

Lecture 28: Input/Output System I

输入/输出系统

磁盘存储器和I/O接口

主要内容

- I/O系统概述
 - I/O系统的性能
 - I/O系统的功能
 - OS在I/O系统中的角色
- I/O设备概述
 - I/O设备的通用模型
- 磁盘存储器
 - 回顾：磁盘存储器的读写原理
 - 回顾：磁盘存储器的性能指标
 - 冗余磁盘阵列 (RAID)
- I/O接口的分类
- I/O控制器的结构
- I/O控制器的职能
- I/O端口的概念
- I/O设备的寻址
- I/O设备、I/O总线、I/O接口、I/O控制器的连接

I/O System的性能

- 两个常用的性能指标:

- **Throughput: I/O bandwidth (吞吐率, 即: I/O带宽):**

- 单位时间内从系统输入/输出多少数据?
 - 单位时间内实现了多少次输入/输出操作?

(服务器更关注)

- **Response time: Latency (响应时间, 即: 等待延迟):**

- 在多长时间完成请求的任务?

(台式机和嵌入式更关注)

- 不同的任务对性能的要求不同:

- 要求吞吐量高的场合:

- 如: 多媒体应用 (音/视频的播放要流畅!)

- 要求响应时间短的场合:

- 如: 事务处理系统 (存/取款的速度要快!)

- 要求吞吐率高且响应时间短的场合:

- **ATM**、文件服务器、**Web**服务器等

I/O System的功能

- 输入/出系统的功能：
 - 解决各种形式信息的输入和输出
即：用户如何将所需的信息（文字、图表、声音、视频等）通过不同的外设输入到计算机中，以及计算机内部处理的结果信息如何通过相应的外设输出给用户
- 要实现上述功能需解决以下一系列的问题：
 - 怎样在**CPU**、主存和外设间建立一个高效信息传输“通路”；
 - 怎样将用户的**I/O**请求转换成设备的命令；
 - 如何对外设进行编址；
 - 怎样使**CPU**方便地寻找到要访问的外设；
 - **I/O**硬件和操作系统如何协调完成主机和外设之间的数据传送等等

以上是本章的主要内容

外设发展与分类

◦ 从交互方式上来分，外设分为：

- 人-机交互设备

- 输入/输出的信息是人可读的
- 如：键盘、鼠标、扫描仪、打印机、显示器等

- 机器可读设备

- 输入/输出的信息是机器可读的，人无法读取
- 如：网络、Modem、D/A、A/D、磁盘、声音输入设备等

◦ 从功能行为来分，外设分为：

- 输入/输出设备（大部分为字符型设备）

- 用于信息的输入/输出
- 输入设备：键盘、鼠标、扫描仪等
- 输出设备：打印机、显示器等

- 外部存储设备（大部分为成块传送设备）

- 用于信息的存储（其输入/出的信息是机器可读的）
- 如：磁盘、磁带、光盘等

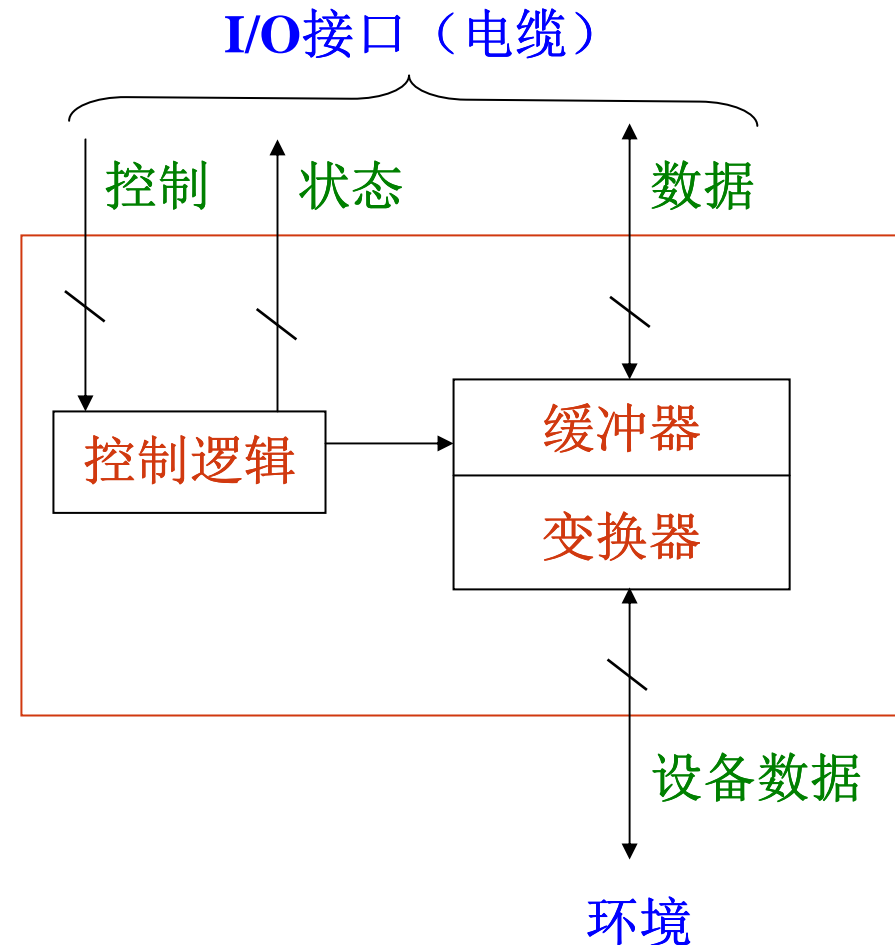
常用外部设备

- 输入设备：
 - 键盘、触摸屏
 - 图形输入设备(鼠标、图形板、跟踪球、操纵杆、光笔)
 - 图像输入设备(摄像机、扫描仪、传真机)
 - 条形码阅读机、光学字符识别设备(OCR)
 - 音、视频输入设备
- 输出设备：
 - 显示器(字符、汉字、图形、图像)
 - 打印设备(点阵、激光、喷墨)
 - 绘图仪(平板式、滚筒式)
 - 声音输出设备
- 其它：
 - 终端设备(键盘+显示器)
 - 外存储器(磁盘、磁带、光盘)

外部设备的通用模型

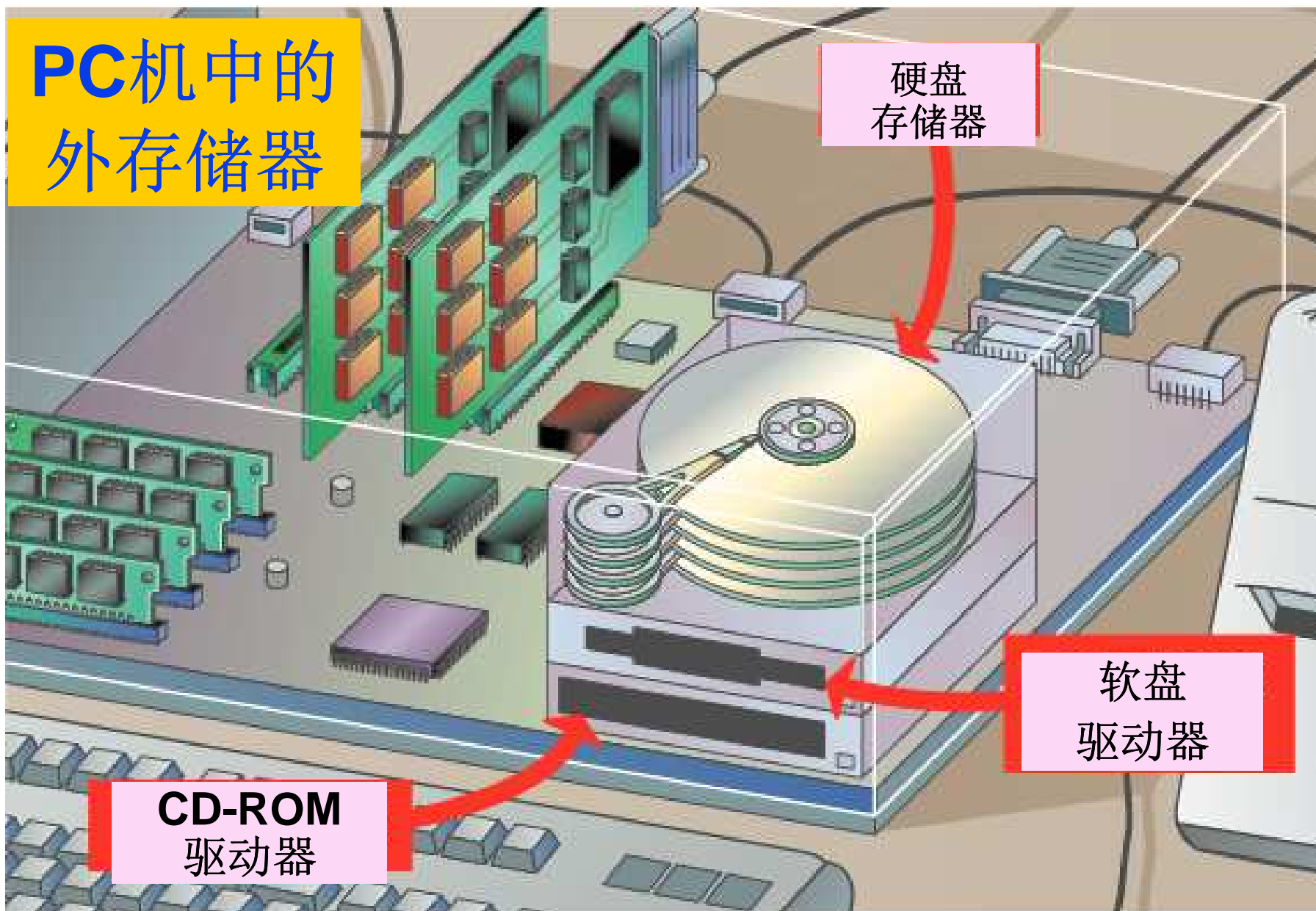
- 通过**电缆**与计算机内部I/O接口进行数据、状态和控制信息的传送
- 控制逻辑**根据控制信息控制设备的操作，并检测设备状态
- 缓冲器**用于保存交换的数据信息
- 变换器**用于在电信号形式（内部数据）和其他形式的设备数据之间进行转换

所有设备都可以抽象成这个通用模型！
设备所用的电缆线中有以下三种信号线：
控制信号、状态信号、数据信号

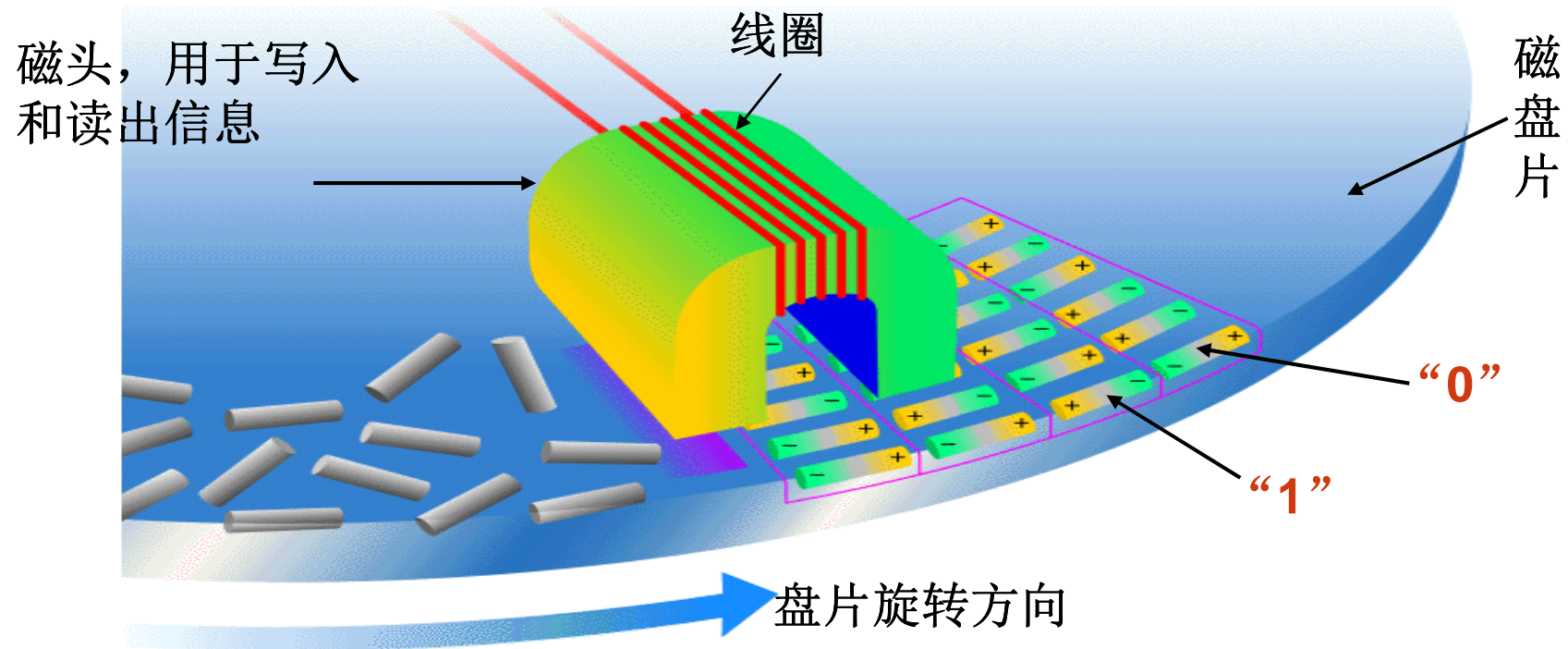


下面以磁盘为例，说明外部设备的工作原理

回顾：PC中的外存储器



回顾：磁盘存储器的信息存储原理



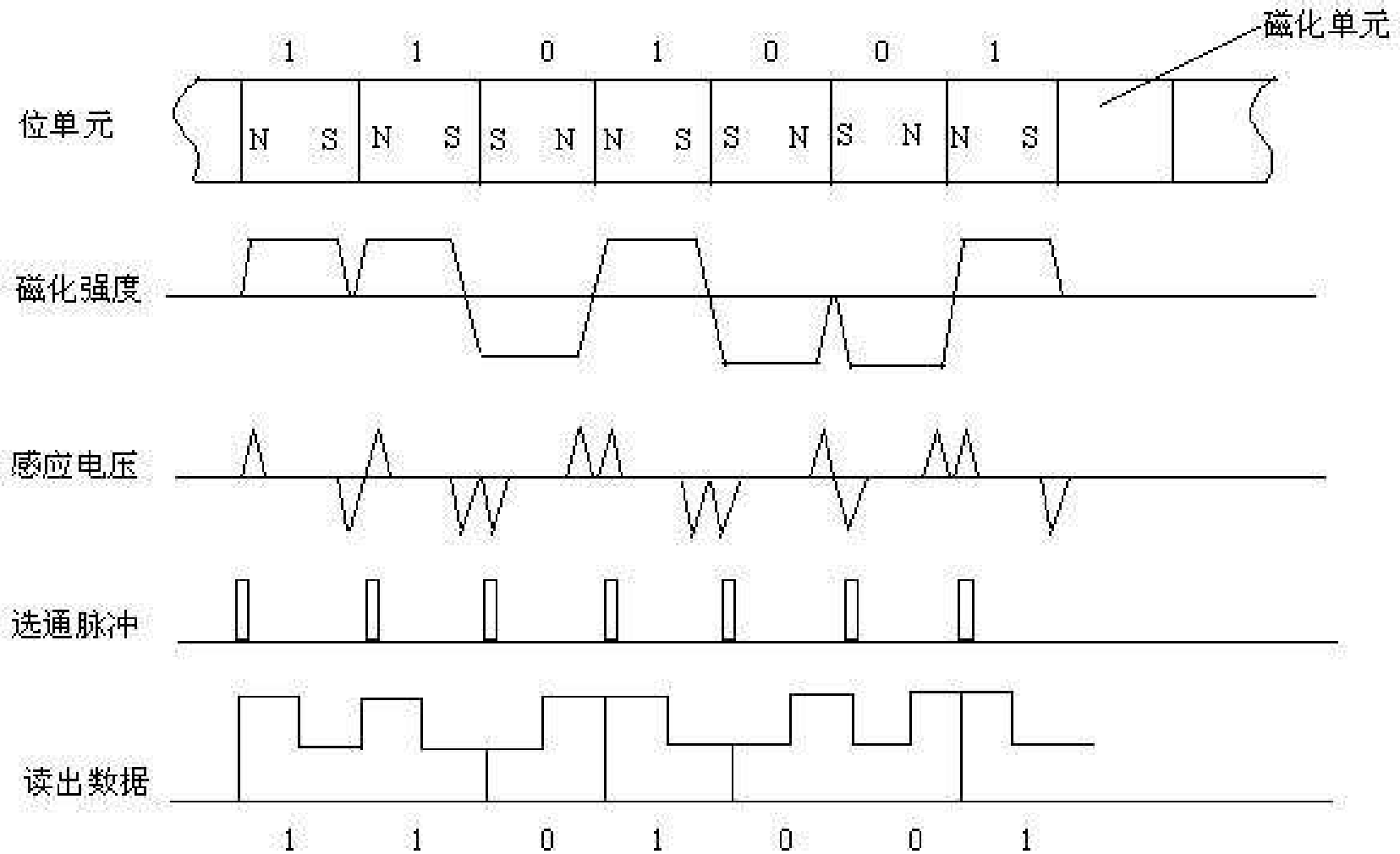
写1: 线圈通以正向电流，使呈**N-S**状态

写0: 线圈通以反向电流，使呈**S-N**状态

} 不同的磁化状态被记录在磁盘表面

读时: 磁头固定不动，载体运动。因为载体上小的磁化单元外部的磁力线通过磁头铁芯形成闭合回路，在铁芯线圈两端得到感应电压。根据不同的极性，可确定读出为**0**或**1**。

磁表面信息读出过程



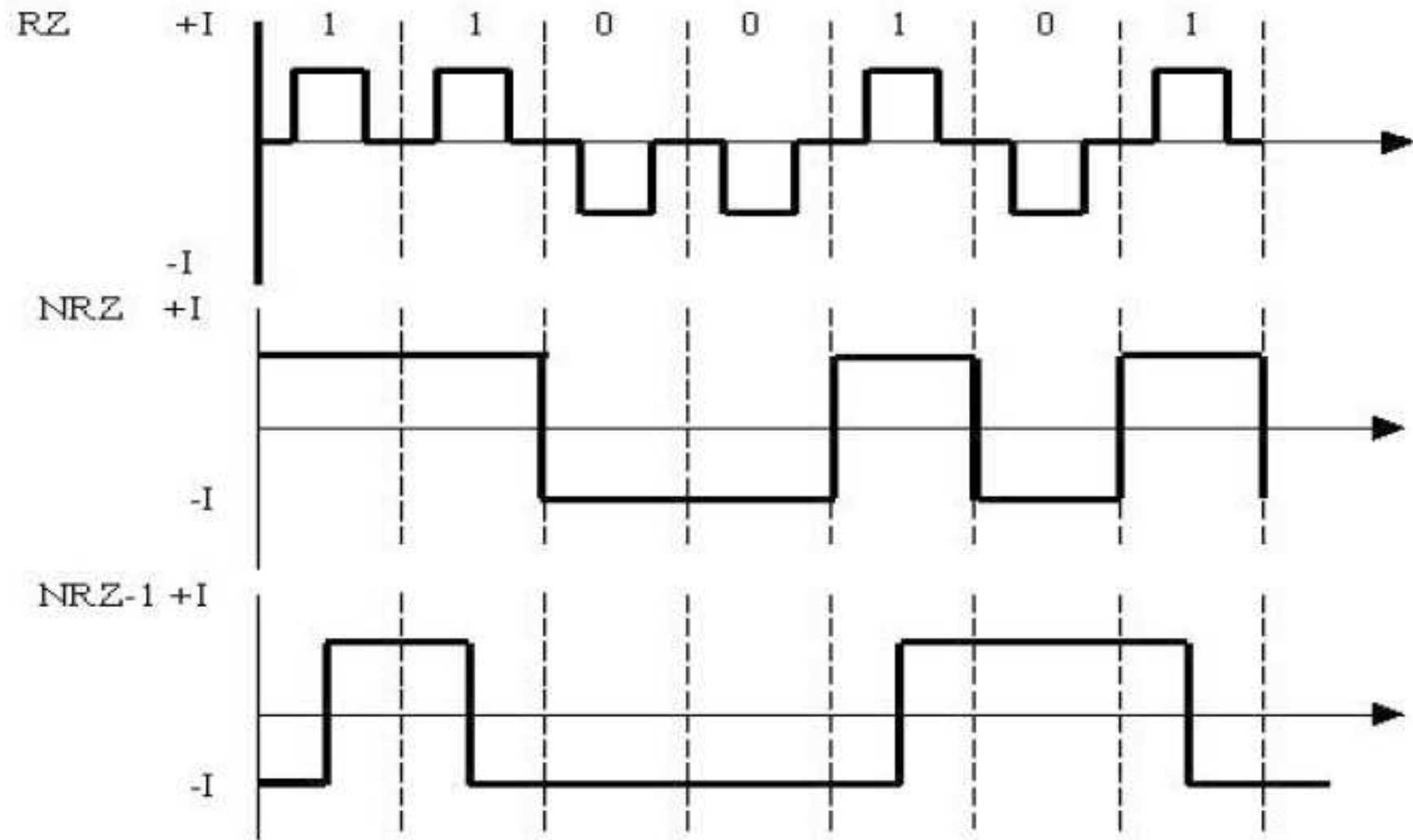
（自学）磁表面记忆原理和记录方式

° 什么叫磁记录方式？有哪几种？

- 对数字信息按一定的规律进行编码，变成相应的写电流序列，通过磁头造成磁层表面上的磁化翻转单元，从而把信息写到磁表面的方式
- 数据记录方式按照写电流波形的极性、频率和相位的不同有：
 - 归零制（RZ）：写1用正脉冲，写0用负脉冲，一位信息写完后，电流总回到零，又叫双向归零制或典型归零制
 - 不归零1制（NRZ-1）：写电流只在写1时改变方向，写0时写电流不变，故又称为“见1就翻”不归零制。各信息位间也无“间隙”，记录密度较高。存1才能读出信号，存0无读出信号，故无自同步能力
 - 调相制（PM）：利用写电流相位的不同实现写1和写0，写0时，先正后负，写1时，先负后正。无论写1还是写0，在一个位信息期间，写电流相位至少有一次改变
 - 调频制（FM）：写0和1时，写电流频率不相同。写1翻转2次（在位单元的前沿和数据位中央各一次），写0翻转1次（在位单元的前沿）
 - 改进调频制（MFM）：逢1在位中央翻转一次；独立一个0不翻转；两个0在位之间翻转一次

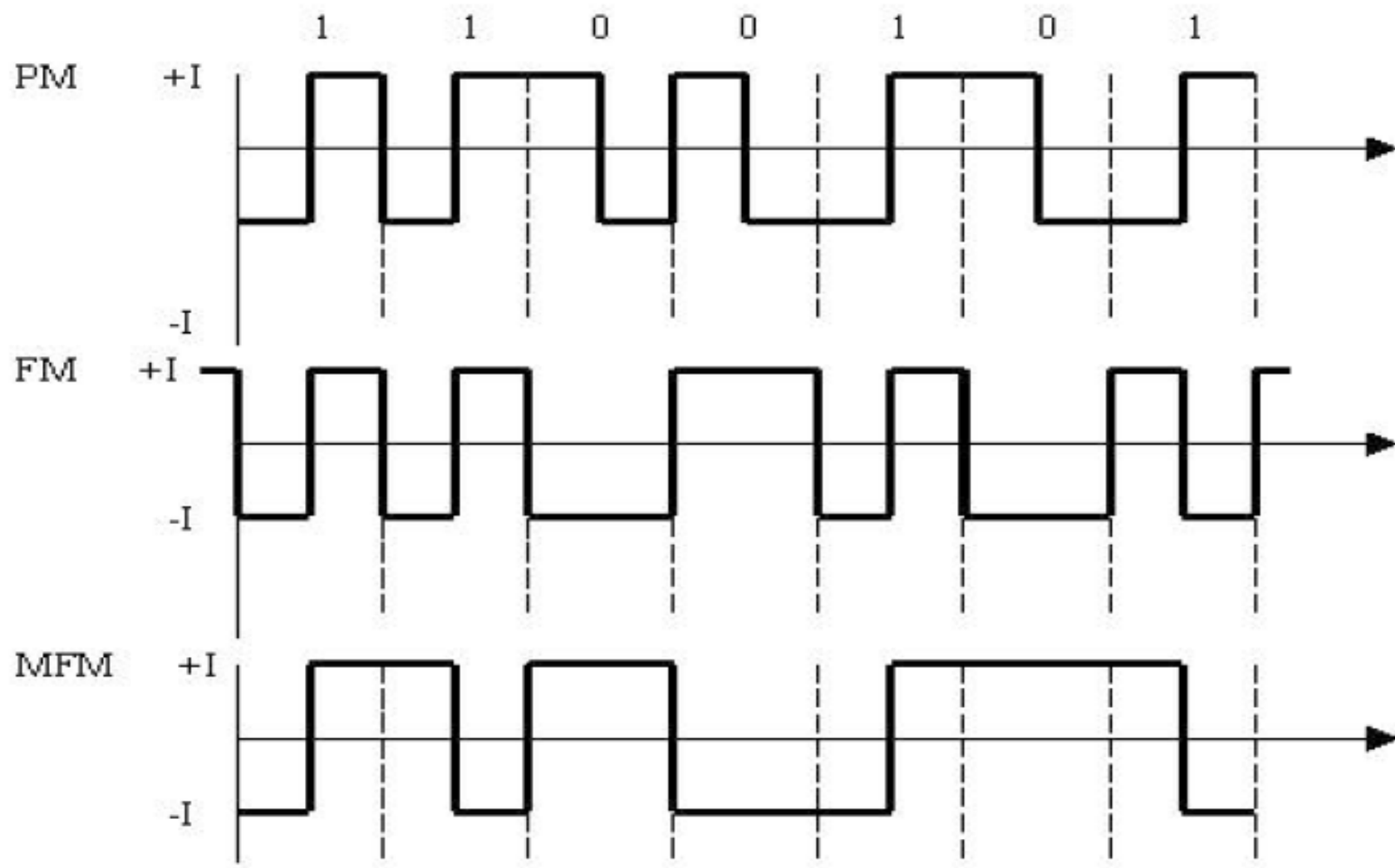
SKIP

(自学) 磁记录方式1



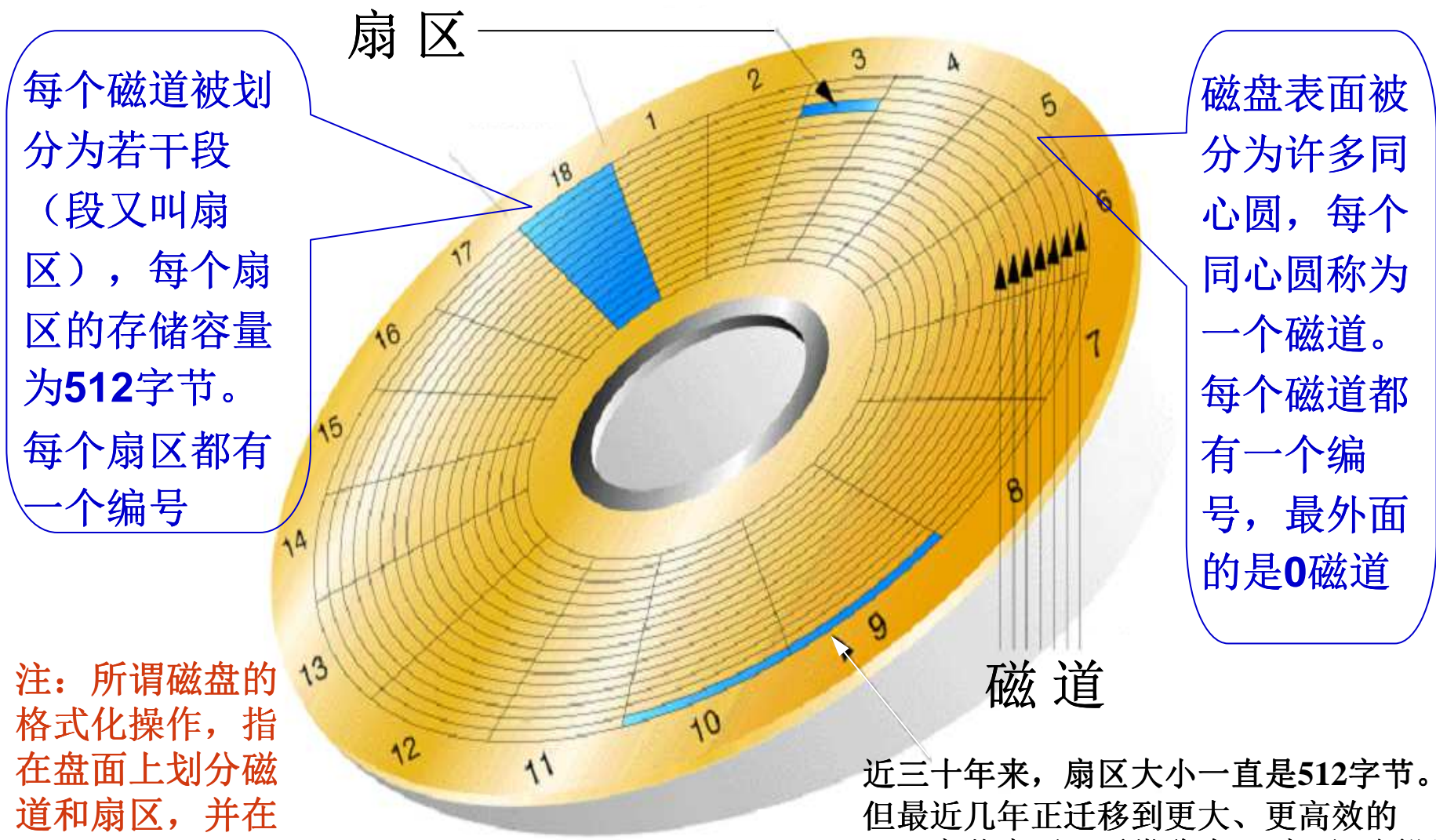
[BACK](#)

(自学) 磁记录方式2



[BACK](#)

磁盘的磁道和扇区

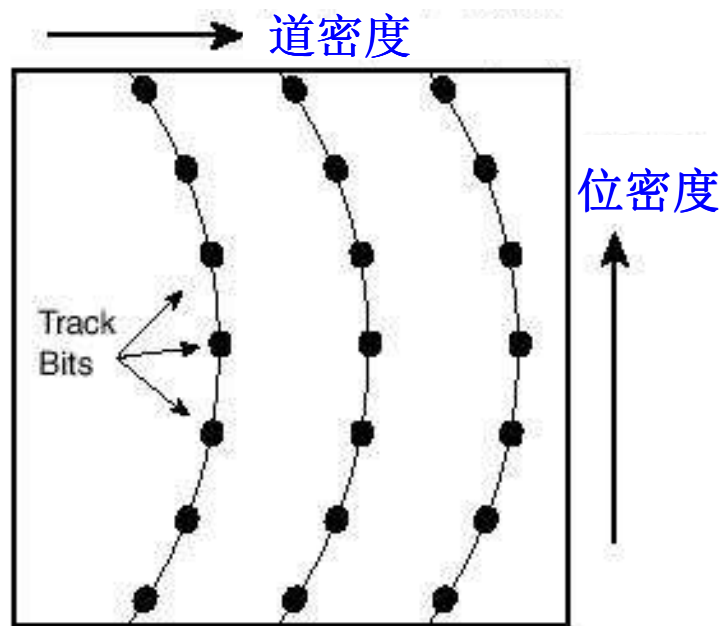


注：所谓磁盘的格式化操作，指在盘面上划分磁道和扇区，并在扇区中填写扇区号等信息的过程

近三十年来，扇区大小一直是512字节。但最近几年正迁移到更大、更高效的4096字节扇区，通常称为4K扇区。国际硬盘设备与材料协会（IDEMA）将之称为高级格式化。

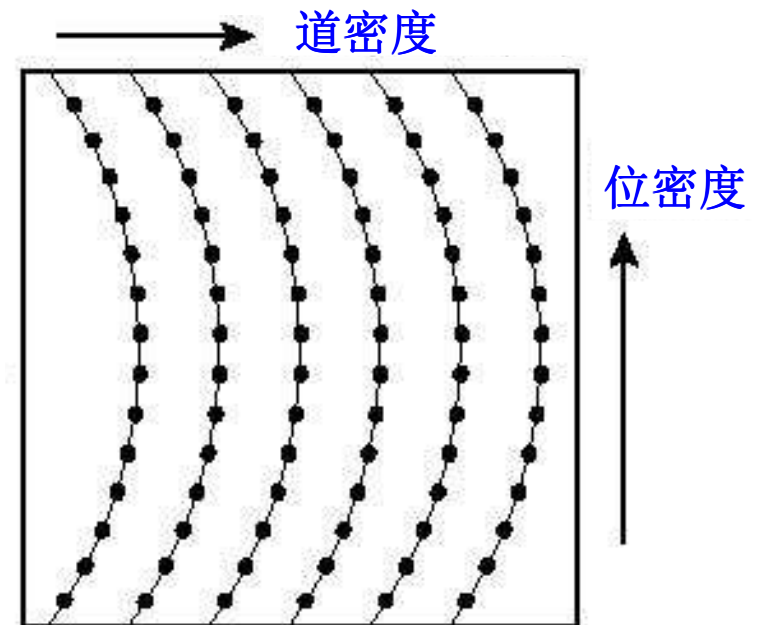
如何增大磁盘片的容量？

- 提高盘片上的信息记录密度！
 - 增加磁道数目——提高磁道密度
 - 增加扇区数目——提高位密度，并采用可变扇区数



低密度存储示意图

早期的磁盘所有磁道上的扇区数相同，所以位数相同，内道上的位密度比外道位密度高



高密度存储示意图

现代磁盘磁道上的位密度相同，所以，外道上的扇区数比内道上扇区数多，使整个磁盘的容量提高

磁盘磁道的格式

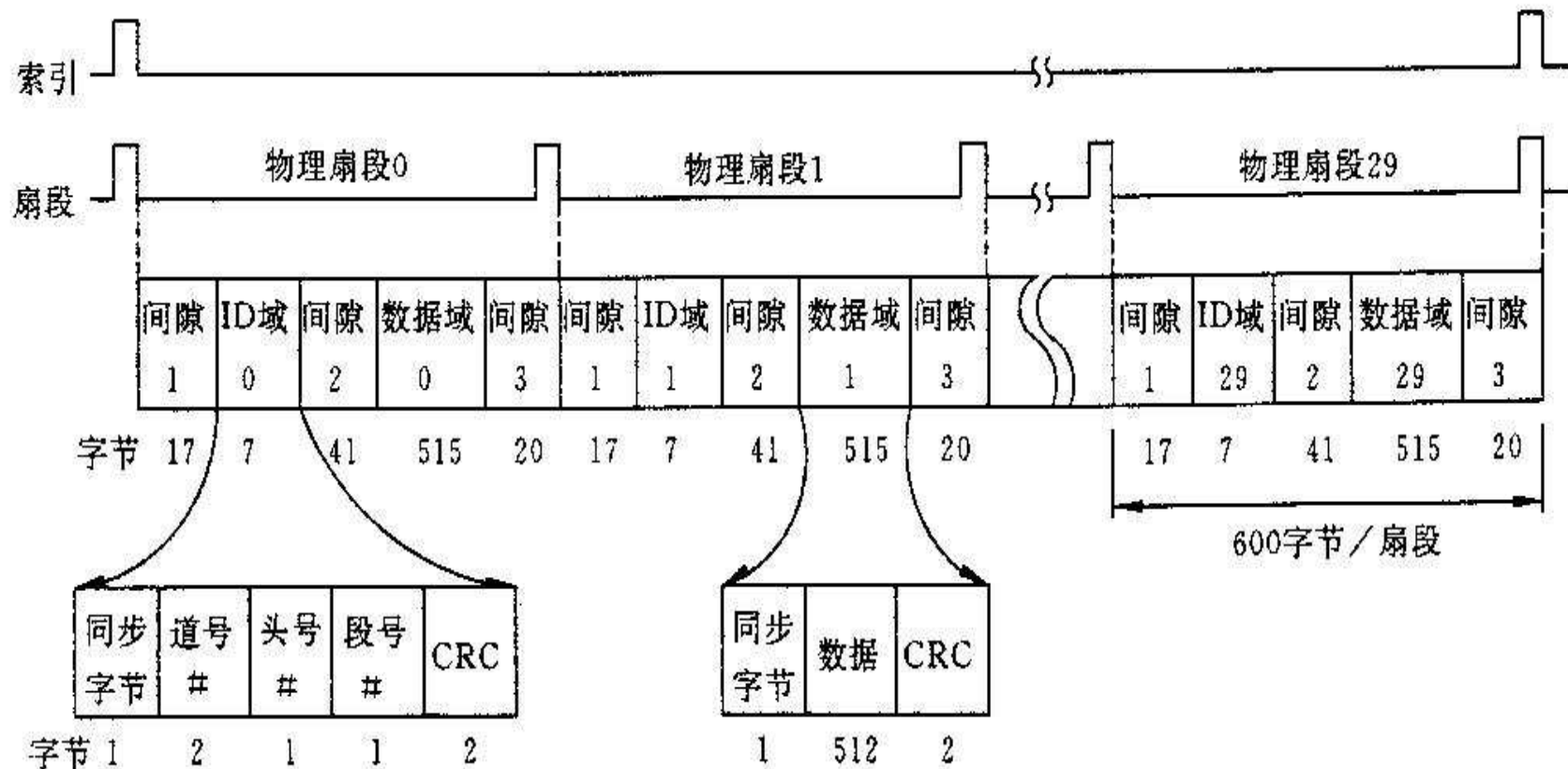
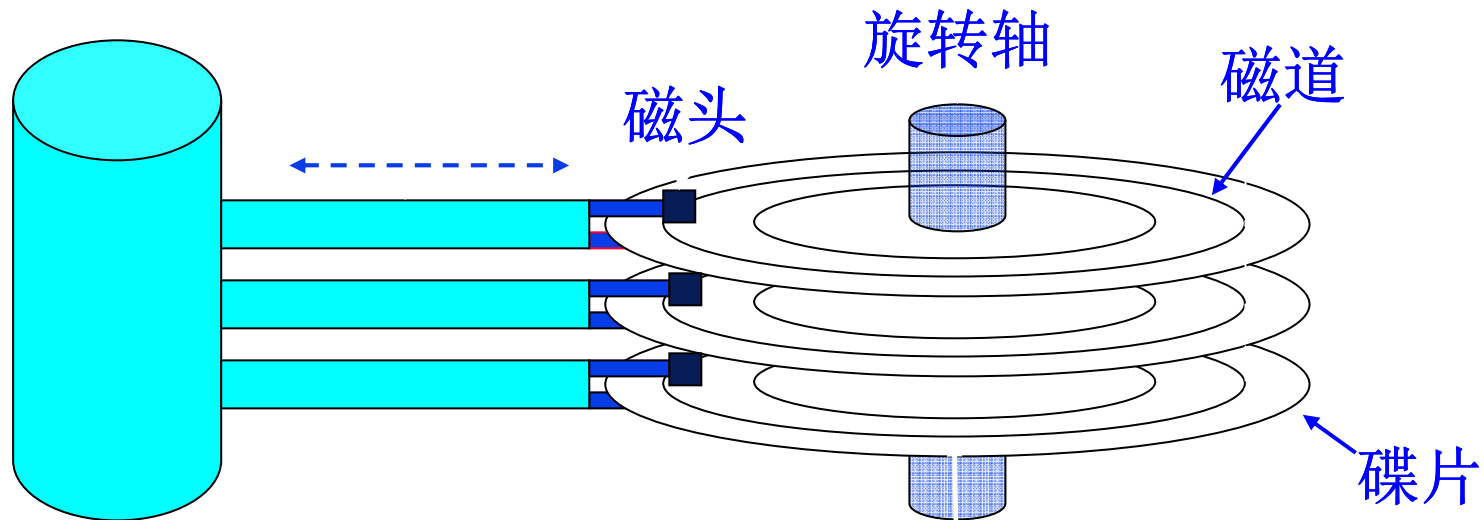


图 5.2 温彻斯特磁盘磁道格式(Seagate ST506)

在此例中，每个磁道包含**30**个固定长度的扇段，每个扇段有**600**个字节 ($17+7+41+515+20=600$)。

平均存取时间



硬盘的操作流程如下：

所有磁头同步寻道（由柱面号控制）→ 选择磁头（由磁头号控制）→
被选中的磁头等待扇区到达磁头下方（由扇区号控制）→ 读写该扇区中的数据

◦ 磁盘上的信息以扇区为单位进行读写，平均存取时间为：

$T = \text{寻道时间} + \text{旋转等待时间} + \text{数据传输时间}$ （忽略不计）

- **寻道时间**——磁头寻找到指定磁道所需时间(大约**5ms**)
- **旋转等待时间**——指定扇区旋转到磁头下方所需要的时间(大约**4~6ms**)（转速：**4200 / 5400 / 7200 / 10000rpm**）
- **数据传输时间**——(大约**0.01ms / 扇区**)

磁盘响应时间计算举例

- 假定每个扇区512字节，磁盘转速为5400 RPM，广告声称寻道时间为12 ms，数据传输率为4 MB/s，磁盘控制器开销为1 ms，不考虑排队时间，则磁盘响应时间为多少？

$$\begin{aligned} \text{Disk Response Time} &= \text{Seek time} + \text{Rotational Latency} + \text{Transfer time} \\ &\quad + \text{Controller Time} + \text{Queuing Delay} \\ &= 12 \text{ ms} + 0.5 / 5400 \text{ RPM} + 0.5 \text{ KB} / 4 \text{ MB/s} + 1 \text{ ms} + 0 \\ &= 12 \text{ ms} + 0.5 / 90 \text{ RPS} + 0.125 / 1024 \text{ s} + 1 \text{ ms} + 0 \\ &= 12 \text{ ms} + 5.5 \text{ ms} + 0.1 \text{ ms} + 1 \text{ ms} + 0 \text{ ms} \\ &= 18.6 \text{ ms} \end{aligned}$$

如果实际的寻道时间只有1/3的话，则为10.6ms，这样旋转等待时间就占了近50%!

$$12/3 + 5.5 + 0.1 + 1 = 10.6 \text{ms}$$

所以，磁盘转速非常重要!

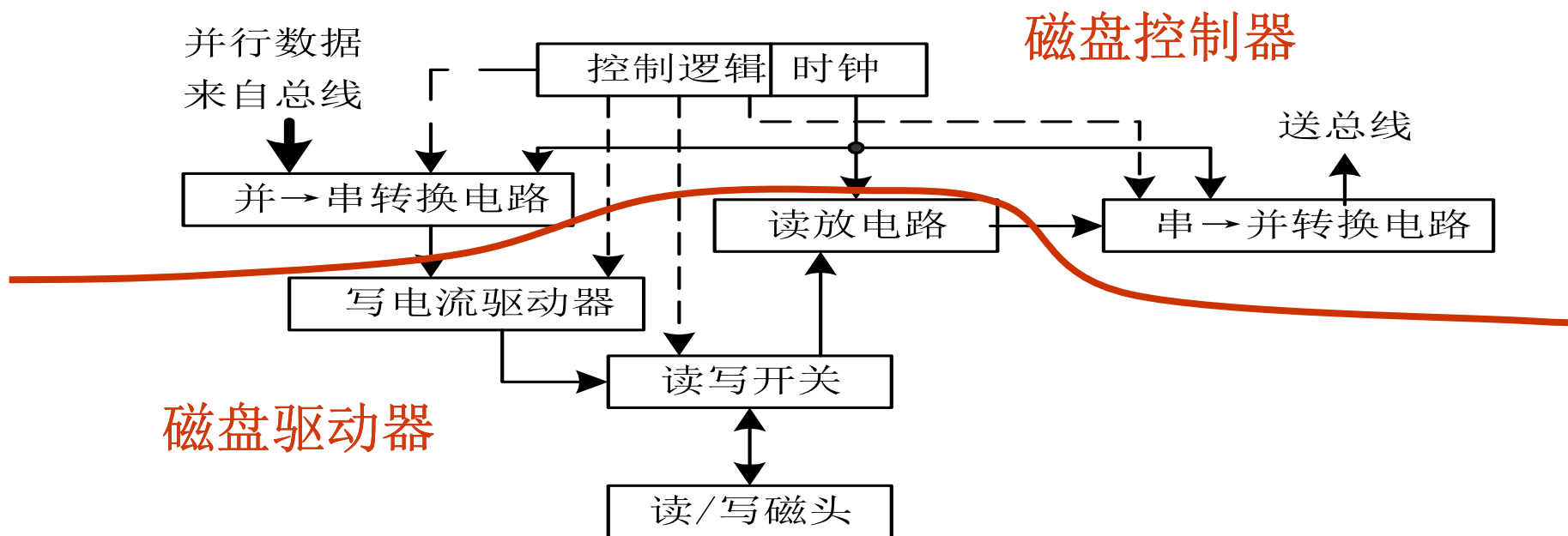
为什么实际的寻道时间只有1/3?

访问局部性使得每次磁盘访问大多在局部磁道!

硬盘存储器的组成

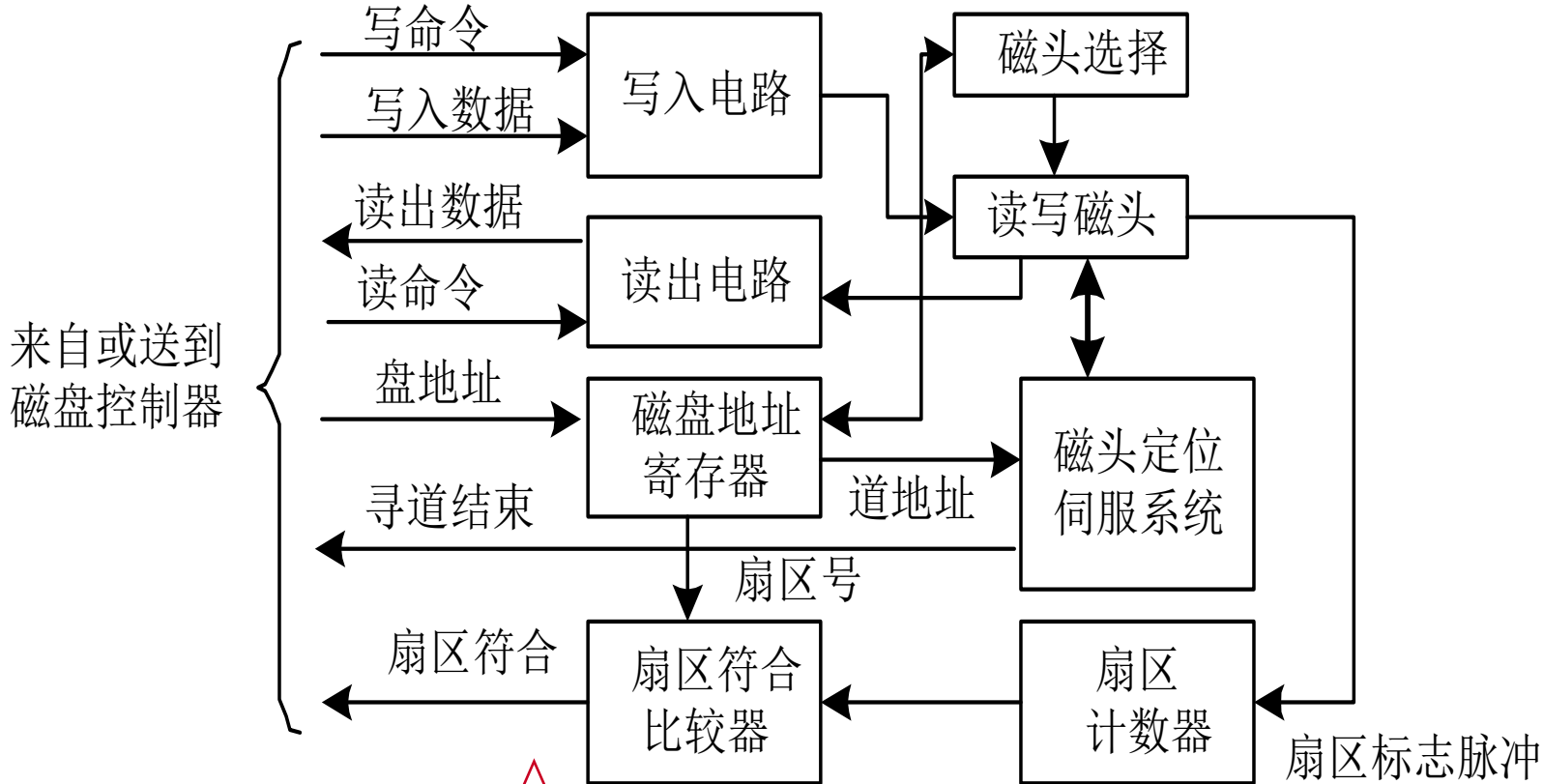
硬盘存储器的基本组成

- 磁记录介质：用来保存信息
- 磁盘驱动器：包括读写电路、读\写转换开关、读写磁头与磁头定位伺服系统等
- 磁盘控制器：包括控制逻辑、时序电路、“并→串”转换和“串→并”转换电路等。
(用于连接主机与盘驱动器)



硬盘存储器的逻辑结构

硬盘驱动器的逻辑结构



与磁盘控制器之间的接口

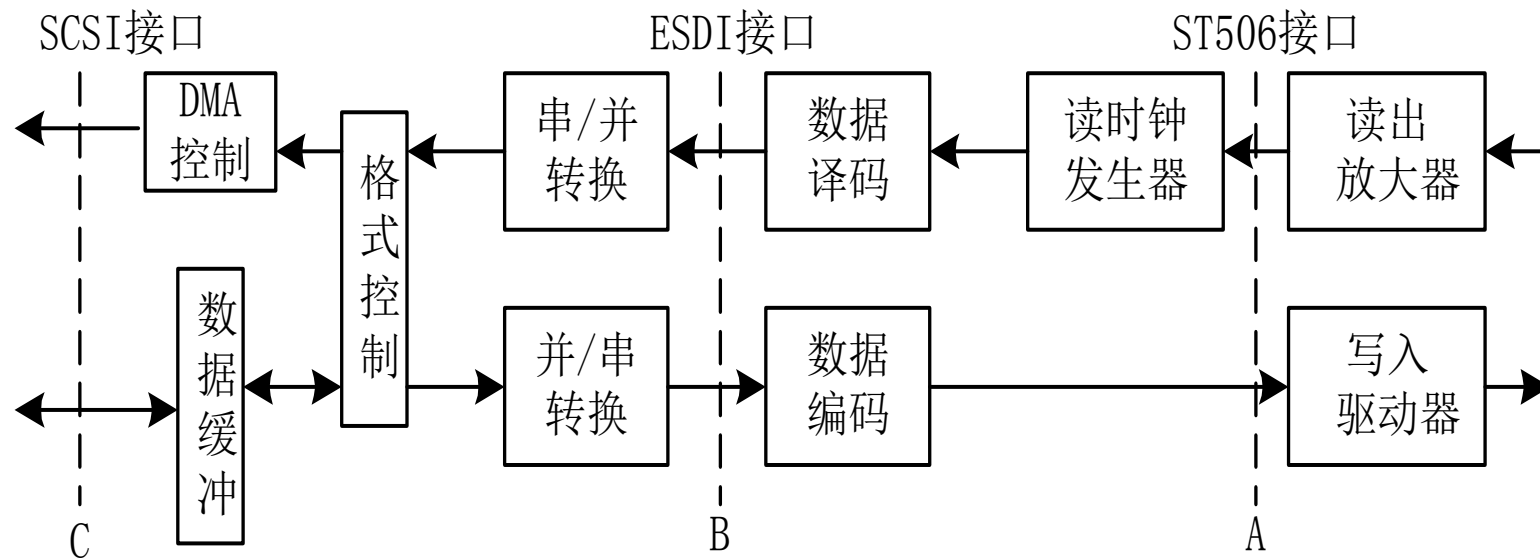


硬盘控制器的逻辑结构

- 硬盘控制器是主机与磁盘驱动器之间的接口
- 硬盘控制器与磁盘驱动器之间并没有明确的界线
(可以在 A点 / B点 / C点)

通过总线与主机连接

通过接口电缆与磁盘驱动器连接



服务器大多使用SCSI接口

PC机前几年大多使用IDE(ATA)接口

近两年PC机开始大量使用SATA接口

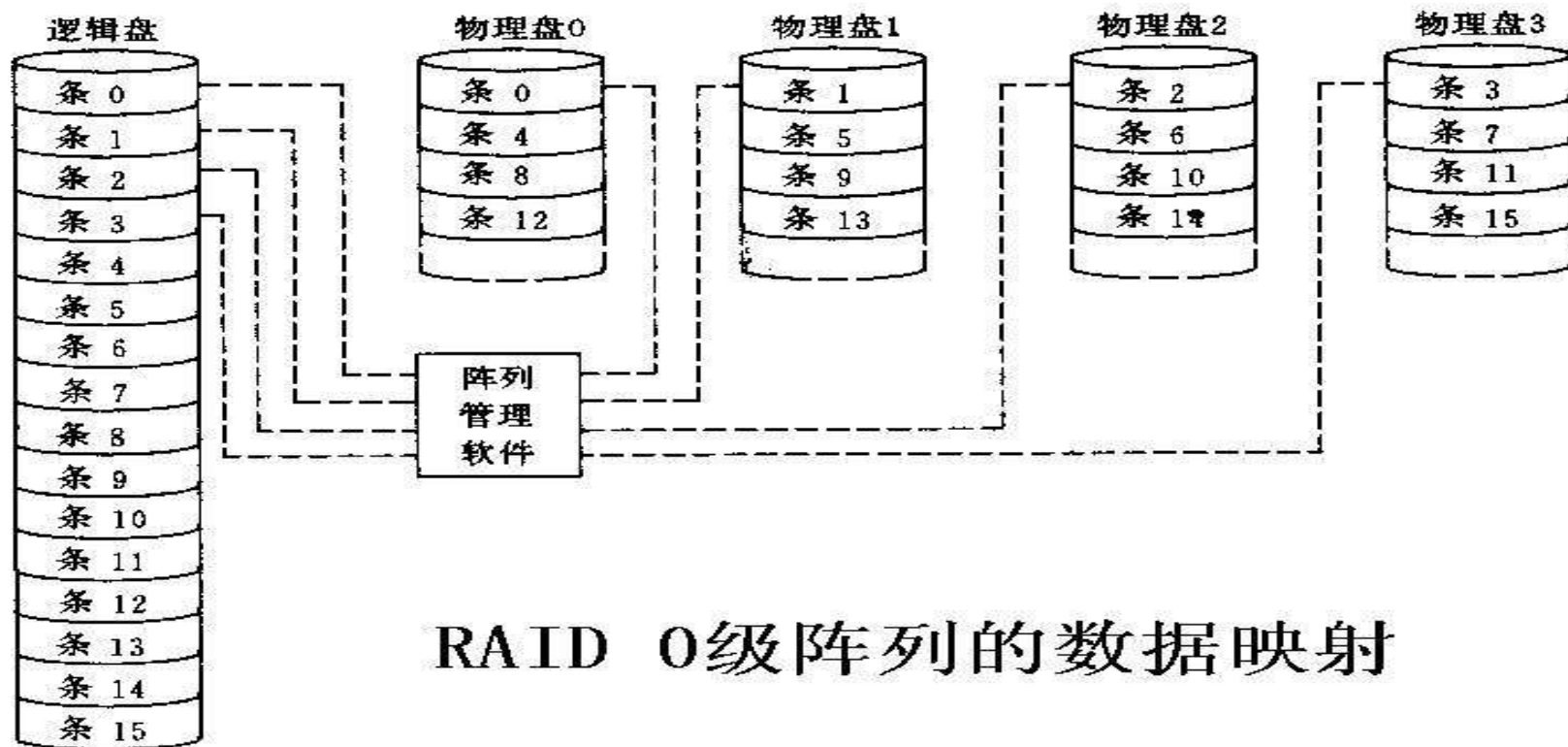
冗余磁盘阵列(RAID)

- 系统总体性能的提高不匹配
 - 处理器和主存性能改进快
 - 辅存性能性能改进慢
- 所用措施: **RAID-Redundant Arrays of Inexpensive Disk** (磁盘冗余阵列)
- **RAID**的基本思想:

将多个独立操作的磁盘按某种方式组织成磁盘阵列(Disk Array), 以增加容量, 利用类似于主存中的多体交叉技术, 将数据存储多个盘体上, 通过使这些盘并行工作来提高数据传输速度, 并用冗余(redundancy)磁盘技术来进行错误恢复(error correction)以提高系统可靠性。
- **RAID**特性:
 - (1) **RAID**是一组物理磁盘驱动器, 在操作系统下被视为一个单个逻辑驱动器。
 - (2) 数据分布在一组物理磁盘上。
 - (3) 冗余磁盘容量用于存储奇偶校验信息, 保证磁盘万一损坏时能恢复数据。
- **RAID**级别
 - 目前已知的**RAID**方案分为8级(0-7级), 以及**RAID10**(结合0和1级)和**RAID30**(结合0和3级)和 **RAID50**(结合0和5级)。但这些级别不是简单地表示层次关系, 而是表示具有上述3个共同特性的不同设计结构。

冗余磁盘阵列 (RAID 0)

- 不遵循特性(3)，所以无冗余。适用于容量和速度要求高的非关键数据存储的场合
 - 与单个大容量磁盘相比有两个优点：
 - (1) 连续分布或大条区交叉分布时，如果两个I/O请求访问不同盘上的数据，则可并行发送。减少了I/O排队时间。具有较快的I/O响应能力。
 - (2) 小条区交叉分布时，同一个I/O请求有可能并行传送其不同的数据块(条区)，因而可达较高的数据传输率。例如，可以用在视频编辑和播放系统中，以快速传输视频流



RAID 0级阵列的数据映射

冗余磁盘阵列 (RAID 1)

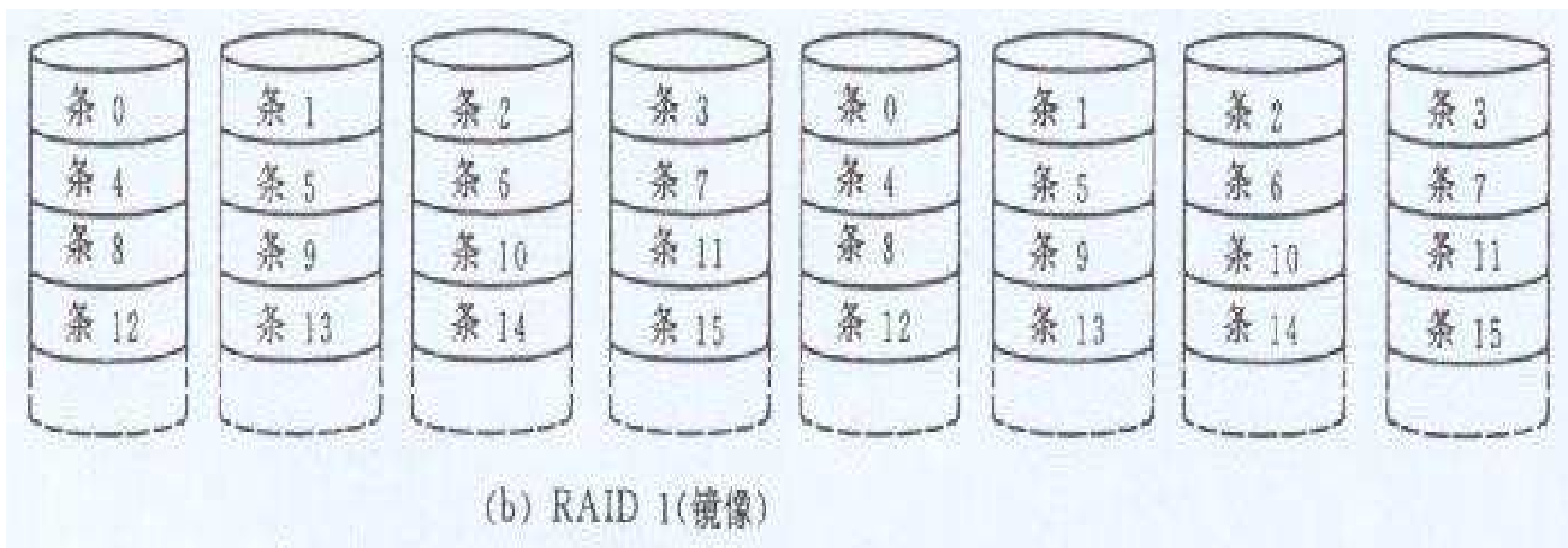
- 镜像盘实现1对1冗余(100% redundancy)

- (1) 读：一个读请求可由其中一个定位时间更少的磁盘提供数据。
- (2) 写：一个写请求对对应的两个磁盘并行更新。故写性能由两次中较慢的一次写来决定，即定位时间更长的那一次。

- (3) 检错：数据恢复很简单。当一个磁盘损坏时，数据仍能从另一个磁盘中读取。

- 特点：可靠性高，但价格昂贵。

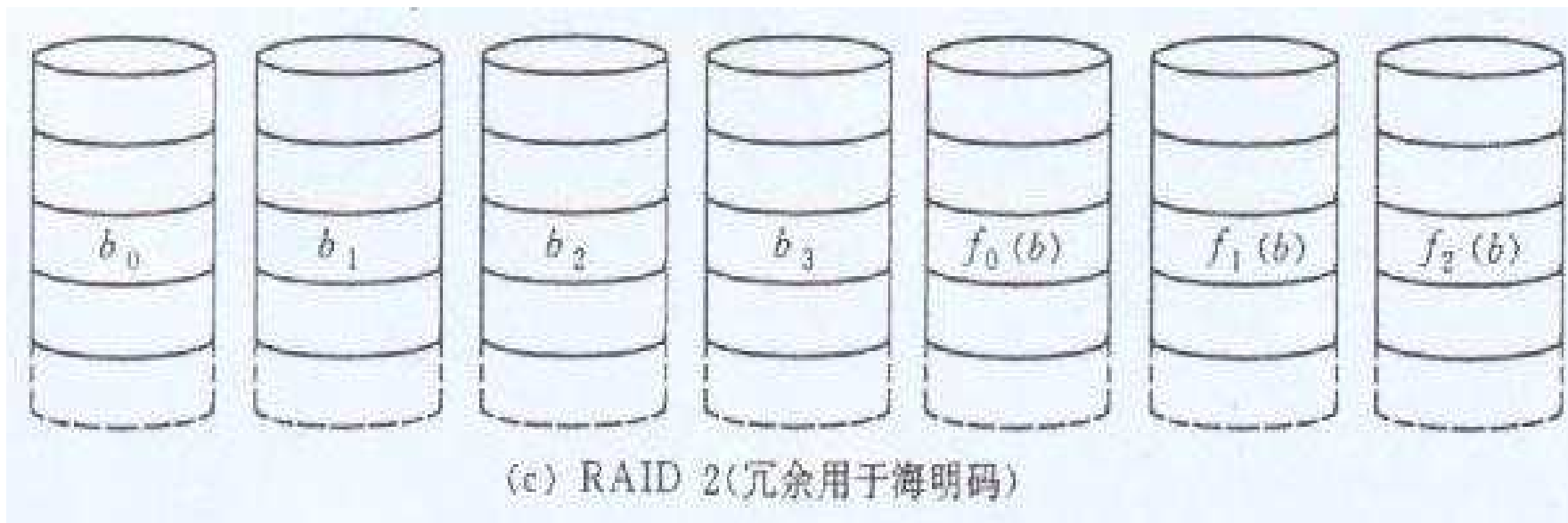
常用于可靠性要求很高的场合，如系统软件的存储，金融、证券等系统。



冗余磁盘阵列 (RAID2)

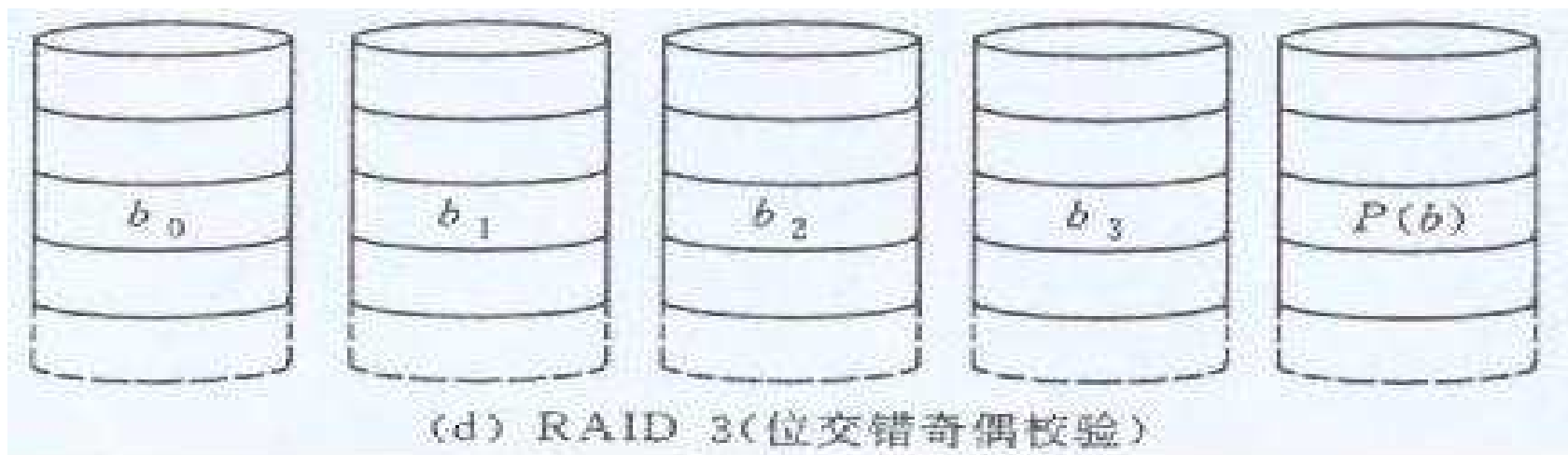
- 用海明校验法生成多个冗余校验盘，实现纠正一位错误、检测两位错误的功能。
- 采用条区交叉分布方式，且条区非常小。这样，可获得较高的数据传输率，但I/O响应时间差。
- 采用海明码，虽然冗余盘的个数比RAID1少，但校验盘与数据盘成正比。所以冗余信息开销还是太大，价格也较贵
- 读操作性能高（多盘并行）。
- 写操作时要同时写数据盘和校验盘。

RAID2已不再使用！ 为什么？



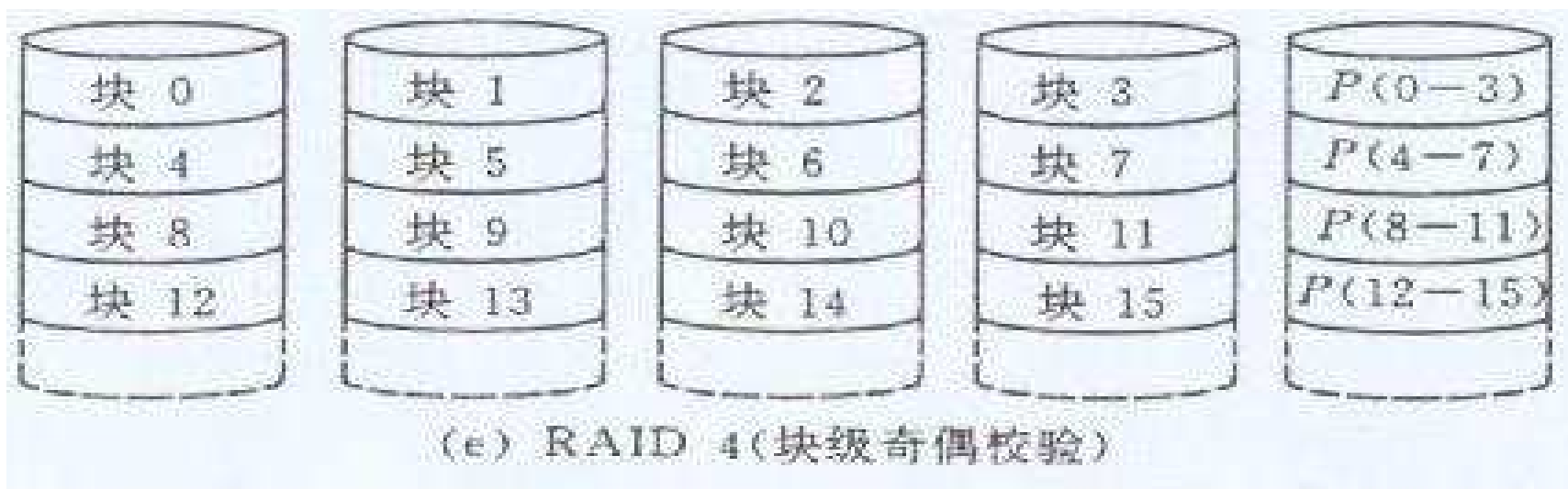
冗余磁盘阵列 (RAID 3)

- 采用奇偶校验法生成单个冗余盘。
- 与RAID 2相同，也采用条区交叉分布方式，并使用小条区。这样，可获得较高的数据传输率，但I/O响应时间差。为什么？
- 用于大容量的 I/O请求的场合，如：图像处理、CAD 系统中。
- 某个磁盘损坏但数据仍有效的情况，称为简化模式。此时损坏的磁盘数据可以通过其它磁盘重新生成。数据重新生成非常简单，这种数据恢复方式同时适用于RAID3、4、5级。



冗余磁盘阵列 (RAID 4)

- 用一个冗余盘存放相应块（较大的数据条区）的奇偶校验位。
- 采用独立存取技术，每个磁盘的操作独立进行，所以可同时响应多个I/O请求。因而它适合于要求I/O响应速度快的场合。
- 对于写操作，校验盘成为I/O瓶颈，因为每次写都要对校验盘进行。
 - 少量写（只涉及个别磁盘）时，有“写损失”，因为一次写操作包含两次读和两次写
 - 大量写（涉及所有磁盘的数据条区）时，则只需直接写入奇偶校验盘和数据盘。因为奇偶校验位可全部用新数据计算得到。而无须读原数据



RAID 3/4/5的数据恢复与RAID4的少量写（自学）

RAID3/4/5的数据恢复操作

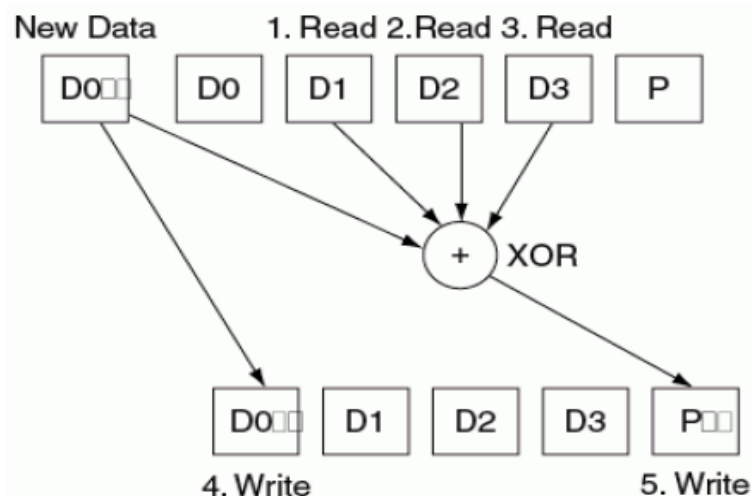
假定考虑一个有5个磁盘的阵列， X_0 到 X_3 保存数据， p 是奇偶校验盘，奇偶校验的第 i 位计算公式如下：

$$p(i) = X_3(i) \oplus X_2(i) \oplus X_1(i) \oplus X_0(i)$$

若磁盘 X_0 损坏，上述等式两边同时异或 $p(i) \oplus X_0(i)$ ，则得到以下等式：

$$X_0(i) = p(i) \oplus X_3(i) \oplus X_2(i) \oplus X_1(i)$$

因此，在阵列中，某一数据磁盘中的任何一个数据条区的内容都能从剩余磁盘的相应条区中重新生成



RAID4的少量写操作

假定考虑一个有5个磁盘的阵列， X_0 到 X_3 保存数据， P 是奇偶校验盘，初始时对每位 i 有下列关系式：

$$p(i) = X_3(i) \oplus X_2(i) \oplus X_1(i) \oplus X_0(i)$$

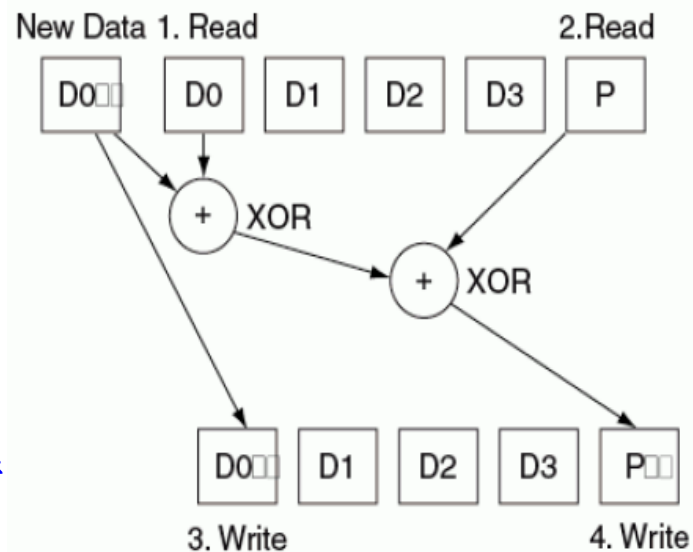
写操作后，可能改变的位数表示如下：

$$P'(i) = X_3(i) \oplus X_2(i) \oplus X_1(i) \oplus X'_0(i)$$

$$= X_3(i) \oplus X_2(i) \oplus X_1(i) \oplus X'_0(i) \oplus X_0(i) \oplus X_0(i)$$

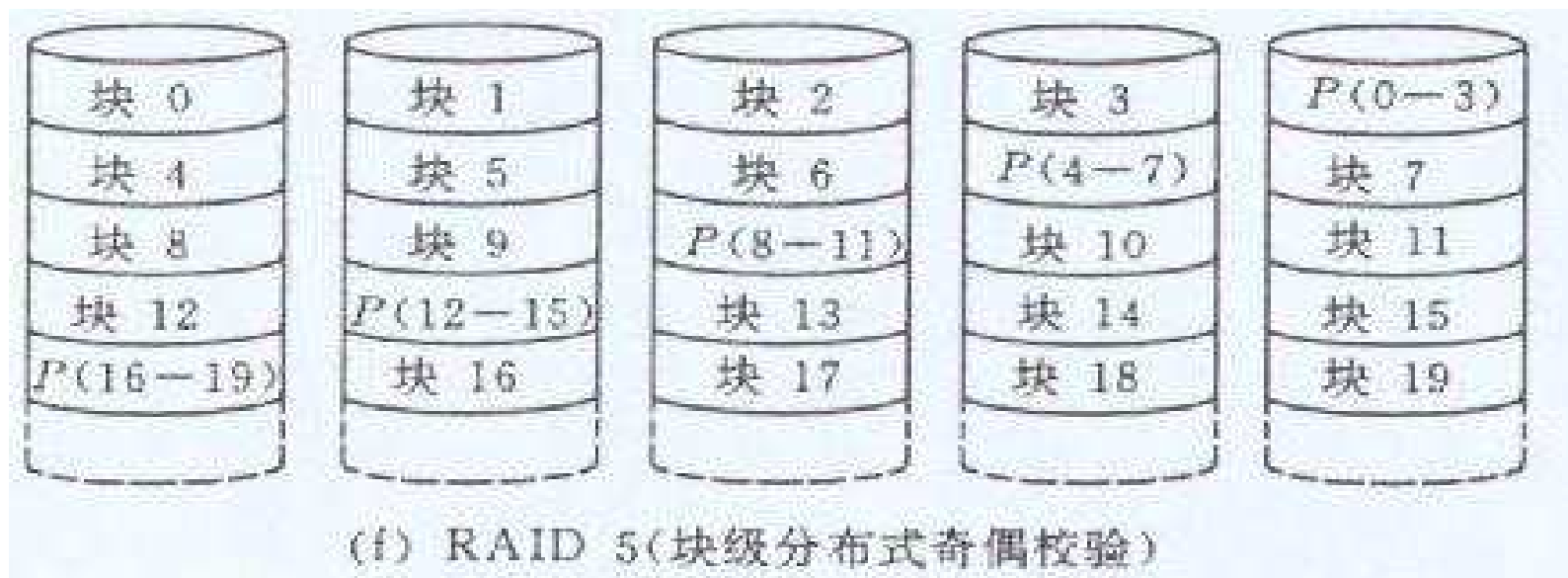
化简后得到： $p'(i) = p(i) \oplus X_0(i) \oplus X'_0(i)$

由此可见，要更新一个 $X_0(i)$ ，必须先读 $p(i)$ 和 $X_0(i)$ ，然后写 $X'_0(i)$ 和 $p'(i)$



冗余磁盘阵列 (RAID 5)

- 与RAID 4组织方式类似，只是奇偶校验块分布在各个磁盘中，所以所有磁盘地位等价，这样可提高容错性，并且避免了使用专门校验盘时潜在的I/O瓶颈。
- 与RAID 4一样，采用独立的存取技术，因而有较高的I/O响应速度。
- 小数据量的操作可以多个磁盘并行操作
- 成本不高但效率高，所以被广泛使用



P块为校验块，分布在不同的磁盘中

冗余磁盘阵列 (RAID 6)

- 冗余信息均匀分布在所有磁盘上，而数据仍以块交叉方式存放
- 双维块交叉奇偶校验独立存取盘阵列，容许双盘出错
- 它是对RAID 5的扩展，主要是用于要求数据绝对不能出错的场合
- 由于引入了第二种奇偶校验值，对控制器的设计变得十分复杂，写入速度也比较慢，用于计算奇偶校验值和验证数据正确性所花费的时间比较多
- RAID 6级以增大开销的代价保证了高度可靠性



RAID 6 级双维块交叉奇偶校验独立存取盘阵列示意图

P0代表第0条区的奇偶校验值，而**PA**代表数据块**A**的奇偶校验值

冗余磁盘阵列 (RAID 7)

- 带**Cache**的盘阵列
- 在**RAID6**的基础上，采用**Cache**技术使传输率和响应速度都有较大提高
- **Cache**分块大小和磁盘阵列中数据分块大小相同，一一对应
- 有两个独立的**Cache**，双工运行。在写入时将数据同时分别写入两个独立的**Cache**，这样即使其中有一个**Cache**出故障，数据也不会丢失
- 写入磁盘阵列以前，先写入**Cache**中。同一磁道的信息在一次操作中完成
- 读出时，先从**Cache**中读出，**Cache**中没有要读的信息时，才从**RAID**中读

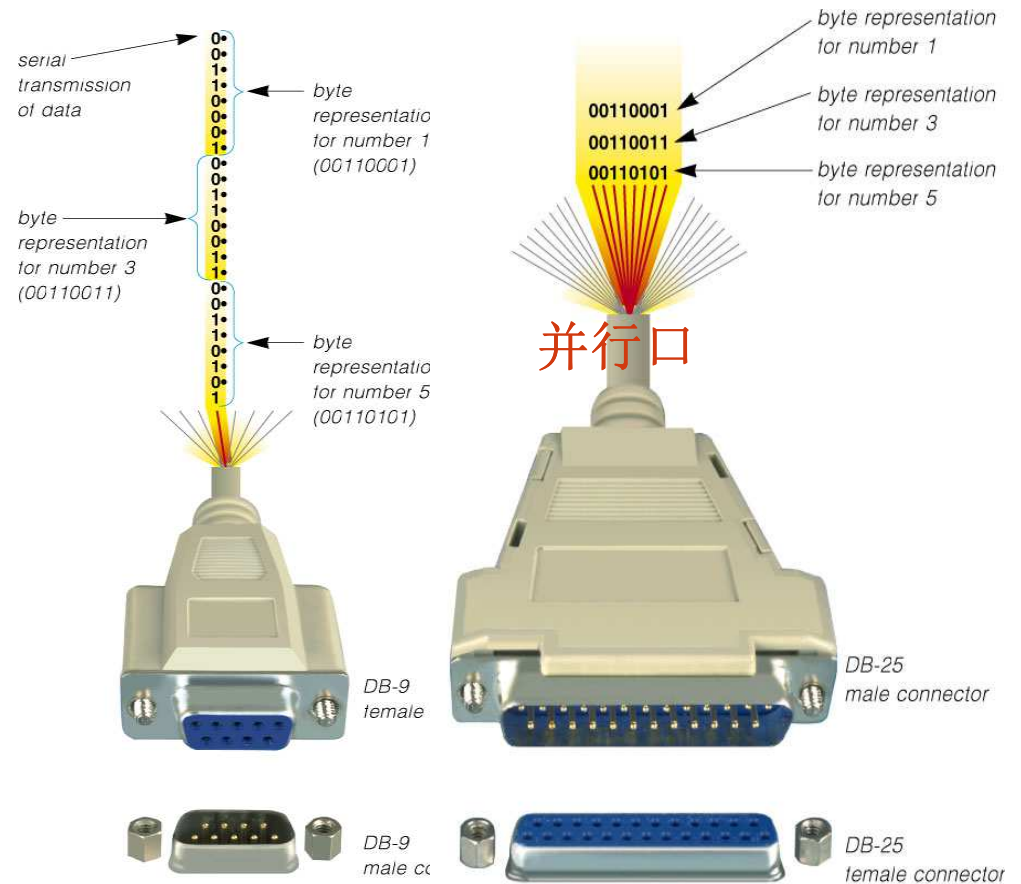
Cache和**RAID**技术结合，弥补了**RAID**的不足（如：分块的写请求响应性能差等），从而以高效、快速、大容量、高可靠性，以及灵活方便的存储系统提供给用户

回顾：关于I/O接口

- I/O接口：I/O设备与I/O控制器之间的连接器
包括：插头 / 插座的形式、通讯规程和电器特性等

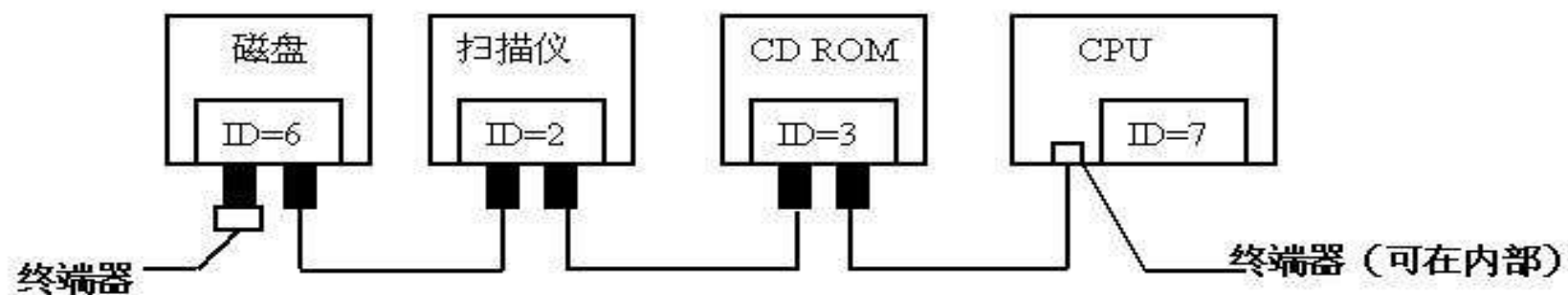
- 分类：

- 从数据传输方式来分：
 - 串行（一次只传输1位）
 - 并行（多位一起进行传输）
- 从是否能连接多个设备来分：
 - 总线式（可连接多个设备）
 - 独占式（只能连接1个设备）
- 从是否符合标准来分：
 - 标准接口（通用接口）
 - 专用接口（专用接口）
- 按功能选择的灵活性来分：
 - 可编程接口
 - 不可编程接口

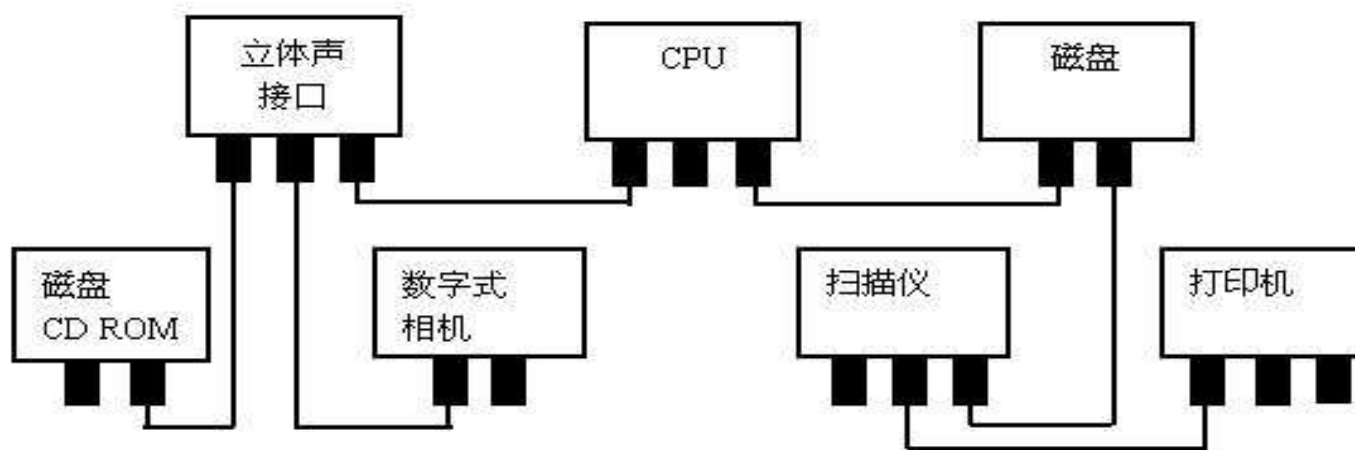


串行口

总线式I/O接口

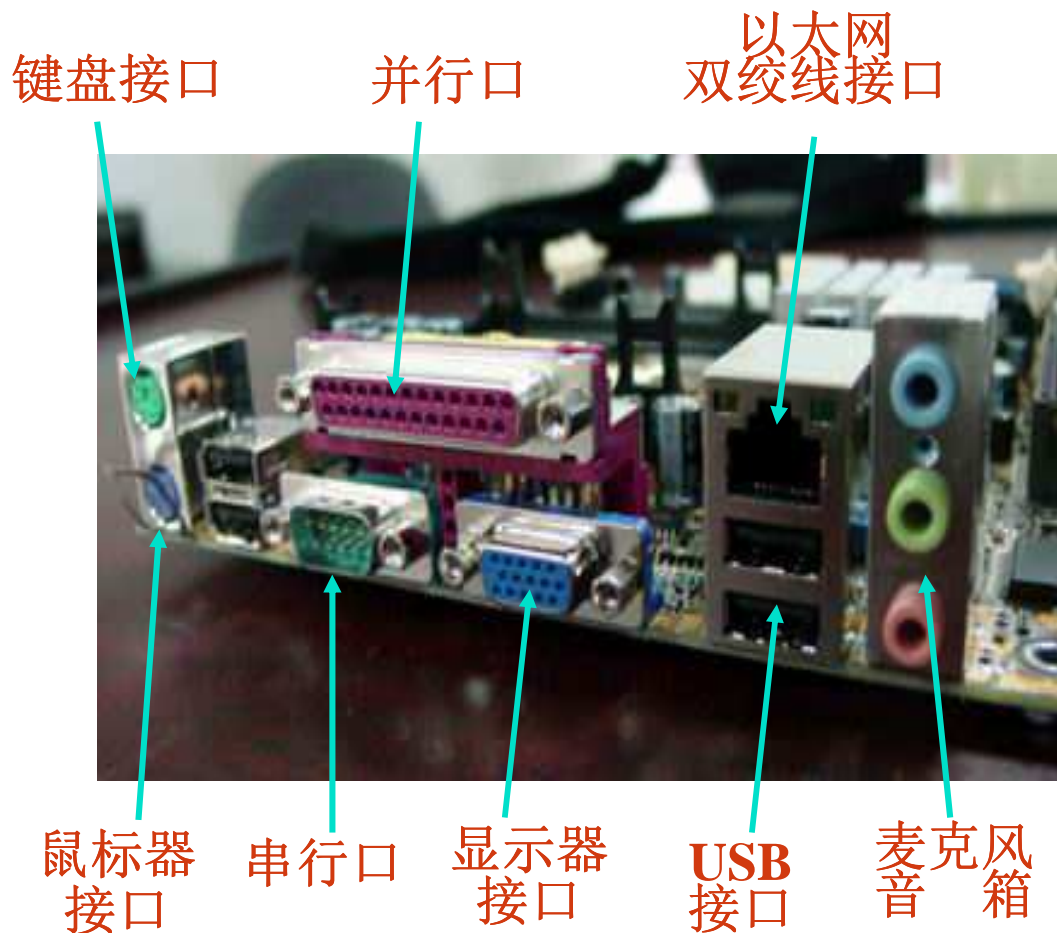


(a) SCSI 配置举例

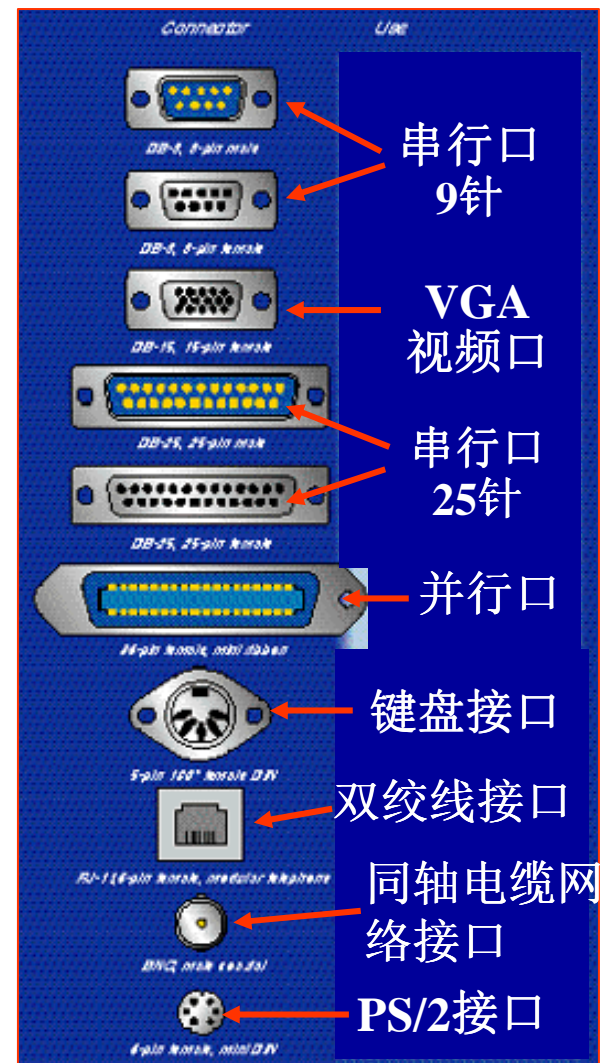


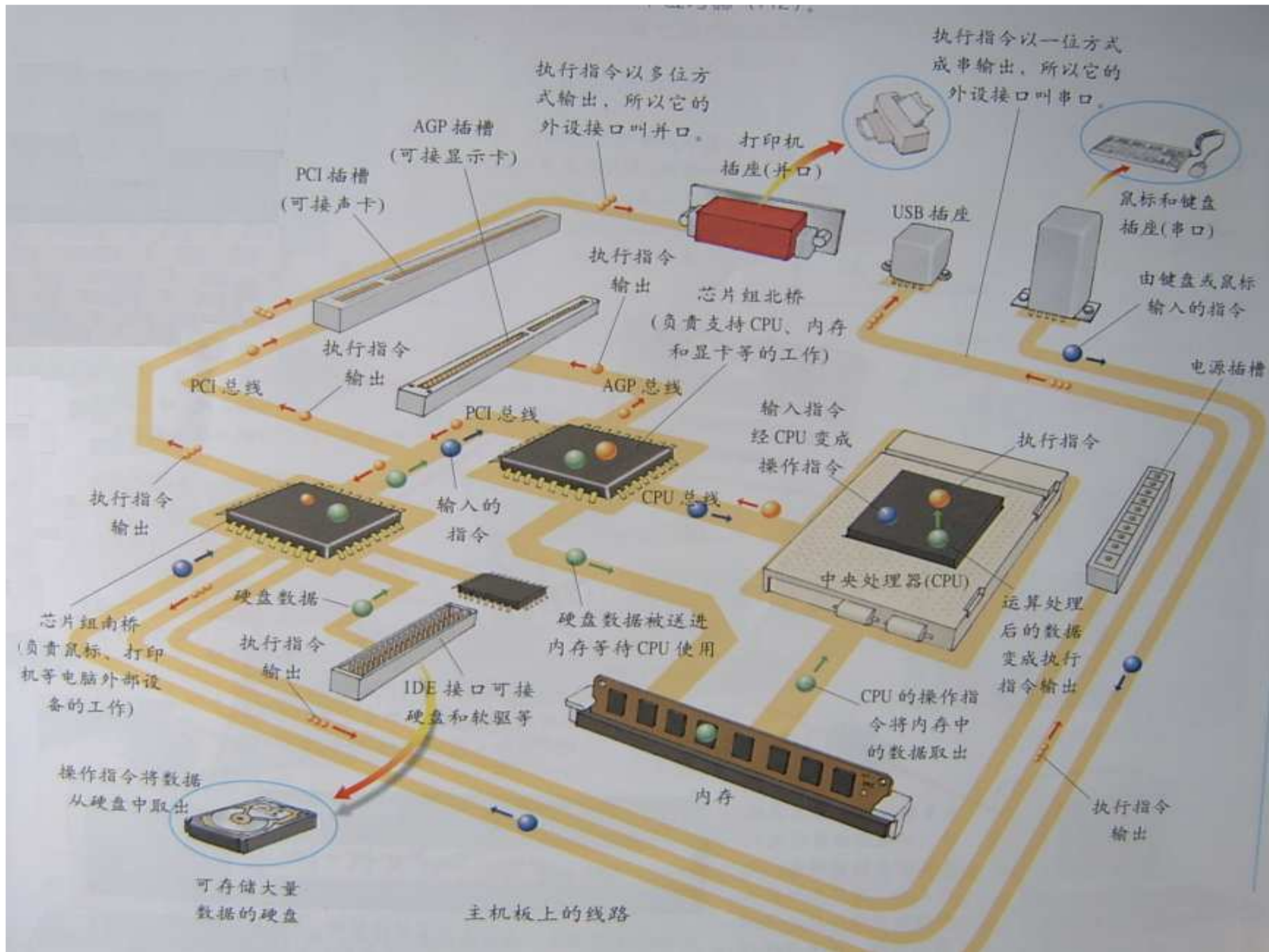
(b) P1394 配置举例

回顾： I/O设备接口

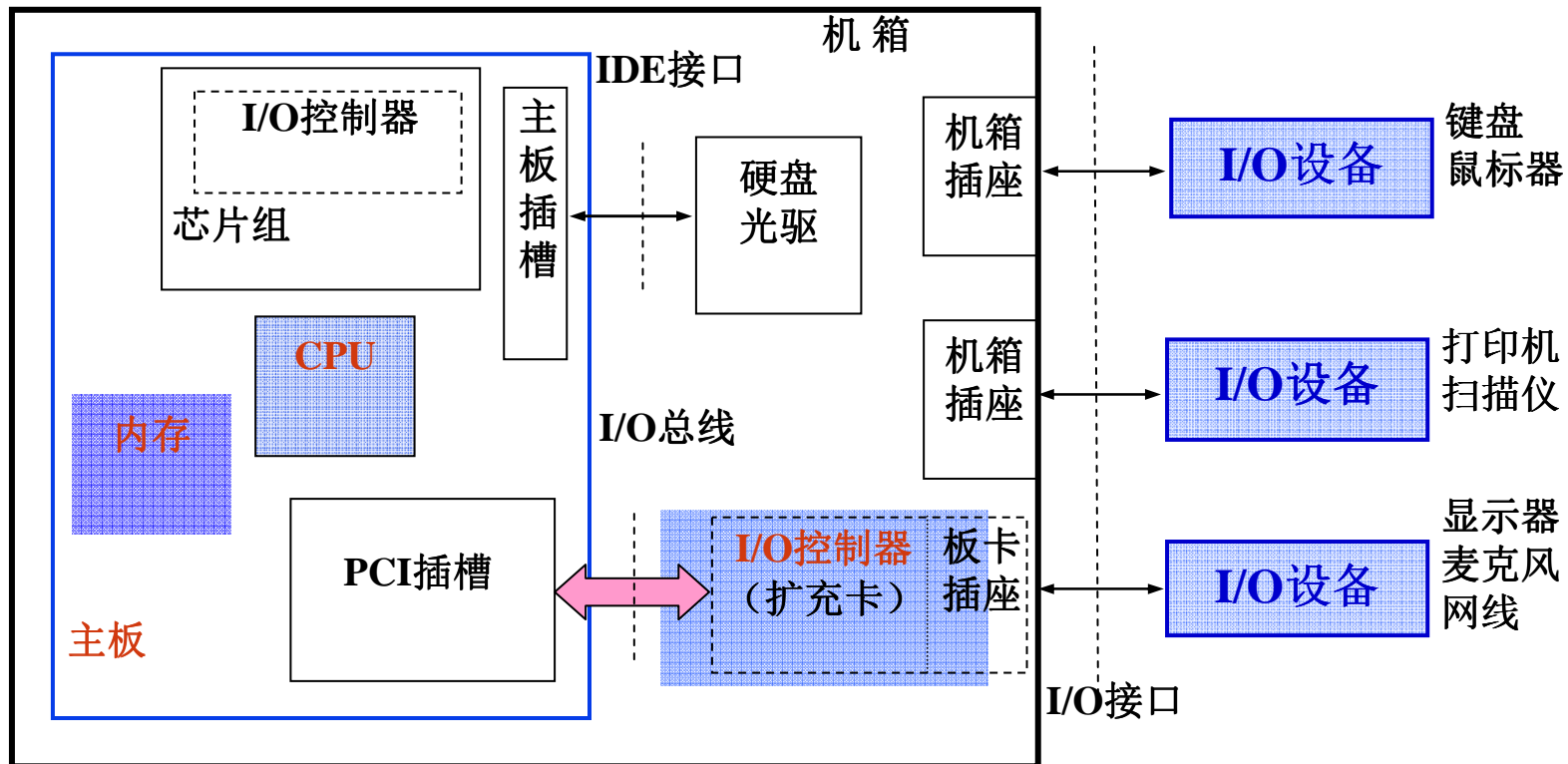


(安装在主板上的I/O设备接口)



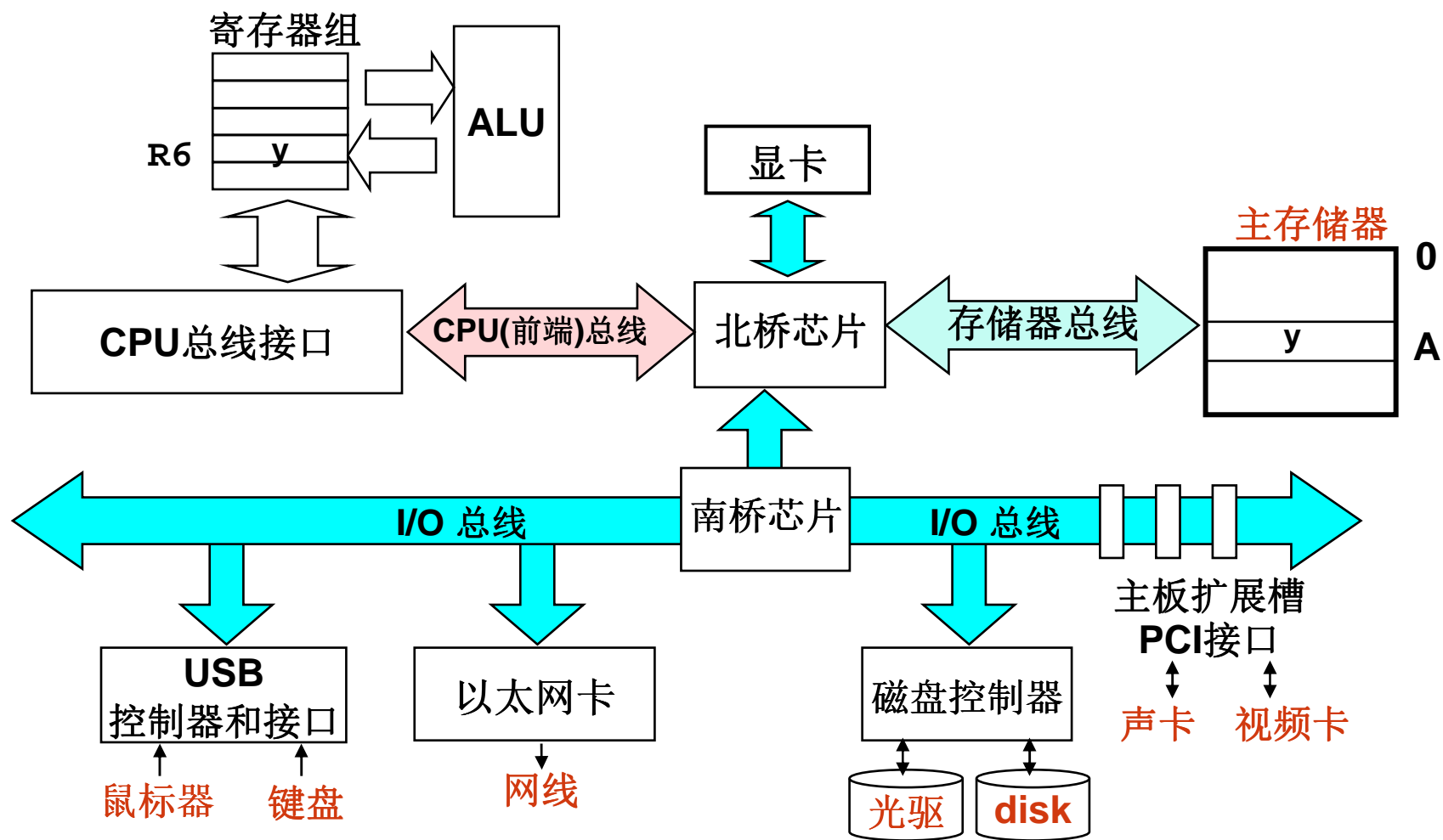


回顾： I/O总线,I/O控制器,I/O接口与I/O设备的关系



- I/O设备通常都是物理上相互独立的设备，它们一般通过I/O接口与I/O控制器连接
- I/O控制器通过扩展卡或者南桥芯片与I/O总线连接
- I/O总线经过北桥芯片与内存、CPU连接

回顾：I/O总线,I/O控制器,I/O接口与I/O设备的关系



本教材把I/O控制器和插座合起来称为I/O接口。

I/O接口（I/O控制器）的职能

- 数据缓冲

提供数据缓冲寄存器，以达到主机和外设工作速度的匹配。

- 错误或状态检测

提供状态寄存器，以保存各种错误或状态信息供**CPU**查用。

- 控制和定时

提供控制和定时逻辑，以接受从系统总线来的控制定时信号。

- 数据格式转换

提供数据格式转换部件使通过外部接口得到的数据转换为内部接口需要的格式，或在相反的方向进行数据格式转换。

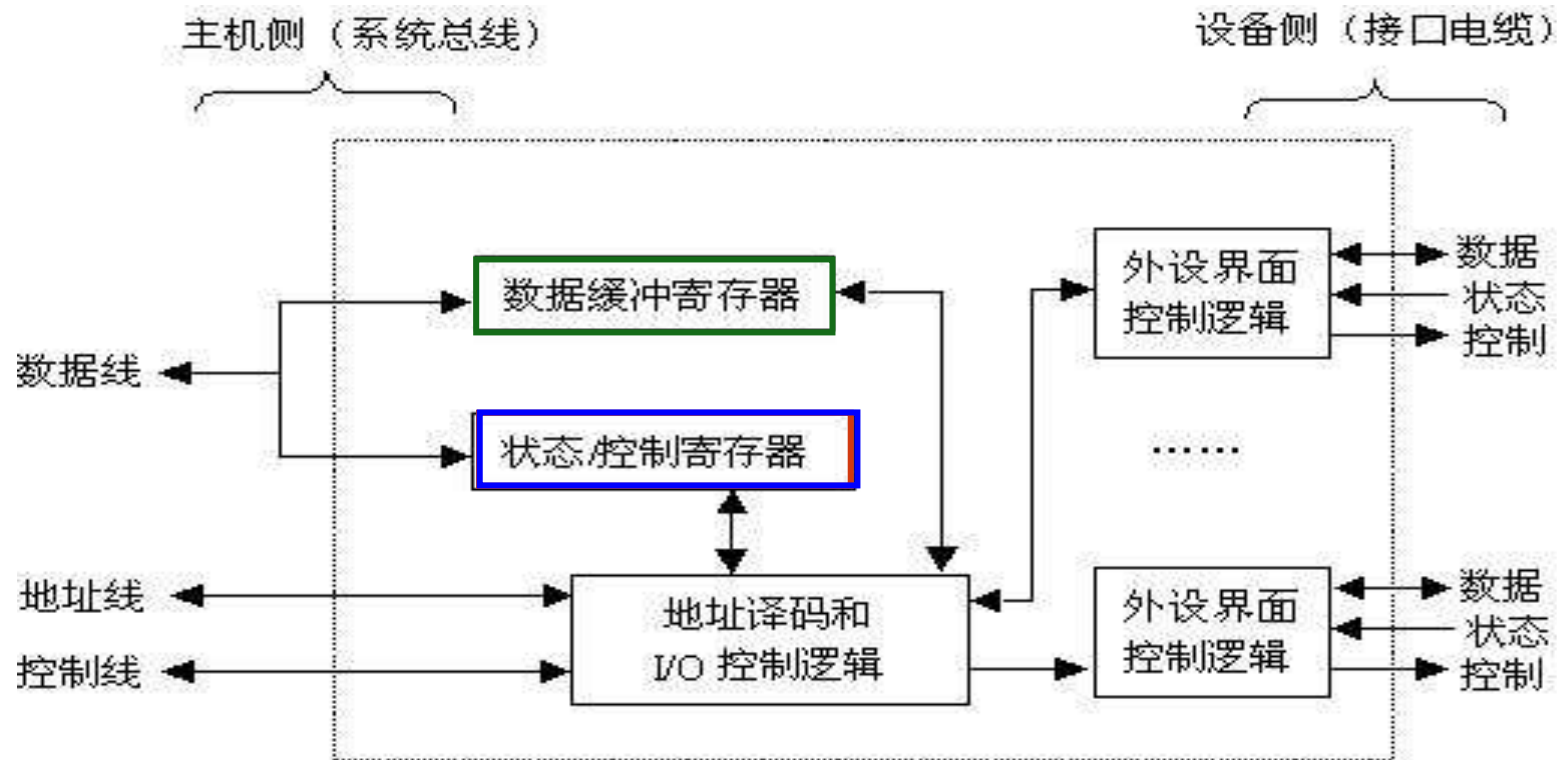
- 与主机和设备通信

上述功能通过**I/O**接口与主机之间、**I/O**接口与设备之间的通信来完成。

I/O接口（I/O控制器）的结构

- I/O控制器的一般结构

（各种不同的I/O模块在复杂性和控制外设的数量上相差很大。）



通过I/O控制寄存器发送命令字来向设备发送命令

通过从状态寄存器读取状态字来获取外设或I/O控制器的状态信息

通过向I/O控制器发送或读取数据来和外设进行数据交换

将I/O控制器中CPU能够访问的各类寄存器称为I/O端口

对外设的访问通过向I/O端口发命令、读状态、读/写数据来进行

I/O设备的寻址方式

- 对I/O端口读写，就是向I/O设备送出命令或从设备取得状态或读/写设备数据
- 一个I/O控制器可能会占有多个端口地址
- I/O端口必须编号后，CPU才能访问
- I/O设备的寻址方式就是I/O端口的编号方式

(1) 统一编址方式（内存映射方式）

与主存空间统一编址，将主存空间分出一部分地址给I/O端口进行编号。

（因该方法是将I/O端口映射到主存空间的某区，故也被称为“存储器映射方式”）

例如，**Motorola**公司生产的处理器就采用该方案

(2) 独立编址方式（特殊I/O指令方式）

不和主存单元一起编号，而是单独编号，使成为一个独立的I/O地址空间

（因需专门I/O指令，故也称为“特殊I/O指令方式”）

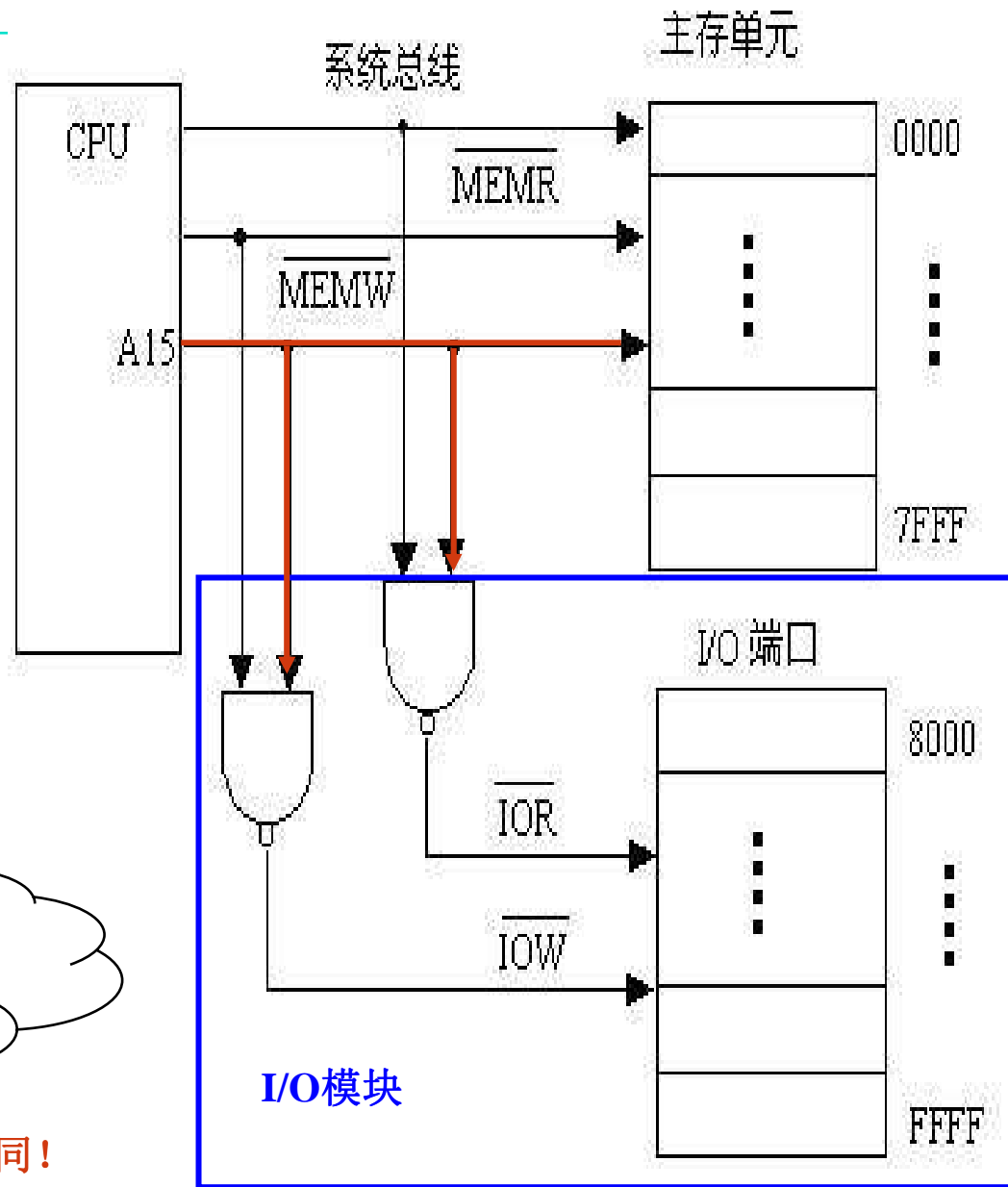
例如，**Intel**公司和**Zilog**公司的处理器就是独立编址方式

统一编址方式

- CPU不直接通过读写控制信号IOR#、IOW#对I/O端口读写，而是根据I/O端口在地址空间的位置，通过地址译码来实现。
- 地址线的高位参与片选控制逻辑。
- 无需设置专门I/O指令，只要用一般访存指令就可存取I/O端口。

MemR#或MemW#命令由访存指令发出，IOR#和IOW#命令怎样呢？

也是访存指令，只是访问的地址范围不同！



统一编址方式的优缺点

- 主要优点:

- 与访存指令一致的存/取指令

- 减少对专门I/O指令的设计

- 利用丰富的访存指令进行I/O操作（不仅可对端口进行数据传送，而且还可直接对端口进行移位、测试等）。

- 便于扩大系统吞吐率

- 外设或I/O寄存器数目几乎不受限制，而只受总存储容量的限制。这在大型控制或数据通信系统等特殊场合很有用。

- 读写控制逻辑简单

- 主要缺点:

- 主存空间减少。因为被I/O占用了存储空间。

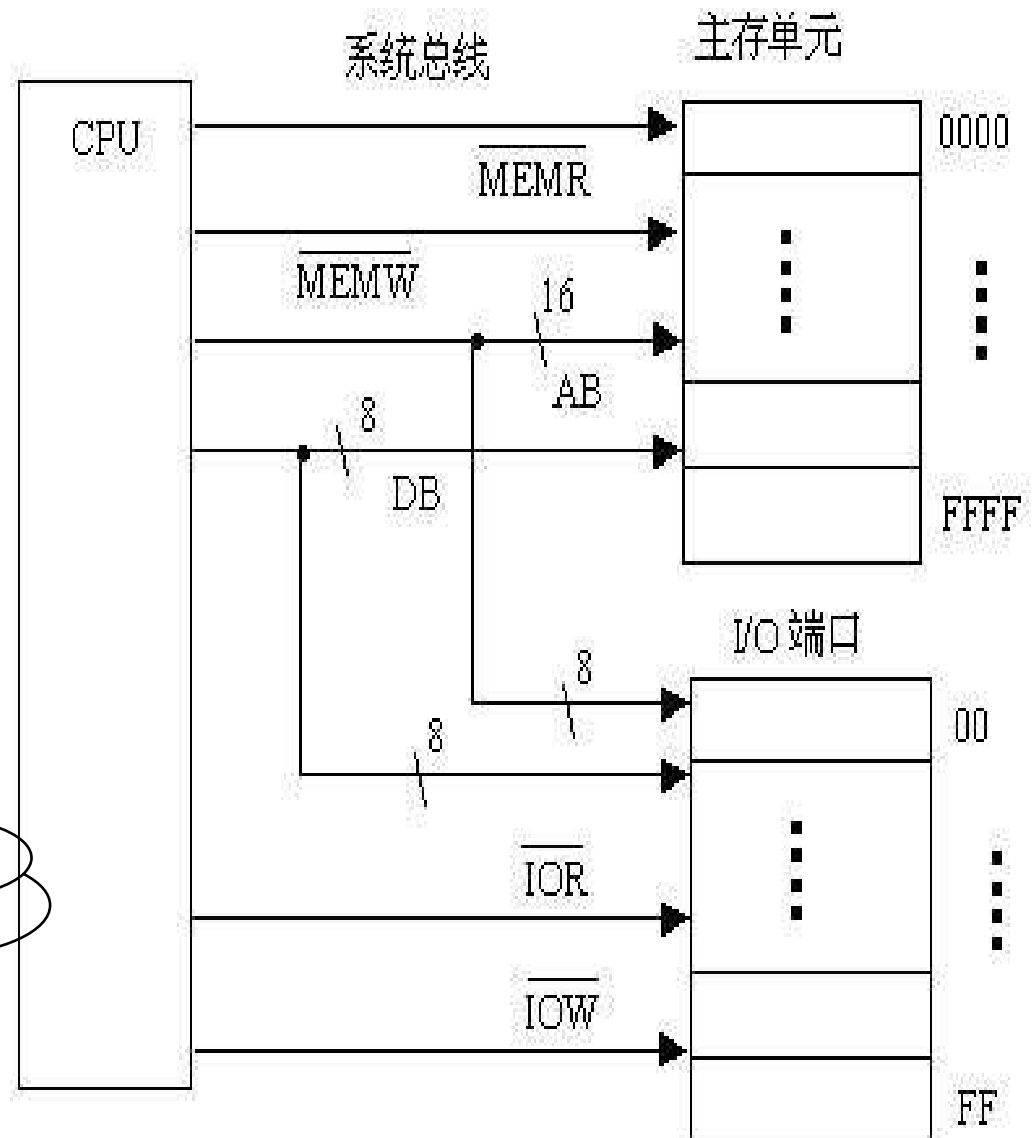
- 外设寻址时间长。为了识别I/O端口,全部地址线都需参与地址译码,使译码电路复杂并需花很长时间。

独立编址方式

- 通过不同的读写控制信号 **IOR#**、**IOW#**和 **MEMR#**、**MEMW#**来实现对**I/O** 端口和存储器的读写。
- 一般**I/O**端口比存储器单元少，所以选择**I/O**端口时，只需少量地址线。
- 指令系统必须设计专门的**I/O**指令。

MemR#或MemW#命令由访存指令发出，IOR#和IOW#命令怎样呢？

是专门的I/O指令，指令中给的地址可能相同，但操作命令不同！



独立编址方式的优缺点

◦ 主要优点：

- **I/O**端口地址不占用存储器地址空间，故主存空间不受**I/O**地址的影响。
- **I/O**地址线较少，所以**I/O**端口译码简单，寻址速度快。
- 使用专用**I/O**指令，程序清晰，便于理解和检查。

◦ 主要缺点：

- 专用 **I/O**指令类型少，只提供简单的传输操作，故程序设计灵活性差。
- 要求处理器提供两组读写命令（**MEMR / MEMW**、**IOR / IOW**），增加了控制逻辑的复杂性和处理器引脚数。

（自学）奔腾机的I/O端口编址方式

- 采用独立编址方式，I/O地址空间由 2^{16} (64K)个8位端口组成
- 虽然具有64K字节的寻址空间，但一般只使用其中1K字节的I/O空间，故只用低10位地址线寻址
- 两个连续的8位端口可作为一个16位端口；四个连续的8位端口可作为一个32位端口。所以一次可传送32位、16位或8位数据
- 采用专门的I/O指令：**IN**和**OUT**（处理器执行到这些指令时产生相应的I/O读写命令信号）
- 部分外设的[I/O地址分配表](#)

(自学) 奔腾机I/O端口地址分配表

部分外设的 I/O 地址分配表

输入/出设备	I/O 地址	占用地址数
DMA 控制器 1	000-01FH	32
中断控制器 1	020-03FH	32
定时器/计数器	040-05FH	32
键盘控制器	060-06FH	32
实时时钟, NMI屏蔽寄存器	070-07FH	16
DMA 页面寄存器	080-09FH	32
中断控制器 2	0A0-0BFH	32
DMA 控制器 2	0C0-0DFH	32
硬盘控制器 2	170-177H	8
硬盘控制器 1	1F0-1F8H	9
游戏 I/O 口	200-207H	8
并行打印机口 2	278-27FH	8
串行口 4	2E8-2EFH	8
串行口 2	2F8-2FFH	8
软盘控制器 2	370-377H	8
并行打印机口 1	378-37FH	8
单色显示器/打印适配器	3B0-3BFH	16
彩色/图形监视器适配器	3D0-3DFH	16
串行口 3	3E8-3EFH	8
软盘控制器 1	3F0-3F7H	8
串行口 1	3F8-3FFH	8

并行传输和串行传输（自学）

并行传输方式

- 多位数据在多条数据线上并行传送
- 最大传输率为：时钟频率 \times 数据线宽度

串行传输方式

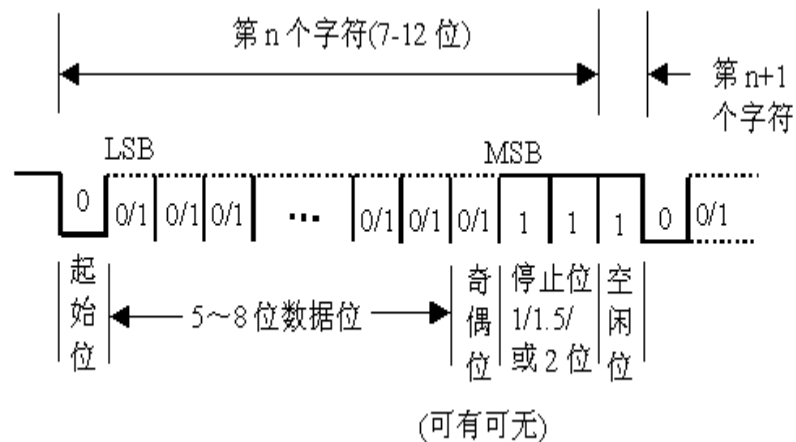
- 波特率：每秒钟通过信道传输的码元数
- 比特率：每秒钟传输的位数
- 两相调制时，波特率=比特率

异步串行

- 每个字符的开始是随机的，字符内的位之间同步
- 有效数据位为5位时，停止位取1位或1.5位，其他情况取1位或2位
- 一个字符可能由7~12位信息组成，称其为一个数据帧
- 缺点：每个字符都有额外信息，实际数据传输率低

同步串行

- 字符之间、字符内的位之间都同步



小结

- I/O系统概述
 - I/O系统的性能主要有吞吐率和响应时间，两者是对立统一的关系
 - I/O系统的具体任务是：构建传输通路、对设备寻址、向设备发命令、取状态，并提供相应的传输机制来读/写设备数据等（后面一讲的内容）
- I/O设备概述
 - I/O设备通过I/O接口和主机相连
 - 外设分类
 - I/O设备和存储设备
 - 机读设备和人读设备
- 磁盘存储器
 - 读写原理：两种不同的磁化状态
 - 性能指标：寻道时间、旋转等待时间、传输时间
 - 冗余磁盘阵列：多个物理盘组成一个逻辑盘，以提高磁盘存取速度、容量和可靠性
- I/O接口（I/O控制器）中一般有数据缓冲器、状态/控制寄存器、串-并转换、设备控制逻辑、地址译码逻辑等。用于在主机和设备之间进行命令、数据、状态信息的传递和转换。
- I/O端口是指I/O控制器中CPU可访问的寄存器，对I/O设备的寻址就是对I/O端口的访问
- I/O端口的编址方式有两种：内存映射方式（统一编址）和特殊I/O指令方式（独立编址）

本教材把I/O控制器和I/O接口综合称为I/O接口，不严格区分控制逻辑部分（I/O控制器）和插座（I/O接口）部分