

5.5 对一个 64 位地址的直接映射 cache 的设计，地址的以下位用于访问 cache。

标签	索引	偏移
63~10	9~5	4~0

5.5.1 [5] < 5.3 > cache 块大小为多少（以字为单位）？

5.5.2 [5] < 5.3 > cache 块有多少个？

5.5.3 [5] < 5.3 > 这种 cache 实现所需的总位数与数据存储器之间的比率是多少？

下表记录了从上电开始 cache 访问的字节地址。

地址												
十六进制	00	04	10	84	E8	A0	400	1E	8C	C1C	B4	884
十进制	0	4	16	132	232	160	1024	30	140	3100	180	2180

5.5.4 [20] < 5.3 > 对每一次访问，列出：它的标签、索引和偏移；指出命中还是失效；替换了哪个字节（如果有的话）。

5.5.5 [5] < 5.3 > 命中率是多少？

5.5.6 [5] < 5.3 > 列出 cache 的最终状态，每个有效表项表示为 < 索引，标签，数据 > 的记录。例如：

< 0, 3, Mem[0xC00]-Mem[0xC1F] >

5.5.1 每块包含4个字。8byte的字对应偏移中的3位，偏移总共是5位，剩余2位是块内偏移，也就是每个块共有 $2^2=4$ 个字。

如果标注了字大小是4字节，最终结果是8个字，也可以。

5.5.2 索引为5位， $2^5=32$ 块

5.5.3 1.215

Cache: $32 \text{ lines} * 4 \text{ words/block} * 8 \text{ bytes/word} = 1024 \text{ bytes} = 8192 \text{ bits}$

数据: 54bit的tag, 1bit的valid。 $8192 + 54 * 32 + 1 * 32 = 9952 \text{ bits}$

比率 $= 9952 / 8192 = 1.215$

反过来比值也没有扣分

5.5.4

Byte Address	Binary Address	Tag	Index	Offset	Hit/Miss	Bytes Replaced
0x00	0000 0000 0000	0x0	0x00	0x00	M	
0x04	0000 0000 0100	0x0	0x00	0x04	H	
0x10	0000 0001 0000	0x0	0x00	0x10	H	
0x84	0000 1000 0100	0x0	0x04	0x04	M	
0xe8	0000 1110 1000	0x0	0x07	0x08	M	
0xa0	0000 1010 0000	0x0	0x05	0x00	M	
0x400	0100 0000 0000	0x1	0x00	0x00	M	0x00-0x1F
0x1e	0000 0001 1110	0x0	0x00	0x1e	M	0x400-0x41F
0x8c	0000 1000 1100	0x0	0x04	0x0c	H	
0xc1c	1100 0001 1100	0x3	0x00	0x1c	M	0x00-0x1F
0xb4	0000 1011 0100	0x0	0x05	0x14	H	
0x884	1000 1000 0100	0x2	0x04	0x04	M	0x80-0x9f

5.5.5 $4/12=33.3\%$

5.5.6

<index, tag, data>

<0, 3, Mem[0xC00]-Mem[0xC1F]>

<4, 2, Mem[0x880]-Mem[0x89f]>

<5, 0, Mem[0x0A0]-Mem[0x0Bf]>

<7, 0, Mem[0x0e0]-Mem[0x0ff]>

5.11 本题研究不同 cache 设计的效果，特别是将组相联 cache 与 5.4 节中的直接映射 cache 进行比较。

有关这些练习，请参阅下面显示的字地址序列：

0x03, 0xb4, 0x2b, 0x02, 0xbe, 0x58, 0xbf, 0x0e, 0x1f,
0xb5, 0xbf, 0xba, 0x2e, 0xce

5.11.1 [10] < 5.4 > 绘制块大小为 2 字、总容量为 48 字的三路组相联 cache 的组织结构图。图中应有类似于图 5-18 的样式，还应该清楚地显示标签和数据字段的宽度。

5.11.2 [10] < 5.4 > 从 5.11.1 中记录 cache 的行为。假设 cache 使用 LRU 替换策略。对于每一次 cache 访问，确定：

- 二进制字地址。
- 标签。
- 索引。
- 偏移。
- 访问会命中还是失效。
- 在处理访问后，cache 每一路中有哪些标签。

5.11.3 [5] < 5.4 > 绘制块大小为 1 字、总容量为 8 字的全相联 cache 的组织结构图。图中应有类似于图 5-18 的样式，还应该清楚地显示标签和数据字段的宽度。

5.11.4 [10] < 5.4 > 从 5.11.3 中记录 cache 的行为。假设 cache 使用 LRU 替换策略。对于每一次 cache 访问，确定：

- 二进制字地址。
- 标签。
- 索引。
- 偏移。
- 访问会命中还是失效。
- 在处理访问后，cache 中的内容。

5.11.5 [5] < 5.4 > 绘制块大小为 2 字、总容量为 8 字的全相联 cache 的组织结构图。图中应有类似于图 5-18 的样式，还应该清楚地显示标签和数据字段的宽度。

5.11.6 [10] < 5.4 > 从 5.11.5 中记录 cache 的行为。假设 cache 使用 LRU 替换策略。对于每一次 cache 访问，确定：

- 二进制字地址。
- 标签。
- 索引。
- 偏移。
- 访问会命中还是失效。
- 在处理访问后，cache 中的内容。

(1) 问题：替换过程出现错误；少写漏写；

(2) 大家Cache替换过程格式不统一，细节处没法一一检查，建议大家复习时对照答案查看是否存在问题。

5.11.7 [10] < 5.4 > 将替换策略改为 MRU (最多最常使用) 策略，再次完成 5.11.6。

5.11.8 [15] < 5.4 > 将替换策略改为最优替换策略 (造成最低失效率的替换策略)，再次完成 5.11.6。

5.11.1 地址划分正确，图简单画下。

1. ↵

Tag↵	Index↵	Offset↵	↵
7,6,5,4↵	3,2,1↵	0↵	↵

2. ↵

本题index和offset位数正确就可以，总位数没关系。
因为本题没有给出字节数，所以Offset是按照字寻址的。考试时如果题目给出字节数，还是要按照字节数进行寻址的。

5.11.2 3路组相联 (LRU)

Word Address	Binary Address	Tag	Index	Offset	Hit/Miss	Way 0	Way 1	Way 2
0x03	0000 0011	0x0	1	1	M	T(1)=0		
0xb4	1011 0100	0xb	2	0	M	T(1)=0 T(2)=b		
0x2b	0010 1011	0x2	5	1	M	T(1)=0 T(2)=b T(5)=2		
0x02	0000 0010	0x0	1	0	H	T(1)=0 T(2)=b T(5)=2		
0xbe	1011 1110	0xb	7	0	M	T(1)=0 T(2)=b T(5)=2 T(7)=b		
0x58	0101 1000	0x5	4	0	M	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5		
0xbf	1011 1111	0xb	7	1	H	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5		
0x0e	0000 1110	0x0	7	0	M	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=0	
0x1f	0001 1111	0x1	7	1	M	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	H	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	H	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	M	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	M	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	M	T(1)=0 T(2)=b T(5)=2 T(7)=b T(4)=5	T(7)=2 T(5)=b	T(7)=c

5.11.3

3.↵

Tag↵
7-0↵

Word Address	Binary Address	Tag	Hit/Miss	Contents
0x03	0000 0011	0x03	M	3
0xb4	1011 0100	0xb4	M	3, b4
0x2b	0010 1011	0x2b	M	3, b4, 2b
0x02	0000 0010	0x02	M	3, b4, 2b, 2
0xbe	1011 1110	0xbe	M	3, b4, 2b, 2, be
0x58	0101 1000	0x58	M	3, b4, 2b, 2, be, 58
0xbf	1011 1111	0xbf	M	3, b4, 2b, 2, be, 58, bf
0x0e	0000 1110	0x0e	M	3, b4, 2b, 2, be, 58, bf, e
0x1f	0001 1111	0x1f	M	b4, 2b, 2, be, 58, bf, e, 1f
0xb5	1011 0101	0xb5	M	2b, 2, be, 58, bf, e, 1f, b5
0xbf	1011 1111	0xbf	H	2b, 2, be, 58, e, 1f, b5, bf
0xba	1011 1010	0xba	M	2, be, 58, e, 1f, b5, bf, ba
0x2e	0010 1110	0x2e	M	be, 58, e, 1f, b5, bf, ba, 2e
0xce	1100 1110	0xce	M	58, e, 1f, b5, bf, ba, 2e, ce

5. ↵

Tag↵	Offset↵
7-1↵	0↵

5.11.6 2路全相联，共8字（LRU）

Word Address	Binary Address	Tag	Offset	Hit/Miss	Contents
0x03	0000 0011	0x01	1	M	[2,3]
0xb4	1011 0100	0x5a	0	M	[2,3], [b4,b5]
0x2b	0010 1011	0x15	1	M	[2,3], [b4,b5], [2a,2b]
0x02	0000 0010	0x01	0	H	[b4,b5], [2a,2b], [2,3]
0xbe	1011 1110	0x5f	0	M	[b4,b5], [2a,2b], [2,3], [be, bf]
0x58	0101 1000	0x2c	0	M	[2a,2b], [2,3], [be, bf], [58, 59]
0xbf	1011 1111	0x5f	1	H	[2a,2b], [2,3], [58, 59], [be, bf]
0x0e	0000 1110	0x07	0	M	[2,3], [58, 59], [be, bf], [e,f]
0x1f	0001 1111	0x0f	1	M	[58, 59], [be, bf], [e,f], [1e,1f]
0xb5	1011 0101	0x5a	1	M	[be, bf], [e,f], [1e,1f], [b4, b5]
0xbf	1011 1111	0x5f	1	H	[e,f], [1e,1f], [b4, b5], [be, bf]
0xba	1011 1010	0x5d	0	M	[1e,1f], [b4, b5], [be, bf], [ba, bb]
0x2e	0010 1110	0x17	0	M	[b4, b5], [be, bf], [ba, bb], [2e, 2f]
0xce	1100 1110	0x67	0	M	[be, bf], [ba, bb], [2e, 2f], [ce,cf]

5.11.7 2路全相联，共8字（MRU）

Word Address	Binary Address	Tag	Offset	Hit/Miss	Contents
0x03	0000 0011	0x01	1	M	[2,3]
0xb4	1011 0100	0x5a	0	M	[2,3], [b4,b5]
0x2b	0010 1011	0x15	1	M	[2,3], [b4,b5], [2a,2b]
0x02	0000 0010	0x01	0	H	[b4,b5], [2a,2b], [2,3]
0xbe	1011 1110	0x5f	0	M	[b4,b5], [2a,2b], [2,3], [be, bf]
0x58	0101 1000	0x2c	0	M	[b4,b5], [2a,2b], [2,3], [58, 59]
0xbf	1011 1111	0x5f	1	M	[b4,b5], [2a,2b], [2,3], [be, bf]
0x0e	0000 1110	0x07	0	M	[b4,b5], [2a,2b], [2,3], [e, f]
0x1f	0001 1111	0x0f	1	M	[b4,b5], [2a,2b], [2,3], [1e, 1f]
0xb5	1011 0101	0x5a	1	H	[2a,2b], [2,3], [1e, 1f], [b4,b5]
0xbf	1011 1111	0x5f	1	M	[2a,2b], [2,3], [1e, 1f], [be, bf]
0xba	1011 1010	0x5d	0	M	[2a,2b], [2,3], [1e, 1f], [ba, bb]
0x2e	0010 1110	0x17	0	M	[2a,2b], [2,3], [1e, 1f], [2e, 2f]
0xce	1100 1110	0x67	0	M	[2a,2b], [2,3], [1e, 1f],, [ce, cf]

5.11.8

Word Address	Binary Address	Tag	Offset	Hit/Miss	Contents
0x03	0000 0011	0x01	1	M	[2,3]
0xb4	1011 0100	0x5a	0	M	[2,3], [b4,b5]
0x2b	0010 1011	0x15	1	M	[2,3], [b4,b5], [2a,2b]
0x02	0000 0010	0x01	0	H	[2,3], [b4,b5], [2a,2b]
0xbe	1011 1110	0x5f	0	M	[2,3], [b4,b5], [2a,2b], [be, bf]
0x58	0101 1000	0x2c	0	M	[58,59], [b4,b5], [2a,2b], [be, bf]
0xbf	1011 1111	0x5f	1	H	[58,59], [b4,b5], [2a,2b], [be, bf]
0x0e	0000 1110	0x07	0	M	[e,f], [b4,b5], [2a,2b], [be, bf]
0x1f	0001 1111	0x0f	1	M	[1e,1f], [b4,b5], [2a,2b], [be, bf]
0xb5	1011 0101	0x5a	1	H	[1e,1f], [b4,b5], [2a,2b], [be, bf]
0xbf	1011 1111	0x5f	1	H	[1e,1f], [b4,b5], [2a,2b], [be, bf]
0xba	1011 1010	0x5d	0	M	[1e,1f], [b4,b5], [ba,bb], [be, bf]
0x2e	0010 1110	0x17	0	M	[1e,1f], [b4,b5], [2e,2f], [be, bf]
0xce	1100 1110	0x67	0	M	[1e,1f], [b4,b5], [ce,cf], [be, bf]

替换不唯一， 这里给出两个例子。

地址 (HEX)	地址 (BINARY)	标签	偏移	命中/失效	内容
03	0000 0011	01	1	失效	[2,3]
B4	1011 0100	5A	0	失效	[2,3], [B4,B5]
2B	0010 1011	15	1	失效	[2,3], [B4,B5], [2A,2B]
02	0000 0010	01	0	命中	[2,3], [B4,B5], [2A,2B]
BE	1011 1110	5F	0	失效	[2,3], [B4,B5], [2A,2B], [BE,BF]
58	0101 1000	2C	0	失效	[58,59], [B4,B5], [2A,2B], [BE,BF]
BF	1011 1111	5F	1	命中	[58,59], [B4,B5], [2A,2B], [BE,BF]
0E	0000 1110	07	0	失效	[0E,0F], [B4,B5], [2A,2B], [BE,BF]
1F	0001 1111	0F	1	失效	[1E,1F], [B4,B5], [2A,2B], [BE,BF]
B5	1011 0101	5A	1	命中	[1E,1F], [B4,B5], [0E,0F], [BE,BF]
BF	1011 1111	5F	1	命中	[1E,1F], [B4,B5], [0E,0F], [BE,BF]
BA	1011 1010	5D	0	失效	[1E,1F], [B4,B5], [BA,BB], [BE,BF]
2E	0010 1110	17	0	失效	[2E,2F], [B4,B5], [BA,BB], [BE,BF]
CE	1100 1110	67	0	失效	[2E,2F], [CE,CF], [BA,BB], [BE,BF]

4.24 一个4体低位交叉的存储器,假设存取周期为 T ,CPU 每隔 $1/4$ 存取周期启动一个存储体,试问依次访问 64 个字需多少个存取周期?

答: 只有访问第一个字需一个存取周期。从第二个字开始,每隔 $1/4$ 存取周期即可访问一个字,因此,依次访问 64 个字需:

$$\text{存取周期个数} = (64-1) \times (1/4)T + T = (63/4 + 1)T = 15.75 + 1 = 16.75T$$