

5. 假定用  $8K \times 8$  位的 EPROM 芯片组成  $32K \times 16$  位的只读存储器, 要求回答以下问题。

- (1) 数据寄存器最少应有多少位?
- (2) 地址寄存器最少应有多少位?
- (3) 共需多少个 EPROM 芯片?

- (1) 数据寄存器最少有16位
- (2) 地址寄存器15位 (按16位编址), 16位 (按照8位(1字节)编址)
- (3)  $(32/8) * (16/8) = 8$ 片

7. 假定一个存储器系统支持四体交叉存取, 某程序执行过程中访问地址序列为 3, 9, 17, 2, 51, 37, 13, 4, 8, 41, 67, 10, 则哪些地址访问会发生体冲突?

位差在4以内且在模4同余下相等则会冲突:

故而: 9和17, 17和37, 37和13, 4和8

8. 假定一个程序重复完成将磁盘上一个 4KB 的数据块读出, 进行相应处理后, 写回到磁盘的另外一个数据区。各数据块内信息在磁盘上连续存放, 并随机地置于磁盘的一个磁道上。磁盘转速为 7200RPM, 平均寻道时间为 10ms, 磁盘最大数据传输率为 40MB/s, 磁盘控制器的开销为 2ms, 没有其他程序使用磁盘和处理器, 并且磁盘读写操作和磁盘数据的处理时间不重叠。若程序对磁盘数据的处理需要 20000 个时钟周期, 处理器时钟频率为 500MHz, 则该程序完成一次数据块“读出-处理-写回”操作所需的时间为多少? 每秒钟可以完成多少次这样的数据块操作?

平均存取时间: 控制器开销 + 寻道时间 + 旋转等待时间 + 数据传输时间

$$= 2 + 10 + (1/(7200/(1000 * 60)))/2 + ((4 * 10^3)/(40 * 10^6/10^3)) = 16.2667 \text{ ms}$$

总时间:

$$16.2667 * 2 + (10^3 * 20000)/(500 * 10^6) = 32.5734 \text{ ms}$$

$$\text{每秒可以完成 } 1000 / 32.5734 = 30.6999 \approx 30 \text{ 次}$$

11. 假定某计算机主存地址空间大小为 1GB, 按字节编址, cache 的数据区 (即不包括标记、有效位等存储区) 有 64KB, 块大小为 128 字节, 采用直接映射和直写 (write through) 方式。请问:

- (1) 主存地址如何划分? 要求说明每个字段的含义、位数和在主存地址中的位置。
- (2) cache 的总容量为多少位?

- (1) 按字节编址得, 地址位数为30位。而cache有  $64 * 2^{10} / 128 = 512$  行, 所以行索引有 9位; 块大小为128->内地址需要有7位。

所以地址字段从高到低可以划分为: 14(标志)+9(行索引)+7(内地址)

也就是  $2^{14}$  组  $\times$   $2^9$  块/组  $\times$   $2^7$  B/块

- (2) 总容量: 数据+标志位+有效位 (直写法无修改位)

$$\text{为 } 512 * (14 + 1) / 2^{10} + 64 * 8 = 519.5 \text{ Kb (1K} = 2^{10}, \text{ b 为 位)}$$

12. 假定某计算机的 cache 共 16 行, 开始为空, 块大小为 1 个字, 采用直接映射方式, 按字编址。CPU

12. 假定某计算机的 cache 共 16 行, 开始为空, 块大小为 1 个字, 采用直接映射方式, 按字编址。CPU 执行某程序时, 依次访问以下地址序列: 2, 3, 11, 16, 21, 13, 64, 48, 19, 11, 3, 22, 4, 27, 6 和 11。

(1) 说明每次访问是命中还是缺失, 试计算访问上述地址序列的命中率。

(2) 若 cache 数据区容量不变, 而块大小改为 4 个字, 则上述地址序列的命中情况又如何?

直接映射, 按字编址

(1)

行索引 要 4 位; 块内地址不用了;  $x(\text{标志}) + 4(\text{行号})$

直接 mod 16 即可判断。

地址	2	3	11	16	21	13	64	48	19	11	3	22	4	27	6	11
是否命中	miss	miss	miss	miss	miss	miss	miss	miss	miss	hit	miss	miss	miss	miss	miss	miss

命中率:  $1/16 = 6.25\%$

(2)

Cache 行索引 2 位置; 块地址 2 位; 划分为  $x + 2 + 2$

主存块号 =  $\text{add} / \text{块地址数} = \text{Add}/4$ , 行号 =  $\text{主存块号} \% \text{Cache 行数} = (\text{Add}/4) \% 4$

地址	2	3	11	16	21	13	64	48	19	11	3	22	4	27	6	11
主存块号	0	0	2	4	5	3	16	12	4	2	0	5	1	6	1	2
行号	0	0	2	0	1	3	0	0	0	2	0	1	1	2	1	2
是否命中	miss	hit	miss	miss	miss	miss	miss	miss	miss	hit	miss	hit	miss	miss	hit	miss

命中率:  $4/16 = 25\%$

13. 假定数组元素在主存按从左到右的下标优先顺序存放。试改变下列函数中循环的顺序, 使得其数组元素的访问与排列顺序一致, 并说明为什么修改后的程序比原来的程序执行时间更短。

```
int sum_array(int a[N][N][N])
{
    int i, j, k, sum=0;
    for(i=0; i<N; i++)
        for(j=0; j<N; j++)
            for(k=0; k<N; k++) sum+=a[k][i][j];
    return sum;
}
```

```
int sum_array(int a[N][N][N]) {
    int i, j, k, sum = 0;
    for (k = 0; k < N; ++k)
        for (i = 0; i < N; ++i)
            for (j = 0; j < N; ++j)
                sum += a[k][i][j];
    return sum;
}
```

// 修改后程序的数组元素访问和排列顺序一致,

// 使得空间局部性比原先要好,

// 故其执行时间更短

```
// 使得空间局部性比原先要好,
// 故而执行时间更短。
```

16. 以下是对矩阵进行转置的程序段：

```
typedef int array[4][4];
void transpose(array dst, array src)
{
    int i, j;
    for(i=0; i<4; i++)
        for(j=0; j<4; j++) dst[j][i]=src[i][j];
}
```

假设该段程序运行的计算机中 sizeof(int)=4,且只有一级 cache,其中 L1 数据 cache 的数据区大小为 32B,采用直接映射、回写方式,块大小为 16B,初始为空。数组 dst 从地址 0000 C000H 开始存放,数组 src 从地址 0000 C040H 开始存放。填写下表,说明对数组元素 src[row][col]和 dst[row][col]的访问是命中(hit)还是缺失(miss)。若将 L1 数据 cache 的数据区容量改为 128B,请重新填写表中内容。

	src 数组				dst 数组			
	col=0	col=1	col=2	col=3	col=0	col=1	col=2	col=3
row=0	miss				miss			
row=1								
row=2								
row=3								

32B	src数组 0000 C040				dst数组 0000 C000			
2块	col = 0	col = 1	col = 2	col = 3	col = 0	col = 1	col = 2	col = 3
row = 0	0-miss	0-miss	0-hit	0-miss	0-miss	0-miss	0-miss	0-miss
row = 1	1-miss	1-hit	1-miss	1-hit	1-miss	1-miss	1-miss	1-miss
row = 2	0-miss	0-miss	0-hit	0-miss	0-miss	0-miss	0-miss	0-miss
row = 3	1-miss	1-hit	1-miss	1-hit	1-miss	1-miss	1-miss	1-miss

128B	src数组 0000 C040				dst数组 0000 C000			
8块	col = 0	col = 1	col = 2	col = 3	col = 0	col = 1	col = 2	col = 3
row = 0	4-miss	4-hit	4-hit	4-hit	0-miss	0-hit	0-hit	0-hit
row = 1	5-miss	5-hit	5-hit	5-hit	1-miss	1-hit	1-hit	1-hit
row = 2	6-miss	6-hit	6-hit	6-hit	2-miss	2-hit	2-hit	2-hit
row = 3	7-miss	7-hit	7-hit	7-hit	3-miss	3-hit	3-hit	3-hit

18. 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 替换算法和回写(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 的总容量有多少位?

(3) 假设 cache 初始为空,CPU 依次从 0 号地址单元顺序访问到 4344 号单元,重复按此序列共访问 16 次。若 cache 命中时间为 1 个时钟周期,缺失损失为 10 个时钟周期,则 CPU 访存的平均时间为多少时钟周期?

(1) 4路组相连,一个组就有  $4 * 64B = 2^8B$ , 共有  $4 * 2^{10} / 2^8 = 16$  组

Cache索引 4位, 块内地址 6位。

地址 需要的位数:  $\log_2(64 * 2^{20}) = 26$  位

字段从高到低划分为: 16(标志)+4(Cache索引)+6(块内地址)

也就是  $2^{16}$  组群  $\times 2^4$  块/组群  $\times 2^6 B$ /块

(2)

$64(\text{块Cache}) * (1(\text{有效位}) + 16(\text{标志位}) + 1(\text{回写法修改位}) + 2(\text{LRU位})) / (2^{10} * 8) + 4 = 4.1563 \text{ KB}$

(3)

$4345 / 64 = 67.8906$  上取整 为 68块, 所以实际是对0~67块进行访问16次

	第0组	第1组	第2组	第3组	第4组	第5组	第6组	第7组	第8组	第9组	第10组	第11组	第12组	第13组	第14组	第15组
第0行	0,64	1,65	2,66	3,67	4	5	6	7	8	9	10	11	12	13	14	15
第1行	16,0	17,1	18,2	19,3	20	21	22	23	24	25	26	27	28	29	30	31
第2行	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
第3行	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63

第一次循环 每一个块内只有一个字未命中; 剩余15次循环, 有20块的第一字未命中, 其余命中未命中:  $68 + 15 * 20 = 368$

全部时间:  $4345 * 1 * 16 + 368 * 10 = 73200$

cpu平均时间:  $73200 / (4345 * 16) = 1.0529$  个时钟周期

22. 假定有 3 个处理器,分别带有以下不同的 cache:

cache 1: 采用直接映射方式,块大小为 1 个字,指令和数据的缺失率分别为 4%和 6%。

cache 2: 采用直接映射方式,块大小为 4 个字,指令和数据的缺失率分别为 2%和 4%。

cache 3: 采用 2 路组相联映射方式,块大小为 4 个字,指令和数据的缺失率分别为 2%和 3%。

在这些处理器上运行同一个程序,其中有一半是访存指令,在 3 个处理器上测得该程序的 CPI 都为 2.0。已知处理器 1 和 2 的时钟周期都为 420ps,处理器 3 的时钟周期为 450ps。若缺失损失为(块大小 + 6) 个时钟周期,请问: 哪个处理器因 cache 缺失而引起的额外开销最大? 哪个处理器执行速度最快?

由于缺失率已知,采用何种映射对我们得计算没有影响, 我这里使用了excel表格:

计算平均一条指令得花销和总时间,

其中 额外花销 = (块大小 + 6) \* (50%\*数据缺失率 + 指令缺失率)

总时间 = 额外花销 + 2

单位cc \* 时钟周期 = 单位ps



	块大小	时钟周期	缺失率	
	无量纲	单位ps	指令	数据
cache1	1	420	4%	6%
cache2	4	420	2%	4%
cache3	4	450	2%	3%
	额外花销		总时间	
	单位 cc	单位 ps	单位 cc	单位 ps
cache1	0.49	205.8	2.49	1045.8
cache2	0.4	168	2.4	1008
cache3	0.35	157.5	2.35	1057.5

发现 处理器1 因cache确实引起得额外开销最大(不论是百分比还是绝对时间)

处理器2的执行速度最快

25. 假定一个计算机系统有一个 TLB 和一个 L1 数据 cache。该系统按字节编址,虚拟地址 16 位,物理地址 12 位,页大小为 128B; TLB 采用 4 路组相联方式,共有 16 个页表项; L1 数据 cache 采用直接映射方式,块大小为 4B,共 16 行。在系统运行到某一时刻时, TLB、页表和 L1 数据 cache 中的部分内容如下:

组号	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	有效位
0	03	—	0	09	0D	1	00	—	0	07	02	1
1	13	2D	1	02	—	0	04	—	0	0A	—	0
2	02	—	0	08	—	0	06	—	0	03	—	0
3	07	—	0	63	0D	1	0A	34	1	72	—	0

(a) TLB(4 路组相联): 4 组、16 个页表项

虚页号	页框号	有效位	行索引	标记	有效位	字节 3	字节 2	字节 1	字节 0
00	08	1	0	19	1	12	56	C9	AC
01	03	1	1	—	0	—	—	—	—
02	14	1	2	1B	1	03	45	12	CD
03	02	1	3	—	0	—	—	—	—
04	—	0	4	32	1	23	34	C2	2A
05	16	1	5	0D	1	46	67	23	3D
06	—	0	6	—	0	—	—	—	—
07	07	1	7	16	1	12	54	65	DC
08	13	1	8	24	1	23	62	12	3A
09	17	1	9	—	0	—	—	—	—
0A	09	1	A	2D	1	43	62	23	C3
0B	—	0	B	—	0	—	—	—	—
0C	19	1	C	12	1	76	83	21	35
0D	—	0	D	16	1	A3	F4	23	11
0E	11	1	E	33	1	2D	4A	45	55
0F	0D	1	F	—	0	—	—	—	—

(b) 部分页表(开始 16 项)

(c) L1 数据 cache: 直接映射, 共 16 行, 块大小为 4B

请问(假定图中数据都为十六进制形式):

- (1) 虚拟地址中哪几位表示虚拟页号? 哪几位表示页内偏移量? 虚拟页号中哪几位表示 TLB 标记? 哪几位表示 TLB 索引?
- (2) 物理地址中哪几位表示物理页号? 哪几位表示页内偏移量?
- (3) 主存物理地址如何划分成标记字段、行索引字段和块内地址字段?
- (4) CPU 从地址 067AH 中取出的值为多少? 说明 CPU 读取地址 067AH 中内容的过程。

(1) 页大小为128B故而低7位表示页内偏移量;

故而, 高9位表示虚拟页号;

虚页号中, 低2位为TLB组索引(因为16项4路共有4组), 高7位为TLB标记

(2) 低7位表示页内偏移量, 故而高5位表示物理页号

(3) 因为物理地址为12位, Cache直接映射,

故而从高到低  $12 = 6(\text{标记}) + 4(\text{Cache行索引}) + 2(\text{块内地址})$

(4)  $0x067A = 0000\ 0110\ 0111\ 1010$

得到 虚拟页号 =  $0x03$  TLB标记 +  $00$  组索引, 查询快表失败;

进入主存中页表查询  $0x00C$  查询成功, 找到物理页号 $0x19 = 1\ 1001$

得到物理地址  $1100\ 1111\ 1010$

得到  $0x33$ 标记;  $0xE$ 行索引; 10块内地址(字节2)

查询cache得  $0x4A = 0100\ 1010$