计 算 机 组 成 原 理

补充部分

存储器

2023年6月3日

本讲安排

1、存储器概述

外部特性，性能参数，层次结构

2、静态存储器和动态存储器存储单元构成

一位存储单元及存储阵列，多端口SRAM，读写时序

3、半导体ROM存储器

MROM，PROM，EPROM，EEPROM，FLASH

4、存储器芯片构成以及存储器主要技术指标

5、存储器扩展技术

位、字、字位扩展

6、数据校验码

奇偶校验码，海明码，CRC码

本讲将解决的主要问题

L、半导体存储器的分类、组成及组成部件的作用及工作原理、读/写操作的基本过程。

2、SRAM、DRAM芯片的组成特点、工作过程、典型芯片的引脚信号、了解DRAM刷新的基本概念。

3、半导体存储器的主要技术指标、芯片的扩充、CPU与半导体存储器间的连接。

简介

在现代计算机中,存储器处于全机中心地位,其原因是:

(1) 当前计算机正在执行的程序和数据(除了暂存于CPU寄存器的)均存放在存储器中。CPU直接从存储器取指令或存取数据。

(2) 计算机系统中输入输出设备数量增多，数据传送速度加快，因此采用了直接存储器存取(DMA)技术和I/O通道技术，在存储器与输入输出系统之间直接传送数据。

(3) 共享存储器的多处理机的出现，利用存储器存放共享数据，并实现处理机之间的通信，更加强了存储器作为全机中心的地位。

由于中央处理器都是由高速器件组成，不少指令的执行速度基本上取决于主存储器的速度。所以，计算机解题能力的提高、应用范围的日益广泛和系统软件的日益丰富，无一不与主存储器的技术发展密切相关。

存储器概述

存储器分类

存储器的层次结构

存储器的技术指标

存储器概述

几个基本概念

1、存储器：是计算机系统中的记忆设备，用来存放程序和数据。

2、存储元：存储器的最小组成单位，用以存储1位二进制代码。

3、存储单元：是CPU访问存储器基本单位，由若干个具有相同操作属性的存储元组成。

4、单元地址：在存储器中用以表识存储单元的唯一编号，CPU通过该编号访问相应的存储单元。

5、字存储单元：存放一个字的存储单元，相应的单元地址叫字地址。

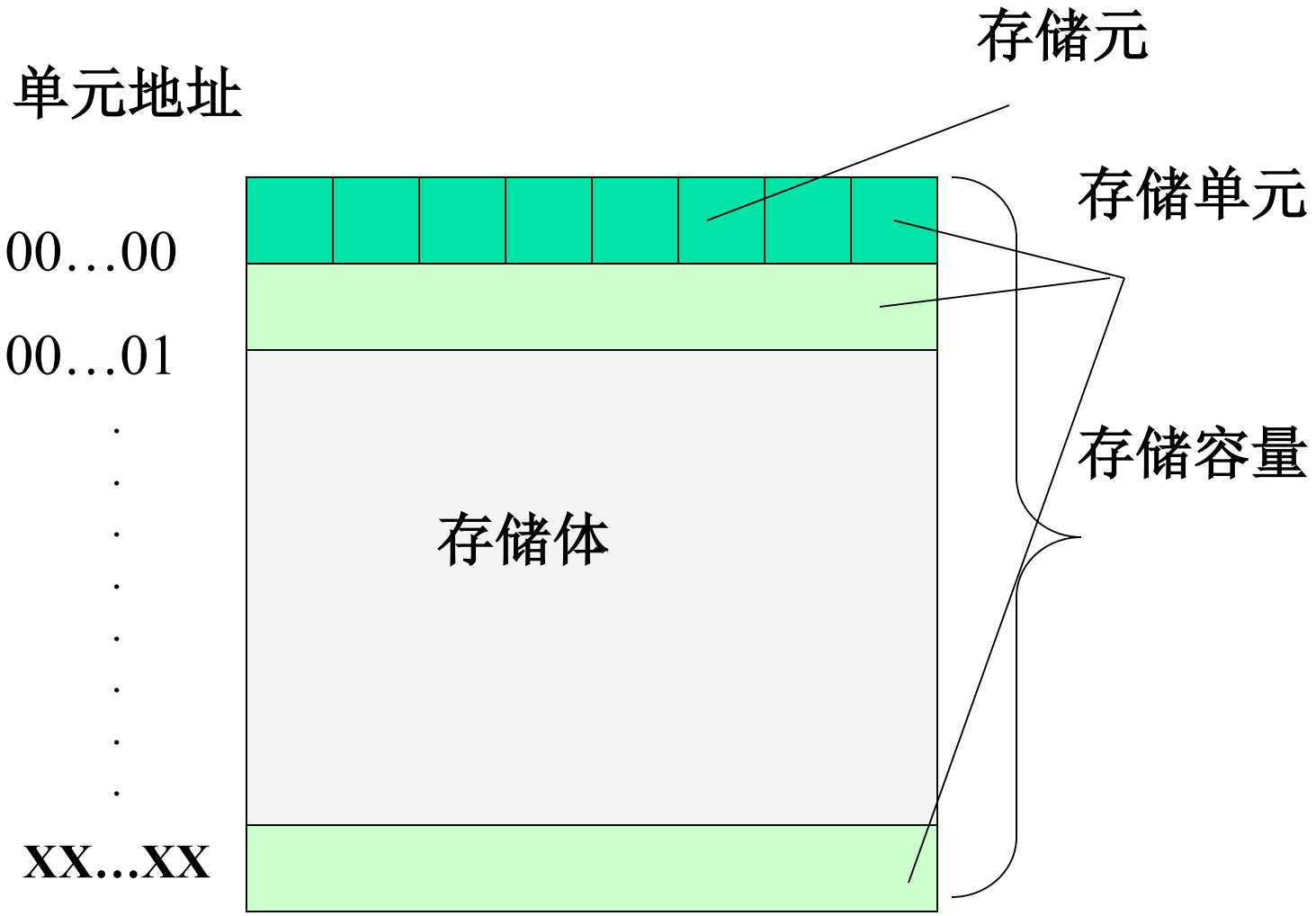
6、字节存储单元：存放一个字节的存储单元，相应的单元地址叫字节地址

7、按字寻址计算机：可编址的最小单位是字存储单元的计算机。

8、按字节寻址计算机：可编址的最小单位是字节的计算机。

9、存储体：存储单元的集合，是存放二进制信息的地方

存储器各个概念之间的关系



存储器分类

1. 按存储介质分

半导体存储器：用半导体器件组成的存储器。

磁表面存储器：用磁性材料做成的存储器。

2. 按存储方式分

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。

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。

3. 按存储器的读写功能分

只读存储器(ROM)：存储的内容是固定不变的，只能读出而不能写入的半导体存储器。

随机读写存储器(RAM)：既能读出又能写入的半导体存储器。

4. 按信息的可保存性分

非永久记忆的存储器：断电后信息即消失的存储器。

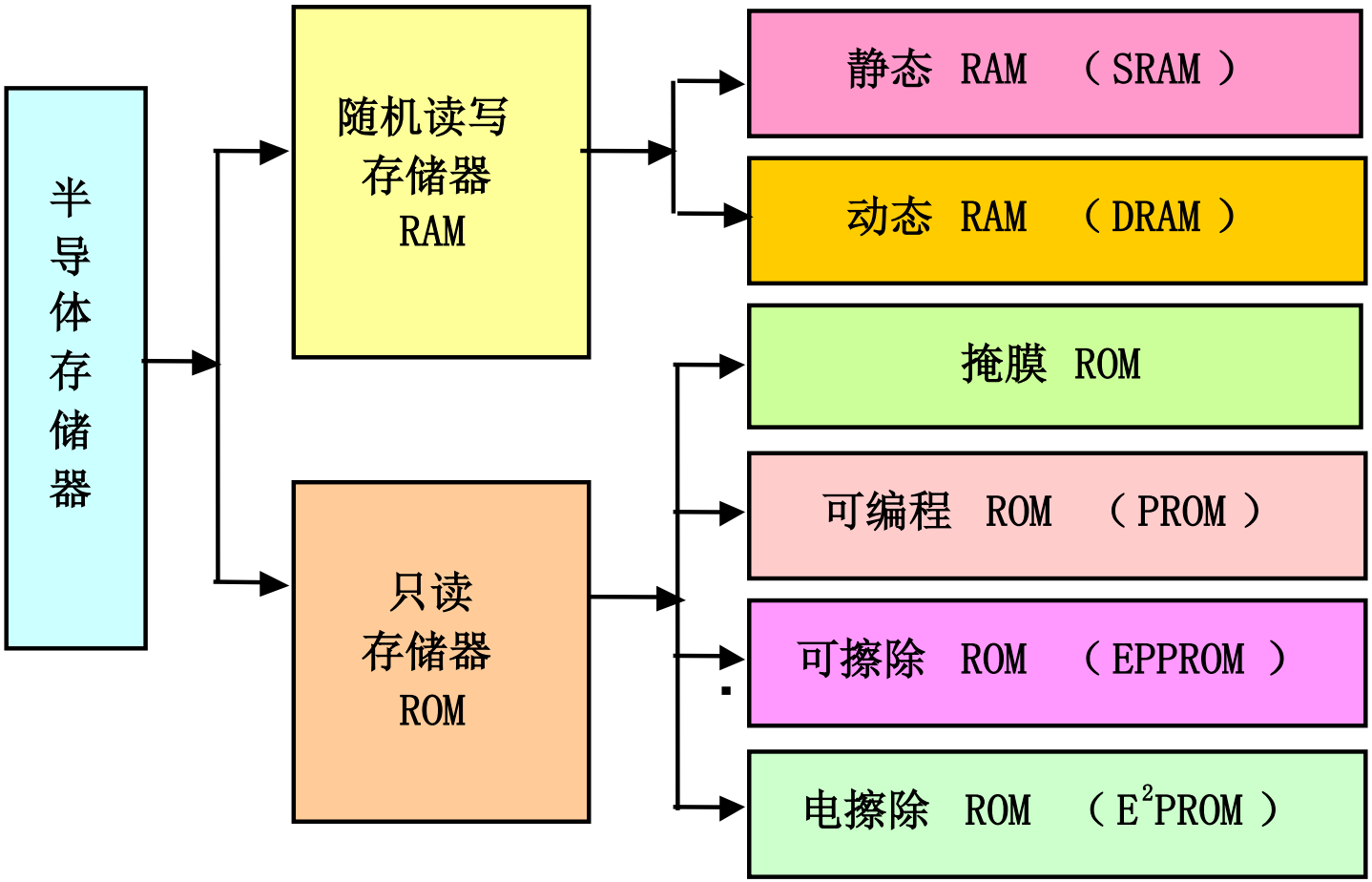
永久记忆性存储器：断电后仍能保存信息的存储器。

5. 按在计算机系统中的作用分

根据存储器在计算机系统中所起的作用，可分为:

主存储器、辅助存储器、高速缓冲存储器、 控制存储器等。

半导体存储器

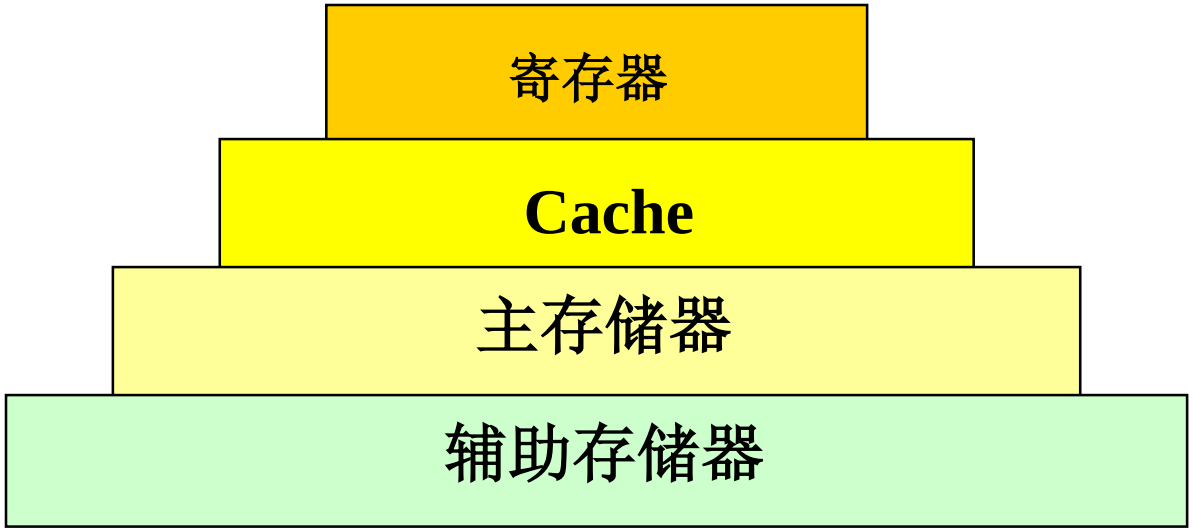

半导体存储器
随机读写存储器 RAM
只读存储器 ROM
静态 RAM （SRAM）
动态 RAM （DRAM）
掩膜 ROM
可编程 ROM （PROM）
可擦除 ROM （EPPROM）
电擦除 ROM （E²PROM）

存储器层次结构

- 对存储器的要求是：

  容量大，速度快，成本低。

- 为解决三者之间的矛盾，目前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。



存储器的用途和特点

名　称	简称	用　　途	特　点
高速缓冲存储器	Cache	高速存取指令和数据	存取速度快，但存储容量小
主存储器	主存	存放计算机运行期间的大量程序和数据	存取速度较快，存储容量不大
外存储器	外存	存放系统程序和大型数据文件及数据库	存储容量大，位成本低

主存储器的技术指标

存储容量；存取时间(存储器访问时间）、存储周期和存储器带宽；可靠性；功耗及集成度。

指 标	含 义	表 现	单 位
存储容量	在一个存储器中可以容纳的存储单元总数	存储空间的大小	字数，字节数
存取时间	启动到完成一次存储器操作所经历的时间	主存的速度	n s
存储周期	连续启动两次操作所需间隔的最小时间	主存的速度	n s
存储器带宽	单位时间里存储器所存取的信息量	数据传输速率技术指标	位/秒，字节/秒

- **可靠性** 主存储器的可靠性通常用平均无故障时间 MTBF (Mean Time Between Failures)来表征。MTBF指连续两次故障之间的平均时间间隔。显然，MTBF越长，意味着主存的可靠性越高，

- **功耗** 作为目前的主存储器的主体的半导体存储器的功耗包括“维持功耗”和“操作功耗”，应在保证速度的前提下尽可能地减小功耗，特别是要减小“维持功耗”。

- **集成度** 所谓集成度是指在一片数平方毫米的芯片上能集成多少个存储单元，每个存储单元存储一个二进制位，所以集成度常表示为位/片。

SRAM存储器

DRAM存储器

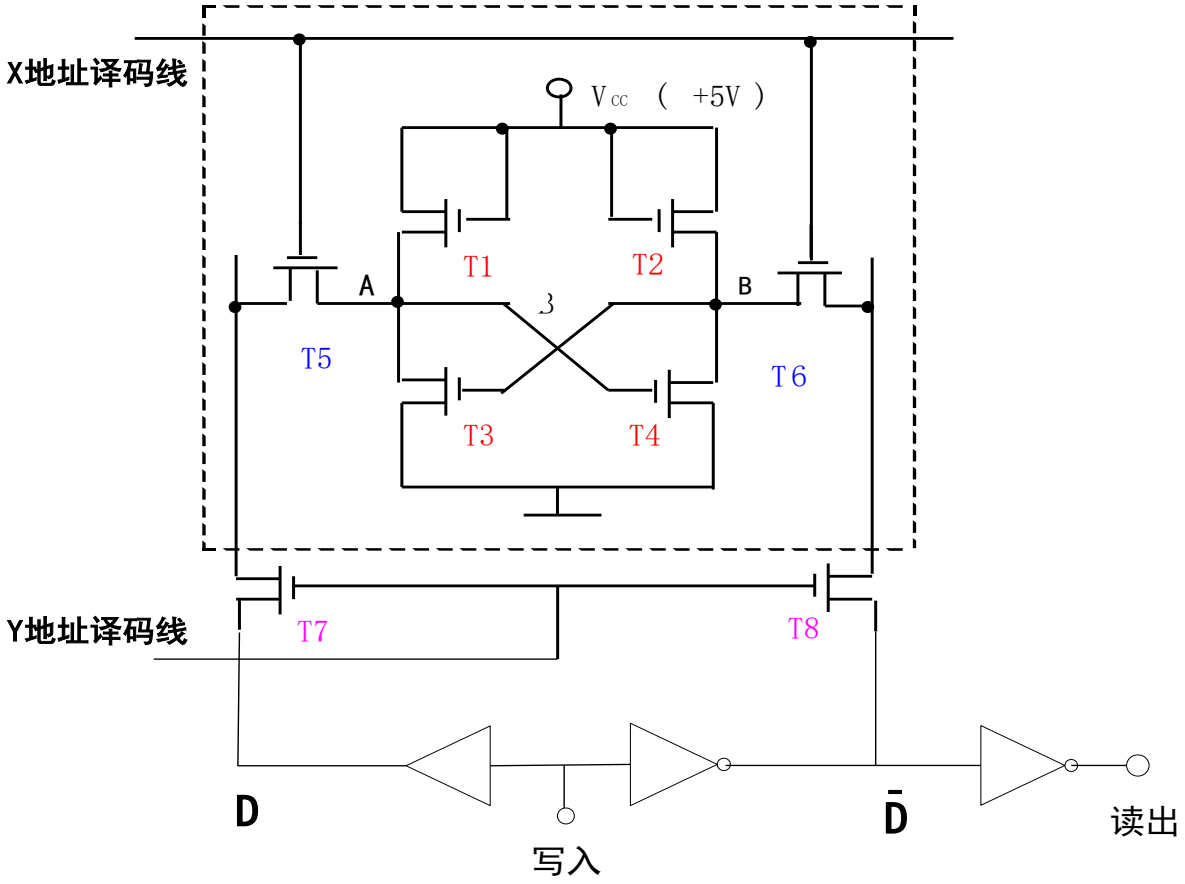
主存储器组成实例

高性能的主存储器

SRAM存储器

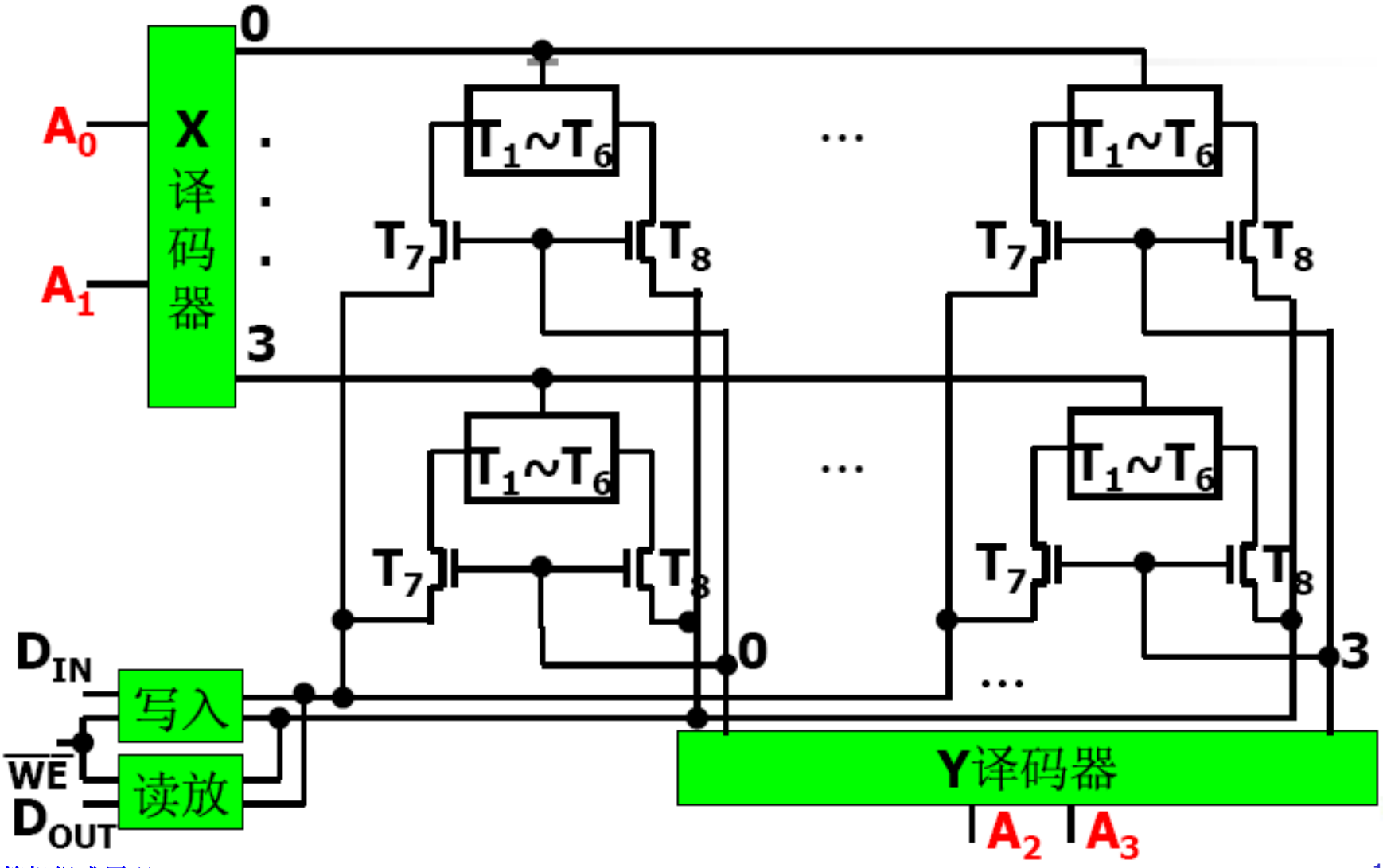
1. 基本存储元

基本存储元是组成存储器的基础和核心,它用来存储一位二进制信息0或1。

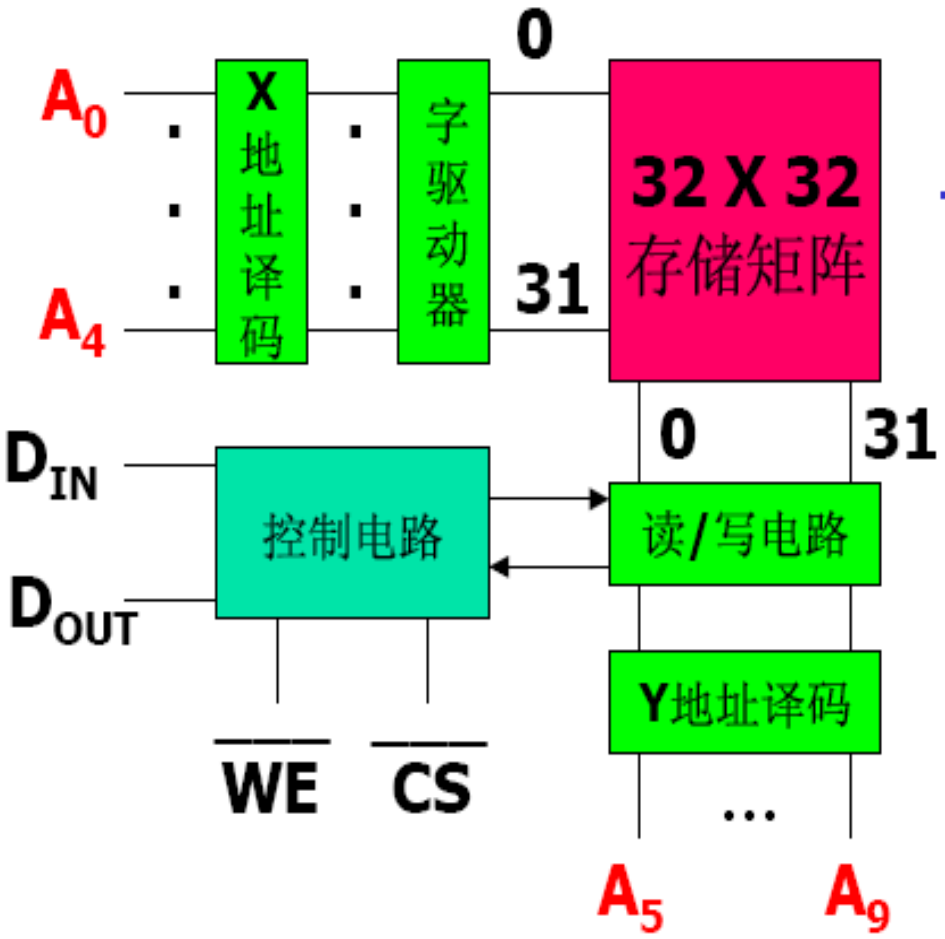


六管基本存储单元电路

16×1 bit SRAM



1K bit SRAM

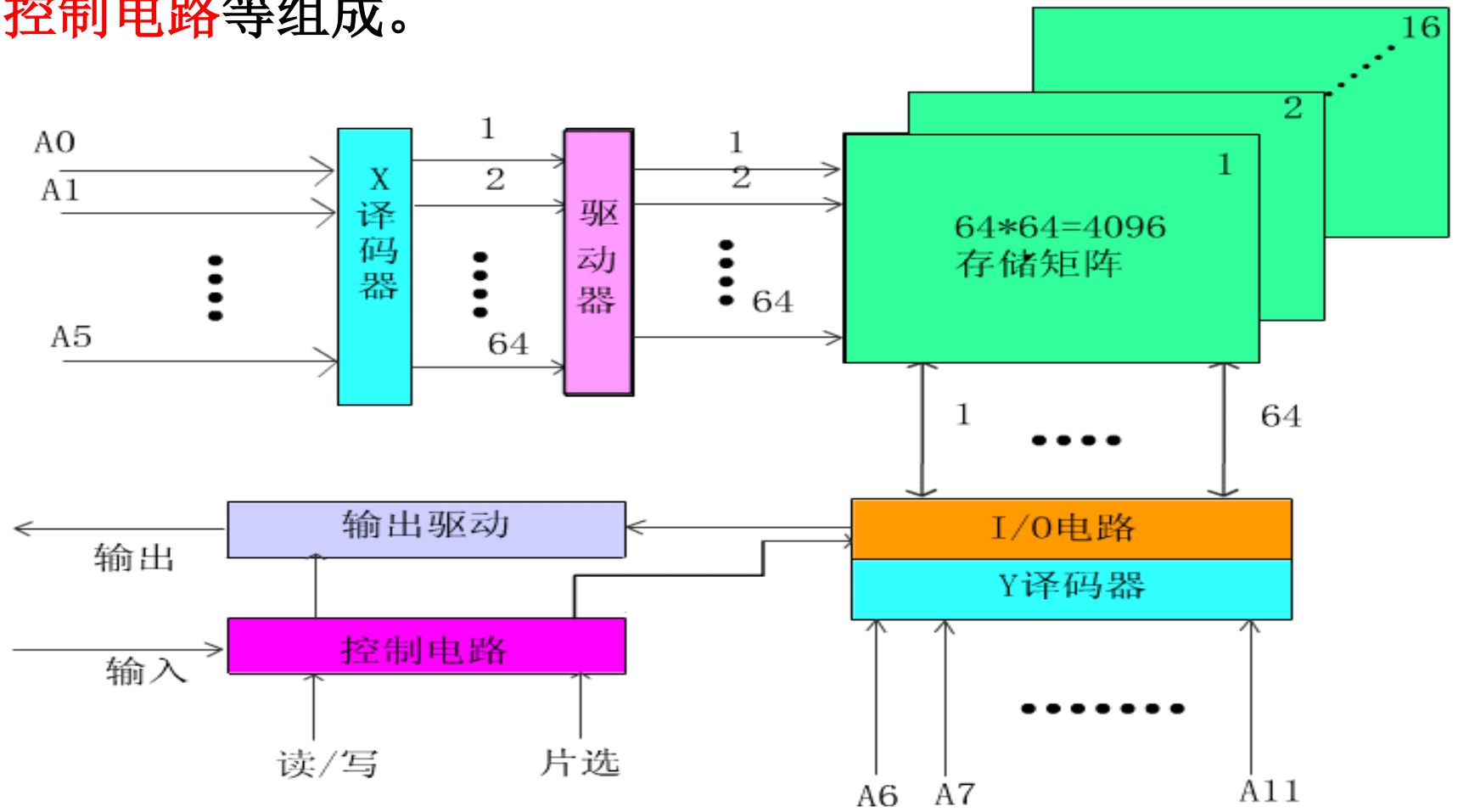


功能表

$\overline{CS}$	$\overline{WE}$	D_{IN}	D_{OUT}	操作方式
H	X	X	H	未选
L	L	L	H	写“0”
L	L	H	H	写“1”
L	H	X	D_{OUT}	读

2. SRAM存储器的组成

一个SRAM存储器由存储体、读写电路、地址译码电路和控制电路等组成。



(1) 存储体

- 一个基本存储电路只能存储一个二进制位。
- 将基本的存储电路有规则地组织起来，就是存储体。
- 存储体又有不同的组织形式：

将各个字的同一位组织在一个芯片中；
将各个字的4位组织在一个芯片中， 如：2114 1K×4；
将各个字的8位组织在一个芯片中， 如：6116 2K×8；

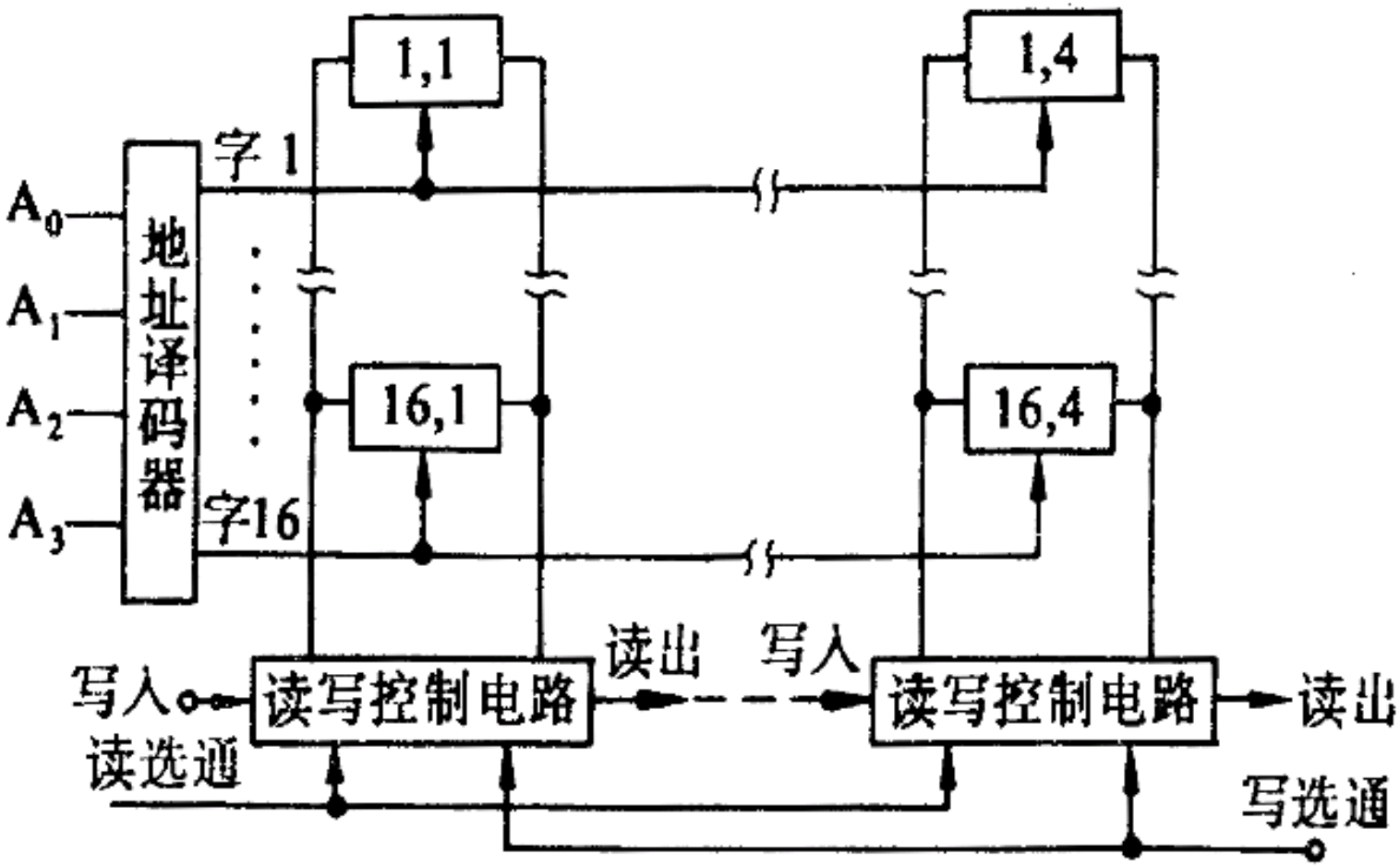
如图所示：

存储体将4096个字的同一位组织在一个集成片中；
需16个片子组成4096×16的存储器；

4096通常排列成矩阵形式，如 64×64，由行选、列选线选中所需的单元。

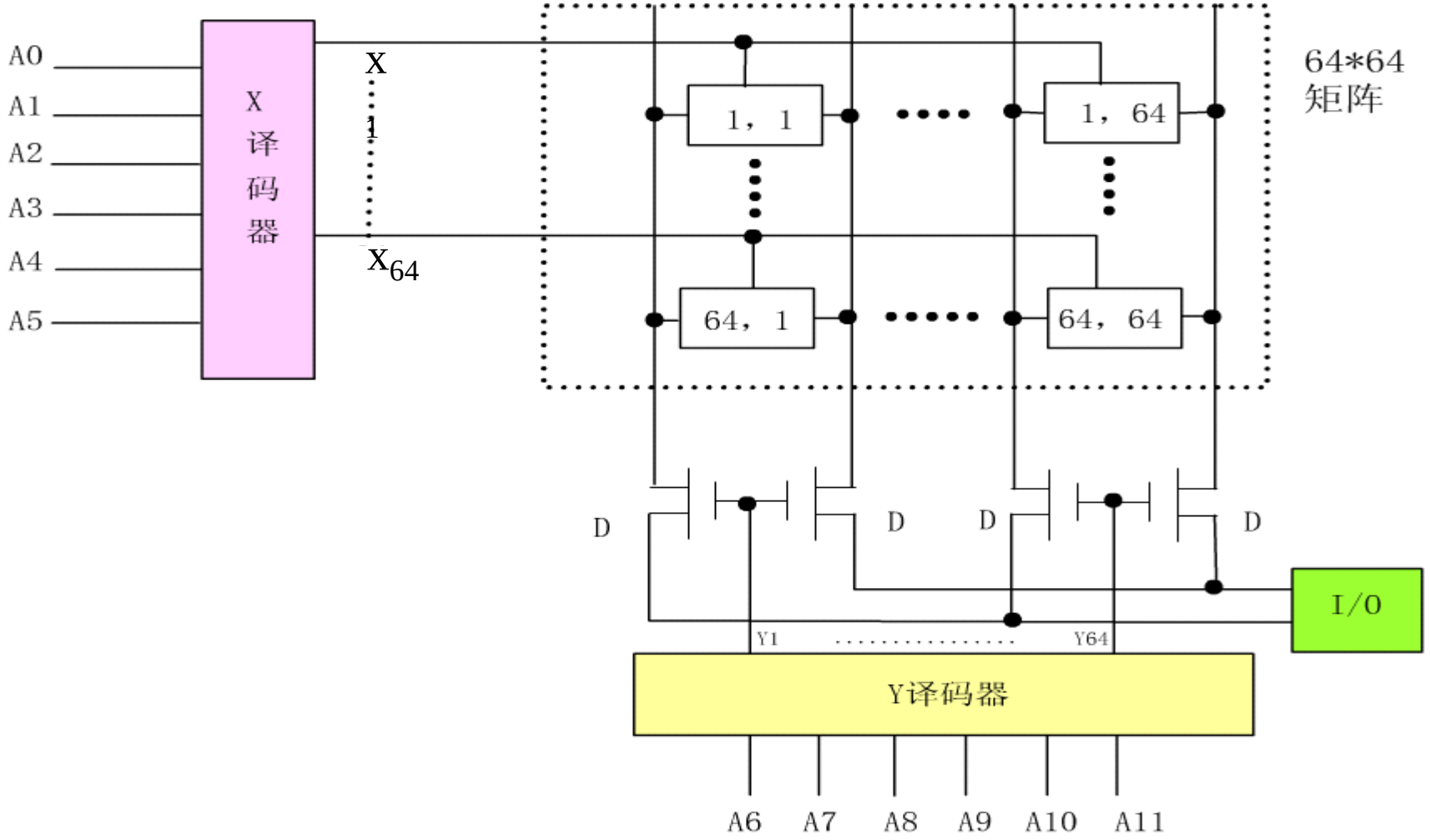
(2) 地址译码器

单译码方式——适用于小容量存储器中，只有一个译码器。



双译码方式

——地址译码器分成两个，可有效减少选择线的数目。



(3) 驱动器

双译码结构中，在译码器输出后加驱动器，驱动挂在各条X方向选择线上的所有存储元电路。

(4) I/O电路

处于数据总线和被选用的单元之间， 控制被选中的单元读出或写入，放大信息。

(5) 片选

在地址选择时，首先要选片,只有当片选信号有效时，此片所连的地址线才有效。

(6) 输出驱动电路

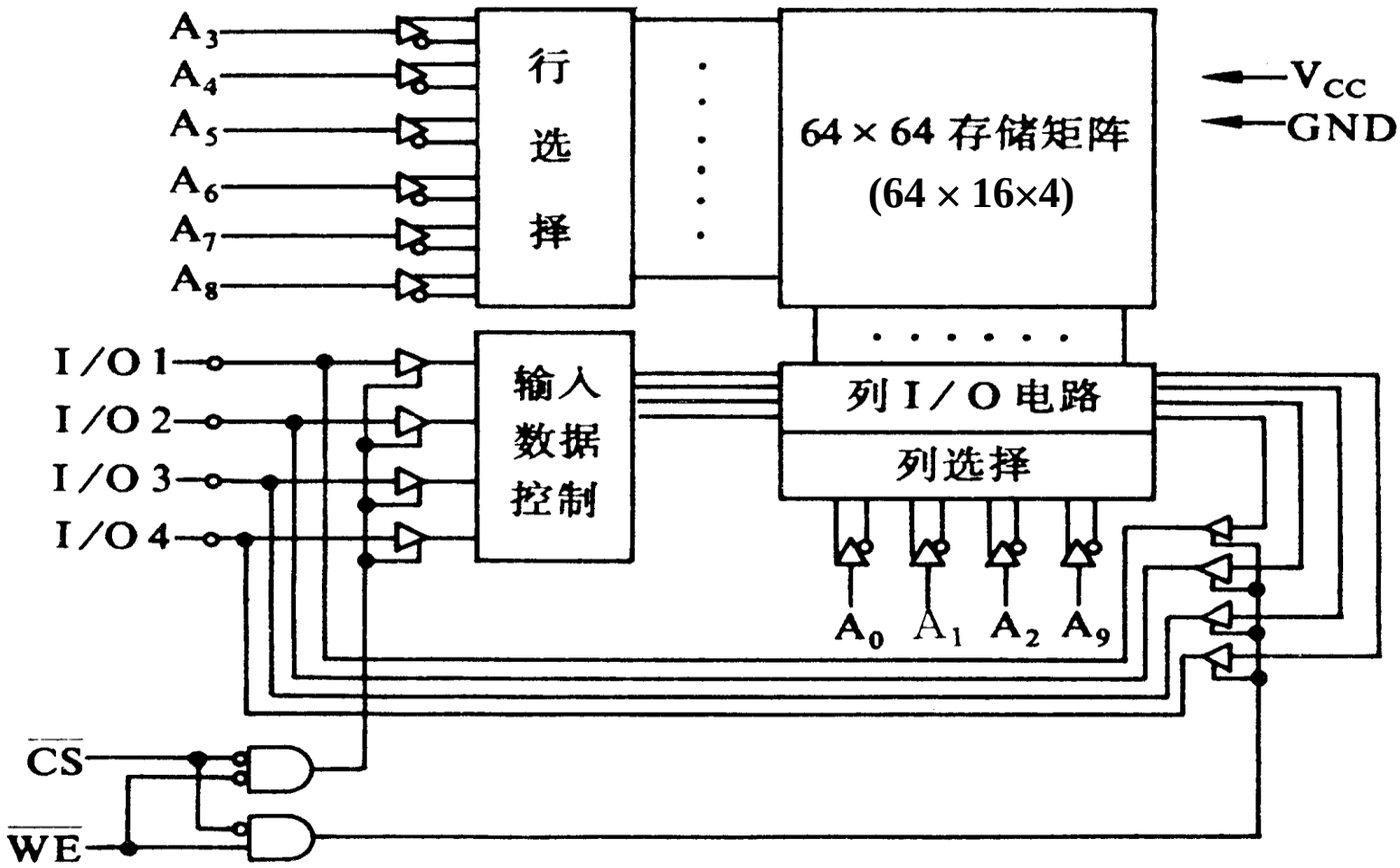
为了扩展存储器的容量，常需要将几个芯片的数据线并联使用；另外存储器的读出数据或写入数据都放在双向的数据总线上。这就用到三态输出缓冲器。

3. SRAM存储器芯片实例

Intel 2114——1024×4 的存储器：

- **4096 个基本存储单元，排成 64×64 (64×16×4) 的矩阵；**
- **需 10 根地址线寻址；**
- **X 译码器输出 64 根选择线，分别选择 1-64 行；**
- **Y 译码器输出 16 根选择线，分别选择 1-16 列控制各列的位线控制门。**

Intel 2114——1K×4 SRAM



4. 存储器的读、写周期

在与CPU连接时，CPU的控制信号与存储器的读、写周期之间的配合问题是非常重要的。

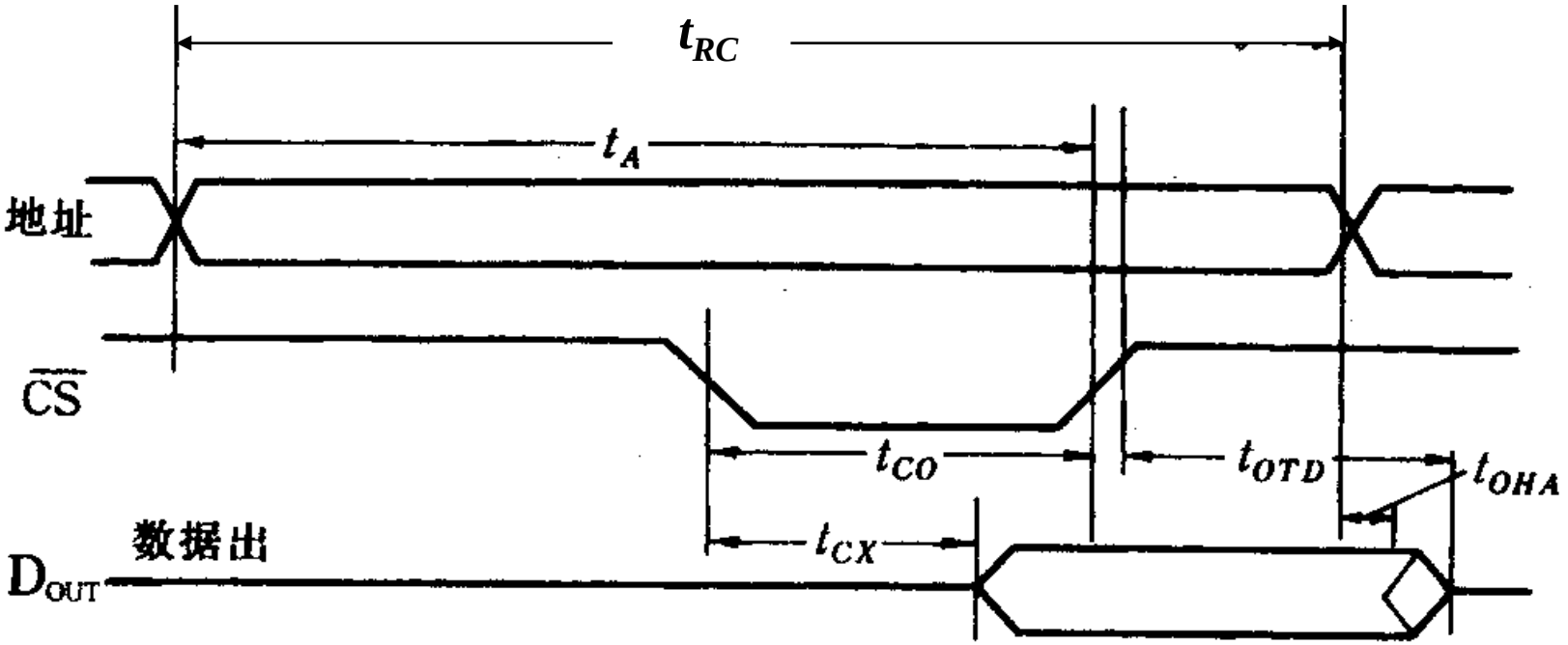
读周期:

读周期与读出时间是两个不同的概念。

读出时间——从给出有效地址到外部数据总线上稳定地出现所读出的数据信息所经历的时间。

读周期时间——则是存储器进行**两次连续读**操作时所必须间隔的时间，它总是大于或等于读出时间。

SRAM存储器时序



静态存储器的读周期

地址有效→CS有效→数据输出→CS复位→地址撤销

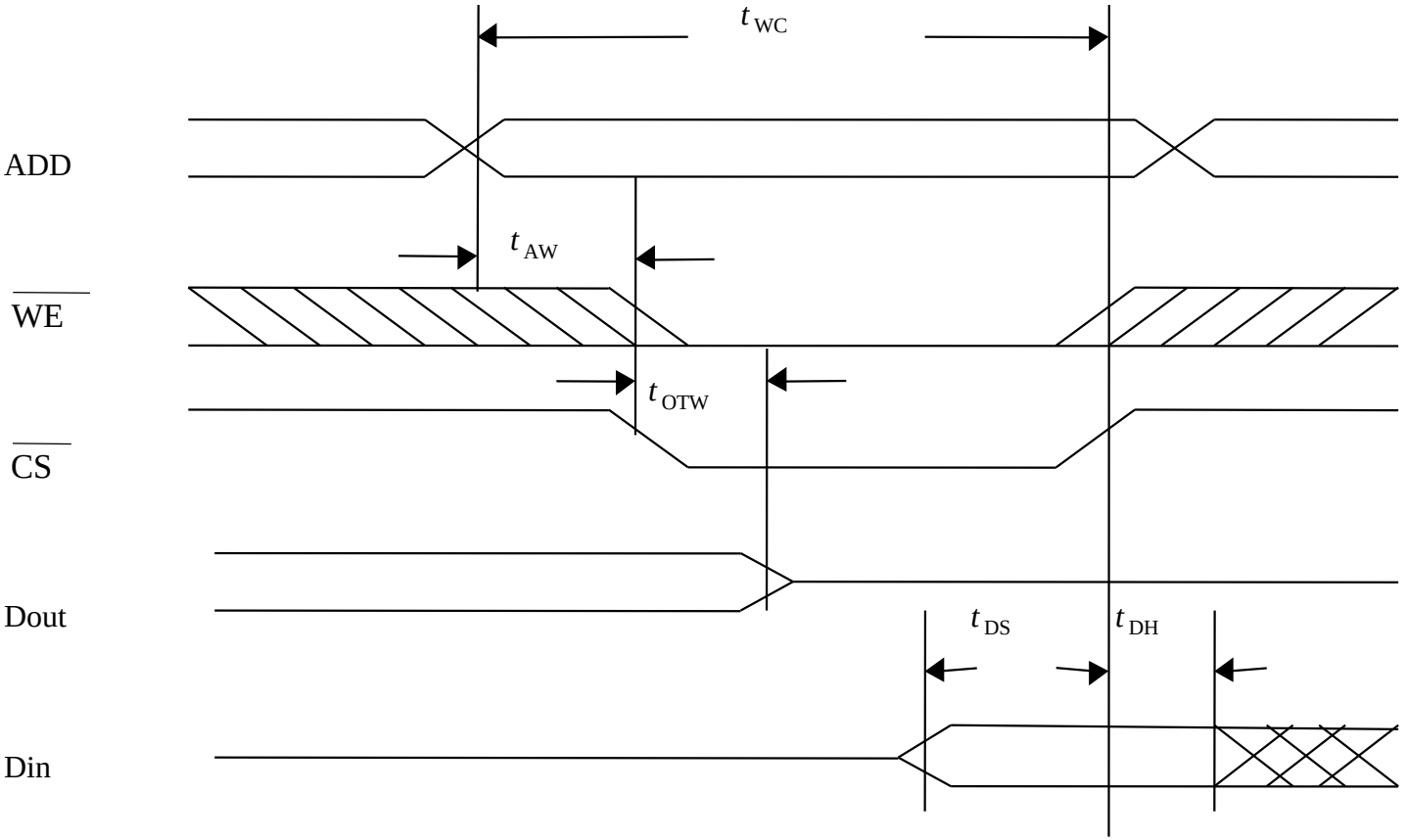
t_{RC} —— 读周期　　t_A —— 读出周期　　t_{CO} —— 片选到数据输出延迟

t_{CX} —— 片选到输出有效　　t_{OTD} —— 从断开片选到输出变为三态

t_{OHA} —— 地址改变后的维持时间

写周期：

地址有效→CS有效→数据有效→CS复位（数据输入）→地址撤销

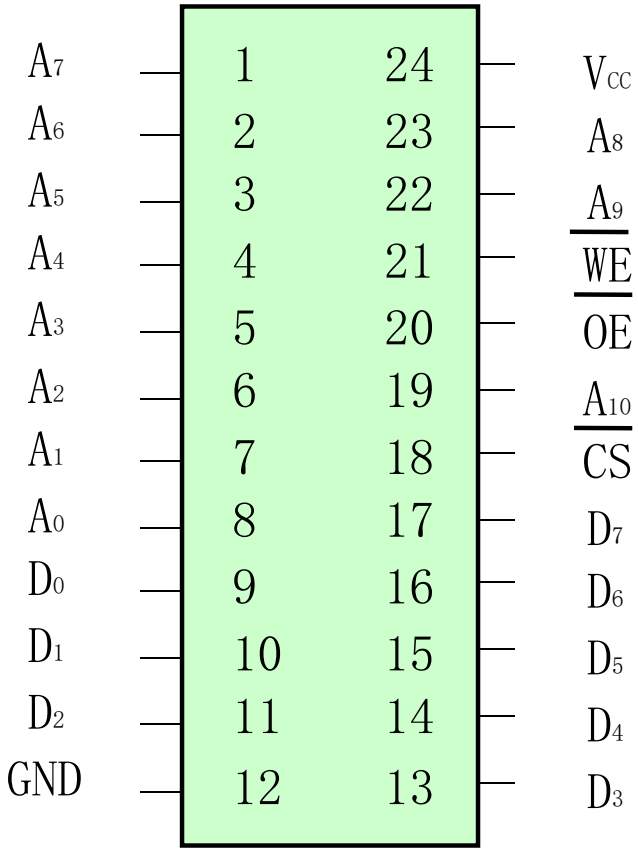


静态存储器的读写周期

SRAM芯片实例

常用典型的SRAM芯片有6116、6264、62256等。

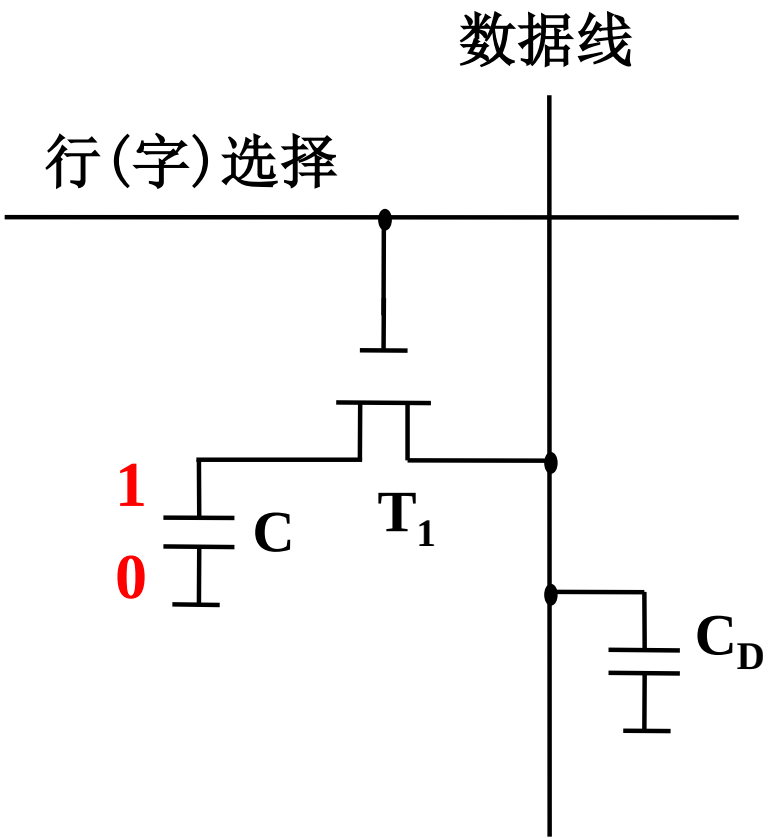
SRAM 6116 (2K × 8)

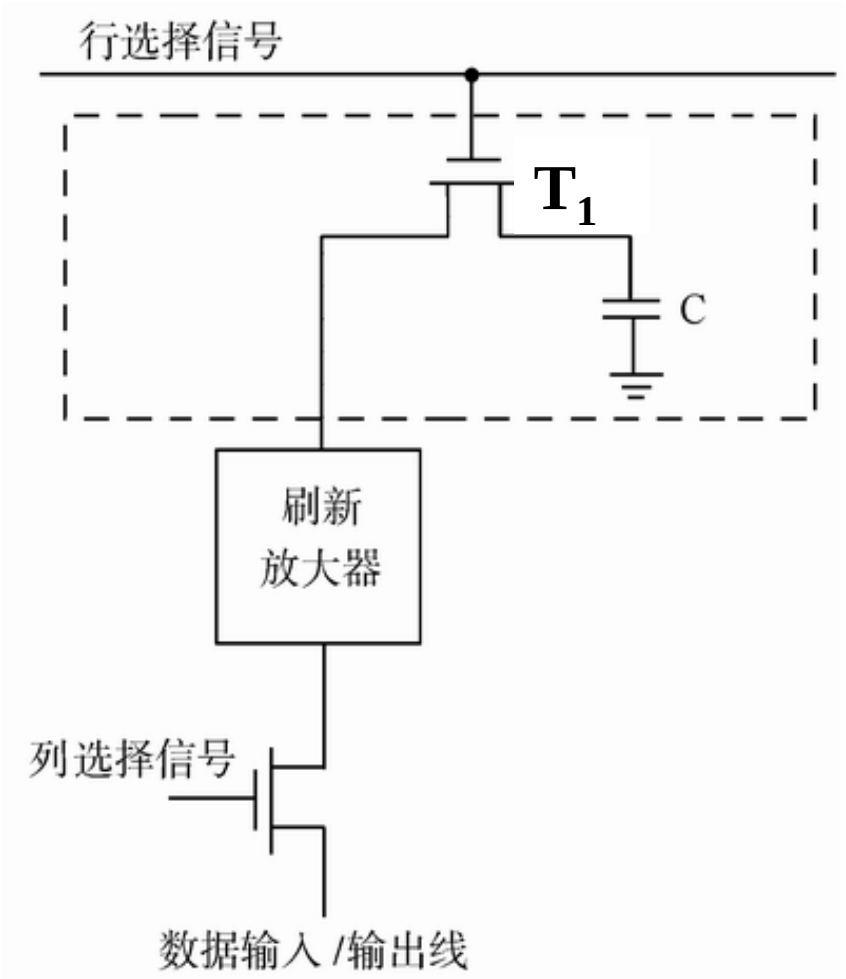


输入			I/O		工作方式
$\overline{CE}$	$\overline{WE}$	$\overline{OE}$	DI	DO	
H	X	X	X	High-Z	非选择
L	H	L	High-Z	DO	读
L	L	H	DI	High-Z	写
L	L	L	DI	High-Z	写
L	H	H	X	High-Z	选择

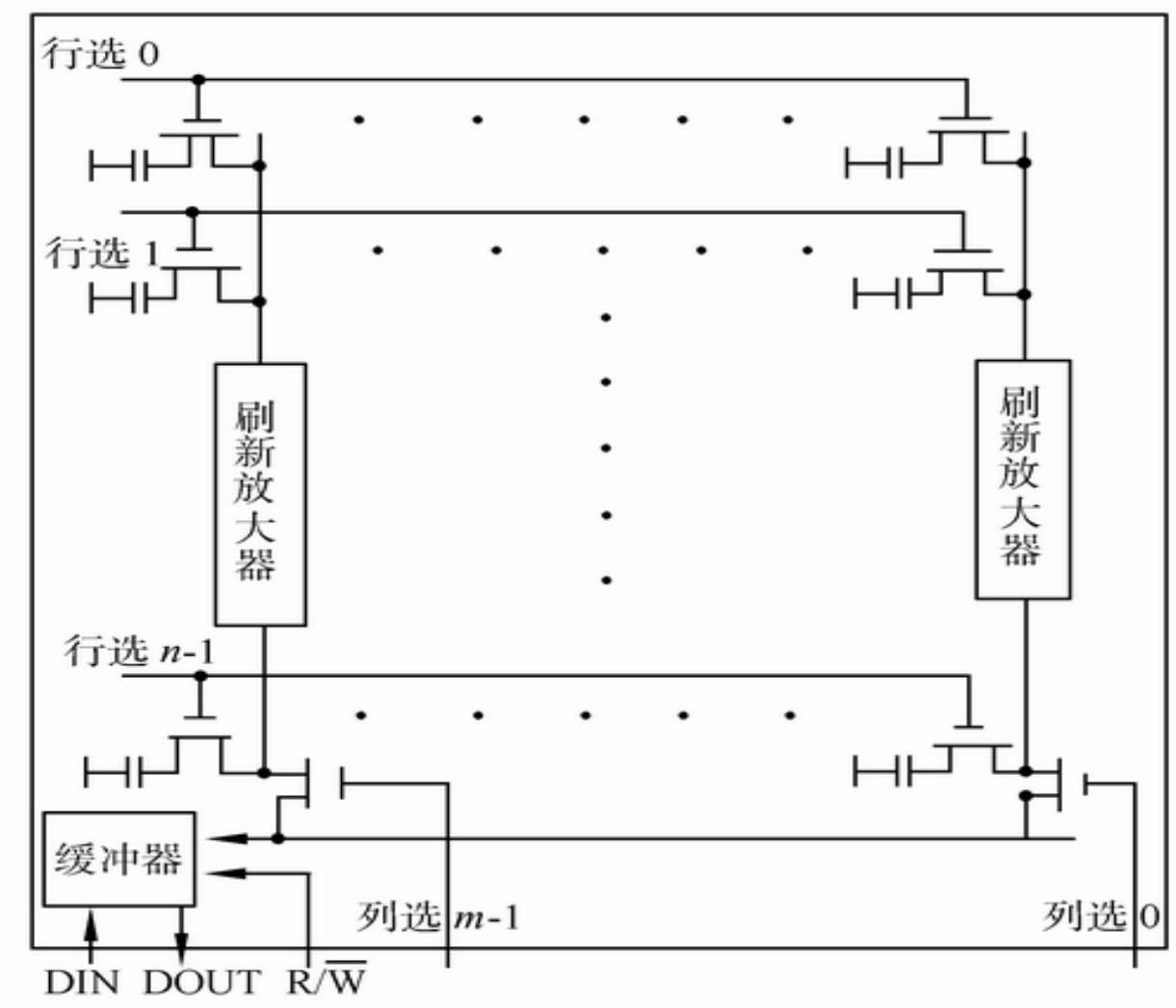
DRAM存储器

1. 单管动态存储元





单管DRAM的存储矩阵

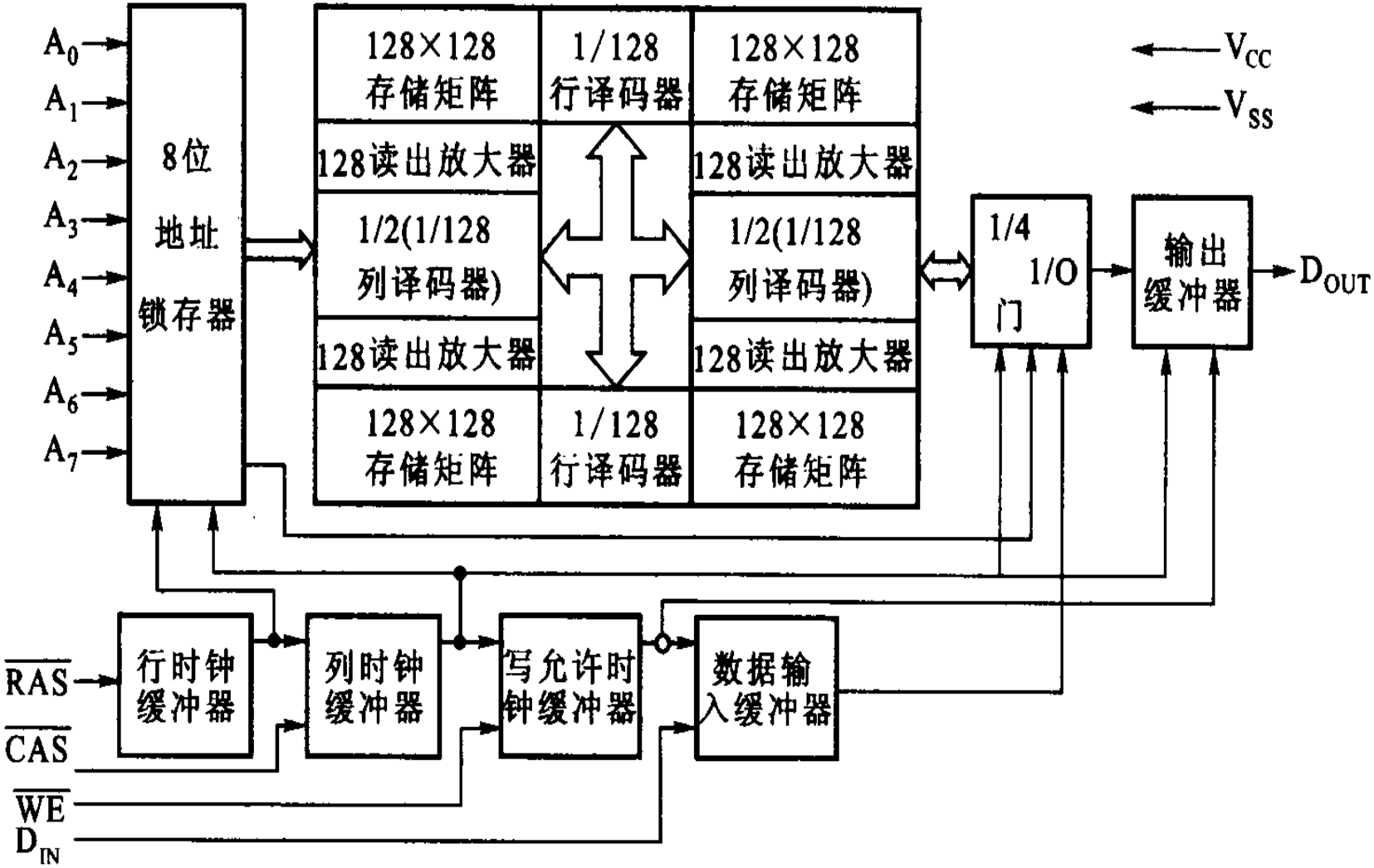


DRAM的电气特征：

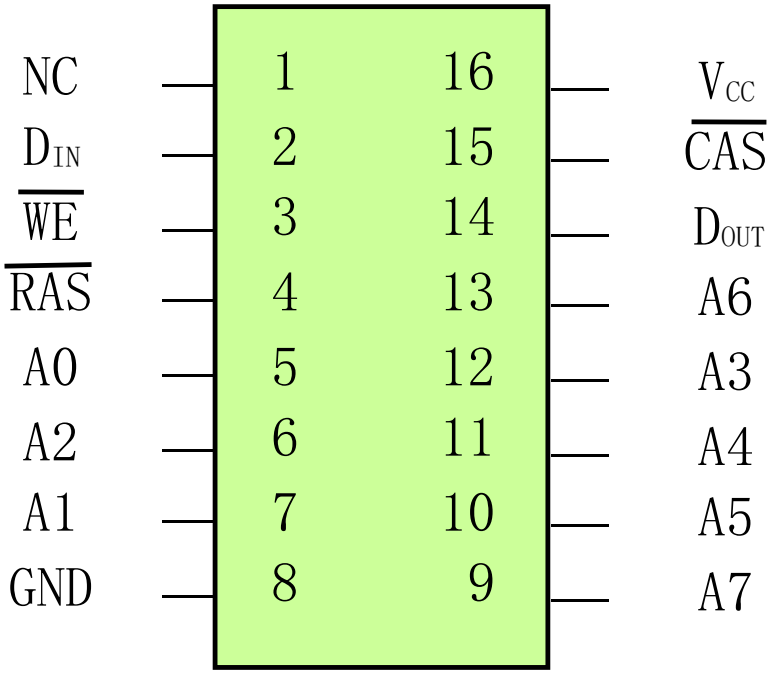
- 集成度高，功耗低
- 具有易失性，必须刷新。
- 破坏性读出，必须读后重写
- 读后重写，刷新均经由刷新放大器进行。
- 刷新时只提供行地址，由各列所拥有的刷新放大器，对选中行全部存储细胞实施同时集体读后重写(再生)。

2. DRAM存储芯片实例

内部结构——Intel2164(64K×1)



Intel 2164(64K×1)引脚



A0～A7：地址输入线

RAS：行地址选通信号线，兼起片选信号作用（整个读写周期，RAS一直处于有效状态）

CAS：列地址选通信号线

WE：读写控制信号 0-写 1-读

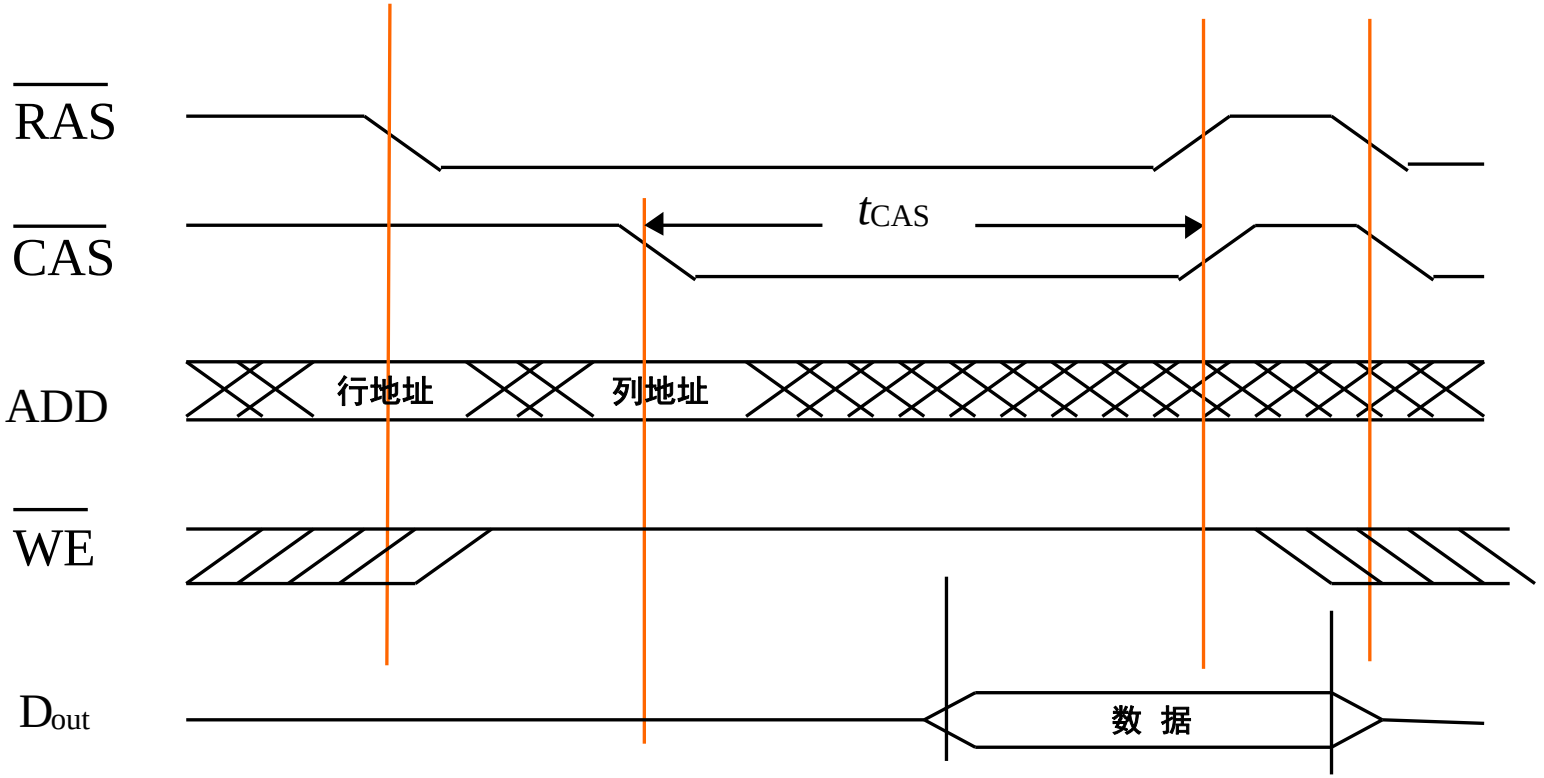
Din：数据输入线

Dout：数据输出线

DRAM时序

读周期：

行地址有效→行地址选通→列地址有效→列地址选通→数据输出→行选通、列选通及地址撤销

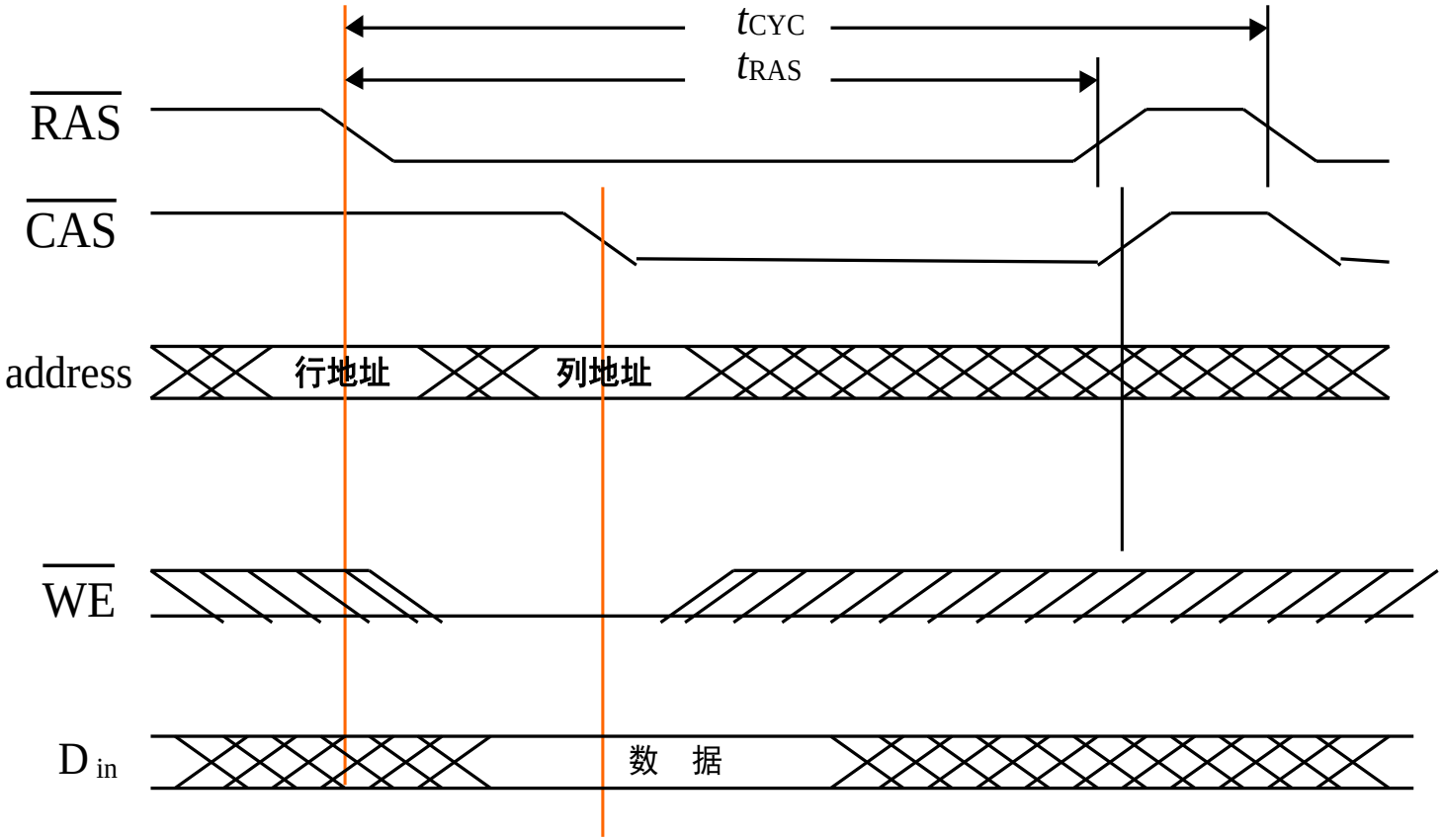


(a) 读周期

DRAM时序

写周期：

行地址有效→行地址选通→列地址、数据有效→列地址选通→数据输入→行选通、列选通及地址撤销



(b)　写周期

3. DRAM的刷新

(1) DRAM的刷新

不管是哪一种动态RAM，都是利用电容存储电荷的原理来保存信息的，由于电容会逐渐放电，所以，对动态RAM必须不断进行读出和再写入，以使泄放的电荷受到补充，也就是进行刷新。

动态MOS存储器采用“读出”方式进行刷新，先将原存信息读出，再由刷新放大器形成原信息并重新写入。

(2) 刷新周期

从上一次对整个存储器刷新结束到下一次对整个存储器全部刷新一遍为止，这一段时间间隔叫刷新周期。

一般为2ms, 4ms, 8ms。

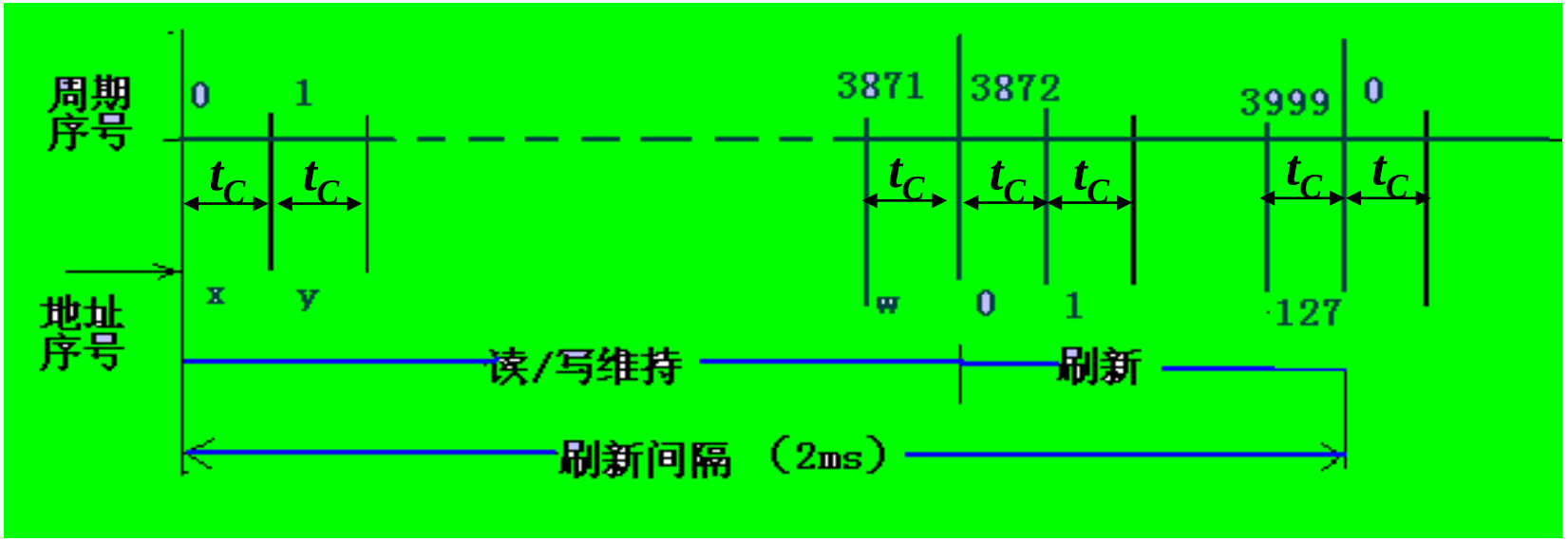
(3) 刷新方式

常用的刷新方式有三种：

集中式、分散式、异步式。

• 集中式刷新

在整个刷新间隔内，前一段时间重复进行读/写周期或维持周期，等到需要进行刷新操作时，便暂停读/写或维持周期，而逐行刷新整个存储器，它适用于高速存储器。



例如：对128×128矩阵存储器刷新。

刷新时间相当于128个读周期；

设刷新周期为2ms，读/写周期为0.5μs，则 刷新周期有4000个周期，其中

3782个周期（1936 μs）用来读/写或维持信息；

128个周期（64 μs）用来刷新操作；

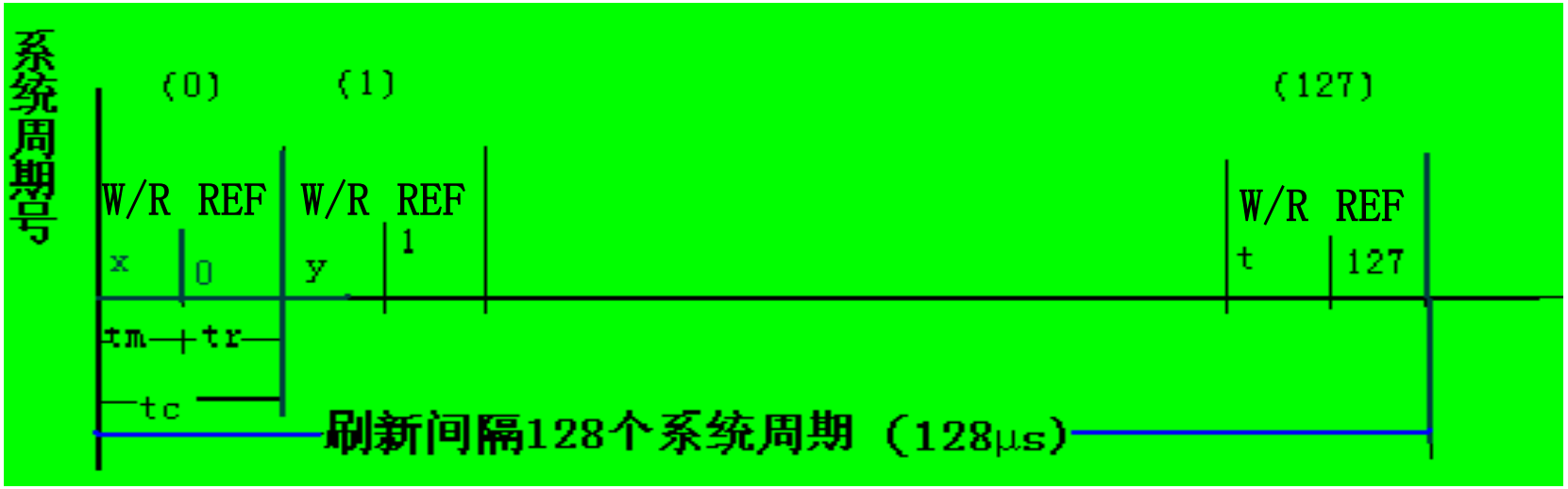
当3781个周期结束，便开始进行128个周期，64 μs的刷新操作。

集中式刷新适用于高速存储器。

存在不能进行读写操作的死区时间.

• 分散式刷新

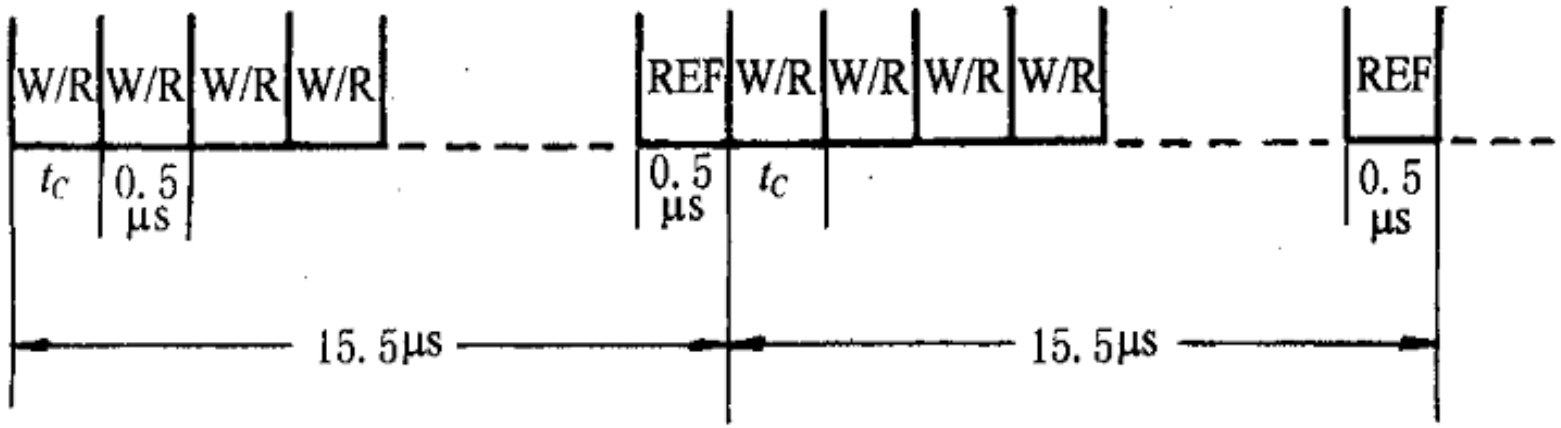
把一个存储周期t_c分为两半，周期前半段时间t_m用来读/写操作或维持信息，周期后半段时间t_r作为刷新操作时间。这样，每经过128个系统周期时间，整个存储器便全部刷新一遍。



分散式刷新系统速度降低，但不存在停止读写操作的死时间。

• 异步式刷新

是前两种方式的结合。



例如：对2116来说，在2ms中内把128行刷新一遍。

2000μs ÷128 ≈ 15.5 μs

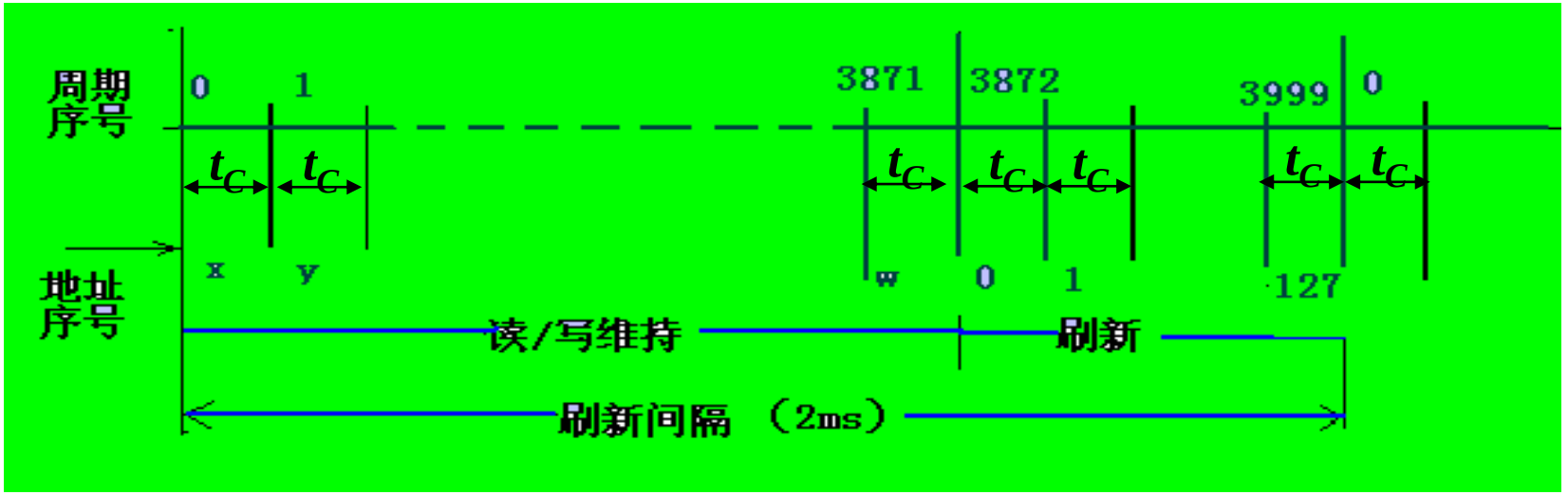
即：每15.5 μs 刷新一行。

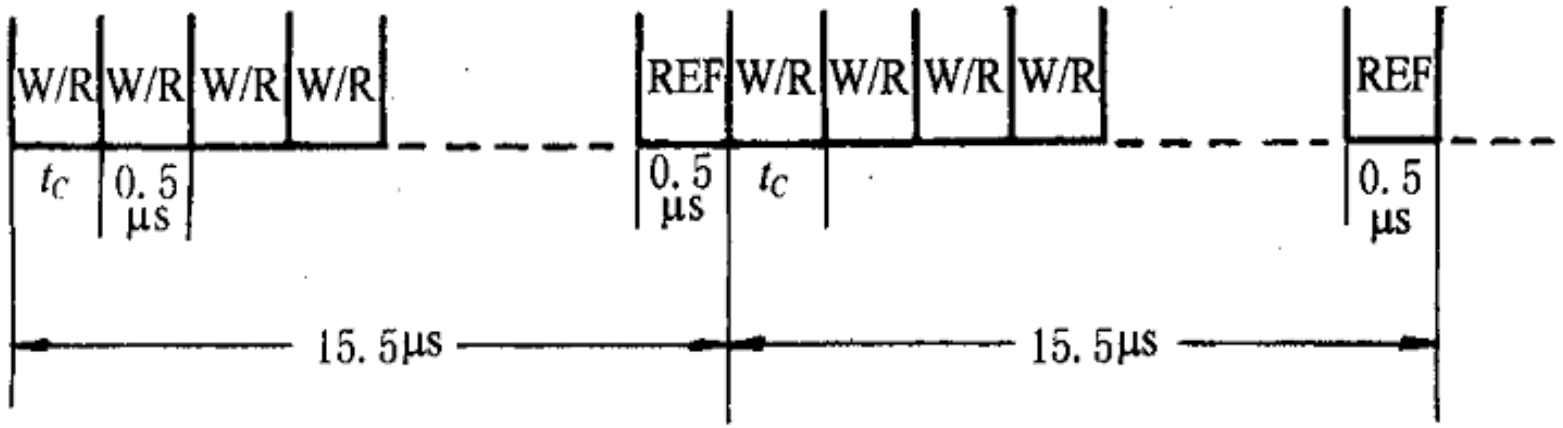
例：说明1M×1位DRAM片子的刷新方法，刷新周期定为8ms。

- 1M位的存储单元排列成 512×2048的矩阵；
- 如果选择一个行地址进行刷新， 刷新地址为A_0~A_8（2^9），因此这一行上的2048个存储元同时进行刷新；
- 在8ms内进行512个周期的刷新；
- 刷新方式可采用：

  在8ms中进行512次刷新操作的集中刷新方式；

  按8ms÷512＝15.5μs刷新一次的异步刷新方式。


周期序号
0
1
3871
3872
3999
0
t_C
x
y
w
0
1
127
地址序号
读/写维持
刷新
刷新间隔 （2ms）


W/R
W/R
W/R
W/R
REF
W/R
W/R
W/R
W/R
REF
t_C
0.5 μs
0.5 μs
t_C
0.5 μs
15.5μs
15.5μs

4. 存储器控制电路

DRAM存储器的刷新需要有硬件电路的支持，包括：

刷新计数器、

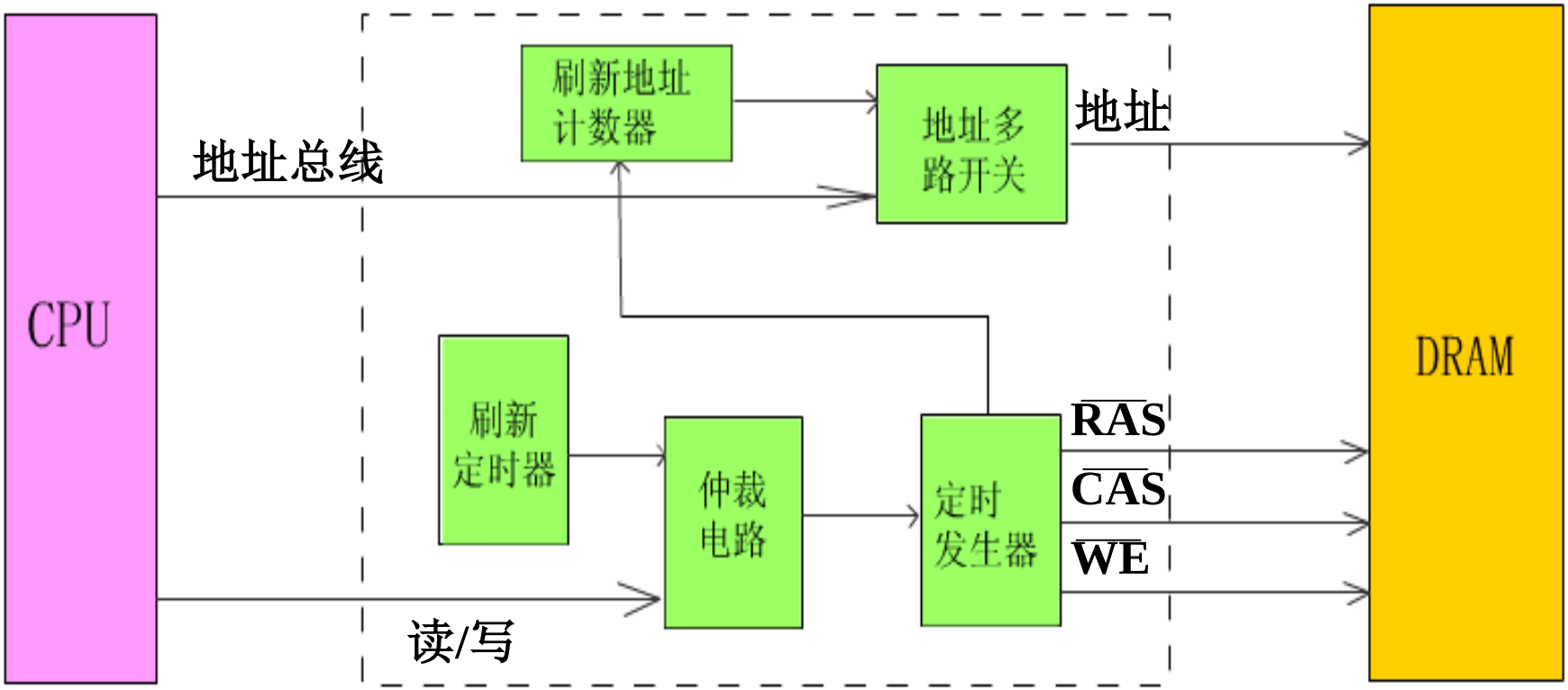
刷新/访存裁决、

刷新控制逻辑等。

这些控制线路形成DRAM控制器。

DRAM控制器是CPU和DRAM的接口电路，它将CPU的信号变换成适合DRAM片子的信号。

DRAM控制器


CPU
地址总线
刷新地址
计数器
地址多
路开关
地址
DRAM
刷新
定时器
仲裁
电路
定时
发生器
RAS
CAS
WE
读/写

(1) 地址多路开关

读写操作时向DRAM片子分时送出行地址和列地址；

刷新时需要提供刷新地址。

(2) 刷新定时器： 定时电路用来提供刷新请求。

(3) 刷新地址计数器：只用$\overline{\text{RAS}}$信号的刷新操作，需要提供刷新地址计数器。对于1M位的片子，需512个地址，故刷新计数器9位。

(4) 仲裁电路：对同时产生的来自CPU的访问存储器的请求和来自刷新定时器的刷新请求的优先权进行裁定。

(5) 定时发生器：提供行地址选通信号$\overline{\text{RAS}}$、列地址选通信号$\overline{\text{CAS}}$和写信号$\overline{\text{WE}}$.

高性能存储器

DDR 2 规格	核心频率 (MHz)	工作电压 (V)	引脚数目 (PIN)	传输带宽 (MB/S)
DDR2 400	100	1.8	200	3200
DDR2 533	133	1.8	200	4300
DDR2 667	167	1.8	200	5300

DDR 3 规格	核心频率 (MHz)	工作电压 (V)	引脚数目 (PIN)	传输带宽 (MB/S)
DDR3 800	100	1.5	204	6400
DDR3 1066	133	1.5	204	8500
DDR3 1333	167	1.5	204	10700

DDR4	核心频率（**MHZ**）	工作电压（**V**）	引脚数	传输带宽
DDR4	2400	1.2	288/260	29.22GB/s
DDR4	3200	1.2	288/260	37.52GB/s

存储器的基本组织

(1) 与CPU的连接

主要是 地址线、控制线、数据线 的连接。

(2) 多个芯片连接

存储器容量与实际存储器的要求多有不符。 如前所述存储器芯片有不同的组织形式，如1024*1、1024*4、4096*8等;

实际使用时，需进行字和位扩展(多个芯片连接），组成你所需要的实际的存储器，如 1K*8、4K*8 等的存储器。

另一当面，系统本身需要将多种类型的存储器（ROM+RAM）。

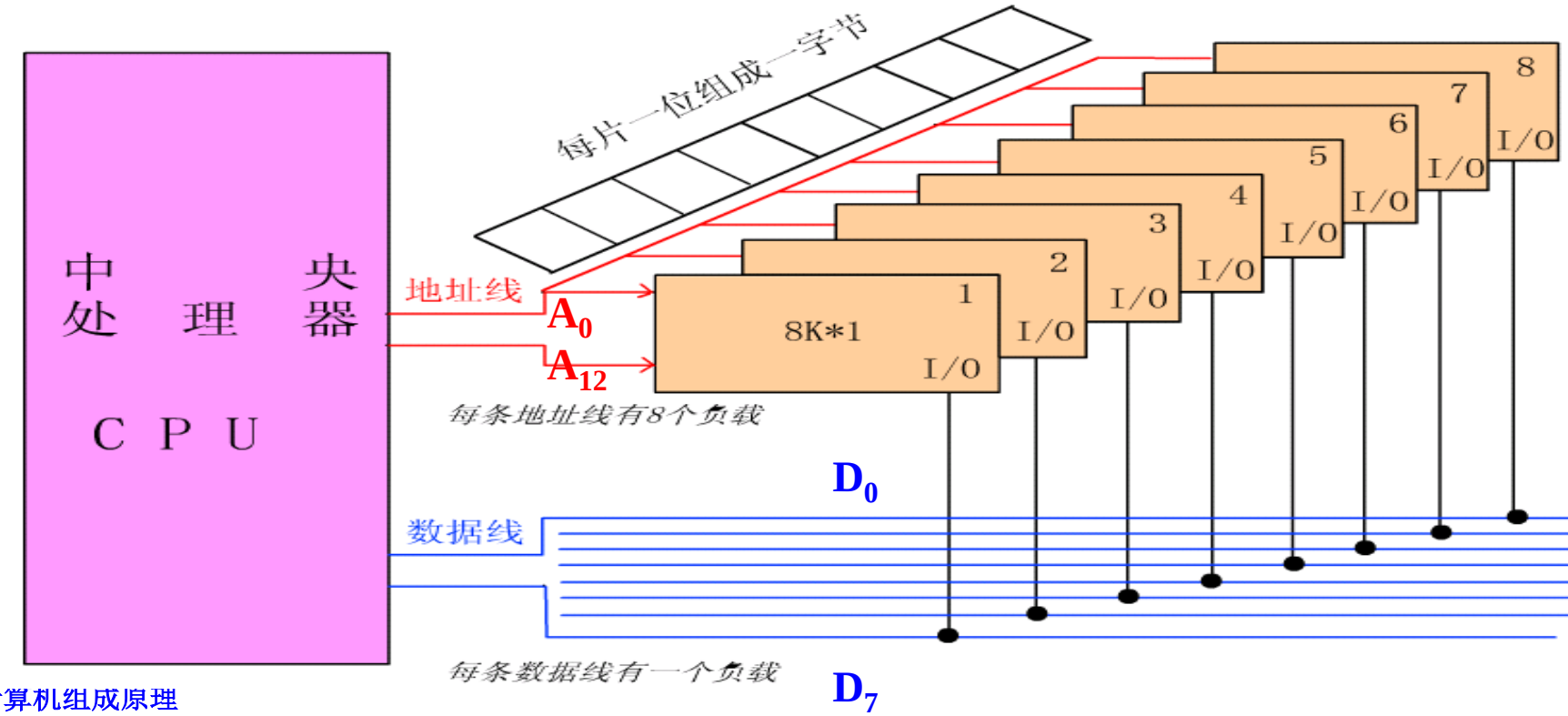
位扩展法

只加大字长，而存储器的字数与存储器芯片字数一致，对片子没有选片要求。

用8k*1的片子组成8k*8的存储器需 8 个芯片

地址线——需 13 根　　数据线——8 根

控制线—— $\overline{WR}$接存储器的$\overline{WE}$

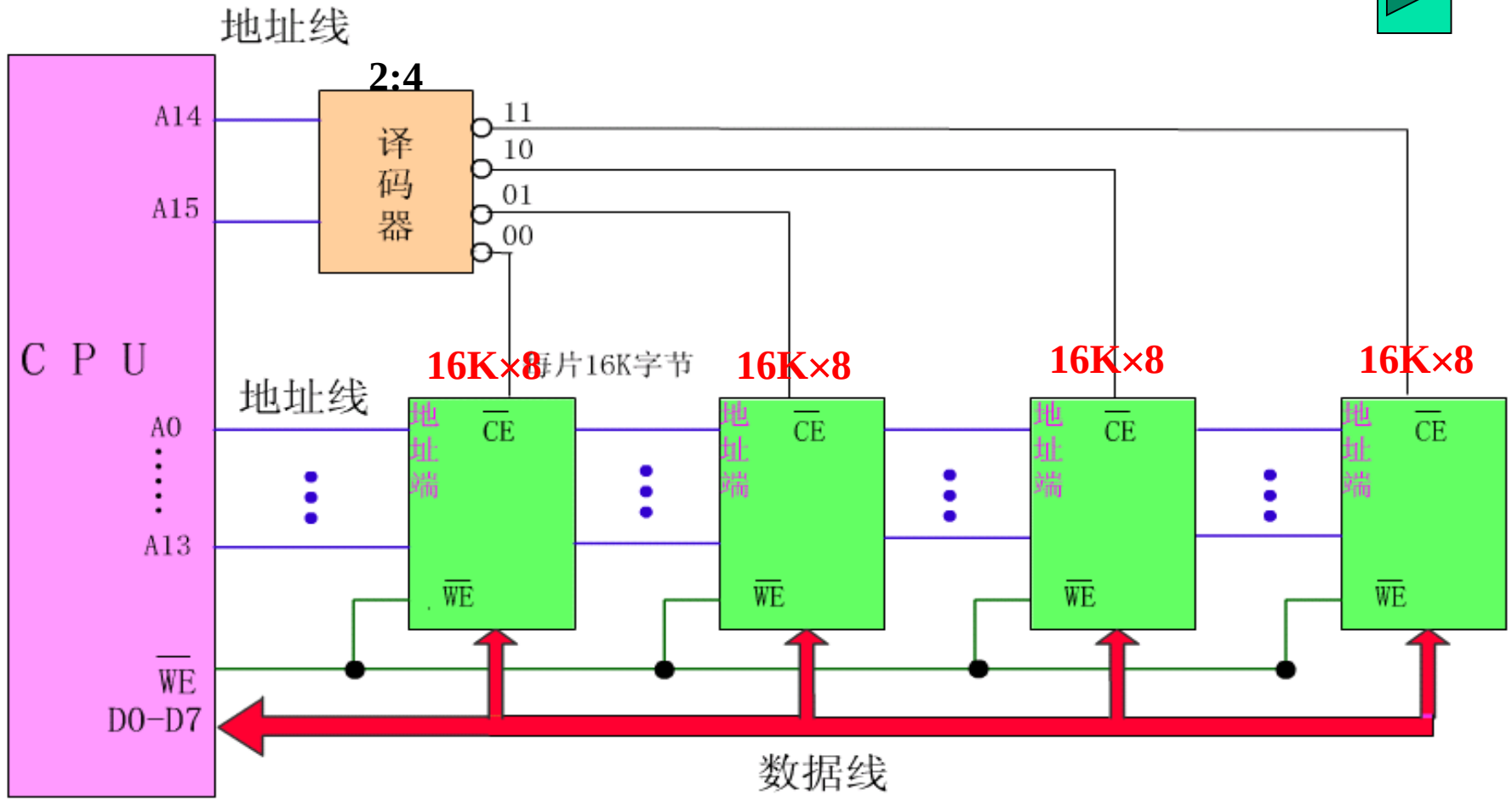


字扩展法

用16K×8位的芯片组成64K×8位的存储器需4个芯片

地址线—— 共需16根 片内：(2^{14} = 16384) 14根，选片：2根

数据线—— 8根　　控制线 —— $\overline{WE}$

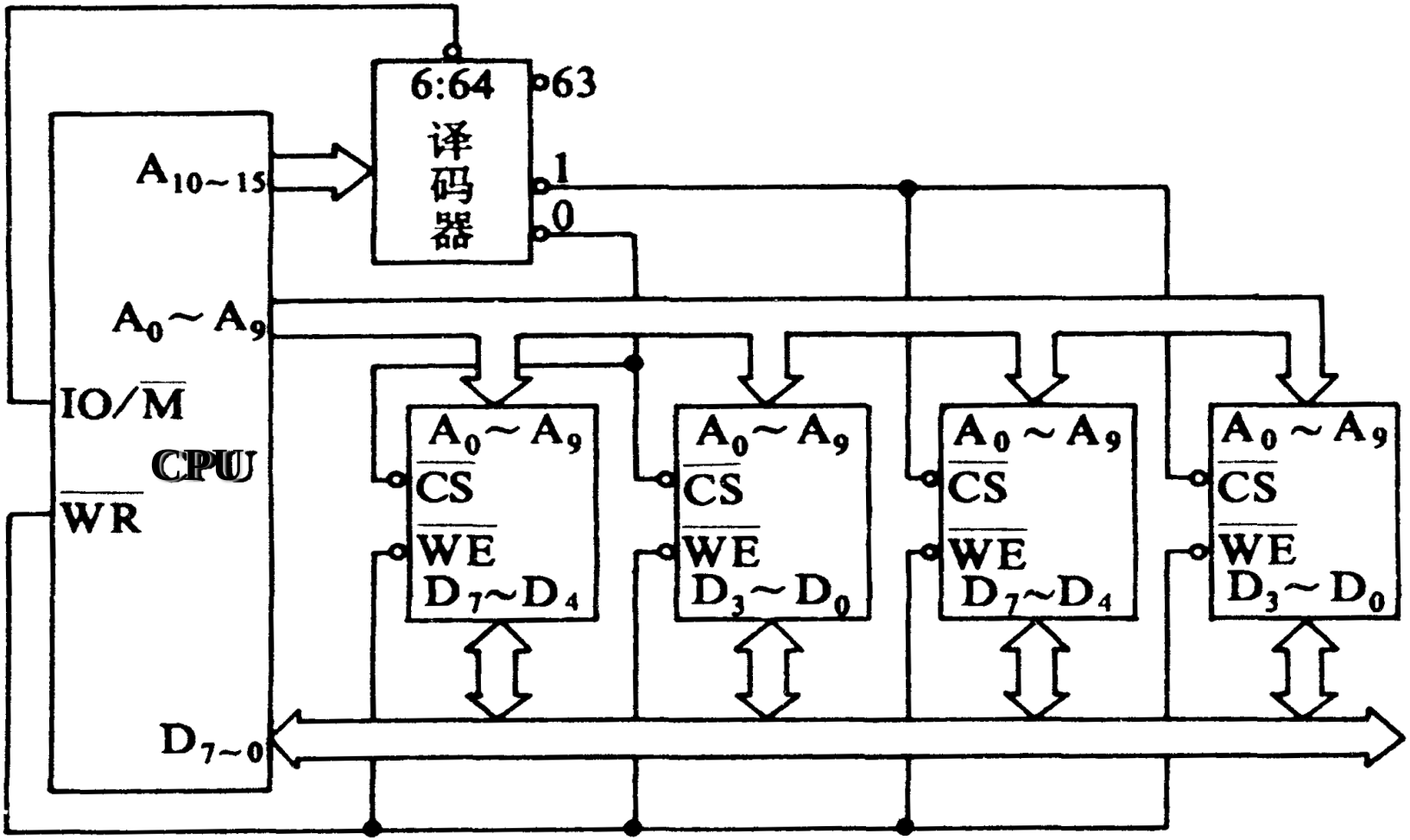


地址空间分配表

地址 / 片号	选片 A_{15} A_{14}	片内 A_{13} A_{12} ……A_1 A_0	总地址	说明
1	00 00	00,0000,0000,0000 11,1111,1111,1111	0000 3FFF	最低地址 最高地址
2	01 01	00,0000,0000,0000 11,1111,1111,1111	4000 7FFF	最低地址 最高地址
3	10 10	00,0000,0000,0000 11,1111,1111,1111	8000 BFFF	最低地址 最高地址
4	11 11	00,0000,0000,0000 11,1111,1111,1111	C000 FFFF	最低地址 最高地址

字位同时扩展法

用1k × 4 的存储器芯片 2114 组成 2k × 8 的存储器



例 有若干片1M×8位的SRAM芯片，采用字扩展方法构成4MB存储器，问
(1) 需要多少片RAM芯片？
(2) 该存储器需要多少地址位？
(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R/W#。
(4) 给出地址译码器的逻辑表达式。

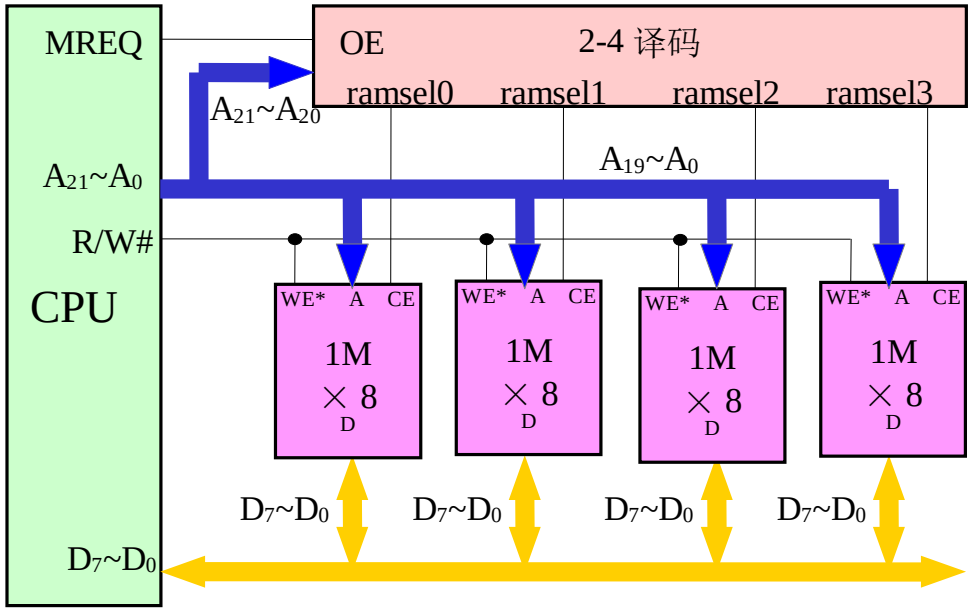
解：(1) 需要4M/1M = 4片SRAM芯片；
(2) 需要22条地址线
(3) 译码器的输出信号逻辑表达式为：

ramsel0 = $\overline{A21}$ * $\overline{A20}$ *MREQ

ramsel1 = $\overline{A21}$ *A20*MREQ

ramsel2 = A21* $\overline{A20}$ *MREQ

ramsel3 = A21*A20*MREQ



课堂练习：

设有若干片256K×8位的SRAM芯片，问：

(1) 采用字扩展方法构成2048KB的存储器需要多少片SRAM芯片？

(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R/W#。

例 设有若干片256K×8位的SRAM芯片，问：

(1) 采用字扩展方法构成2048KB的存储器需要多少片SRAM芯片？

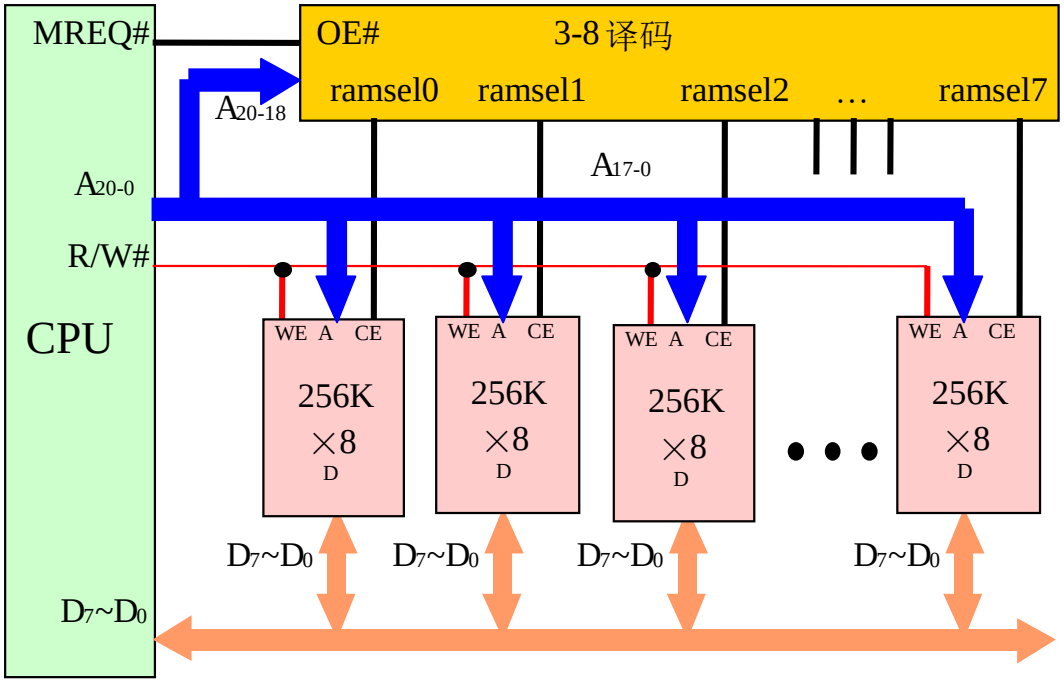
(2) 该存储器需要多少字节地址位？

(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R/W#。

解：（1）该存储器需要2048K/256K = 8片SRAM芯片；

（2）需要21条地址线，因为2^{21}=2048K，其中高3位用于芯片选择，低18位作为每个存储器芯片的地址输入。

（3）该存储器与CPU连接的结构图如下。



例 设有若干片256K×8位的SRAM芯片，问：

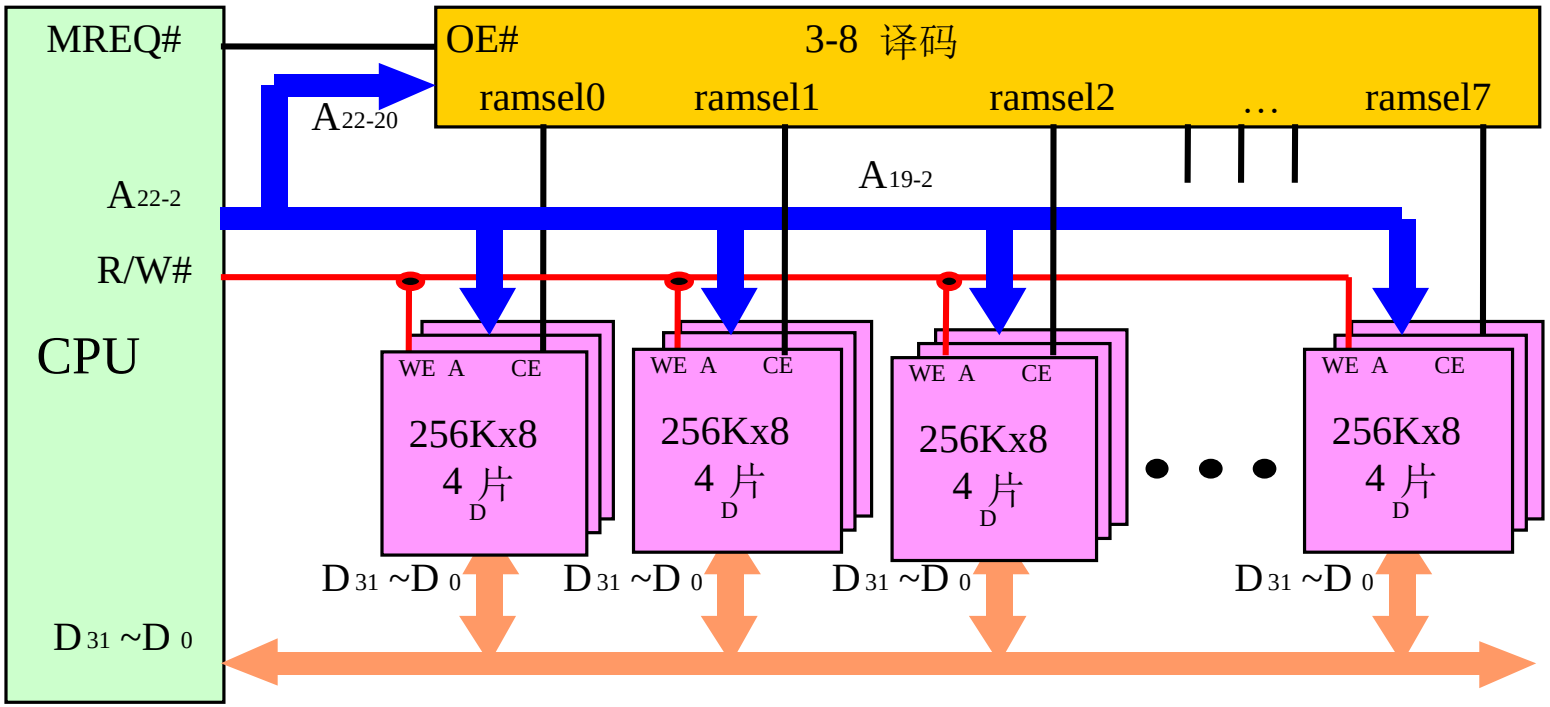
(1) 如何构成2048K×32位的存储器？

(2) 需要多少片RAM芯片？

(3) 该存储器需要多少字节地址位？

(4) 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R/W#。

解： 采用字位扩展的方法。需要32片SRAM芯片。



只读存储器

闪速存储器

高速存储器

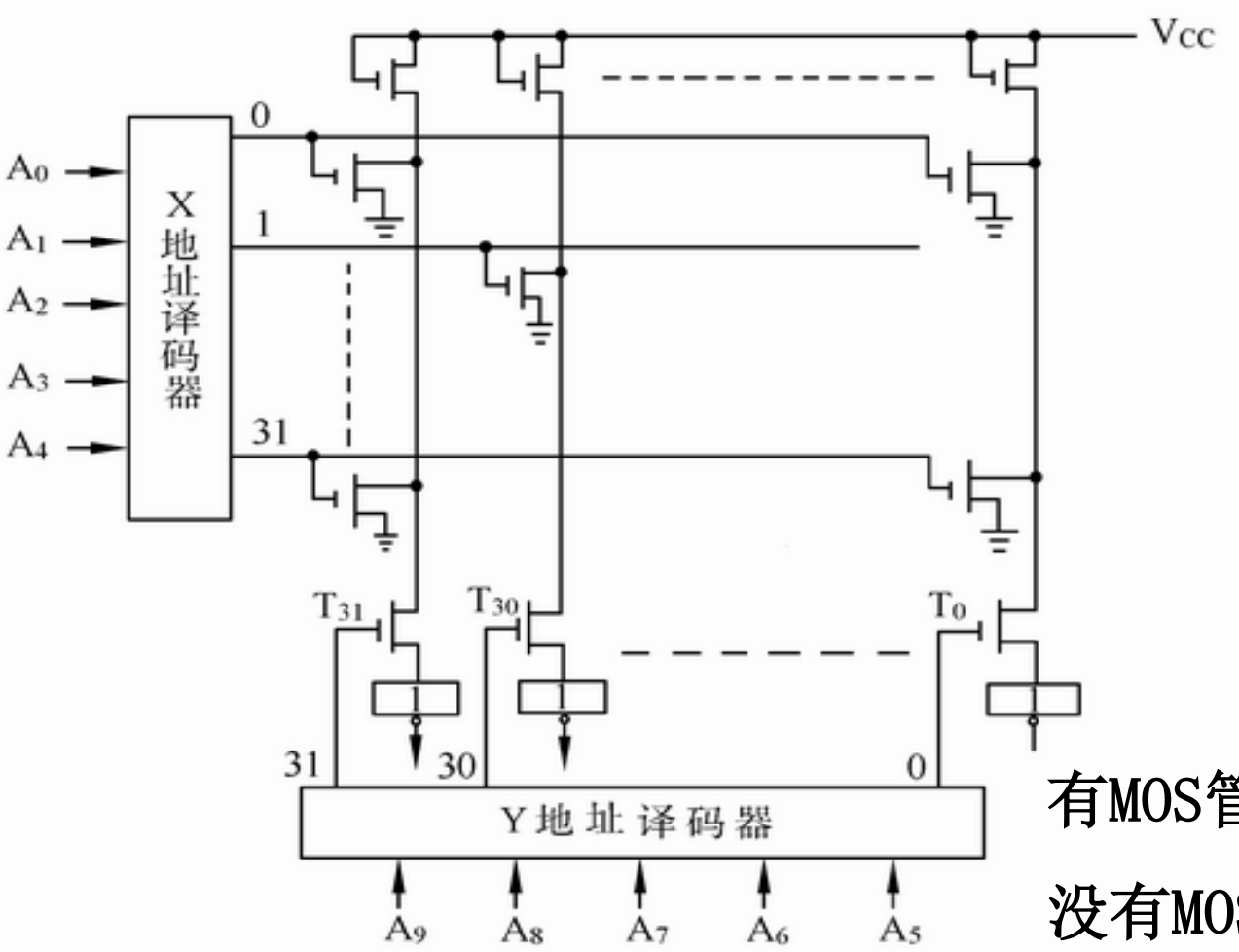
只读存储器

1. ROM的分类

只读存储器	定　　义	优　　点	缺　点
掩模式(ROM)	数据在芯片制造过程中就确定	可靠性和集成度高，价格便宜	不能重写
一次编程(PROM)	用户可自行改变产品中某些存储元	可以根据用户需要编程	只能一次性改写
多次编程(EPROM)(EEPRPM)	可以用紫外光照 射或电擦除原来的数据，然后再重新写入新的数据	可以多次改写ROM中的内容	
闪速存储器 Flash memory			

(1) 掩模式ROM

采用掩模工艺制成，其内容由厂方生产时写入，用户只能读出使用而不能改写。

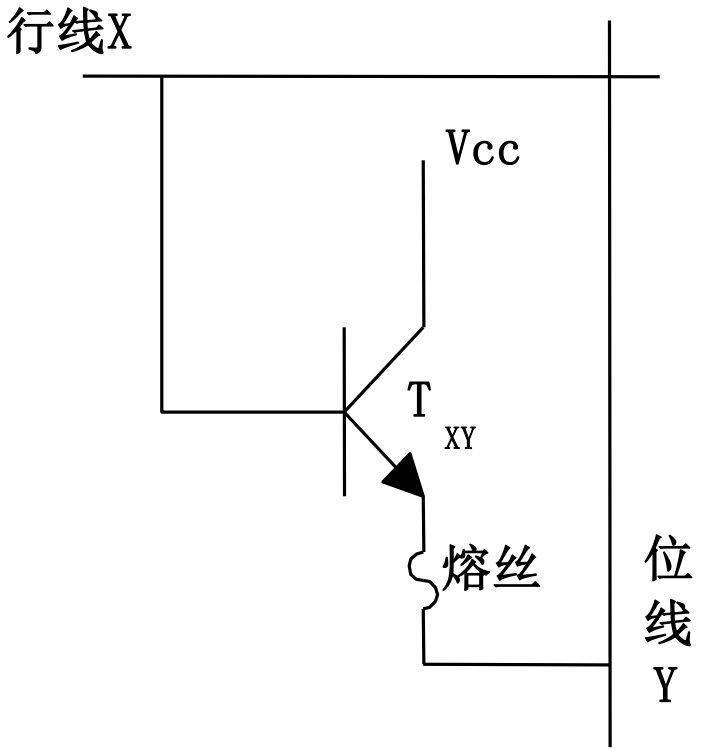


有MOS管的位表示存1，

没有MOS管的位表示存0。

(2) 可写入（可编程）只读存储器PROM

例：熔丝烧断型



写“0”时：

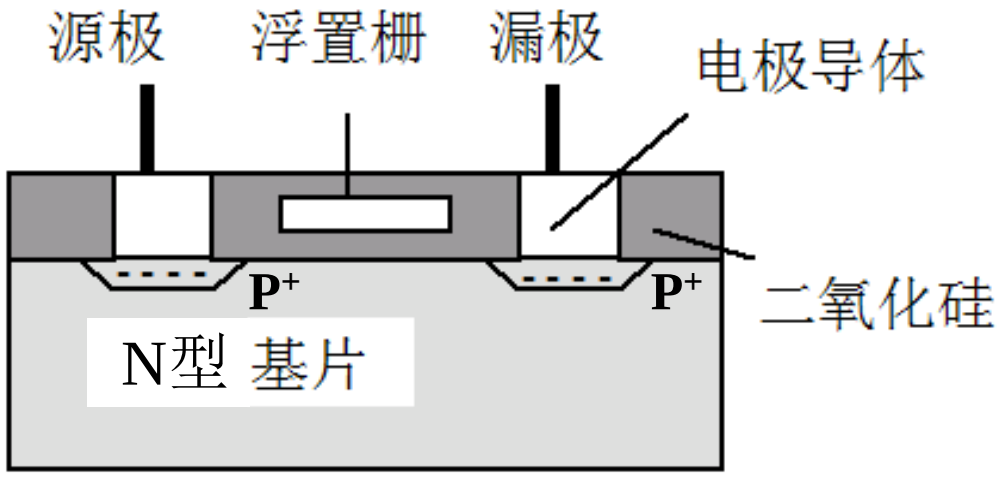
烧断熔丝

写“1”时：

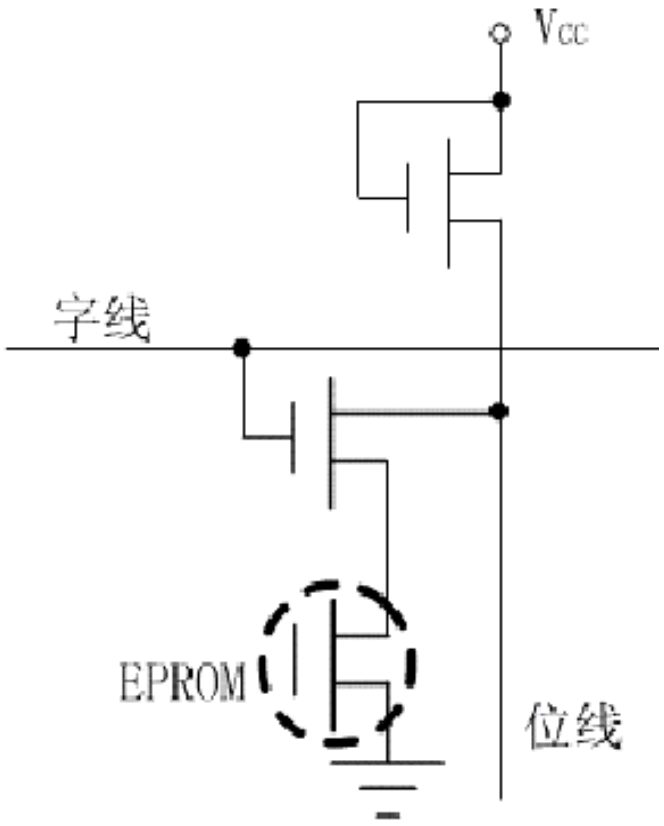
保留熔丝

(3)光擦可编程只读存储器EPROM

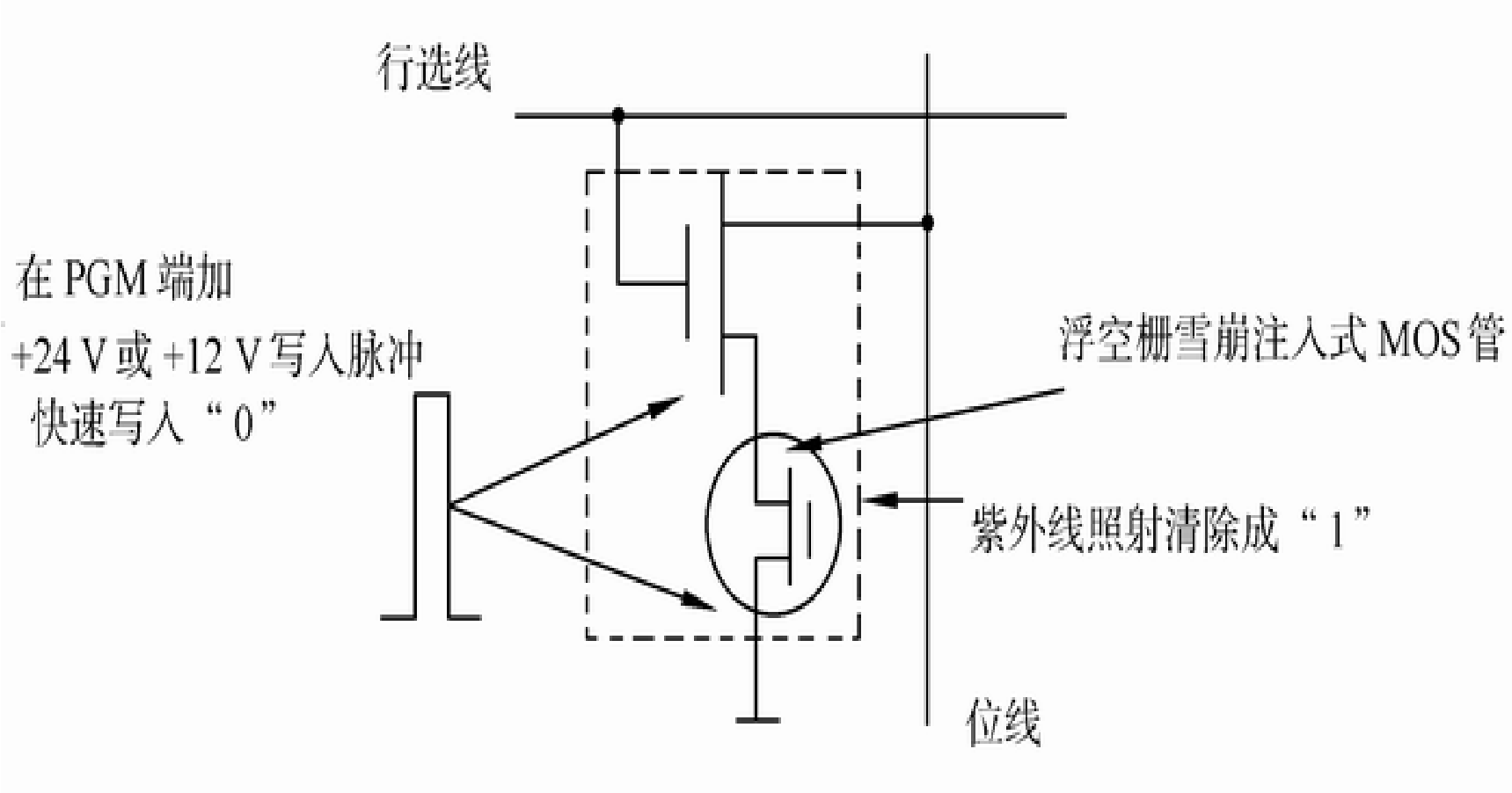
- 基本存储元电路



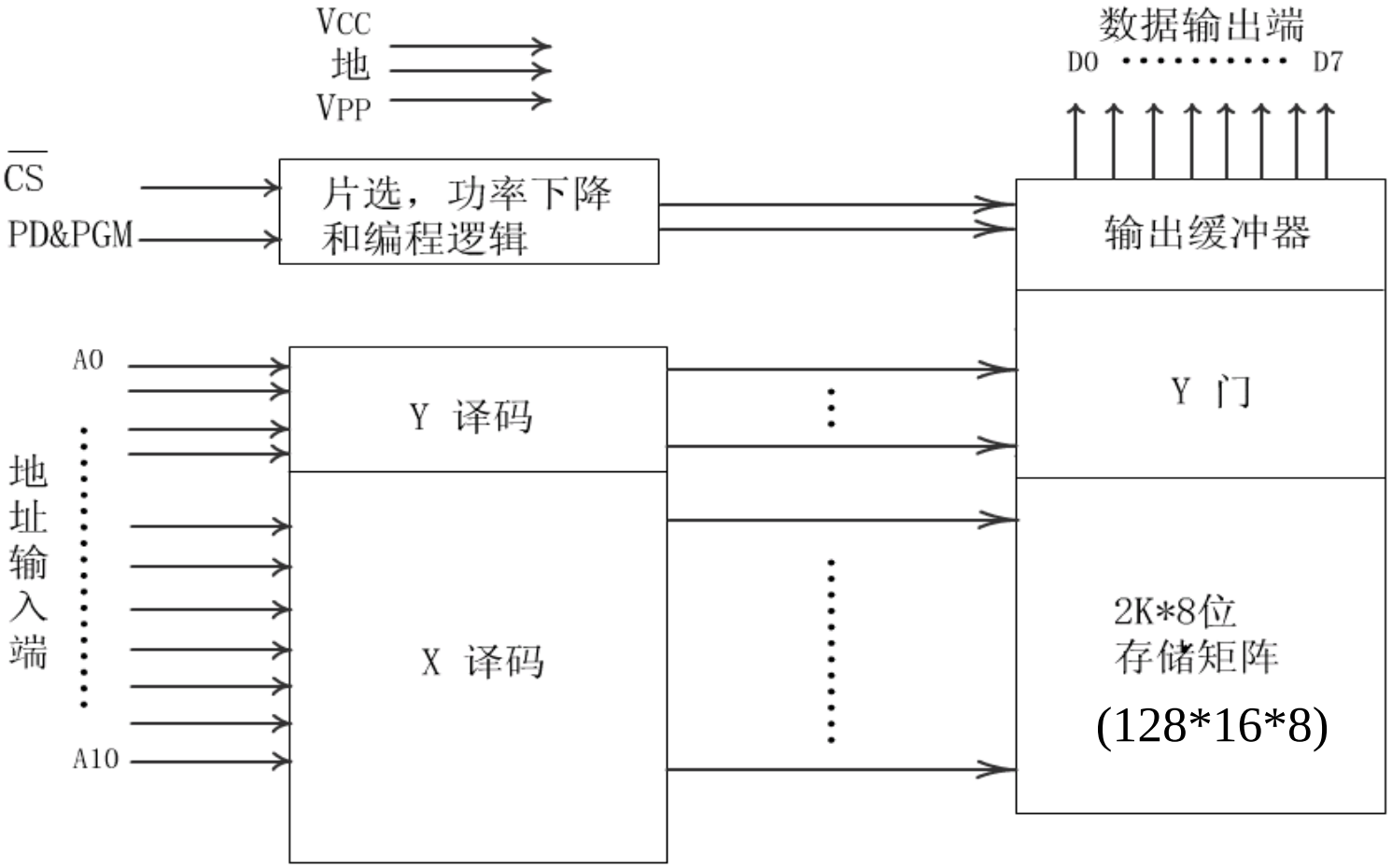
(a) 单元结构



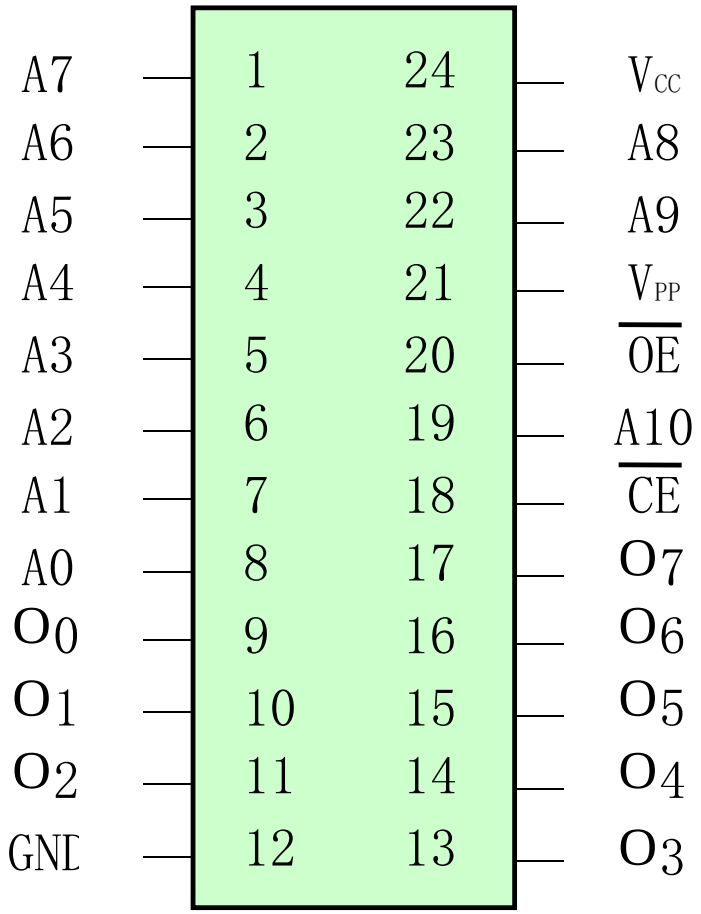
EPROM的基本存储元


行选线
在 PGM 端加
+24 V 或 +12 V 写入脉冲
快速写入“0”
浮空栅雪崩注入式 MOS 管
紫外线照射清除成“1”
位线

EPROM实例


VCC
地
VPP
数据输出端
D0 ····· D7
CS
PD&PGM
片选，功率下降
和编程逻辑
输出缓冲器
A0
地址输入端
A10
Y 译码
Y 门
X 译码
2K*8位
存储矩阵
(128*16*8)

EPROM实例



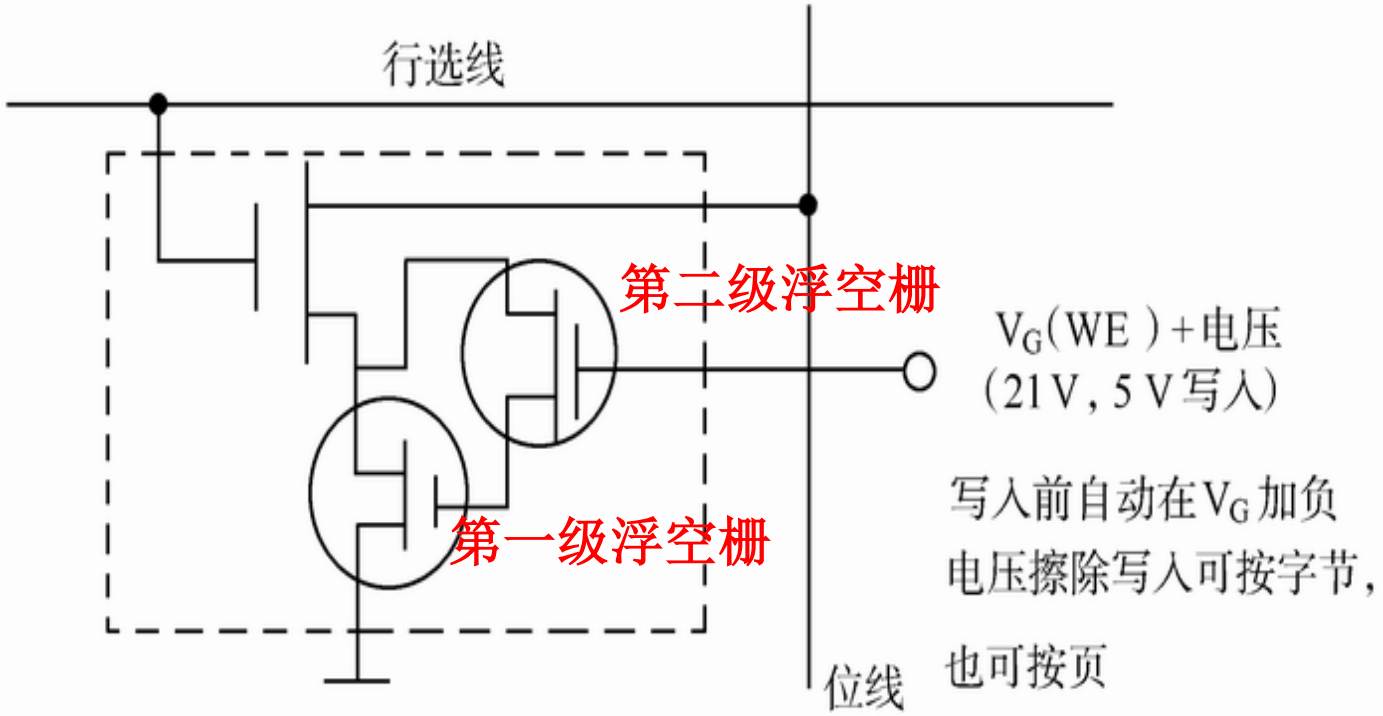
工作模式选择

操作 \ 引脚	PD/PGM	CS	Vpp	Vcc	数据输出
读	低	低	+5V	+5V	输出
未选中	无关	高	+5V	+5V	高阻
功率下降	高	无关	+5V	+5V	高阻
编程	由低到高脉冲	高	+25V	+5V	输入

EPROM 2716 2K×8引脚

(4) 电擦可编程只读存储器EEPROM

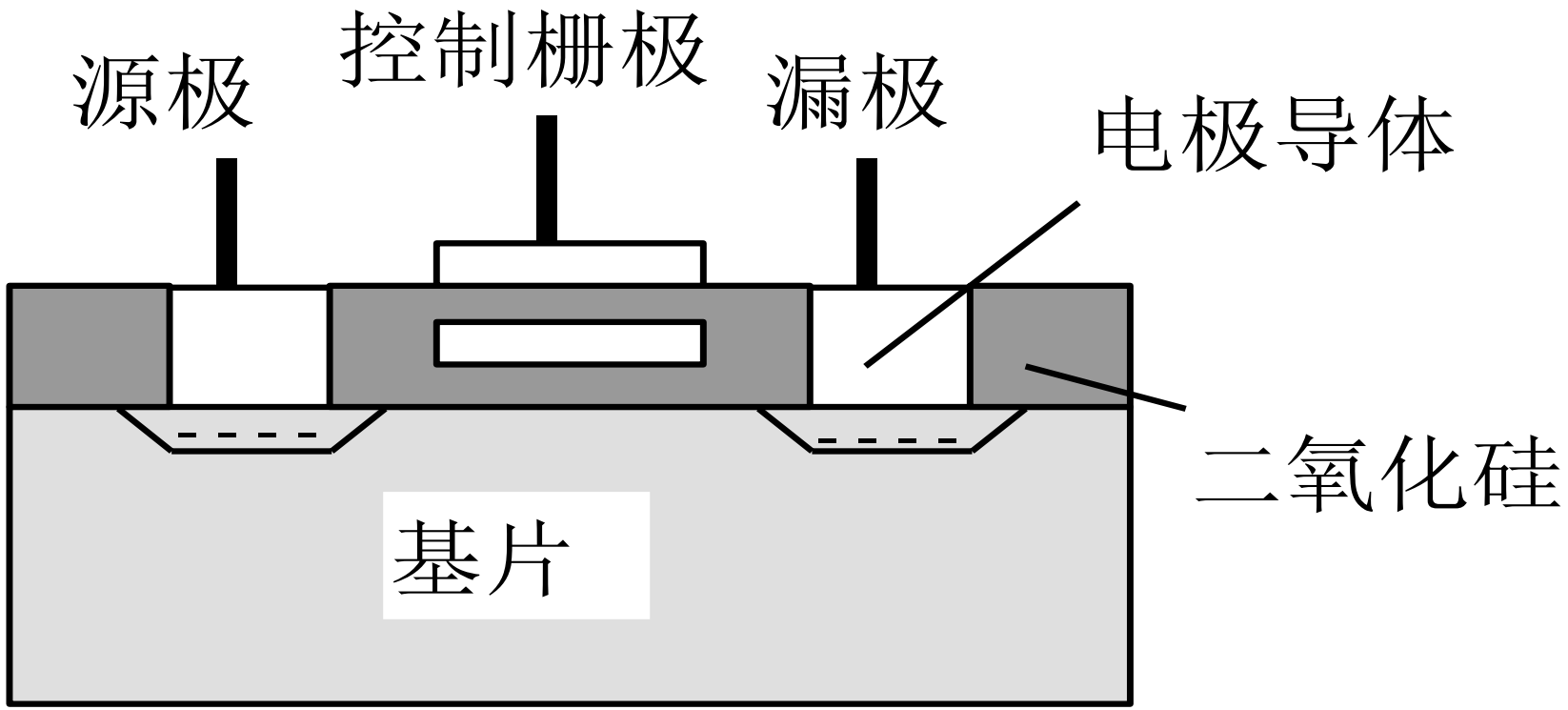
它的主要特点是能在应用系统中在线改写，断电后信息保存，因此目前得到广泛应用。



- 若V_G为正电压，第一浮空栅极与漏极之间产生隧道效应，使电子注入第一浮空栅极，即编程写入。
- 若使V_G为负电压，强使第一级浮空栅极的电子散失，即擦除。
- EEPROM的编程与擦除电流很小，可用普通电源供电，而且擦除可按字节进行。

电可擦写ROM

——EEPROM及Flash存储器



2. 存储器举例

(1) CPU的地址总线16根(A_{15}—A_0，A_0为低位)；双向数据总线8根(D_7—D_0)，控制总线中与主存有关的信号有：

$\overline{MREQ}$，$R/\overline{W}$。

(2) 主存地址空间分配如下：

0—8191为系统程序区，由只读存储芯片组成；

8192—32767为用户程序区；最后(最大地址)2K地址空间为系统程序工作区。

(3) 现有如下存储器芯片：

EPROM：8K×8位(控制端仅有$\overline{CS}$);

SRAM：16K×1位，2K×8位，4K×8位，8K×8位.

请从上述芯片中选择适当芯片设计该计算机主存储器，画出主存储器逻辑框图，注意画出选片逻辑(可选用门电路及3：8译码器74LS138)与CPU 的连接，说明选哪些存储器芯片，选多少片。

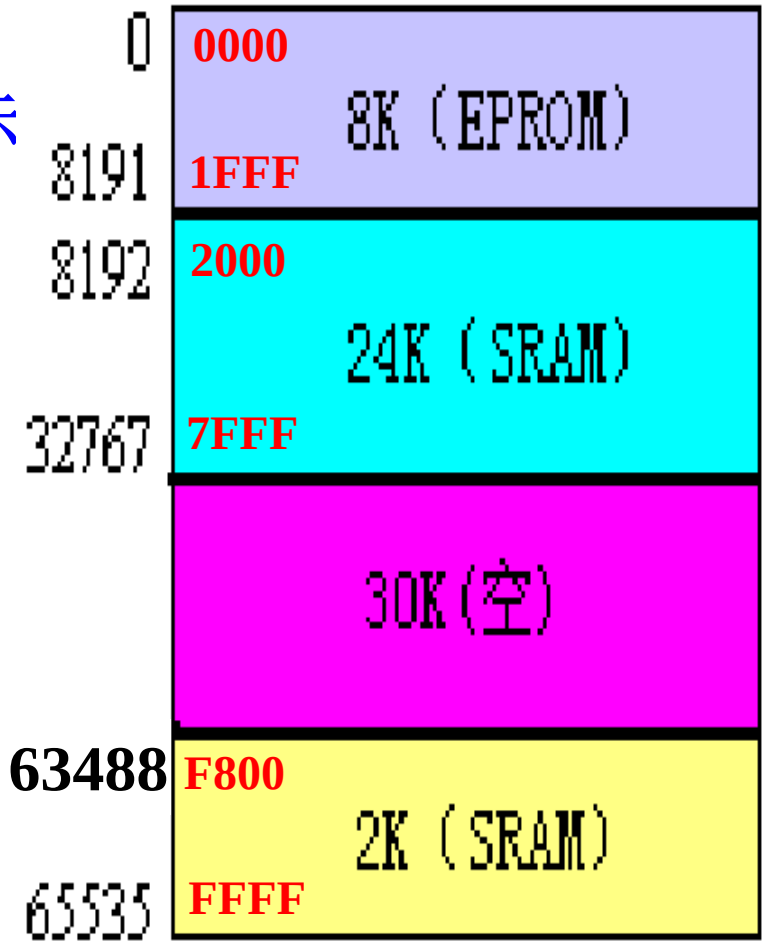
解： (1) 主存地址空间分布如图所示

16根地址线寻址 —— 64K

0000 ~ FFFFH(65535)

EPROM：8K×8位

SRAM：16K×1位，2K×8位，4K×8位，8K×8位.



(2) 连接电路

片内寻址：

8K芯片——片内13根　　A_{12}~A_0

2K芯片——片内11根　　A_{10}~A_0

片间寻址：

前32K　$A_{15}A_{14}A_{13}$

0　0　0

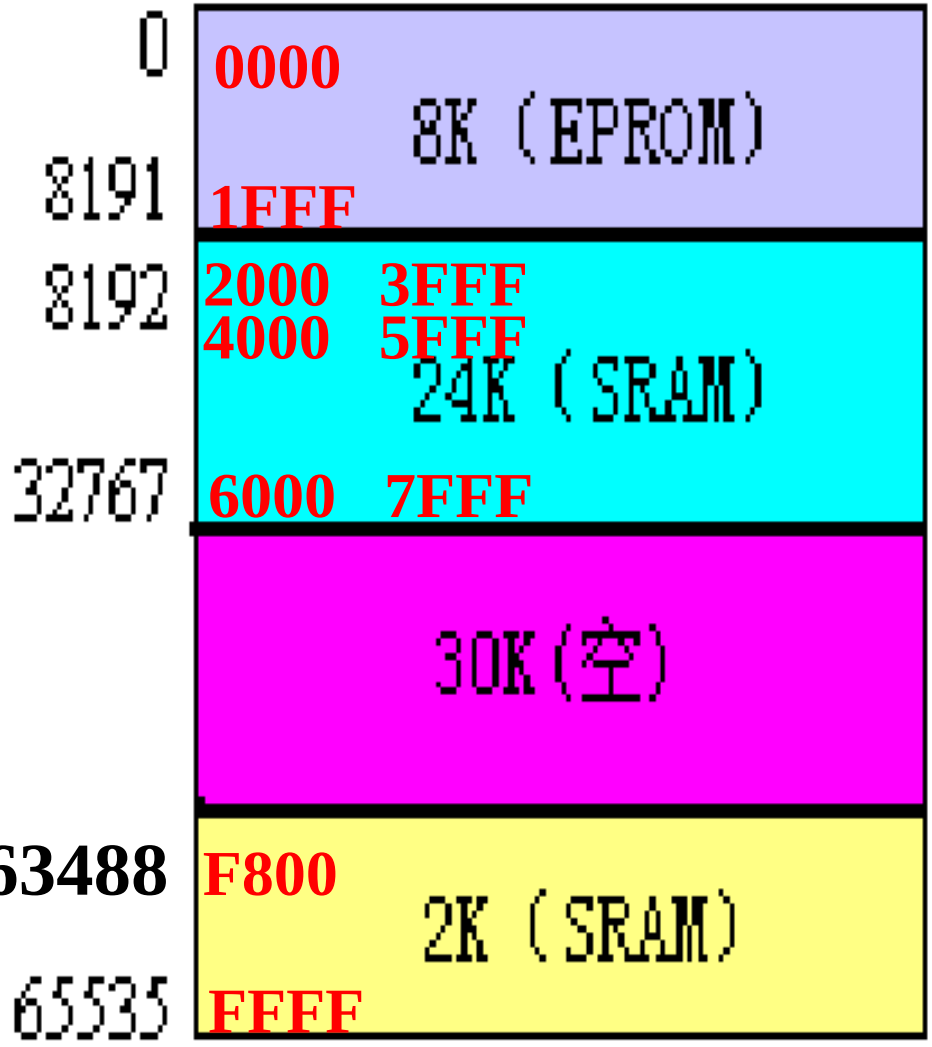
0　0　1

0　1　0

0　1　1

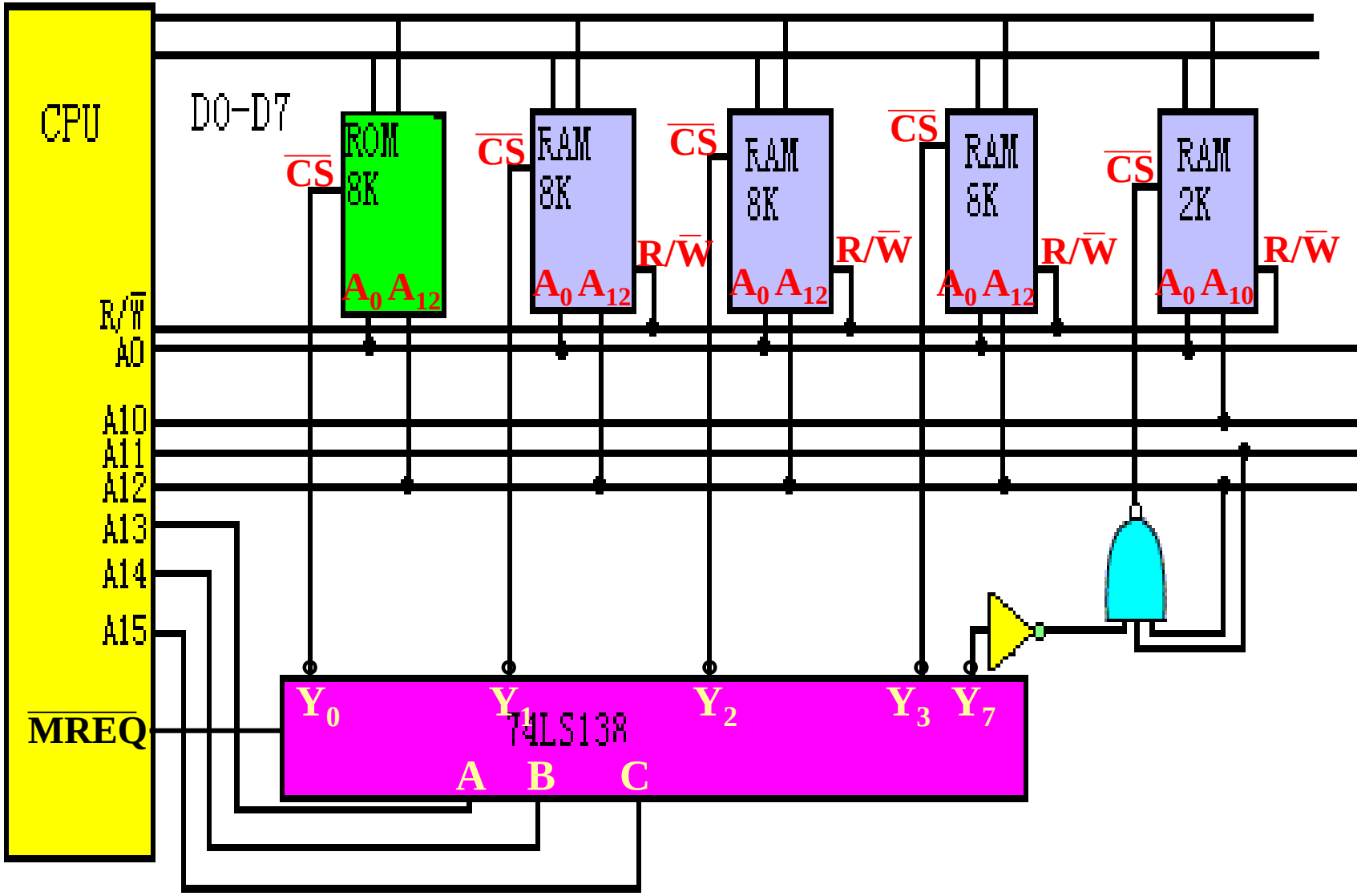
最后2K　1　1　1　加 $A_{12}A_{11}$

1　1



课堂练习：

画出主存储器逻辑框图，注意画出选片逻辑(可选用门电路及3：8译码器74LS138)与CPU 的连接，说明选哪些存储器芯片，选多少片。


CPU
D0-D7
ROM
8K
RAM
8K
RAM
8K
RAM
8K
RAM
2K
CS
R/W
A0 A12
A0 A10
R/W
A0
A10
A11
A12
A13
A14
A15
MREQ
74LS138
Y0
Y1
Y2
Y3
Y7
A B C

闪速存储器

1. 什么是闪速存储器

Flash Memory

闪速存储器是一种高密度、非易失性的读/写半导体存储器，它突破了传统的存储器体系，改善了现有存储器的特性。

特点：

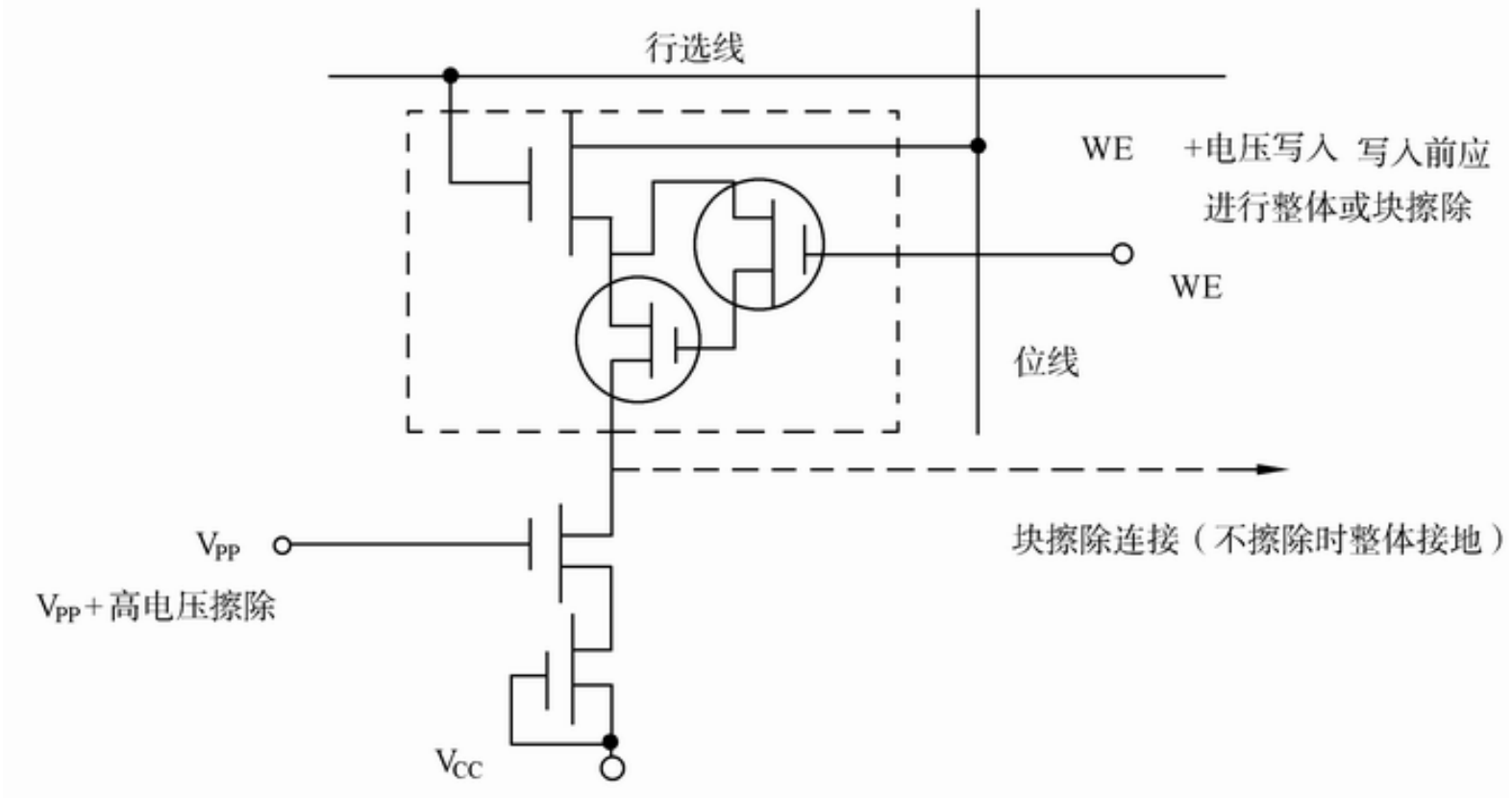
(1) 固有的非易失性

(2) 廉价的高密度

(3) 可直接执行

(4) 固态性能

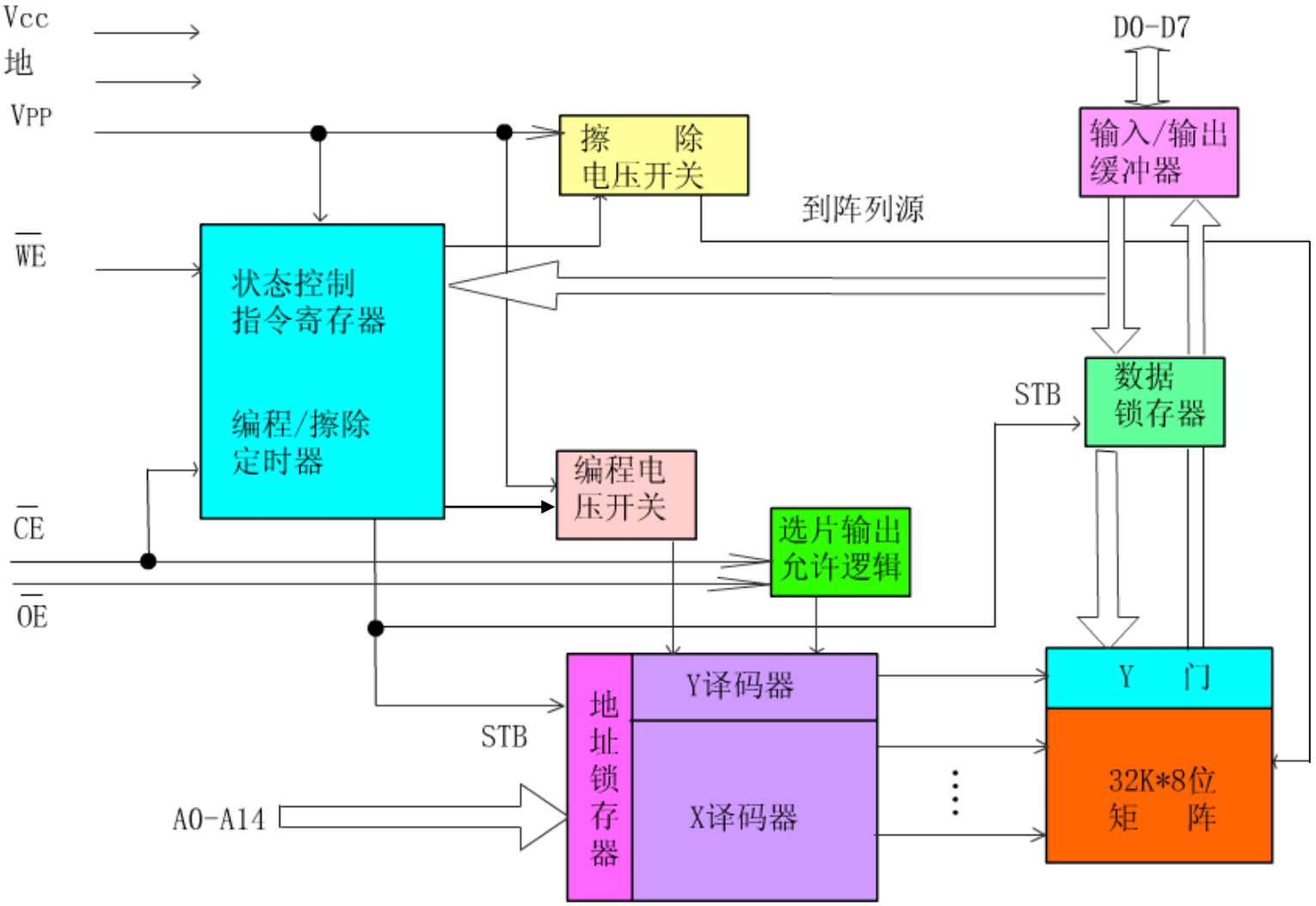
2. 基本单元电路



擦除方法是在源极加正电压利用第一级浮空栅与源极之间的隧道效应，把注入至浮空栅的负电荷吸引到源极。由于利用源极加正电压擦除，因此各单元的源极联在一起，这样，快擦存储器不能按字节擦除，而是全片或分块擦除。

3. 闪速存储器的逻辑结构

28F256A, 存储容量256K位 (32K*8)



(整体擦除Flash Memory)

4. 闪速存储器的工作原理

电擦除和重新编程能力

闪速存储器是在EPROM功能基础上增加了电路的电擦除和重新编程能力。28F256A引入一个指令寄存器来实现这种功能。其作用是：

(1)保证TTL电平的控制信号输入；

(2)在擦除和编程过程中稳定供电；

(3)最大限度的与EPROM兼容。

高速存储器

由于CPU和主存储器在速度上不匹配，限制了高速计算。

为了使CPU不至因为等待存储器读写操作的完成而无事可做，可以采取一些加速CPU和存储器之间有效传输的特殊措施。

(1) 芯片技术

研究开发高性能芯片技术，如： DRAM→FPMD→EDO→EDRAM→CDRAM→SDRAM→RambusDRAM→DDR2/3/4。

(2) 结构技术

- 采用并行操作方式 ---双端口存储器
- 采用并行主存储器, 提高读出并行性 ---多模块交叉存储器
- 主存储器采用更高速的技术来缩短存储器的读出时间 ---相联存储器

双端口存储器

多模块交叉存储器

相联存储器

高性能存储器

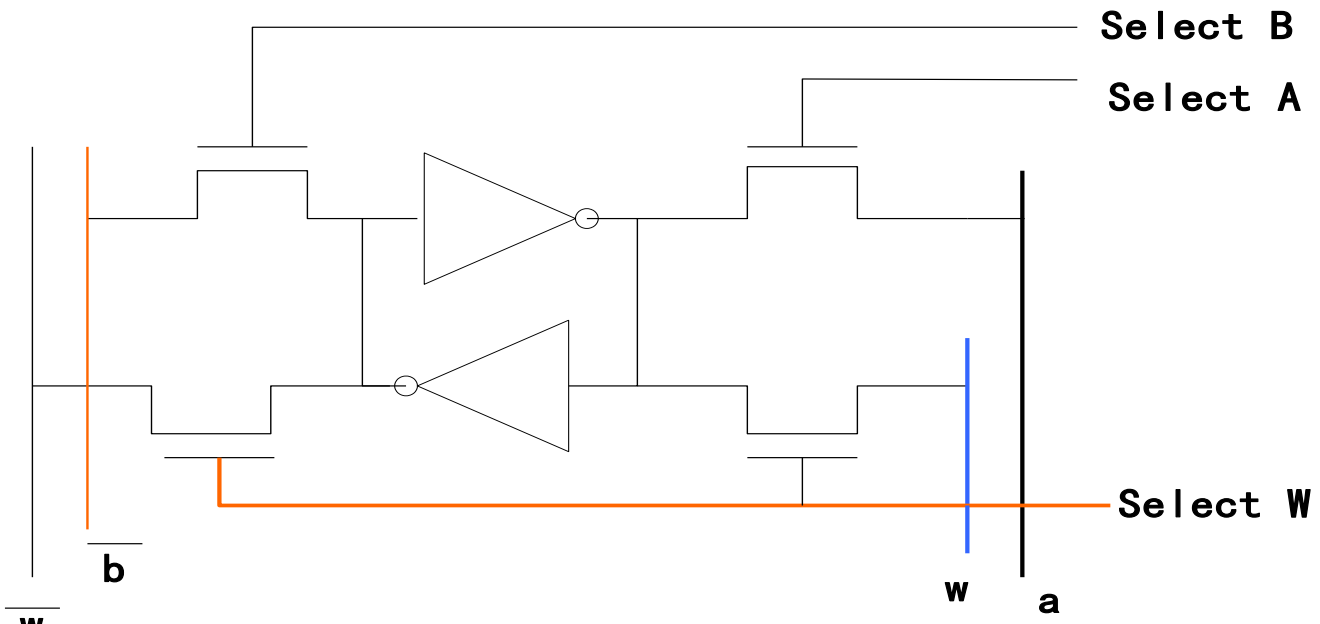
双端口存储器

1. 双端口存储器的逻辑结构

双端口存储器

——指同一个存储器具有两组相互独立的读写控制线路，是一种高速工作的存储器。

双读单写端口存储器单元结构

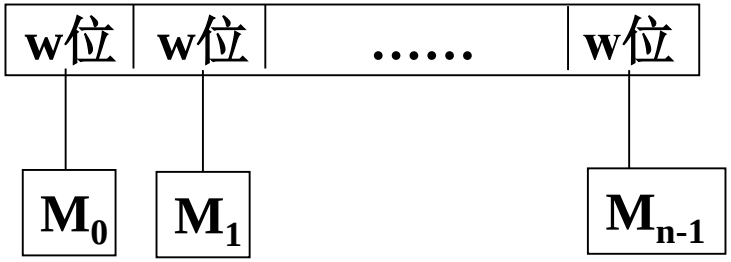


多模块交叉存储器

1. 并行主存系统

并行主存系统

大存储器在一个存储周期中读出的不是一个存储单元的w位信息，而是n个字，这样在单位时间里存储器提供的信息量可提高n倍，这样组织的主存系统称为并行主存系统。



2. 多模块交叉存储器

1） 存储器的模块化组织

一个由若干个模块组成的主存储器是线性编址的。

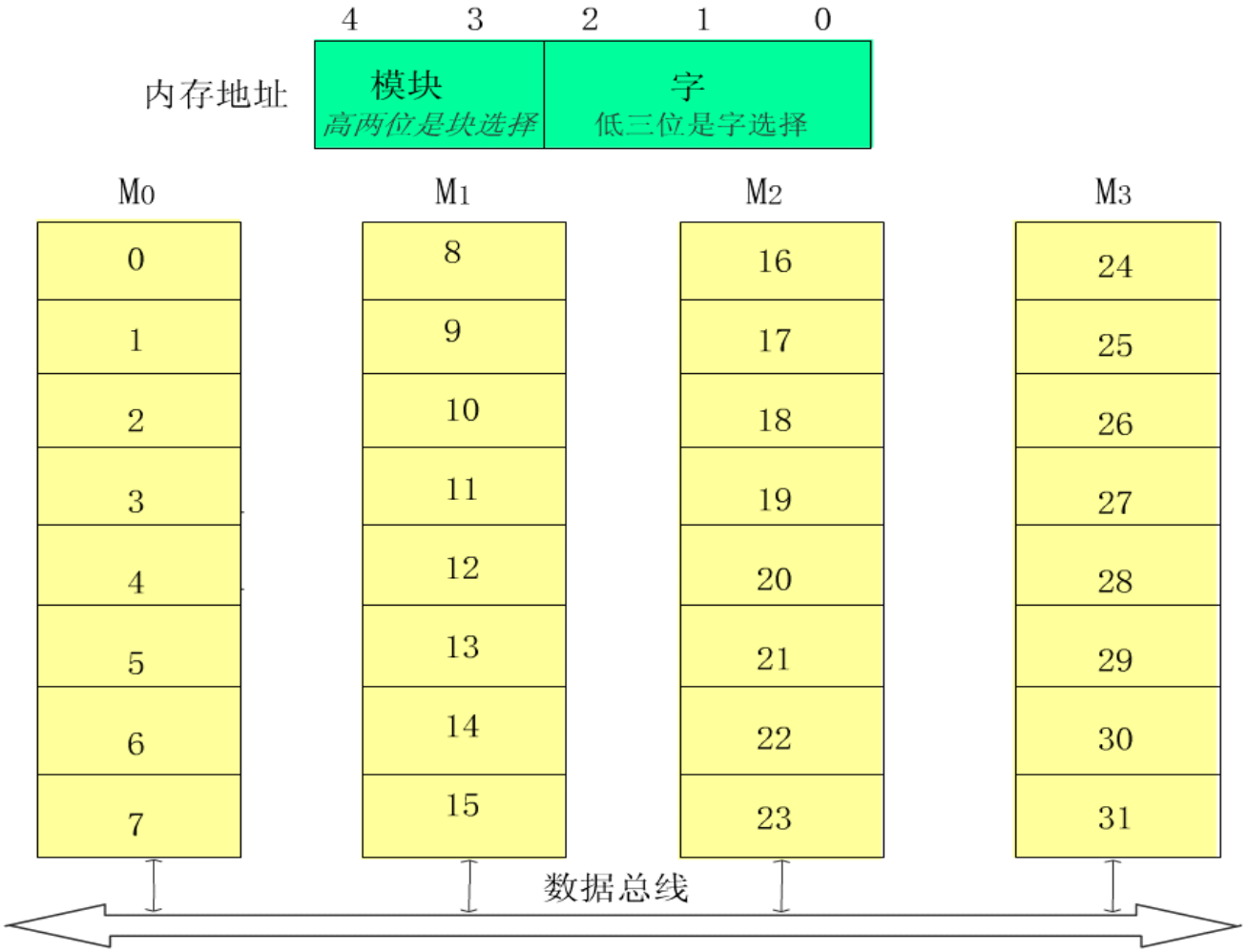
这些地址在各模块中有两种安排方式：

顺序方式

交叉方式

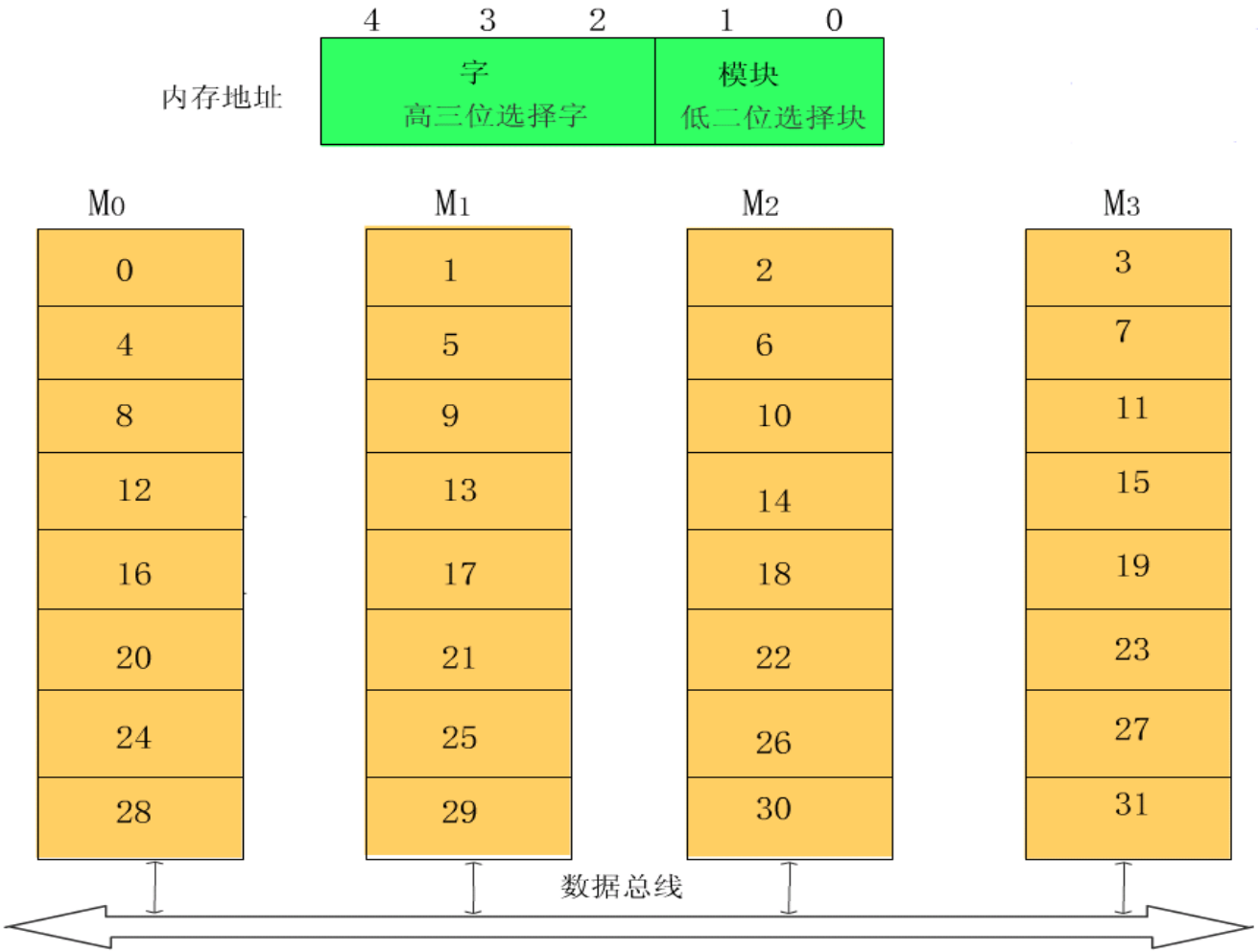
顺序方式

各模块一个接一个串行工作。



交叉方式

连续地址分布在相邻的不同模块内，同一个模块内的地址都是不连续的。对连续字的成块传送可实现多模块流水式并行存取，大大提高存储器的带宽。



2）.多模块交叉存储器编址方式

如果在M个模块上交叉编址($M=2^k$），则称为模M交叉编址。

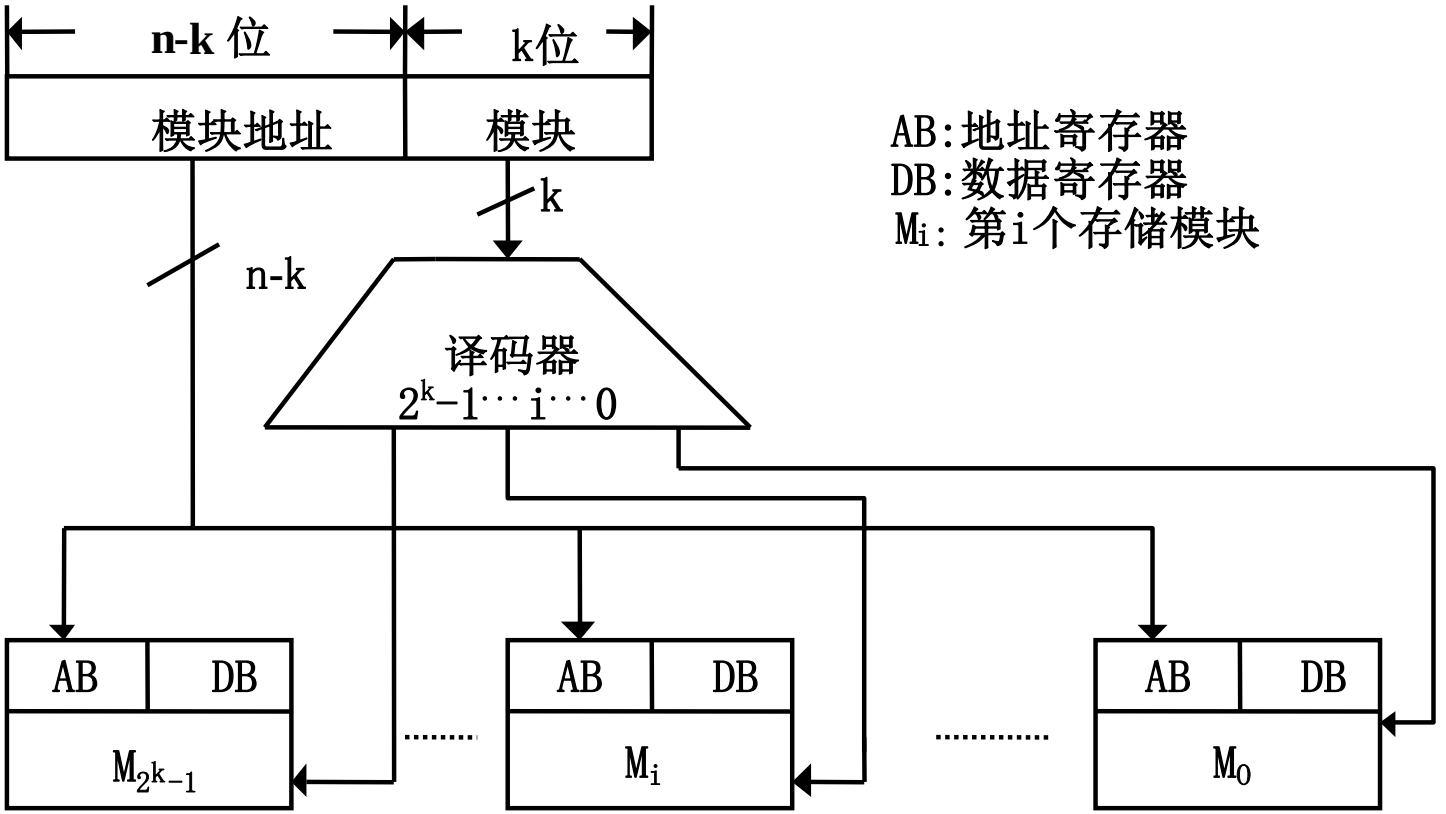
设存储器包括M个模块，每个模块的容量为L ，各存储模块进行低位交叉编址，连续的地址分布在相邻的模块中。第i个模块M_i的地址编号应按下式给出：

$$M \cdot j + i$$

其中， $j=0,1,2,\ldots,L-1$

$i=0,1,2,\ldots,M-1$

一般模块数M取2的k次幂，高档微机M值可取2或4，大型计算机M取16至32。


n-k 位
k位
模块地址
模块
k
n-k
译码器
$2^k-1\cdots i\cdots 0$
AB:地址寄存器
DB:数据寄存器
M_i: 第i个存储模块
AB
DB
M_{2^k-1}
AB
DB
M_i
AB
DB
M_0

多体交叉编址方式

模四交叉各模块的编址序列

模体	地址编址序列	对应二进制地址最低二位
M_0	0, 4, 8, 12, . . . 4·j+0, . . .	0 0
M_1	1, 5, 9, 13, . . . 4·j+1, . . .	0 1
M_2	2, 6, 10, 14, . . . 4·j+2, . . .	1 0
M_3	3, 7, 11, 15, . . . 4·j+3, . . .	1 1

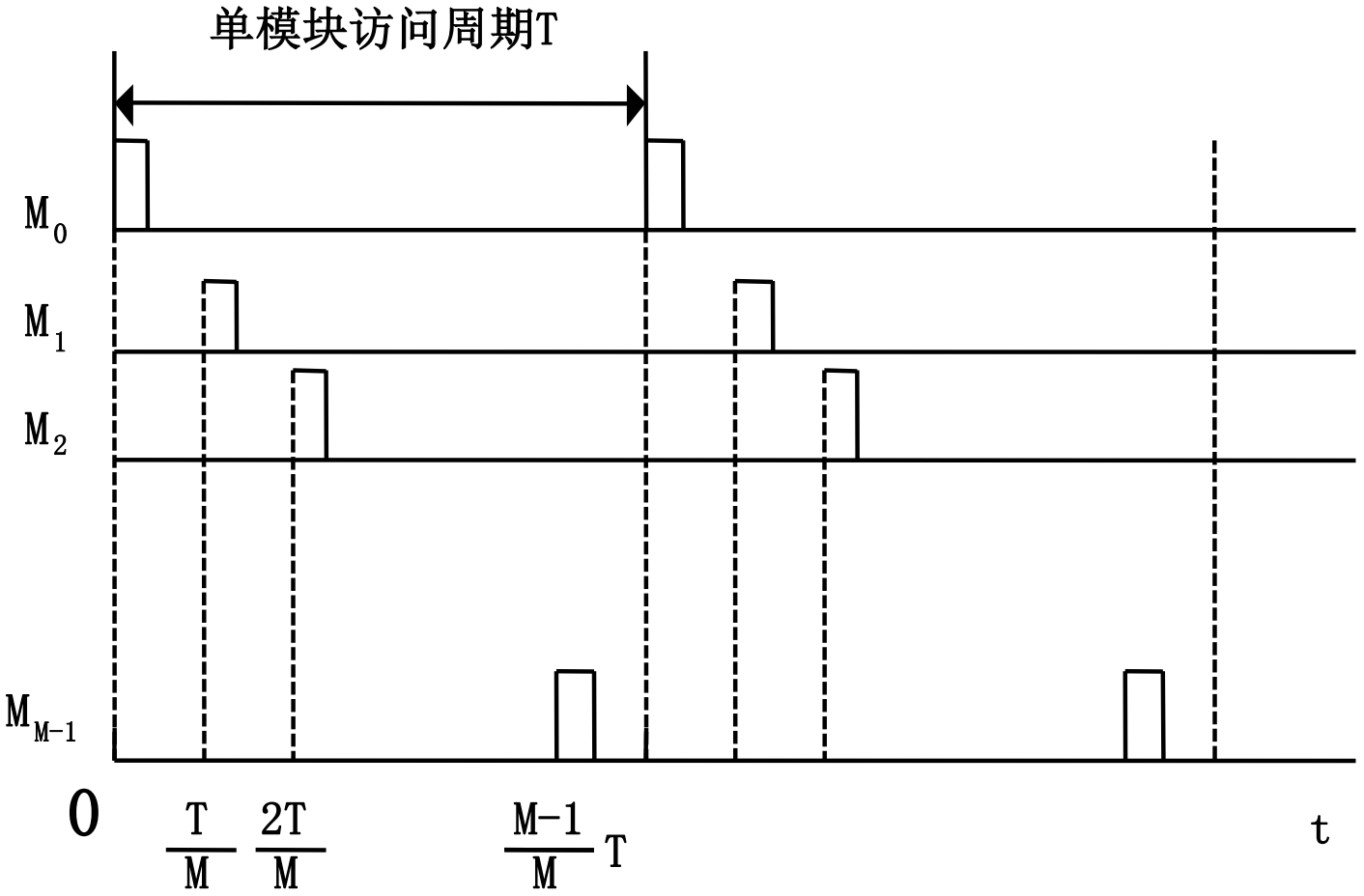
3）.多模块交叉存储器存取控制方式

多模块交叉存储器可以有两种不同的方式进行访问：

(1) 一种是所有模块同时启动一次存储周期，相对各自的数据寄存器并行地读出或写入信息；称为“同时访问”，同时访问要增加数据总线宽度。

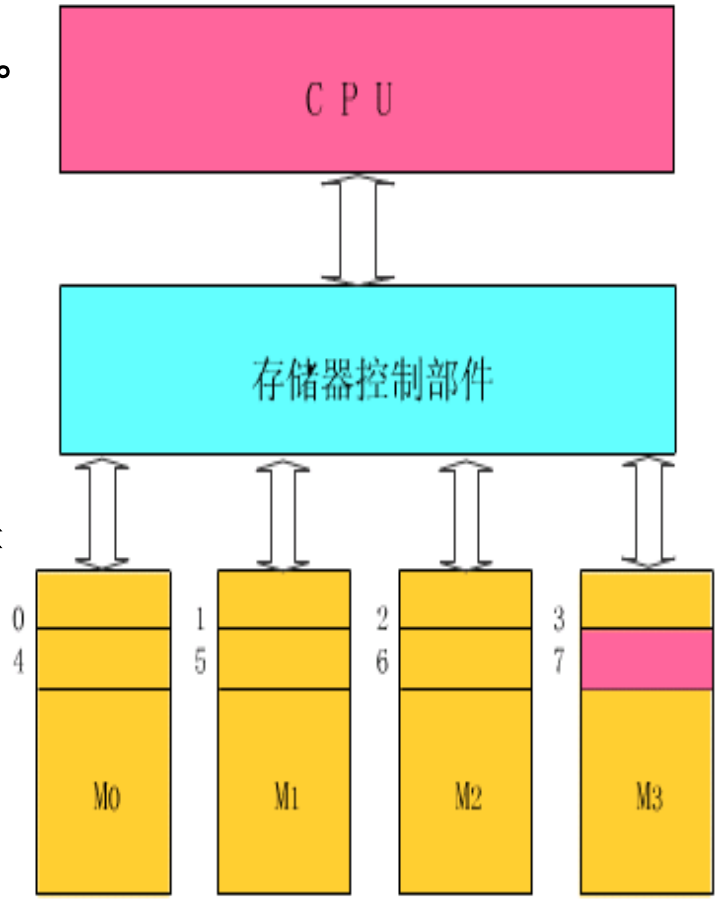
(2) 另一种是M个模块按一定的顺序轮流启动各自的访问周期，启动两个相邻模块的最小时间间隔等于单模块访问周期的1/M。称为“交叉访问”。

交叉访问的存储器工作时间图



4）.多模块交叉存储器的基本结构

- 每个模块各自以等同的方式与CPU传送信息。
- CPU同时访问四个模块，由存储器控制部件控制它们分时使用数据总线进行信息传递。
- 对每一个模块来说，从CPU给出访存命令直到读出信息仍然使用了一个存取周期时间；
- 对CPU来说，它可以在 一个存取周期中连续访问4个模块；
- 各模块的读写过程重叠进行，所以这是一种并行存储器结构。



相联存储器

1. 相联存储器的基本原理

相联存储器不是按地址访问的存储器，而是按内容寻址的存储器。如下表：

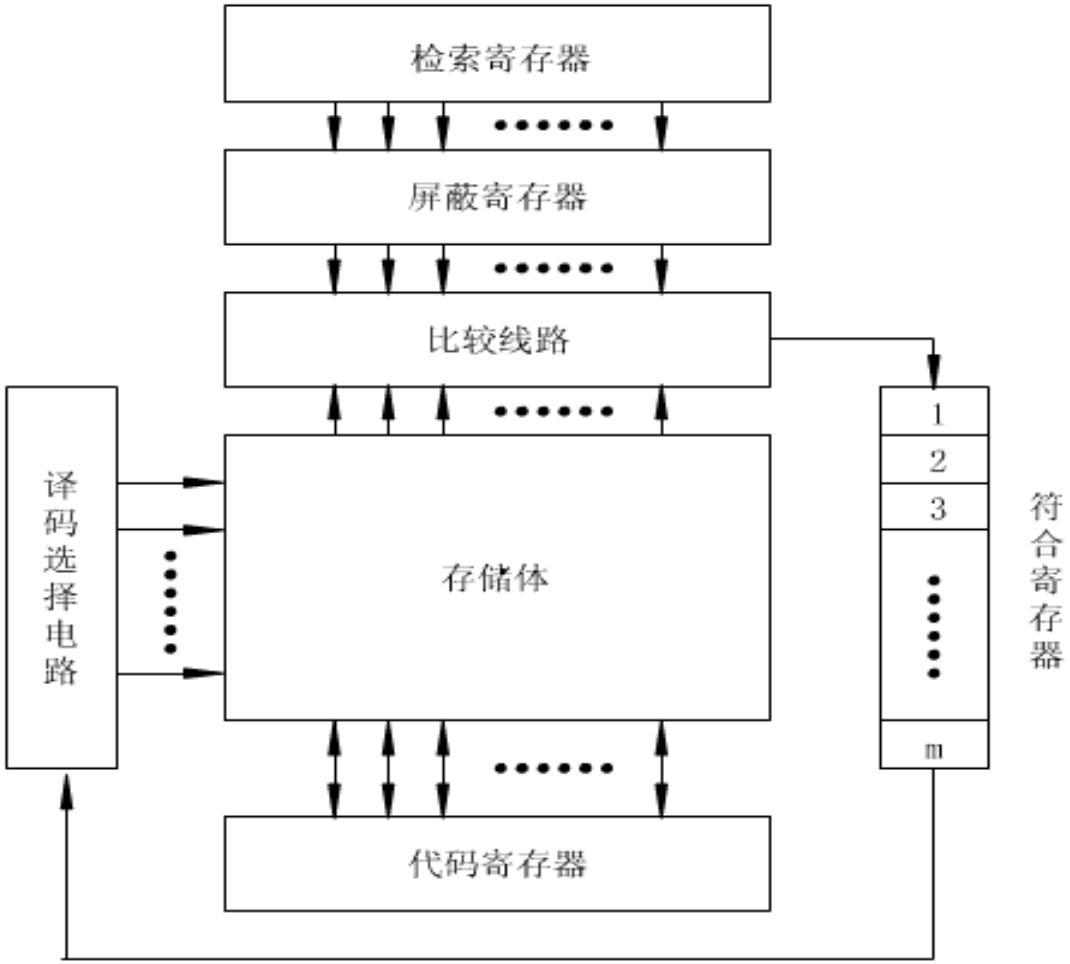
物理地址	职工号	姓名	出生年月	工资数
n	800	张明	1940.2	2000
n+1	540	王芳	1960.1	1200
n+2	920	李平	1943.3	1500
n+3	750	赵洪	1945.2	1400
n+4	610	周进	1965.9	1000

- 相联存储器是指其中任一存储项内容作为地址来存取的存储器。
- 选用来寻址存储器的子段叫做关键字，简称“键”。
- 这样，存放在相联存储器中的项可以看成具有下列格式：

  KEY，DATA

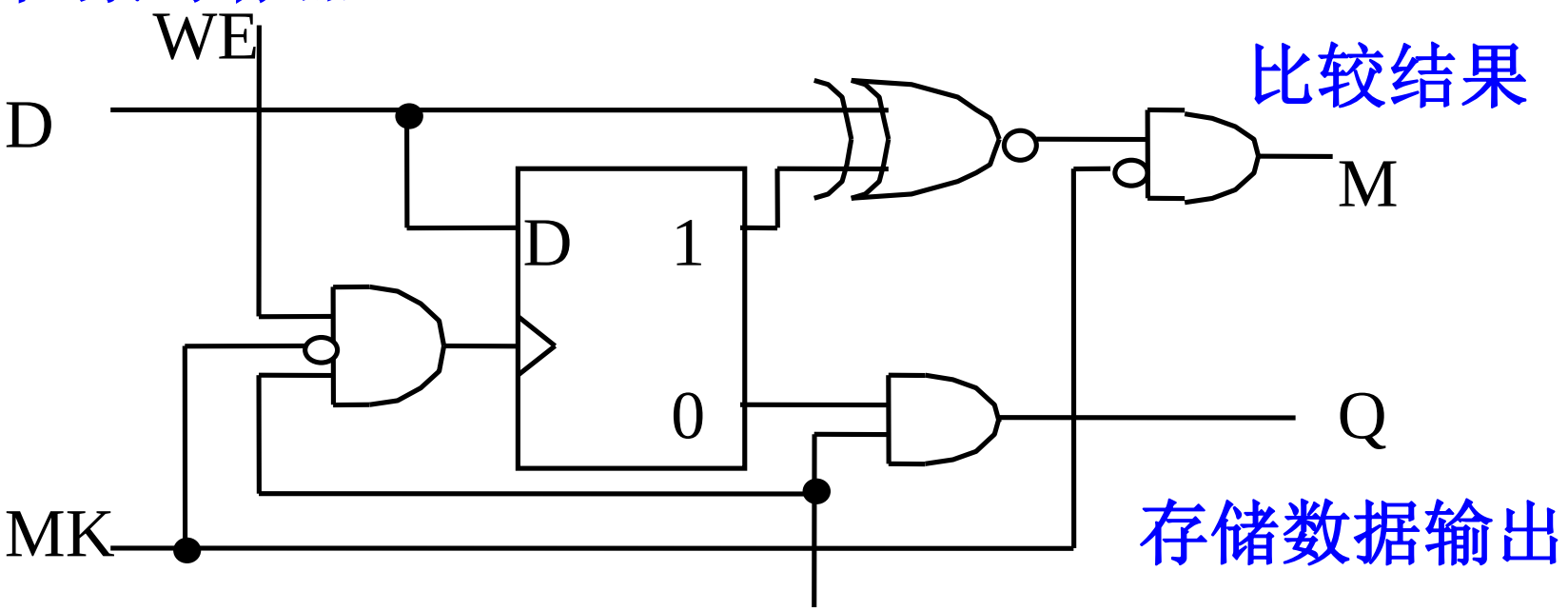
  其中KEY是地址，DATA是被读写信息。
- 相联存储器的基本原理是把存储单元所存内容的某一部分作为检索项(即关键字项)，去检索该存储器，并将存储器中与该检索项符合的存储单元内容进行读出或写入。

2. 相联存储器的组成

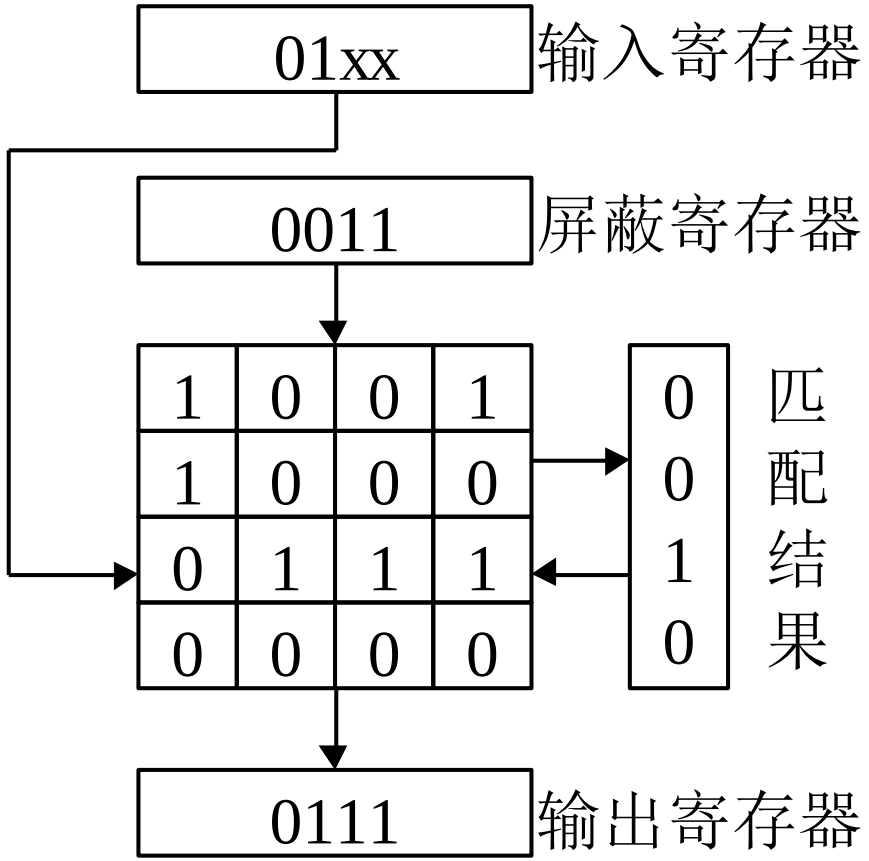


相联存储器——存储Key的单元结构

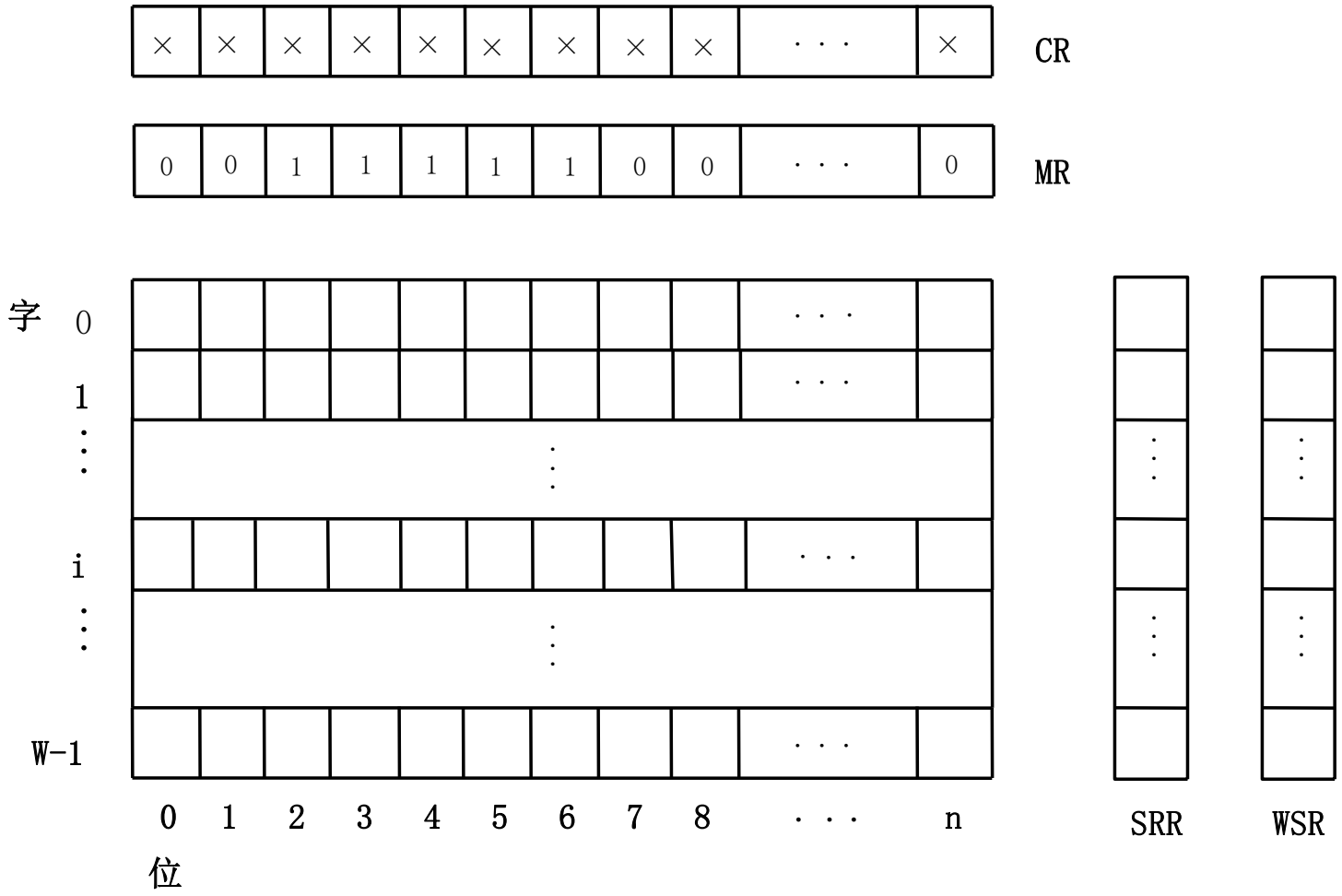
来自检索寄存器



相联存储器——访问实例


01xx
输入寄存器
0011
屏蔽寄存器
1 0 0 1
1 0 0 0
0 1 1 1
0 0 0 0
0
0
1
0
匹配结果
0111
输出寄存器

3. 相联存储器举例



- 设存储器有W个字，字长n位。
- CR位比较寄存器，字长也为n位，存放要比较的数（或要检索的内容）。
- MR为屏蔽寄存器，与CR配合适用，字长也为n位。当按比较数的部分内容进行检索时，相应地把MR中要比较的位设置成“1”，不要比较的位设置成“0”。图中表示需要按2～6位的内容

  进行比较，所以 MR的2—6位为“1”，其余各位均置“0”。置成

  “1”的字段称为关键字段。
- SRR为查找结果寄存器，字长为W位，假如比较结果第i个字满足要求，则SRR中的第i位为“1”，其余各位均为“0”，若同 时有n个字满足要求，则相应地就有n位为“1”。

- 有的相联存储器还设置有字选择寄存器WSR，用来确定哪些字参与检索，若字选择寄存器某位为“1”，则表示其对应的存储字参与检索；若某位为“0”，则表示其对应的存储字不参与检索。下面举例说明之。

假如某高校学生入学考试总成绩已存入相联存储器，如图所示。

今要求列出“总分”在560分和600分范围内的考生名单。

可以用二次查找完成：

第一次找出“总分”大于559分的考生名单；

第二次从名单中再找出总分小于601分的考生；

因此分别将559分和601分作为关键字段内容置于比较寄存器中。

0	0	0	0	0	559

第一次查找寄存器内容

0...0	0...0	0...0	0...0	0...0	11...1

屏蔽寄存器内容

准考证号	姓名	性别	年龄	志愿	总分	SRR 第二次查找结果	WSR 第一次查找结果
1	赵××	男	17	××系	582	1	1
2	钱××	男	18	××系	611	0	1
3	孙××	女	18	××系	584	1	1
4	李××	男	18	××系	530	0	0
5	周××	女	17	××系	604	0	1
6	吴××	女	18	××系	580	1	1
7	陈××	男	18	××系	572	1	1
8	王××	男	18	××系	578	1	1
⋮							
N	丁××	女	19	××系	520	0	0

- 为了进行检索，还要求相联存储器能进行各种比较操作（相等、不等、小于、大于、求最大值和最小值等）。
- 比较操作是并行进行的，即CR中的关键字段与存储器的所有W个字的相应字段同时进行比较。这由相联存储器的具体电路实现，极大地提高了处理速度。

在计算机系统中，相联存储器主要用于虚拟存储器中存放分段表、页表和快表；在高速缓冲存储器中，相联存储器作为存放cache的行地址之用。这是因为，在这两种应用中，都需要快速查找。

校验码

1. 为什么设置校验码

元件故障、噪声干扰等各种因素常常导致计算机在处理信息过程中出现错误。为了防止错误，可将信号采用专门的逻辑线路进行编码以检测错误，甚至校正错误。

通常的方法是：在每个字上添加一些校验位，用来确定字中出现错误的位置。

常用方法：

奇偶校验码；

海明校验与纠错码；

循环冗余校验码。

数据校验码原理

1、码字：由若干位代码组成，满足某种编码规律的一个代码字。

例：编码规则“代码中1的个数为奇数”则

“01001001”合法　“11001001”不合法

2、码距：码距指任何一种编码的任两组二进制代码中，其对应位置的代码最少有几个二进制位不相同。

例：若用4位二进制数表示16种状态，16种状态都用，则码距L=1。若用4位二进制数表示8种状态，而把另外8种状态作为非法编码，此时的码距L=2。

3、最小码距：指一种编码的任意两个码字中间，对应位置代码变化的最少个数。8421BCD码0111→1001 L=3 而 0100→0101 L=1

4、数据校验的实现原理：数据校验码是在合法的数据编码之间，加进一些不允许出现的(非法的)编码，使合法的数据编码出现错误时成为非法编码。这样就可以通过检测编码的合法性达到发现错误的目的。

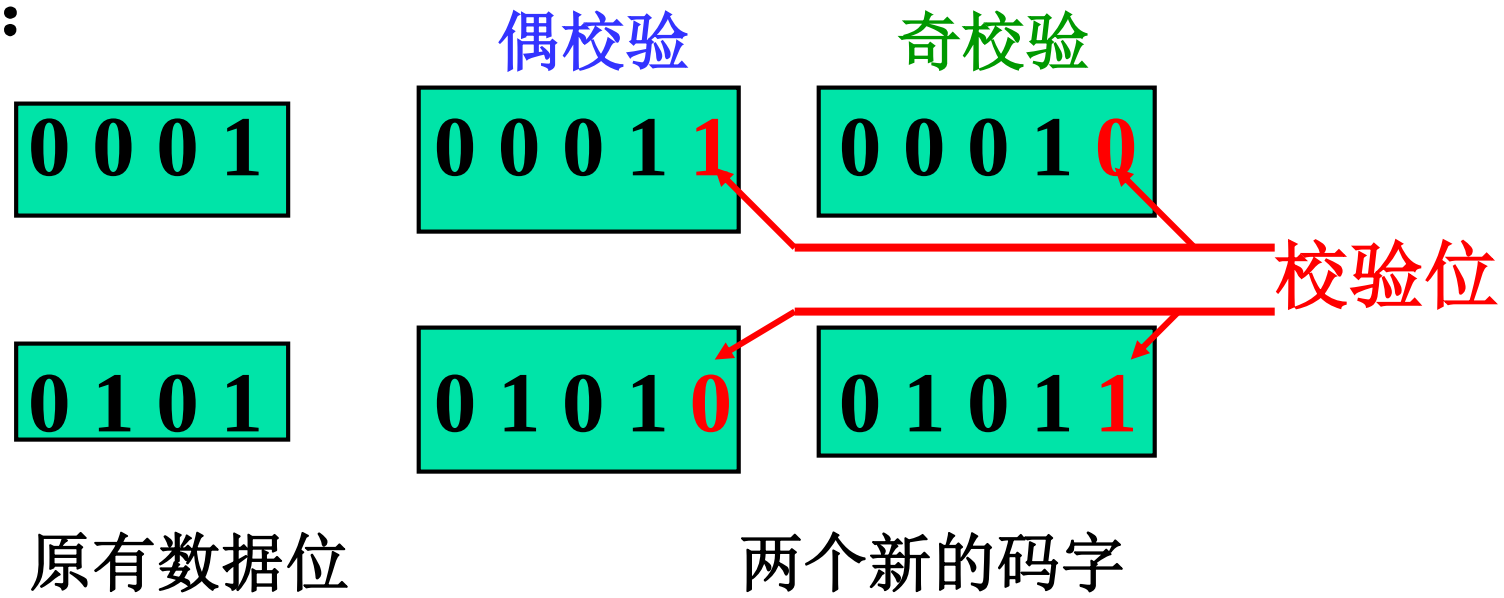
2. 奇偶校验

原理： 在 k 位数据码之外增加 1 位校验位，

使 k+1 位码字中取值为 1 的位数保持为

偶数（偶校验）或 奇数（奇校验）

例如：



定义：

设$x=(x_0\ x_1 \ldots x_{n-1})$是一个n位字，则奇校验位 C定义为

$$C = \overline{x_0 \oplus x_1 \oplus \ldots \oplus x_{n-1}}$$

式中⊕代表按位加，只有当x中包含有奇数个1时，$C=0$。

同理，偶校验位 C定义为

$$C = x_0 \oplus x_1 \oplus \ldots \oplus x_{n-1}$$

即x中包含偶数个1时，才使$C=0$。

例 已知下表中左面一栏有5个字节的数据。请分别用奇校验和偶校验进行编码。

数　据	偶校验编码 C	奇校验编码 C
1 0 1 0 1 0 1 0	1 0 1 0 1 0 1 0 0	1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 0	0 1 0 1 0 1 0 0 1	0 1 0 1 0 1 0 0 0
0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 1
0 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1 0
1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 0	1 1 1 1 1 1 1 1 1

校验方法：(以偶校验为例)

发送：$x_0\,x_1 \ldots x_{n-1}C$　（算出C加到需发送字的后面）

接收：$x_0'\,x_1' \ldots x_{n-1}'\,C'$

计算：$F = x'_0 \oplus x'_1 \oplus \ldots \oplus x'_{n-1} \oplus C'$

结果：若$F=1$，意味着收到的信息有错；

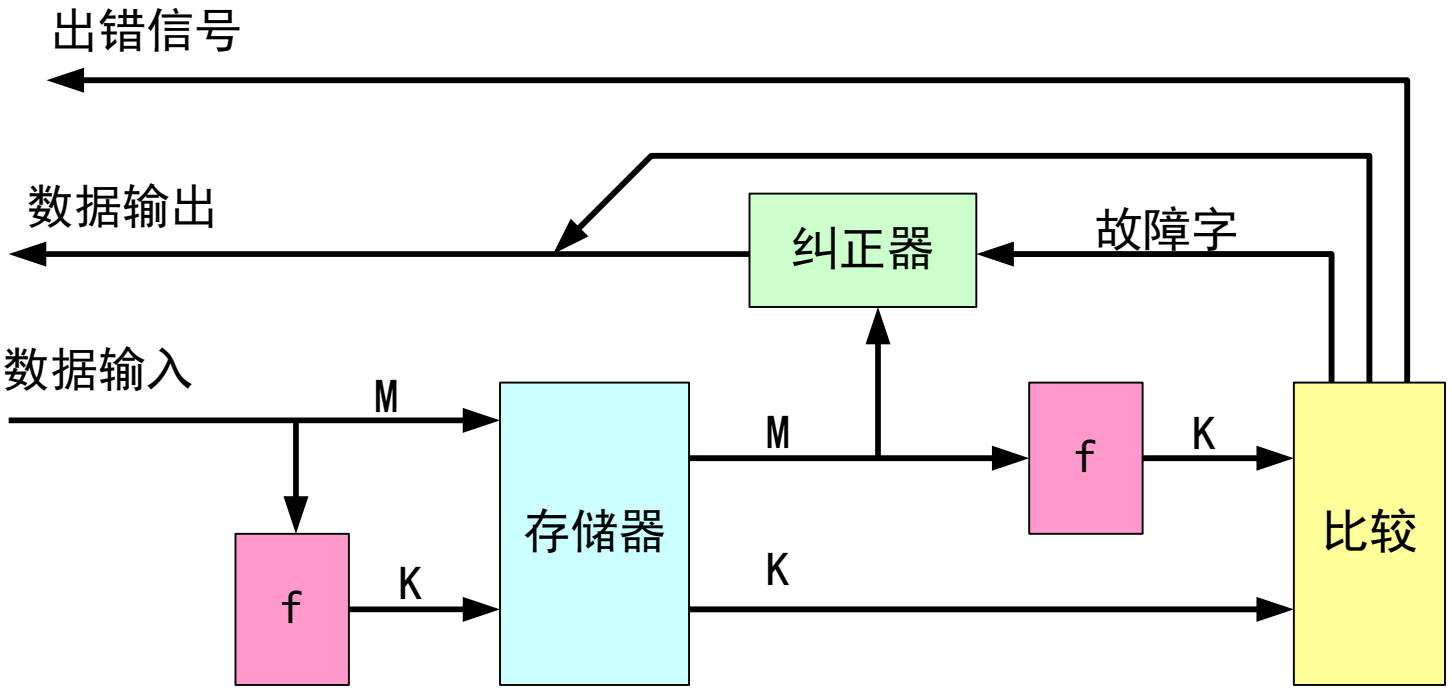
若$F=0$，表明x字传送正确。

特点：

奇偶校验可提供单（奇数）个错误检测，
但无法检测多（偶数）个错误，
更无法识别错误信息的位置及纠正错误。

奇偶校验码常用于存储器读写检查，或ASCII字符传送过程中的检查。

纠错码功能



从M位数据中产生一组新的K位校验码与取出的纠错码功能校验位码作比较：

1、无错误

2、检测到差错，并可以纠正。

3、检测到差错，但无法纠正。

海明校验码

1. 原理

海明校验码的实现原理是：在数据位中加入几个校验位，将数据代码的码距均匀地拉大，并把数据的每个二进制位分配在几个奇偶校验组中。当某一位出错后，就会引起有关的几个校验位的值发生变化，这不但可以发现错误，还能指出是哪一位出错，为进一步自动纠错提供了依据。

2. 编码规则

若海明码的最高位号为**m**，最低位号为**1**，即$H_m H_{m-1} \cdots H_2 H_1$，则海明码的编码规则是：

（**1**）校验位与数据位之和为**m**，每个校验位P_i在海明码中被分在位号2^{i-1}的位置上，其余各位为数据位，并按从低向高逐位依次排列的关系分配各数据位。

（**2**）海明码的每一位位码H_i（包括数据位和校验位）由多个校验位校验，其关系是被校验的每一位位号要等于校验它的各校验位的位号之和。

3. 增添校验位

假设欲检测的有效信息为n位，需增加的校验位为k位，则校验码的长度为n+k位。校验位的状态组合，应当具有指出n+k位中任一位有错或无错的能力，即需要区别出n+k+1种状态。应满足以下关系式：

$$2^k \geq n+k+1$$

这个关系式称为海明不等式，若信息位长度n确定后，由此可得到校验位k的最短长度。

确定校验位后，就可以与信息位组成海明校验位。假设数据位是7位二进制编码，据上所述，校验位的位数k为4,故海明码的总位数为11。它们的排列关系可表示为：

海明码位号：$H_{11}\ H_{10}\ H_9\ H_8\ H_7\ H_6\ H_5\ H_4\ H_3\ H_2\ H_1$

海明码：$D_7\ \ D_6\ \ D_5\ P_4\ D_4\ D_3\ D_2\ P_3\ D_1\ P_2\ P_1$

可知：每个校验位由其本身校验；

每个数据位由若干校验位校验。

4. 校验位校验任务的分配

根据海明码的编码规则，每一位海明码都有多个校验位校验，且被校验的每一位的位号等于参与校验它的几个校验位的位号之和。

占据各权位上的校验位按权组成的**8421**码，正好等于海明码的位号，即海明码的位号H_i正好等于要校验它的校验位所占权位权值之和。

例如：$H_{11}=P_4\times2^3+P_2\times2^2+P_1\times2^1$

这说明了H_{11}位将由 P_4、P_2、P_1进行校验。

校验位P_1可以校验：H_1、H_3、H_5、H_7、H_9、H_{11}、H_{13}、H_{15}

校验位P_2可以校验：H_2、H_3、H_6、H_7、H_{10}、H_{11}、H_{14}、H_{15}

校验位P_3可以校验：H_4、H_5、H_6、H_7、H_{12}、H_{13}、H_{14}、H_{15}

校验位P_4可以校验：H_8、H_9、H_{10}、H_{11}、H_{12}、H_{13}、H_{14}、H_{15}

根据校验时偶校验，可以写出相应的校验方程。

例：设有一个7位信息码位0110001，求它的海明码。

解： 此例中，信息位n=7，根据海明不等式，可求得校验位最短长度k=4。

其海明码先表示如下：

海明码位号：$H_{11}\ H_{10}\ H_9\ H_8\ H_7\ H_6\ H_5\ H_4\ H_3\ H_2\ H_1$

海明码：$0\quad 1\quad 1\quad P_4\quad 0\quad 0\quad 0\quad P_3\quad 1\quad P_2\quad P_1$

按偶校验写出校验方程为：

$H_1 \oplus H_3 \oplus H_5 \oplus H_7 \oplus H_9 \oplus H_{11} = 0$　　（$P_1 = H_1$）

$H_2 \oplus H_3 \oplus H_6 \oplus H_7 \oplus H_{10} \oplus H_{11} = 0$　　（$P_2 = H_2$）

$H_4 \oplus H_5 \oplus H_6 \oplus H_7 = 0$　　（$P_3 = H_4$）

$H_8 \oplus H_9 \oplus H_{10} \oplus H_{11} = 0$　　（$P_4 = H_8$）

由此可得：$P_1=0$、$P_2=0$、$P_3=0$、$P_4=0$，所以0110001的海明码为01100000100。

5. 检错与纠错

方法：将错了的码字重新代入校验方程校验一次即可。假设上面例子中的海明码**01100000100**传送后，若H_6位发生了错误，变成了**01100100100**，这时把它们代入上面的偶校验校验方程，如下：

$$H_1 \oplus H_3 \oplus H_5 \oplus H_7 \oplus H_9 \oplus H_{11} = 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 \oplus 0 = 0 = E_1$$

$$H_2 \oplus H_3 \oplus H_6 \oplus H_7 \oplus H_{10} \oplus H_{11} = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 \oplus 0 = 1 = E_2$$

$$H_4 \oplus H_5 \oplus H_6 \oplus H_7 = 0 \oplus 0 \oplus 1 \oplus 0 = 1 = E_3$$

$$H_8 \oplus H_9 \oplus H_{10} \oplus H_{11} = 0 \oplus 1 \oplus 1 \oplus 0 = 0 = E_4$$

可以把$E_4E_3E_2E_1 = 0110$看成一个“指误字”，因为其二进制码为**0110**，说明H_6出了错，是H_6错成了**1**，所以要纠错，纠错时将H_6位取反值，即让它恢复到正确值**0**。这样纠错后即可得到正确的海明码**01100000100**。

循环冗余校验码

1．CRC的编码方法

任何一个二进制序列中的各位看成一个多项式的系数

如：1101→$1\times X^3+1\times X^2+0\times X^1+1\times X^0$

设：n是有效数据信息位位数，r是校验位位数。

总长k=n+r位，称（k，n）码。

设待编码的有效信息以多项式M(x)表示，将M(x)左移r位得到多项式$M(x)*X^r$，使低r位二进制位全为零，以便与r位校验位拼接。使用多项式$M(x)*X^r$除以生成多项式G(x)，求得的余数即为校验位。为了得到r位余数(校验位)，G(X)必须是r+1位的。

G(X)最高项的指数决定了r的位数

假设M(x)*X^r除以生成多项式G(x)，求得的余数用表达式R(x)表示，商的表达式用Q(x)表示，它们之间的关系如下：

$$\frac{M(x)*X^r}{G(X)} = Q(X) + \frac{R(X)}{G(X)}$$

这时将余数R(X)与M(x)*X^r相加，就得到n+r位CRC编码：

$$M(x)*X^r+R(x) = Q(x)*G(x) + R(x) + R(x)$$

因为“两个相同数据的模2和为零”，即R(x) + R(x) = 0，所以

$$M(x)*X^r+R(x) = Q(x)*G(x)$$

可以看出，所求得的CRC码是一个可被G(X)表示的数码除尽的数码。

例 设四位有效信息位是1100，选用生成多项式 G(X)=1011，试求有效信息位1100的CRC编码。

解：

(1)将有效信息位1100表示为多项式M(x)

$M(X) = X^3 + X^2 = 1100$

(2)M(X)左移r=3位，得$M(x)*X^3$

$M(x)*X^3 = X^6 + X^5 = 1100000$

(3)用r+1位的生成多项式 G(X)，对$M(x)*X^r$作“模2除”

1100000/1011 = 1110 + 010/1011

(4)$M(x)*X^3$ 与r位余数R(X) 作“模2加”，即可求得它的CRC编码

$M(x)*X^3 + R(X) = 1100000 + 010 = 1100010$ (模2加)

因为k=7、n=4，所以编好的CRC码又称为(7，4)码。

2．模2运算：不考虑借位和进位

（1）模2加减：可用异或门实现，即：

0+0=0；0+1=1；1+0=1；1+1=0；

0−0=0；0−1=1；1−0=1；1−1=0；

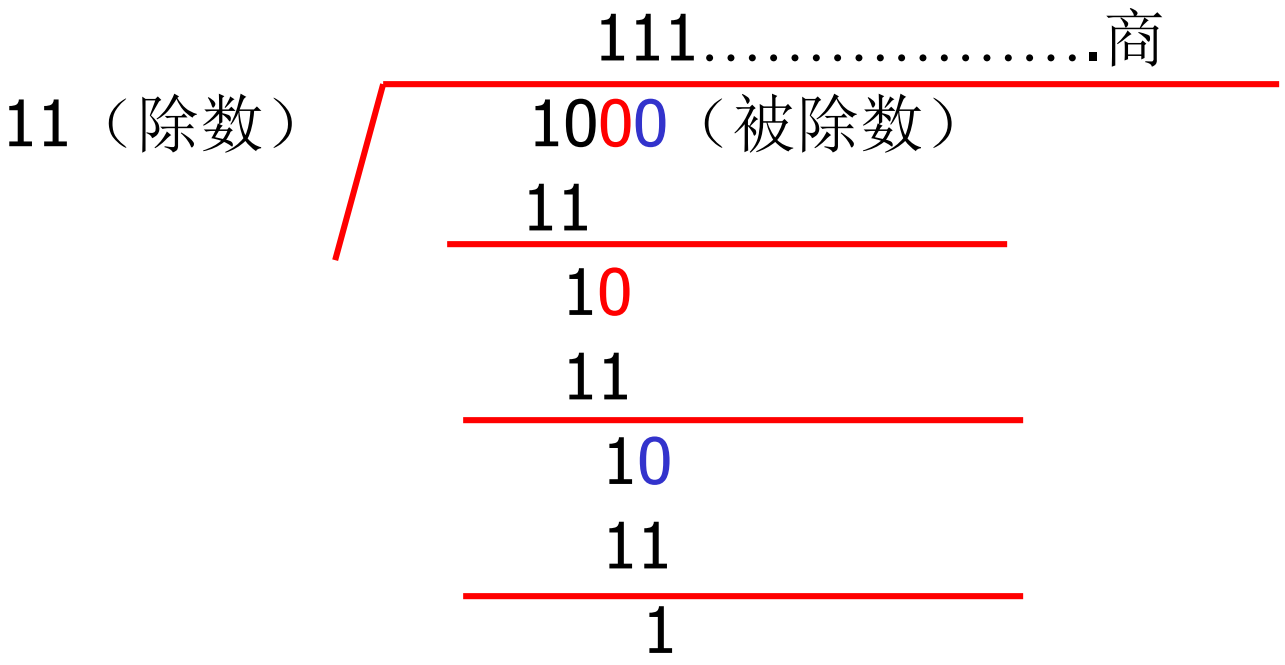
（2）模2乘法：用模2加求部分积之和

例如：

$$
\begin{array}{rr}
 & 1011 \\
\times & 11 \\
\hline
 & 1011 \\
+ & 1011 \\
\hline
 & 11101
\end{array}
$$

（3） 模2除法：按模2减求部分余数，每上一位商，部分余数要减少一位，上商规则是：只要余数最高位为1，则商1，否则为0。当部分余数的位数小于除数时，该余数为最后余数。

例如：



3．CRC的译码及纠错

CRC码传送到目标部件，用约定的多项式G(x)对收到的CRC码进行“模2除”，若余数为0，则表明该CRC校验码正确；否则表明有错，不同的出错位，其余数是不同的。由余数具体指出是哪一位出了错，然后加以纠正。

不同的出错位，其余数也是不同的。

可以证明：更换不同的有效信息位，余数与出错位的对应关系不会发生变化，只与码制和生成多项式G(X)有关。

4．关于生成多项式

不是任何一个(k+1)位多项式都能作为生成多项式，从检错、纠错的要求来看，生成多项式应满足下列要求：

(1)任何一位发生错误，都应使余数不为零；

(2)不同位发生错误，都应使余数不同；

(3)用余数补零，继续作“模2除”，应使余数循环。

常用的CRC生成多项式：

CRC-12	12位	$x^{12}+x^{11}+x^3+x^2+1$	
CRC-16	16位	$x^{16}+x^{15}+x^2+1$	(IBM)
CRC-16	16位	$x^{16}+x^{12}+x^5+1$	(CCITT)
CRC-32	32位	$x^{32}+x^{26}+x^{23}+x^{16}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$	

5、CRC产生电路

CRC校验码不仅检错率高，而且硬件实现简单，因而到底广泛应用。

小结

- 存储器的基本工作原理
- 存储器的组织方式
- 存储器的分类
- 高性能存储器