

# 王道考研——计算机组成原理

WWW.CSKAOYAN.COM

听完直播做王道 3.5、3.6 课后真题(大题)

强化直播P2: 存储系统大题总结

关注公众号: 研途小时 获取后续课程完整更新



## 强化阶段（二轮）复习策略

### 核心策略——抓住主要矛盾和矛盾的主要方面

第一轮复习：主要矛盾——过线，打基础

主攻小题，打好基础，对课程形成较完整的认知

第二轮复习：主要矛盾——迅速提分，强化考试重点

把握考试重点，对命题重点范围进行训练

第三轮复习：主要矛盾——把握做题节奏，查缺补漏

做模拟题、真题。  
全面查缺补漏

Tips: 很少有人能把王道书所有题目一题不漏的做完做透，因此要有选择地训练。做题在精不在多



# 考试形式和试卷结构

## 考试形式和试卷结构

一、试卷满分及考试时间 —— 本试卷满分为 150 分、考试时间为 180 分钟

二、答题方式 —— 答题方式为闭卷、笔试

### 三、试卷内容结构

数据结构 45 分 —— 11道小题22分，2道大题23分

计算机组成原理 45 分 —— 11道小题22分，2道大题23分

操作系统 35 分 —— 10道小题20分，2道大题15分

计算机网络 25 分 —— 8道小题16分，1道大题9分

### 四、试卷题型结构

单项选择题（共80分，40小题，每小题2分）

综合应用题（共70分）



# 408真题构成

一、单项选择题：第 1~40 小题，每小题 2 分，共 80 分。下列每题给出的四个选项中，只有一个选项最符合试题要求。

1~11 ——数据结构 (22)

12~22 ——计组 (22)

23~32 ——操作系统 (20)

33~40 ——计网 (16)

二、综合应用题：第 41~47 小题，共 70 分。

数据结构 (23分)

计组 (23分)

操作系统 (15分)

计网 (9分)



2009	2010	2011	2012	2013	2014	2015	2016
中断控制方式的处理过程；DMA控制方式的处理过程。	指令格式；寻址范围；指令执行的微操作过程	C语言中常见变量的表示；强制类型转换；补码加法的应用；溢出判断。	CPU性能指标的计算；引入Cache后的访存原理；虚拟内存的工作原理；DMA控制方式的工作原理；低位交叉存储的流水线	CPU性能指标、总线性能指标的计算；低位交叉存储方式；总线的突发传送过程；引入Cache后的访存原理；	结合C语言，读懂各条指令的作用；条件转移指令的工作原理；五段式指令流水线	指令执行的数据通路；数据通路上各种常见的硬件部件及作用；控制信号连线；	中断控制方式的处理过程；
指令执行的详细过程；说明每一步的微操作、微命令。并安排合理的时序	Cache和主存的映射；C语言二维数组的存储原理；Cache命中率的计算	虚拟存储系统的地址结构；Cache的工作原理；TLB的工作原理；	数据的移位运算；五段式指令流水线；流水线的“冲突”原因；	条件转移指令的工作原理；CPU内部常见的硬件部件（根据处理逻辑推测）	Cache的工作原理；指令的溢出判断；虚拟存储，缺页异常的产生原因；TLB的工作原理；	指令格式；各步微操作对应的微命令；微操作的时序安排；	TLB的工作原理；Cache的工作原理；虚拟存储，缺页异常；Cache淘汰策略、页面淘汰处理；
2017	2018	2019	2020	2021	2022	2023	2024
C语言强制类型转换；各种数的精度问题、溢出问题	程序定时查询方式的工作过程；中断查询方式的工作过程；DMA方式的工作过程；	C语言对应的指令序列；条件转移指令、无条件转移指令、函数调用call指令的原理；数据的精度、溢出问题	数据的运算：二进制乘法；溢出问题	指令格式；数据的运算、溢出问题；	一条指令的执行过程；指令执行的电路数据通路原理	Cache、虚拟页式存储	一条指令的执行过程；指令执行的电路数据通路原理
C语言对应的指令序列；比较指令cmp、条件转移指令的工作原理；数据的运算，算数左移	虚拟存储，地址结构；TLB的工作原理；Cache的工作原理；有TLB、Cache的地址变换过程	虚拟分页存储；Cache的工作原理；	Cache的工作原理；结合C语言分析Cache命中情况；	虚拟存储，地址结构；TLB的工作原理；	磁盘+IO控制方式	结合C语言，分析指令序列的工作原理	结合C语言，分析指令序列的工作原理



• 图示说明:

主考第二章

数据的运算; 强制类型转换; 精度、溢出问题

3+n

主考第三章

Cache、TLB、虚拟分页

9

主考第五章

一条指令的执行过程

7

主考第四章

指令序列的工作过程

6

主考第七章

三种IO控制方式

4

杂七杂八

3







# Cache 一类 $\Rightarrow$ 地址结构

$2^n$  行  $\Rightarrow$  行号  $n$  bit

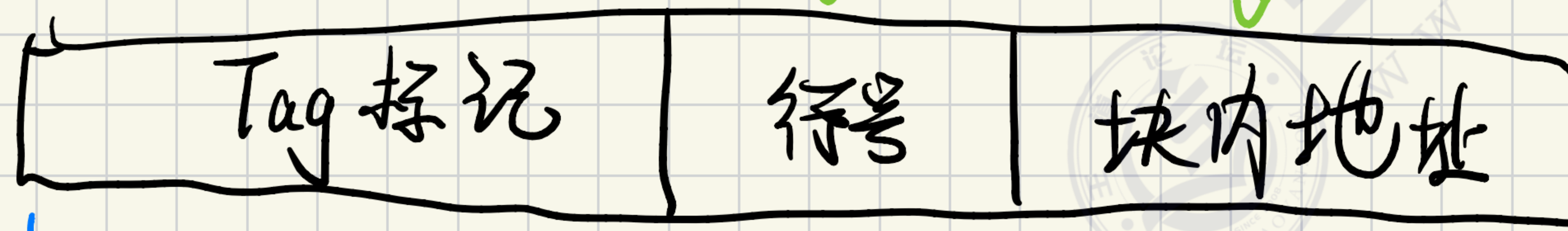
关注条件:

- ① PA 位数
- ② Cache 总行数
- ③ Cache 块大小
- ④ 映射方式 { 直组全  $\rightarrow$  关注“路数”

Cache 总大小/总容量  
包含: Tag, 有效位, ...

Cache 数据区 总大小

④ 直接映射

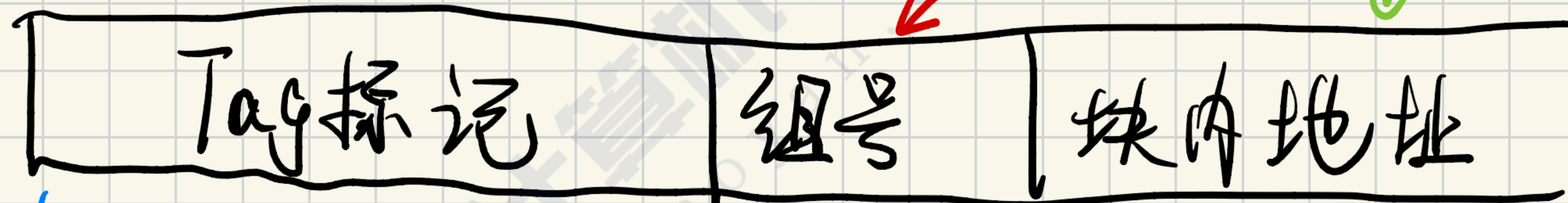


②  $\downarrow$

③  $\downarrow$

① 分“几组” =  $\frac{\text{总行数}}{\text{路数}}$  ③

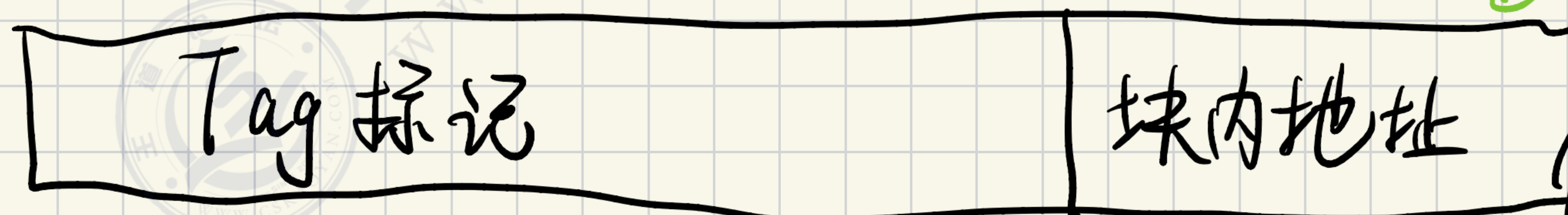
④ 组相联



①

③

④ 全相联



①



# Cache 行完整构成:

关注条件:

① PA 位数

② Cache 总行数

③ Cache 块大小

④ 映射方式

直组全 → 关注“路数”

⑤ 写策略

直写法 → 不需要脏位  
回写法 → 1bit 脏位

⑥ 替换算法

随机替换  
其它

Cache 总大小/总容量  
↓ 包含: Tag, 有效位

Cache 数据区 总大小

Tag

① ② ③ ④

一定存在

有效位

一定存在 1bit

块内数据

③

一定存在

脏位

⑤

写回法

1bit

直写法

0bit

替换信息位

⑥

随机替换 0bit  
其他 FIFO, LRU

④

行号/组号 (隐含)

不考虑

直接 → 0bit  
组相联 →  $\log_2$  路数 bit  
全相联 →  $\log_2$  行数 bit

②



# TLB 需关注:

① VA 多少位, 虚页号多少位?

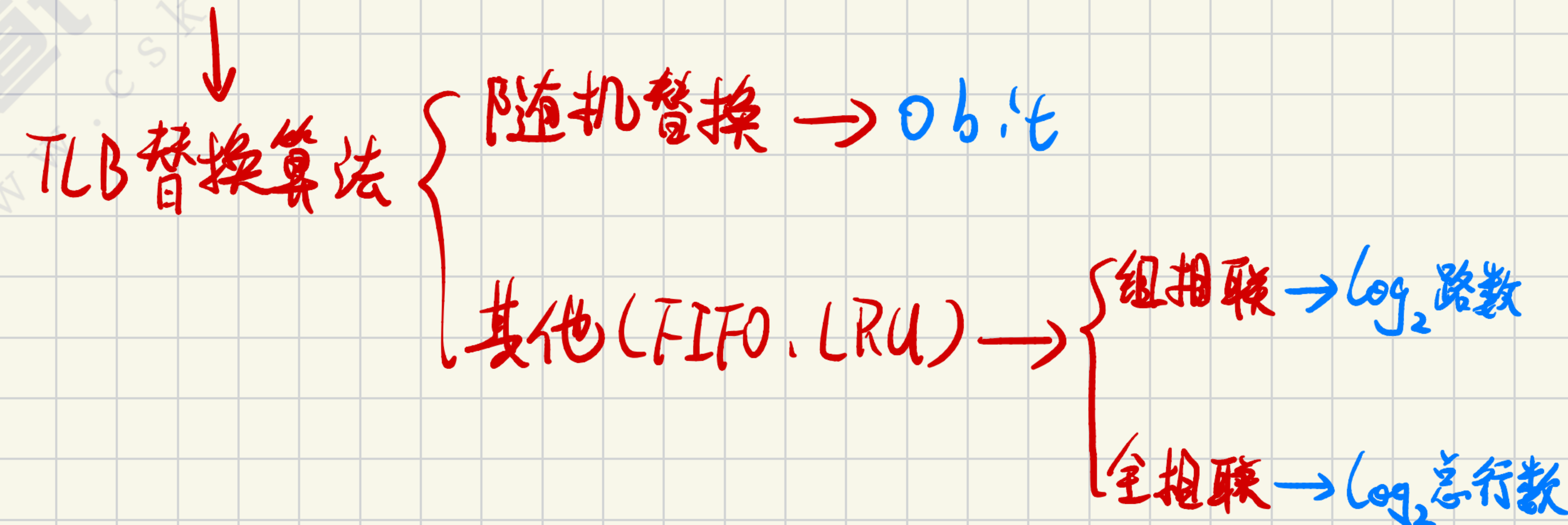
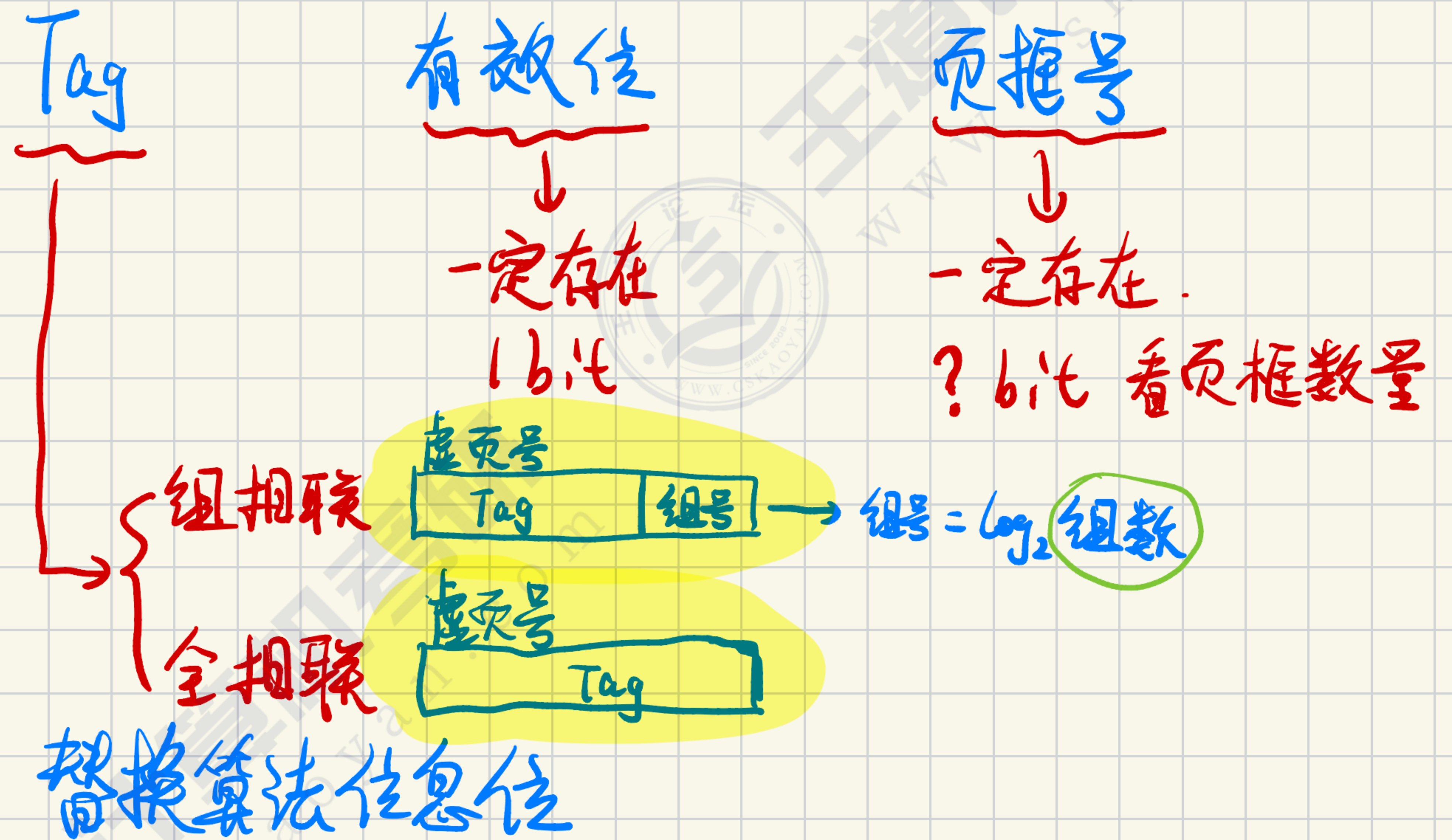
↓  
页号, 页内地址

② 映射方式

- 可能 TLB
- 直接映射 (抖动, 效率低) X
- 组相联 → K 路
- 全相联

③ 替换算法 (同 Cache)

每个 TLB 行 (表项) 完整构成:





分界线

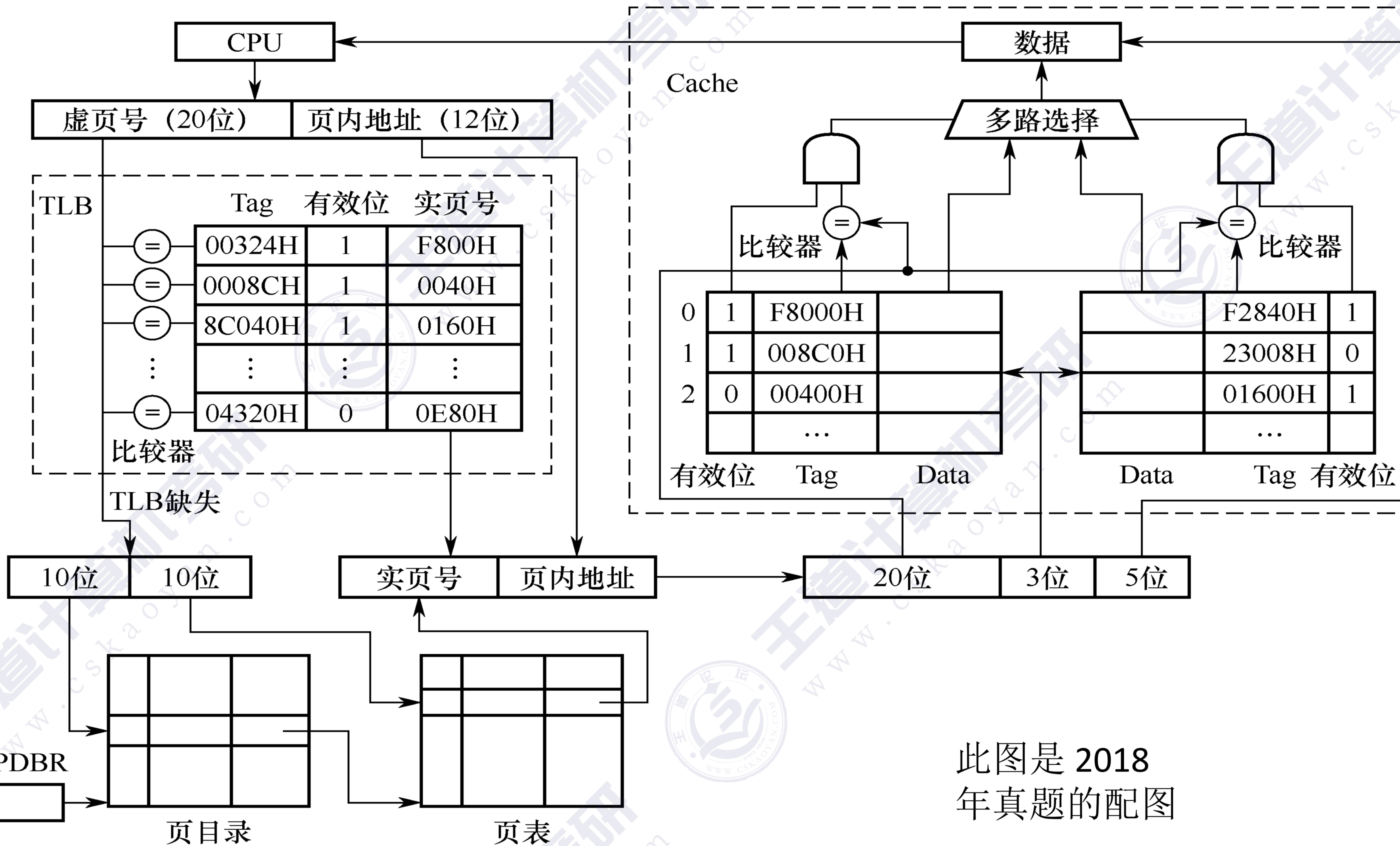
# 补充：存储系统硬件图示

咸鱼注：多看存储系统的各种图示画法，预防题目里套上新马甲

关注公众号：研途小时 获取后续课程完整更新

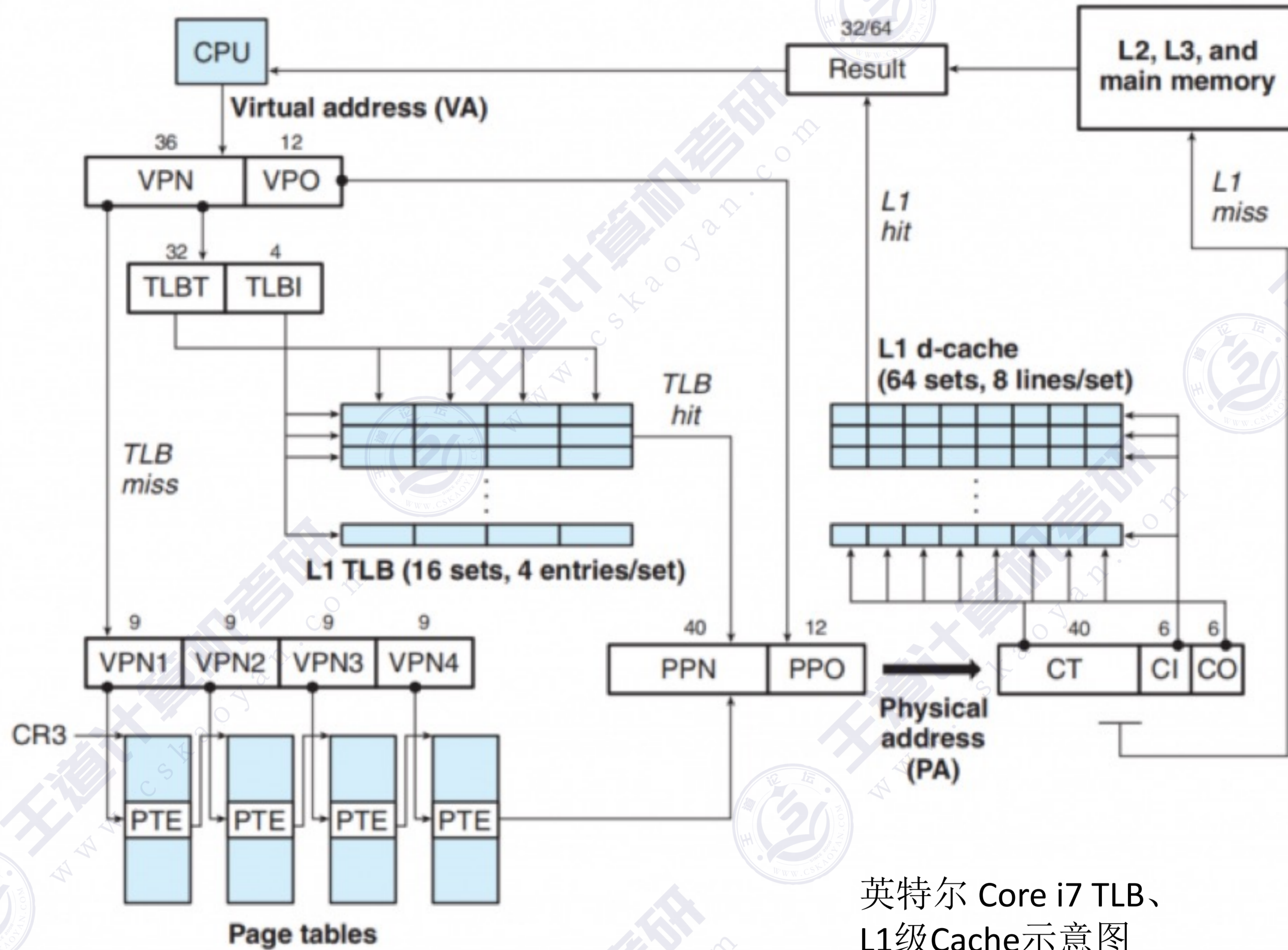


# 全相联映射TLB+2路组相联映射Cache



此图是 2018 年真题的配图





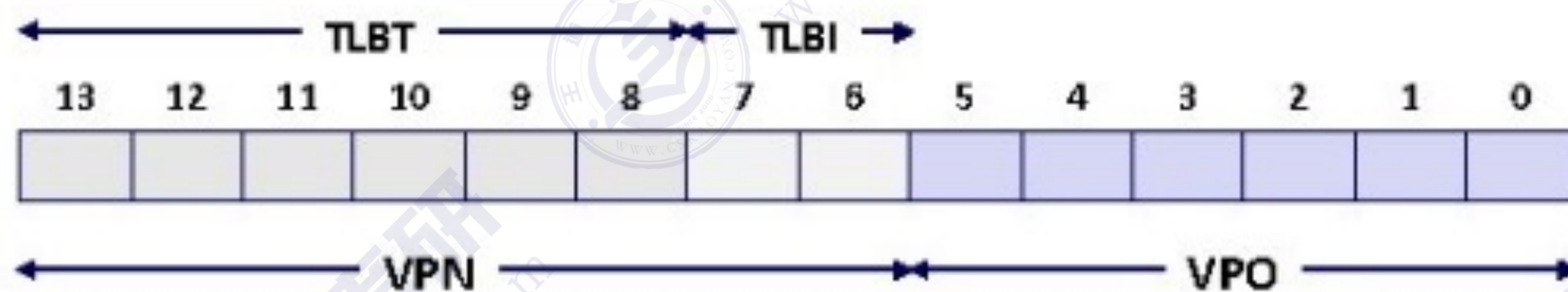
英特尔 Core i7 TLB、L1级Cache示意图



# TLB 4路组相连映射图示

## Simple Memory System TLB

- 16 entries
- 4-way associative

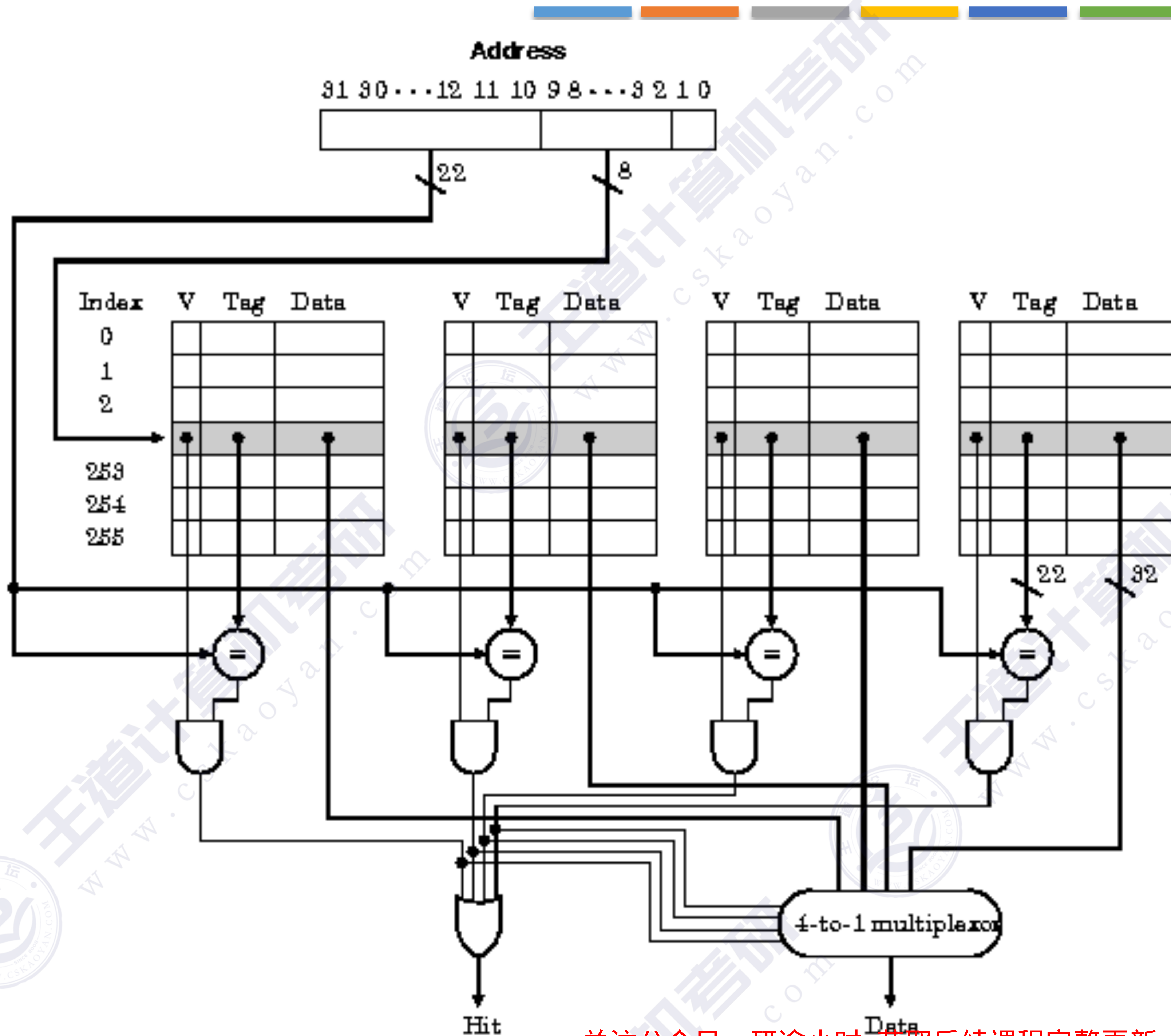


此图的画法中，着重突出了4路组相联TLB的内容，并没有给出硬件结构的示意图

Set	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid	Tag	PPN	Valid
0	03	-	0	09	0D	1	00	-	0	07	02	1
1	03	2D	1	02	-	0	04	-	0	0A	-	0
2	02	-	0	08	-	0	06	-	0	03	-	0
3	07	-	0	03	0D	1	0A	34	1	02	-	0

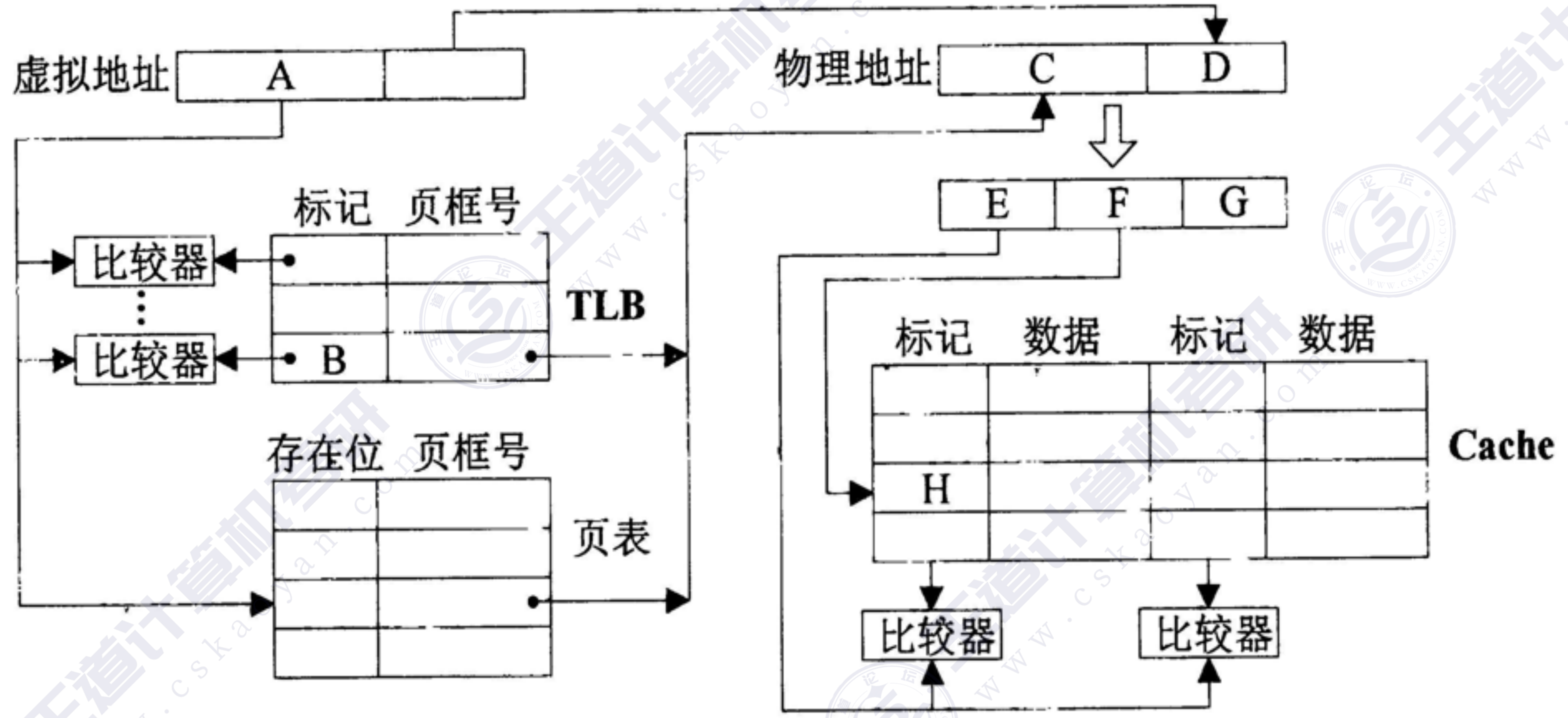


# 4路组相连映射图示





# TLB 全连映射图示

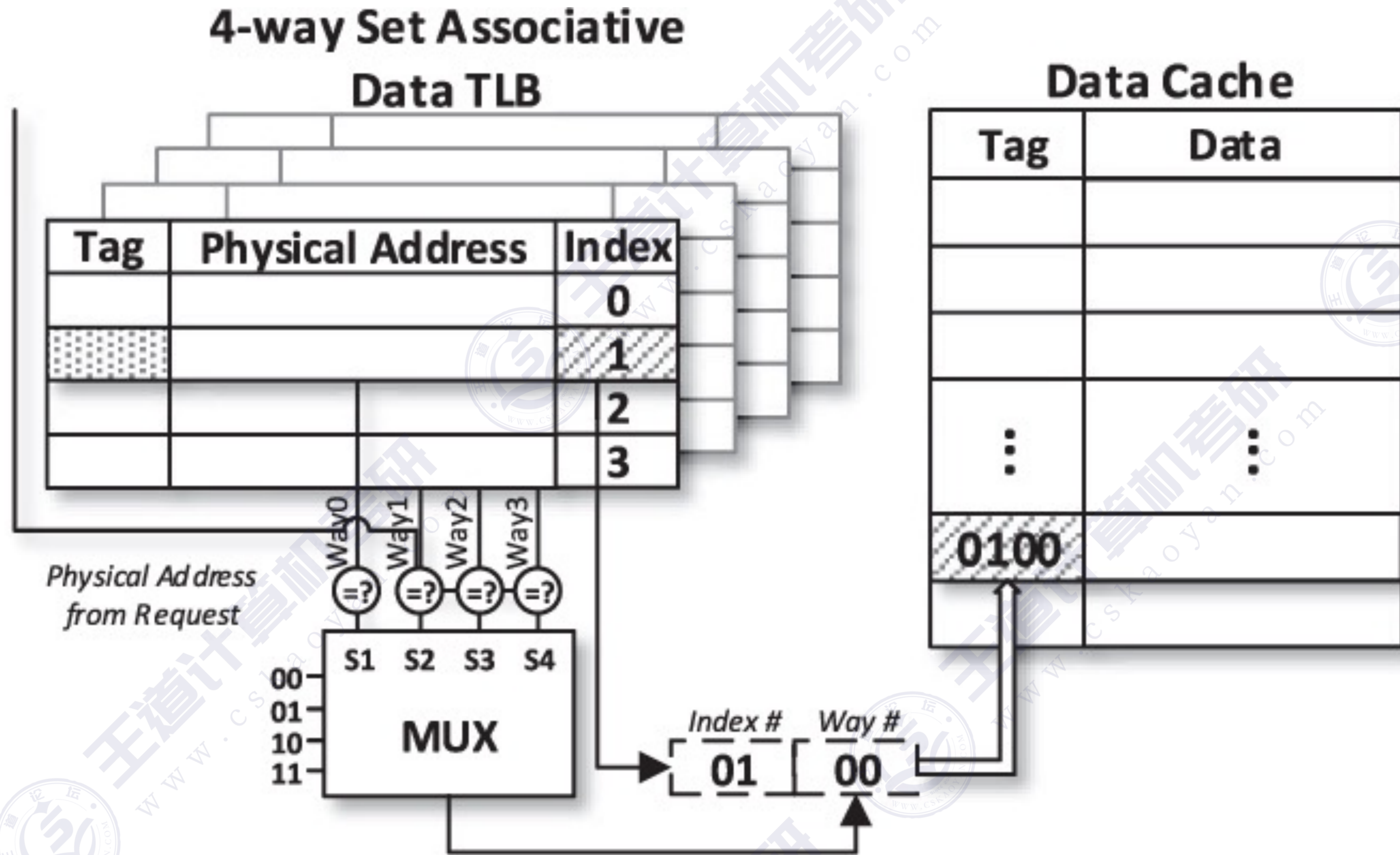


Cache 采用二路组相联映射

2016真题45题



# TLB 4路组相连映射图示





分 界 线

# 存储系统相关的历年真题



## 2023年真题

43. (14分) 已知计算机 M 字长为 32 位, 按字节编址, 采用请求调页策略的虚拟存储管理方式, 虚拟地址为 32 位, 页大小为 4 KB; 数据 Cache 采用 4 路组相联映射方式, 数据区大小为 8KB, 主存块大小为 32B。现有 C 语言程序段如下:

```
int a[24][64];  
.....  
for ( i = 0; i < 24; i++ )  
    for ( j = 0; j < 64; j++ ) a[i][j] = 10;
```

已知二维数组 a 按行优先存放, 在虚拟地址空间中分配的起始地址为 0042 2000H,  $\text{sizeof}(\text{int}) = 4$ , 假定在 M 上执行上述程序段之前数组 a 不在主存, 且在该程序段执行过程中不会发生页面置换。请回答下列问题。



## 2023年真题

- 1) 数组  $a$  分布在几个页面中? 对于数组  $a$  的访问, 会发生几次缺页异常? 页故障地址各是什么?
- 2) 不考虑变量  $i$  和  $j$ , 该程序段的数据访问是否具有时间局部性? 为什么?
- 3) 计算机  $M$  的虚拟地址 ( $A_{31} \sim A_0$ ) 中哪几位用作块内地址? 哪几位用作 Cache 组号?  $a[1][0]$  的虚拟地址是多少? 其所在主存块对应的 Cache 组号是多少?
- 4) 数组  $a$  占用多少主存块? 假设上述程序段执行过程中数组  $a$  的访问不会和其他数据发生 Cache 访问冲突, 则数组  $a$  的 Cache 命中率是多少? 若将循环中  $i$  和  $j$  的次序按如下方式调换:

```
for (j = 0; j < 64; j++)
```

```
    for (i = 0; i < 24; i++) a[i][j] = 10;
```

则数组  $a$  的 Cache 命中率又是多少?



## 2021年真题

44. (8分) 假设计算机 M 的主存地址为 24 位，按字节编址；采用分页存储管理方式，虚拟地址为 30 位，页大小为 4 KB；TLB 采用 2 路组相联方式和 LRU 替换策略，共 8 组。请回答下列问题。
- 1) 虚拟地址中哪几位表示虚页号？哪几位表示页内地址？
  - 2) 已知访问 TLB 时虚页号高位部分用作 TLB 标记，低位部分用作 TLB 组号，M 的虚拟地址中哪几位是 TLB 标记？哪几位是 TLB 组号？
  - 3) 假设 TLB 初始时空，访问的虚页号依次为 10, 12, 16, 7, 26, 4, 12 和 20，在此过程中，哪一个虚页号对应的 TLB 表项被替换？说明理由。
  - 4) 若将 M 中的虚拟地址位数增加到 32 位，则 TLB 表项的位数增加几位？



## 2020年真题

44. (10分) 假定主存地址为32位, 按字节编址, 指令Cache和数据Cache与主存之间均采用8路组相联映射方式, 直写(Write Through)写策略和LRU替换算法, 主存块大小为64B, 数据区容量各为32KB。开始时Cache均为空。请回答下列问题。

(1) Cache每一行中标记(Tag)、LRU位各占几位? 是否有修改位?

(2) 有如下C语言程序段:

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

若数组s及其变量k均为int型, int型数据占4B, 变量k分配在寄存器中, 数组s在主存中的起始地址为0080 00C0H, 则该程序段执行过程中, 访问数组s的数据Cache缺失次数为多少?

(3) 若CPU最先开始的访问操作是读取主存单元0001 003H中的指令, 简要说明从Cache中访问该指令的过程, 包括Cache缺失处理过程。



46. (7分) 对于题 45, 若计算机 M 的主存地址为 32 位, 采用分页存储管理方式, 页大小为 4KB, 则第 1 行的 push 指令和第 30 行的 ret 指令是否在同一页中(说明理由)? 若指令 Cache 有 64 行, 采用 4 路组相联映射方式, 主存块大小为 64B, 则 32 位主存地址中, 哪几位表示块内地址? 哪几位表示 Cache 组号? 哪几位表示标记 (tag) 信息? 读取第 16 行的 call 指令时, 只可能在指令 Cache 的哪一组中命中 (说明理由)?



(19年45题是46题的前置条件)

## 2019年真题

45. (16分) 已知  $f(n) = n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$ , 计算  $f(n)$  的 C 语言函数 f1 的源程序 (阴影部分) 及其在 32 位计算机 M 上的部分机器级代码如下:

int	f1(int n){	
1	00401000 55	push ebp
...	...	...
	if(n>1)	
11	00401018 83 7D 08 01	cmp dword ptr [ebp+8],1
12	0040101C 7E 17	jle f1+35h (00401035)
	return n*f1(n-1);	
13	0040101E 8B 45 08	mov eax, dword ptr [ebp+8]
14	00401021 83 E8 01	sub eax, 1
15	00401024 50	push eax
16	00401025 E8 D6 FF FF FF	call f1 ( 00401000)
...	...	...
19	00401030 0F AF C1	imul eax, ecx
20	00401033 EB 05	jmp f1+3Ah (0040103a)
	else return 1;	
21	00401035 B8 01 00 00 00	mov eax,1
	}	
...	...	...
26	00401040 3B EC	cmp ebp, esp
...	...	...
30	0040104A C3	ret

其中, 机器级代码行包括行号、虚拟地址、机器指令和汇编指令, 计算机 M 按字节编址, int 型数据占 32 位。请回答下列问题:



44. (15分)某计算机采用页式虚拟存储管理方式,按字节编址。CPU进行存储访问的过程如题44图所示。

(1) 主存物理地址占多少位?

(2) TLB采用什么映射方式? TLB用SRAM还是DRAM实现?

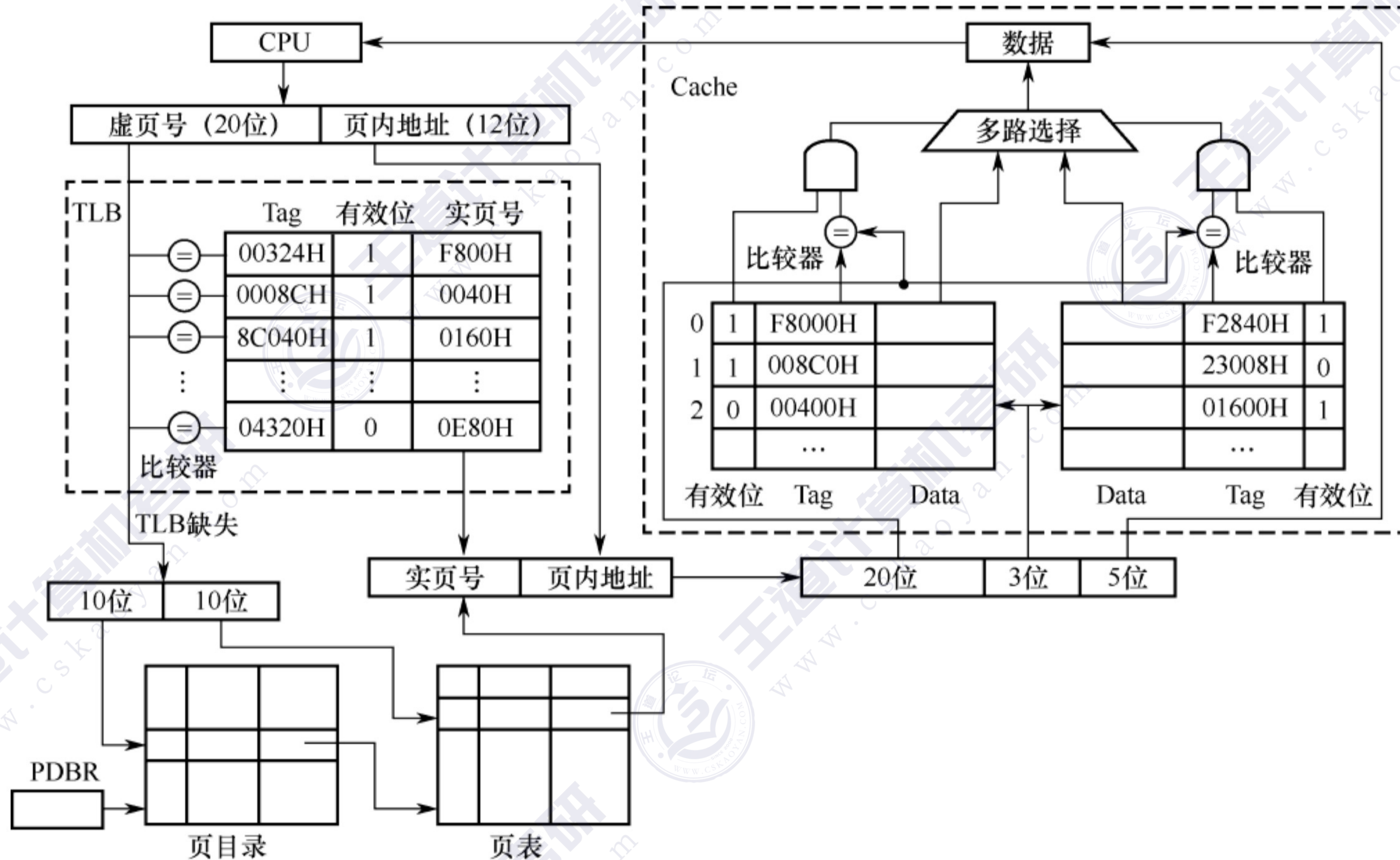
(3) Cache采用什么映射方式?若Cache采用LRU替换算法和回写(Write Back)策略,则Cache每行中除数据(Data)、Tag和有效位外,还应有哪些附加位?Cache总容量是多少?Cache中有效位的作用是什么?

(4)若CPU给出的虚拟地址为0008C040H,则对应的物理地址是多少?是否在Cache中命中?说明理由,若CPU给出的虚拟地址为0007C260H,则该地址所在主存块映射到的Cache组号是多少?



(18年44题为计组题, 考TLB、Cache)

# 2018年真题 (计组)





(18年45题为操作系统题，  
基于44题条件考请求页式存储)

## 2018年真题（操作系统）

45. (8分)请根据题44图给出的虚拟存储管理方式，回答下列问题。

(1) 某虚拟地址对应的页目录号为6，在相应的页表中对应的页号为6，页内偏移量为8，该虚拟地址的十六进制表示是什么？

(2) 寄存器PDBR用于保存当前进程的页目录起始地址，该地址是物理地址还是虚拟地址？进程切换时，PDBR的内容是否会变化？说明理由。同一进程的线程切换时，PDBR的内容是否会变化？说明理由。

(3) 为了支持改进型CLOCK置换算法，需要在页表项中设置哪些字段？



王道计算机考研  
WWW.CSKAOYAN.COM

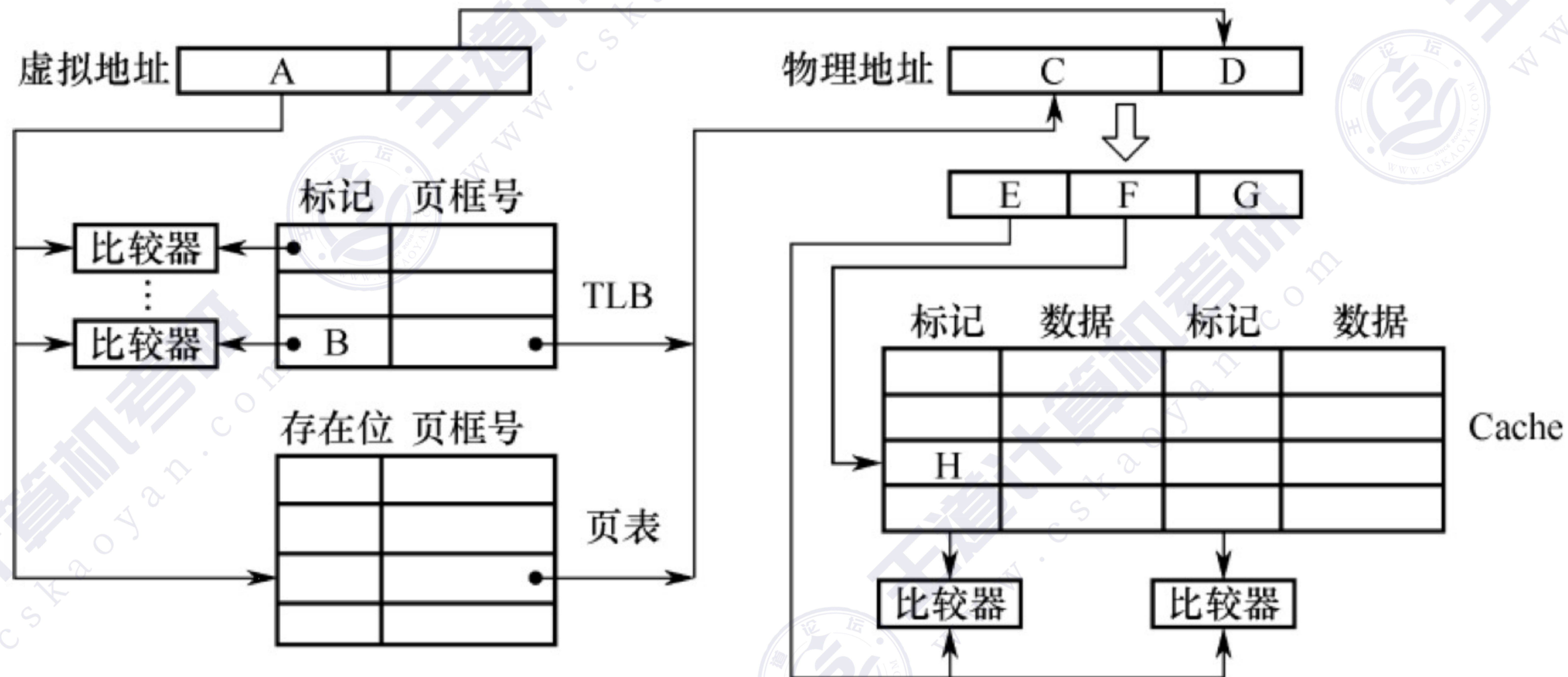


王道计算机考研  
WWW.CSKAOYAN.COM



# 2016年真题

45. (14分) 某计算机采用页式虚拟存储管理方式，按字节编址，虚拟地址为32位，物理地址为24位，页大小为8KB；TLB采用全相联映射；Cache数据区大小为64KB，按2路组相联方式组织，主存块大小为64B。存储访问过程的示意图如下。





## 2016年真题

请回答下列问题。

- 1) 图中字段 A~G 的位数各是多少? TLB 标记字段 B 中存放的是什么信息?
- 2) 将块号为 4099 的主存块装入到 Cache 中时, 所映射的 Cache 组号是多少? 对应的 H 字段内容是什么?
- 3) Cache 缺失处理的时间开销大还是缺页处理的时间开销大? 为什么?
- 4) 为何 Cache 可采用直写 (Write Through) 策略, 而修改页面内容时总是采用回写 (Write Back) 策略。



44. (12分) 某程序中有如下循环代码段 P: “for(int i = 0; i < N; i++) sum+=A[i];”。假设编译时变量 sum 和 i 分别分配在寄存器 R1 和 R2 中。常量 N 在寄存器 R6 中, 数组 A 的首地址在寄存器 R3 中。程序段 P 起始地址为 0804 8100H, 对应的汇编代码和机器代码如下表所示。

编号	地址	机器代码	汇编代码	注释
1	08048100H	00022080H	loop: sll R4, R2, 2	(R2)<<2 → R4
2	08048104H	00083020H	add R4, R4, R3	(R4) + (R3) → R4
3	08048108H	8C850000H	load R5, 0(R4)	((R4) + 0) → R5
4	0804810CH	00250820H	add R1, R1, R5	(R1) + (R5) → R1
5	08048110H	20420001H	add R2, R2, 1	(R2) + 1 → R2
6	08048114H	1446FFFAH	bne R2, R6, loop	if(R2)!=(R6) goto loop

执行上述代码的计算机 M 采用 32 位定长指令字, 其中分支指令 bne 采用如下格式:

31	26	25	21	20	16	15	0
OP		Rs		Rd		OFFSET	

OP 为操作码; Rs 和 Rd 为寄存器编号; OFFSET 为偏移量, 用补码表示。请回答下列问题, 并说明理由。



45. 假设对于44题中的计算机M和程序P的机器代码, M采用页式虚拟存储管理; P开始执行时,  $(R1) = (R2) = 0$ ,  $(R6) = 1000$ , 其机器代码已调入主存但不在Cache中; 数组A未调入主存, 且所有数组元素在同一页, 并存储在磁盘同一个扇区。请回答下列问题并说明理由。

1) P执行结束时, R2的内容是多少?

2) M的指令Cache和数据Cache分离。若指令Cache共有16行, Cache和主存交换的块大小为32字节, 则其数据区的容量是多少? 若仅考虑程序段P的执行, 则指令Cache的命中率为多少?

3) P在执行过程中, 哪条指令的执行可能发生溢出异常? 哪条指令的执行可能产生缺页异常? 对于数组A的访问, 需要读磁盘和TLB至少各多少次?



# 2011年真题

44. (12分) 某计算机存储器按字节编址, 虚拟(逻辑)地址空间大小为16MB, 主存(物理)地址空间大小为1MB, 页面大小为4KB; Cache采用直接映射方式, 共8行; 主存与Cache之间交换的块大小为32B。系统运行到某一时刻时, 页表的部分内容和Cache的部分内容分别如题44-a图、题44-b图所示, 图中页框号及标记字段的内容为十六进制形式。

虚页号	有效位	页框号	...
0	1	06	...
1	1	04	...
2	1	15	...
3	1	02	...
4	0	—	...
5	1	2B	...
6	0	—	...
7	1	32	...

题44-a图 页表的部分内容

行号	有效位	标记	...
0	1	020	...
1	0	—	...
2	1	01D	...
3	1	105	...
4	1	064	...
5	1	14D	...
6	0	—	...
7	1	27A	...

题44-b图 Cache的部分内容



# 2011年真题

请回答下列问题。

- 1) 虚拟地址共有几位，哪几位表示虚页号？物理地址共有几位，哪几位表示页框号（物理页号）？
- 2) 使用物理地址访问 Cache 时，物理地址应划分成哪几个字段？要求说明每个字段的位数及在物理地址中的位置。
- 3) 虚拟地址 001C60H 所在的页面是否在主存中？若在主存中，则该虚拟地址对应的物理地址是什么？访问该地址时是否 Cache 命中？要求说明理由。
- 4) 假定为该机配置一个 4 路组相联的 TLB，该 TLB 共可存放 8 个页表项，若其当前内容（十六进制）如题 44-c 图所示，则此时虚拟地址 024BACH 所在的页面是否在主存中？要求说明理由。

组号	有效位	标记	页框号	有效位	标记	页框号	有效位	标记	页框号	有效位	标记	页框号
0	0	—	—	1	001	15	0	—	—	1	012	1F
1	1	013	2D	0	—	—	1	008	7E	0	—	—

题 44-c 图 TLB 的部分内容



## 2010年真题

44. (12分) 某计算机的主存地址空间大小为 256MB，按字节编址。指令 Cache 和数据 Cache 分离，均有 8 个 Cache 行，每个 Cache 行大小为 64B，数据 Cache 采用直接映射方式。现有两个功能相同的程序 A 和 B，其伪代码如下所示：

程序 A:

```
int a[256][256]
...
int sum_array1()
{
    int i, j, sum=0;
    for(i=0; i<256; i++)
        for(j=0; j<256; j++)
            sum+=a[i][j];
    return sum;
}
```

程序 B:

```
int a[256][256]
...
int sum_array2()
{
    int i, j, sum=0;
    for(j=0; j<256; j++)
        for(i=0; i<256; i++)
            sum+=a[i][j];
    return sum;
}
```



## 2010年真题

假定 `int` 类型数据用 32 位补码表示，程序编译时 `i`、`j`、`sum` 均分配在寄存器中，数组 `a` 按行优先方式存放，其首地址为 320（十进制数）。回答下列问题，要求说明理由或给出计算过程。

- 1) 若不考虑用于 Cache 一致性维护和替换算法的控制位，则数据 Cache 的总容量为多少？
- 2) 数组元素 `a[0][31]` 和 `a[1][1]` 各自所在的主存块对应的 Cache 行号分别是多少（Cache 行号从 0 开始）？
- 3) 程序 A 和 B 的数据访问命中率各是多少？哪个程序的执行时间更短？