计组第八次作业答案

不作答疑，答案供参考

5.1，5.2，5.4，5.5，5.6，5.10，5.11

5.1

5.1.1

2

5.1.2

I,J 以及 B[I][0]

5.1.3

A[I][J]

5.1.4

使用 Matlab 时为 32004 个，使用 C 语言时为 32008 个。

代码引用了矩阵 A 中的 8*8000=64000 个整数。由于每 16 字节块存储两个整数，所以供需要 32000 个块。

另外，代码还引用了矩阵 B 的每八行的第一个元素。Matlab 以列主序存储矩阵数据，因此所有八个整数是连续的，可以放在四个块中。而 C 语言以行主序存储矩阵数据，因此每行的第一个元素位于不同的块中。

5.1.5

A(J, I)和 B[I][0]。

5.1.6

I，J 和 B[I][0]。

5.2

5.2.1

Word Address	Binary Address	Tag	Index	Hit/Miss
0x03	0000 0011	0	3	M
0xb4	1011 0100	b	4	M
0x2b	0010 1011	2	b	M
0x02	0000 0010	0	2	M
0xbf	1011 1111	b	f	M
0x58	0101 1000	5	8	M
0xbe	1011 1110	b	e	M
0x0e	0000 1110	0	e	M
0xb5	1011 0101	b	5	M
0x2c	0010 1100	2	c	M
0xba	1011 1010	b	a	M
0xfd	1111 1101	f	d	M

5.2.2

Word Address	Binary Address	Tag	Index	Hit/Miss	
0x03	0000 0011	0	1	1	M
0xb4	1011 0100	b	2	0	M
0x2b	0010 1011	2	5	1	M
0x02	0000 0010	0	1	0	H
0xbf	1011 1111	b	7	1	M
0x58	0101 1000	5	4	0	M
0xbe	1011 1110	b	6	0	H
0x0e	0000 1110	0	7	0	M
0xb5	1011 0101	b	2	1	H
0x2c	0010 1100	2	6	0	M
0xba	1011 1010	b	5	0	M
0xfd	1111 1101	f	6	1	M

5.2.3

Word Address	Binary Address	Tag	Cache 1		Cache 2		Cache 3	
			index	hit/miss	index	hit/miss	index	hit/miss
0x03	0000 0011	0x00	3	M	1	M	0	M
0xb4	1011 0100	0x16	4	M	2	M	1	M
0x2b	0010 1011	0x05	3	M	1	M	0	M
0x02	0000 0010	0x00	2	M	1	M	0	M
0xbf	1011 1111	0x17	7	M	3	M	1	M
0x58	0101 1000	0x0b	0	M	0	M	0	M
0xbe	1011 1110	0x17	6	M	3	H	1	H
0x0e	0000 1110	0x01	6	M	3	M	1	M
0xb5	1011 0101	0x16	5	M	2	H	1	M
0x2c	0010 1100	0x05	4	M	2	M	1	M
0xba	1011 1010	0x17	2	M	1	M	0	M
0xfd	1111 1101	0x1F	5	M	2	M	1	M

缓存 1 的未命中率 = 100%
缓存 1 的总周期数 = 12 × 25 + 12 × 2 = 324
缓存 2 的未命中率 = 10/12 = 83%
缓存 2 的总周期数 = 10 × 25 + 12 × 3 = 286
缓存 3 的未命中率 = 11/12 = 92%
缓存 3 的总周期数 = 11 × 25 + 12 × 5 = 335
因此缓存 2 提供了最佳的性能。

5.4

这是可能的。要实现一个直接映射缓存，我们只需要一个函数，该函数将地址作为输入并产生 10 位的输出。虽然以这种方式实现缓存是可能的，但并不明确这种实现方式是否会带来好处。这是因为这种缓存将需要更大的标记字段，而且可能会有更多的冲突未命中。

5.5

5.5.1

每个缓存块由四个 8 字节的单词组成，总偏移量为 5 位。这 5 位中的 3 位是字偏移量（即进入 8 字节单词的偏移量），剩下的两位是块偏移量。两位允许我们枚举 2^2=4 个单词。

5.5.2

有五位索引位，则缓存中有 2^5=32 行。

5.5.3

比率是 1.21。缓存总共存储 32 行*每块 4 个单词*每词 8 字节=1024 字节=8192 比特。
除了数据之外，每一行还包括 54 位标签位和 1 位有效位。因此，所需的总比特数=8192+5432+132=9952 比特。

5.5.4

见下页

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]>

5.6

5.6.1

L1 缓存具有较低的写未命中惩罚，而 L2 缓存具有较高的写未命中惩罚。在 L1 和 L2 缓存之间设置一个写缓冲区可以隐藏 L2 缓存的写未命中延迟。当 L2 缓存替换脏块时，写缓冲区特别有益，因为新块可以在脏块物理写入内存之前先被读取进来。

5.6.2

当 L1 发生写未命中时，该字直接写入 L2，无需将其所在块调入 L1 缓存。如果这导致了 L2 的未命中，那么必须将该块调入 L2 缓存，可能需要替换一个脏块，而这个脏块必须先被写回内存。

5.6.3

在 L1 写未命中之后，该块将驻留在 L2 中，但不在 L1 中。随后对该同一块的读取未命中将要求 L2 中的块先被写回内存，然后传输到 L1，并在 L2 中失效。

5.10

5.10.1

P1	1.515 GHz
P2	1.11 GHz

5.10.2

P1	6.31ns	9.56 个周期
P2	5.11ns	5.68 个周期

对于 P1，所有内存访问至少需要一个周期（来访问 L1）。8%的内存访问额外需要 70 纳秒的主存访问时间。这相当于 70/0.66 = 106.06 个周期。然而，我们不能分割周期，因此必须向上取整到 107 个周期。因此，平均内存访问时间为

1+0.08*107 = 9.56 个周期，或者 6.31ps。

对于 P2，主存访问需要 70 纳秒。这相当于 70/0.90 = 77.78 个周期。由于我们不能分割周期，必须向上取整到 78 个周期。因此，平均内存访问时间为 1+0.06*78 = 5.68 个周期，或者 6.11ps。

5.10.3

P1	12.64CPI	8.34ns/周期
P2	7.36CPI	6.63ns/周期

对于 P1，每条指令至少需要一个周期。另外，所有指令中有 8%在指令缓存中未命中，会产生 107 个周期的延迟。此外，36%的指令涉及数据访问，其中 8%的数据访问会发生缓存未命中，这又会增加额外的 107 个周期。因此，计算公式为：1 + 0.08*107 + 0.36*0.08*107 = 12.64 个周期，考虑到时钟周期为 0.66 皮秒，每条指令需要的时间为 12.64 * 0.66 ps = 8.34 纳秒。

按照同样的逻辑分析，可以得出 P2 的每周期指令数(CPI)为 7.36，平均每个指令仅需 6.63 纳秒。

5.10.4

AMAT = 9.85 周期	更差

L2 访问需要九个周期（5.62/0.66，向上取整到最接近的整数）。所有内存访问至少需要一个周期。8%的内存访问在 L1 缓存中未命中，会进行 L2 访问，这需要九个周期。而 L2 访问中有 95%未命中，需要额外 107 个周期的内存查找时间。因此，计算公式为：1+0.08*(9 + 0.95*107) = 9.85 个周期。

5.10.5

13.04

可以通过以下公式计算答案：AMAT + %内存访问*(AMAT - 1)。利用此公式，可以计算出带有 L2 缓存的 P1 的 CPI 为 9.85 * 0.36 * 8.85=13.04。

5.10.6

由于两种版本的 P1 具有相同的时钟周期时间和内存指令百分比，我们只需关注 AMAT。我们希望带有 L2 的 AMAT 小于仅带有 L1 的 AMAT

则有：1 + 0.08*(9 + m*107)<9.56。于是，m < 0.916。

5.10.7

我们希望 P1 每条指令的平均时间少于 6.63 纳秒，则有(CPI_P1 * 0.66) < 6.63。于是，CPI_P1 < 10.05。

易知 CPI_P1 = AMAT_P1 + 0.36(AMAT_P1-1)，则有 AMAT_P1+0.36(AMAT_P1-1)<10.05。于是，AMAT_P1<7.65。

最后，有以下方程：

$$1 + 0.08(9 + m*107)<7.65$$

解得，m<0.693，所以未命中率最多可以达到 69.3%。

5.11

本题难度过大，答案过长，批阅时会酌情给分。

5.11.1

缓存中的每一行将总共有六个字（三个组中各有两个）。因此，总共有 48/6=8 行。

5.11.2

T(x)表示索引 x 处的标签。

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

无答案（标答这么写的）

5.11.4

由于这个缓存是全相联的，并且每个块只包含一个字，所以不存在索引和偏移量。因此，字地址实质上就是标签。

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, e, 1f, b5, 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, 23, ce

5.11.5

无答案（标答这么写的）

5.11.6

因为这个缓存是全相联的，所以没有索引。（内容按照数据被访问的顺序展示。顺序并不表示实际的物理位置。）

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 1111	0x5a	1	M	[be,bf], [e,f], [1e,1f], [b4,b5]
0xbf	1011 1111	0x5f	0	H	[e,f], [1e,1f], [b4,b5], [ba,bb]
0xba	1011 1010	0x5d	0	M	[1e,1f], [b4,b5], [b3,bf], [ba,bb]
0x2e	0010 1110	0x17	0	M	[b4,b5], [b3,bf], [ba,bb], [2e,2f]
0xce	1100 1110	0x67	0	M	[be,bf], [ba,bb], [2e,2f], [ce,cf]

5.11.7、5.11.8 见下

5.11.7 (Contents shown in the order the data were accessed. Order does not imply physical location.)

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	H	[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 1111	0x5a	1	H	[2a,2b], [2,3], [1e,1f], [b4,b5]
0xbf	1011 1111	0x5f	1	M	[2a,2b], [2,3], [1e,1f], [ba,bb]
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 Because this cache is fully associative, there is no index.

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 1111	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]