

姓名: 李雅帆

学号: 2213041

5.1

5.1.1 16字节 = 128位 = 4个32位整数

5.1.2 I, J

5.1.3 按行存储: A[1][1]

按列存储: A[1][1]

5.1.4 $8 \times 800 \div 4 \times 2 - 8 \times 8 \div 4 + 8000 \div 4 = 3596$

5.1.5 I, J

5.1.6 A(J, 1)

5.2

5.2.1

Word	Address	Binary Address	Tag	Index	Hit / Miss
3		0000 0011	0	3	M
180		1011 0100	11	4	M
43		0010 1011	2	11	M
2		0000 0010	0	2	M
191		1011 1111	11	15	M
88		0101 1000	5	8	M
190		1011 1110	11	14	M
14		0000 1110	0	14	M
181		1011 0101	11	5	M
44		0010 1100	2	12	M
186		1011 1010	11	10	M
253		1111 1101	15	13	M

5.2.2

Word	Address	Binary Address	Tag	Index	Hit/Miss
3		0000 0011	0	1	M
180		1011 0100	11	2	M
43		0010 1011	2	5	M
2		0000 0010	0	1	H
191		1011 1111	11	7	M
88		0101 1000	5	4	M
190		1011 1110	11	7	H
14		0000 1110	0	7	M
181		1011 0101	11	2	H
44		0010 1100	2	6	M
186		1011 1010	11	5	M
253		1111 1101	15	6	M

5.2.3

Word	Address	Binary Address	Tag	cache1		cache2		cache3	
				index	hit/miss	index	hit/miss	index	hit/miss
3		0000 0011	0	3	M	1	M	0	M
180		1011 0100	22	4	M	2	M	1	M
43		0010 1011	5	3	M	1	M	0	M
2		0000 0010	0	2	M	1	M	0	M
191		1011 1111	23	7	M	3	M	1	M
88		0101 1000	11	0	M	0	M	0	M
190		1011 1110	23	6	M	3	H	1	H
14		0000 1110	1	6	M	3	M	1	M
181		1011 0101	22	5	M	2	H	1	M
44		0010 1100	5	4	M	2	M	1	M
186		1011 1010	23	2	M	1	M	0	M
253		1111 1101	31	5	M	2	M	1	M

5.4 可以直接映射 cache. 但需要确保对异或结果进行适当的限制, 以确保它在 cache 大小范围内。

5.5

5.5.1 word size = 4 bytes

block size = 32 bytes / word size = 8

5.5.2 $2^5 = 32$ (块)

5.5.3 1个8字节大小的 block 需要 22 位的标志位

$$1 + 22 \div (32 \times 8) = 1.086$$

5.5.4.

Word Address	Tag	Index	偏移	Hit / Miss	替换
00 (0000 000)	0	00000	00000	M	无
04 (0000 0100)	0	00000	00100	H	无
10 (0000 10000)	0	00000	10000	H	无
84 (10000 0100)	0	00100	00100	M	无
E8 (11101 000)	0	00111	01000	M	无
A0 (1010 0000)	0	00101	00000	M	无
400 (0100 000000)	1	00000	00000	M	替 0 → 1
1E (0001 1110)	0	00000	11110	M	替 1 → 0
8C (1000 1100)	0	00100	01100	H	无
C1C (1100 0011 00)	3	00000	11100	M	替 0 → 3
B4 (1011 0100)	0	00101	10100	H	无
884 (1000 0000 100)	2	00100	00100	M	替 0 → 2

5.5.5 命中率 $\frac{4}{12} \approx 33\%$

索引	标记	Data
000001	0001	mem[1024]
000001	0011	mem[16]
001011	0000	mem[170]
001000	0010	mem[216]
001110	0000	mem[224]
001010	0000	mem[160]

5.6

5.6.1 L_1 高速缓存的写失误惩罚较低, 而 L_2 高速缓存的写失误惩罚较高。

在 L_1 和 L_2 高速缓存之间有一个写缓冲区, 可以隐藏 L_2 高速缓存的写失误延迟。

当替换一个脏块时, L_2 高速缓存将受益于写缓冲区, 因为新块将在脏块被物理写入内存之前被写入。

5.6.2 L_1 写缺失时, 字直接被写到 L_2 , 而不把它的块带入 L_1 缓存, 如果这导致了 L_2 的缺失, 它的块必须被带入 L_2 缓存, 可能会替换掉一个必须先写入内存的脏块。

5.6.3 L_1 缺失后, 该块留在 L_2 中, 但不在 L_1 中。随后对同一区块的读取失误将要求把 L_2 中的区块写回内存, 转移到 L_1 中, 并在 L_2 中失效。

5.10.

5.10.1 $P_1 = 1.52 \text{ GHz}$ $P_2 = 1.11 \text{ GHz}$

5.10.2 $P_1: 0.08 \times 70 + 0.66 = 6.26 \text{ ns}$ 9.48 cycles

$P_2: 0.06 \times 70 + 0.9 = 5.046 \text{ ns}$ 5.6 cycles

5.10.3 $P_1: 12.64 \text{ CPI}$ 8.34 ns

$P_2: 7.36 \text{ CPI}$ 6.63 ns P_2 更快

5.10.4 $0.92 \times 0.66 + 0.08 \times 0.95 \times 70 + 0.08 \times 0.95 \times 5.62 \approx 6.50 \text{ ns}$
AMAT 更差了

5.10.5 13.04

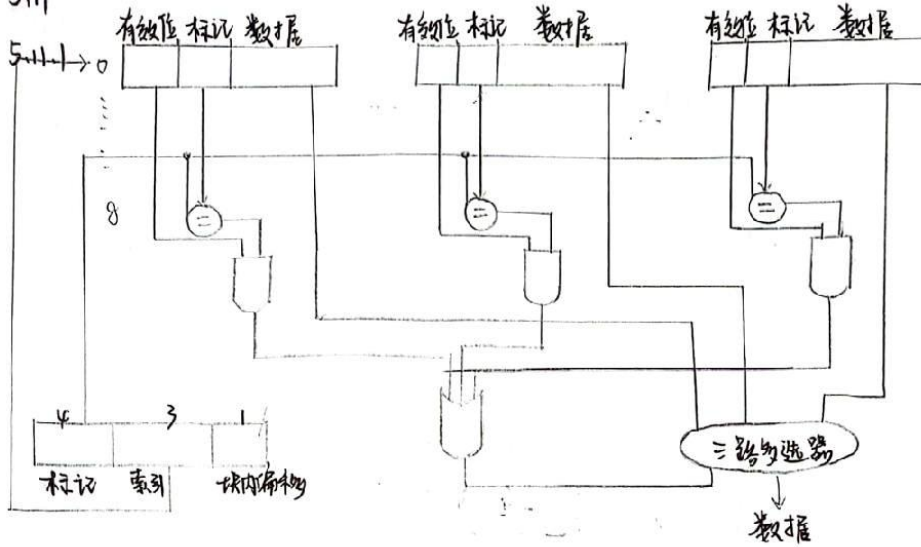
5.10.6 $P_1 \text{ AMAT} = 0.66 + 0.08 \times 70 = 6.26 \text{ ns}$

$P_2 \text{ AMAT} = 0.9 + 0.06 \times (5.62 + 0.95 \times 70) = 5.23 \text{ ns}$

$5.23 = 0.66 \text{ ns} + \text{MR} \times 70 \text{ ns}$

$\text{MR} = 6.5\%$

5.11



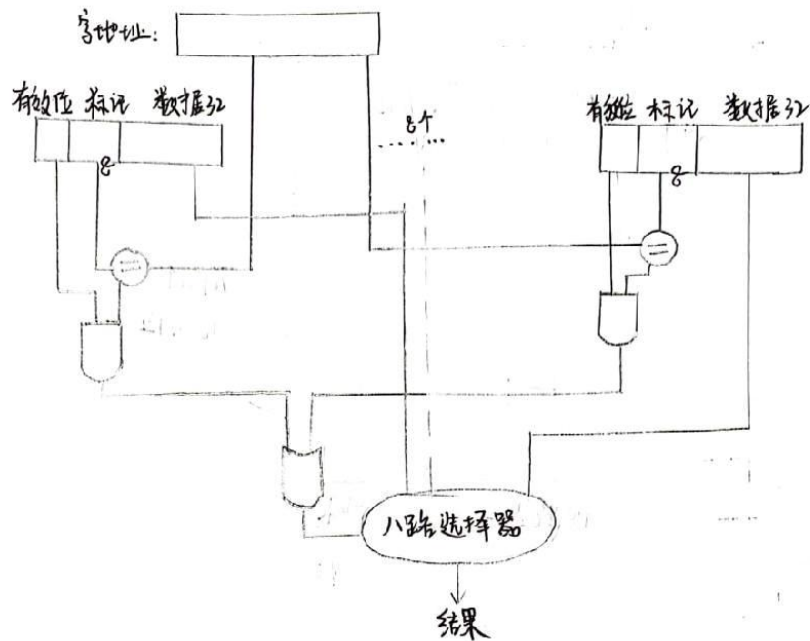
5.11.2

word address	标记	索引	偏移	Hit/Miss
0x03 (0000 0011)	0000	00	1	M
0x04 (0000 0100)	0001	00	0	M
0x0b (0000 1011)	0010	01	1	M
0x0c (0000 1100)	0011	01	0	H
0x0e (0000 1110)	1011	11	0	M
0x58 (0101 1000)	0101	10	0	M
0xb7 (1011 1111)	1011	11	1	H
0x0e (0000 1110)	0000	11	0	M
0x1f (0001 1111)	0001	11	1	M
0xb5 (1011 0101)	1011	01	1	H
0xb7 (1011 1111)	1011	11	1	M
0xbac (1011 1111)	1011	10	0	M
0x2e (0010 1110)	0010	11	0	M
0x0e (0000 1110)	1110	11	0	M

最终可知每个访问被处理后 cache 每路中的标记为:

0	无	001: 0000	111: 0000
1	0000	010: 10 11	111: 0001
2	1011	101: 0010	010: 不变
3	无	001: 不变	111: 1011
4	0011	111: 1011	101: 1011
5	1011	100: 0101	111: 0010
6	无	111: 不变	110: 1110
7	1110		

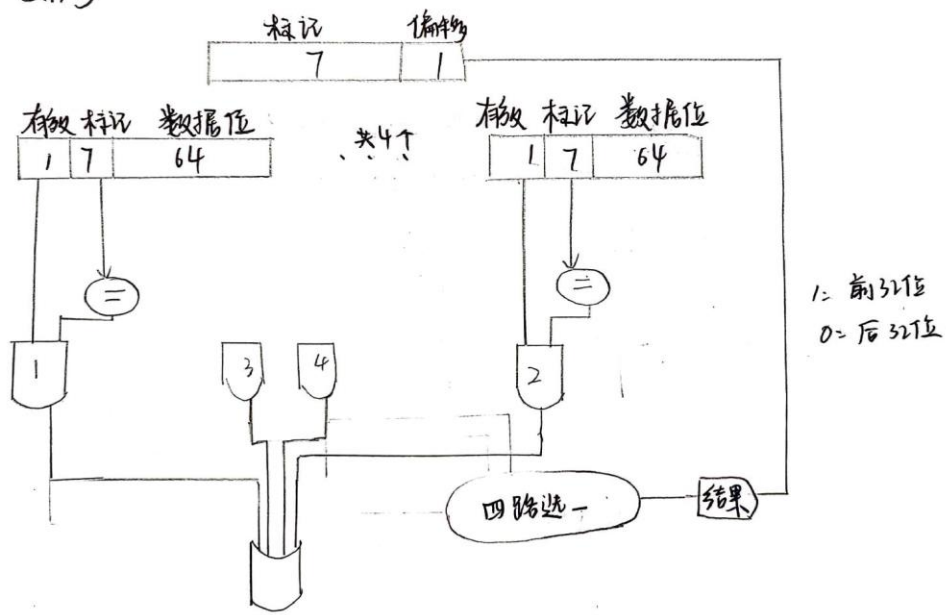
5.1.3



5.1.4

word Address	Tag	Hit/Miss	新增块 (无索引/偏码)
0x03 (000 0011)	0x03	M	+ 0x03
0xb4 (1011 1011)	0xb4	M	+ 0xb4
0x2b (0010 1011)	0x2b	M	+ 0x2b
0x02 (0000 0010)	0x02	M	+ 0x02
0xbe (1011 1110)	0xbe	M	+ 0xbe
0x58 (0101 1000)	0x58	M	+ 0x58
0xbf (1011 1111)	0xbf	M	+ 0xbf
0x0e (0000 1110)	0x0e	M	+ 0x0e
0xf (1000 1111)	0xf	M	0xf 替 0x03
0xb5 (1011 1011)	0xb5	M	0xb5 替 0xb4
0xbf (1011 1111)	0xbf	H	H
0xba (1011 1110)	0xba	M	0xba 替 0x2b
0x2e (0010 1110)	0x2e	M	0x2e 替 0x02
0xce (1100 1110)	0xce	M	0xce 替 0xbe

5.11.5



5.11.6

	标记	一致性	Hit / Miss
0x03	0000001	1	M
0xb4	1011010	0	M
0x2b	0010101	1	M
0x02	0000000	0	H
0xbe	0111111	0	M
0x3b	0101100	0	M
0xbf	1011111	1	H
0x0e	0000111	0	M
0x1f	0001111	1	M
0xb5	1011010	1	H
0xbf	1011111	1	M
0xba	1011010	0	M
0x2e	0010111	0	M
0xce	1100111	0	M

(LRV)

5.11.7 ① 0000001 ② 0000001 101010

③ 0000001 101010 0010101

④ 101010 001010 0000001 ∈ (Hit)

⑤ 101010 001010 0000001 101111

⑥ 101010 001010 0000001 0101100

⑦ 101010 001010 0000001 101111

⑧ 101010 001010 0000001 0000111

⑨ 101010 001010 0000001 0001111

⑩ 001010 0000001 000111 101010 ∈ (Hit)

⑪ 001010 0000001 000111 101111

⑫ 001010 0000001 000111 101101

⑬ 001010 0000001 000111 001011

⑭ 001010 0000001 000111 1100111

5.11.8: 即LRV