第6章 主存储器

主要知识点

- 掌握RAM单元的工作原理
- 掌握ROM单元的工作原理
- 掌握存储器的字、位扩展
- 了解多体交叉存储器的工作原理
- 掌握Cache存储器的工作原理
- 掌握虚拟存储器的工作原理

百年同濟 TONGJI UNIVERSITY

6.1.1存储器分类

能用来作为存储器的器件和介质，除了其基本存储单元有两个稳定的物理状态来存储二进制信息以外，还必须满足一些技术上的要求。另外价格也是一个很重要的因素。

- **主存储器的类型：**

（1）随机存储器(random access memory，RAM)

随机存储器(又称读写存储器)指通过指令可以随机地、个别地对各个存储单元进行访问，一般访问所需时间基本固定，而与存储单元地址无关。

(2) 只读存储器(read only memory,简称ROM)

只读存储器是一种对其内容只能读不能写入的存储器，在制造芯片时预先写入内容。它通常用来存放固定不变的程序、汉字字型库、字符及图形符号等。由于它和读写存储器分享主存储器的同一个地址空间，故仍属于主存储器的一部分。

(3) 可编程序的只读存储器(programmable ROM,简称PROM)

一次性写入的存储器，写入后，只能读出其内容，而不能再进行修改。

（4）可擦除可编程序只读存储器（erasable PROM，简称EPROM）

可用紫外线擦除其内容的PROM，擦除后可再次写入。

（5）可用电擦除的可编程只读存储器（electrically EPROM,简称E^2PROM）

可用电改写其内容的存储器，近年来发展起来的快擦型存储器（flash memory）具有E^2PROM的特点。

6. 1基本概念
6.1.1 存储器分类

上述各种存储器，除了RAM以外，即使停电，仍能保持其内容，称之为“非易失性存储器”，而RAM为“易失性存储器”。

百年同濟 TONGJI UNIVERSITY

6.1.2 存储器主要技术指标

主存储器的主要性能指标为主存容量、存储器存取时间和存储周期时间。

1.寻址空间和地址分配

计算机可寻址的最小信息单位是一个存储字，相邻的存储器地址表示相邻存储字。

CPU地址线的位数决定了主存储器的可直接寻址的最大空间。例如，32位CPU提供32位物理地址线，支持对2^{32}=4G的物理主存空间的访问。每个存储单元都应有一个唯一的地址，存储器的编址有两种编址方式。

（1）字编址

以一个字为单元进行编址，字可以是16位或32位，即每个单元为16位或32位。

这种机器称为“字可寻址”计算机。一个存储字所包括的二进制位数称为字长。

（2）字节编址

以一个字节为单元进行编址，即每个单元为8位二进制位。有些计算机可以按“字节”寻址，因此，这种机器称为“字节可寻址”计算机。

以字或字节为单位来表示主存储器存储单元的总数，就得到了主存储器的容量。

2.存储器的速度

主存储器的另一个重要的性能指标是存储器的速度，一般用存储器存取时间和存储周期来表示。

存储器存取时间(memory access time)----又称存储器访问时间，是指从启动一次存储器操作到完成该操作所经历的时间。

6. 1基本概念

6.1.2 存储器主要技术指标

存储周期(memory cycle time)----指连续启动两次独立的存储器操作(例如连续两次读操作)所需间隔的最小时间。

通常，存储周期略大于存取时间。

在整个计算机系统中，比较关心的是容量和速度两项指标。

容量：一般取决于CPU的寻址能力（地址线的宽度）。

速度：一般取决于所用的器件。

3. 存储器带宽

也称为“数据传输率”，是指单位时间内通过存储器的数据量。

6.1.3主存储器的基本操作

主存储器用来暂时存储CPU正在使用的指令和数据，它和CPU的关系最为密切。

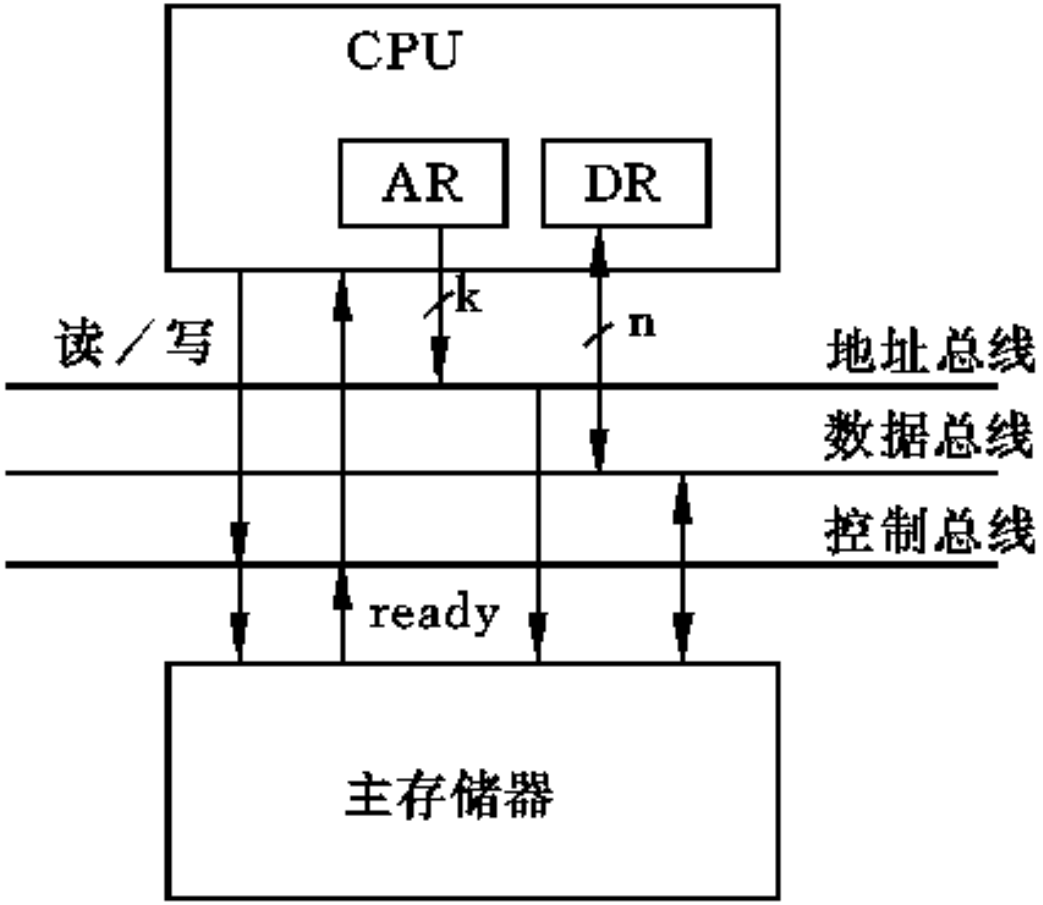
AR 为K位字长，则允许主存包含2^K个可寻址单位 (字节或字)。

DR 为n位字长，CPU和主存之间通过总线进行n位数据传送。

控制总线包括控制数据传送的读(read)、写(write)和表示存储器功能完成的(ready)控制线。

6. 1基本概念

6.1.3主存储器的基本操作



存储器的“读”操作

为了从存储器中取一个信息字，CPU必须指定存储器字地址，并进行“读”操作。

① **CPU需要把信息字的地址送到AR，经地址总线送往主存储器。**

② **同时，CPU通过控制线(read)发一个“读”请求。此后，CPU等待从主存储器发来的回答信号，通知CPU“读”操作完成。**

③ **主存储器通过ready（MFC)线做出回答，若ready信号为“1”，说明存储字的内容已经读出，并放在数据总线上，送入DR。这时，“取”数操作完成。**

6. 1基本概念
6.1.3主存储器的基本操作

- **存储器的“写”操作**

为了“写”一个字到主存。

① **CPU先将信息字在主存中的地址经AR送地址总线，并将信息字送DR。**

② **发出“写”命令。此后，CPU等待写操作完成信号。**

③ **主存储器从数据总线接收到信息字并按地址总线指定的地址存储，然后经ready控制线发回存储器操作完成信号。这时，“存”数操作完成。**

从以上讨论可见，CPU与主存之间采取异步工作方式，以ready信号表示一次访存操作的结束。

6.2随机存储器RAM

半导体读/写存储器按存储元件在运行中能否长时间保存信息来分：

静态存储器 **利用双稳态触发器来保存信息，只要不断电，信息是不会丢失的；**

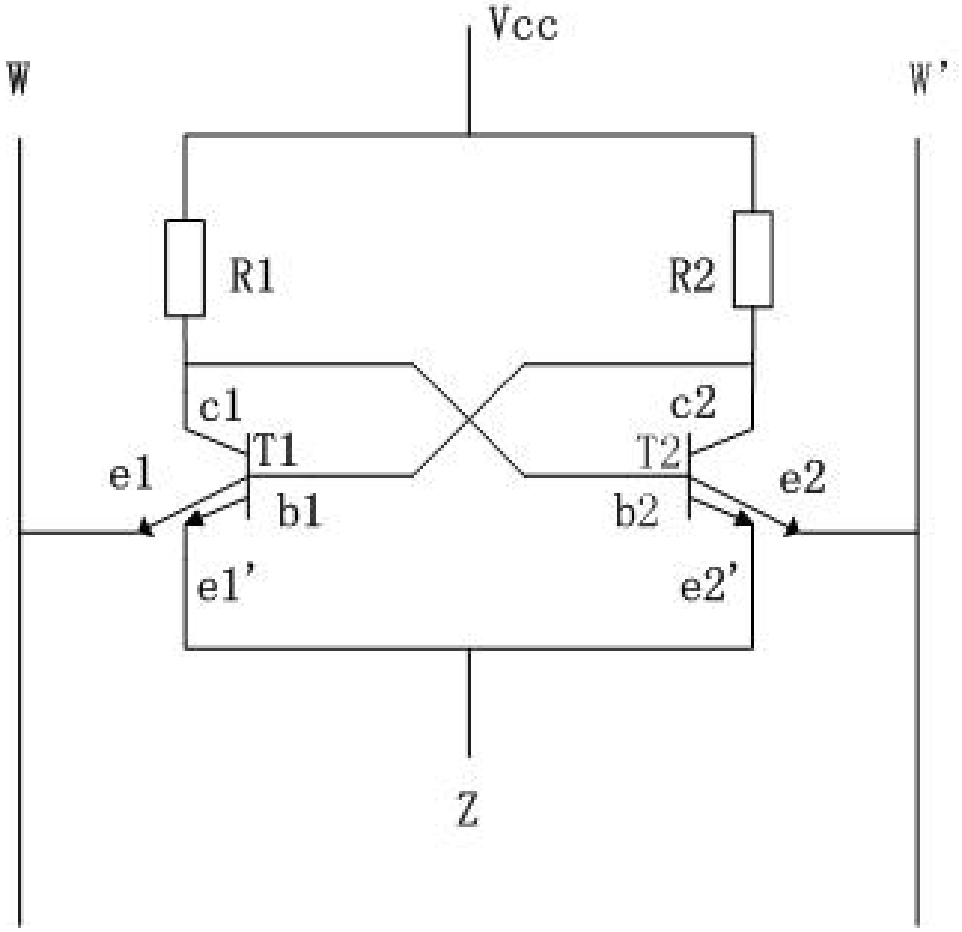
动态存储器 **动态存储器利用MOS电容存储电荷来保存信息，使用时需不断给电容充电才能使信息保持。**

静态存储器的集成度低，功耗较大，主要用于Cache；

动态存储器的集成度高，功耗小，主要用于主存储器。

6.2.1双极型随机存储器

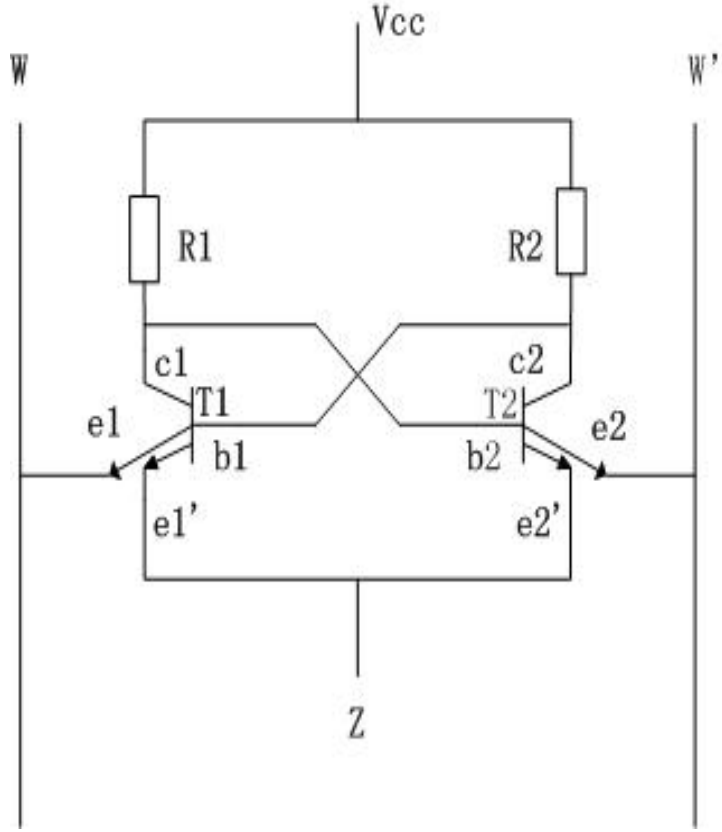
1.基本存储单元



6.2 随机存储器RAM

6.2.1双极型随机存储器

- **T1和T2构成双稳态电路**
- **R1和R2为负载电阻**
- **W为读写线**
- **W’为读写参考线**
- **Z为选通线**



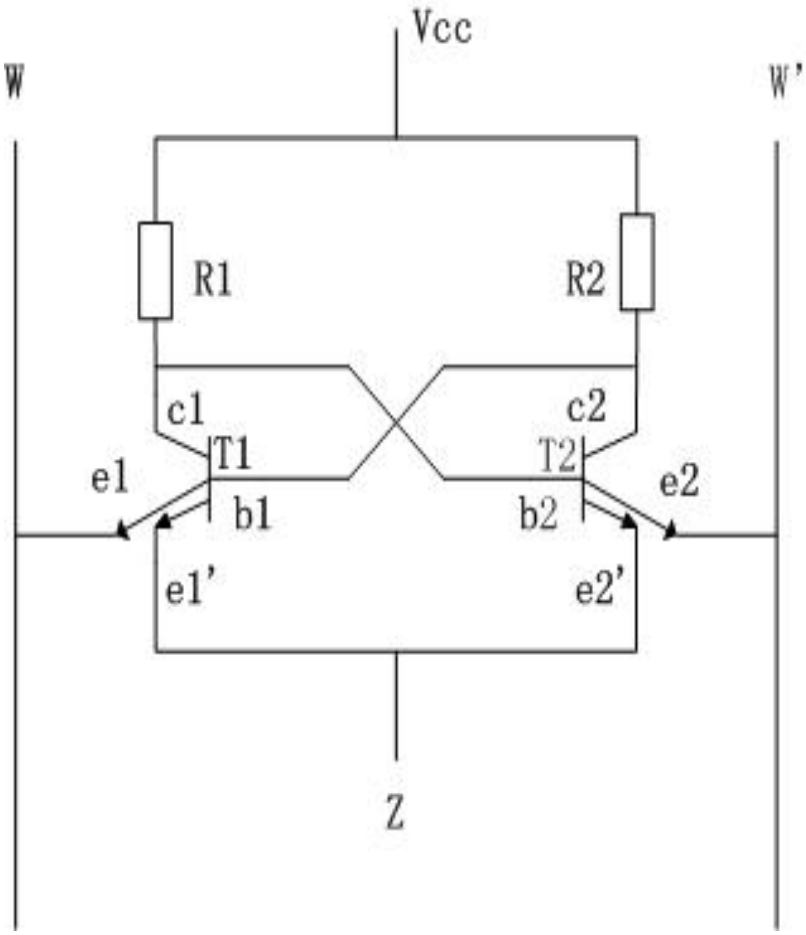
假设： T1导通，T2截止为“1”态

T1截止，T2导通为“0”态

2.保持状态

Vz=0.2V,Vw=1.2V

其原来的状态是不会改变的，即：原T1导通，T2截止或T1截止，T2导通的状态是不会改变的。

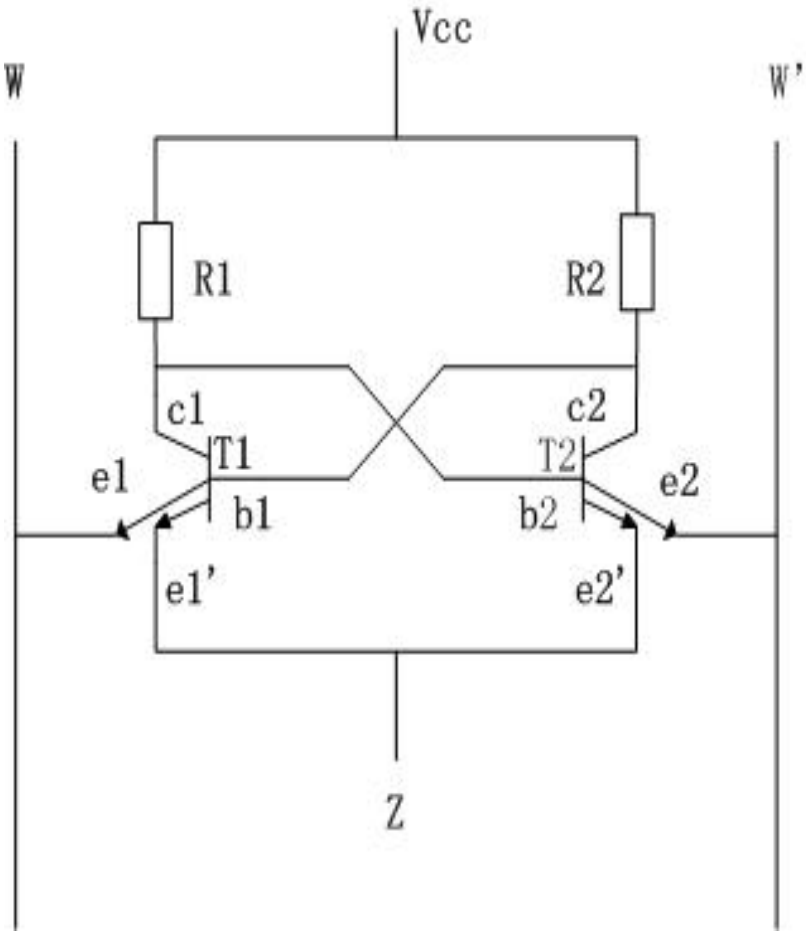


3.读操作状态

Vz=2.5V,Vw=1.2V

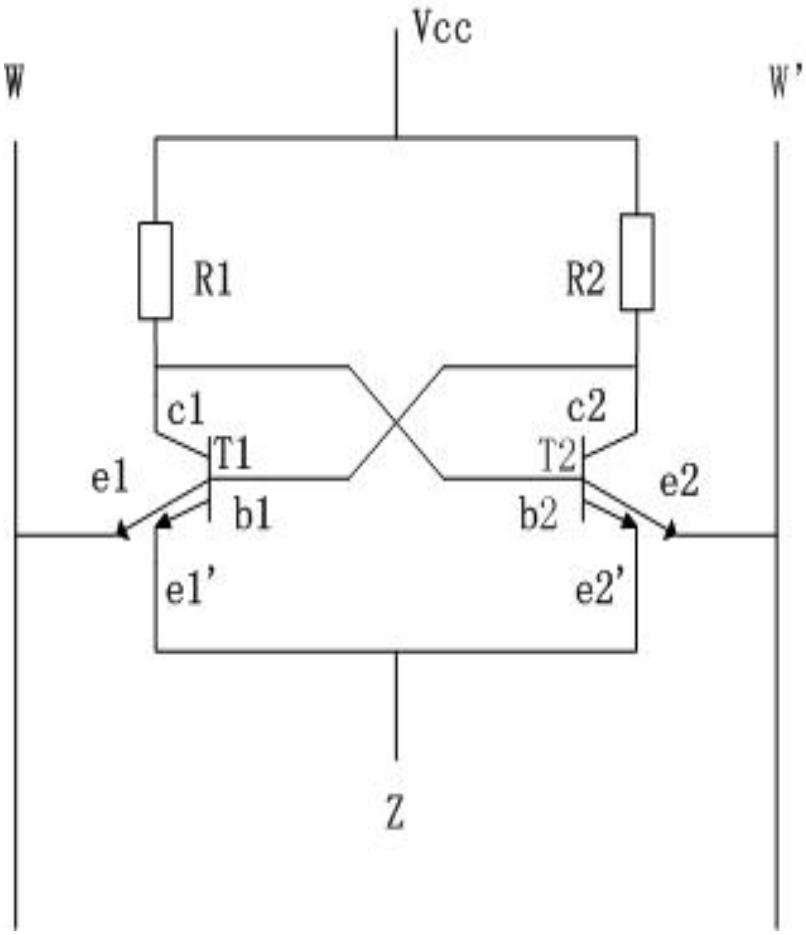
（1）读“1”时

所谓的“1”态就是T1导通，T2截止。当Vz上升到2.5V时，而Vw=1.2V不变，原来流过e1’的电流转向e1流出，在W线上产生一个电流。



(2) 读“0”时

所谓的“0”态就是T1截止，T2导通。当Vz上升到2.5V时，而Vw=1.2V不变，原来流过e2’的电流转向e2流出，在W’线上产生一个电流，而W线上无电流。

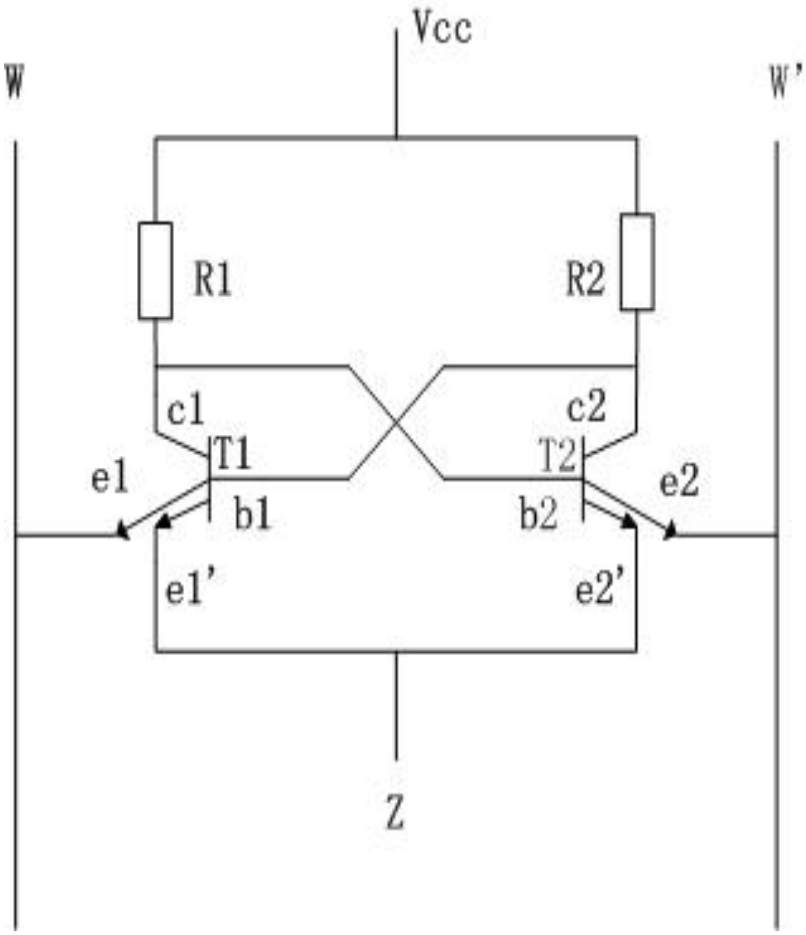


4.写操作状态

Vz=2.5V

(1) 写“1”时

Vw↘0.2V,由于Ve1的电压最低，所以T1导通，而T2截止。写“1”结束。Vw↗1.2V,Vz↘0.2V,进入保持状态。

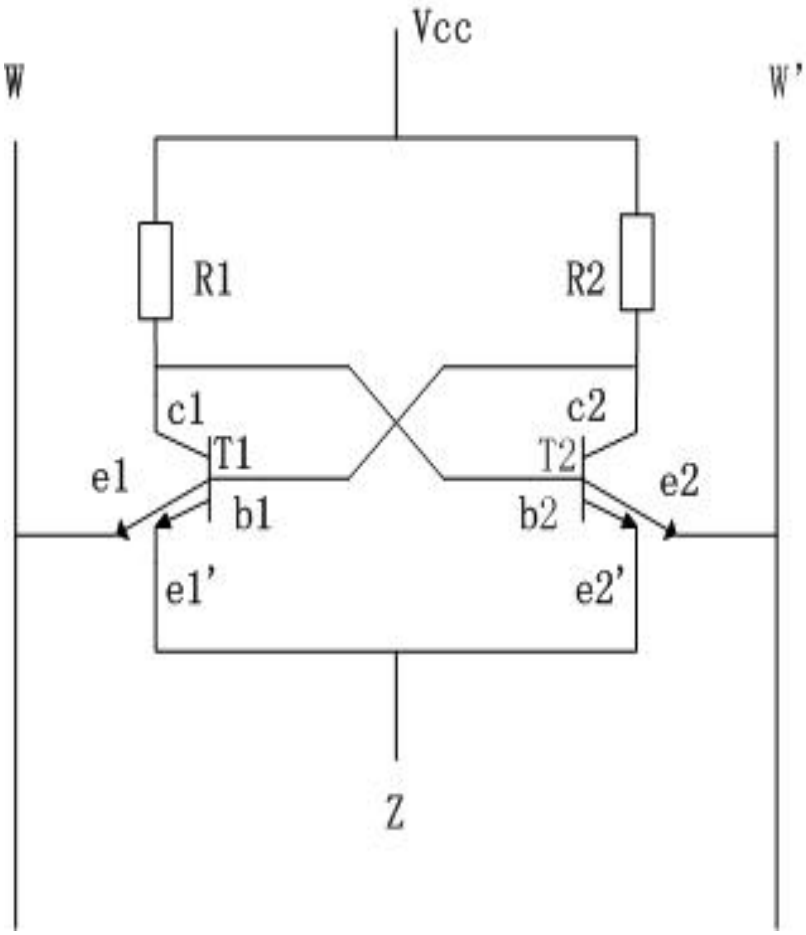


6.2 随机存储器RAM

6.2.1双极型随机存储器

(2) 写“0”时

Vw↗2.5V,由于Ve2的电压最低，所以T2导通，而T1截止。写“0”结束，Vw↘1.2V,Vz↘0.2V，进入保持状态。



6.2 随机存储器RAM

6.2.1双极型随机存储器

- **综合上述：**

①在保持状态下

Vz=0.2, Vw=1.2V

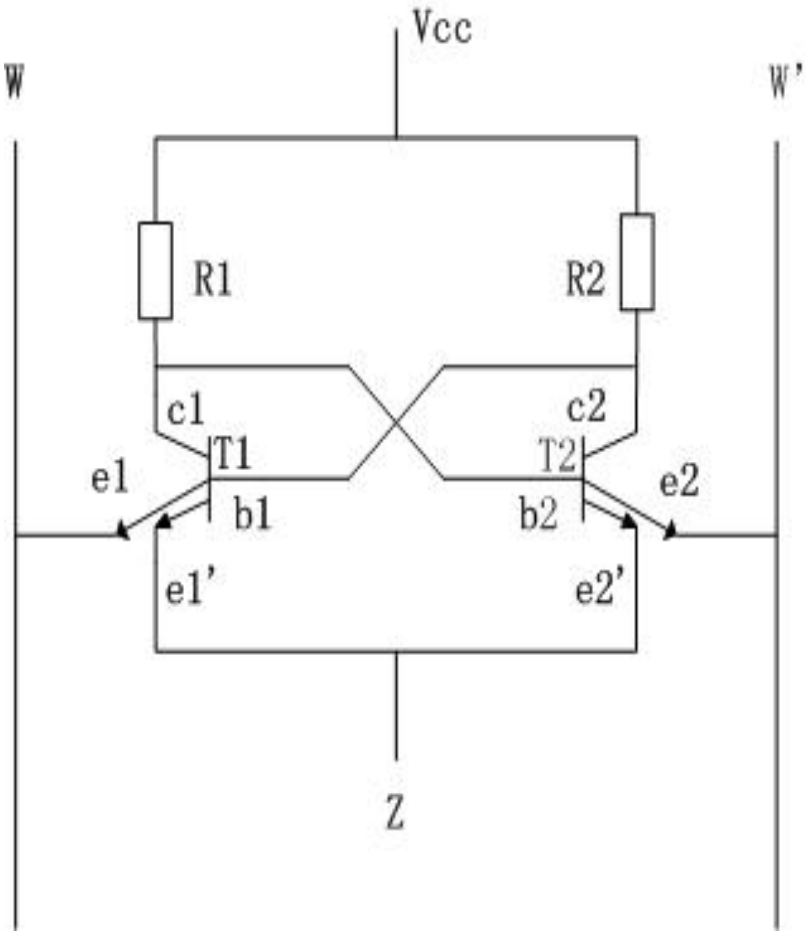
②在读操作状态下

Vz=2.5V, Vw=1.2V

③在写操作状态下

写“1” Vz=2.5V,Vw=0.2V

写“0” Vz=2.5V,Vw=2.5V



百年同济 TONGJI UNIVERSITY

6.2.2 MOS型随机存储器

由于MOS集成电路工艺简单，功耗低，集成度高，价格底，所以应用较为广泛。

MOS型随机存储器分为：

静态存储器SRAM（Static RAM）

动态存储器DRAM（Dynamic RAM）

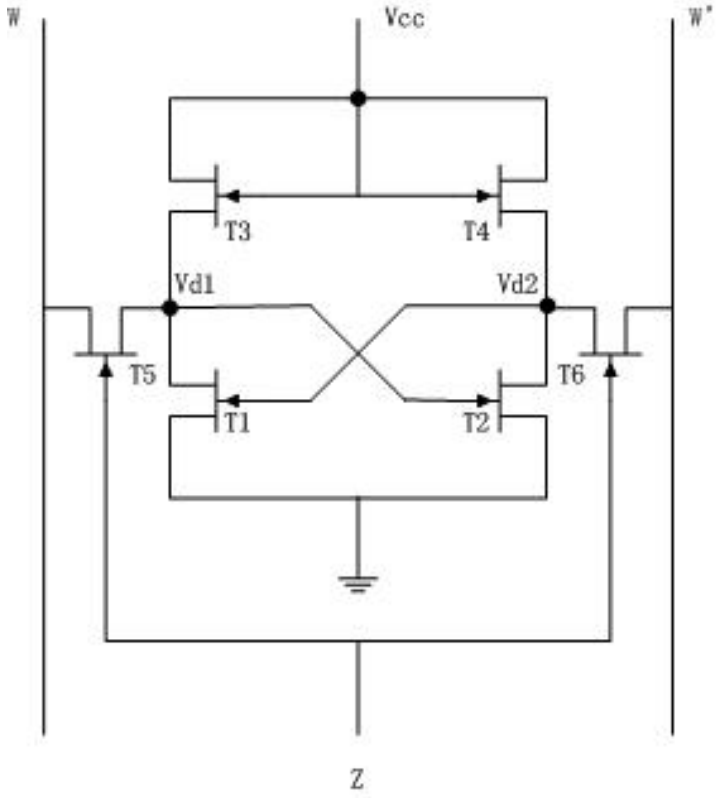
双极型的只有静态存储器

6. 2 随机存储器RAM

6.2.2 MOS型随机存储器

这节主要介绍六管静态存储器和四管动态存储器。

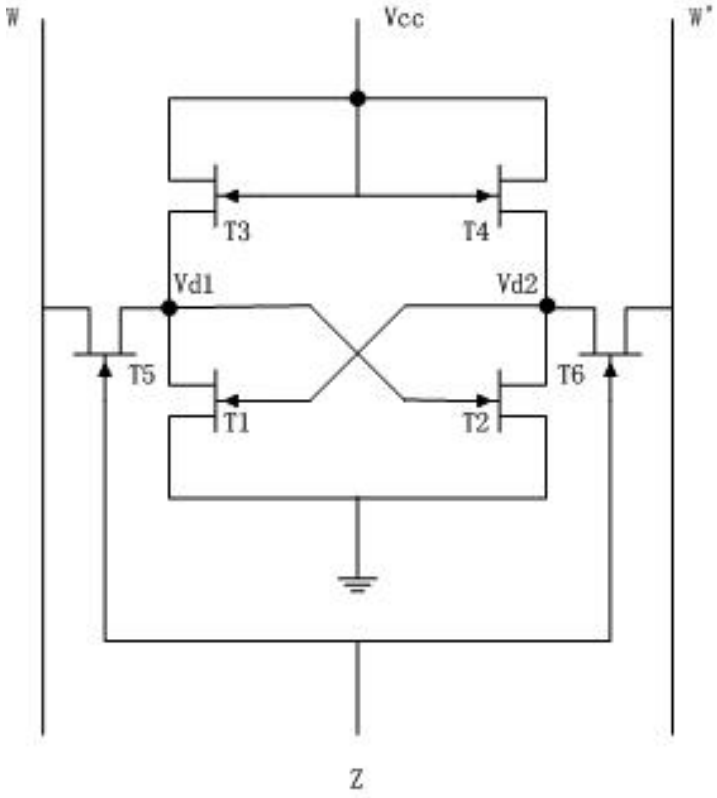
1.六管静态存储器



6.2 随机存储器RAM

6.2.2 MOS型随机存储器

- **T1和T2构成双稳态电路**
- **T3和T4为负载管**
- **T5和T6为门控管**
- **W、W’为读写线**
- **Z为选通线**



假设： T1导通，T2截止为“0”态

T1截止，T2导通为“1”态

6.2 随机存储器RAM

6.2.2 MOS型随机存储器

（1）保持状态

Vz=0V（低电位）

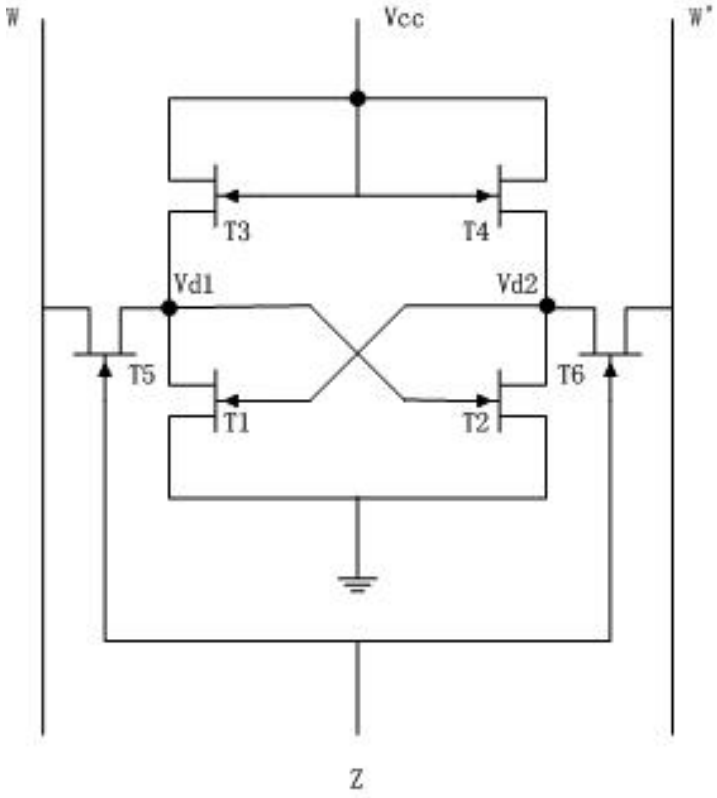
T5，T6均截止，存储单元和W、W’断开。

（2）读操作状态

Vz=5V（高电位）

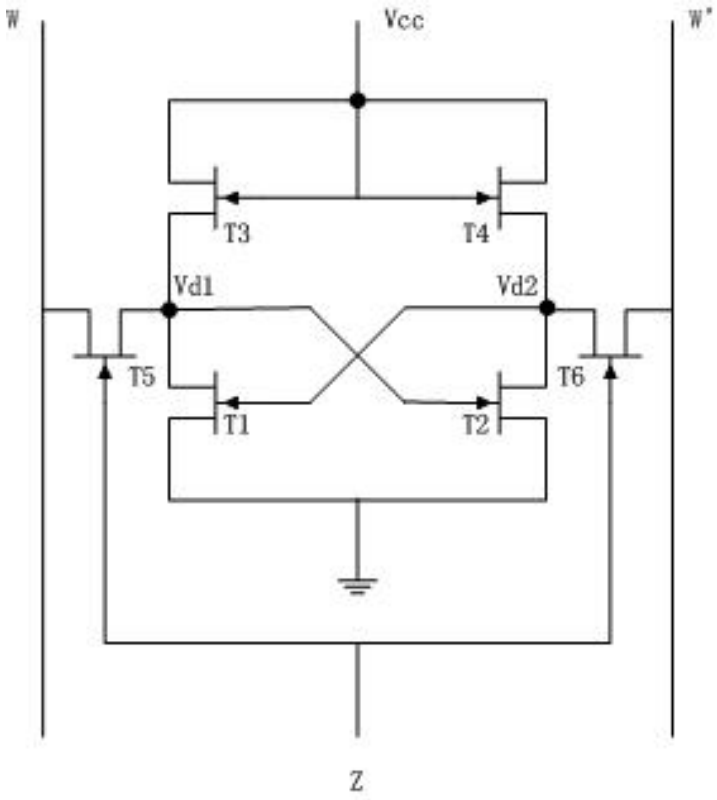
T5，T6导通

如原存信息“1”，即T1截止，T2导通，则Vd1的高电位通过T5传到W线上，Vd2点的低电位通过T6传到W’。



如原存信息“0”，即T1导通，T2截止，则Vd1的低电位通过T5传到W线上，Vd2点的高电位通过T6传到W’。

所以，根据W上是高电位还是W’上是高电位，就可判断读出的是“0”还是“1”。



(3) 写操作状态

Vz=5V（高电位）

T5，T6导通

- **写入“1”**

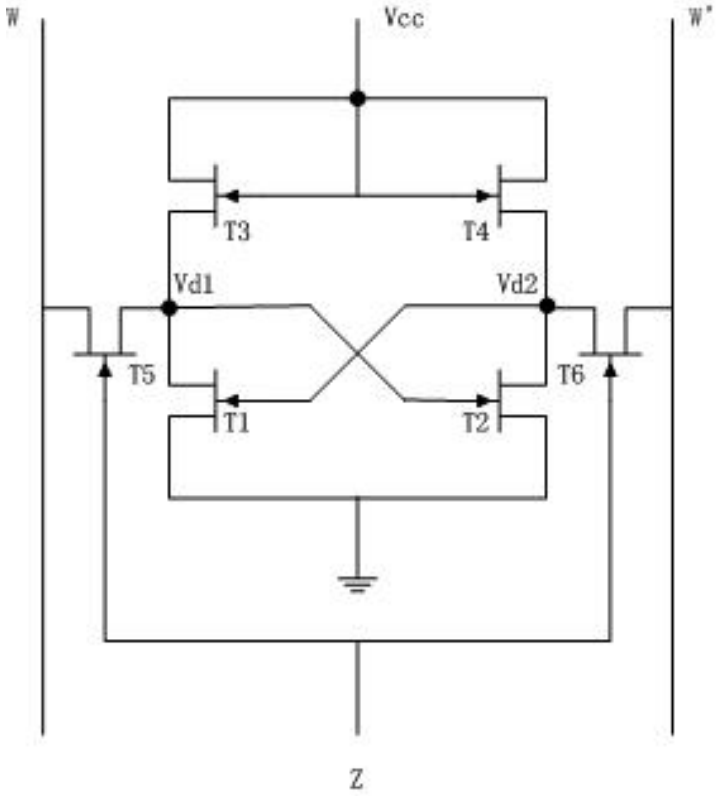
W线上加一个高电位（5V），W’线上加一个低电位（0V），使T1截止，T2导通，写“1”结束。

Vz↘0V，进入保持状态。

- **写入“0”**

W线上加一个低电位（0V），W’线上加一个高电位（5V），使T1导通，T2截止，写“0”结束。

Vz↘0V， 进入保持状态。



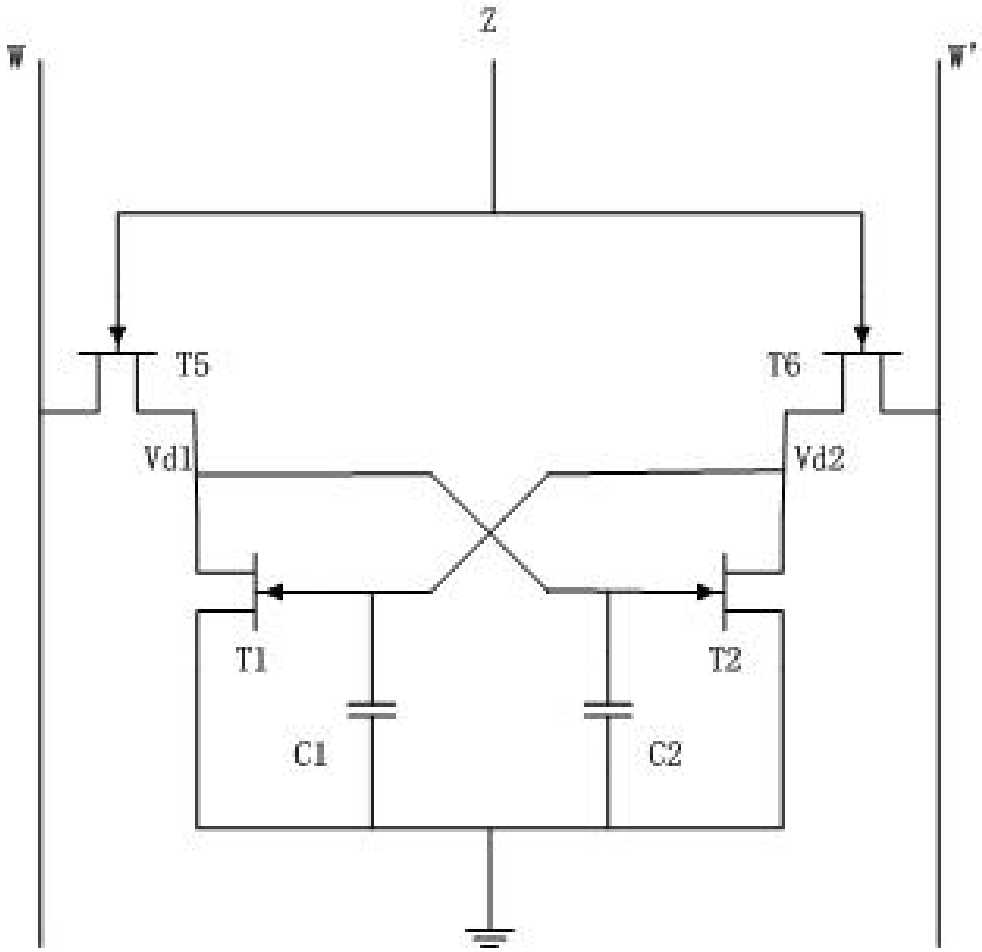
（4）静态存储器特点

①不需要刷新外围电路

②结构简单，可靠性高

③集成度低，功耗高

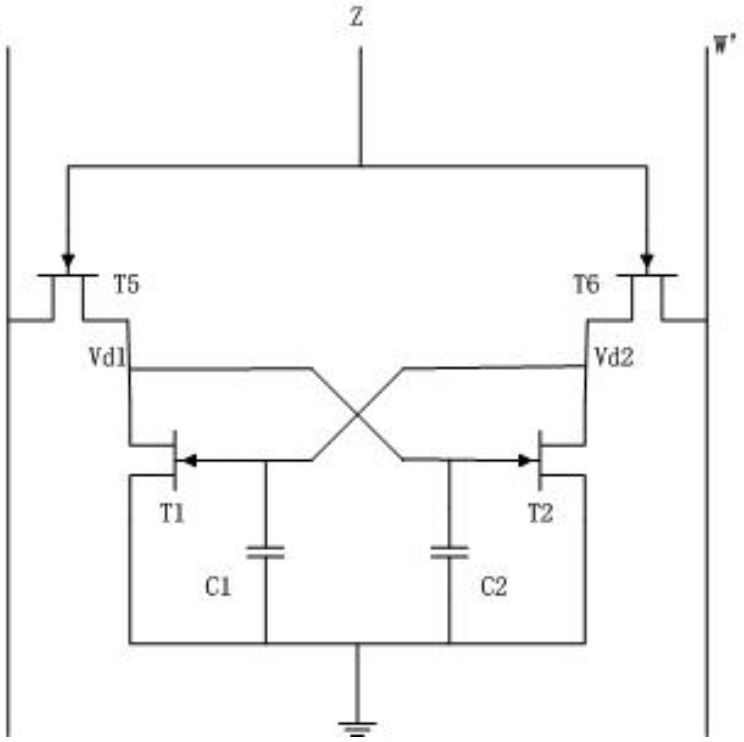
2.四管动态存储器



6.2 随机存储器RAM

6.2.2 MOS型随机存储器

从图中看出，它省去了六管存储电路中的T3和T4，而二进制信息是以电荷形式存在MOS晶体管T1和T2的栅源极电容器C1和C2上。



如C2被充电到高电位，C1为低电位，则T1截止，T2导通。

如C1被充电到高电位，C2为低电位，则T1导通，T2截止。

假设：T1导通，T2截止为“0”态；

T1截止，T2导通为“1”态

6.2 随机存储器RAM

6.2.2 MOS型随机存储器

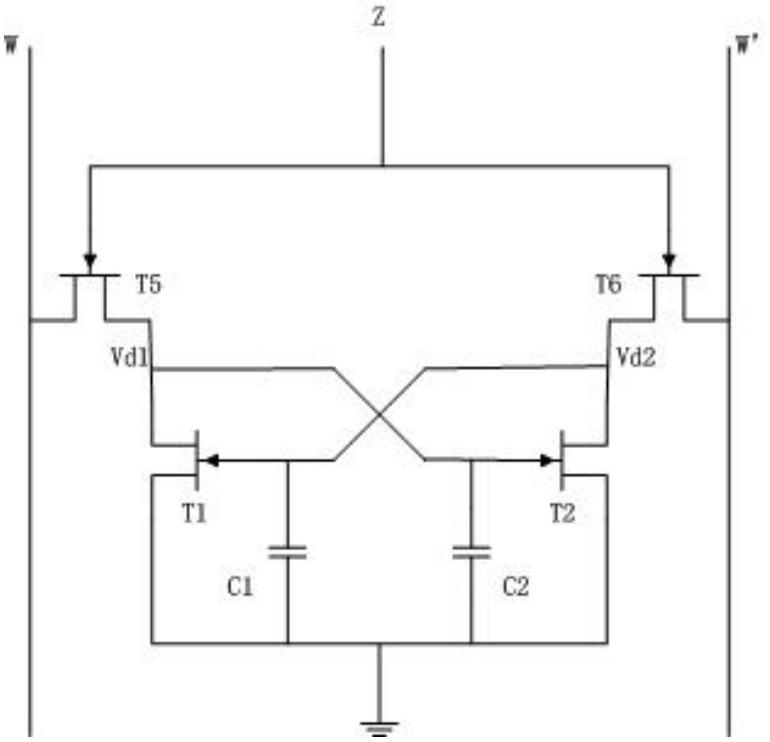
（1）读操作状态

Vz=高电位，Vw、Vw’=高电位，T5、T6导通。

如原存信息“1”，即T1截止，T2导通。W线上无电流，W’线上有电流。

如原存信息“0”，即T1导通，T2截止。W线上有电流，W’线上无电流。

所以，通过判断W线上有无电流就可知道读出的是“1”还“0”。



6.2 随机存储器RAM

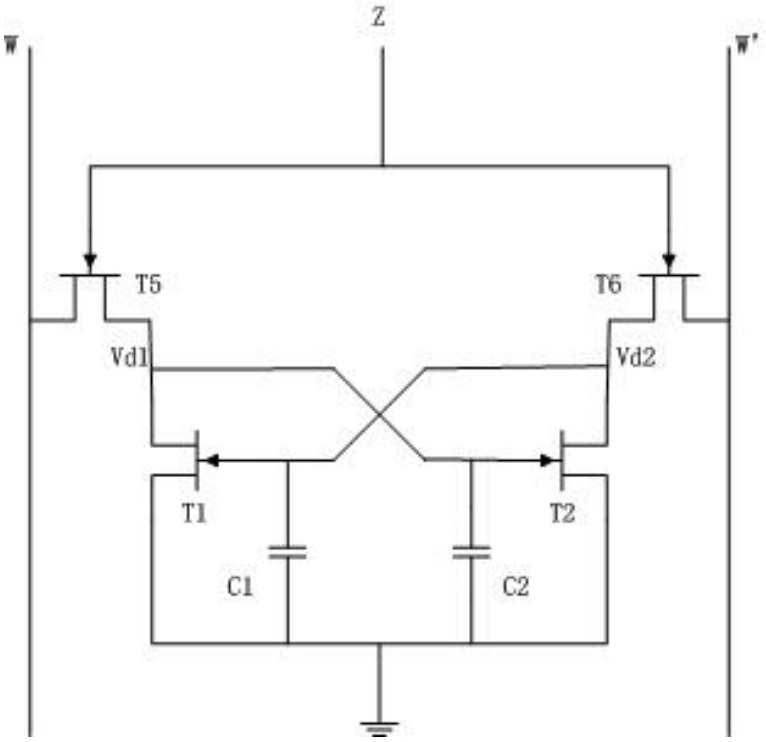
6.2.2 MOS型随机存储器

（2）写操作状态

Vz为高电位，T5、T6导通

如写“1”，则Vw为高电位，Vw’为低电位，W线上的高电位通过T5向C2充电到高电位，W’线的低电位通过T6使C1放电到低电位，使得T1截止，T2导通，写“1”结束，Vz↘低电位，进入保持状态。

如写“0”，则Vw为低电位，Vw’为高电位，W线上的低电位通过T5使C2放电到低电位，W’线的高电位通过T6向C1充电到高电位，使得T1导通，T2截止，写“0”结束，Vz↘低电位，进入保持状态。

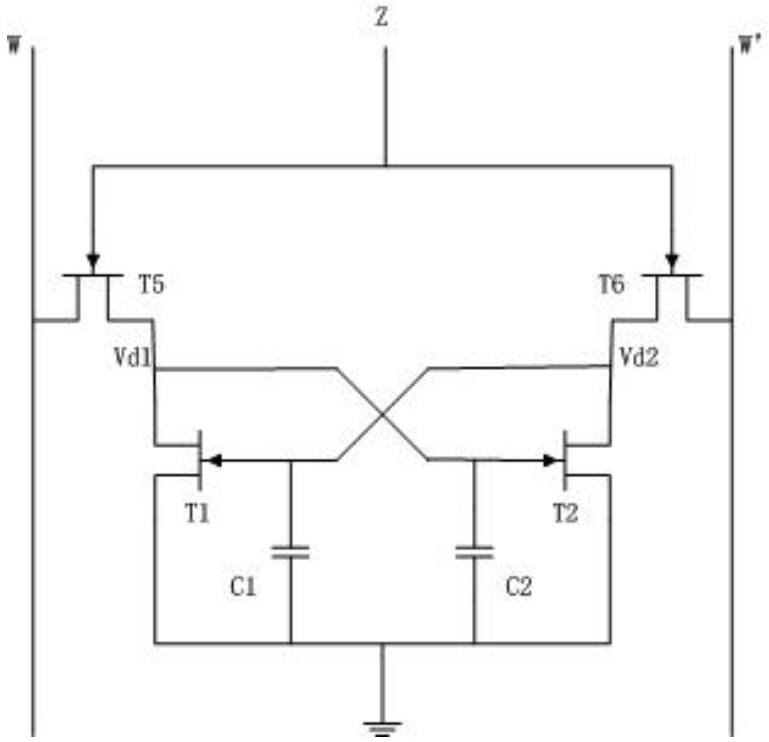


（3）保持状态

Vz为低电位，T5、T6截止。

由于信息是存放在C1或C2上，虽然MOS晶体管栅源极泄漏电流很小，但总会有一点，因而C1或C2上电荷就会慢慢地释放，一旦C1或C2上的电荷释放完，T1、T2都截止，信息丢失。

一般泄漏电流在10^{-10}A以下，但C1、C2也很小，一般在10^{-1}PF以下，电荷在C1、C2上的保持时间计算公式：



$$T = \frac{C\Delta V}{I}$$

C----单位为法（F)

ΔV----节点允许电压变化值，单位V

I----泄漏电流，单位A

如：C=0.2PF， ΔV=1V，I=10^{-10}A

则 $T = \frac{0.2\times10^{-12}\times1}{10^{-10}} = 2ms$

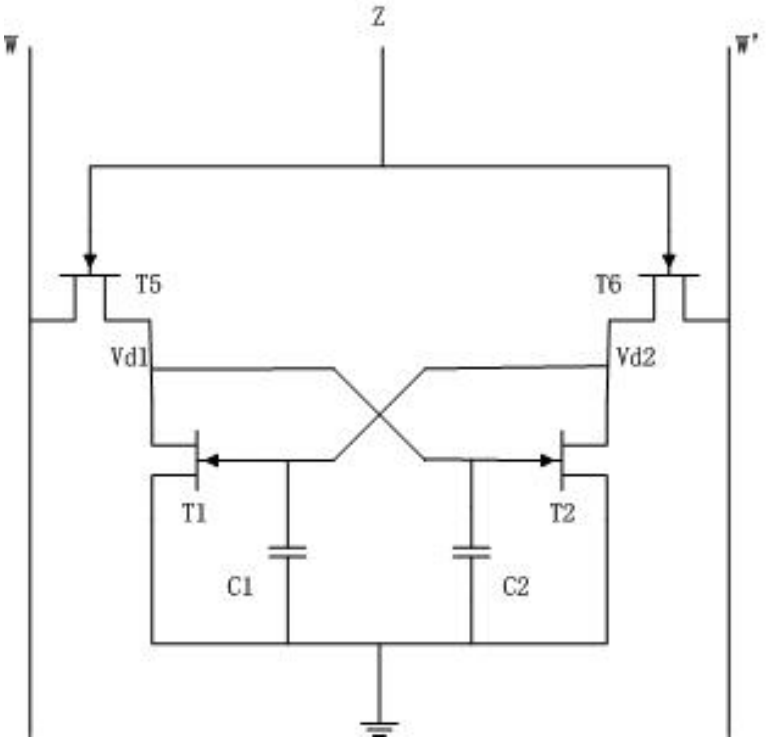
所以，动态存储器在保持状态下，每隔2ms要进行刷新一次，否则，存放在存储器中的信息将丢失。

(4) 刷新过程

刷新过程和读的过程基本相同。

Vz为高电位，Vw、Vw'为高电位，则W、W'线上的高电位通过T5、T6对C1或C2进行充电，保持C1或C2上的电荷。

与读的区别：刷新时，不理会W、W'线上的电流。



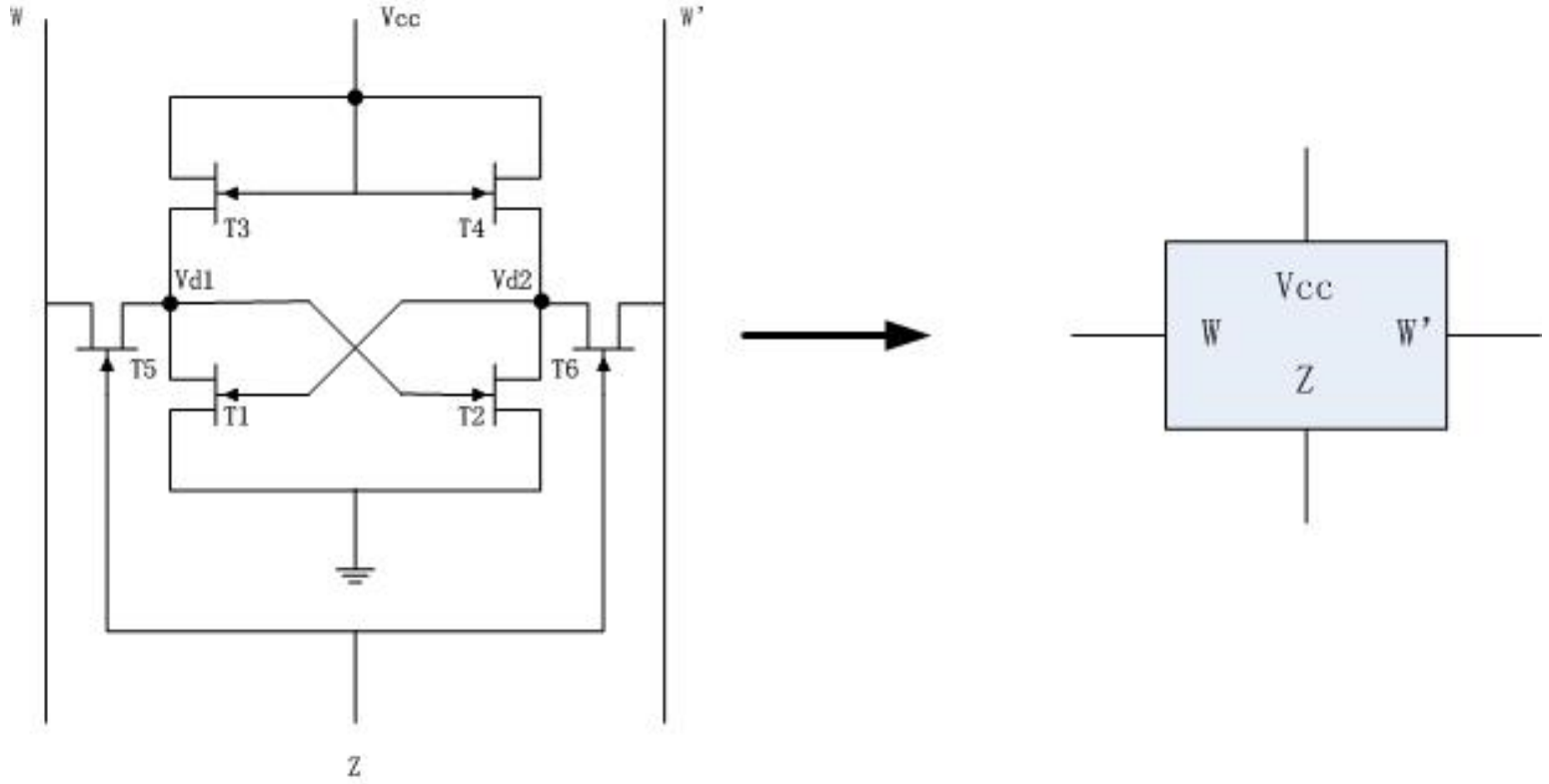
6.2 随机存储器RAM
6.2.2 MOS型随机存储器

- **动态存储器的特点**

①集成度高，功耗低

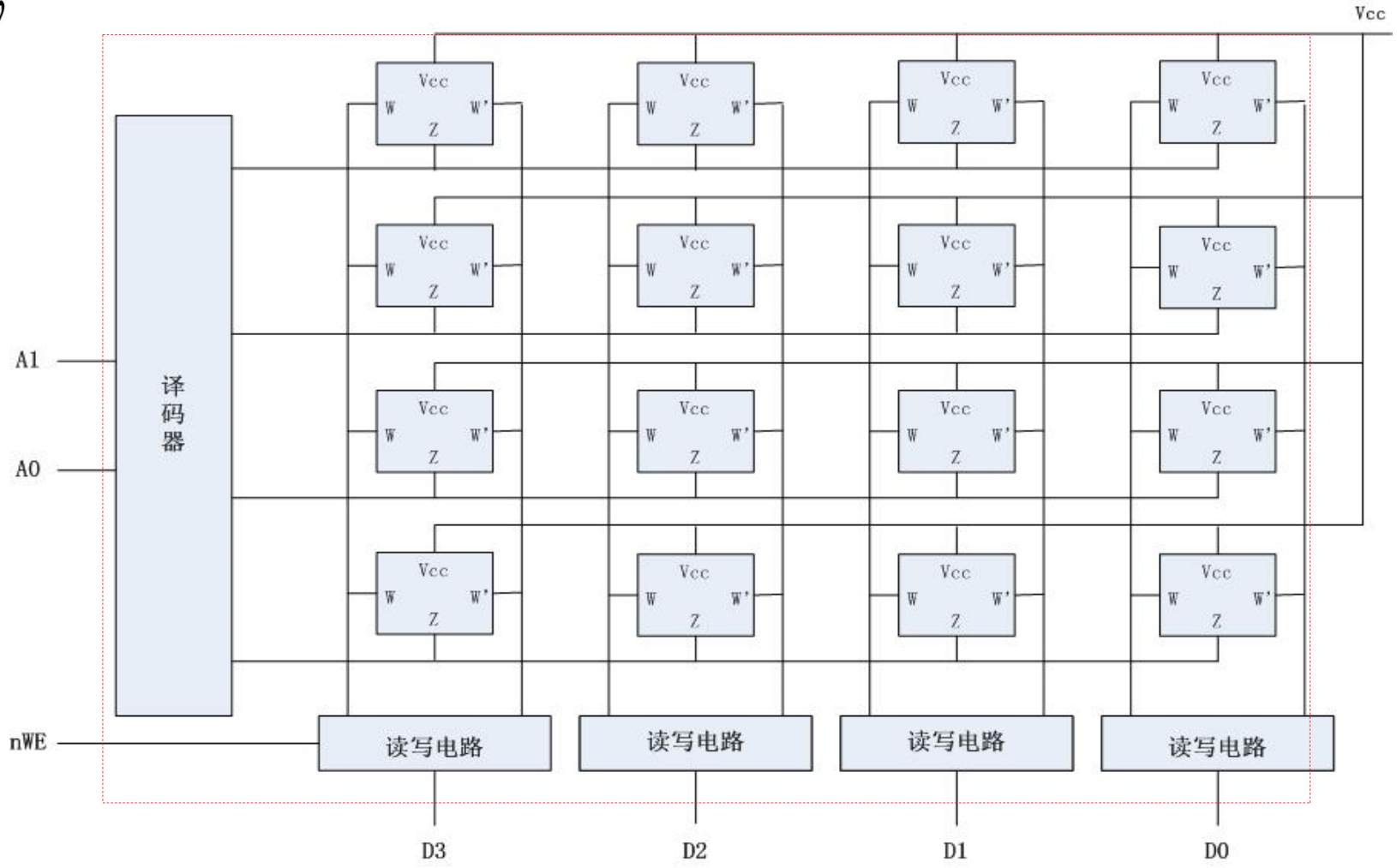
②外围电路复杂，在刷新时不能读写

3.MOS静态存储系统结构图



6.2 随机存储器RAM

6.2.2 MOS型随机存储器



6.3 只读存储器ROM

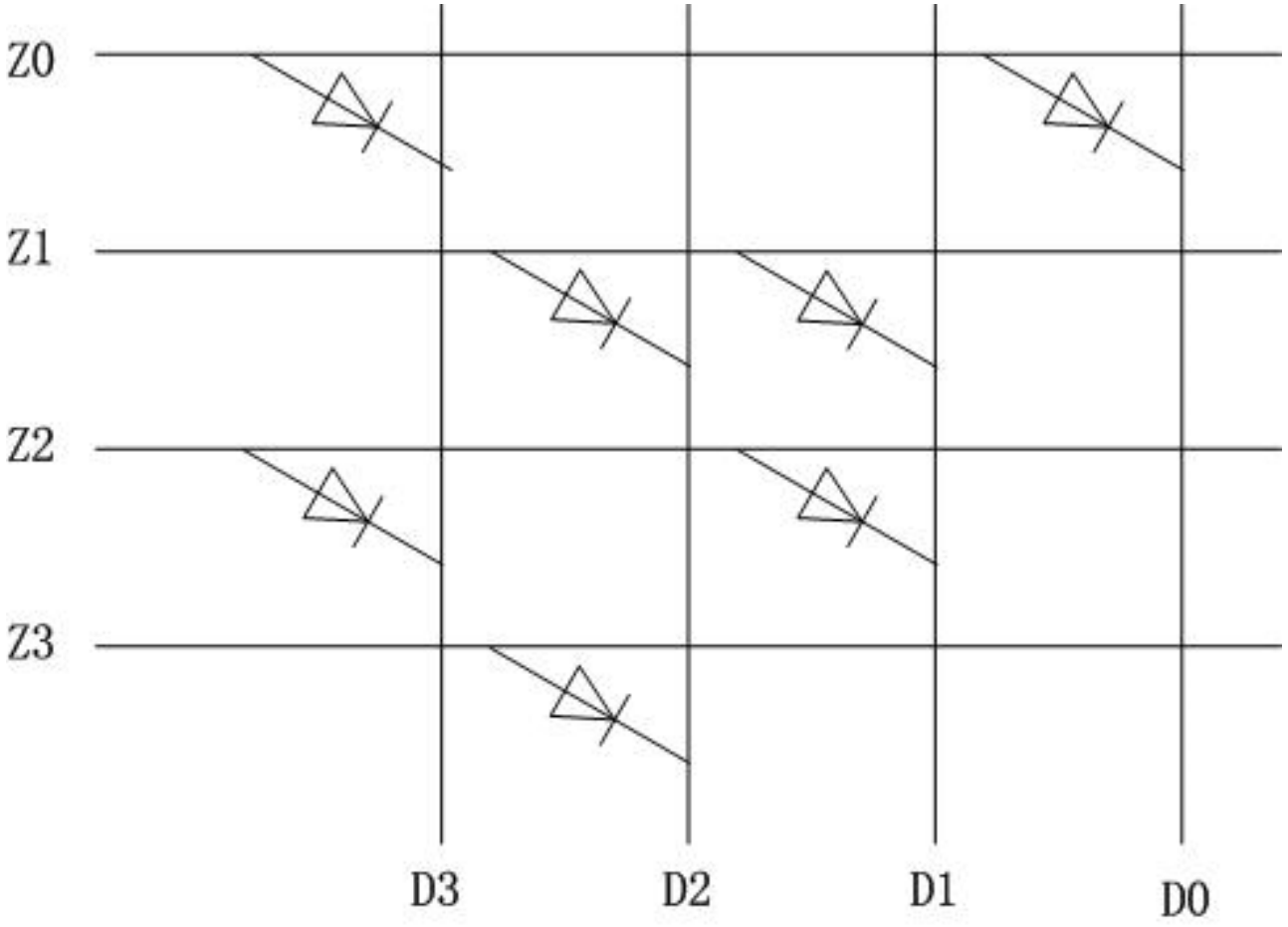
前面介绍的DRAM和SRAM均为可任意读/写的随机存储器，当掉电时，所存储的内容立即消失，所以是易失性存储器。下面介绍的半导体存储器，即使停电，所存储的内容也不会丢失。根据半导体制造工艺的不同，可分为ROM,PROM,EPRO，E^2PROM和Flash Memory。

6.3.1 只读存储器ROM

掩模式ROM由芯片制造商在制造时写入内容，以后只能读而不能再写入。其基本存储原理是以元件的“有/无”来表示该存储单元的信息(“1”或“0”)，可以用二极管或晶体管作为元件，显而易见，其存储内容是不会改变的。

6.3 只读存储器ROM

6.3.1 只读存储器ROM

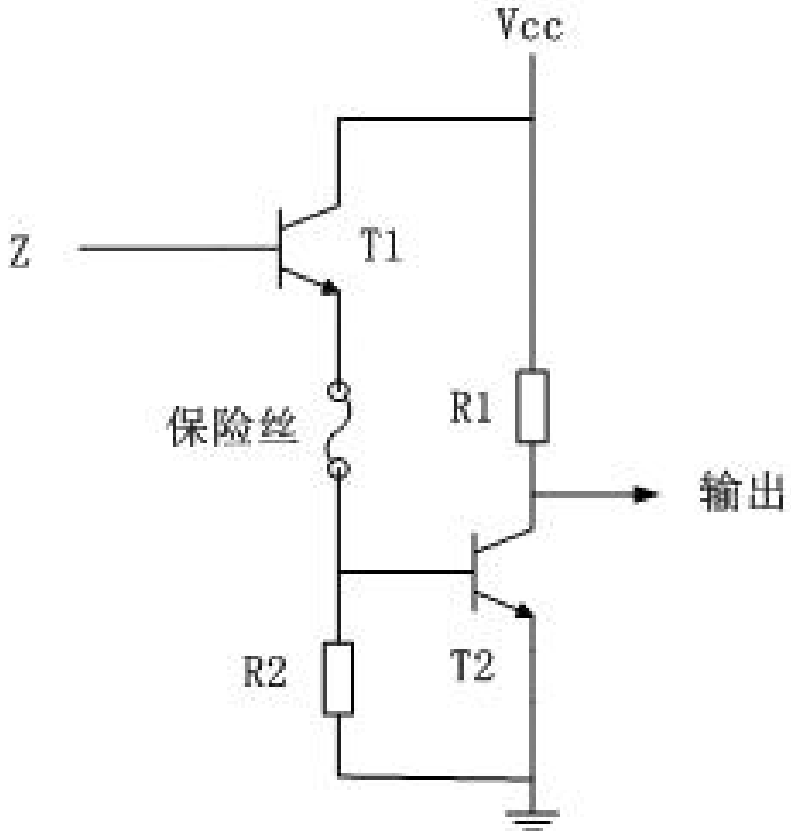


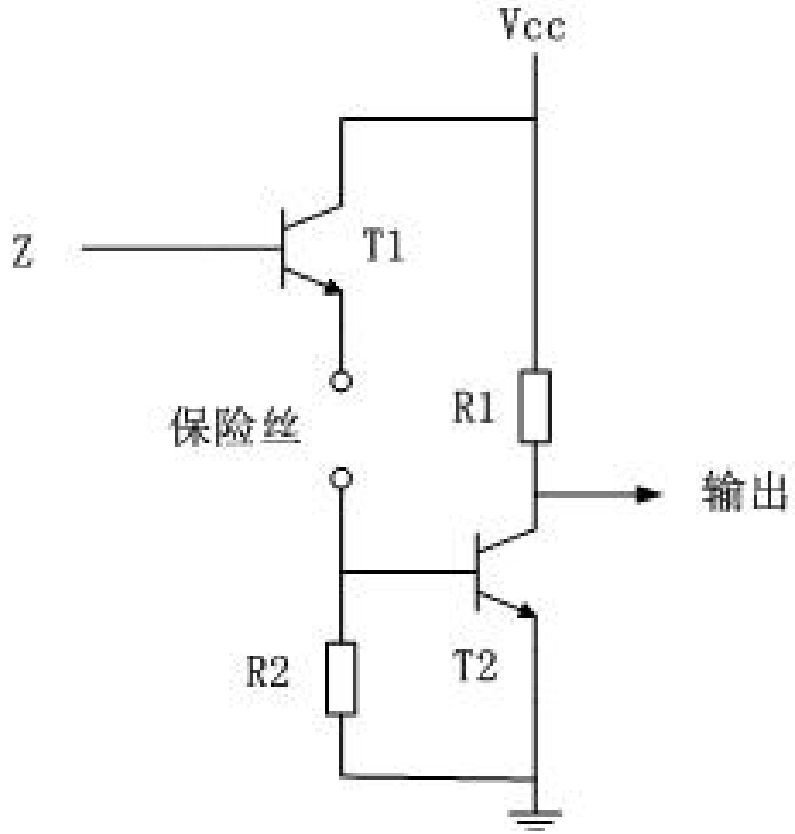
6.3.2可编程序的只读存储器(PROM)

PROM可由用户根据自己的需要来确定ROM中的内容，常见的熔丝式PROM是以熔丝的接通和断开来表示所存的信息为“1”或“0”。刚出厂的产品，其熔丝是全部接通的，使用前，用户根据需要断开某些单元的熔丝(写入)。显而易见，断开后的熔丝是不能再接通了，因此，它是一次性写入的存储器。掉电后不会影响其所存储的内容。

6.3 只读存储器ROM

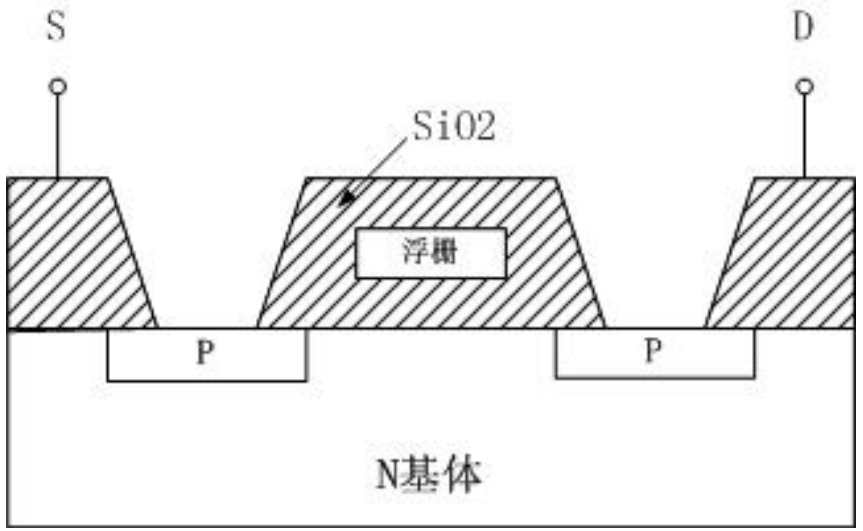
6.3.2 可编程序的只读存储器(PROM)



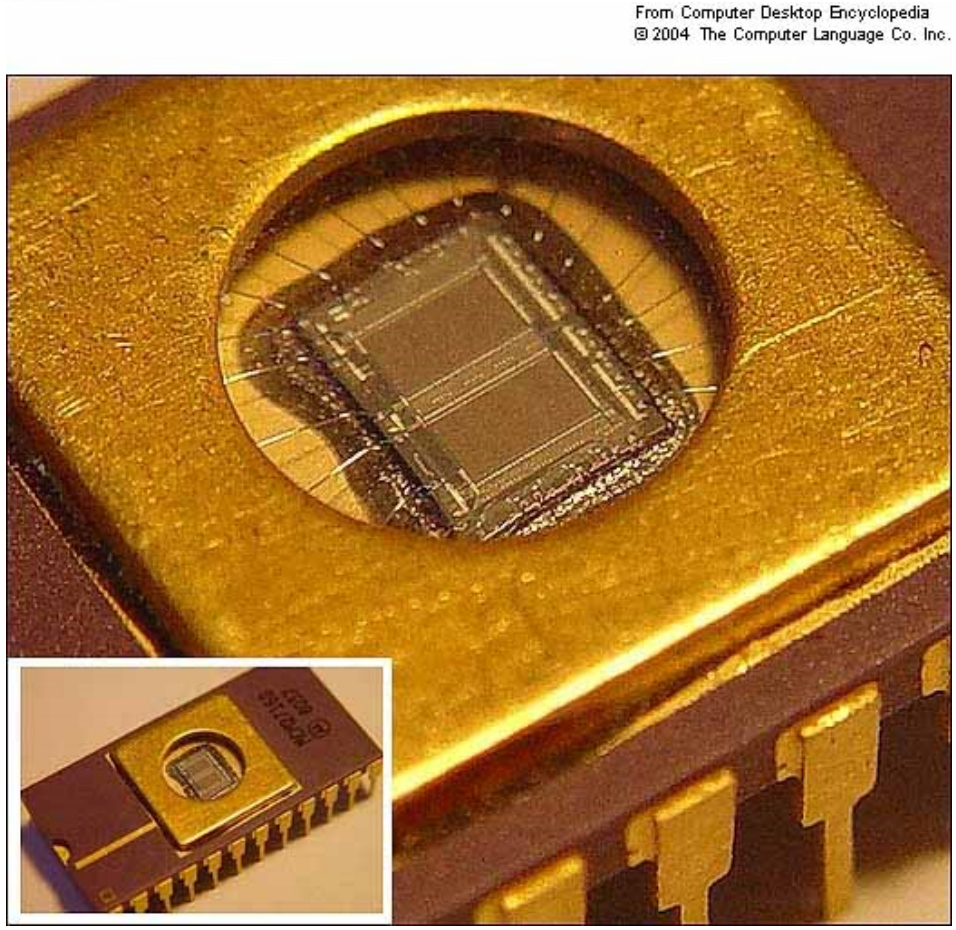


6.3.3可擦可编程序的只读存储器(EPROM)

为了能多次修改ROM中的内容，产生了EPROM。其基本存储单元由一个管子组成，但与其他电路相比管子内多增加了一个浮置栅，如图所示。



可擦可编程序的只读存储器(EPROM)



6.3.4可电擦可编程序只读存储器(E²PROM)

E2PROM的编程序原理与EPROM相同，但擦除原理完全不同，重复改写的次数有限制(因氧化层被磨损)，一般为10万次。其读写操作可按每个位或每个字节进行，类似于SRAM，但每字节的写入周期要几毫秒，比SRAM长得多。E2PROM每个存储单元采用两个晶体管。其栅极氧化层比EPROM薄，因此具有电擦除功能。

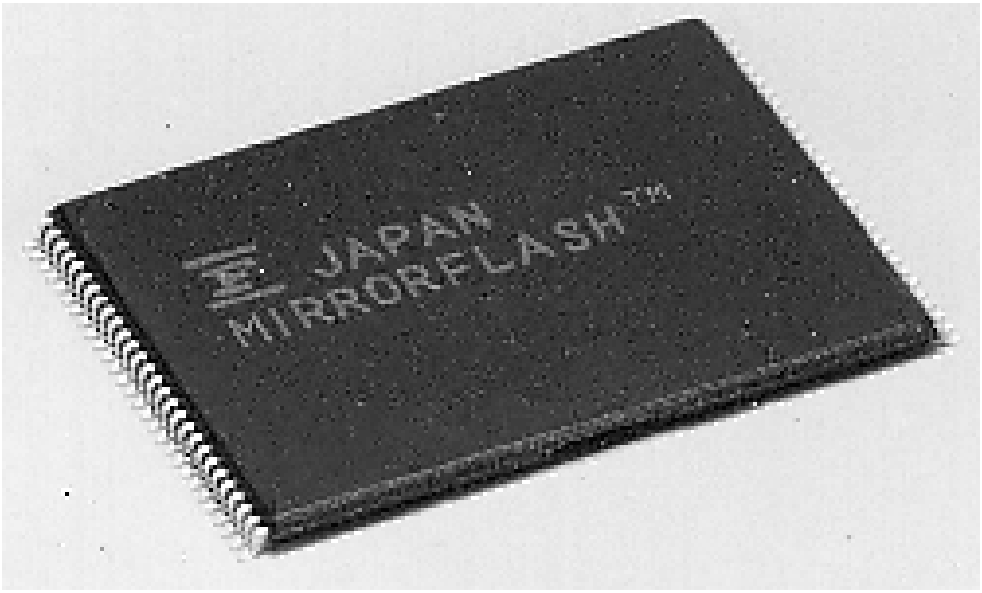
可电擦可编程序只读存储器(E^2PROM)

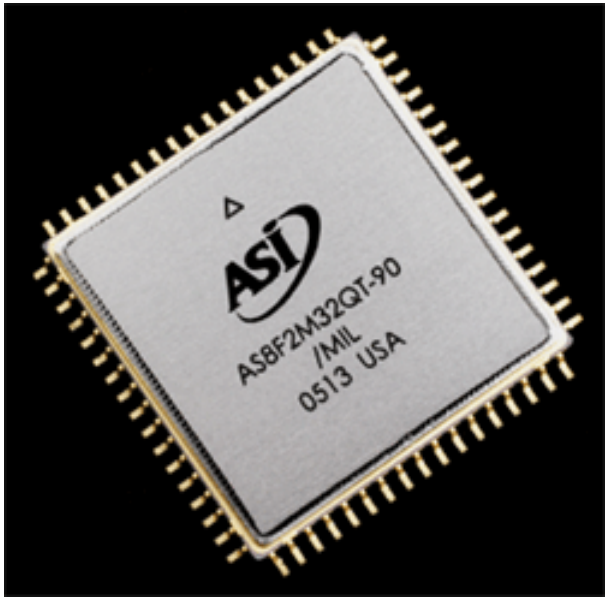
6.3.5快擦除读写存储器(Flash Memory)

Flash Memory是在EPROM与E2PROM基础上发展起来的，它与EPROM一样，用单管来存储一位信息，它与E2PROM相同之处是用电来擦除。但是它只能擦除整个区或整个器件。

快擦除读写存储器兼有ROM和RAM两者的性能，又有ROM，DRAM一样的高密度。目前价格已略低于DRAM，芯片容量已接近于DRAM，是唯一具有大存储量、非易失性、低价格、可在线改写和高速度(读)等特性的存储器。它是近年来发展很快很有前途的存储器。

快擦除读写存储器(Flash Memory)





6. 5半导体存储系统的组成

半导体存储器的读写时间一般在十几至几百毫微秒之间，其芯片集成度高，体积小，片内还包含有译码器和寄存器等电路。常用的半导体存储器芯片有多字一位片和多字多位(4位、8位)片，如16M位容量的芯片可以有16M×1位和4M×4位等种类，如何利用存储器件来构成实际的存储系统。

6.5.1存储系统的一般设计原则和方法

- **设计原则**

根据使用要求，结合实际条件进行设计，使其具有良好的性能价格比。

6. 5半导体存储系统的组成

6.5.1 存储系统的一般设计原则和方法

- **使用要求**

1. 设计的存储器是主存、缓存还是控存。
2. 存储器的主要指标，如速度、容量、字长等。

- **实际条件**

现有的器件，技术条件和生产条件。

存储系统的设计分为三步：

1.系统设计

要从计算机系统的角度，提出对存储系统的主要指标、功能要求以及结构形式等，如存储容量、存取时间、控制方式等。

2.逻辑设计

根据系统设计提出的要求，进行逻辑设计，在设计中要考虑逻辑电路的扇出和负载，信号的传送和衰减，逻辑级的延迟与匹配，在满足技术指标下，要尽量简化逻辑，节省器件。

3.工艺设计

依据系统设计和逻辑设计的资料，进行工艺设计，如器件的排列、印刷电路等设计。

本节只介绍逻辑设计

6.5.2 存储系统的逻辑设计

一个存储器的芯片的容量是有限的，它在容量或字长方面与实际存储器的要求都有很大差距，所以需要在字向和位向进行扩充才能满足需要。

1.存储器容量扩展

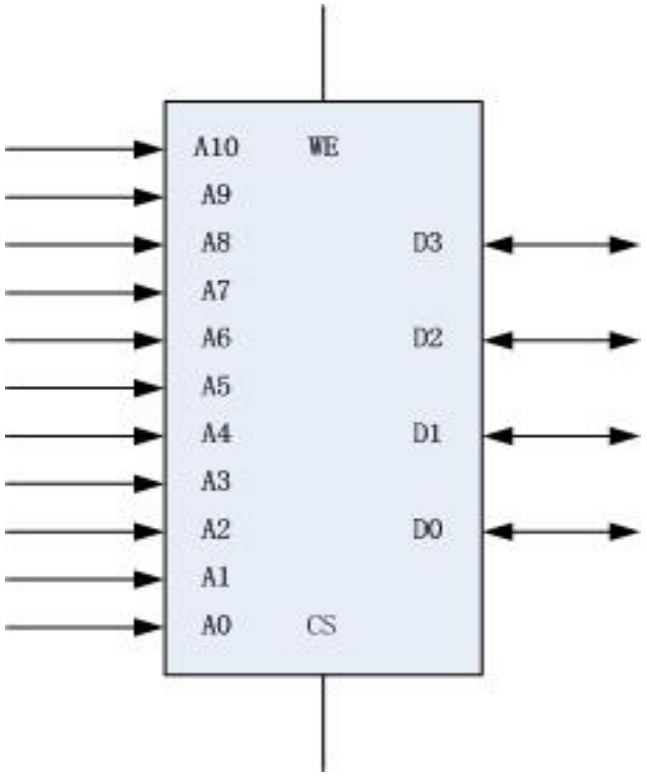
(1) 位扩展

位扩展指的是用多个存储器器件对字长进行扩充。位扩展的连接方式是将多片存储器的地址、片选CS、读写控制端R/W相应并联，数据端分别引出。

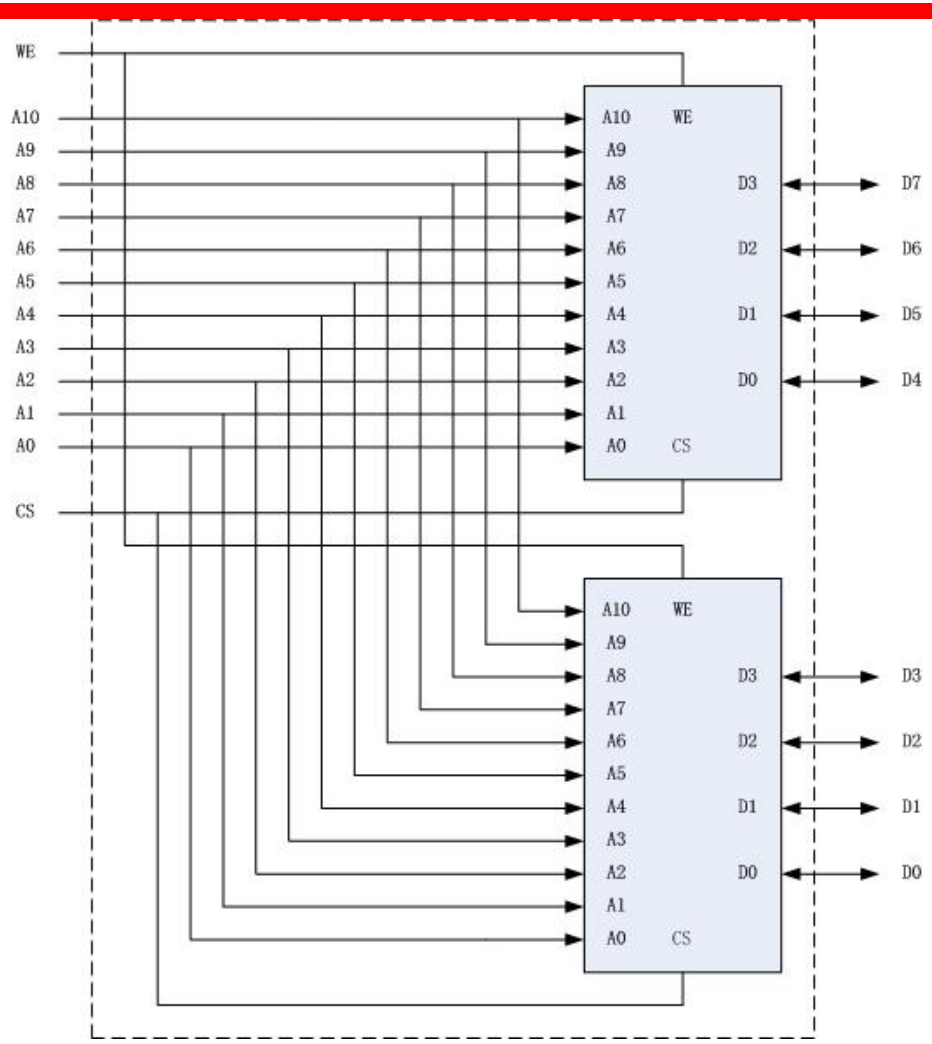
6. 5半导体存储系统的组成

6.5.2 存储系统的逻辑设计

例： 用2K×4位的芯片构成2K×8位的存储系统。



位扩展

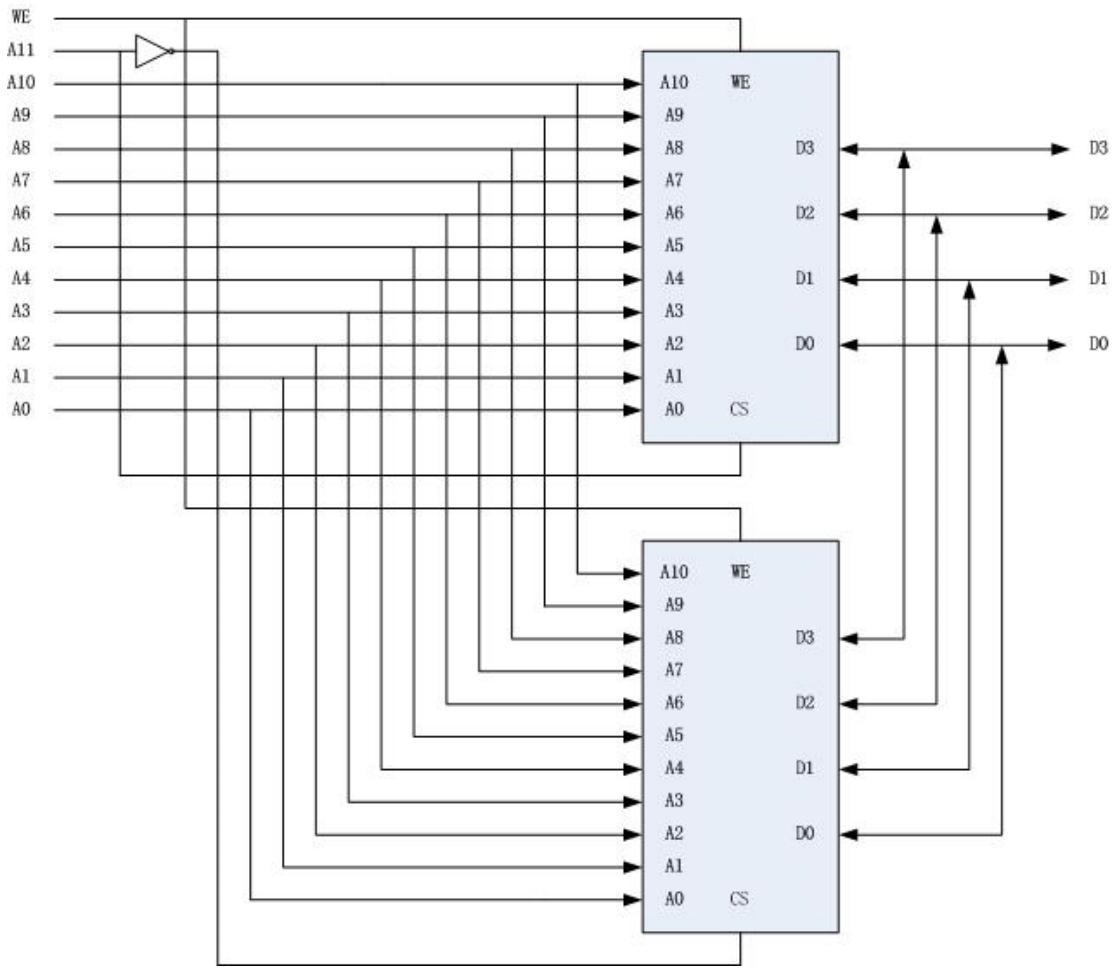

WE
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
CS
A10 WE
A9
A8 D3
A7
A6 D2
A5
A4 D1
A3
A2 D0
A1
A0 CS
D7
D6
D5
D4
A10 WE
A9
A8 D3
A7
A6 D2
A5
A4 D1
A3
A2 D0
A1
A0 CS
D3
D2
D1
D0

(2) 字扩展

字扩展指的是增加存储器中字的数量(即容量扩展）。静态存储器进行字扩展时，将各芯片的地址线、数据线、读写控制线相应并联，而由片选信号来区分各芯片的地址范围。

例：用2K×4位的芯片构成4K×4位的存储系统。

字扩展

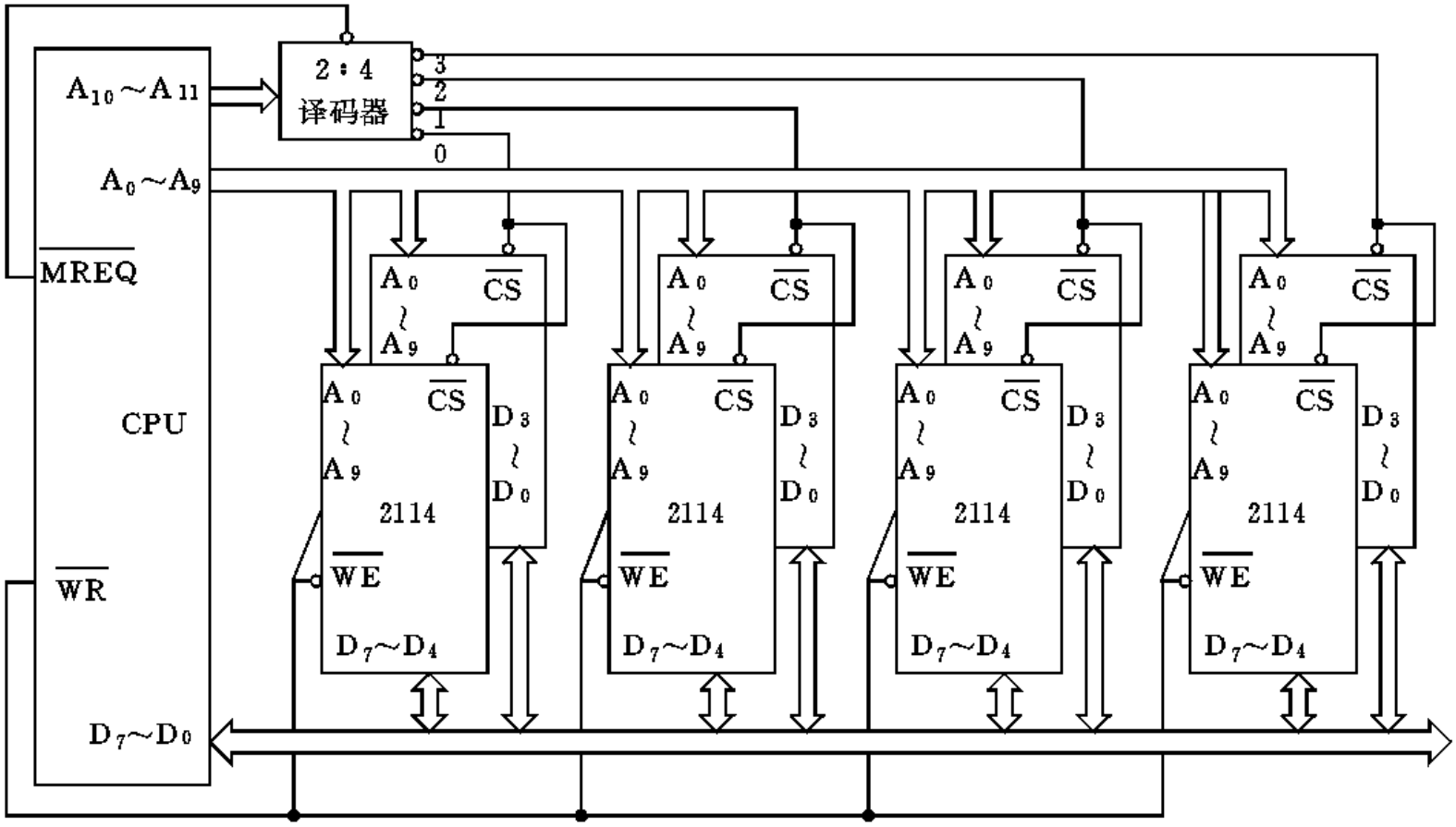

WE
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
A10
WE
A9
A8
D3
A7
A6
D2
A5
A4
D1
A3
A2
D0
A1
A0
CS
D3
D2
D1
D0

(3) 字位同时扩展

实际存储器往往需要字向和位向同时扩充。一个存储器的容量为M×N位，若使用L×K位存储器芯片，那么，这个存储器共需要M/L×N/K个存储器芯片。一般讲，先进行位扩展，再进行字扩展。

一个小容量存储器与CPU的连接方式如图所示。

字位同时扩展


2：4
译码器
3
2
1
0
A10～A11
A0～A9
MREQ
CPU
WR
D7～D0
A0～A9
CS
D3～D0
2114
WE
D7～D4

2. 存储控制

在存储器中，往往需要增设附加电路。这些附加电路包括地址多路转换线路、地址选通、刷新逻辑，以及读/写控制逻辑等。

在大容量存储器芯片中，为了减少芯片地址线引出端数目，将地址码分两次送到存储器芯片，因此芯片地址线引出端减少到地址码的一半。

刷新逻辑是为动态MOS随机存储器的刷新准备的，通过定时刷新、保证动态MOS存储器的信息不致丢失。

6. 5半导体存储系统的组成

6.5.2 存储系统的逻辑设计

动态MOS存储器采用“读出”方式进行刷新。因为在读出过程中恢复了存储单元的MOS栅极电容电荷，并保持原单元的内容，所以，读出过程就是再生过程。

但是存储器的访问地址是随机的，不能保证所有的存储单元在一定时间内都可以通过正常的读写操作进行刷新，因此需要专门予以考虑。通常，在再生过程中只改变行选择线地址，每次再生一行，依次对存储器的每一行进行读出，就可完成对整个RAM的刷新。

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，这一段时间间隔称作再生周期，又叫刷新周期，一般为2ms。

通常有两种刷新方式。

(1) 集中刷新

集中式刷新指在一个刷新周期内，利用一段固定的时间，依次对存储器的所有行逐一再生，在此期间停止对存储器的读和写。

例：一个存储器有1024行，系统工作周期为200ns。RAM刷新周期为2ms。这样，在每个刷新周期内共有10000个工作周期，其中用于再生的为1024个工作周期，用于读和写的为8976个工作周期。

集中刷新的缺点是在刷新期间不能访问存储器，有时会影响计算机系统的正确工作。

(2) 分布式刷新

采取在2ms时间内分散地将1024行刷新一遍的方法，具体做法是将刷新周期除以行数，得到两次刷新操作之间的时间间隔t，利用逻辑电路每隔时间t产生一次刷新请求。

动态MOS存储器的刷新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。这些线路可以集中在RAM存储控制器芯片中。

百年同济 TONGJI UNIVERSITY

3. 存储校验线路

计算机在运行过程中，主存储器要和CPU、各种外围设备频繁地高速交换数据。由于结构、工艺和元件质量等种种原因，数据在存储过程中有可能出错，所以，一般在主存储器中设置差错校验线路。

实现差错检测和差错校正的代价是信息冗余。信息代码在写入主存时，按一定规则附加若干位，称为校验位。在读出时，可根据校验位与信息位的对应关系，对读出代码进行校验，以确定是否出现差错，或可纠正错误代码。

习 题（1）

习题：

P86 1，4，5，6，7

6.6多存储体存储器及交叉存取

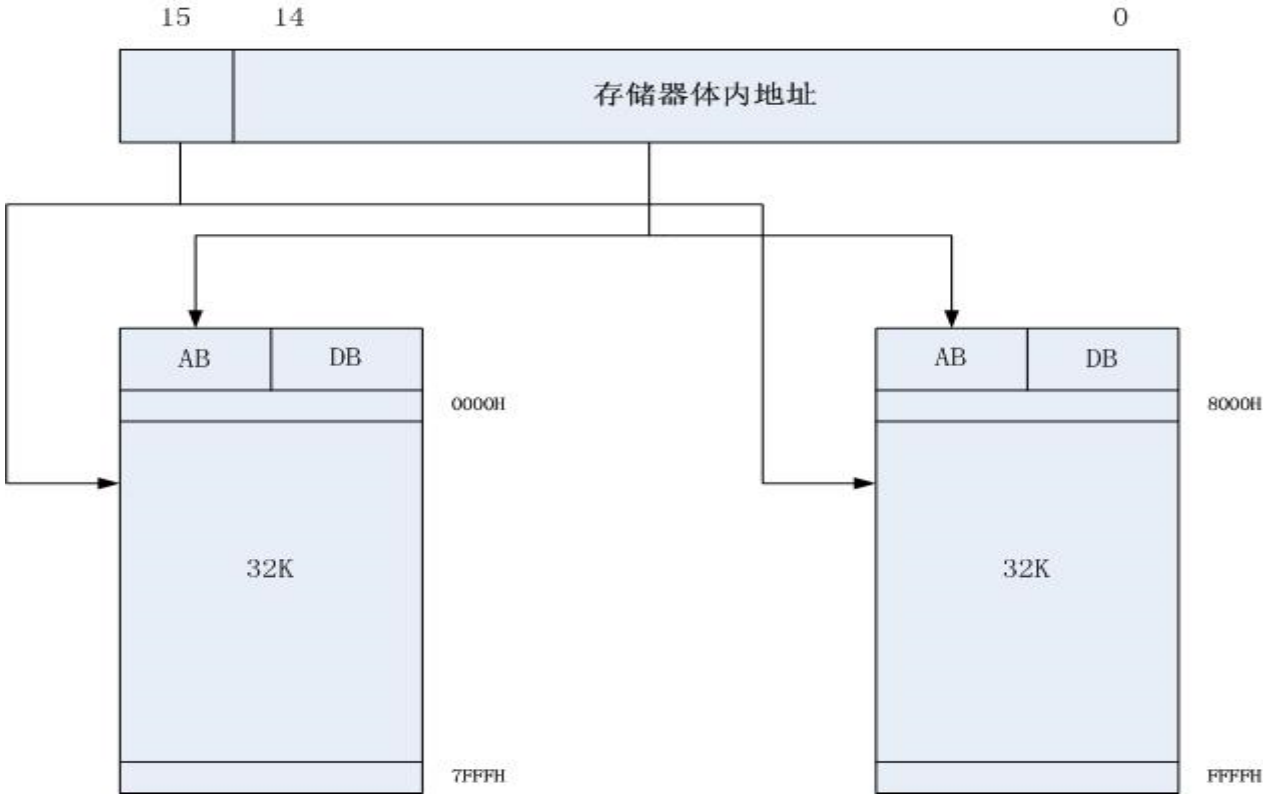
如主存储器由物理上分离的几个存储体构成，每一个存储体带有自己的地址缓冲寄存器（AR）和数据缓冲寄存器（DR），在任一给定时间，就可能有几个存储体来完成读写操作，因而，增加了整个系统传送的平均速率。

多体存储器地址划分：

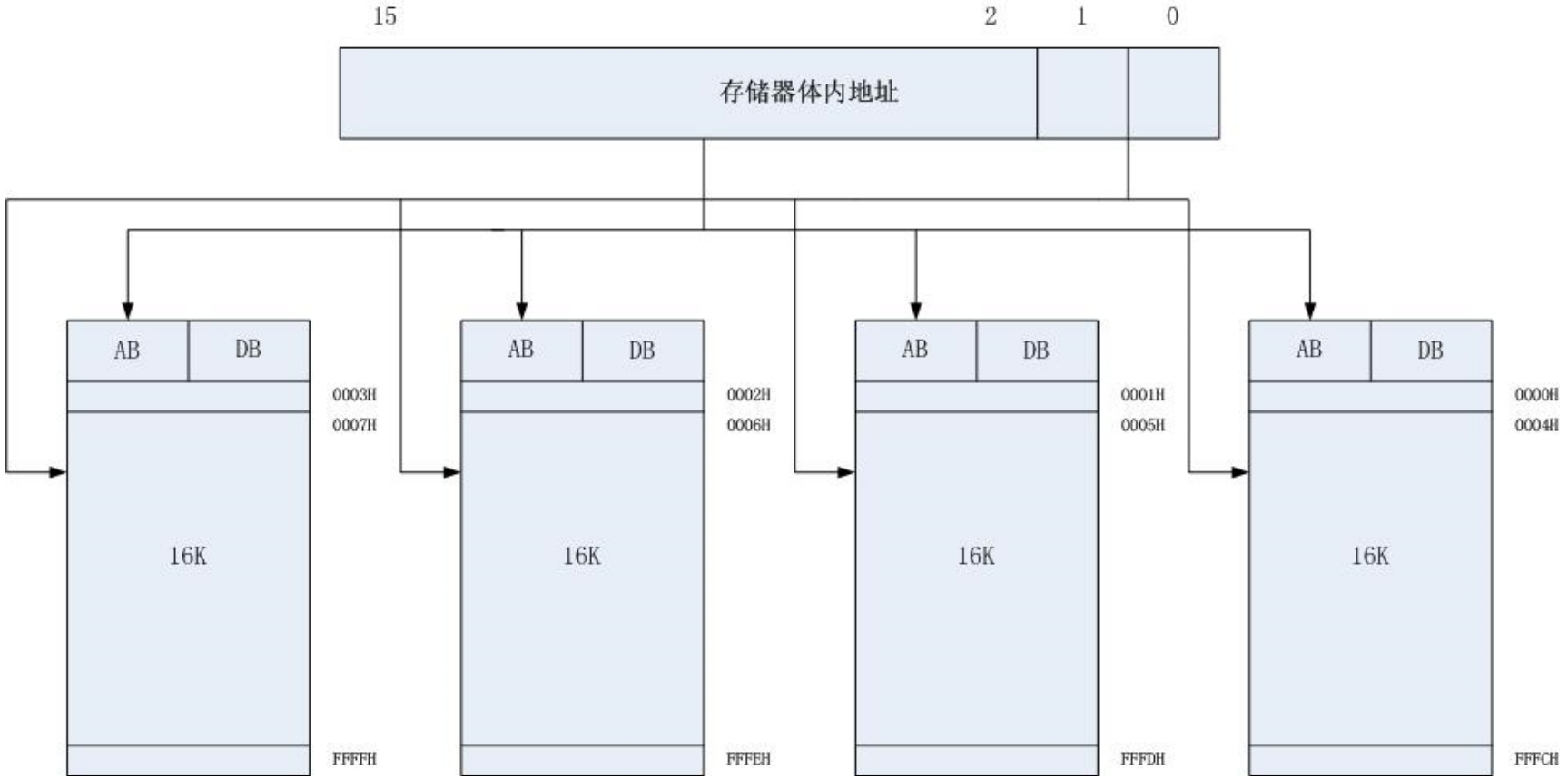
1.按高位地址划分

2.按低位地址划分

6.6.1 按高位地址划分



6.6.2 按低位地址划分


15
2
1
0
存储器体内地址
AB
DB
16K
0003H
0007H
FFFFH
0002H
0006H
FFFEH
0001H
0005H
FFFDH
0000H
0004H
FFFCH

按低位划分（交叉编址）有三个问题：

1. 模块数必须是2的整数幂，否则，地址不连续。
2. 任一模块失效就会造成地址空间的缺陷而导致整个程序无法运行。
3. 不利于存储器模块数的增量式扩展。

6.6.3 多体存储体访问控制

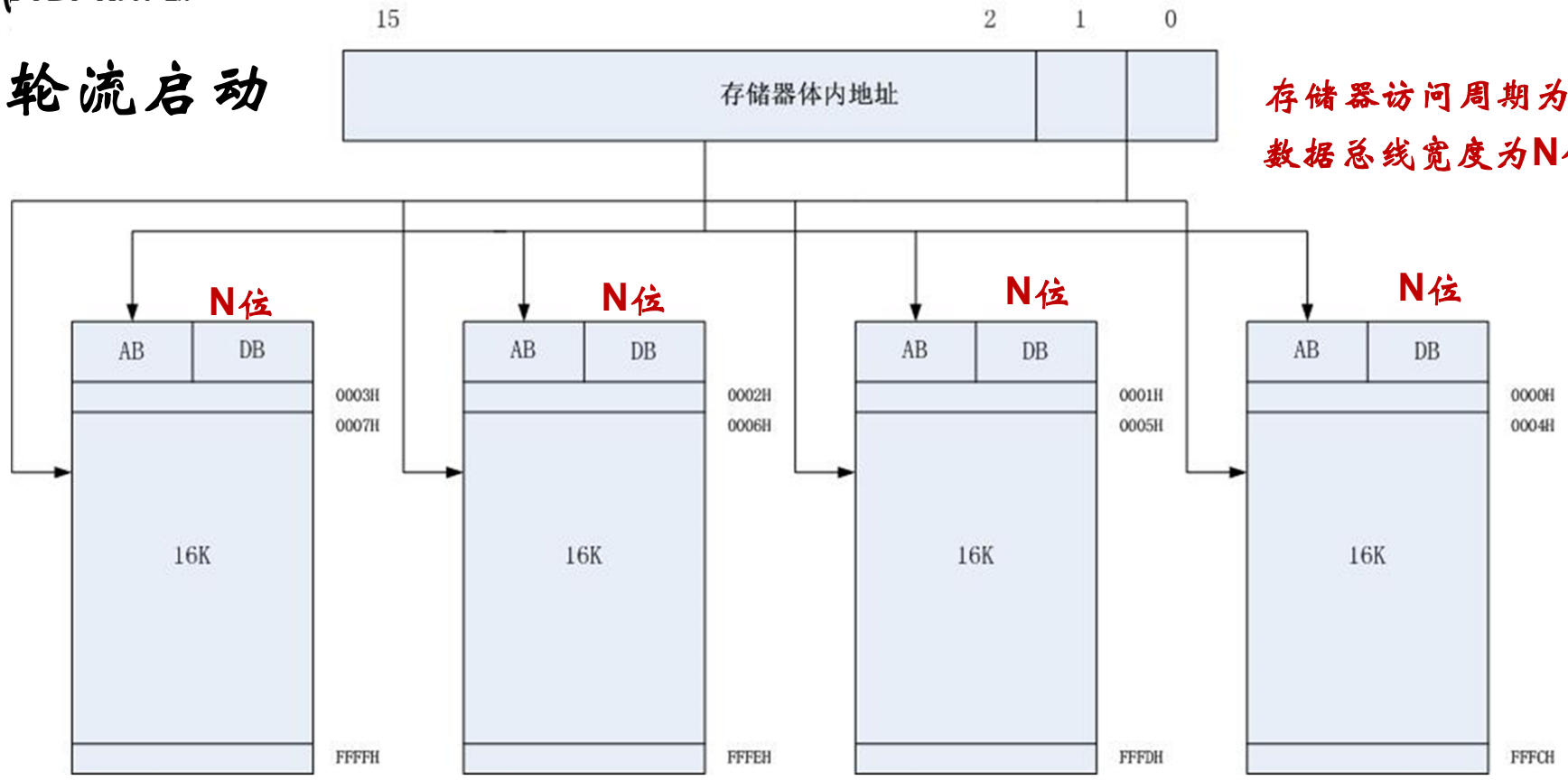
多体存储体可以有两种不同的方式进行访问：

1. 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。
2. 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。

6. 6多存储体存储器及交叉存取

6.6.3 多体存储体访问控制

轮流启动



存储器访问周期为T

数据总线宽度为N位

启动访问时刻3T/4　下次访问时刻7T/4

启动访问时刻2T/4　下次访问时刻6T/4

启动访问时刻T/4　下次访问时刻5T/4

启动访问时刻0　下次访问时刻T

前一种称为“同时访问”，后一种称为“交叉访问”。

同时访问能一次提供多个数据或多条指令，比较适合用于对多数据流或多指令流进行并行处理。

交叉访问最适合用于以流水线方式向中央处理器提供指令和数据。

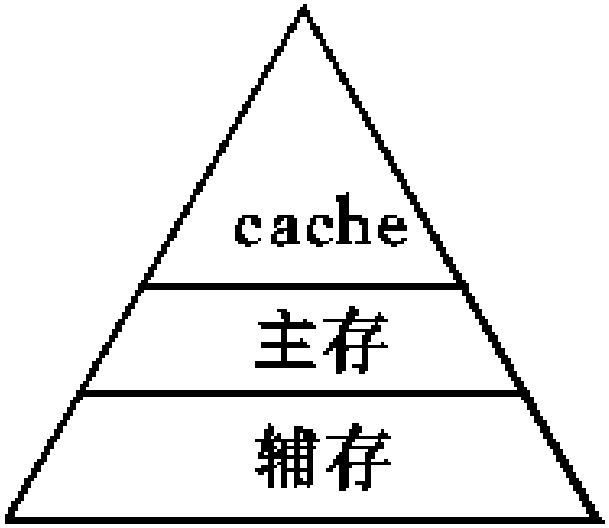
6.7 高速缓冲存储器

衡量存储器有三个指标： 容量、速度和价格 / 位。一般来讲，速度高的存储器，每位价格也高，因此容量不能太大。

主-辅存层次满足了存储器的大容量和低成本需求。设置高速缓冲存储器(cache)是解决存取速度的重要方法。

现代大多数计算机同时采用这种存储层次，cache-主存-辅存三级存储层次解决存储容量和速度问题。

6.7 高速缓冲存储器



三级存储

6.7高速缓冲存储器

6.7.1 Cache存储器工作原理

在一个较短的时间间隔内，CPU访问存储器的地址往往集中在存储器逻辑地址空间的很小范围内。

这种对局部范围的存储器地址频繁访问，而对此范围以外的地址则访问甚少的现象就称为程序访问的局部化性质。

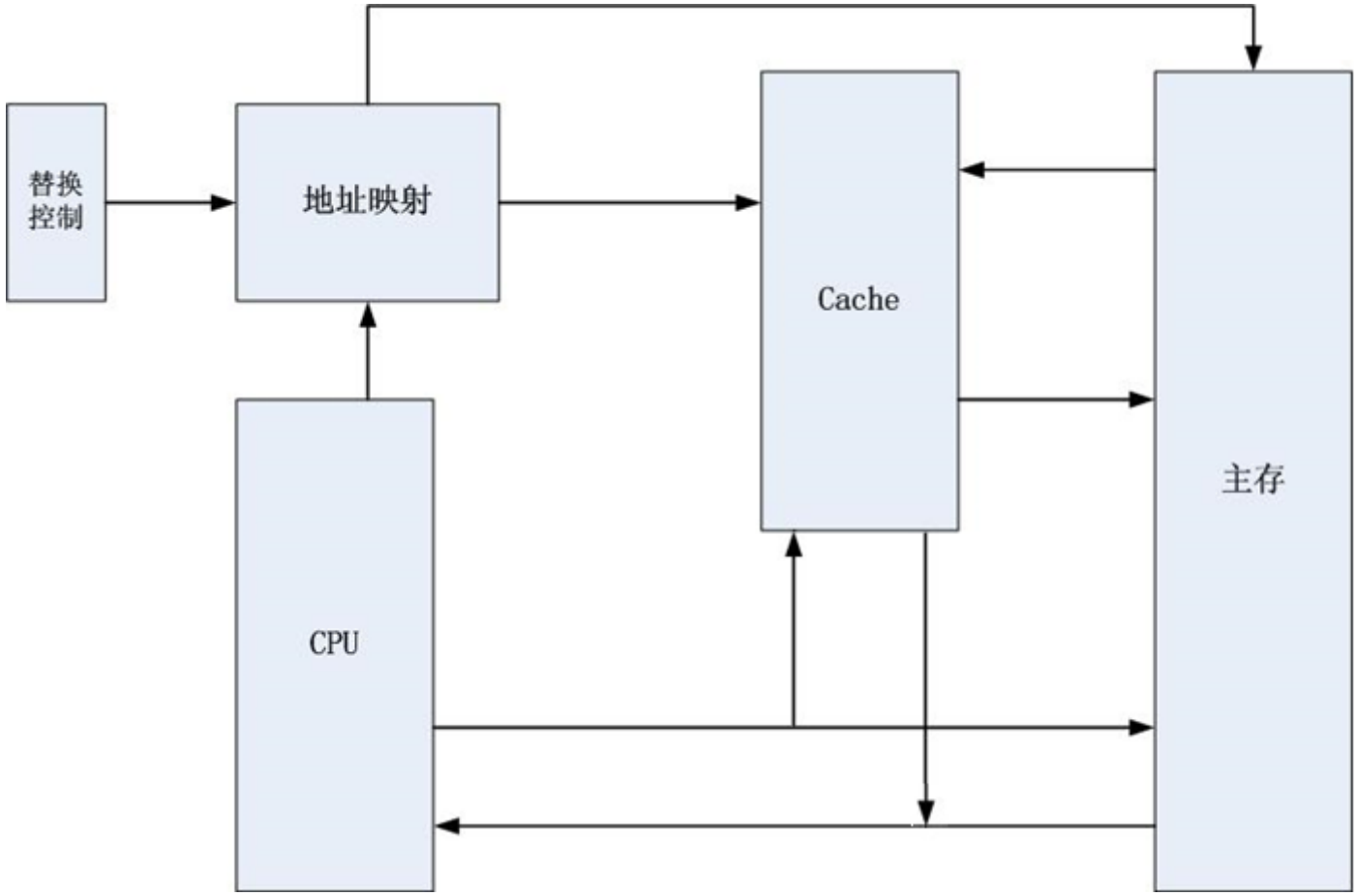
程序访问的局部化性质是实现Cache存储器的基本原理。

数据分布的这种集中倾向不如指令明显，但对数组的存储和访问以及工作单元的选择都可以使存储器地址相对集中。所以，对数组的访问局部化性质也较为明显。

Cache存储器介于CPU和主存之间，它的工作速度数倍于主存，全部功能由硬件实现，并且对程序员是透明的。

6. 7高速缓冲存储器

6.7.1 Cache存储器工作原理



Cache的容量和块的大小是影响Cache的效率的重要因素。通常用“命中率”来测量Cache的效率。

命中率=访问信息在Cache中的次数/访问Cache的总次数

一般来说，Cache的存储容量比主存的容量小得多，但不能太小，太小会使命中率太低；过大又会增加成本，而且当容量超过一定值后，命中率随容量的增加将不会有明显地增长。

1.Cache存储器的读操作

CPU发出读请求时，要根据产生的主存地址分为两种不同的情况：

① 需要的信息已在Cache中，直接访问Cache存储器。

② 需要的信息不在Cache中，就把该信息所在的整个字块从主存一次调入。

两种方案实现：

6. 7高速缓冲存储器

6.7.1 Cache存储器工作原理

- 等整个字块都进入Cache存储器后，再从中把需要的信息读出来送到CPU使用。
- 在送字块到Cache存储器的同时就把需要的信息送到CPU，让信息立即可使用。

后一种方法称为“直通取数”或“通过式加载”（Load-through），也称为“通过式读”（Read-through）。

百年同濟 TONGJI UNIVERSITY

2. Cache存储器的写操作

Cache存储器中保存的字块是主存中相应字块的一个副本。如果程序执行过程中要对该字块的某个单元进行写操作，就会遇到如何保持Cache与主存的一致性问题。

通常有两种写入方式：

① 一种方式是暂时只向Cache存储器写入，并用标志加以注明，直到经过修改的字块被从Cache中替换出来时才一次写入主存。

② 每次写入Cache存储器时也同时写入主存，使Cache和主存保持一致。

前一种方式称为标志交换(flag-swap)方式。只有写标志“置位”的字块才有必要最后从Cache存储器一次写回主存，所以又称其为“写回法”。这种方式写操作速度快，但在此之前，主存中的字块未经即时修改而失效。

后一种方式称为通过式写(write-through)，又称写直达法。向Cache存储器某一单元写入多少次，也要向主存相应单元写入多少次。这种方式实现简单，且能随时保持主存数据的正确性。但是，有可能要增加多次不必要的向主存的写入。

另有一种情况写操作是： 当被修改的单元根本就不在Cache存储器时，写操作直接对主存进行。

具有Cache的主存储器，其平均存取时间计算如下：

$$\text{平均存取时间} = h \times Tc + (1-h) \times (Tc + Tm)$$
$$= Tc + (1-h) \times Tm$$

h — 命中率

Tc — Cache的存取时间

Tm — 主存的存取时间

6.7.2 Cache存储器的地址映像

为了把信息放到Cache存储器中，必须应用某种函数把主存地址映像到Cache，称作地址映像。在信息按照这种映像关系装入Cache后，执行程序时，应将主存地址变换成Cache地址，这个变换过程叫做地址变换。

地址映像有三种方式：直接映像、全相联映像、组相联映像。

1.直接映像

假设主存储器空间被分为Mm(0)，Mm(1)，…，Mm(i)，…，Mm(2^m-1)共2^m个块，字块大小为2^b个字节；Cache存储空间被分为Mc(0)，Mc(1)，…，Mc(j)，…，Mc(2^c-1)共2^c个同样大小的块。

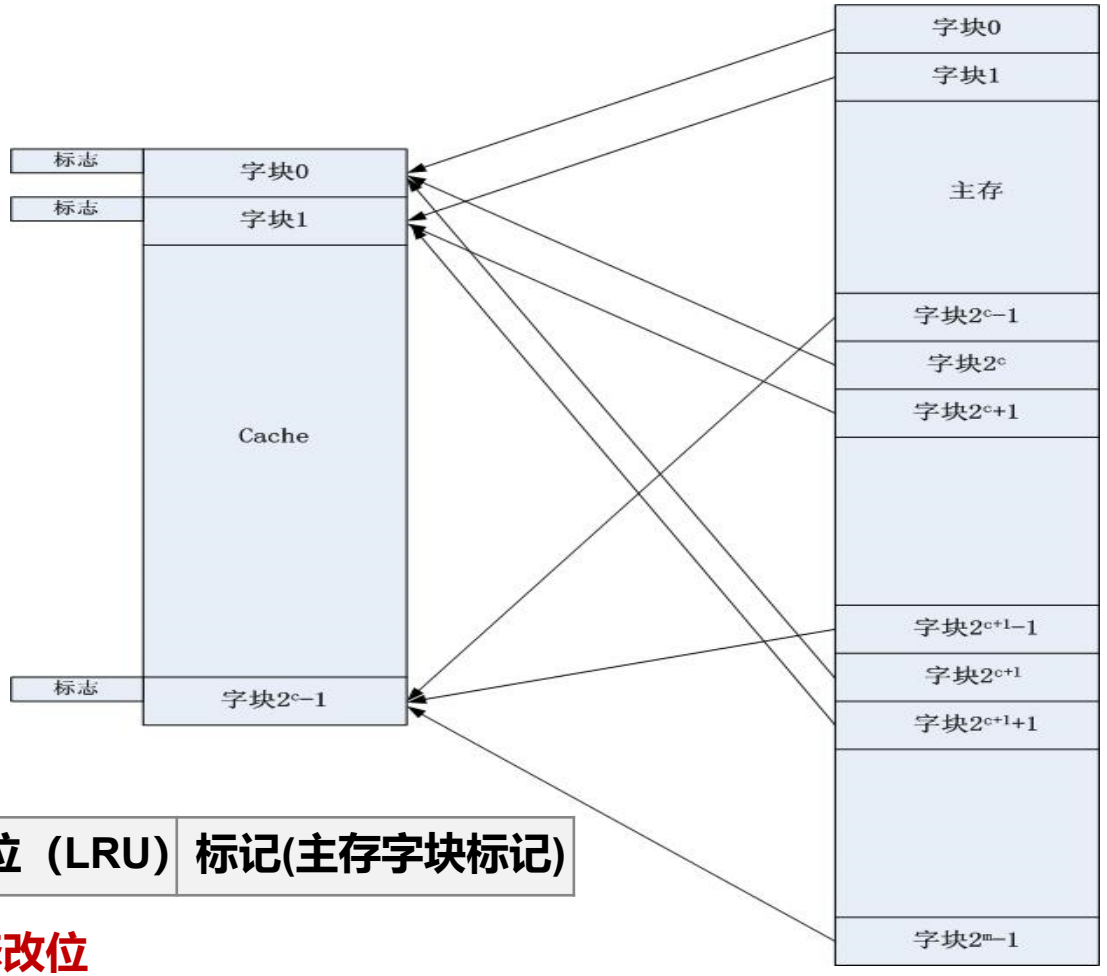
直接映像函数定义为：

$$j=i \mod 2^c$$

其中：j是Cache的字块号，i是主存的字块号

6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像



标志（标记项）包括:

有效位	脏位（修改位）	替换位（LRU）	标记(主存字块标记)

Write-through模式下不需要修改位

6.7高速缓冲存储器

6.7.2 Cache存储器的地址映像

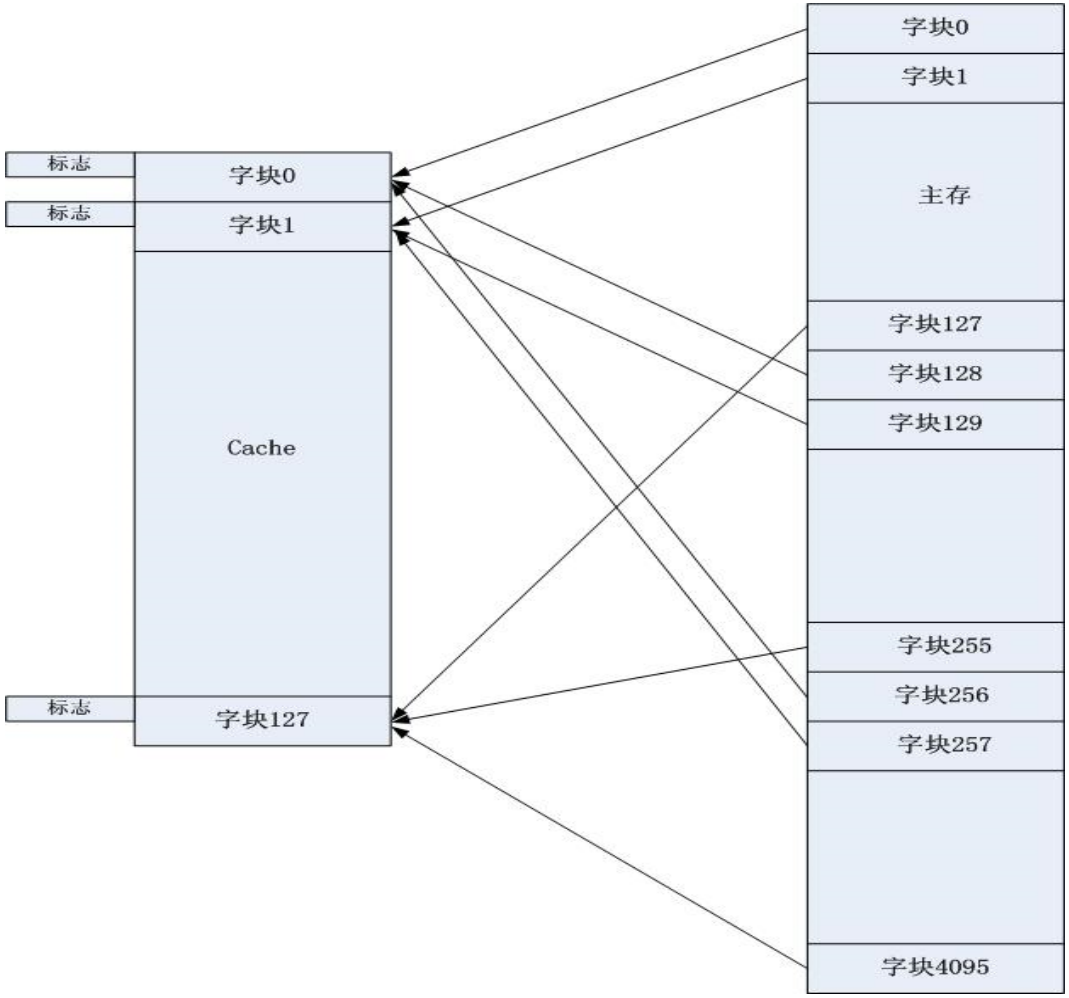
例：Cache为2048（2KB）字节，每块容量为16B，即Cache有128块组成。设主存为64KB,每块容量也为16B ,共4096块组成。

$$j=i \mod 2^7$$

主存各块映射到Cache如图所示：

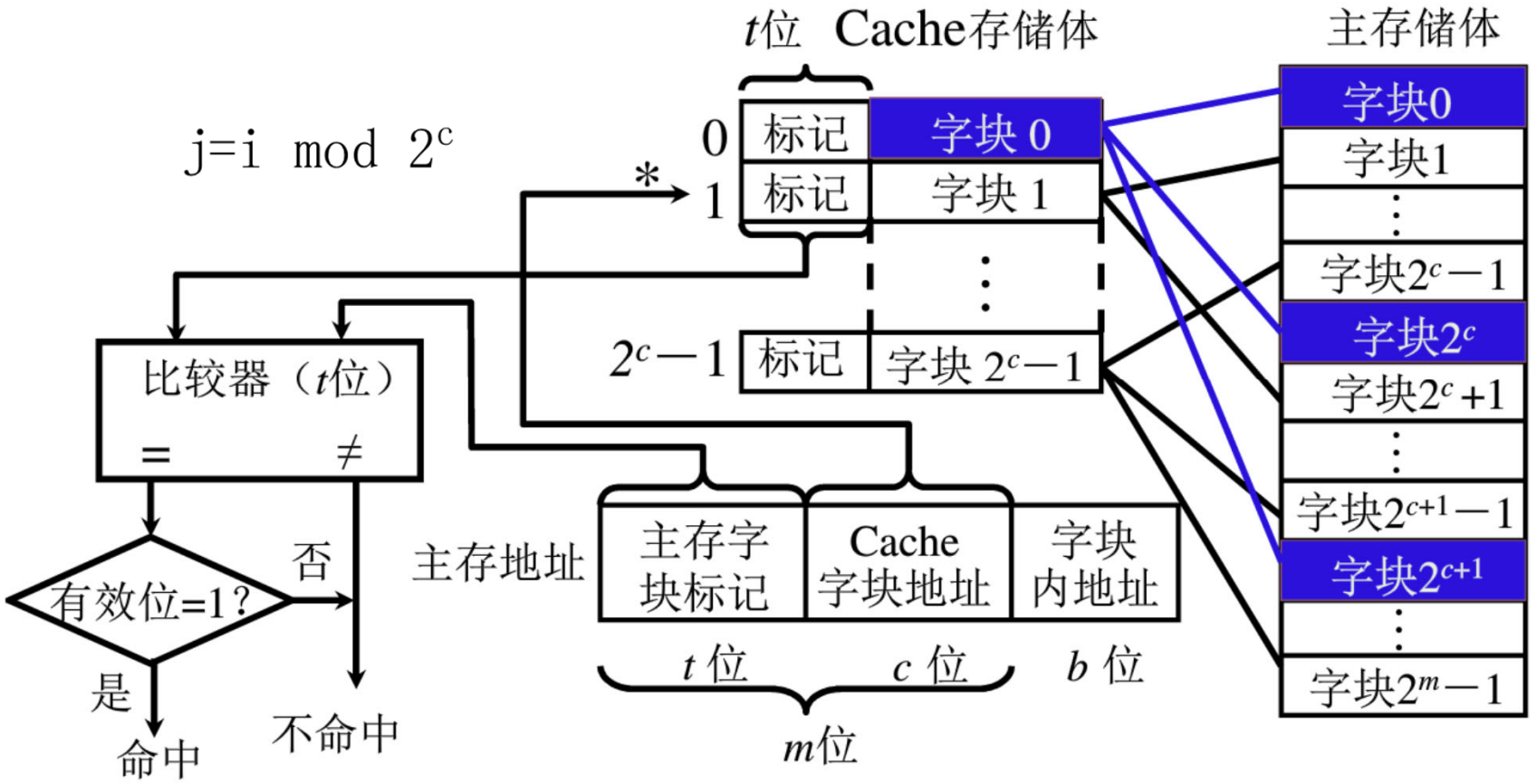
6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像



6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像



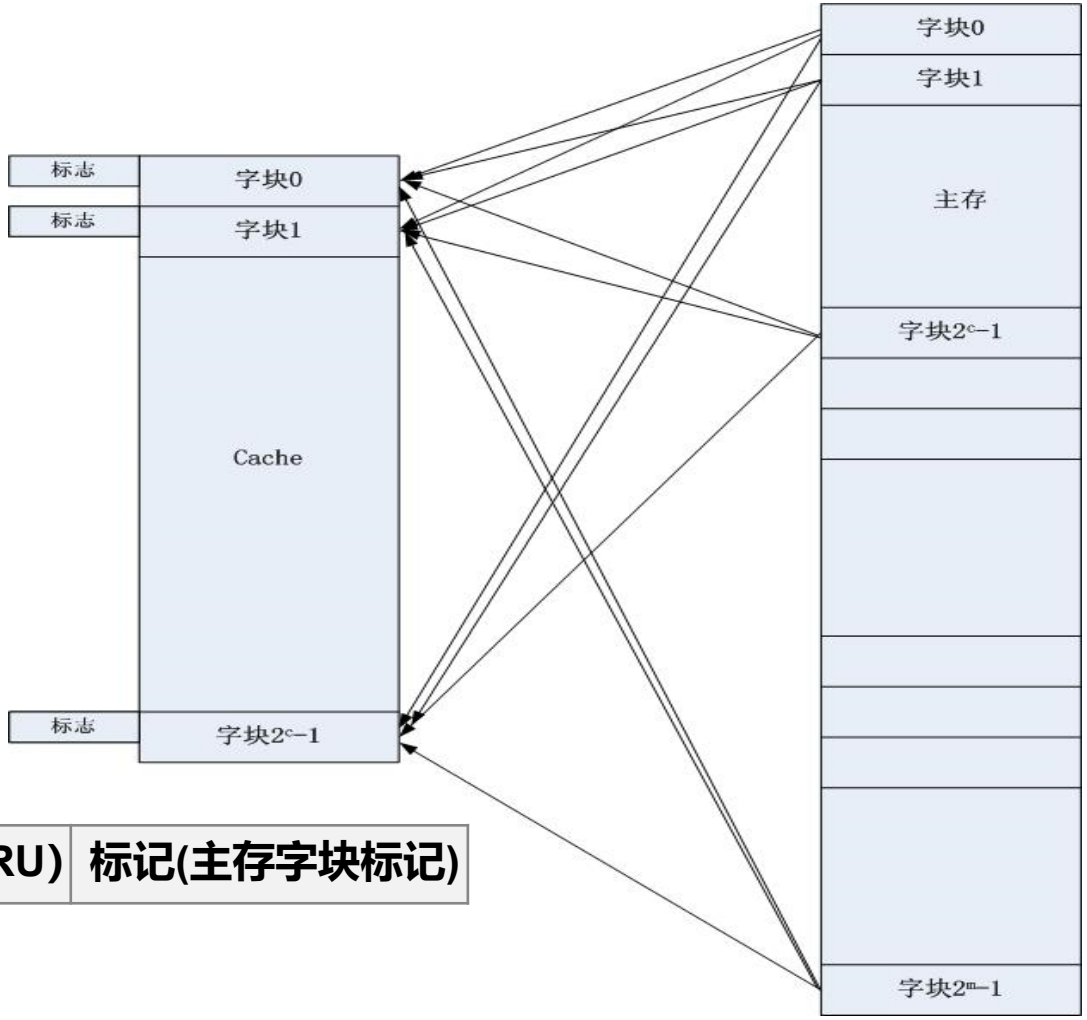
直接映像的优点是实现简单，只需利用主存地址按某些字段直接判断，即可确定所需字块是否已在Cache存储器中。

2.全相联映像

全相联映像方式是最灵活但成本最高的一种方式，如图所示。

6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像



标志（标记项）包括:

有效位	脏位（修改位）	替换位（LRU）	标记(主存字块标记)

Write-through模式下不需要修改位

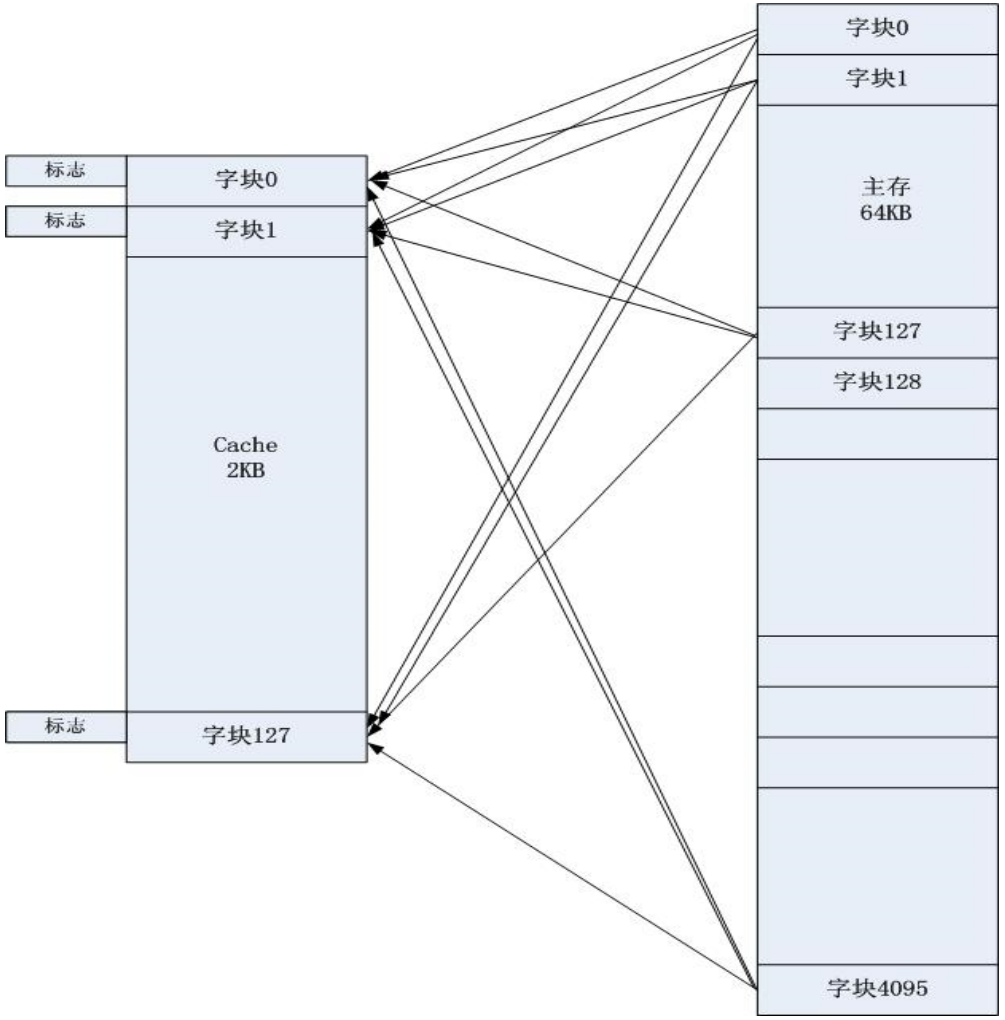
6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像

以前例为例，主存各块映射到Cache如图所示：

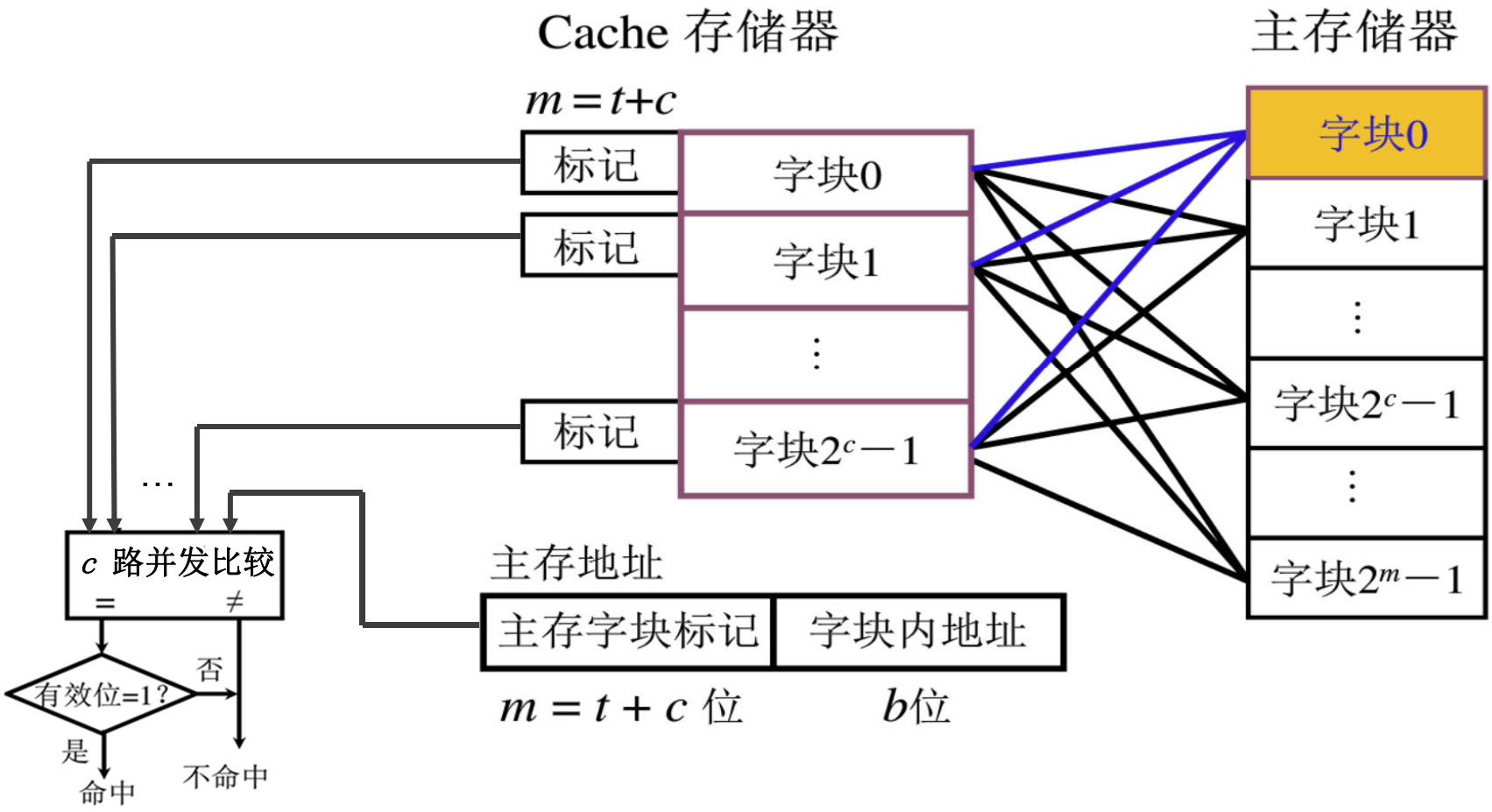
6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像



6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像

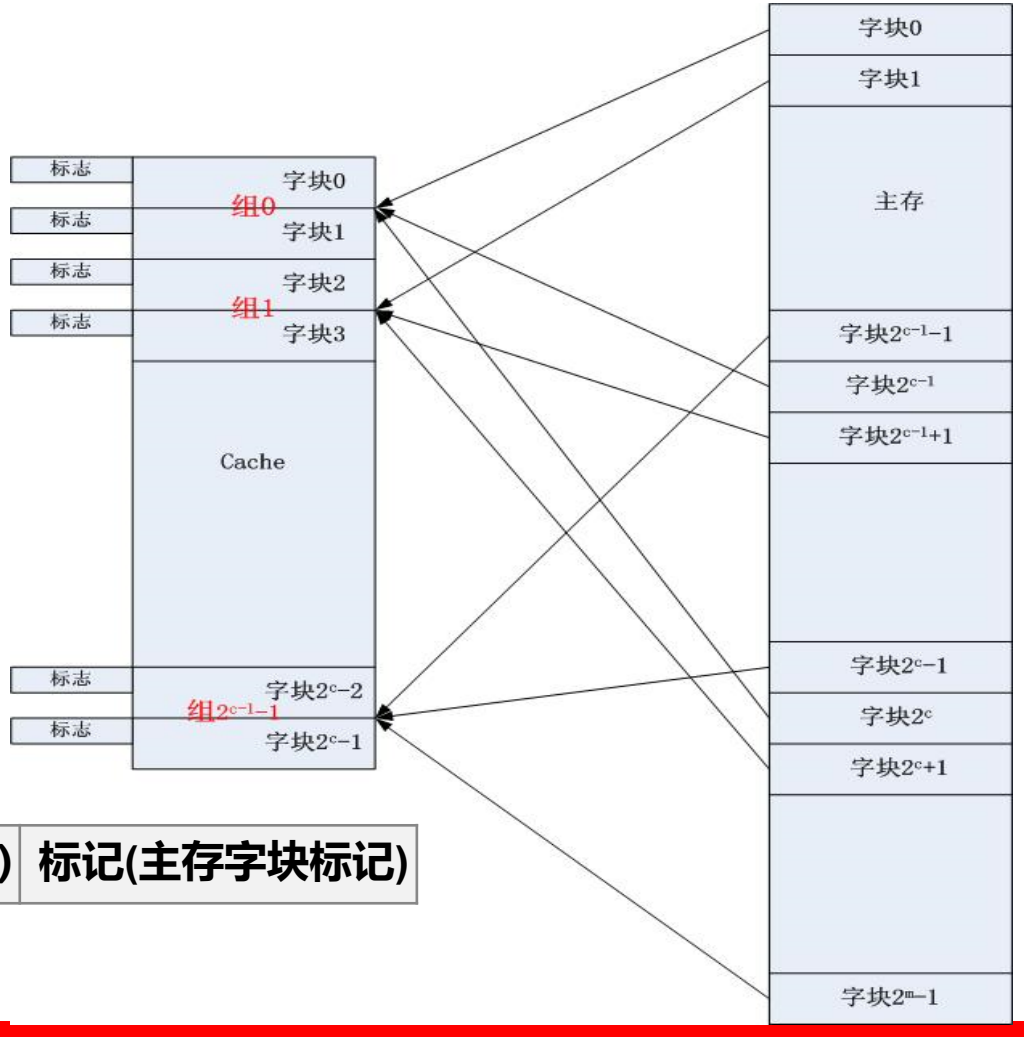


3.组相联映像

组相联映像方式是直接映像和全相联映像方式的一种折衷方案。组相联映像Cache组织如图所示。

6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像



标志（标记项）包括:

有效位	脏位（修改位）	替换位（LRU）	标记(主存字块标记)

Write-through模式下不需要修改位

6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像

以前为例，将Cache分成64组，每组2块，每块16B。同样块大小的主存有4096块。

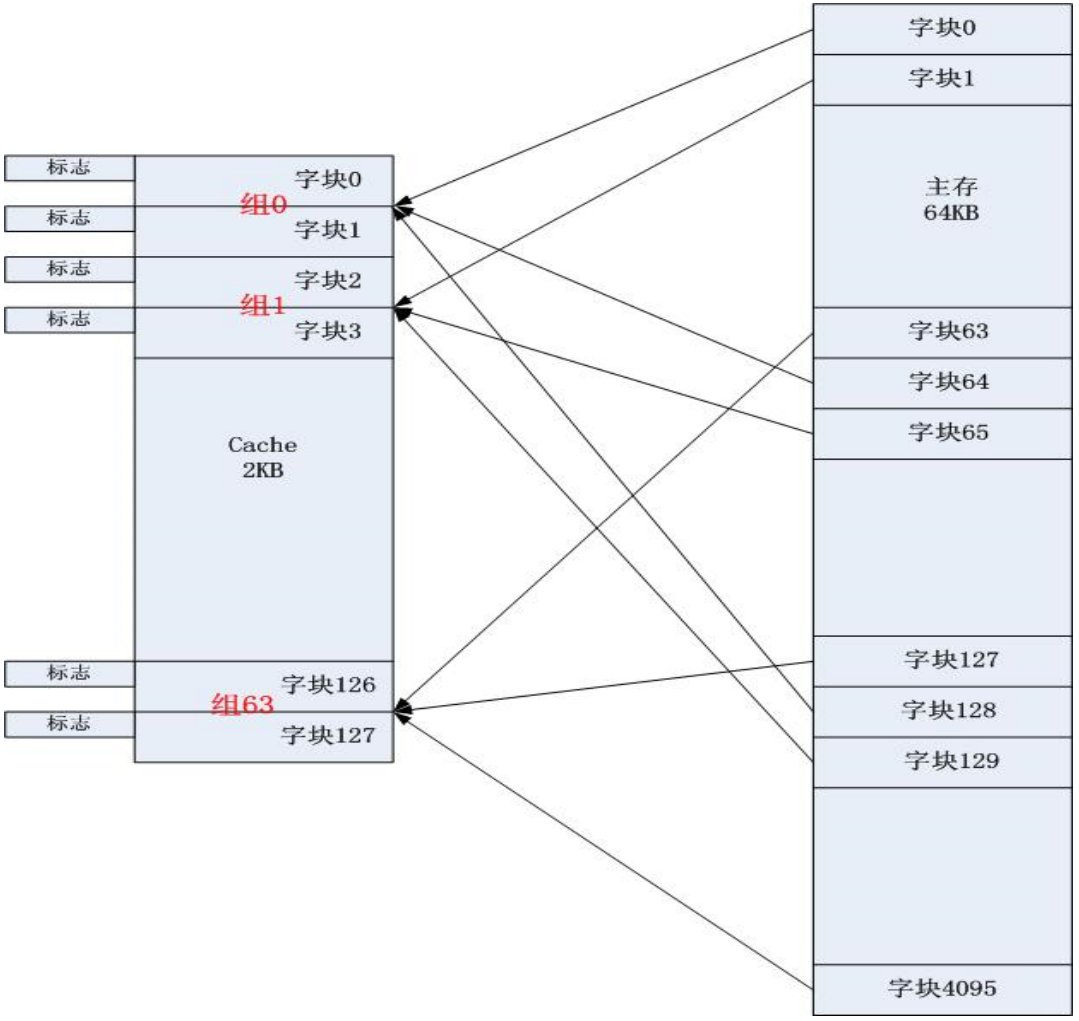
$$j=i \mod 2^6$$

其中：j是Cache的某组，i是主存的字块号

主存各块映射到Cache如图所示：

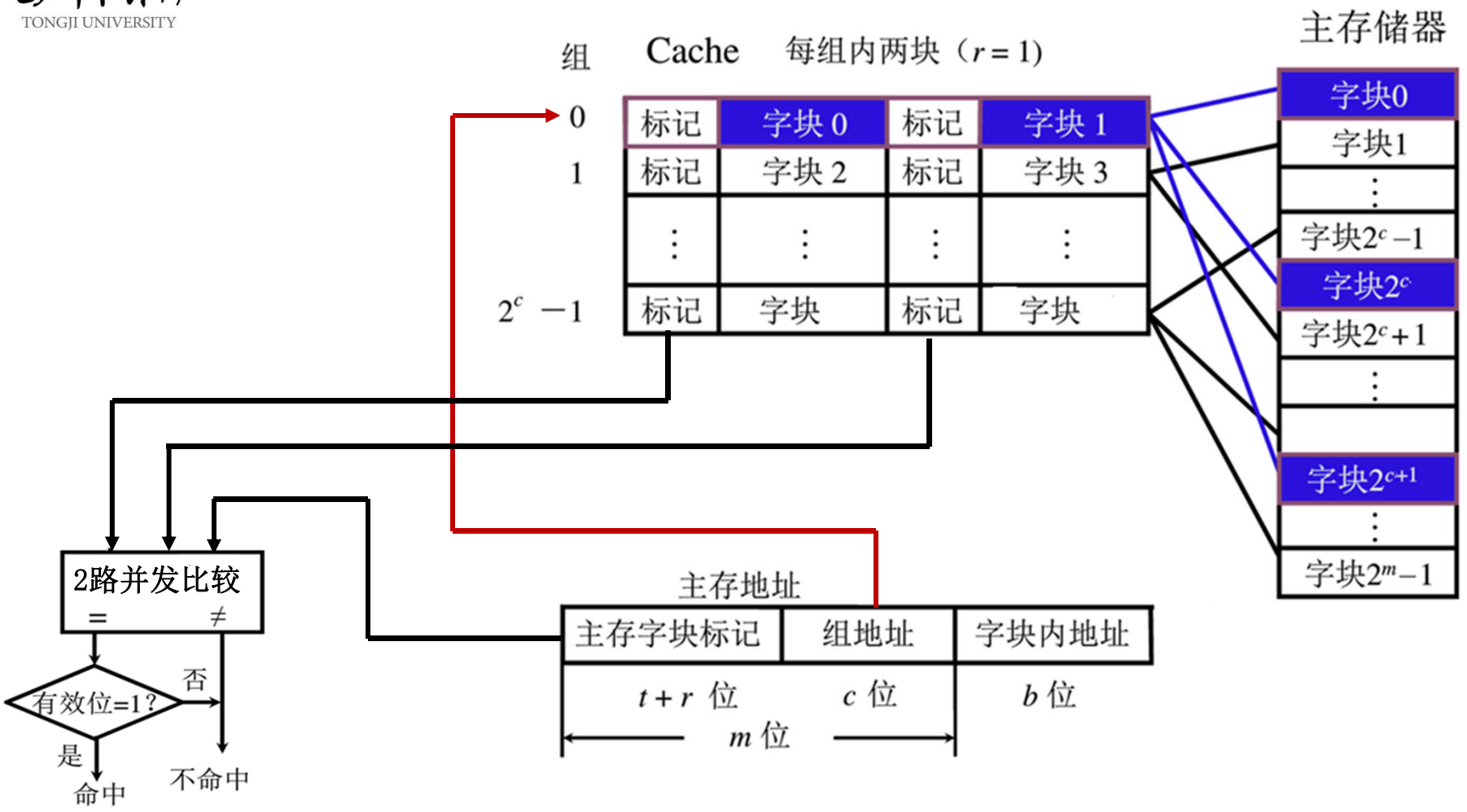
6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像



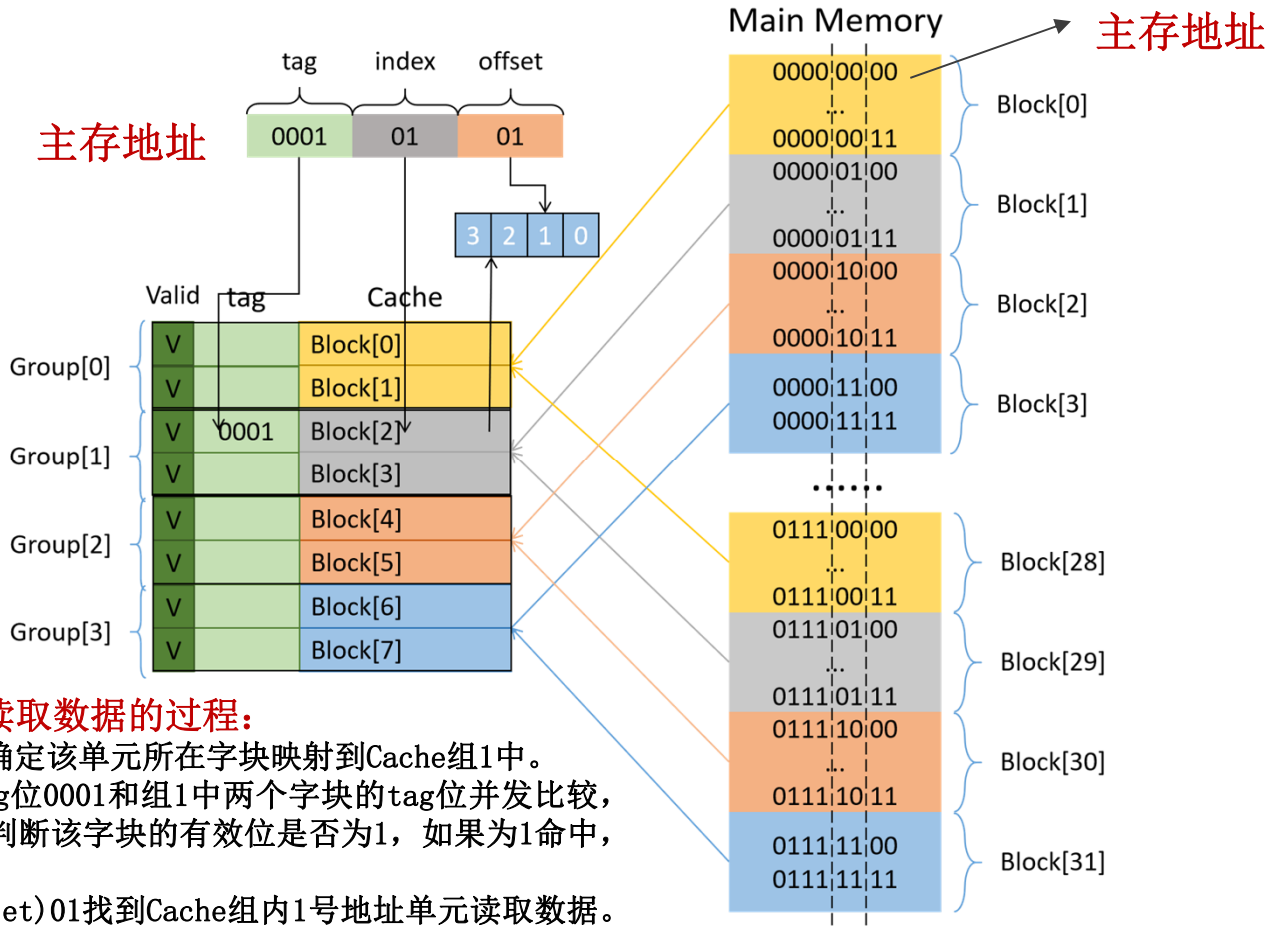
6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像



6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像



以访问00010101地址单元为例说明读取数据的过程：

- 按照访问单元地址中的组地址(Index)确定该单元所在字块映射到Cache组1中。
- 因为是2路组相联，所以将地址中的tag位0001和组1中两个字块的tag位并发比较，确定和组内一个字块的tag位相同，再判断该字块的有效位是否为1，如果为1命中，否则未命中。
- 若Cache命中，再根据字块内地址(offset)01找到Cache组内1号地址单元读取数据。若未命中，需将该单元所在字块5搬入Cache组1中的空闲字块，并修改该字块有效位为1，再根据字块内地址(offset)01找到Cache组内1号地址单元读取数据。

6. 7高速缓冲存储器

6.7.2 Cache存储器的地址映像

组相联映像方式的性能与复杂性介于直接映像与全相联映像两种方式之间。

当组数为2^c时，就是直接映像方式；

当组数为2^0时，就是全相联映像方式。

6.7.3 替换算法

当新的主存字块需要调入Cache存储器而它的可用位置又已被占满时，就产生替换算法问题。介绍两种替换算法：先进先出(FIFO)算法和近期最少使用(LRU)算法。

1. FIFO算法

FIFO算法总是把一组中最先调入Cache存储器的字块替换出去，它不需要随时记录各个字块的使用情况，所以实现容易，开销小。

6. 7高速缓冲存储器

6.7.3 替换算法

例：如有一程序，其块地址流为1、2、3、4、1、2、5、1、2、3、4、5，假定：这五个块都映射到同一组，在组相联映像方式中，每组为3块和每组为4块情况下的命中率。

1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	4	4	4	5	5	5	5	5	5
	2	2	2	1	1	1	1	1	3	3	3
		3	3	3	2	2	2	2	2	4	4
x	x	x	x	x	x	x	o	o	x	x	o

1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1	1	1	5	5	5	5	4	4
	2	2	2	2	2	2	1	1	1	1	5
		3	3	3	3	3	3	2	2	2	2
			4	4	4	4	4	4	3	3	3
x	x	x	x	o	o	x	x	x	x	x	x

2.LRU算法

LRU算法是把一组中近期最少使用的字块替换出去。这种替换算法需随时记录Cache存储器中各个字块的使用情况，以便确定那个字块是近期最少使用的字块。

LRU算法规则

Cache的每一块都附设有一个计数器，用于记录该块的累计不使用次数。

① 每当某一块被命中时，它的计数值就清零。而所有计数值低于它原来计数值的那些块计数值加1，其余块计数值不变。

② 当分组未满而调入新的字块时，新块的计算值为0，其余块计数值加1。

③ 当分组已满而调入新块时，具有计数值为最大的那个块就是被替换得字块，由计数值为0的新字块替代，而其余字块的计数值加1。

以前例为例，计算LRU的命中率。

6. 7高速缓冲存储器

6.7.3 替换算法

1	2	3	4	1	2	5	1	2	3	4	5
10	11	12	40	41	42	50	51	52	30	31	32
	20	21	22	10	11	12	10	11	12	40	41
		30	31	32	20	21	22	20	21	22	50
x	x	x	x	x	x	x	o	o	x	x	x

1	2	3	4	1	2	5	1	2	3	4	5
10	11	12	13	10	11	12	10	11	12	13	50
	20	21	22	23	20	21	22	20	21	22	23
		30	31	32	33	50	51	52	53	40	41
			40	41	42	43	43	43	30	31	32
x	x	x	x	o	o	x	o	o	x	x	x

6.7高速缓冲存储器

6.7.3 替换算法

如：假设：地址流为1、2、3、4、1、2、3、4、……，而容量为3字块/组，那么无论是FIFO还是LRU算法命中率都为0。这种现象称为颠簸。

6.8 虚拟存储器

6.8.1 虚拟存储器概述

虚拟存储器指的是“主存—辅存”层次，使得程序员可以按比主存大得多的空间来编制程序，即按虚存空间编址。当然，主存实际容量的大小是会影响到系统的工作效率，如果程序过大而主存容量过小，则程序运行速度会明显下降。

1. 主存—辅存层次与cache—主存层次的比较

（1）主存 / cache存储器的访问“时间比”较小，而辅存 / 主存的访问“时间比”就要大得多；

（2）主存 / cache每次传送的基本信息单元(字块)比较小。辅存 / 主存每次传送的基本信息单元(段或页面)很大。

6.8 虚拟存储器

6.8.1 虚拟存储器概述

从原理角度看，主存—辅存层次和cache—主存层次有很多相似之处。它们采用的地址变换及映像方法和替换策略，从原理上看是相同的。

虚拟存储系统所采取的映像方式同样有全相联映像、组相联映像和直接映像等，替换算法也多采用LRU算法。实际上，这些替换算法和地址映像方式最早应用于虚拟存储系统中，后来才发展到cache系统中。

2. 主存—辅存层次信息传送单位和存储管理

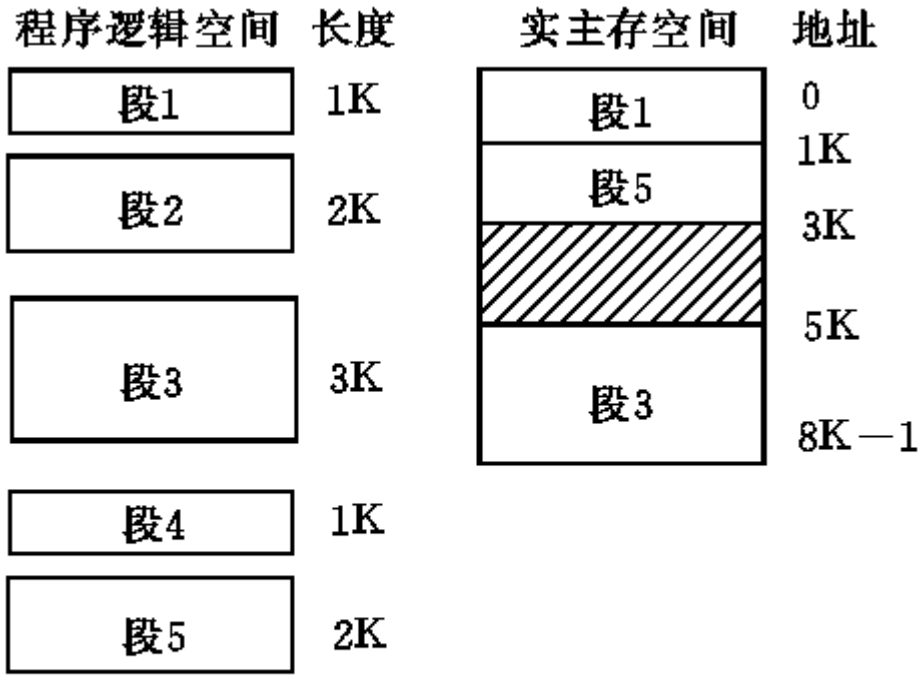
主存—辅存层次的信息传送单位可采用几种不同的方案： 段、页或段页。

（1）段存储管理

段是利用程序的模块化性质，按照程序的逻辑结构划分成的多个相对独立部分。段作为独立的逻辑单位可以被其他程序段调用，这样就形成段间连接，产生规模较大的程序。

6.8 虚拟存储器

6.8.1 虚拟存储器概述



(a)

段表

段号	段起点	装入位	段长
1	0	1	1K
2		0	
3	5K	1	3K
4		0	
5	1K	1	2K

(b)

主存按段分配的存储管理方式称为段式管理。其优点是：

1、段的分界与程序的自然分界相对应；

2、段的逻辑独立性使它易于编译、管理、修改和保护，也便于多道程序共享。

缺点是：

容易在段间留下许多空余的零碎存储空间不好利用，造成浪费。

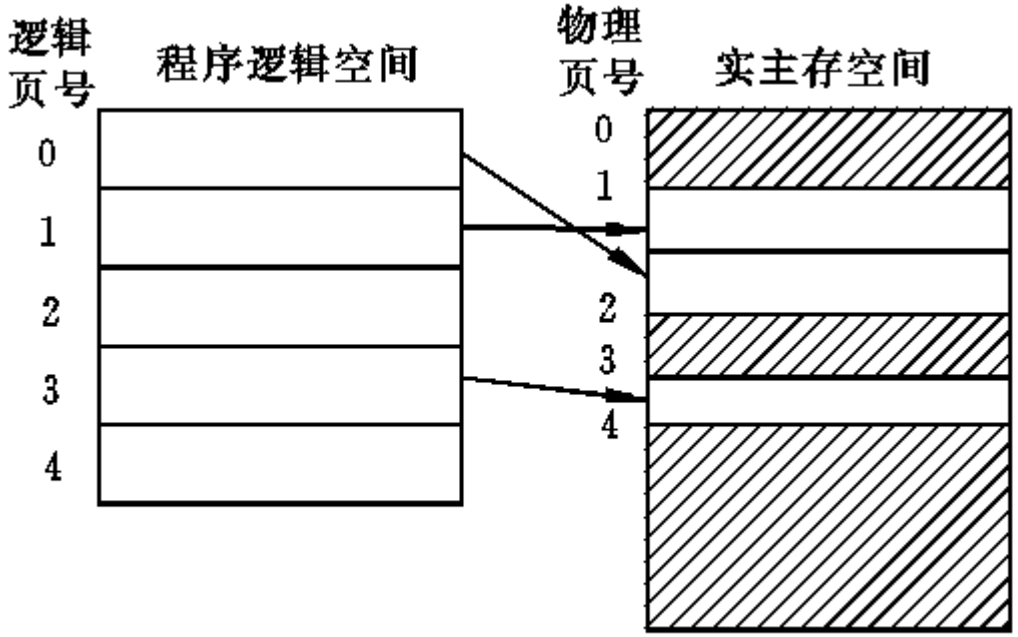
（2）页存储管理

页式管理系统的信息传送单位是定长的页，主存的物理空间也被划分为等长的固定区域，称为页面。新页调入主存只要有空白页面就可。可能造成浪费的是程序最后一页的零头，它比段式管理系统的空间浪费要小得多。

页式管理系统的缺点正好和段式管理系统相反，由于页不是逻辑上独立的实体，所以处理、保护和共享都不及段式来得方便。

6.8 虚拟存储器

6.8.1 虚拟存储器概述



(a)

页表

逻辑页号	实主存页号	装入位
0	2	1
1	1	1
2		0
3	4	1
4		0

(b)

6. 8 虚拟存储器

6.8.1 虚拟存储器概述

段式和页式存储管理各有其优缺点，可以采用段和页结合的段页式存储管理系统。程序按模块分段，段内再分页，出入主存仍以页为信息传送单位，用段表和页表(每段一个页表)进行两级管理。

6.8 虚拟存储器

6.8.2 页式虚拟存储器

在页式虚拟存储系统中，把虚拟空间分成页，主存空间也分成同样大小的页，称为实页或物理页，而把前者称为虚页或逻辑页。假设虚页号为0，1，2，…，M，实页号为0，1，…，L，显然有M>L。

由于页的大小都取2的整数幂个字，所以，页的起点都落在低位字段为零的地址上。可把虚拟地址分为两个字段，高位字段为虚页号，低位字段为页内字地址。

6.8 虚拟存储器

6.8.2 页式虚拟存储器

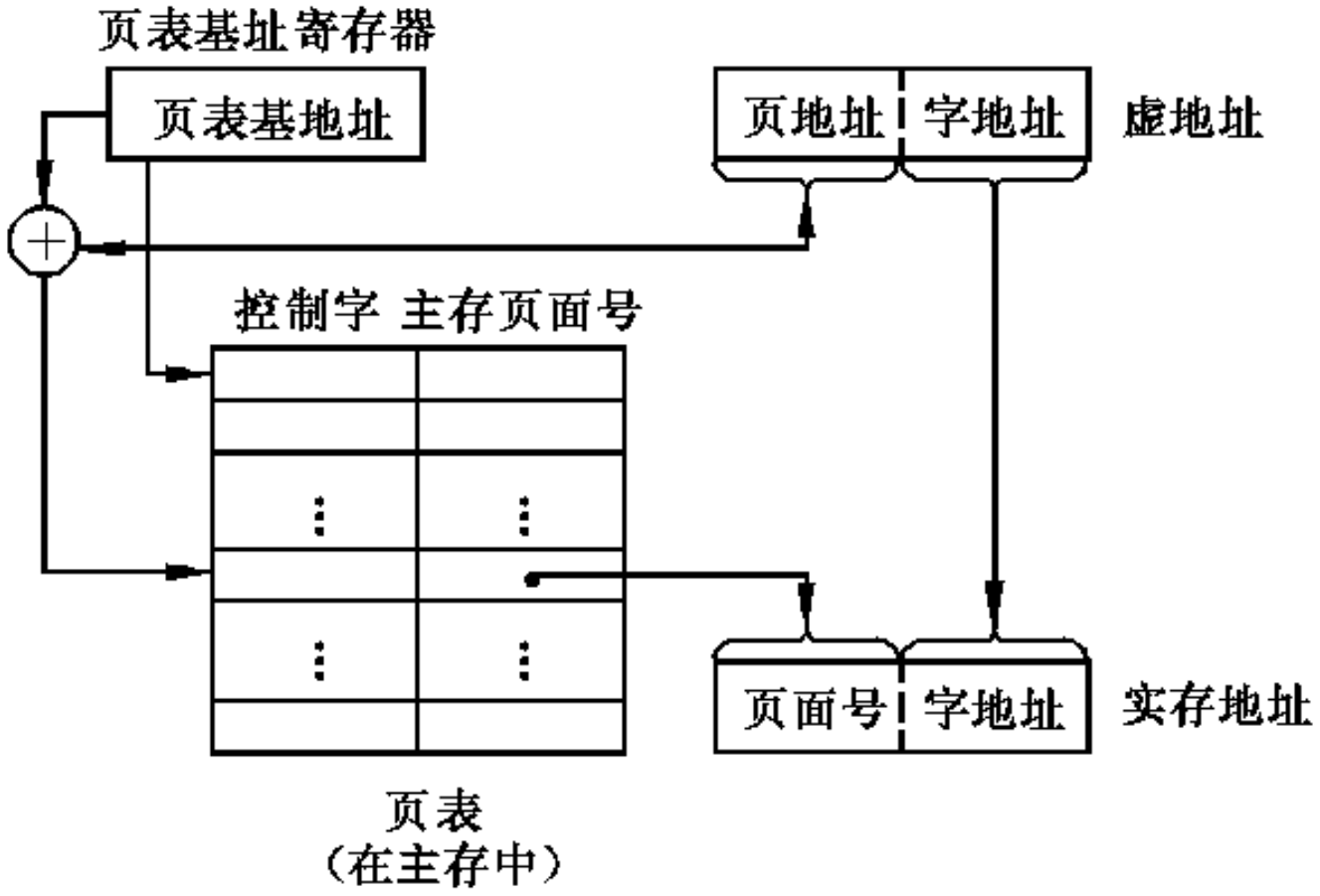
虚拟地址到主存实地址的变换是由页表来实现的。

在页表中，对应每一个虚存页号有一个表目。表目内容至少要包含该虚页所在的主存页面地址(页面号)，用它作为实(主)存地址的高字段，与虚拟地址的字地址字段相拼接，就产生完整的实主存地址，据此访问主存。

页式管理的地址变换如图所示。

6. 8 虚拟存储器

6.8.2 页式虚拟存储器



控制字：由装入位(有效位)、修改位、替换控制位及其他保护位等组成。

装入位：为“1”表示该虚页已从辅存调入主存；如装入位为“0”，则表示对应的虚页尚未调入主存。

修改位：指出主存页面中的内容是否被修改过，替换时是否要写回辅存。

替换控制位：指出需替换的页。

6.8 虚拟存储器

6.8.2 页式虚拟存储器

假设页表是保存在(或已调入)主存储器中，那么，在访问存储器时，首先要查页表。

- 即使页面命中，也得先访问一次主存去查页表，再访问主存才能取得数据，这就相当于主存速度降低了一倍。
- 如果页面失效，要进行页面替换，页面修改，访问主存次数就更多了。

6.8 虚拟存储器

6.8.2 页式虚拟存储器

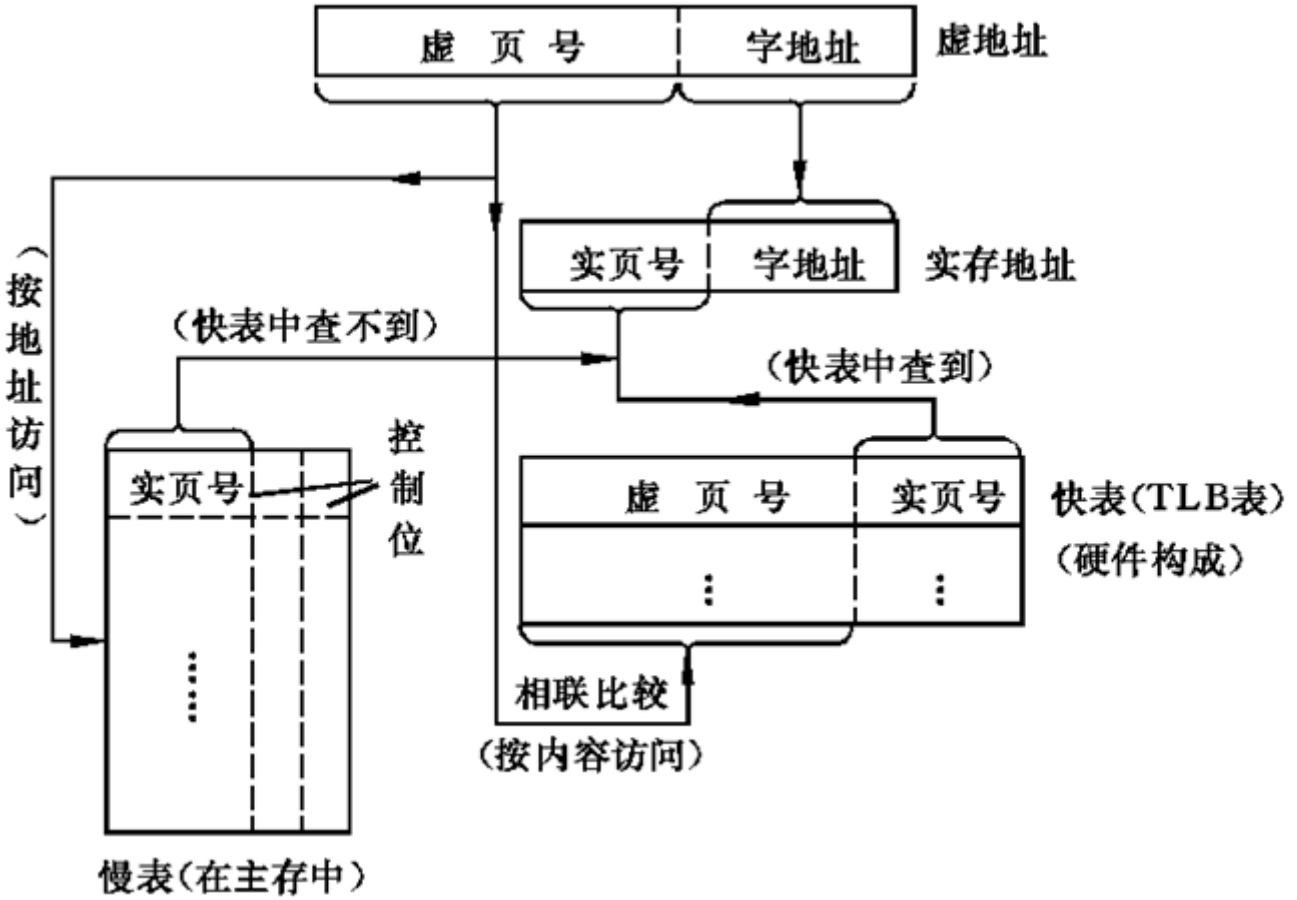
把页表的最活跃部分存放在快速存储器中组成快表，这是减少时间开销的一种方法。

此外，在一些影响工作速度的关键部分引入了硬件支持。例如，采用按内容查找的相联存储器并行查找，也是可供选择的技术途径。

一种经快表与慢表实现内部地址变换的方式如图所示。

6.8 虚拟存储器

6.8.2 页式虚拟存储器



6.8 虚拟存储器

6.8.3 段页式虚拟存储器

- 在段页式虚拟存储器中，把程序按逻辑结构分段以后，再把每段分成固定大小的页。
- 程序对主存的调入调出是按页面进行的，但它又可以按段实现共享和保护，它可以兼取页式和段式系统的优点。它的缺点是在地址映像过程中需要多次查表。

6.8.3 段页式虚拟存储器

- 在段页式虚拟存储系统中，虚拟地址转换成物理地址是通过一个段表和一组页表来进行定位的。
- 段表中的每个表目对应一个段，每个表目有一个指向该段的页表的起始地址(页号)及该段的控制保护信息。
- 由页表指明该段各页在主存中的位置以及是否已装入、已修改等标志。

6.8 虚拟存储器

6.8.3 段页式虚拟存储器

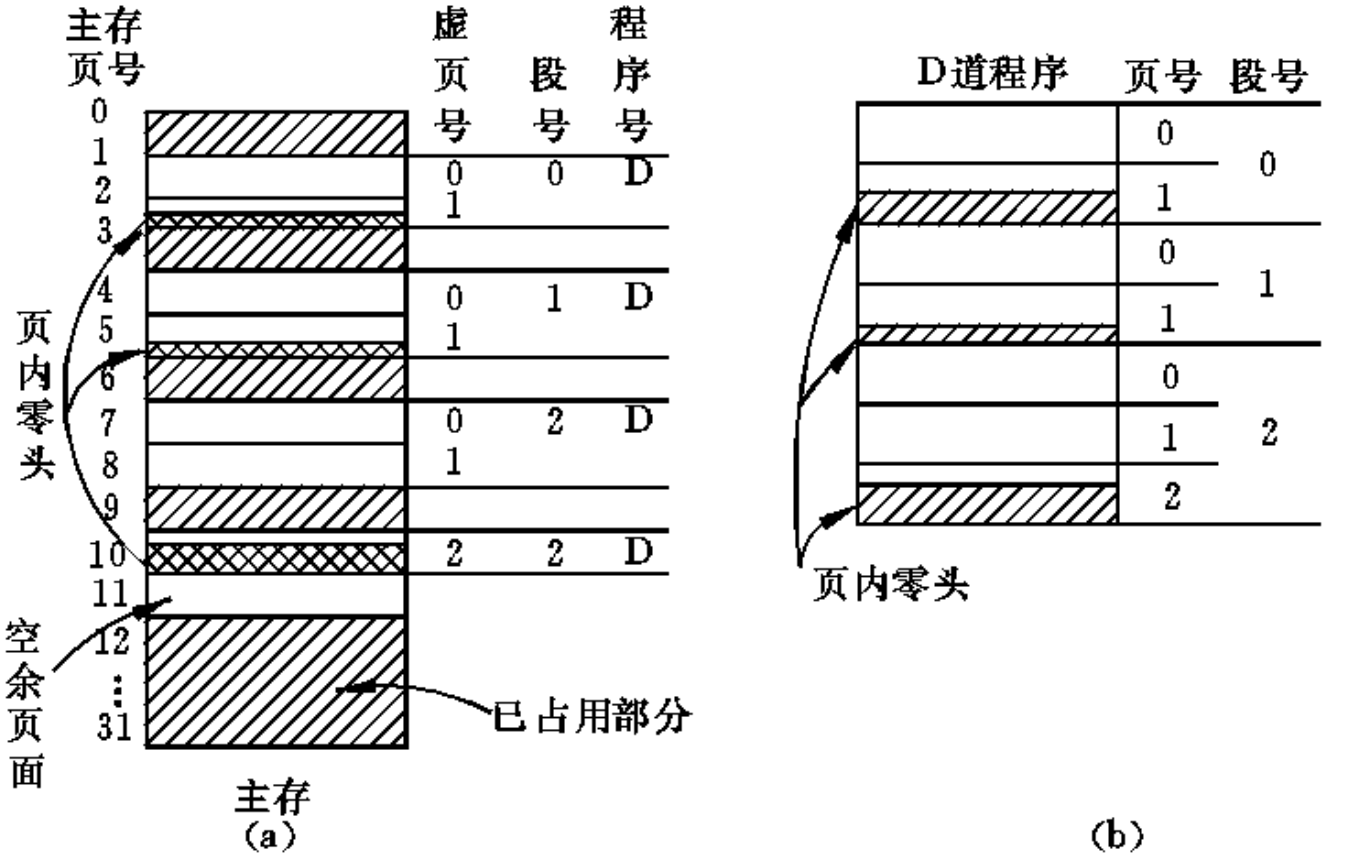
如果有多个用户在机器上运行，称为多道程序，多道程序的每一道(每个用户)需要一个基号(用户标志号)，可由它指明该道程序的段表起点(存放在基址寄存器中)。这样，虚拟地址应包括基号D、段号S、页号P、页内地址d。格式如下：

基号D	段号S	页号P	页内地址d

现举例说明段页式地址变换过程。如图所示。

6.8 虚拟存储器

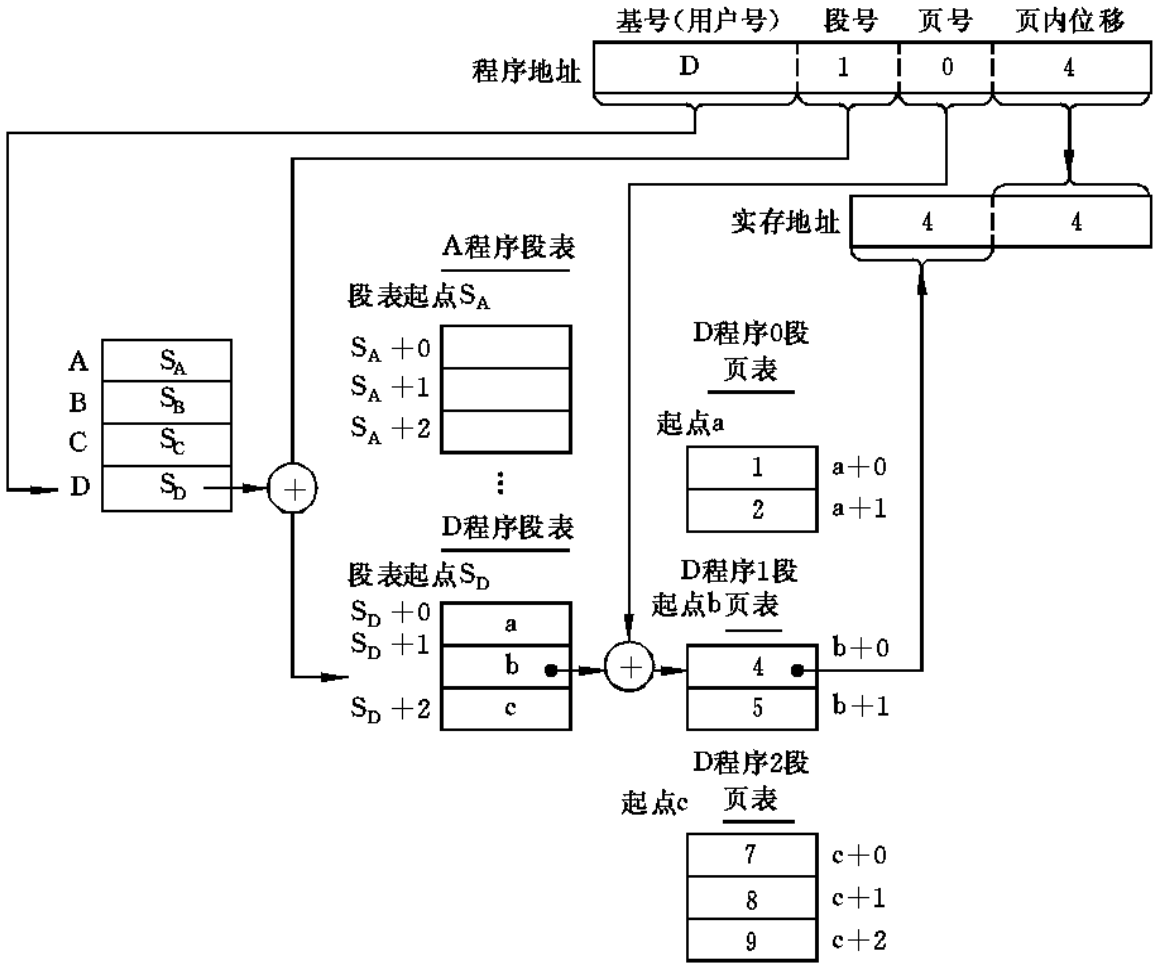
6.8.3 段页式虚拟存储器



当要访问的程序地址为D道1段0页4单元时，其地址变换过程如下图所示 。

6. 8 虚拟存储器

6.8.3 段页式虚拟存储器



6.8.4 虚拟存储器工作的全过程

对虚拟存储器来说，程序员按虚存储空间编制程序，在直接寻址方式下由机器指令的地址码给出地址。这个地址码就是虚地址，可由虚页号及页内地址组成，如下所示：

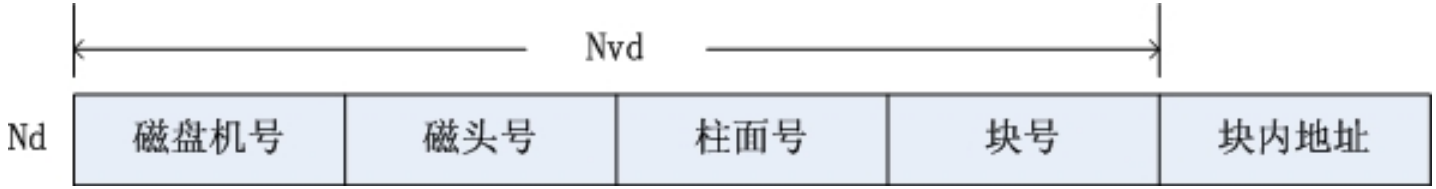
虚地址	虚页号Nv	页内地址Nr

这个虚地址实际上不是辅存的实地址，而是辅存的逻辑地址。

6.8 虚拟存储器

6.8.4 虚拟存储器工作的全过程

因此，在虚拟存储器中还应有虚拟地址到辅存实地址的变换。辅存一般按信息块编址，而不是按字编址，若使一个块的大小等于一个虚页面的大小，这样就只需把虚页号变换到Nvd即可完成虚地址到辅存实地址的变换。

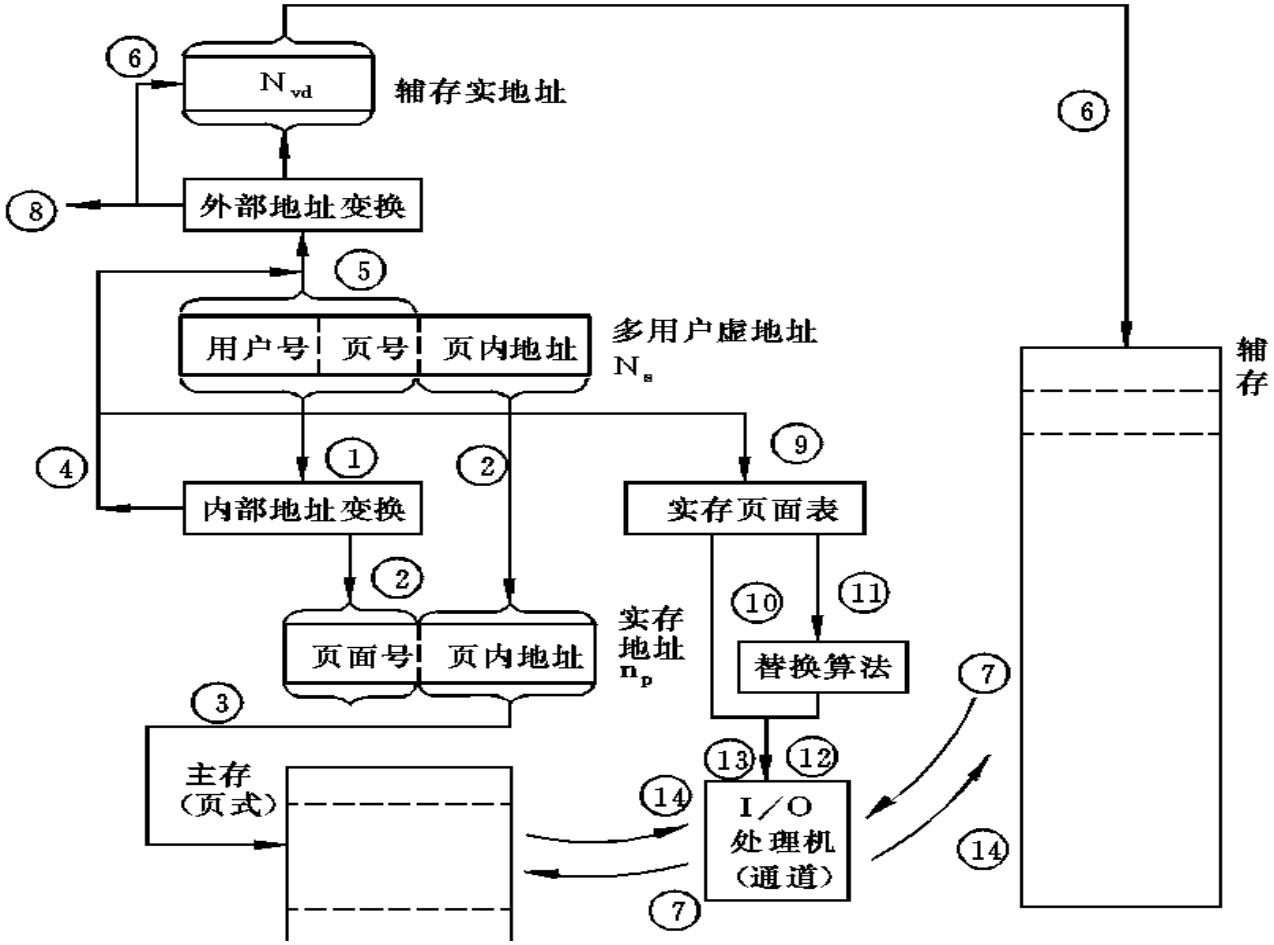


为此，可采用页表的方式，把由Nv变换成Nvd的表称为外页表，而把Nv变换到主存页号的表称为内页表。

虚拟存储器的工作全过程如图所示。

6.8 虚拟存储器

6.8.4 虚拟存储器工作的全过程



6.8 虚拟存储器

6.8.5 Pentium处理机的虚拟存储器

Pentium存储器的地址转换过程(从虚拟地址到物理地址)如图所示。

Pentium存储器结构有很大灵活性，根据其段表和页表是否设置可以有4种组合情况。

(1) 无段表和无页表的存储器。非虚拟存储器其逻辑地址即为物理地址，可减少复杂性，在高性能的控制机中经常被采用。

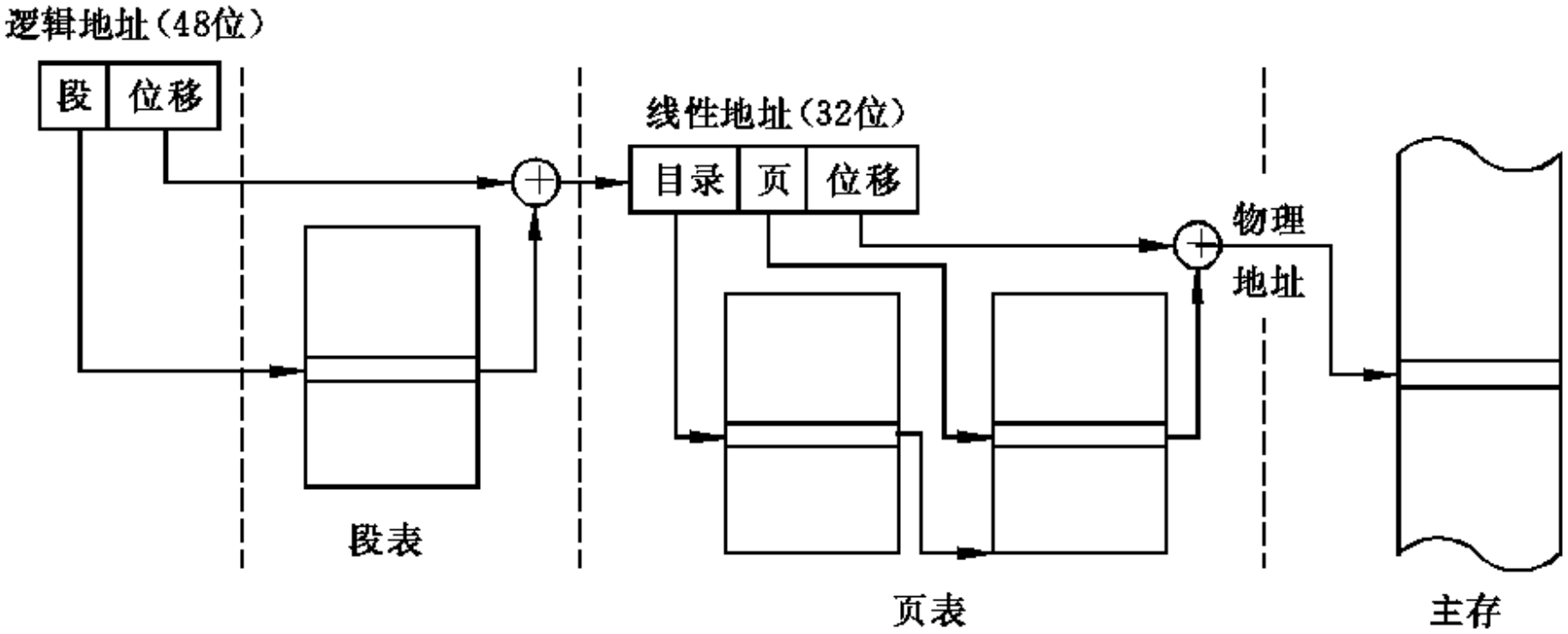
(2) 无段表和有页表的存储器。页式虚拟存储器，此时存储器的管理和保护是通过页面转换实现的。

(3) 有段表和无页表的存储器。段式虚拟存储器。

(4) 有段表和有页表的存储器。段页式虚拟存储器。

6. 8 虚拟存储器

6.8.5 Pentium处理机的虚拟存储器



习 题（2）

习题：

P179 3，5，6，7,10,12