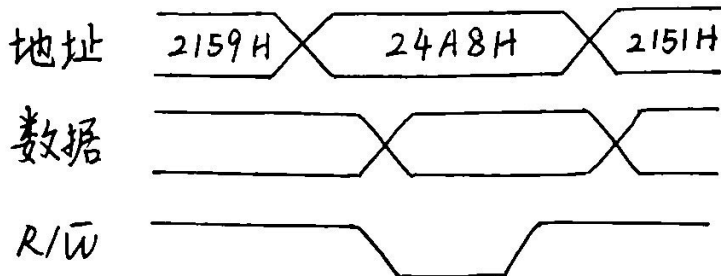




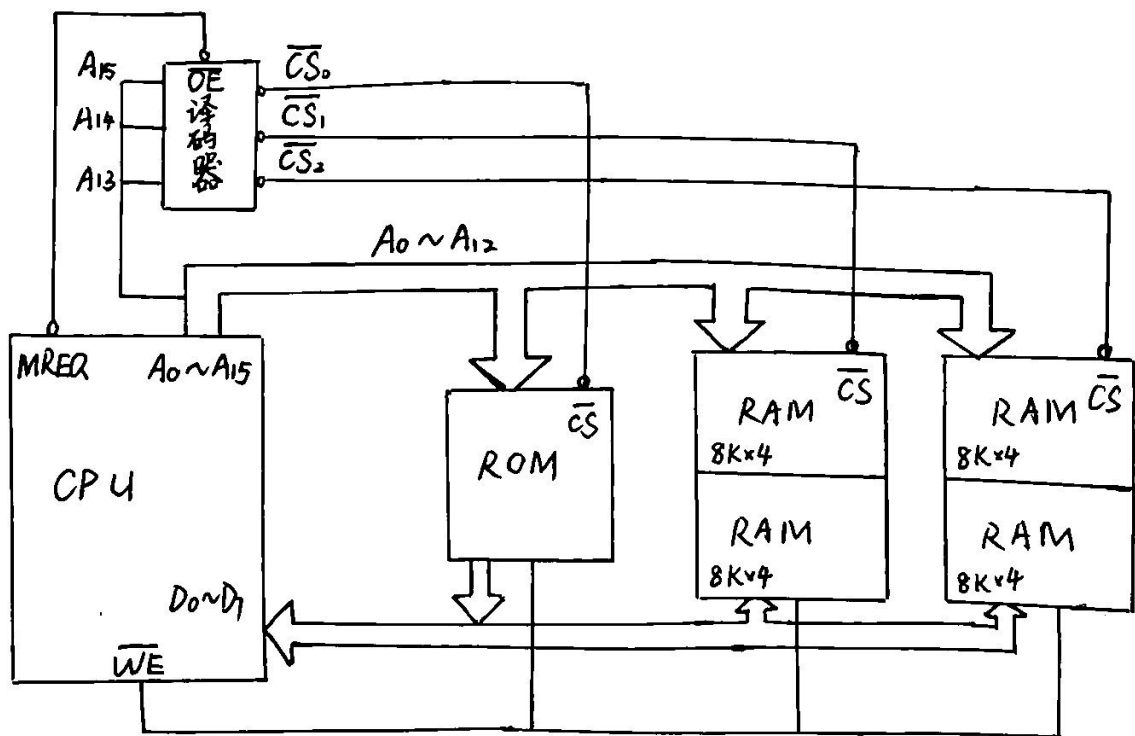
4.4 写入存储器时地址不应发生变化



4.5 (1) $(512K \times 16) / (16K \times 1) = 128 (\text{个})$

(2) $\frac{2ms}{128} = 15.6 \mu s (\text{不超过})$

4.6



4.7 (1) SRAM无需刷新, 但DRAM需要刷新以避免丢失数据;

(2) SRAM访问速度较DRAM更高;

(3) SRAM较DRAM成本高、功耗大。



4.11 地址线共需 $\log_2(1G) = 30$ (位), 从高到低位记为 $A_{29} \sim A_0$

4体交叉存储方案共分4个存储体, 其中每个存储体有32个DRAM芯片
 每个存储体中的32个芯片RAS端相连, 所有芯片的地址端输入均为 $A_2 \sim A_{29}$ 。

低位地址 A_1, A_0 输入译码器后与RAS相乘形成 $\overline{RAS_3}, \overline{RAS_2}, \overline{RAS_1}, \overline{RAS_0}$ 分别对应4个存储体用于交叉选择。

4.12 (1) $16 \times (1 + 4 + 1) = 96$ 个时钟周期

(2) 最后会余3个传输周期 $4 \times 6 + 3 = 27$ 个时钟周期

(3) 与4字宽类似, 余下3个传输周期 $4 \times 6 + 3 = 27$ 个时钟周期

7.1

层次	介质	
(寄存器组)	电路	
Cache	SRAM	
主存	DRAM	
辅存	磁表面存储	

小

↓

大

容量

高

↓

低

存取速度

7.2 主存: 随机按字存取;

辅存: 通过直接内存访问(DMA)成组传送。



7.3 设置Cache:

平均访存时间为

$$\begin{aligned} & (0.98 \times 10\text{ns} + 0.02 \times (10+100)\text{ns}) + \frac{1}{5} \times (0.95 \times 10\text{ns} + 0.05 \times (10+100)\text{ns}) \\ &= (9.8 + 2.2)\text{ns} + (9.5 + 5.5)/5\text{ns} \\ &= 15\text{ns} \end{aligned}$$

无Cache:

平均访存时间为 $100\text{ns} + \frac{1}{5} \times 100\text{ns} = 120\text{ns}$

\therefore 提高了 $\frac{120\text{ns}}{15\text{ns}} = 8\text{倍}$

7.4 若采用哈佛结构, 则有Cache时平均访存时间为 $9.8 + 2.2 = 12\text{ns}$

提高了 $\frac{120\text{ns}}{12\text{ns}} = 10\text{倍}$

7.5 (1) 因为主存容量为2MB, 故地址为21位

4路组相连对应 $r=2$

Cache 16KB 对应 $2^{4+10} = 2^{14}$ 块内地址 $\log_2 8 = 3\text{位}$

每个字节对应 $\log_2 \frac{32}{8} = 2\text{位}$ 故组号 $14 - 2 - 3 - 2 = 7\text{位}$

主存高位地址	组号	块号	块内地址	字节
7	7	2	3	2

(2) 命中率为 $10/11 \approx 91\%$

有Cache
无Cache $= \frac{0.91 \times 1 + 0.09 \times (5+1)}{5} = 0.29$

$1 / 0.29 = 3.45\text{倍}$



7.6 (1) 0

(2) $0.95 \times 10 + 0.05 \times (10 + 100) = 15 \text{ ns}$

7.7 (1) 主存: $\log_2(8192 \times 32) = 18 \text{ 位}$

Cache: $\log_2(64 \times 32) = 11 \text{ 位}$

4路组相连

(2) 区号7位, 组号4位, 块号2位, 块内地址5位

$18 - 11 = 7$ $\log_2(64/4)$ $\log_2 4$ $\log_2 32$

7.9

若有虚存, 则用户编程无需考虑允许使用的主存容量;

若无虚存, 则需要考虑。

7.10 虚拟地址为30位

物理地址为 $2 + 20 = 22 \text{ 位}$

根据寻址方式计算的是虚拟地址

页表长度为 $2^{30-12} = 2^{18}$

7.11 (1) 80324 H

(2) 96128 H

(3) 去慢表中查找(在主存中), 可能需重新分配



7.12(1)																	
FIFO		3	4	2	6	4	3	7	4	3	6	3	4	8	4	6	命中率 $3/15 = 20\%$
	④	3	3	3	4	4	2	6	3	3	7	4	4	6	3	8	
	③	/	4	4	2	2	6	3	7	7	4	6	6	3	8	4	
	②	/	/	2	6	6	3	7	4	4	6	3	3	8	4	6	
	命中	x	x	x	x	✓	x	x	x	✓	x	x	✓	x	x	x	
LRU	④	3	3	3	4	2	6	4	3	7	4	4	6	3	3	8	命中率 $6/15 = 40\%$
	③	/	4	4	2	6	4	3	7	4	3	6	3	4	8	4	
	②	/	/	2	6	4	3	7	4	3	6	3	4	8	4	6	
	命中	x	x	x	x	✓	x	x	✓	✓	x	✓	✓	x	✓	x	
	(2) FIFO	④	3	3	3	3	3	3	4	4	2	2	2	6	7	7	
③		/	4	4	4	4	4	2	2	6	6	6	7	3	3	4	
②		/	/	2	2	2	2	6	6	7	7	7	3	4	4	8	
①		/	/	/	6	6	6	7	7	3	3	3	4	8	8	6	
命中		x	x	x	x	✓	✓	x	✓	x	✓	✓	x	x	✓	x	
LRU	④	3	3	3	3	3	2	6	6	6	7	7	7	6	6	3	命中率 $9/15 = 60\%$
	③	/	4	4	4	2	6	4	3	7	4	4	6	3	3	8	
	②	/	/	2	2	6	4	3	7	4	3	6	3	4	8	4	
	①	/	/	/	6	4	3	7	4	3	6	3	4	8	4	6	
	命中	x	x	x	x	✓	✓	x	✓	✓	✓	✓	✓	x	✓	✓	

(3) 命中率会显著提升, 因为CPU会多次访问相同的页面(局部性原理)