北京郵電大學
BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS

Computer Organization and Architecture

Chapter 5

Internal Memory

School of Computer Science（National Pilot Software Engineering School）
AO XIONG (熊翱)
xiongao@bupt.edu.cn

Assignment- 1

3.3 假设 32 位微处理器的 32 位指令有两个字段：第一个字节包含操作码，其余的字节为立即数操作数或操作数地址。

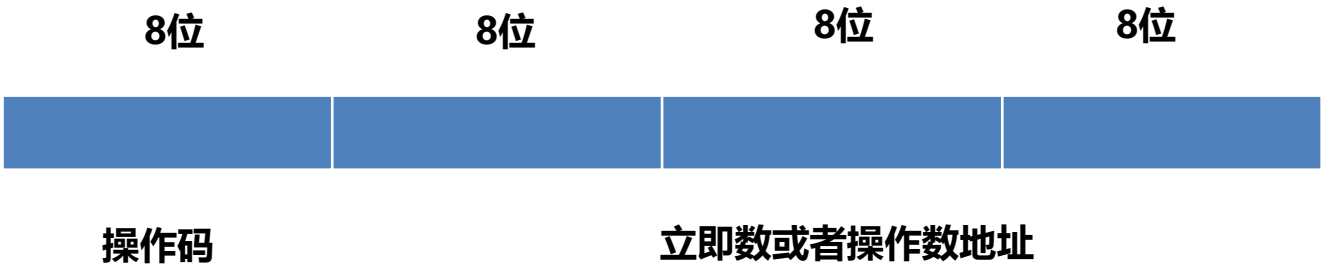
a. 可直接寻址的最大内存容量是多少（以字节计）？

b. 若微处理器总线如下，讨论其对系统速度的影响：

1）32 位局部地址总线和 16 位局部数据总线。

2）16 位局部地址总线和 16 位局部数据总线。

c. 程序计数器和指令寄存器需要多少位？



Assignment- 1

- （1）可寻址的最大内存容量

地址的位数为24位，地址从0000 0000 0000 0000 0000 0000~1111 1111 1111 1111 1111 1111

最大可寻址空间为2^{24}=16M（可寻址单元）

Assignment- 1

- （2）总线对系统速度的影响

a. 地址总线32位，数据总线16位。地址只需要24位，所以一次可以将地址传给内存。但是因为指令是32位，所以需要访问2次才能获取完整的指令或数据。

b. 地址总线和数据总线均为16位。由于寻址空间为16M，需要24位地址。这样一次就不能传完地址，需要2次才能传送完整的地址。数据也是一样，需要2次才能得到完整的指令或数据。

Assignment- 1

- （3）程序计数器和指令寄存器的位数

a. 地址总共是24位。程序计数器保存的是下一个指令的地址，所以最小只需要24位。

b. 指令寄存器保存的当前要执行的指令。指令为32位，所以指令寄存器需要32位。

Assignment- 2

3.14 微处理器有一条加 1 的内存直接指令，它把内存位置中的内容加 1。该指令分 5 个步骤：取操作码（4 个总线时钟周期），取操作数地址（3 个周期），取操作数（3 个周期），操作数加 1（3 个周期），存储操作数（3 个周期）。

a. 如果必须在每个内存读和内存写操作中插入两个总线等待状态，指令的持续时间会增加多少（以百分比形式）？

b. 若加 1 操作需要 13 个周期，而不是 3 个周期，重做上题。

取操作码	取操作数地址	取操作数	操作数+1	存操作数
4	3	3	3/13	3

（a）没有内存读写的等待状态，总共需要16个周期。内存操作每个需要2个周期，需要增加8个周期。16到24，增加了50%

（b）没有等待状态，需要26个周期。内存操作增加8个周期后，需要34个周期。从26到34，增加了30%

Assignment- 3

4.7：Intel80486有一个统一的片内cache，其容量为8KB，采用四路组相联结构，每块包含4个32位字。cache组织成128组，每行有一个“行有效位”与另外3个位B0,B1,B2(LRU）位。在cache未命中时，80486在一个总线存储器读周期从主存读取一个16字节的行。画出简化的cache框图，并说明地址的不同字段是如何转换的。

Assignment- 3

1985年，Intel 推出80386处理器，内部和外部数据总线是32位，地址总线也是32位，寻址范围4GB

2001年5月，Intel公司正式推出了第一种64位微处理器Itanium

Intel80486有32位地址线，也就是地址是32位，按字节寻址

Cache总共128组，组号字段用$log_2 128 = 7$ 位表示

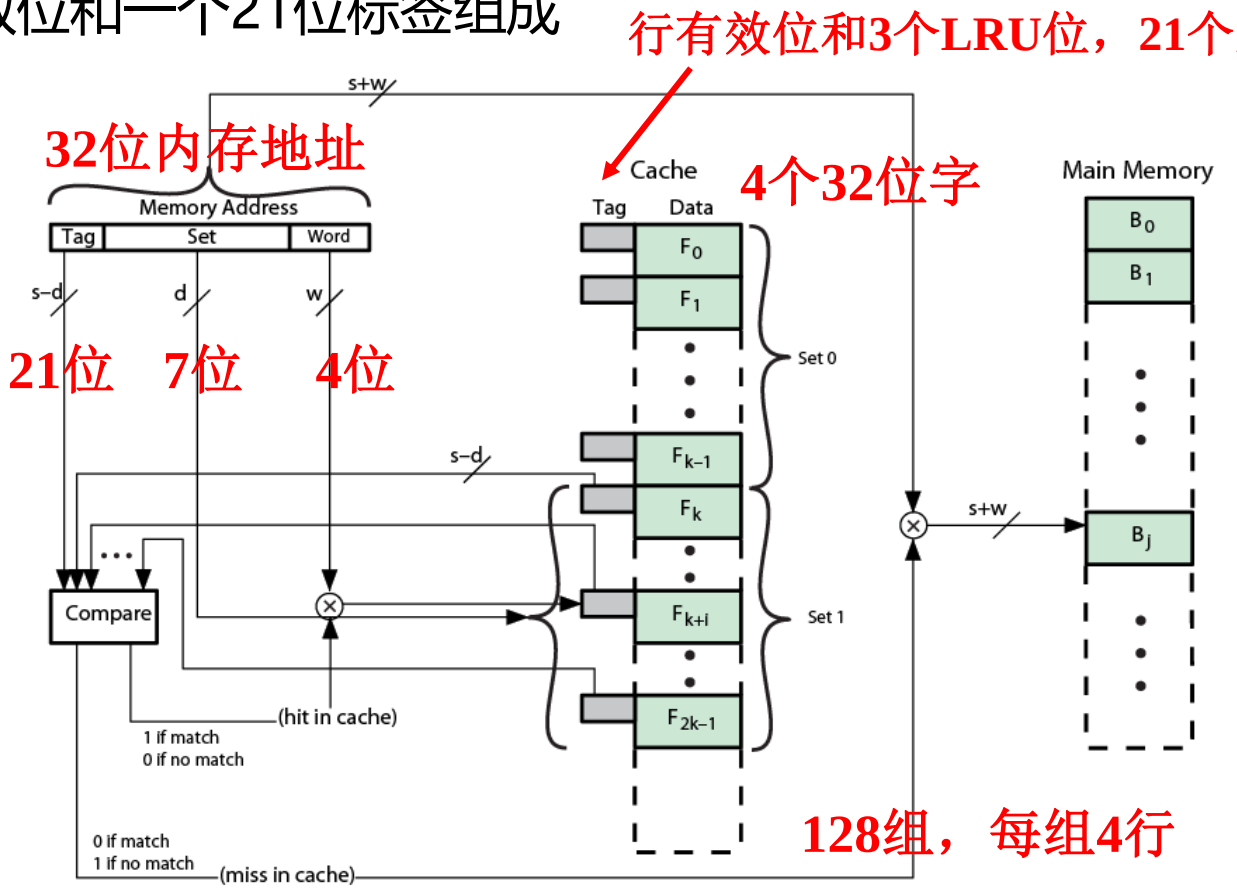
Cache每行包含4*4 = 16个Byte

字字段用$log_2 16 = 4$ 位表示

标签字段用$32 - 4 - 7 = 21$位表示

Assignment- 3

32位地址由21位标签字段、7位组号字段和4位字字段组成。高速缓存中的每组包括4行。每行由4个32位字、 3个LRU位、一个有效位和一个21位标签组成



Assignment- 4

- 4.10 一个组相联cache，每块（行）大小为4个16位字，组大小为2，cache总容量为4096个字。可缓存的内存容量为64k*32位。设计一种cache的结构，并说明如何进行转换。

分析

- Cache，4096个字，每行4个16位字，每组2行

Cache行号	行内字号
0	0，1，2，3
1	
2	
3	
4	
……	
1022	
1023	

- 总共1024行
- 每行4个字，每个字是16个bit
- Cache总共512个组

分析

- 内存64K*32位

	内存单元	单元内字号
内存块	0	0，1
	1	
内存块	2	
	3	
	4	
	……	
内存块	65534	
	65535	

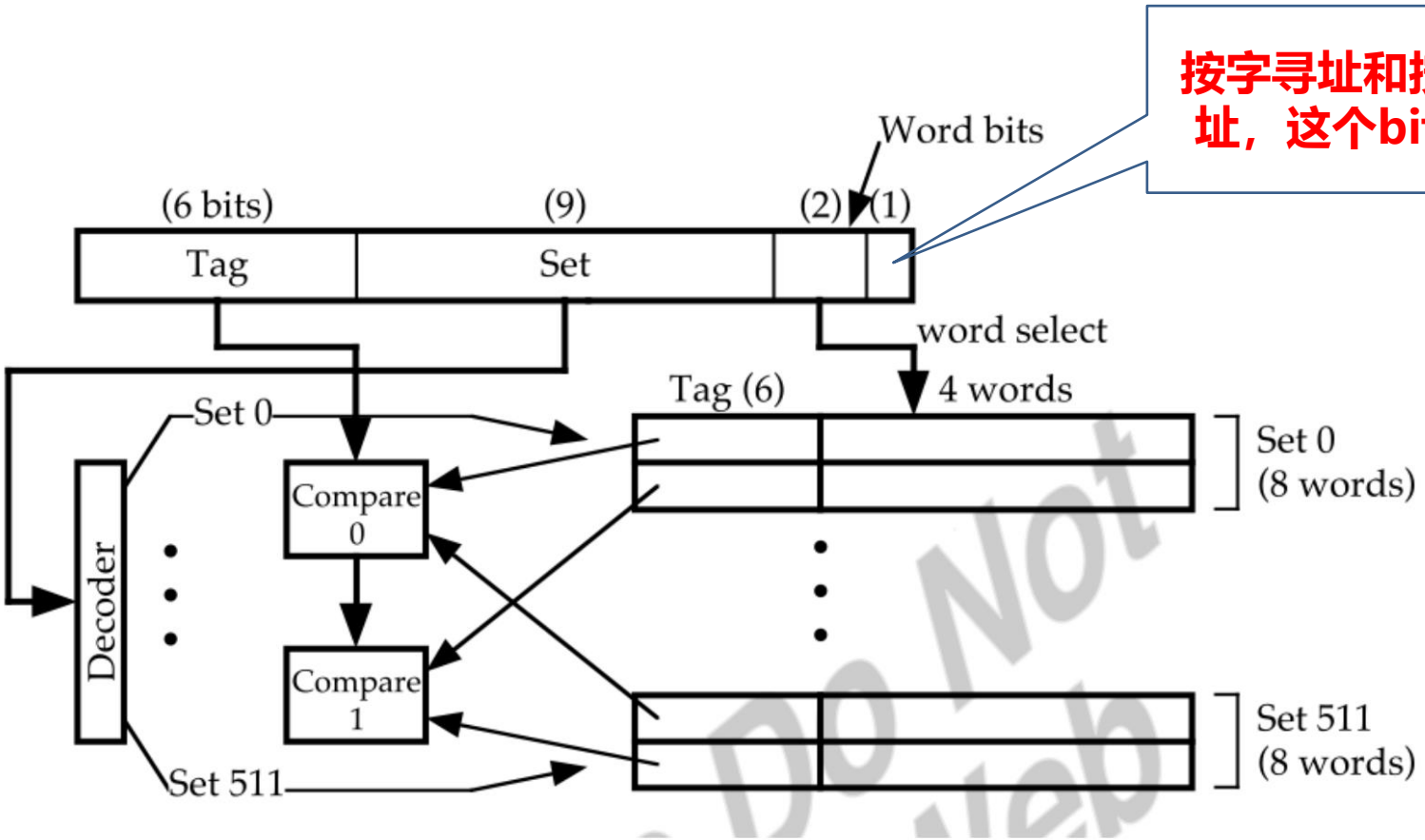
- Cache一行是4个16位字，也就是8个字节。
- 一个内存单元是32位，4个字节。
- 内存块对应cache行，所以内存总共有32k个块，每块是2个内存单元（8个字节）
- 如果按字寻址，总共有128k字，需要17位地址
- 如果按字节寻址，总共有256k字节，需要18位地址

分析

- 如果内存按字寻址，也就是以字位单位，那么内存总地址位：17位地址。
- 块内地址为2位。内存总共32k块，cache组为512组，所以cache的组号需要9位
- 标记位为：17-2-9=6位

- 如果内存按字节寻址，那么内存总地址位：18位地址。
- 一个块内有8个字节，块内地址为3位。内存总共32k块，cache组为512组，所以cache的组号需要9位
- 标记位为：18-3-9=6位

分析


按字寻址和按字节寻址，这个bit不一样
Word bits
(6 bits)
(9)
(2)
(1)
Tag
Set
word select
Tag (6)
4 words
Set 0
Compare 0
Compare 1
Decoder
Set 511
Set 0 (8 words)
Set 511 (8 words)

Preface

We have learned:

- Basic Concepts and Computer Evolution 基本概念和计算机发展历史
- Performance Issues 性能问题
- Top level view of computer function and interconnection 计算机功能和互联结构顶层视图
- Cache Memory cache存储器
  - Memory system overview 存储系统概述
  - Cache memory principles cache工作原理
  - Elements of cache design cache设计要素
  - Pentium 4 Cache Organization P4 cache组织

Review

- Cache的替换策略有哪些?
- 为什么需要写策略？有哪些写策略?
- Cache设计中的几个要素

Review

- Cache的替换策略
  - 直接映射，不需要策略
  - 全相联映射和组相联映射：最近使用原则，最少使用原则，FIFO，随机
- 为什么需要写策略？有哪些写策略？
  - 写直达，写回法
- Cache设计中的几个要素
  - Cache地址，容量，映射，替换，写策略，行大小，cache数目

Preface

We will focus the following contents today:

- Internal Memory 内部存储器
  - What are the characteristics of internal memory? 内部存储器的特点
  - How to correct error in internal memory? 内部存储器如何纠错
  - How to improve the performance of internal memory? 如何提高内部存储器的性能

Outline

- Key Terms 关键术语
- Semiconductor main memory 半导体内存
- Error correction 纠错算法
- DDR DRAM 双倍速率DRAM
- Flash Memory 闪存

Key terms （1）

- semiconductor memory：半导体存储器
- dynamic RAM (DRAM)：动态RAM
- static RAM (SRAM)：静态RAM
- read-only memory (ROM)：只读存储器
- programmable ROM(PROM)：可编程ROM
- erasable programmable ROM(EPROM):可擦除/可编程ROM
- electrically erasable programmable (EEPROM)：电可擦除/可编程ROM
- read-mostly memory：主读存储器
- flash memory：快闪存储器

Key terms （2）

- error correcting code (ECC):纠错码
- error correction：错误纠正
- single-error-correcting (SEC) code：单纠错码
- single-error-correcting , double-error-detecting (SEC-DED) code：单纠错双检错码
- Hamming code：汉明码
- volatile memory：易失性存储器
- nonvolatile memory：非易失性存储器

Key terms （3）

- hard failure：硬故障
- soft error：软差错
- Syndrome：故障，综合故障
- synchronous DRAM (SDRAM)：同步DRAM
- RamBus DRAM (RDRAM)：总线式DRAM
- cache DRAM (CDRAM):带高速缓存的DRAM

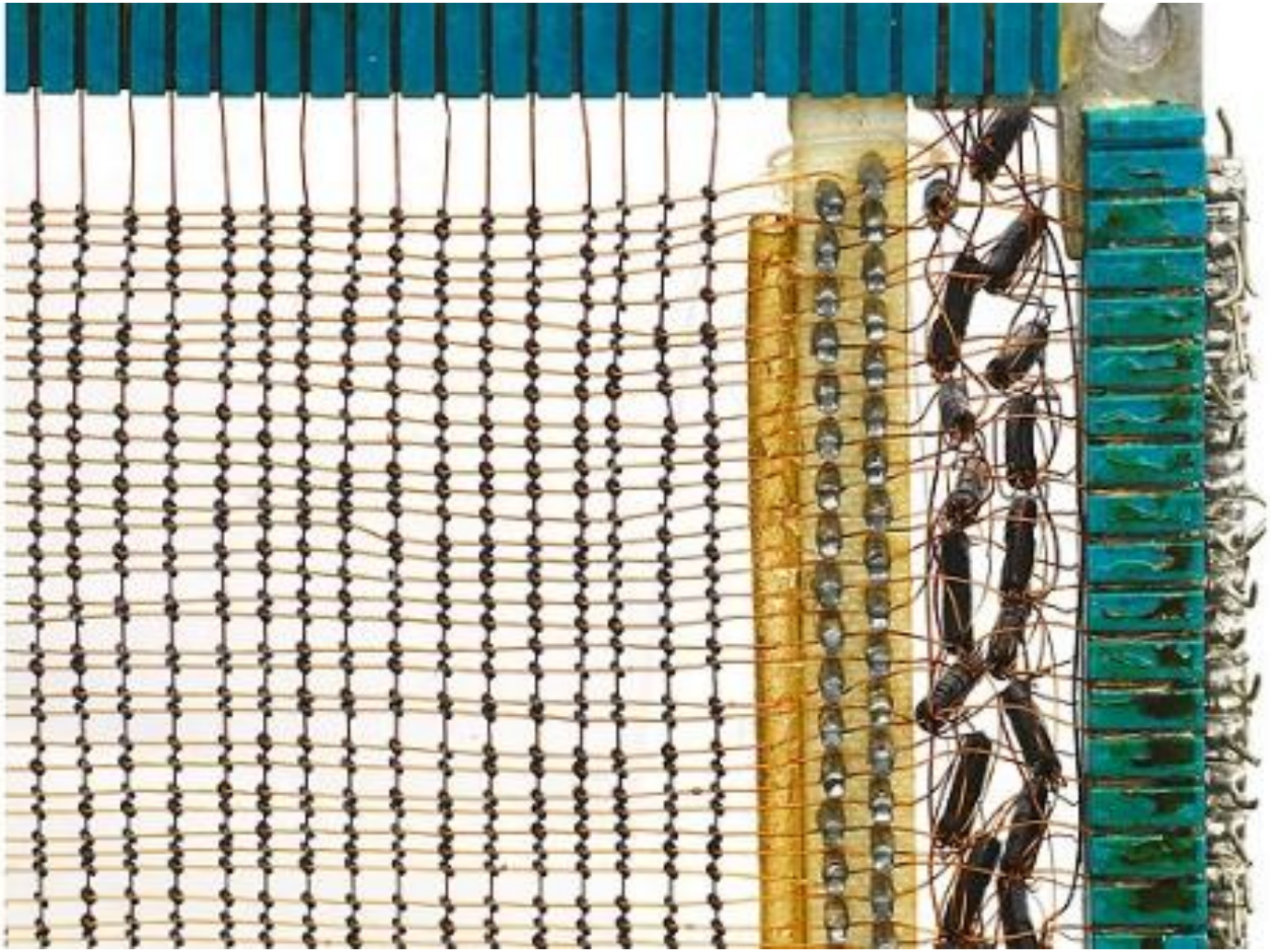
Outline

- Key Terms 关键术语
- Semiconductor main memory 半导体内存
- Error correction 纠错算法
- DDR DRAM 双倍速率DRAM
- Flash Memory 闪存

Core memory 磁芯存储器

- Memory used earlier 早期使用的存储器
  - Used in the 1950s and 1960s 50~60年代使用
  - tiny rings of ferromagnetic material 很小的铁磁体环做成
  - each about a sixteenth of an inch in diameter 每个环的大小大概是1/16英寸
- Characteristics 特点
  - Fast, about 1us 速度快，大概1us
  - Large volume 体积大
  - Expensive 昂贵
  - Volatile，must be written immediately after reading 易失，读出之后必须立即写入

Core Memory 磁芯存储器



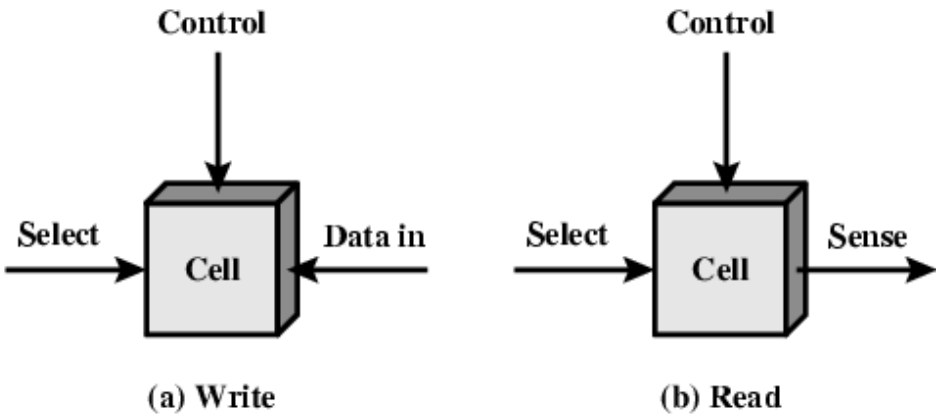
Semiconductor memory 半导体存储器

- By 1970, Fairchild produced first semiconductor memory with large capacity 1970年，仙童公司生产第一个大容量半导体存储器
  - With a capacity of 256 bits 容量是256 bit
  - As large as a magnetic core 一个磁芯大小差不多
  - High storage density 存储密度大
  - Nonvolatile 非易失
  - Faster than core, about 70ns 存取快，大概70ns
- 1974 – the price per bit of semiconductor memory dropped below the price of core memory 1974年，半导体存储器单位bit价格低于磁芯
- since then, cost continue decreasing, density continue increasing 从那时起，半导体存储器价格持续下降，密度持续上升

Semiconductor main memory 半导体存储器

- DRAM and SRAM DRAM和SRAM
- Types of ROM ROM的类型
- Organization 组织
- Chip Logic 芯片逻辑
- Chip Packaging 芯片封装
- Module Organization 模块组织
- Interleaved Memory 多体交叉存储器

Memory cell operation 存储位元操作



- Basic element of memory is cell 存储器的基本元件是位元
  - has two stable states, representing 0 and 1 有2个基本状态
  - can be changed from 0 to 1, or from 1 to 0 能够在0和1之间变化
  - can read its current state by some way 能够读取当前状态
- Cell has two operations 位元有2个基本操作
  - Read 读
  - Write 写

Two basic memory types 两种基本的存储类型

- RAM – random access memory 随机访问存储器
  - SRAM – static RAM 静态RAM
  - DRAM – dynamic RAM 动态RAM
- ROM – read only memory 只读存储器
  - ROM 只读
  - PROM – programmable ROM 可编程ROM
  - EPROM –Erasable programmable ROM 可擦除可编程ROM
  - EEPROM – electronically erasable programmable ROM 电可擦除可编程ROM
  - Flash (EEPROM type) Flash，属于EEPROM

Semiconductor Memory Types 半导体存储器类型

<table>
<tr><th>Memory Type</th><th>Category</th><th>Erasure</th><th>Write Mechanism</th><th>Volatility</th></tr>
<tr><td>RAM</td><td>Read-write memory</td><td>Electrically, byte-level</td><td>Electrically</td><td>Volatile</td></tr>
<tr><td>ROM</td><td rowspan="2">Read-only memory</td><td rowspan="2">Not possible</td><td>Masks</td><td rowspan="6">Nonvolatile</td></tr>
<tr><td>PROM</td><td rowspan="4">Electrically</td></tr>
<tr><td>EPROM</td><td rowspan="3">Read-mostly memory</td><td>UV light, chip-level</td></tr>
<tr><td>EEPROM</td><td>Electrically, byte-level</td></tr>
<tr><td>Flash memory</td><td>Electrically, block-level</td></tr>
</table>

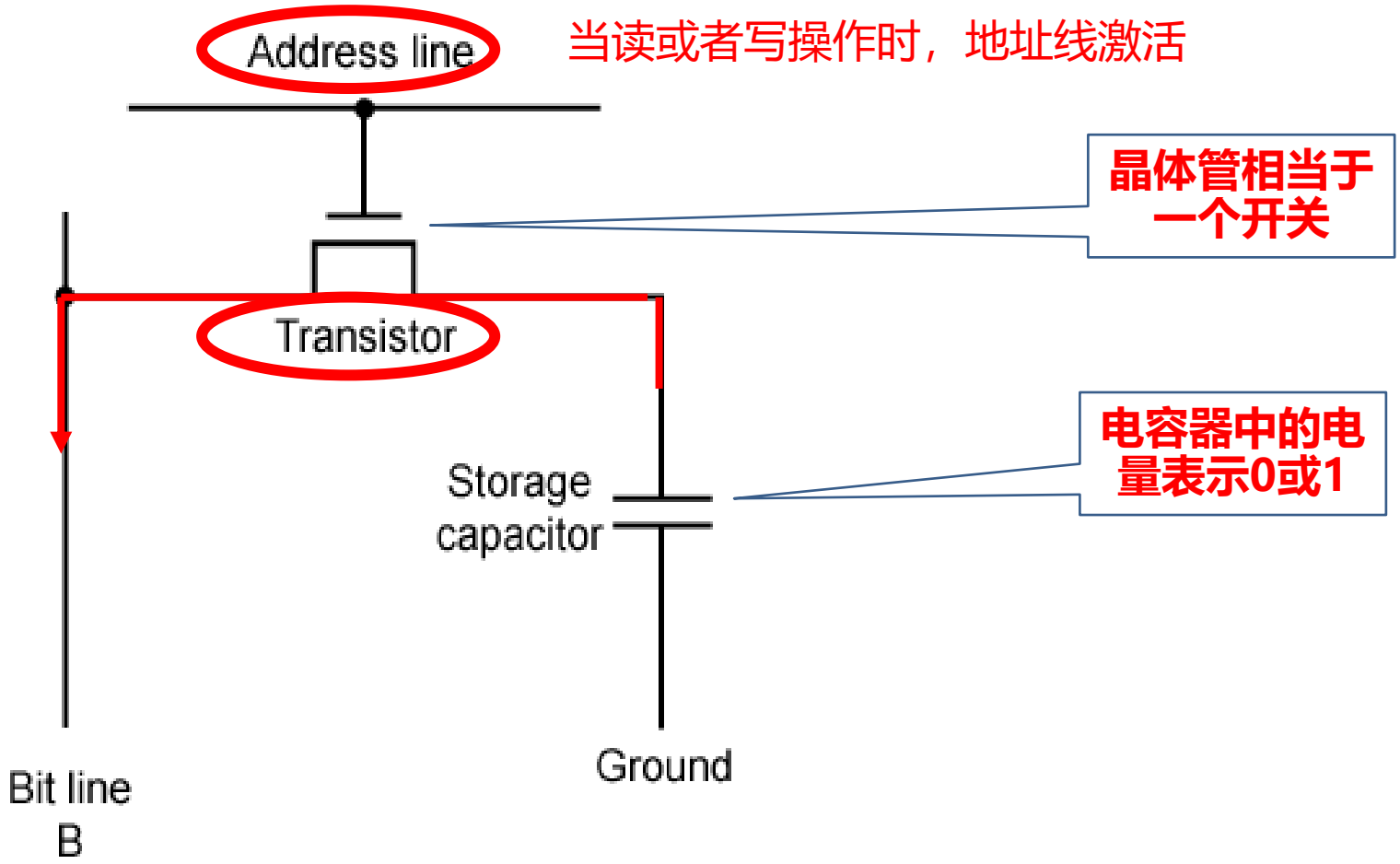
RAM

- The most commonly used as main memory 在内存中最常用
- Can be read and write at any time 能够随时读写
- Volatile – must be provided with a constant power supply
  易失性的，因此需要持续供电
  - When power is on, the content of the RAM cell stays 有电的时候，RAM中的内容能保持
  - When power is off, the data is gone 一旦没电了，数据就丢了
- Two major forms of RAM 两种基本的RAM
  - DRAM 动态RAM
  - SRAM 静态RAM

Dynamic RAM 动态RAM

- Use one capacitor to record 1 bit 用1个电容器来记录1bit
- Bits stored as charge in capacitors 用电容中的电荷来存储位
  - 1：capacitor is charged 电容器充电
  - 0：no electricity in capacitor 电容器没电
- Charge of capacitor leaks automatically 电容器电量会自动泄露
  - Need refreshing even when powered加电后需要持续刷新
- Characteristics
  - Simpler construction 构造简单
  - Smaller per bit 每一位的体积小

Dynamic RAM structure DRAM结构



DRAM operation DRAM的操作

- Address line active when bit read or written 当位元需要读写的时候，地址线激活
  - Transistor as a switch 晶体管类似一个开关
- Write 写操作
  - Voltage to bit line 位线施加电压
    - High for 1 low for 0 高电压表示1，低电压表示0
  - Then signal address line 给出地址线信号
    - Transfers charge to capacitor 晶体管导通

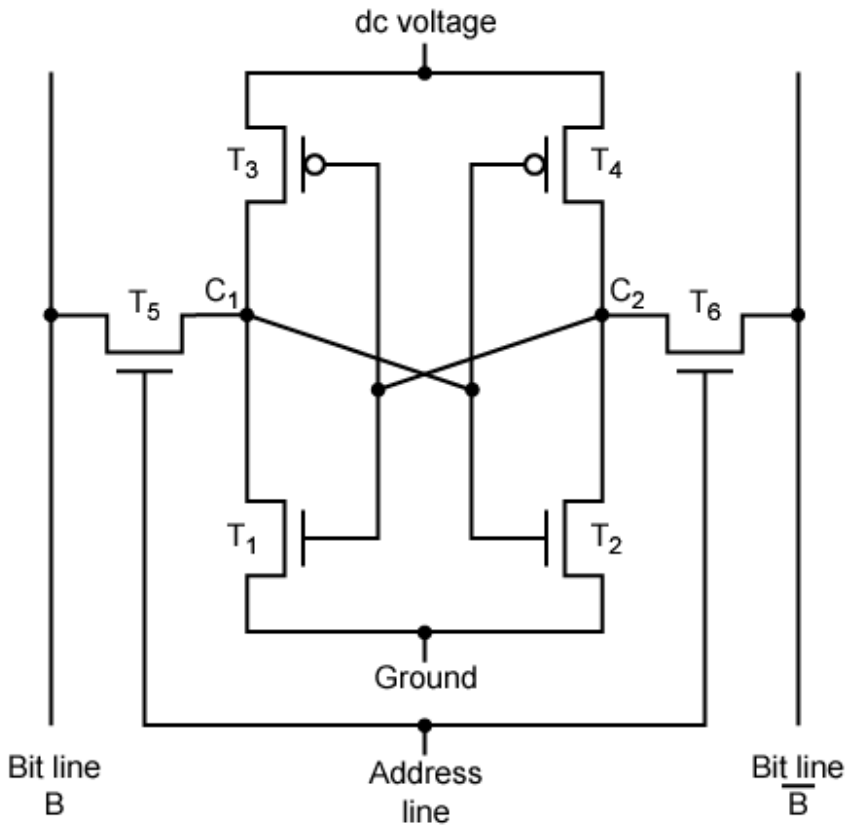
DRAM operation DRAM的操作

- Read 读
  - Address line selected 选择地址线
  - transistor turns on 晶体管导通
  - Charge from capacitor fed via bit line to sense amplifier 电容中存储的电荷通过位线给放大器
  - Compares with reference value to determine 0 or 1 通过比较确定是0还是1
  - When reading, the capacitor charges discharge 读出时，电容电荷放电
  - Capacitor charge must be restored 电容中的电荷需要重新存储

Characteristics of Dynamic RAM DRAM的特点

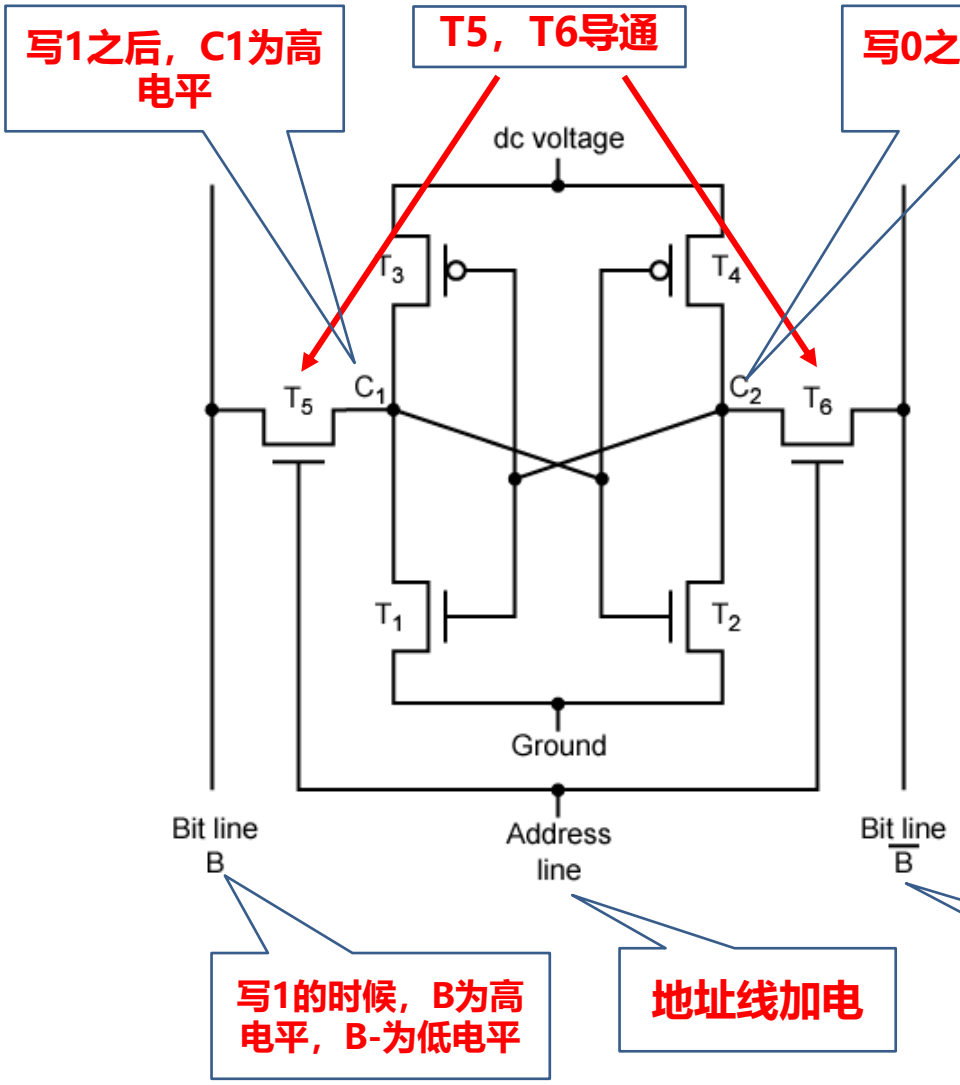
- Less electronic components, high integration and low price 采用的电子元件少，集成度高，价格便宜
- The charge stored in the capacitor will discharge，refresh circuits needed 电容存储的电荷会放电，需要刷新电路
- Voltage of the capacitor is an analog value 电容的电压是一个模拟值
  - Level of charge determines value 电荷量决定值
  - Speed of reading and writing is relatively slow Slower 读出和写入的速度相对比较慢
- Main memory 用作主存储器

Static RAM structure SRAM结构



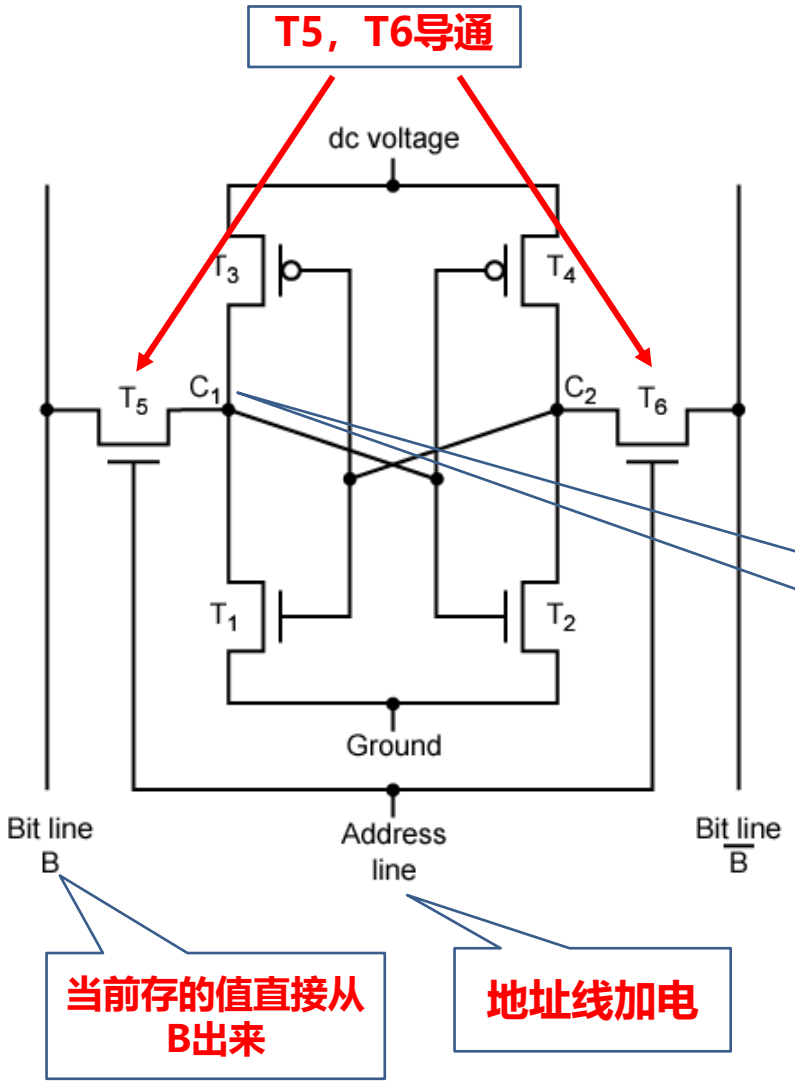
- 静态RAM位元由6个晶体管组成
- T1~T4交叉连接形成有稳定逻辑状态的阵列。T5和T6由地址线控制，进行位元的读或者写操作
- 在逻辑状态1时，C1是高电平，C2是低电平，这样T1和T4处于截止状态，而T2和T3处于导通的状态
- 在逻辑状态为0时，C1是低电平，C2是高电平，这样T2和T3处于截止状态，T1和T4处于导通状态

Static RAM write SRAM写



- 写操作的时候，施加信号给地址线，T5和T6导通
- 写1的时候，给B加正值，B-加负值， T1~T4进入到值为1的稳定值
- 写0的时候，给B加负值，B-加正值， T1~T4就进入到值为0的稳定状态

Static RAM write SRAM读



- 值从线B出来

Static RAM 静态RAM

- Advantage 优点
  - No charges to leak 电荷不泄露
  - No refreshing needed when powered 只需要供电，不需要刷新
  - Faster 快
  - Digital 数字的，不是模拟的
- Disadvantage 缺点
  - More complex construction 结构复杂
  - Larger per bit 每一位的体积大
  - More expensive 价格昂贵
- Used as Cache 主要用作cache

SRAM vs. DRAM SRAM和DRAM比较

Both volatile, Power needed to preserve data

都是易失性的，需要持续电力供应

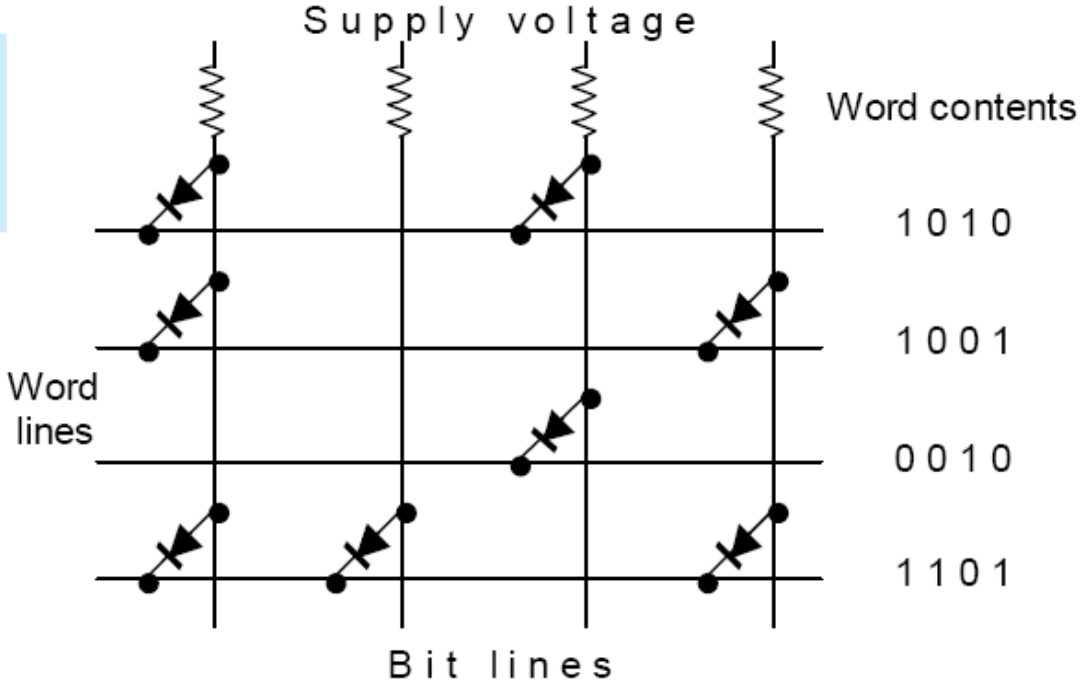
	SRAM	DRAM
Complexity	Complex	Simpler
Speed	Faster	Slower
Size	Bigger	Small
Price	Expensive	Cheaper
Refresh	No	Yes
use	cache	Main memory

Read only memory ROM 只读存储器

- Permanent storage 永久储存
  - Nonvolatile 非易失性
- While it is possible to read a ROM, it is not possible to write new data into it 可以读，不能写
- Important application 重要的应用
  - Microprogramming 微程序
  - Library subroutines 子程序库
  - Systems programs (BIOS) 系统程序，比如BIOS
  - Function tables 函数表

Structure of ROM ROM的结构

ROM
PROM
EPROM



- 存储的值固定在电路里
- 选择某行，在这一行加电
- 值从位线出来

Types of ROM – 1 ROM的类型1

- True ROM：written during manufacture 生产的时候写入
  - Very expensive for small runs 小规模生产很贵
  - ROM 作为只读
  - Factory programmed, cannot be changed. Older style 工厂写入，不能修改。老的形式
  - There is no room for error 不能出错
- Programmable (once) 一次可写，也就是可编程
  - PROM 称为PROM
  - Needs special equipment to program 需要特殊的设备去写入

Types of ROM – 2 ROM的类型2

- Read “mostly” 大部分是读操作
  - Erasable Programmable (EPROM) 可擦写PROM
    - Erased by UV 通过紫外线擦除
    - Can be written repeatedly for many times 可以重复写入多次
    - More expensive 价格比较贵
  - Electrically Erasable (EEPROM) 电可擦除PROM
    - Write directly after locating one or more bytes 直接对一个或多个字节进行定位后写入
    - Takes much longer to write than read 写的时间比读长很多
    - More flexible，more expensive 更灵活，价格更贵

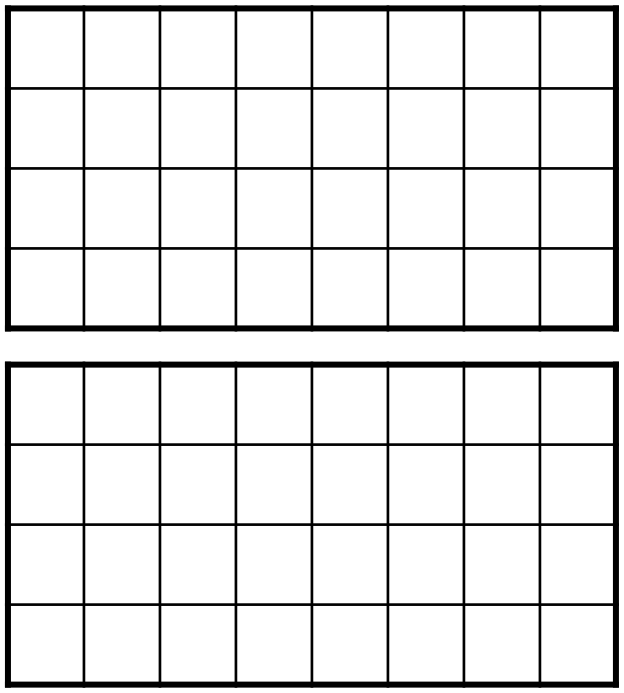
Types of ROM – 2 ROM的类型2

- Read “mostly” 大部分是读操作
  - Flash memory 闪存盘，U盘
    - Invented by Toshiba in the 1980s 20世纪80年代东芝公司发明
    - Block storage 采用块存储的方式
    - Erase whole memory electrically 电擦除整个内存块
    - Only one transistor is used to store bit 存储位元只用了一个晶体管
    - High integration and high storage density 集成度高，存储密度较大
    - Widespread use 广泛应用

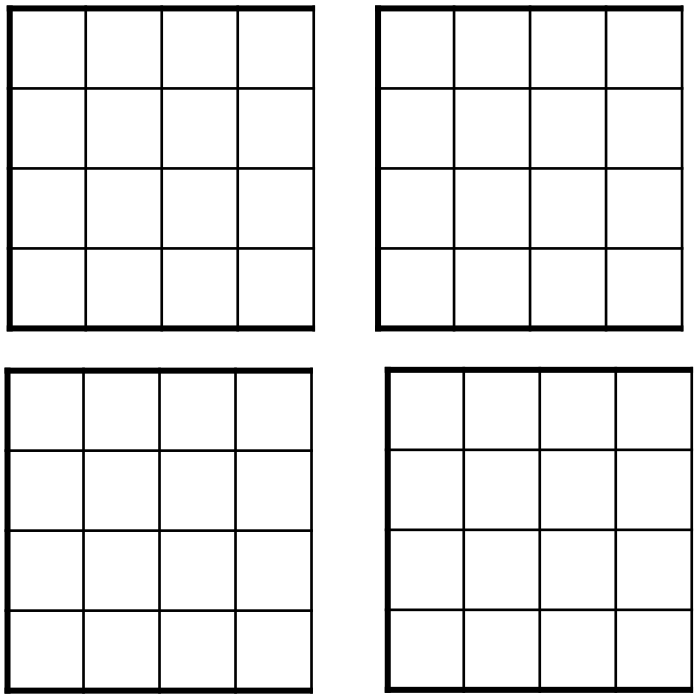
Memory organization 内存组织

- Semiconductor memory comes in packaged chips 半导体存储器封装成芯片
- The capacity of a single chip cannot meet the requirements of the storage system 单个芯片的容量不能满足存储系统的要求
- Multiple memory chips form a memory array 多个存储芯片组成存储阵列
  - Some chips are organized as one dimensional array 有些组织成一维阵列
  - Some are organized as two dimensional array 有些芯片组成成二维阵列
  - The data is read/written in bits 数据一次读写多位

Memory organization 存储器组织

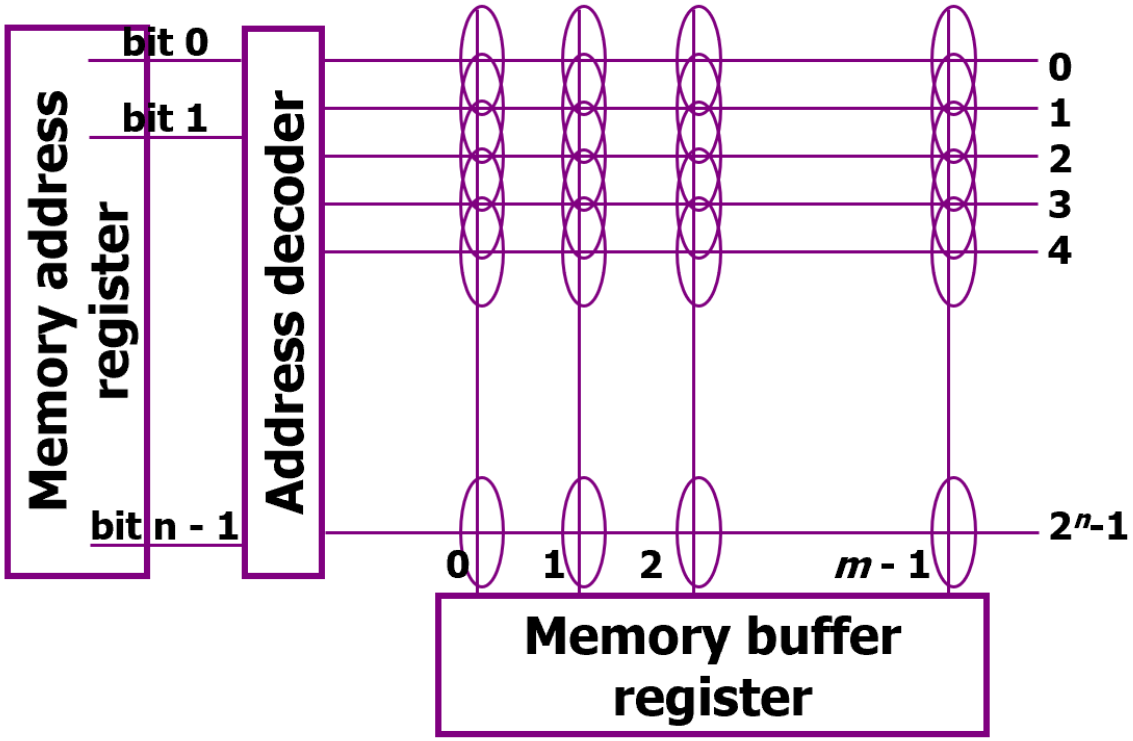


- 每个存储芯片是n*8bit
- m个存储芯片，组成一维的阵列，总容量为m*n byte



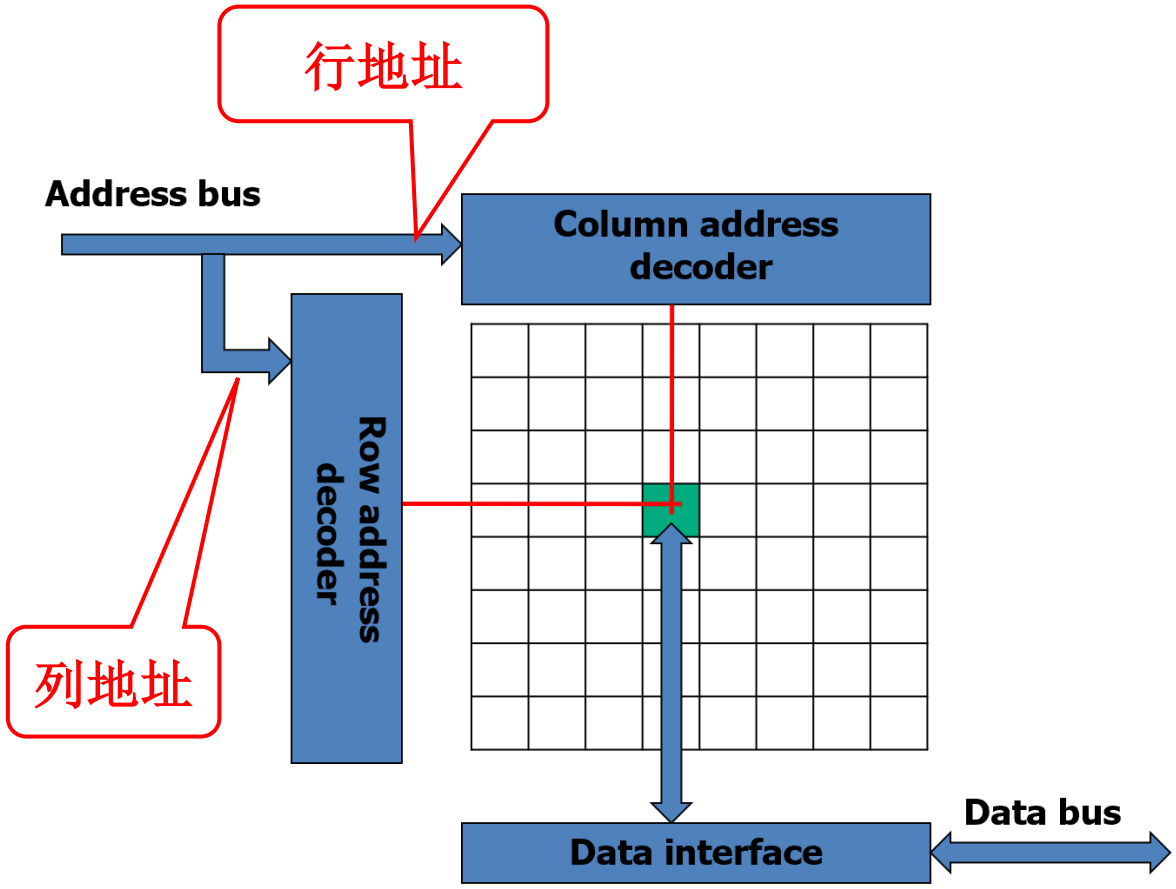
- 每个存储芯片是n*4bit
- 2个存储芯片一组，形成n byte
- 多个这样的组，构成一个二维阵列

Memory line access 存储器行访问



- 存储阵列中每行有m位，总共有2^n行。按行访问
- 2^n需要n位地址
- 地址输入到MAR，给地址解码器解码，从2^n行中选择跟地址对应的行，该行的m个bit的数据到MBR中

Memory grid access 存储器网格访问

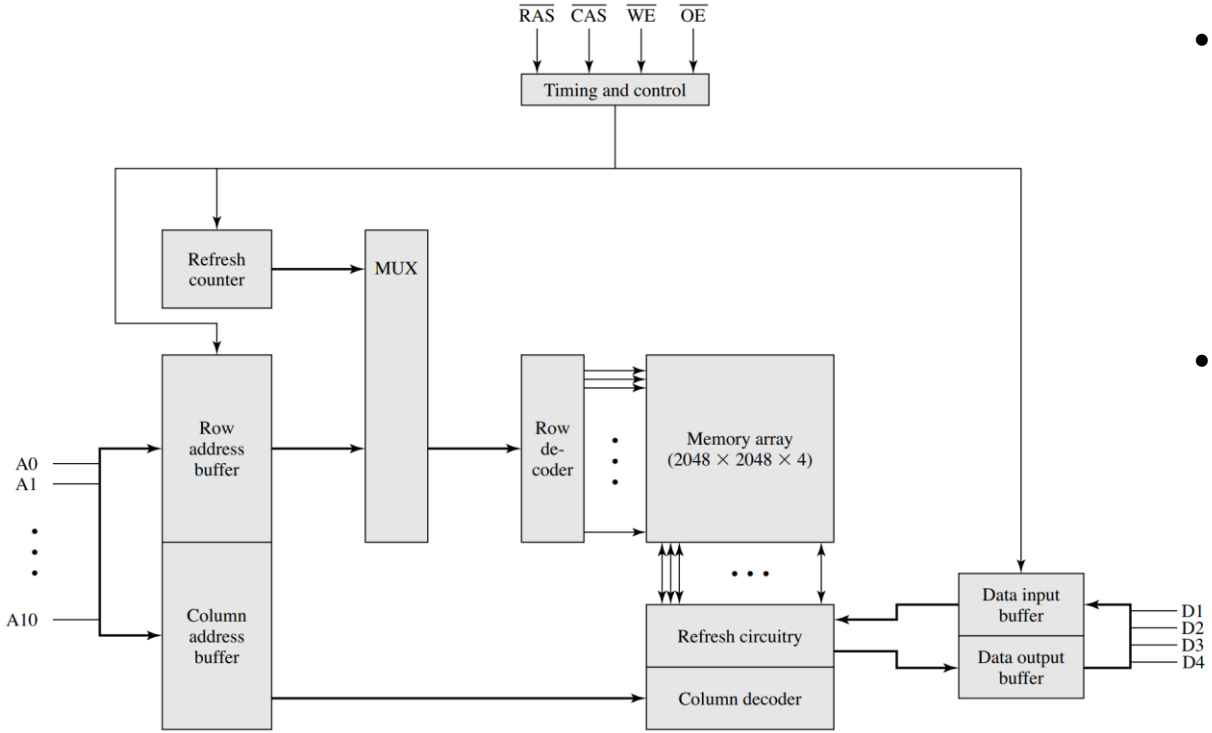


- 在网格式的存储器组织方式中，需要提供行地址和列地址
- 每个存储单元可能会存储多个bit。地址总线会把行地址和列地址分别给行地址解码器和列地址解码器
- 行地址解码器和列地址解码器解码后，选中对应的存储单元，将数据传送给数据接口，并送到数据总线上

Organization in detail 存储器组织细节

- 1M words, 16bit of each word can be organized in two ways 1M个字，每个字16bit的存储单元可以有两种组织方式
  - A bit per chip system has 16 lots of 1Mbit chip with bit 1 of each word in chip 1 , and so on 16个1Mbit的芯片，每个字的第一个bit在芯片1上，如此类推
  - Two 1M * 8bit chips, the high 8 bits of each word are on one chip, and the low 8 bits are on the other chip 2个 1M*8bit的芯片，每个字的高8位在一个芯片上，低8位在另一个芯片上
- A 16Mbit chip can be organised as a 2048 x 2048 x 4bit array 16Mbit的芯片也可组织成 2048*2048*4bit的阵列
  - Reduces number of address pins 减少地址引脚的数量
  - Multiplex row address and column address 复用行和列地址
  - 11 pins to address (2^{11}=2048) 11个地址引脚
  - Adding one more pin doubles range of values so x4 capacity 增加1个引脚，容量*4

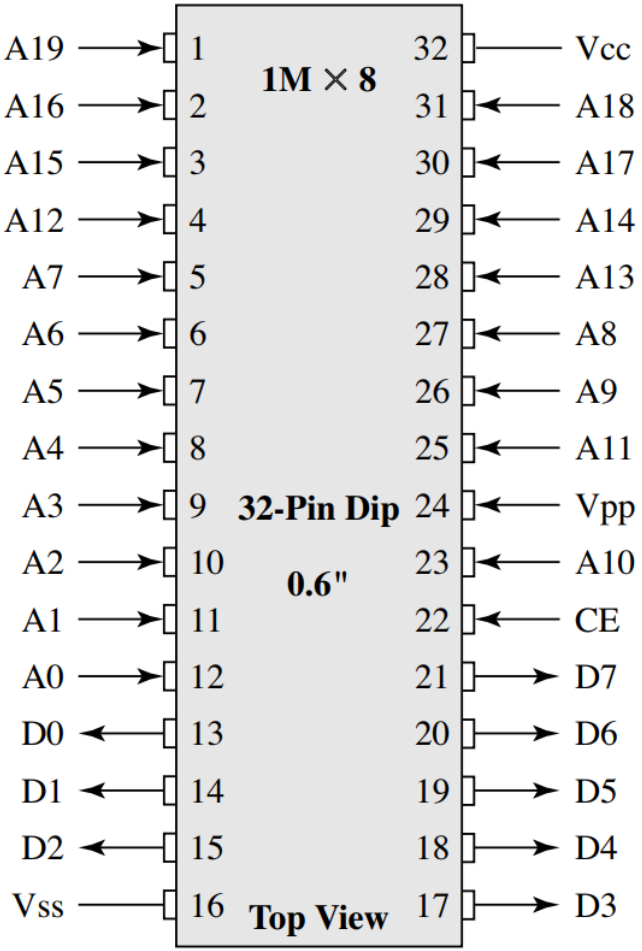
Typical 16Mb DRAM 典型的16Mb的存储器



- 典型的16Mb的DRAM的结构，按2K*2K*4的形式组成存储阵列
- 4M个访问单元，需要22个地址线。按照2k*2K的阵列来组织的，需要11个行地址、11个列地址

- 用11个地址线来分时作为行地址和列地址，RAS和CAS用于区分
- 先将行地址的11位送过来，由RAS确定是给行地址寄存器，通过行地址解码器解码成行地址；然后再给11位列地址，进行地址解码生成列地址

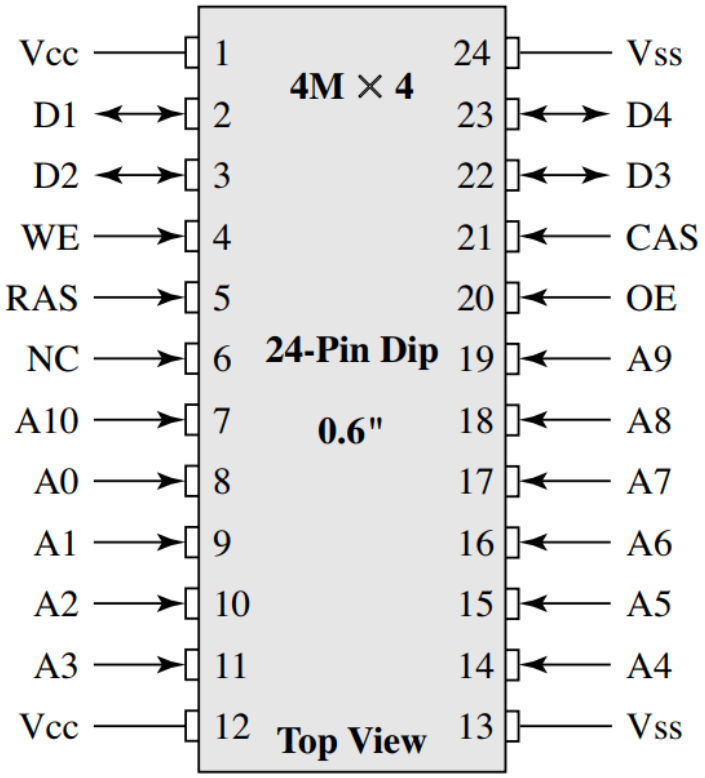
Chip packaging 芯片封装



8Mb EPROM，1M*8bit

- 集成电路封装成组件，然后加上引脚
- 8Mbit的EPROM的芯片，1M*8bit来组织的。1M个寻址单元，需要有20个地址线，在芯片上从A0~A19。
- 按字节来进行读取，有8个数据引脚
- 除了地址引脚和数据引脚之外，还有Vcc电源，Vss地线
- CE芯片允许引脚，表示该地址是否对这个芯片有效。还有一个Vpp，在EPROM写操作时提供

Chip packaging 芯片封装



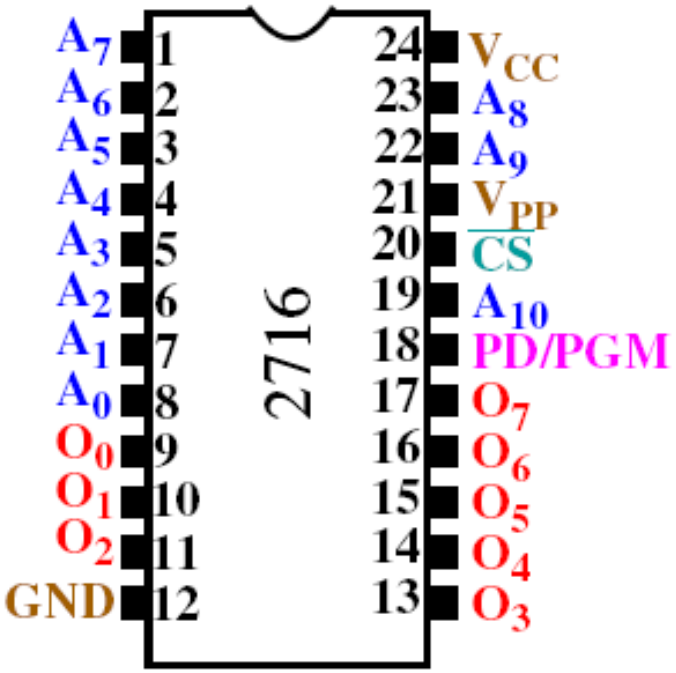
16Mb DRAM，4M*4bit

- 16Mbit的DRAM芯片，4M*4bit的结构，按4个bit来进行读取，总共有4个数据引脚
- 4M的寻址空间，需要22个地址线。为了节省引脚，采用11个行和11个列共用引脚的方式
- RAS和CAS确定当前的引脚是用作行还是列地址。A0~A10为地址引脚
- Vcc和Vss为电源和地线。WE和OE确定当前的操作是写操作，还是读操作。NC凑数，使引脚总数为偶数

Generic pin configuration 通用引脚配置

- Besides the address lines and data lines, each memory device has at least one chip select pin that enables the memory device. 除了地址和数据线之外，每个存储芯片至少有一个片选引脚，表示选中这个芯片
- Each RAM device has read or write control pin 每个RAM还需要有读写控制引脚
  - WE：Write Enable 写允许
  - OE：Output Enable 输出允许

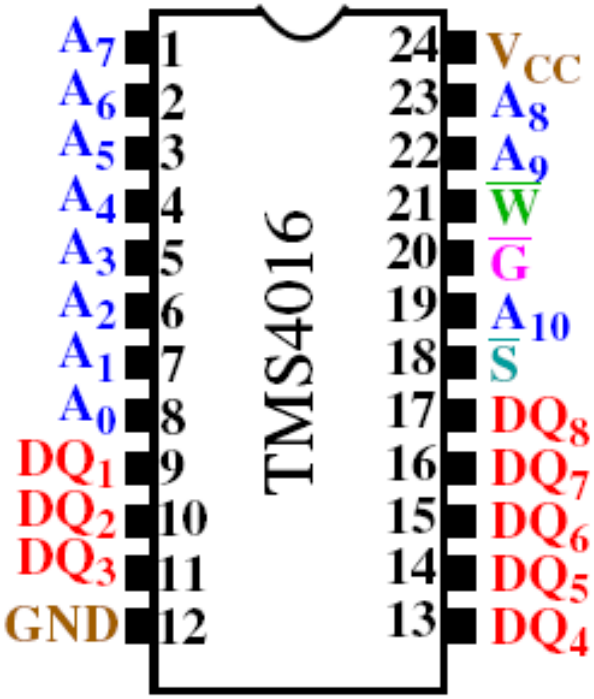
2K*8 EPROM chip 2K*8的ERPOM芯片



Pin(s)	Function
A_0-A_{10}	Address
PD/PGM	Power down/Program
$\overline{CS}$	Chip Select
O_0-O_7	Outputs

- 8个数据引脚
- 11个地址引脚
- 控制引脚，包括片选引脚，电源和接地引脚，写入控制引脚等
- Vpp引脚，写操作时使用

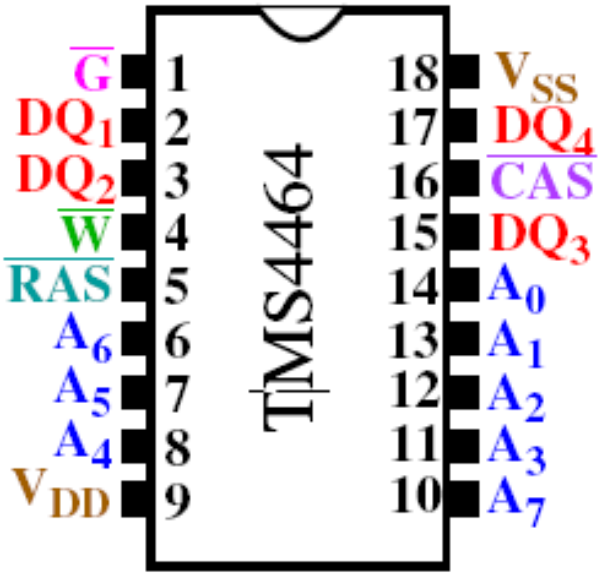
2K*8 SRAM chip 2K*8的SRAM芯片



Pin(s)	Function
A_0-A_{10}	Address
DQ_0-DQ_7	Data In/Data Out
$\overline{S}$ ($\overline{CS}$)	Chip Select
$\overline{G}$ ($\overline{OE}$)	Read Enable
$\overline{W}$ ($\overline{WE}$)	Write Enable

- 8个输入/输出引脚DQ1~8
- 11个地址引脚A0~A11
- 控制引脚，包括片选引脚
- 芯片是RAM，能够读写，还需要一个读允许引脚，一个写允许引脚。

64K*4 DRAM chip 64K*4的SRAM芯片



Pin(s)	Function
A_0-A_7	Address
DQ_0-DQ_4	Data In/Data Out
$\overline{RAS}$	Row Address Strobe
$\overline{CAS}$	Column Address Strobe
$\overline{G}$	Output Enable
$\overline{W}$	Write Enable

- 4个数据读写引脚
- 16位地址线
- 芯片采用256*256的阵列，行和列都需要8位地址，用8个引脚进行复用
- 控制引脚还包括了RAS和CAS，也就是行地址和列地址控制
- 还包括一个读允许引脚，一个写允许引脚

Example

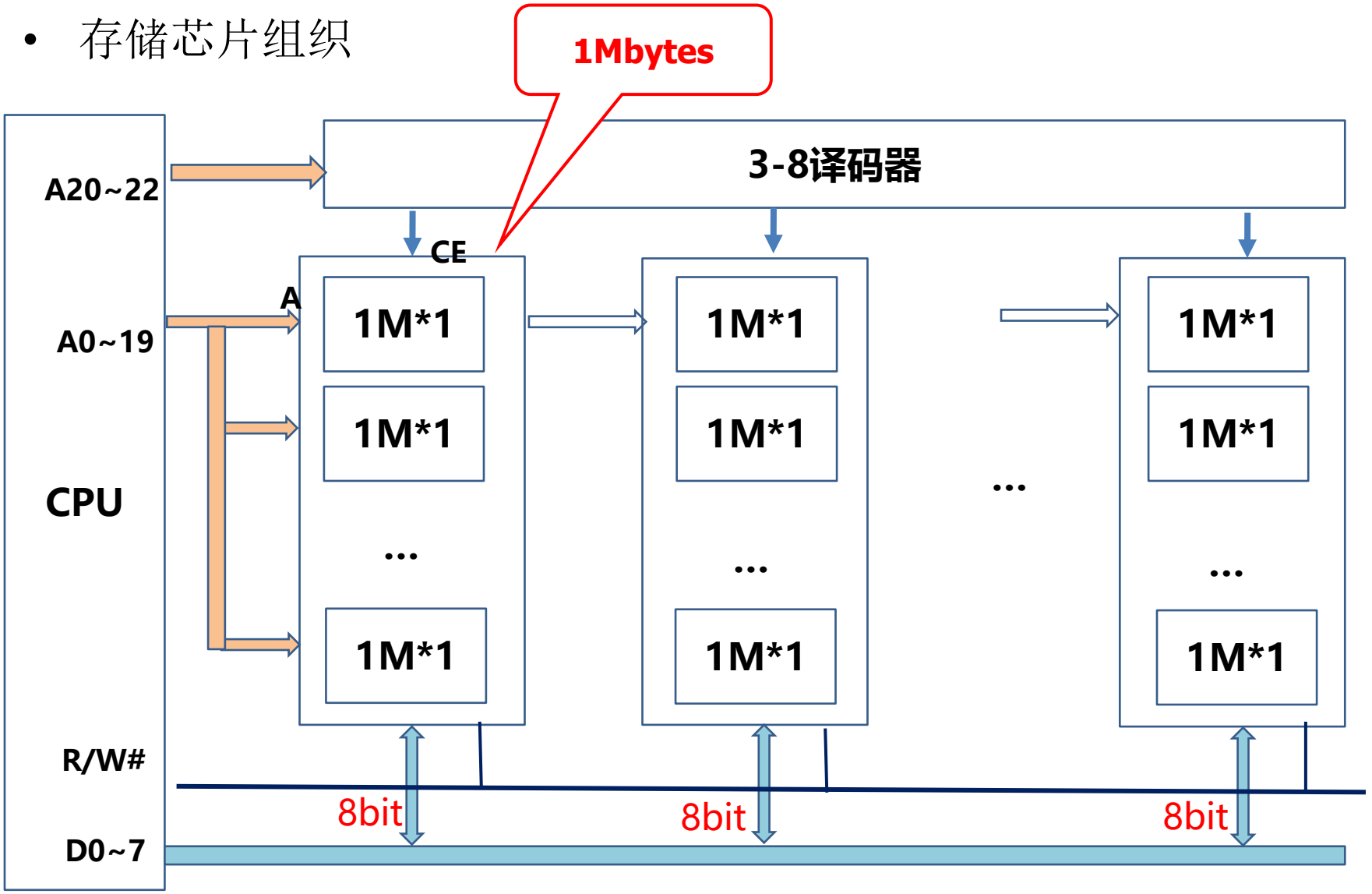
- 存储器芯片为1Mbit，需要组织成8MB的按字节寻址的存储系统。
  - 需要多少个这样的芯片？
  - 需要多少位存储器地址？
  - 存储芯片如何组织？

Example

- 芯片数量
  - 8Mbyte/1M bit=64片
- 存储器地址位数
  - 8Mbyte按字节存储，需要寻址8M个存储单元，所以需要23位地址

Example of Memory Organization

- 存储芯片组织



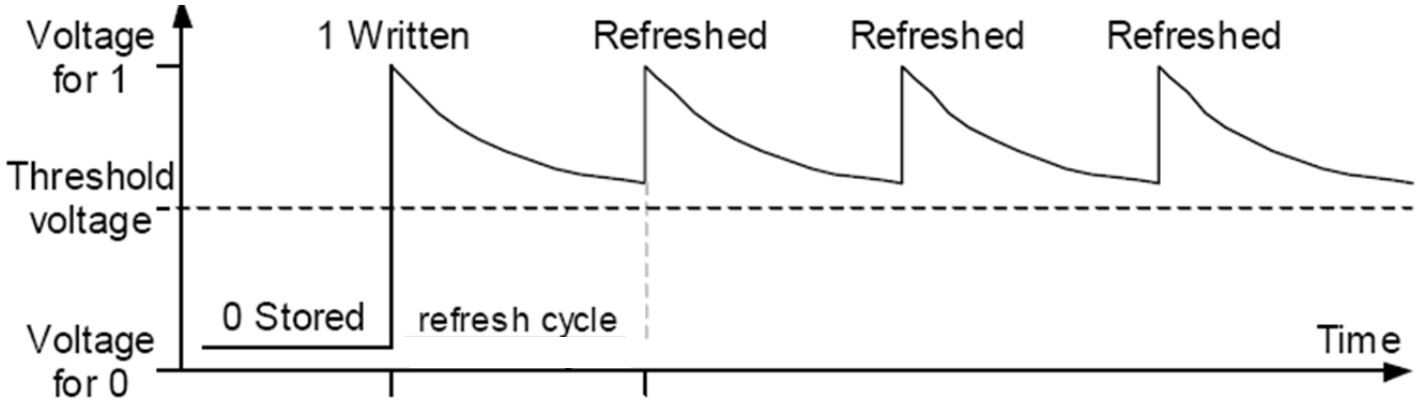
DRAM refresh and row access – 1 内存刷新和行访问

- DRAMs **MUST** be refreshed (rewritten) every 2 to 4ms DRAM必须每2~4ms刷新一次
- This refresh is performed by a special circuit in the DRAM which refreshes the entire memory 刷新由DRAM中的特殊电路完成，它刷新整个内存
  - Usually accomplished by a "RAS-only" cycle 通常由RAS-only完成
  - The row address is placed on the address lines and RAS (row address select) asserted. This refreshed the entire row 行地址放在地址线上，然后声明RAS信号，刷新一整行

DRAM refresh and row access – 2 内存刷新和行访问

- CAS (column address select) is not asserted 列地址没有插入
- The absence of a CAS phase signals the chip that a row refresh is requested 缺少列地址表示当前正在进行行刷新操作
- No data is placed on the external data lines 数据不用放到数据线上
- Most DRAM vendors also supply on-chip DRAM controllers that encapsulate the refresh and other functions 现在大多数DRAM供应商还提供片上DRAM控制器，用于进行DRAM的刷新和其他一些功能

DRAM refresh cycles and refresh rate 刷新示意图



写入1和后续刷新操作后，DRAM单元电容器上的电压变化。

- 写入1和后续刷新操作后，DRAM单元电容器上的电压恢复到正常值
- 需要定时去刷新，否则由于电容自然放电，导致存储的电荷丢失，存储的1就变成了0

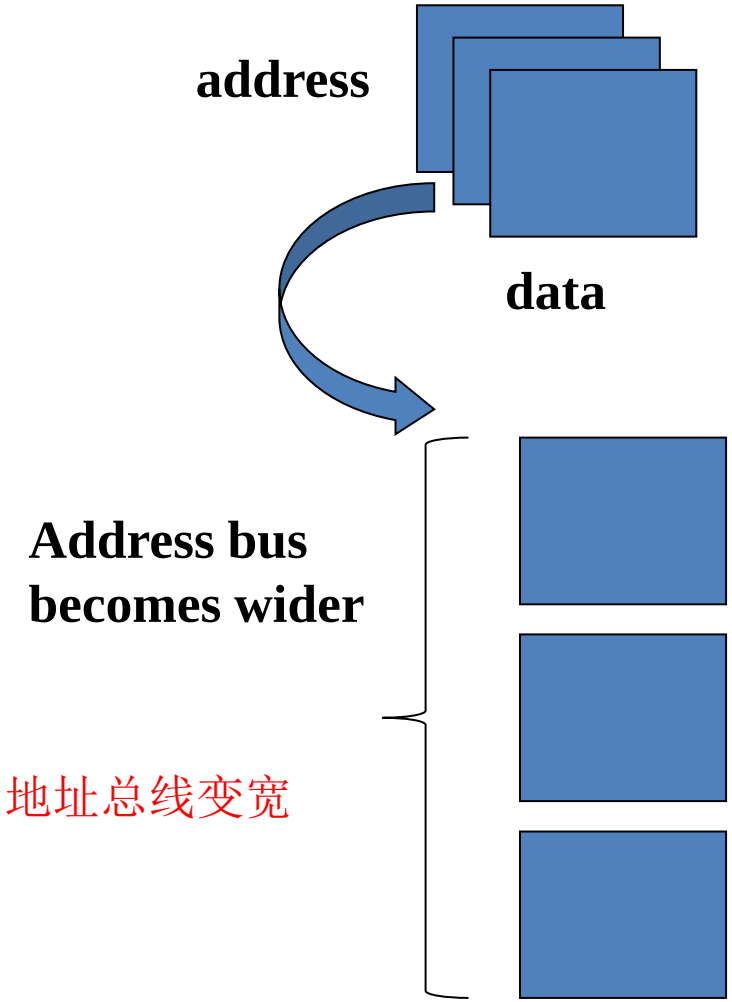
About Refresh

- DRAMs **MUST** be refreshed periodically DRAM必须周期性刷新
- Question: What is the effect of memory refresh？刷新的影响是什么？

Loss of Bandwidth to Refresh Cycles 刷新导致带宽损失

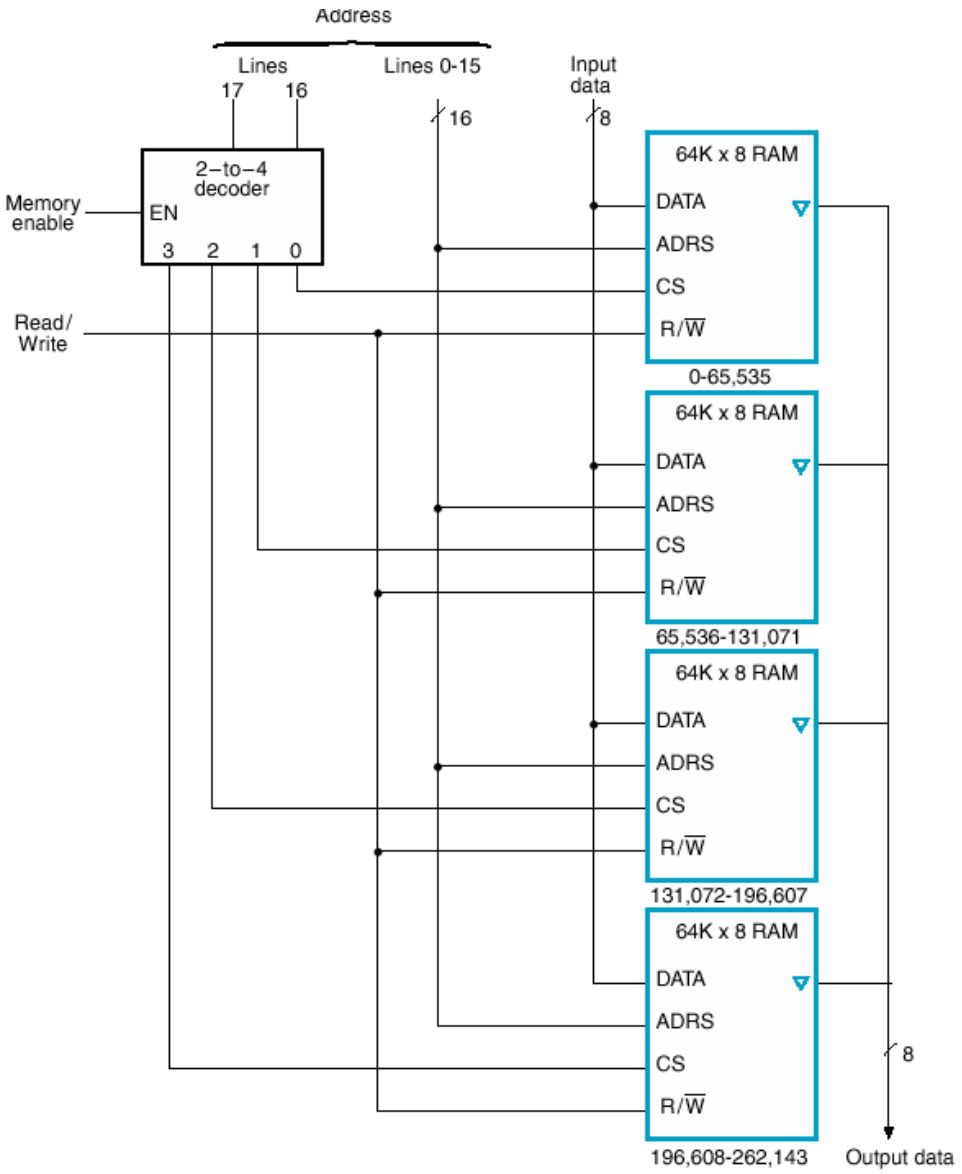
- Example：A 1Mb DRAM chip is organized as a 1K × 1K array. 1Mb DRAM组织成 1k*1k的阵列
- Rows must be refreshed at least once every 5 ms to forestall data loss; refreshing a row takes 100 ns. 每5ms必须刷新一次，刷新一行需要100ns
- What fraction of the total memory bandwidth is lost to refresh cycles? 由于刷新损失了多少带宽
- Refreshing all 1K rows takes 1024×100 ns = 0.1ms. 刷新1k行需要0.1ms
- Loss of 0.1ms every 5 ms amounts to 0.1/5 = 2% of the total bandwidth 每5ms损失了0.1ms，损失了2%

Expand memory to be larger 扩展内存使内存变大



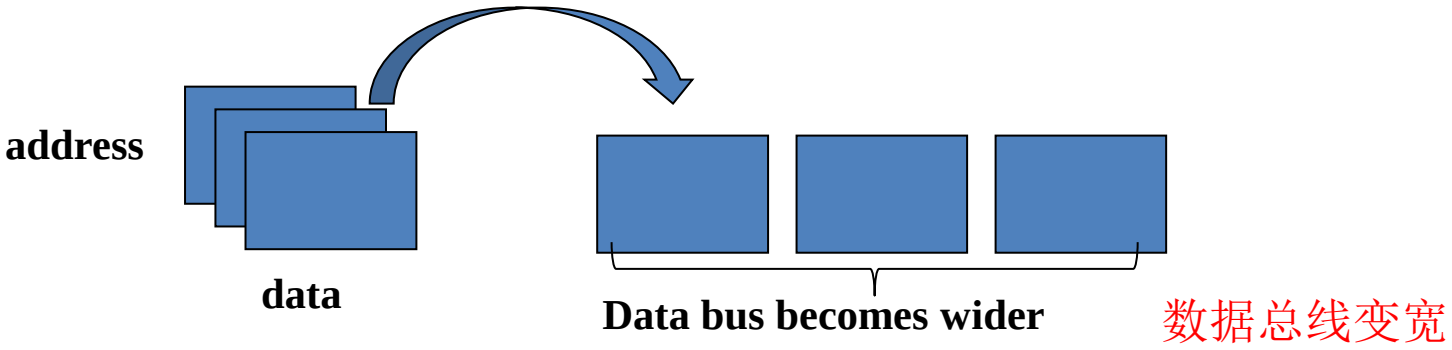
- 单个储存器芯片的容量有限，通过多个存储器芯片用某种方式来组合，以扩大存储器的总容量
- 选用多个存储器芯片，通过地址线来选择存储器芯片
- 地址线的一部分用于选择芯片
- 其余地址线选择芯片上的存储单元
- 可寻址空间变大

Expand memory to be larger 扩展内存使内存变大



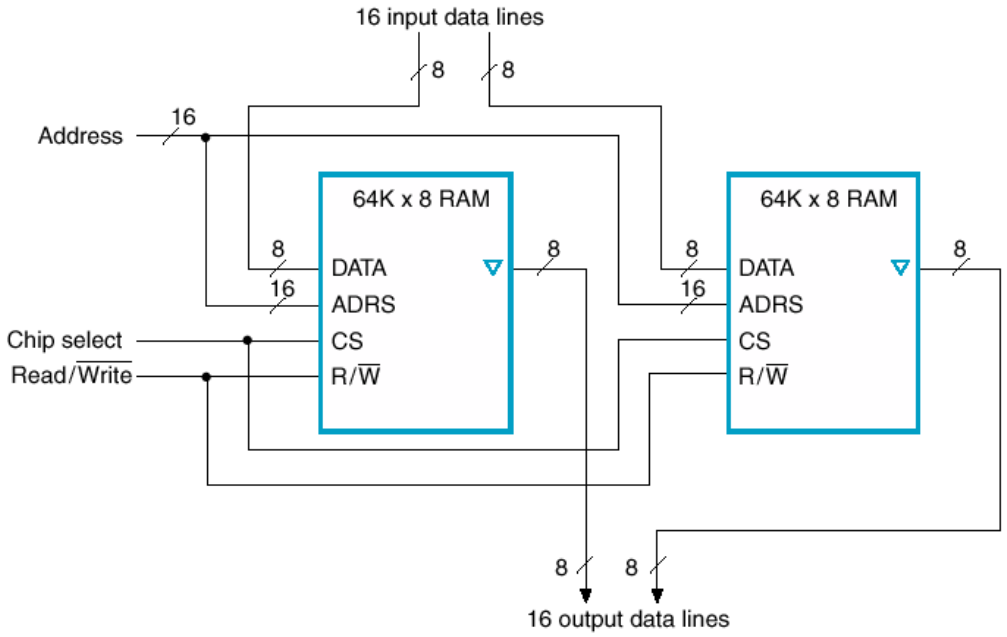
- 单个存储器芯片64KB。通过4片芯片，扩展成256KB的存储系统
- 地址线总共是18位，低16位给每个芯片，高2位用于选择芯片
- 高2位的地址送入一个2-4的解码器，解码后将信号输入给每个芯片的CS片选口
- 地址总线变的更宽，用增加的地址总线来选择存储芯片，寻址空间变的更大

Expand memory to be wider 扩展内存使内存变宽



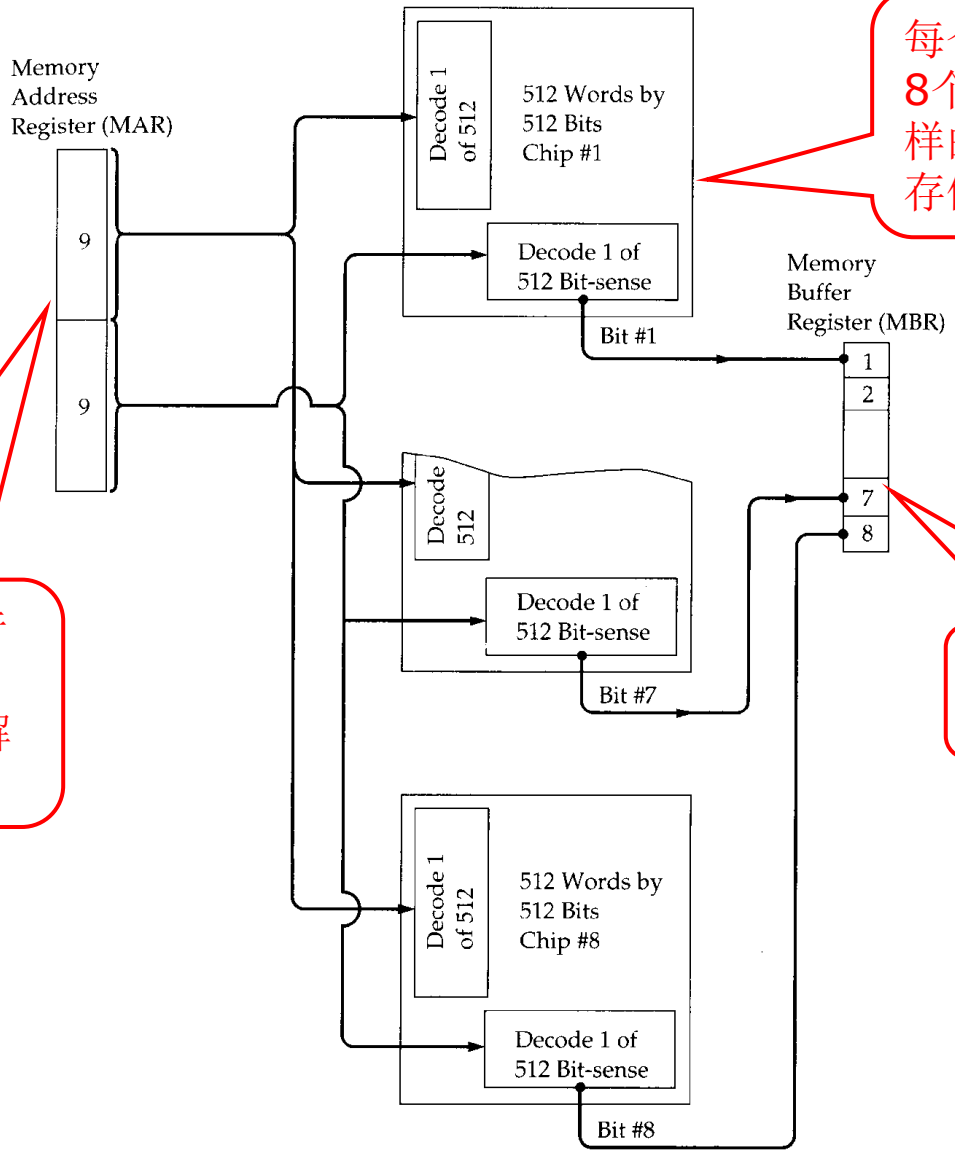
- 使用多个存储器芯片，将存储器的数据位数进行扩宽
- 地址线用于所有的芯片
- 每个芯片保存每次输出的一部分内容单元，每次输出的数据是所有芯片输出之和
- 数据总线变宽，存储容量变大（每个可寻址单元变大，寻址空间没有变大）

Expand memory to be wider 扩展内存使内存变宽



- 2个64K*8bit的存储芯片，组合成一个字长为16bit的存储系统（64k*16bit）
- 地址长度为16位，这16位同时送给2个芯片
- 数据位的高8位用左边的芯片存储，低8位用右边的芯片存储
- 地址空间没有扩展，但是数据总线的宽度扩展，存储容量扩大了

256k Byte module organization 256k byte模块构成

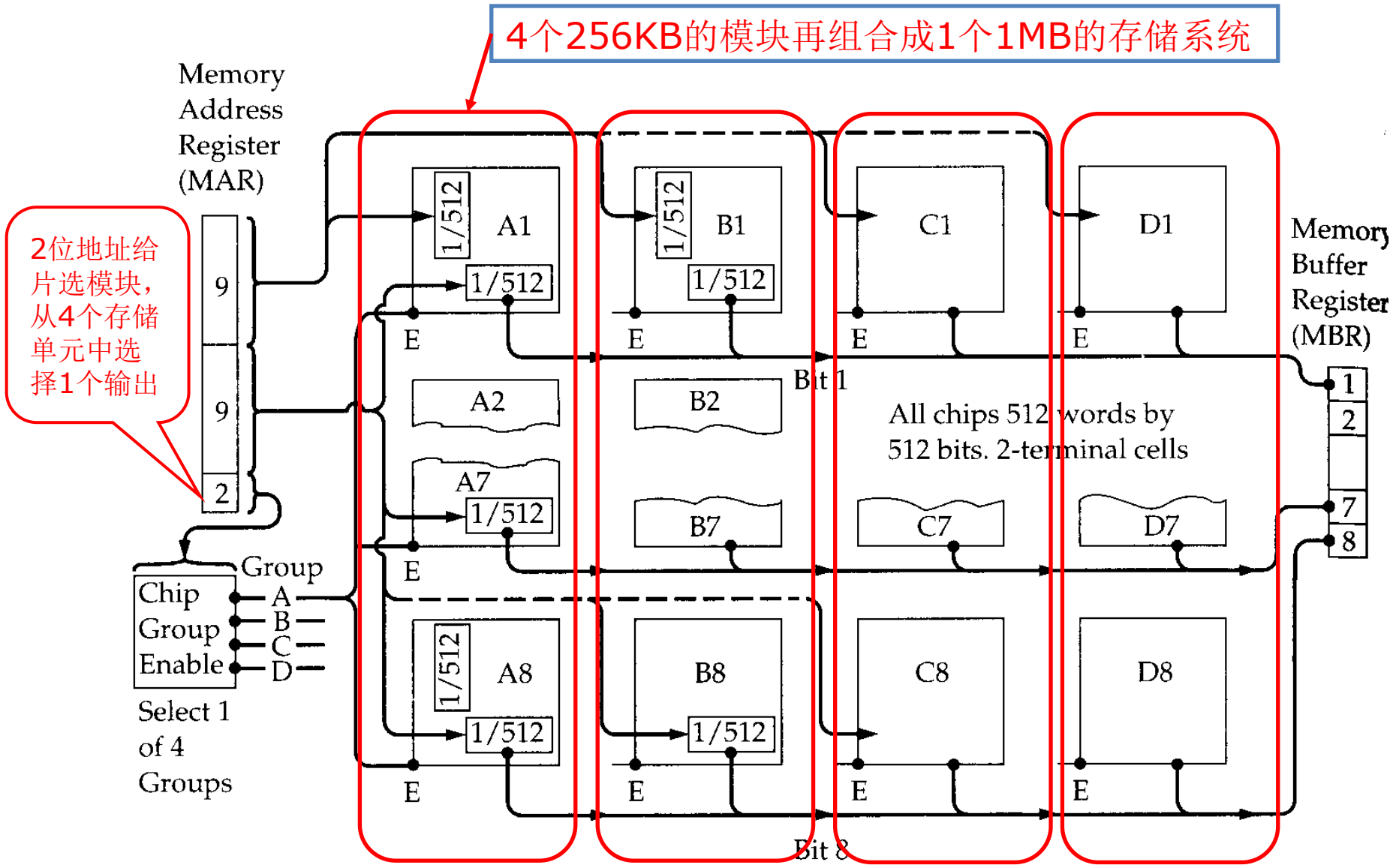


每个芯片包含256kb，8个这样的芯片通过同样的地址构成256kB的存储单元

18位地址，分为9个行地址和9个列地址，分别送给行解码器和列解码器

每个芯片贡献1位给MBR，总共8位

1M byte module organization 1M模块的组成

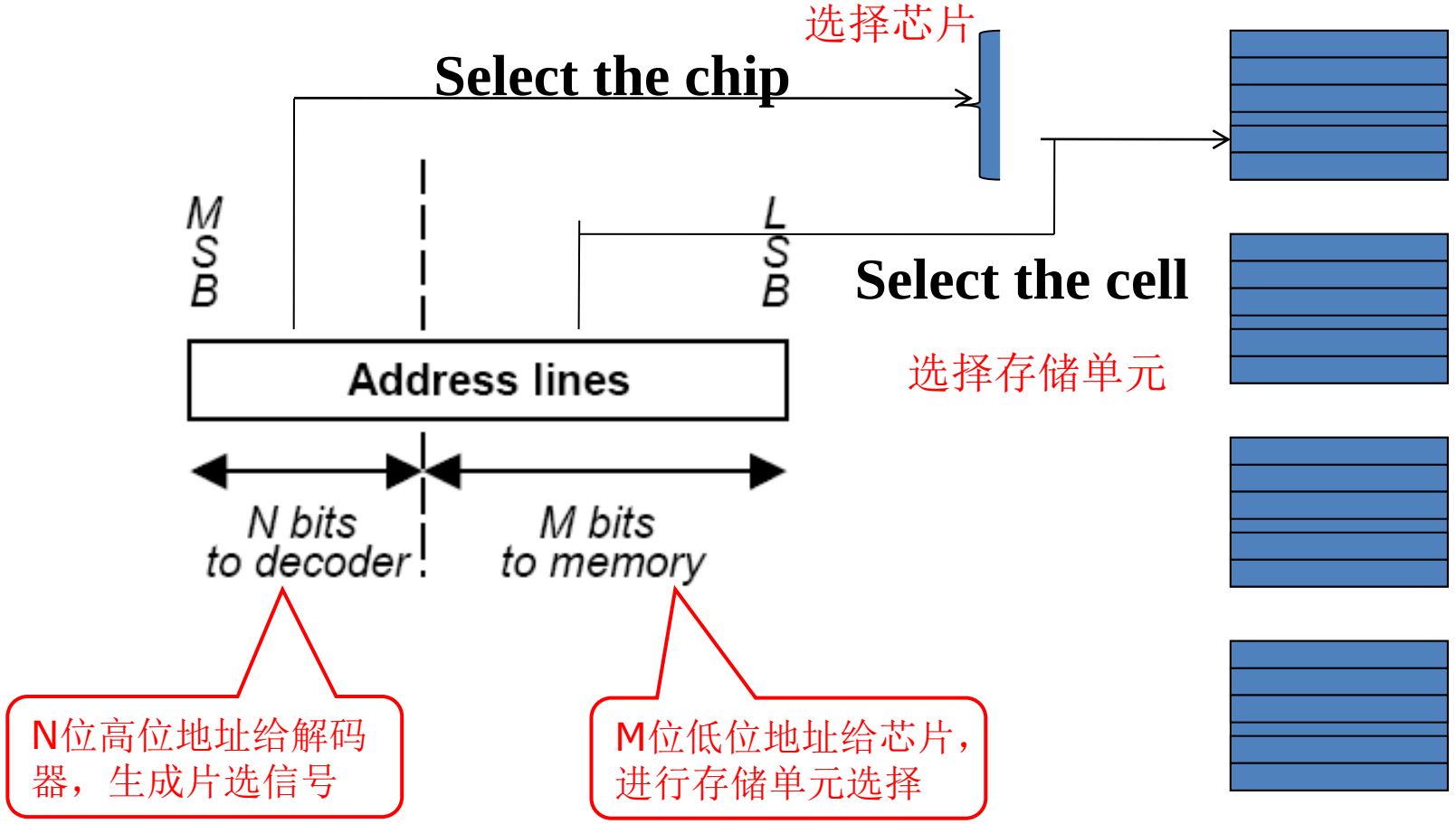


Memory address decoding 地址解码

- A memory system consists of a few memory chips 存储系统包含多个存储芯片
- How do these chips fit into the system address space? 这些存储芯片如何使用系统地址空间呢？
  - Decoding 解码
- Address decoding is the process of generating chip select (CS*) signals from the address bus for each device in the system 地址解码就是为每个设备生成片选信号的过程

Memory address decoding 地址解码

MSB：most significant bit 最高有效位
LSB： least significant bit 最低有效位



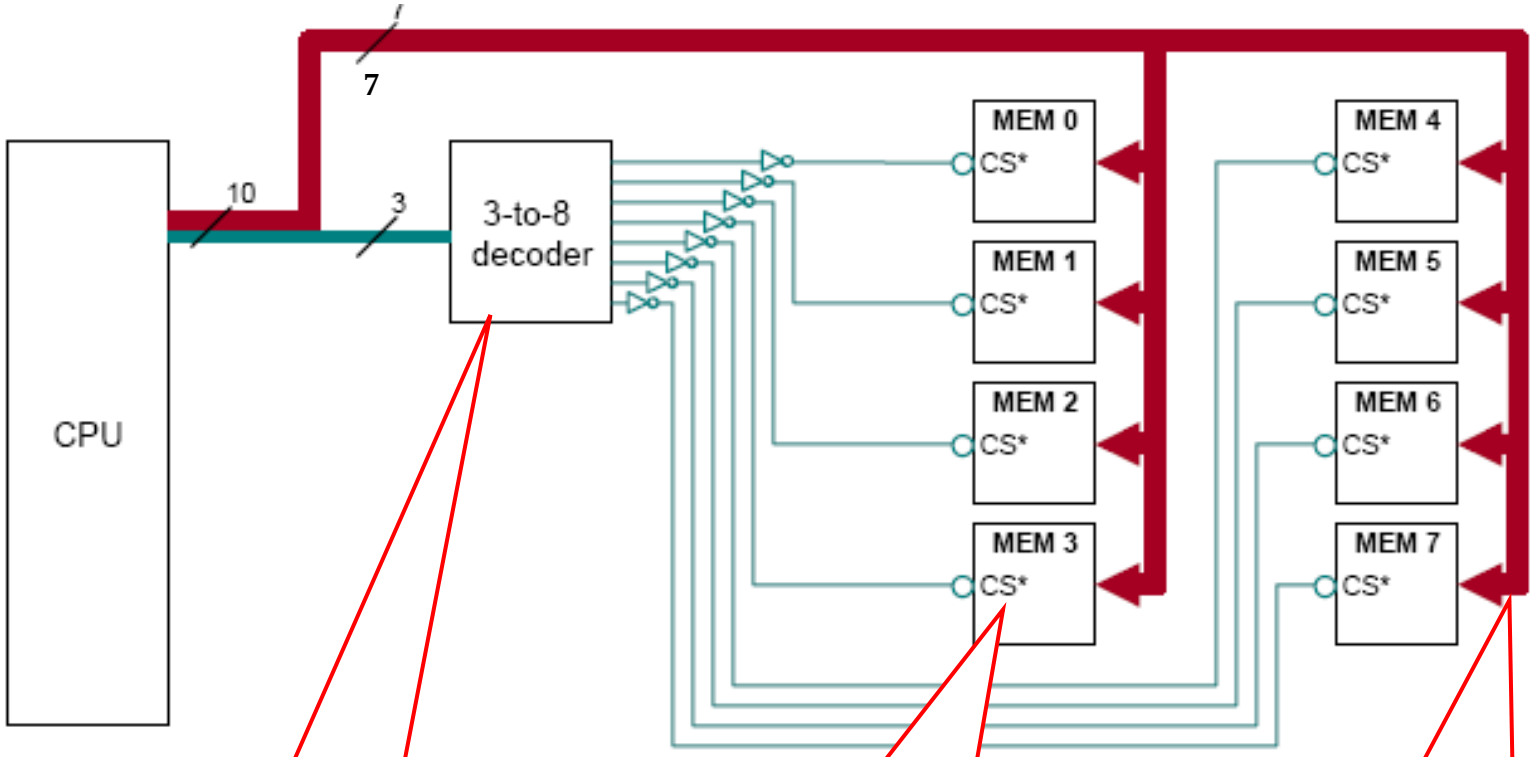
Example

- Assume a simple CPU need access 1KB memory locations 假定CPU访问1KB的存储空间
- We use 128x8bit memory chip 使用128*8bit 的存储芯片
- Question：
  - 需要多少个存储芯片
  - 总共需要多少根地址线，如何使用？

Example

- Assume a simple CPU need access 1KB memory locations 假定CPU访问1KB的存储空间
- We use 128x8bit memory chip 使用128*8bit 的存储芯片
  - 8 memory chips are needed 128x8 = 1024 需要8个存储芯片才能达到1024的存储空间
  - We need 3 address lines to select each of the memory chips 需要3个地址线来选择存储芯片
  - 7 LS address lines to select the cells within the chips 7个低位地址线来选择芯片内的存储单元

Example of memory address 存储地址举例



3位地址给3-8解码器，从8个存储芯片中选择一个芯片进行读取

选中这个芯片，就将这个芯片的CS信号设为有效，表示选中了这个芯片

其余7位地址同时给所有的芯片，进行片内存储单元的选择和读取

Two methods of decoding strategy 两种解码策略

- Full address decoding 全地址解码
  - All the address lines are used to specify a memory location 所有的地址线用于指定存储位置
  - Each physical memory location is identified by a unique address 每个物理存储位置都由唯一的地址来标识
- Partial address decoding 部分地址解码
  - If only a portion of the addressable space is going to be implemented, then not all the address space is implemented 如果只需要使用部分地址，就不需要实现全部的地址空间寻址
  - Only a subset of the address lines are needed to point to the physical memory locations 只需要地址线的一部分用于物理寻址

Example 举例

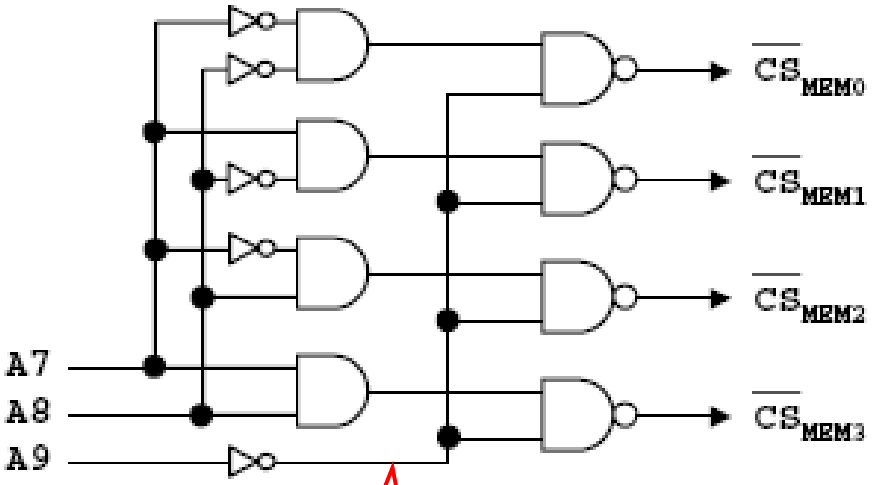
- Assume a simple CPU with 10 address lines that can access 1K memory locations 假定CPU有10个地址线，可以访问1k的地址
- However, this time we wish to implement only 512 bytes of memory 这时候我们只需要512的存储空间
- We still use 128x8 memory chip 我们还是使用128*8的芯片
- Physical memory must be placed on the upper half of the memory map 物理存储需要放在存储表的高半部分

Used for Address Decoding

Used to reference memory cells on each memory IC

Device	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
MEM 0	0	0	0	X	X	X	X	X	X	X
MEM 1	0	0	1	X	X	X	X	X	X	X
MEM 2	0	1	0	X	X	X	X	X	X	X
MEM 3	0	1	1	X	X	X	X	X	X	X

Example 举例



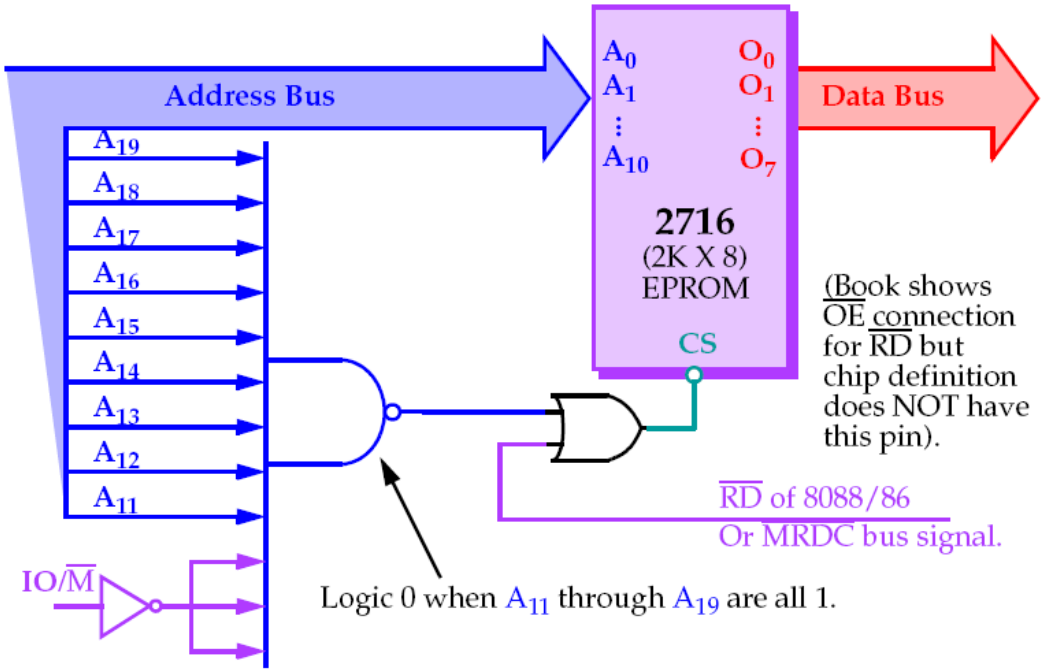
Memory map

MEM 0
MEM 1
MEM 2
MEM 3
Not used
Not used
Not used
Not used

A7、A8实际用于片选，
A9通过非门直接连接

只用了低位地址空间，
高位地址空间不用

Example 举例



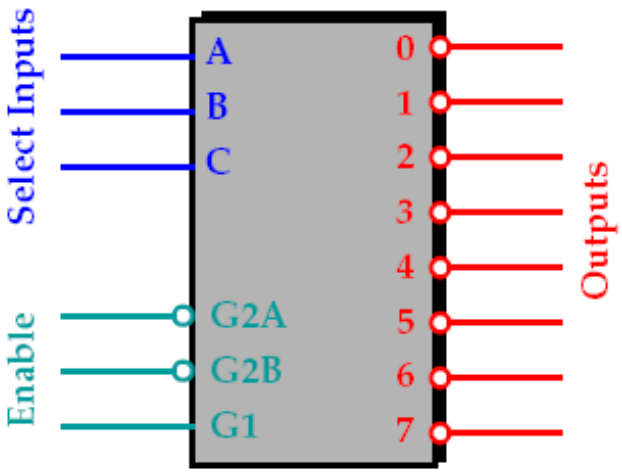
A19~A11 A10~A0

1 1 1 1 1 1 1 1 1 x x x x x x x x x x x x

1111 1111 1000 0000 0000 to 1111 1111 1111 1111 1111

- 2716芯片是2K*8bit的存储芯片，需要11位地址
- 高9位地址用于选择芯片，所以最多可以支持2^9=512个这样芯片
- 图上只用了从A12~A14的三位地址线，用于进行芯片的寻址，总共支持8个芯片。
- 存储系统的总容量就是16K byte

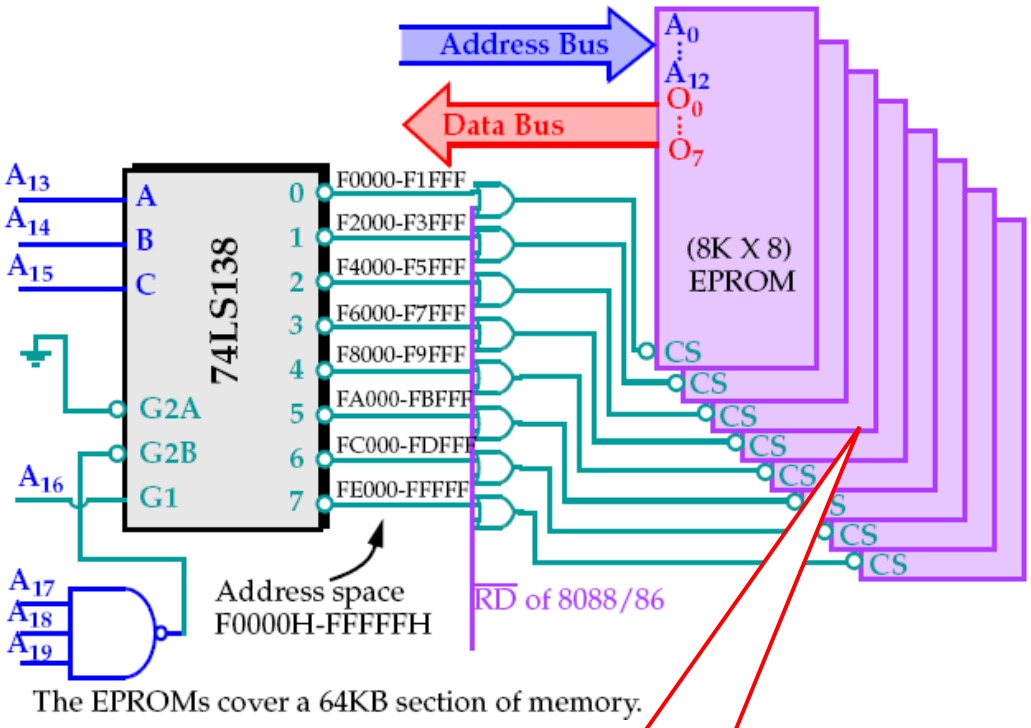
3-to-8 decoder (74LS138) 3-8解码器



Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	0	1	2	3	4	5	6	7
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

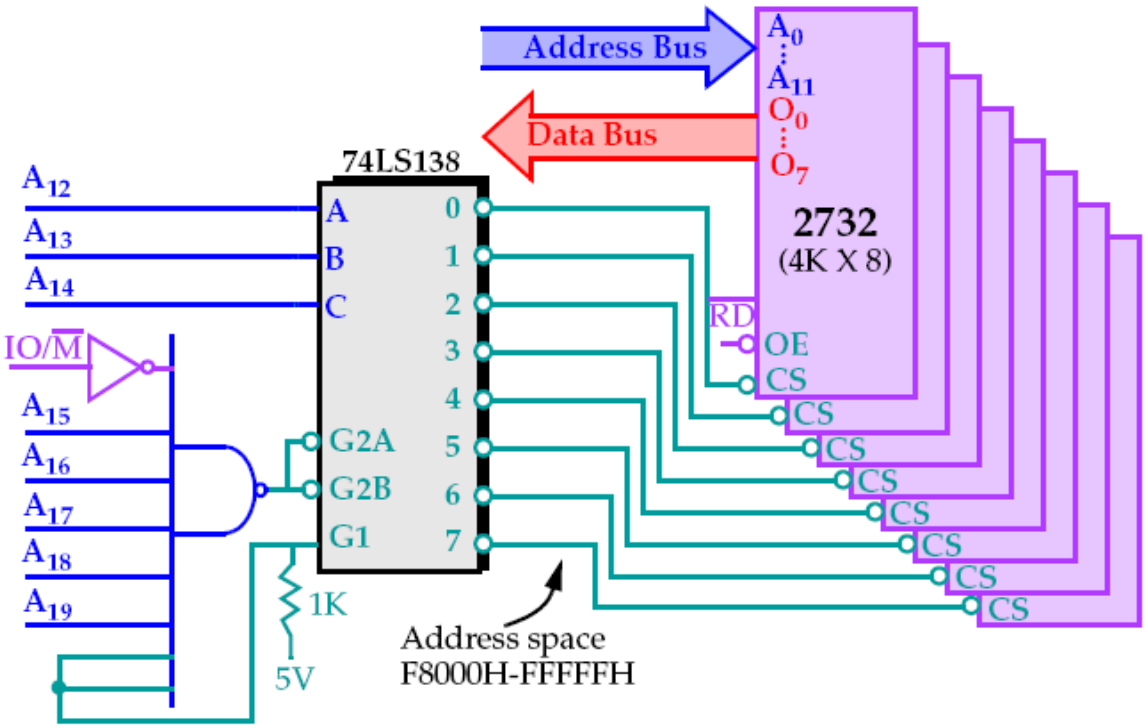
- 74LS138，3-8的解码器
- 输入为3，输出为8的解码器。用于3选8，也就是用3个地址线，选择8个存储芯片

3-to-8 decoder (74LS138) 3-8解码器的使用



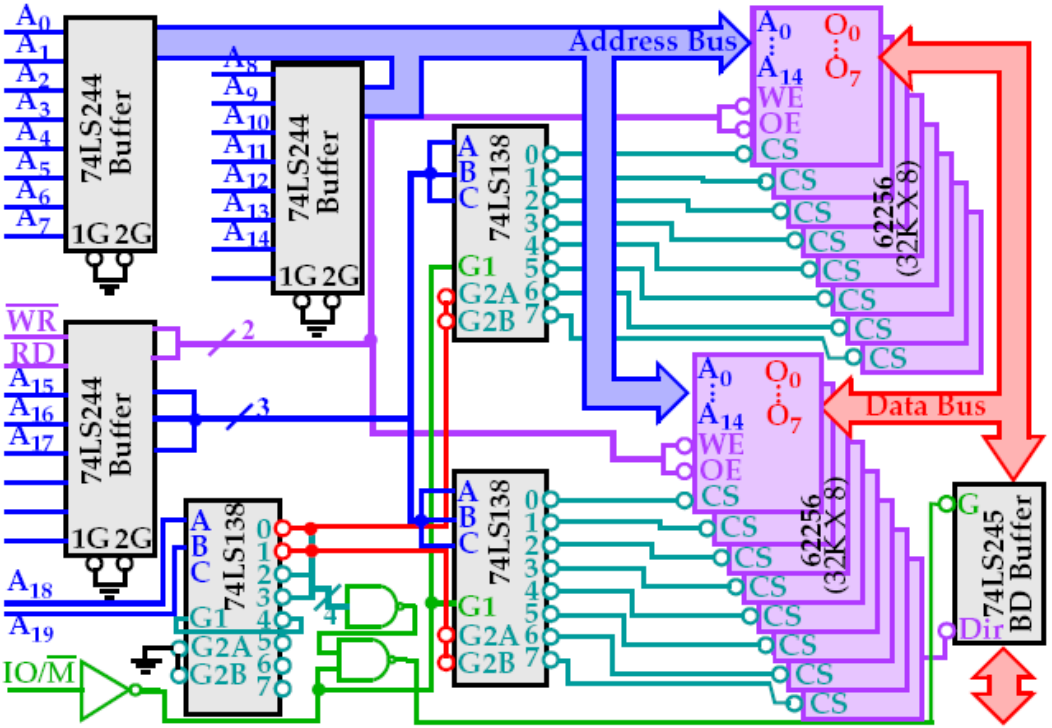
- 8个8Kbyte的存储芯片组合成一个64Kbyte的存储系统
- A13~A15通过74LS138，实现从8个芯片选择一个芯片
- A0~A12地址通过总线给存储芯片
- A16~A19实现了解码器的控制。

8088 EPROM 8088 EPROM



- 使用8个4KB的2732 芯片，构成一个32KB的EPROM
- A0~A11为片内地址
- A12~A14为片选地址
- A15~A19用于控制解码器

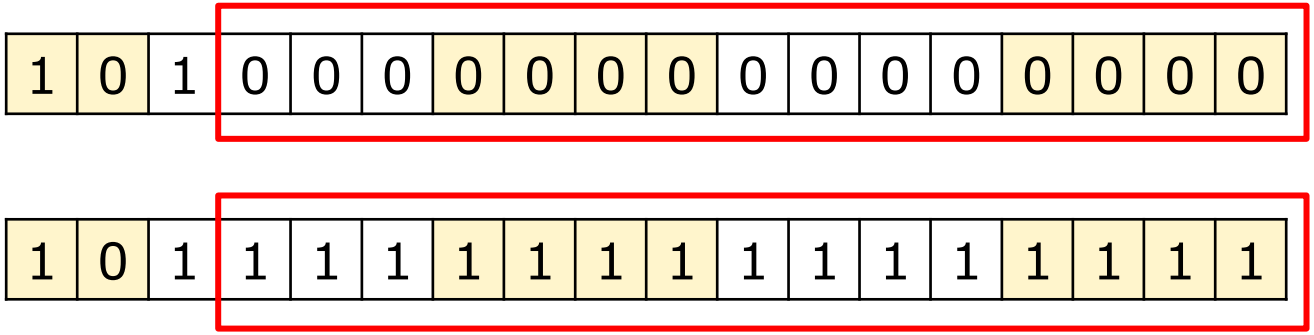
8088 RAM



- 最基础的存储芯片的容量是32KB
- 用这样8个芯片，构成32KB*8=256KB的存储单元
- 8088的字长是16位，用上面的2个256KB的存储单元，形成8088的存储系统。一个存储单元提供高8位数据，另一个提供低8位数据

Example of address 地址举例

- The chip select address is 101. what is the starting and ending address of 32Kx16 memory chip? 如果片选信号是101，32K*16的存储芯片，开始和结束地址是什么？
  - 32K space needs 15 address lines to access 32K地址，需要15位地址
  - Plus 3 chip select address lines 加上3位片选
  - Range 28000 to 2FFFF 地址从28000到2FFFF



Interleaved memory 交叉存储器

- Collection of DRAM chips 多个DRAM芯片的集合
  - Multiple RAM Grouped into memory bank多个RAM分组形成存储体
  - Multiple memory banks form a Interleaved 多个存储体形成多体交叉存储器
- Banks independently service read or write requests 存储体独立提供读和写的请求
- K banks can service k requests simultaneously K个存储库能同时提供服务K个请求
- Improve the response speed of the storage system 提高存储系统的响应速度

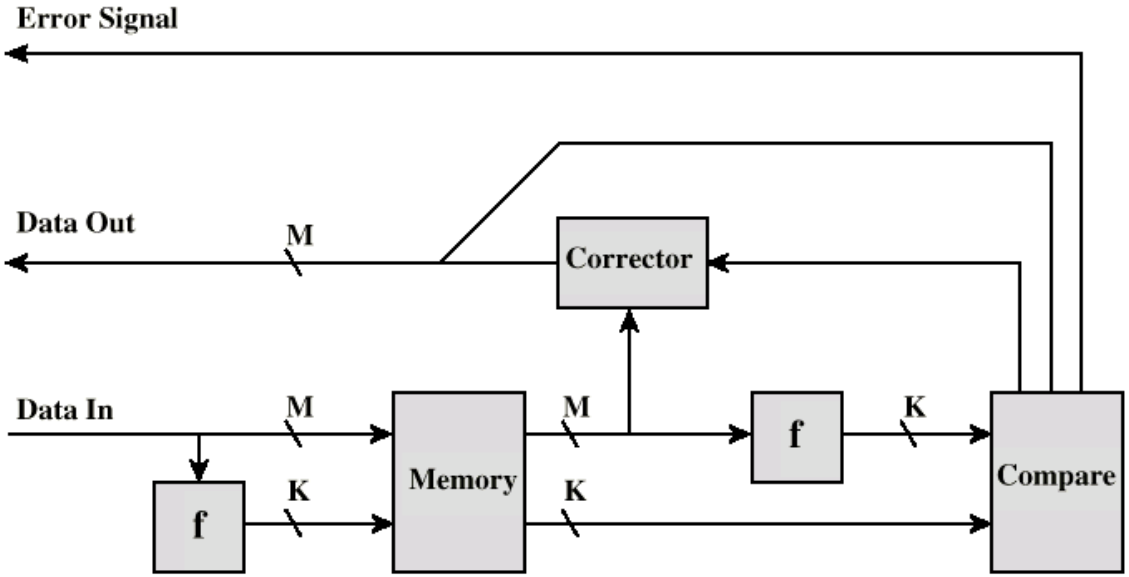
Outline

- Key Terms 关键术语
- Semiconductor main memory 半导体内存
- Error correction 纠错算法
- DDR DRAM 双倍速率DRAM
- Flash Memory 闪存

Error correction 纠错

- Semiconductor memories are subject to errors 半导体器件容易因为多种原因导致错误
  - Hard Failure: Permanent defect 硬件错误：永久性错误
  - Soft Error: Random, non-destructive, no permanent damage to memory 软件错误：随机的，非永久性错误
- Most modern main memory systems include logic for both detecting and correcting errors 大多数现在存储系统包含检错和纠错的逻辑
  - Fault tolerance of the system is improved 系统容错能力提高
  - Width of memory word increased 增加了存储字的宽度

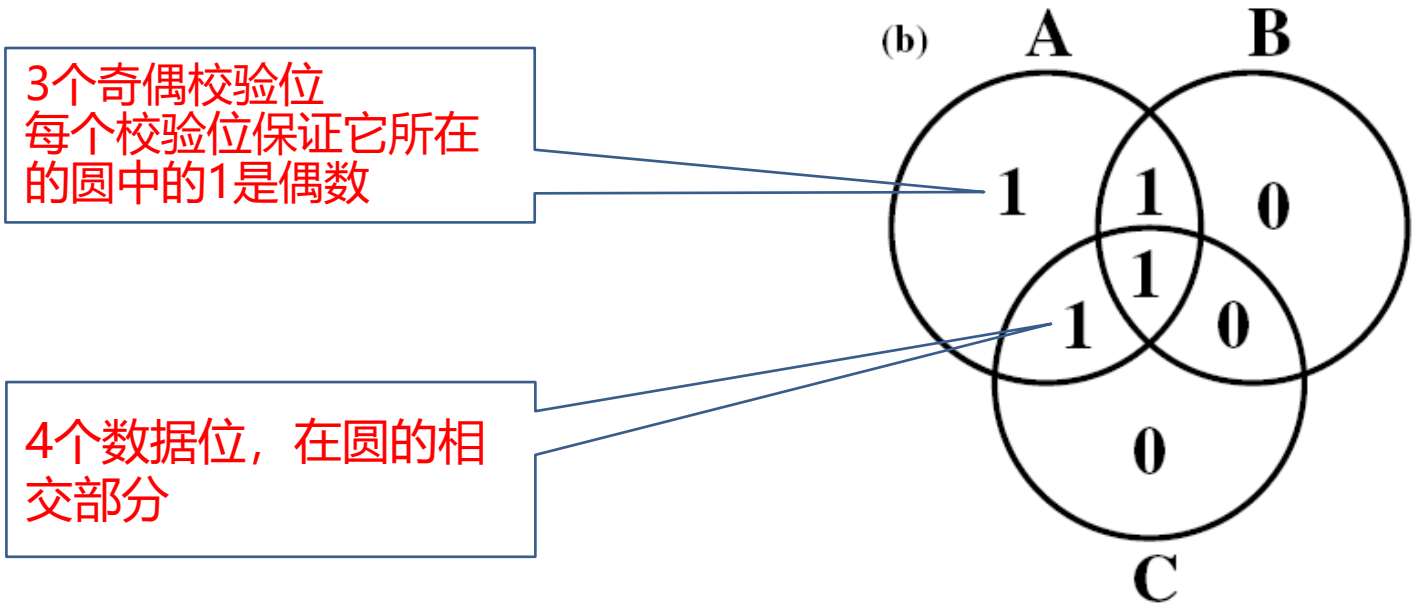
Error correcting code function 检错功能实现



- 原始数据M通过函数f计算得到校验码K
- 校验码K和M同时存入到存储器中
- 输出时，用存储器中存储的M，再次通过函数f进行计算，得到K’ ，和存储的K进行比较
  - 没有错误，直接将存储的M输出
  - 有错误，但是可以纠错，将存储的M和K进入纠错器，纠错后将正确数据输出
  - 有错误，并且无法纠错，直接输入错误信号

Hamming Error-correcting code 汉明纠错码

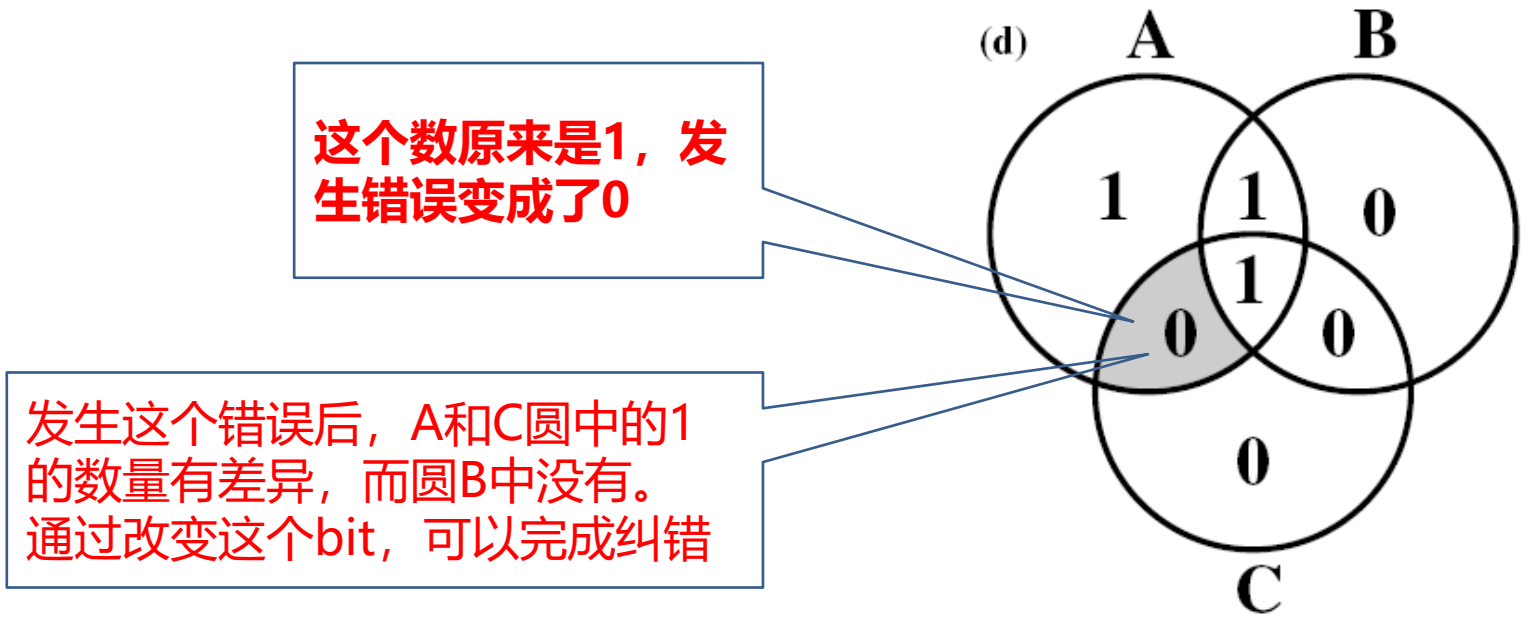
- Invented by Richard Hamming of Bell Laboratories 贝尔实验室的理查德.汉明发明
- The simplest error-correcting codes is the Hamming code 最简单的纠错码是汉明码



Hamming Error-correcting code 汉明纠错码

- If an error changes one of the data bits, it is easily found

  如果发生了1个错误，很容易发现

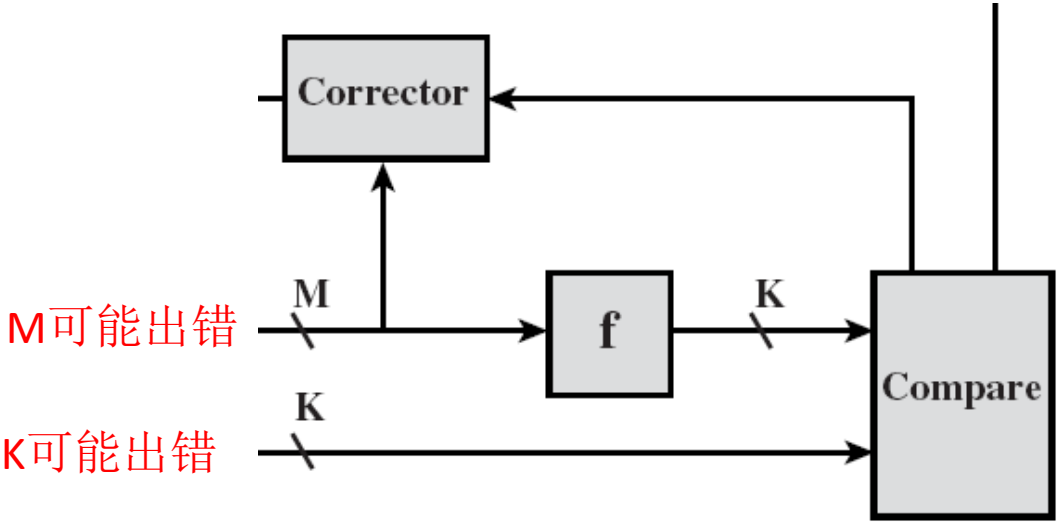


Choosing width of syndrome 纠错码的位数选择

- Hamming Code uses three error correction codes to correct four original data, and can only correct one bit error 汉明码使用了3个纠错码来对4个原始数据进行纠错，并且只能纠正一位错误
- For data with more bit length, how many bits of error correction code do you need? 针对位长更多的数据，纠错码的位数需要多少呢?
- M bit data，K bit Syndrome word 数据位数M，纠错码位数为K
- The original error correction code K is compared with the error correction code K 'generated from the data bit through the function 原始纠错码K和通过函数对数据位生成的纠错码K’进行比较
  - 0 = there is a match 0，表示匹配
  - 1 = there is not a match 1，表示不匹配

Choosing width of syndrome 纠错码的位数选择

- For the case that K and K 'do not match, there are $2^K - 1$ cases where K "is not 0 对于K和K’ 不匹配的情况，比较的结果K”不为0的情况有$2^K - 1$ 种
- It is required that $2^K - 1$ can indicate which bit of M and K has changed 要求这$2^K - 1$ 能够指示M和K的哪一位发生了变化



Choosing width of syndrome 纠错码的位数选择

- Because an error could occur on any of the M data bits or K check bits, we must have 因为错误可能发生在M位数据，也可能发生在k位纠错码上，所有需要满足如下公式：
  - $2^K - 1 >= M + K$
  - M: the number of data bits M是数据位
  - K: the number of check bits K是纠错位
- Both the code and the data must be stored in memory 数据和纠错码都要保存
- If an M-bit word of data is to be stored and the code is of length K bits, then the actual size of the stored word is M+K bits 如果数据位数为m，纠错码的位数为k，那么保存在存储器中的位数为m+k

Increase in word length 增加的字长

单纠错

单纠错双检错

	Single-Error Correction		Single-Error Correction/ Double-Error Detection	
Data Bits	Check Bits	% Increase	Check Bits	% Increase
8	4	50	5	62.5
16	5	31.25	6	37.5
32	6	18.75	7	21.875
64	7	10.94	8	12.5
128	8	6.25	9	7.03
256	9	3.52	10	3.91

- 根据上述公式，可以得到数据位长为不同情况下的纠错码的位数。
- 数据位数越多，需要的纠错码位数越长，但是占比越小
- 太长的数据位数，也会带来额外的问题，计算时间长，多位数据出错几率大

Layout of Data Bits and Check Bits 数据和校验位布局

Hamming error correcting code 汉明纠错码

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check bit					C8				C4		C2	C1

Figure 5.9 Layout of Data Bits and Check Bits

$$
\begin{aligned}
C1 &= D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7 \\
C2 &= D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7 \\
C4 &= D2 \oplus D3 \oplus D4 \oplus D8 \\
C8 &= D5 \oplus D6 \oplus D7 \oplus D8
\end{aligned}
$$

- 纠错码的位置在1、2、4、8这四位
- 计算C1的所有数据位，它的位置的最后一位都是1
- 计算C2的数据位，它的位置的倒数第二位是1，如此类推

Example – 1 举例-1

- Assume that the 8-bit input word is 00111001, with data bit D1 in the rightmost position. The calculations are as follows: 假设8位数据位是00111001，数据位D1在最右边。校验位计算如下

$$C1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$$
$$C2 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$$
$$C4 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$$
$$C8 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

- Suppose now that data bit 3 sustains an error and is changed from 0 to 1.When the check bits are recalculated, we have: 假如数据位3发生了错误，从0变成了1。校验位重新计算的时候，我们得到如下结果：

Example – 2 举例-2

- When the new check bits are compared with the old check bits, the syndrome word is formed: 根据数据生成的新纠错码和老纠错码进行对比，得到故障字：

$$C1 = 1 \oplus 0 \oplus 1 \oplus 1 \oplus 0 = 1$$
$$C2 = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0$$
$$C4 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$
$$C8 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

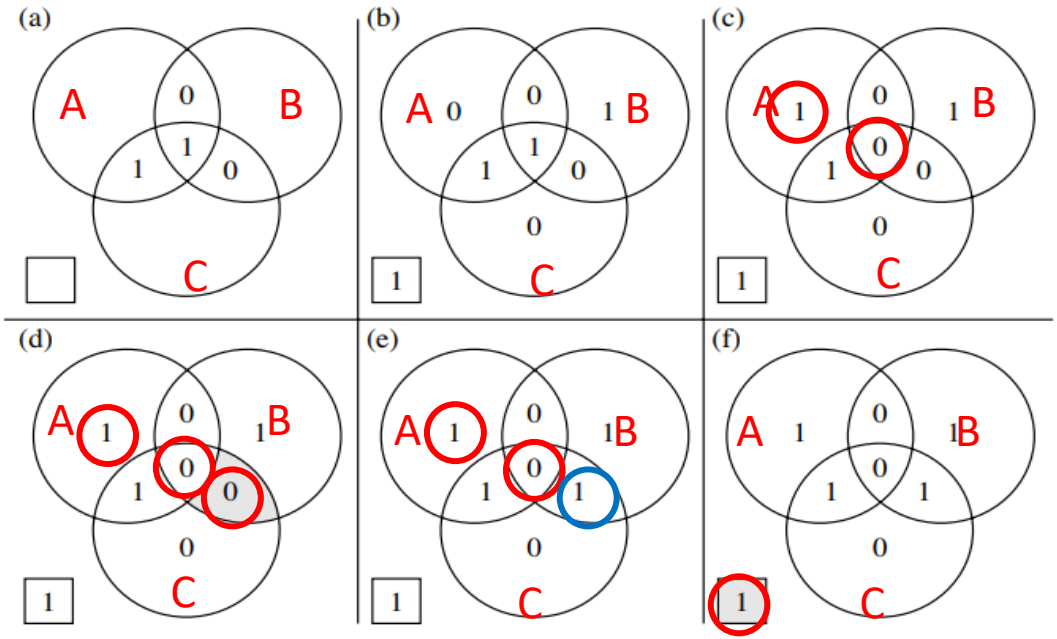
	C8	C4	C2	C1
	0	1	1	1
⊕	0	0	0	1
	0	1	1	0

- The result is 0110, indicating that bit position 6, which contains data bit 3, is in error. 故障字是0110，表示第6位，也就是数据位D3，发生了错误。

Steps of error correcting 纠错步骤

- 8 bits data, 4 bits code 8位数据位，4位纠错码
- Compare two error correction codes to generate syndrome 两个纠错码对比生成故障字
  - If all 0，no error 如果故障字全0，表示没有错误
  - If contains only one bit set to 1, an error has occurred in one of the 4 check bits. No correction is needed 如果故障字只包含1个1，那么错误只能是在4个纠错码中的1位。此时不需要纠错。
  - If contains more than one bit set to 1, then the numerical value of the syndrome indicates the position of the data bit in error. This data bit is inverted for correction. 如果故障字包含多个1，那么故障位的数据值能确定是哪个数据位出错了，这样对这个数据位取反就可以得到正确的值。

Hamming SEC-DED code 汉明单纠错双检错码



- 上面的方法只适合单错误，对于多错误，会发生什么？
- 图a是原始的4个数据位，图b是根据汉明纠错码生成的3个纠错码，分别是0、1和0。
- 如图c，发生了2个错误。一个错误是数据位，中间的数据位由1变成了0。另一个是纠错码，A圈的纠错码从0变成了1
- 如图d和e，认为是B和C交界的地方发生了错误，把这个0修改成了1。结果导致发生第三个错误
- 如图f，增加一个奇偶校验位，保证1的个数为偶数

Summary 小结

- Error correction is to improve its reliability by processing data and increasing its internal correlation 纠错就是通过对数据进行处理，增加其内部相关性来提高其可靠性
- In memory, SEC-DED （single-error-correcting, double-error-detecting）code is commonly used 在内存中，常用的是SEC-DED码
- The more data bits processed in a single time, the less space waste 单次进行处理的数据位数越多，空间损失越小

Outline

- Key Terms 关键术语
- Semiconductor main memory 半导体内存
- Error correction 纠错算法
- DDR DRAM 双倍速率DRAM
- Flash Memory 闪存

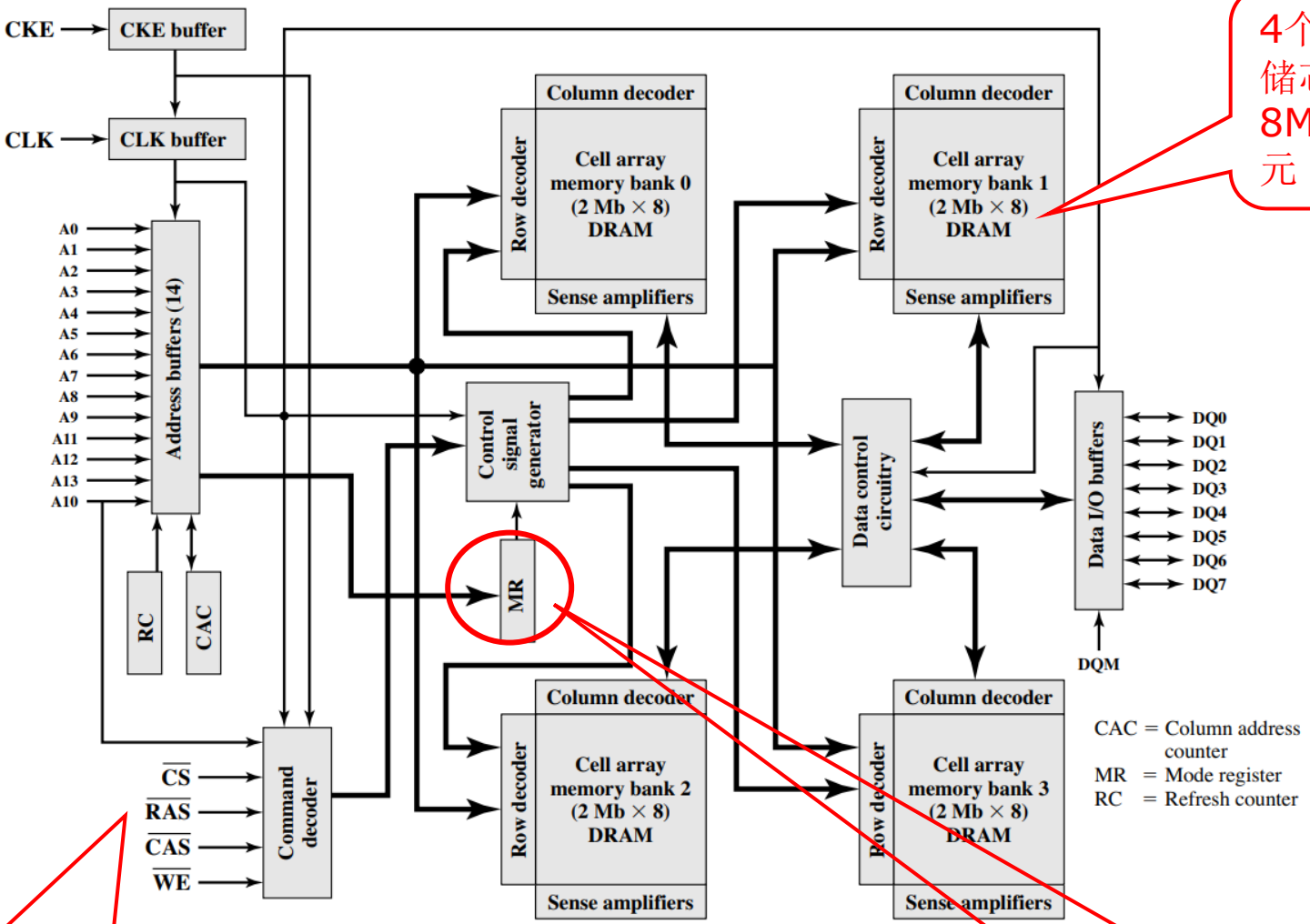
DRAM organization DRAM组织

- The performance of DRAM seriously affects the performance of the computer DRAM的性能严重影响了计算机的性能
- Cache can play a great role, but it is still not enough Cache虽然能发挥很大作用，但是仍然不够
- Many methods are proposed to improve the performance of memory itself 提出了很多方法来提高内存本身的性能
  - SDRAM
  - Rambus DRAM
  - DDR SDRAM
  - Cache DRAM

Synchronous DRAM (SDRAM) 同步DRAM

- Access is synchronized with an external clock 通过外部时钟同步访问
- Address is presented to RAM，RAM finds data 地址给RAM后，RAM去找数据
  - CPU does not have to wait, it can do something else. (CPU waits in conventional DRAM) CPU 不需要等待，可以去做其他事情。在传统的RAM中，此时CPU需要等待。
  - Since SDRAM moves data in time with system clock, CPU knows when data will be ready 由于SDRAM通过系统时钟工作，数据准备好之后，CPU马上就能知道
- Burst mode allows SDRAM to set up stream of data and fire it out in block 爆发模式允许SDRAM建立流数据并且按块发送

Diagram of SDRAM SDRAM结构图

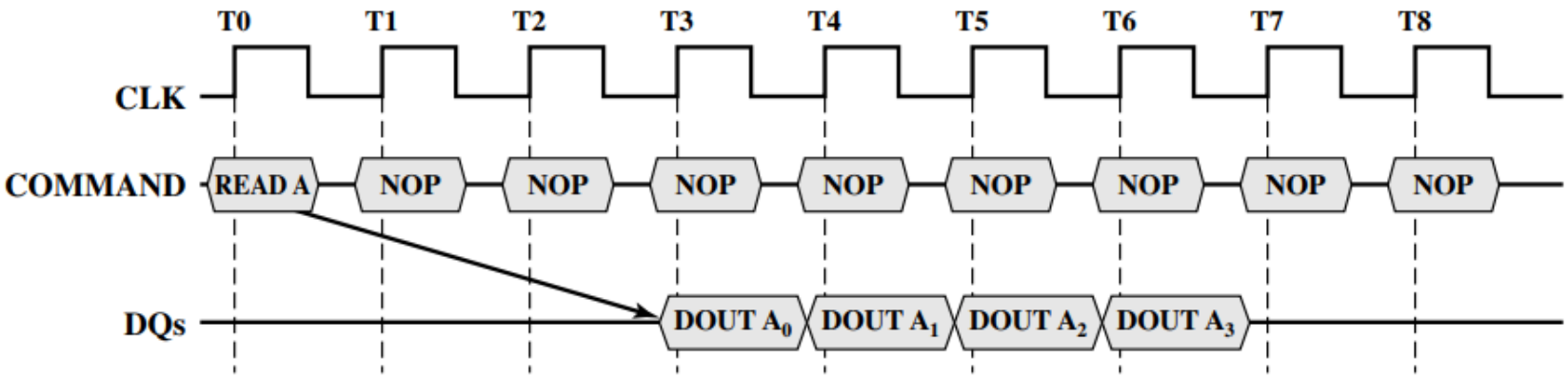


4个2MB的存储芯片，构成8MB的存储单元

地址线复用，分别进行行地址和列地址解码

采用爆发模式，提高存取效率。MR为模式寄存器

SDRAM read timing SDRAM读取时序图

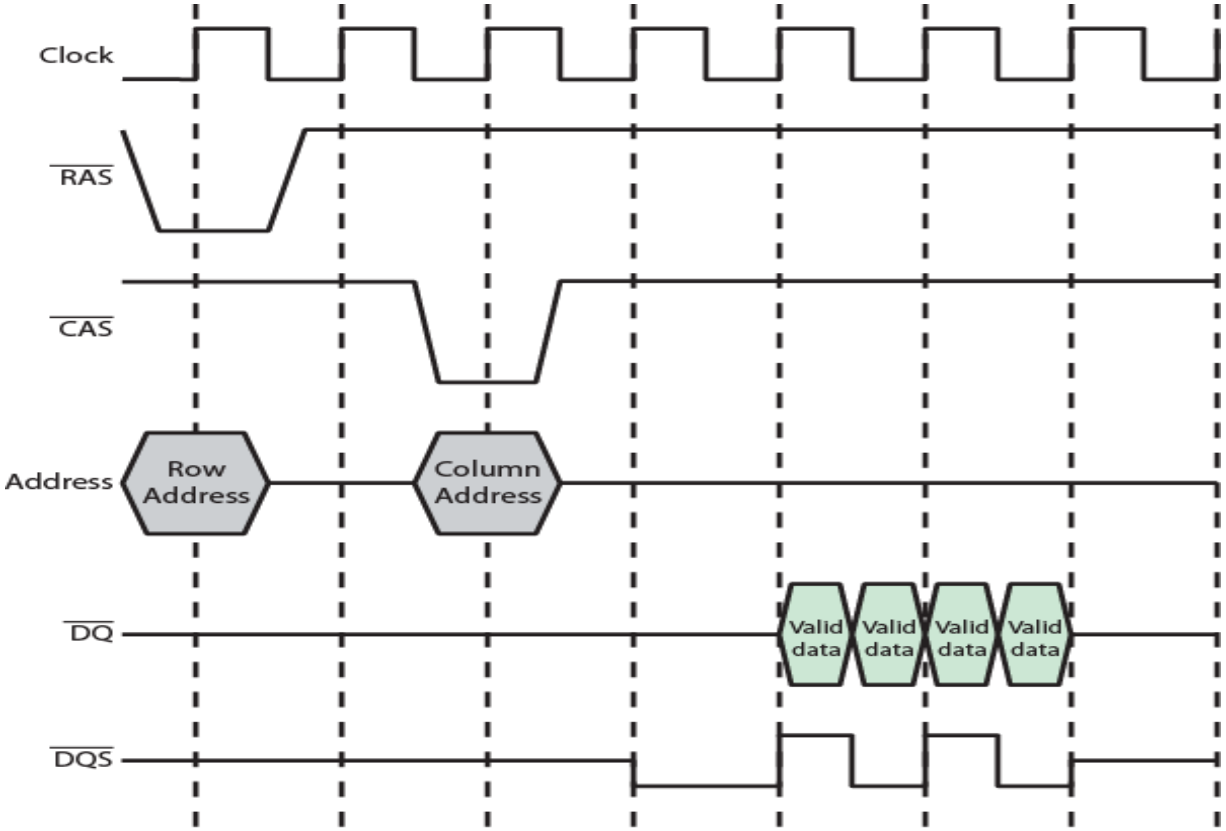


- 字处理、电子表格、多媒体应用，这些会有连续的读取需求
- SDRAM的爆发模式在连续传输大数据块的时候，能起到很好的效果
- 从读地址到第一个数据的输出有2个时钟周期的延迟
- 爆发长度是4，在读取第一个数据后，可以快速输出4个单元的数据

DDR DRAM 双倍速DRAM

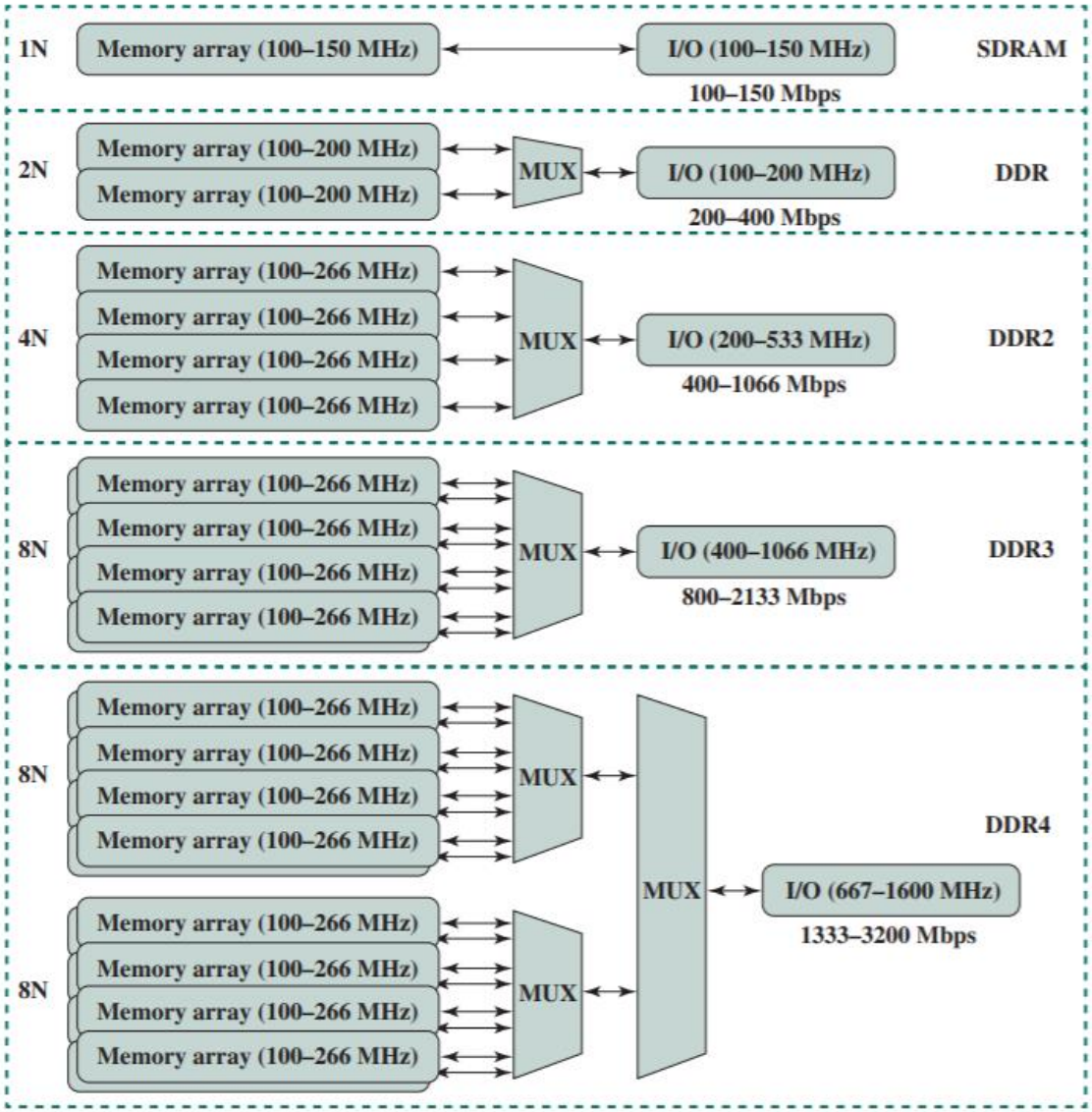
- SDRAM can only transmit data once in a clock cycle SDRAM只能在一个时钟周期传输一次数据
- DDR SDRAM can transmit data twice a clock cycle DDR-SDRAM可以一个时钟周期传2次数据
- DDR improves data rate in three ways DDR通过如下三种方式提高数据速率
  - Both the rising and falling edges of the clock can transmit data 时钟的上升沿和下降沿都可以传输数据
  - DDR uses a higher bus clock frequency to increase transmission rate DDR使用更高的总线时钟频率提高传输速率
  - Adopting prefetch buffering mechanism 采用预取缓冲机制

DDR DRAM read timing DDR的读时序图



RAS = row address select
CAS = column address select
DQ = data (in or out)
DQS = DQ select

DDR Generations 各代DDR

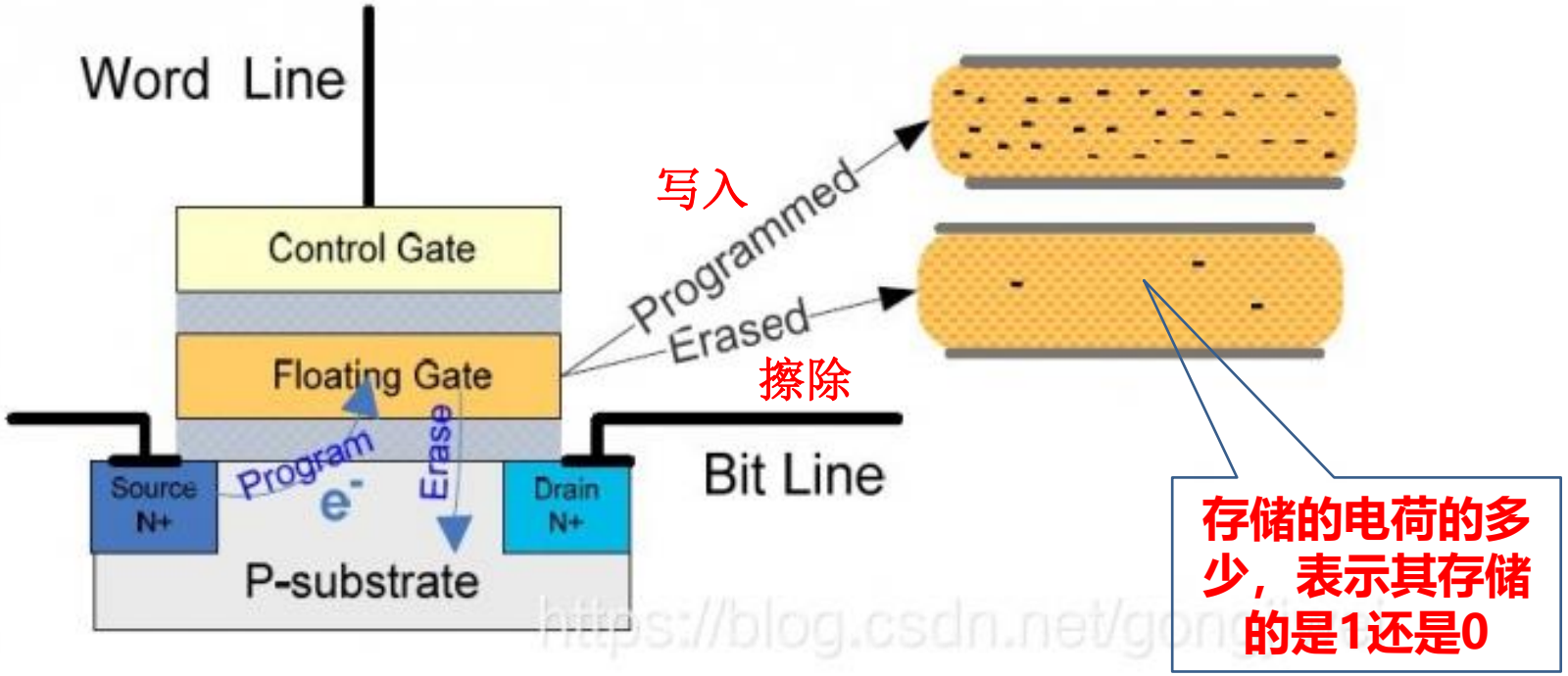


Outline

- Key Terms 关键术语
- Semiconductor main memory 半导体内存
- Error correction 纠错算法
- DDR DRAM 双倍速率DRAM
- Flash Memory 闪存

How Flash Works Flash工作原理

Flash的内部存储是金属-氧化层-半导体-场效晶体管 MOSFET，里面有个悬浮门 Floating Gate，是真正存储数据的单元



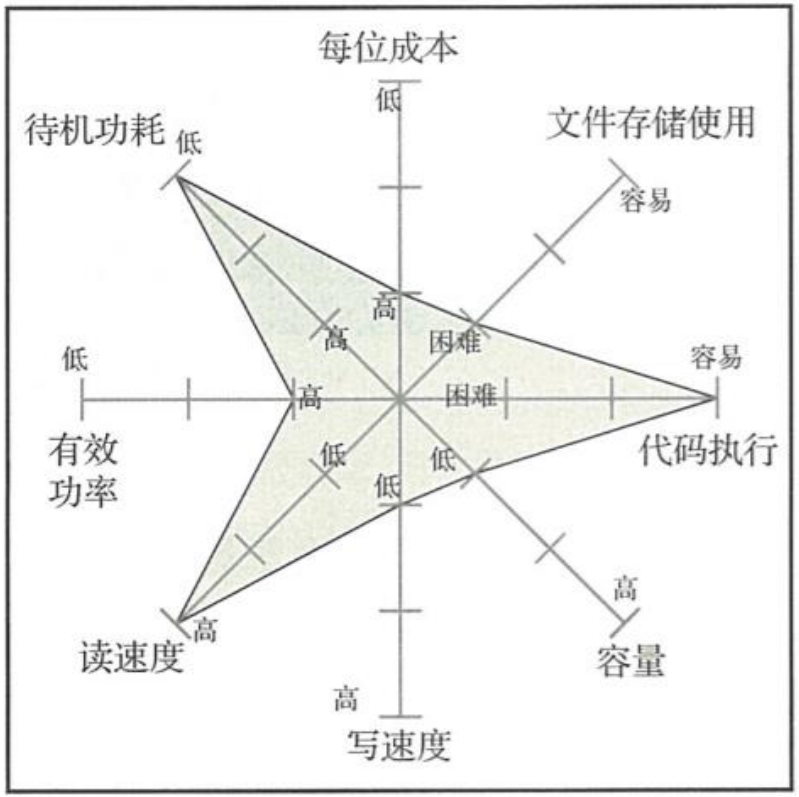
Compare of NOR and NAND -1 NOR和NAND的比较1

- Flash memory include NOR Flash and NAND FLASH 包括NOR Flash和NAND FLASH两类
- NOR
  - Developed by Intel in 1988 Intel公司1988年开发
  - With dedicated address and data line (similar to SRAM), read and write in byte mode 专用地址和数据线，跟SRAM类似，读写采用字节模式
  - Fast access speed and small storage capacity 高速访问，容量小
  - Data can be read randomly by byte 可以按字节随机读取
  - Program can be executed in the NOR 程序可以在NOR内执行
  - Speed of writing and erasing is not fast, which affects its performance 写入和擦除的速度都不快，影响了它的性能
  - Suitable for program storage, such as BIOS 适合程序存储，比如BIOS

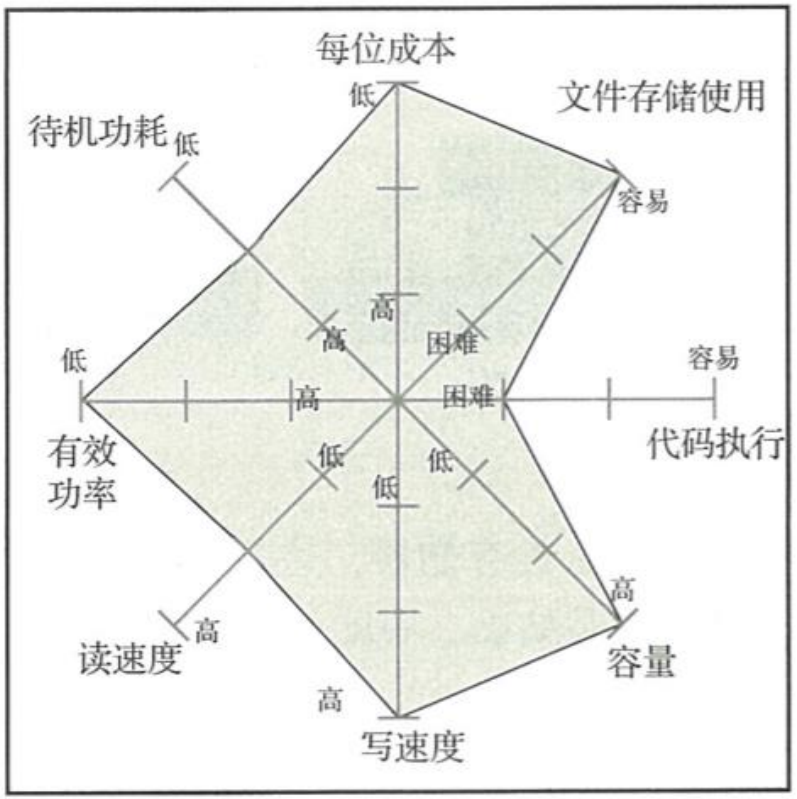
Compare of NOR and NAND -2 NOR和NAND的比较2

- NAND
  - Reading and writing in blocks 按块读写
  - Slower in reading, but much faster in writing and erasing than NOR
    比NOR读要慢，但写和擦除要快很多
  - Smaller volume and higher storage density than NOR flash memory
    体积小，存储密度大
  - Suitable for storing large amount of data 适合存储大量的数据
  - Including four types: SLC（Single-Level Cell）,MLC（Multi-Level Cell）,TLC（Trinary-Level Cell）,QLC（Quad-Level Cell）包括四种类型：SLC、MLC、TLC和QLC

Compare of NOR and NAND -3 NOR和NAND的比较3



a) NOR



b) NAND

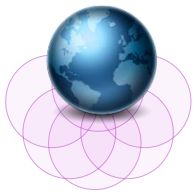
Summary and Question

- 小结
  - 对计算机内存单元的工作原理进行了学习
  - 讨论了半导体存储器的工作原理，包括存储器的组织、DRAM和SRAM、ROM的类型、芯片逻辑、芯片封装、存储器组织等
  - 对存储器的纠错原理和方法进行了分析
  - 对内存的几种优化的组织方式进行了分析解
- 问题
  - 问题1：SRAM和DRAM有什么区别？
  - 问题2：存储器的容量扩展有哪两种方法？
  - 问题3：简述纠错的工作原理

Assignments

- review questions: 5.1, 5.4, 5.10

- problem: 5.4, 5.6, 5.8, 5.11



谢谢大家！