第3章 层次结构的存储器系统

考试内容范围

1. 存储器的分类
2. 存储器的层次化结构
3. 半导体随机存取存储器
(1) SRAM 存储器的工作原理
(2) DRAM 存储器的工作原理
4. 只读存储器
5. 主存储器与 CPU 的连接
6. 双口 RAM 和多模块存储器
7. 高速缓冲存储器(Cache)
(1) 程序访问的局部性原理
(2) Cache 的基本工作原理
(3) Cache 和主存之间的映射方式
(4) Cache 中主存块的替换算法
(5) Cache 写策略
8. 虚拟存储器
(1) 虚拟存储器的基本概念
(2) 页式虚拟存储器
(3) 段式虚拟存储器
(4) 段页式虚拟存储器
(5) TLB(块表)

3.1 存储器的性能指标和分类

存储器的性能指标有 3 个,即存储容量、单位成本和存储速度。

(1) 存储容量=存储字数×字长。

(2) 单位成本(也称每位价格):

$$C(\text{每位价格})=C(\text{总成本})/S(\text{总容量})$$

(3) 存储速度

$$\text{BM}(\text{数据传输率})=W(\text{数据的宽度})/T_M(\text{存储周期})$$

这是3个重要的相互制约的主要性能指标。设计存储系统所追求的目标是大容量、低成本和高速度。

通常的分类方法有3种：按存储介质、存取方式、所处位置或层次。

(1) 按存储介质分为磁表面存储器、半导体存储器和光存储器3种。

(2) 按存取方式分为随机存储器、只读存储器和串行存储器3种。

(3) 按所处位置或层次分为内存(主存)、外存(辅助存储器)、缓冲存储器。

另外，还有一些所处位置不同的存储器，如处于并行多处理机中共享位置的共享存储器，以实现各处理机的数据共享和数据通信；又如分布在网络中各个不同位置、实现网络系统中更大的存储容量、更安全可靠的存储和资源共享的网络存储器。

3.2 存储器的三层结构

为解决存储系统大容量、高速度和低成本三个互相制约的矛盾，计算机系统采用高速缓存、主存和辅存组成的多级结构的存储器系统。在应用程序员看来它是一个存储器。它的速度接近最快的那个存储器，容量与最大的那个存储器相等或接近，单位价格接近最便宜的那个存储器。如图3.1所示是计算机系统普遍采用的三级结构的存储系统的示意图。

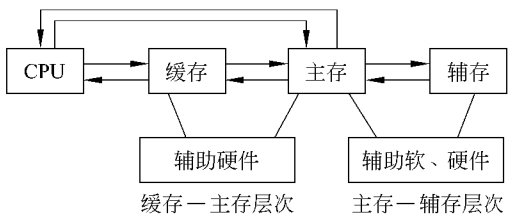


图3.1 三级结构的存储系统示意图

该系统的控制由辅助的硬件或软件加硬件支撑，其中Cache系统控制由纯硬件实现，对所有程序员透明，虚拟存储器系统的管理既有硬件也有软件(一般是操作系统)，对应用程序员透明。

三级结构的存储系统能够有高的性能/价格比是因为程序运行过程中存在局部性原理，即在一小段时间内(体现出时间上的局部性)只会用到一小部分程序和一小部分的数据，这一小部分程序和数据往往集中在一小片存储空间(体现出空间上的局部性)中，程序中顺序执行的语句比转移执行的语句要多很多。这就意味着，只要统一管理和调度好这三级结构的存储器系统，把最频繁使用的、数量不是太多的指令和数据分配到速度最快、容量不是太大的高速缓存中，把最近可能会用到的、数量要更多的指令和数据装入到主存储器中，而把一段时间内不大可能用到的指令和数据(数量可以非常大)暂存在存储容量非常大的虚拟存储器中，并随着时间推进和程序运行的实际情况，处理好信息在三层存储介质中的调度与交换，使不同存储介质充分发挥各自的优势，整体上就能取得比较理想的性能/价格比。

层次结构的存储器系统中不同层次上的存储器介质的具体关键特征如图3.2所示，它

们在速度、容量、价格方面是不同的，越往上存储速度越快、访问频度越高、存储容量越小、存储单位信息的成本越高。

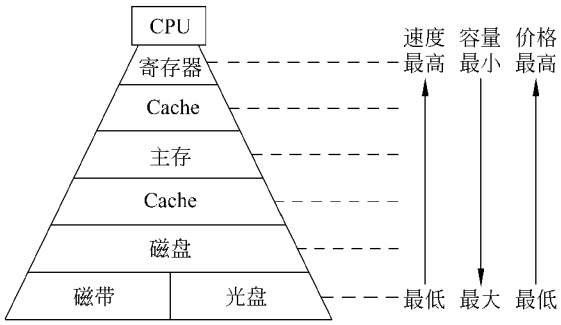


图 3.2 层次存储器结构示意图

要保证这样的存储器系统有正确的运行结果，在程序运行过程中还必须满足以下两项原则：

(1) 一致性原则。同一个信息会同时存放在几个层次的存储器中，这一信息在几个层次的存储器中必须保持相同的值。

(2) 包含性原则。处在内层(更靠近 CPU)存储器中的信息一定被包含在各外层的存储器中，即内层存储器中的全部信息一定是各外层存储器中所存信息中一小部分的副本。

3.3 半导体随机存取存储器

3.3.1 存储芯片的内部组成

静态存储器芯片中，通常由存储阵列、译码器电路、读写控制电路和数据缓冲电路等几个部分组成，如图 3.3(a)所示。

(1) 存储阵列(存储体)：由大量相同的位存储单元阵列构成。

(2) 译码器电路：将来自地址总线的信号，翻译成某单元(存储字或字节)的选通信号，使该单元能够被读写。

(3) 控制电路：对存储芯片进行片选控制、读写控制和输出控制等操作，它们分别通过$\overline{\text{CE}}$(Chip Enable)、$\overline{\text{WE}}$(Write Enable)和$\overline{\text{OE}}$(Output Enable)引脚实现。

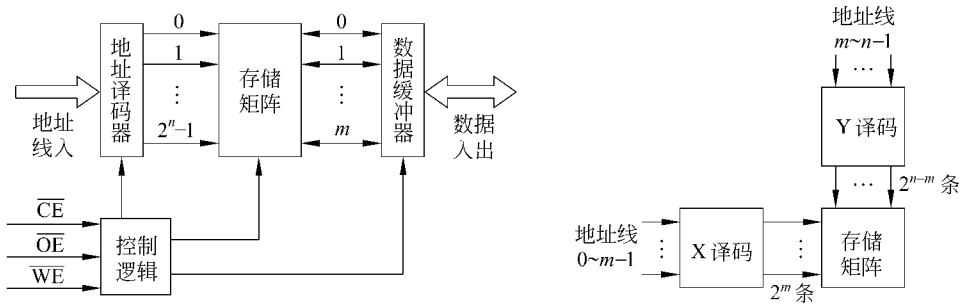
(4) 数据缓冲电路：用于暂存写入的数据或从存储体内读出的数据，具有三态控制。

注意：当芯片控制$\overline{\text{CE}}$无效，其他所有的信号都不起作用，即芯片不能工作；通常$\overline{\text{OE}}$信号与$\overline{\text{WE}}$信号是互斥的，即进行读操作时，$\overline{\text{OE}}$信号为低，而$\overline{\text{WE}}$信号为高。

译码方式分为单译码和双译码两种，双译码方式结构的存储器如图 3.3(b)所示。

【例 1】 通常大容量存储器采用双译码方式，其优点是可以节省大量的_____，例如，10 条地址线，单译码方式它需要_____，而双译码方式仅需要_____。

答案： 选通线；$2^{10}=1024$ 条；$2^5+2^5=64$ 条



(a) 存储芯片的内部组成　　(b) 存储芯片的双译码方式示意图

图 3.3　存储器芯片的内部组成与双译码示意图

3.3.2　静态 RAM 和动态 RAM 的工作原理

RAM 存储器有静态(SRAM)和动态(DRAM)之分。

1. 典型的 6 管 1 位 NMOS 静态 RAM 单元原理

其原理图如图 3.4 所示。

(1) 中间 4 个管子为 2 个反相器,交叉耦合组成双稳态触发器,能存 1 位二进制信息。

(2) 每个反相器中上面的管子作负载电阻用,两个反相器各输出 D 和$\overline{D}$信号。

(3) 触发器两边的管子用于外部连通或隔离,连通时兼传送读写的数据信号。

(4) 使这两个管子由字选(也称行选)信号控制工作与否;位选(也称列选)信号控制传送的信号能否传至芯片的数据引脚 I/O。当行和列选都选通,这个 1 位单元才被选中。

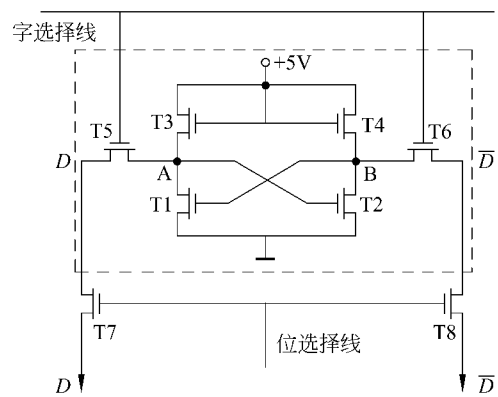


图 3.4　1 位 MOS 存储单元的组成

2. 典型的 1 管 NMOS 动态 RAM 单元原理

其原理图如图 3.5 所示。

(1) Cs 为 MOS 管的栅源电容,容量很小,C 存有电荷时记忆 1 状态,反之为 0 状态。

(2) 用于控制的 MOS 管,由字选信号控制其工作与否(即导通与否),导通时才能进行

读写操作。因用较少的晶体管构成一个位存储单元，因此提高了芯片单位面积上的容量，同时也降低了每位价格和功耗。

(3) 由于破坏性读出它需要读后重写，由于 Cs 漏电必须对它进行定时刷新。

(4) 常用的刷新(也称再生)方法有 3 种：集中式、分散式、异步式。

(5) DRAM 的刷新需要的控制电路包括刷新计数器、刷新/访存裁决、刷新控制逻辑等。刷新控制电路是 CPU 和 DRAM 的接口电路，目前这些电路通常都已集成在 DRAM 芯片中。

(6) DRAM 采用地址复用技术，地址线是原来的 1/2，且地址信号分行、列 2 次传送，分别由$\overline{RAS}$(行选通)和$\overline{CAS}$(列选通)控制，并用$\overline{RAS}$替代片选$\overline{CE}$信号。可进一步减小芯片的体积。

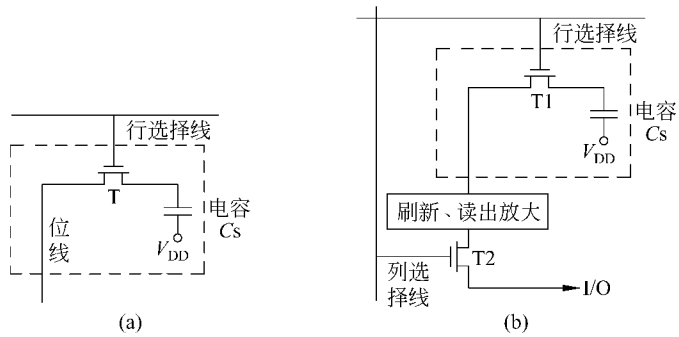


图 3.5 MOS 单管动态存储器的组成

3. SRAM 与 DRAM 的比较

其比较如表 3.1 所示。

表 3.1 SRAM 与 DRAM 的比较

名称 / 内容	SRAM	DRAM	名称 / 内容	SRAM	DRAM
存储信息	触发器	电容	运行速度	快	慢
破坏性读出	非	是	集成度	低	高
是否需要刷新	不需要	需要	发热量	大	小
送行列地址	同时送	分两次送	存储成本	高	低

4. 存储器读写过程中的时序简介

作为看懂时序的方法，仅介绍总线的信号表示和 SRAM 的读周期时序。

(1) 总线的信号表示：典型的 5 种信号表示如图 3.6 所示。

(2) SRAM 的读周期时序。

读周期主要关心：地址有效→CS 有效→数据输出→CS 复位→地址无效。

SRAM 的读周期时序示意图如图 3.7 所示。

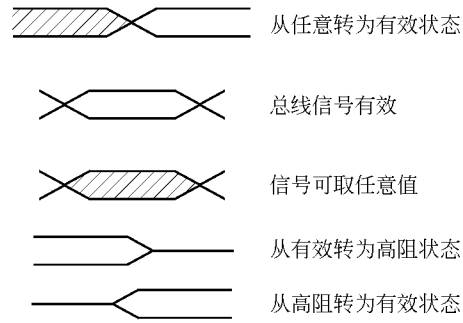


图 3.6 总线信号的表示

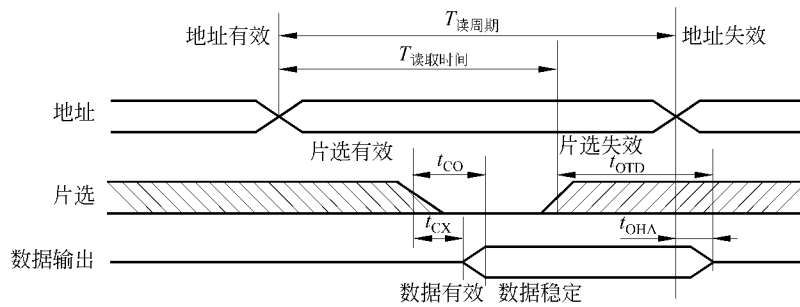


图 3.7 SRAM 的读周期时序示意图

写周期情况类似：地址有效 → CS 有效 → 数据有效 → CS 无效→地址无效。

(3) 对DRAM 的读周期时序也有类似情况。

读周期：由于分行、列两次送地址，过程分析如下：

行地址有效→行地址选通→列地址有效→列地址选通→数据输出→(行选通、列选通及地址)无效。

写周期：行地址有效→行地址选通→列地址、数据有效→列地址选通→数据输入→(行选通、列选通及地址)无效。

【例 2】 系统有 20 位地址线 A19～A0，现手头有 8 片 64K×1 位的存储器芯片，需要把它们组织起来，问：

(1) 如何使它们一起同时工作？

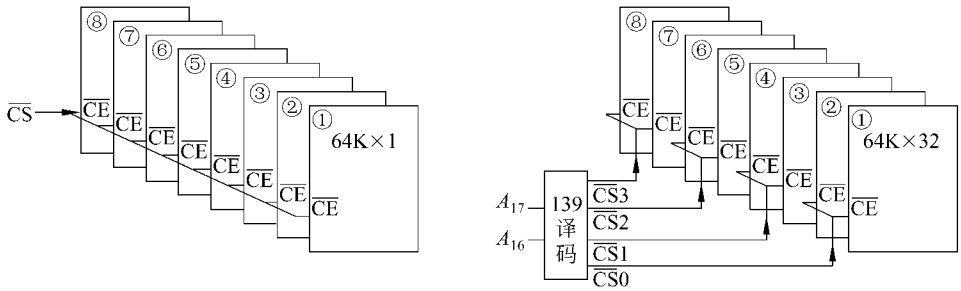
(2) 如何使 8 片(64K×32 位)这 4 组分时工作？

答案：

(1) 如图 3.8(a)所示，将 8 片一起工作的存储芯片的$\overline{CE}$连接在一起，并用一根高位地址线 A_{16}(我们假设为$\overline{CS}$)去选通它，就能实现这个目的。

提示：只要存储芯片中的$\overline{CE}$无效，芯片就不能工作，利用此功能可以达到这个目的。

(2) 如图 3.8(b)所示，将 8 片中每 2 片的$\overline{CE}$分别连接；为使各组分时工作，采用 1 片地址译码器，用 2 位高地址作为输入，再将译码输出$\overline{CS}3$、$\overline{CS}2$、$\overline{CS}1$ 和$\overline{CS}0$ 与各组的$\overline{CE}$分别连接，这样，当高地址为 11、10、01、00 时，就实现了各组芯片分时工作的目的。



(a) 8片芯片可以同时工作　　(b) 4组芯片只能分时工作

图 3.8　芯片允许信号$\overline{CE}$的应用

3.4　只读存储器

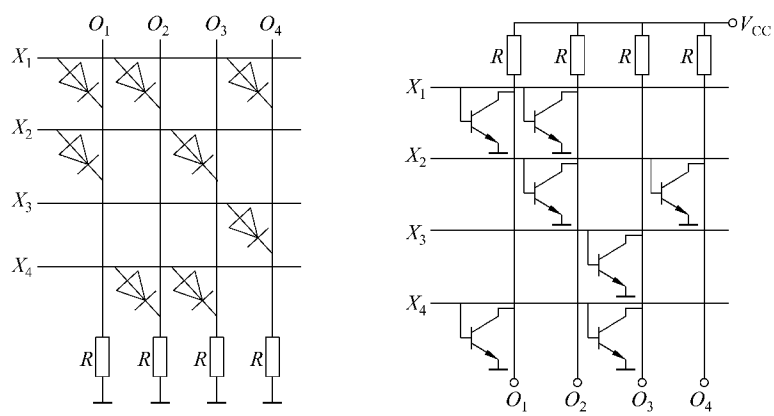
3.4.1　用二极管或三极管构成的只读存储器

1. 二极管组成的 ROM

利用二极管的单向导电性，可以构成只读存储器，如图 3.9(a)所示，输入的字线 X 和输出的位线 O 之间接一个二极管的位相当于存入 1 信号，无二极管的位相当于存入 0 信号。

2. 三极管存储单元

用三极管构成的只读存储器如图 3.9(b)所示，可用理想三极管的开关特性分析其原理。



(a) 二极管只读存储器　　(b) 三极管只读存储器

图 3.9　二极管和三极管只读存储器

3.4.2 可编程 ROM

1. 掩模式 ROM(MROM)的原理

内容由半导体制造厂按用户的要求,在芯片的生产过程中直接写入,写入之后无法改变其内容。

2. 一次性编程 ROM(PROM)的原理

(1) 以熔丝作为可编程(PROM),如图 3.10 所示。
(2) 所有的行列交叉点都保存了 0。
(3) 将某些位熔丝烧断,这些位就写入了 1,即为编程。

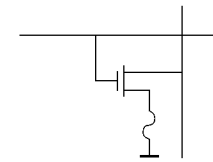


图 3.10 熔丝式 ROM(PROM)

3. 可擦除 PROM(EPROM)的原理

(1) EPROM 和多数闪存的基础如图 3.11 所示。

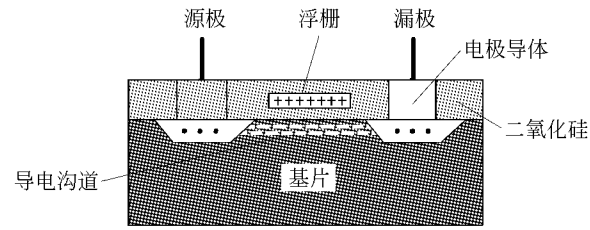


图 3.11 EPROM 的原理图

(2) 其栅极为浮栅,出厂时所有浮栅上没有电荷,即每位都存入 1。
(3) 浮栅的电荷可反复注入和擦除,所以可多次编程。

4. 电可擦 PROM(EPROM)的原理

在 EPROM 的浮栅上叠加擦除电极,就可使其方便地注入或擦除。

5. 闪速存储器(Flash Memory)的原理

主要是浮栅做得更薄。除了也用电擦除外,系统编程能力更强,并具有软件和硬件保护能力,可按字节、区块(Sector)或页面(Page)进行擦除和编程操作,内部可以自行产生编程电压(V_{PP}),所以只用单电源 V_{CC} 供电。

3.5 主存储器与 CPU 的连接

3.5.1 连接关系

主存储器通过数据总线 DB、地址总线 AB 和控制总线 CB 与 CPU 连接,如图 3.12

图 3.12　主存储器与 CPU 的连接

所示。

(1) 通过 DB 传送数据信息,DB 的位数与工作频率的乘积正比于数据传输率。

(2) 通过 AB 传送地址信息,AB 的位数决定了可寻址的最大内存空间。

(3) 通过 CB 传送控制信号,指出总线周期的类型和本次输入/输出操作完成的时刻。

构成主存的基本器件是半导体存储器芯片,了解它们的基本原理和相关技术,才能对主存部件进行更有效的组织与设计。由于存储器芯片的规格型号很多,单个存储芯片又很难满足系统容量或数据位的要求,往往需要用多片或不同型号的存储芯片来组织主存部件,通常采用**位扩展**、**字扩展**和**字位扩展技术**来实现。

1. 位扩展

如图 3.13 所示,用于增加存储字的长度,在存储芯片数据线位数小于 DB 位数的时候,把多个芯片的数据线并列起来,每个芯片的数据线代表存储字的不同数据位,相关的各存储芯片需要同时工作,并使用相同的地址信号和控制信号。

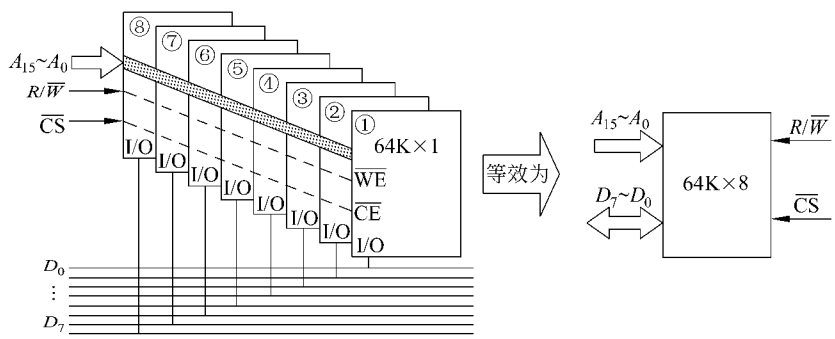


图 3.13　8 片存储芯片的位扩展

【例 3】 已知 CPU 的 DB 为 8 位,请用容量为 64K×1 位的存储芯片,将存储系统容量扩展为 64K×8 位,即 64KB。

答案:

(1) 将各存储芯片的数据线依次接 D_0,D_1,…,D_7,就实现了存储系统的位扩展。

(2) 由于它们要同时工作,所以将各芯片允许$\overline{CE}$都接在一起。

(3) 再将地址线和写允许 $\overline{WE}$也分别接在一起,这样,对选中的(地址相同)存储单元才能进行读写。扩展后的系统可等效为图左的一片 64K×8 位的存储器。

提示:因存储芯片的数据宽度为 1 位,需要在字长方向扩展为 8 位,所以是位扩展,用 8

块相同的芯片就能满足要求。具体连接如图 3.13 所示。

2. 字扩展

如图 3.14 所示，用于增加存储字的数量，在存储器芯片的容量小于存储器部件容量的要求时，就要用多个芯片来提供更大的存储器空间。这些存储芯片需要分时工作，此时连接各片的$\overline{CE}$一定不同(对地址的高几位译码产生)，其他控制线(读/写等)相同，各存储芯片数据线要连接在一起，实现芯片内寻址的地址线都相同。

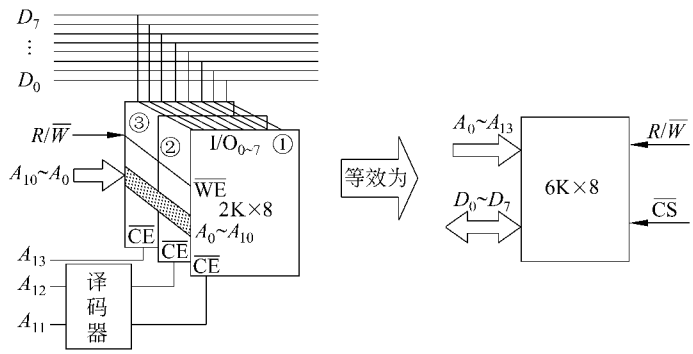


图 3.14 3 片存储芯片的字扩展

【例 4】 已知 CPU 数据总线为 8 位，地址总线为 16 位，请用 2K×8 位的存储芯片，将存储系统扩展成 6K×8 位，并指出需要几块内存芯片。

解：

(1) 采用 3 块相同的存储芯片，总容量即为 6KB。

(2) 显然各存储芯片的地址、读写控制和数据线接法都相同，差别是不同芯片$\overline{CE}$引脚信号不同，需要分别接对地址 A_{11}、A_{12}和 A_{13}译码产生的不同信号。

(3) 接到芯片内部的 11 位地址，为选中片内某 1 存储单元，即片内寻址；剩余的 3 位地址，是选中某 1 块存储芯片，即片外寻址，CPU 送出地址信号后，它们共同完成寻址工作。扩展后的存储系统等效为一块 6KB 的存储芯片。

提示：显然，由于存储芯片与 DB 的数据宽度相同，只需进行字容量方向的扩展，这是字扩展，具体连接如图 3.14 所示。

3. 字位扩展

如图 3.15 所示，即同时实现位扩展和字扩展，在存储器芯片的位数和字数都不够大时，用一组芯片实现存储器的字长扩展，用多组芯片实现存储器的容量扩展，汇总了前两种扩展的功能，此时用于位扩展的 8 个芯片同时工作，用于字扩展的 3 组分时工作。

【例 5】 已知 CPU 的数据宽度 8 位，地址线 16 位，请用 2K×1 位的存储芯片，扩展为 6K×8 位的存储系统。

解：

(1) 由于存储芯片与 CPU 的 DB 数据宽度不一致，需要位扩展，即 8 个芯片组成位宽为 8 位(一个字长)，容量为 2KB 的组。