



北京航空航天大学
BEIHANG UNIVERSITY

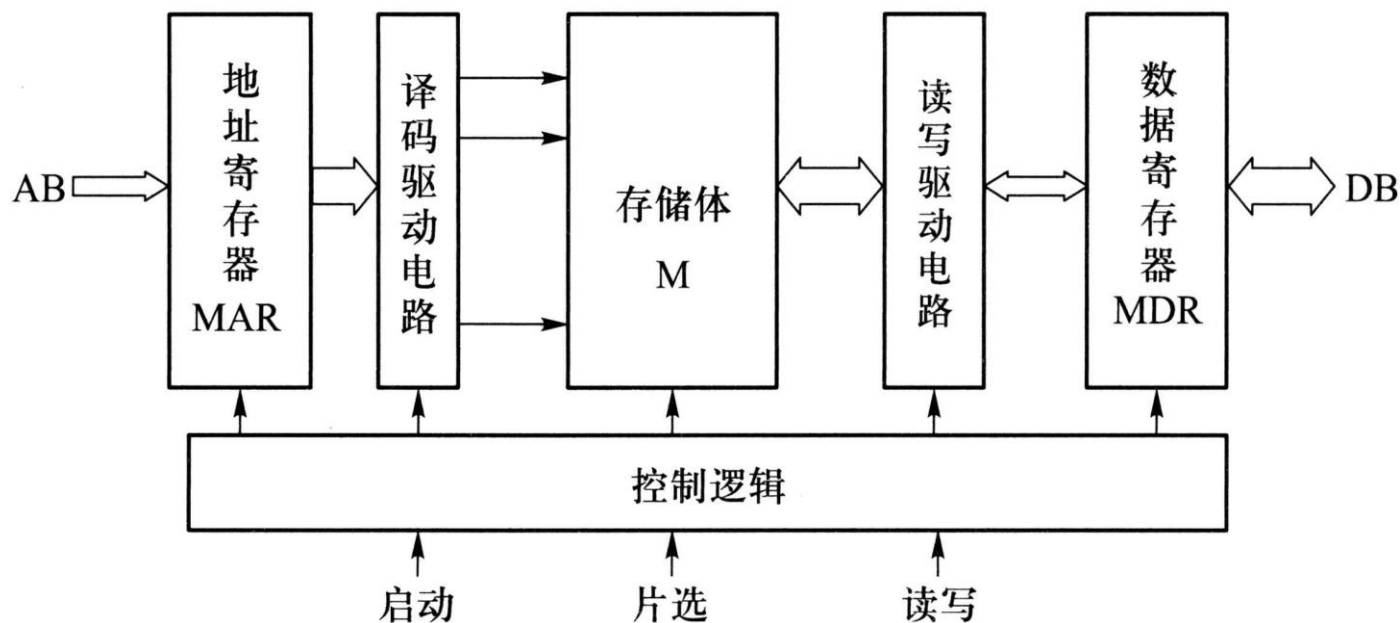
微机原理与接口技术

第14讲 存储器 (2)



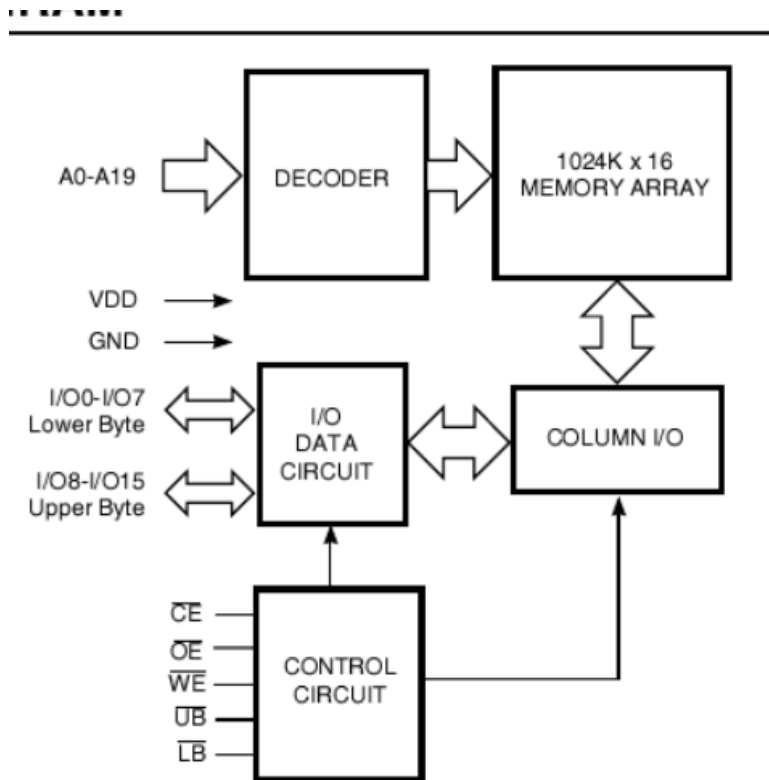
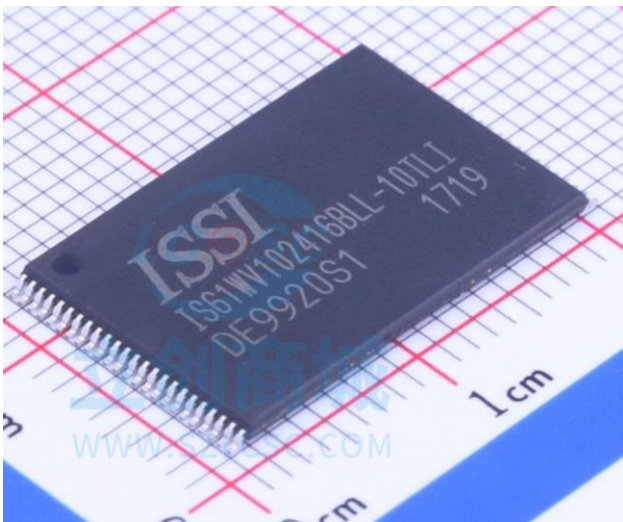
半导体存储器的结构

- 6个部分



半导体存储器的读写时序

- 主要控制信号：CS、OE、WE



半导体存储器的读写时序

- 特性参数

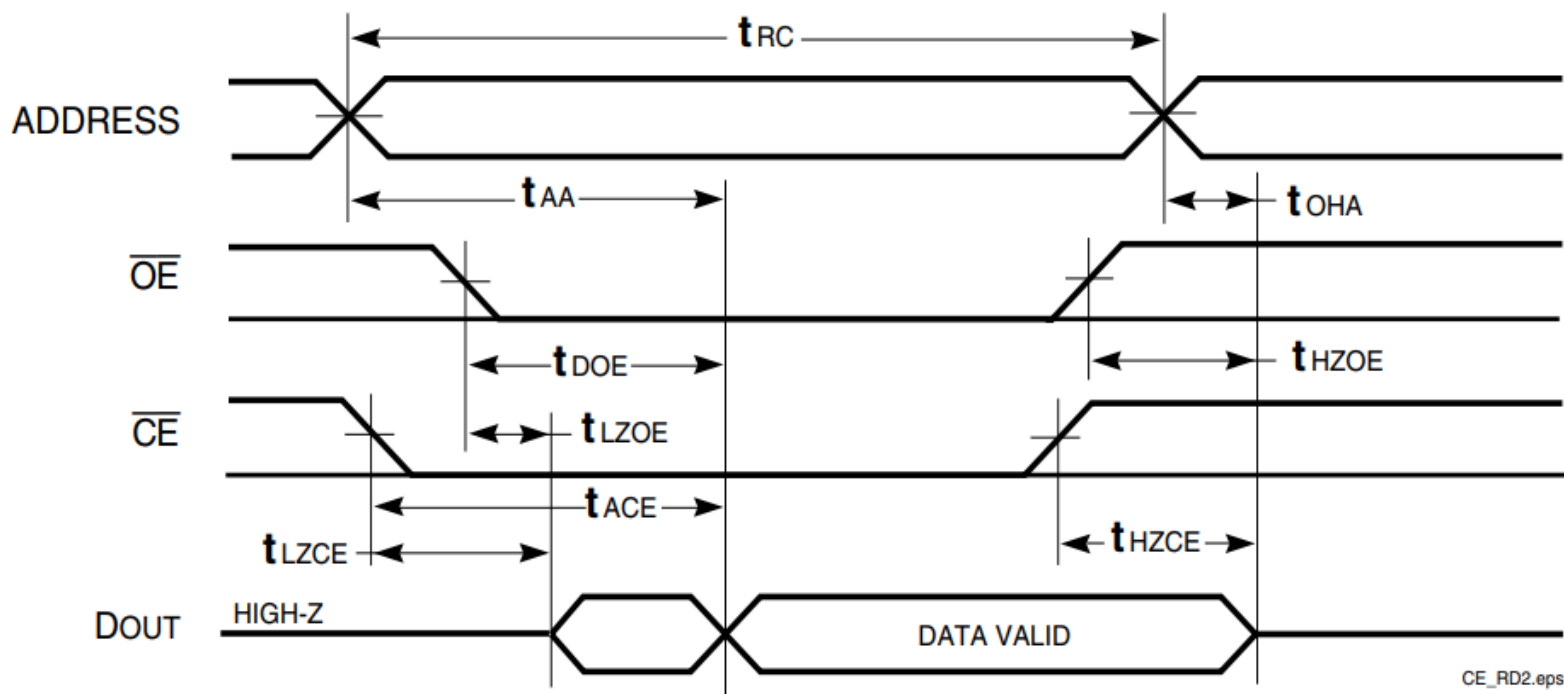


READ CYCLE SWITCHING CHARACTERISTICS⁽¹⁾ (Over Operating Range)

Symbol	Parameter	-8		-10		Unit
		Min.	Max.	Min.	Max.	
t _{RC}	Read Cycle Time	8	—	10	—	ns
t _{AA}	Address Access Time	—	8	—	10	ns
t _{OH} A	Output Hold Time	2.5	—	2.5	—	ns
t _{ACE}	$\overline{\text{CE}}$ Access Time	—	8	—	10	ns
t _{DOE}	$\overline{\text{OE}}$ Access Time	—	5.5	—	6.5	ns
t _{HZOE} ⁽²⁾	$\overline{\text{OE}}$ to High-Z Output	—	3	—	4	ns
t _{LZOE} ⁽²⁾	$\overline{\text{OE}}$ to Low-Z Output	0	—	0	—	ns
t _{HZCE} ⁽²⁾	$\overline{\text{CE}}$ to High-Z Output	0	3	0	4	ns
t _{LZCE} ⁽²⁾	$\overline{\text{CE}}$ to Low-Z Output	3	—	3	—	ns
t _{BA}	$\overline{\text{LB}}, \overline{\text{UB}}$ Access Time	—	5.5	—	6.5	ns
t _{HZB} ⁽²⁾	$\overline{\text{LB}}, \overline{\text{UB}}$ to High-Z Output	0	3	0	3	ns
t _{LZB} ⁽²⁾	$\overline{\text{LB}}, \overline{\text{UB}}$ to Low-Z Output	0	—	0	—	ns
t _{PU}	Power Up Time	0	—	0	—	ns
t _{PD}	Power Down Time	—	8	—	10	ns

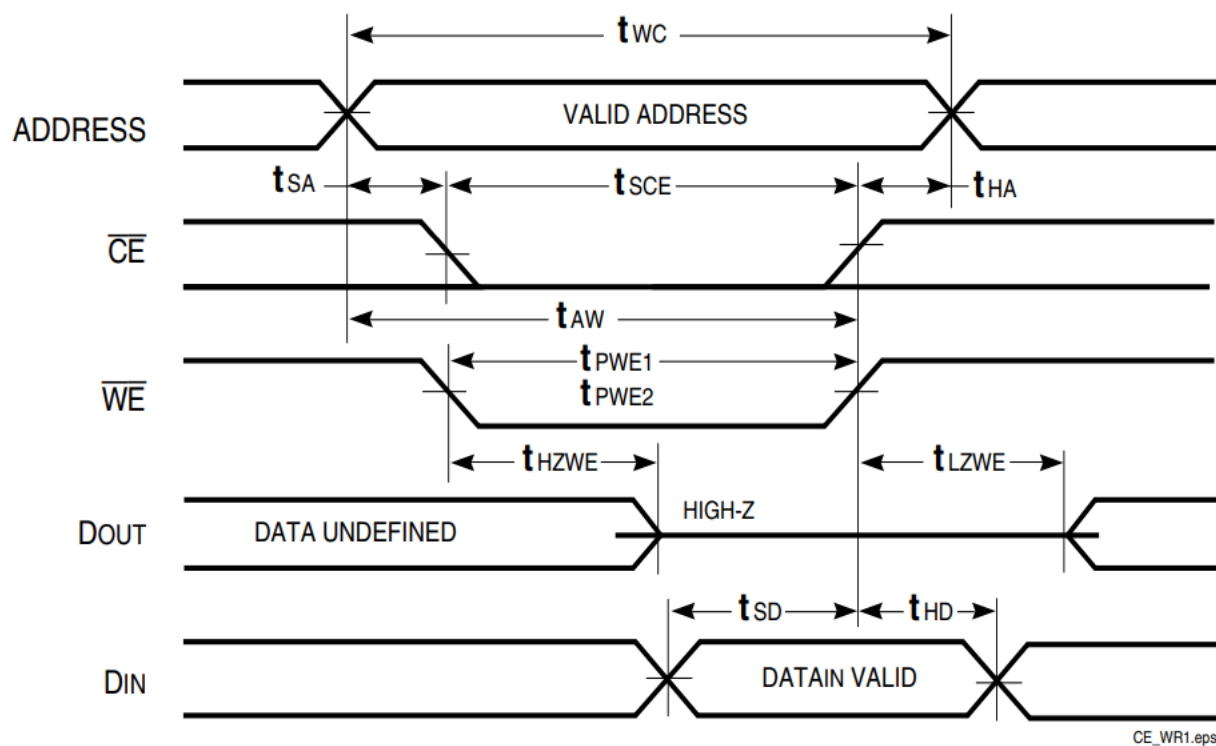
半导体存储器的读写时序

- 数据读出



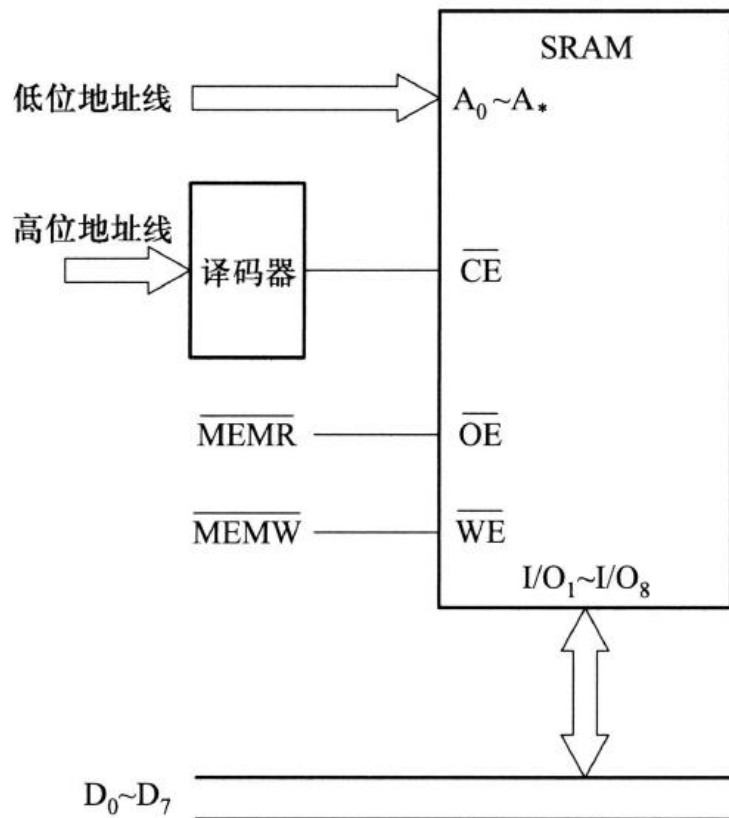
半导体存储器的读写时序

• 数据写入



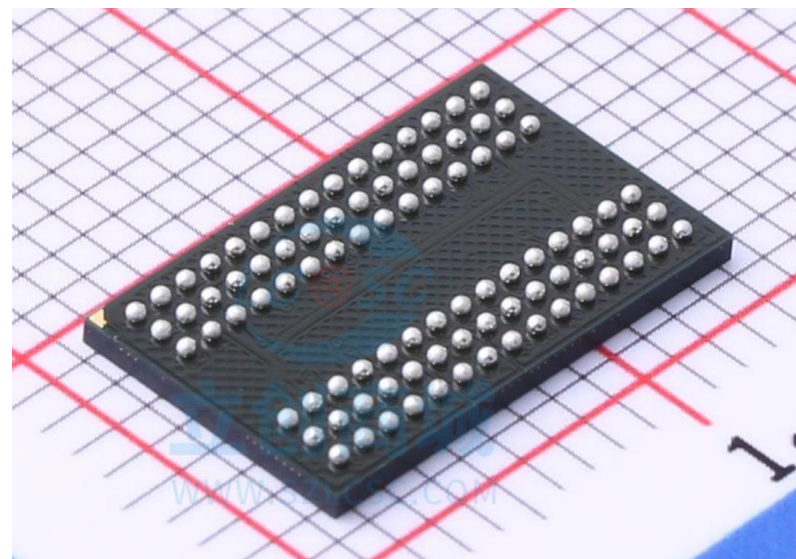
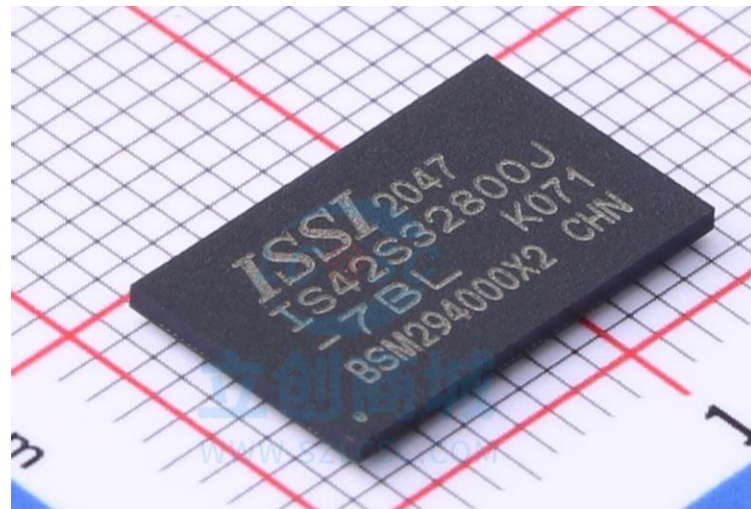
半导体存储器的读写时序

- 如何与CPU连接???



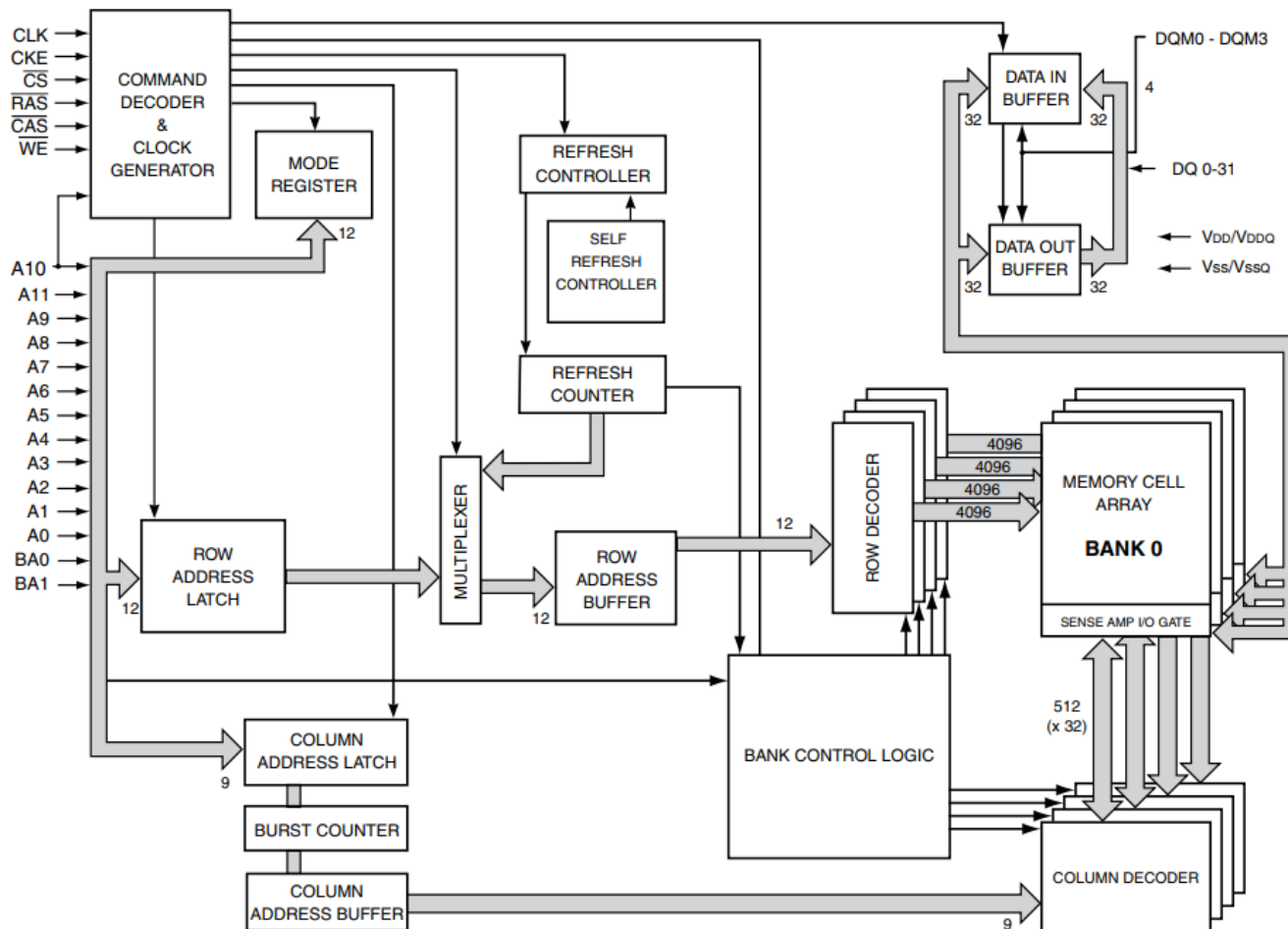
• IS42S32800

安装类型	表面贴装型
技术	SDRAM
存储器格式	DRAM
时钟频率	143MHz
存储容量	256Mb (8M x 32)
访问时间	5.4ns
封装/外壳	90-TFBGA
工作温度	0°C ~ 70°C (TA)
供应商器件封装	90-TFBGA (8x13)
写周期时间-字, 页	-
存储器类型	易失
存储器接口	并联
电压-供电	3V ~ 3.6V



• IS42S32800

FUNCTIONAL BLOCK DIAGRAM (FOR 2Mx32x4 BANKS)





课程内容



第一节 概述

第二节 ROM存储器

第三节 RAM存储器

第四节 FLASH

第五节 高速缓冲存储技术

第六节 存储器的接口设计



第二节 ROM存储器

Read-Only Memory (ROM):

Memory content fixed and cannot be changed (easily)

- **特点：**非易失性存储器，造价比RAM低，集成度高，组成结构比RAM简单。
 - **用途：**存放软件；存放微程序；存放特殊编码
- 掩模式只读存储器
 - 一次编程只读存储器 (PROM)
 - 多次编程只读存储器 (EPROM, EEPROM)



第二节 ROM存储器

1. 掩模式ROM(MROM)

厂家制造芯片时把数据用光刻掩模写入芯片，不能改。

特点：可靠性高，集成度高，批量生产价格便宜，但用户对厂家依赖性大，灵活性差。

存储单元可用二极管、双极型晶体管和MOS三极管作为耦合元件。



第二节 ROM存储器



1.掩模式ROM(MROM)

依用户要求，由厂家按确定工序，用是否制造元器件的方法，实现信息写入。制造完成后，不能再改变的只读存储器

■ MROM的特点

- 结构简单、集成度高、价格便宜
- 宜于标准化和大批量生产
- 使用不灵活、一经生产为成品，则无法修改
- 读出时间：10~500ns
- 最大集成度：32Mb/片

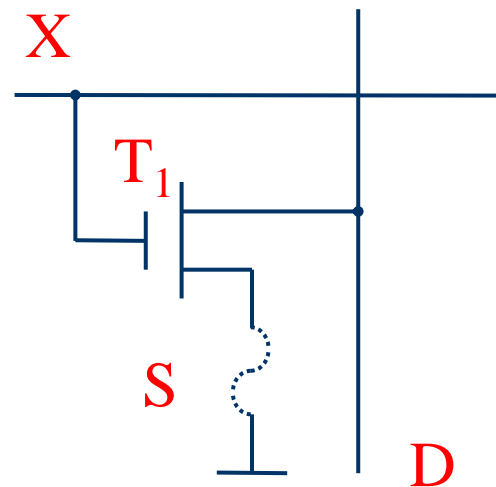


第二节 ROM存储器

2、熔丝式ROM(PROM)

有时也称为可编程只读存储器(PROM)。

制造厂家生产的PROM在出厂时，各个单元都处于相同状态，用户根据需要在特殊条件下编程(专用的设备上)写入自己需要的信息，**且是一次性的**，也就是软件灌入后，就无法修改了。PROM类似于掩膜ROM，适合于批量使用。





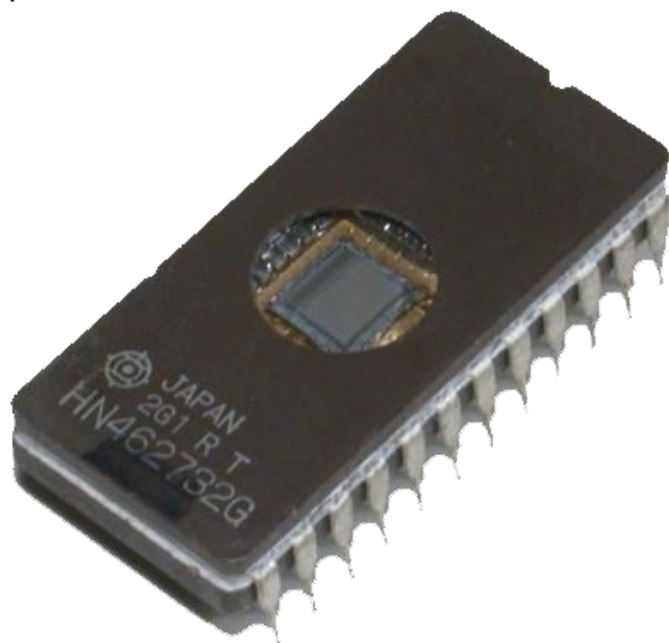
第二节 ROM存储器



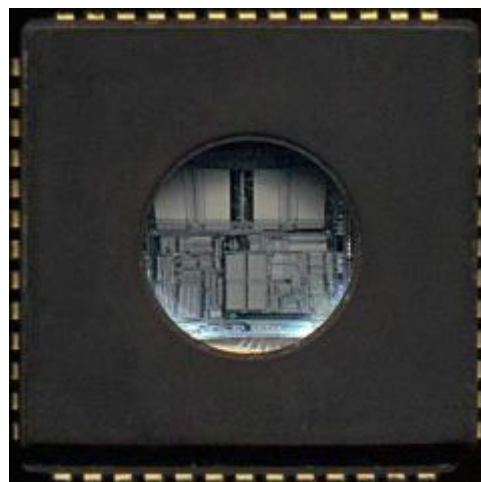
(3) 可擦除的EPROM

可擦除PROM简称EPROM(Erasable Programable ROM)。这种存储器可由用户按规定的方法多次编程，如编程之后想修改，可用紫外线灯制作的擦除器照射7~30分钟左右(新的芯片擦除时间短，多次擦除过的芯片擦除时间长)，使存储器复原，再编程。

相比PROM而言，EPROM可以经过紫外光的照射擦除原先的程序，根据需要重写且能改写多次。在修改时，需要额外的条件，写的速度慢。即使要改写其中已经写入的一位，也必须把整个内容全部擦去。



高压写入



紫外线光照擦除



第二节 ROM存储器

(4) 电可擦除的PROM

简称为 **EEPROM** 或 **E²PROM**(Electrically Erasable PROM):

能以字节为单位擦除和改写，在用户系统中可以直接操作，且不需要把芯片拔下来，直接可以通过电子擦除原数据，并写入新代码。价格很高，写入时间很长，写入很慢。

随着技术的进步，E²PROM的擦写速度不断加快，可作为**非易失性RAM**使用。



课程内容



第一节 概述

第二节 ROM存储器

第三节 RAM存储器

第四节 FLASH

第五节 高速缓冲存储技术

第六节 存储器的接口设计



第三节 RAM存储器

随机读写存储器(RAM)

这种存储器在使用过程中既可利用程序随时写入信息，又可随时读出信息。它分为双极型和MOS型两种，前者读写速度高，但功耗大，集成度低，故在微型机中几乎都用后者。RAM可分为三类。

1) 静态RAM

静态RAM即SRAM(Static RAM)，其存储电路以双稳态触发器为基础，状态稳定，只要不掉电，信息不会丢失。优点是不需刷新，缺点是集成度低。它适用于不需要大存储容量的微型计算机(例如，单板机和单片机)中。

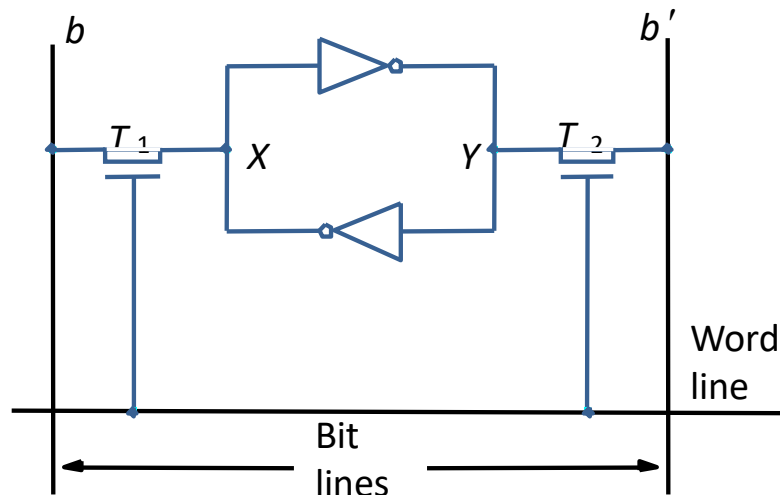
1) SRAM (静态RAM)

1. 静态RAM的构成

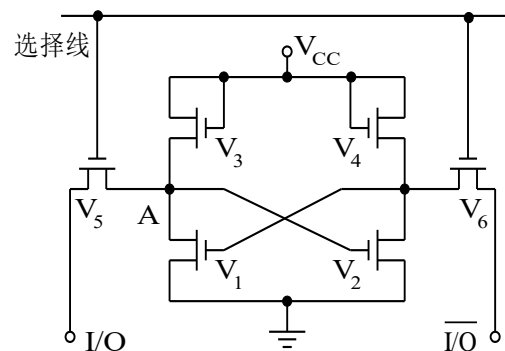
由地址译码器，存储矩阵，控制逻辑和三态数据缓冲器组成。

基本存储单元所包含的管子数目较多，且功耗也较大。

适合在小容量存储器中使用。



静态随机存储单元电路



六个NMOS管组成的静态RAM存储电路



1) SRAM (静态RAM)

- a. 由6个MOS管构成的触发器作为基本存储电路，集成度介于双极型和动态RAM之间。
- b. 无需刷新，故可省去刷新电路，功耗比双极型低，但比动态RAM高。
- c. 可以用电池做后备电源，因而不需刷新逻辑电路(RAM中一个最大的问题就是：一旦RAM掉电，其存储的信息便会丢失。这就要求当交流电源掉电时，能够自动切换到一个用电池供电的低压后备电源，以此来保持RAM中的信息)
- d. 较高的集成度。



2) 动态RAM

2) 动态RAM

动态RAM即DRAM(Dynamic RAM)，其存储单元以电容为基础，电路简单，集成度高。但也存在问题，即电容中电荷由于漏电会逐渐丢失，因此DRAM需定时刷新。它适用于大存储容量的计算机。

- a. 基本存储电路由单管线路组成(靠电容存储电荷);
- b. 需要刷新电路，典型要求是每隔2毫秒刷新一遍;
- c. 较高的集成度，比SRAM的集成度高;



2) 动态RAM

DRAM 刷新相关概念(1)

- DRAM靠电容电荷存储信息。电容电荷容易泄漏，需定期补充电荷以保持信息不变，补充电荷的过程称为刷新过程
- 泄漏完毕之前如不能补充电荷，存储信息发生丢失，信息存储到信息泄漏完毕之间必须完成刷新过程，称为最大刷新周期，
- 从上一次对存储器刷新结束到下一次对整个存储器刷新结束所需要的时间称为刷新周期，刷新一块芯片所需的刷新周期数由芯片矩阵的行数决定。



2) 动态RAM



DRAM 刷新相关概念(2)

- DRAM在原理上是通过对电容充、放电来存入信息的，会因漏电而导致存储的信息随时间的推移逐渐减弱直至消失，因此需要定时给电容进行充电以保持当前存储的信息。这种周期性的刷新操作称为**动态刷新**。
- 与SRAM相比，DRAM芯片的内部增加了**动态刷新电路**。
- DRAM 的集成度高，容量通常都比较大，为了减少芯片引脚，DRAM芯片采用**时分复用**的方式传输地址。

2) 动态RAM

自DRAM问世以来，设计就发生了许多变化。下图显示了基本DRAM体系结构从时钟到异步到快速页面模式（FPM）到扩展数据输出（EDO）到突发模式EDO（BEDO）到同步（SDRAM）到DDR SDRAM的演变。这些变化本质上是结构性的，在实现成本方面相对较小，并且显著提高了DRAM吞吐量。

DDR SDRAM，是目前电脑中用得最多的内存。

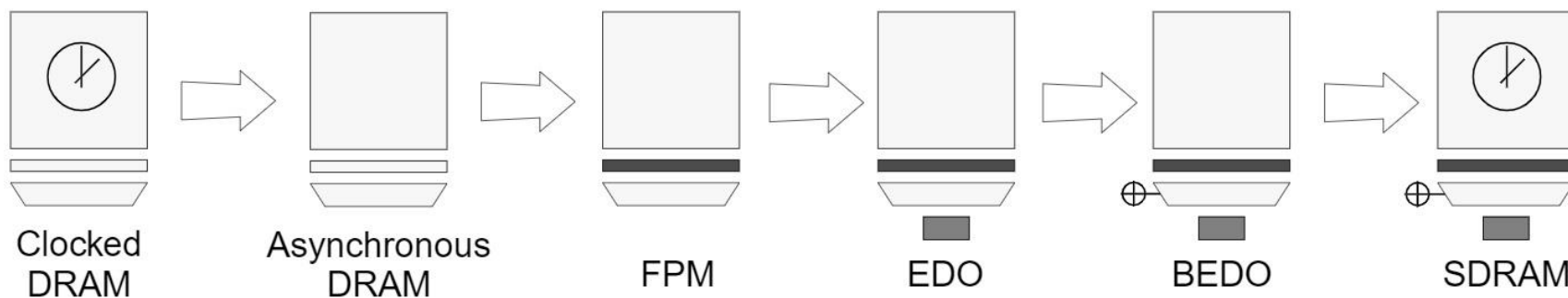


Figure 7: Evolution of the DRAM Architecture

Each step along DRAM's evolution has been incremental. Original designs were clocked; in the mid-1970's the clock disappeared; fast page mode (FPM) kept the sense amplifiers active; extended data-out (EDO) added a latch; burst EDO (BEDO) added an internal counter; SDRAM carried full circle by reinstating a clock signal.

动态 RAM 和静态 RAM 的比较

主存

DRAM

SRAM

缓存

存储原理

电容

触发器

集成度

高

低

芯片引脚

少

多

功耗

小

大

价格

低

高

速度

慢

快

刷新

有

无



3) 非易失RAM

非易失RAM或称掉电自保护RAM，即NVRAM(Non Volative RAM)，这种RAM是由SRAM和EEPROM共同构成的存储器，正常运行时和SRAM一样，而在掉电或电源有故障的瞬间，它把SRAM的信息保存在EEPROM中，从而使信息不会丢失。NVRAM多用于存储非常重要的信息和掉电保护。

其他新型存储器还有很多，如快擦写ROM(即Flash ROM)以及Integrated RAM，它们已得到应用，详细内容请参阅存储器数据手册。



第三节 RAM存储器



注：

1. SRAM速度非常快，是目前读写最快的存储设备了，但是它也非常昂贵，所以只在要求很苛刻的地方使用，譬如CPU的一级缓冲，二级缓冲。
2. DRAM保留数据的时间很短，速度也比SRAM慢，不过它还是比任何的ROM都要快，但从价格上来说DRAM相比SRAM要便宜很多，计算机内存就是DRAM的。



课程内容



第一节 概述

第二节 ROM存储器

第三节 RAM存储器

第四节 FLASH

第五节 高速缓冲存储技术

第六节 存储器的接口设计

第四节 FLASH

FLASH存储器又称**闪存**，它结合了ROM和RAM的长处，不仅具备电子可擦除可编程（EEPROM）的性能，还不会断电丢失数据同时可以快速读取数据（NVRAM的优势），

常见有**NOR**和**NAND**两类



桀冈富士雄
Fujio Masuoka

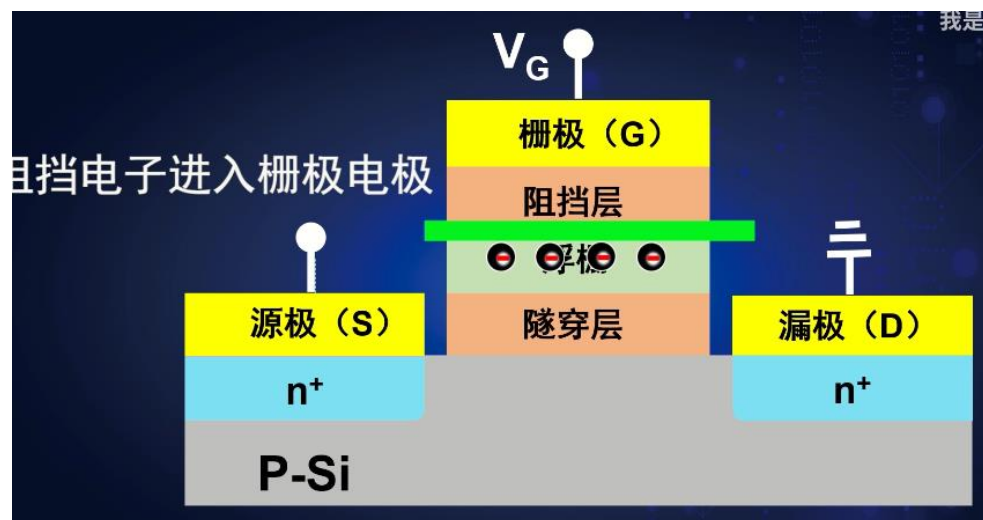
第四节 FLASH——NOR Flash

NOR Flash的读取和常见的SDRAM的读取是一样，用户可以直接运行装载在NOR FLASH里面的代码，这样可以减少SRAM的容量从而节约了成本。

一般小容量的用NOR Flash，因为其读取速度快，多用来存储操作系统等重要信息，通常用来存放固件（Firm Ware）、系统启动代码（BootCode）、PC的BIOS等。

NAND Flash强调的是降低每位成本，能够提供极高的单元密度，实现单个存储芯片的大容量存储。NAND Flash的读取以页为单位，所以存放在NAND 中的代码是不能直接运行的，需要读到RAM中然后再执行。

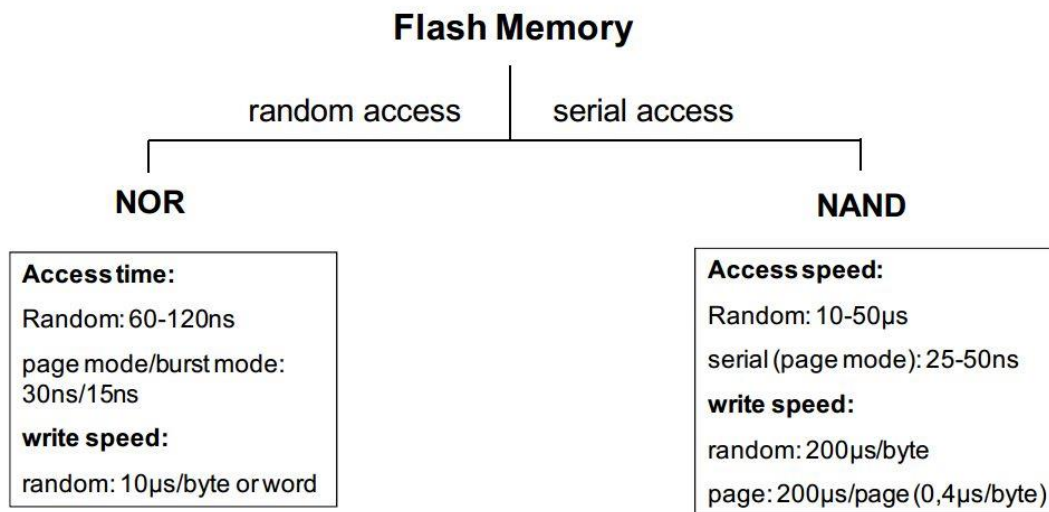
大多数SSD硬盘，U盘、存储卡、eMMC存储芯片都是用的NAND Flash。



第四节 FLASH

NOR Flash 和 NAND Flash比较

- NOR的读速度比NAND稍快一些。
- NAND的写入速度比NOR快很多。
- NAND的擦除速度远比NOR的快。
- 大多数写入操作需要先进行擦除操作。
- NAND的擦除单元更小，相应的擦除电路更少。





第四节 FLASH



Nand Flash 目前分以下4种颗粒：

SLC、MLC、TLC和QLC闪存芯片区别

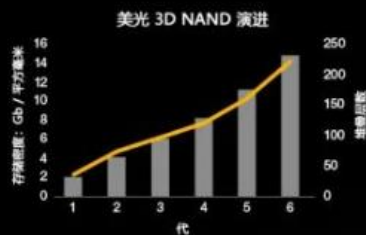
- SLC = Single-Level Cell，即1bit/cell，速度快寿命长，价格贵（约MLC 3倍以上的价格），约10万次擦写寿命；
- MLC = Multi-Level Cell，即2bits/cell，速度一般寿命一般，价格一般，约3000---10000次擦写寿命；
- TLC = Trinary-Level Cell，即3bits/cell，速度慢寿命短，价格便宜，约500-1000次擦写寿命；
- QLC= Quad-Level Cell，即4bits/cell，速度最慢，成本最低，寿命最短。

美光 232 层 NAND

为新一波端到端技术创新浪潮奠定基础

- 更多堆叠层数
- 更大存储密度 (每平方毫米比特数)
- 更快输入/输出速度

建立在美光经过验证的
闪耀业界 176 层 NAND 先进技术之上



注释:

*速度与耐用性能提升数据基于与上一代 176 层 NAND 对比

**封装尺寸数据与上一代相比: B47R VS B58R

©2022 Micron Technology, Inc. 保留所有权利。信息、产品和服务/规格如有变更,恕不另行通知。
Micron, Micron 徽标和其他 Micron 商标均为 Micron Technology Inc. (美光科技股份有限公司) 所属商标。
所有其他商标均为其各自所有者所有。



应用与服务

232 层 NAND
是数据密集型应用
与高要求存储应用
的理想之选



PC 客户端



移动设备



智能边缘



数据中心

美光 232 层
堆叠层数
和 6 平面架构
的优势

100%
写入带宽提升 *

>75%
读取带宽提升 *

50%
传输速率提高
达到每秒 2.4GB (ONFI总线) *

28%
封装尺寸减小 **



2020年, 美光发布了 176 层 3D NAND, 确立成为NAND 技术领域前沿厂商。如今凭借下一代 2



第一节 概述

第二节 ROM存储器

第三节 RAM存储器

第四节 FLASH

第五节 高速缓冲存储技术

第六节 存储器的接口设计

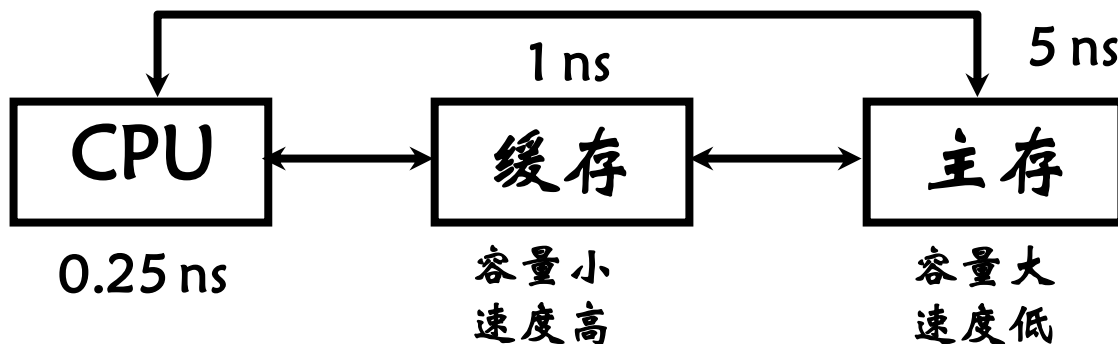


第五节 高速缓冲存储技术

1. 问题的提出

避免 CPU “空等” 现象

CPU 和主存 (DRAM) 的速度差异



程序访问的局部性原理



第五节 高速缓冲存储技术

- **CACHE的作用**：在原有的内存和**CPU**之间增加了更高速的**CACHE**，把当前正在执行的指令地址附近的一部分指令或当前正在使用的数据可以从内存中调入**CACHE**中，以提高系统速度；
- **引入CACHE的依据**：程序的局部性原理（**Locality of Reference**），也就是在某一段时间内频繁访问某一局部的存储器地址，而对此以外的地址则很少访问的现象；
- **命中率**：**CPU**访问**CACHE**，找到信息的百分比；高速缓存器的设计目标是使**CPU**访问尽可能在**CACHE**中进行；



第五节 高速缓冲存储技术

2. 解决方法--数据访问局部性(程序局部性)

□ **程序局部性**---程序仅需要访问内存很小一部分空间。

■ **空间局部性**:如果内存某个区域刚刚被访问, 那么不久的将来其相邻的区域很有可能被访问。

■ **时间局部性**:如果内存某个区域刚刚被访问, 那么不久的将来该区域可能会被重复访问。

□ **优化手段**

■ 预读优化空间局部性

■ 调度算法优化时间局部性

■ 将即将访问或频繁访问的数据调度到上层快速存储器



第五节 高速缓冲存储技术

程序局部性举例

□ 数据

- 数组元素访问 (空间)
- 结构体、数据库记录访问 (空间)
- 局部变量, 计数器, 指针等被重复使用. (时间)

□ 指令

- 顺序访问的指令 (空间)
- 重复使用的循环体 (时间)
- 子函数 (时间)

```
int a[M][N];  
for (i = 0; i < M; i++)  
    for (j = 0; j < N; j++)  
        sum += a[i][j];
```



第五节 高速缓冲存储技术

3、关键问题

- 如何判断一个数据在cache中
 - 数据查找 Data Identification
- 如需访问的数据在cache中，存放在什么地方
 - 地址映射 Address Mapping
- Cache满了以后如何处理
 - 替换策略 Placement Policy
- 如何保证cache与memory的一致性
 - 写入策略 Write Policy



第五节 高速缓冲存储技术

4、Cache命中率

- N_c 表示Cache完成存取访问的总次数
- N_m 表示主存完成存取访问的总次数
- Cache命中率 h

$$h = N_c / (N_c + N_m)$$

- t_c 表示命中Cache时的访问时间
- t_m 表示命中主存时的访问时间
- t_a 平均访问时间

$$t_a = ht_c + (1-h)t_m$$



第五节 高速缓冲存储技术

命中与未命中

缓存共有 C 块

主存共有 M 块 $M \gg C$

命中 主存块 **调入** 缓存

主存块与缓存块 **建立** 了对应关系

用**标记记录**与某缓存块建立了对应关系的**主存块块号**

未命中 主存块 **未调入** 缓存

主存块与缓存块 **未建立** 对应关系



第五节 高速缓冲存储技术

5.Cache的结构

Cache一般由两部分组成，一部分存放由主存储器来的数据，另一部分存放该数据在主存储器中的地址。(此部分称地址标记存储器，记为Tag)。

由关联性，高速缓冲存储器结构可分为：

- 全相联Cache – 保存数据块及其在主存中的地址。
- 直接映象Cache – 主存分页，Tag保存页号，索引字段保存页中偏移地址。
- 成组相联Cache – Cache分组，每组采用直接映像结构，组之间采用全相联结构。



第五节 高速缓冲存储技术

6.Cache的架构

1) 读取结构

➤ 旁视结构

Cache和主存并行地连接到系统总线，同时监视总线周期。当CPU发出数据请求时，Cache如果命中，响应并终止总线周期。如果没命中，主存响应，同时Cache获取数据。

➤ 通视结构

Cache位于CPU和系统总线之间，CPU发出的所有数据请求必须经过Cache的监视。只有在Cache没命中时，数据请求才传给主存。



第五节 高速缓冲存储技术

2) 写入策略

通写法

当CPU写入数据到Cache后，Cache立即将其写入主存。

回写法

在系统总线空闲时，才将Cache中的内容回写到主存。

3) 替换策略

当Cache装满后，新的数据要不断地替换Cache中过时的数据。一般采用：

随机法、先进先出法、最近最少使用法



第五节 高速缓冲存储技术

cache基本组织方式

- cache由速度较快的SRAM构成
- cache与主存之间以数据块为单位交换数据
- cache分为若干数据块，块大小固定
- 每个数据块包括若干字节
- 相联存储器存放已调入cache的数据块地址
- 相联存储器的字数与cache块数相等
- cpu给出的地址如与相联存储器中某个单元相同，表示数据命中，否则缺失



课程内容



第一节 概述

第二节 ROM存储器

第三节 RAM存储器

第四节 FLASH

第五节 高速缓冲存储技术

第六节 存储器的接口设计



第六节 存储器的接口设计

CPU对存储器进行读写操作，首先由地址总线给出地址信号，然后发出读写控制信号，最后才能在数据总线上进行数据的读写。连接接口设计时应注意：

1. CPU总线的带负载能力

存储器主要是电容负载，在简单系统中，CPU可直接与存储器相连，在较大系统中，需加驱动器再与存储器相连。

2. CPU时序与存储器存取速度之间的配合

CPU的取指周期和对存储器读写都有固定的时序，由此决定了对存储器存取速度的要求。



第六节 存储器的接口设计

3. 存储器地址分配和片选

内存分为ROM区和RAM区，RAM区又分为系统区和用户区，每个芯片的片内地址由CPU的低位地址来选择，芯片的片选信号由CPU的高位地址译码取得。

4. 控制信号的连接

存储器系统通常由许多存储器芯片组成，对存储器的寻址必须有两部分：低位地址线连到所有存储器芯片，实现片内寻址；高位地址线通过译码器或线性组合后输出作为芯片的片选信号，实现片间寻址。

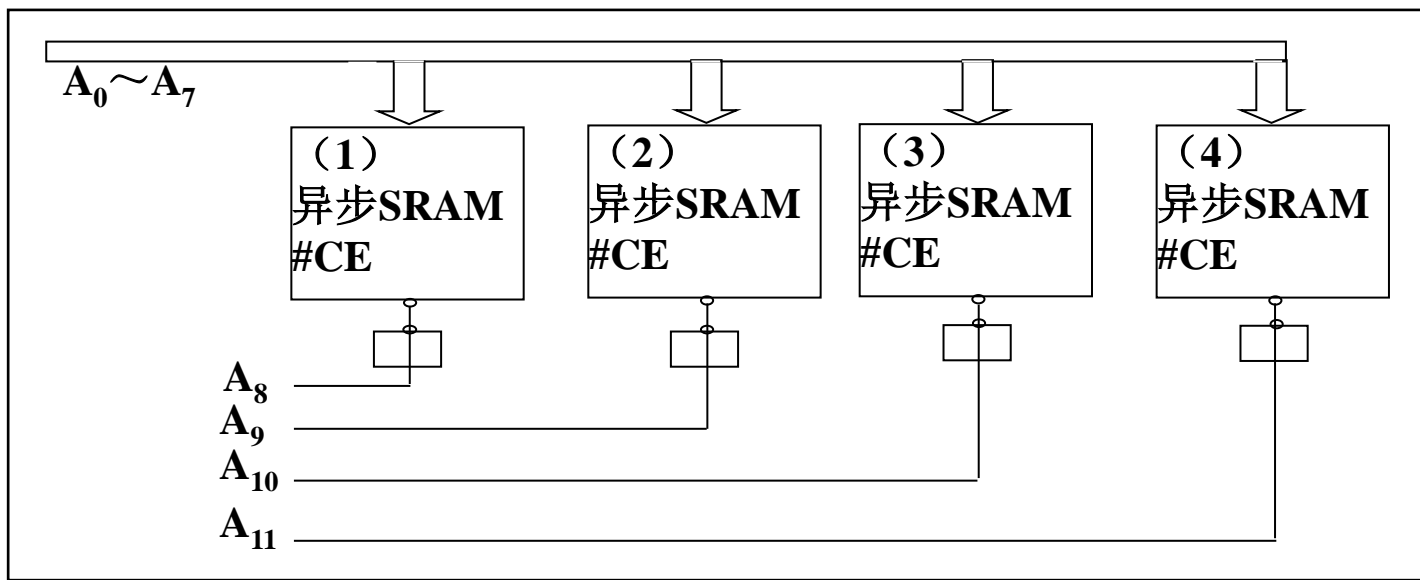
存储器地址选择有三种方法：

线性选择、部分译码和全译码

第六节 存储器的接口设计

• 地址译码的方法：

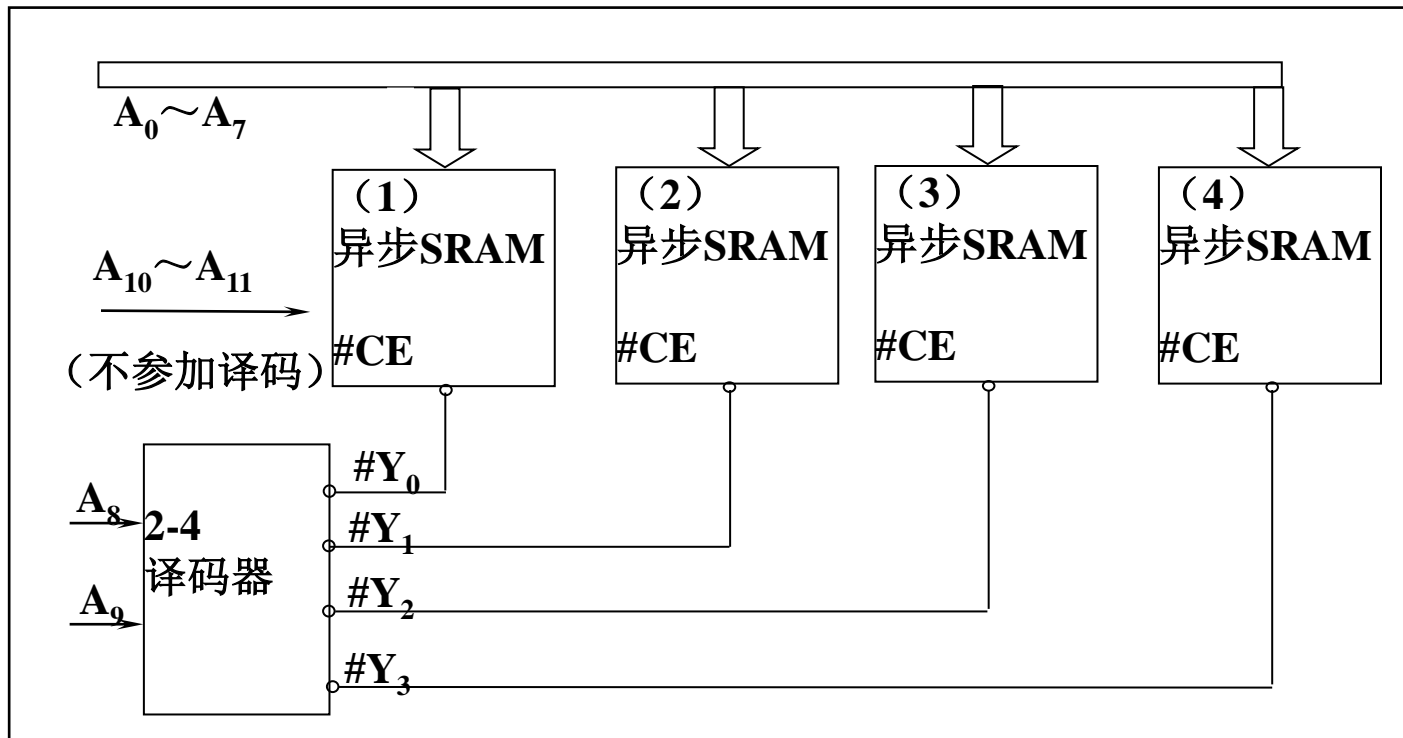
- **线选法译码**：将某根高位地址线直接作为芯片的片选。电路简单，但地址分配重叠，且地址空间不连续。适于用在容量小且不要求扩充的系统中。



第六节 存储器的接口设计

• 地址译码的方法：

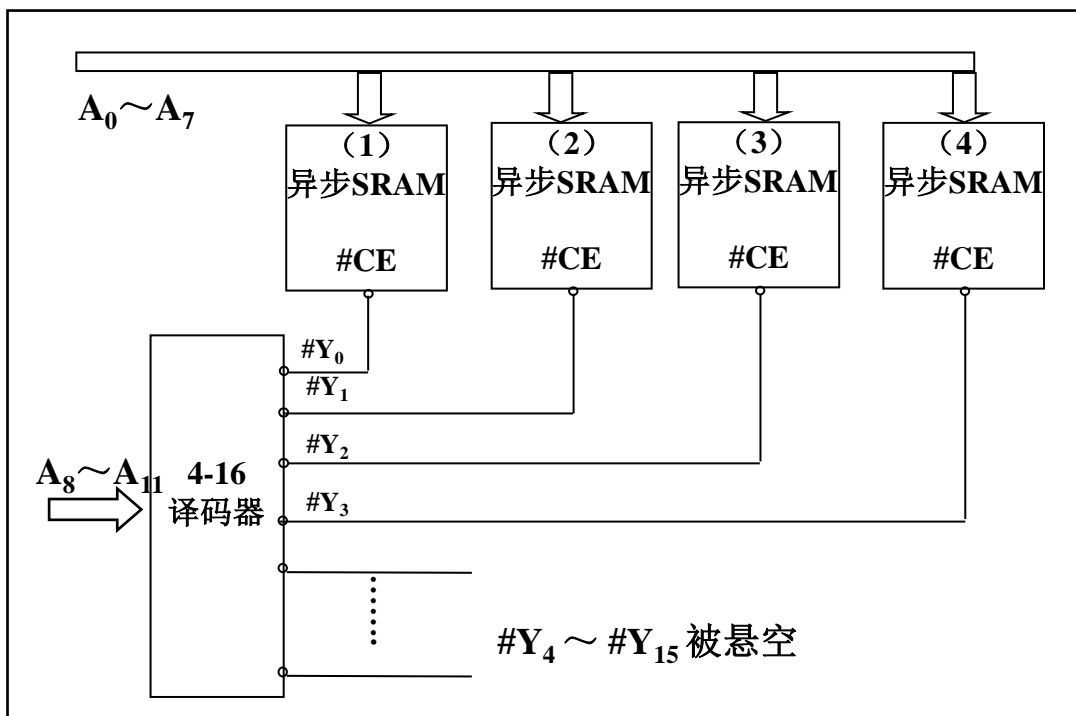
- **部分译码法：**将高位地址线中的几位经过译码后作为片选控制。



第六节 存储器的接口设计

地址译码的方法：

全译码法对全部高位地址进行译码，输出作为片选。译码电路复杂，但所得地址是唯一且连续的，并且便于内存扩充。





第六节 存储器的接口设计

存储器扩展方法：

□ 字长扩展(数据总线扩展)

- 各芯片并行工作

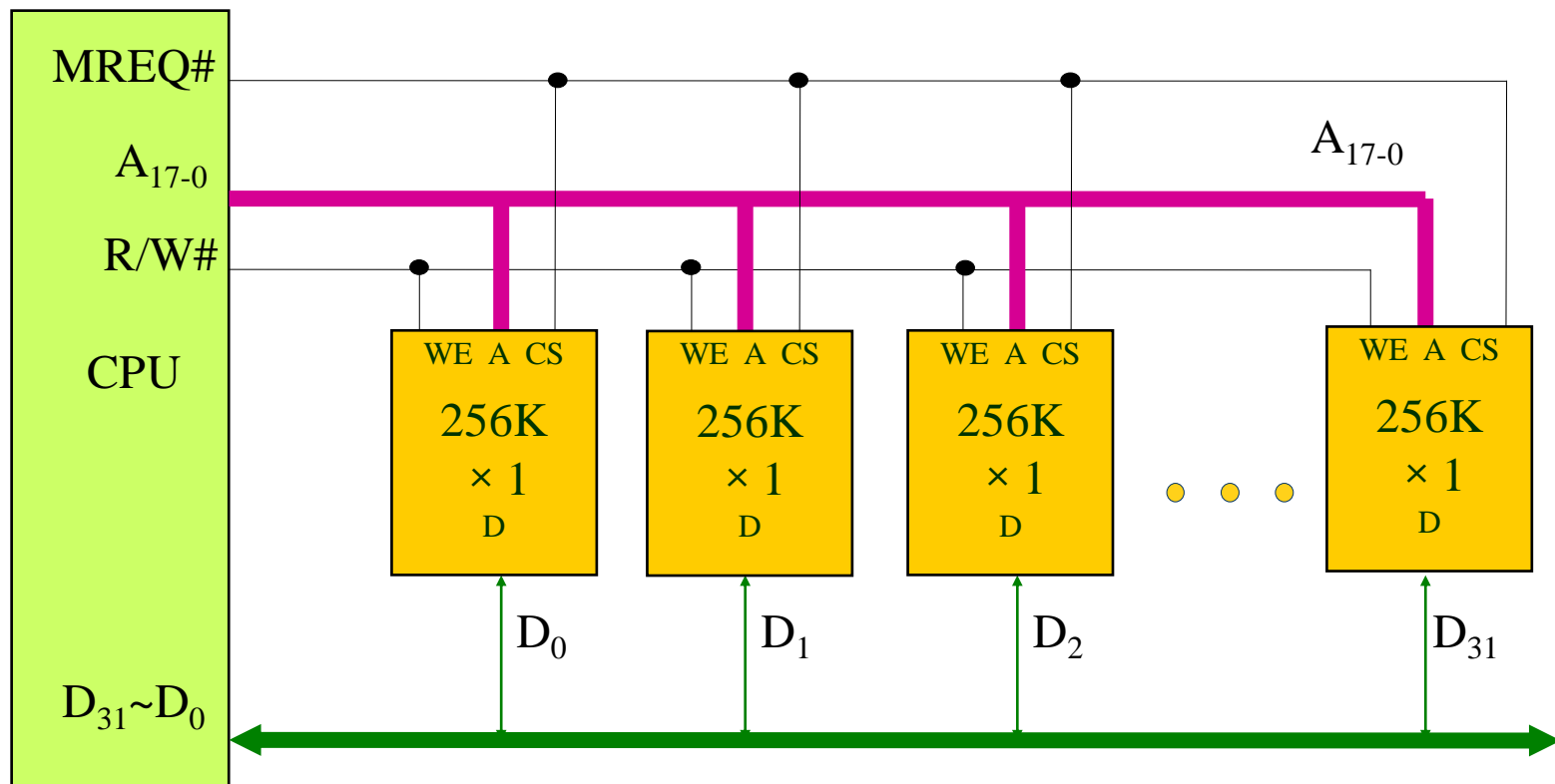
□ 字数扩展(地址总线扩展)

- 同一时刻仅一芯片工作



第六节 存储器的接口设计

(1) 字长扩展 (DBUS)

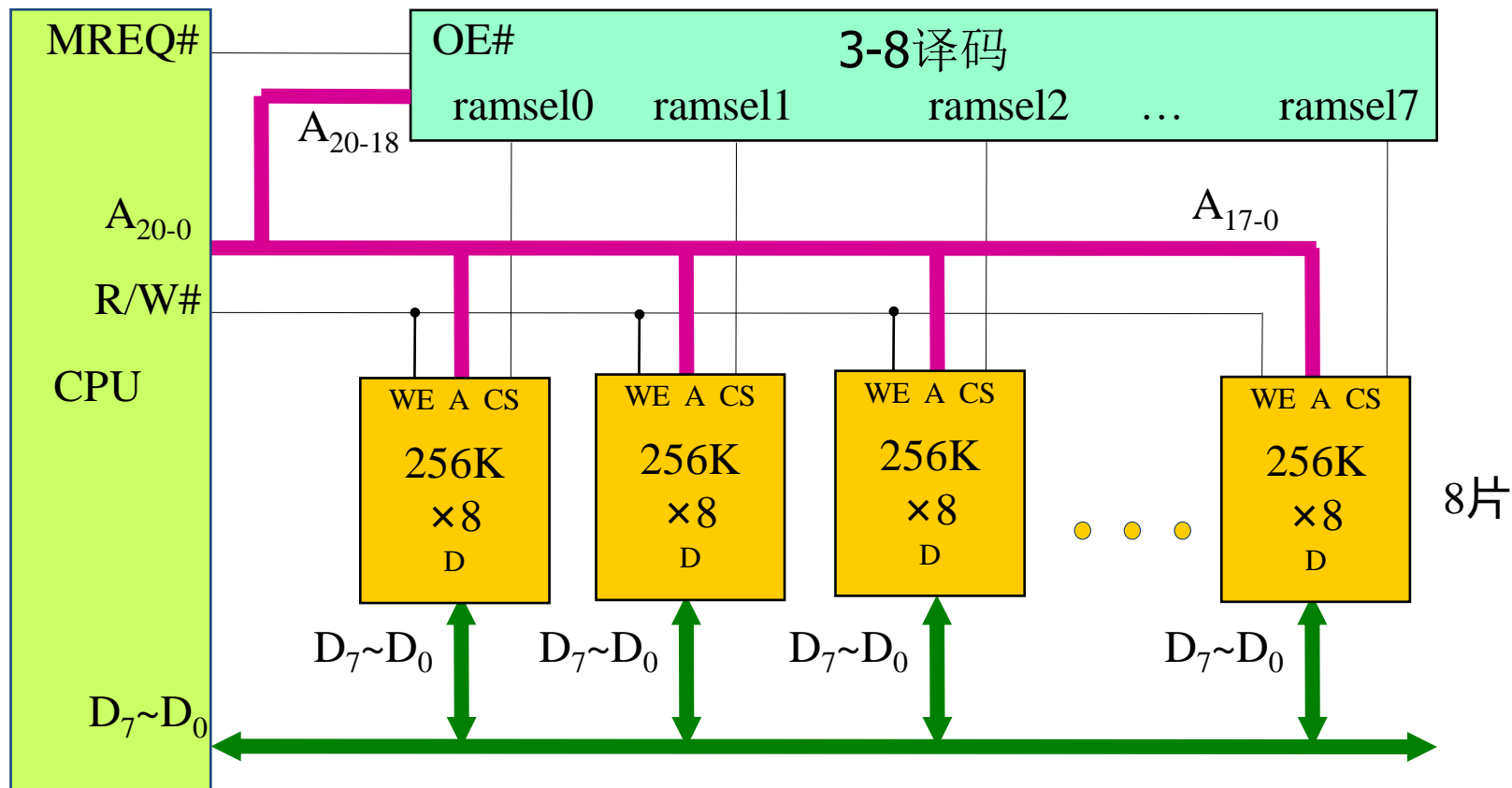


一个存储系统容量为 N 位，若使用 k 位的芯片， $k < N$ ，共需要 (N/k) 个芯片



第六节 存储器的接口设计

(2) 字数扩展 (ABUS)

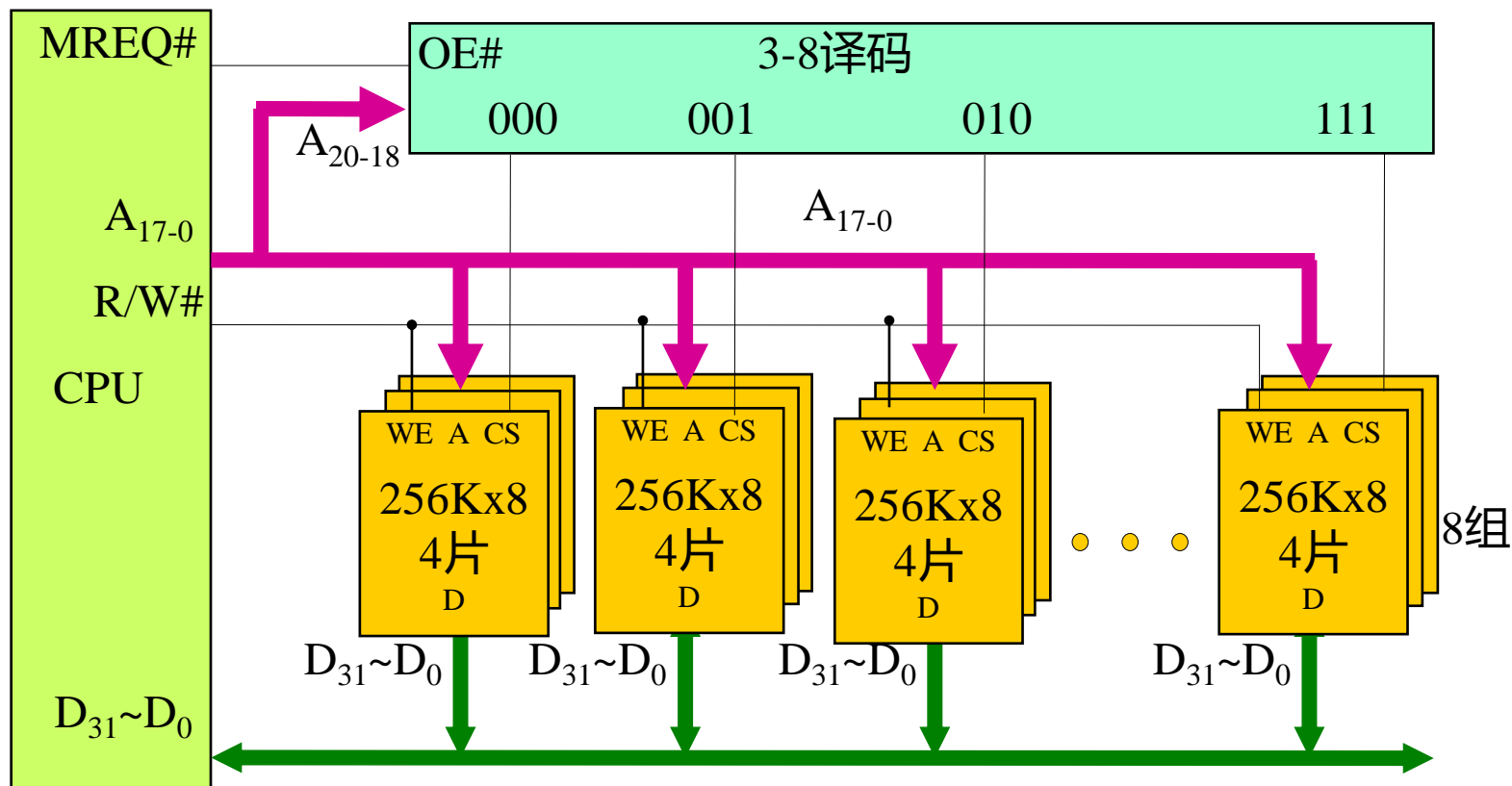


一个存储系统容量为 M ，若使用容量 l 的芯片， $l < M$ ，共需要 (M/l) 个芯片



第六节 存储器的接口设计

(3) 综合扩展 (ADBUS)



一个存储系统容量为 $M*N$ 位，若使用 $l*k$ 位的芯片， $l < M, k < N$ ，共需要 $(M/l) * (N/k)$ 个芯片



第六节 存储器的接口设计

例2. 设有32片 $256\text{K} \times 1$ 位的SRAM芯片

(1) 采用位扩展方法可构成多大容量的存储器？

(2) 该存储器需要多少字节地址位？

(3) 画出该存储器与CPU连接的结构图，设CPU的接口信号有地址信号、数据信号、控制信号 MREQ\# 和 $\text{R/\text{W\#}}$ 。

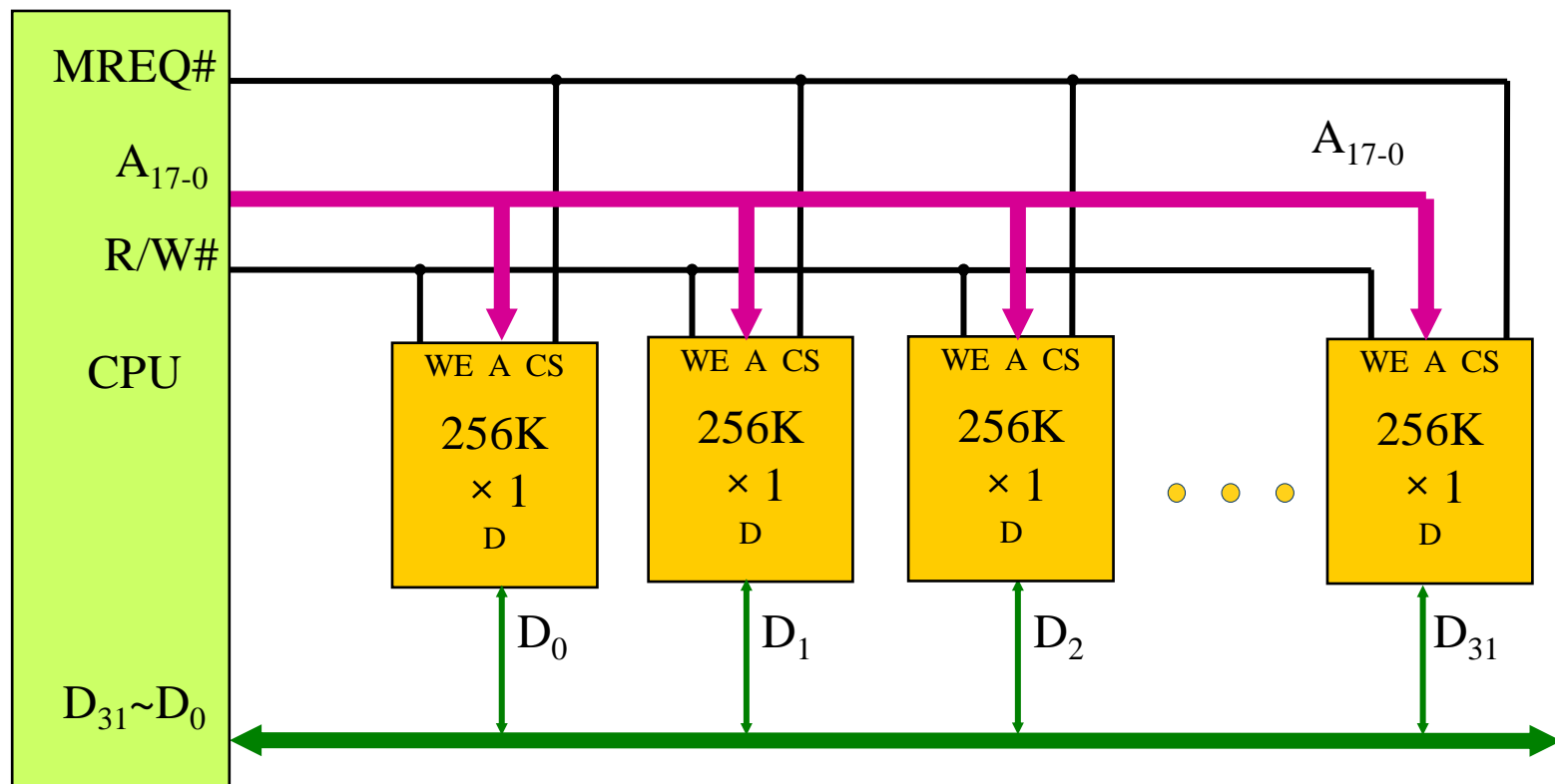
解： $256\text{K} \times 1$ 位SRAM芯片包含18根地址线

(1) 32片 $256\text{K} \times 1$ 位的SRAM芯片可构成 $256\text{K} \times 32$ 位的存储器。

(2) 如采用32位字编址方式，则需要18条地址线，因为 $2^{18} = 256\text{K}$ Word。

如果采用的字节编址方式，则需要20条地址线，因为 $2^{20} = 1024\text{K byte}$ 。

采用4片256k×1的芯片SRAM（字编码）





第六节 存储器的接口设计

- 应考虑的问题：
 - 存储器容量
 - 总容量：多少M/G字节
 - 单芯片容量： $2^m \times n$ 位， n 为数据线宽度， m 为地址线宽度
 - 存储空间的安排
 - 总线上的时序
 - 端口的逻辑电平与驱动能力
 - 数据总线的宽度



第六节 存储器的接口设计

小结：

- **数据线：**

- 根据存储系统的数据线宽度和芯片的数据线宽度构造成组
- 每组有多片时注意数据线连接时的分配

- **读写控制：**

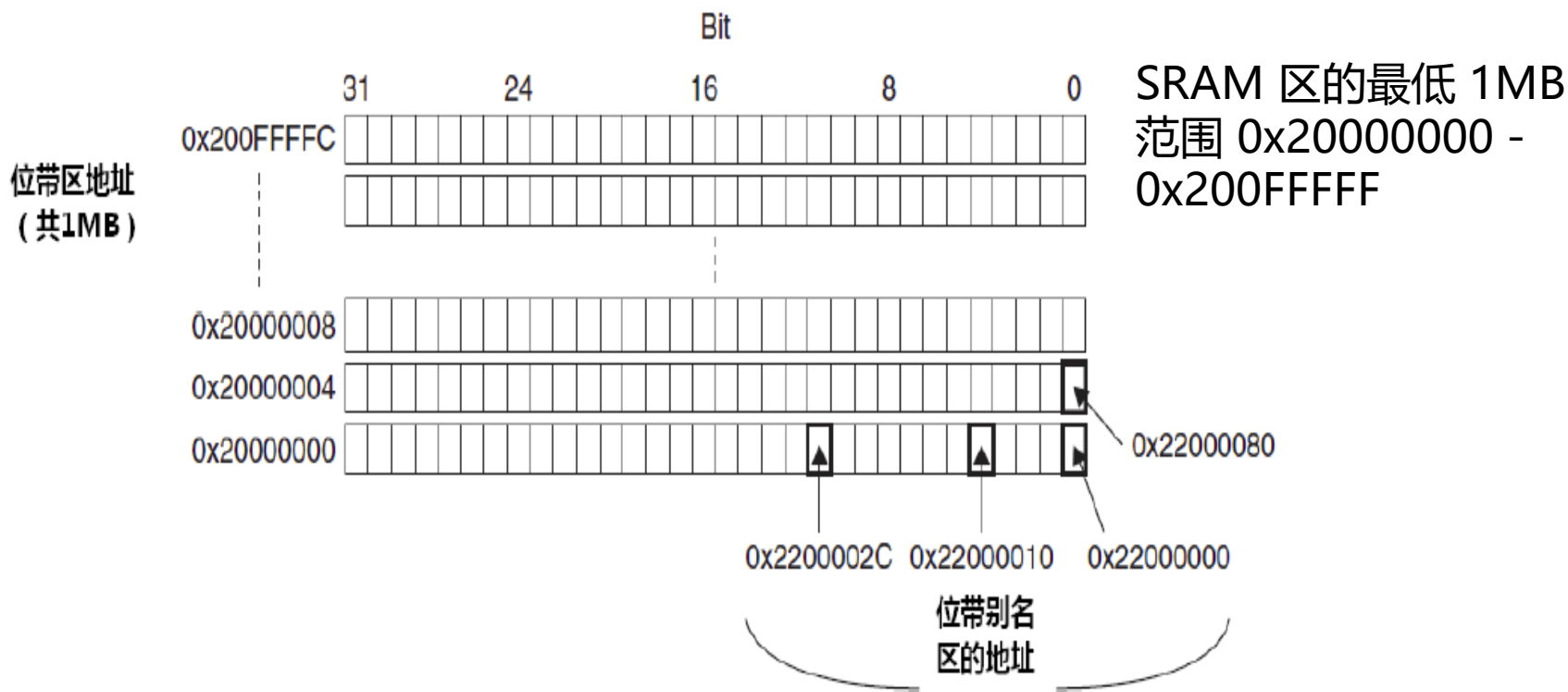
- 存储系统的数据宽度与CPU的数据线宽度不一致时，写访问控制需要额外的电路及控制信号

- **地址线：**

- CPU固定按8位编址，如果存储系统不是8位宽度，低位地址线需要错位连接
- 可以使用BANK的片选，或者用高位地址线译码生成片选

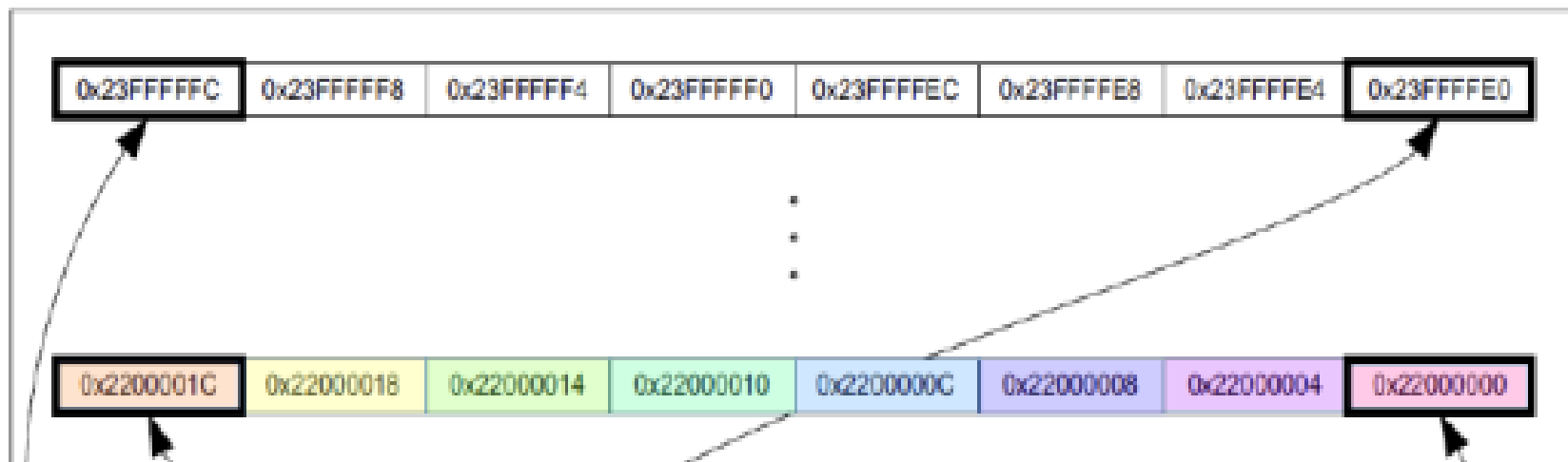
位带操作Bit-Band

Cortex-M4 等支持了位操作后，可以使用普通的加载/存储指令来对单一的比特进行读写。有两个区中实现了位带。

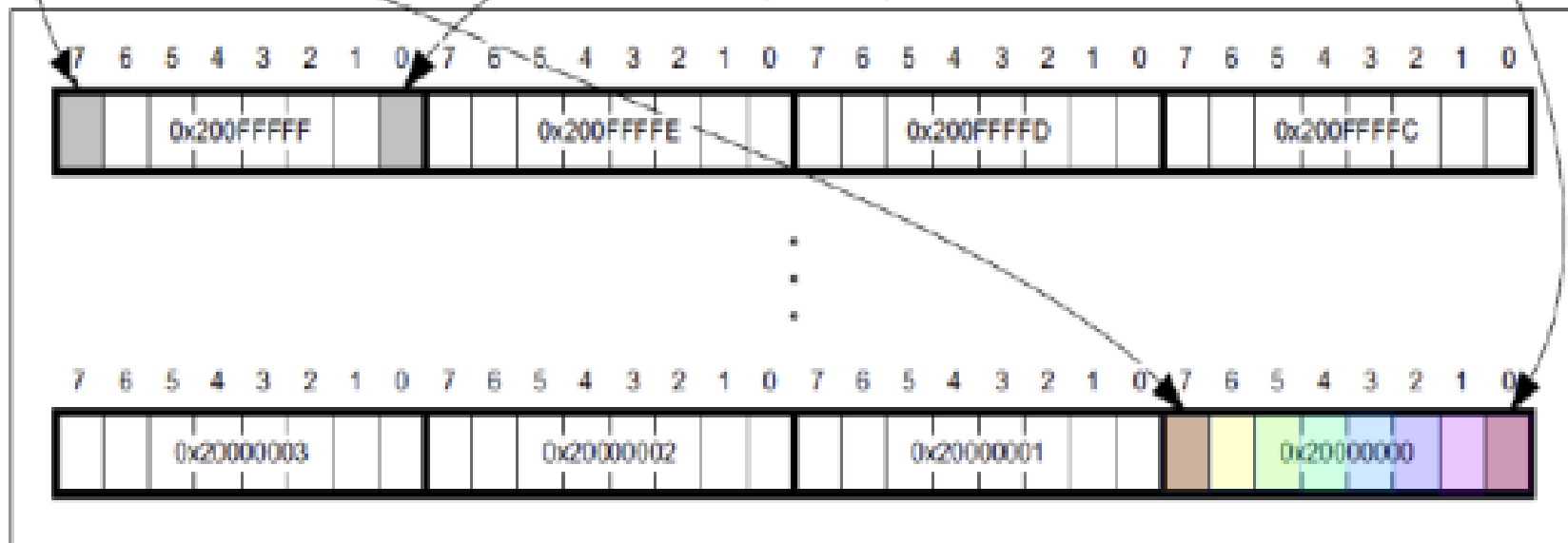


位带区与位带别名区

位带别名区 (共32MB)



SRAM位带区 (共1MB)



位带区与位带别名区的膨胀对应关系图



- **但是：M7不支持位带**
- 1) *Bit band and data cache doesn't match very well. One of the bit band address ranges is in SRAM region and is cacheable, With bit band, the same information is available from two address spaces and if we need to support this, the data cache need to remap the address on the fly which can be problematic.*
- 2) *bit band requires bus level lock mechanism, In AHB bus protocol this is relatively easy to implement, but in AXI bus protocol this is a bit more messy and during the lock sequence, it could cause longer latency to other bus masters..*

谢谢！