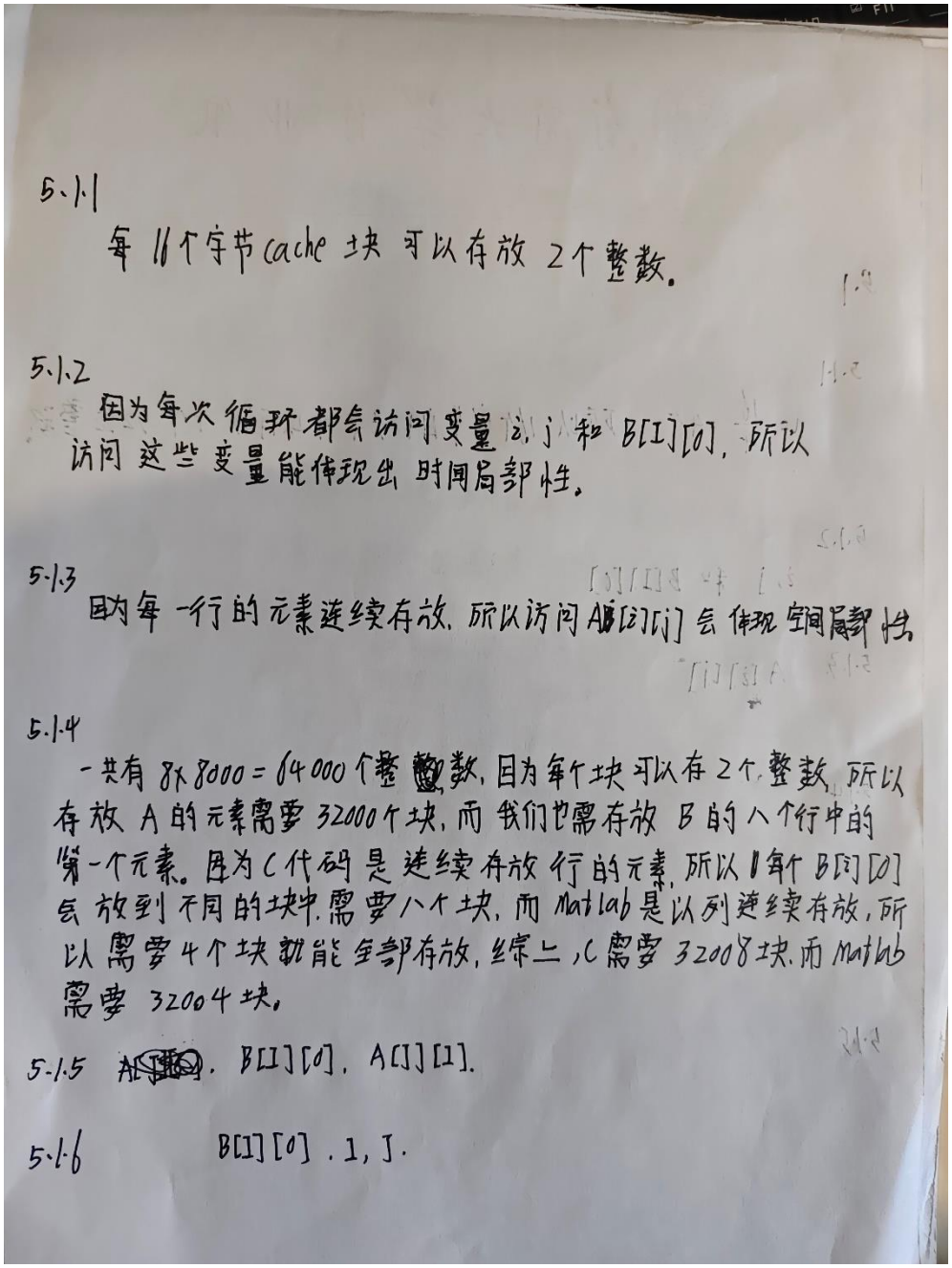
5.1



5.1.1

每 16 个字节 cache 块 可以存放 2 个整数.

5.1.2

因为每次循环都会访问变量 i, j 和 B[I][0], 所以访问 这些变量能体现出 时间局部性.

5.1.3

因为每一行的元素连续存放, 所以访问 A[i][j] 会体现空间局部性.

5.1.4

一共有 8×8000 = 64000 个整数, 因为每个块可以存 2 个整数, 所以存放 A 的元素需要 32000 个块, 而我们也需存放 B 的八个行中的第一个元素. 因为 C 代码是连续存放行的元素, 所以 16 个 B[i][0] 会放到不同的块中, 需要八个块, 而 Matlab 是以列连续存放, 所以需要 4 个块就能全部存放, 综上, C 需要 32008 块, 而 Matlab 需要 32004 块.

5.1.5 A[i][j], B[I][0], A[I][1].

5.1.6 B[I][0] . I, J.

5.2

南开大学 作业纸

系别_____ 班级_____ 姓名_____ 第 页

5.2.1

因为是按字编址的，所以我们可以用字为单位，而不是用字节

0x03 0000(标记) 0011(索引)

0x2b 0010(标记) 1011(索引)

0xbf 1011(标记) 1111(索引)

0xbe 1011(标记) 1110(索引)

0xb5 1011(标记) 0101(索引)

0xba 1011(标记) 1010(索引)

0xb4 1011(标记) 0100(索引)

0x02 0000(标记) 0010(索引)

0x58 0101(标记) 1000(索引)

0x0e 0000(标记) 1110(索引)

0x2c 0010(标记) 1100(索引)

0xfd 1111(标记) 1101(索引)

观察之后发现，没有哪一次访问是命中的，都是缺失的。

南开大学 作业纸

系别_____ 班级_____ 姓名_____ 第 页

5.2.2

因为每个块有2个字，所以字偏移就有一位，然后索引是3位。

0x03	0000 0011	0000	001	缺失
0xb4	1011 0100	1011	010	缺失
0x2b	0010 1011	0010	101	缺失
0x02	0000 0010	0000	001	命中
0xbf	1011 1111	1011	111	缺失
0x58	0101 1000	0101	100	缺失
0xbe	1011 1110	1011	111	命中
0x0e	0000 1110	0000	111	缺失
0xb5	1011 0101	1011	010	命中
0x2c	0010 1100	0010	110	缺失
0xba	1011 1010	1011	101	缺失
0xfd	1111 1101	1111	110	缺失

5.2.3

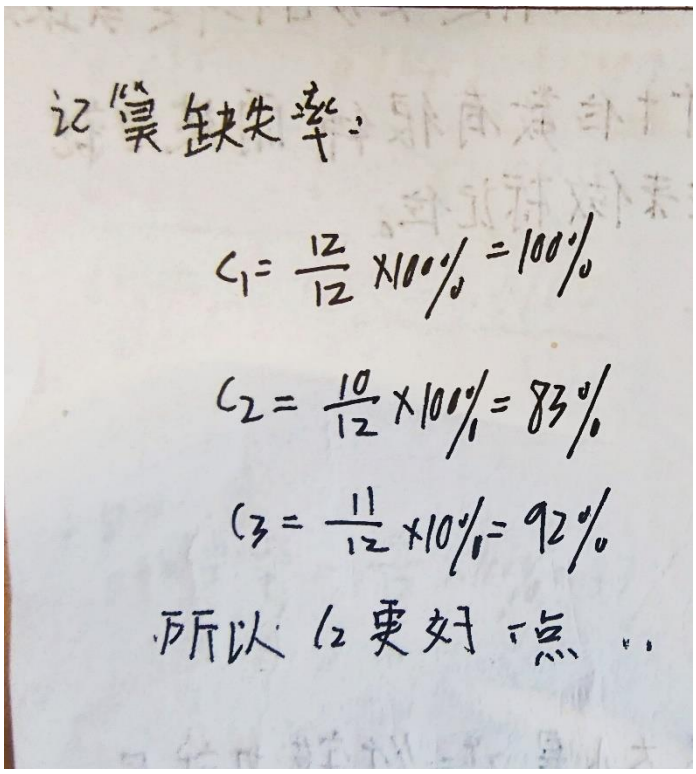
对(1)来说 $\frac{8}{1}=8$个块，所以索引位有3位，标记位有五位

对(2)来说 $\frac{8}{2}=4$个块，所以有#2位索引位，标记位有五位

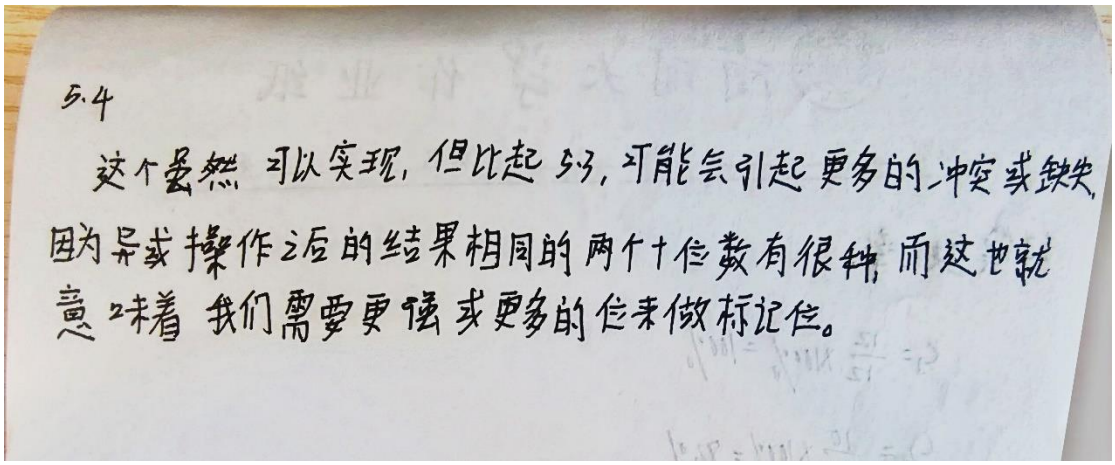
对(3)来说 $\frac{8}{4}=2$个块，所以有一位索引位，标记位有五位

所以三种方案的标记位都是一样的。

地址	=进制地址	标记	(1)索引	状态	(2)索引	状态	(3)索引	状态
0x03	0000 0011	00000	011	缺失	01	缺失	0	缺失
0xb4	1011 0100	10110	100	缺失	10	缺失	1	缺失
0x2b	0010 1011	00101	011	缺失	01	缺失	0	缺失
0x02	0000 0010	00000	010	缺失	01	缺失	0	缺失
0xbf	1011 1111	10111	111	缺失	11	缺失	1	缺失
0x58	0101 1000	01011	000	缺失	00	缺失	0	缺失
0xbe	1011 1110	10111	110	缺失	11	命中	1	命中
0x0e	0000 1110	00001	110	缺失	11	缺失	1	缺失
0xb5	1011 0101	10110	101	缺失	10	命中	1	缺失
0x2c	0010 1100	00101	100	缺失	10	缺失	1	缺失
0xba	1011 1010	10111	010	缺失	01	缺失	0	缺失
0xfd	1111 1101	11111	101	缺失	10	缺失	1	缺失


记号缺失率：
$C_1=\frac{12}{12}\times100\%=100\%$
$C_2=\frac{10}{12}\times100\%=83\%$
$C_3=\frac{11}{12}\times10\%=92\%$
所以C2更好一点。。

5.4


5.4
这个虽然可以实现，但比起5.3，可能会引起更多的冲突或缺失，
因为异或操作之后的结果相同的两个十位数有很种，而这也就
意味着我们需要更强或更多的位来做标记位。

5.5

5.5

5.5.1

因为字节偏移量有5位，所以大小是 $2^5=32$ 字节，也就是 $\frac{32}{4}=8$ 个字

5.5.2 因为索引位有5位，所以有 $2^5=32$ 个 cache.

5.5.3

cache 总位数：

$$32\times(1+22+32\times8)=8928(位)$$

数据总位数：

$$32\times32\times8=8192(位)$$

比率：$\frac{8928}{8192}=1.09$

5.5.4

地址	二进制地址	标记	索引	偏移	状态	替代
0x00	0000 0000 0000	0x00	0x00	0x00	缺失	
0x04	0000 0000 0100	0x00	0x00	0x04	命中	
0x10	0000 0001 0000	0x00	0x01	0x10	命中	
0x84	0000 1000 0100	0x00	0x04	0x04	缺失	
0xe8	0000 1110 1000	0x00	0x07	0x08	缺失	
0xA0	0000 1010 0000	0x00	0x05	0x00	缺失	
0x400	0100 0000 0000	0x01	0x00	0x00	缺失	0x00-0x1F
0x1e	0000 0001 1110	0x00	0x01	0x1e	缺失	0x400-0x41F
0x8C	0000 1000 1100	0x00	0x04	0x0C	命中	

南开大学 作业纸

系别_____ 班级_____ 姓名_____ 第 页

0xC1C	0 1100 0001 1100	0x03	0x00	0x1C	缺失	0x00-0x1F
0xb4	0000 1011 0100	0x00	0x05	0x14	命中	
0x884	1000 1000 0100	0x02	0x04	0x04	缺失	0x80-0x9f

5.5 命中率 = $\frac{4}{12} \times 100\% = 33\%$

5.6

可以从表中可知，索索引位 0 为 0 的最后是标记位为 3，

索引位为 4 的 最后的标记位是 2.

索引位为 5 的 最后的标记位是 0

索引位为 7 的 最后的标记位是 0

所以 cache 的最后状态为：

<0, 3, 0x00-0x1F>

<4, 2, 0x80-0x9F>

<5, 0, 0x0A0-0x0BF>

<7, 0, 0x0e0-0x0ff>

5.6

南开大学 作业纸

系别_____ 班级_____ 姓名_____ 第 页

5.6.1

因为 L_1 的策略是写直达和写不分配，这也就是说当 L_1 写命中时，它要在 L_1 和 L_2 中同时写入，而写不命中时，它直接写入 L_2，所以当 L_1 和 L_2 的延迟比较大时，会导致时差会很大，所以要在 L_1 和 L_2 中间要有一个 buffer，用来缓冲。而 L_2 采用的是写回和写分配，并不会出现上述情况，所以就不需要 buffer。

5.6.2

就像 5.6.1 中说得那样，当 L_1 发生写缺失的时候，因为采用的是写不分配，所以会直接写入 L_2，而不是把块调到 L_1，而这种操作使 L_2 也发生写缺失的话，它就必须要从内存中取出相应的块放到 L_2 中之后再修改，但这也可能导致 L_2 中已经被修改过的块被替换出去，这时候，被修改过的块就必须要把修改过的内容写入内存了。

5.6.3

在 L_1 写入失败以后，这个块将会保存在 L_2 中，而当对同一块读取失效以后，需要将 L_2 中的块修改的地方写入内存，并把相应的块转入 L_1，并在 L_2 中失效。

5.10

5.10.1

由表可知，

$T_1 = 0.66ns$ ∴ $P_1 = \frac{1}{0.66ns} = 1.515GHz$

$T_2 = 0.90ns$ ∴ $P_2 = \frac{1}{0.90ns} = 1.11GHz$

5.10.2

AMAT = 命中时间 + 缺失率 × 缺失代价

∴ $AMAT_1 = 0.66ns + 0.08 \times 70ns = 6.26ns$

$AMAT_2 = 0.90ns + 0.06 \times 70ns = 5.1ns$

10.3

P_1:

① 先把 70ns 转换为 时钟周期数: $\frac{70ns}{0.66ns} = 106.1$

② 设总指令数为 I

③ 在指令缺失上阻塞的时钟周期数为: $0.08 \times 106.1 \times I$

④ 在数据缺失上阻塞的时钟周期数为: $0.08 \times 0.36 \times 106.1 \times I$

⑤ $CPI = \frac{(1 + 0.08 \times 106.1 + 0.08 \times 0.36 \times 106.1)I}{I} = 12.54$

P_2:

① 把 70ns 转换为 时钟周期数: $\frac{70ns}{0.90ns} = 77.8$

② 设总指令数为 I

③ 在指令缺失上阻塞的时钟周期数为: $0.06 \times 77.8 \times I$

④ 在数据缺失上阻塞的时钟周期数为: $0.06 \times 0.36 \times 77.8 I$

⑤ $CPI = \frac{I + 0.06 \times 77.8 \times I + 0.06 \times 0.36 \times 77.8 \times I}{I} = 7.348$

对于 P_1 来说，一个指令需要 12.54 个时钟周期，也就是 $12.54 \times 0.66 = 8.27ns$

对于 P_2 来说，一个指令需要 7.348 个时钟周期，也就是 $7.348 \times 0.9 = 6.61ns$

所以 P_2 更快一点

5.10.4

AMAT = 命中时间 + 缺失率 × 缺失代价.

先计算 L_2 的AMAT：$5.62ns + 0.95 \times 70ns = 72.12$

再计算 L_1 的AMAT：$0.66 + 0.08 \times AMAT(L_2) = 0.66 + 0.08 \times 72.12$

$= 6.43(ns)$

≈ 9.85 个时钟周期.

AMAT 比以前变大了，所以是变差了。

5.10.5

指令缺失的阻塞为：$I \times 0.08 \times (9 + 0.95 \times 107) = 8.852$ I.

数据缺失的阻塞为：$I \times 0.08 \times 0.36 \times (9 + 0.95 \times 107) = 3.186$ I.

∴ CPI = $1 + 8.852 + 3.186 = 13.04$

5.10.6.

也就是 $AMAT(L_1, L_2) < AMAT(L_1)$

$0.66 + 0.08(5.62 + X \times 70) < 0.66 + 0.08 \times 70$.

可得 $X < 0.91$

5.10.7

AMAT(L_1,L_2) P_1 = 0.08 + 0.08(5.02 + X.70)

AMAT(L_1) P_2 = 0.90 + 0.06(70)

AMAT(L_1,L_2) P_1 < AMAT(L_1) P_2

0.16 + 0.08(5.02 + 70X) < 0.90 + 0.06.70

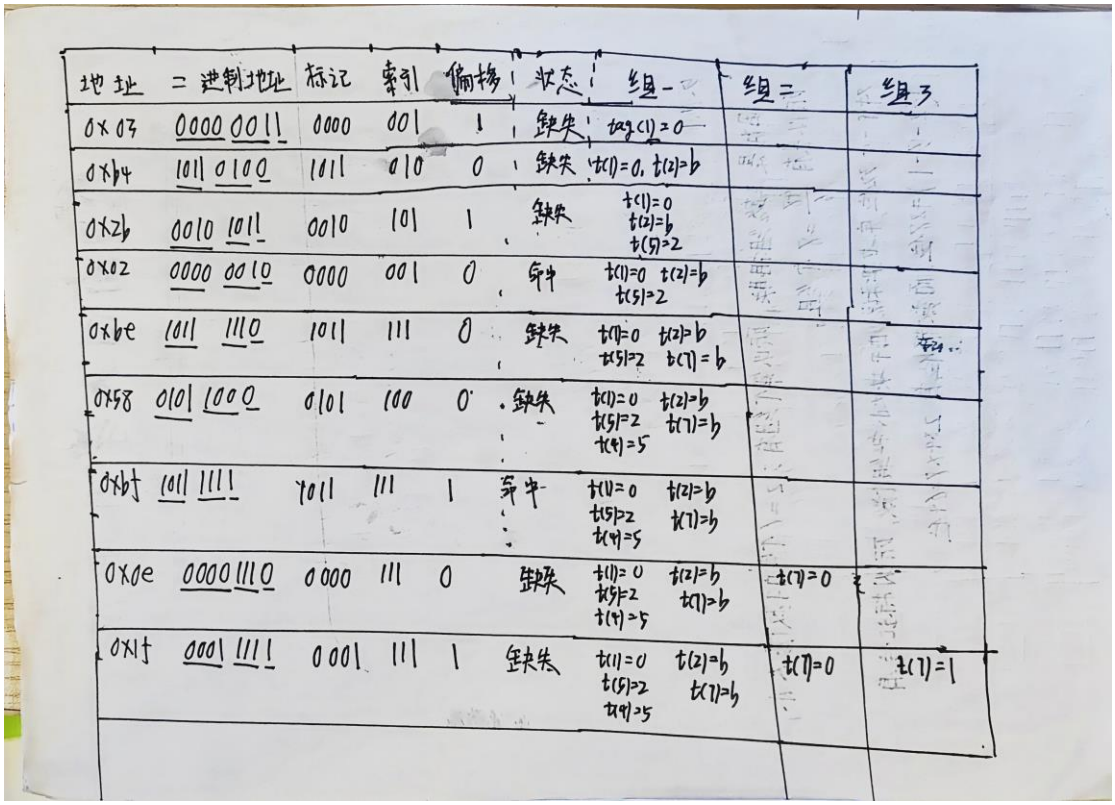
计算可得 X < 0.7

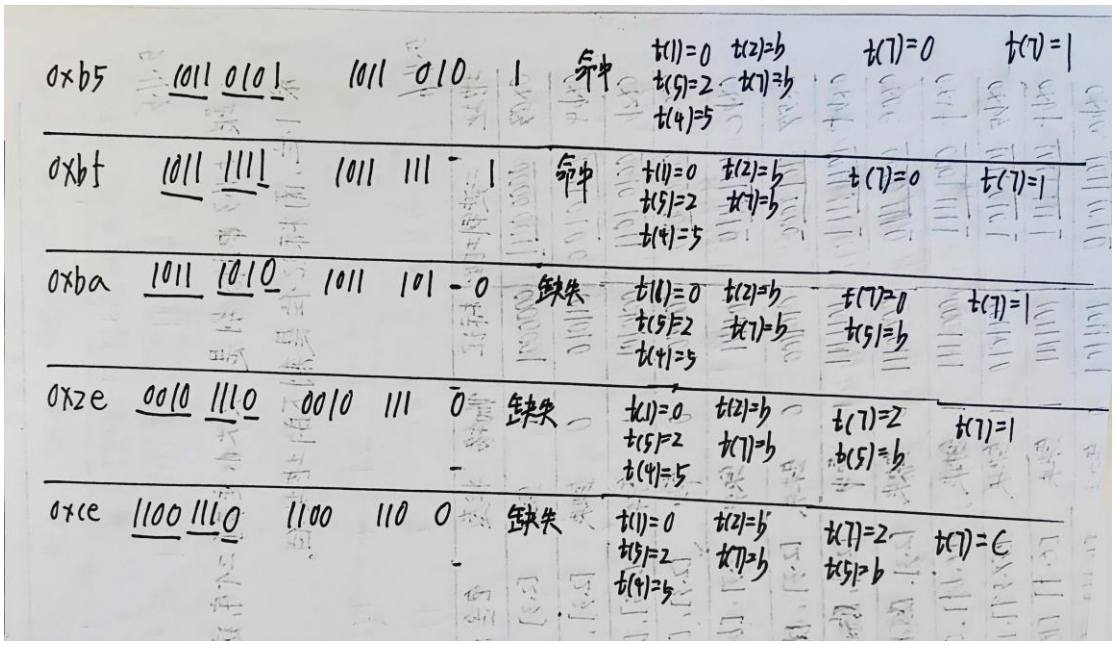
5.11

5.11.1

因为是三路组相联，所以每个组有 3×2 = 6 个字的数据大小，所以共有 $\frac{48}{6}$ = 8 个组。

对于一个 32 位字地址来说，由于块大小和组个数，所以标记位有 (32-3-1) = 28 位，而数据位有 2×4×8 = 64 位


地址 二进制地址 标记 索引 偏移 状态 组一 组二 组三
0x03 0000 0011 0000 00 1 缺失 tag(1)=0
0xb4 1011 0100 1011 010 0 缺失 t(1)=0, t(2)=b
0x92 1001 0010 1101 0010 101 1 缺失 t(1)=0 t(2)=b t(5)=2
0x02 0000 0010 0000 0001 0 命中 t(1)=0 t(2)=b t(5)=2
0xbe 1011 1110 1011 111 0 缺失 t(1)=0 t(2)=b t(5)=2 t(7)=b
0x58 0101 1000 0101 100 0 缺失 t(1)=0 t(2)=b t(5)=2 t(7)=b t(4)=5
0xbf 1011 1111 1011 111 1 命中 t(1)=0 t(2)=b t(5)=2 t(7)=b t(4)=5
0x0e 0000 1110 0000 111 0 缺失 t(1)=0 t(2)=b t(5)=2 t(7)=b t(4)=5 t(7)=0
0x1f 0001 1111 0001 111 1 缺失 t(1)=0 t(2)=b t(5)=2 t(7)=b t(4)=5 t(7)=0 t(7)=1


0x54 1010 1010 1010 101 0 命中 t(1)=0 t(2)=b t(5)=2 t(7)=b t(4)=5 t(7)=0 t(7)=1
0xbf 1011 1111 1011 111 1 命中 t(1)=0 t(2)=b t(5)=2 t(7)=b t(4)=5 t(7)=0 t(7)=1
0xba 1011 1010 1011 101 0 缺失 t(1)=0 t(2)=b t(5)=2 t(7)=b t(4)=5 t(7)=0 t(5)=b t(7)=1
0x2e 0010 1110 0010 111 0 缺失 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 1100 110 0 缺失 t(1)=0 t(2)=b t(5)=2 t(7)=b t(4)=5 t(7)=2 t(5)=b t(7)=3

5.11.3

因为块容量是8字节，所以一共有8个块，而对于一个32位字地址来说，标记位就是全部的32位，而数据位是4×8=32位

5.11.4 分析可知，该cache没有索引和偏移位。

地址	二进制地址	标记	状态	包含的块
0x03	0000 0011	0000 0011	缺失	0x03
0xb4	1011 0100	1011 0100	缺失	0x03, 0xb4
0x2b	0010 1011	0010 1011	缺失	0x03, 0xb4, 0x2b
0x02	0000 0010	0000 0010	缺失	0x03, 0xb4, 0x2b, 0x02
0xbe	1011 1110	1011 1110	缺失	0x03, 0xb4, 0x2b, 0x02, 0xbe
0x58	0101 1000	0101 1000	缺失	0x03, 0xb4, 0x2b, 0x02, 0xbe, 0x58
0xbf	1011 1111	1011 1111	缺失	0x03, 0xb4, 0x2b, 0x02, 0xbe, 0x58, 0xbf
0x0e	0000 1110	0000 1110	缺失	0x03, 0xb4, 0x2b, 0x02, 0xbe, 0x58, 0xbf, 0x0e
0x1f	0001 1111	0001 1111	缺失	0xb4, 0x2b, 0x02, 0xbe, 0x58, 0xbf, 0x0e, 0x1f
0xb5	1011 0101	1011 0101	缺失	0x2b, 0x02, 0xbe, 0x58, 0xbf, 0x0e, 0x1f, 0xb5
0xbf	1011 1111	1011 1111	命中	0x2b, 0x02, 0xbe, 0x58, 0xbf, 0x0e, 0x1f, 0xb5
0xba	1011 1010	1011 1010	缺失	0x02, 0xbe, 0x58, 0xbf, 0x0e, 0x1f, 0xb5, 0xba

0xze	0010 1110	00101110	缺失	0xbe. 0x58. 0x0e 0x1f 0xb5 0xbf 0xba 0xze	
0xce	1100 1110	11001110	缺失	0x58, 0x0e. 0x1f , 0xb5 0xbf 0xba 0xze 0xce	

5.11.5

跟 5.11.3 中的区别是 块大小变成了二个字，这也就意味着，偏移位要变成1位，而标记位是剩下的地址位。

5.11.6

地址	二进制地址	标记	偏移	状态	包含的字.
0x03	00000011	0000001	1	缺失	[2.3]
0xb4	10110100	1011010	0	缺失	[2.3], [b4. b5]
0x2b	0010 1011	0010101	1	缺失	[2.3]. [b4. b5]. [2a. 2b].
0x02	0000 0010	0000001	0	命中	[2.3], [b4.b5]. [2a.2b]
0xbe	1011 1110	1011111	0	缺失.	[2.3]. [b4. b5]. [2a. 2b]. [be. bf].
0x58	0101 1000	0101100	0	缺失	[2.3], [illegible]. [2a. 2b] [be. bf]. [58. 59]
0xbf	1011 1111	1011111	1	命中	[2.3], [illegible]. [be. bf]. [58. 59]
0x0e	0000 1110	0000111	0	缺失	[2.3]. [illegible]. [be. bf]. [58. 59]. [ce. f]
0x1f	0001 1111	0001111	1	缺失	[be. bf]. [58. 59]. [ce. f]. [1e. 1f]
0xb5	1011 0101	1011010	1	缺失	[58. 59]. [ce. f] [1e. 1f] [b4. b5]
0xbf	1011 1111	1011111	1	缺失	[ce. f] [1e. 1f] [b4. b5], [be. bf]
0xba	1011 1010	1011101	0	缺失	[1e. 1f]. [b4. b5]. [be. bf]. [ba. bb]

系别 班级 姓名 第 页

0xze	00101010	0010111	0	缺失	[b4. b5] [be. bf]. [ba. bb]. [2e. 2f]
0xce	11001110	1100111	0	缺失	[be. bf]. [ba. bb]. [2e. 2f]. [ee. ff].

5.11.7

替换的是最近使用的那个块。

地址	二进制地址	标记	偏移	包含的字	状态
0x03	0000 0011	000000	1	[2,3]	缺失
0xb4	1011 0100	1011010	0	[2,3], [b4,b5]	缺失
0x2b	0010 1011	0010101	1	[2,3], [b4,b5], [2a,2b]	缺失
0x02	0000 0010	0000001	0	②[b4,b5] [2a,2b] [2,3]	命中
0xbe	1011 1110	1011111	0	[b4,b5] [2a,2b] [2,3] [be,bf]	缺失
0x58	0101 1000	0101100	0	[b4,b5] [2a,2b] [2,3] [58,59]	缺失
0xbf	1011 1111	1011111	1	[b4,b5] [2a,2b] [2,3], [be,bf]	缺失
0x0e	0000 1110	0000111	0	[b4,b5] [2a,2b] [2,3] [e,f]	缺失
0x1f	0001 1111	0001111	1	[b4,b5] [2a,2b] [2,3] [1e,1f]	缺失
0xb5	1011 0101	1011010	1	[2a,2b] [2,3] [1e,1f] [b4,b5]	命中
0xbf	1011 1111	1011111	1	[2a,2b] [2,3] [1e,1f] [be,bf]	缺失
0xba	1011 1010	1011101	0	[2a,2b] [2,3] [1e,1f] [ba,bb]	缺失
0x2e	0010 1110	0010111	0	[2a,2b] [2,3] [1e,1f] [2e,2f]	缺失
0xce	1100 1110	1100111	0	[2a,2b] [2,3] [1e,1f] [ce,cf]	缺失

南開大學 作业纸

系别_____ 班级_____ 姓名_____ 第 页

5-11.8

想提高命中率，就不能把接下来要访问的块替换出去。
就比如 [be.bf] [b4.b5] 等。

地址	二进制地址	标记	偏移	状态	包含的字
0x03	00000011	0000001	1	缺失	[2.3]
0xb4	10110100	1011010	0	缺失	[2.3] [b4.b5]
0x2b	00101011	0010101	1	缺失	[2.3] [b4.b5] [2a.2b]
0x02	00000010	0000001	0	命中	[2.3] [b4.b5] [2a.2b]
0xbe	10111110	1011111	0	缺失	[2.3] [b4.b5] [2a.2b] [be.bf]
0x58	01011000	0101100	0	缺失	[58.59] [b4.b5] [2a.2b] [be.bf]
0xbf	10111111	1011111	1	命中	[58.59] [b4.b5] [2a.2b] [be.bf]
0x0e	00001110	0000111	0	缺失	[0e.0f] [b4.b5] [2a.2b] [be.bf]
0x1f	00011111	0001111	1	缺失	[1e.1f] [b4.b5] [2a.2b] [be.bf]
0xb5	10110101	1011010	1	命中	[1e.1f] [b4.b5] [2a.2b] [be.bf]
0xbf	10111111	1011111	1	命中	[1e.1f] [b4.b5] [2a.2b] [be.bf]
0xba	10111010	1011101	0	缺失	[ba.bb] [b4.b5] [2a.2b] [be.bf]
0x2e	00101110	0010111	0	缺失	[ba.bb] [b4.b5] [2e.2f] [be.bf]
0xce	11001110	1100111	0	缺失	[ba.bb] [ce.cf] [2e.2f] [be.bf]