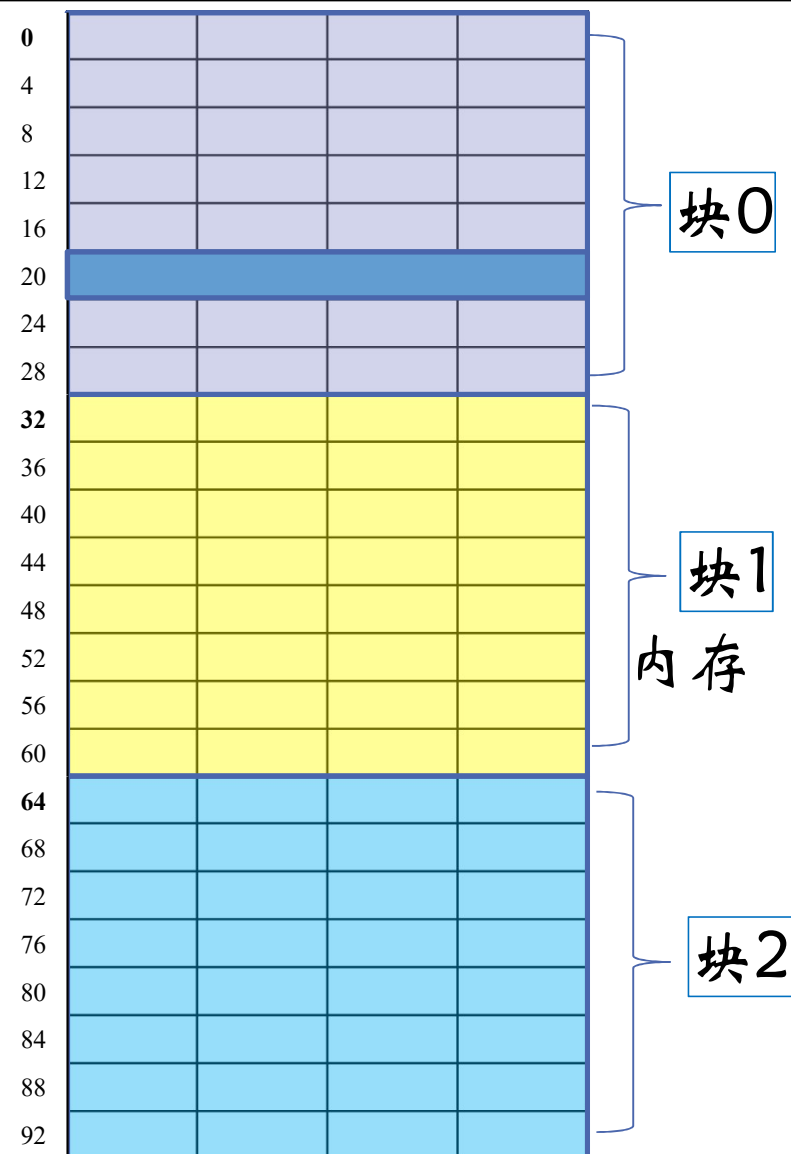
Cache映射机制

内存的容量是Cache容量的整数倍，
并按cache容量大小分块

0				
4				
8				
12				
16				
20				
24				
28				

cache

内存和Cache按照同样大小分行，
行大小为 2^n



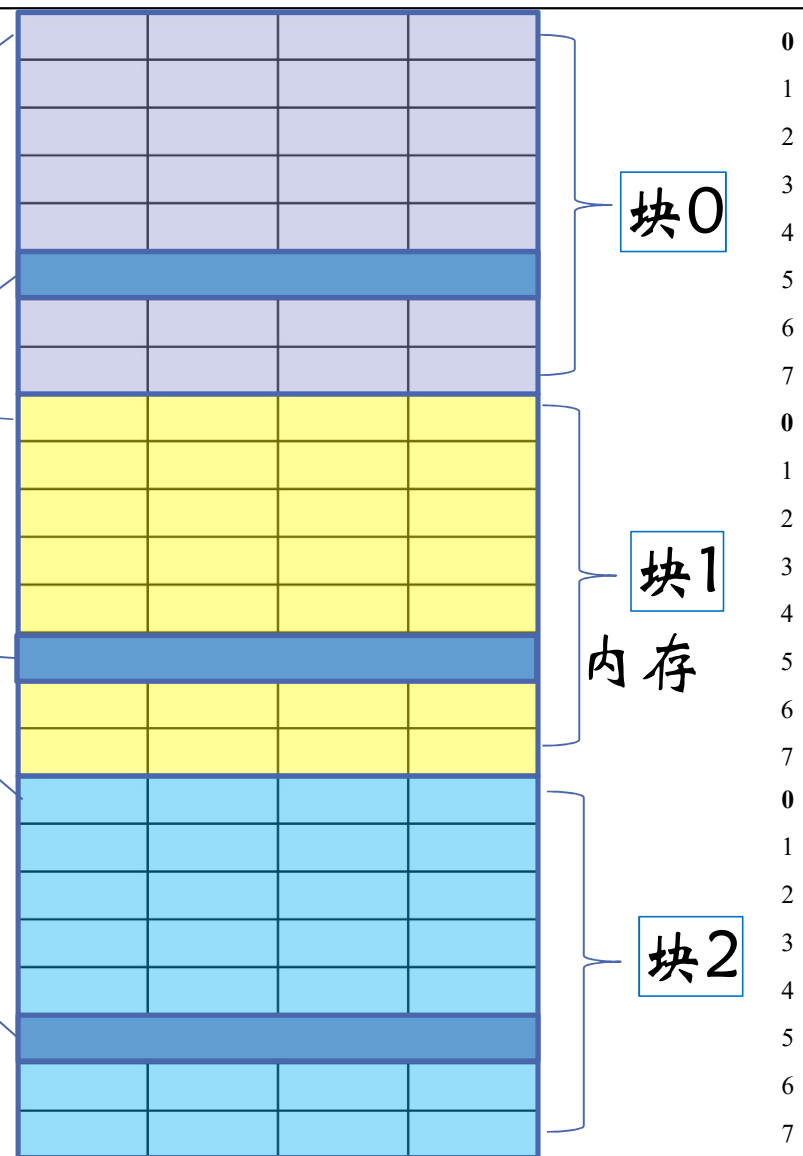
直接映射机制

行地址

0				
1				
2				
3				
4				
5				
6				
7				

cache

内存中任意一块中某行的数据存入缓存时只能存入缓存中编号相同的行



直接映射策略

内存物理地址

块号	块内行号	行内偏移
$2b$	$3b = \log_2 8$	$2b = \log_2 4$

缓存一行4个字节

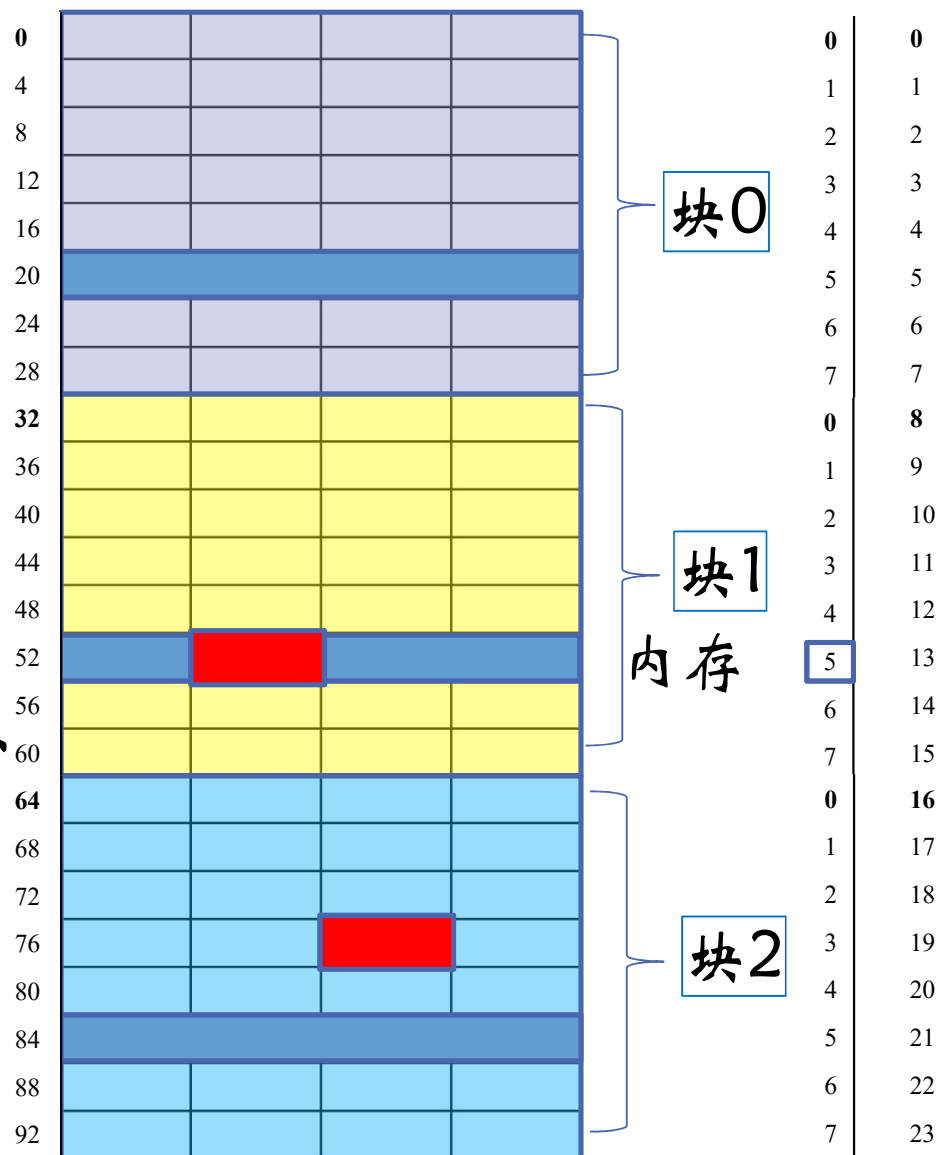
内存行号: $\frac{53}{4} = 13$ 内存地址
行大小

行内偏移: $53 \% 4 = 1$ 内存地址 % 行大小

$$13 \% 8 = 5$$

对应缓存行号: 内存行号 % 缓存行数

内存块号: $\frac{13}{8} = 1$ 内存行号
缓存行数



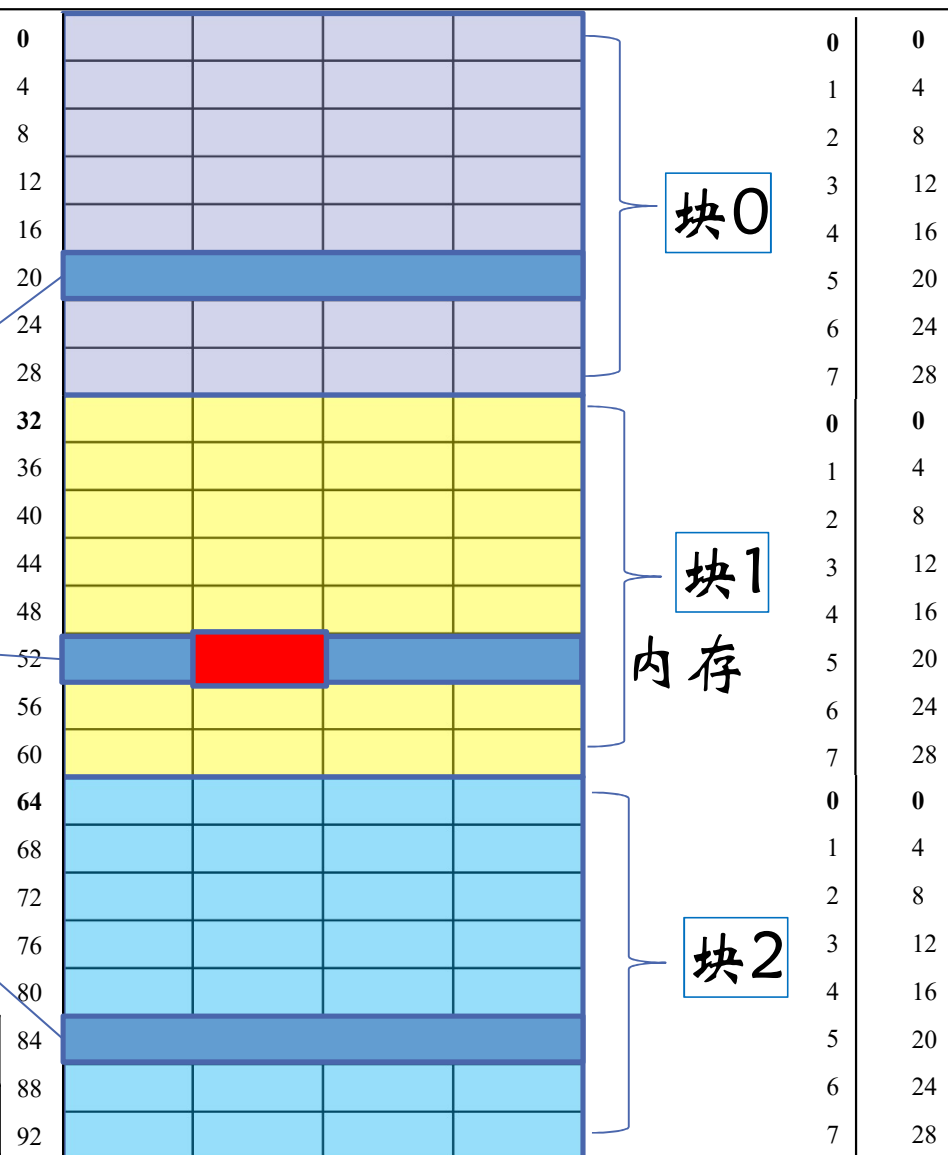
直接映射机制

		Tag 块号 v				
0	0					
1	4					
2	8					
3	12					
4	16					
5	20				00	
6	24					
7	28					

cache

有数据还是初始状态?

V (有效位)	标志Tag (块号)	数据
1b	2b	32b



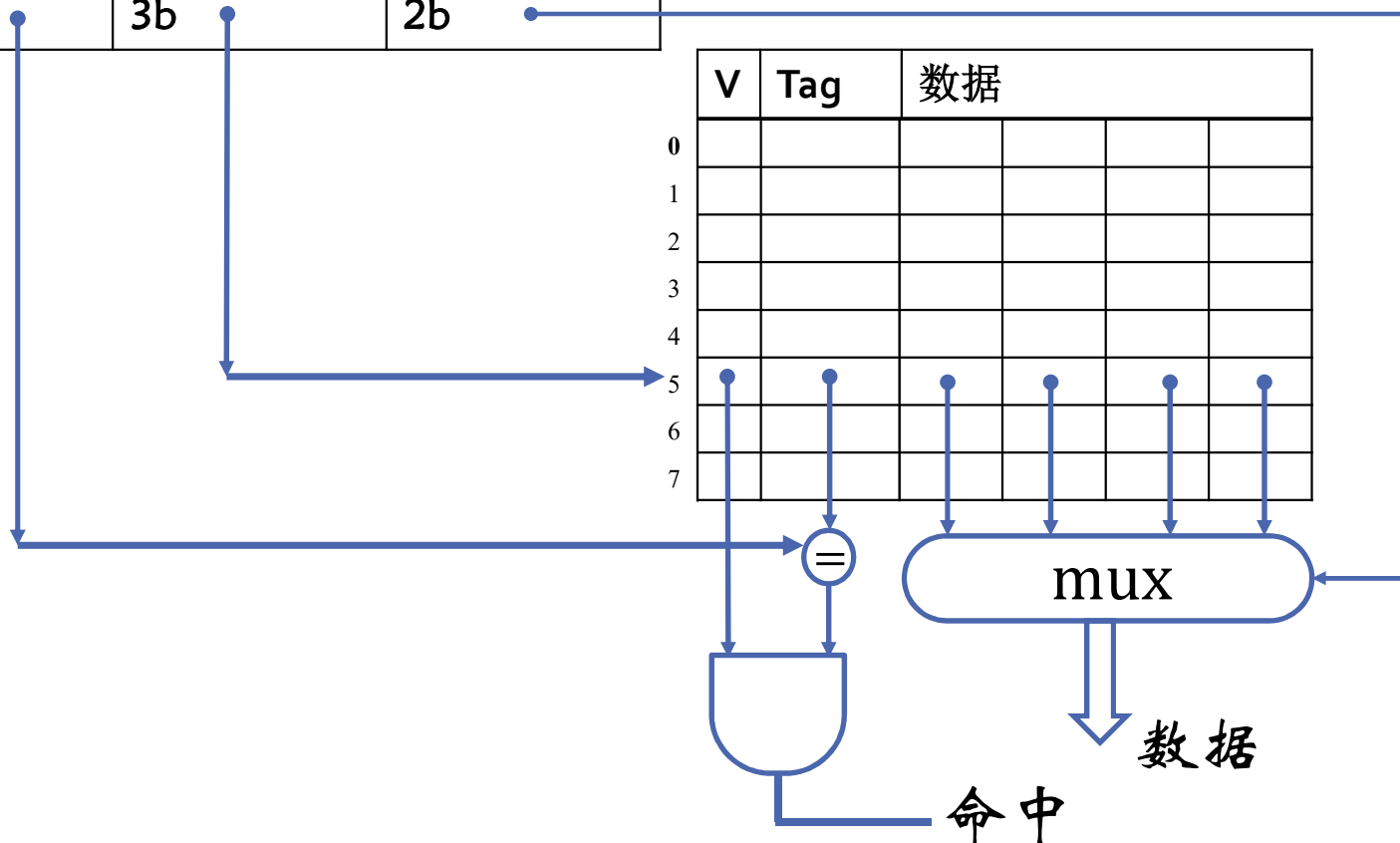
直接映射策略

内存物理地址

块号	块内行号	行内偏移
2b	3b	2b

缓存一行4字节，共8行

	V	Tag	数据			
0						
1						
2						
3						
4						
5						
6						
7						

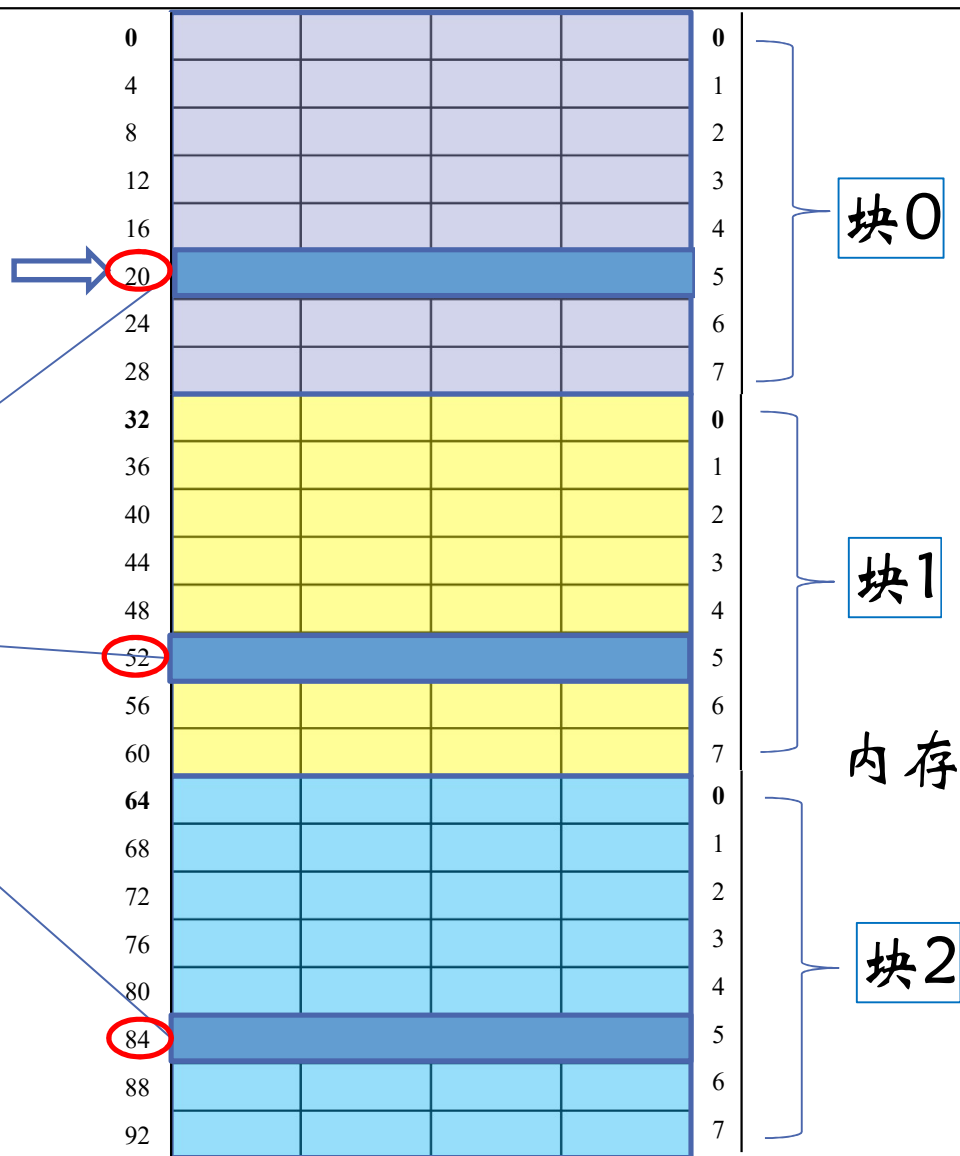


直接映射机制

	v	Tag					
0							
1							
2							
3							
4							
5	1	00					
6							
7							

cache

都未命中且Cache有空闲空间
Cache利用率低，命中率低



全相联映射机制

内存物理地址

行号	行内偏移
5b	2b

v Tag

1	5				
1	13				
1	21				
	行号				

命中

cache

内存和Cache按照同样大小分行，
行大小为 2^n

内存行可映射到cache任意行

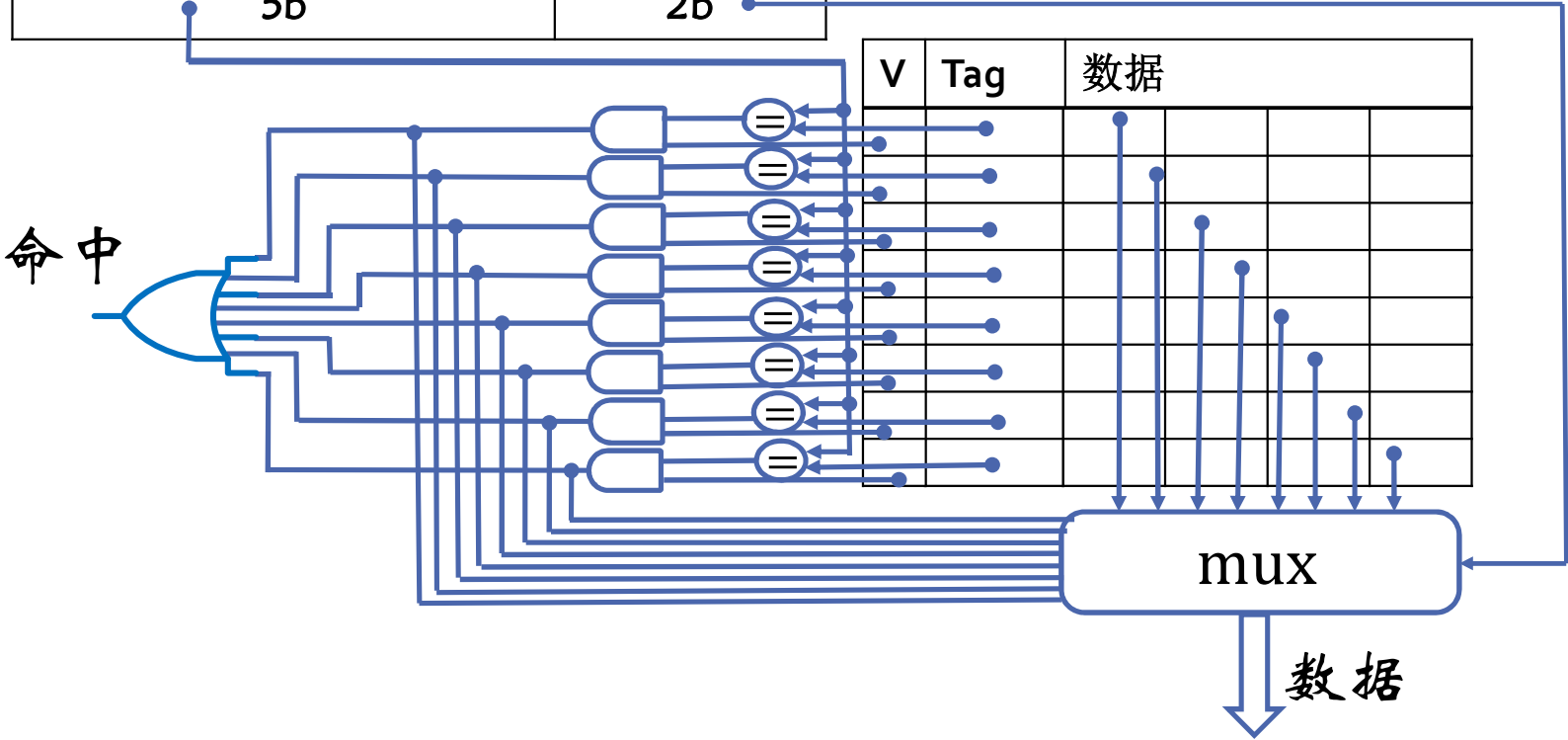
0					0
4					1
8					2
12					3
16					4
20					5
24					6
28					7
32					8
36					9
40					10
44					11
48					12
52					13
56					14
60					15
64					16
68					17
72					18
76					19
80					20
84					21
88					22
92					23

全相联映射策略

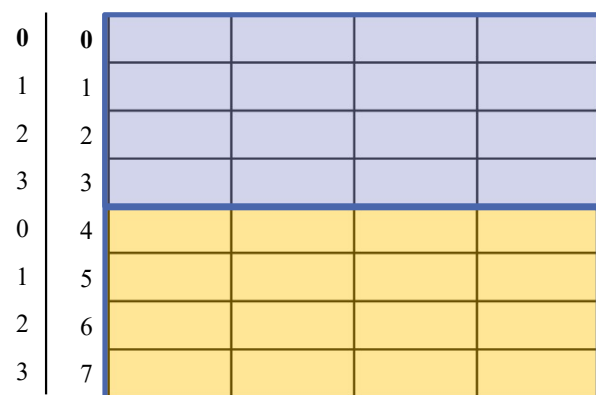
内存物理地址

行号	行内偏移
5b	2b

缓存一行4字节，共8行



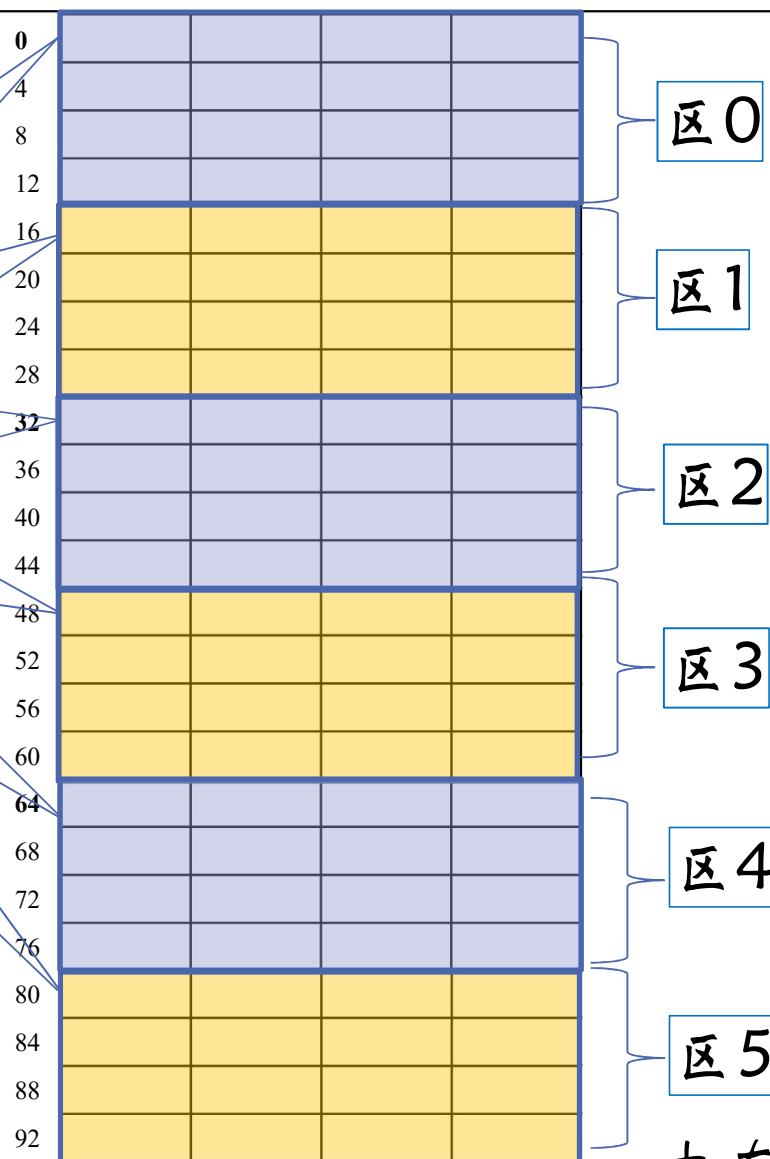
组相联映射-n way set association



cache

内存和缓存按照同样大小分区，
每区大小为 2^m 行

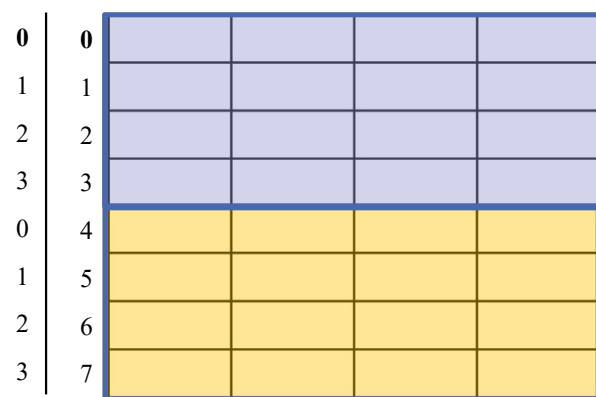
缓存各区行号相同的行属于同一组，
行号也称为组号



内存

0	0
1	1
2	2
3	3
0	4
1	5
2	6
3	7
0	8
1	9
2	10
3	11
0	12
1	13
2	14
3	15
0	16
1	17
2	18
3	19
0	20
1	21
2	22
3	23

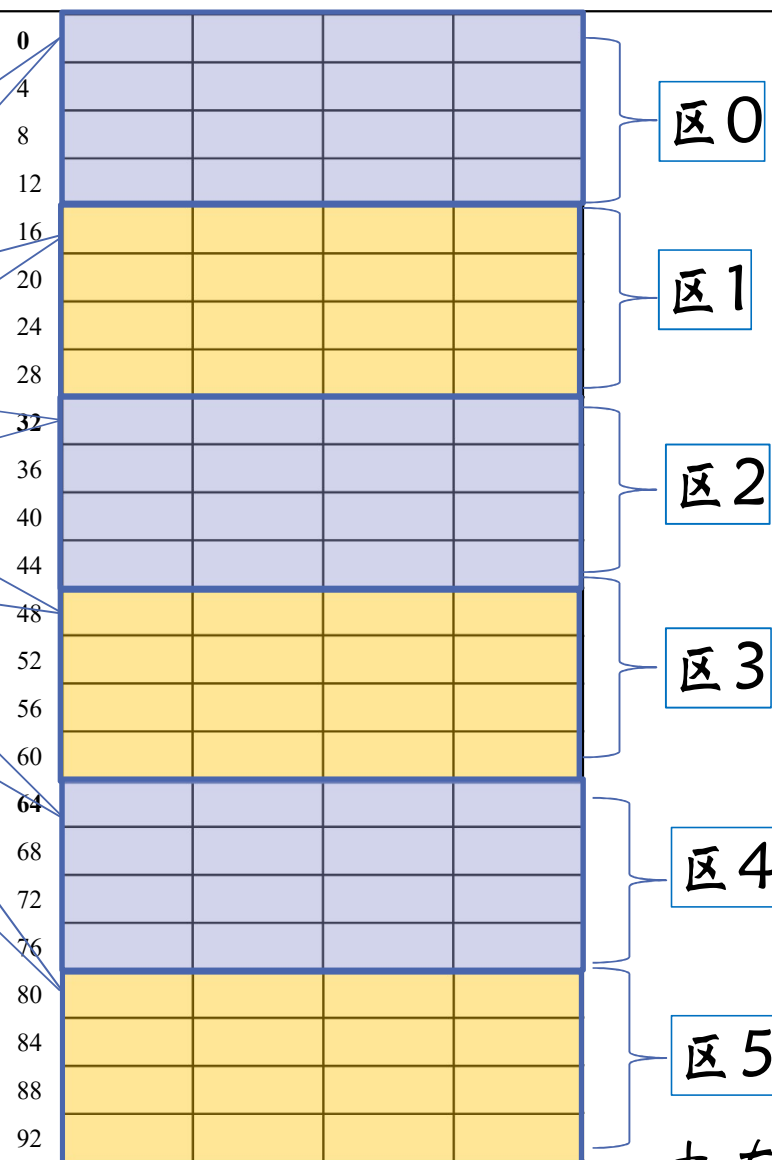
组相联映射-n way set association



cache

内存各区中的行直接映射到缓存中行号相同的组，组内可选择任意行映射

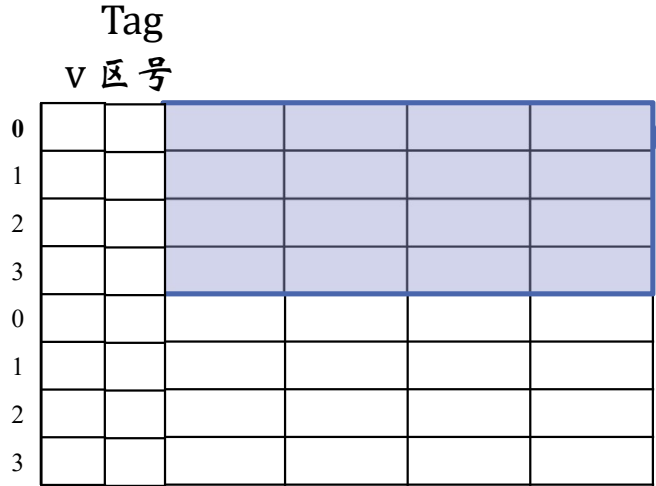
$n=2$



内存

0	0
1	1
2	2
3	3
0	4
1	5
2	6
3	7
0	8
1	9
2	10
3	11
0	12
1	13
2	14
3	15
0	16
1	17
2	18
3	19
0	20
1	21
2	22
3	23

组相联映射-n way set association

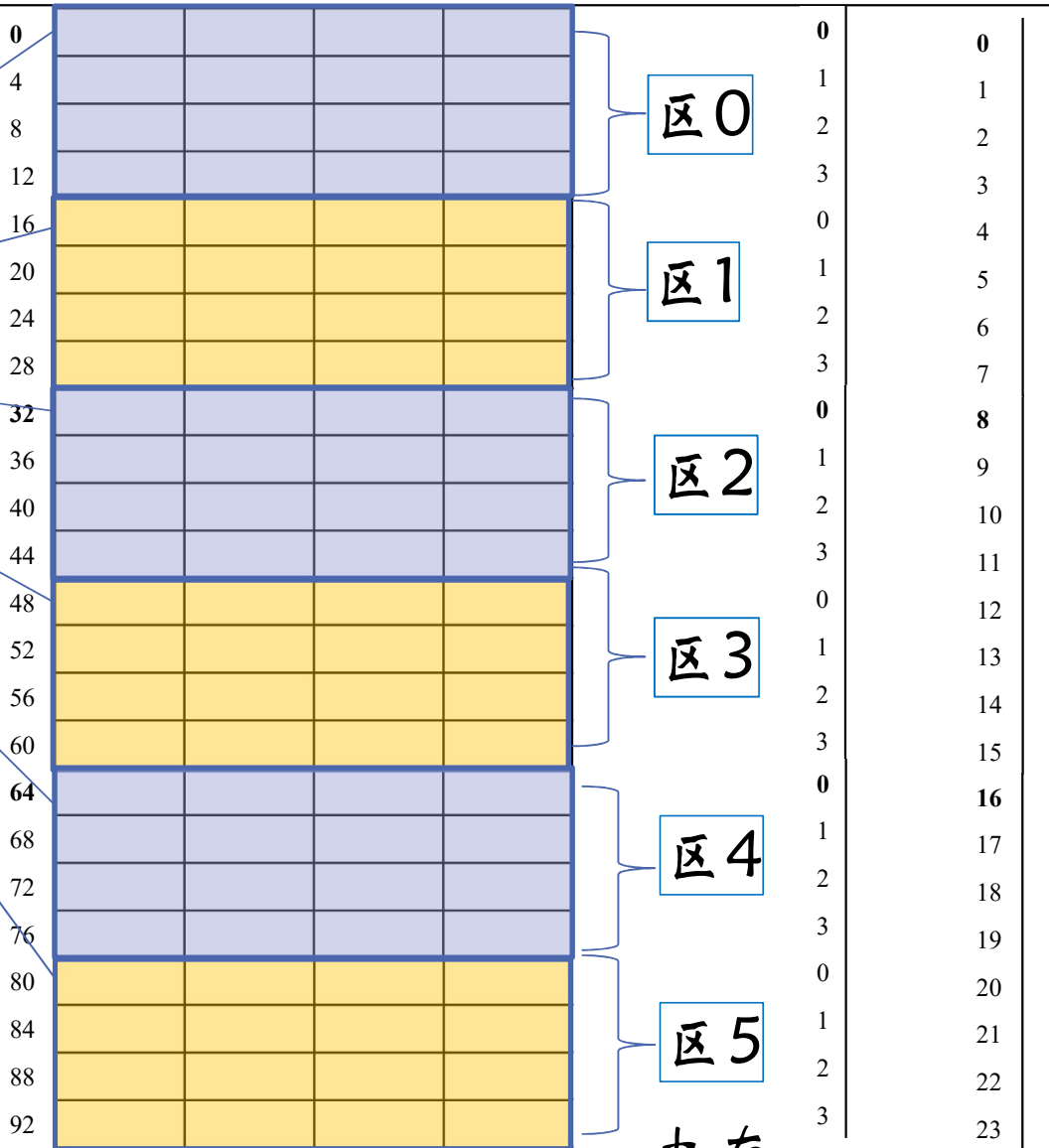


cache

与直接映射类似

内存物理地址

区号	组号(区内行号)	行内偏移
3b	2b	2b



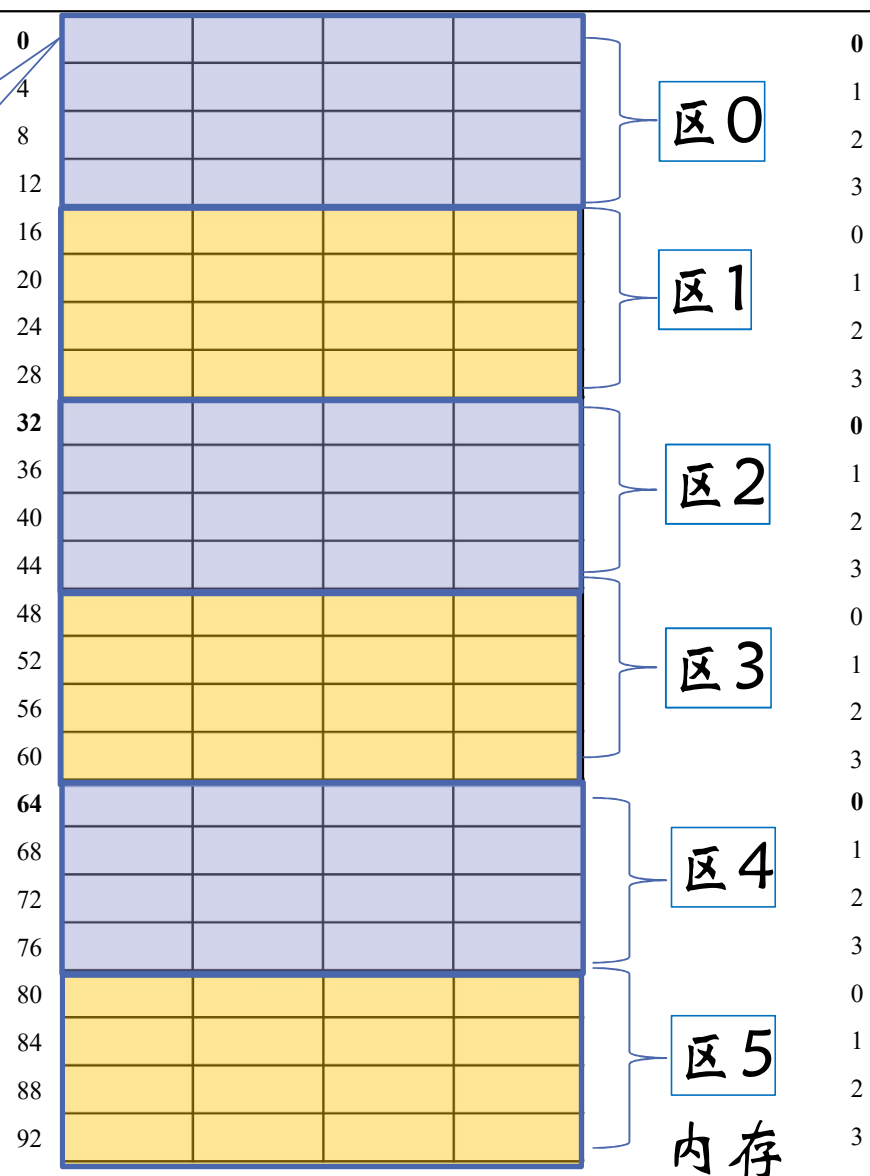
内存

组相联映射-n way set association

0	0				
1	1				
2	2				
3	3				
0	4				
1	5				
2	6				
3	7				

cache

组内全相联 $n=2$



组相联映射策略

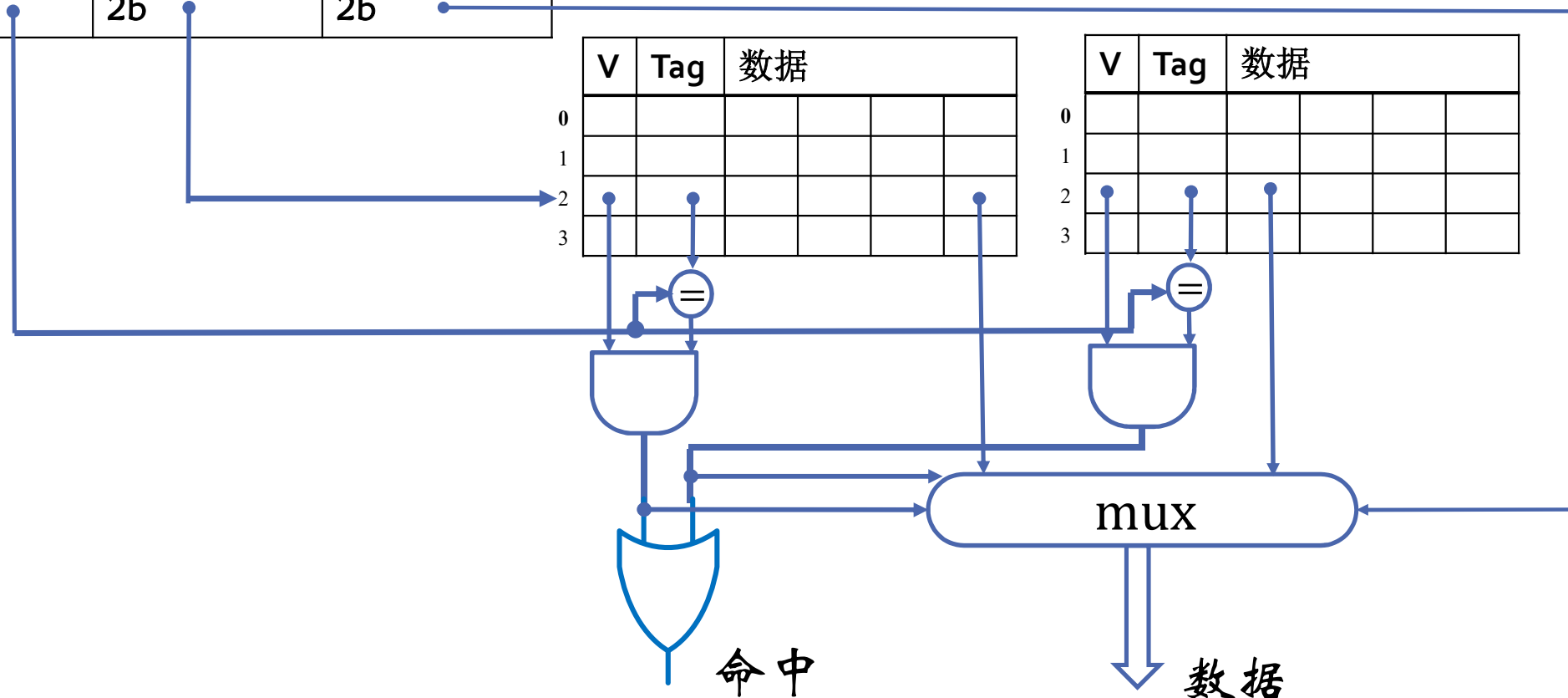
内存物理地址

块号	区内行号	行内偏移
3b	2b	2b

缓存一行4字节，共8行，2路组相联

	V	Tag	数据			
0						
1						
2						
3						

	V	Tag	数据			
0						
1						
2						
3						



组相联映射

内存物理地址

块号	区内行号	行内偏移
3b	2b	2b

缓存一行4字节，共8行，2路组相联

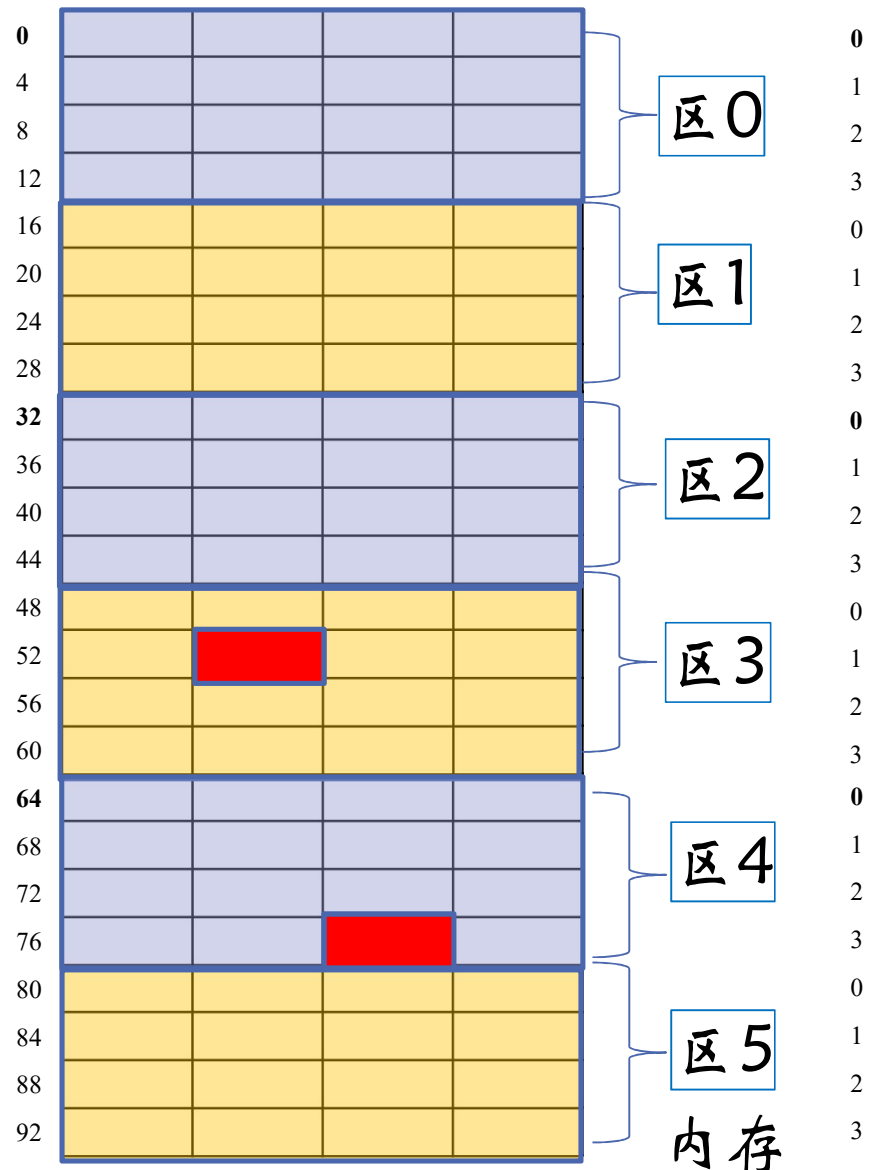
内存行号: $\frac{53}{4}=13$ $\frac{\text{内存地址}}{\text{行大小}}$

行内偏移: $53 \% 4 = 1$ 内存地址 $\%$ 行大小

对应缓存组号: $13\%4=1$

内存行号%缓存区内行数

$$\text{内存区号: } \frac{13}{4} = 3 \quad \frac{\text{内存行号}}{\text{缓存区内行数}}$$

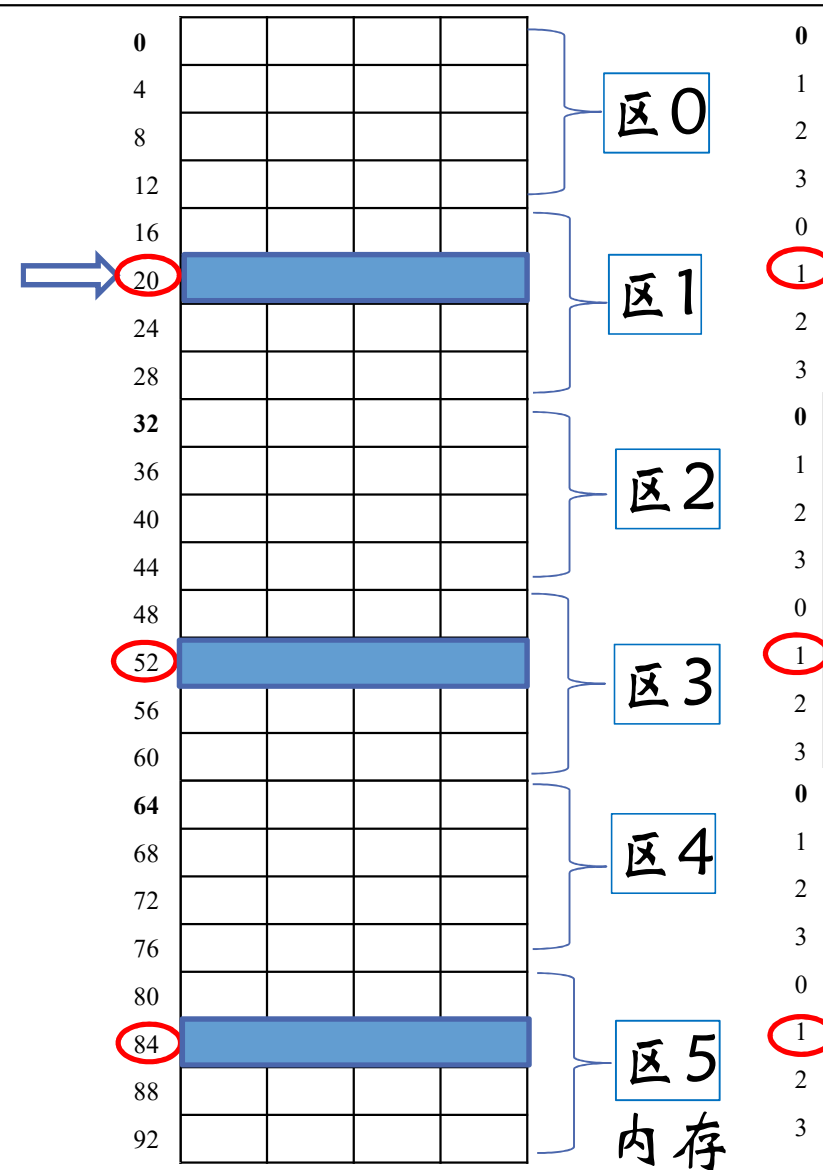


2路组相联映射机制

v Tag		v Tag	
0		0	
1	1 101	1	1 001
2		2	
3		3	

cache

Cache命中率有所提高



组相联映射

处理器 | **缓存** | 主板 | 内存 | SPD | 显卡 | 测试分数 | 关于

一级数据缓存

大小: 32 KBytes **x 4**

描述: 8-way set associative, 64-byte line size

一级指令缓存

大小: 32 KBytes **x 4**

描述: 8-way set associative, 64-byte line size

二级缓存

大小: 256 KBytes **x 4**

描述: 4-way set associative, 64-byte line size

三级缓存

大小: 8 MBytes

描述: 16-way set associative, 64-byte line size

处理器 | 缓存 | 主板 | **内存** | SPD | 显卡 | 测试分数 | 关于

常规

类型: DDR4

大小: 8 GBytes

通道数: 单通道

DC模式:

北桥频率: 1297.1 MHz

4核

Cache行结构（各个域的位宽）

v	Tag	数据
1	33-6-6	64×8

寻址8G内存的物理地址如何划分
33b

区号	组号（区内行号）	行内偏移（字节）
33-6-6	$\log_2 64$	$\log_2 64$

行大小: 64B 一级Cache大小: 32KB

一级Cache分区数（路数）: 8

一级Cache总行数: $32\text{KB}/64\text{B}=512$

一级Cache各区行数: 总行数/分区数= $512/8=64$

Cache映射策略总结

直接映射

	V	Tag	数据
0			
1			
2			
3			
4			
5			
6			
7			

一路组相联映射

V	Tag	数据	V	Tag	数据	V	Tag	数据	V	Tag	数据	V	Tag	数据	V	Tag	数据

8路组相联映射

全相联映射

	V	Tag	数据	V	Tag	数据
0						
1						
2						
3						

2路组相联映射

	V	Tag	数据	V	Tag	数据	V	Tag	数据	V	Tag	数据
0												
1												

4路组相联映射

小结

- 三种cache映射策略
 - Cache行结构
 - Cache控制器构成
 - 内存物理地址划分

下一讲：Cache管理策略