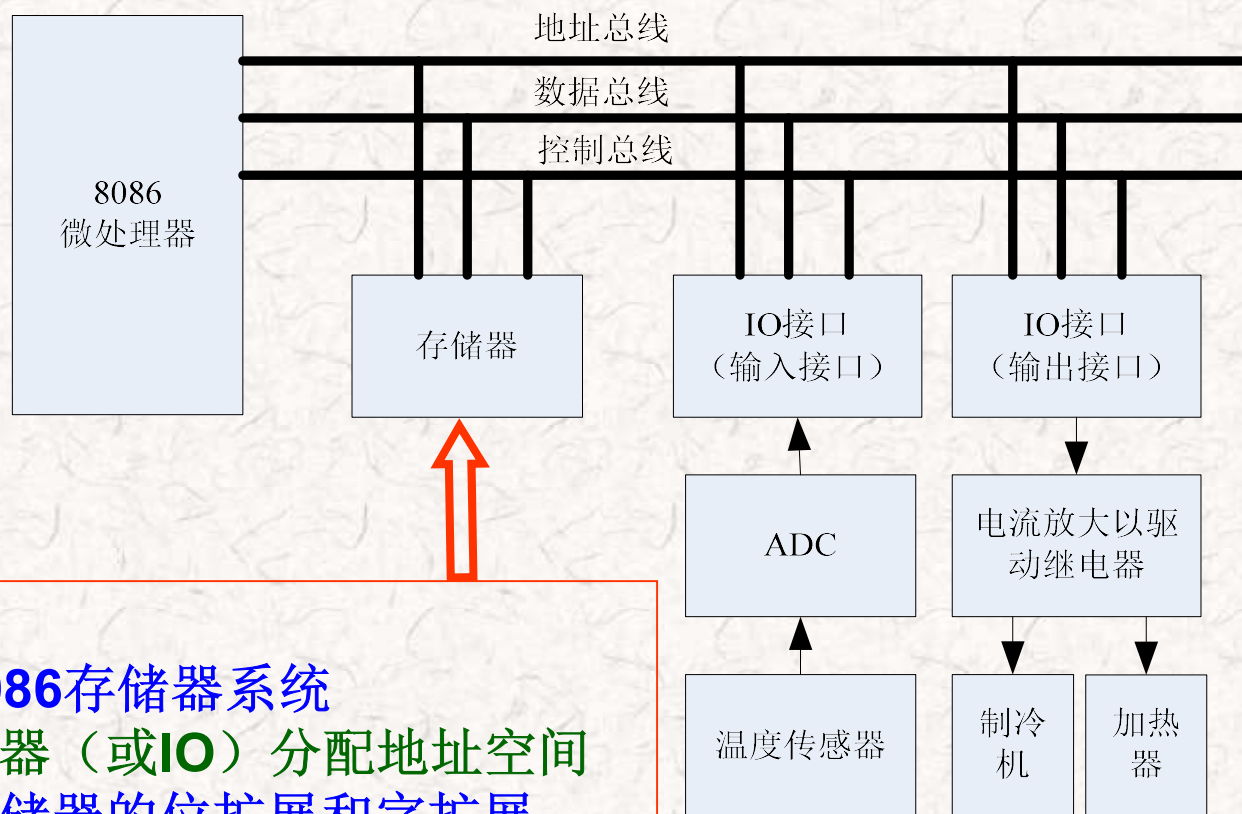




微机原理与接口技术

第五章 存储器

课程内容之构造8086存储器系统



第5章

如何设计8086存储器系统

如何为存储器（或IO）分配地址空间

如何实现存储器的位扩展和字扩展

1.计算机存储器系统概述

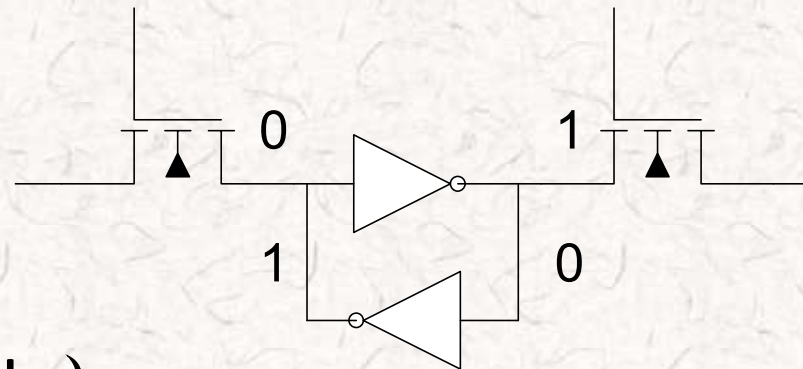
- 什么是**SRAM**？特点是什么？什么是**DRAM**？特点是什么？什么是**ROM**？生活中遇到的**ROM**有哪些？
- 计算机的主存是什么存储器构成的？计算机的高速缓存是什么存储器构成的？高速缓存用于解决什么问题？
- 在**CPU**系统中，存储器中存放的是什么？**CPU**通过哪些信号与存储器进行访问？
- 什么是地址空间？为什么给不同的存储器芯片分配不同的地址空间？

阅读课本 5.1~5.3节
其中重点5.2.1节

半导体存储器的名词解释1

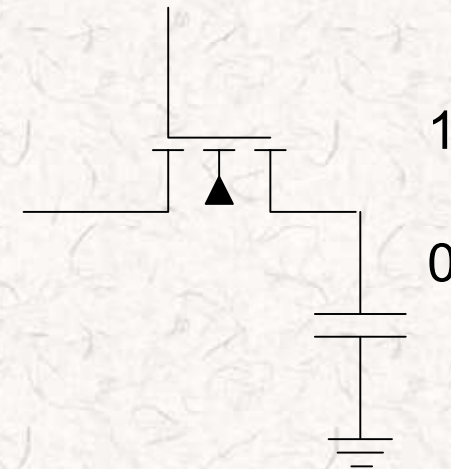
- 静态存储器

- 数据静态保持
- 单元读写速度快
- 存储密度低，成本较高
- 常用作片上存储或高速缓存（cache）



- 动态存储器

- 数据以电荷形式保存在电容上
- 存储密度高，成本较低
- 需要周期性刷新
- 访问延迟大，访问模式复杂
- 常用作现代微处理器的主存



半导体存储器的名词解释2

- 易失性存储器：没有电数据就会丢失
- 非易失性存储器：没有电数据也能保持
- 随机访问存储器：通过地址指定存储单元，相邻两次访问之间地址没有相关性
- 非随机访问存储器：地址不可指定，相邻两次访问存储单元的地址之间有相关性，例如堆栈
- 常见的名词解释：
- **SRAM**：静态随机读写存储器
- **DRAM**：动态随机读写存储器
- **ROM**：随机只读存储器

存储器的种类

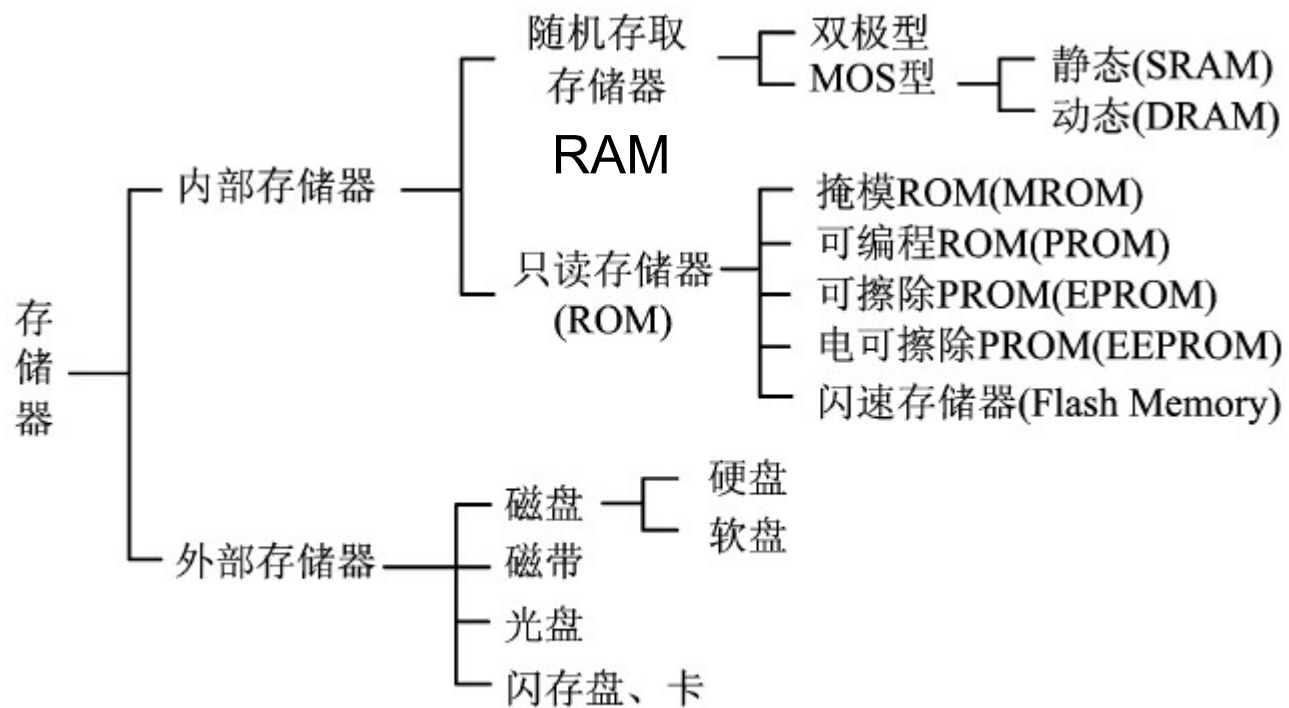
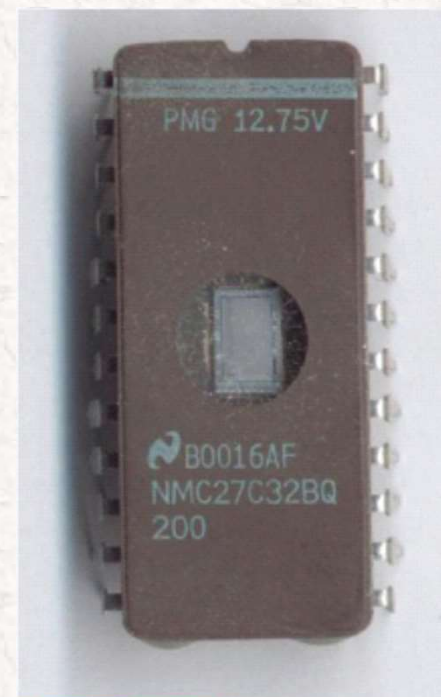
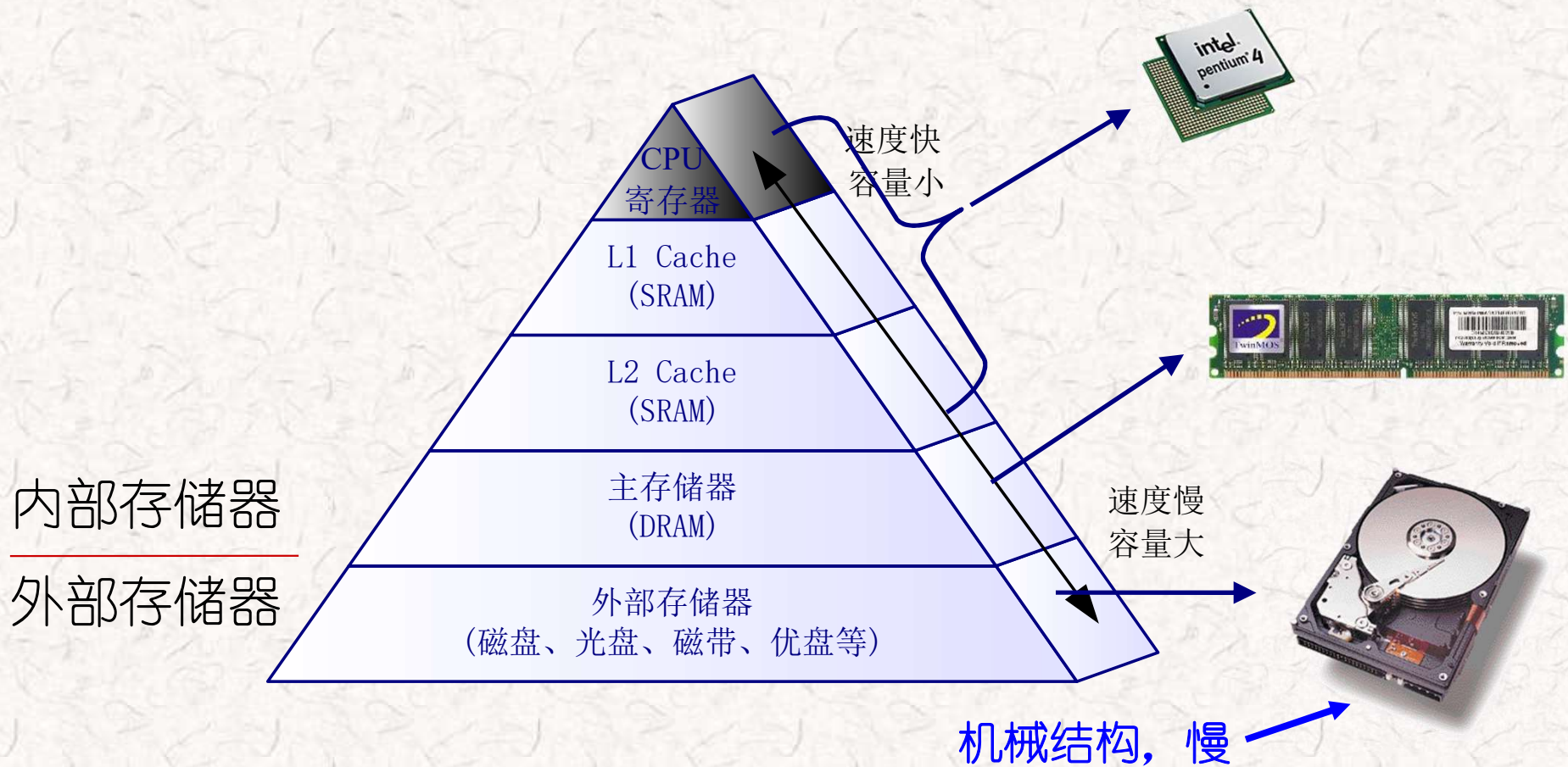


图 5.1 存储器的分类



现代计算机的存储层次



Cache简介

容量小，访问速度快，
减少CPU等待时间

容量大，访问延迟大，长突发

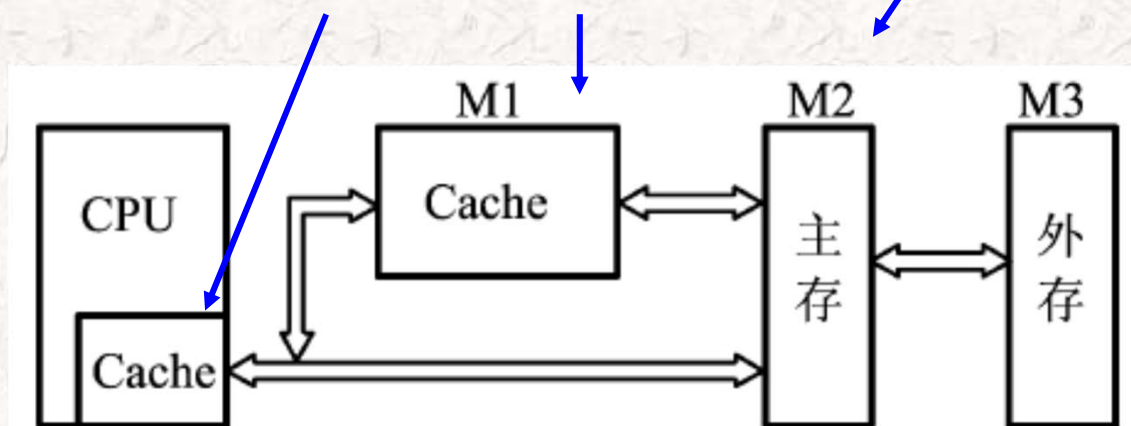


图 5.24 计算机的三级存储系统

高速缓存的设计依据：
利用存储器数据访问的
时间局部性和空间局部性

Cache的功能

1. 命中判断
2. 自动加载
3. 缓存内容更替

cache保存数据和数据对应的地址，通过比较地址判断相应的数据数否在**Cache**中

本章内容安排

- 了解存储器的接口、信号时序
- 如何将**CPU**的信号连接到存储器/存储器模块
- 如何将多片存储器联合起来实现位宽和深度的扩展

本课程中常用存储器

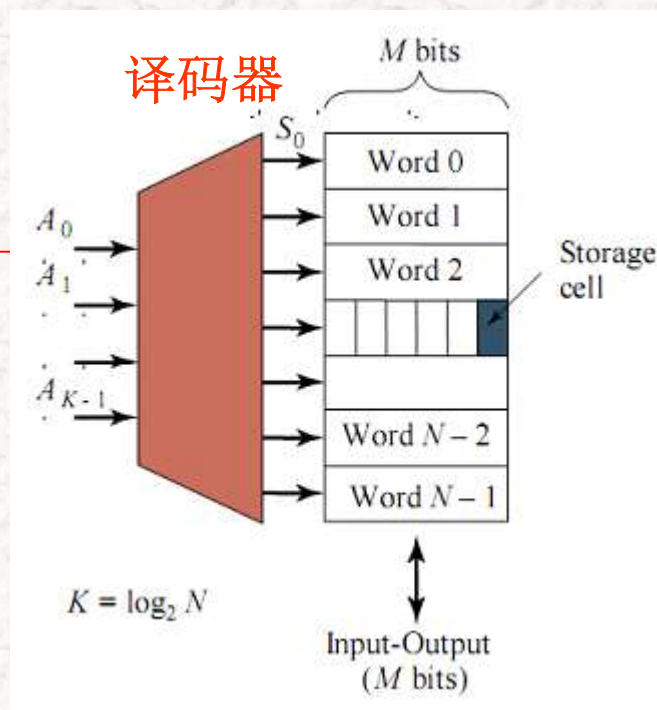
- **SRAM**
- 根据地址随机读写
- 存放数据，也可用于存放程序
- **EPROM**
- 根据地址随机只读，特殊条件下可编程
- 存放程序
- 二者访问接口相似
- 地址线、数据线、片选线、读/写（读）控制线



存储器基本结构与容量表示

- 存储器芯片中每个单元位宽相同，且有唯一的地址
- K 根地址线可以选择 2^K 个存储单元
- 存储器容量表示：
- 单元个数 * 每个单元位宽
- $2^K * M \text{ bit}$

A_{k-1}	A_{k-2}		A_0
0	0	00000000000000	0
0	0	00000000000000	1
.....			
1	1	11111111111111	1



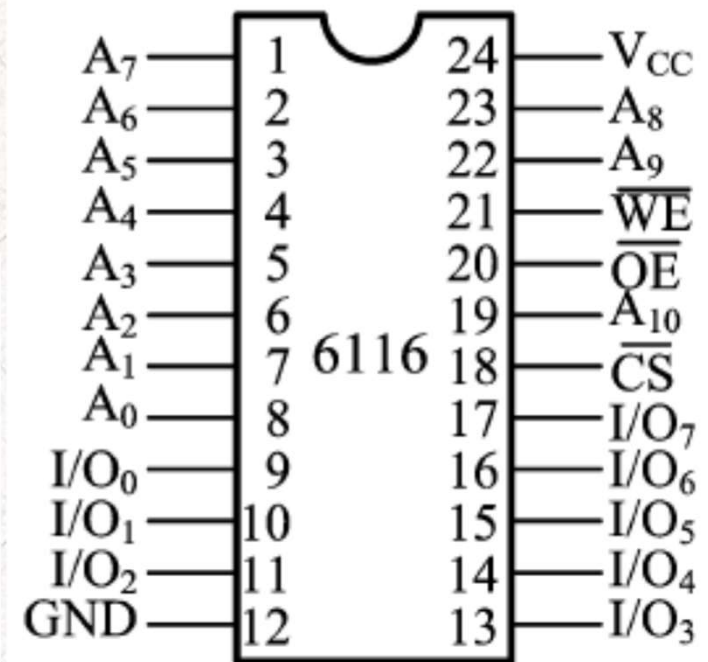
一个SRAM的引脚

- **A10:0** 地址线
- **I/O7:0** 数据线
- **/WE**: 写信号, 低电平有效
- **/OE**: 读信号, 低电平有效
- **/CS**: 片选信号, 低电平有效
- **VCC**: +5V电源
- **GND**: 地

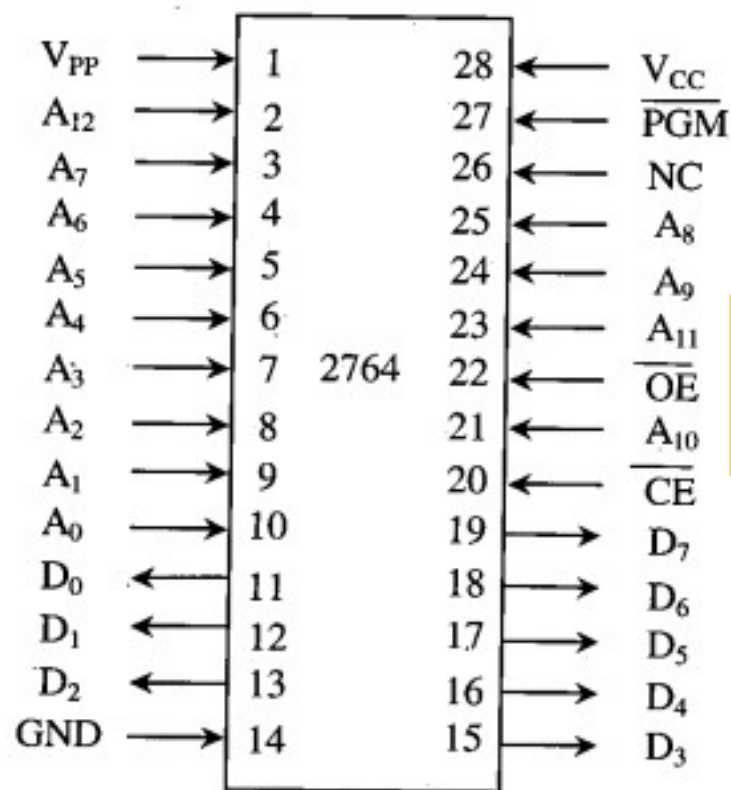
2048*8bit
(2K*8bit)

表 5.1 6116 存储器的工作方式

\overline{CS}	\overline{OE}	\overline{WE}	工作方式
0	0	1	读
0	1	0	写
1	X	X	未选



一种EPROM芯片的引脚



A12:0: 地址线

D7:0: 数据线，读出时输出，编程时输入

/OE: 输出使能，读信号，低电平有效

/CE: 芯片使能，片选，低电平有效

/PGM: 编程时输入低电平脉冲

VPP: 编程时接+12V电压，读出时接+5V（早期曾用+21V、+24V等）

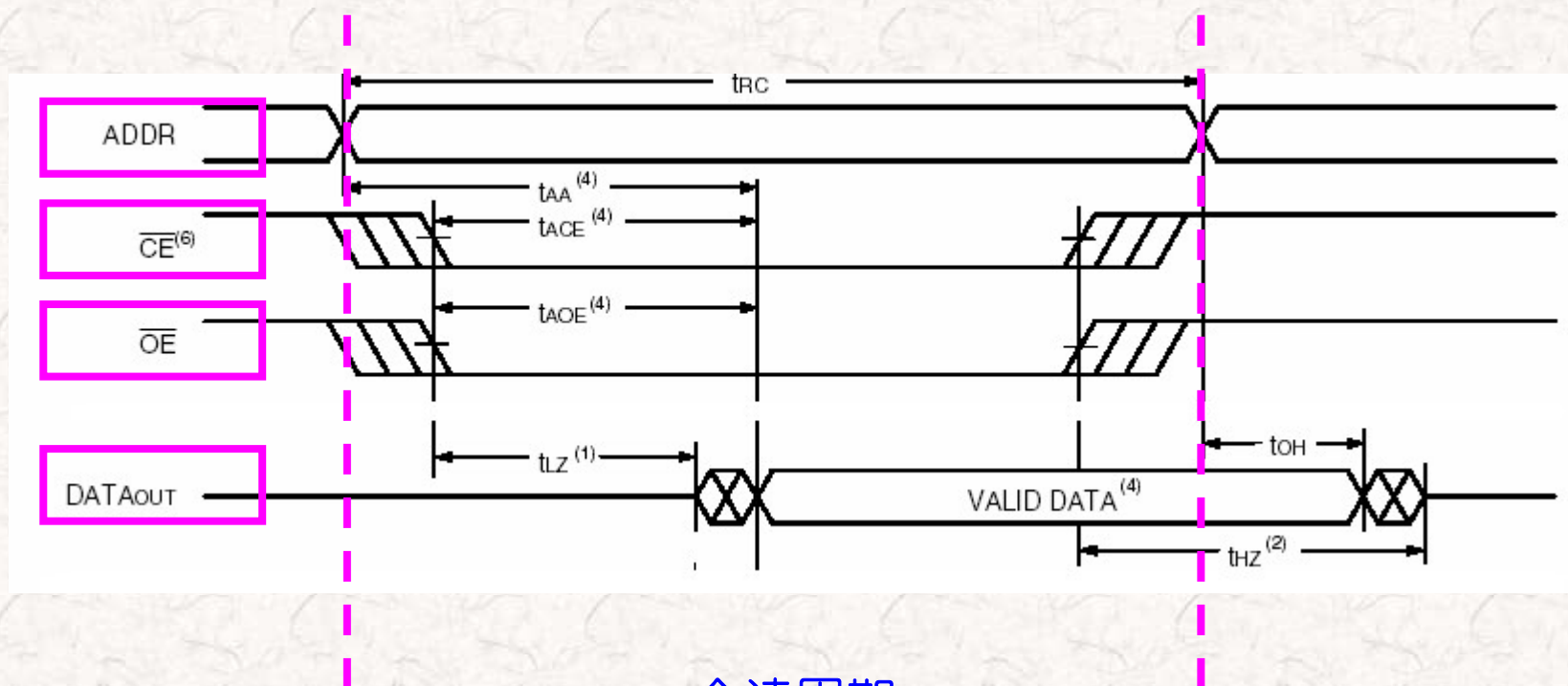
VCC: +5V电源

GND: 地

8K*8bit

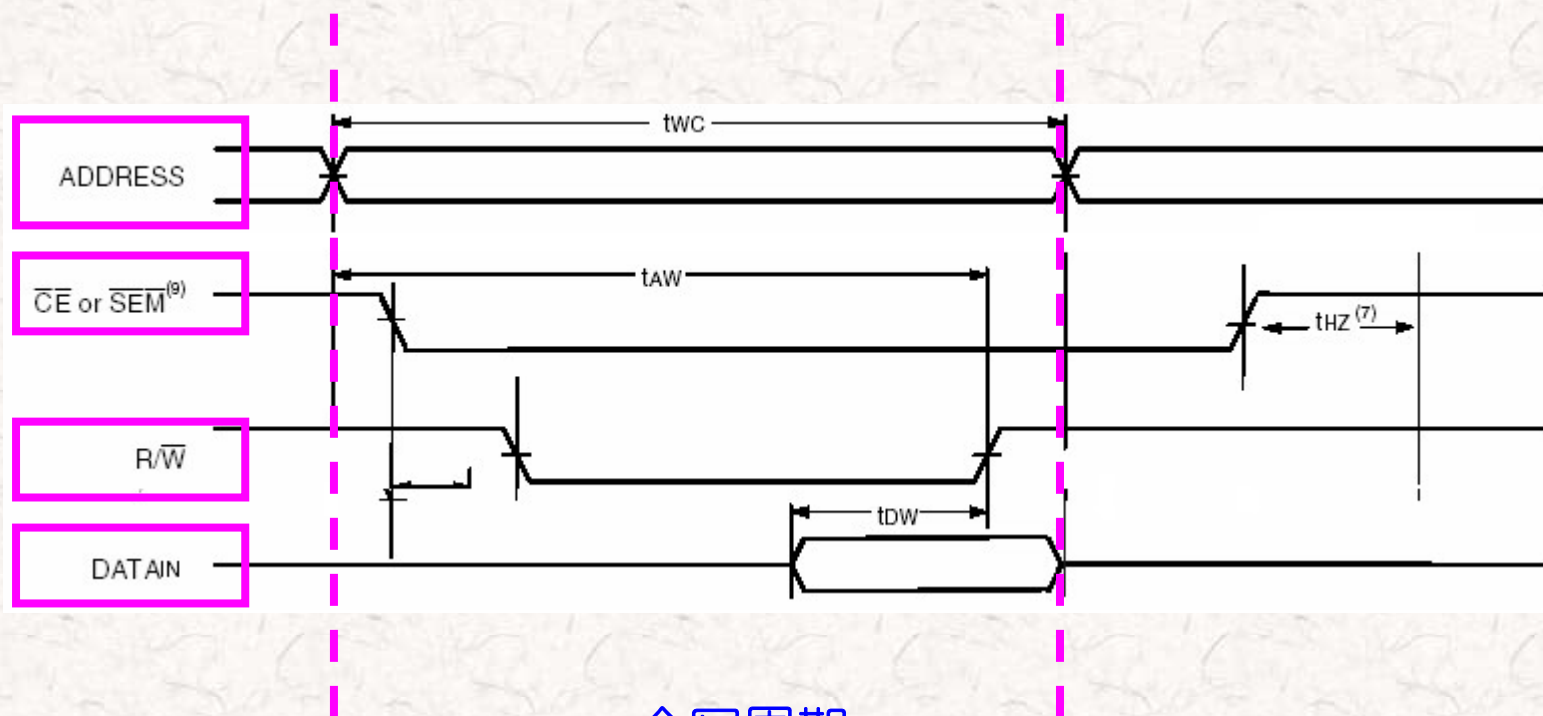
EPROM、FLASH等读操作与SRAM相同

SRAM的操作--读操作



一个读周期

SRAM的操作--写操作



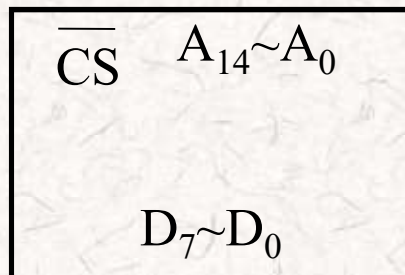
一个写周期

存储器访问端口信号总结

- 存储器端口信号有：地址线、数据线、片选、读/写
- 以地址稳定为一个读或写周期
- 片选有效是芯片被读写的前提条件
- 数据总线是双向的
- 写操作中，数据应和读写信号同时施加，由CPU驱动
- 读操作中，数据在地址、读写、片选施加之后经过一定响应延时出现在数据端口，由存储器驱动
- 读写信号指示当前访问周期是读操作还是写操作
- K根地址线的存储器通常含有 2^K 个存储单元，每个单元的位宽为该存储器输入输出数据端口的数量

练习1-认识存储器

下图是存储器模块的示意图，请说明引脚的含义、有效电平、存储器的容量



32K*8bit

A14:0: 地址线

D7:0: 数据线

/CS: 芯片使能，片选，低电平有效
(读写信号有时不画出)

练习2-存储器类型

- 1、高速缓存的存取速度^C。
- A、比内存慢、比辅存快 B、比内存慢，比内部寄存器快
- C、比内存快，比内部寄存器慢
- 2、下面的说法中，正确的是：^D。
- A、EPROM是不能改写的
- B、EPROM是可改写的，所以也使一种读写存储器
- C、EPROM只能改写一次
- D、EPROM是可改写的，它不能作为读写存储器
- 3、若256Kbits的SRAM具有8条数据线,则它具有^B条地址线
- A、16 B、15 C、20 D、32

每单元8bit位宽， $256\text{Kbit}=32\text{K}\times 8\text{bit}$

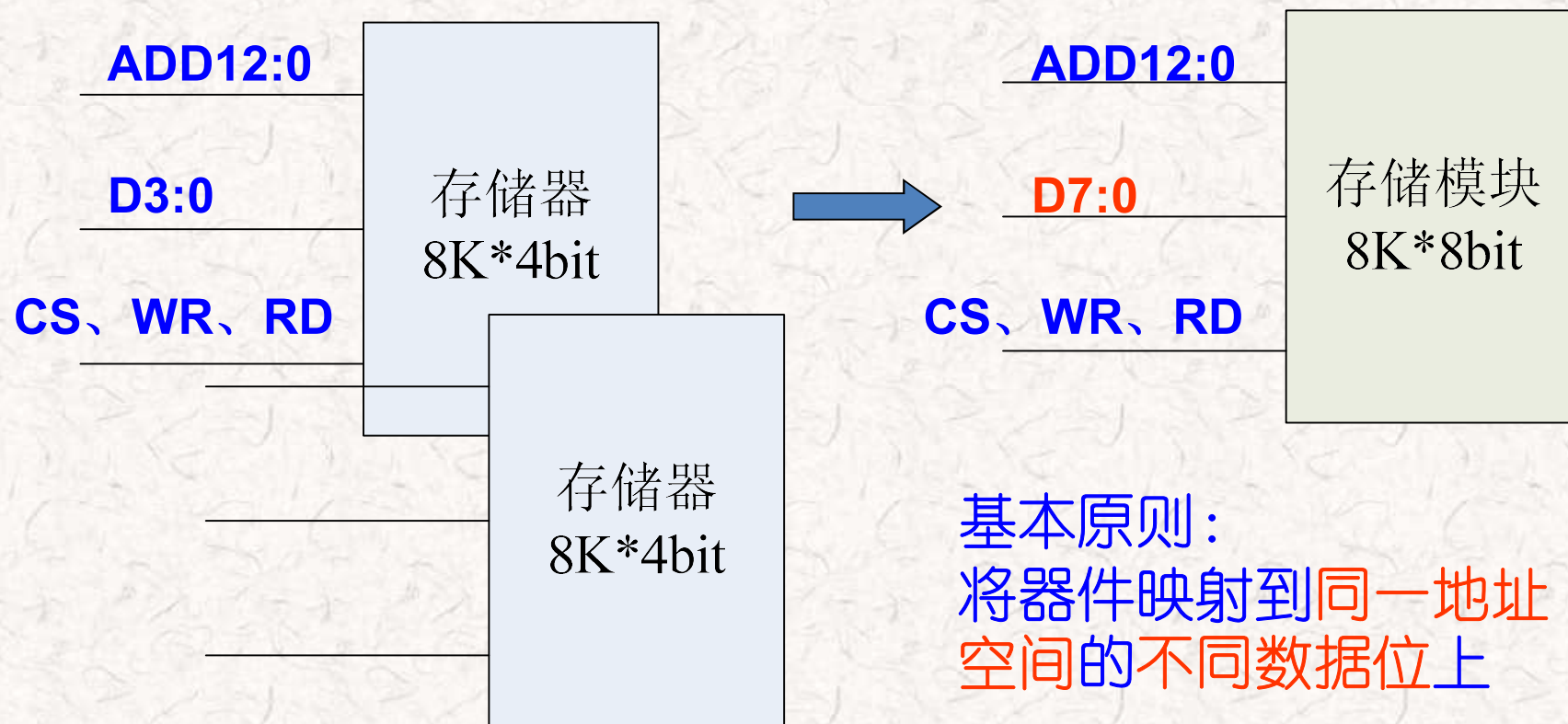
构成所需容量存储器的方



- 什么是存储器扩展？
- 例：内存颗粒**1G*16bit**，内存条**4G*64bit**
- **位扩展**-- 使用较小位宽的存储器组成较大位宽的存储模块
- 例：单个存储器芯片**2K*4bit**，存储模块要求**2K*8bit**
- **字扩展**-- 使用较少存储单元数量的存储器组成较多存储单元数量的存储模块
- 例：单个存储器芯片**2K*8bit**，存储模块要求**4K*8bit**

存储器位扩展分析

用8K*4bit存储器组成8K*8bit存储模块



存储器位扩展的实现

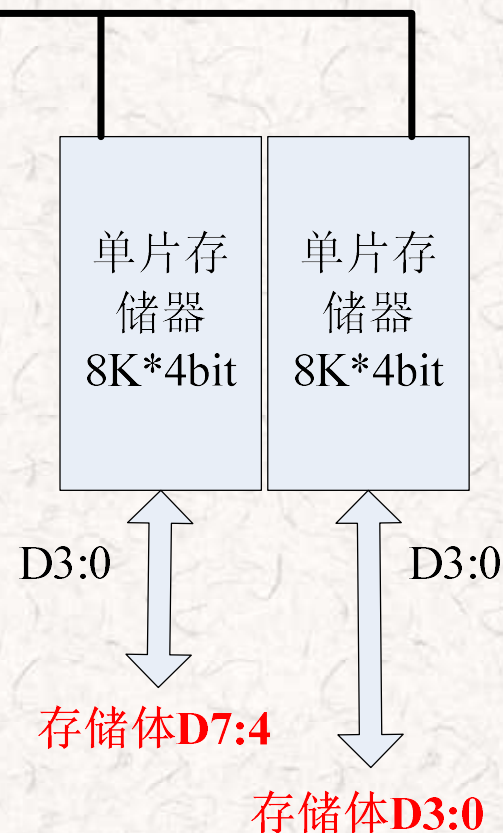
两片8K*4bit存储器组成8K*8bit存储模块

存储体与存储器信号之间的连接特点（仅存在位扩展的情况）：

- (1) 每一片地址和读写控制信号并联
- (2) 数据接口各自独立，并行输出，实现位扩展
- (3) 无片间寻址，所有片选信号并联

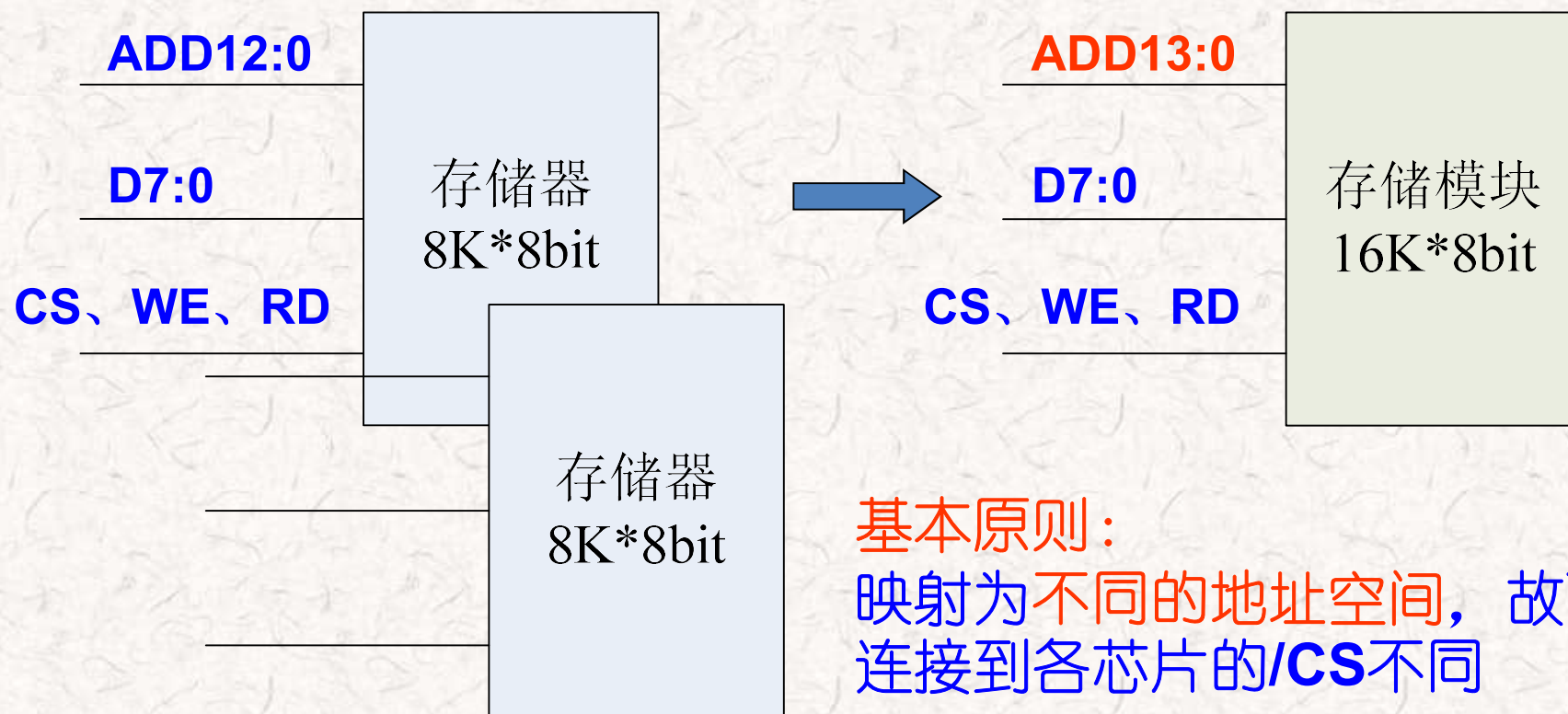
/CS决定芯片所在地址空间，两片位扩展存储器映射在同一地址空间，故**/CS**并联

存储模块 ADD12:0 & Control



存储器的字扩展分析

用8K*8bit存储器组成16K*8bit存储模块

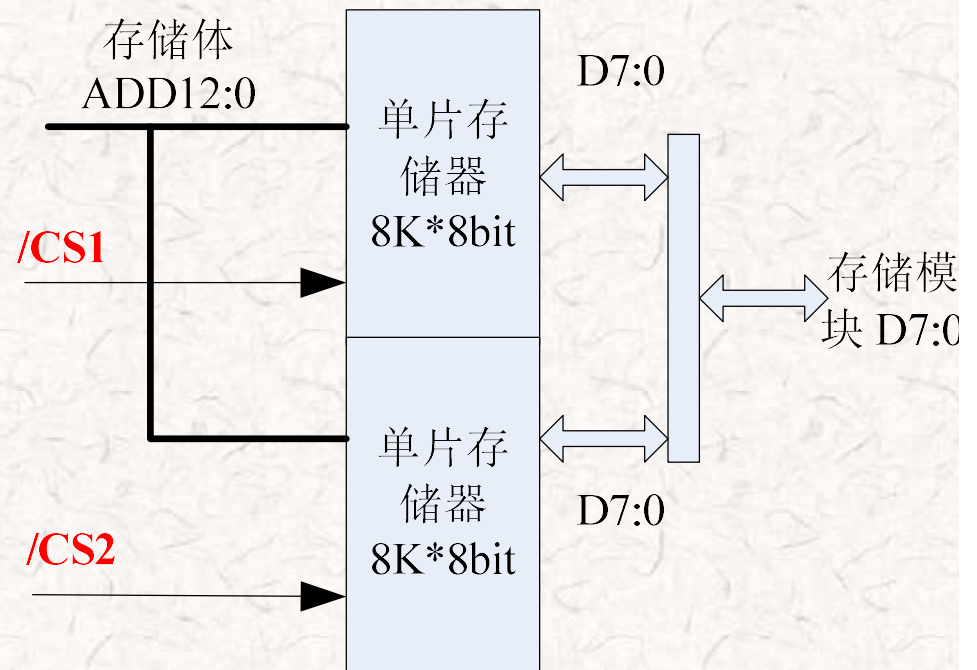


存储器字扩展的实现

设采用两片8K*8bit存储器组成
16K*8bit存储体

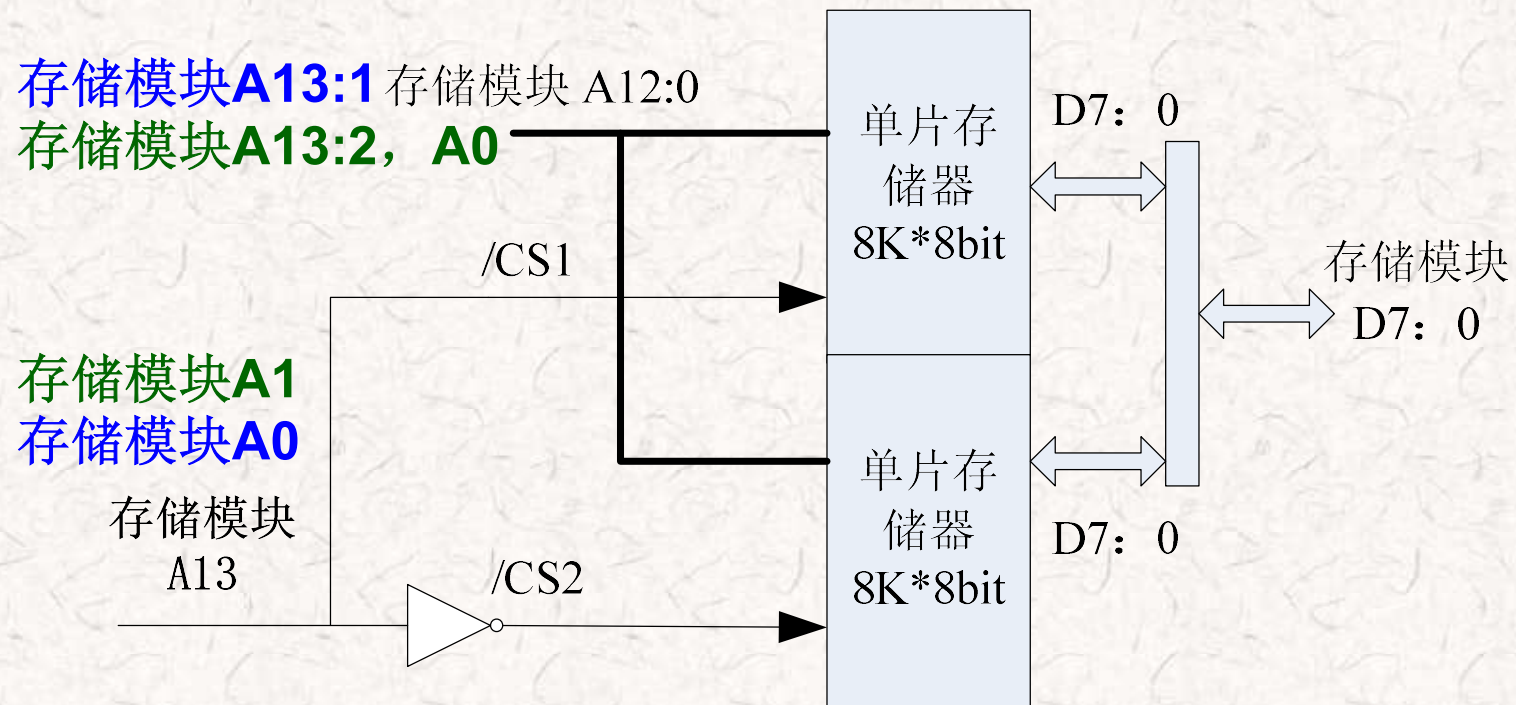
存储体与存储器信号之间的连接特点
特点（仅存在字扩展的情况）：

- (1) 单片地址和读写控制信号并联
- (2) 数据接口并联
- (3) 片选信号各自独立，且同一时刻只允许一个片选有效



问：总地址空间大于单个器件覆盖的地址空间，多出来的地址线做什么用？

生成两个片选信号的方式之一



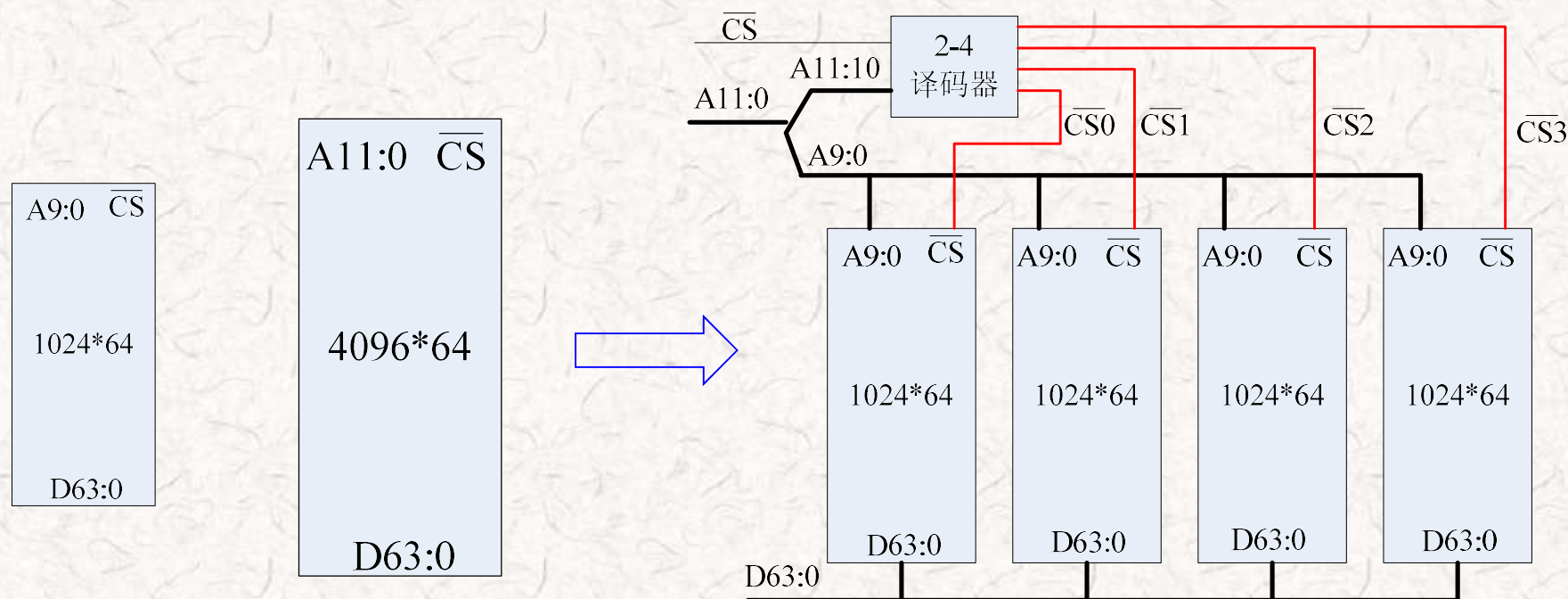
总结：将高位地址线做片选，低位地址线用于片内单元寻址，可以保证片内单元映射地址连续

IC设计中存储器

扩展举例1

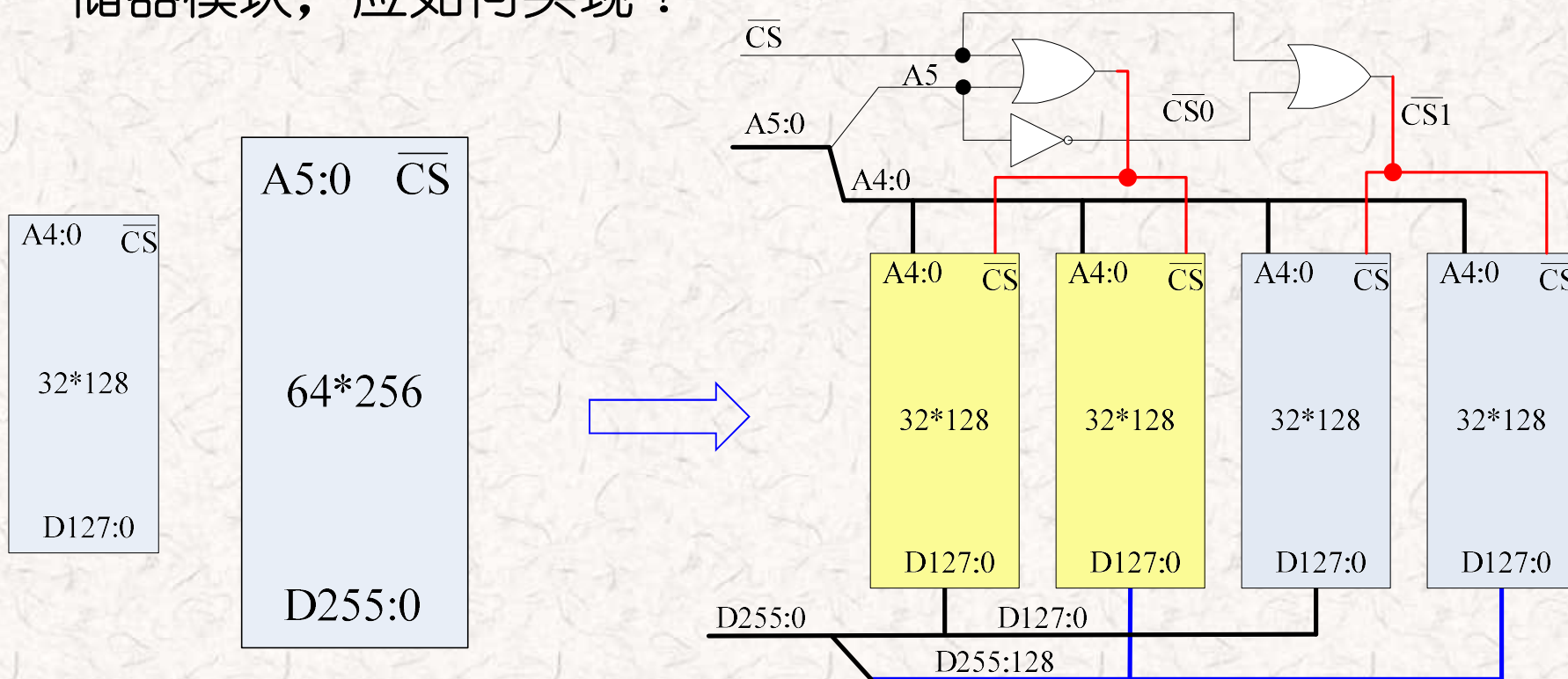
```
assign cs0=( !cs && A[11:10]==2'b00) ? 1'b0:1'b1 ;
```

- 假设MC生成单块最大容量为**1024*64bit**，需要使用**4096*64bit**的存储器模块，应如何实现？



IC设计中存储器扩展举例2 `assign cs0 = cs | A[5];`

- 假设MC生成单块位宽为 $32 \times 128\text{bit}$ ，需要使用 64×256 的存储器模块，应如何实现？



存储器在单一位宽的微处理器地址空间的映射

设某8位处理器地址空间1MB，要求将128KB的存储模块映射到该地址空间的40000H开始连续的128K地址空间

A19	A18	A17	A16	A1	A0
0	1	0	0		0	0
0	1	0	0		0	1
					
0	1	0	1		1	1

存储模块内部地址连续，表明应采用低位地址
A16:0对模块内部单元寻址

高位地址参与片选生成，决定其映射的地址空间

FFFFFH

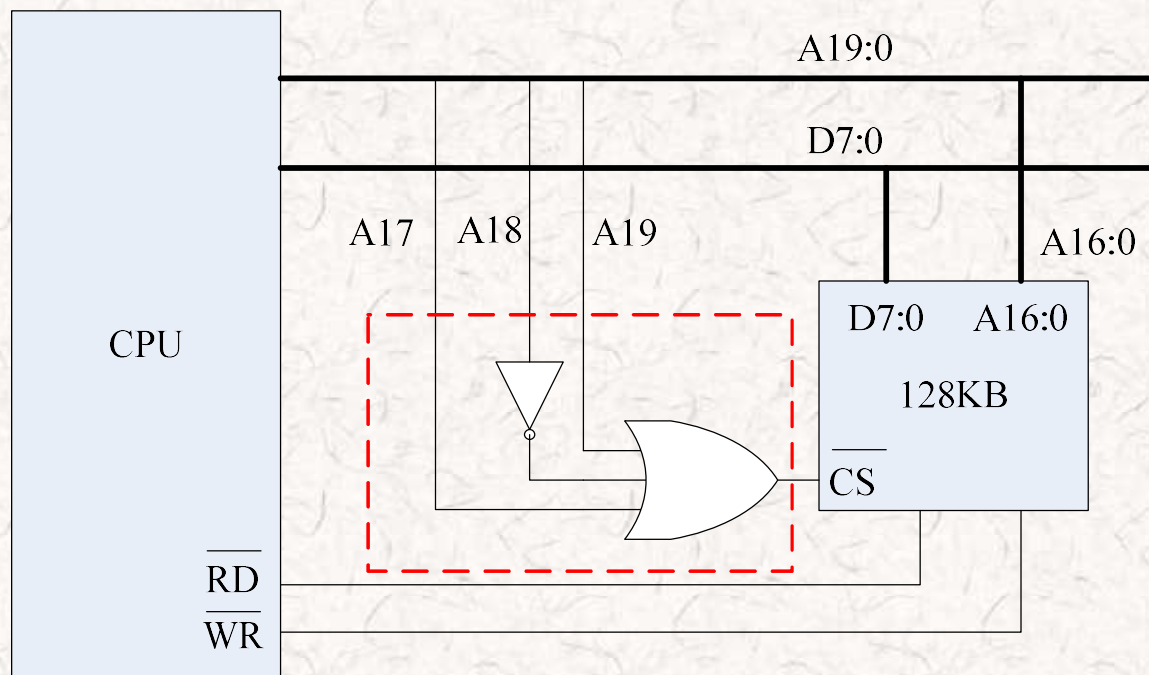
5FFFFH

40000H

0000H

128KB

前述设计的实现

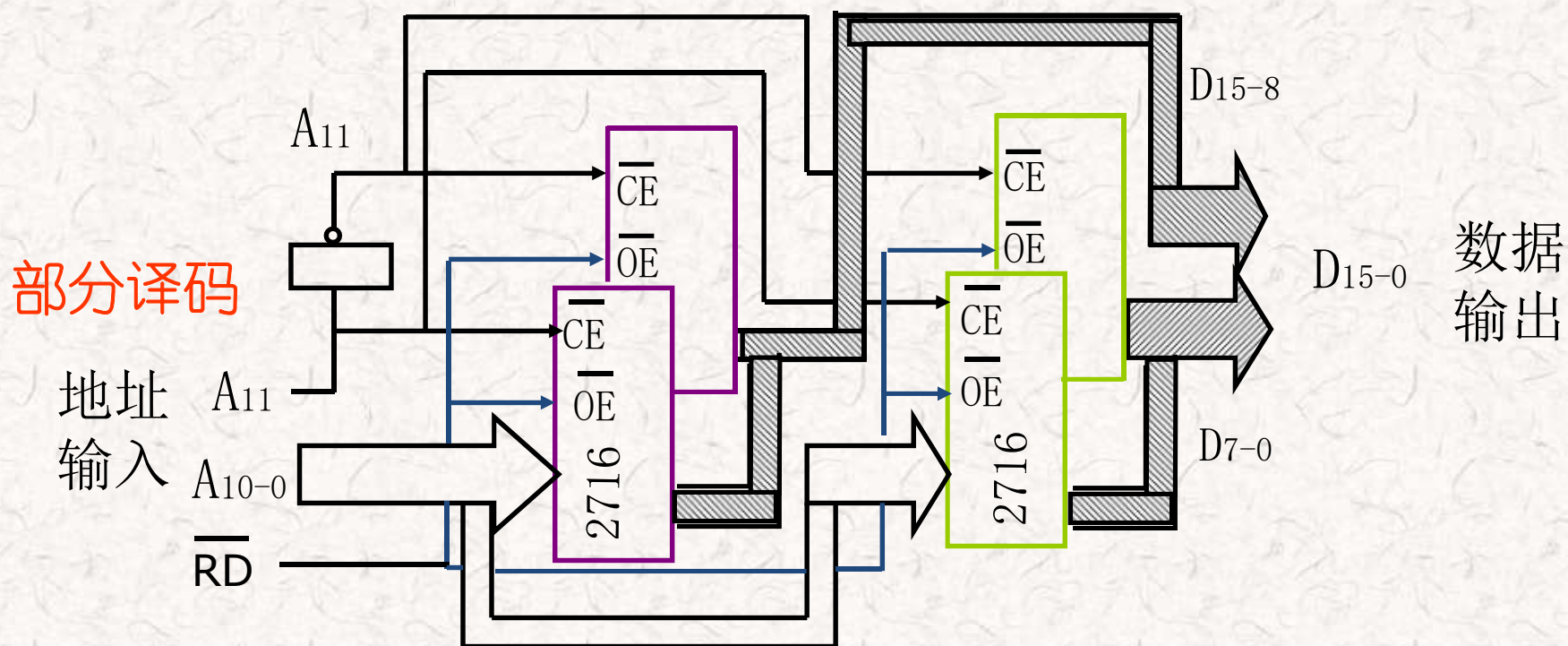


除了模块内部寻址的地址线，其它地址线均参与片选生成译码，称为**全译码**

存储器在地址空间的映射

下图，4片**2716**组成某具有**14根**地址线的**16位CPU**的程序存储系统，分析存储系统的总容量和各芯片的映射空间

1. 所构成的存储系统的总容量？
2. 哪些字扩展，哪些位扩展？
3. 每片存储器地址范围？设未使用的高位地址线为零
4. 该CPU的地址空间？



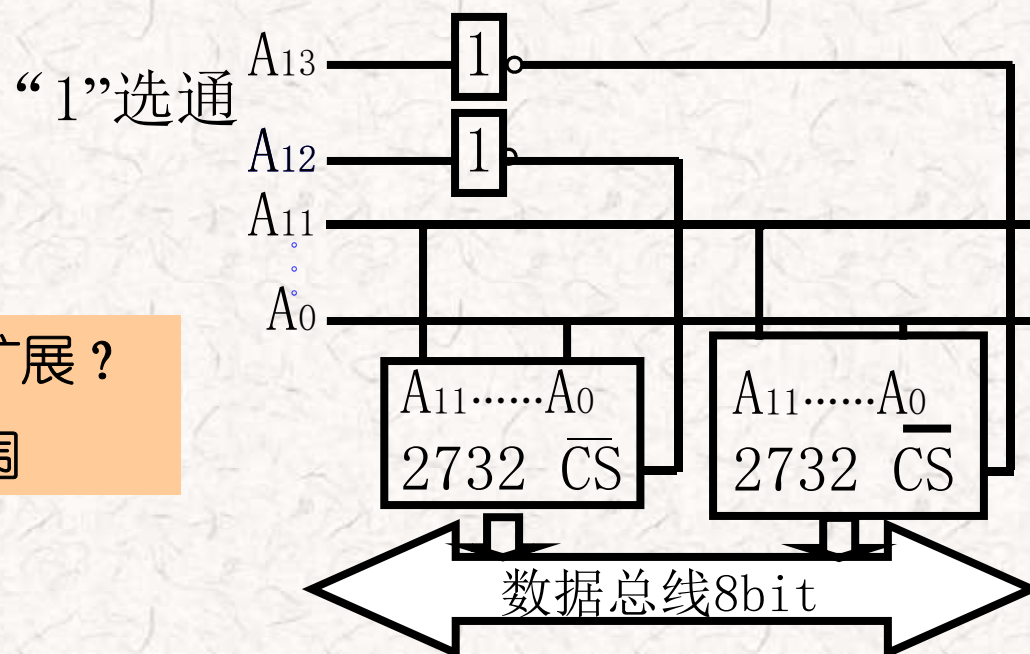
部分译码的影响—浪费地址空间

A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
X	X	0	0	0	0	0	0	0	0	0	0	0	0
X	X	0	0	0	0	0	0	0	0	0	0	0	1
.....				
X	X	1	1	1	1	1	1	1	1	1	1	1	1

- 存储器总容量：**4K*16bit**
- 实际占用了**16K*16bit**空间：**0000H~3FFFH**

注意译码冲突--线选法

1. 两块存储器是字扩展还是位扩展？
2. 请写出两块存储器的地址范围

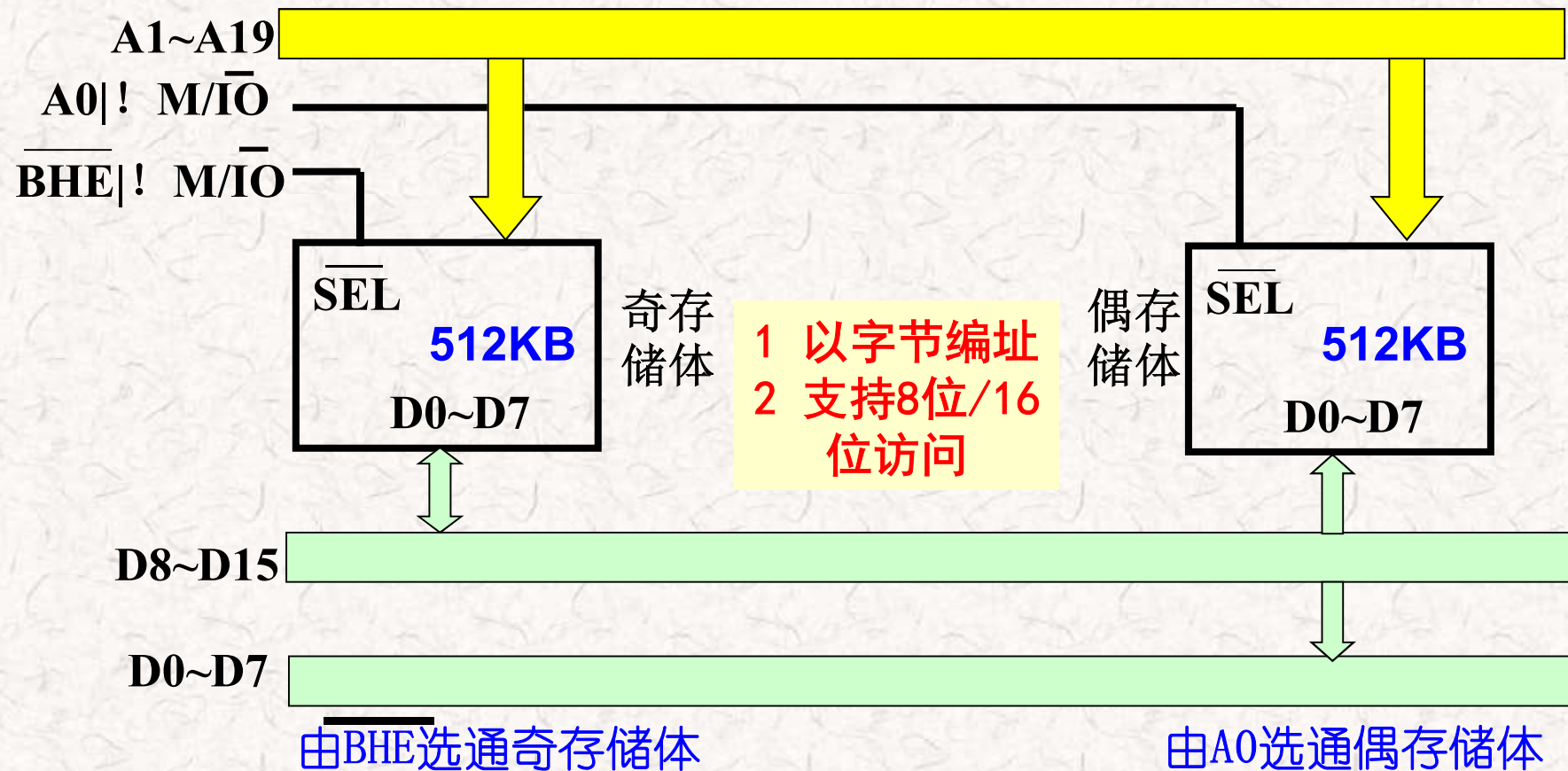


	最高地址	最低地址
第一块 (左)	X1 1111111111111	X1 0000000000000
第二块 (右)	1X 1111111111111	1X 0000000000000

如何为8086设计存储系统

- 1. 必须遵守的特点：
 - 8086地址空间的构成特点
 - 8086存储器系统的构成特点
- 2. 针对具体系统的设计需求：
 - 根据系统的需要确定：存储系统的总容量
 - 根据系统的需要为存储系统分配地址空间
- 3. 具体实现：
 - 用什么存储器芯片来实现
 - 信号如何连接

如何为8086设计1MB存储空间？

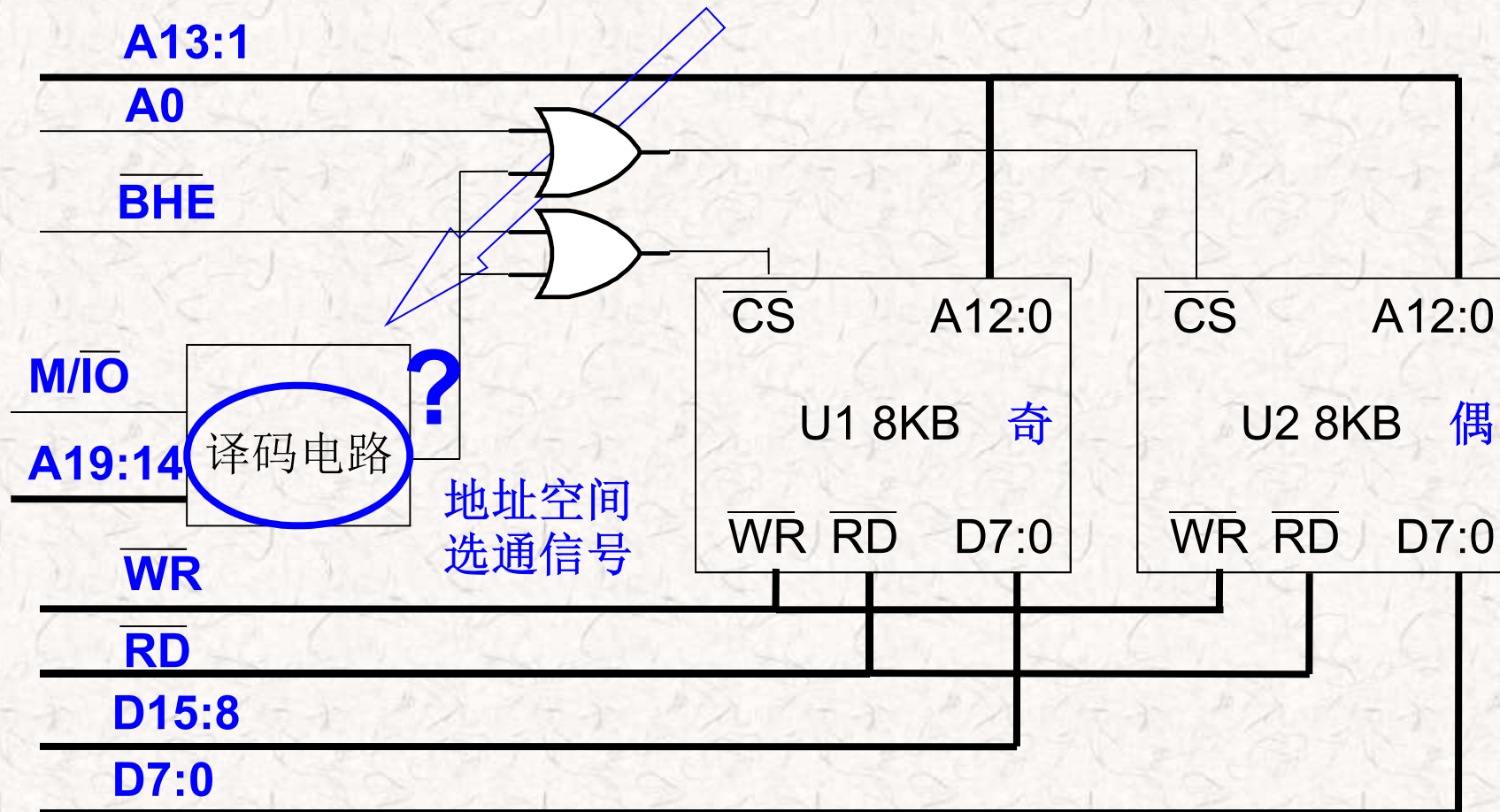


为8086设计指定容量与指定地址范围的存储系统

- 举例：如何为8086设计一个总容量为16KB的可读写存储器？要求占用90000H开始连续的16K地址空间，采用单片容量8K*8bit的SRAM（A12:0, D7:0）芯片实现
- 分析设计目标
- 了解拥有的器件，确定8086每个存储体的容量，需要几片？
- 单个存储体8KB，1片/存储体
- 根据题目要求，存储系统和每个存储体覆盖的地址空间分别是什么？
- 存储系统覆盖90000H~93FFFH，偶存储体占据其中的偶地址，奇存储体占据其中的奇地址
- 如何实现地址空间分配，以及如何区分奇偶存储体？
- 采用未使用的高位地址译码，译码信号作为存储系统的片选，再用/BHE和A0区分奇偶存储体

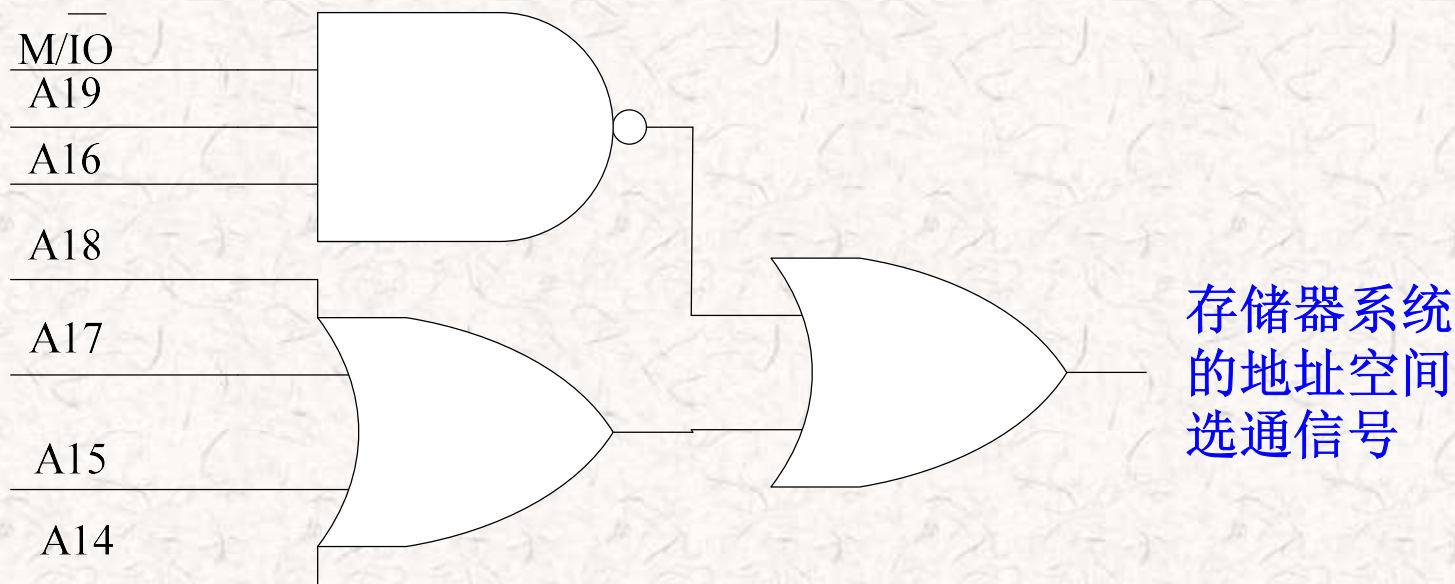
电路连接

需要当: A19 A18 A17 A16 A15 A14 输出=0
1 0 0 1 0 0



方法1：用逻辑门进行译码

需要当： **M/IO** **A19** **A18** **A17** **A16** **A15** **A14** 输出=0
 1 1 0 0 1 0 0

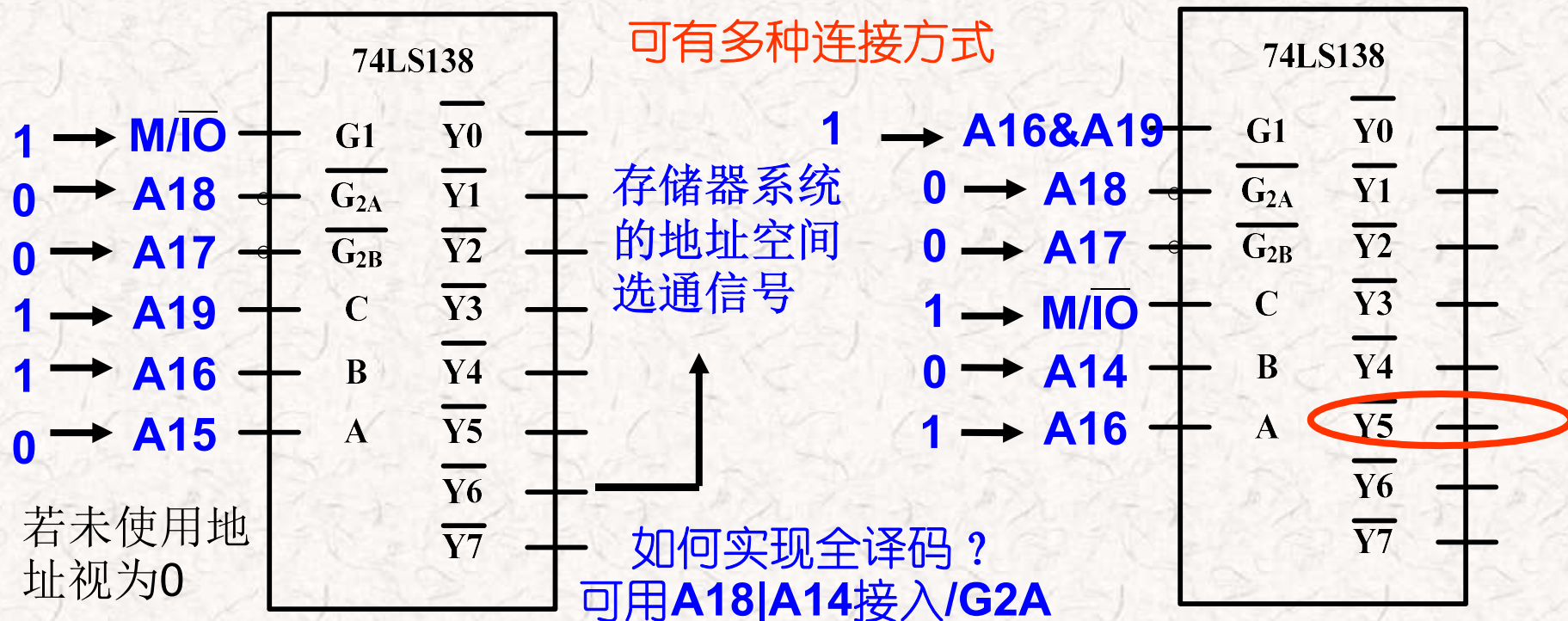


方法2：使用138译码器进行译码

需要当：M/I/O A19 A18 A17 A16 A15 A14 输出=0

1 1 0 0 1 0 0

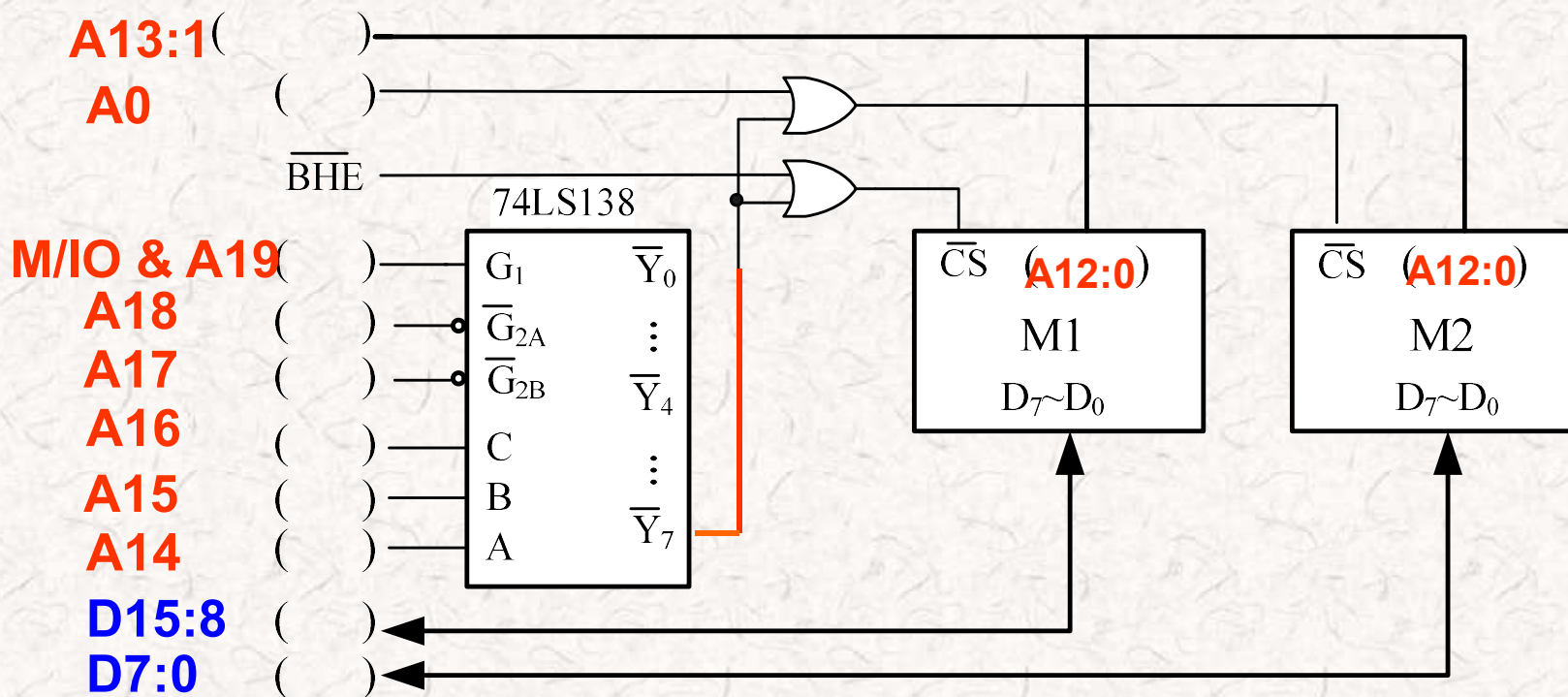
此方案缺点？



全译码练习

提示: A19 A18 A17 A16 A15 A14 A13 A12 A11 ~ A1 A0
 1 0 0 1 1 1 0 0 0 0 0 0

要求: 分配给M1和M2的地址空间从9C000H开始分配, 请填空, 并连接74LS138输出线, 要求全译码



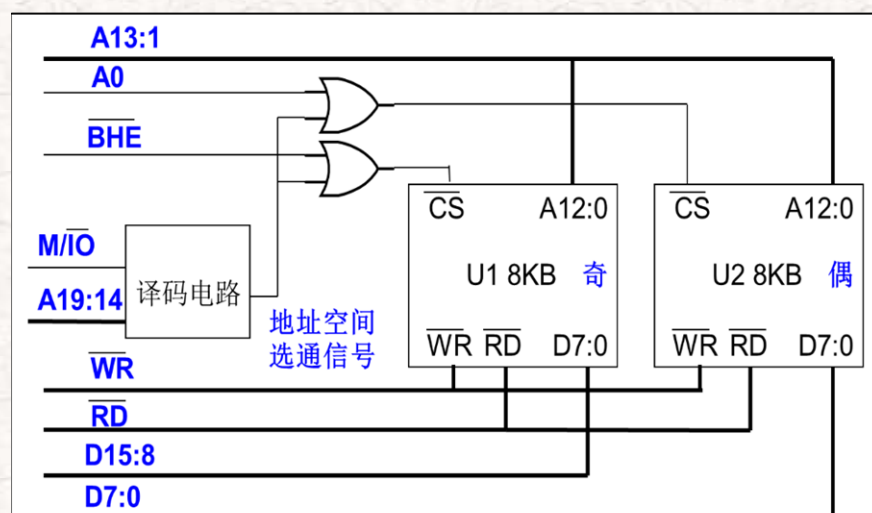
把指令和硬件动作联系起来

- 建立系统的概念：什么时候会产生上述信号去选通存储器？
- 对相应存储器/IO地址进行读/写的时候（如果不考虑取指操作）
- MOV AX, 9000H
- MOV DS, AX
- MOV BX, 3FFEh
- MOV AX, [BX]

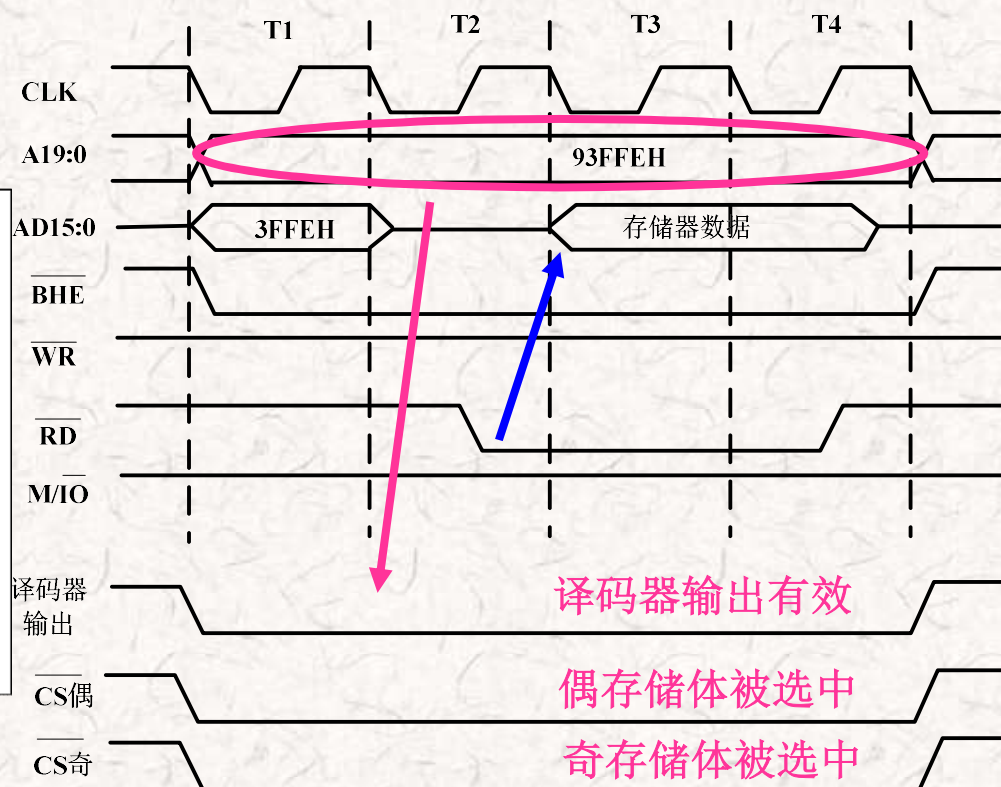
物理地址：9000H*10H+3FFEh=93FFEh

把指令和硬件动作联系起来

- 8086执行MOV AX, [BX]时, BIU产生总线周期, T1~T4总线上输出以下信号: **A19:0=93FFE_H**, **/RD=0**, **/BHE=0**, **M/IO=1**



D7:0 偶存储体中物理地址为**93FFE_H**的单元内容
D15:8 奇存储器中物理地址为**93FFF_H**的单元内容

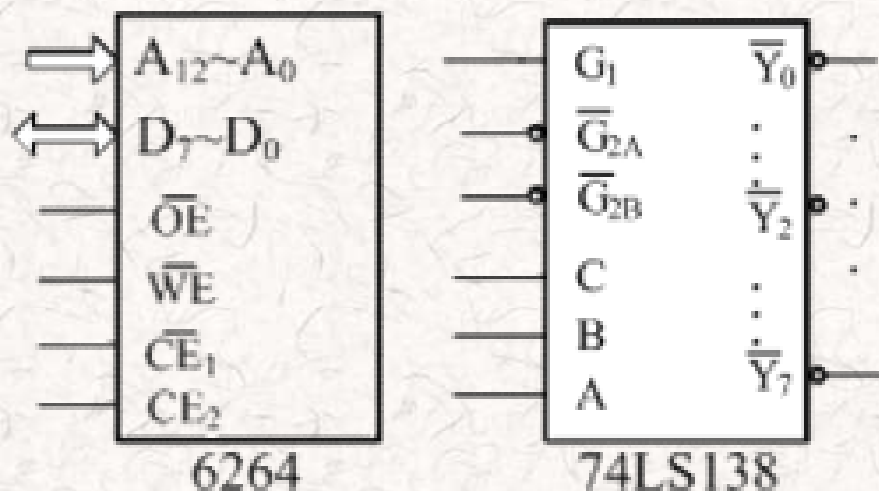


字访问, **A0=0**, **/BHE=0**

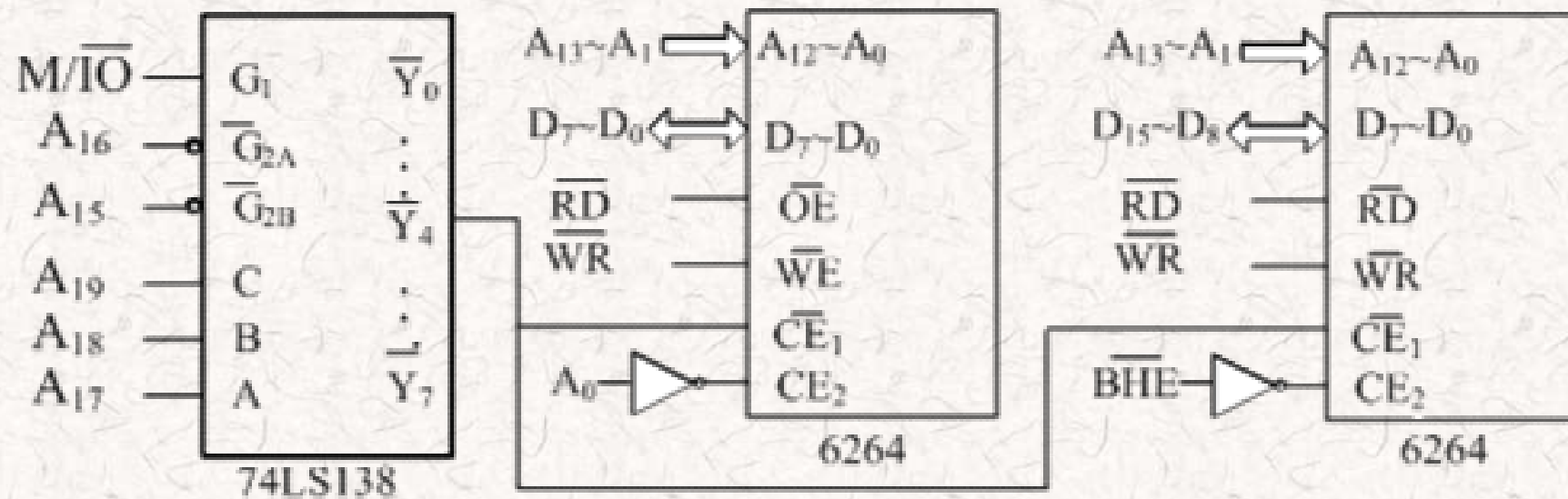
使用具有双片选信号的6264构造 存储系统

- 已知SRAM芯片6264与译码器74LS138功能图如下, 为8086设计16KB的存储系统, 且起始地址为80000H, 画出系统连接图 (可采用部分译码)

每个存储体8K*bit,
共需要2片



参考设计



两个片选信号必须同时有效方可选中该存储器

存储器读图与编程

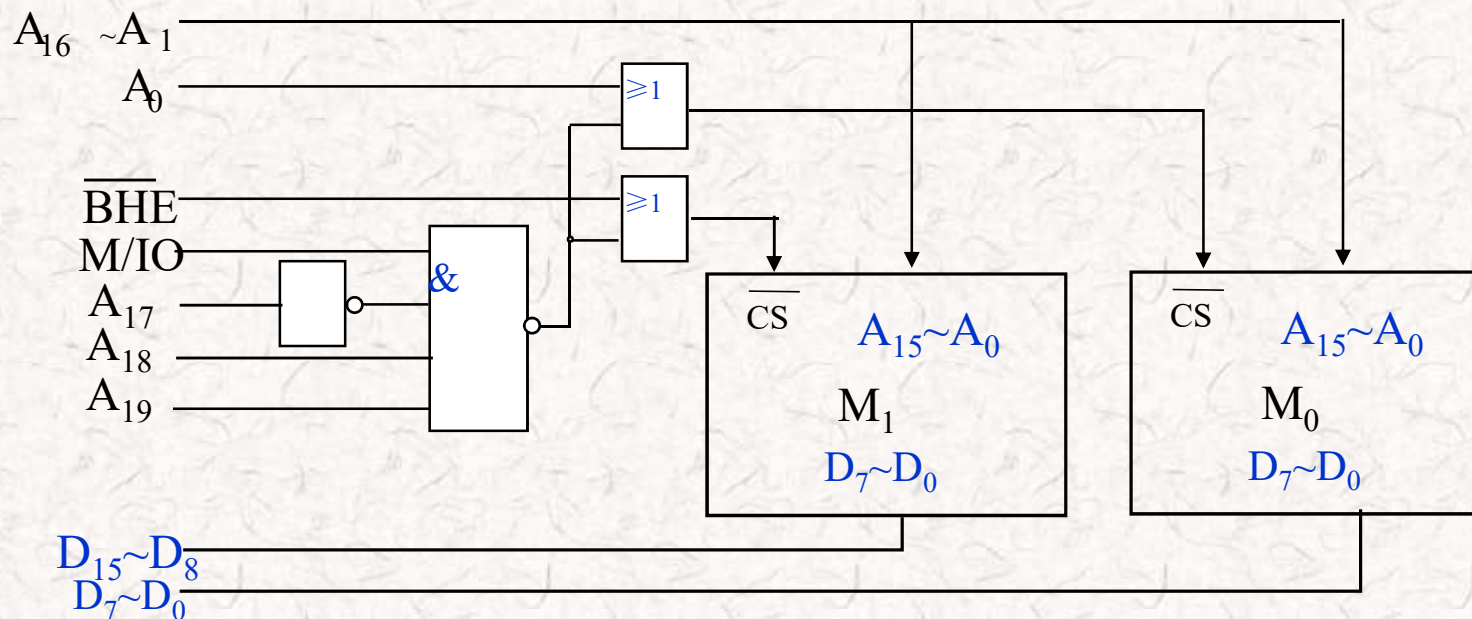
M1: 0C0000~0DFFFFH 的奇地址

M0: 0C0000~0DFFFFH 的偶地址

总容量: 128KB

下图为8086存储器的部分电路连线图。问:

- (1) 存储总容量为多少?
- (2) M0和M1的占据的地址空间分别是什么?
- (3) 编程将从起始地址开始的1KB存储空间赋值为'*'



参考代码

根据题目的要求，需要填充0C0000H~0C03FFH字节 '*'

通过段寄存器和偏移地址联合操作

数据段: DS-->0C000H

偏移地址: DI (或者SI、BX) 从0开始 可以用BP吗?

```
INIT_MM:MOV AX, 0C000H    LOP1:MOV BYTE PTR[SI], '*'
        MOV DS, AX        INC BP
        MOV SI, 0         DEC CX
        MOV CX, 1024      JNZ LOP1
```


代码优化

寄存器寻址快于立即数寻址

INIT_MM:MOV AX, 0C000H

MOV DS, AX

MOV SI, 0

MOV AL, '*'

MOV CX, 1024

LOP1:MOV [SI], AL

INC BP

DEC CX

JNZ LOP1

继续优化

```
INIT_MM:MOV AX, 0C000H
        MOV DS, AX
        MOV BP, 0
        MOV AX, '**'
        MOV CX, 512
```

```
INIT_MM:MOV AX, 0C000H
        MOV DS, AX
        MOV BP, 1024
        MOV AX, '**'
```

```
LOP1:MOV DS:[BP], AX
      INC BP
      INC BP
      DEC CX
      JNZ LOP1
```

将DEC放在MOV指令下面
可以吗?

```
LOP1:DEC BP
      DEC BP
      MOV DS:[BP], AX
      JNZ LOP1
```

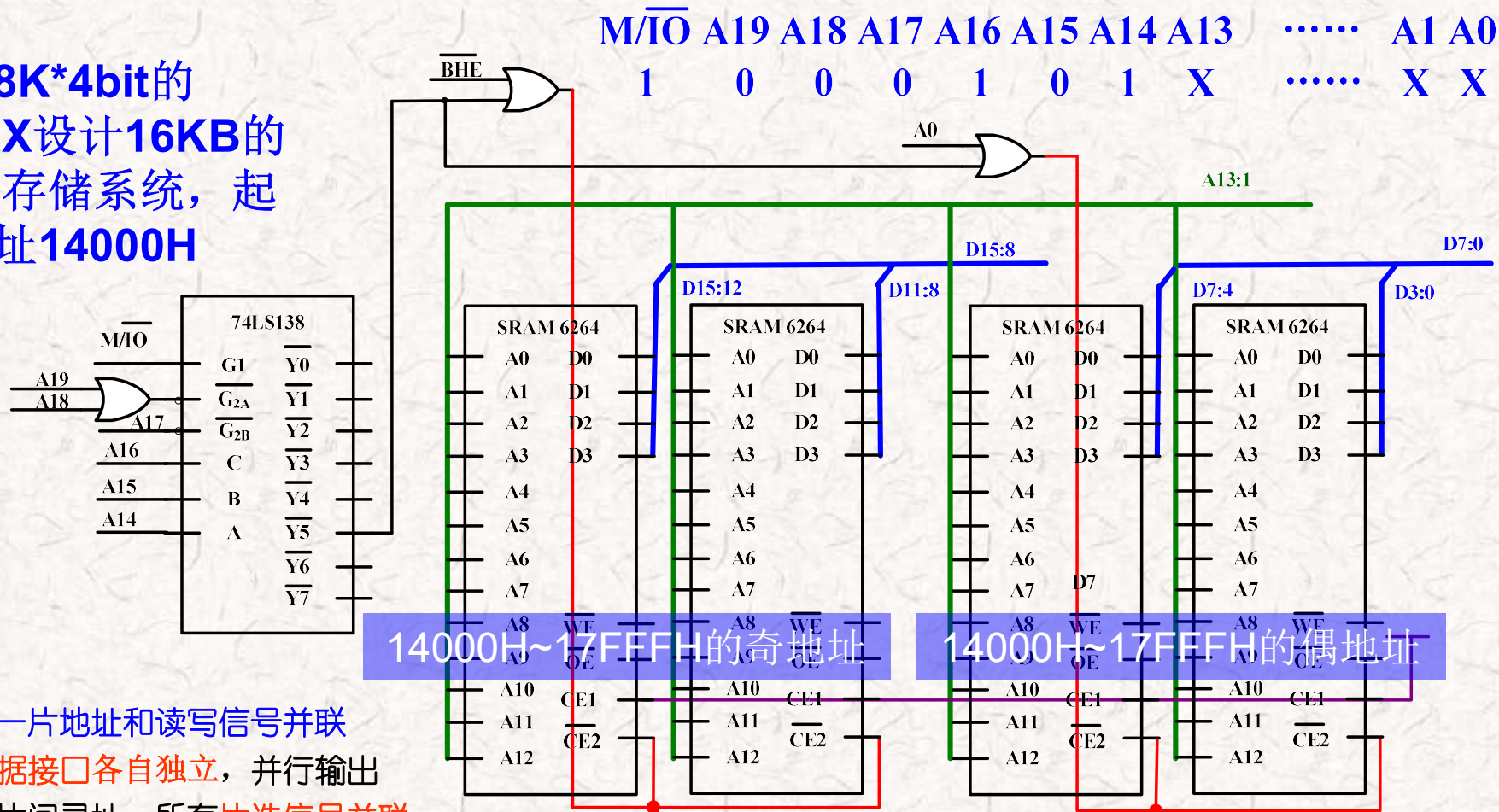
需要位扩展的8086存储系统举例

- 为8086设计16KB存储系统，占据14000H开始的连续地址空间，采用6232X SRAM（假想的）芯片如何实现？
- 假设：6232X容量为8K*4bit
- 分析
- (1) 要求每个存储体容量_____？ 8K*8bit
- (2) 各需要___片6232X？ 2片
- (3) 如何设计该存储系统？

在每个存储体内部进行位扩展，增加选通存储单元时能够得到的数据位宽

需要位扩展的8086存储系统举例

使用8K*4bit的
6232X设计16KB的
8086存储系统，起
始地址14000H



位扩展:

- (1) 每一片地址和读写信号并联
- (2) 数据接口各自独立，并行输出
- (3) 无片间寻址，所有片选信号并联

需要字扩展的8086存储系统举例

- 为8086设计32KB存储系统，占据90000H开始的连续地址空间，采用6264 SRAM芯片如何实现？
- 说明：6264容量为8K*8bit
- 分析
- (1) 要求每个存储体容量_____？ 16K*8bit
- (2) 各需要___片6264？ 2片
- (3) 如何设计该存储系统？

在每个存储体内部进行字扩展，增加每个存储体包含的存储单元的数量

需要字扩展的8086存储系统举例

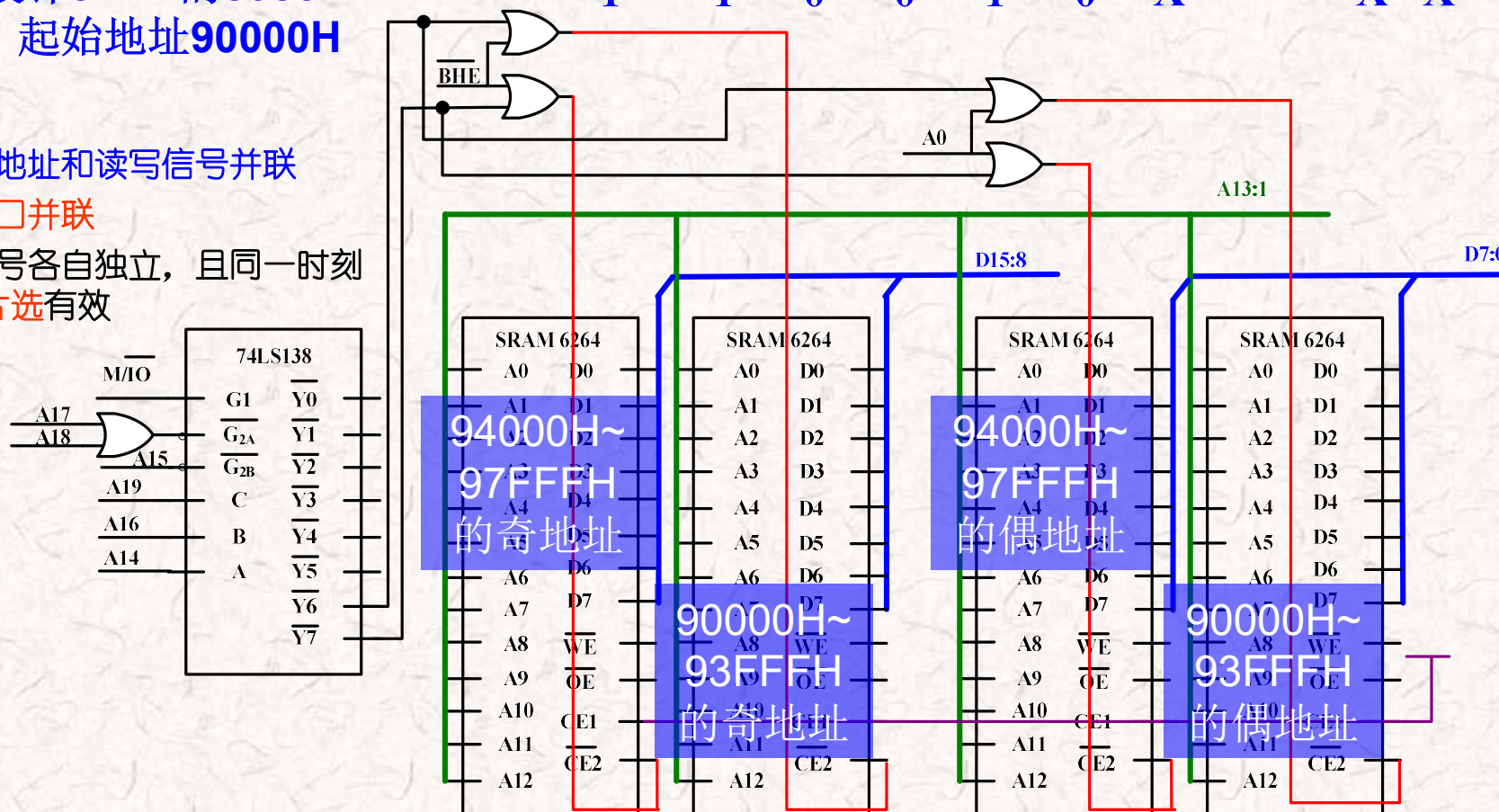
M/I \bar{O} A19 A18 A17 A16 A15 A14 A1 A0

1 1 0 0 1 0 X X X

使用6264设计32KB的8086
存储系统，起始地址90000H

字扩展:

- (1) 每一片地址和读写信号并联
- (2) 数据接口并联
- (3) 片选信号各自独立，且同一时刻只允许一个片选有效



需要字和位扩展的8086存储系统

- 假设已经生成了8086的20根地址总线和16根数据总线，
- (1) 使用6232x (8K*4bit) 芯片为8086设计32KB存储系统，占据20000H开始的存储空间；
- (2) 说明以下哪些指令会访问存储器
- (3) 该指令执行时，总线周期T1~T4期间总线上出现的波形是怎样的？哪些存储器片选有效？
- **MOV AX,2000H**
- **MOV DS,AX**
- **MOV DI, 1110H**
- **MOV CL,55H**
- **MOV 1[DI],CL**

设计思路

- 每个存储体容量：**16KB**，需要**4**片存储器
- 两片位扩展，得到 **4K*8bit** (一个存储模块)
- 两个存储模块进行字扩展，得到 **8K*8bit** (单个存储体)
- 生成地址译码信号
- 按照奇偶存储体连接原则构成**8086**指定地址空间的存储系统

参考电路

位扩展：

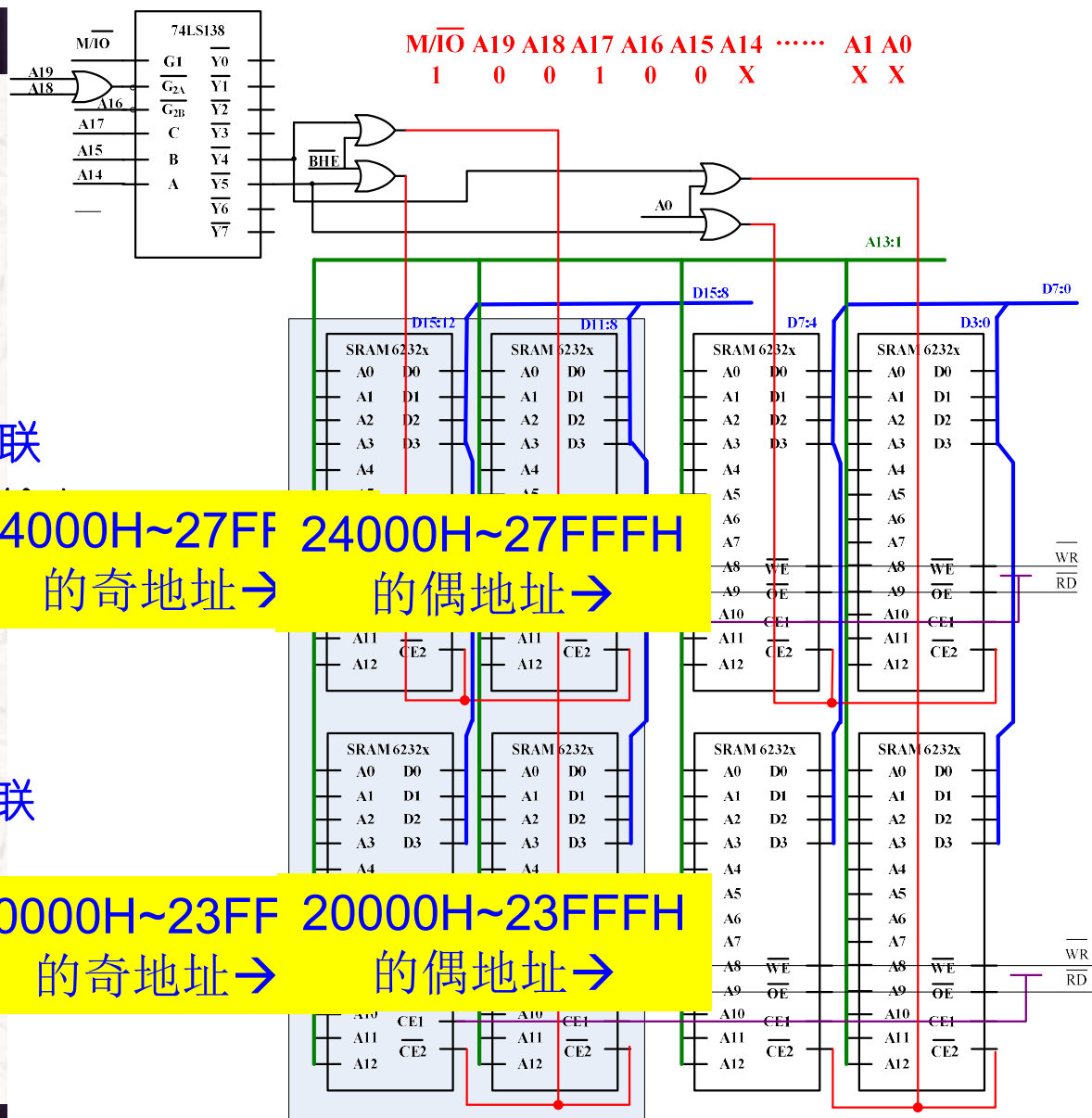
- (1) 每一片地址和读写信号并联
- (2) 数据总线各自独立，并
- (3) 无片间寻址，所有片选

24000H~27FF 24000H~27FFFH
的奇地址→ 的偶地址→

字扩展：

- (1) 每一片地址和读写信号并联
- (2) 数据总线并联
- (3) 片选信号各自独立，且时刻只允许一个片选有效

20000H~23FF 20000H~23FFFH
的奇地址→ 的偶地址→



总线波形

MOV AX,2000H

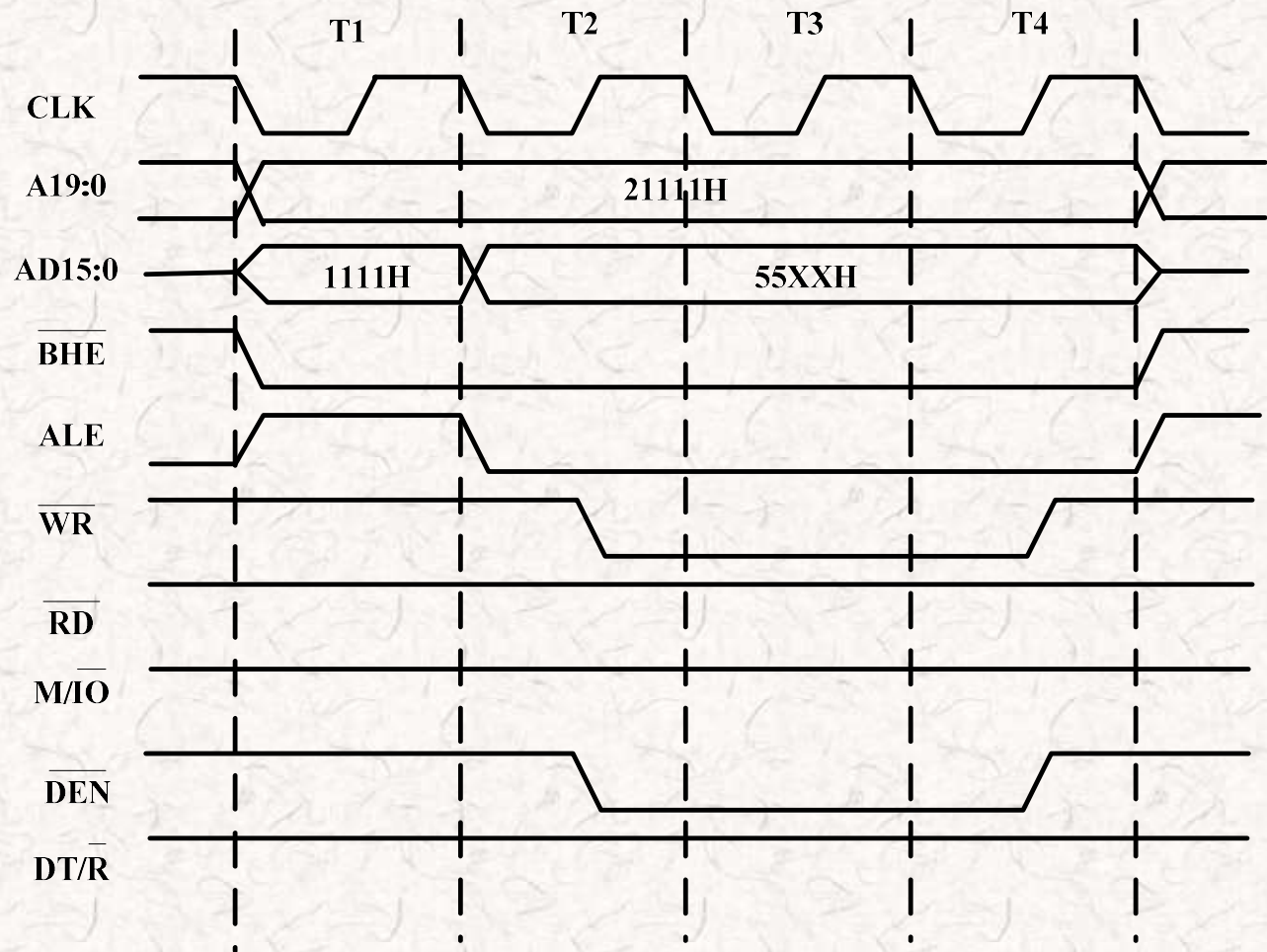
MOV DS,AX

MOV DI, 1110H

MOV CL,55H

MOV 1[DI],CL

写奇存储体的较低地址存储器，
即左边下面的两块存储器

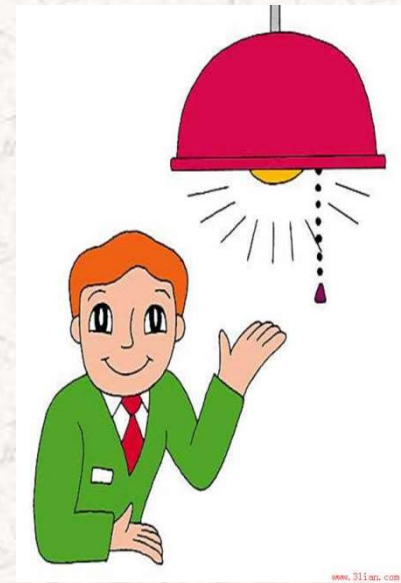


8086存储器设计电路连接练习

- 采用8K*4bit的SRAM器件和74ls138器件为8086构造48KB的存储系统，并占据A0000H开始的连续地址空间，要求不浪费任何地址空间。
- 需要完成器件实例化和连线，实现存储器的位扩展，字扩展和地址分配；
- 运行代码结束ok_start和ok_end应为高电平（仅中间为高不正确）；

作业

- P192~193
- 1~3
- 12~13
- 15~18



预告-第六章需要掌握的内容

- 6.1.1~6.1.3 掌握
- 6.2 8255方式0
- 6.3.1~6.3.2 8255的应用
- 其它：不要求