

一、对下面给出的十六进制存储器字地址：3、B4、2B、2、BF、58、BE、0E、B5、2C、BA、FD，这里有三种 cache 设计方案，每个容量都为 8 个字：C1 块大小为 1 个字，C2 块大小为 2 个字，C3 块大小为 4 个字。如果采用直接映像，根据缺失率，哪种 cache 设计最好？如果缺失阻塞时间为 25 个周期，C1 的访问时间为 2 个周期，C2 为 3 个周期，C3 为 5 个周期，那么哪种 cache 设计最好？

参考解答： C2 最好

方案 1： 块内地址 0 位，cache 共分 8 块，cache 块号就是 cache 地址，内存块号就是内存地址，因此：内存地址的低 3 位就是索引位，而内存地址的高位部分就是标记位。

	16 进制地址流	2 进制地址流	Tag	Index	Result	备注
1	3	0000 0011	00000	011	Miss	强制缺失
2	B4	1011 0100	10110	100	Miss	强制缺失
3	2B	0010 1011	00101	011	Miss	强制缺失
4	2	0000 0010	00000	010	Miss	强制缺失
5	BF	1011 1111	10111	111	Miss	强制缺失
6	58	0101 1000	01011	000	Miss	强制缺失
7	BE	1011 1110	10111	110	Miss	强制缺失
8	0E	0000 1110	00001	110	Miss	强制缺失
9	B5	1011 0101	10110	101	Miss	强制缺失
10	2C	0010 1100	00101	100	Miss	强制缺失
11	BA	1011 1010	10111	010	Miss	强制缺失
12	FD	1111 1101	11111	101	Miss	强制缺失

说明： 红色表示内存块号，下同！

结果：命中率为 0，访问时间为 $12 \times 2 + 12 \times 25 = 324$

方案 2： 块内地址 1 位，cache 共分 4 块，cache 块号就是 cache 地址的高 2 位，内存块号就是内存地址最低位左边的所有高位部分，因此：内存块号的低 2 位就是索引位，而内存块号的高位部分就是标记位。

	16 进制地址流	2 进制地址流	Tag	Index	Result	备注
1	3	0000 0011	00000	01	Miss	强制缺失
2	B4	1011 0100	10110	10	Miss	强制缺失
3	2B	0010 1011	00101	01	Miss	强制缺失
4	2	0000 0010	00000	01	Miss	冲突缺失
5	BF	1011 1111	10111	11	Miss	强制缺失
6	58	0101 1000	01011	00	Miss	强制缺失
7	BE	1011 1110	10111	11	Hit	

8	0E	0000 1110	00001	11	Miss	强制缺失
9	B5	1011 0101	10110	10	Hit	
10	2C	0010 1100	00101	10	Miss	强制缺失
11	BA	1011 1010	10111	01	Miss	强制缺失
12	FD	1111 1101	11111	10	Miss	强制缺失

命中率为 2/12，访问时间为 $12 \times 3 + 10 \times 25 = 286$

方案 3：块内地址 2 位，cache 共分 2 块，cache 块号就是 cache 地址的最高位，内存块号就是内存地址最低 2 位之外的所有高位部分，因此：内存块号的最低位就是索引位，而内存块号的高位部分就是标记位。

	16 进制地址流	2 进制地址流	Tag	Index	Result	备注
1	3	0000 0011	00000	0	Miss	强制缺失
2	B4	1011 0100	10110	1	Miss	强制缺失
3	2B	0010 1011	00101	0	Miss	强制缺失
4	2	0000 0010	00000	0	Miss	冲突缺失
5	BF	1011 1111	10111	1	Miss	强制缺失
6	58	0101 1000	01011	0	Miss	强制缺失
7	BE	1011 1110	10111	1	Hit	
8	0E	0000 1110	00001	1	Miss	强制缺失
9	B5	1011 0101	10110	1	Miss	冲突缺失
10	2C	0010 1100	00101	1	Miss	强制缺失
11	BA	1011 1010	10111	0	Miss	强制缺失
12	FD	1111 1101	11111	1	Miss	强制缺失

命中率为 1/12，访问时间为 $12 \times 5 + 11 \times 25 = 335$

二、一个 8KB 的 cache，采用 LRU 替换算法。假设 cache 初始时空，按照下列 8 个地址访问 cache (16 位 16 进制字节地址)，采取直接映射时 0xc600 与 0x a6ff 冲突、只有 0x a700 命中：

0x779f、0xa7ff、0xb7dd、0xa6ff、0x7700、0xa700、0xc600、0xa60f。

1. 根据上述缺失和命中情况分析 cache 块的大小，并以图示意分析过程。

参考解答：在直接映像方式下，内存块号的低位部分就是缓存索引（即缓存块号），高位部分就是缓存标记。

内存地址：缓存标记+缓存索引+块内地址=16 位

缓存地址：缓存块号+块内地址=13 位

因此：缓存标记=16-13=3 位

	16 进制地址流	2 进制地址流	Tag	Index	命中情况
1	0x779f	0111 0111 1001 1111	011		Miss
2	0xa7ff	1010 0111 1111 1111	101		Miss
3	0xb7dd	1011 0111 1101 1101	101		Miss
4	0xa6ff	1010 0110 1111 1111	101		Miss
5	0x7700	0111 0111 0000 0000	011		Miss
6	0xa700	1010 0111 0000 0000	101		Hit
7	0xc600	1100 0110 0000 0000	110		Miss
8	0xa60f	1010 0110 0000 1111	101		Miss

0xc600 与 0xa6ff 发生冲突，说明 offset 至少 8 位，index 至多 5 位；

仅 0xa700 命中，说明 index 至少 5 位（否则，0xa6ff 等也会命中）。

因此，tag 占 3 位，index 占 5 位，offset 占 8 位。

而且：

	16 进制地址流	2 进制地址流	访问情况
1	0x779f	011 10111 1001 1111	强制缺失
2	0xa7ff	101 00111 1111 1111	强制缺失
3	0xb7dd	101 10111 1101 1101	强制缺失
4	0xa6ff	101 00110 1111 1111	强制缺失
5	0x7700	011 10111 0000 0000	冲突缺失
6	0xa700	101 00111 0000 0000	命中
7	0xc600	110 00110 0000 0000	强制缺失
8	0xa60f	101 00110 0000 1111	冲突缺失

Cache 块大小为 $2^8=256$ 字节。

2. 对 1 中分析出的块大小，如果采用 2 路组相联，以图分析每次访问发生冲突缺失和命中的情况。

参考解答：缓存共 32 块，分 16 组。内存地址的高 8 位为内存块号，其中的低 4 位（红色部分）为对应的缓存组号（即索引位），余下的高 4 位为缓存的 Tag。

	16 进制地址流	2 进制地址流	访问情况
1	0x779f	0111 0111 1001 1111	强制缺失
2	0xa7ff	1010 0111 1111 1111	强制缺失（已满）
3	0xb7dd	1011 0111 1101 1101	强制缺失（覆盖 1）
4	0xa6ff	1010 0110 1111 1111	强制缺失
5	0x7700	0111 0111 0000 0000	冲突缺失（覆盖 2）

6	0xa700	1010	0111	0000 0000	冲突缺失（覆盖 3）
7	0xc600	1100	0110	0000 0000	强制缺失（已满）
8	0xa60f	1010	0110	0000 1111	命中

3. 对 1 中分析出的块大小，如果采用全相联，以图分析每次访问发生冲突缺失和命中的情况。

参考解答：内存地址的高 8 位为内存块号，同时也是缓存的 Tag。

	16 进制地址流	2 进制地址流	访问情况
1	0x779f	0111 0111 1001 1111	强制缺失
2	0xa7ff	1010 0111 1111 1111	强制缺失
3	0xb7dd	1011 0111 1101 1101	强制缺失
4	0xa6ff	1010 0110 1111 1111	强制缺失
5	0x7700	0111 0111 0000 0000	命中
6	0xa700	1010 0111 0000 0000	命中
7	0xc600	1100 0110 0000 0000	强制缺失
8	0xa60f	1010 0110 0000 1111	命中

三、在一个采用字节编址的系统中，缓存容量为 256 字节，按 16 字节分块，内存和缓存之间采用 4 路组相联映像。如果需要将某个内存块调入缓存，在选择缓存块时优先使用该组块号最小的空闲块。设某时刻的缓存状态(不含数据位)如下：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	0	x	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	0	x

从此时刻起，系统需要连续访问 10 个内存地址：0x31, 0x43, 0x51, 0x78, 0x67, 0x81, 0x92, 0x55, 0x130, 0x171。请按下列表格形式分析在下列各种情况下的命中情况(用 1 表示命中, 0 表示未命中)并画出访问结束后的缓存状态：

内存地址	内存块号	组号	标记	命中情况

说明：此题并未说明内存地址为 12 位长度！

1. 采用 FIFO 替换策略;

参考解答:

内存地址	内存块号	组号	标记	命中情况
0x31	3	3	0	0
0x43	4	0	1	1
0x51	5	1	1	0
0x78	7	3	1	1
0x67	6	2	1	0
0x81	8	0	2	0
0x92	9	1	2	0
0x55	5	1	1	1
0x130	19	3	4	0
0x171	23	3	5	0

访问结束后缓存的状态:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	4	1	5	1	3	1	0

分析如下: 每块 16 字节, 缓存共分 4 组, 每组 4 块。内存地址的低 4 位为块内地址, 高位部分为块号。内存块号的低 2 位为对应的缓存组号 (即索引位), 高位部分为标记位。

地址 0x31: 内存块号为 0x3, 映射到 3 号组, 标记为 0

检索缓存:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	0	x	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	0	x

检索结果: 未找到。

缺失处理: 到内存找到相应的块, 复制到 3 号组 3 号块, 同时更新 3 号组 3 号块的有效

效位和标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	0	x	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

地址 **0x43**：内存块号为 0x4，映射到 0 号组，标记为 1

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	0	x	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：找到

地址 **0x51**：内存块号为 0x5，映射到 1 号组，标记为 1

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	0	x	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：未找到

缺失处理：到内存找到相应的块，复制到 1 号组 0 号块，同时更新 1 号组 0 号块的有效位和标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x

1	1	1	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

地址 **0x78**: 内存块号为 0x7, 映射到 3 号组, 标记为 1

检索缓存:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	1	1	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果: 找到

地址 **0x67**: 内存块号为 0x6, 映射到 2 号组, 标记为 1

检索缓存:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	1	1	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果: 未找到

缺失处理: 到内存找到相应的块, 复制到 2 号组 1 号块, 同时更新 2 号组 1 号块的有效位和标记位:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	1	1	0	x	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

地址 **0x81**: 内存块号为 0x8, 映射到 0 号组, 标记为 2

检索缓存:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	1	1	0	x	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果: 未找到

缺失处理: 到内存找到相应的块, 复制到 0 号组 2 号块, 同时更新 0 号组 2 号块的有效位和标记位:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	0	x	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

地址 0x92: 内存块号为 0x9, 映射到 1 号组, 标记为 2

检索缓存:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	0	x	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果: 未找到

缺失处理: 到内存找到相应的块, 复制到 1 号组 1 号块, 同时更新 1 号组 1 号块的有效位和标记位:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x

1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

地址 **0x55**：内存块号为 0x5，映射到 1 号组，标记为 1

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：找到

地址 **0x130**：内存块号为 0x13，映射到 3 号组，标记为 4

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：未找到

缺失处理：到内存找到相应的块，复制到 3 号组 0 号块，同时更新 3 号组 0 号块的标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	4	1	2	1	3	1	0

地址 **0x171**：内存块号为 0x17，映射到 3 号组，标记为 5

检索缓存:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	4	1	2	1	3	1	0

检索结果: 未找到

缺失处理: 到内存找到相应的块, 复制到 3 号组 1 号块, 同时更新 3 号组 1 号块的标记位:

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	4	1	5	1	3	1	0

2. 采用 LRU 替换策略, 假设: 0 号组中, 最近用到的是 0 号块; 3 号组中, 最近用到的是 1 号块, 最近最久未用的是 0 号块.

参考解答:

内存地址	内存块号	组号	标记	命中情况
0x31	3	3	0	0
0x43	4	0	1	1
0x51	5	1	1	0
0x78	7	3	1	1
0x67	6	2	1	0
0x81	8	0	2	0
0x92	9	1	2	0
0x55	5	1	1	1
0x130	19	3	4	0
0x171	23	3	5	0

访问结束后缓存的状态:

组号	组内 0 号块	组内 1 号块	组内 2 号块	组内 3 号块
----	---------	---------	---------	---------

	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	5	1	4	1	0

分析如下：过程同前，只是缺失处理时如果需要替换的话，替换策略变了而已！

地址 0x31：内存块号为 0x3，映射到 3 号组，标记为 0

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	0	x	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	0	x

检索结果：未找到。

缺失处理：到内存找到相应的块，复制到 3 号组 3 号块，同时更新 3 号组 3 号块的有效位和标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	0	x	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

同时 3 号组的使用情况为 0、2、1、3（0 号块最近最久未用，3 号块刚刚用过）

地址 0x43：内存块号为 0x4，映射到 0 号组，标记为 1

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	0	x	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x

3	1	1	1	2	1	3	1	0
---	---	---	---	---	---	---	---	---

检索结果：找到！

同时 0 号组的使用情况为 1、0（1 号块最近最久未用，0 号块刚刚用过）

地址 0x51：内存块号为 0x5，映射到 1 号组，标记为 1

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	0	x	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：未找到

缺失处理：到内存找到相应的块，复制到 1 号组 0 号块，同时更新 1 号组 0 号块的有效位和标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	1	1	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

同时 1 号组的使用情况为 0（0 号块刚刚用过）

地址 0x78：内存块号为 0x7，映射到 3 号组，标记为 1

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	1	1	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：找到

同时 3 号组的使用情况为 2、1、3、0（2 号块最近最久未用，0 号块刚刚用过）

地址 0x67：内存块号为 0x6，映射到 2 号组，标记为 1

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	1	1	0	x	0	x	0	x
2	1	2	0	x	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：未找到

缺失处理：到内存找到相应的块，复制到 2 号组 1 号块，同时更新 2 号组 1 号块的有效位和标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	1	1	0	x	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

同时 2 号组的使用情况为 0、1（0 号块最近最久未用，1 号块刚刚用过）

地址 0x81：内存块号为 0x8，映射到 0 号组，标记为 2

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	0	x	0	x
1	1	1	0	x	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：未找到

缺失处理：到内存找到相应的块，复制到 0 号组 2 号块，同时更新 0 号组 2 号块的有效位和标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	0	x	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

同时 0 号组的使用情况为 1、0、2（1 号块最近最久未用，2 号块刚刚用过）

地址 0x92：内存块号为 0x9，映射到 1 号组，标记为 2

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	0	x	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：未找到

缺失处理：到内存找到相应的块，复制到 1 号组 1 号块，同时更新 1 号组 1 号块的有效位和标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

同时 1 号组的使用情况为 0、1（1 号块刚刚用过）

地址 0x55：内存块号为 0x5，映射到 1 号组，标记为 1

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：找到

同时 1 号组的使用情况为 1、0（0 号块刚刚用过）

地址 0x130：内存块号为 0x13，映射到 3 号组，标记为 4

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	3	1	0

检索结果：未找到

缺失处理：到内存找到相应的块，根据 LRU 策略复制到 3 号组 2 号块，同时更新 3 号组 2 号块的标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	4	1	0

同时 3 号组的使用情况为 1、3、0、2（1 号块最近最久未用，2 号块刚刚用过）

地址 0x171：内存块号为 0x17，映射到 3 号组，标记为 5

检索缓存：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位

0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	2	1	4	1	0

检索结果：未找到

缺失处理：到内存找到相应的块，根据 LRU 策略复制到 3 号组 1 号块，同时更新 3 号组 1 号块的标记位：

组号	组内 0 号块		组内 1 号块		组内 2 号块		组内 3 号块	
	有效位	标记位	有效位	标记位	有效位	标记位	有效位	标记位
0	1	1	1	3	1	2	0	x
1	1	1	1	2	0	x	0	x
2	1	2	1	1	0	x	0	x
3	1	1	1	5	1	4	1	0

同时 3 号组的使用情况为 3、0、2、1（3 号块最近最久未用，1 号块刚刚用过）