



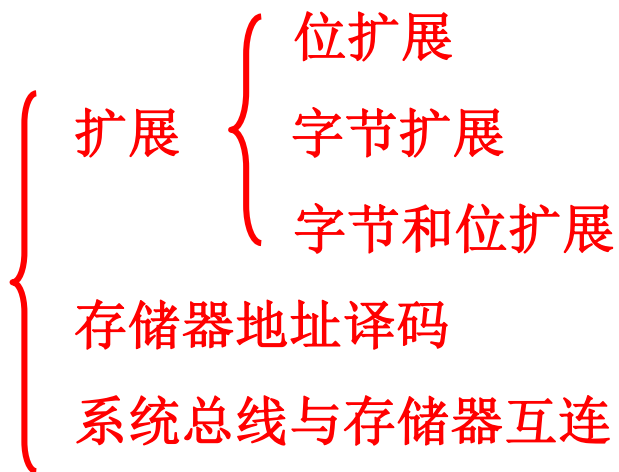
扩展存储器设计

问题

- ❖ 单个存储芯片容量有限（位数或字节数），如何满足系统所需的更高位数或字节数的要求？
- ❖ 如何实现系统总线与存储器互连？

主要内容

扩展存储器设计





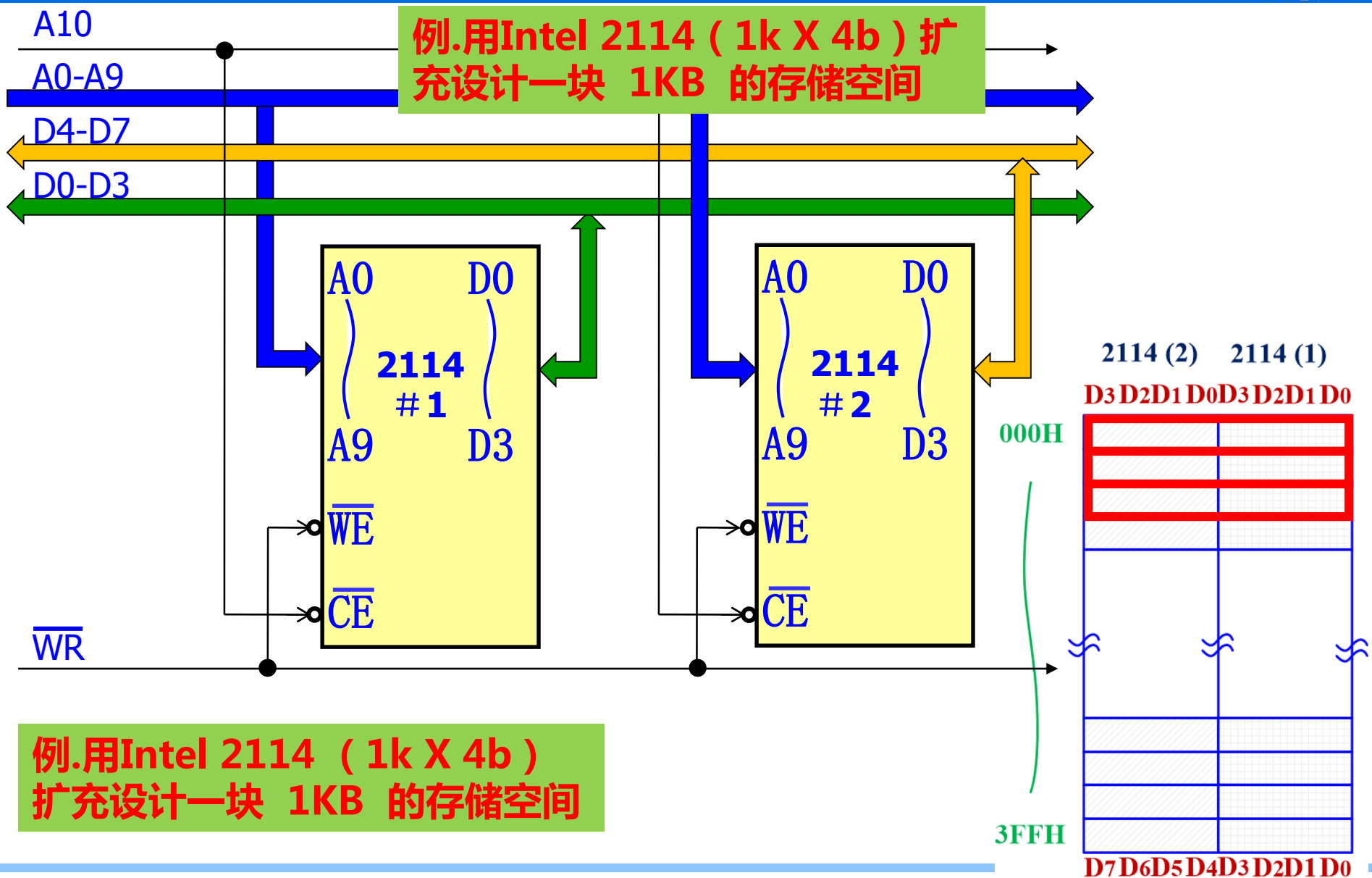
扩展存储器设计

1. 位扩展

- ❖ 针对数据宽度；
- ❖ 微机系统存储器按字节构成，存储器芯片字长不足8位时，必须进行位扩展；



扩展存储器设计





扩展存储器设计

位扩展总结

- 用几片存储器芯片并起来，增加存储字长；
- 每个芯片的数据线分别引出；
- 每个芯片的地址线并联；
- 每个芯片的控制线并联；
- 每个芯片的片选线并联。



扩展存储器设计

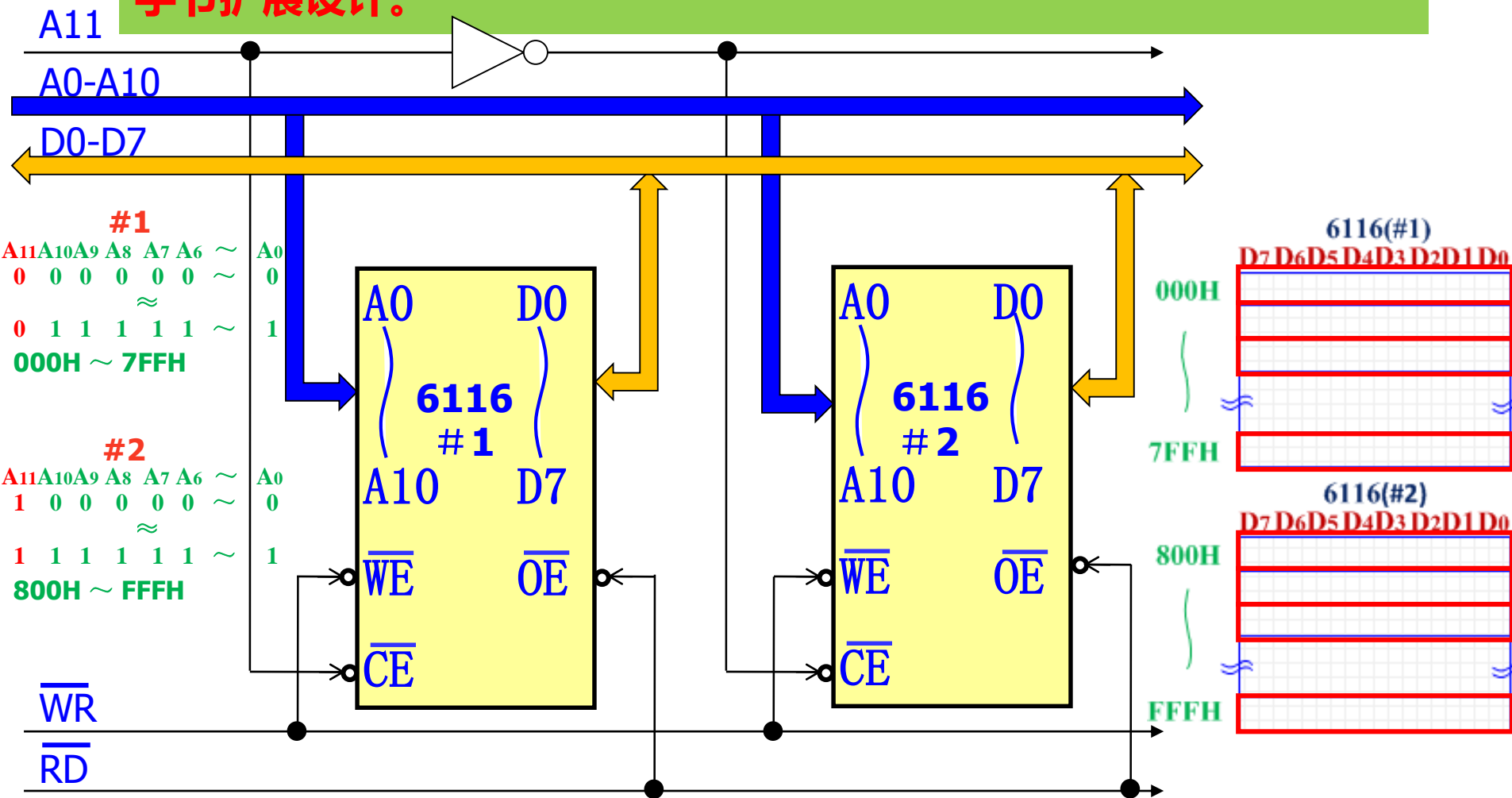
2 . 字节扩展（地址空间扩展）

- 芯片字长不变的情况下增加存储器字节的数量;
- 存储芯片的字节数不够，需用若干芯片组成总容量较大的存储器



扩展存储器设计

例. 用2片2k×8b的SRAM芯片6116，组成4k×8b的存储器，实现字节扩展设计。



字节扩展总结

- 各芯片地址线并联；
- 各芯片数据线并联；
- 各芯片控制线并联；
- 片选信号分别引出，每个芯片具有不同的地址范围。



扩展存储器设计

3 . 字节和位扩展

存储芯片的位数和字节数都不满足要求，需要同时进行字节和位的扩展（**字节扩展和位扩展的组合**）。

方法：

- **先进行位扩展**以满足字长（位数）要求（芯片组）；
- **再进行字节扩展**以满足字节数要求(可利用芯片组)；
- **根据所需容量及芯片容量确定所需存储芯片数：**

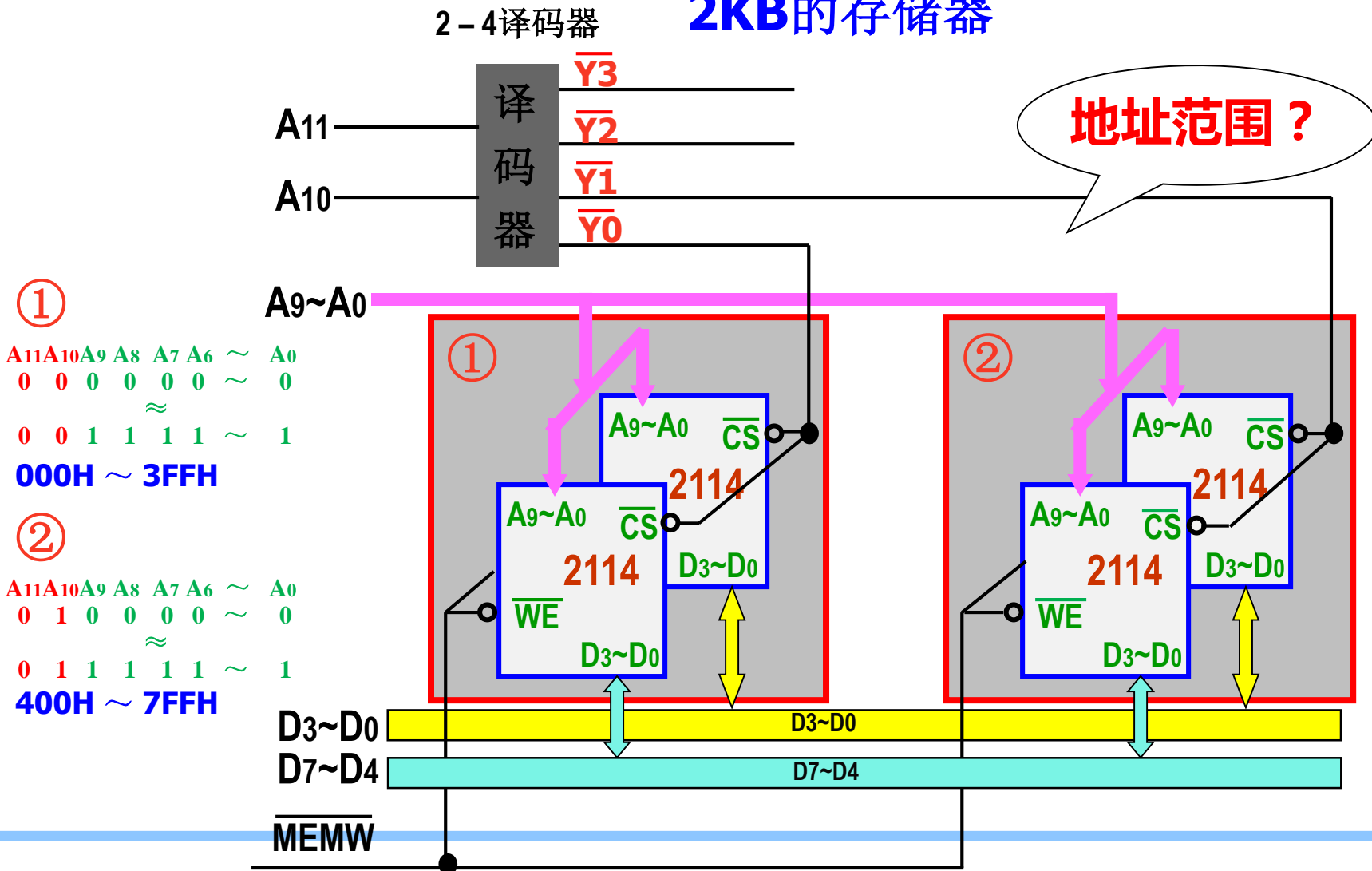
芯片的容量为 m (字节数) $\times n$ (位数)，要构成容量为 $M \times N$ 的存储器，需要的芯片数为 $= (M/m) \times (N/n)$



扩展存储器设计

3. 字节和位扩展

例. 用**2114**(**1k×4b**) 芯片组成**2KB**的存储器





扩展存储器设计

4 . 存储器地址译码

译码方法（三种）

所有低位地址接存储器地址线

(1) 全地址译码方式： 所有剩余高位地址参加译码

(2) 部分地址译码方式： 部分高位地址参加译码

(3) 线选译码方式： 高位地址一根选择一个存储芯片

译码电路

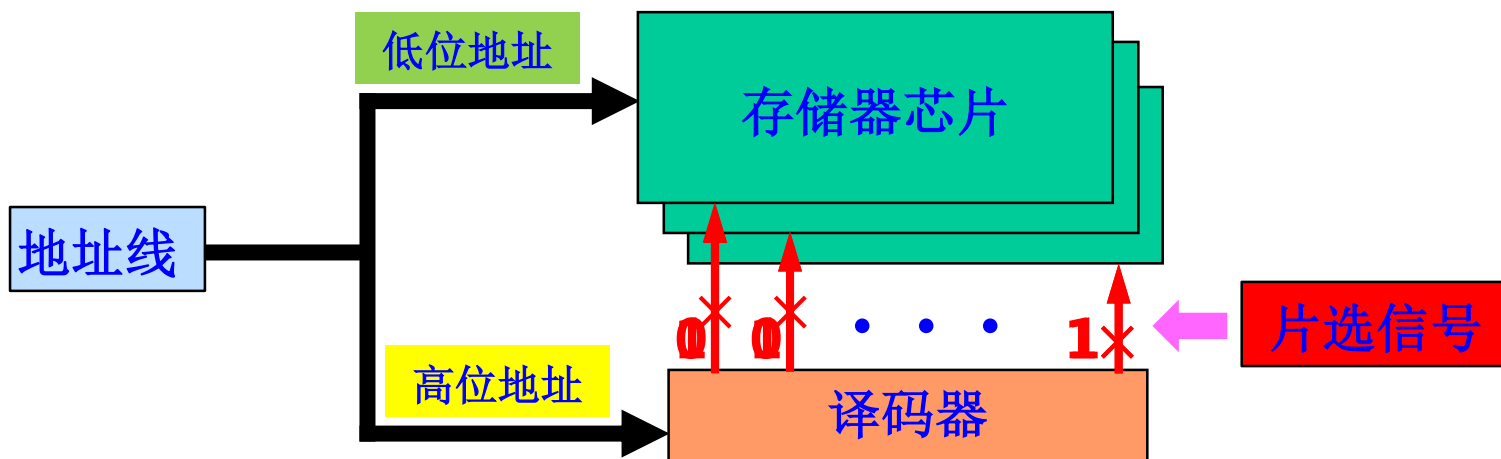
存储器的地址译码电路形式很多，主要关注：

- 组合电路（门电路）形式；
- 专用译码器形式，如3-8译码器74LS138；

扩展存储器设计

全地址译码方式

- ❖ 存储芯片地址和CPU低位地址——连接；
- ❖ 剩余所有高位地址信号全部作为译码输入信号；
- ❖ 译码后连接不同芯片的片选，仅有一个（组）存储芯片被选中；
- ❖ 每个（组）存储芯片的地址不重叠，存储芯片所占地址连续；





扩展存储器设计

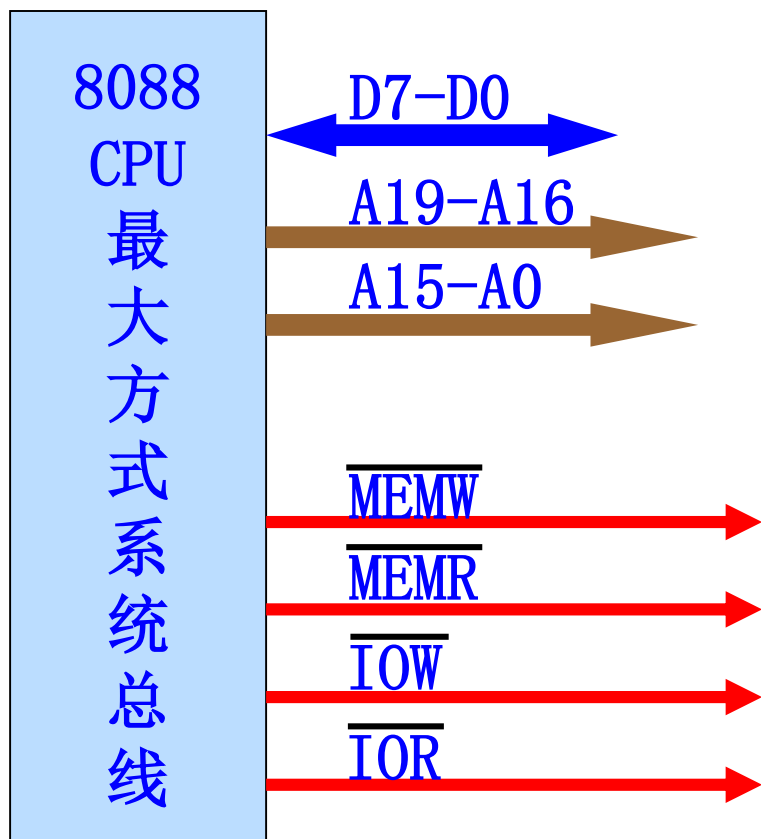
4 . 系统总线与存储器互连

例. 8088 CPU工作在最大方式组成的微机应用系统中，扩充设计 8kB 的 SRAM 电路，SRAM 芯片用 Intel 6264。若分配给该SRAM的起始地址为62000H，片选信号（ \overline{CS} ）为低电平有效。请用全地址译码方法设计该SRAM存储器的片选信号形成电路，并实现与系统总线的连接。



扩展存储器设计

①确定总线及总线信号



②确定存储器芯片数

因为Intel 6264芯片容量为 $8k \times 8b(8kB)$ ，因此只需要1片Intel 6264存储器芯片，既不需要位扩展，也不需要字节扩展。

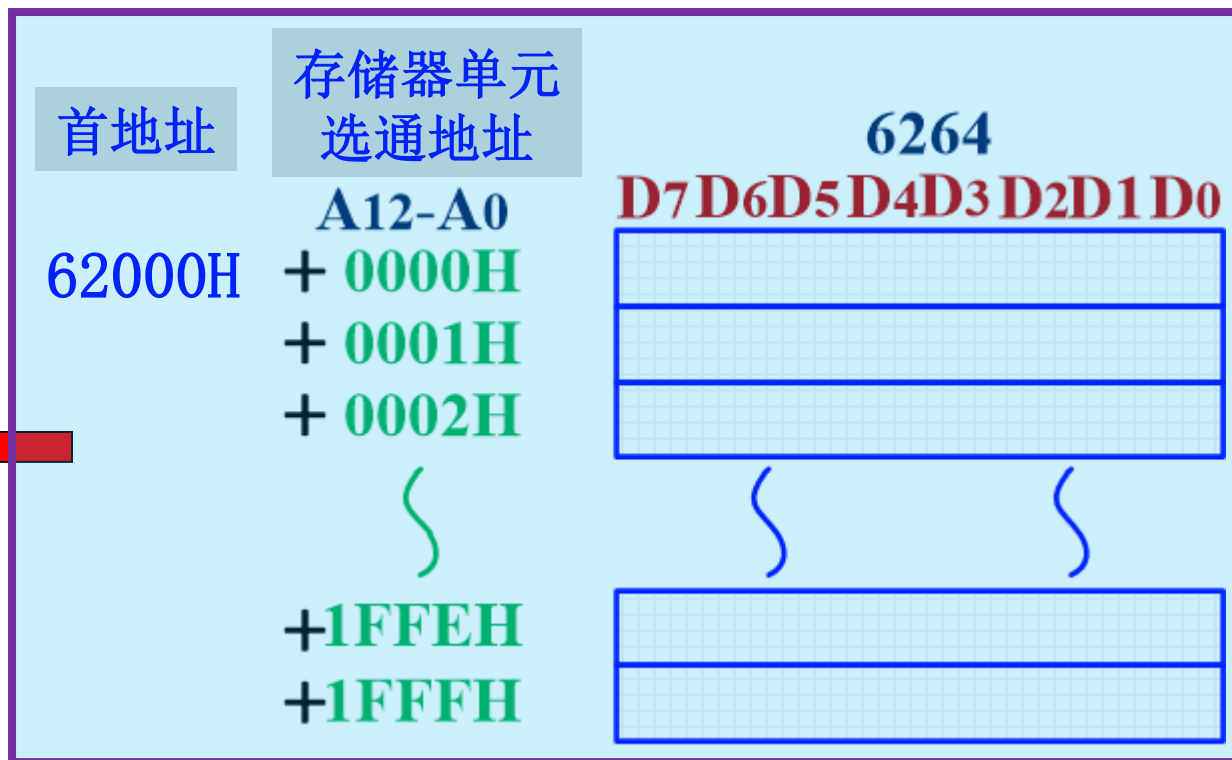


扩展存储器设计

③地址分析

存储器地址范围:
62000H ~ 63FFFH

片外寻址



A_{19} ~ A_{13}

0 1 1 0 0 0 1

0 1 1 0 0 0 1

A_{12}

~

A_0

0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1

与6264
芯片的
 $A_{12} \sim A_0$
相连,
作片内
寻址

高位不变地址部分, 去参加译码, 作6264的片选信号。



扩展存储器设计

④电路连接

❖ 地址线：

8088CPU 系统地址总线的低 13 位 A12 ~ A0 直接与 Intel6264的片内地址引脚A12 ~ A0相连接，作片内寻址，来选择片内具体的存储单元；

❖ 数据线：

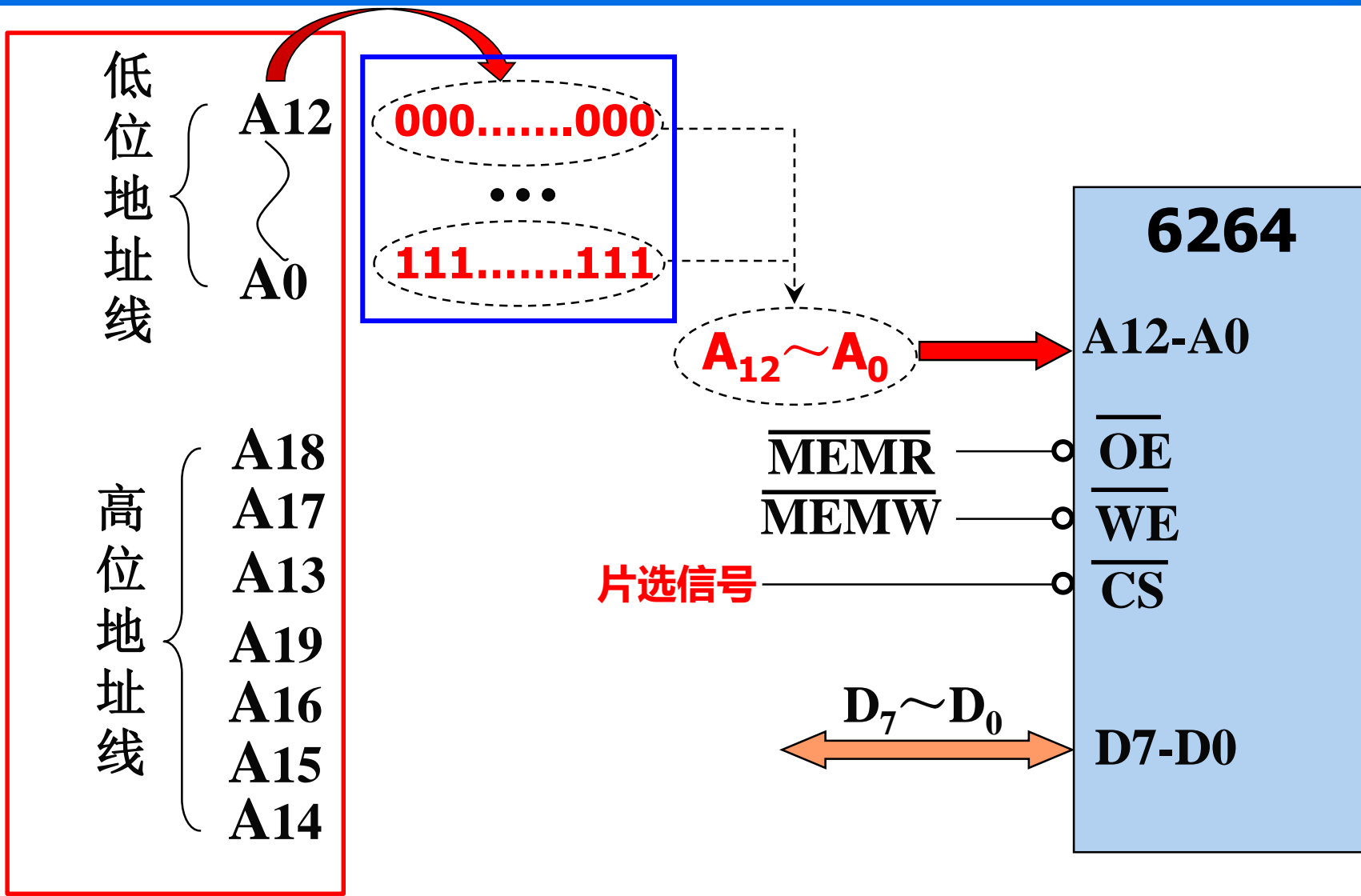
数据总线D7 ~ D0与存储器数据管脚D7 ~ D0相连；

❖ 控制线：

$\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$ 与存储器 $\overline{\text{OE}}$, $\overline{\text{WE}}$ 相连。



扩展存储器设计



⑤存储器地址译码电路

- 全地址译码：

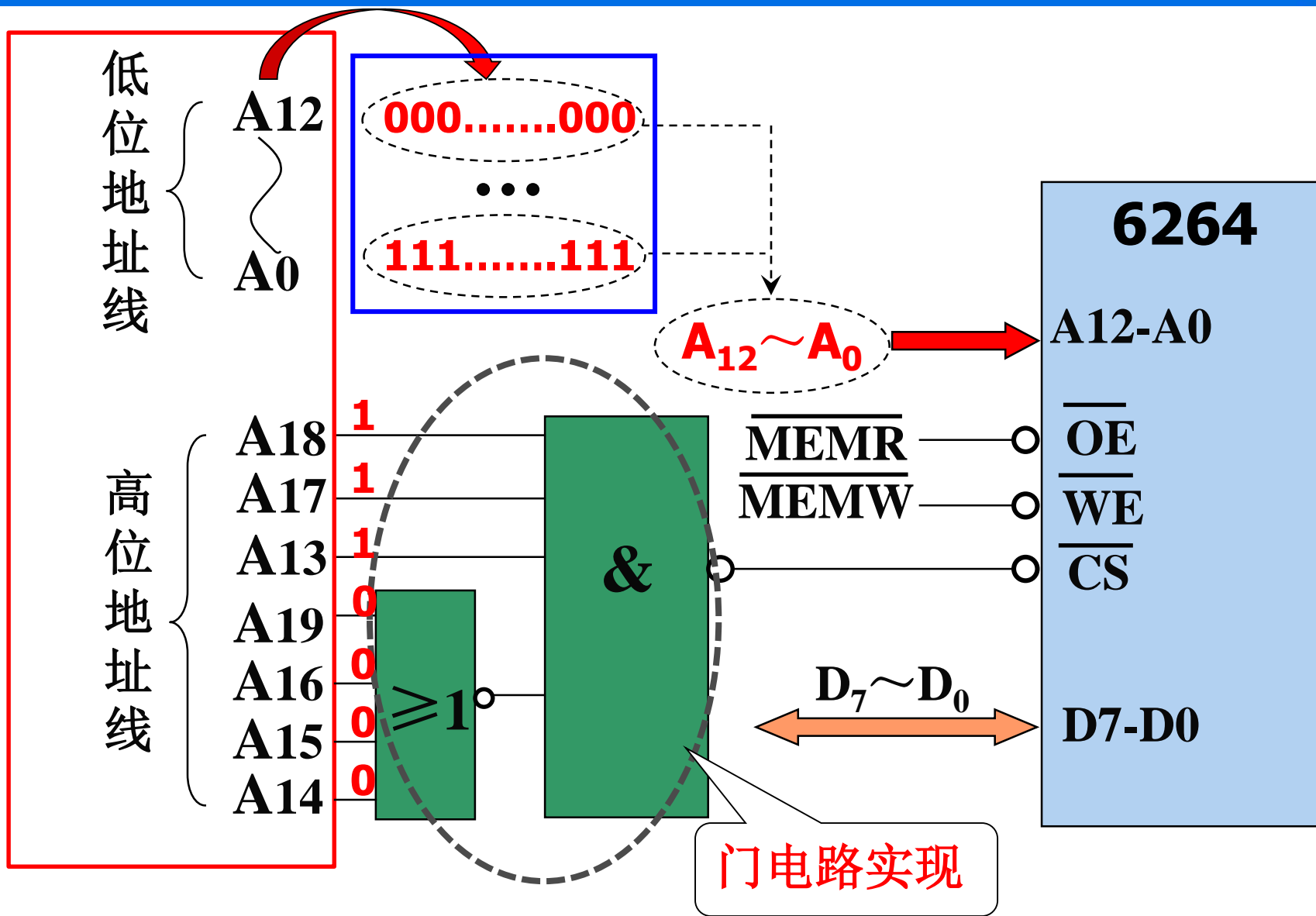
8088CPU系统地址总线的高7位A19 ~ A13全部参加译码；

- 译码输出作为存储器芯片的片选信号

对应的存储器地址范围为62000H ~ 63FFFH连续的8kB存储区域。



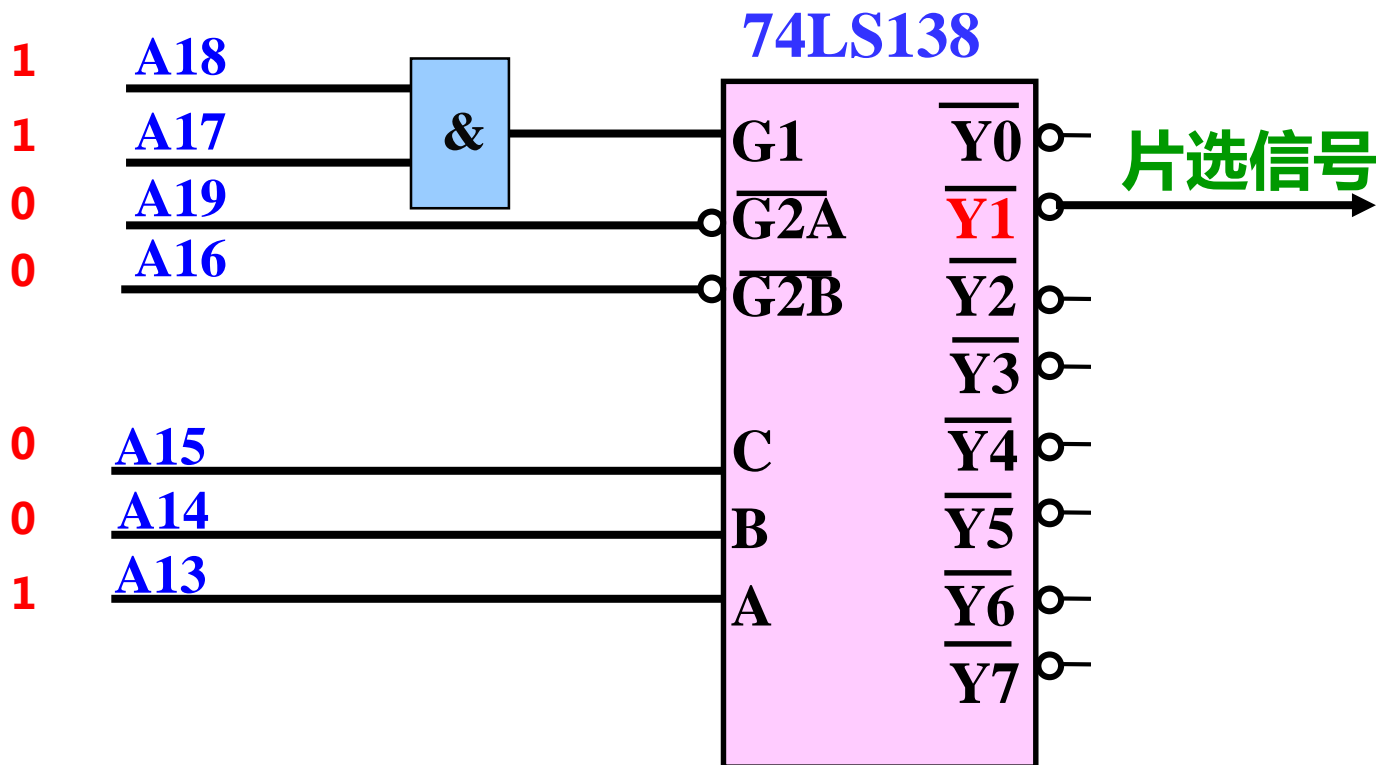
扩展存储器设计





扩展存储器设计

译码电路（用74LS138专用译码器译码）



例. 在8088系统总线上（最小方式）扩充设计8K字节的SRAM存储器电路。SRAM芯片选用Intel6264，起始地址从04000H开始，译码电路采用74LS138。

- (1) 计算此RAM存储区的最高地址为多少？**
- (2) 画出此存储器电路与系统总线的连接图；**
- (3) 编写程序实现对此存储器区域进行自检。**



扩展存储器设计

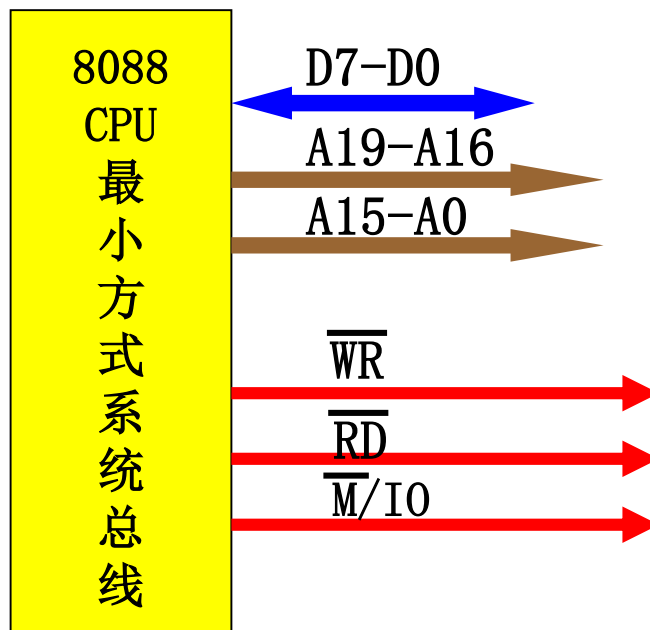
(1) 计算此RAM存储区的最高地址为多少？

因为Intel 6264存储容量为 $8K \times 8$ （字节），所以设计此存储电路共需1片6264芯片。因此最高地址为：

$$04000H + 02000H - 1 = 05FFFH$$

(2) 画出此存储器电路与系统总线的连接图

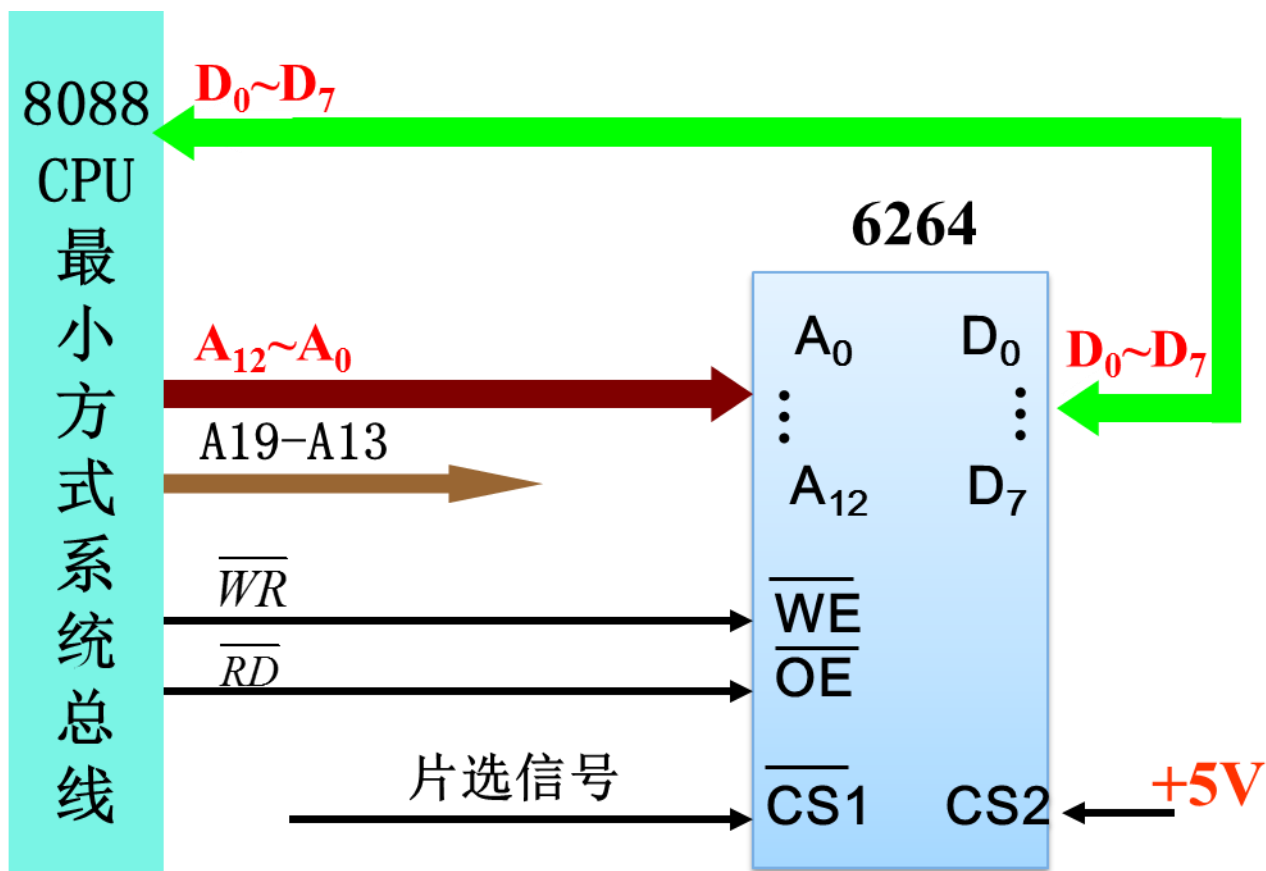
确定总线及总线信号





扩展存储器设计

电路连接



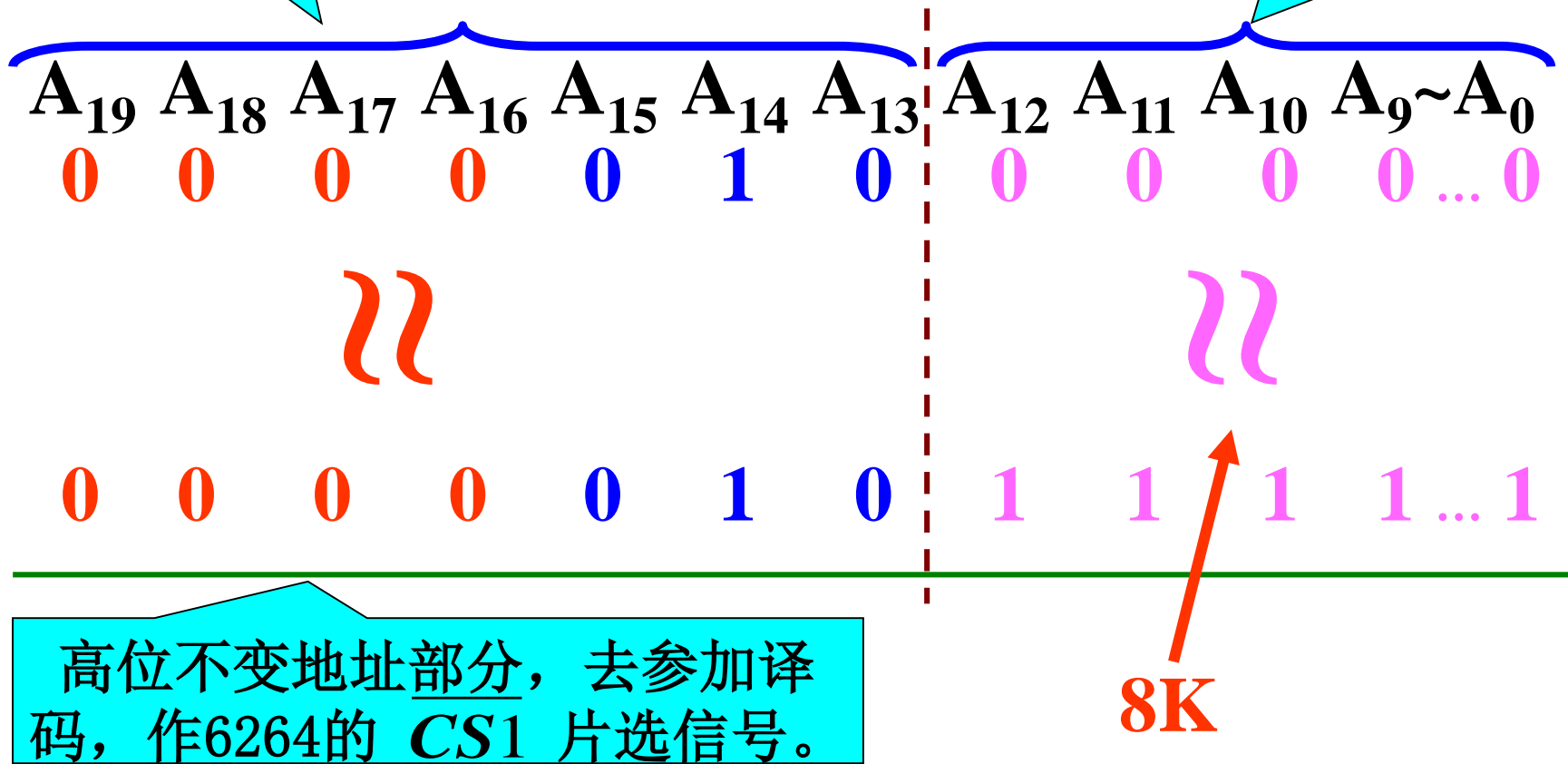


扩展存储器设计

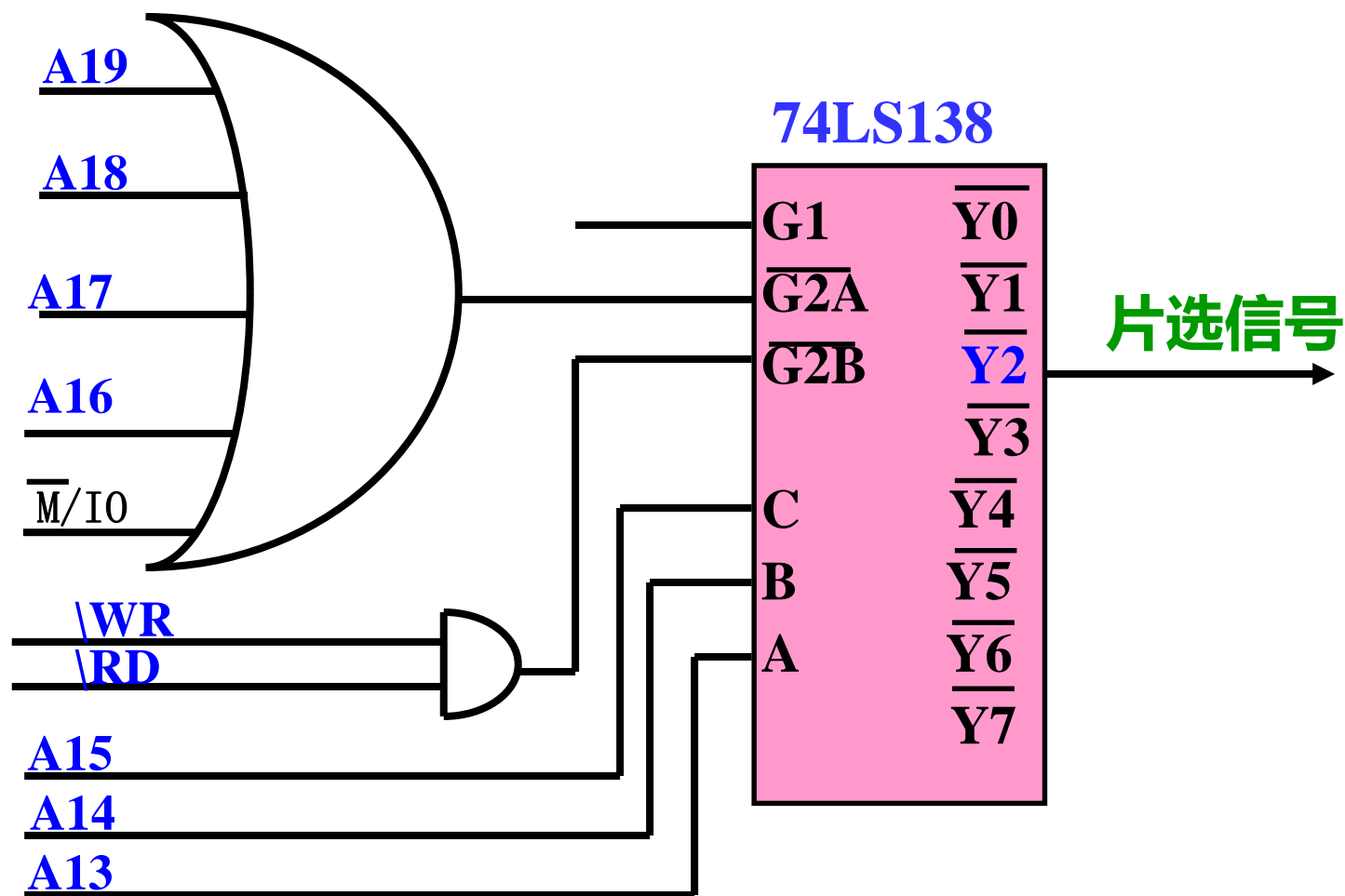
地址分析

片外寻址

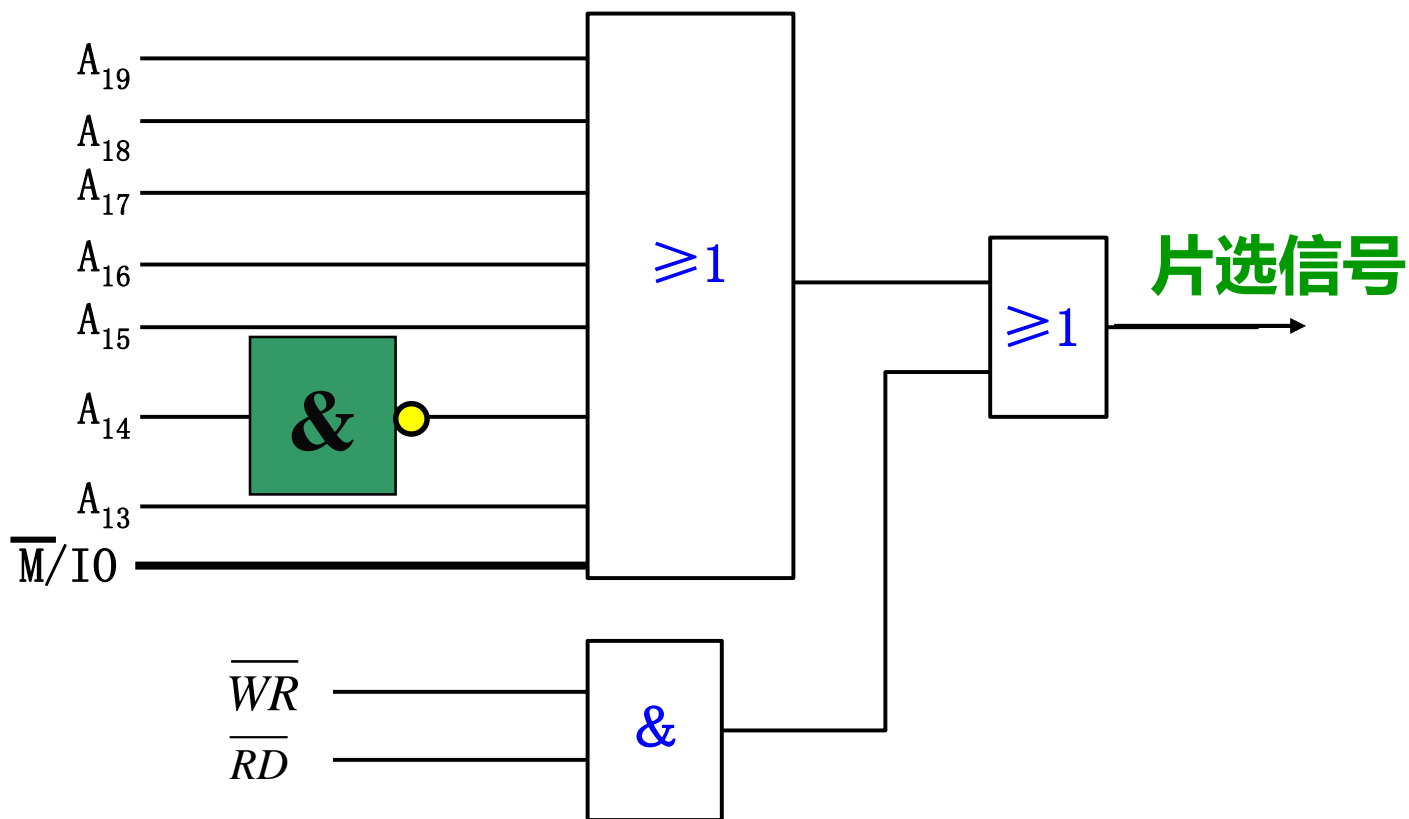
与6264芯片的 $A_{12} \sim A_0$
相连，作片内寻址



译码电路设计方法1（用74LS138专用译码器译码）



译码电路设计方法2（用门电路译码）



(3) 编写程序实现对此存储器区域进行自检

```
MOV AX,0400H
MOV DS,AX
MOV BX,0000H
MOV CX,8*1024
MOV AL,55H
NEXT1:
MOV [BX],AL
CMP [BX],AL
JNZ ERROR
INC BX
LOOP NEXT1
...
ERROR:
...
```

(注：保护现场)



扩展存储器设计

例. 在8088最大方式系统总线上扩充设计8K字节的ROM存储器电路。ROM芯片选用Intel 2764，起始地址从FE000H开始，译码电路采用74LS138。

- 1.计算此ROM存储区的最高地址为多少？
- 2.画出此存储器电路与系统总线的连接图。



扩展存储器设计

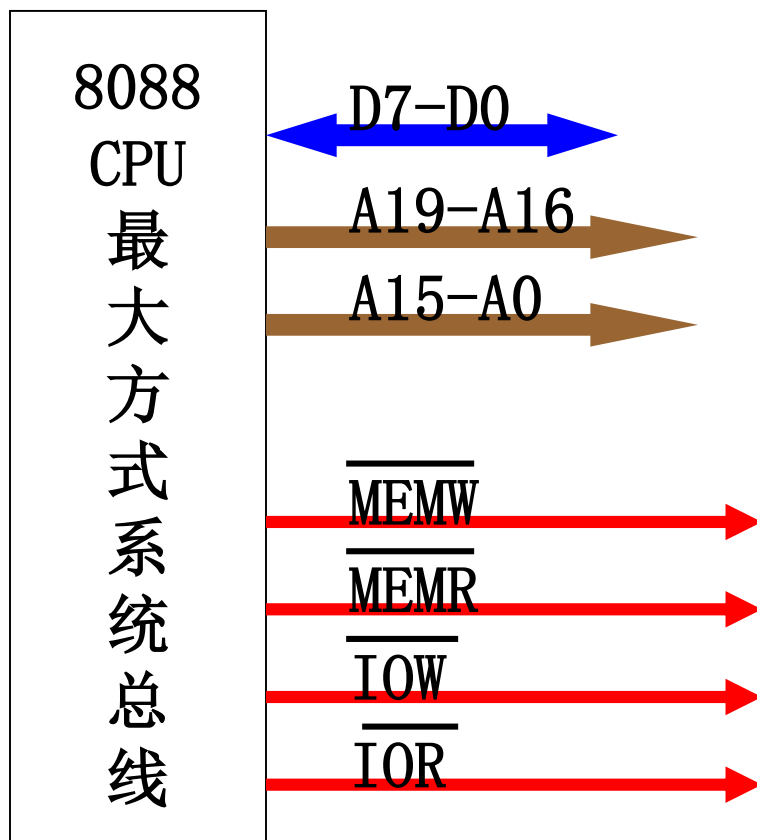
① 求存储区的最高地址

因为Intel 2764存储容量为8K × 8（字节），
所以设计此存储电路共需1片2764芯片。因此最高
地址为：

$$\text{FE000H} + 02000\text{H} - 1 = \text{FFFFFFH}$$

② 电路设计

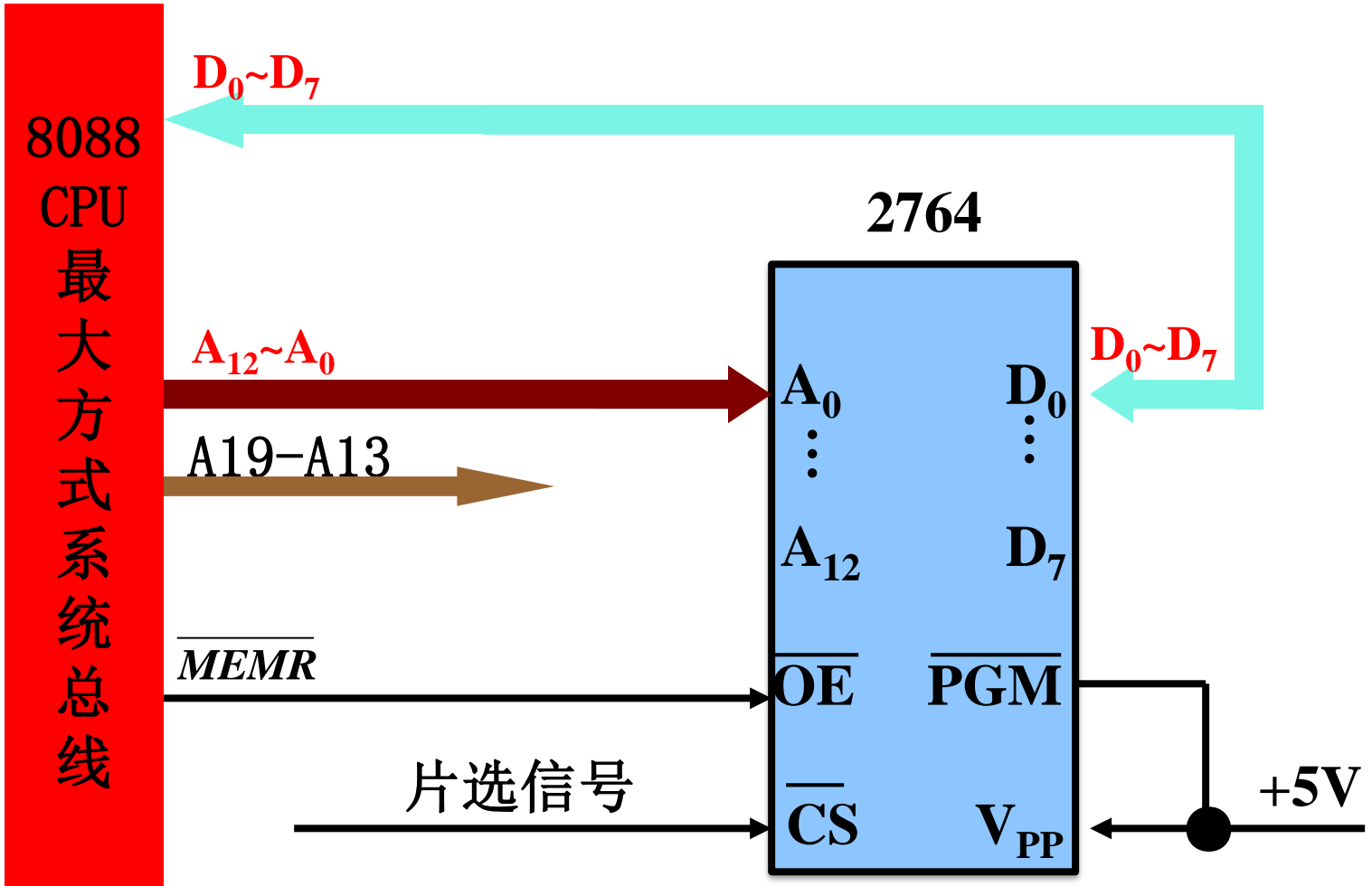
确定总线及总线信号





扩展存储器设计

电路 连接



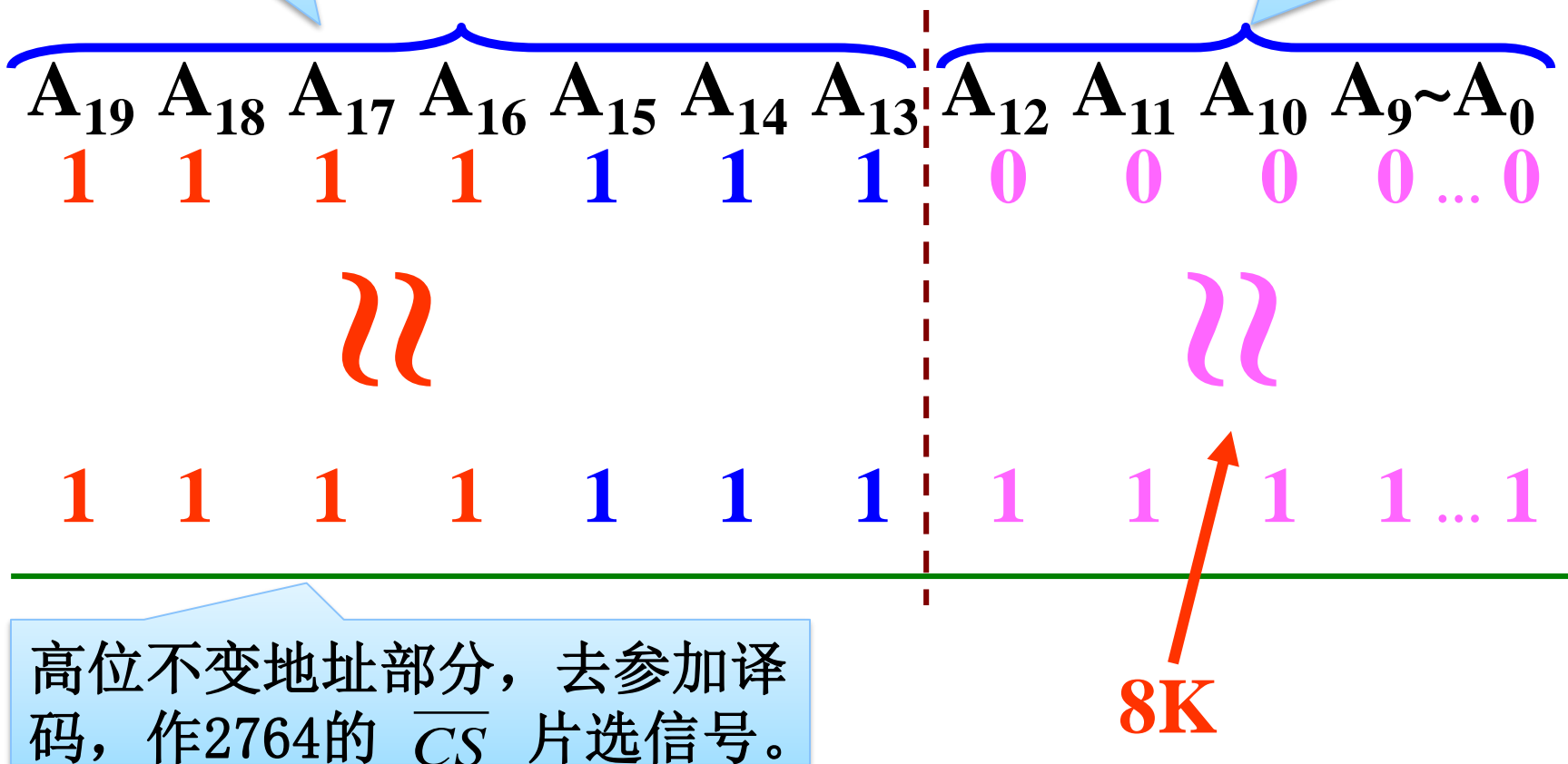


扩展存储器设计

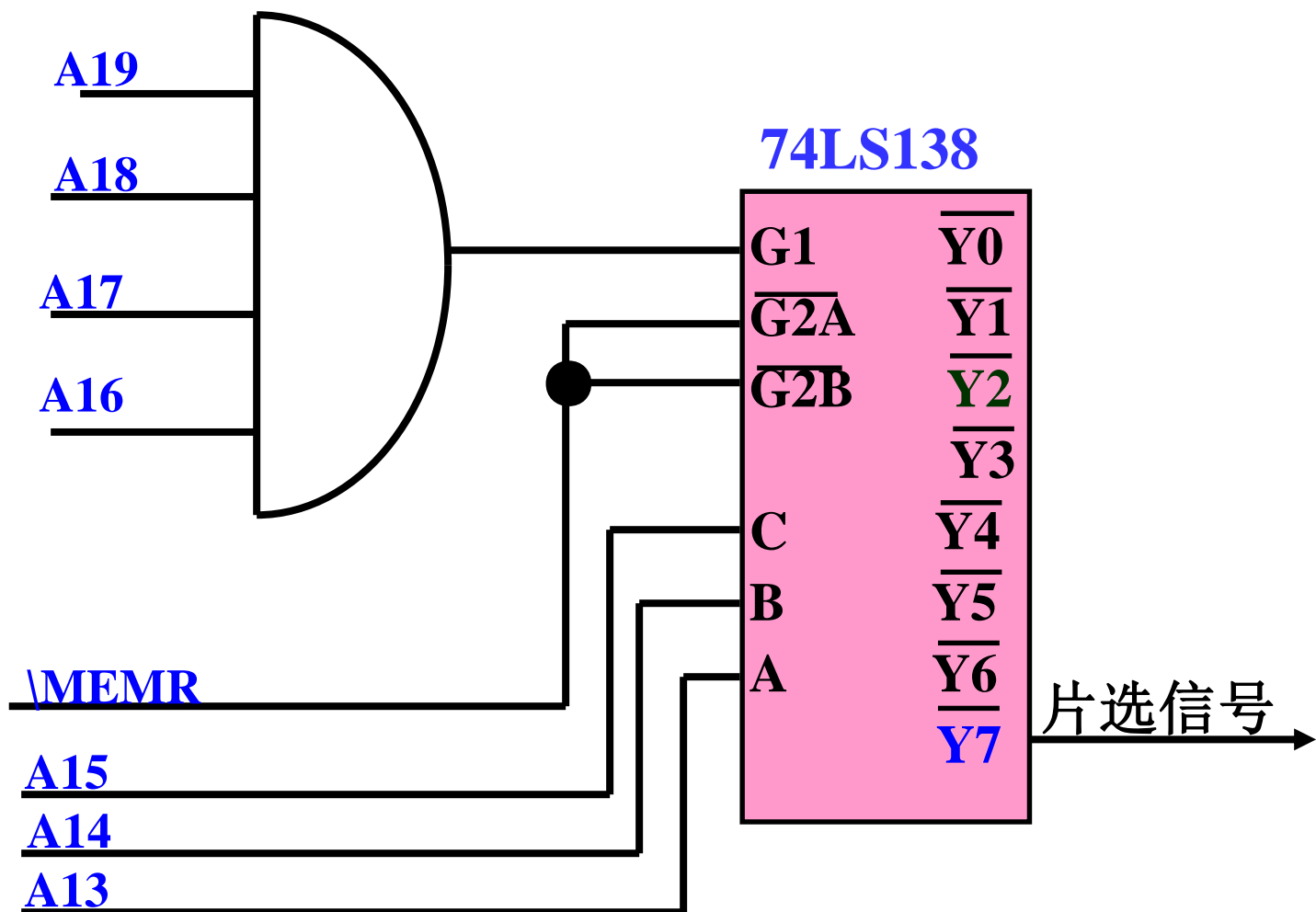
地址分析

片外寻址

与2764芯片的 $A_{12} \sim A_0$
相连，作片内寻址



译码电路设计

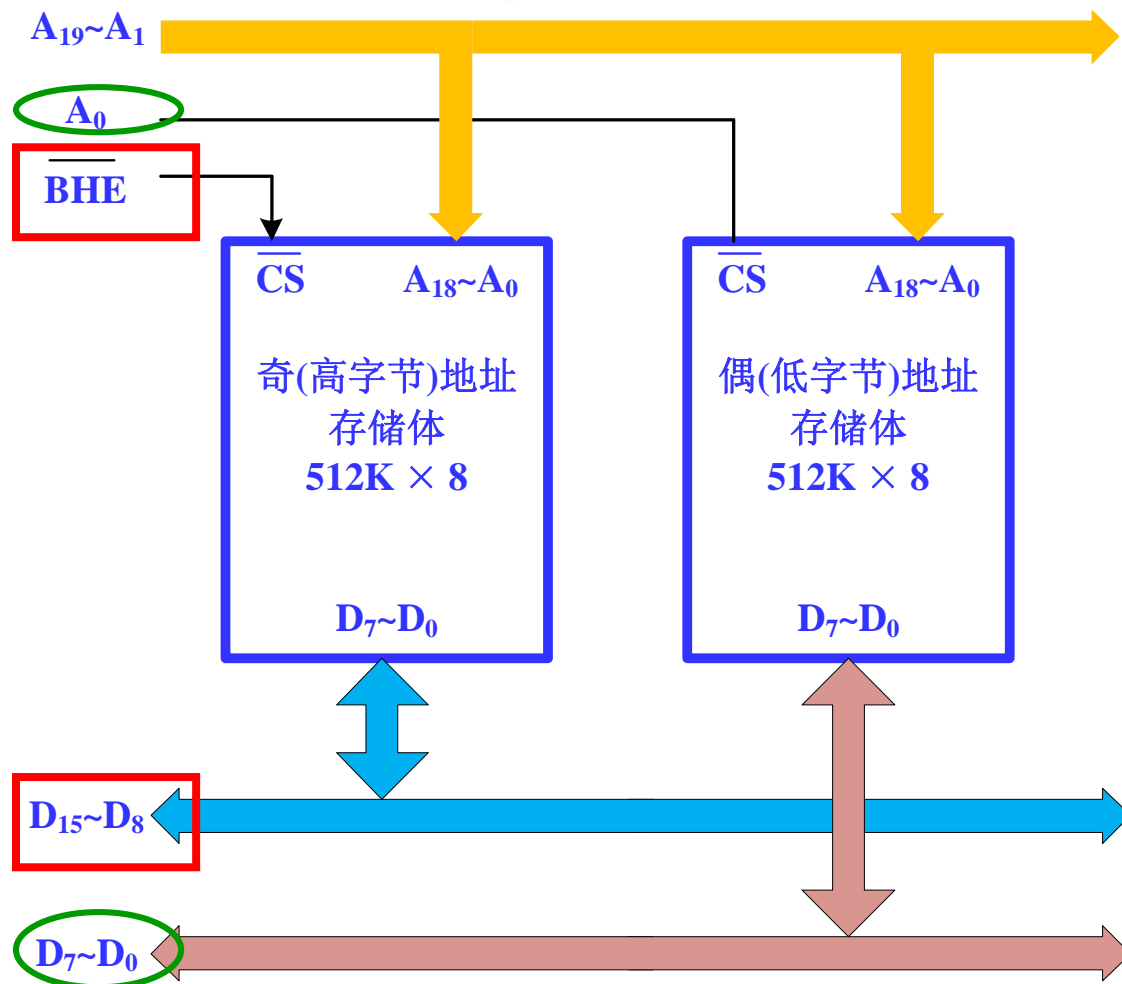




扩展存储器设计

8086系统中存储器译码电路设计

8086系统中存储器的组成





扩展存储器设计

例. 在8086最小方式系统总线上扩充设计16K字节的SRAM存储器电路。
 SRAM芯片选用Intel 6264，起始地址从04000H开始，译码电路采用74LS138。

1. 计算此RAM存储区的最高地址为多少？
2. 画出此存储器电路与系统总线的连接图。

1.求存储区的最高地址

❖ 地址范围：

A19 A18 A17 A16 A15 A14 | A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0

起始地址：04000H+0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 B

最高地址：04000H+0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 B

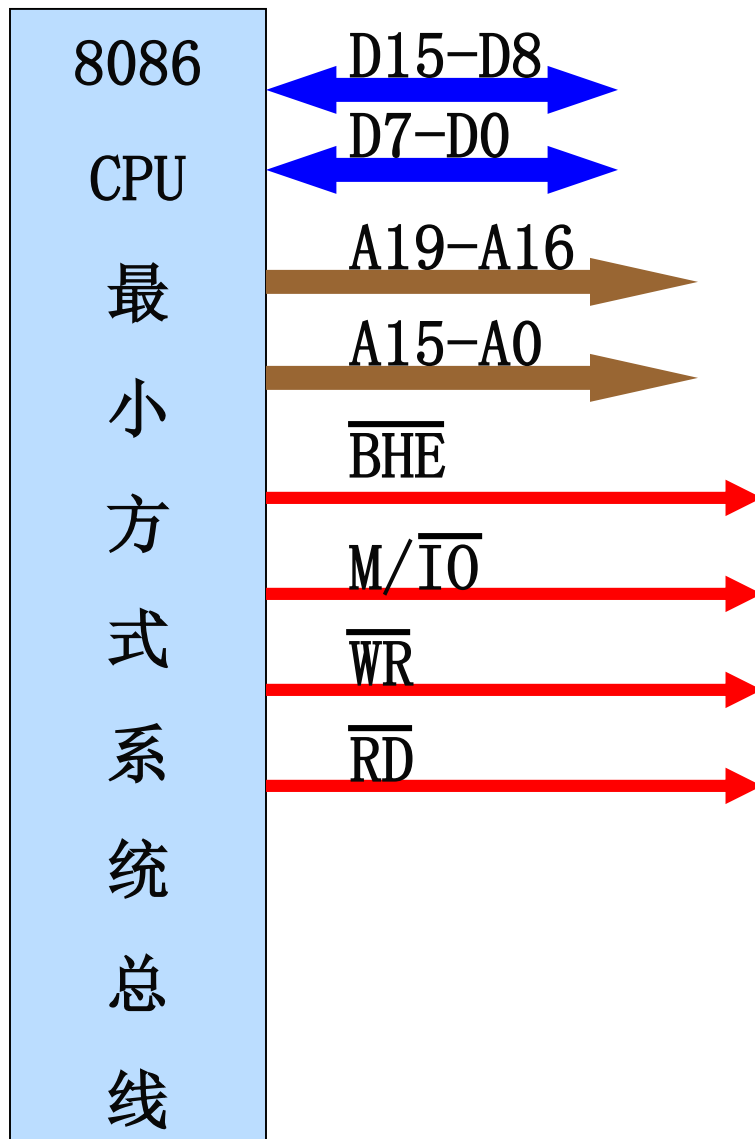
04000H+03FFFH=07FFFH



扩展存储器设计

2. 电路设计

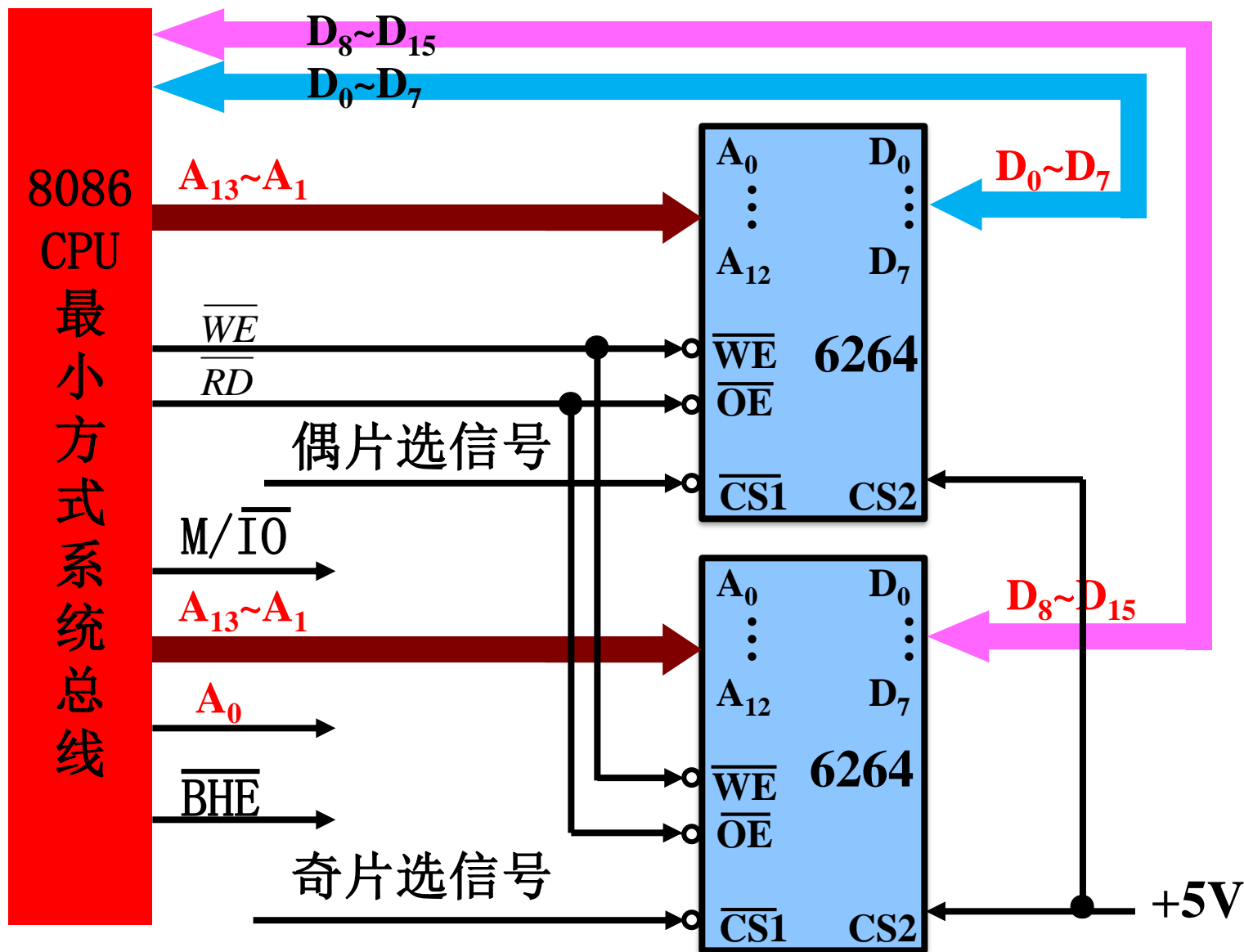
① 确定总线及总线信号





扩展存储器设计

② 电路连接



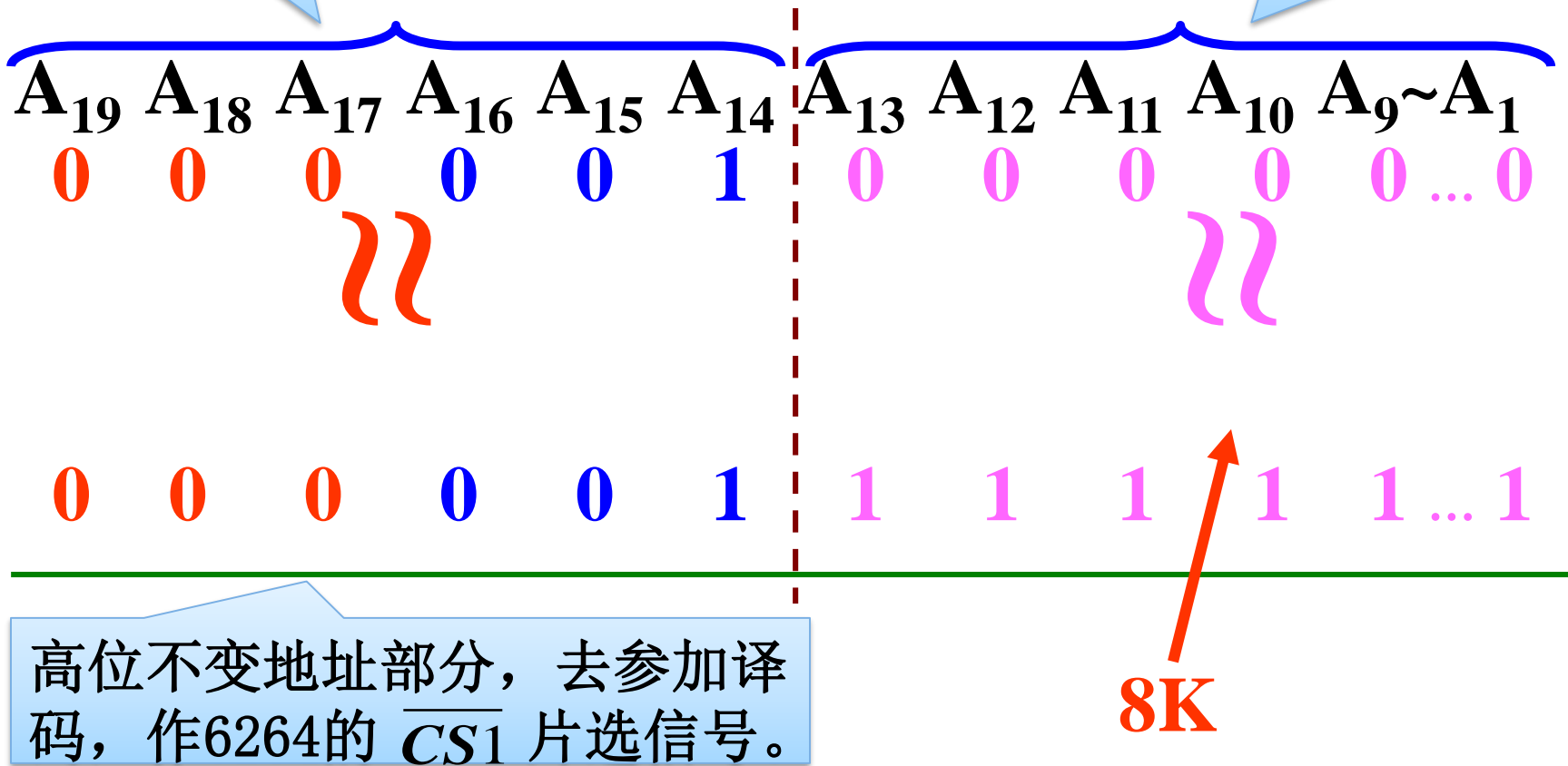


扩展存储器设计

③ 地址分析

片外寻址

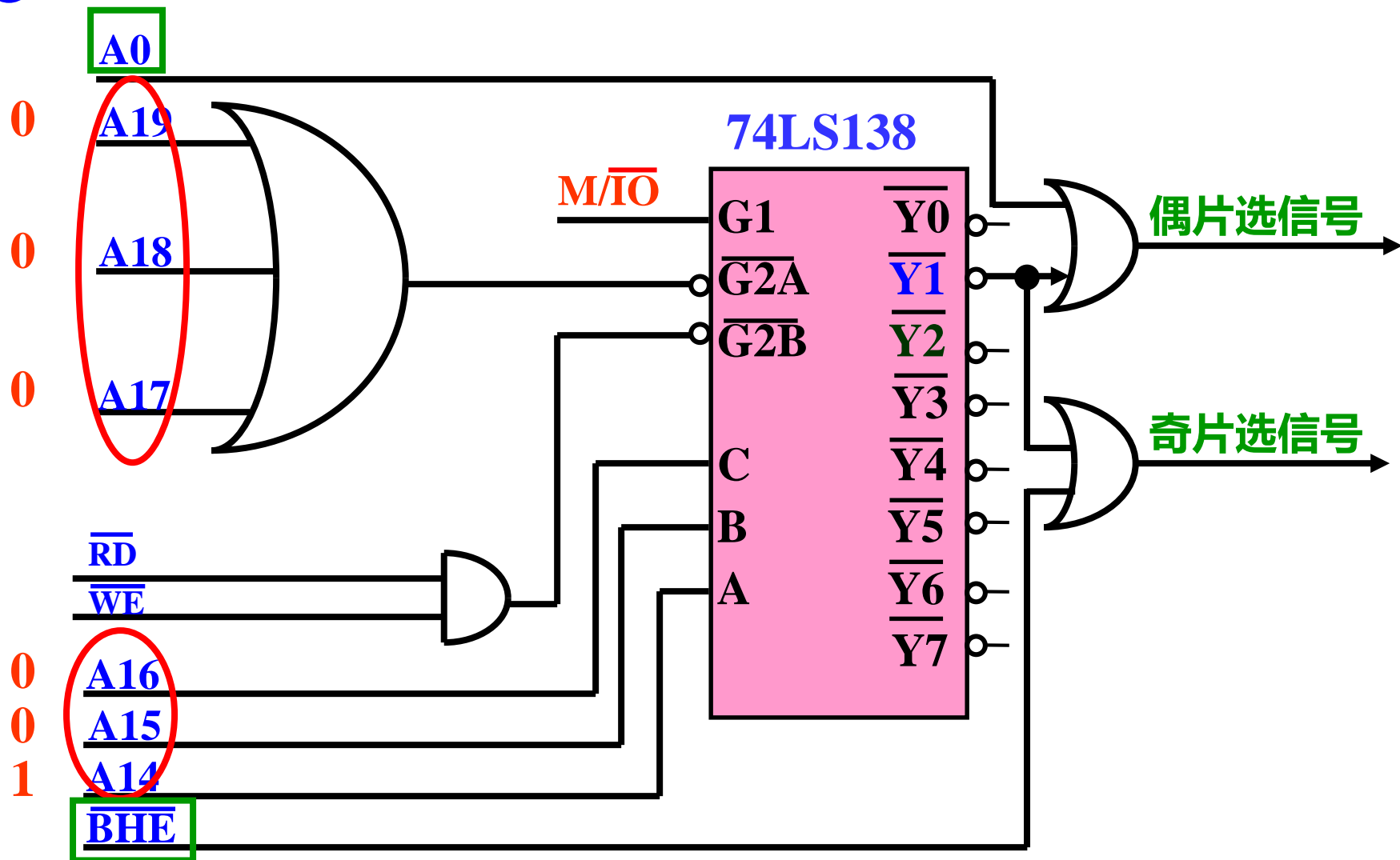
与6264芯片的 $A_{12} \sim A_0$ 相连，作片内寻址





扩展存储器设计

④译码电路设计





扩展存储器设计

总结：

8088系统中存储器扩展连接

1. 数据总线： D0—D7与存储器——连接；

2. 地址总线： A0—An与存储器——连接；

3. 控制总线：

最小方式： \overline{RD} ， \overline{WR} 分别与存储器读/写——连接；

最大方式： \overline{MEMR} ， \overline{MEMW} 分别与存储器读/写——连接；

4. 译码使能（片选）：

➤ 译码器：找出高位地址不变部分（片外地址），顺次与74LS138译码器A、B、C——连接，其余地址完成使能，控制信号（最小： \overline{RD} ， \overline{WR} ，IO/M或最大： \overline{MEMR} ， \overline{MEMW} ）纳入使能，译码输出为片选信号；

➤ 门电路：片外地址所有为1的地址分别做与非（反相器），并与所有为0的地址相或，控制信号纳入使能，逻辑输出为片选信号；



扩展存储器设计

总结：

8086系统中存储器扩展连接

1. **数据总线**：D0—D7与偶存储体——连接，D8—D15与奇存储体——连接；
2. **地址总线**：A1—An与存储器A0—A (n-1) ——连接（**错位接**）；

3. **控制总线**：

最小方式： \overline{RD} ， \overline{WR} 分别与存储器读/写——连接；

最大方式： \overline{MEMR} ， \overline{MEMW} 分别与存储器读/写——连接；

4. **译码使能（片选）**：

➤ 译码器：找出高位地址不变部分（片外地址），顺次与74LS138译码器A、B、C——连接，其余地址完成使能，控制信号（最小： \overline{RD} ， \overline{WR} ，M/IO或最大： \overline{MEMR} ， \overline{MEMW} ）纳入使能；

➤ 门电路：片外地址所有为1的地址分别做与非（反相器），并与所有为0的地址相或，控制信号纳入使能；

➤ 上面方式所产生的译码信号分别与 \overline{BHE} 和A0相或，分别产生奇存储体和偶存储体的片选信号；