

# 存储系统

吕熠娜

厦门大学信息学院



# >>> 存储系统

- 随机读写存储器
- 只读存储器和闪速存储器
- Cache存储器
- 虚拟存储器
- 存储保护

# >>> 概述

存储器是计算机系统记忆设备，用来存放程序和数据。

**存储元**是存储器中最小的存储单位，可以存储一位二进制代码。由若干个存储元组成一个**存储单元**，再由许多存储单元组成一个**存储器**。

存储器有不同的分类方法。

## ★ 按存储介质分

- 半导体存储器：主要有MOS型存储器和双极型存储器两大类。
- 磁表面存储器：在金属或塑料基体上，涂覆一层磁性材料，用磁层存储信息，常见的有磁盘、磁带等。
- 光存储器：采用激光技术访问的存储器。

## ★ 按存取方式分

- 随机存储器：任何存储单元的内容都能被**随机存取**，且存取时间和存储单元的物理位置无关。如半导体存储器。
- 顺序存储器：只能按**某种顺序**来存取，存取时间和存储单元的物理位置有关。如磁带存储器。

磁盘存储器既不像随机存储器那样能随机地访问任一个存储单元，也不像顺序存储器那样完全按顺序存取，而是介于两者之间。存取信息时，第一步指向整个存储器中的某个小区域（磁盘上的磁道）；第二步在小区域内顺序检索，直至找到目的地后再进行读/写操作。其存取时间和信息的物理位置有一定关系。

## ★ 按存储器的读写功能分

- 只读存储器(ROM): 存储的内容固定不变, 只能读出而不能写入。
- 随机读写存储器(RAM): 既能读出又能写入。

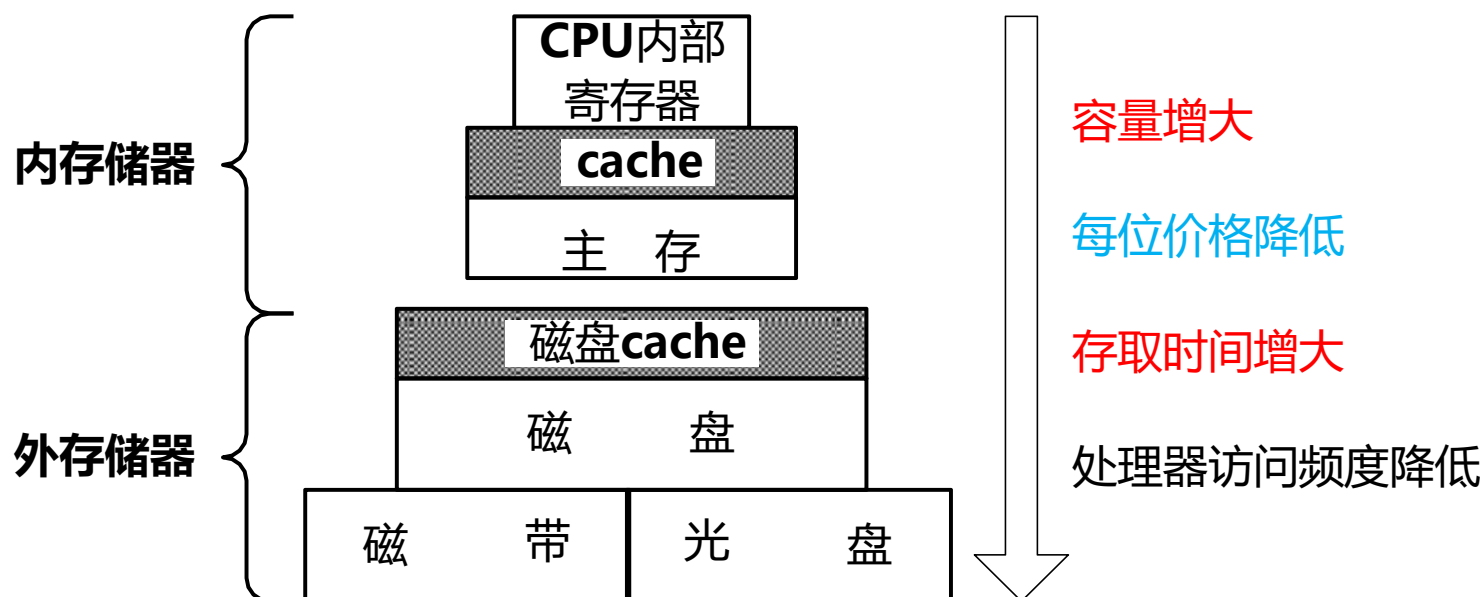
## ★ 按信息的可保存性分

- 易失性存储器: 断电后信息即消失的存储器。
- 非易失性存储器: 断电后仍能保存信息的存储器。

## ★ 按在计算机系统中的作用分

- 根据存储器在计算机系统所起的作用, 可分为主存储器、辅助存储器、高速缓冲存储器、控制存储器 (用于存放微程序, 由ROM构成) 。

# >>> 存储系统的层次结构



- 为了解决存储容量、存取速度和价格之间的矛盾，计算机中通常采用多级存储器体系结构，即使用高速缓冲存储器、主存储器和外存储器。CPU能直接访问的存储器称为**内存储器**，包括高速缓冲存储器和主存储器。CPU不能直接访问**外存储器**，外存储器的信息必须调入内存储器才能被CPU处理。



# >>> 存储系统的层次结构

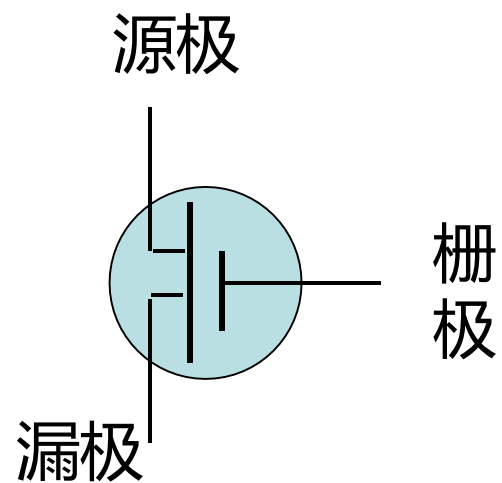
- **高速缓冲存储器 - 主存 层次：** CPU的处理速度比主存的存取速度快。为弥补主存速度的不足，在主存和CPU之间增加一级**高速缓冲存储器** (Cache)。其特点是速度高而容量小。它所存放的是**主存中部分内容的复制**，是当前最有可能被CPU访问的信息。
- 从整体看，Cache - 主存层次的存取速度接近于Cache的速度，而容量接近于主存的容量。Cache存储器全部由硬件调度，对程序员是透明的。
- **主存 - 辅存 层次：** 主存的存储量仍不能满足程序运行的要求，因此利用大容量、低价格的外部存储器作为辅助存储器。当前要用到或经常用到的信息存储在主存，未用到或不常用到的信息存储在辅存，需要时调往主存。主存和辅存一起构成了现在广泛使用的“虚拟存储系统”。
- 从整体看，主存 - 辅存层次具有接近于主存的速度和接近于辅存的容量。虚拟存储系统需要由操作系统来调度，因此对系统程序员是不透明的，但对应用程序员是透明的。

# >>> 存储器的性能指标

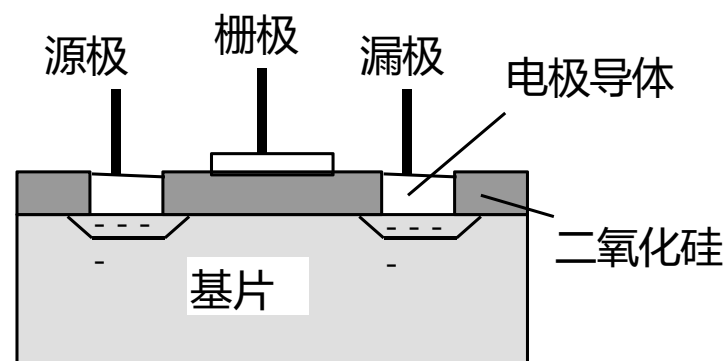
- **存储容量**：存储器所包含的存储单元的总数称为存储容量。存储容量用字数或字节数表示。一个字节定义为8个二进制位，一个字包括2个或4个字节。
- **存取时间**：从启动一次存储器操作到完成该操作所经历的时间。例如：读出时间是指从CPU向主存发出有效地址和读命令开始，直到将被选单元的内容读出为止所用的时间；写入时间是指从CPU向主存发出有效地址和写命令开始，直到信息写入被选中单元为止所用的时间。
- **存取周期**：连续两次访问存储器操作之间所需要的最短时间。一般情况下，存取周期大于存取时间。这是因为对于任何一种存储器，在读写操作之后，总要有一段恢复内部状态的复原时间。
- **存储器带宽**：又称数据传输率，指单位时间内存储器可读写的的数据量，用位/秒或字节/秒度量。由存取周期和字长决定。
- **可靠性**：用平均无故障时间来衡量。
- **其它参数**：功耗、价格等。



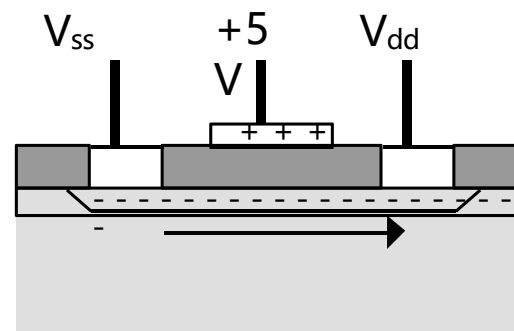
# >>> 场效应管



场效应管



(a) MOS 晶体管结构

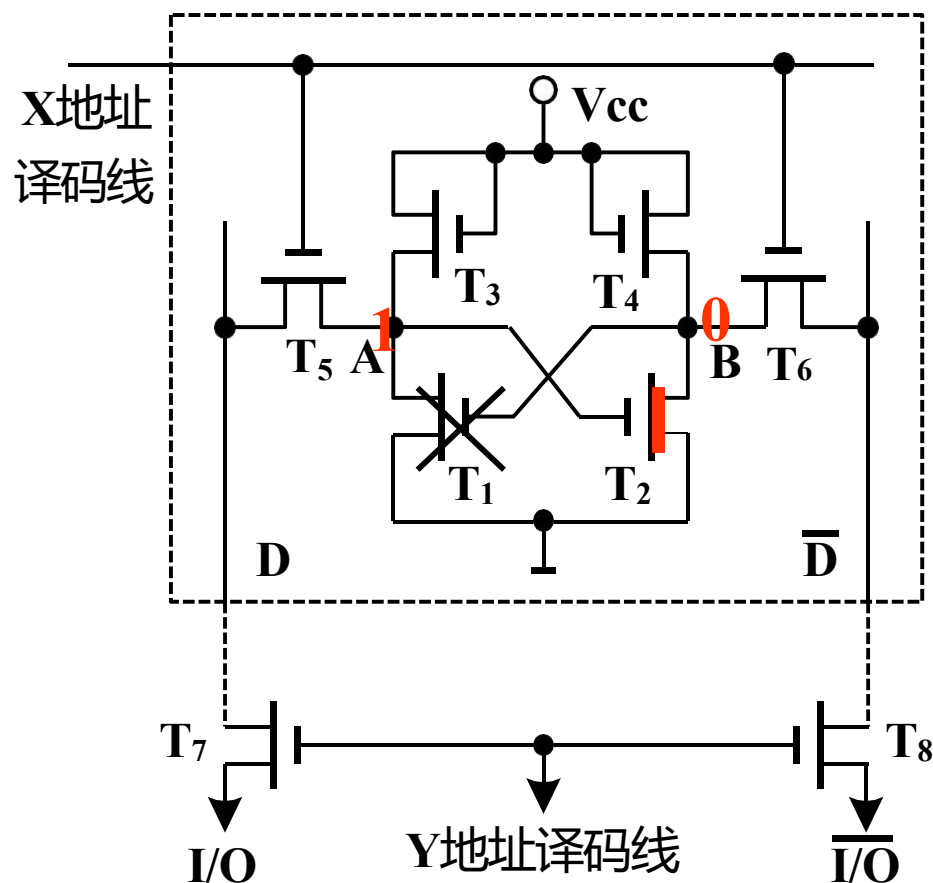


(b) MOS 晶体管导通状态

# 静态随机读写存储器SRAM

目前广泛使用的半导体存储器是MOS型半导体存储器，可以分为静态MOS型存储器（Static RAM）和动态MOS型存储器（Dynamic RAM）。

右图是六管SRAM存储元的电路图。T<sub>3</sub>、T<sub>4</sub>相当于负载电阻，T<sub>1</sub>和T<sub>2</sub>构成双稳态触发器。若T<sub>1</sub>截止，A为高电平，使T<sub>2</sub>导通，B为低电平，而B的低电平又使T<sub>1</sub>更加截止；反之，若B为高电平，则A为低电平。可见该电路有两个稳定状态，且A和B两点电位总是互反的。如果用A点高电平代表“1”，A点低电平代表“0”，该电路可存储一位二进制数。

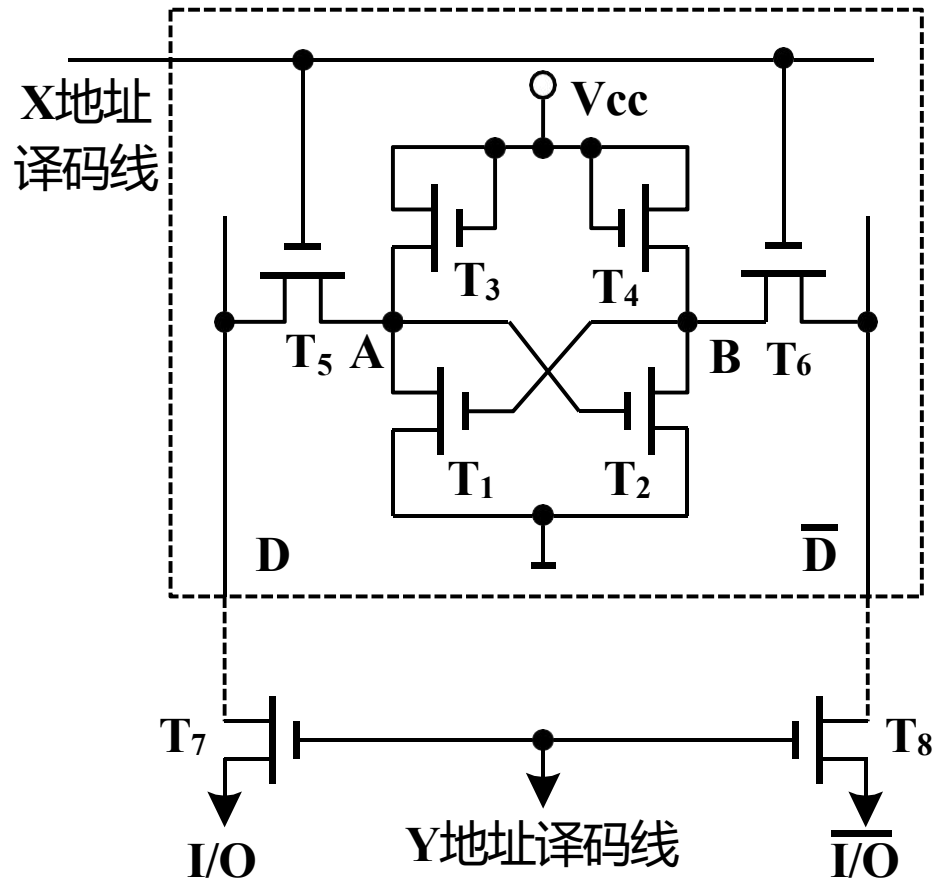


# 静态随机读写存储器SRAM

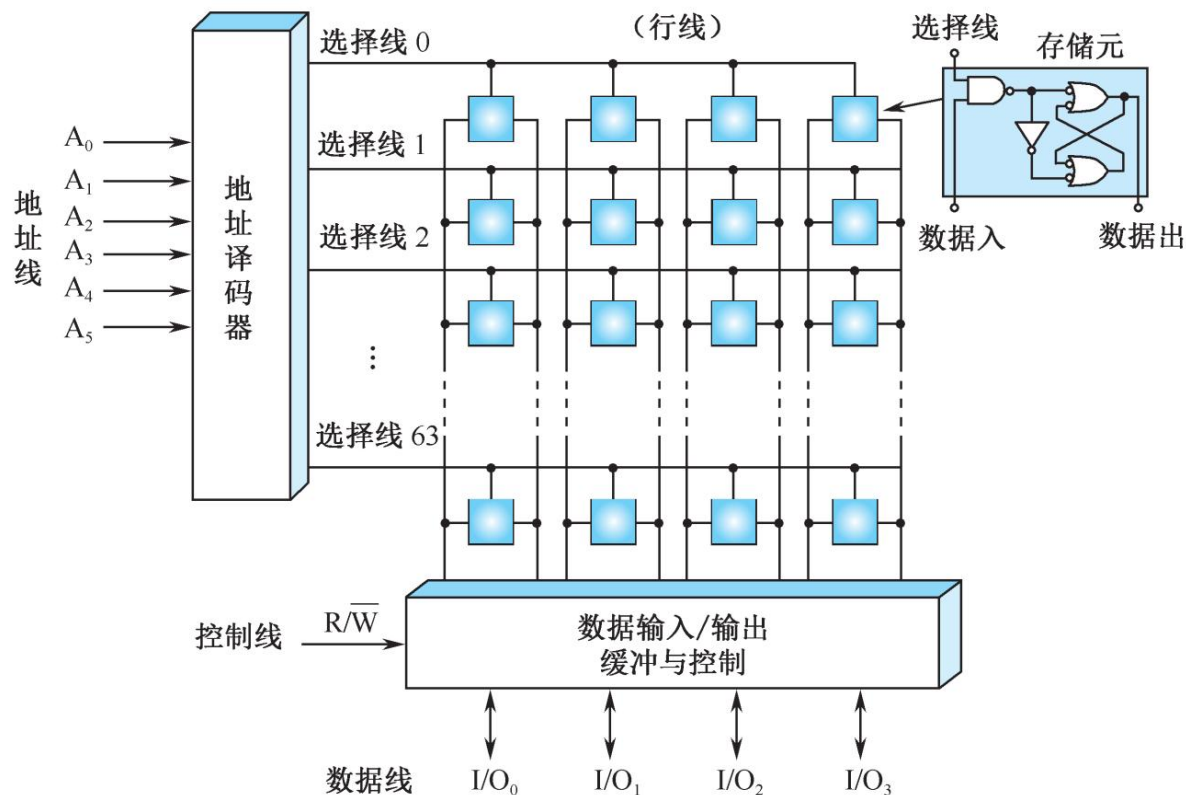
$T_5$ 、 $T_6$ 、 $T_7$ 和 $T_8$ 为控制管。如果某存储元被选中，X、Y地址译码线均处于高电平，使 $T_5 \sim T_8$ 导通，输入输出电路I/O和 $\overline{I/O}$ 分别与A点和B点相连，A点和B点的电平状态就能输出到I/O和 $\overline{I/O}$ 上，完成**读操作**。

**写操作**时，如果要写入“1”，在I/O线上输入高电位，在 $\overline{I/O}$ 线上输入低电位，开启 $T_5 \sim T_8$ 四个MOS管把高、低电位分别加在A、B点，使 $T_1$ 管截止，使 $T_2$ 管导通，将“1”写入存储元。

如果要写入“0”，在I/O线上输入低电位，在 $\overline{I/O}$ 线上输入高电位，打开 $T_5 \sim T_8$ ，把低、高电位分别加在A、B点，使 $T_1$ 管导通， $T_2$ 管截止，将“0”写入存储元。



# >>> 基本的静态存储元阵列



任何一个 SRAM，都有三组信号线与外部打交道：

- ①地址线，本例中有 6 条，即  $A_0$ 、 $A_1$ 、 $A_2$ 、 $A_3$ 、 $A_4$ 、 $A_5$ ，它指定了存储器的容量是  $2^6=64$  个存储单元。
- ②数据线，本例中有 4 条，即  $I/O_0$ 、 $I/O_1$ 、 $I/O_2$  和  $I/O_3$ ，说明存储器的字长是 4 位，因此存储位元的总数是  $64 \times 4=256$ 。
- ③控制线，本例中  $R/\overline{W}$  控制线，它指定了对存储器进行读( $R/\overline{W}$  高电平)，还是进行写( $R/\overline{W}$  低电平)。注意，读写操作不会同时发生。

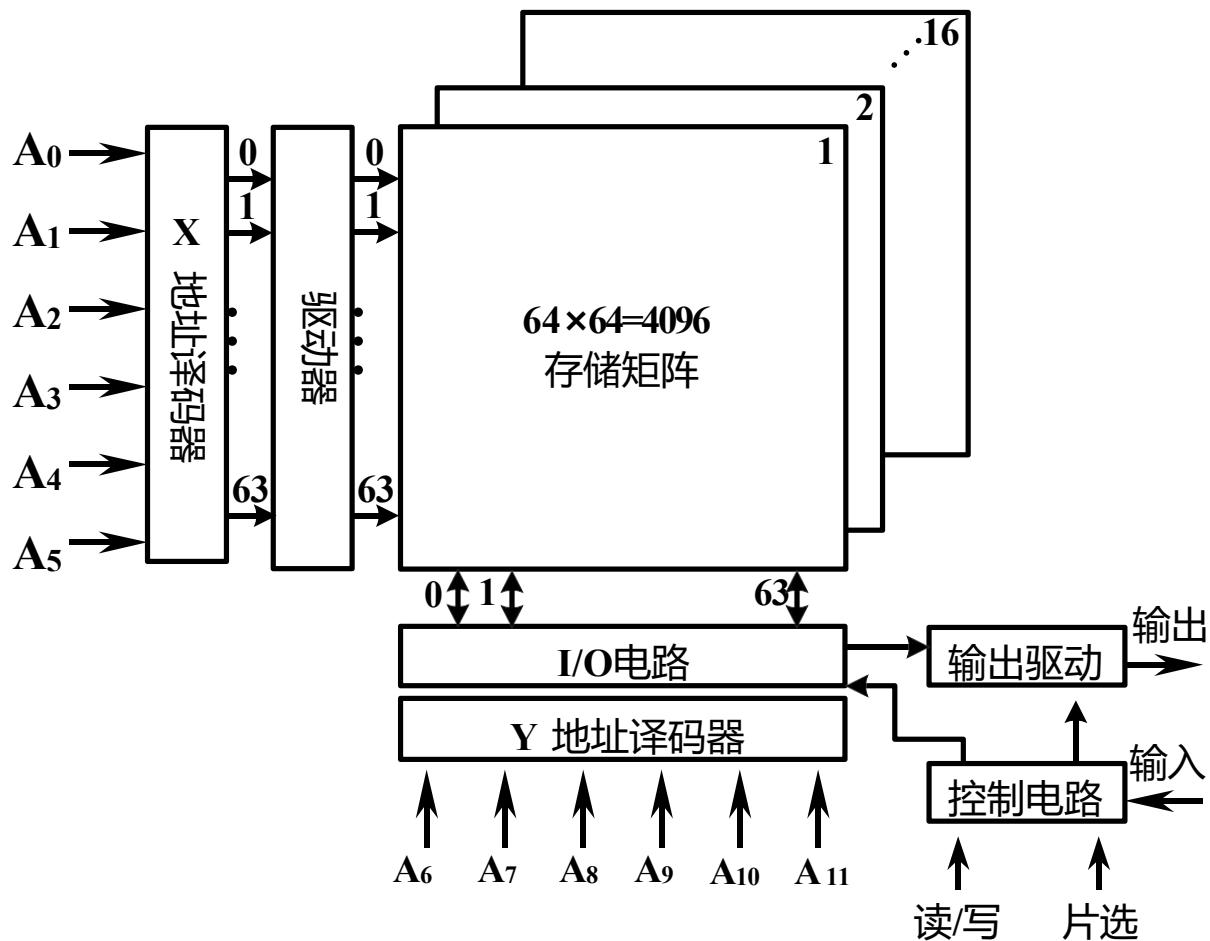


# SRAM存储器的组成

SRAM存储器由存储体、地址译码电路、读写电路和控制电路组成。

› **存储体：**存储体是存储单元的集合。

较大容量的存储器中，往往把各个字的同一位集成在一块芯片内，并排列成矩阵形式，由行选择线和列选择线进行选择。例如把4096个字的同一位集成在一块4096×1位的芯片中，用16块这样的芯片就可以组成4096×16位的存储器。

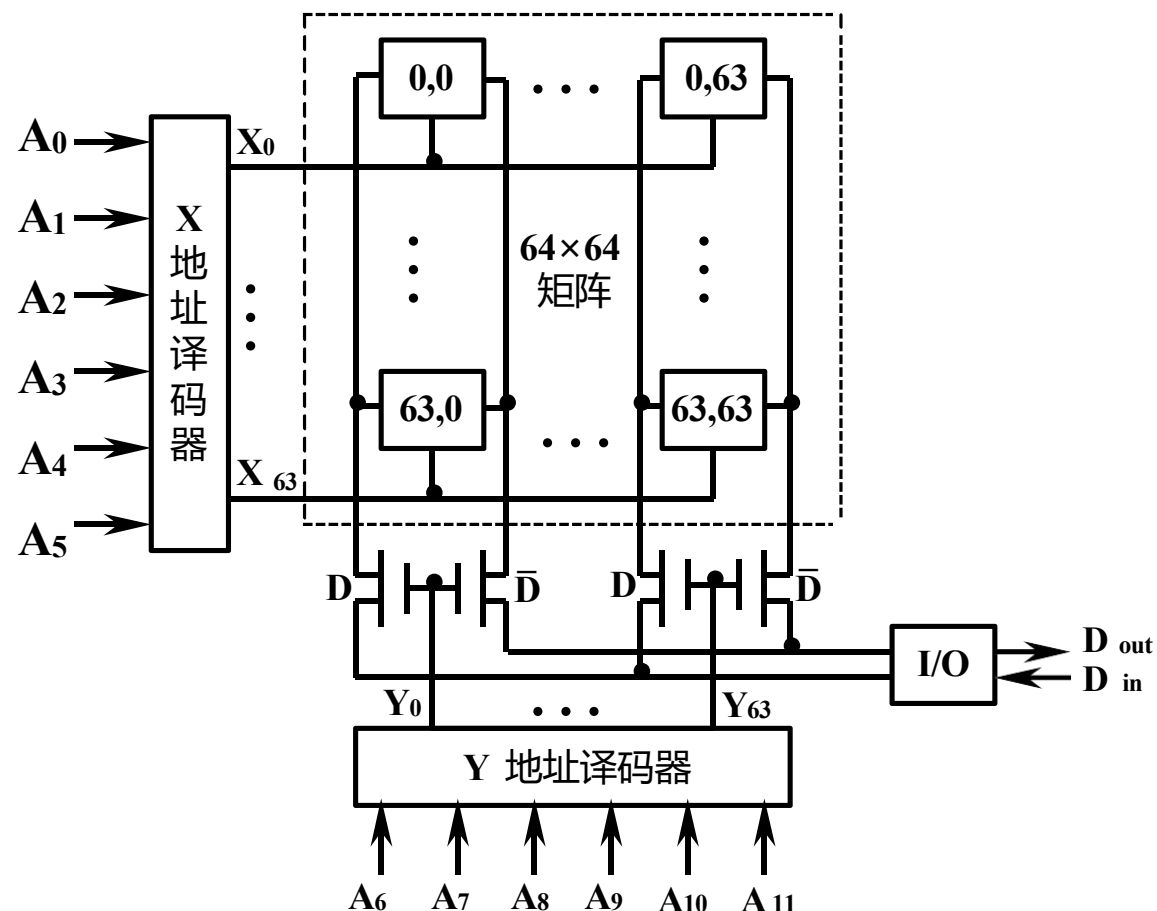


# SRAM存储器的组成

- **地址译码器：** CPU要选择某一存储单元，就在地址总线上输出该单元的地址信号给地址译码器，译码器把二进制代码表示的地址转换成电平信号，选中要访问的存储单元。

右图是双译码结构的地址译码器，包括X向和Y向两个译码器。若每个译码器有  $n/2$  个输入端， $2^{n/2}$  个输出端，两个译码器交叉译码的结果，可以选择  $2^{n/2} \times 2^{n/2} = 2^n$  个存储单元。这种结构可以减少译码线数目，适用于大容量存储器。

- **片选与读写控制电路：** 读写控制线用来控制芯片是进行读操作还是写操作；片选线用来决定该芯片是否被选中。



**在片选信号 $\overline{\text{CS}}$ 有效（低电平）情况下，写命令 $\overline{\text{WE}}$ 有效（低电平）进行写操作； $\overline{\text{WE}}$ 无效（高电平）进行读操作。**



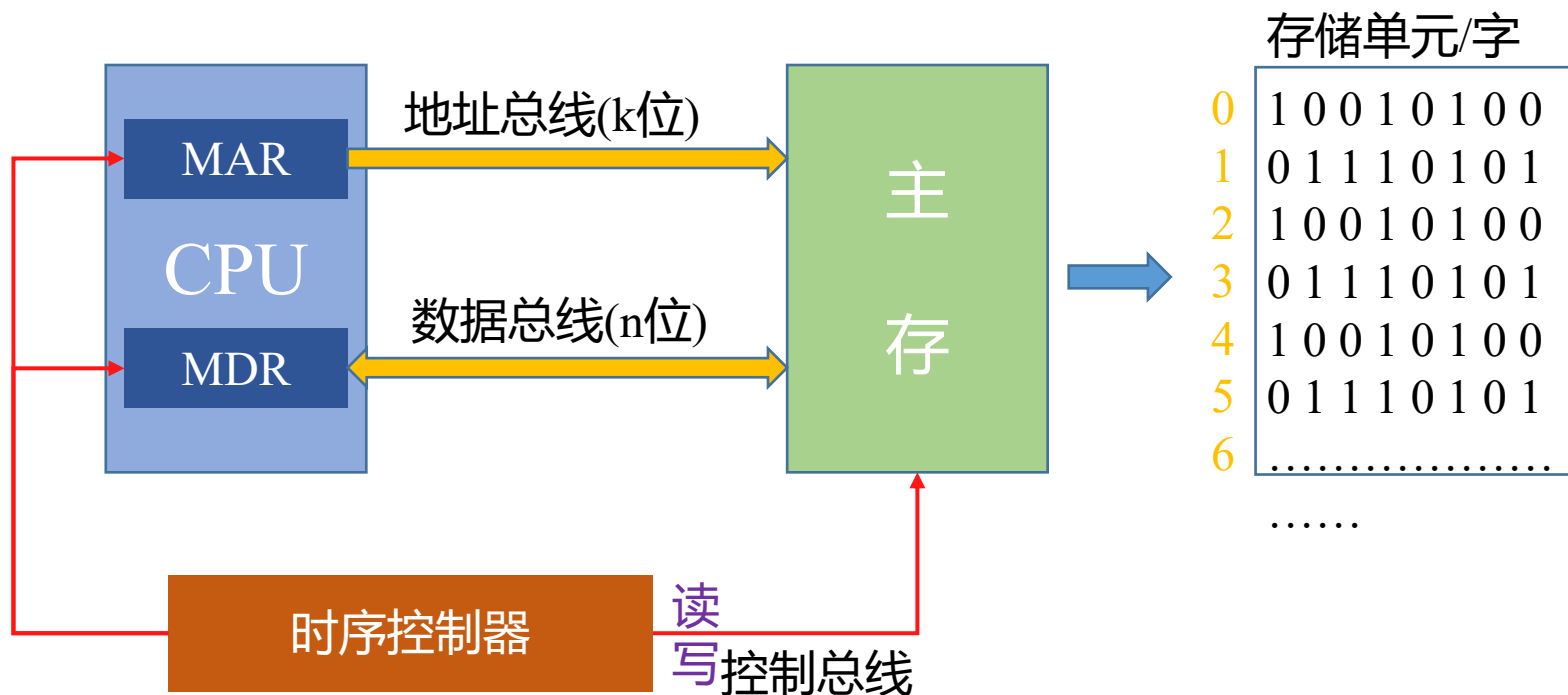
# >>> 存储器与CPU的连接

## • 数据总线

- 存取周期: 两次连续存取的最小时间间隔
- 存储器总线宽度  
= 数据总线位数  
= 存储字长
- 数据传输率  
= 存储器总线宽度 / 存取周期
- 单位时间内存取二进制信息的位数

## • 地址总线

## • 控制总线



存储容量: 半导体存储芯片所能存储的二进制信息位数(单位bit)

可寻址最大范围:  $2^k \times n$  (单位bit, 1Byte=8bit)

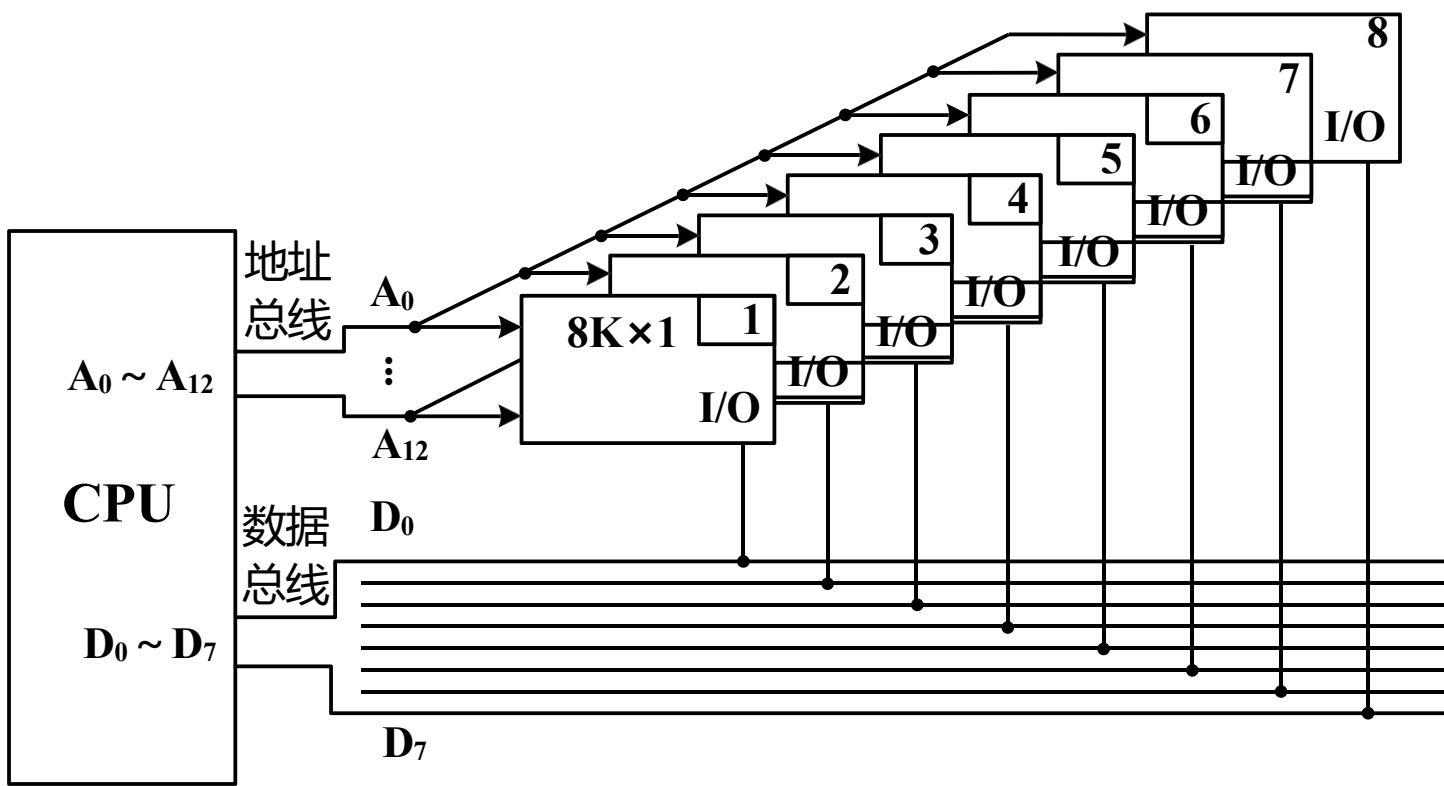


# >>> 存储器与CPU的连接

- RAM芯片通过地址线、数据线和控制线与外部连接。存储器同CPU的连接就是要完成**地址线、数据线和控制线**的连接。
  - 地址线是单向输入的，数据线是双向的，既可输入也可输出。
  - 单个芯片的存储容量往往不能满足要求，需要进行扩展。扩展的方法有：**位扩展法、字扩展法、字位同时扩展法**。
- 1、存储芯片存储容量为 $8K \times 1$ ， $16K \times 8$ 其含义分别表示什么？
  - 2、存储容量与地址线位数有关还是与数据线位数有关？
  - 3、假设某存储器的地址线为A0~A12，数据线为D0~D7，从地址单元001(H)中读出的数据为03H，则地址线，数据线上的信号分别为多少？
  - 4、能否从一片 $8K \times 1$ 的存储芯片中读出数据02 (H) ？
  - 5、常见的数字换算关系， $1K = 2^{10} =$  400 (H)

# >>> 位扩展法

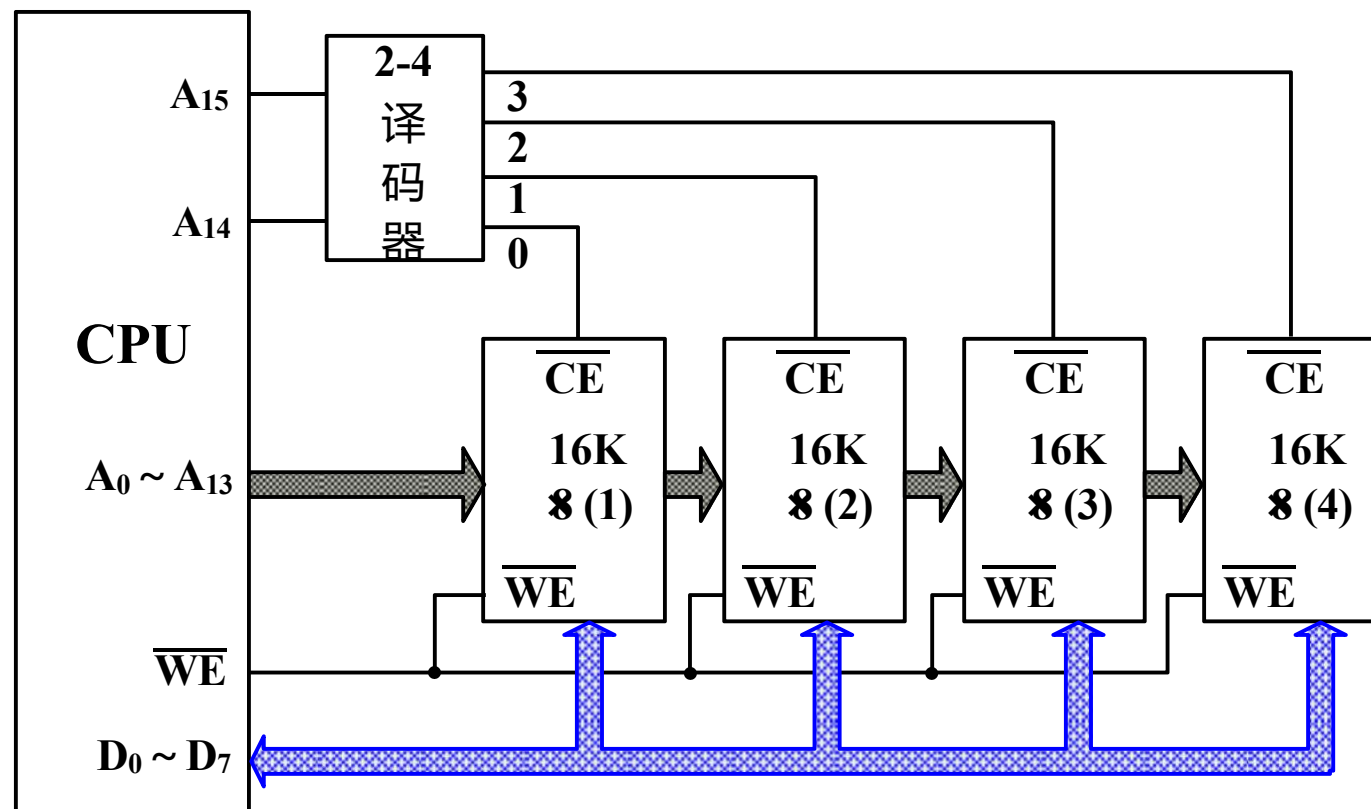
**位扩展法：**如果存储器芯片的字数满足要求，而位数不够，需进行位扩展。  
例如：将 $8K \times 1$ 的芯片组成 $8K \times 8$ 的存储器。方法是将芯片的地址线、控制线并联，数据线分联。对片选信号 $\overline{CS}$ 没有要求，直接接地。每一条地址总线有8个负载，每一条数据总线接一个负载。当给出某个地址时，同时选中8个芯片，将8个芯片上的数据从数据总线上读入或写入。



# 字扩展法

**字扩展法：**如果存储器芯片的位数满足要求，而字数不够，需进行字扩展。方法是将芯片的低位地址线、数据线、读写控制线并联，利用高位地址线经译码后作为片选信号。下图利用4片16K×8芯片经字扩展组成64K×8存储器。

4个芯片的数据线与数据总线 $D_0 \sim D_7$ 相连，地址线与地址总线低位地址 $A_0 \sim A_{13}$ 相连，写允许信号 $\overline{WE}$ 与CPU的 $\overline{WE}$ 相连。高位地址 $A_{14}$ 和 $A_{15}$ 经译码器和4个片选端相连。





# 思考



- 1、要将多少位的低位地址线进行并联？多少位的高位地址进行译码？
- 2、地址译码器要如何选择？选择3：8译码，2：4译码还是其他？若需要用16K的芯片组成80K的存储器呢？
- 3、译码器的输出是高电平还是低电平？
- 4、写出各个芯片的起始地址
- 5、当地址为4001H时，是如何选中需要的存储单元？  
A0 ~ A13, A14, A15分别为多少？译码结果如何？

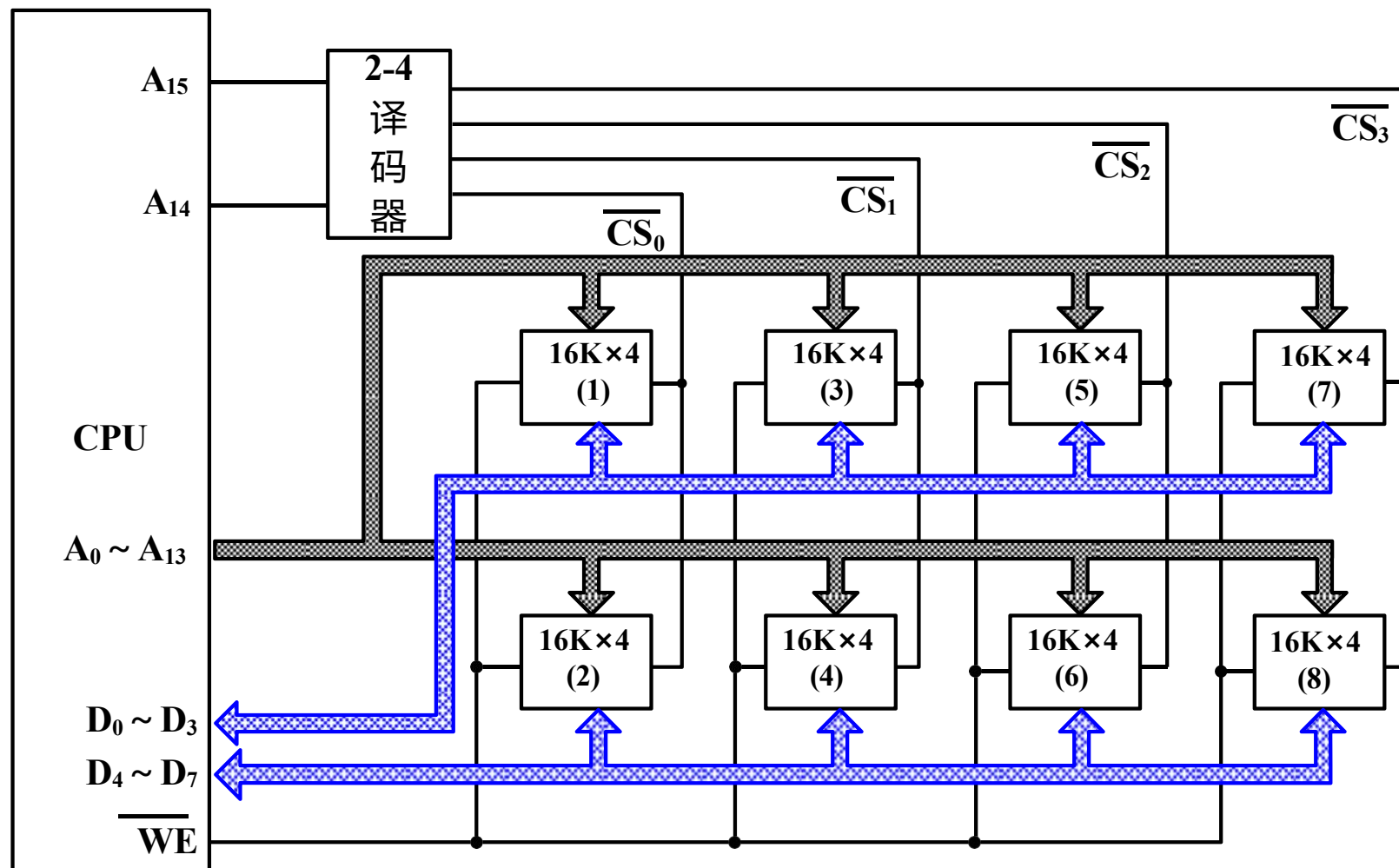
同一时刻4个芯片中只能有一个芯片被选中。 $A_{15}A_{14}=00$ 选中第一片,  
 $A_{15}A_{14}=01$ 选中第二片, .....。4个芯片的地址分配如下:

	$A_{15}A_{14}$	$A_{13} \sim A_0$	
第一片 最低地址	00	00 0000 0000 0000	= 0000 H
最高地址	00	11 1111 1111 1111	= 3FFF H
第二片 最低地址	01	00 0000 0000 0000	= 4000 H
最高地址	01	11 1111 1111 1111	= 7FFF H
第三片 最低地址	10	00 0000 0000 0000	= 8000 H
最高地址	10	11 1111 1111 1111	= BFFF H
第四片 最低地址	11	00 0000 0000 0000	= C000 H
最高地址	11	11 1111 1111 1111	= FFFF H

# >>> 字位同时扩展法

**字位同时扩展法：**如果存储器芯片的位数和字数都不满足要求，就需要字位同时扩展。

用 $m \times n$ 位芯片构成 $M \times N$ 位存储器需要 $(M/m) \times (N/n)$ 个芯片。  
右图用8片 $16K \times 4$ 位芯片构成 $64K \times 8$ 位存储器。





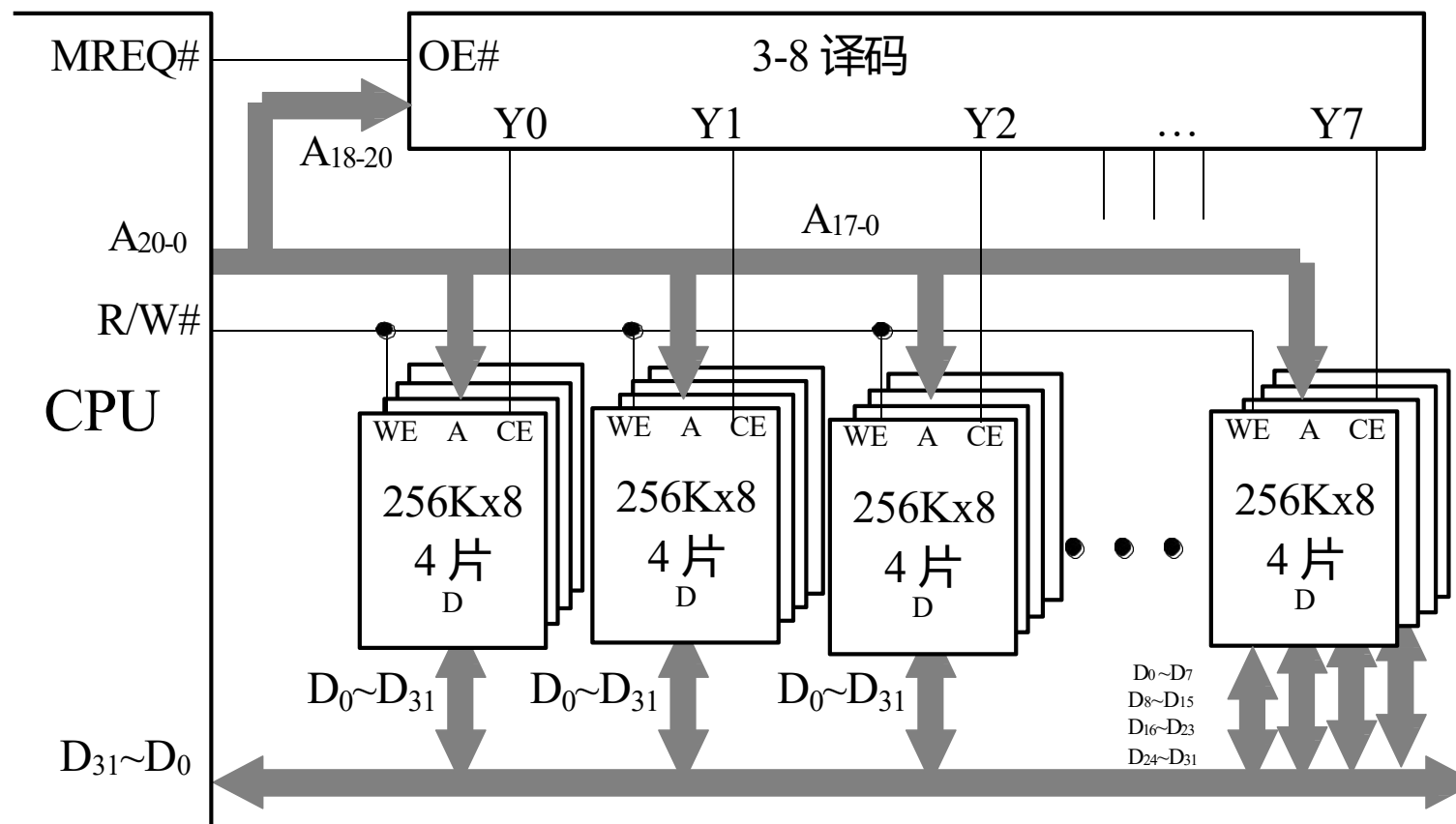
# 字位同时扩展法

- 字位同时扩展法是以上两种方法的综合，需要完成数据线、地址线、控制线的连接。先完成位扩展，按照位扩展的方式将数据线低位、高位分别接不同芯片，这些位扩展后的芯片形成一组。按照字扩展的方式对这些芯片组的地址线进行连接。注意译码器的选择。最后完成控制线（如读写信号）的连接。

**例1：256K×8的芯片组成2M×32的存储器**

# 字位同时扩展法

## 例1：256K×8的芯片组成2M×32的存储器





## 存储器与CPU的连接完成数据线、地址线、控制线的连接

**数据线：**位扩展的方式，位数不够时将高位、低位数据线分别连接；数据线是双向的（ROM除外）。

**控制线：**读写信号并联接入CPU；译码器的片选信号也直接接入CPU的访存允许；存储器片选信号由高位地址经过译码器得到的结果给出，注意两种情况：一个译码结果对应多个芯片的片选，或几个译码结果对应一个芯片的片选，有时还需要与低位一起经过门电路后进行片选。RW读写控制信号要接RAM。

**地址线：**高位进行译码，低位直接接入芯片，注意接入芯片进行片内选址的地址位数，芯片多大，就需要多少根片内地址。两种情况：位数不够或位数太多。地址线为单向的。



# >>> 地址线

1、16K 8K 4K 2K 1K 别需要多少根地址线

14 13 12 11 10

2、5片8K芯片组成40K，高几位进行译码？

高3位， A13\A14\A15

3、A12\A13\A14三位进行地址译码时，一个译码的结果(Y0-Y7)对应多大的地址空间？

A0-A11,  $2^{12} = 4K$

4、若译码的结果(Y0-Y7)对应4K地址空间，如何选择其中的1K的地址空间？

$4K/1K = 4 = 2^2$  高2位来处理

$1K = 2^{10}$  A0-A9 接下去2位 A10 A11



## 若芯片的存储容量不一致，如何连接相应的连接线？

例2：为某8位微机系统设计一个具有40 KB RAM和8 KB ROM的存储器。  
RAM用SRAM芯片6264 (8 K×8位) 组成，地址从0000 H开始；ROM  
用 EPROM芯片2732 (4 K×8位) 组成，与RAM地址空间相连。

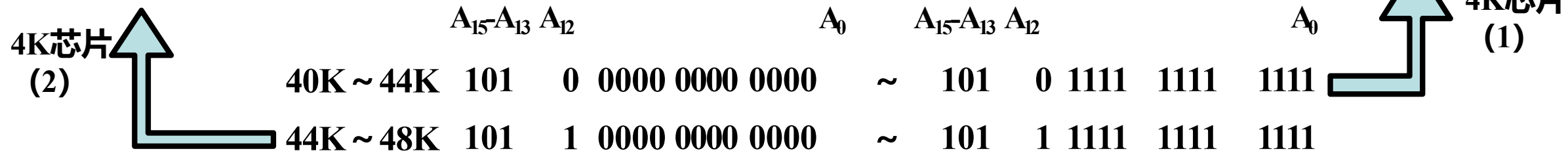
$\overline{Y5}$	A12	$\overline{CE}$
0	0	1
0	1	0
1	0	1
1	1	1

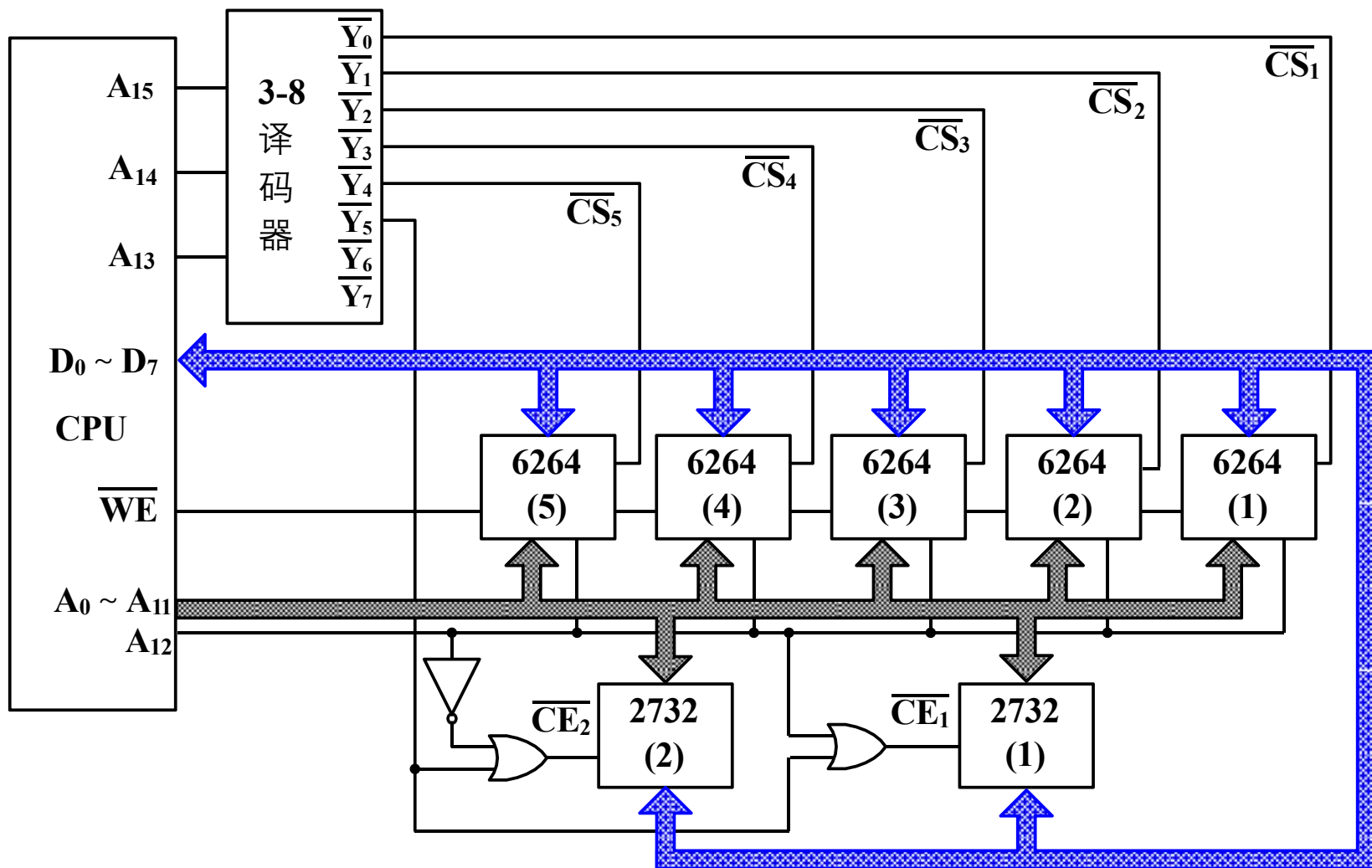
译码器如何选择？

8K需要多少根地址线？ 4K需要多少根地址线？

4K芯片的地址范围是多少？ 4K芯片的片选如何连接？

$\overline{Y5}$	A12	$\overline{CE}$
0	0	0
0	1	1
1	0	1
1	1	1







- **ROM芯片为只读存储器，  
所以数据只读不写，数据线为单向的，  
不需要读写控制信号RW**



**例3：CPU地址总线为A15 ~ A0，数据总线为D7 ~ D0， $\overline{\text{MREQ}}$ 为允许访存， $\text{R}/\overline{\text{W}}$ 为读写命令。地址空间分配如下：0 ~ 8 K为系统程序区，由8K × 8的ROM组成；8 K ~ 32 K为用户程序区；由8K × 8的RAM芯片组成，地址空间最后2K为系统程序区，由一片2K × 8的RAM芯片。请设计存储器系统。**



## 地址空间分配如下:

地址空间分配如下: 0 ~ 8 K为系统程序区, 由ROM组成; 8 K ~ 32 K为用户程序区; 最后2K为系统程序区。

A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	...	...	A <sub>0</sub>	
0	0	0	0	0	0	...	...	0	} 一片8 KB EPROM
0	0	0	1	1	1	...	...	1	
0	0	1	0	0	0	...	...	0	} 第一片8 KB SRAM
0	0	1	1	1	1	...	...	1	
0	1	0	0	0	0	...	...	0	} 第二片8 KB SRAM
0	1	0	1	1	1	...	...	1	
0	1	1	0	0	0	...	...	0	} 第三片8 KB SRAM
0	1	1	1	1	1	...	...	1	
...			...	...	...	空 (30 KB)			
1	1	1	1	1	0	...	...	0	} 一片2 KB SRAM
1	1	1	1	1	1	...	...	1	



## 最后8K地址空间分配如下:

Y7	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	... ..	A <sub>0</sub>		$\overline{\text{CE}}$
0	1	1	1	0	0	0	... ..	0	} 第一个2 KB 空间	1
0	1	1	1	0	0	1	... ..	1		
0	1	1	1	0	1	0	... ..	0	} 第二个2 KB 空间	1
0	1	1	1	0	1	1	... ..	1		
0	1	1	1	1	0	0	... ..	0	} 第三个2 KB 空间	1
0	1	1	1	1	0	1	... ..	1		
0	1	1	1	1	1	0	... ..	0	} 第四个2 KB 空间	0
0	1	1	1	1	1	1	... ..	1		





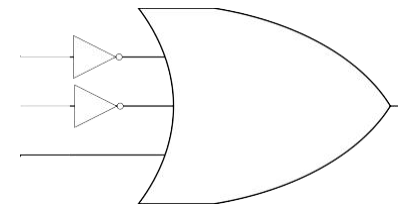
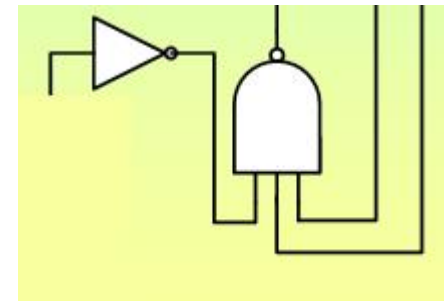
## 最后2K的片选逻辑电路如何设计？

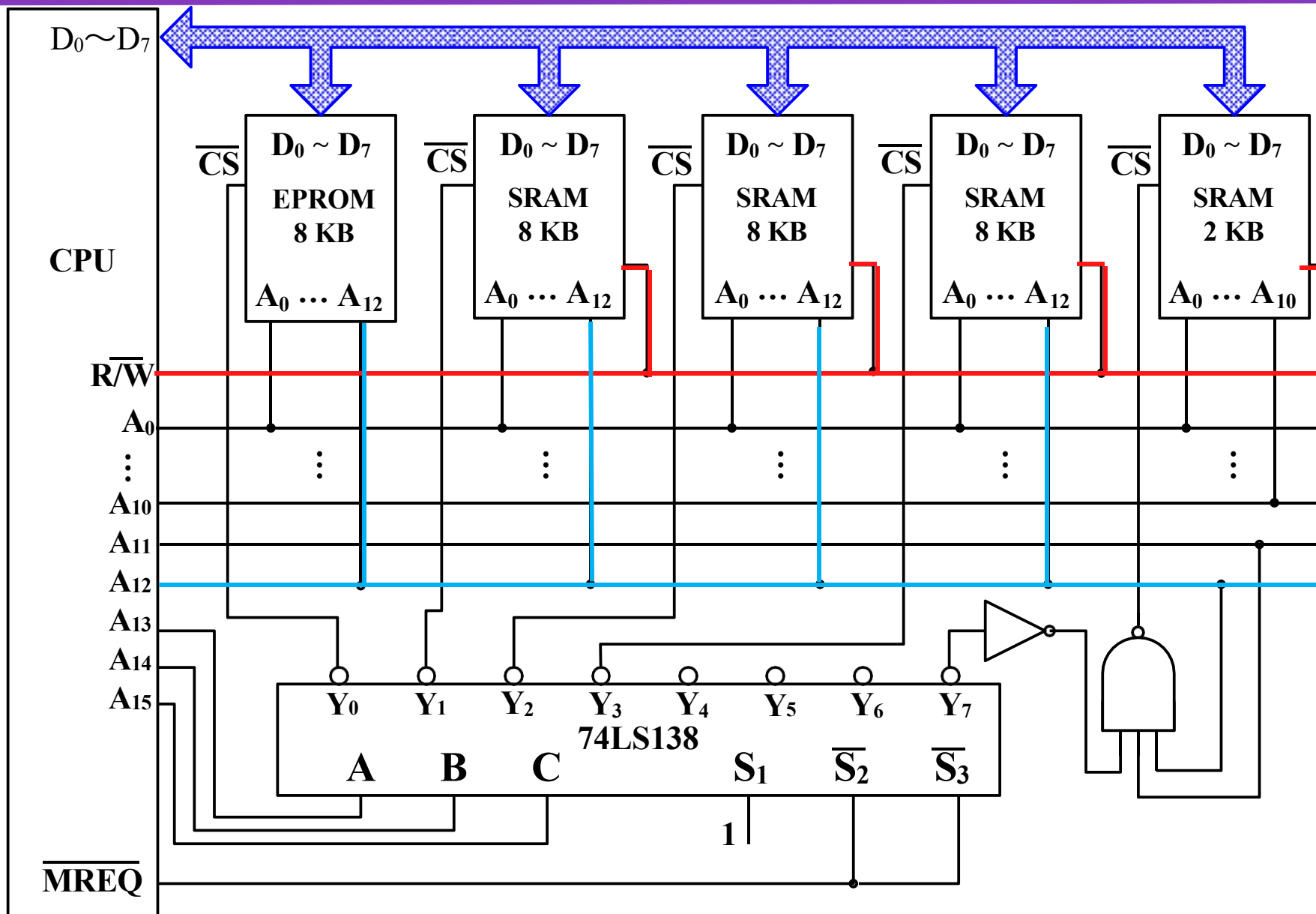
1、与最后2K地址片选有关的信号有哪些？

2、

Y7	A12	A11	CE
1	×	×	1
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0

→ 相关的信号  
都转化为全  
0或者全1







**例4：某 8 位机器中，地址总线12根，数据总线 8 根（D7~D0）控制总线中与主存有关的有（访存允许，低电平有效；读写控制信号，高电平为读命令，低电平为写命令）。**

**选片译码电路使用2:4译码器，主存的地址空间分布如下：**

- 1) 用1K×8位的ROM芯片构成一个地址空间为2KB的系统程序区；起始地址为000H。**
- 2) 用2K×4位的RAM芯片构成2KB的用户程序区，与ROM地址空间相连**

**要求：**

- 1) 计算所需的芯片数目；**
- 2) 画出地址空间分布图；**
- 3) 画出主存和CPU的连接逻辑图。**



1)系统程序区大小 $2K \times 8$ 位； 用户程序区 $2K \times 8$ 位。

故需要芯片：

$1K \times 8$ 位ROM 2片；  $2K \times 4$ 位RAM芯片2片。

2) 地址空间分布图如图：

000~3FFH	ROM 系统程序区
400~7FFH	
800~FFFH	RAM 用户程序区



**1、需要多少地址线对RAM芯片寻址？**

**用户程序区 $2K \times 8$ 位       $A_0 \sim A_{10}$**

**2、ROM芯片需要多少地址线？**

**$1K \times 8$ 位       $A_0 \sim A_9$**

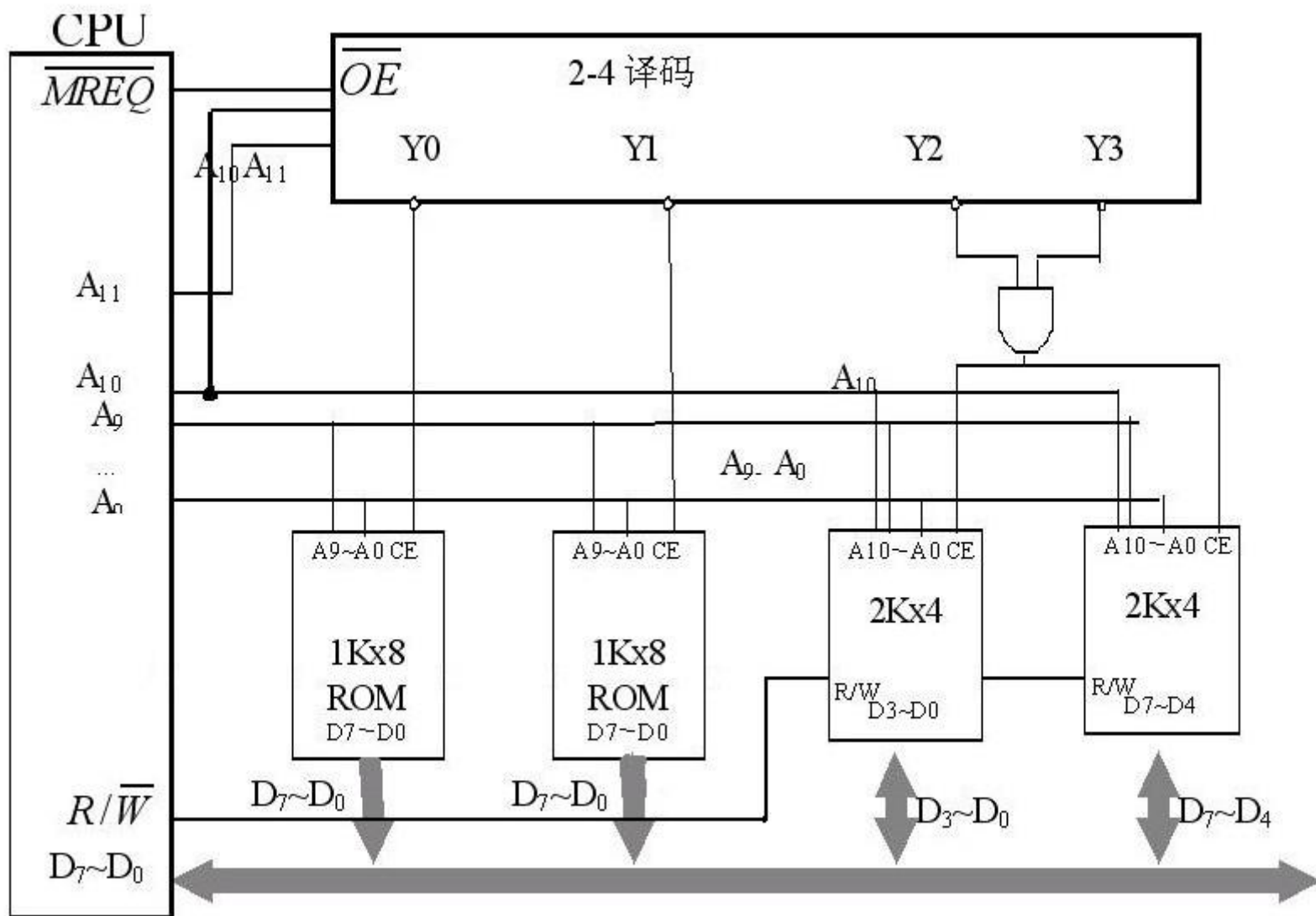
**3、需要几位的地址译码作为片选？  $A_{10}$ 、 $A_{11}$**

**4、地址线 $A_{10}$ 如何处理？**

**5、RAM区域对应着几个译码片选信号？**



- **分析：使用2：4译码器，译码的使能端 $\sim$ OE连接 $\sim$ MEMQ，将A10、A11进行译码；ROM的地址线A0~A9，2片，其CE分别用Y0、Y1连接；**
- **RAM芯片，2片，地址线A0~A10，CE分别用Y2、Y3相与后连接；数据线D0~D7直接和ROM相连，分成两组D0~D3、D4~D7和RAM相连； $\sim$  R/W和RAM的读写控制端相连。**





注意:

- A10既参加译码又作为片内地址寻址

**A11   A10   A9   A8   ... ..   A0**

**1   0   0   0   ... ..   1   801H**

**1   1   0   0   ... ..   1   B01H**

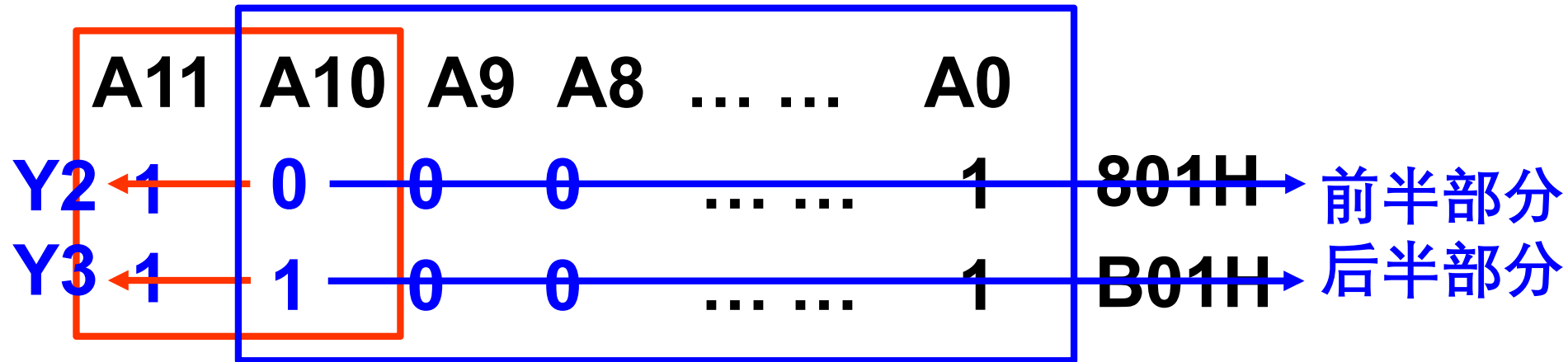
**801H与B01H地址的后10位相同**

- 译码片选Y2 Y3 同时用来片选RAM





- A10既参加译码又作为片内地址寻址



801H与B01H地址的后10位相同

- 译码片选Y2 Y3 同时用来片选RAM

思考：A10的两种作用会不会起“冲突”？

不会！

A10与A11配合，选Y2或Y3(A10=0或1)

A10与A0~A9配合，片内选址，芯片的前半部分 (A10=0) 或后半部分 (A10=1)



## 例5: P115.7

7. 某机器中，已知配有一个地址空间为  $0000H \sim 3FFFH$  的 ROM 区域。现在再用一个 RAM 芯片 ( $8K \times 8$ ) 形成  $40K \times 16$  位的 RAM 区域，起始地为  $6000H$ 。假设 RAM 芯片有  $\overline{CS}$  和  $\overline{WE}$  信号控制端。CPU 的地址总线为  $A_{15} \sim A_0$ ，数据总线为  $D_{15} \sim D_0$ ，控制信号为 R/W (读/写)， $\overline{MREQ}$  (访存)，要求：

- (1) 画出地址译码方案。
- (2) 将 ROM 与 RAM 同 CPU 连接。

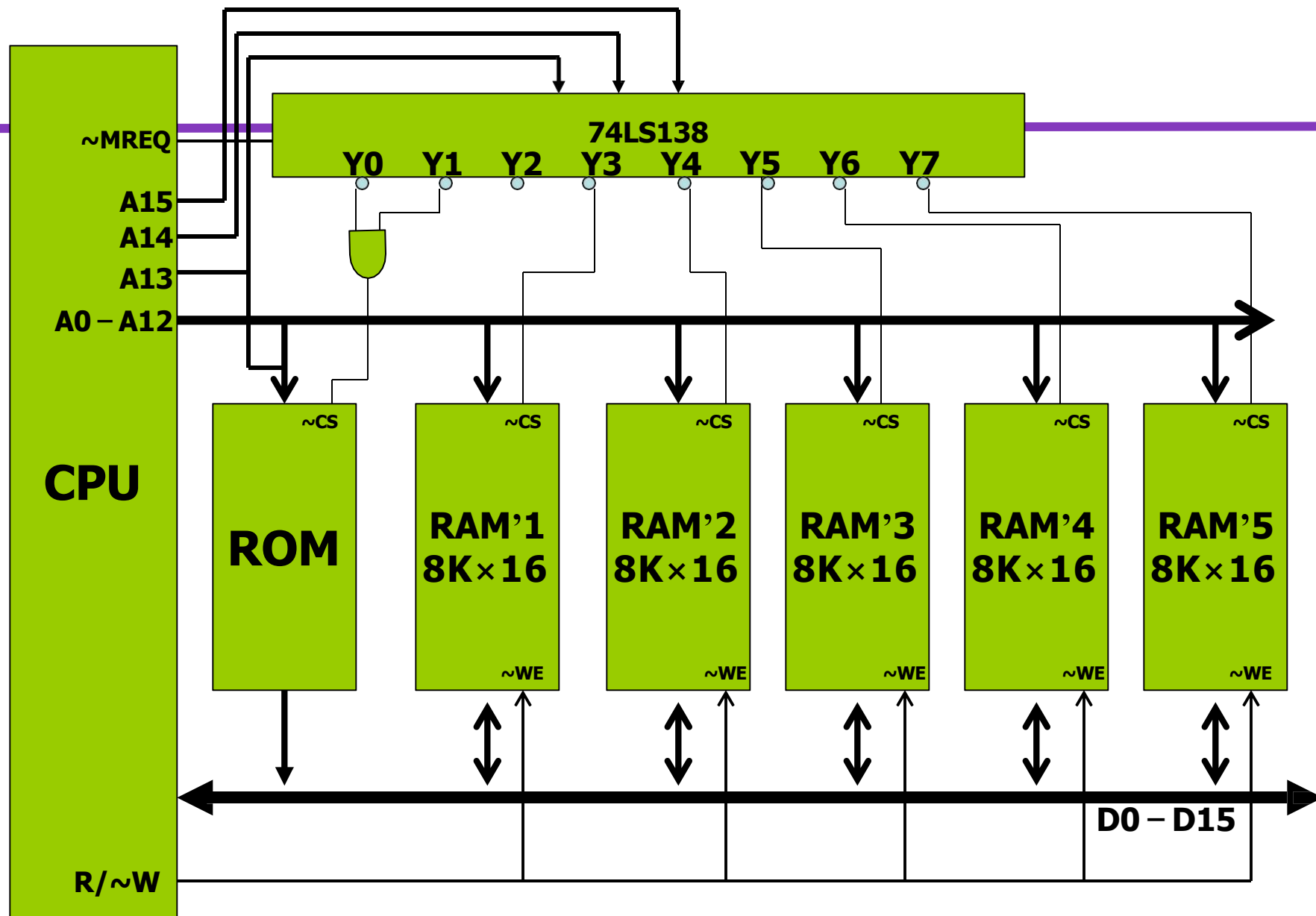


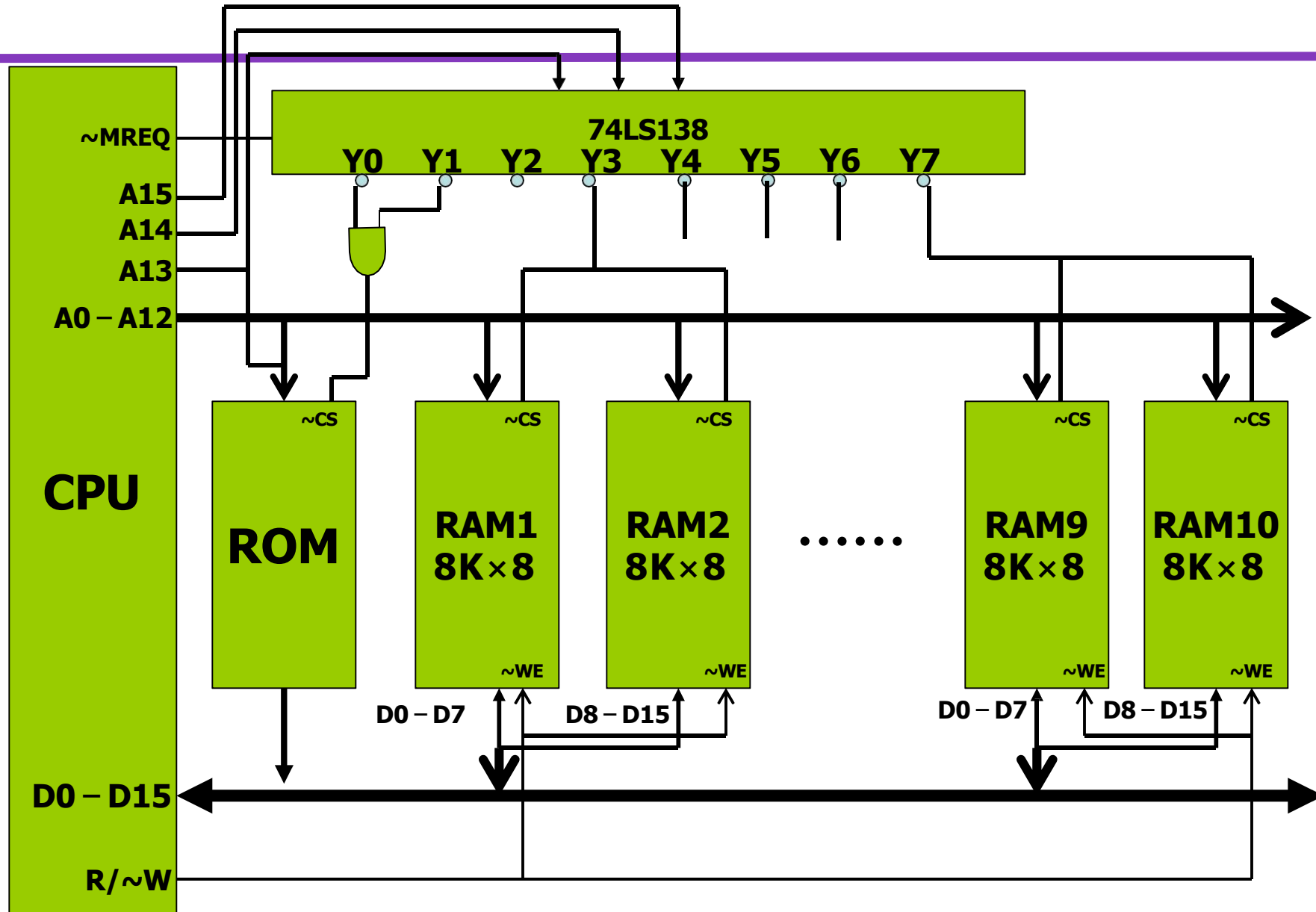
• 解：存储器的地址空间分布如下图所示：

0000H	ROM	16K	
3FFFH			
4000H	空	8K	
5FFFH			
6000H	RAM1'	8K	8K=2000H
	RAM2'	8K	
	RAM3'	8K	
	RAM4'	8K	
	RAM5'	8K	
FFFFH			



- RAM1' ~ RAM5'是8K×16的模块，需要两片的8K×8的 RAM采用位扩展的方法组成。
- 存储器的地址空间可以看做分为8组，每组8K，其中ROM区域看做2组，所以需要采用3：8译码来控制片选信号。（也可以采用将各个模块的起止地址写出，找出其中的规律进行译码。提示：采用地址线的高2位进行译码，就是将地址空间平均分为4组，高3位译码就是将地址空间平均分为8组）
- CPU的R/W信号与RAM的~WE连接，ROM只读不写。
- ROM区域的地址空间有16K， $16K=2^{14}$ ，所以ROM的片内寻址地址需要有14位，**A13既作为片内寻址，也用来译码片选。**
- ROM区域的地址空间有16K，译码后的结果对应着两组，低电平有效，所以采用与门来实现。

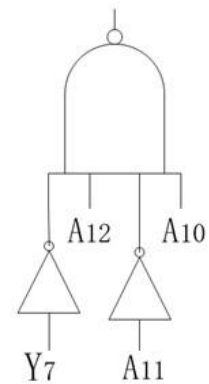
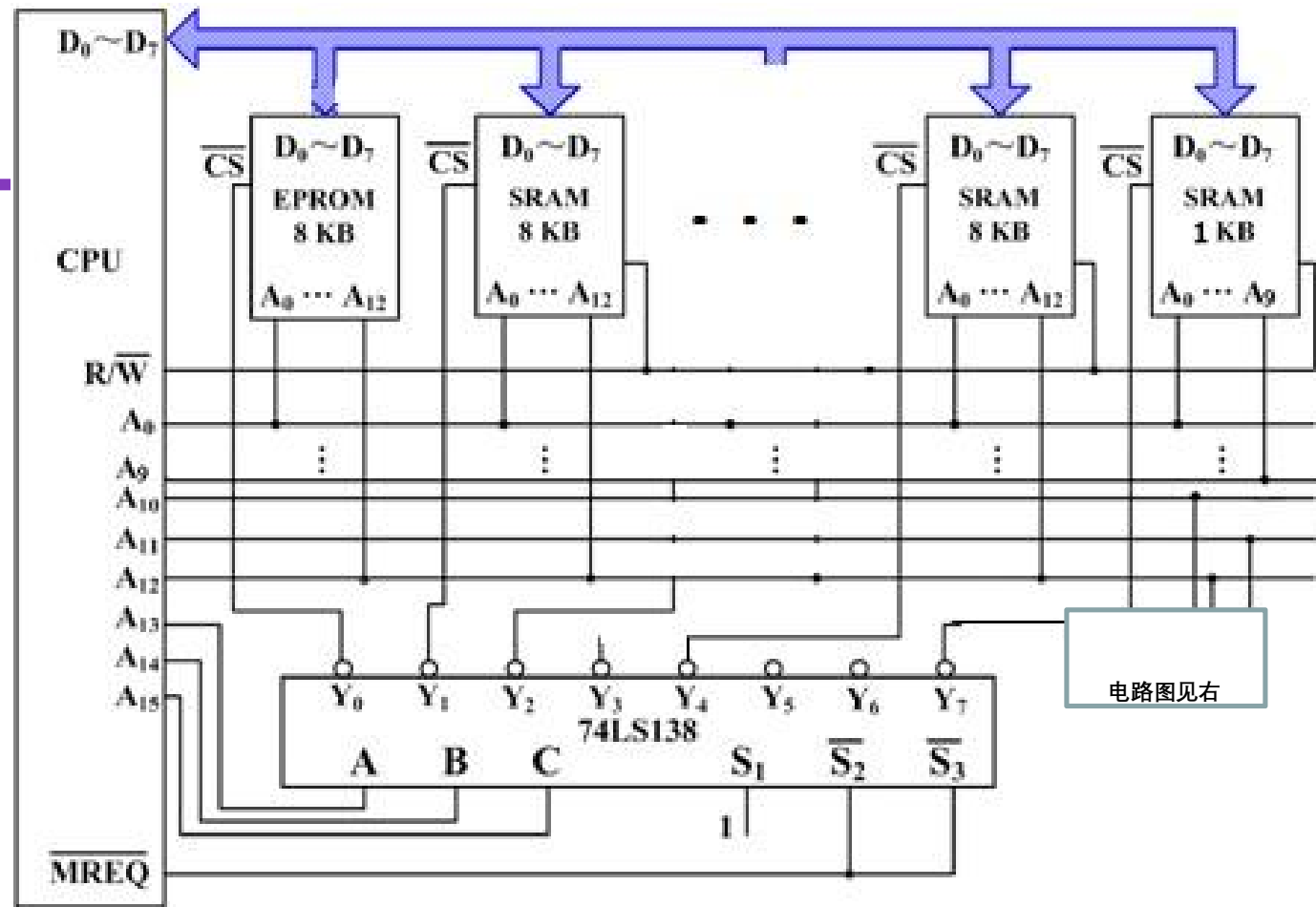




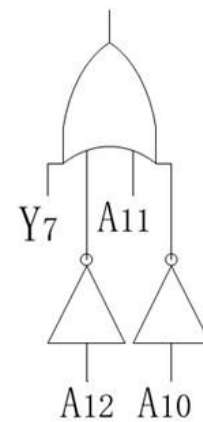


三、某 CPU 地址总线为 A15~A0，数据总线为 D7~D0， $\overline{MREQ}$  为允许访存信号，RAM 芯片有  $\overline{CS}$  和  $\overline{WE}$  信号控制端。存储空间分配如下：系统程序区 8KB，起始地址为 0000H，由  $8K \times 8$  的 ROM 组成；用户程序区共 32KB，起始地址为 2000H，由  $8K \times 8$  的 RAM 芯片组成；系统程序区 1KB，起始地址为 F400H，由  $1K \times 8$  的 RAM 芯片组成。存储器按字节访问，请设计该系统。

- 1) 计算所需各个芯片数。(2 分)
- 2) 画出地址空间分布图；(2 分)
- 3) 画出 RAM、ROM 与 CPU 的连接图以及相关的控制信号。(12 分)



或







- 评分标准：

- 1、基本结构对（有译码、有芯片）
  - 2、有MREQ信号
  - 3、A13、A14、A15作为高位译码信号
  - 4、RAM有WE信号，ROM不能有WE
  - 5、片选的CS有连到3:8译码端Y0-Y4，Y7
  - 6、地址线8K的为A0-A12
  - 7、地址线1K的为A0-A9
  - 8、数据线D0-D7有标
  - 9、ROM数据线单向的
- 以上一点为1分
- 10、最后的门电路图（3分）



二、（14 分）某 CPU 地址总线为  $A_0 \sim A_{15}$ ，数据总线为  $D_0 \sim D_7$ ， $\overline{MREQ}$  为允许访存， $R/\overline{W}$  为读写命令。地址空间分配如下： $0 \sim 8\text{ K}$  为系统区，由  $8\text{ K} \times 8$  的 ROM 组成； $8\text{ K} \sim 32\text{ K}$  为用户区，由  $8\text{ K} \times 8$  的 RAM 芯片组成；程序区有  $2\text{ K}$ ，由  $2\text{ K} \times 4$  的 RAM 芯片组成，程序区的地址空间与用户区的地址空间相连。

1. 计算所需的芯片。
2. 画出地址空间分布图。
3. 请画出 CPU 与存储器系统的连线，并注明相关的控制信号。

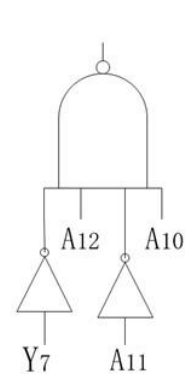
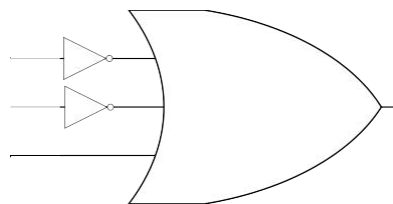
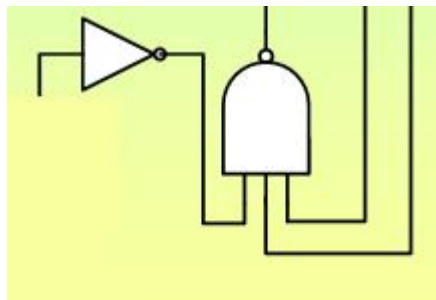
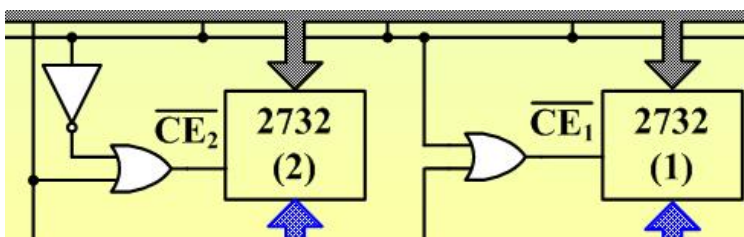




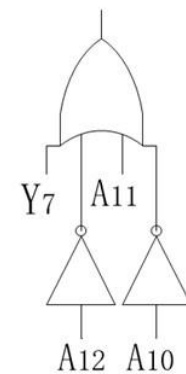
- 有CPU、芯片、译码器（1分）
- A13-A15接138译码器，Y0-Y3外接片选（1分）
- 最后2K位扩展，标明D0-D3，D4-D7（1分）
- 8K芯片有标明D0-D7（1分）
- R/W信号有接芯片，ROM不能接R/W（1分）
- 8K芯片的地址线为A0-A12（1分）
- 2K芯片地址线为A0-A10（1分）
- MREQ有接（1分）
- 最后2K的片选接对（1分）



# 思考



或

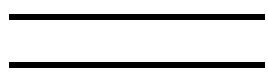
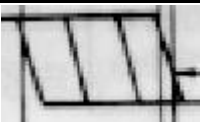





- 不用门电路，用相对应的2:4 (3:8) 译码器是否可以实现所需功能？

时序图提供的信息：

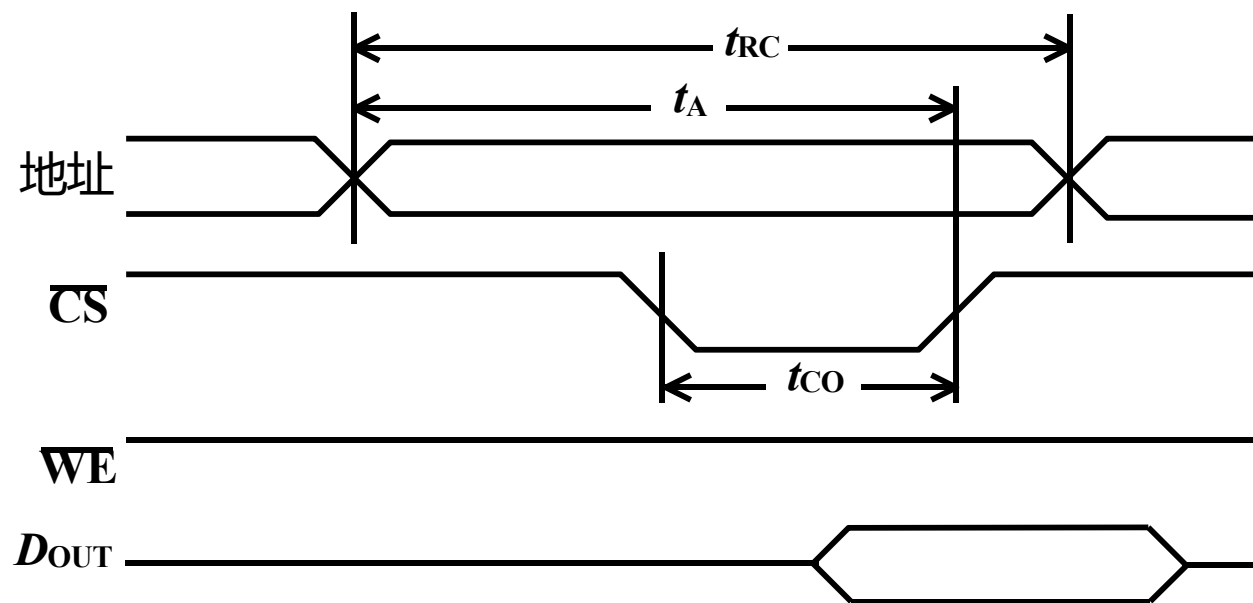
- 各输入信号的有效情况
- 各输入信号的极限时间参数（如最小脉冲宽度）
- 各输入信号之间应遵循的极限时间配合关系（如建立时间）
- 各输出的最大传输延迟

# 常用的时序符号

符号	输入	输出
	稳定的 'H'或 'L'态	稳定的 'H'或 'L'态
	由 'H'向 'L'变化，斜线为过渡区	
	允许由 'L'向 'H'变化或由 'H'向 'L'变化	不知状态如何变化
		中线为高阻态
	中间为信息有效，其它的为信息无效	

# >>> 存储器的读、写周期

**读周期：**从给出有效地址，经过译码电路、驱动电路延迟，到读出选中单元内容，再经过I/O电路延迟后在外部数据总线上稳定地出现所读出的数据信息，这一过程所需时间为 $t_A$ ，称为读出时间。读周期 $t_{RC}$ 表示存储器进行两次连续读操作所必须间隔的时间，它大于等于读出时间。片选信号 $\overline{CS}$ 必须保持到数据稳定输出， $t_{CO}$ 为片选保持时间。读周期中 $\overline{WE}$ 为高电平。

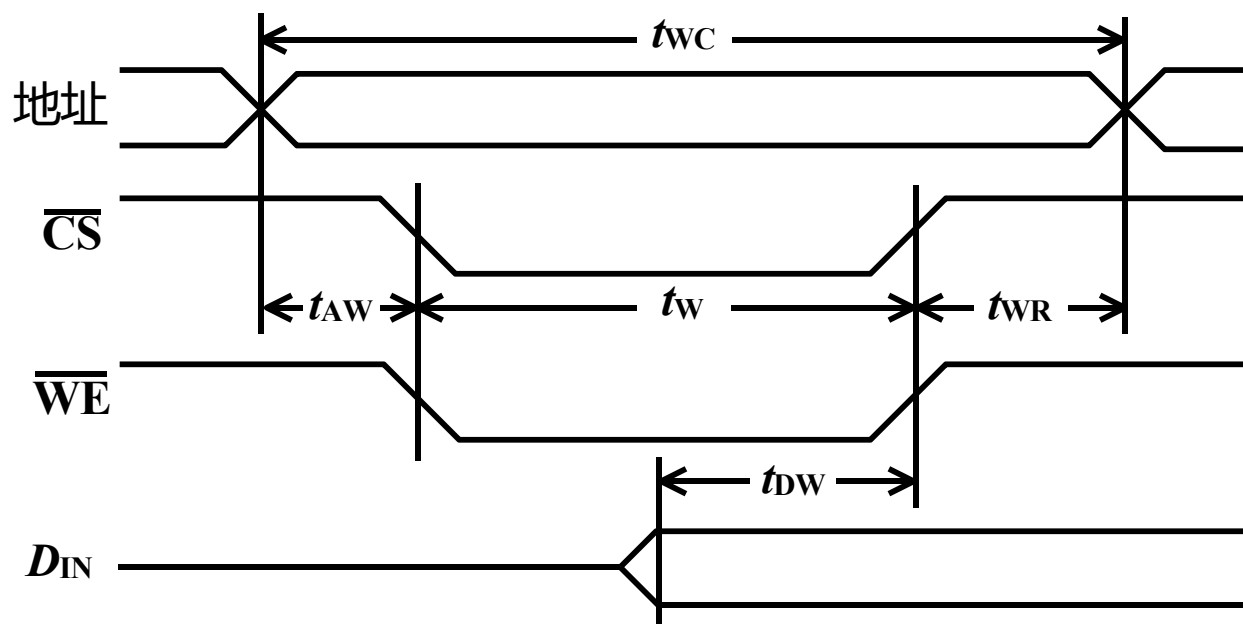


- $t_{RC}$  - 读周期
- $t_A$  - 读出时间
- $t_{CO}$  - 片选保持时间



# 存储器的读、写周期

**写周期：**要实现写操作，要求 $\overline{CS}$ 和 $\overline{WE}$ 都为低电平。为使数据总线上的数据可靠写入存储器，要求 $\overline{CS}$ 和 $\overline{WE}$ 同时有效的宽度至少为 $t_{WC}$ 。为了使地址变化期间不会把信息写入错误地址， $\overline{WE}$ 在地址变化期间必须为高电平。为了保证 $\overline{WE}$ 、 $\overline{CS}$ 无效前把数据可靠地写入，数据必须在 $t_{DW}$ 前稳定地出现在数据总线上。



- $t_{WC}$  - 写周期
- $t_{W}$  - 写数时间
- $t_{WR}$  - 写恢复时间
- $t_{DW}$  - 数据有效时间

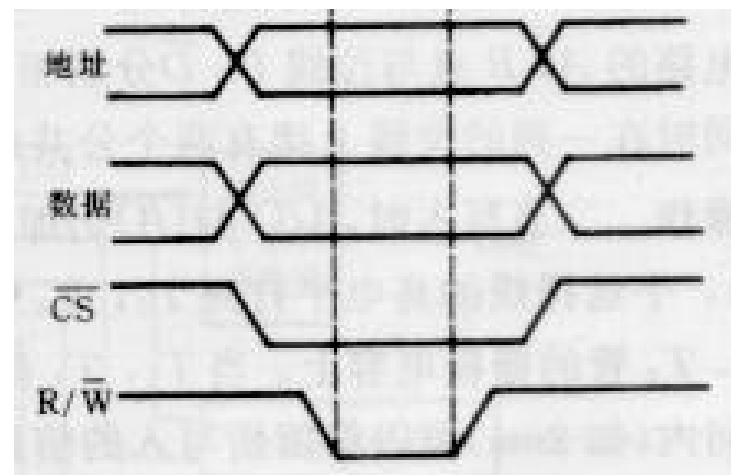
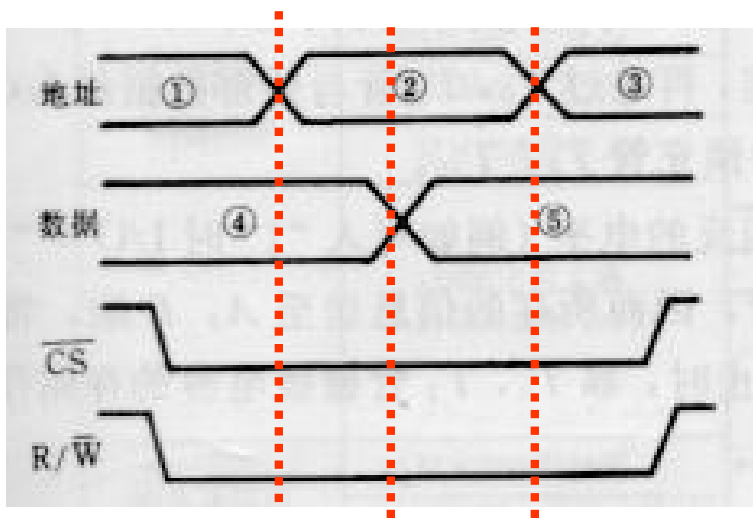
# >>> 存储器的读、写周期

如图为SRAM的写入时序图，请指出图中错误地方，并画出正确的时序图

**读写信号加负脉冲时，地址线和数据线应稳定**

**错误1：读写信号加低电平时，数据线改变，会存入新的数据**

**错误2：读写信号加低电平时，地址线改变，会存入新的地址**





谢谢