

# 《计算机组成原理》第八次作业

信息安全 胡博浩 2212998

## 5.1

1)

一个32位整数需要四字节表示，16字节的cache块可以存放 $16/4=4$ 个

2)

I, J, B[I][0]。在第二个循环内，I和J会被频繁访问，并且对于每个I，B[I][0]被使用了8000次，显示出时间局部性

3)

A[I][J]。按行存储，A[I][J]重复访问，显示出空间局部性

4)

$$(8000 \times 8 \times 2 - 8 \times 8 + 8) / 4 = 31986$$

5)

同理，I、J、B[I][0]

6)

同理，A[J,I]

## 5.2

1)

二进制地址	标记	索引	命中/缺失
0000 0011	0000	0011	X
1011 0100	1011	0100	X

0010 1011	0010	1011	X
0000 0010	0000	0010	X
1011 1111	1011	1111	X
0101 1000	0101	1000	X
1011 1110	1011	1110	X
0000 1110	0000	1110	X
1011 0101	1011	0101	X
0010 1100	0010	1100	X
1011 1010	1011	1010	X
1111 1101	1111	1101	X

2)

二进制地址	标记	索引	命中/缺失
0000 0011	0000	001	X
1011 0100	1011	010	X
0010 1011	0010	101	X
0000 0010	0000	001	√
1011 1111	1011	111	X
0101 1000	0101	100	X
1011 1110	1011	111	√
0000 1110	0000	111	X
1011 0101	1011	010	√
0010 1100	0010	110	X
1011 1010	1011	101	X
1111 1101	1111	110	X

3)

1. 块大小为1:

缺失率100%，周期为 $25 \times 12 + 2 \times 12 = 324$

2. 块大小为2:

缺失率为 $10/12 = 83.3\%$ ，周期为 $25 \times 10 + 3 \times 12 = 286$

3. 块大小为4:

缺失率为 $11/12 = 91.7\%$ ，周期为 $25 \times 11 + 5 \times 12 = 335$

所以块大小为2的时候最优

5.4

可以，块地址[63:54]和块地址[53:44]各占10位，将这两个10位数进行XOR运算，结果仍然是一个10位数。因此可以把64位字地址映射到1024个块上。

但是，由于异或运算会丢失一些信息，不同的块地址可能会生成相同的索引，因此需要更多的标记位来区分这些块地址。

5.5

1)

$5 - 2 = 3$ ，块大小为 $2^3 = 8$ 字

2)

Cache有 $2^5 = 32$ 项

3)

标志位为22位，则比率为 $1 + 22 / (8 \times 32) = 1.086$

4)

十六进制地址	十六进制标记	十六进制索引	十六进制偏移	命中/缺失	被替换
00	0	00	00	X	
04	0	00	04	√	

10	0	00	10	√	
84	0	04	04	X	
E8	0	07	08	X	
A0	0	05	00	X	
400	1	00	00	X	√
1E	0	00	1E	X	√
8C	0	04	0C	√	
C1C	3	00	1C	X	√
B4	0	05	14	√	
884	2	04	04	X	√

5)

4/12=33%

6)

索引	标记	数据
0	3	3072-3103
4	2	2176-2207
5	0	160-191
7	0	224-255

5.6

1)

L1和L2之间写缓存，当L1Cache缺失时启用

L2和RAM之间写缓存，当Cache中数据被修改写回时启用

2)

当发生写入未命中时，数据将直接写入L2，因为L1具有非写入分配策略（我们不会将块从L2移动到L1）。这可能会导致L2中的写入未命中。在这种情况下，我们写这个数据到RAM并将块移动到L2（L2具有写入分配策略）。但是，我们必须注意不要更换脏块，并且如果它是，我们必须首先将其写入RAM。

### 3)

在L1写入未命中后，块将驻留在L2中，但不驻留在L1中。同一块的后续读取未命中将要求将L2中的块写回内存，传输到L1，并使L2中有效位为0。

## 5.10

### 1)

$$P1 \text{ 为 } 1 / (0.66 * 10^{-9}) = 1.52 \text{ GHz}$$

$$P2 \text{ 为 } 1 / (0.90 * 10^{-9}) = 1.11 \text{ GHz}$$

### 2)

对于 P1，所有内存访问至少需要一个周期来访问 L1。8%的内存访问还需要70ns的主存访问。即  $70 / 0.66 = 106.06$ ，四舍五入107个周期。因此，平均内存访问时间为  $1 + 0.08 * 107 = 9.56$  个周期，即 6.31ps。

对于 P2，主存访问需要70ns。即  $70 / 0.66 = 77.78$ ，四舍五入78个周期。平均内存访问时间为  $1 + 0.06 * 78 = 5.68$  个周期，即 6.11ps。

### 3)

对于 P1，每条指令至少需要一个周期。此外，8%的指令在指令缓存中丢失，并产生 107个周期的延迟。而且，36%的指令是数据访问。这36%中有8%缓存未命中，这增加了额外的107个周期。CPI为  $1 + 0.08 * 107 + 0.36 * 0.08 * 107 = 12.64$ ，时钟周期为 0.66 ps，则平均时间为 8.34ns。

同理，我们可以得到P2的CPI为7.36，平均每条指令仅为6.63ns。

### 4)

$5.62 / 0.66 = 8.52$ ，则L2访问需要9个周期。所有内存访问至少需要一个周期。8%的内存访问在L1缓存中缺失并进行L2访问，这需要9个周期。95%的L2访问都是未命中，需要 107个周期的内存查找。因此，AWAT为  $1 + 0.08 * (9 + 0.95 * 107) = 9.85$ ，比没有L2Cache更差了。

### 5)

和3相同，CPI =  $1 + 0.08 * 110.65 + 0.08 * 0.36 * 110.65 = 13.04$

### 6)

由于两个版本的P1的时钟周期时间和内存指令百分比相同，因此只需关注 AMAT就足够了。要使有L2的AMAT<只有L1的AMAT，即 $1+0.08 \cdot (9+m \cdot 107) < 9.56$ ，解得 $m < 0.916$ ，故缺失率最多可以达到91.6%。

7)

要使P1每条指令的平均时间小于6.63 ns，即 $(CPI\_P1 \cdot 0.66) < 6.63$ ，得 $CPI\_P1 < 10.05$ 。由于 $CPI\_P1 = AMAT\_P1 + 0.36 \cdot (AMAT\_P1 - 1)$ ，代入得 $AMAT\_P1 < 7.65$ 。即 $1+0.08 \cdot (9+m \cdot 107) < 7.65$ ，解得 $m < 0.693$ 。故缺失率最多可以达到69.3%。

## 5.11

1)

块数为 $48/2=24$ ，每组有 $24/3=8$ 个块。偏移位为 $\log_2 2=1\text{bit}$ ，索引位为 $\log_2 8=3\text{bits}$ ，标记位为 $8-(3+1)=4\text{bits}$ 。因此，大体组织结构为

Tag	Index	Offset
4bits	3bits	1bit

2)

设T(x)是索引x的标记位。则有

Word Address	Binary Address	Tag	Index	Offset	Hit/Miss	Way 0	Way 1	Way 2
0x03	0000 0011	0x0	1	1	X	T(1)=0		
0xb4	1011 0100	0xb	2	0	X	T(1)=0 T(2)=b		
0x2b	0010 1011	0x2	5	1	X	T(1)=0 T(2)=b T(5)=2		
0x02	0000 0010	0x0	1	0	✓	T(1)=0 T(2)=b T(5)=2		
0xbe		0xb	7	0	X			

	1011 1110					T(1)=0 T(2)=b T(5)=2 T(7)=b		
0x58	0101 1000	0x5	4	0	X	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5		
0xbf	1011 1111	0xb	7	1	√	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5		
0x0e	0000 1110	0x0	7	0	X	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=0	
0x1f	11111	0x1	7	1	X	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=0	T(7)=1
0xb5	1011 0101	0xb	2	1	√	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=0	T(7)=1
0xbf	1011 1111	0xb	7	1	√	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=0	T(7)=1
0xba	1011 1010	0xb	5	0	X	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=2 T(5)=b	T(7)=1

0x2e	0010 1110	0x2	7	0	X	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=2 T(5)=b	T(7)=1
0xce	1100 1110	0xc	7	0	X	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=2 T(5)=b	T(7)=c

### 3)

块数为 $8/1=8$ 。因为这个Cache是全相联的，并且块大小一个字，所以无偏移量，没有索引，address等同于标记、大小为8bits。因此，大体组织结构为

Tag
8bits

### 4)

因为这个Cache是全相联的，并且每个块大小一个字，所以没有索引，无偏移量。因此，address等同于标记。

Word Address	Binary Address	Tag	Hit/Miss	Contents
0x03	0000 0011	0x03	X	3
0xb4	1011 0100	0xb4	X	3, b4
0x2b	0010 1011	0x2b	X	3, b4, 2b
0x02	0000 0010	0x02	X	3, b4, 2b, 2
0xbe	1011 1110	0xbe	X	3, b4, 2b, 2, be
0x58	0101 1000	0x58	X	3, b4, 2b, 2, be, 58
0xbf	1011 1111	0xbf	X	3, b4, 2b, 2, be, 58, bf



0x0e	0000 1110	0x0e	X	3, b4, 2b, 2, be, 58, bf, e
0x1f	0001 1111	0x1f	X	b4, 2b, 2, be, 58, bf, e, 1f
0xb5	1011 0101	0xb5	X	2b, 2, be, 58, e, 1f, b5, b5
0xbf	1011 1111	0xbf	√	2b, 2, be, 58, e, 1f, b5, bf
0xba	1011 1010	0xba	X	2, be, 58, e, 1f, b5, bf, ba
0x2e	0010 1110	0x2e	X	be, 58, e, 1f, b5, bf, ba, 2e
0xce	1100 1110	0xce	X	58, e, 1f, b5, bf, ba, 23, ce

## 5)

块数为 $8/2=4$ 。由于此Cache是全相联的，因此没有索引，偏移位为 $\log_2 2=1\text{bit}$ ，标记位为 $8-1=7\text{bits}$ 。因此，大体组织结构为

Tag	Offset
7bits	1bit

## 6)

由于此Cache是全相联的，因此没有索引。显示的内容按访问数据的顺序排列，顺序并不意味着物理位置。

Word Address	Binary Address	Tag	Offset	Hit/Miss	Contents
0x03	0000 0011	0x01	1	X	[2,3]
0xb4	1011 0100	0x5a	0	X	[2,3], [b4,b5]

0x2b	0010 1011	0x15	1	X	[2,3],[b4,b5], [2a,2b]
0x02	0000 0010	0x01	0	$\sqrt{\quad}$	[b4, b5], [2a,2b], [2,3]
0xbe	1011 1110	0x5f	0	X	[b4,b5], [2a,2b], [2,3], [be,bf]
0x58	0101 1000	0x2c	0	X	[2a,2b], [2,3], [be,bf], [58,59]
0xbf	1011 1111	0x5f	1	$\sqrt{\quad}$	[2a,2b], [2,3], [58,59], [be,bf]
0x0e	0000 1110	0x07	0	X	[2,3], [58,59], [be,bf], [e,f]
0x1f	0001 1111	0x0f	1	X	[58,59], [be,bf], [e,f], [1e,1f]
0xb5	1011 1111	0x5a	1	X	[be,bf], [e,f], [1e,1f], [b4,b5]
0xbf	1011 1111	0x5f	0	$\sqrt{\quad}$	[e,f],[1e,1f], [b4,b5], [ba,bb]
0xba	1011 1010	0x5d	0	X	[1e,1f], [b4,b5], [b3,bf], [ba,bb]
0x2e	0010 1110	0x17	0	X	[b4,b5], [b3,bf], [ba,bb], [2e,2f]
0xce	1100 1110	0x67	0	X	[be,bf], [ba,bb],

					[2e,2f], [ce,cf]
--	--	--	--	--	---------------------

7)

按数据访问顺序显示的内容,不代表真实的物理位置。

Word Address	Binary Address	Tag	Offset	Hit/Miss	Contents
0x03	0000 0011	0x01	1	X	[2,3]
0xb4	1011 0100	0x5a	0	X	[2,3], [b4,b5]
0x2b	0010 1011	0x15	1	X	[2,3], [b4,b5], [2a,2b]
0x02	0000 0010	0x01	0	√	[b4, b5], [2a,2b], [2,3]
0xbe	1011 1110	0x5f	0	X	[b4,b5], [2a,2b], [2,3], [be,bf]
0x58	0101 1000	0x2c	0	X	[b4,b5], [2a,2b], [2,3], [58,59]
0xbf	1011 1111	0x5f	1	√	[b4,b5], [2a,2b], [2,3], [be,bf]
0x0e	0000 1110	0x07	0	X	[b4,b5], [2a,2b], [2,3], [e,f]
0x1f	0001 1111	0x0f	1	X	[b4,b5], [2a,2b], [2,3], [1e,1f]
0xb5	1011 1111	0x5a	1	√	[2a,2b], [2,3], [1e,1f], [b4,b5]
0xbf	1011 1111	0x5f	1	X	

					[2a,2b], [2,3], [1e,1f], [ba,bb]
0xba	1011 1010	0x5d	0	X	[2a,2b], [2,3], [1e,1f], [ba,bb]
0x2e	0010 1110	0x17	0	X	[2a,2b], [2,3], [1e,1f], [2e,2f]
0xce	1100 1110	0x67	0	X	[2a,2b], [2,3], [1e,1f], [ce,cf]

## 8)

由于此Cache是全相联的，因此没有索引。

Word Address	Binary Address	Tag	Offset	Hit/Miss	Contents
0x03	0000 0011	0x01	1	X	[2,3]
0xb4	1011 0100	0x5a	0	X	[2,3], [b4,b5]
0x2b	0010 1011	0x15	1	X	[2,3], [b4,b5], [2a,2b]
0x02	0000 0010	0x01	0	√	[2,3], [b4,b5], [2a,2b]
0xbe	1011 1110	0x5f	0	X	[2,3], [b4,b5], [2a,2b], [be,bf]
0x58	0101 1000	0x2c	0	X	[58,59], [b4,b5],

					[2a,2b], [be,bf]
0xbf	1011 1111	0x5f	1	$\sqrt{\quad}$	[58,59], [b4,b5J, [2a,2b], [be,bf]
0x0e	0000 1110	0x07	0	X	[e,f], [b4,b5], [2a,2b], [be,bf]
0x1f	0001 1111	0x0f	1	X	[b4,b5J, [2a,2b], [be,bf]
0xb5	1011 1111	0x5a	1	$\sqrt{\quad}$	[1e,1f], [b4,b5], [2a,2b], [be,bf]
0xbf	1011 1111	0x5f	1	$\sqrt{\quad}$	[1e,1f], [b4,b5], [2a,2b], [be,bf]
0xba	1011 1010	0x5d	0	X	[1e,1f], [b4,b5], [ba,bb], [be,bf]
0x2e	0010 1110	0x17	0	X	[1e,1f], [b4,b5], [2e,2f], [be,bf]
0xce	1100 1110	0x67	0	X	[1e,1f], [b4,b5], [ce,cf], [be,bf]