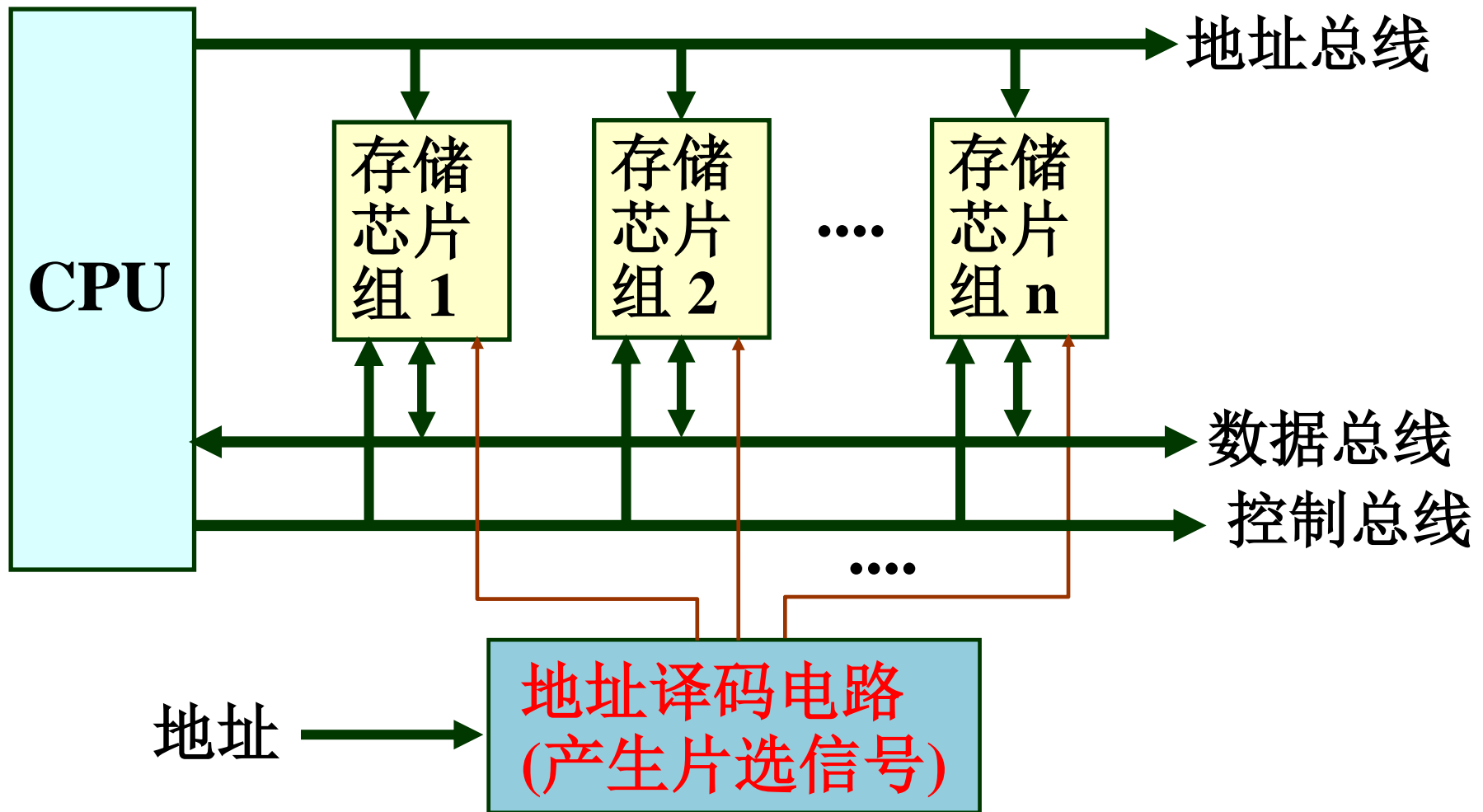




8.3 主存储器的组织

如何由存储芯片→存储器？

- ◆ 确定设计容量 → 多大设计容量
- ◆ 芯片的选用 → 多大容量, 位数, 多少片, 如何分组
- ◆ 地址分配 → 将CPU提供的全部地址空间的哪些地址空间分配给所设计的存储器, 确定各芯片组的片内地址
- ◆ 片选逻辑 → 高位地址产生芯片组所需的片选信号
- ◆ 信号线的连接 → 所有信号线如何连接



涉及地址分配、译码选片和译码选单元等。

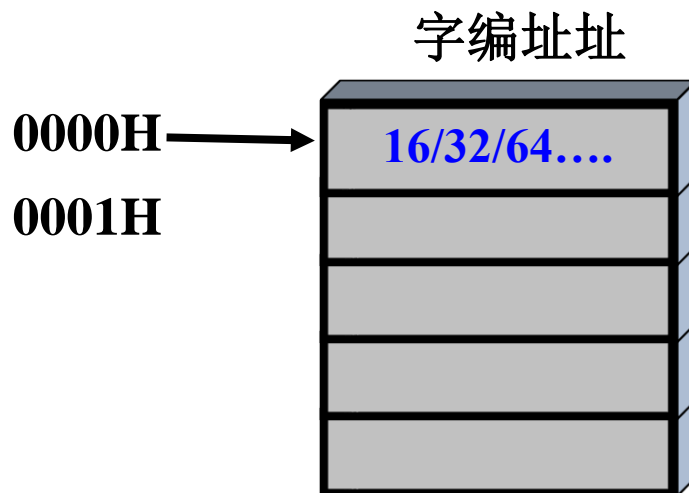
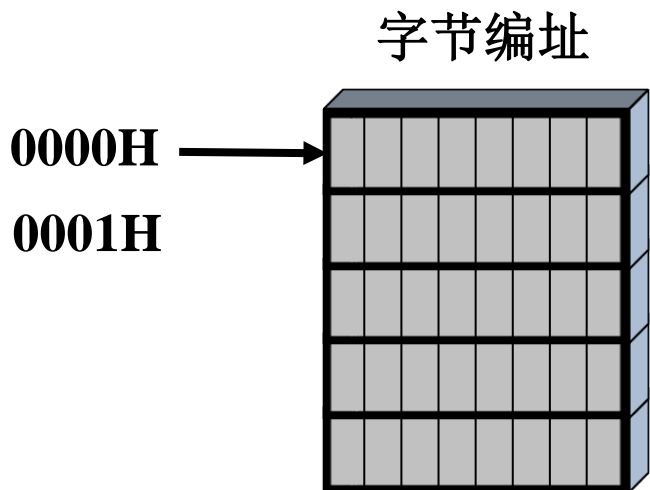


8.3.1 主存容量的扩展

主存容量 = 字数 × 位数

字数指可编址存储单元数

位数指每个编址存储单元的位数



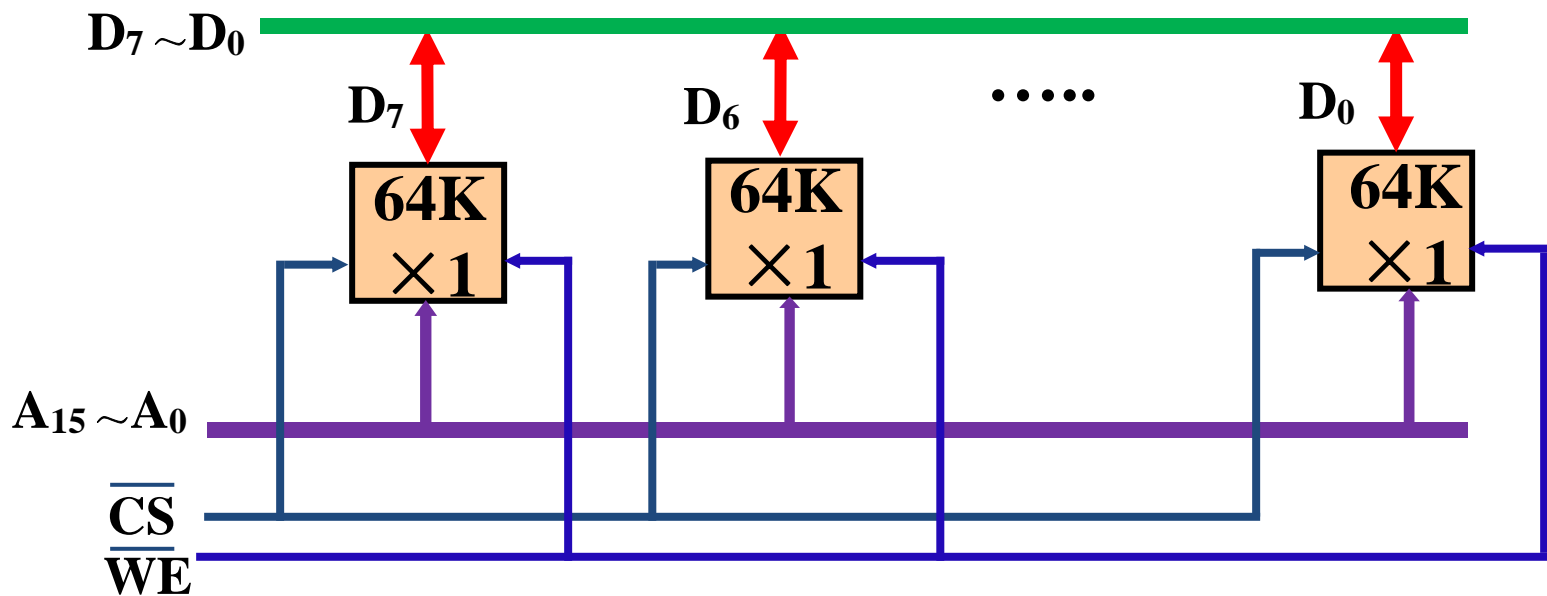


1.位扩展

位扩展指只在**位数方向**扩展。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**64K×1**的SRAM**存储芯片**组成**64K×8**的**存储器**，需要**8**块芯片。

	容量	地址	数据
存储器	64K×8	16	8
存储芯片	64K×1	16	1



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。

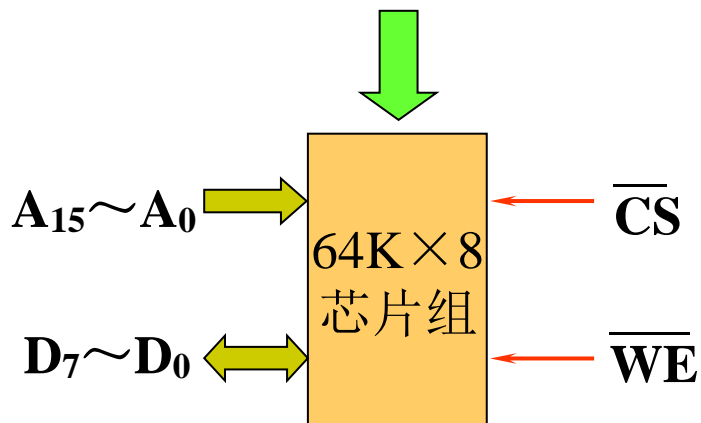
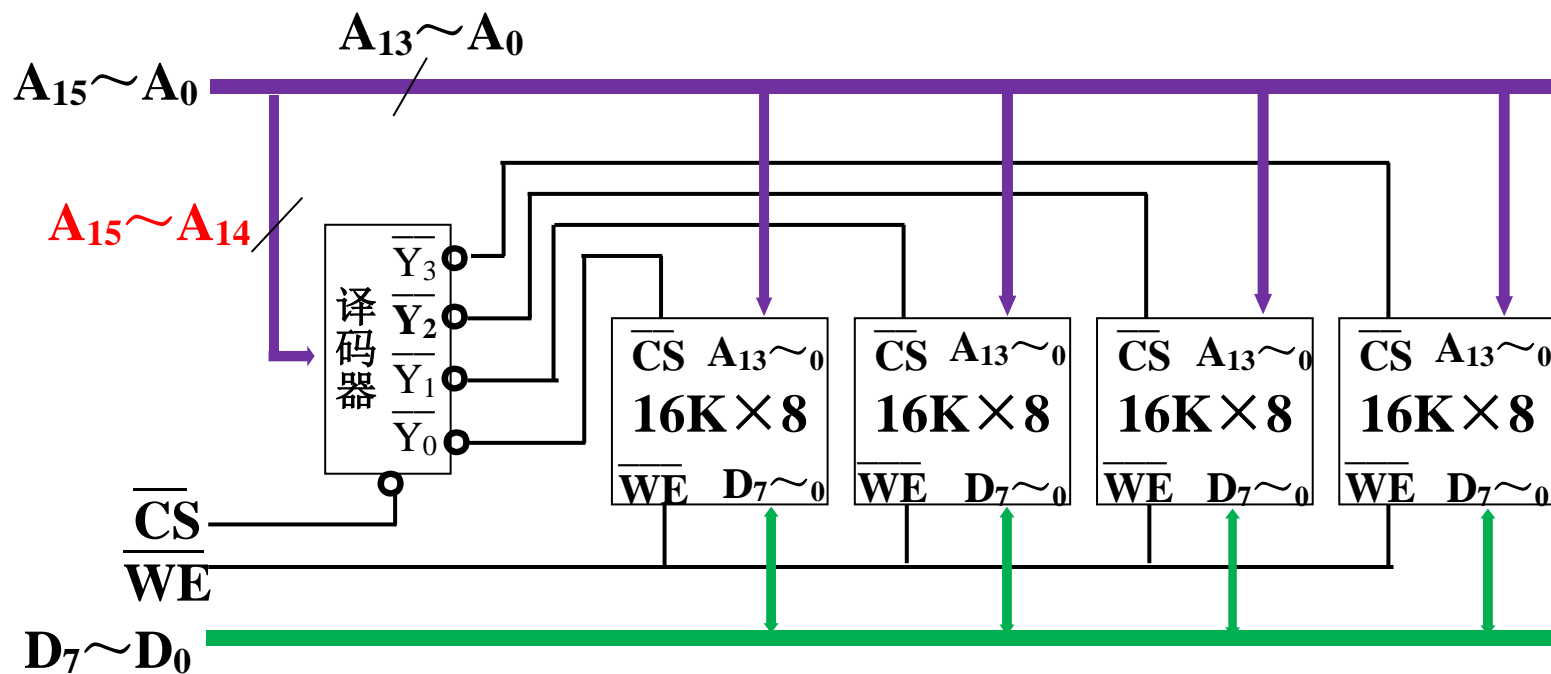


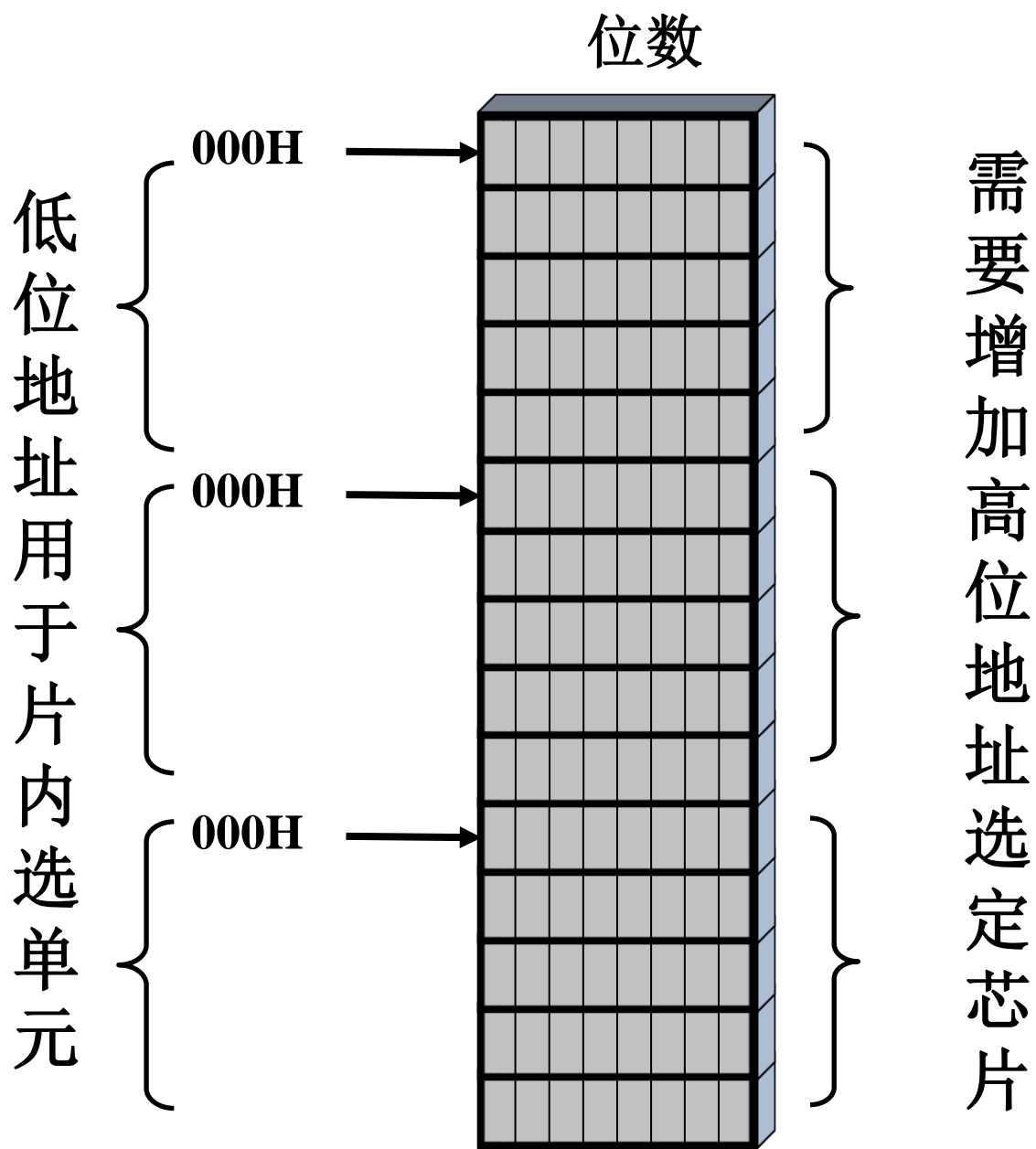
2. 编址空间扩展（字扩展）

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**数据线**、**读/写线**并联，由片选信号（**高位地址线译码**）选定存储芯片，**低位地址线**直接连接各芯片。

如用**16K×8**的SRAM组成**64K×8**的存储器，需要4块芯片。

	容量	地址	数据
存储器	64K×8	16	8
存储芯片	16K×8	14	8





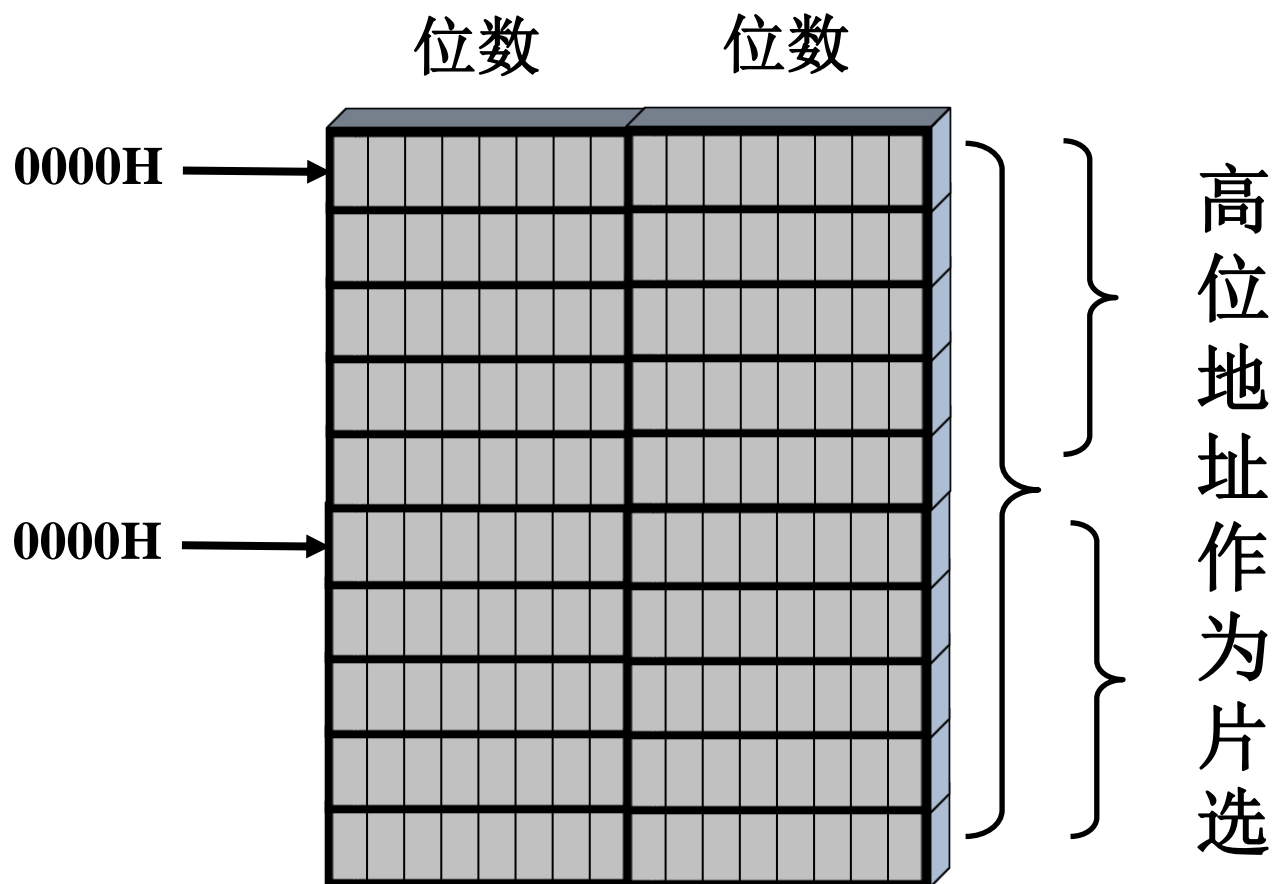


3.字和位同时扩展

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。

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



例1. 用2114(**1K×4**)SRAM芯片组成容量为**4K×8**的存储器，存储器的起始地址为1000H。假设CPU地址总线A₁₅~A₀，数据总线D₇~D₀，读/写信号线R/ \overline{W} 。
给出芯片地址分配与片选逻辑，画出存储器框图

设计步骤:

1. 根据存储器设计容量计算芯片数与分组

先扩展位数，再扩展单元数

2片1K×4 → 1K×8
4组1K×8 → 4K×8 } 8片



2. 地址分配与片选逻辑

- 1)明确CPU可访问存储空间： CPU地址总线 $A_{15} \sim A_0$
- 2)存储器容量设计及芯片分组：存储器所需的地址线
- 3)为每组芯片分配哪几位地址,以寻找片内的存储单元;
- 4)由哪几位地址形成芯片选择逻辑,以寻找各芯片组。

因为， 4KB存储器（需12位地址）由4组芯片组成，
且每组为1KB（需10位地址），在16位地址
空间(64KB)中的起始地址为1000H。

A15...A12 **A11** **A10** A9 A0

64KB

0 0 0 1 0	0	0 0			
0 0 0 1 0	0	1 1	1K×4	1K×4	1000-13FF
0 0 0 1 0	1	0 0	1K×4	1K×4	
0 0 0 1 0	1	1 1	1K×4	1K×4	1400-17FF
0 0 0 1 1	0	0 0	1K×4	1K×4	
0 0 0 1 1	0	1 1	1K×4	1K×4	1800-1BFF
0 0 0 1 1	1	0 0	1K×4	1K×4	
0 0 0 1 1	1	1 1	1K×4	1K×4	1C00-1FFF

4KB存储器的存储空间为1000H-1FFFH



片选逻辑:

第一组: $A_{15} \dots A_{12} A_{11} A_{10} = 000100$ (=CS0)

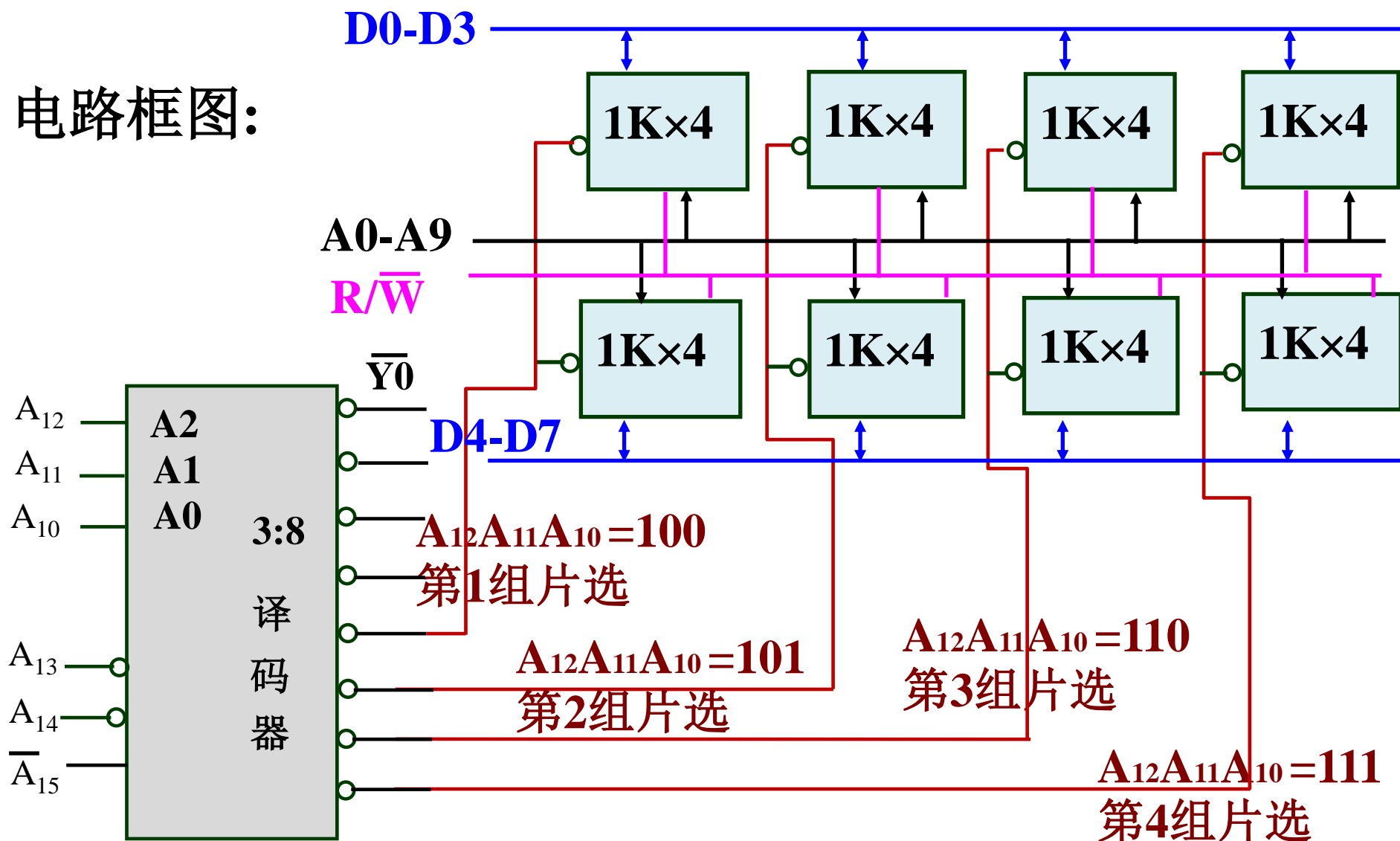
第二组: $A_{15} \dots A_{12} A_{11} A_{10} = 000101$ (=CS1)

第三组: $A_{15} \dots A_{12} A_{11} A_{10} = 000110$ (=CS2)

第四组: $A_{15} \dots A_{12} A_{11} A_{10} = 000111$ (=CS3)

如果使用3:8译码器来产生片选信号, 可将上述状态看作: $A_{15}A_{14}A_{13}=000$ 时, $A_{12}A_{11}A_{10}$ 分别=100、101、110、111。

电路框图:



将除了与芯片连接的地址以外的所有高位地址用于译码产生片选信号,称为全译码方式。

例2. 某存储器按字节编址。其中, 0000H~07FFH为ROM区, 选用EPROM芯片(2KB/片); 0800H~13FFH为RAM区, 选用RAM芯片(2KB/片和1KB/片)。地址总线A₁₅~A₀。给出地址分配和片选逻辑。

1. 计算容量和芯片数

ROM区: 2KB RAM区: 3KB 共3片(总容量5K)

(0800~0FFFH, 1000 ~13FFH)

2. 地址分配与片选逻辑

5KB需13位地址寻址: $A_{12} \sim A_0$

	A15	A14	A13	A12	A11	A10	A9	...	A0
0000 H	0	0	0	0	0	0	0	
07FFH	0	0	0	0	0	1	1	
0800 H	0	0	0	0	1	0	0	
0FFFH	0	0	0	0	1	1	1	
1000 H	0	0	0	1	0	0	0	0	
13FFH	0	0	0	1	0	0	1	1	

64KB

2KB
(ROM)

2KB
(RAM)

1KB
(RAM)

⋮

即: 对2K的ROM, 有 $A_{12}A_{11} = 00$

对2K的RAM, 有 $A_{12}A_{11} = 01$

对1K的RAM, 有 $A_{12}A_{11}A_{10} = 100$

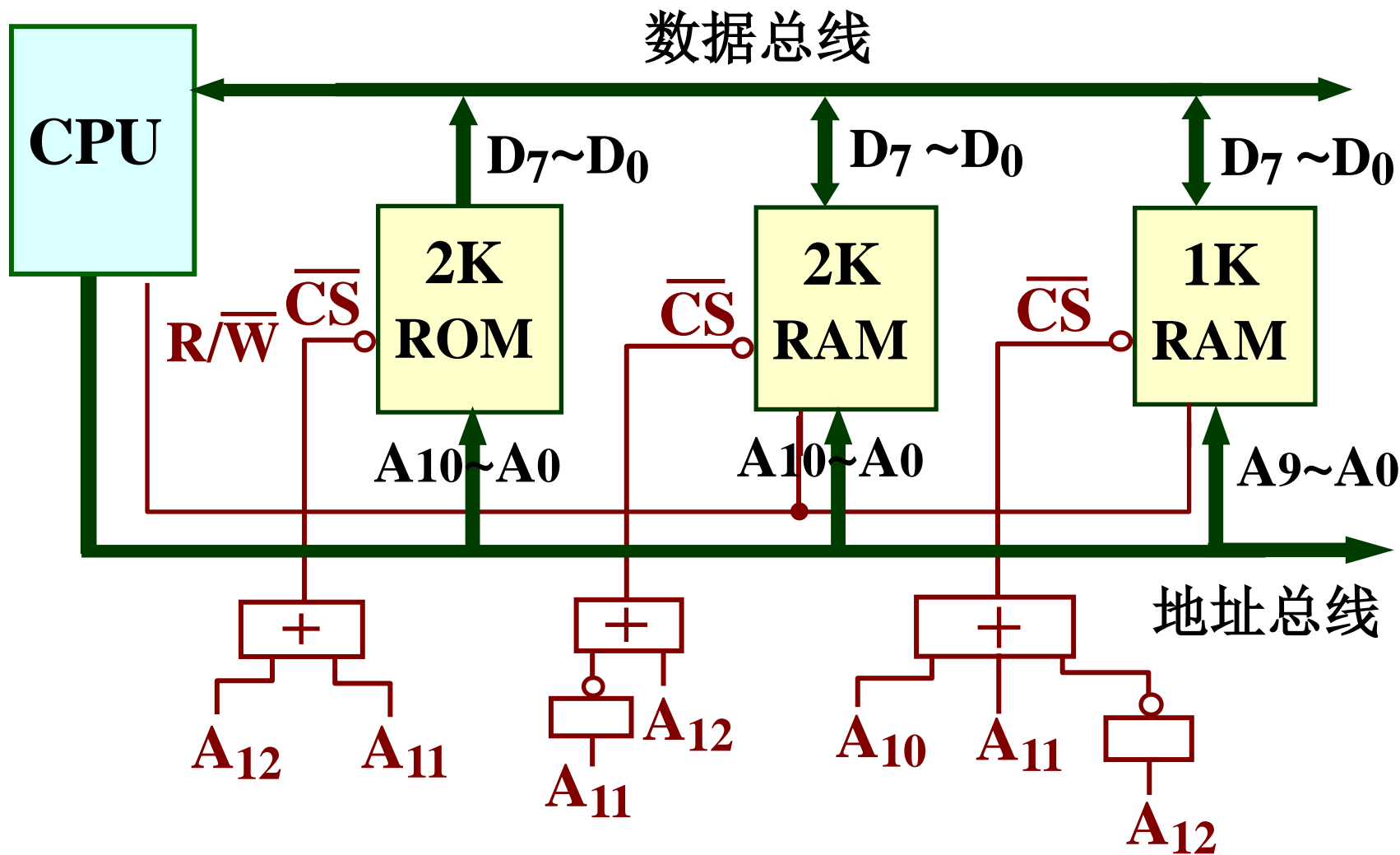
低位地址分配给芯片, 高位地址形成片选逻辑。

芯片	芯片地址	片选信号	片选逻辑
2K	A10~A0	CS0	$\overline{A}_{12}\overline{A}_{11}$
2K	A10~A0	CS1	$\overline{A}_{12}A_{11}$
1K	A9 ~ A0	CS2	$A_{12}\overline{A}_{11}\overline{A}_{10}$

$A_{15}A_{14}A_{13}$ 为全0

3. 连接方式

- 芯片每个单元长度为1个字节。
- 片选信号低电平有效 (假设ROM的片选信号有效时, 便能实现读操作)。
- 由于 $A_{15}A_{14}A_{13}$ 为全0, 在产生片选的译码电路中可以不体现。 (有无影响?)

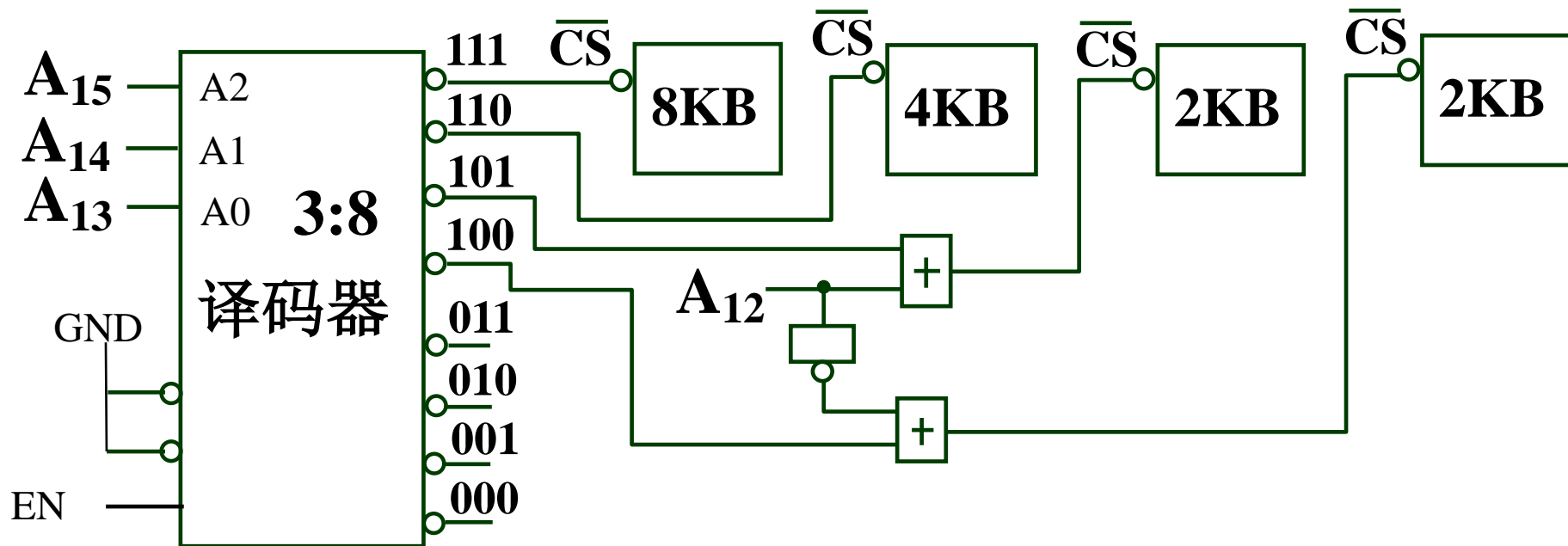


$$CS_0 = \overline{A}_{12}\overline{A}_{11} \quad CS_1 = \overline{A}_{12}A_{11} \quad CS_2 = A_{12}\overline{A}_{11}\overline{A}_{10}$$

将除了与芯片连接的地址外的部分高位地址用于译码产生片选信号,称为部分译码方式。



例3. CPU地址总线16条, 数据总线8条, 有以下存储器连接电路:



1. 写出每片芯片的地址范围;
2. 分析芯片地址重叠情况;



1. 写出每片芯片的地址范围

$A_{15}A_{14}A_{13}A_{12}A_{11}A_{10}A_9\dots A_2A_1A_0$

8KB	$\left\{ \begin{array}{l} \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \dots \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \\ \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \dots \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \end{array} \right\}$	E000~FFFF
4KB	$\left\{ \begin{array}{l} \mathbf{1} \quad \mathbf{1} \quad \mathbf{0} \quad \boxed{\mathbf{0}} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \quad \dots \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \\ \mathbf{1} \quad \mathbf{1} \quad \mathbf{0} \quad \boxed{\mathbf{1}} \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \quad \dots \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \end{array} \right\}$	C000~DFFF
2KB ^①	$\left\{ \begin{array}{l} \mathbf{1} \quad \mathbf{0} \quad \mathbf{1} \quad \mathbf{0} \quad \boxed{\mathbf{0}} \quad \mathbf{0} \quad \mathbf{0} \quad \dots \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \\ \mathbf{1} \quad \mathbf{0} \quad \mathbf{1} \quad \mathbf{0} \quad \boxed{\mathbf{1}} \quad \mathbf{1} \quad \mathbf{1} \quad \dots \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \end{array} \right\}$	A000~AFFF
2KB ^②	$\left\{ \begin{array}{l} \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{1} \quad \boxed{\mathbf{0}} \quad \mathbf{0} \quad \mathbf{0} \quad \dots \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{0} \\ \mathbf{1} \quad \mathbf{0} \quad \mathbf{0} \quad \mathbf{1} \quad \boxed{\mathbf{1}} \quad \mathbf{1} \quad \mathbf{1} \quad \dots \quad \mathbf{1} \quad \mathbf{1} \quad \mathbf{1} \end{array} \right\}$	9000~9FFF

4K地址范围 ←

8K地址范围 ←



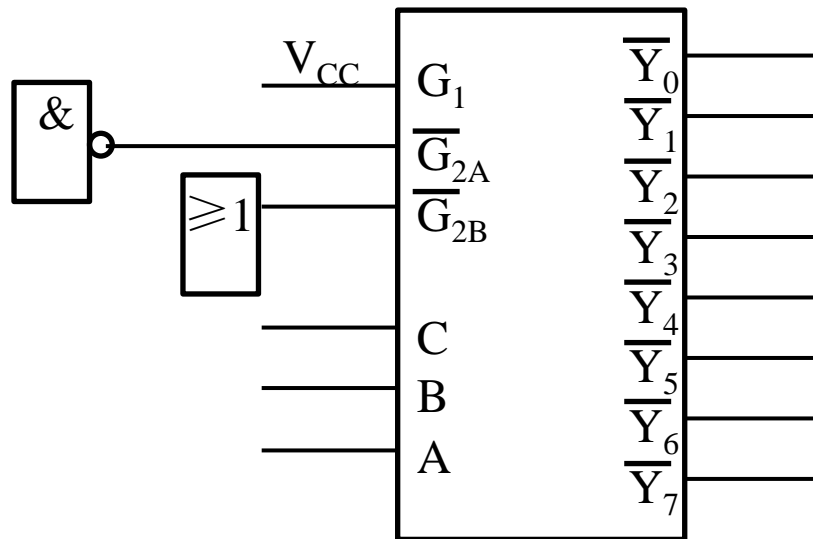
2. 分析芯片地址重叠情况

- (1) 8KB芯片地址空间E000~FFFF, 没有地址重叠;**
- (2) 4KB芯片地址空间C000~DFFF(即两个4K空间:
C000~CFFF、D000~DFFF), 有4K地址重叠;**
- (3) 2KB ①芯片地址空间A000~AFFF(即两个2K空间:
A000~A7FF、A800~AFFF), 有2K地址重叠;**
- (4) 2KB ②芯片地址空间9000~9FFF(即两个2K空间:
9000~97FF、9800~9FFF), 有2K地址重叠;**



例4，用 $2K \times 4b$ 的芯片(若干片)构成一个12KB的存储器，其地址范围在C0000~C2FFFH以及C4000~C6FFF之间。数据总线为D0~D7，地址总线为A0~A19(高地址)，对芯片读写采用 R/\bar{W} 控制，且片选要求采用3-8译码器的输出，如下图所示。

- (1) 需要 $2K \times 4b$ 的芯片多少片构成12KB的存储器？
- (2) 每组芯片地址线如何分配？
- (3) 选择哪些地址线作为译码器的使能端？
- (4) 选择哪些地址线作为译码器的输入端？
- (5) 在不增加其他器件的条件下，画出存储器逻辑电路图。





解: $C2FFFH - C0000H = 2FFFH$

$C6FFFH - C4000H = 2FFFH$

$2FFFH + 1 = 3000H = 0011,0000,0000,0000B = 12KB$

因此属于部分译码，有地址重叠

$C0000H = 1100,0000,0000,0000,0000B$

$C4000H = 1100,0100,0000,0000,0000B$

可以得到A14地址线没有参与片选译码

(1) $12KB = (2K \times 4b \times 2) \times 6$ ，所以需要12片

(2) 由于每组为2KB，所以需要11根地址线用于片内单元译码，即地址线A0~A10。



(3) (4) 已知CPU地址线为A0~A19，每组芯片为A0~A10，同时A14不参与片选译码，因此能够参加片选译码的只有A11~A13，A15~A19。

由于共有6组芯片，所以用于译码器译码输入的必须有3位地址线，因此译码器的输入端为A11~A13。

所以，译码器的使能端为A15~A19。

$C0000H = 1100, 0000, 0000, 0000, 0000B$

$C4000H = 1100, 0100, 0000, 0000, 0000B$

得到，A19=A18=1, A17=A16=A15=0。



译码器使能

片选地址

译码器输入

第一组

C0000H= 1100,0000,0000,0000,0000B
~ C07FFH= 1100,0000,0111, 1111,1111B

第二组

C0800H= 1100,0000,1000,0000,0000B
~ C0FFFH= 1100,0000,1111, 1111,1111B

第三组

C1000H= 1100,0001,0000,0000,0000B
~ C17FFH= 1100,0001,0111, 1111,1111B

第四组

C1800H= 1100,0001,1000,0000,0000B
~ C1FFFH= 1100,0001,1111, 1111,1111B

第五组

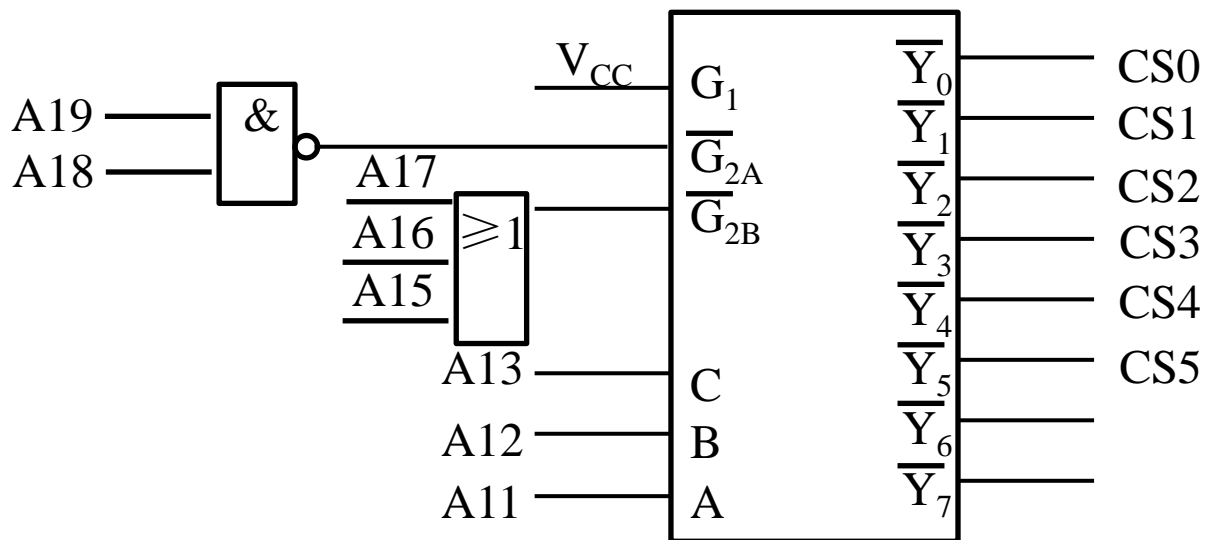
C2000H= 1100,0010,0000,0000,0000B
~ C27FFH= 1100,0010,0111, 1111,1111B

第六组

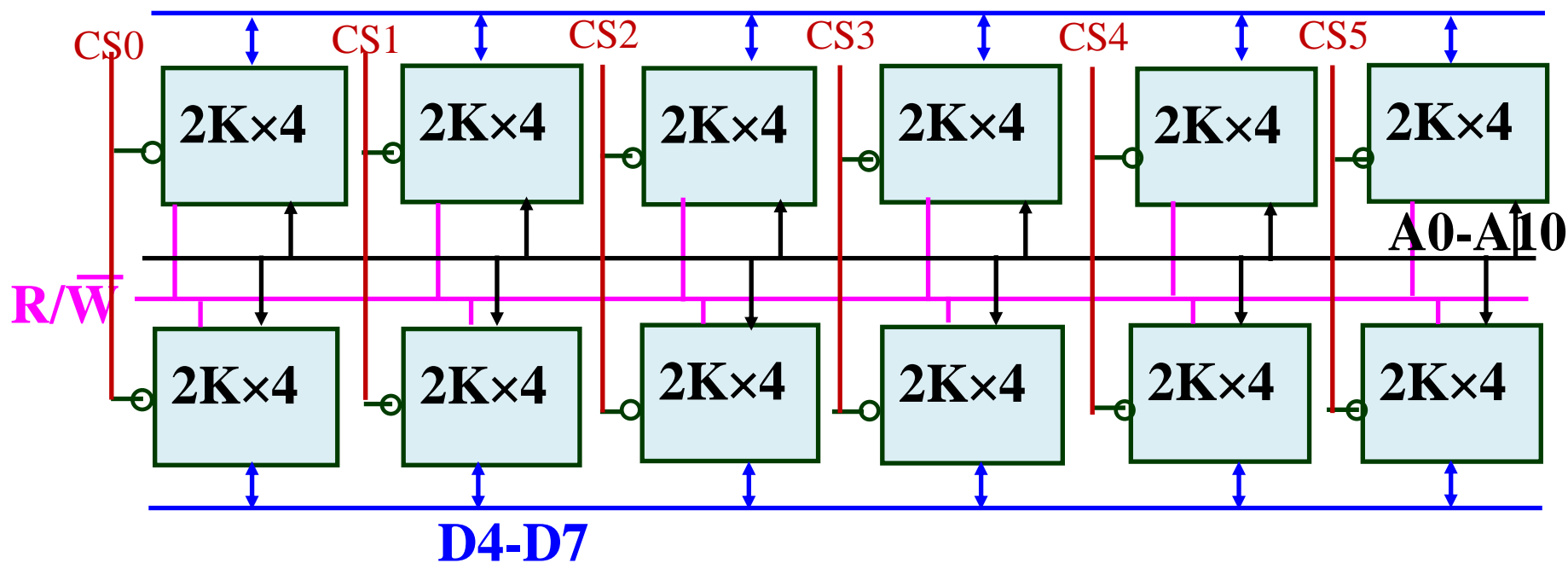
C2800H= 1100,0010,1000,0000,0000B
~ C2FFFH= 1100,0010,1111, 1111,1111B



(5) 逻辑电路图



D0-D3





8.4 数据校验

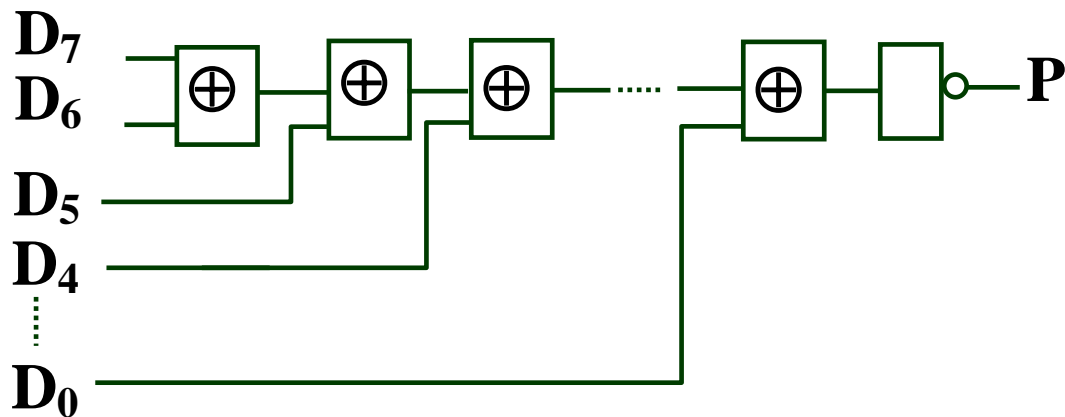
1、奇偶校验

(1) 奇校验

对给定的一个需传送的**代码** $D_{n-1} \sim D_0$, 设置一个**校验位** P , 要求 $PD_{n-1} \sim D_0$ (共 $n+1$ 位)中, “1”的个数须为奇数。校验位 P 连同代码 $D_{n-1} \sim D_0$ 一起传送。

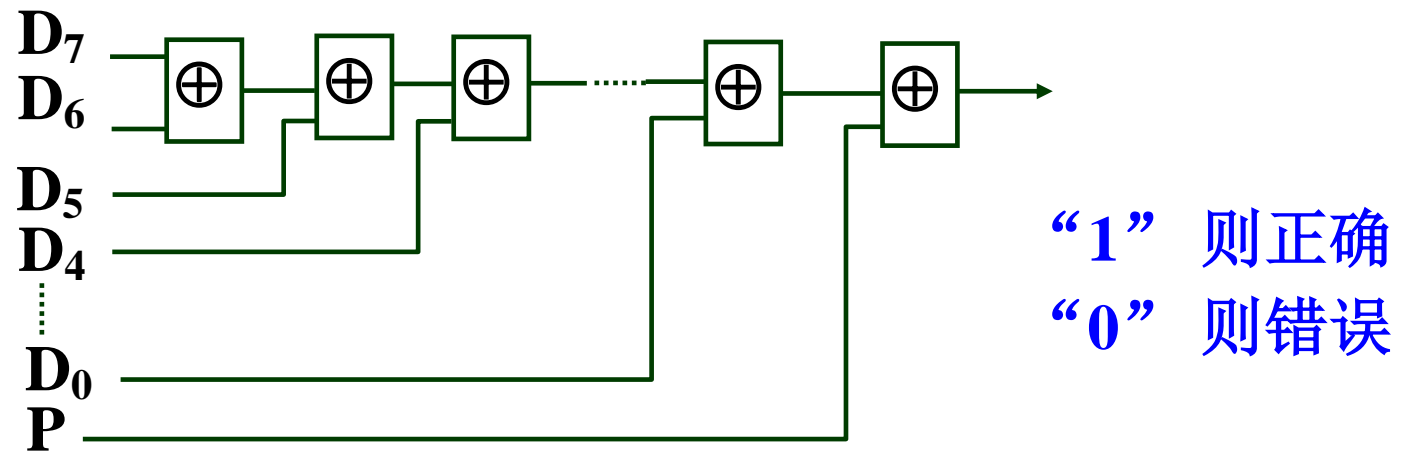
比如: 代码串10001101, 则 P 应为1, 形成110001101

电路: 对一个八位的数据 $D_7 \sim D_0$, 则:





对代码的接收方:



(2) 偶校验

对给定的一个需传送的代码 $D_{n-1} \sim D_0$, 设置一个校验位 P , 要求 $PD_{n-1} \sim D_0$ (共 $n+1$ 位)中, “1” 的个数须为偶数。校验位 P 连同代码 $D_{n-1} \sim D_0$ 一起传送。

奇偶校验简单, 但是只能校验一位错误, 且不能指出错误位置。