第五章 存储器

5.1

(1) 4个

(2) B[I][0]， I, J

(3) A[I][J]

(4) 需要31986个16字节的cache块，计算如下：

$$
\begin{aligned}
Num\_block &= \frac{total\_int}{cache\_wordsize} \\
&= \frac{8 * 8000 * 2 - 8 * 8 + 8}{4} \\
&= 31986
\end{aligned}
$$

(5) B[I][0], I , J

(6) A[J][I]

5.2

(1)

十进制地址	二进制地址	标志	索引	访问
3	0000 0011	0000	0011	×
180	1011 0100	1011	0100	×
43	0010 1011	0010	1011	×
2	0000 0010	0000	0010	×
191	1011 1111	1011	1111	×
88	0101 1000	0101	1000	×
190	1011 1110	1011	1110	×
14	0000 1110	0000	1110	×
181	1011 0101	1011	0101	×
44	0010 1100	0010	1100	×
186	1011 1010	1011	1010	×
253	1111 1101	1111	1101	×

(2)

十进制地址	二进制地址	标志	索引	访问
3	0000_001_1	0000	001	×
180	1011_010_0	1011	010	×
43	0010_101_1	0010	101	×
2	0000_001_0	0000	001	√
191	1011_111_1	1011	111	×
88	0101_100_0	0101	100	×
190	1011_111_0	1011	111	√
14	0000_111_0	0000	111	×
181	1011_010_1	1011	010	√
44	0010_110_0	0010	110	×
186	1011_101_0	1011	101	×
253	1111_110_1	1111	110	×

(3)

C1: blocksize = 1 word, blockbit = 0, blocks = 8, indexbit = 3 bit
C2: blocksize = 2 word, blockbit = 1, blocks = 4, indexbit = 2 bit
C1: blocksize = 4 word, blockbit = 2, blocks = 2, indexbit = 1 bit

十进制地址	二进制地址	标志	c1-索引	C1-访问	C2-索引	C2-访问	C3-索引	C3-访问
3	0000 0011	0	3	×	1	×	0	×
180	1011 0100	22	4	×	2	×	1	×
43	0010 1011	5	3	×	1	×	0	×
2	0000 0010	0	2	×	1	×	0	×
191	1011 1111	23	7	×	3	×	1	×
88	0101 1000	11	0	×	0	×	0	×
190	1011 1110	23	6	×	3	√	1	√
14	0000 1110	1	6	×	3	×	1	×
181	1011 0101	22	5	×	2	√	1	×
44	0010 1100	5	4	×	2	×	1	×
186	1011 1010	23	2	×	1	×	0	×
253	1111 1101	31	5	×	2	×	1	×

第一问：按照缺失率，C2块和C3块设计效果更好

第二问：
C1块：$CPI = 25 * 12 + 2 * 12 = 324$
C2块：$CPI = 25 * 10 + 3 * 12 = 286$
C3块：$CPI = 25 * 11 + 5 * 12 = 335$

故C2块设计更好

(4)

第一问：

块大小(blocksize) = 2 word * 4 = 8 Byte，则blockbit = 3，故需要3位地址；
块数量(blocks) = 32 KB / 8 B = 2^{12}，则 indexbit = 12，故需要12位地址表示；
标志地址位数(tagbit) = 32 - 12 - 3 = 17 位；
cache总位数 = 数据容量 + 标志容量 = 32KB + 2^{12} * (1 + 17) / 8 Byte = 41 KB

第二问：
根据题意，给定cache的总位数大小不超过41KB，$blocksize = 16\text{word} = 64\text{B}$。设块数量为$blocks$，则

$$
\begin{aligned}
&blockbit = \log_2(blocksize) = 6 \\
&tagbit = 32 - \log_2(blocks) - blockbit \\
&\text{数据容量} = blocks \times blocksize\ (\text{Byte}) \\
&\text{标志容量} = blocks \times (tagbit + 1)/8\ (\text{Byte}) \\
&\text{数据容量} + \text{标志容量} \leq 41\text{KB}
\end{aligned} \tag{1}
$$

得到$blocks \leq 2^9$.

第三问：
块大小，越大，会导致命中时间增加、缺失代价增加，块的数量减少进而导致块冲突和块替换的频率增加，故访问速度反而慢了。

(5)

利用二路组相联cache，优点在于可以降低缺失率，但缺点在于增加了命中时间。

(6)

可以。但是经过异或运算，容易丢失块地址[31:27]这五位信息，需要增加更多的标志位来验证地址信息。

5.3

(1) $\text{block\_size} = 2^5\text{Byte} = 8\text{word}$

(2) $\text{block\_num} = 2^{\text{index}} = 2^5 = 32$

(3)

$$
\begin{aligned}
r &= \frac{\text{data\_size}}{\text{data\_size} + (\text{valid} + \text{tag\_size}) * \text{block\_num}} \\
&= \frac{2^5 * 32\ \text{Byte}}{2^5 * 32\ \text{Byte} + (1 + 22) * 32 \div 8\ \text{Byte}} \\
&= 0.9175627240143369
\end{aligned}
$$

(4) 3个块被替换

(5) $\text{命中率} = \frac{3}{12} \times 100\% = 25\%$

(6) cache有效项最终状态如下表（位地址 --> 字节地址）

索引	标记	数据
1	1	mem[1024]
1	3	mem[16]
11	0	mem[176]
8	2	mem[2176]
14	0	mem[224]
10	0	mem[160]

5.5

(1)

设该地址为字节地址，根据题意，地址流将是一连串偶数，其中cache块大小为32Byte，故在0~31的地址流中，共16条地址访问，只有第一条缺失，以此类推，接下来每16条偶数地址，都将是第一条缺失，故缺失率为1/16。

在该地址中，cache的缺失率与cache的大小和原始数据工作集变化无关，与cache的块大小和地址流访问形式有关。

按照题目的偶数地址流，每一个块都将第一次访问，故为强制缺失

(2)

块大小为16Byte时，缺失率 = 1/8；
块大小为64Byte时，缺失率 = 1/32；
块大小为128Byte时，缺失率 = 1/64；

空间局部性。

(3)

由题意，只有第一次访问cache有缺失，之后所有的都将命中，缺失率分子为1，分母过大，可近似视为0.

(4)

最佳块大小为8

$\text{blocksize} = 8$：$0.04 * 20 * 8 = 6.4$;
$\text{blocksize} = 16$：$0.03 * 20 * 16 = 9.6$;
$\text{blocksize} = 32$：$0.02 * 20 * 32 = 12.8$;
$\text{blocksize} = 64$：$0.015 * 20 * 64 = 19.2$;
$\text{blocksize} = 128$：$0.01 * 20 * 128 = 25.6$;

(5)

最佳块大小为32

$\text{blocksize} = 8$：$0.04 * (24 + 8) = 1.28;$
$\text{blocksize} = 16$：$0.03 * (24 + 16) = 1.2;$
$\text{blocksize} = 32$：$0.02 * (24 + 32) = 1.12;$
$\text{blocksize} = 64$：$0.015 * (24 + 64) = 1.3198;$
$\text{blocksize} = 128$：$0.01 * (24 + 128) = 1.52;$

(6)

当缺失代价恒定，则缺失率最小，cache块大小最佳，即128

5.6

（1）时钟频率：

$f_{\text{P1}} = \frac{1}{0.66*10^{-9}} = 1.5151\text{Ghz}$

$f_{\text{P2}} = \frac{1}{0.90*10^{-9}} = 1.1112\text{Ghz}$

（2）平均存储器访问时间

$AMAT_{\text{P1}} = 0.66 + 8\% * 70 = 6.2600\text{ns}$

$AMAT_{\text{P2}} = 0.90 + 6\% * 70 = 5.1000\text{ns}$

（3）P2处理器更快。

$CPI_{\text{P1}} = 1 + 36\% * 8\% * 70 * 1.5151 = 4.0544$

$CPI_{\text{P2}} = 1 + 36\% * 6\% * 70 * 1.1112 = 2.6801$

（4）增加二级Cache，AMAT更好了

$AMAT_{\text{P1}} = 0.66 + 8\% * 5.62 + 5\% * 70 = 4.6096\text{ns} < 6.2600\text{ns}$

（5）

$CPI_{\text{P1}} = 1 + 36\% * (8\% * 5.62 + 5\% * 70) * 1.5151 = 3.1543$

（6）

处理器时间，$CPI_{\text{P1}} * 0.66 = 2.0818\text{ns} < CPI_{\text{P2}} * 0.90 = 2.4121\text{ns}$，故处理器P1更快。

解如下不等式，得到$miss_rate_{\text{P2}} < 0.0469$，即P2处理器的缺失率需要小于4.69%才能匹配P1的性能。

$$1 * 0.90 + 36\% * miss\_rate_{\text{P2}} * 70 < 2.0818 \tag{2}$$

5.7

(1)

$\text{group\_num} = 24 \div (3 \times 2) = 4$，$\text{group\_block} = 3$，故需要index 2位表示.

十进制地址	二进制地址	索引	标志	访问	组1	组2	组3
3	00000_01_1	1	0	×	mem[2,**3**]		
180	10110_10_0	2	22	×	mem[**180**,181]		
43	00101_01_1	1	5	×	mem[42,**43**]		
2	00000_01_0	1	0	√	mem[**2**,3]		
191	10111_11_1	3	23	×	mem[190,**191**]		
88	01011_00_0	0	11	×	mem[**88**,89]		
190	10111_11_0	3	23	√	mem[**190**,191]		
14	00001_11_0	3	1	×	mem[190,191]	mem[**14**,15]	
181	10110_10_1	2	22	√	mem[180,**181**]		
44	00101_10_0	2	5	×	mem[**44**,45]		
186	10111_01_0	1	23	×	mem[42,43]	mem[**186**,187]	
253	11111_10_1	2	31	×	mem[44,45]	mem[252,**253**]	

(2)

$\text{group\_num} = 1, \text{group\_block} = 8$,不需要索引位

十进制地址	二进制地址 32bit	索引 3bit	标志 28bit	访问	组内
3	0000 0011	无	3	×	group[0]
180	1011 0100	无	180	×	group[1]
43	0010 1011	无	43	×	group[2]
2	0000 0010	无	2	×	group[3]
191	1011 1111	无	191	×	group[4]
88	0101 1000	无	88	×	group[5]
190	1011 1110	无	190	×	group[6]
14	0000 1110	无	14	×	group[7]
181	1011 0101	无	181	×	group[0]
44	0010 1100	无	44	×	group[1]
186	1011 1010	无	186	×	group[2]
253	1111 1101	无	253	×	group[3]

(3)

$\text{group\_block} = 8 \div 2 = 4, \text{group\_num} = 1$，全相联不需要索引位，块数据需要1位，标志位为32-2-1 = 29位。

LRU替换原则：数据越靠后，越新；缺失率 = 10/12 = 83.33%

十进制地址	二进制地址 32bit	索引 3bit	标志 28bit	访问	数据
3	0000001_1	无	1	×	2,3
180	1011010_0	无	90	×	2,3; 180,181
43	0010101_1	无	21	×	2,3; 180,181;42,43
2	0000001_0	无	1	√	180,181;42,43; 2,3
191	1011111_1	无	95	×	180,181;42,43; 2,3; 190,191
88	0101100_0	无	44	×	42,43; 2,3; 190,191; 88,89
190	1011111_0	无	95	√	42,43; 2,3; 88,89;190,191;
14	0000111_0	无	7	×	2,3; 88,89;190,191; 14,15
181	1011010_1	无	90	×	88,89;190,191; 14,15; 180,181
44	0010110_0	无	22	×	190,191; 14,15; 180,181; 44,45
186	1011101_0	无	93	×	14,15; 180,181; 44,45; 186,187
253	1111110_1	无	126	×	180,181; 44,45; 186,187; 252, 253

MRU替换原则：缺失率 = 11/12 = 91.67%

十进制地址	二进制地址	索引	标志	访问	数据
3	0000001_1	无	1	×	2,3
180	1011010_0	无	90	×	2,3; 180,181
43	0010101_1	无	21	×	2,3; 180,181;42,43
2	0000001_0	无	1	√	180,181;42,43; 2,3
191	1011111_1	无	95	×	180,181;42,43; 2,3; 190,191
88	0101100_0	无	44	×	180,181;42,43; 2,3; 88,89
190	1011111_0	无	95	×	180,181;42,43; 2,3; 190,191
14	0000111_0	无	7	×	180,181;42,43; 2,3; 14,15
181	1011010_1	无	90	×	180,181;42,43; 2,3; 180,181
44	0010110_0	无	22	×	180,181;42,43; 2,3; 44,45
186	1011101_0	无	93	×	180,181;42,43; 2,3; 186,187
253	1111110_1	无	126	×	180,181;42,43; 2,3; 252, 253

最好的情况下，为（1）中的方案，此时cache缺失率为 75%。

(4)

根据题意，主存时钟周期数 = 100 ns * 2 Ghz = 200

处理器总的CPI计算如下：

访主存CPI	只有一级cache	一个直接映射的二级cache	一个8路组相联的二级cache
200	1.5+0.07*200=15.5	1.5+0.07*12+0.035*200=9.34	1.5+0.07*28+0.015*200=6.46
400	1.5+0.07*400=29.5	1.5+0.07*12+0.035*400=16.34	1.5+0.07*28+0.015*400=9.46
100	1.5+0.07*100=8.5	1.5+0.07*12+0.035*100=5.84	1.5+0.07*28+0.015*100=4.96

(5)

改进后的处理器总的CPI = 1.5+0.07*12+0.035*50+0.013*200 = 6.69 < 9.34（原处理器总的CPI），故这种设计能提供更好的性能；

优点：减少了缺失代价

缺点：一旦出线写缺失等情况，写直达和写回需要更大的代价；且占用较多的缓存资源。

(6)

相比于直接映射的二级cache：

$$1.5+7\%\times 50+4\%\times(1-0.7\%)^{\frac{\text{cache\_size}}{512}}\times 200\geq 9.34$$

得到$\text{cache_size}\geq 44575.027$，即cache容量至少为44576 KiB.

相比于8路组相联的二级cache：

$$1.5+7\%\times 50+4\%\times(1-0.7\%)^{\frac{\text{cache\_size}}{512}}\times 200\geq 6.46$$

得到$\text{cache_size}\geq 123980.406$，即cache容量至少为123981KiB.

5.8

（1）$MTBF=MTTF+MTTR=365*3+1=1096$天。

（2）$可用性=\frac{MTTF}{MTTF+MTTR}\times 100\%=\frac{1095}{1095+1}\times 100\%=99.91\%$

（3）当MTTR接近0时，可用性接近1，替换时间成本为0的情形不合理。

（4）当MTTR非常高时，可用性将下降；不一定，因为当MTTF远大于MTTR时，该器件的可用性也可以达到很高的值。