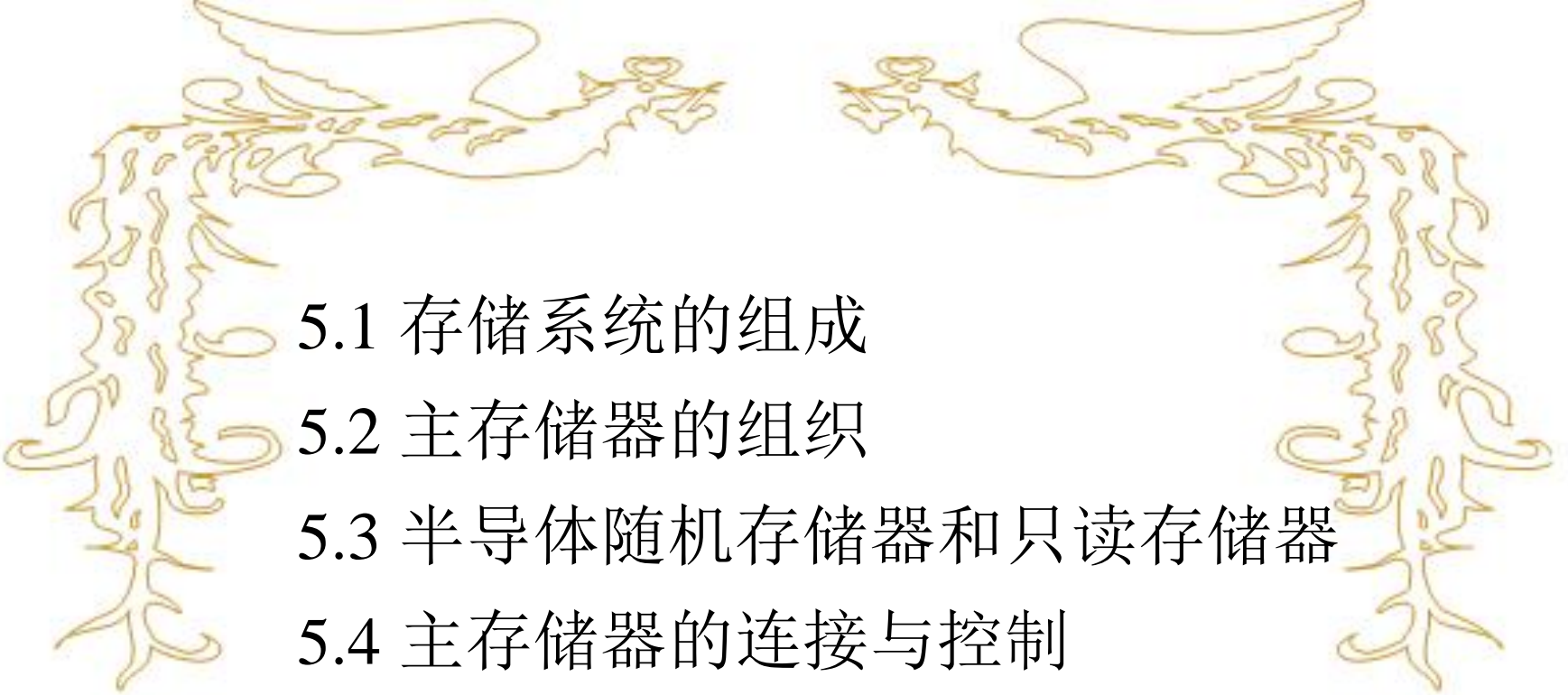





河北师范大学软件学院
Software College of Hebei Normal University

计算机组成原理

第五章 存储系统和结构

- 
- 5.1 存储系统的组成
 - 5.2 主存储器的组织
 - 5.3 半导体随机存储器和只读存储器
 - 5.4 主存储器的连接与控制
 - 5.5 提高主存读写速度的技术
 - 5.6 多体交叉存储技术
 - 5.7 高速缓冲存储器
 - 5.8 虚拟存储器
- 



5.1 存储系统的组成

5.2 主存储器的组织

5.3 半导体随机存储器和只读存储器

5.4 主存储器的连接与控制

5.5 提高主存读写速度的技术

5.6 多体交叉存储技术

5.7 高速缓冲存储器

5.8 虚拟存储器



5.4 主存储器的连接与控制

5.4.1 主存容量的扩展

要组成一个主存，首先要考虑选片的问题，然后就是如何把芯片连接起来的问题。根据存储器所要求的容量和选定的存储芯片的容量，就可以计算出总的芯片数，即

$$\text{总片数} = \frac{\text{总容量}}{\text{芯片容量}}$$

例如：存储器容量为 $8K \times 8$ ，若选用 $1K \times 4$ 的存储芯片，则需要：

$$\frac{8K \times 8}{1K \times 4} = 8 \times 2 \text{ 片} = 16 \text{ 片}$$

将多片组合起来常采用位扩展法、字扩展法、字和位同时扩展法。

5.4 主存储器的连接与控制

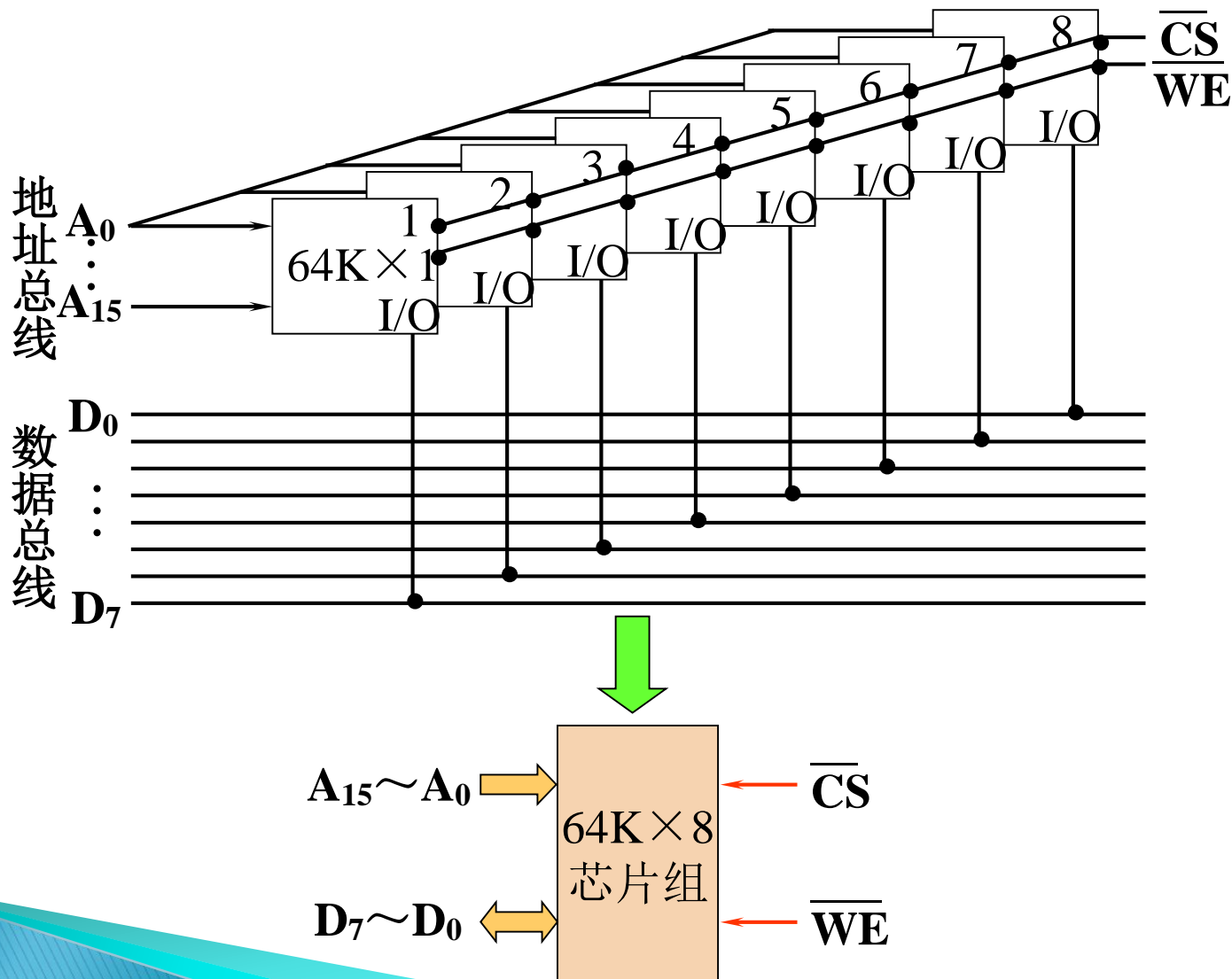
1. 位扩展

位扩展指只在**位数方向扩展**（加大字长），而芯片的字数和存储器的字数是一致的。位扩展的连接方式是将各存储芯片的地址线、片选线和读/写线相应地并联起来，而将各芯片的数据线单独列出。

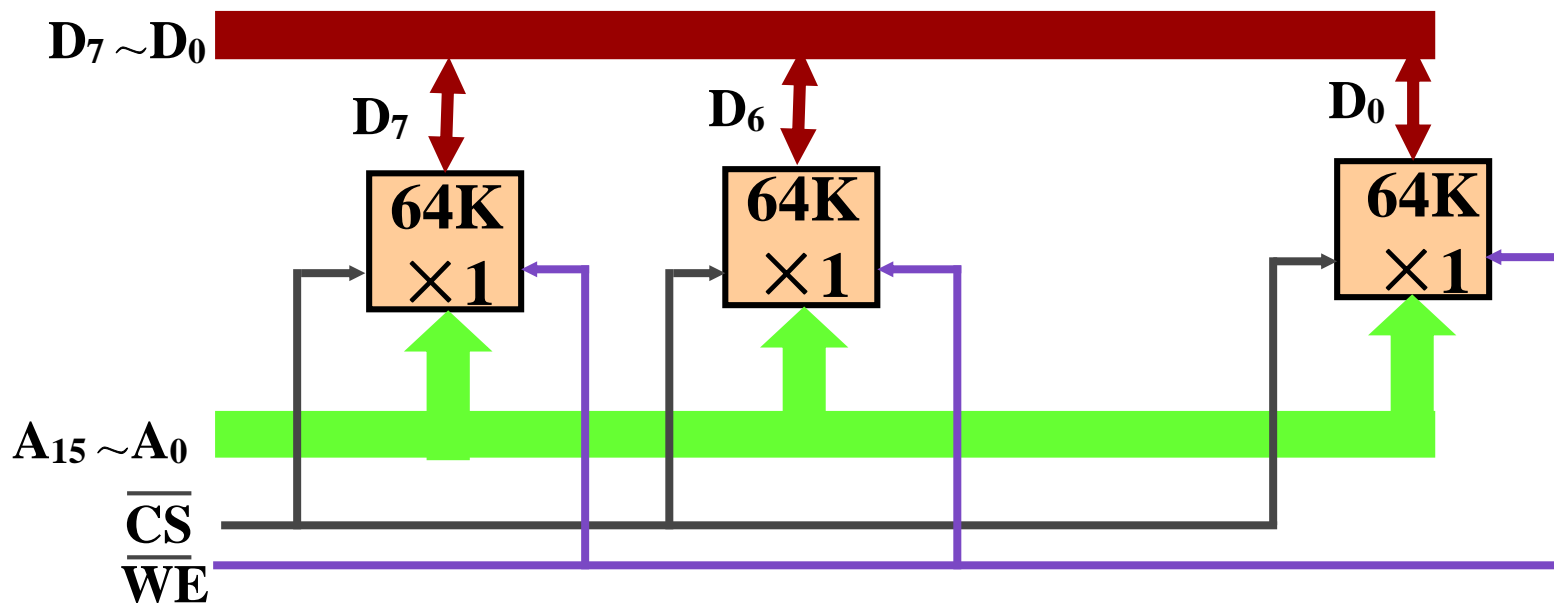
如用 $64\text{K} \times 1$ 的SRAM芯片组成 $64\text{K} \times 8$ 的存储器，需要8个芯片。

	容量	地址	数据
存储器	$64\text{K} \times 8$	16	8
存储芯片	$64\text{K} \times 1$	16	1

5.4 主存储器的连接与控制



5.4 主存储器的连接与控制



当CPU访问该存储器时，其发出的地址和控制信号同时传给8个芯片，选中每个芯片的同一单元，其单元的内容被同时读至数据总线的相应位，或将数据总线上的内容分别同时写入相应单元。

5.4 主存储器的连接与控制

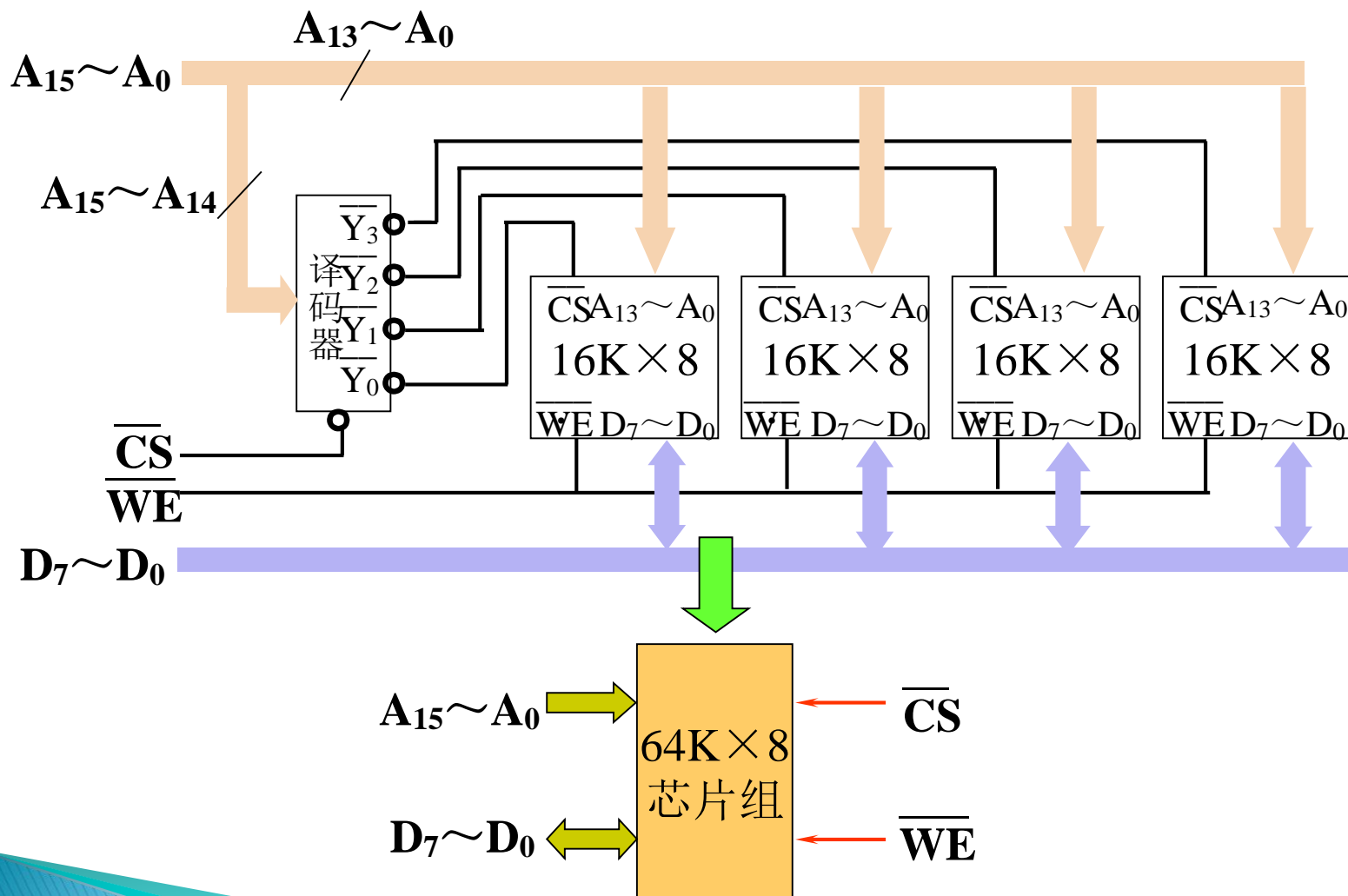
2. 字扩展

字扩展是指**仅在字数方向扩展，而位数不变**。字扩展将芯片的地址线、数据线、读/写线并联，由片选信号来区分各个芯片。

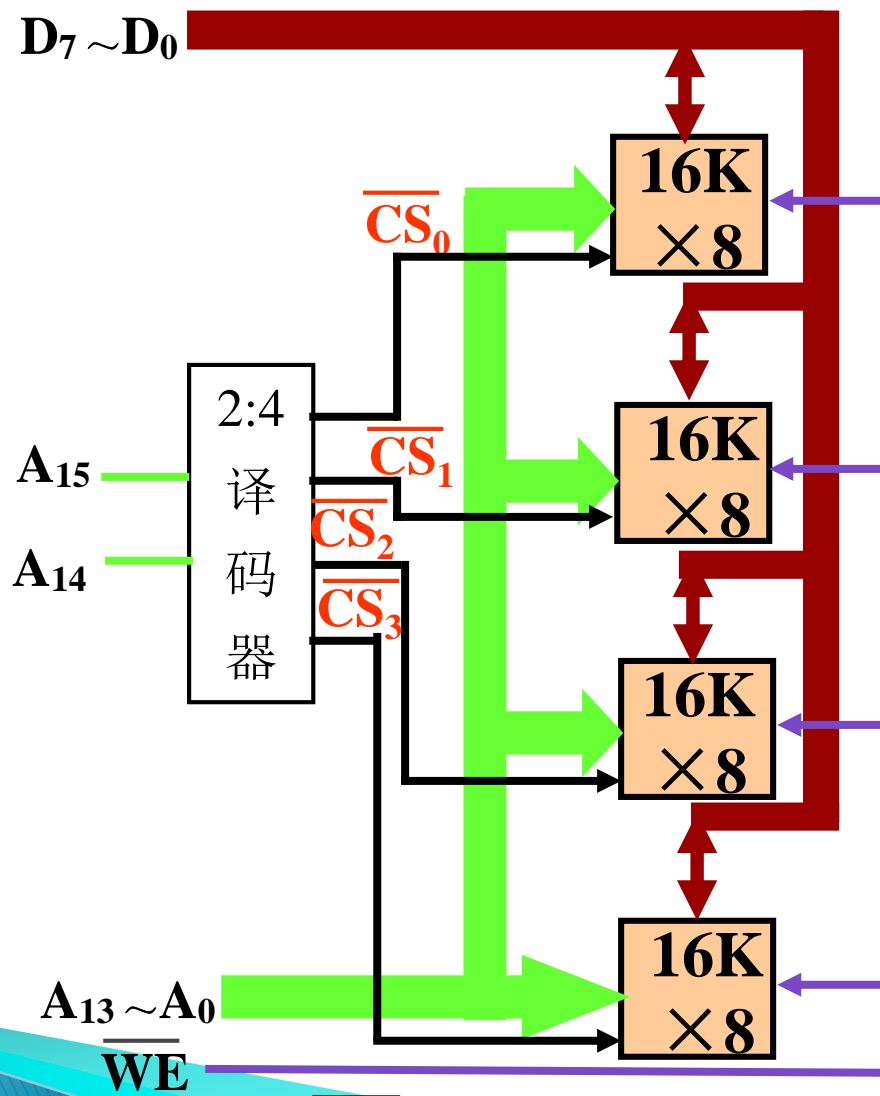
如用 $16\text{K} \times 8$ 的SRAM组成 $64\text{K} \times 8$ 的存储器，需要4个芯片。

	容量	地址	数据
存储器	$64\text{K} \times 8$	16	8
存储芯片	$16\text{K} \times 8$	14	8

5.4 主存储器的连接与控制



5.4 主存储器的连接与控制



5.4 主存储器的连接与控制

在同一时间内四个芯片中只能有一个芯片被选中。
四个芯片的地址分配如下：

第一片 最低地址 0000H

最高地址 3FFFH

第二片 最低地址 4000H

最高地址 7FFFH

第三片 最低地址 8000H

最高地址 BFFFH

第四片 最低地址 C000H

最高地址 FFFFH

5.4 主存储器的连接与控制

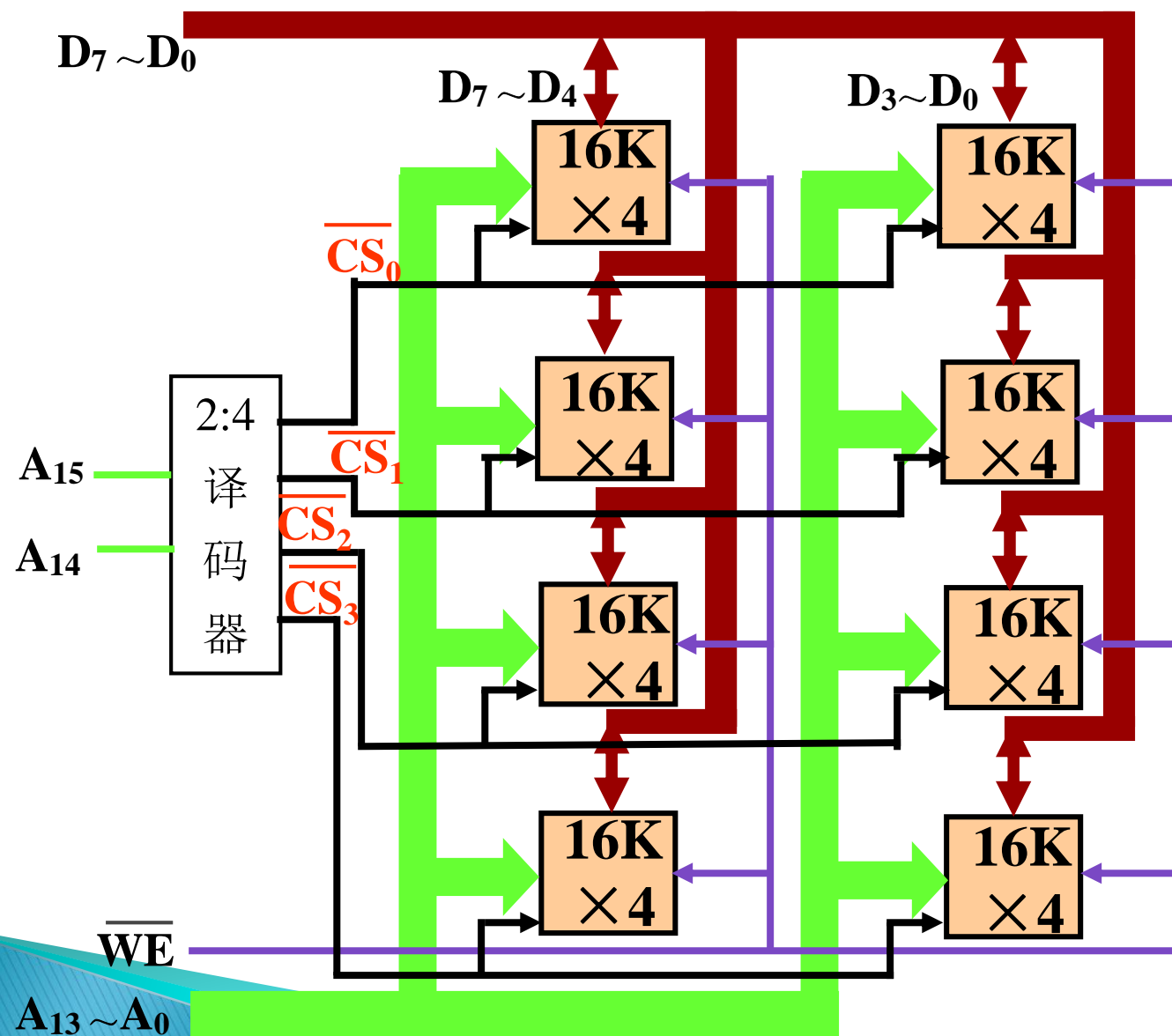
3. 字和位同时扩展

当构成一个容量较大的存储器时，往往需要在**字数方向和位数方向上同时扩展**，这将是前两种扩展的组合，实现起来也是很容易的。

如用 $16\text{K} \times 4$ 的SRAM组成 $64\text{K} \times 8$ 的存储器，需要8个芯片。

	容量	地址	数据
存储器	$64\text{K} \times 8$	16	8
存储芯片	$16\text{K} \times 4$	14	4

5.4 主存储器的连接与控制



5.4 主存储器的连接与控制

5.4.2 存储芯片的地址分配和片选

CPU要实现存储单元的访问，首先要选择存储芯片，即进行片选；然后再从选中的芯片中依地址码选择出相应的存储单元，以进行数据的存取，这称为字选。片内的字选是由CPU送出的N条低位地址线完成的，地址线直接接到所有存储芯片的地址输入端（N由片内存储容量 2^N 决定），而片选信号则是通过高位地址得到的。实现片选的方法可分为3种：即线选法、全译码法和部分译码法。

5.4 主存储器的连接与控制

1. 线选法

线选法就是用除片内寻址外的高位地址线直接（或经反相器）分别接至各个存储芯片的片选端，当某地址线信息为“0”时，就选中与之对应的存储芯片。请注意，这些片选地址线每次寻址时只能有一位有效，不允许同时有多位有效，这样才能保证每次只选中一个芯片（或组）。

芯片	$A_{14} \sim A_{11}$	$A_{10} \sim A_0$	地址范围
0#	1 1 1 0	00...0 11...1	7000~ 77FFH
1#	1 1 0 1	00...0 11...1	6800~ 6FFFH

5.4 主存储器的连接与控制

2#	1	0	1	1	00...0	5800~
					11...1	5FFFH
3#	0	1	1	1	00...0	3800~
					11...1	3FFFH

线选法的优点是不需要地址译码器，线路简单，选择芯片不需要外加逻辑电路，但仅适用于连接存储芯片较少的场合。同时，线选法不能充分利用系统的存储器空间，且把地址空间分成了相互隔离的区域，给编程带来了一定的困难。

5.4 主存储器的连接与控制

2.全译码法

全译码法将片内寻址外的全部高位地址线作为地址译码器的输入，把经译码器译码后的输出作为各芯片的片选信号，将它们分别接到存储芯片的片选端，以实现存储芯片的选择。

全译码法的优点是每片（或组）芯片的地址范围是唯一确定的，而且是连续的，也便于扩展，不会产生地址重叠的存储区，但全译码法对译码电路要求较高。

5.4 主存储器的连接与控制

芯片	$A_{19} \sim A_{13}$	A_{12}	A_{11}	$A_{10} \sim A_0$	地址范围
0#	0 ... 0	0	0	00...0 11...1	00000~ 007FFH
1#	0 ... 0	0	1	00...0 11...1	00800~ 00FFFH
2#	0 ... 0	1	0	00...0 11...1	01000~ 017FFH
3#	0 ... 0	1	1	00...0 11...1	01800~ 01FFFH

5.4 主存储器的连接与控制

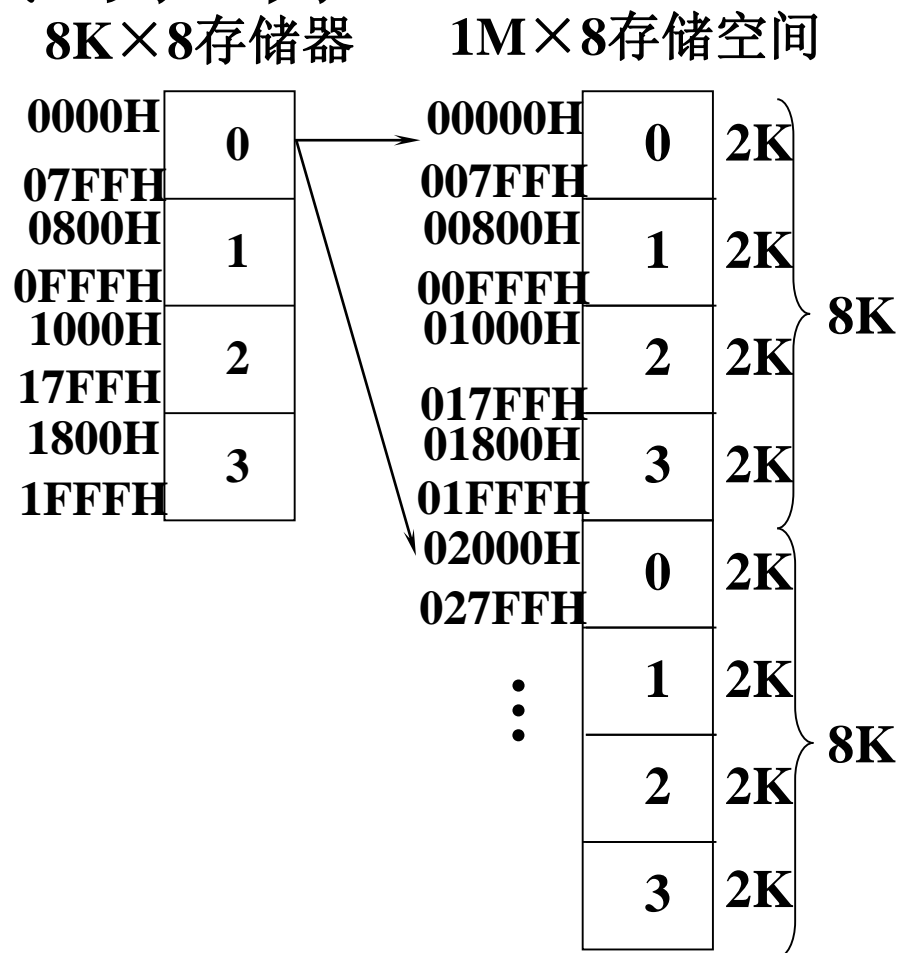
3.部分译码

所谓部分译码即用片内寻址外的高位地址的一部分来译码产生片选信号。

如用4片 $2K \times 8$ 的存储芯片组成 $8K \times 8$ 存储器，需要四个片选信号，因此只要用两位地址线来译码产生。设地址总线有20位（ $A_{19} \sim A_0$ ），则寻址 $8K \times 8$ 存储器时，无论 $A_{19} \sim A_{13}$ 取何值，只要 $A_{12} = A_{11} = 0$ ，而均选中第一片，只要 $A_{12} = 0$ ， $A_{11} = 1$ ，均选中第二片，……。也就是说，8K RAM中的任一个存储单元，都对应应有 $2^{(20-13)} = 2^7$ 个地址，这种一个存储单元出现多个地址的现象称地址重叠。

5.4 主存储器的连接与控制

从地址分布来看，这
8KB存储器实际上占用
了CPU全部的空间
(1MB)。每片 $2K \times 8$
的存储芯片有
 $1/4M=256K$ 的地址重
叠区。



5.4 主存储器的连接与控制

令未用到的高位地址全为0，这样确定的存储器地址称为基本地址，本例中 $8K \times 8$ 存储器的基本地址即00000H~01FFFH。部分译码法较全译码法简单，但存在地址重叠区。

用2114 ($1K \times 4$) SRAM芯片组成容量为 $4K \times 8$ 的存储器。地址总线A15~A0 (低), 双向数据总线D7~D0 (低), 读/写信号线 R/\bar{W} 。给出芯片地址分配与片选逻辑, 并画出M框图。

(1) 先扩展位数, 再扩展单元数。

$$\begin{array}{l} 2\text{片 } 1K \times 4 \longrightarrow 1K \times 8 \\ 4\text{组 } 1K \times 8 \longrightarrow 4K \times 8 \end{array} > 8\text{片}$$

(2) 先扩展单元数, 再扩展位数。

$$\begin{array}{l} 4\text{片 } 1K \times 4 \longrightarrow 4K \times 4 \\ 2\text{组 } 4K \times 4 \longrightarrow 4K \times 8 \end{array} > 8\text{片}$$

存储器寻址逻辑 { 芯片内的寻址系统
芯片外的地址分配与片选逻辑

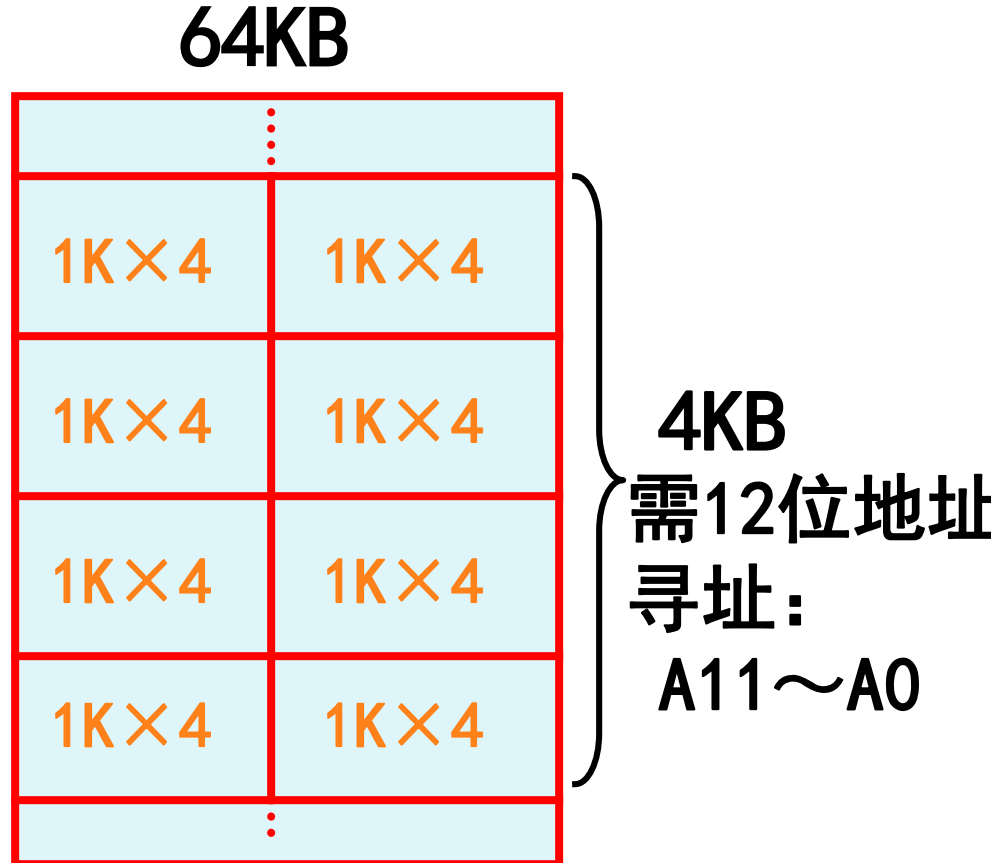
为芯片分配哪几位地址，
以便寻找片内的存储单
元

由哪几位地址形
成芯片选择逻辑，
以便寻找芯片

存储空间分配：

4KB存储器在16位地址空间（64KB）中占据
任意连续区间。

任意值	片选		芯片地址			
A15...A12	A11	A10	A9	A0	
<	0	0	0	0	
	0	0	1	1	
<	0	1	0	0	
	0	1	1	1	
<	1	0	0	0	
	1	0	1	1	
<	1	1	0	0	
	1	1	1	1	



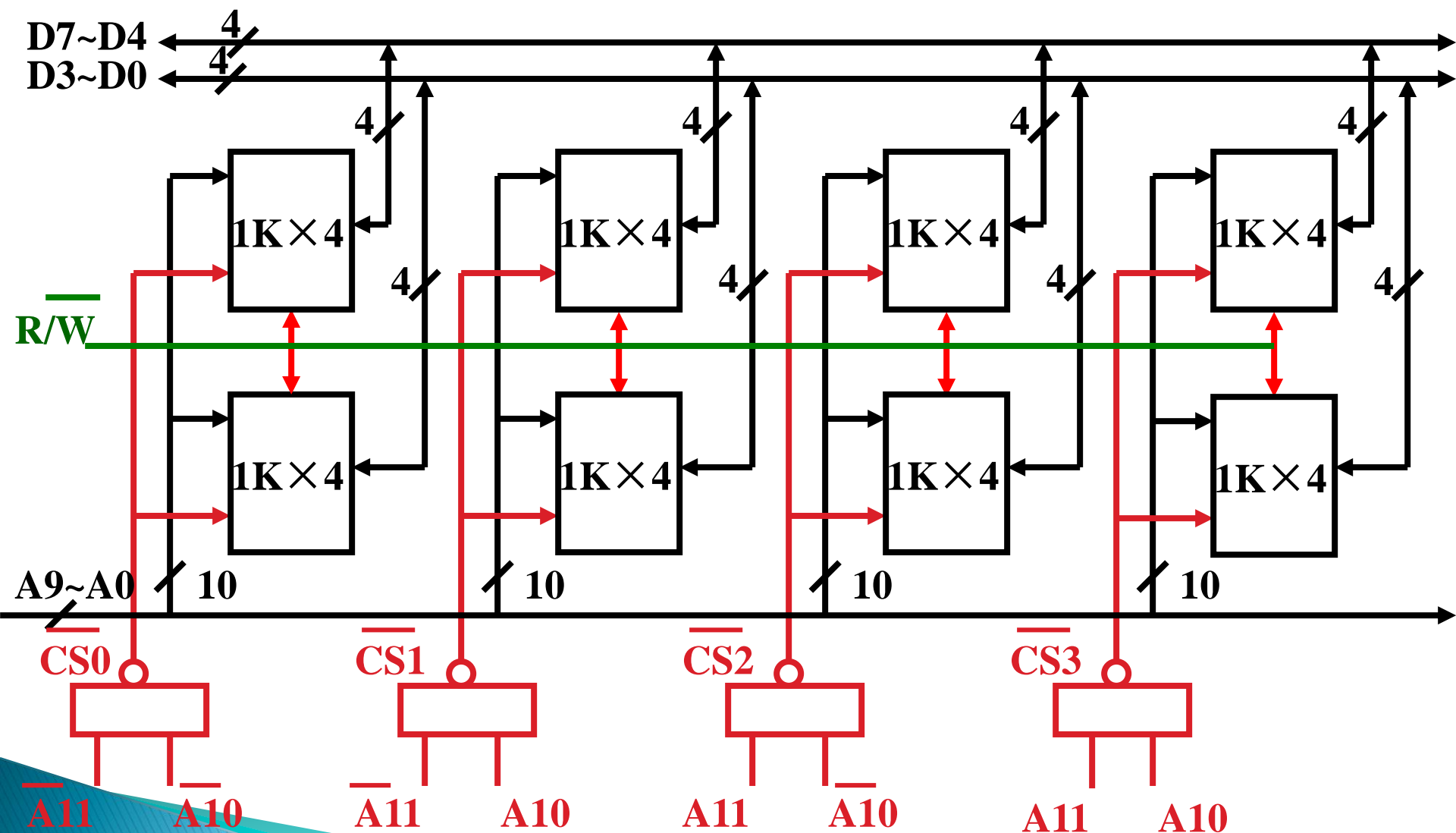
低位地址分配给芯片，高位地址形成片选逻辑。

芯片	芯片地址	片选信号	片选逻辑
1K	A9~A0	CS0	$\overline{A_{11}}\overline{A_{10}}$
1K	A9~A0	CS1	$\overline{A_{11}}A_{10}$
1K	A9~A0	CS2	$A_{11}\overline{A_{10}}$
1K	A9~A0	CS3	$A_{11}A_{10}$

(1) 扩展位数

(2) 扩展单元数

(3) 连接控制线



(4) 形成片选逻辑电路

某半导体存储器容量 $4K \times 8b$ 。其中固化区2KB选用EPROM芯片2716 ($2K \times 8b$)；工作区2KB，选用RAM芯片2114 ($1K \times 4b$)。地址总线A15~A0 (低)，双向数据总线D7~D0 (低)，读/写信号线R/W。

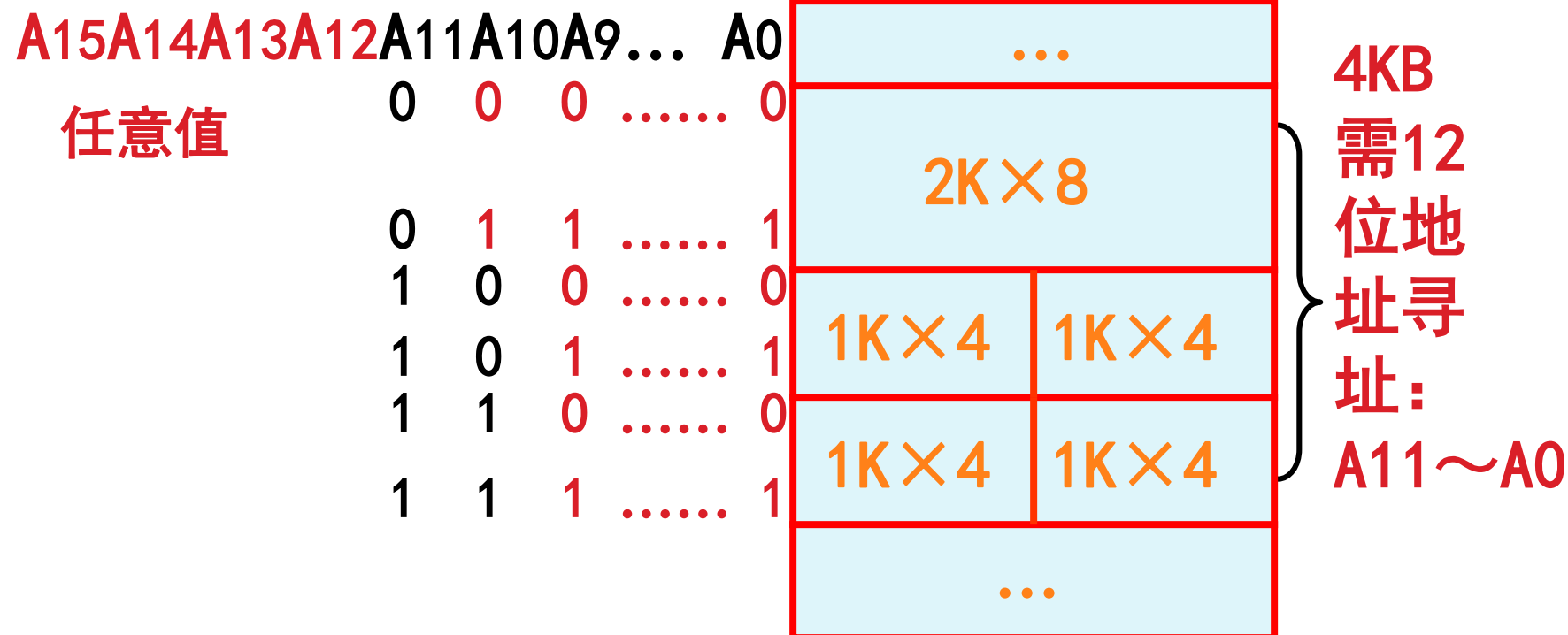
1. 计算容量和芯片数

ROM区：2KB

RAM区：2KB 共5片

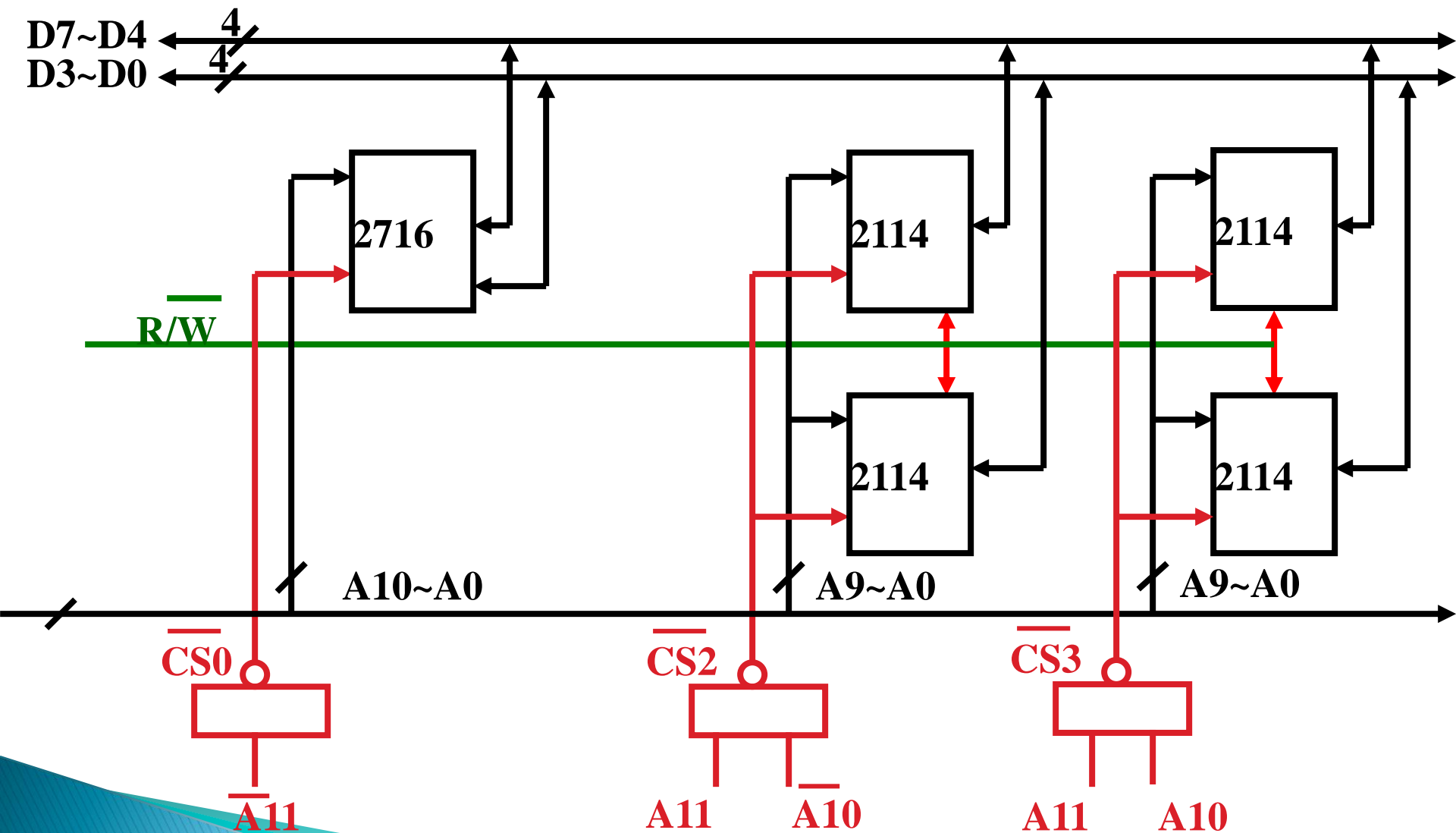
存储空间分配：先安排大容量芯片（放地址低端），再安排小容量芯片。
便于拟定片选逻辑。

4K × 8b



低位地址分配给芯片，高位地址形成片选逻辑。

芯片	芯片地址	片选信号	片选逻辑
2K	A10 ~ A0	CS0	$\overline{A_{11}}$
1K	A9 ~ A0	CS1	$A_{11} \overline{A_{10}}$
1K	A9 ~ A0	CS2	$A_{11} A_{10}$



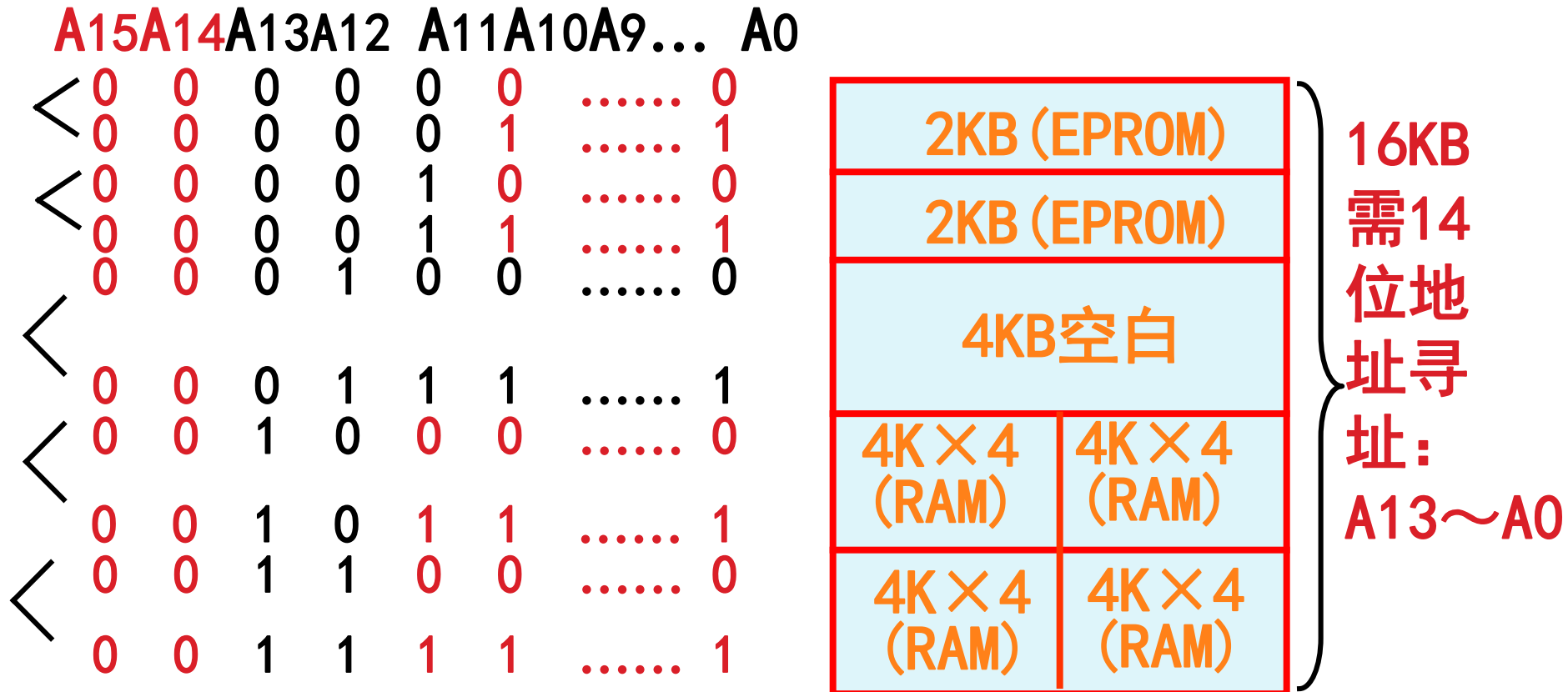
设计一个容量为16KB (按字节编址) 的存储器, 从低地址向高地址依次为4KB固化区 (选用2KB/片的EPROM芯片)、4KB的空区 (无存储芯片)、8KB的随机读/写区 (选用4K×4位的RAM芯片)。CPU的地址总线为A15~A0 (低), 双向数据总线为D7~D0 (低), 读/写控制信号为R/W, 访存请求信号为MREQ。请设计出该存储器, 并与CPU连接。

1. 芯片的选择

固化区: 4KB (2片) 随机读/写区: 8KB (4片)

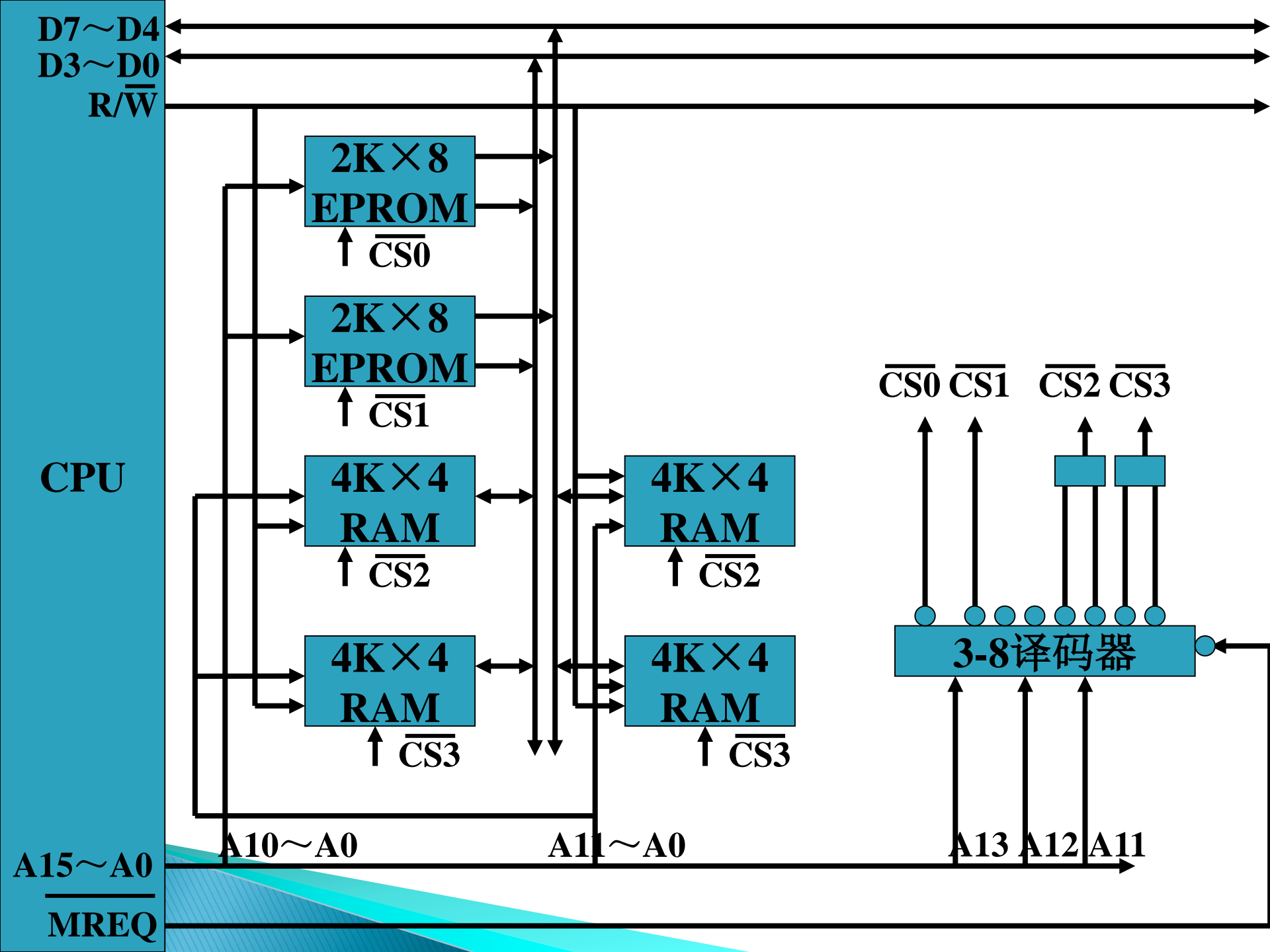
共6片

2. 地址分配与片选逻辑



低位地址分配给芯片，高位地址形成片选逻辑。

芯片	芯片地址	片选信号	片选逻辑
2K	A10~A0	CS0	$\overline{A13}\overline{A12}\overline{A11}$
2K	A10~A0	CS1	$\overline{A13}\overline{A12}A11$
4K	A11~A0	CS2	$A13\overline{A12}$
4K	A11~A0	CS3	$A13A12$



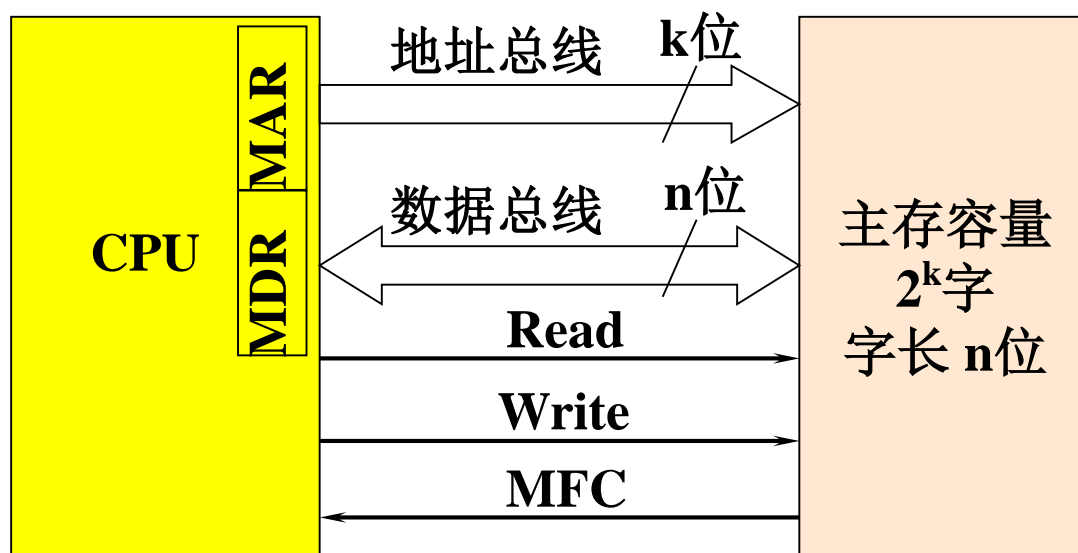
5.4 主存储器的连接与控制

5.4.3 主存储器和CPU的连接

1.主存和CPU之间的硬连接

主存与CPU的硬连接有三组连线：地址总线（AB）、数据总线（DB）和控制总线（CB）。此时，我们把主存看作一个黑盒子，存储器地址寄存器（MAR）和存储器数据寄存器（MDR）是主存和CPU之间的接口。MAR可以接受来自程序计数器的指令地址或来自运算器的操作数地址，以确定要访问的单元。MDR是向主存写入数据或从主存读出数据的缓冲部件。

5.4 主存储器的连接与控制



5.4 主存储器的连接与控制

2.CPU对主存的基本操作

CPU与主存的硬连接是两个部件之间联系的物理基础，而两个部件之间还有软连接，即CPU向主存发出的读或写命令，这才是两个部件之间有效工作的关键。

CPU对主存进行读/写操作时，首先CPU在地址总线上给出地址信号，然后发出相应的读或写命令，并在数据总线上交换信息。

5.4 主存储器的连接与控制

(1)读

读操作是指从CPU送来的地址所指定的存储单元中取出信息，再送给CPU，其操作过程是：

地址→MAR→AB CPU将地址信号送至地址总线

Read CPU发读命令

Wait for MFC 等待存储器工作完成信号

((MAR))→DB→MDR 读出信息经数据总线送至CPU

5.4 主存储器的连接与控制

(2)写

写操作是指将要写入的信息存入CPU所指定的存储单元中，其操作过程是：

地址→MAR→AB CPU将地址信号送至地址总线

数据→MDR→DB CPU将要写入的数据送至数据
总线

Write CPU发写命令

Wait for MFC 等待存储器工作完成信号

5.4 主存储器的连接与控制

由于CPU和主存的速度存在着差距，所以两者之间的速度匹配是很关键的，通常有两种匹配方式：同步存储器读取和异步存储器读取。上面给出的读/写基本操作是以异步存储器读取来考虑的，CPU和主存间没有统一的时钟，由存储器工作完成信号（MFC）通知CPU存储器工作已完成。

对于同步存储器读取，CPU和主存采用统一时钟，因为主存速度较慢，所以CPU与之配合必须放慢速度。在这种存储器中，不需要存储器工作完成信号。

5.4 主存储器的连接与控制

5.4.4 PC系列微机的存储器接口

数据总线一次能并行传送的位数，称为总线的数据通路宽度，常见的有8位、16位、32位、64位几种。但大多数主存储器常采取字节编址，每次访存允许读/写8位，以适应对字符类信息的处理。

1.8位存储器接口

如果数据总线为8位（如微机系统中的PC总线），而主存按字节编址，则匹配关系比较简单。一个总线周期中读/写8位。

5.4 主存储器的连接与控制

2.16位存储器接口

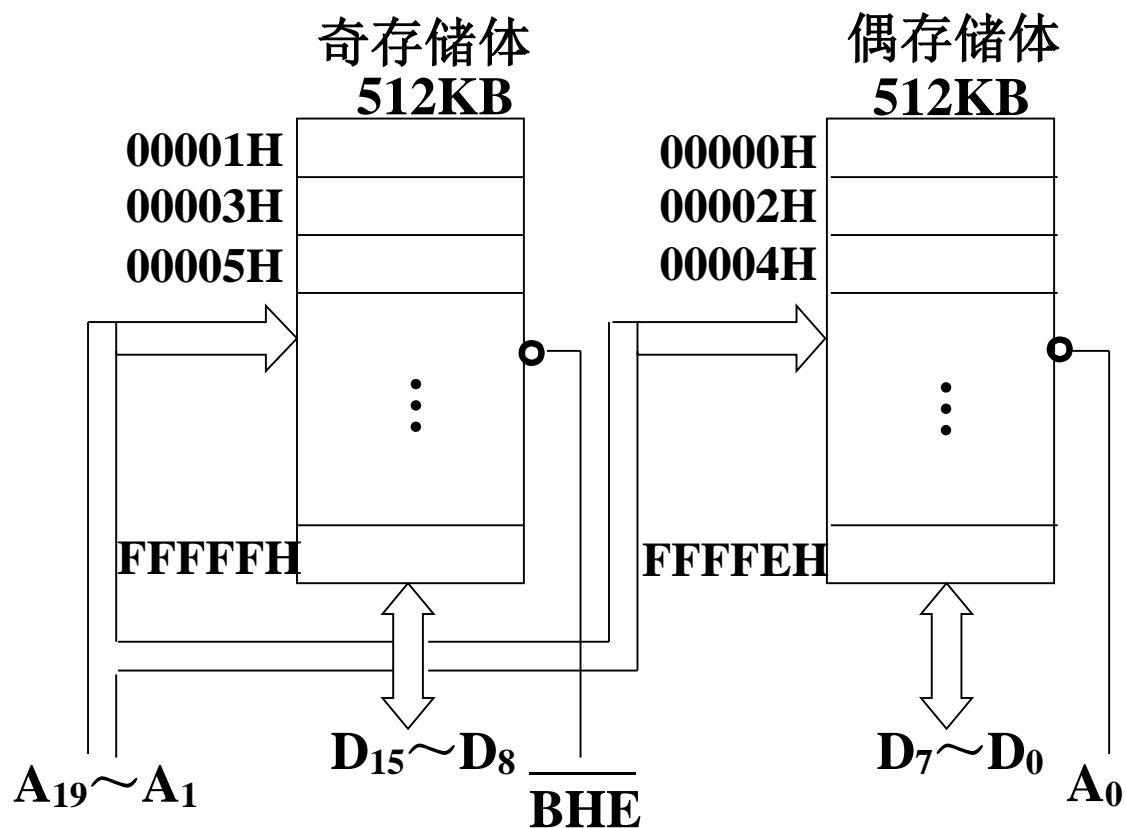
对于16位的微处理器8086（或80286），在一个总线周期内可读/写两个字节，即先送出偶地址，然后同时读/写这个偶地址单元和随后的奇地址单元，用低8位数据总线传送偶地址单元的数据，用高8位数据总线传送奇地址单元的数据，这样读/写的字（16位）被称为规则字。如果读/写的是非规则字，即是从奇地址开始的字，这时需要安排两个总线周期才能实现。

5.4 主存储器的连接与控制

为了实现这样的传送，需要将存储器分为两个存储体，一个存储体的地址均为偶数，称为偶地址（低字节）存储体，它与低8位数据线相连；另一个存储体的地址均为奇数，称为奇地址（高字节）存储体，与高8位数据线相连。8086和主存之间可以传送一个字节（8位）数据，也可以传送一个字（16位）数据。任何两个连续的字节都可以作为一个字来访问，地址值较低的字节是低位有效字节，地址值较高的字节是高位有效字节。

8086微处理器的地址线 $A_{19} \sim A_1$ 同时送至两个存储体， \overline{BHE} （高位存储体）和最低位地址线 A_0 用来选择一个或两个存储体进行数据传送。

5.4 主存储器的连接与控制



5.4 主存储器的连接与控制

8086微处理器的地址线 $A_{19} \sim A_1$ 同时送至两个存储体， \overline{BHE} （高位存储体）和最低位地址线 A_0 用来选择一个或两个存储体进行数据传送。

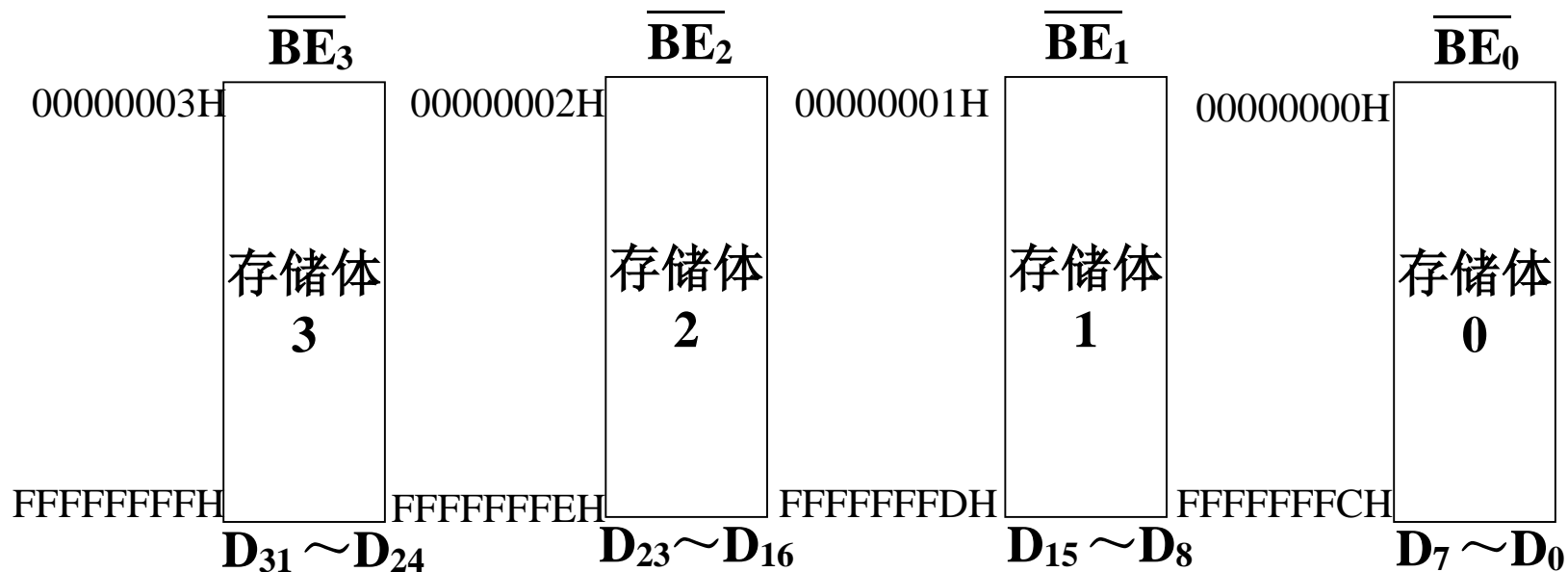
\overline{BHE}	A_0	特征
0	0	全字（规则字）传送
0	1	在数据总线高8位上进行字节传送
1	0	在数据总线低8位上进行字节传送
1	1	备用

5.4 主存储器的连接与控制

3.32位存储器接口

32位微处理器的存储器系统由4个存储体组成，每个存储体的存储空间为1GB，存储体选择通过选择信号 $\overline{\text{BE3}}$ 、 $\overline{\text{BE2}}$ 、 $\overline{\text{BE1}}$ 和 $\overline{\text{BE0}}$ 实现。如果要传送一个32位数，那么4个存储体都被选中；若要传送一个16位数，则有2个存储体（通常是 $\overline{\text{BE3}}$ 和 $\overline{\text{BE2}}$ 或者 $\overline{\text{BE1}}$ 和 $\overline{\text{BE0}}$ ）被选中；若传送的是8位数，只有一个存储体被选中。

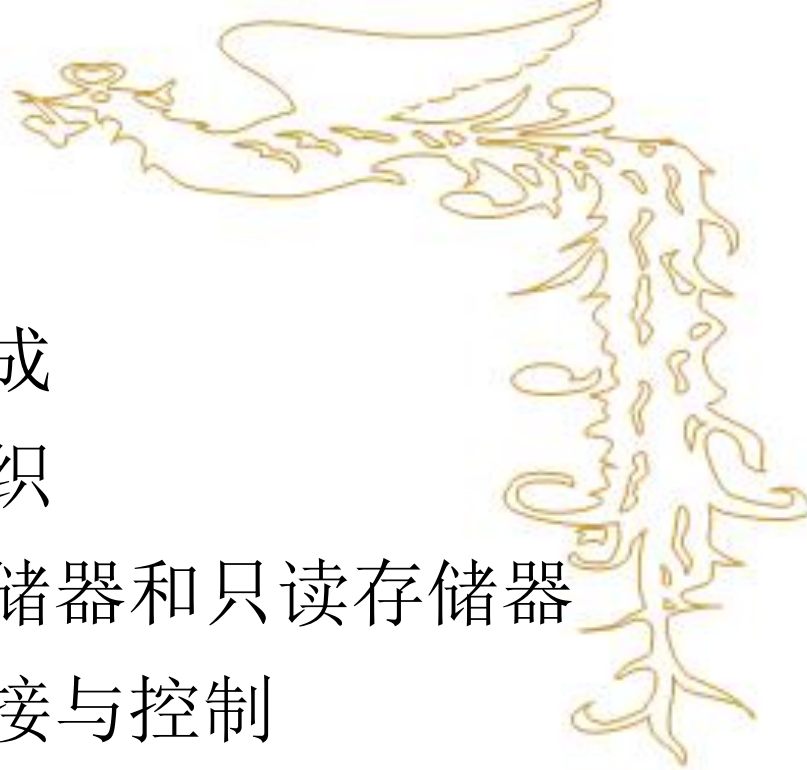
5.4 主存储器的连接与控制



5.4 主存储器的连接与控制

4.64位存储器接口

64位微处理器的存储器系统由8个存储体组成，每个存储体的存储空间为512MB（Pentium）或8GB（Pentium Pro），存储体选择通过选择信号 $\overline{BE7} \sim \overline{BE0}$ 实现。如果要传送一个64位数，那么8个存储体都被选中；如果要传送一个32位数，那么4个存储体都被选中；若要传送一个16位数，则有2个存储体被选中；若传送的是8位数，只有一个存储体被选中。



5.1 存储系统的组成

5.2 主存储器的组织

5.3 半导体随机存储器和只读存储器

5.4 主存储器的连接与控制

5.5 提高主存读写速度的技术

5.6 多体交叉存储技术

5.7 高速缓冲存储器

5.8 虚拟存储器



5.5 提高主存读写速度的技术

5.5.1 RAM与CPU速度的匹配

RAM的速度通常以ns表示，而CPU速度总是被表示为MHz，最近一些更快更新的RAM也用MHz来表示速度。

我们希望RAM的速度与CPU速度相等，然而实际RAM的速度远远落后于CPU的速度。以PC机为例，在1998年以前，DRAM的存取时间为60ns或更大，这相当于16.7MHz或更慢的速度，而当时CPU的速度已达到300MHz或更高的速度，两者之间存在着很大的差距。目前RAM的速度达到100MHz~266MHz，更高的RAM带宽可达到2.12GB/s，而CPU的速度则达到了2GHz或更快。

5.5 提高主存读写速度的技术

5.5.2 FPM DRAM（快速页模式随机存储器）

传统的DRAM在存取数据时，必须分别输入行地址和列地址信息，而FPM DRAM对这种方式做了改进。FPM DRAM的速度之所以能提高是基于这样一个事实——计算机中大量的数据是连续存放的。比如，若一个数据与前一个数据的行地址相同，内存控制器就不必再传一个行地址，只要再传一个列地址就可以了。由于大量的数据是连续存放的，这种触发行地址后连续输出列地址的方式使我们能用较少的时钟周期读较多的数据。

5.5 提高主存读写速度的技术

5.5.3 EDO DRAM（扩展数据输出 DRAM）

在 FPM DRAM基础上加以改进的存储器控制技术。传统的DRAM和FPM DRAM在存取每一数据时，输入行地址和列地址后必须等待电路稳定，然后才能有效的读写数据，而下一个地址必须等待这次读/写周期完成才能输出。而EDO输出数据在整个CAS周期都是有效的，EDO不必等待当前的读/写周期完成即可启动下一个读/写周期，即可以在输出一个数据的过程中准备下一个数据的输出。EDO DRAM在读/写一个存储单元时，同时启动下一个（连续）存储单元的读/写周期，从而节省了重选地址的时间，提高了读/写速度。

5.5 提高主存读写速度的技术

5.5.4 SDRAM（同步DRAM）

前面介绍的几种DRAM主存都属于“非同步存取的存储器”，即它们的工作速度并没有和系统时钟同步，存取数据时，系统须等待若干时钟周期才能接收和发送数据。如EDO DRAM须等待2个时钟周期，FPM DRAM则须等待3个时钟周期，这种等待制约了系统的数据传送速率。通常，FPM DRAM和EDO DRAM的速度不能超过66MHz。

5.5 提高主存读写速度的技术

5.5.5 DDR SDRAM（双数据传输率同步动态随机存储器）

它是SDRAM的升级版本，与SDRAM的主要区别是：DDR SDRAM不仅能在时钟脉冲的上升沿读出数据而且还能在下降沿读出数据，不需要提高时钟频率就能加倍提高SDRAM的速度。

5.5 提高主存读写速度的技术

5.5.6 DDR2 SDRAM和DDR3 SDRAM

DDR2 SDRAM是新生代内存技术标准，它与上一代DDR SDRAM技术标准最大的不同在于，虽然同是采用了在时钟的上升/下降沿同时进行数据传输的基本方式，但DDR2 SDRAM却拥有两倍于上一代DDR SDRAM的预读取能力（即：4bit数据读预取）。换句话说，DDR2 SDRAM每个时钟能够以4倍于外部总线的速度读/写数据，即在同样100MHz的工作频率下，DDR的实际频率为200MHz，而DDR2则可以达到400MHz。

DDR3 SDRAM可以看作是DDR2的改进版，DDR2的预取设计位数是4bit，也就是说DRAM内核的频率只有接口频率的1/4，而DDR3的预取设计位数提升至8bit，其DRAM内核的频率达到了接口频率的1/8。

5.5 提高主存读写速度的技术

5.5.7 Rambus DRAM

是继SDRAM之后的新型高速动态随机存储器。

RDRAM在内部结构上进行了重新设计，并采用了新的信号接口技术，其对外接口也不同于以前的DRAM。

使用FPM/EDO或SDRAM的传统主存系统称为宽通道系统，它们的主存通道和处理器的数据总线一样宽。RDRAM却是一种窄通道系统，它一次只传输16位数据（加上2个可选的校验位），但速度却快得多。



5.1 存储系统的组成

5.2 主存储器的组织

5.3 半导体随机存储器和只读存储器

5.4 主存储器的连接与控制

5.5 提高主存读写速度的技术

5.6 多体交叉存储技术

5.7 高速缓冲存储器

5.8 虚拟存储器



5.6 多体交叉存储技术

5.6.1 并行访问存储器

常规的主存是单体单字存储器，只包含一个存储体。在高速的计算机中，普遍采用并行主存系统，即在一个存取周期内可以并行读出多个字，依靠整体信息吞吐率的提高，以解决CPU与主存之间的速度匹配问题。

多个并行工作的存储器共有一套地址寄存器和译码电路，按同一地址并行地访问各自的对应单元。

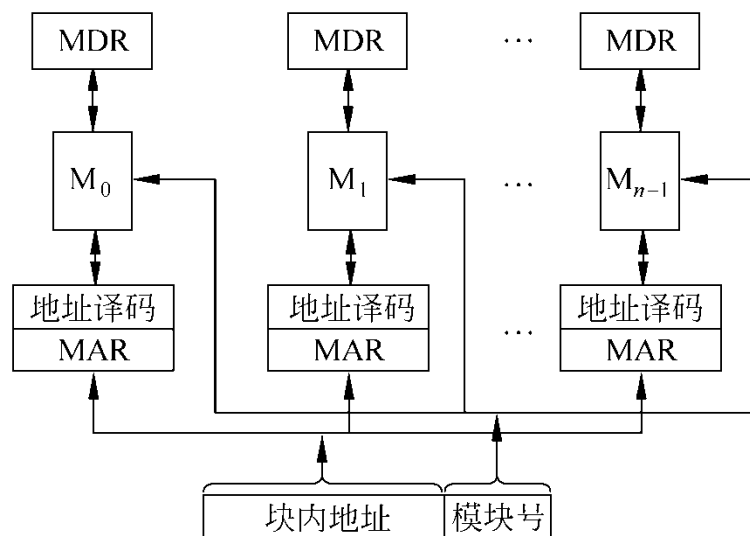
并行访问存储器按地址在一个存取周期内可读出 $n \times w$ 位的指令或数据，使主存带宽提高 n 倍。

5.6 多体交叉存储技术

5.6.2 交叉访问存储器

交叉访问存储器中有多个容量相同的存储模块（存储体），而且各存储模块具有各自独立的地址寄存器、读写电路和数据寄存器，这就是多体系统。各个存储体能并行工作，又能交叉工作。

多体交叉访问存储器地址寄存器的低位部分经过译码选择不同的存储体，而高位部分则指向存储体内的存储字。



1.有一个 $16\text{K} \times 16$ 位的存储器，由 $1\text{K} \times 4$ 位的 DRAM 芯片构成（芯片是 64×64 结构）。问：

(1)共需要多少 RAM 芯片？

(2)存储体的组成框图。

2.某半导体存储器容量 $7\text{K} \times 8\text{b}$ 。其中固化区 $4\text{K} \times 8\text{b}$ 可选 EPROM 芯片： $2\text{K} \times 8\text{b}/\text{片}$ ；随机读写区 $3\text{K} \times 8\text{b}$ ，可选 SRAM 芯片 $2\text{K} \times 4\text{b}/\text{片}$ 、 $1\text{K} \times 4\text{b}/\text{片}$ 。地址总线 $A_{15} \sim A_0$ (低)，双向数据总线 $D_{7} \sim D_0$ (低)，R/W 控制读/写。另有控制信号 MREQ，低电平是允许存储器工作。设计并画出存储器逻辑图。