

王宇涵 202200400253 第四章作业

4.3① 体现在缓存-主存 和 主存-辅存两个存储层次上。

② 分缓存-主存是为了解决 CPU 速度不匹配的问题

分主存-辅存是为了解决存储容量的问题。

③ 对于缓存-主存，计算机将 CPU 近期要用的信息调入缓存，CPU 直接从缓存中获取信息，从而提高访问速度。因为缓存容量小，故而不断将主存内容调入缓存，替换缓存内容，由硬件自动完成。

对于主存-辅存，辅存容量比主存大得多，可用于存放大量暂时未用到的信息。当 CPU 需要这些信息时，计算机将辅存内容调入主存，使 CPU 直接访问，由硬件和操作系统共同完成。

4.5 存储器的带宽是单位时间内存储器存取的信息量。

带宽为: $32 \text{ bit} / 200 \times 10^{-9} \text{ s} = 16 \times 10^7 \text{ bit/s} = 20 \text{ MB/s}$

4.8 (SRAM) 静态 RAM: 用触发器存储信息, 只要不断电, 信息就不会丢失。

不需要刷新, 但集成度低, 功耗大, 适合做 cache。

(DRAM) 动态 RAM: 用电容存储信息, 为保持信息须每隔 2 ms 内对高电平电容进行刷新, 因此需设计刷新电路, 较复杂, 但集成度高, 功耗低, 适合做主存。

4.9 刷新: 对 DRAM 定期进行的全部重写过程。

原因: 电容漏电, 引起 DRAM 存储消息衰减, 需及时补充重写, 故应及时定期刷新。

方法: 集中刷新, 分散刷新, 异步刷新。

集中刷新: 在最大的刷新间隔时间内, 集中安排一段时间进行刷新。

分散刷新: 在每个读写周期插入一个刷新周期, 无 CPU 访存死区。

异步刷新: 每次刷新一行, 2 ms 之内刷新完全即行, 是前两者折中。

4.14

(1) $2^{18} \times 8 \text{ bit} = 256 \text{ K} \cdot 8 \text{ bit} = 256 \text{ KB}$

(2) $256 \text{ KB} / 32 \text{ KB} = 8 (\text{个})$

(3) $32 \text{ KB} / 2 \text{ KB} = 16 (\text{片})$

256 K 8位

32 K 8位

(4) $16 \times 8 = 128 (\text{片})$

4 K 4位

(5) 存储器需 8 个模块板, 故需 3 位, 每个板内 16 片 RAM, 每 2 片 RAM 一组,

位扩展为 $4 \text{ K} \times 8 \text{ 位}$, 共 8 组, 故需 3 位, 片内地址共 $4 \text{ K} = 2^{12}$, 故需 12 位。

则共需 $12 + 3 + 3 = 18$ 位选择模块板。

3	3	12
板地址	片地址	片内地址

4.15
1)

(2¹²)

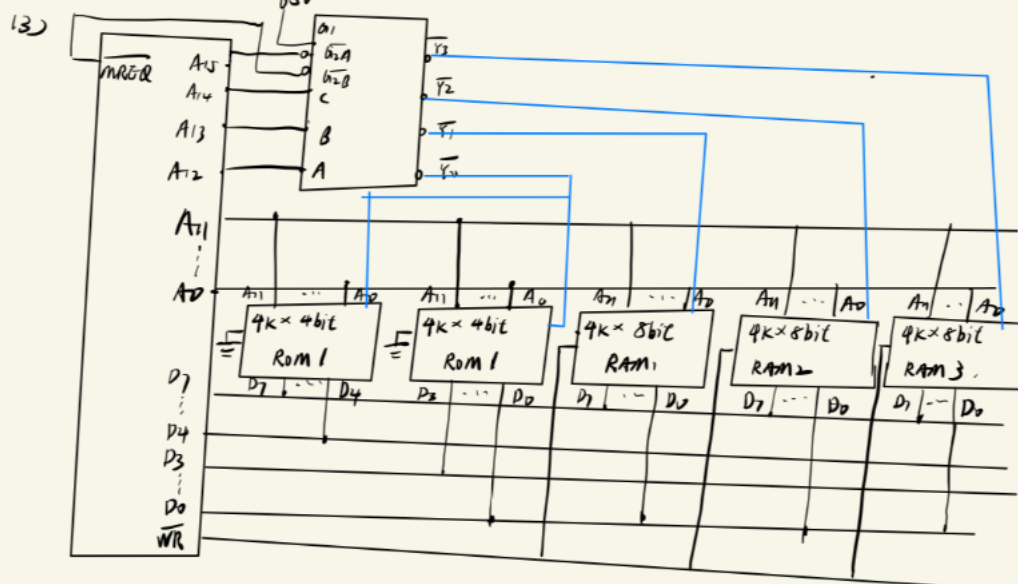
0	0000	0000	0000	0000	
4095	0000	1111	1111	1111	> Rom. 4K×8bit
4096	0001	0000	0000	0000	
16383	0011	1111	1111	1111	> RAM. 12K×8bit

2) 地址分配: 2ROM: 4K×4bit. 3RAM 4K×8bit.

(2¹²)

地址划分为

	A ₁₅ ~A ₁₂	A ₁₁ ~A ₈	A ₇ ~A ₄	A ₃ ~A ₀	
0	0000	0000	0000	0000	
4095	0000	1111	1111	1111	> Rom.
4096	0001	0000	0000	0000	
	0001	1111	1111	1111	> RAM ₁
	0010	0000	0000	0000	
	0010	1111	1111	1111	> RAM ₂
	0011	0000	0000	0000	
	0011	1111	1111	1111	> RAM ₃
16383	0011	1111	1111	1111	



4.17. (1) 1100, 1101 有双位. 4位 设为 $a_1 a_2 a_3 a_4$

$$2^K \geq n+K+1 \rightarrow K=3. \text{ 设为 } b_1 b_2 b_3$$

则 汉明码为 $b_1 b_2 a_1 b_3 a_2 a_3 a_4$

$$b_1 = a_1 \oplus a_2 \oplus a_4$$

$$b_2 = a_1 \oplus a_3 \oplus a_4$$

$$b_3 = a_2 \oplus a_3 \oplus a_4$$

	1	0	0	1	0	0
b_1	b_2	a_1	b_3	a_2	a_3	a_4
001	010	011	100	101	110	111
g_1	✓		✓		✓	✓
g_2		✓	✓			✓
g_3				✓	✓	✓

例 1100: $b_1 b_2 b_3 = 011$ 汉明码为 0111100

1101: $b_1 b_2 b_3 = 100$ 汉明码为 1010101

4.18 汉明码 0 0 1 0 $g_1 g_2 g_3$

$$(1) 1100100 \quad g_1 = b_1 \oplus a_1 \oplus a_2 \oplus a_4$$

110 第6位即 a_3 出错

$$(2) 1100111 \quad g_2 = b_2 \oplus a_1 \oplus a_3 \oplus a_4$$

111 第7位即 a_4 出错

$$g_3 = b_3 \oplus a_2 \oplus a_3 \oplus a_4$$

4.28

(1) Cache 2K字 块大小为4字 则共有 $\frac{2K}{4} = 2^9 = 512$ 块.
地址格式:

Cache 块地址 (9位)	字块内地址 (2位)
----------------	------------

(2) 主存 256K字 共 $256 \times 2^{10} = 2^{18}$ 18位

$$\text{主存共 } \frac{256K}{4} = 64K \text{ 块} = 2^{16} \text{ 位}$$

主存块标记 (7位)	Cache 块地址 (9位)	字块内地址 (2位)
------------	----------------	------------

(3) Cache 共 $\frac{512}{4} = 2^7$ 组

主存块标记 (9位)	Cache 组号 (7位)	字块内地址 (2位)
------------	---------------	------------

(4)

主存块标记 (16位)	字块内地址 (2位)
-------------	------------

(5). 存储字长 32位 (4 byte).

则主存容量: $256k \times 4 = 2^{20} B$ (20位).

Cache容量: $2k \times 4 = 2^{13} B$ (13位).

直接映射:

主存块标记 (17位)	Cache块号 (9位)	字块内地址 (4位)
-------------	--------------	------------

四路组映射

主存块标记 (9位)	Cache组号 (7位)	字块内地址 (4位)
------------	--------------	------------

全相联映射

主存块标记 (16位)	字块内地址 (4位)
-------------	------------

4.29.

命中率: $\frac{4800}{4800+200} = 96\%$

平均访问时间: $96\% \times 30ns + 4\% \times 150ns = 34.8ns$

效率: $\frac{30ns}{34.8ns} = 86.2\%$

辱时间: 150ns 则提高了 $\frac{150}{34.8} - 1 = 3.31$ 倍

4.31 主存地址划分 $1MB = 2^{20} \text{ byte} = 20$ 位

Cache共有 $16KB / 4 \times 4B = 2^{10}$ 块 块大小为 $2^4 B$.

主存块标记 (6位)	Cache块号 (10位)	字块内地址 (4位)
------------	---------------	------------

例 ABCDEH:

即映射到 Cache 中 1111001101 | 1110 位置

1010 1011 1100 1101 | 1110

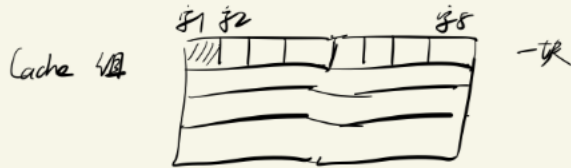
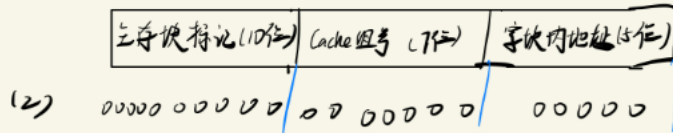
即 BCDEH 位置

4.32

(1) 主存类 $4 \times 2^{20} = 2^{22}$ byte 共 22 位 块大小 $8 \times 4 = 2^5$ B 5 位

Cache 类 $\frac{16KB}{8 \times 4B} = 2^9$ 块 共 $\frac{2^9}{4} = 2^7$ 组 共 7 位

主存划分



每次读第一个字都装入一块，共装入了 $\lceil \frac{90}{8} \rceil = 12$ 块，以后重复操作均命中，则命中率为 $\frac{8 \times 90 - 12}{8 \times 90} = 0.984$ 。

(3) 设 Cache 为 7，主存为 60

则提高了

$$\frac{60}{(0.984 \times 60 + 0.016 \times 60)} - 1 = 454 \text{ 倍}。$$