**第一章 计算机系统概论**

一、计算机系统简介

1、计算机的软硬件概念

计算机系统：

（1）硬件：计算机的实体，如主机、外设等。

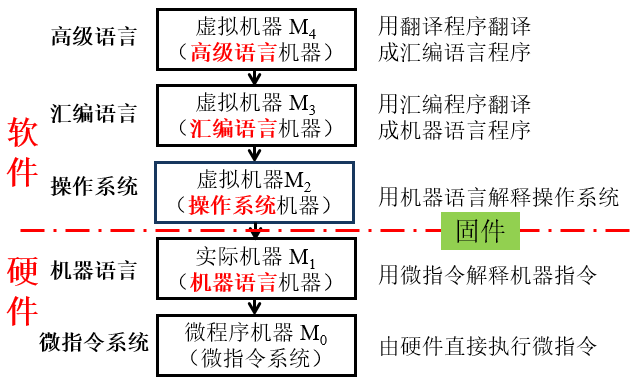
（2）软件：由具有各类特殊功能的信息（程序）组成

①系统软件：用来管理整个计算机系统

语言处理程序、操作系统、服务性程序、数据库管理系统、网络软件

②应用软件：按任务需要编制成的各种程序

2、计算机系统的层次结构



汇编程序（Assembler）：将汇编语言程序翻译成对应的目标程序。

编译程序（Compiler）：将高级语言程序翻译成对应的目标程序。

链接程序（Linker）：将一个或多个目标程序与一个或多个相关的程序库（Library）组织在一起，产生可执行代码，并存入外存，在需要运行时再由操作系统加载后执行。

解释程序（Interpreter）：对高级语言程序逐句地进行翻译，产生对应的机器语言指令序列并执行之。

计算机唯一能直接执行的语言是机器语言。

二、计算机的基本组成

1、冯·诺依曼计算机的特点

冯·诺依曼计算机以“存储程序”为基础，特点为：

（1）五大组成部件：运算器、存储器、控制器、输入设备、输出设备；

（2）指令和数据以同等地位存于存储器，可按地址寻访；

（3）指令和数据用二进制表示；

（4）指令由操作码和地址码组成；

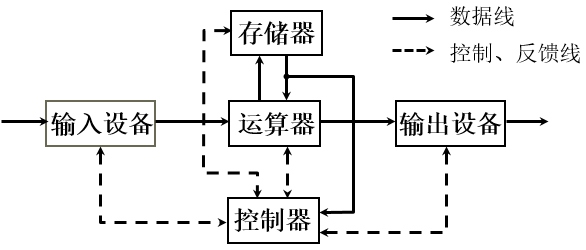
（5）指令在存储器内按顺序存放（存储程序）；

（6）以运算器为中心。

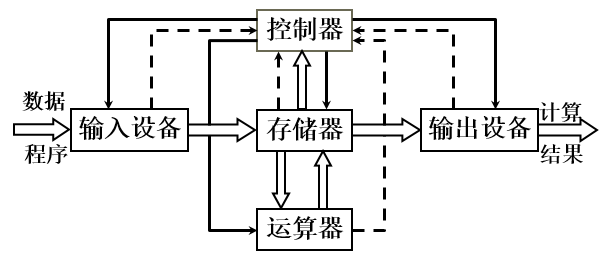
世界上第一台冯·诺伊曼结构的计算机EDVAC是1949年诞生的。

2、**计算机硬件框图**

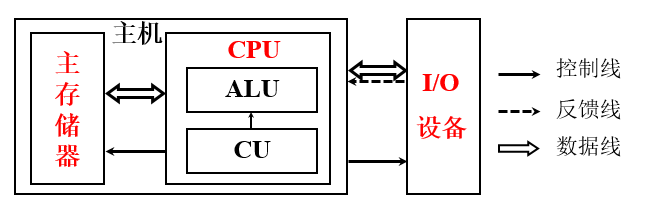
（1）典型冯·诺依曼计算机以**运算器**为中心（改进前）：



改进后：以**存储器**为中心



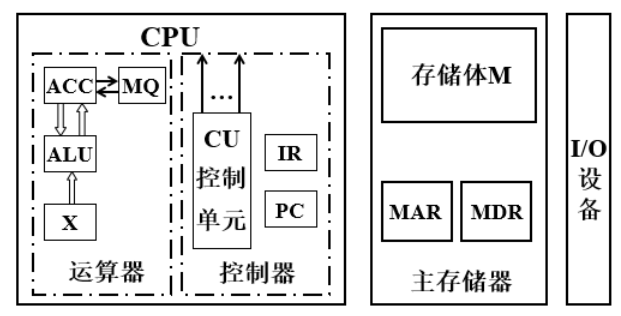
（2）现代计算机结构框图：现代计算机以存储器为中心



如何区分从存储器中取出的是指令还是数据？

计算机硬件主要通过不同的时间段来区分指令和数据，即：取指周期（或取指微程序）取出的既为指令，执行周期（或相应微程序）取出的既为数据。  
 另外也可通过地址来源区分，从PC指出的存储单元取出的是指令，由指令地址码部分提供操作数地址。

3、计算机的工作步骤



（1）存储器的基本组成

MAR：存储器地址寄存器，反映存储单元的个数；

MDR：存储器数据寄存器，反映存储字长。

存储单元：存放一串二进制代码/存放一个存储字的所有存储元集合

存储字：存储单元中二进制代码的组合

存储字长：存储单元中二进制代码的位数，往往是8字节的整数倍

每个存储单元赋予一个地址号

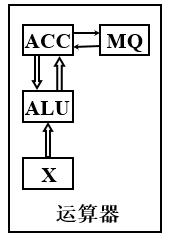
按地址寻访

存储容量：存储器内二进制信息的位数

计算机存储数据的基本单位为字节。

（2）运算器的基本组成及操作过程

运算器：是计算机的数据处理中心，完成各种算术运算、逻辑运算、移位操作等功能。

ACC：累加器

MQ：乘商寄存器

X：操作数寄存器

|  |  |  |  |
| --- | --- | --- | --- |
|  | ACC | MQ | X |
| 加法 | 被加数、和 |  | 加数 |
| 减法 | 被减数、差 |  | 减数 |
| 乘法 | 乘积高位 | 乘数、乘积地位 | 被乘数 |
| 除法 | 被除数、余数 | 商 | 除数 |

ALU：算术逻辑单元

①加法操作过程

初态：ACC 被加数 [M]->X [ACC]+[X]->ACC

②减法操作过程

初态：ACC 被减数 [M]->X [ACC]-[X]->ACC

③乘法操作过程

初态：ACC 被乘数 [M]->MQ [ACC]->[X] 0->ACC [X]×[MQ]->ACC//MQ

④除法操作过程

初态：ACC 被除数 [M]->X [ACC]÷[X]->MQ 余数在ACC中

（3）控制器的基本组成

功能：控制指令的读出、解释和执行，中断事件的处理等。

组成：程序计数器PC—存放当前待执行指令所在地址；

指令寄存器IR—寄存当前的指令（现行的指令）；

控制单元CU—解释现行指令。

程序：有序指令的集合，用来解决某一特定问题。

指令：机器完成某种操作的命令，包括操作码和地址码。

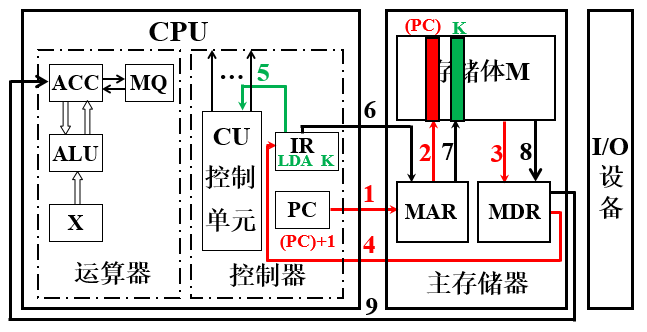
一条指令的执行过程：取指令—分析指令—执行指令。

PC 存放当前欲执行指令的地址，具有计数功能（PC）+1->PC；

IR 存放当前欲执行的指令。

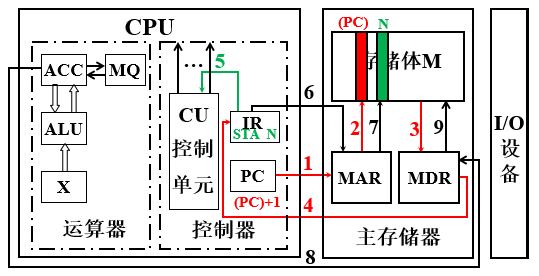
**指令的执行过程（信息流向）：取指令->分析指令->执行指令。**

①取数指令

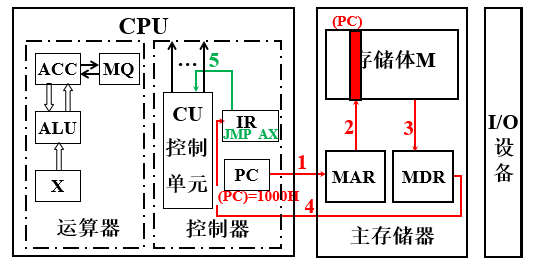




②存数指令：STA N



③跳转指令：JMP AX，寄存器(AX)=1000H



三、计算机硬件的主要技术指标

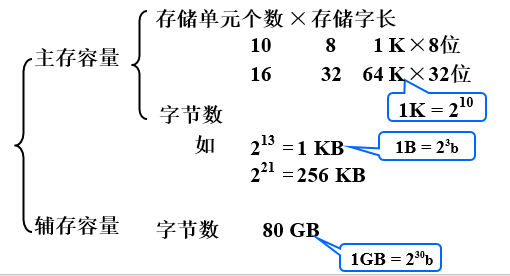
1、机器字长

CPU一次能处理的数据的位数，通常与CPU中寄存器的位数有关。

字长位数越多，精度越高，运算速度越快；通常是8的整倍数，使能存放整数个字符的编码。

机器字长、指令字长和存储字长可以相等也可以不相等。

2、存储容量：存放二进制信息的总位数。



3、运算速度

（1）平均运算速度Vm（指令平均执行速度）

定义：计算机每秒钟执行指令的条数；单位：MIPS（百万条指令每秒）；

求法：先求平均运算时间（平均指令周期）Tm

式中：n：指令的种类；：第i种指令出现的频度(%) ：第i种指令的指令周期(秒)

则平均运算速度：Vm＝1 ⁄ Tm

（2）指令的平均周期数CPI

定义：计算机执行一条指令所需的时钟周期数。

求法：

相关参数：IPC(CPU 每一时钟周期内所执行的指令条数)

说明：一旦处理器设计完成，IPC值就不会再改变了

平均运算速度Vm 与平均周期数CPI的关系：

（3）每秒浮点运算次数FLOPS

定义：每秒所执行的浮点运算次数。

**第二章 计算机的发展及应用**

1、计算机发展的阶段（按照采用的电子器件划分）：

第一代：电子管计算机；第二代：晶体管计算机；第三代：中、小规模集成电路计算机；

第四代：大规模集成电路计算机；第五代：超大规模集成电路计算机。

2、计算机的应用

CAD：计算机辅助设计；CAM：计算机辅助制造；CAI：计算机辅助教学；

CIMS：计算机/现代集成制造系统。

办公自动化和管理信息系统、虚拟现实、人工智能、多媒体技术。

**第三章 系统总线**

一、总线的基本概念

1、总线是连接多个部件的公共信息传输线，是各个部件分时、共享的传输介质。

2、同一时刻仅允许一个部件向总线发送信息，多个部件可同时接受相同的信息。

二、总线分类

按数据传送方式不同,总线可分为串行传输总线和并行传输总线。

系统总线是用来连接CPU、主存和外设部件。按连接部件不同,总线通常可分为片内总线、系统总线和通信总线三种。

1、片内总线—芯片内部的总线

CPU内部，寄存器与寄存器之间、寄存器与算术逻辑单元之间都由片内总线连接。

2、系统总线—指计算机各部件之间的信息传输线

按传输信息的不同,系统总线通常可分为数据总线、地址总线和控制总线三种。

（1）数据总线

双向传输、与机器字长、存储字长有关。传输的是指令，数据，中断类型（入口地址）

若数据总线宽度为8位，指令字长为16位，则CPU取指阶段必须两次访问主存。

（2）地址总线

单向传输、与存储地址、 I/O地址有关。传输的是内存地址、I/O设备地址（端口号）

（3）控制总线

有出（中断请求、总线请求）、有入（存储器读、存储器写、总线允许、中断确认）。

对象不同，方向不同。

3、通信总线

用于计算机系统之间或计算机系统与其他系统之间的通信。

按传输方式分为串行通信总线和并行通信总线。

（1）串行通信总线：使用一条传输线，采用脉冲传送；

优点：只需要一条传输线，成本比较低廉。 缺点：速度慢。

（2）并行传送：每一位数据需一条传输线，一般采用电位传送；

优点：短距离内，传输速度快。缺点：成本高；距离短。

三、总线性能指标

总线特性：机械特性、电气特性、功能特性、时间特性。

1、总线宽度

数据线的根数，用位表示。

2、总线带宽

每秒传输的最大字节数（MBps）；最大数据传输率=总线宽度×总线工作频率。

总线工作频率为33MHz，总线宽度为32位，则总线带宽为33×（32÷8）=132MBps。

3、总线复用

一种信号线在不同的时间传输不同的信息，可以减少总线中信号线的数量。

不同信号共用一组信号线,分时传送,这种总线传输方式是总线复用。

四、总线控制

总线控制：统一管理总线上多个部件如何发送信息，如何接收信息，如何防止信息丢失等一系列问题。

1、判优控制—解决总线使用权问题

当多个设备同时请求占用总线时，由总线判优控制器按其优先级别仲裁，决定由哪个主设备占用总线。主设备：对总线有控制权，可向总线发送请求。从设备：对总线没有控制权，只能响应主设备发来的总线命令。

（1）**集中式—将控制功能集中在一处（如在CPU中）**

①链式查询（对电路故障最敏感）

优点：需3根控制线，线路少，优先级固定，易扩展；

缺点：优先级低的设备可能难获得总线使用权，对电路故障敏感。

在链式查询方式下，若有N个设备，则只有一条总线请求线。

②计数器定时查询

优点：设备的优先级可改变，系统故障敏感度降低；

缺点：需（log2N+2）条控制线，线路复杂。其中N为设备总数。

在计数器定时查询方式下，若计数从0开始，则设备号小的优先级高。

计数器定时查询，若每次计数从上一次计数的终止点开始，则每个设备使用总线的机会相等。

③独立请求方式（响应时间最快）

优点：设备的优先级可灵活改变，响应速度快；

缺点：控制线数量多（2N），结构复杂。

（2）分布式—将控制功能分散在各个部件或设备上

2、通信控制—解决获得使用权后，通信双方协调配合问题

目的：主要解决通信双方如何获知传输开始和传输结束、以及通信双方如何协调如何配合。

总线周期：完成一次总线操作的时间。（申请分配阶段、寻址阶段、传数阶段、结束阶段）

（1）**同步通信—由统一时钟控制数据的传送**

同步控制方式是由统一的时序信号控制的方式。

在同步通信中,一个总线周期的传输过程是先传送地址，再传输数据。

优点：规定明确、统一，模块间配合简单一致。

缺点：主从模块的时间配合属于“强制性同步”，对于速度不同的部件而言，严重影响总线工作效率，缺乏灵活性。

适用场合：总线长度短（短距离）、各部件存取时间一致（或差异不大）的场合。

（2）**异步通信—没有公共时钟标准，采用应答方式**

应答方式：又称握手方式，即主模块发出请求，从模块响应，然后开始通信。

特点：没有同步时钟，允许不同速度的模块和谐工作。但是结构复杂，且对传送的字符有格式要求。

格式：1个起始位（低电平），5-8个数据位，1个奇偶校验位（校验位+信息位），1或1.5或2个终止位（高电平）。起始位后紧跟传送字符的最低位。

①不互锁（速度最快）

主设备请求信号的撤除与是否收到应答无关；从设备应答信号的撤除与主设备是否撤除请求信号无关。

举例：单机系统中，CPU向主存写信息，所给出的地址、数据、写命令以及写入数据。

②半互锁

主设备请求信号的撤除必须在收到应答信号后；从设备应答信号的撤除与主设备是否撤除请求信号无关。

举例：多机系统中，CPU访问共享存储器，发出访存请求后，必须收到未占用的应答信号方可进行访存。

③全互锁

主设备请求信号的撤除必须在收到应答信号后；从设备应答信号的撤除必须得到主设备撤除请求信号后。

举例：网络通信。

（3）半同步通信—同步、异步结合（即采用时钟信号，又采用握手信号）

特点：既有公共时钟控制，又允许速度不同的工作模块协调的工作，只需插入等待周期的措施来协调通信双方的配合问题，称作半同步控制。

优点：控制方式比异步通信简单，可靠性高。

缺点：对系统时钟频率不能要求太高，系统工作速度不高。总线传输周期变长。

（4）分离式通信—充分挖掘系统总线每个瞬间的潜力

特点：充分提高了总线的有效占用。

在分离式通信方式中，总线上所有模块都可以成为主模块。

**第四章 存储器**

一、存储器分类

计算机的存储系统是指Cache、主存储器和外存储器。

计算机的存储器采用分级方式是为了解决容量、速度、价格三者之间的矛盾。

1、主存

（1）随机存储器RAM—在程序的执行过程中可读可写

主要用于：主存，高速缓冲存储器



①静态RAM（SRAM）

存储原理：用触发器工作原理存储信息。

特点：信息读出后，仍保持原状态，不需要再生；断电后，原信息丢失。

存取速度快，但集成度低，功耗大，一般用于组成高速缓冲存储器。

某一RAM芯片，容量为1K×4位,其数据线为4条，地址线为10条。若重合法驱动方式，其字线条数为80条。

②动态RAM（DRAM）

存储原理：电容存储电荷的原理存储信息。

特点：所存储信息需要刷新；集成度高，功耗更低。

DRAM的刷新是以行为单位的。

常见的动态RAM基本单元电路有三管式和单管式两种，它们共同的特点都是靠电容存储电荷的原理来寄存信息。

动态半导体存储器需要刷新是因为存储电荷的电容放电。

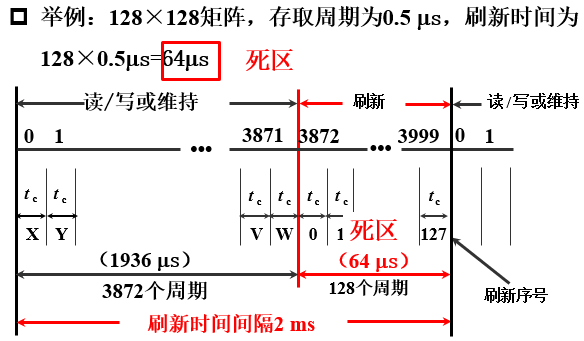
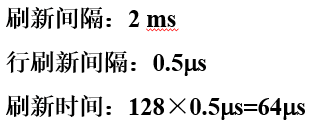
刷新（间隔、死区）

实质：将原存信息读出，再经由刷新放大器形成原信息并重新写入的过程。

刷新原因：电容上的电荷一般只能维持1~2ms，即使不断电，信息也会自动消失。如果存储单元长期得不到访问，则其原信息将会慢慢消失。

动态 RAM 刷新方式：

a.集中刷新：在规定的一个刷新周期（2ms）内，对全部存储单元集中一段时间逐行进行刷新，刷新时必须停止读/写操作。

优点：读写操作时不受刷新工作的影响，因此系统的存取速度比较高。

缺点：在集中刷新期间（死区）不能进行读写，而且存储容量越大，死区就越长。 这增加了存储管理的困难，显然对于高速高效的计算机系统工作是不利的。

b.分散刷新（不存在死时间）

把刷新操作分散到每个存取周期内进行。

优点：没有死区。

缺点：加长了系统的存取周期，降低了整机速度；刷新过于频繁，从而使刷新时间增加。没有充分利用所允许的最大时间间隔（2ms）。

c.异步刷新

前两种方式的结合，充分利用了最大刷新间隔时间，把刷新操作平均分配到整个最大刷新间隔时间内进行。故：相邻两行的刷新间隔=最大刷新间隔时间÷行数

优点：死区小，刷新占用时间少。

缺点：系统结构变复杂。

（2）只读存储器ROM—在程序的执行过程中只读

主要用于：BIOS芯片，U盘

a. 掩模 ROM ( MROM，不可改写)

b. PROM (可编程只读存储器，一次性编程)

c. EPROM (可擦除可编程只读存储器，多次性编程)

EPROM是可改写的,但不能作为随机存储器

d. EEPROM (电擦除只读存储器，多次性编程)

e. Flash Memory (闪速型存储器) U盘

（3）技术指标

存储容量=存储单元个数 × 存储字长

存取周期=存取时间+复原时间

存储器的存取周期是存储器进行连续读或写操作所允许的最短时间间隔。

存取时间：启动一次存储器操作到完成该操作所需的全部时间；

读出时间：存储器收到有效地址开始，到产生有效输出；

写入时间：存储器收到有效地址开始，到数据写入被选中单元。

存储器的带宽是单位时间内存储器存取的信息量；单位：位/秒或字节/秒。

提高存储器带宽的方法：缩短存取周期、增大存储字长、增加存储体。

2、辅存

磁盘是直接存取，磁带是顺序存取。

3、缓存

将主存块调入缓存的任务全部由硬件完成。

4、闪存

闪存是EEPROM的进一步发展,可读可写，用MOS管的浮栅上有无电荷来存储信息。闪存是ROM的一种，写入时必须先擦除原有数据,所以写速度比读速度慢很多。闪存是一种非易失存储器，采用随机访问方式。现在常见的SSD固态硬盘就由Flash芯片构成。

存储元由MOS管组成是一种半导体存储器；掉电后信息不易失，是一种非易失性存储器；采用随机访问方式，可替代计算机外部存储器。

二、存储器的层次结构

1、主存-辅存层次

主存：速度快、容量小、价格高；辅存：速度慢、容量大、价格低。

2、主存-缓存层次

主存和缓存按块存储，块的大小相同，块内地址相同。

三、**主存中地址单元的分配：地址码能够指定的最小存储单位**

1、按字编制—访问的单位为字长

计算机可寻址的最小信息单位是一个存储字，一个存储字所包含的二进制位数称为存储字长。

某机字长32位，存储容量为4MB，若按字编址，它的寻址范围是1M。

2、按字节编制—访问的单位为一个字节

计算机可寻址的最小信息单位是一个字节。

某机字长16位，存储容量为64KB，若按字节编址，它的寻址范围是0-(64K-1)。

四、半导体存储芯片的译码驱动方式

1、线选法

特点：一根字选择线可直接选中一个存储单元的各位。结构简单，但只适用于容量不大的存储芯片。

某一RAM芯片，容量为1K×1位,其数据线为1条，地址线为10条。若采用线选法驱动方式,其位线条数为1条， 其字线条数为1K或1024条。

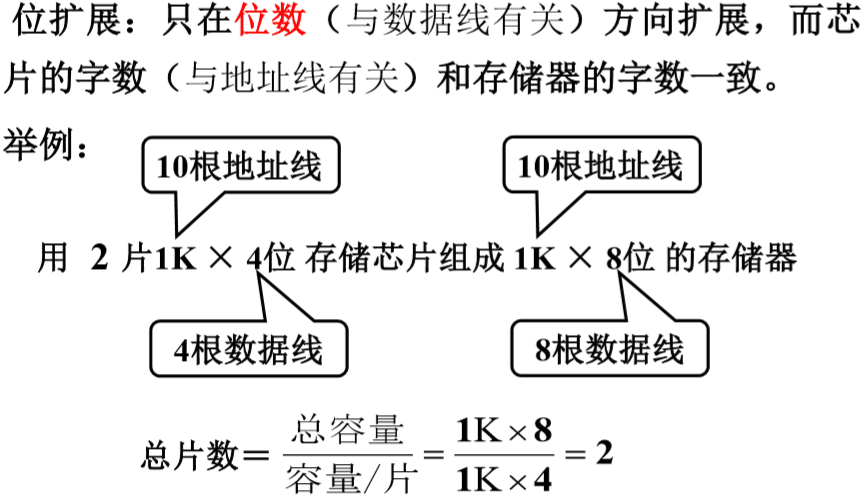
2、重合法

被选单元由X和Y两个方向的地址决定，故称为重合法。

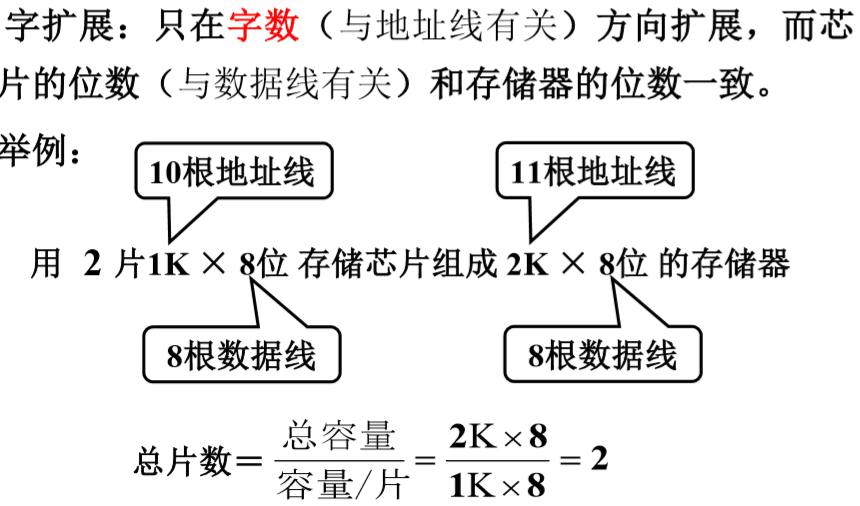
某一RAM芯片，容量为1K×1位,其数据线为1条，地址线为10条。若重合法驱动方式,其位线条数为1条，其字线条数为64条。

五、**存储器的CPU的连接**

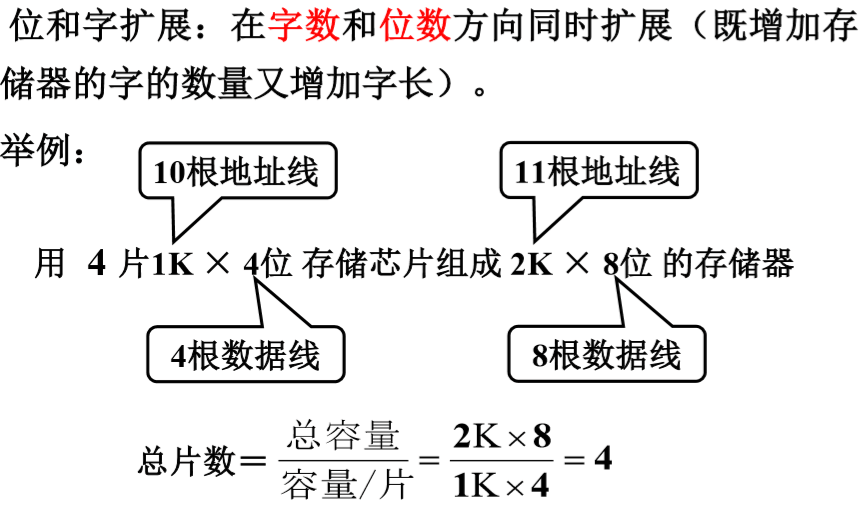
1、位扩展（增加存储字长）

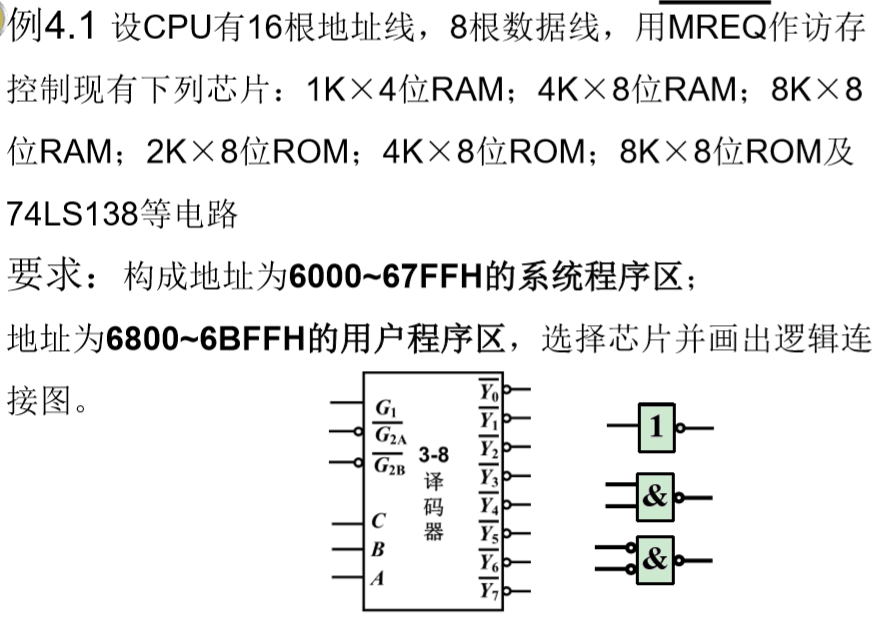


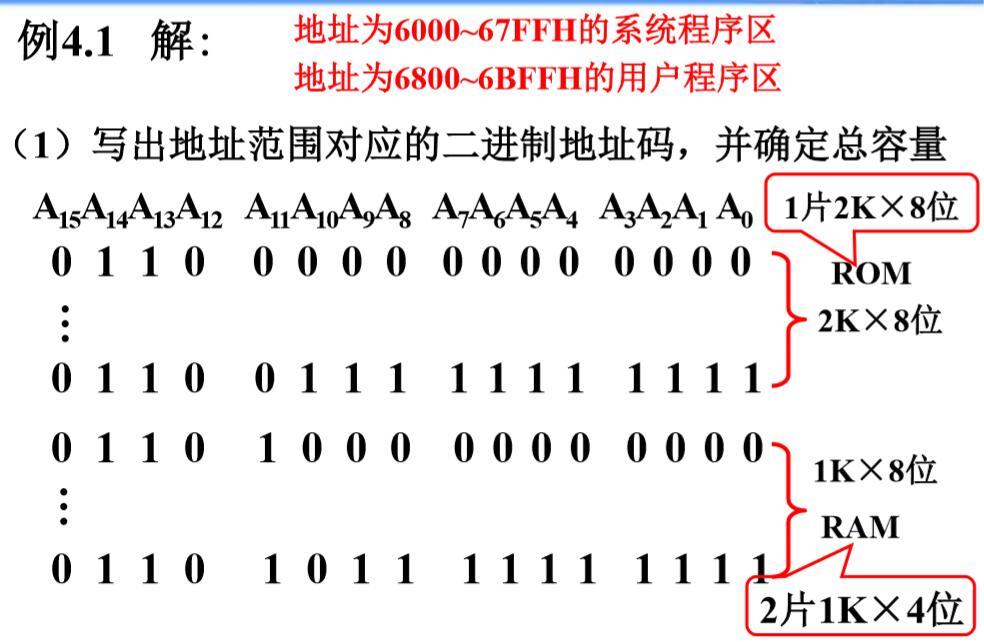
2、字扩展—增加存储字（存储单元）的数量

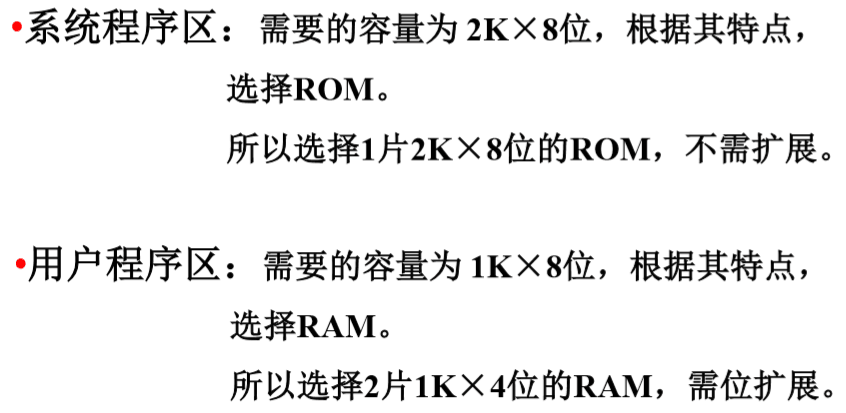


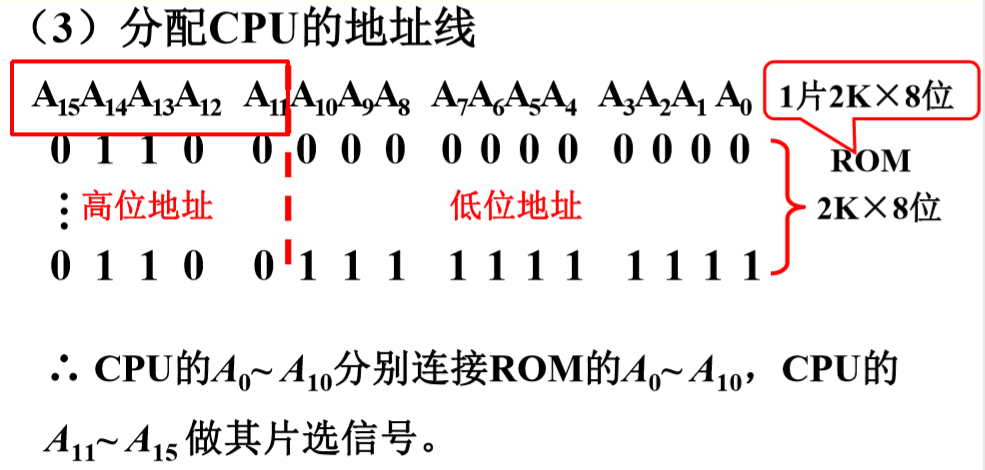
3、位、字扩展

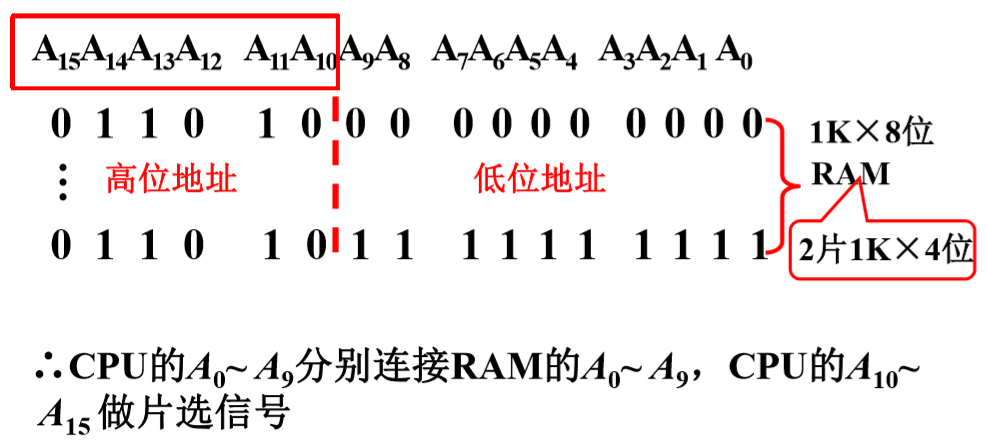


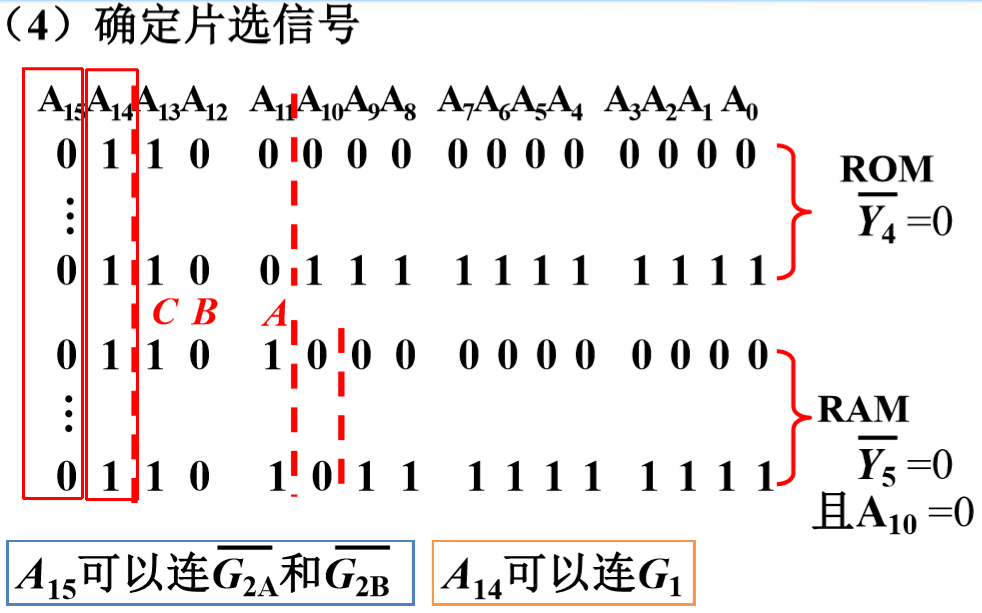


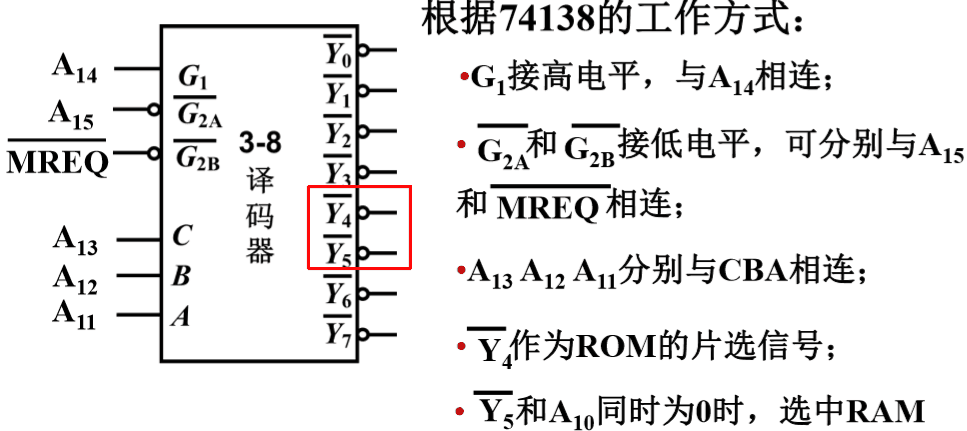


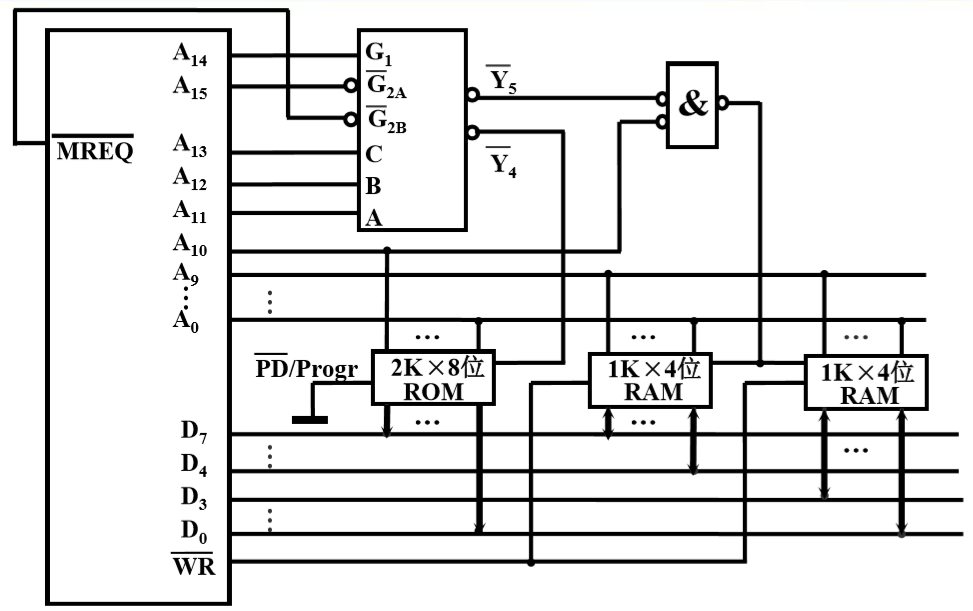






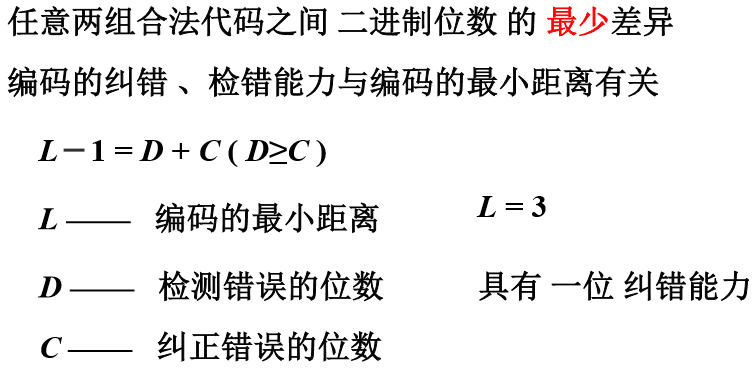




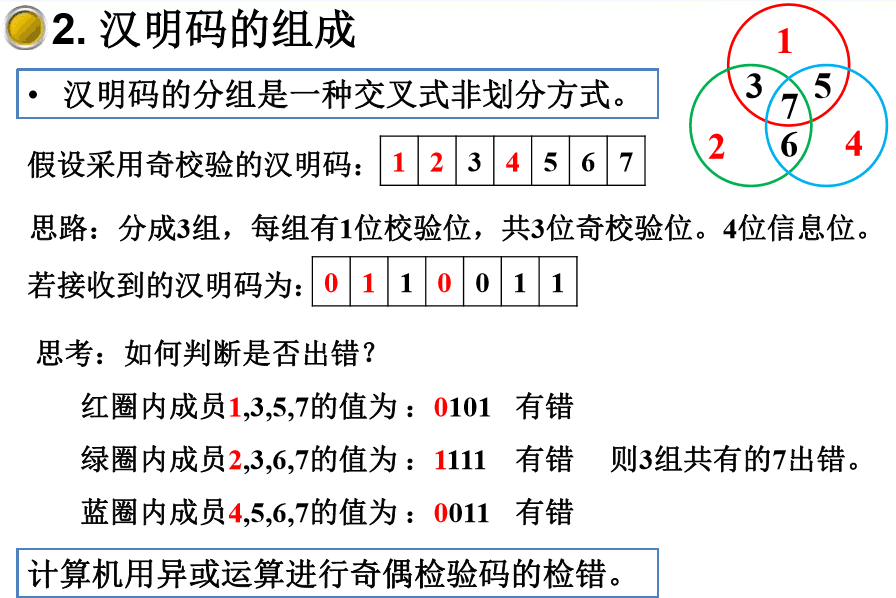


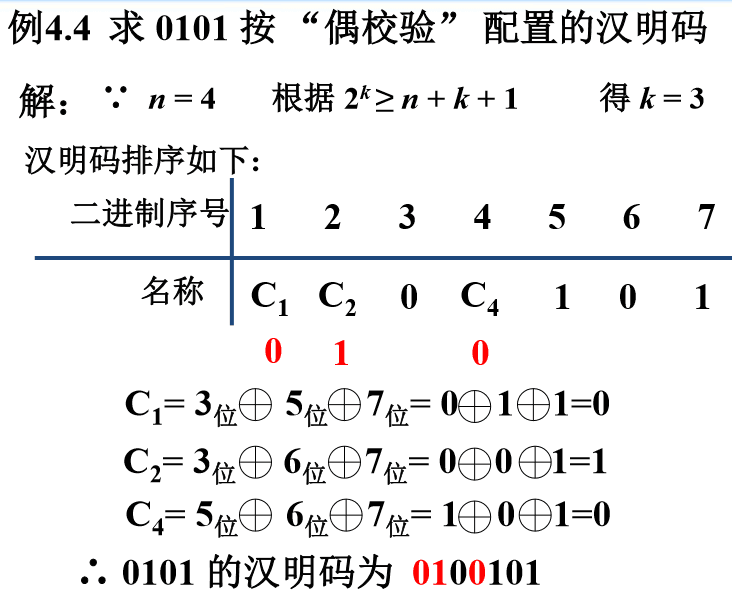
六、汉明码

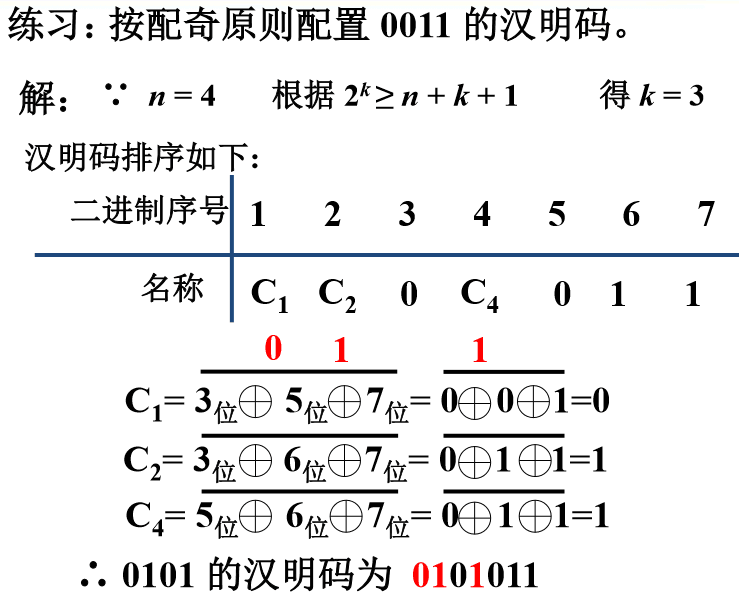
1、具有一位纠错能力

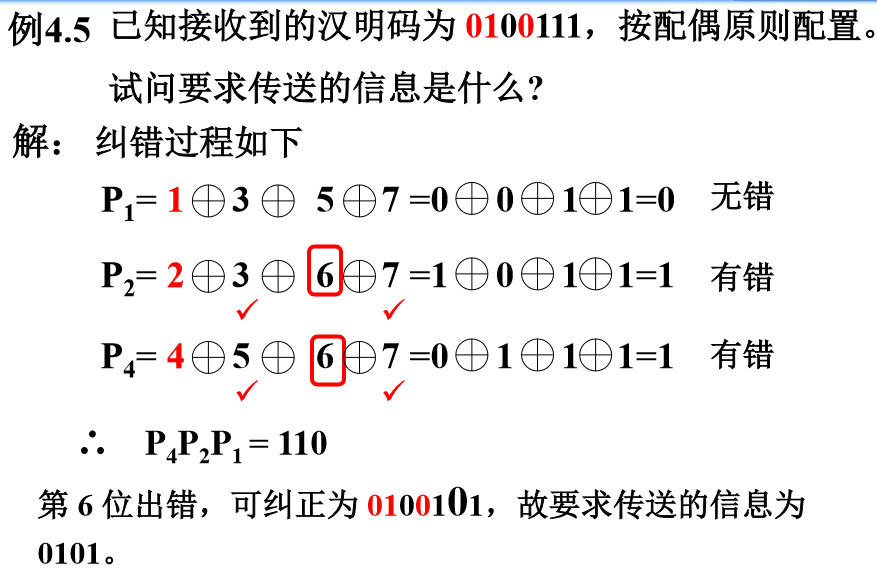


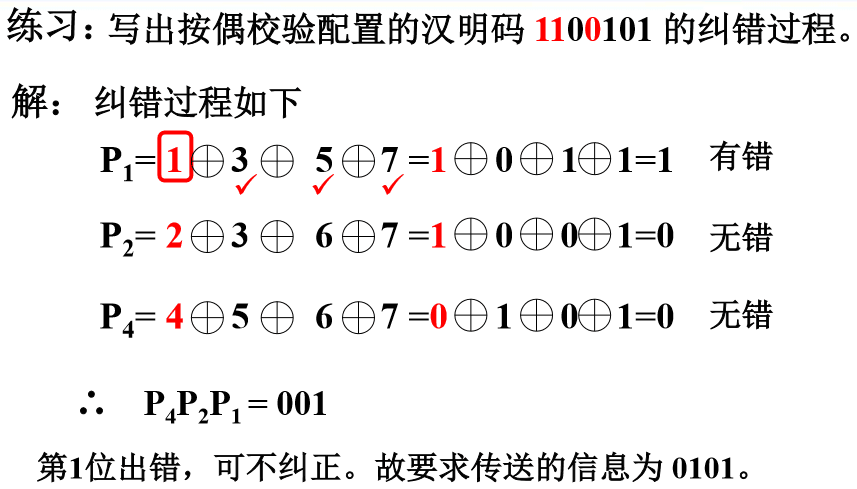
2、汉明码构成

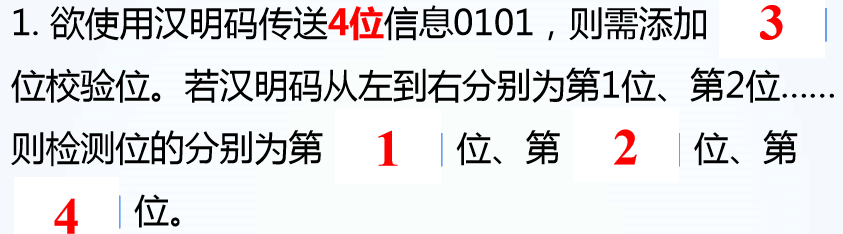




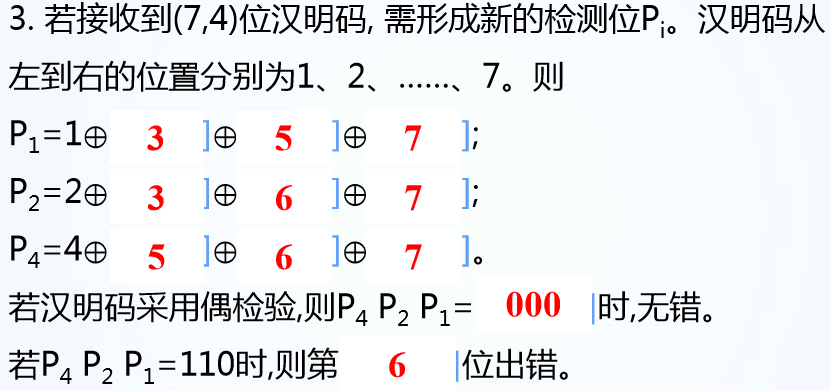








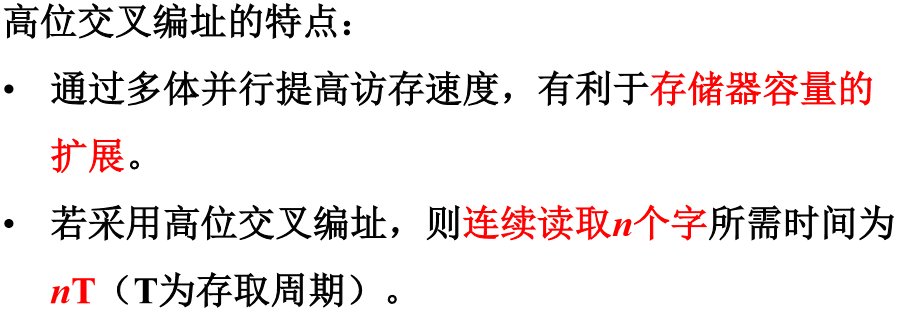


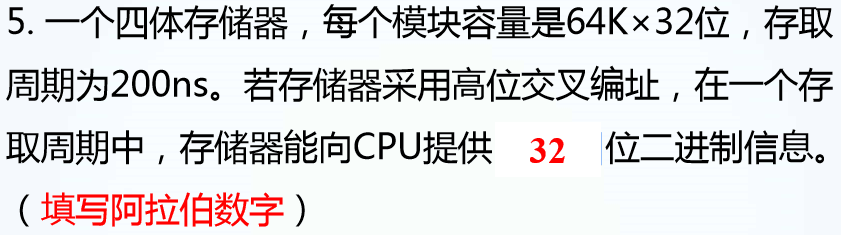




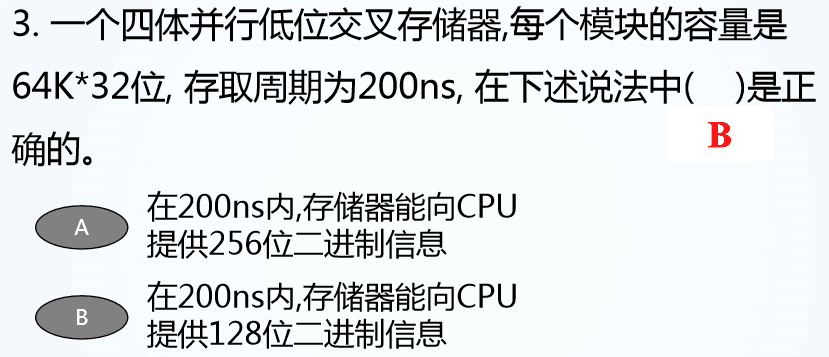
七、高位交叉和低位交叉

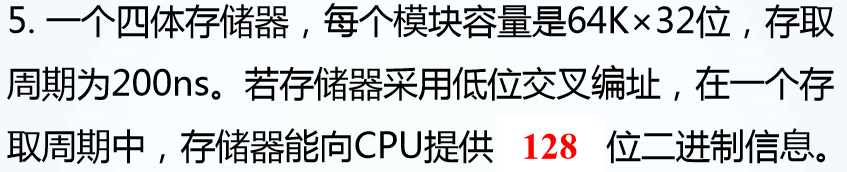
（1）高位交叉：高位体号，低位体内地址，适合存储器容量的扩展。

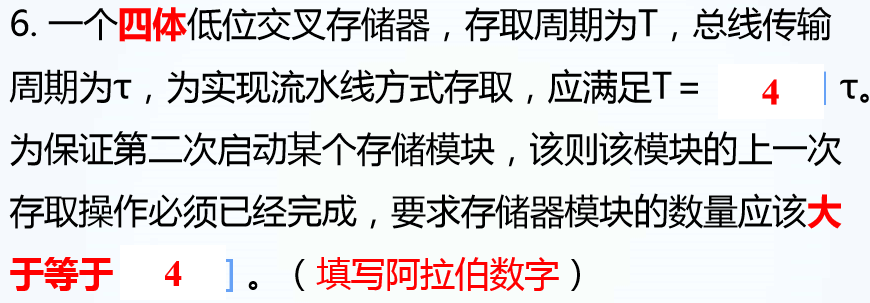


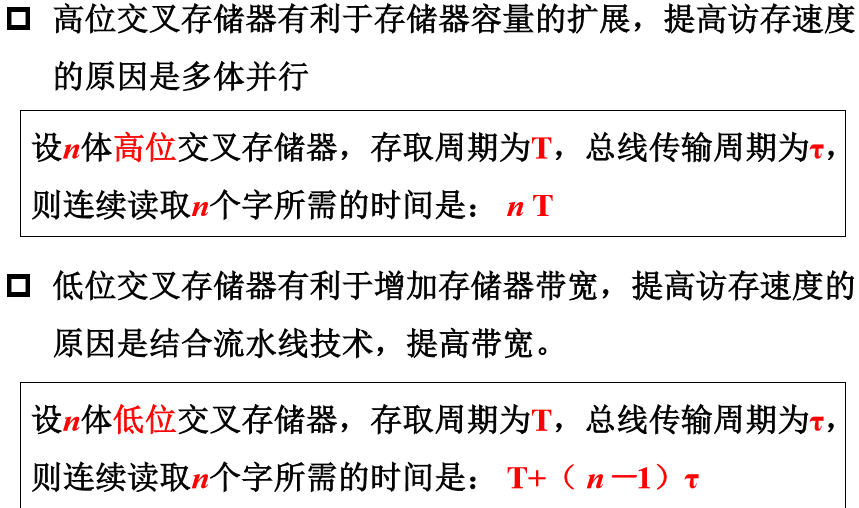


（2）低位交叉：结合流水线技术，适合提高存储器带宽和访问速度。





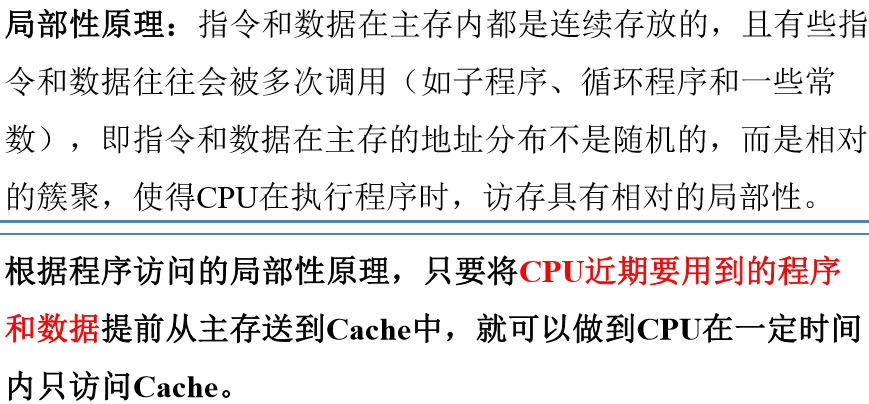




八、高速缓存

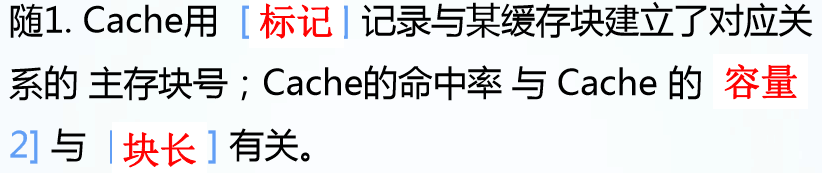
目的：解决CPU和主存之间的速度匹配问题。

1、基本原理：程序访问局部性原理。

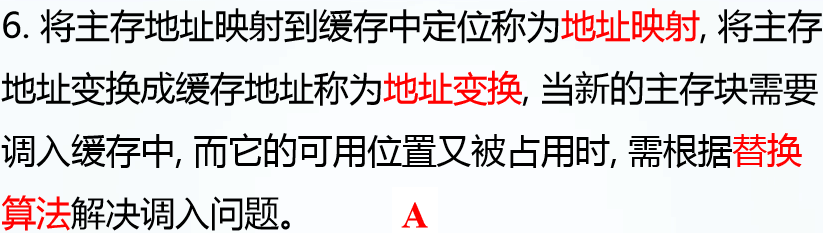


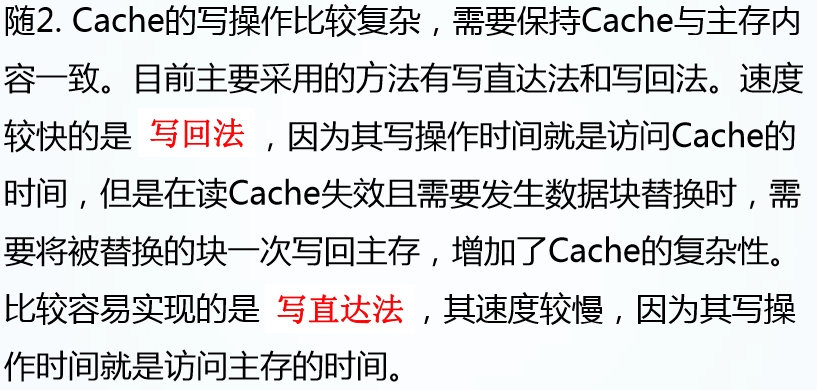
2、命中率

缓存容量越大，命中率越高。命中率=总命中次数/（总命中次数+访问总次数）



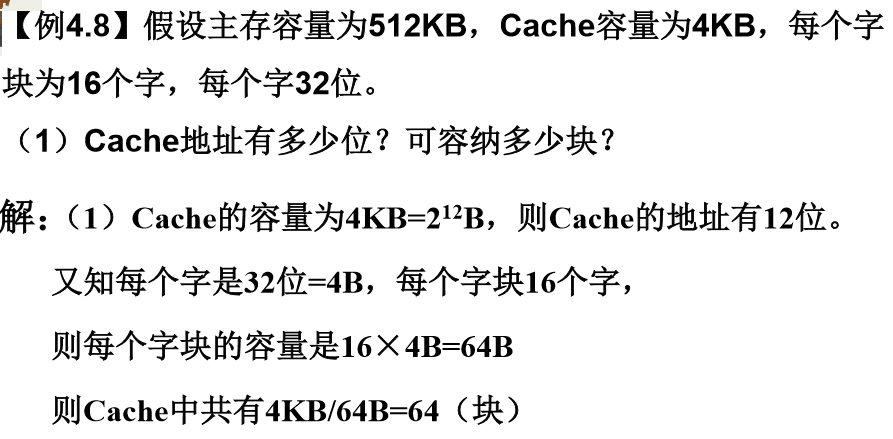
3、**地址映射**



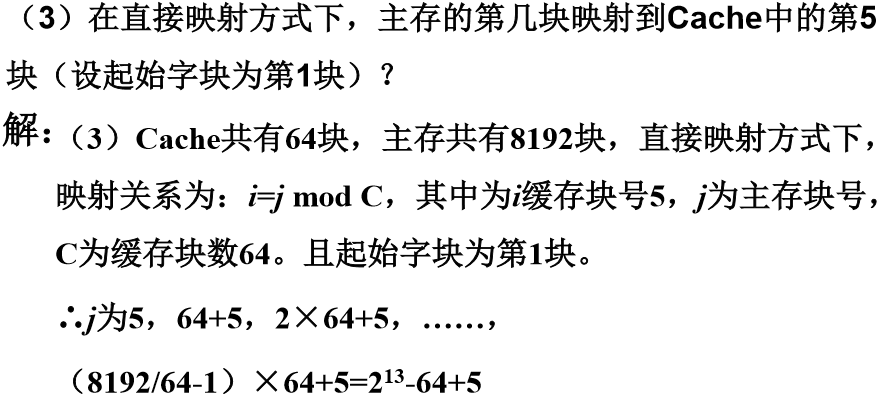


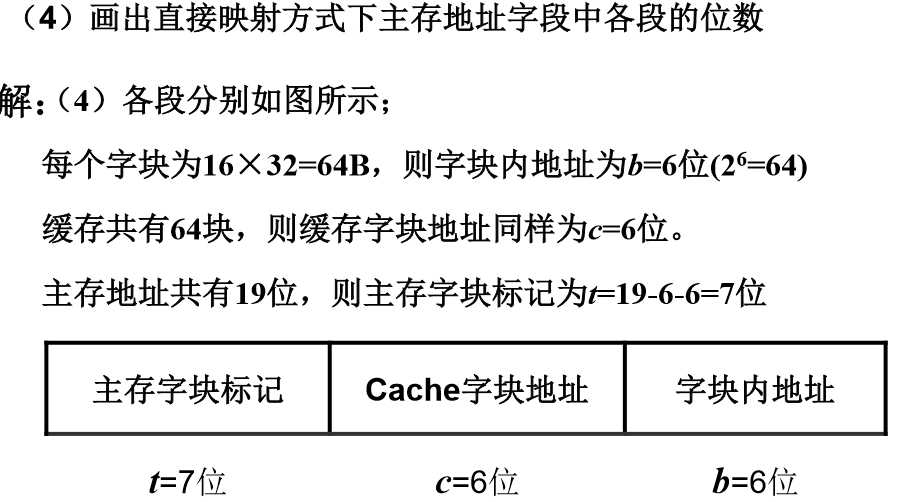
（1）直接映射

每个主存块只能固定的映射到某个缓存块。最不灵活。



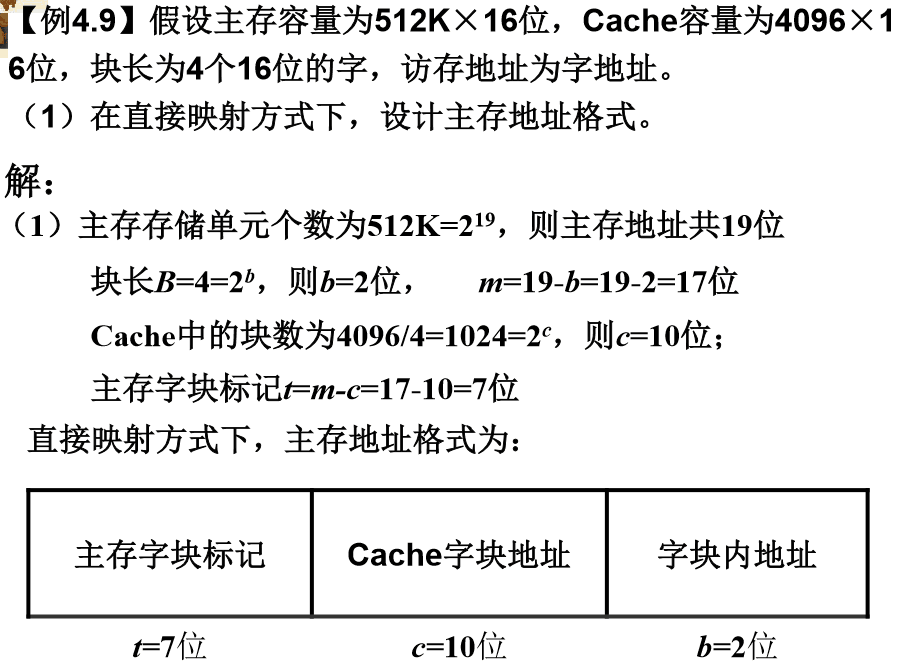
![F:\System\Documents\Tencent Files\838148032\Image\C2C\Image2\%4](`$AJ57S`B~5JY`LYF9Y.png](data:image/png;base64,)

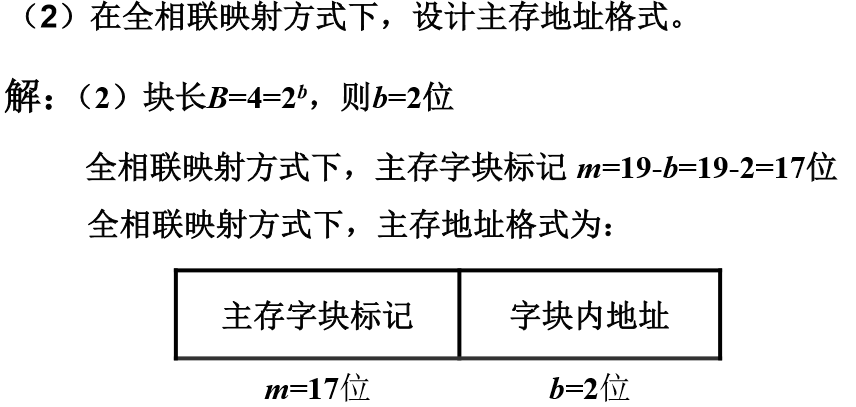




（2）全相联

每个主存块可以映射到任何一个缓存块。灵活性最高，成本最高。

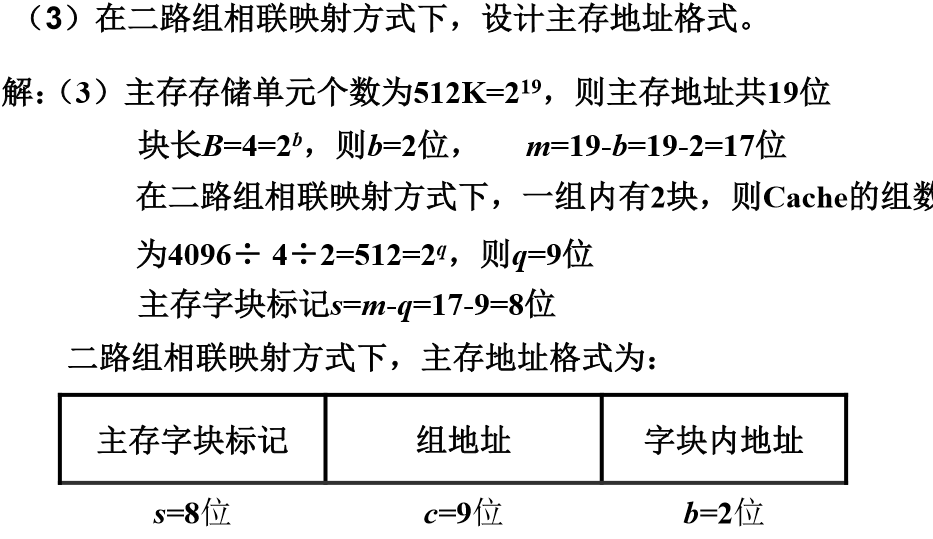


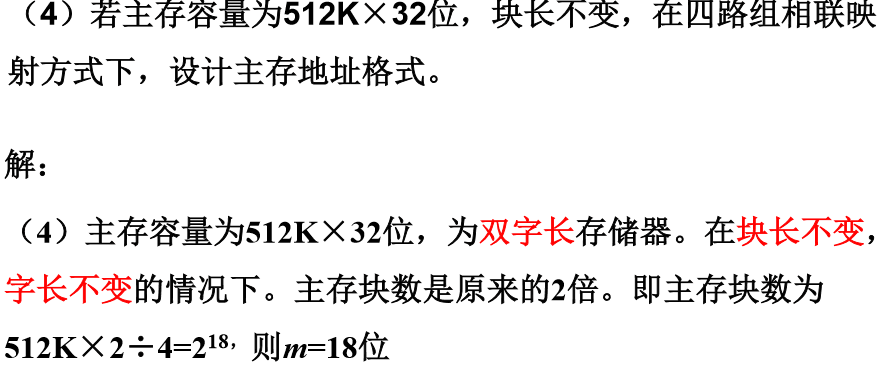


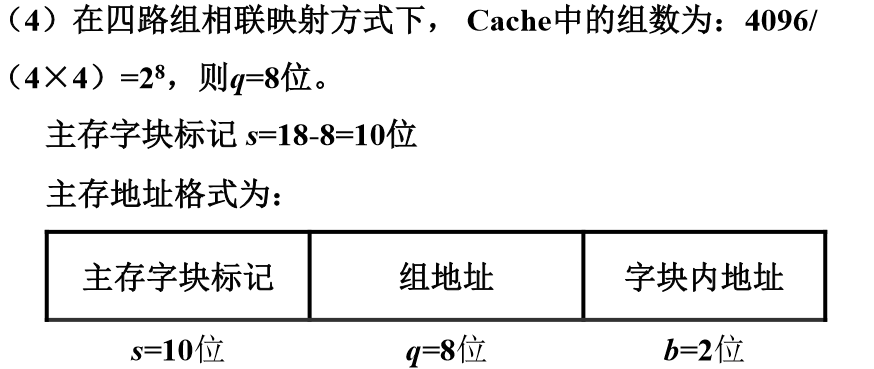
（3）组相联

某一主存块只能映射到某一缓存组中的任意一块。

n路组相联映射是指每个缓存组中有n个缓存块。







4、替换策略

（1）先进先出FIFO

（2）近期最少使用LRU

（3）随机法

**第六章 计算机的运算方法**

一、无符号数和有符号数

1、无符号数

M位寄存器存储无符号数的范围是[0，2^M-1]。

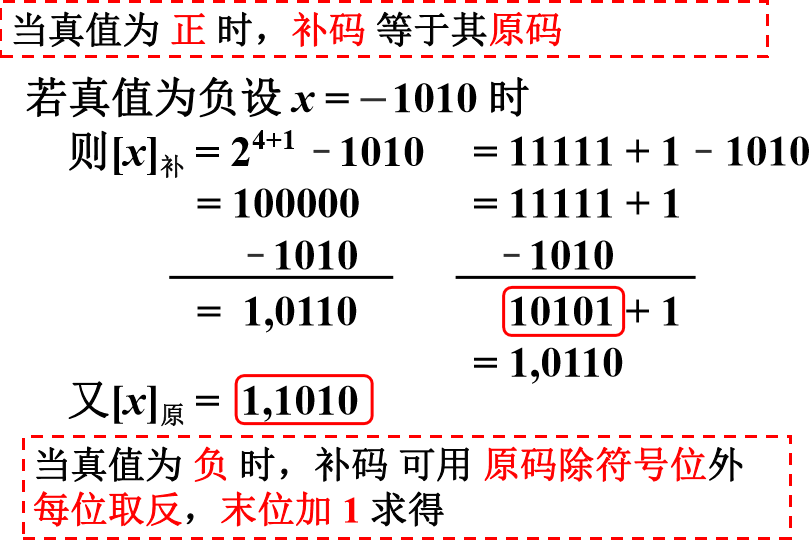
2、有符号数

（1）原码

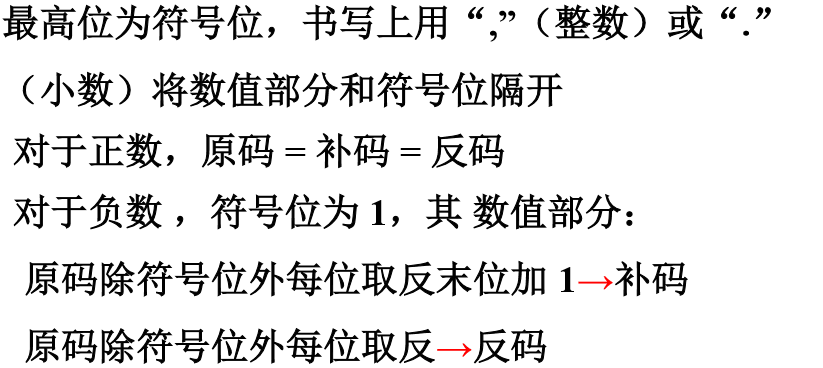
只变符号位。看小数点还是逗号。

（2）**补码**

**-8的补码是1000**



（3）反码

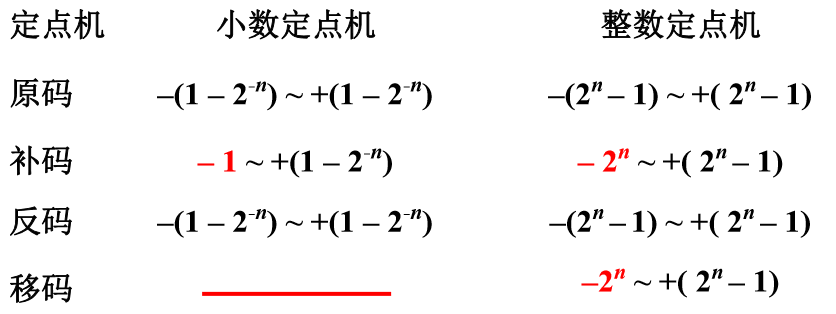


（4）移码

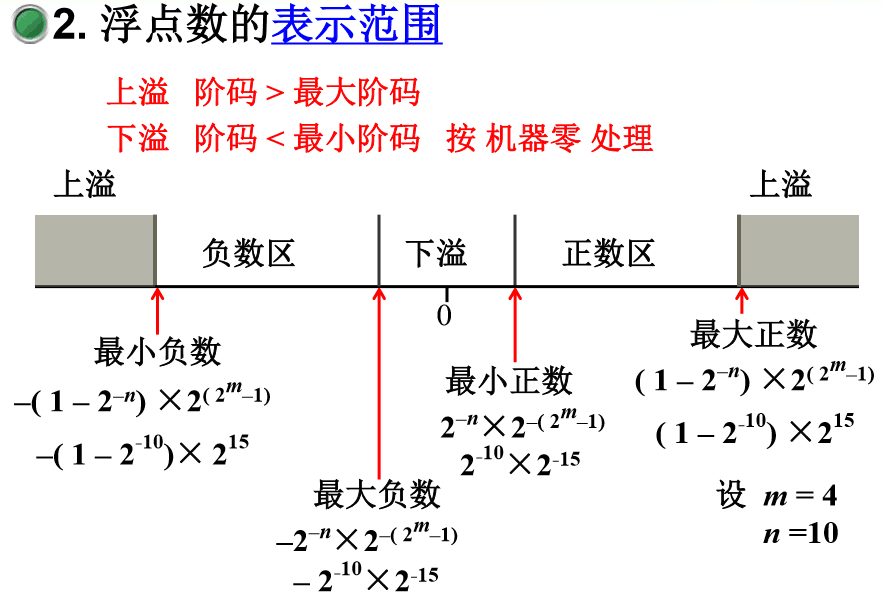
[x]移=2^n+x 移码和补码只差一个符号位。

二、数的定点表示和浮点表示

1、定点表示

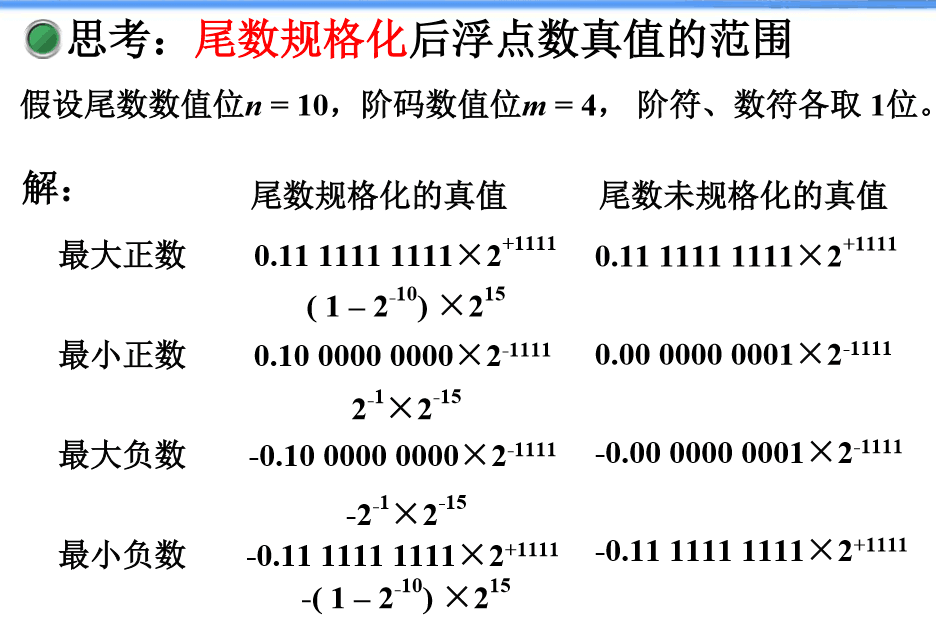


2、浮点表示

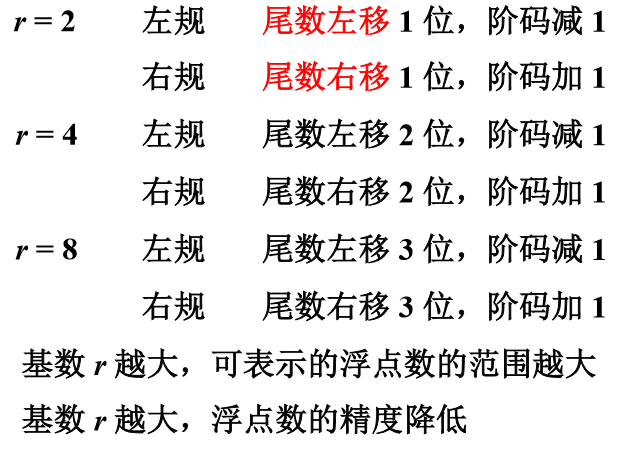


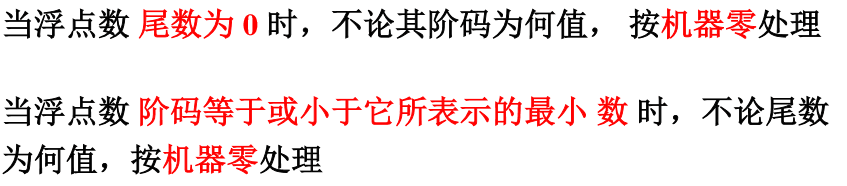
（1）尾数

真值规格化标准



（2）阶码



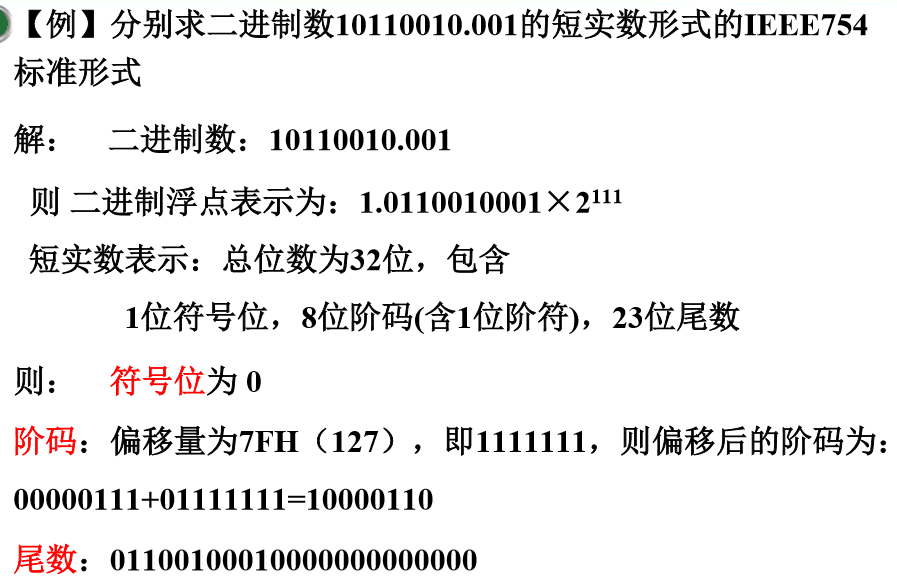


（3）溢出

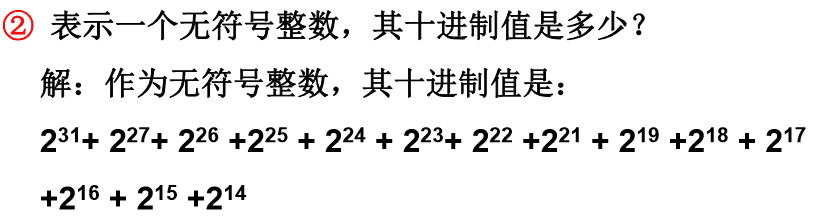
①上溢：中断

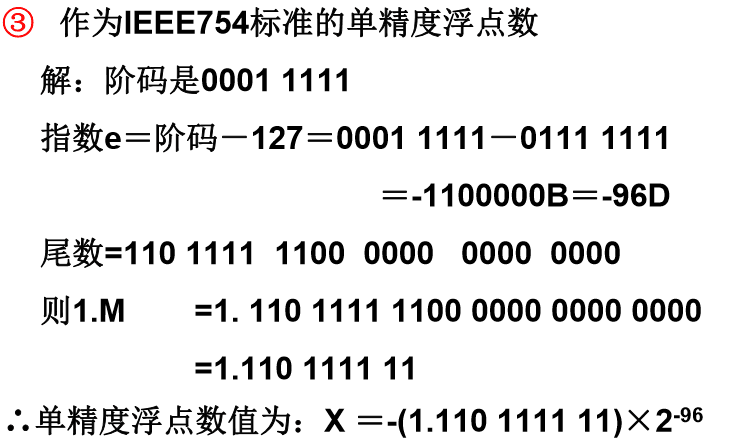
②下溢：机器零

（4）IEEE 754标准





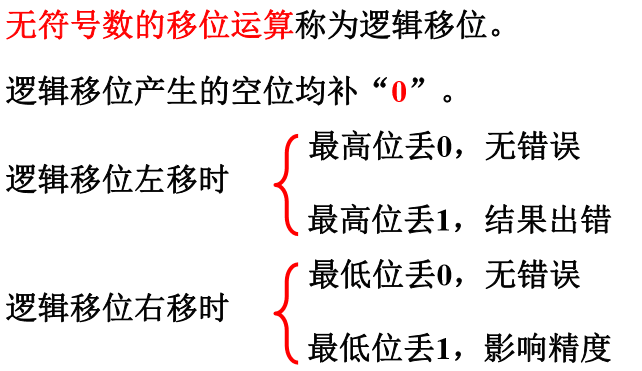




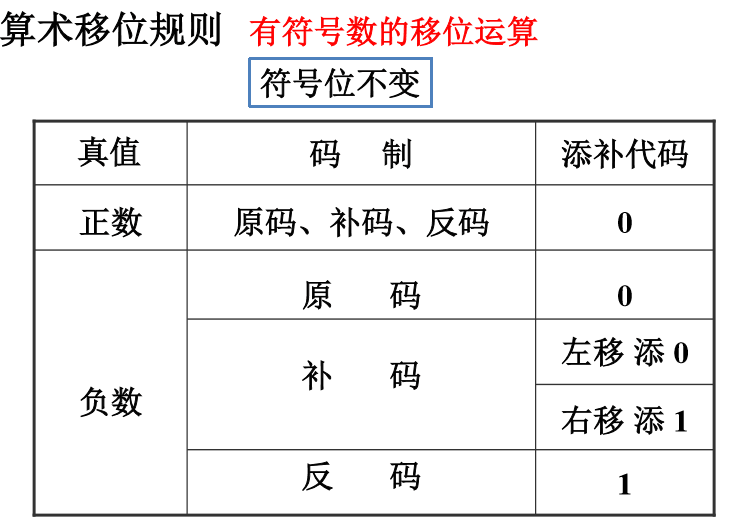
三、定点运算

1、移位运算

（1）逻辑移位

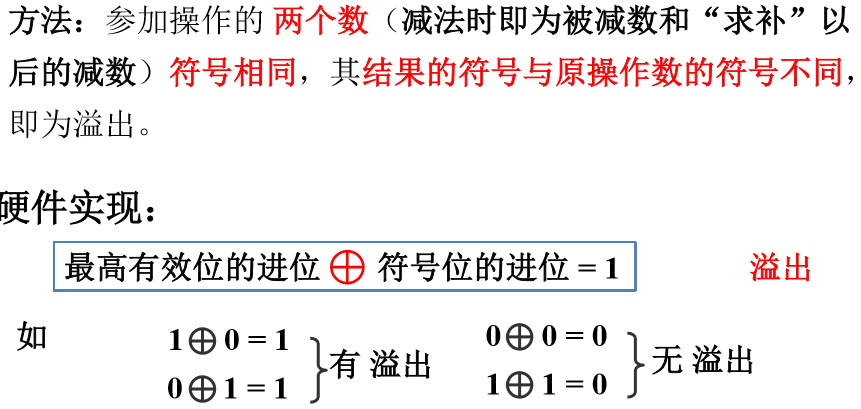


（2）**算术移位**

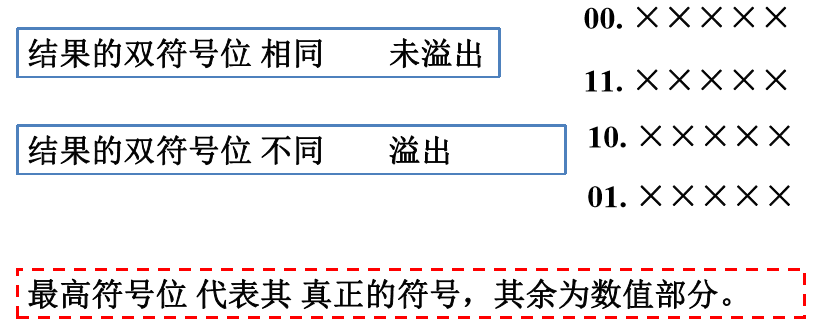


2、加减法运算

（1）单符号位补码溢出判断

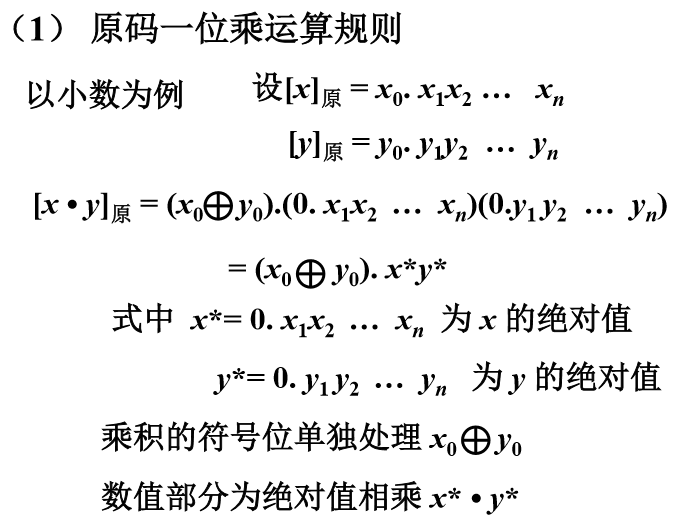


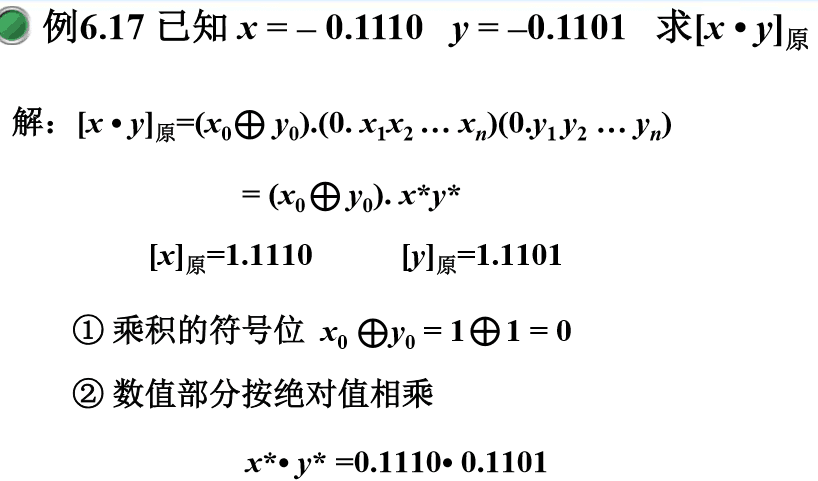
（2）双符号位补码溢出判断



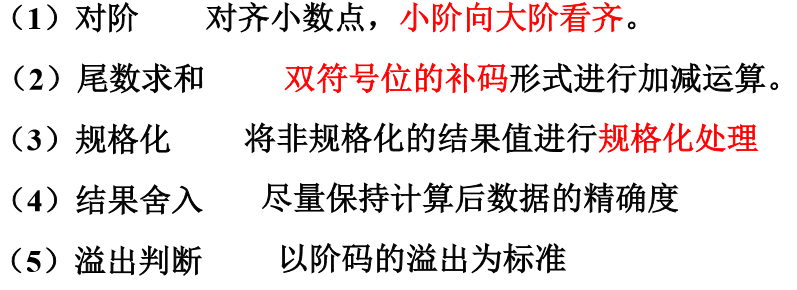
3、乘法运算

小数原码乘法运算





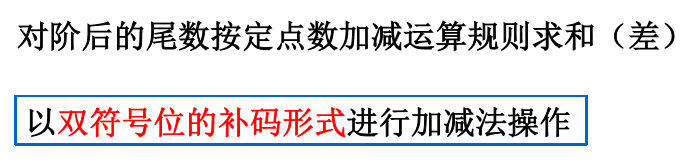
四、浮点数加减运算



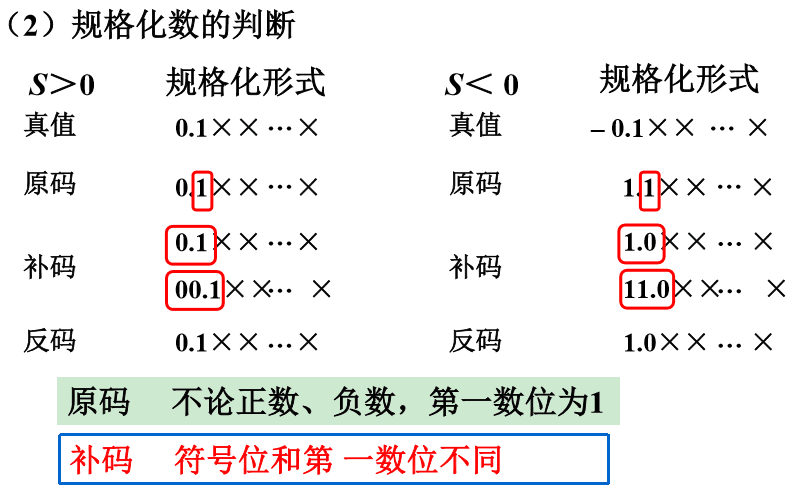
1、对阶

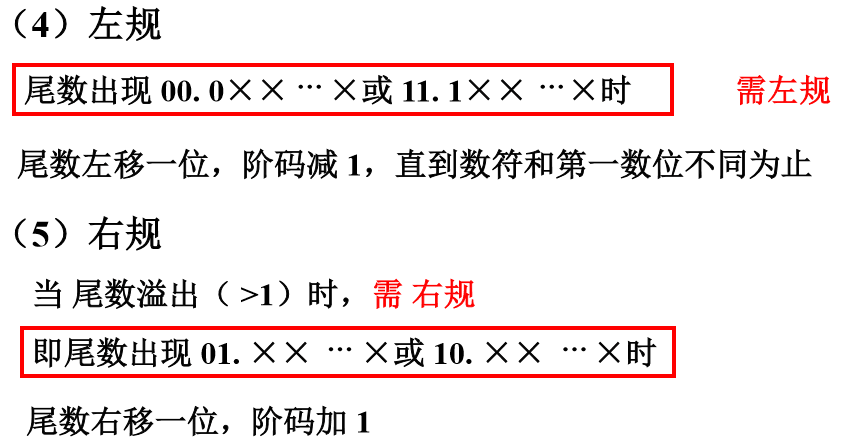
小阶向大阶看齐

2、尾数求和



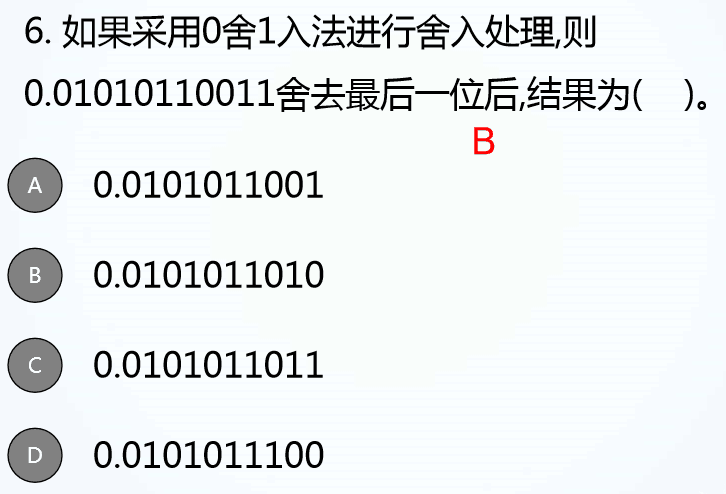
3、规格化



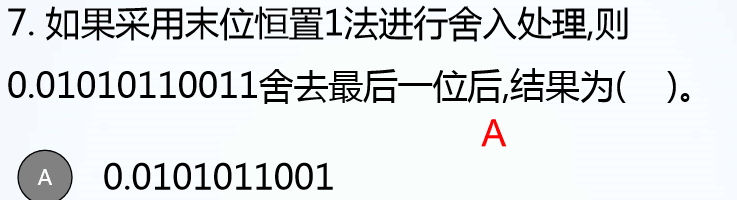


4、舍入

（1）0舍1入



（2）恒置1

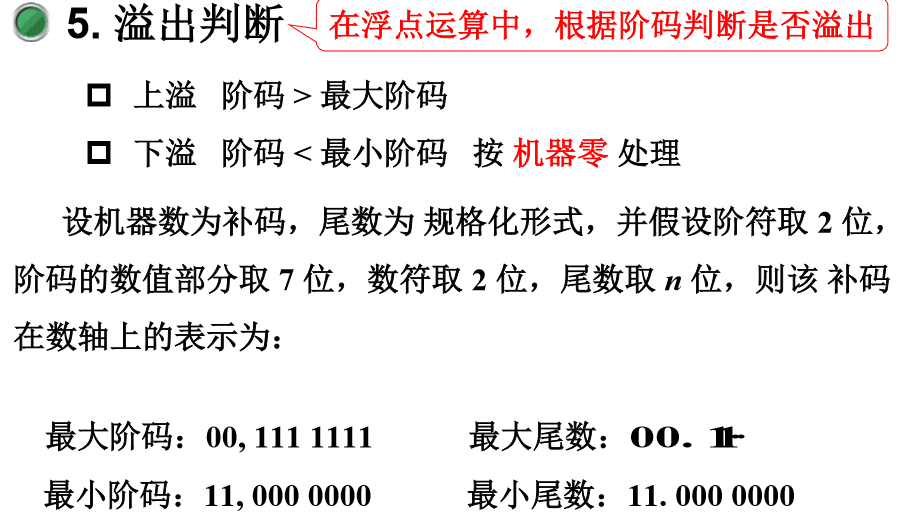


5、溢出判断

用阶码判断溢出

（1）上溢：中断

（2）下溢：机器零



**第七章 指令系统**

一、机器指令

分类：微指令、机器指令、宏指令。

指令：用二进制代码表示的结构形式，有操作码和地址码两部分组成。

1、**操作码—指明该指令所要完成的操作**

操作码的位数反映了机器的操作种类。

操作码字段为8位，最多包含指令的数目为2^8=256条。

（1）操作码固定

特征：所有指令长度均相同。

优点：控制简单，速度快，适用于字长较长的计算机。

（2）扩展操作码

操作码的位数随地址数的减少而增加。目的：保持指令字长度不变而增加指令的数量。

特征：利用操作码扩展技术将操作码扩展到地址码字段，使各类指令的操作码长度不同。

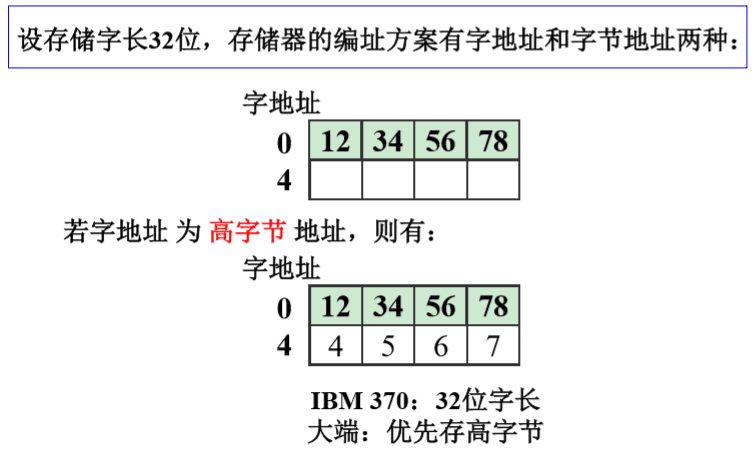
优点：有效的压缩操作码的平均长度，适用于字长较短的计算机。

2、地址码

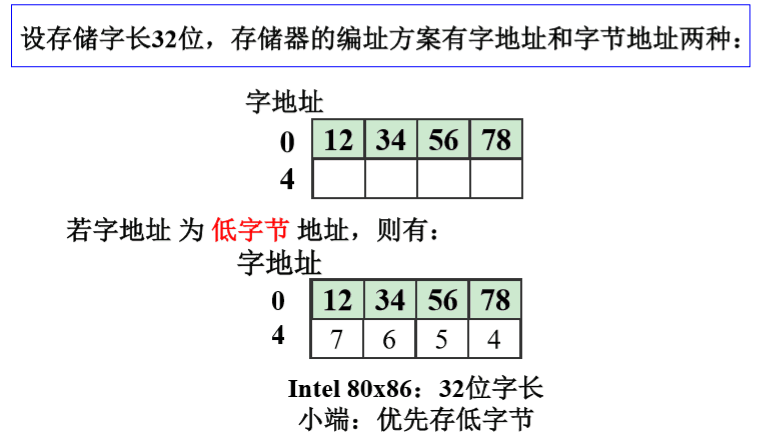
四地址、三地址、二地址、一地址、零地址。

二、数据在存储器中的存放方式

（1）大端（优先存高字节）



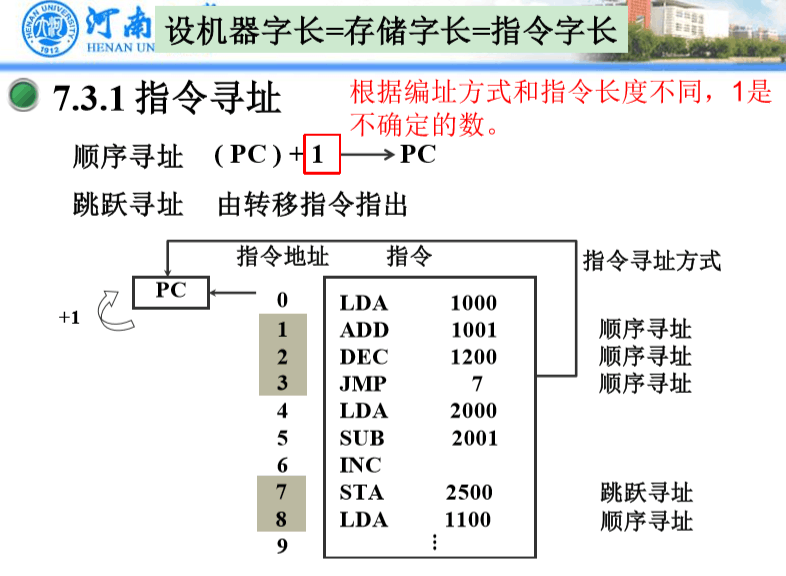
（2）小端（优先存低字节）



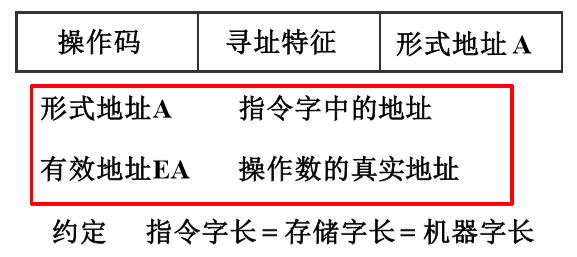
三、寻址方式

确定本条指令的操作数地址；下一条欲执行指令的指令地址。

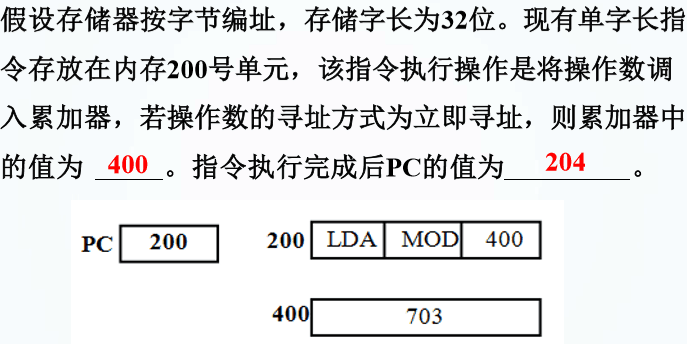
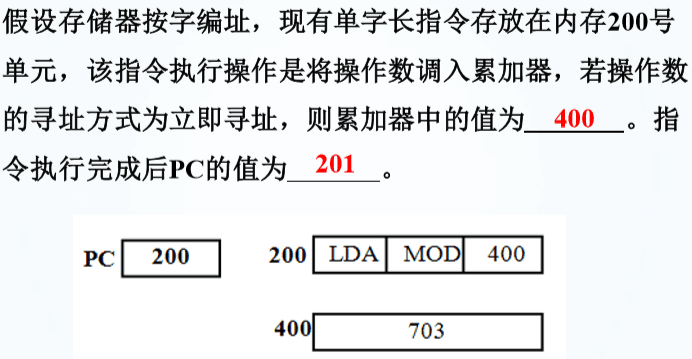
1、指令寻址



2、数据寻址



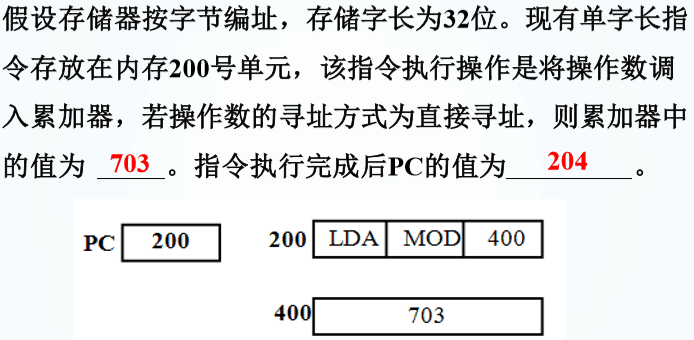
（1）**立即寻址（获取操作数最快）**



（2）**直接寻址**

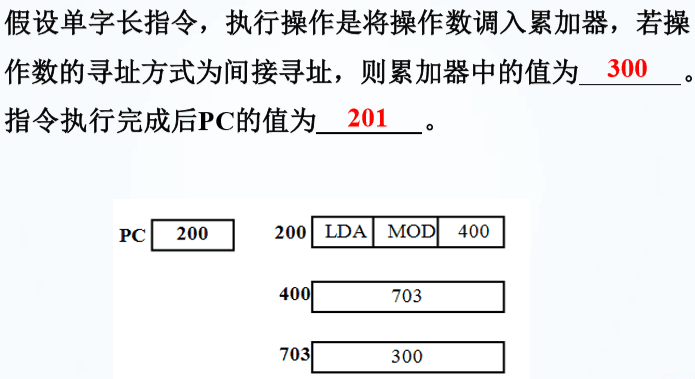
指令中地址码所给出的是操作数的有效地址，采用直接寻址。





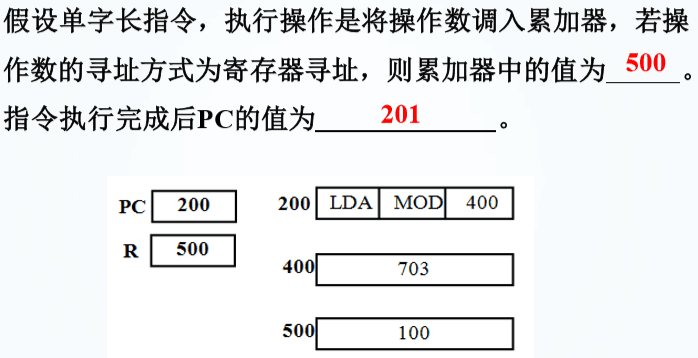
（3）**间接寻址**

有效地址由形式地址间接提供。



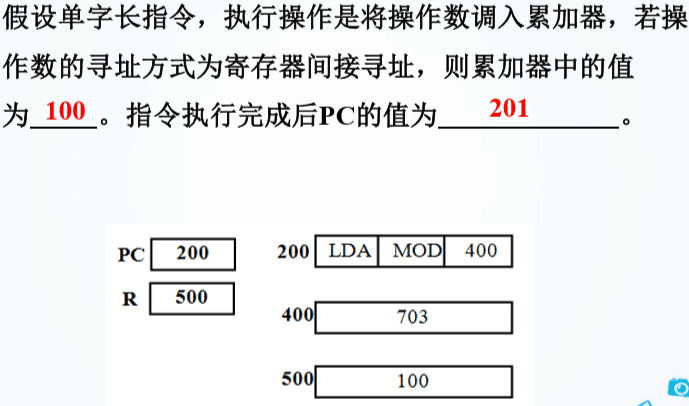
（4）**寄存器寻址**

操作数在寄存器中的寻址 方式称为寄存器直接寻址。

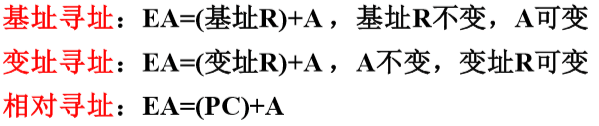


（5）**寄存器间接寻址**

有效地址在寄存器中，操作数在存储器中，执行阶段访存；便于编制循环程序。



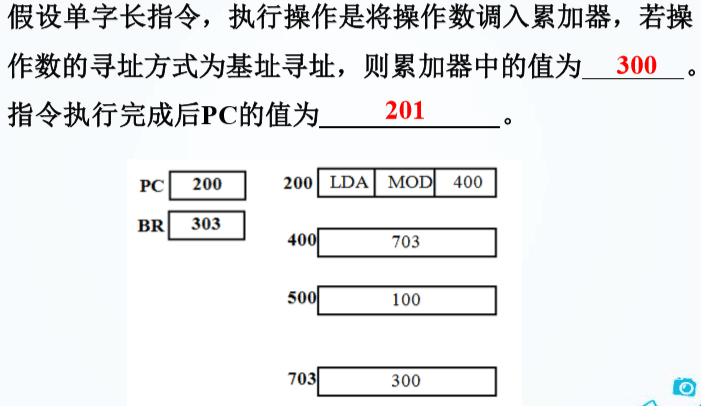
（6）偏移寻址

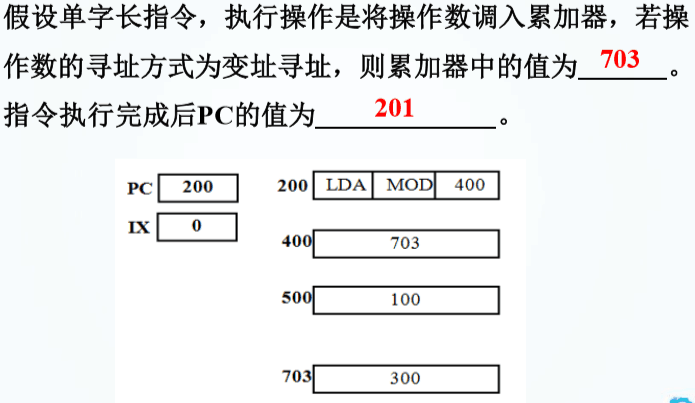


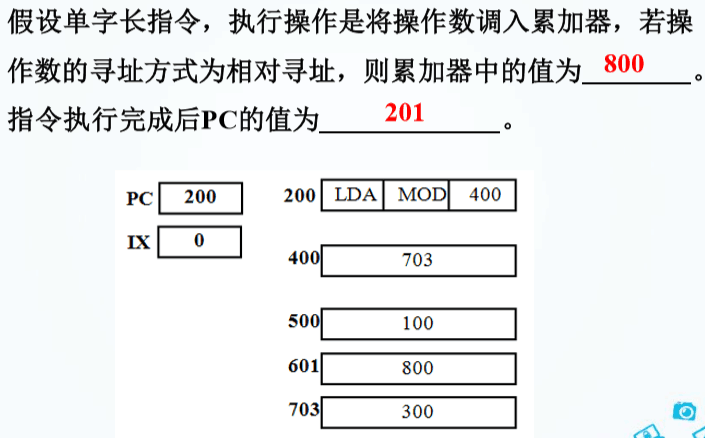
基址寻址的操作数有效地址EA是（BR）+A。

变址寻址的操作数有效地址EA是（X）+A。

相对寻址的操作数有效地址EA是（PC）+A。







（7）堆栈寻址

