

Computer Architecture

15. IO System

Jianhua Li

College of Computer and Information
Hefei University of Technology

内容概要

6.1 引言

6.2 外部存储设备

6.3 可靠性、可用性和可信性

6.4 廉价磁盘冗余阵列RAID

6.5 I/O设备与CPU和存储器的连接

6.6 I/O与操作系统

引言

□ 输入/输出系统简称I/O系统

- I/O设备：键盘、鼠标等
- I/O设备与处理机的连接

□ I/O系统的重要性

- I/O是Von Neumann架构的重要组成部分

□ 衡量指标：

- 响应时间 (Response Time)
- 可靠性 (Reliability)

I/O系统性能与CPU性能的一个谬论

- ❑ **谬论：**使用多进程技术可以忽略I/O性能对系统性能的影响。
 - 多进程技术只能够提高系统吞吐率，并不能够减少系统响应时间；
 - 进程切换时可能需要增加I/O操作；
 - 可切换的进程数量有限，当I/O处理较慢时，仍然会导致CPU处于空闲状态；

性能示例

例 6.1 假设一台计算机的I/O处理占10%，当其CPU性能按照如下方式改进，而I/O性能保持不变时，系统总体性能会出现什么变化？

- 如果CPU的性能提高10倍
- 如果CPU的性能提高100倍

性能示例

解：假设原来的程序执行时间为**1**个单位时间。如果CPU的性能提高10倍，程序的计算（包含I/O处理）时间为：

$$(1 - 10\%)/10 + 10\% = 0.19$$

也就是说：整机性能只能提高约5倍，差不多有**50%**的时间浪费在I/O上。

如果CPU性能提高100倍，程序的计算时间为：

$$(1 - 10\%)/100 + 10\% = 0.109$$

而整机性能只能提高约10倍，表示有**90%**的时间浪费在没有改进的I/O上了。

I/O系统的可靠性

- 当前处理器性能已经很高，在很多场景下人们更加关注系统可靠性。
 - 为什么可靠性这么重要？
- 可信性是存储系统的基础（云存储！）
 - 可靠性？
 - 可用性？
 - 可信性？

内容概要

6.1 引言

6.2 外部存储设备

6.3 可靠性、可用性和可信性

6.4 廉价磁盘冗余阵列RAID

6.5 I/O设备与CPU和存储器的连接

6.6 I/O与操作系统

外部存储设备

□ 各种外部存储器的相似之处

- 记录原理类似：磁性、电阻、电压等；
- 作为计算机部件，均包括驱动机制；
- 作为存储设备，均包括控制器及接口逻辑；
- 均采用定位和校正技术以及相似的读写系统。

□ 目前常用的存储设备：

- 磁盘
- Flash, SSD等
- 光盘、磁带（逐渐消退。。。)

磁盘-Hard Disk

□ 磁盘占据着非易失性存储器的主导地位

- 它是存储层次中主存的下一级存储层次
- 它是虚拟存储器技术的物质基础
- 它是操作系统和所有应用程序的驻留介质

□ 磁盘的结构组成

- 盘片:
 - 数量: 1 ~ 12 转速: 3600 ~ 15000转
 - 盘径: 1.0英寸(25.4厘米) ~ 3.5英寸(8.9厘米)
- 磁道: 每面5000 ~ 30000道
- 扇区: 每道100 ~ 500个

磁盘结构图



图6.2 磁盘结构图（动态）

磁盘性能

□ 访问时间

■ 磁盘的工作过程：

- 磁头首先移动到目标磁道上
- 使期望的扇区旋转至磁头下
- 读取扇区中的数据

■ 工作均在磁盘控制器的控制下完成

磁盘访问时间 = 寻道时间 + 旋转时间 +
传输时间 + 控制器开销

磁盘性能

□ 磁盘容量

- 磁盘容量与盘片数量和单碟容量有关。
- 受工业标准的限制，硬盘中能安装的盘片数目有限：3~4片。一般都是5片以内。
- 提高单碟容量的途径有两个：
 - 提高道密度
 - 提高位密度
- 目前的主流3.5英寸硬盘的单碟容量已经达到了百GB。最新的可以达TB级别。

磁盘性能

□ 磁盘缓存 (Disk Cache)

- 可以弥补磁盘和主存之间的速度差距
- 利用被访问数据的局部性原理
 - 时间局部性
 - 空间局部性
- 磁盘缓存的管理和实现一般是软硬件协同

磁盘性能

□ 磁盘的发展

- 磁盘容量的提高通常用面密度来衡量
 - 面密度为单位面积可以记录的数据位数
 - 面密度 = 磁盘面的道密度 × 磁道的位密度
- 容量不断提升，每位价格不断下降。
- 磁盘的性能价格比高于主存，但访问速度却要低得多。换句话说，性能价格比与速度要求差距太大。

磁盘性能

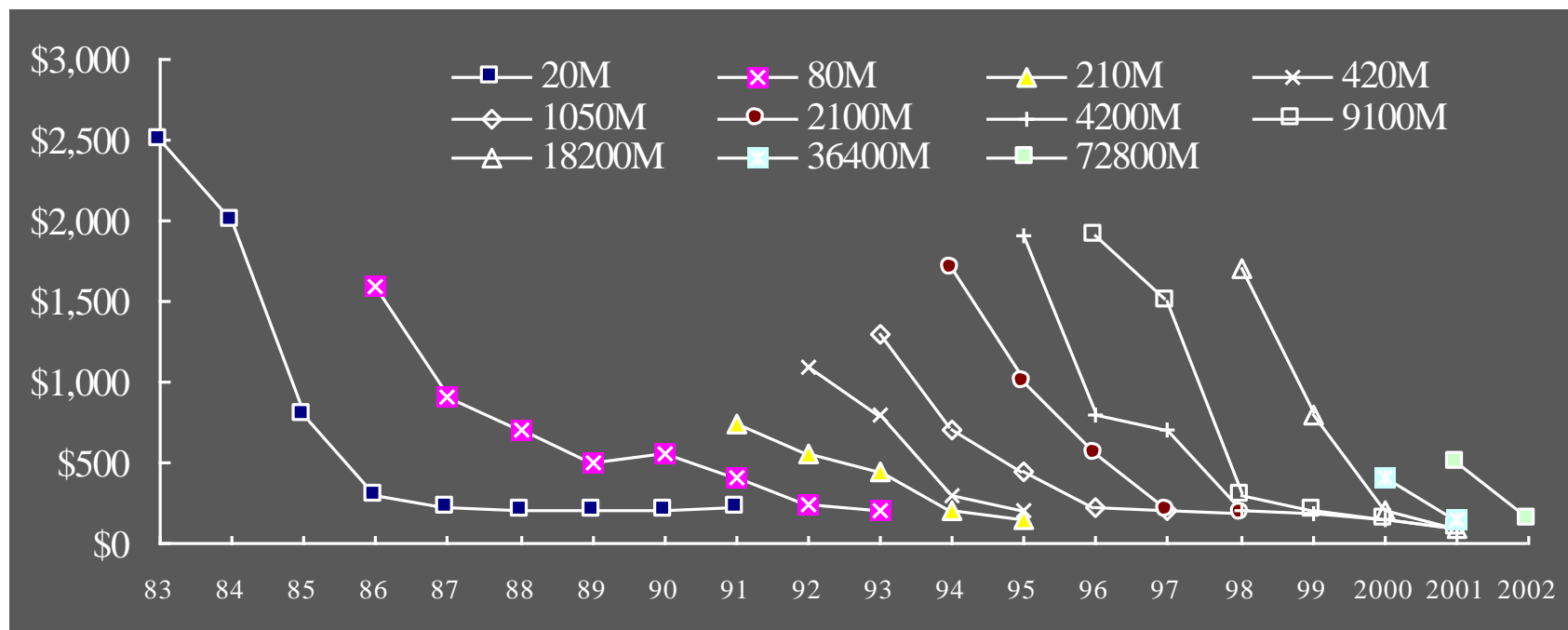


图6.3 磁盘价格的变化

磁盘性能

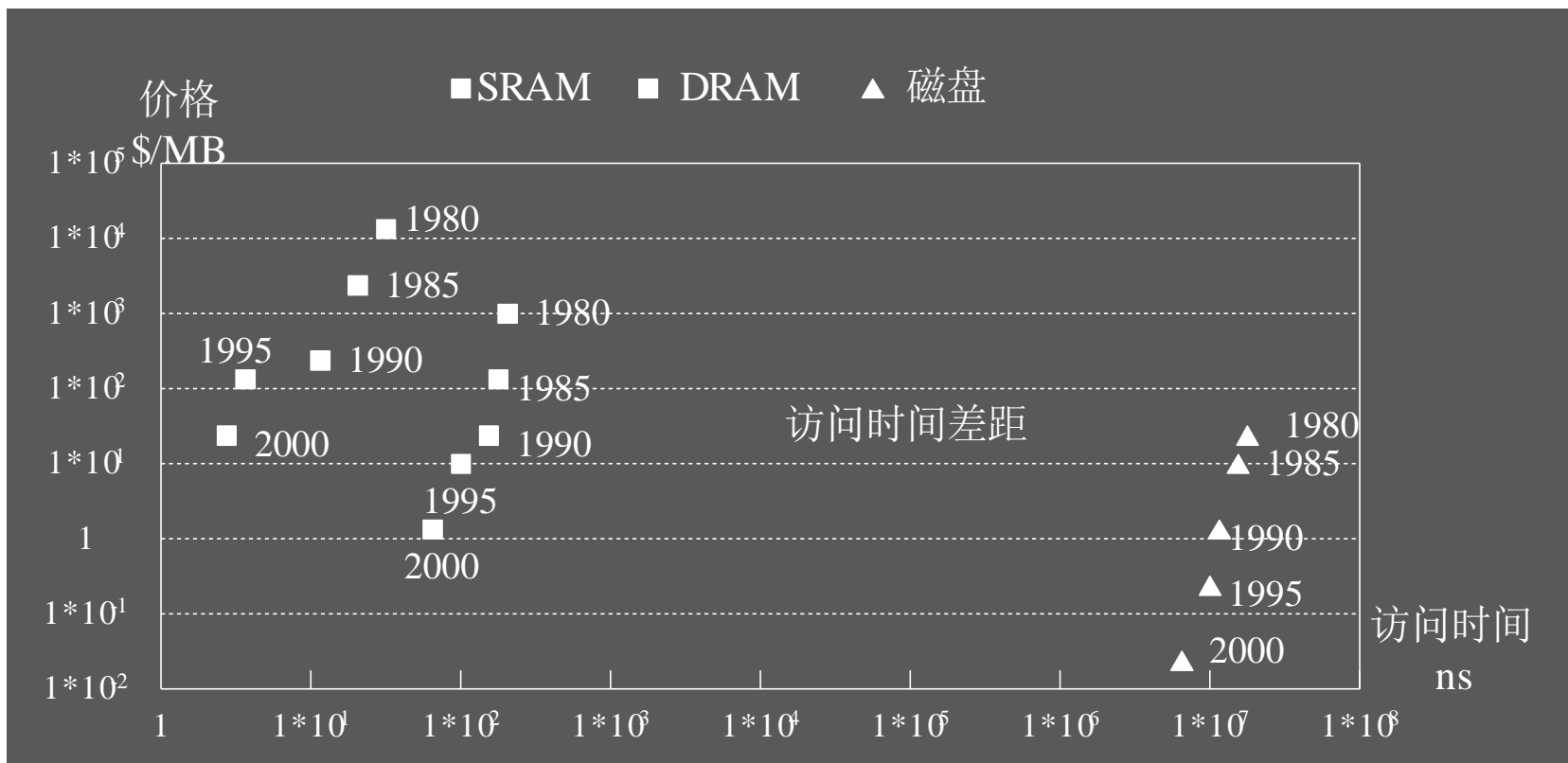


图6.4 磁盘和半导体存储器之间的访问时间差距

Flash存储器

- ❑ 工作原理同E²PROM，容量比E²PROM大。
- ❑ 与磁盘相比的主要特点：
 - 功耗小 ($\leq 50\text{mw}$)
 - 尺寸小/存储密度大
 - 提供与DRAM相仿的访问速度
 - 价格高

固态硬盘 (SSD)

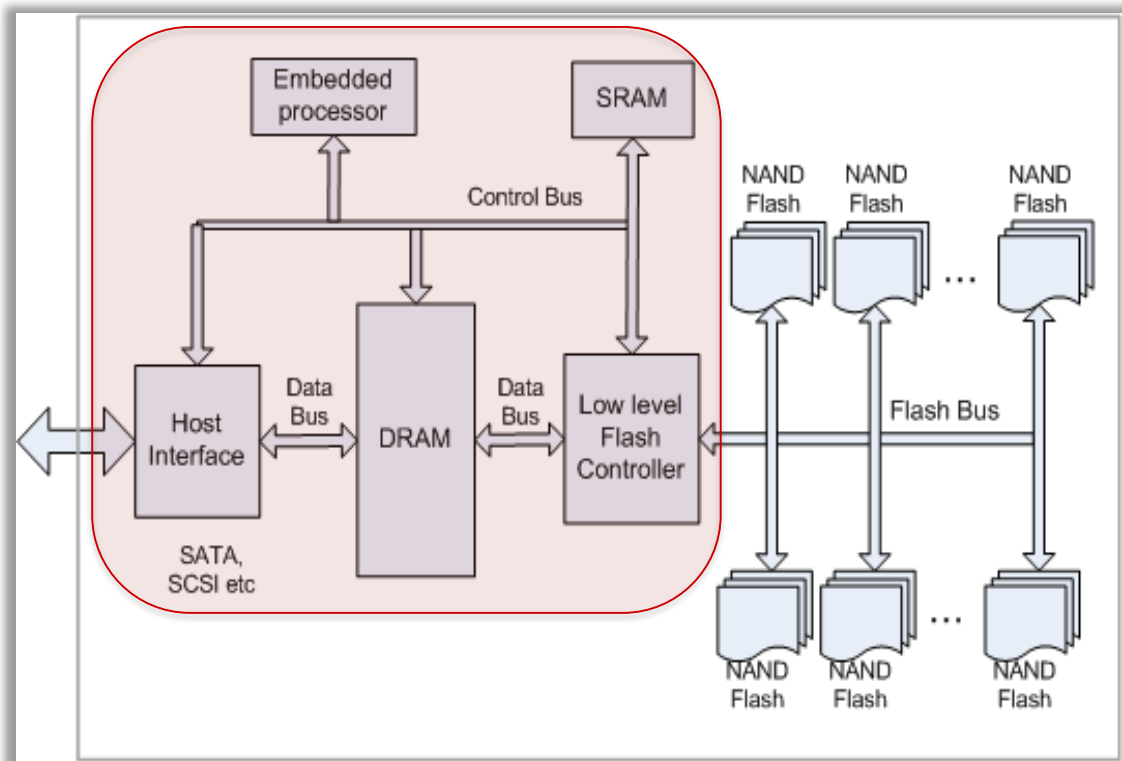


图6.5 固态硬盘的结构图和架构图

固态硬盘优点

- ❑ 极好的读延迟和顺序写带宽
- ❑ 较低的 \$/IOPS/GB
- ❑ 较低的功耗开销
- ❑ 无机械部件，更加可靠
- ❑ 封装尺寸更小，无noise

内容概要

6.1 引言

6.2 外部存储设备

6.3 可靠性、可用性和可信性

6.4 廉价磁盘冗余阵列RAID

6.5 I/O设备与CPU和存储器的连接

6.6 I/O与操作系统

可靠性、可用性和可信性

□ 系统可靠性

- 系统从初始状态开始一直提供服务的能力
- 用平均无故障时间MTTF来衡量

□ 系统可用性

- **工作时间**在**连续两次正常服务间隔时间**中所占比率
- 用MTTF/MTBF（平均失效间隔时间）来衡量

□ 系统可信性

- 多大程度上可以合理地认为服务是可靠的
- 可信性不可度量

可靠性、可用性和可信性示例

例 6.2 假设磁盘子系统的组成部件和它们的MTTF如下：

- ① 磁盘子系统由10个磁盘构成，每个磁盘的MTTF为1000000小时；
- ② 1个SCSI控制器，其MTTF为500000小时；
- ③ 1个不间断电源，其MTTF为200000小时；
- ④ 1个风扇，其MTTF为200000小时；
- ⑤ 1根SCSI连线，其MTTF为1000000小时。

可靠性、可用性和可信性示例

假定每个部件的正常工作时间服从指数分布,即部件的工作时间与故障出现的概率无关;同时假定各部件的故障相互独立,试计算整个系统的MTTF。

解: 整个系统的失效率为:

$$\begin{aligned} & \text{系统失效率} \\ &= 10 \times \frac{1}{1000000} + \frac{1}{500000} + \frac{1}{200000} + \frac{1}{200000} + \frac{1}{1000000} \\ &= \frac{23}{1000000} \end{aligned}$$

可靠性、可用性和可信性示例

系统的MTTF为系统失效率的倒数，即：

$$\text{MTTF} = \frac{1000000}{23} = 43500 \text{ 小时}$$

大约为5年

可靠性、可用性和可信性

□ 提高系统可靠性的方法

- 有效构建方法
- 纠错方法

□ 具体的说，可分为：

- 故障避免技术
- 故障容忍技术
- 错误消除技术
- 错误预报技术

内容概要

6.1 引言

6.2 外部存储设备

6.3 可靠性、可用性和可信性

6.4 廉价磁盘冗余阵列RAID

6.5 I/O设备与CPU和存储器的连接

6.6 I/O与操作系统

廉价磁盘冗余阵列RAID

❑ 廉价磁盘冗余阵列

- Redundant Array of Inexpensive Disks

❑ 独立磁盘冗余阵列

- Redundant Array of Independent Disks

❑ 简称盘阵列技术

- 1988年, David Patterson的研究小组首先提出。

[PDF] [A Case for Redundant Arrays of Inexpensive Disks \(RAID\)](https://www.cs.cmu.edu/~garth/RAIDpaper/Patterson88.pdf)

<https://www.cs.cmu.edu/~garth/RAIDpaper/Patterson88.pdf> ▼ [翻译此页](#)

作者: DA Patterson - 被引用次数: 4077 - [相关文章](#)

A Case for **Redundant Arrays of Inexpensive Disks** (RAID). David A Patterson, Garth Gibson, and Randy H Katz. Computer Science Department, Department of ...

各级RAID的共性

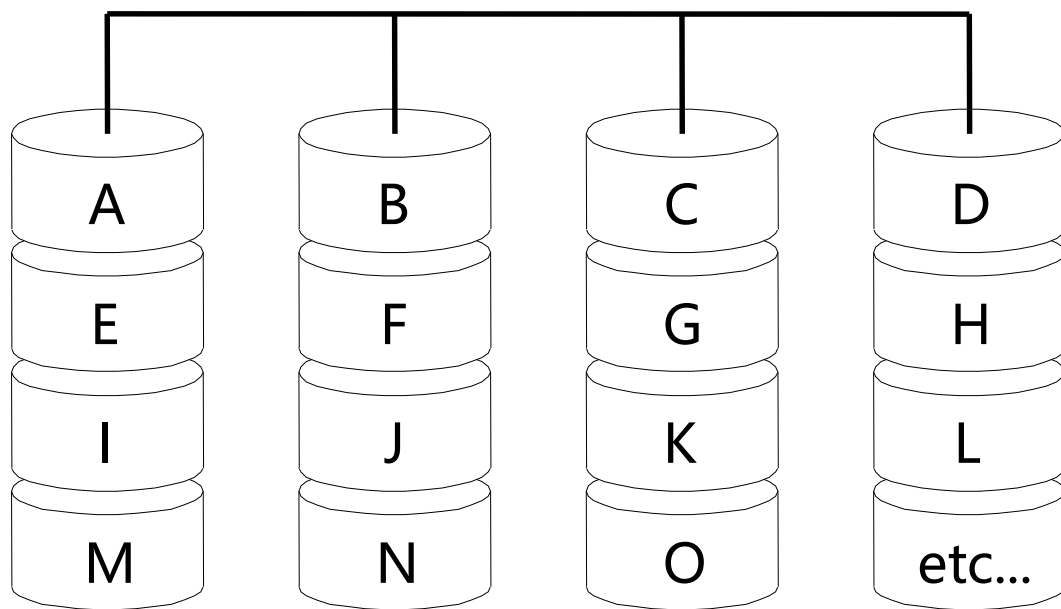
- ① RAID由一组物理磁盘驱动器组成，操作系统视之为一个逻辑驱动器。
- ② 数据分布在一组物理磁盘上
- ③ 冗余信息被存储在冗余磁盘空间中，保证磁盘在万一损坏时可以恢复数据。
- ④ 其中第②③个特性在不同的RAID级别中的表现不同，RAID-0不支持第3个特性。

RAID面对的问题

- 关键问题：如何发现磁盘的失效
 - 磁盘技术(扇区中)提供了故障检测操作的信息。
- 设计的另一个问题
 - 如何减少平均修复时间MTTR
 - 典型的做法：在系统中增加热备份盘
- 热切换技术
 - 与热备份盘相关的一种技术
 - 不用关机就能更换设备

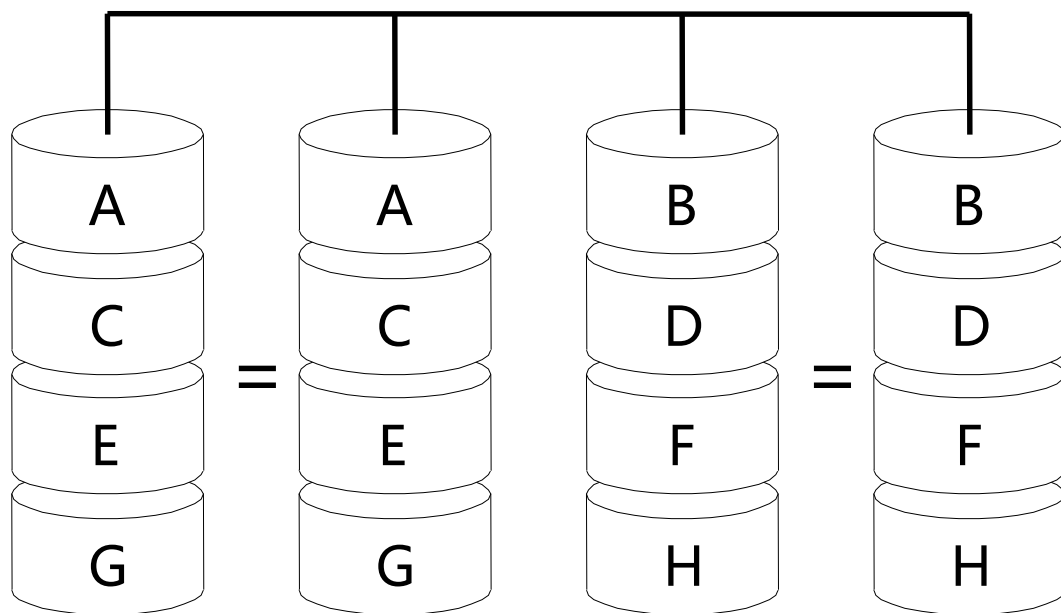
RAID-0

- ❑ 数据分块，即把数据分布在多个盘上。
- ❑ 非冗余阵列、无冗余信息。
- ❑ 严格地说，它不是RAID。



RAID-1

- ❑ 亦称镜像盘，使用双备份磁盘。
- ❑ 每当数据写入一个磁盘时，将该数据也写到另一个冗余盘，形成信息的两份复制品。



RAID-1特点

□ 读性能好

- RAID-1的性能能够达到RAID-0性能的两倍。

□ 写性能由写性能最差的磁盘决定。

- 相对后面各级RAID来说，RAID-1的写速度较快。

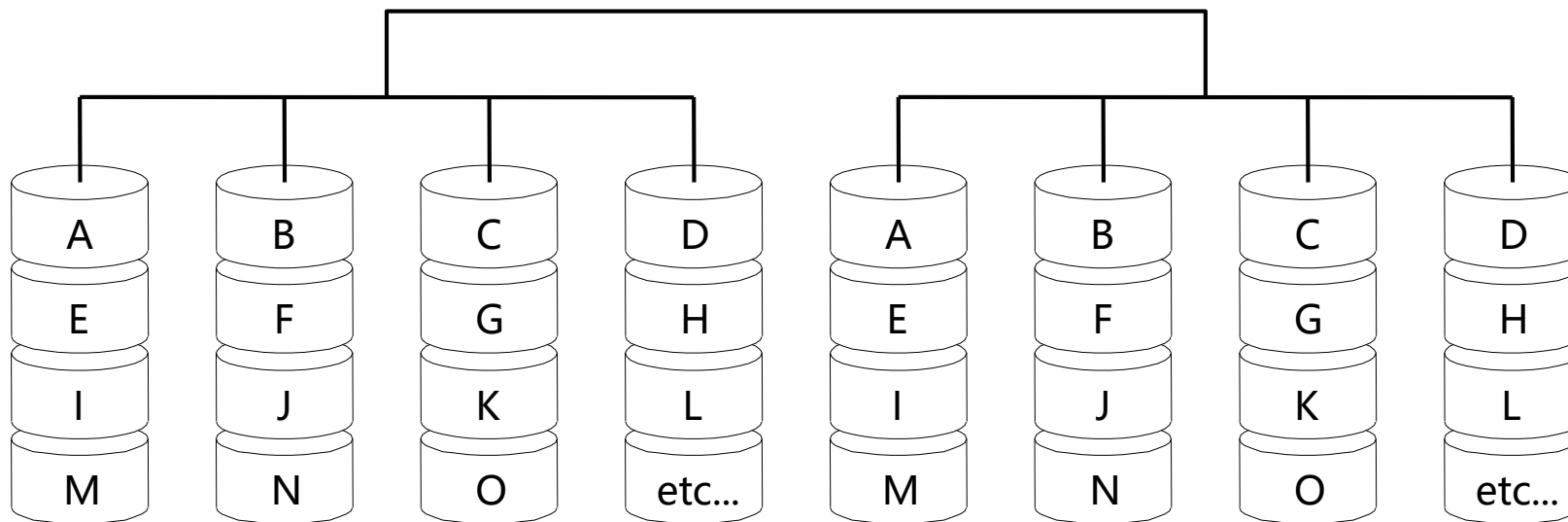
□ 可靠性很高 (很显然)

□ 最昂贵的解决方法

- 物理磁盘空间是逻辑磁盘空间的两倍。

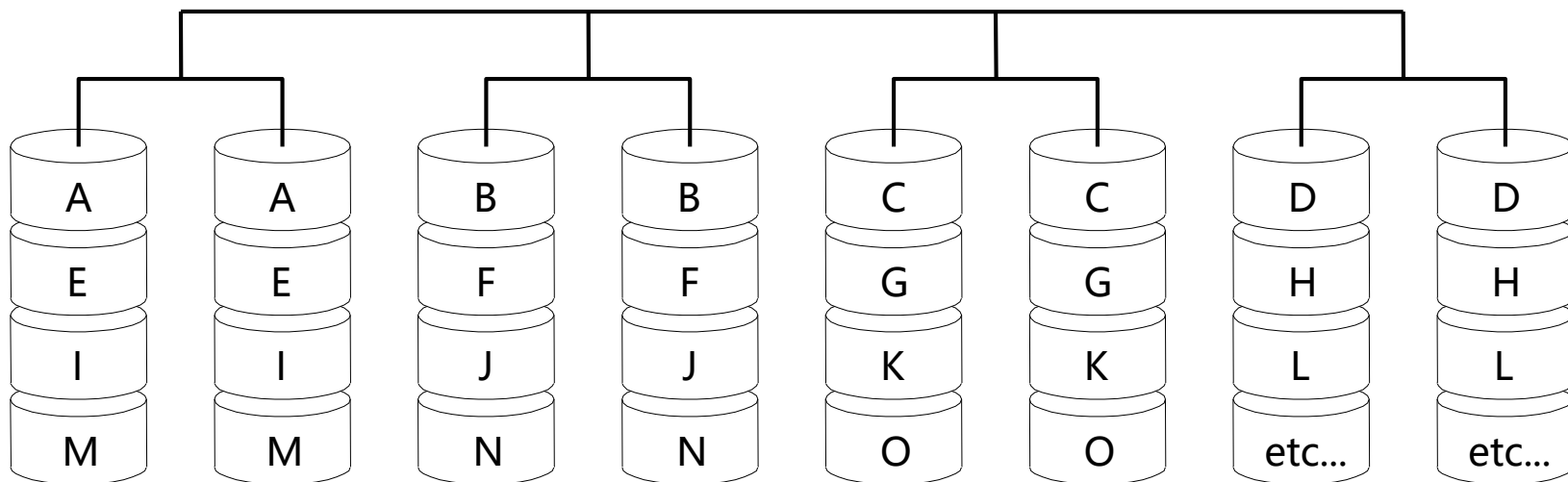
RAID-01

□ RAID-0 —> RAID-1 先分块后镜像



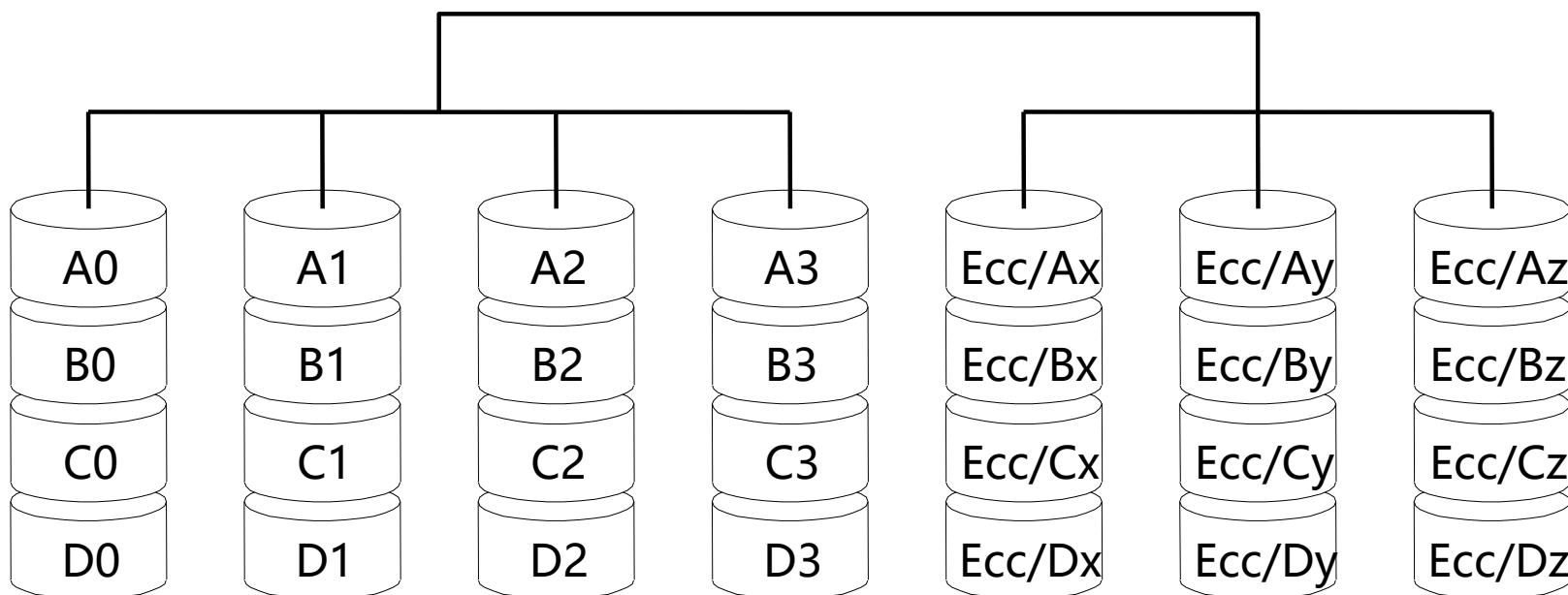
RAID-10

□ RAID-1 —> RAID-0 先镜像后分块



RAID-2 (bit-interleaved hamming code)

- ❑ 位交叉式海明编码阵列
- ❑ 各个数据盘上的相应位计算海明校验码，编码位被存放在多个校验（ECC）磁盘的对应位上。

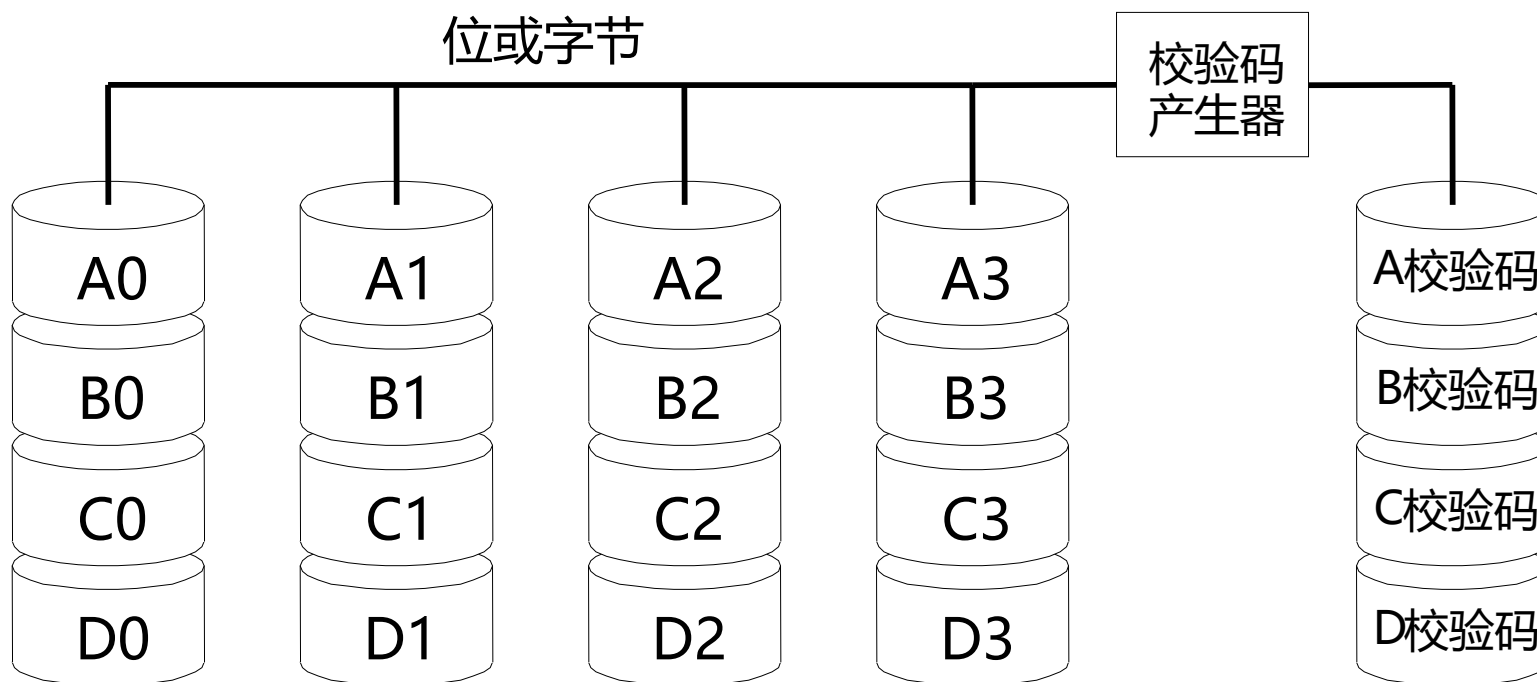


RAID-2特点

- 并行存取，各个驱动器同步工作。
- 使用海明编码来进行错误检测和纠正，数据传输率高。
- 需要多个磁盘来存放海明校验码信息，冗余磁盘数量与数据磁盘数量的对数成正比。
- 是一种在多磁盘易出错环境中的有效选择。并未被广泛应用，目前还没有商业化产品。
 - 为什么没有商业化产品？

RAID-3 (bit/byte-interleaved parity)

- ❑ 位交叉奇偶校验盘阵列
- ❑ 单盘容错并行传输：数据以位或字节交叉存储，奇偶校验信息存储在一台专用盘上。



RAID-3特点

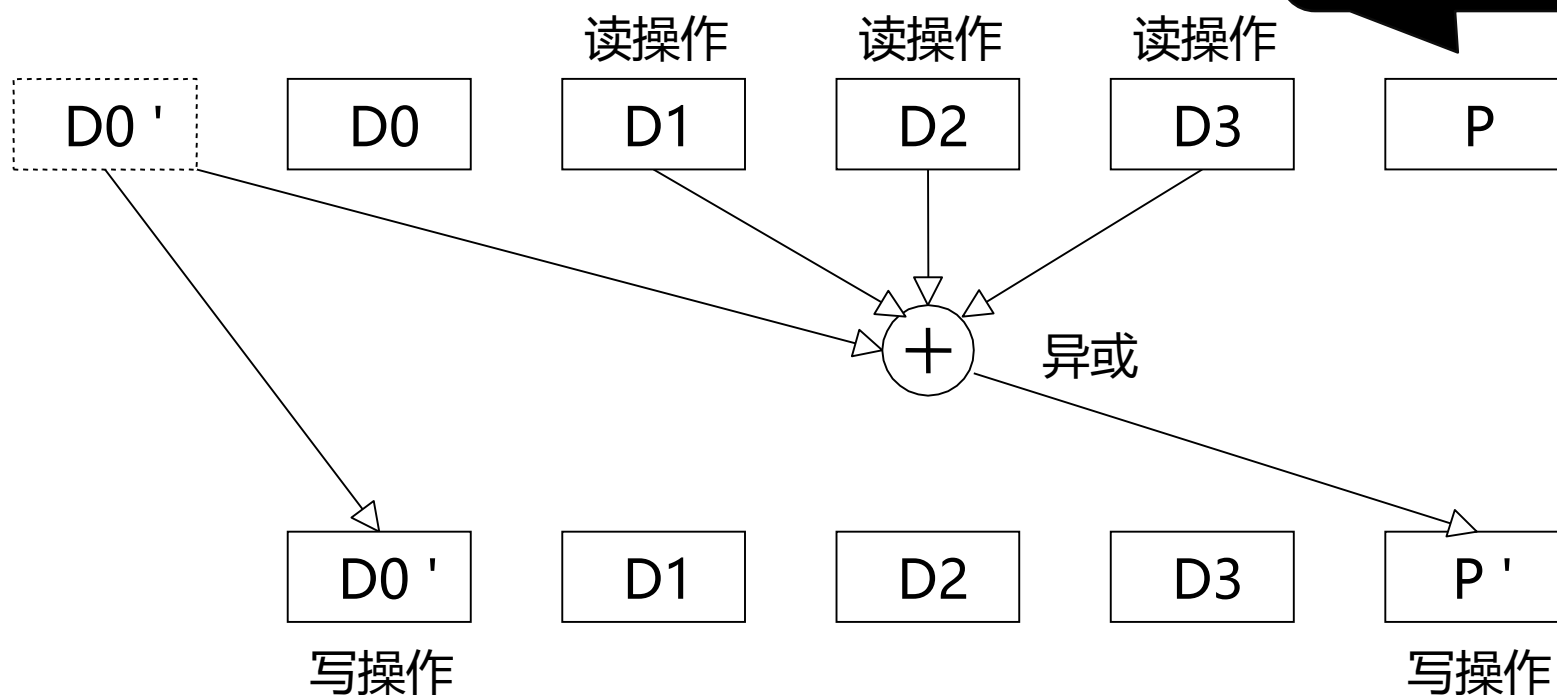
- ❑ 将磁盘分组，读写要访问组中所有盘，每组中有一个盘作为校验盘。
- ❑ 校验盘一般采用奇偶校验。
- ❑ 简单理解：先将分布在各个数据盘上的一组数据加起来，将和存放在冗余盘上。一旦某一个盘出错，只要将冗余盘上的和减去所有正确盘上的数据，得到的差就是出错的盘上的数据。
 - 优点：高数据传输率，空间开销小。
 - 缺点：恢复时间较长，不能同时支持多个IO请求。

RAID-3读写特点

假定：有4个数据盘和1个冗余盘：

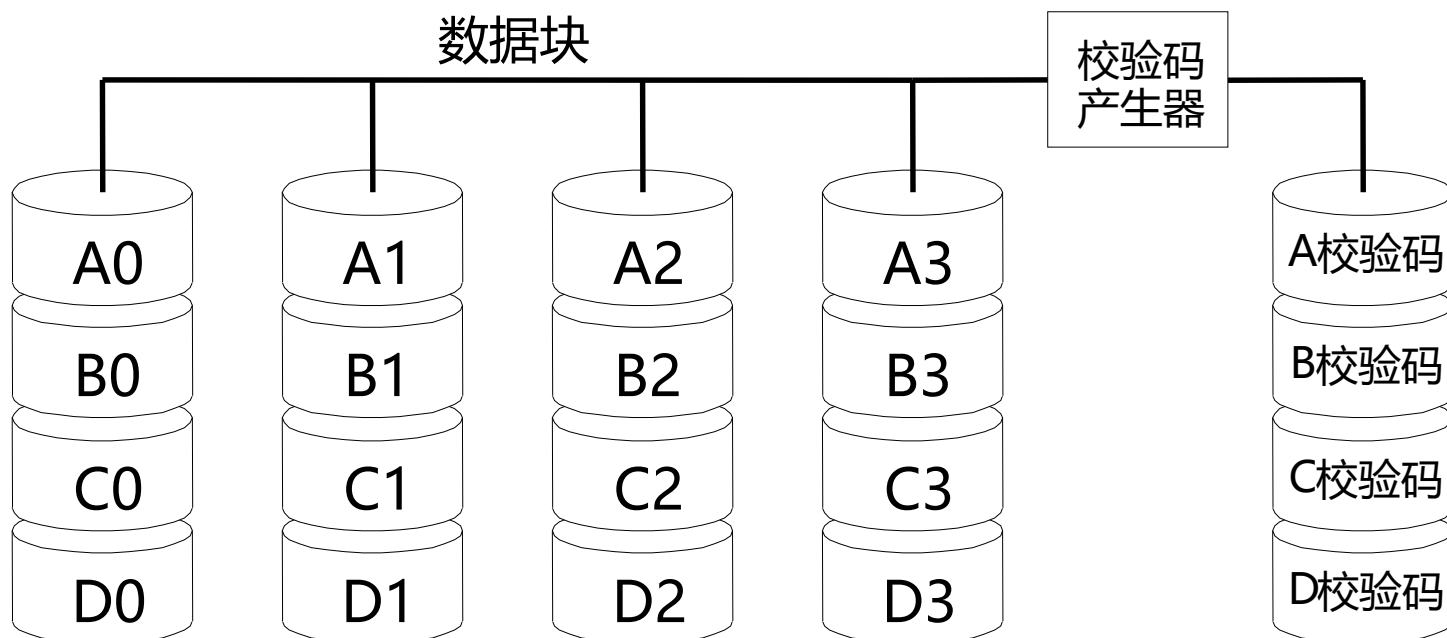
- 读出数据时通常涉及所有磁盘；
- 写数据需要3次磁盘读和2次磁盘写操作【如

校验盘是系统瓶颈！！！！



RAID-4 (block-interleaved parity)

- ❑ 专用奇偶校验独立存取盘阵列
- ❑ 数据以块（块大小可变）交叉的方式存于各盘，奇偶校验信息存在一台专用盘上。



RAID-4特点

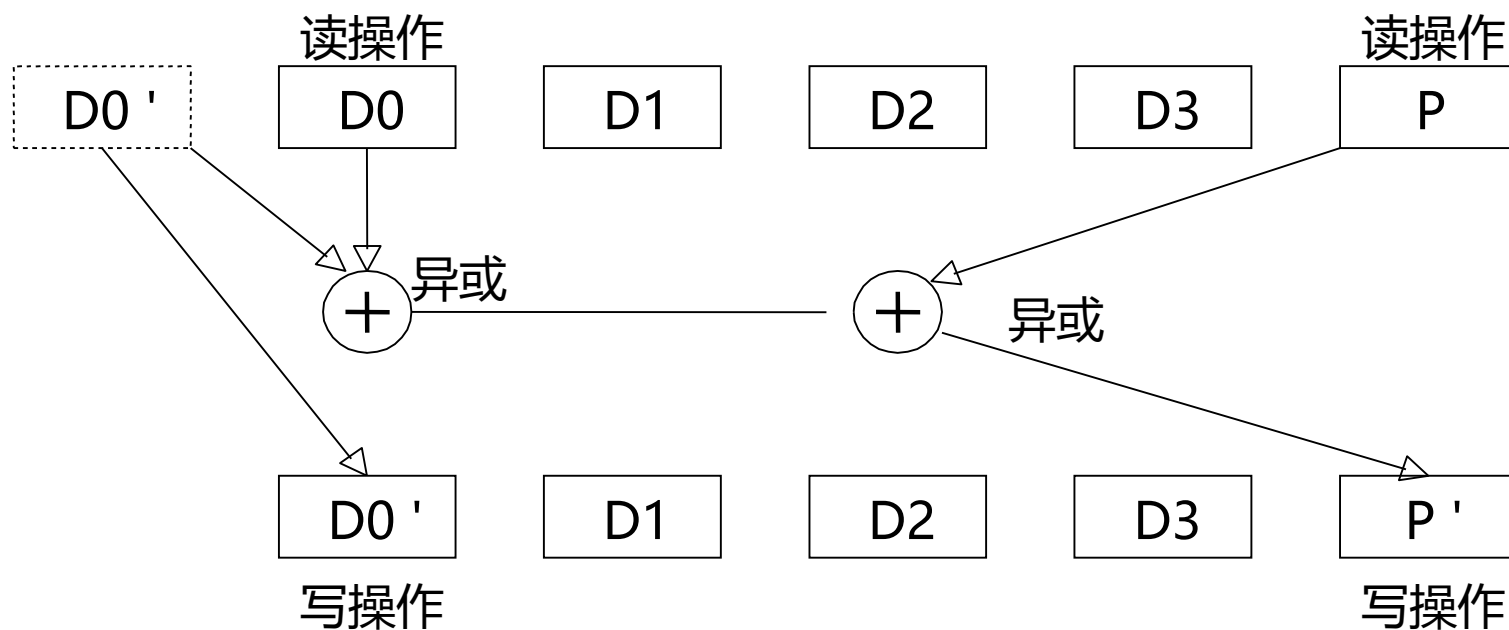
- 冗余代价与RAID-3相同
- 访问数据的方法与RAID-3不同
 - 在RAID-3中，一次磁盘访问将对磁盘阵列中的所有磁盘进行操作。
 - RAID-4出现的原因：
 - 希望使用较少的磁盘参与操作，以使磁盘阵列可以并行进行多个数据的磁盘操作。

RAID-4读写特点

假定:有4个数据盘和1个冗余盘

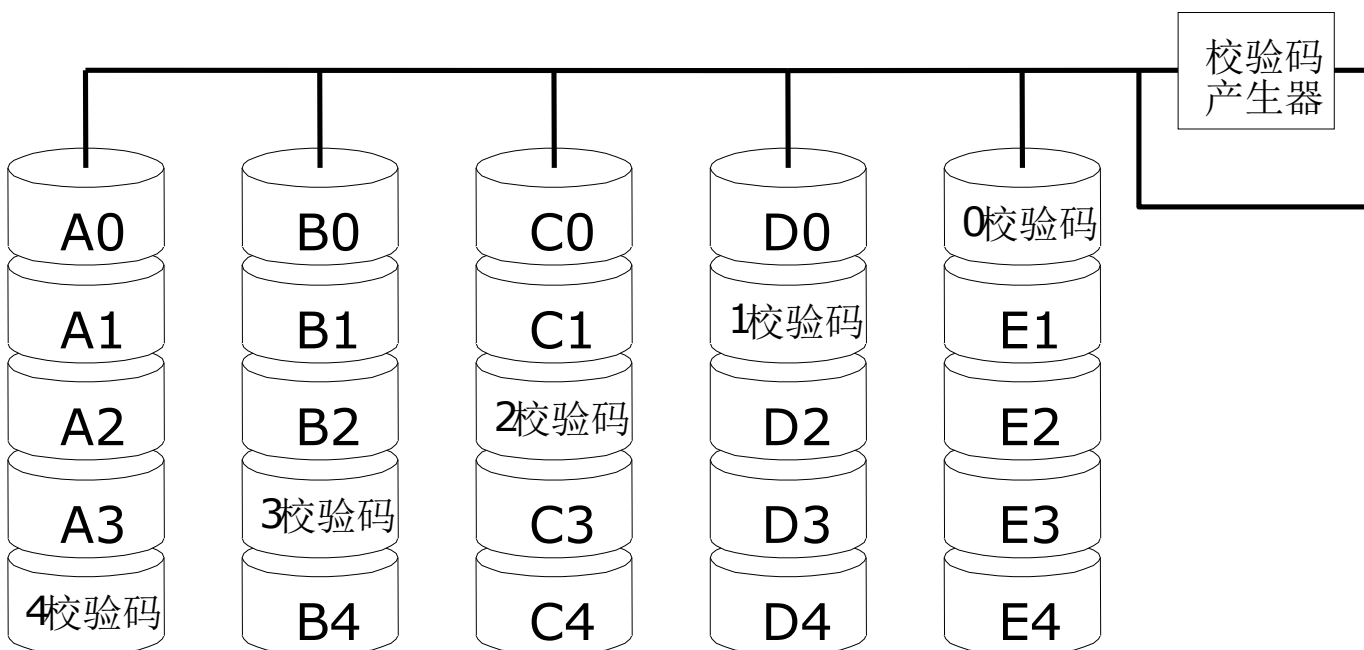
- 读出数据，仅仅需要读取数据对应的磁盘；
- 写数据需要2次磁盘读和2次磁盘写操作。

校验盘也是系统瓶颈！！！！

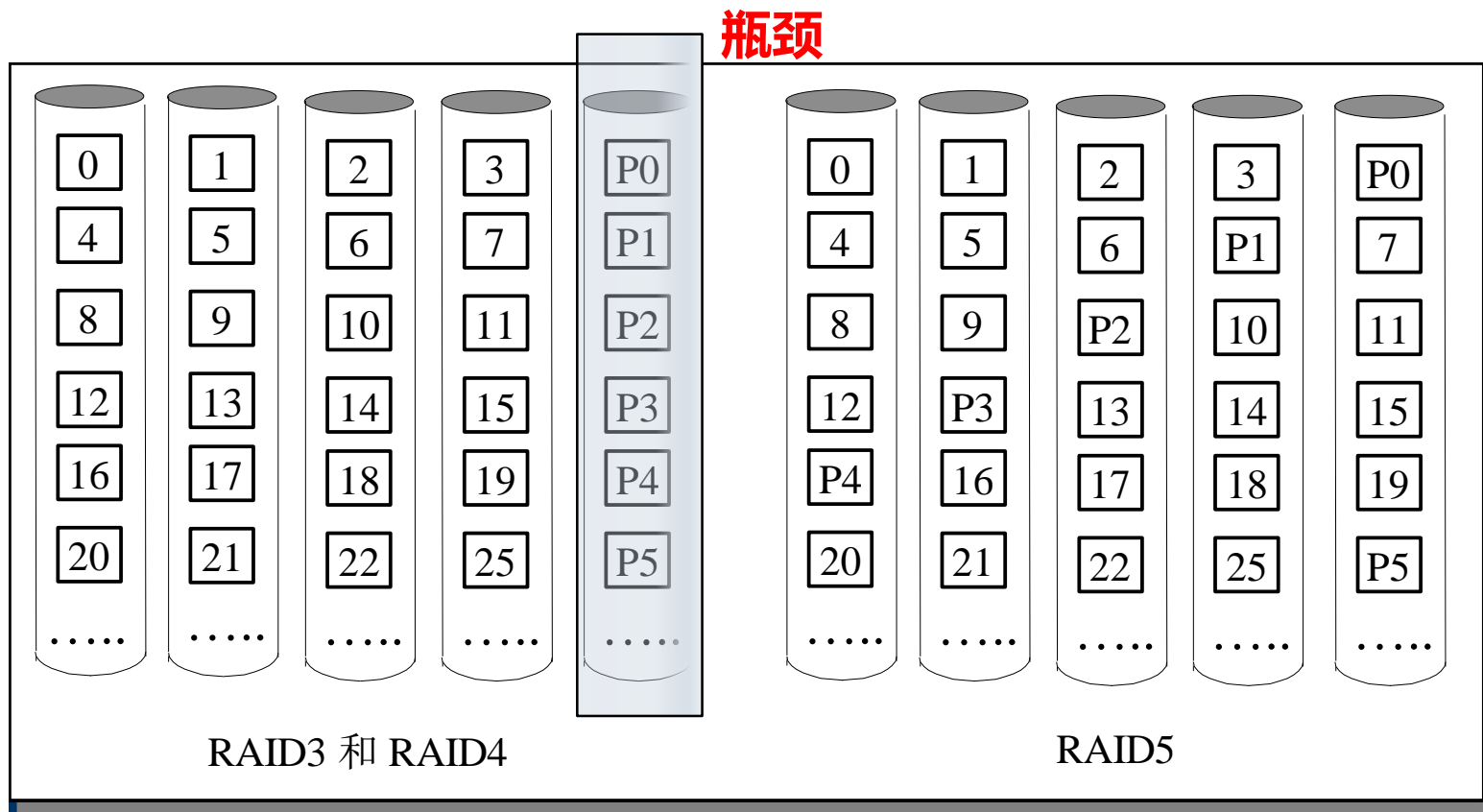


RAID-5 (block-interleaved distributed parity)

- ❑ 块交叉分布式奇偶校验盘阵列
- ❑ 数据以块交叉的方式存于各盘，无专用冗余盘，奇偶校验信息均匀分布在所有磁盘上。

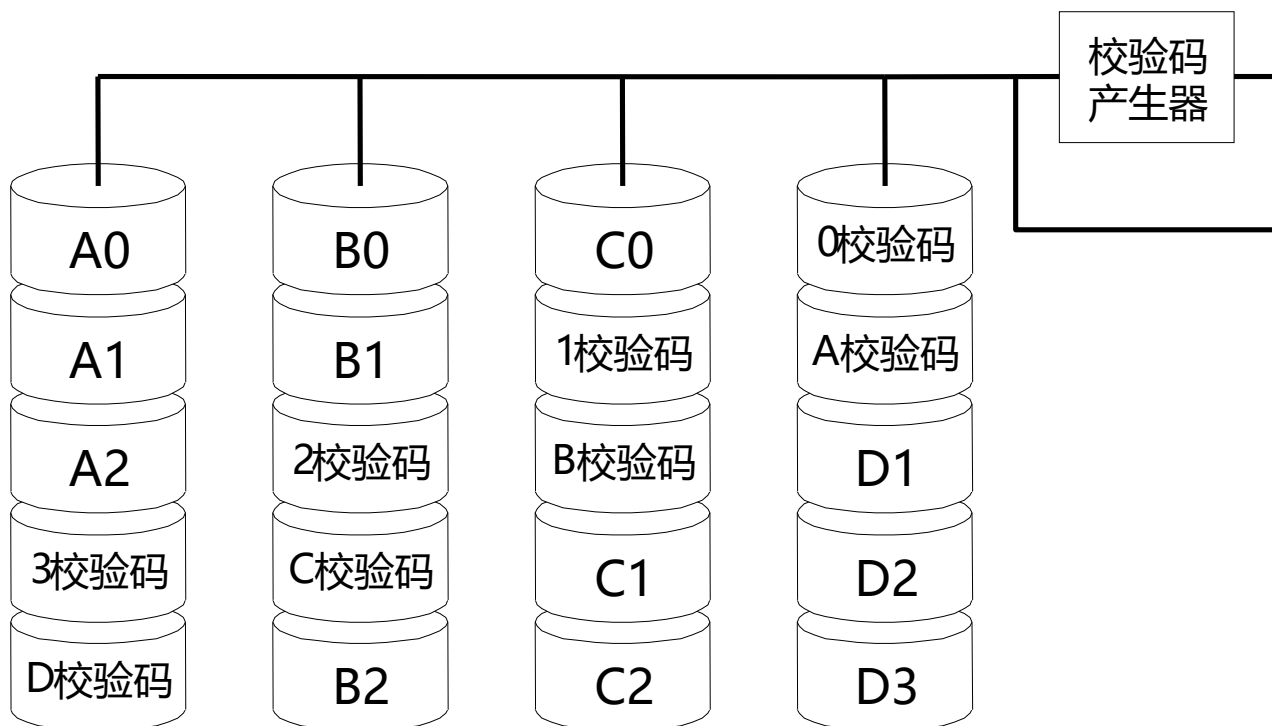


RAID-5的校验信息分布



RAID-6

- ❑ 双维奇偶校验(P+Q校验)独立存取盘阵列
- ❑ 数据以块(块大小可变)交叉方式存于各盘，检、纠错信息均匀分布在所有磁盘上。



RAID-6特点

- ❑ 读操作与RAID-5类似;
- ❑ 可容忍双盘出错, 适用于高可靠性要求的场景;
- ❑ 存储开销是RAID-5的两倍;
- ❑ RAID-6的写过程需要6次磁盘操作。
 - 过程是怎么样的呢?

RAID的实现与发展

□ 实现盘阵列的方式主要有三种

■ 软件方式：阵列管理软件由主机来实现

- 优点：成本低

- 缺点：过多地占用主机时间，并且带宽指标上不去。

■ 阵列卡方式

- 把RAID管理软件固化在I/O控制卡上

- 从而可不占用主机时间，一般用于工作站和PC机。

■ 子系统方式

- 这是一种基于通用接口总线的开放式平台，可用于各种主机平台和网络系统。

RAID的实现与发展

REFINE YOUR SEARCH

Refine by Keywords

Refine by People

[Names](#)
[Institutions](#)
[Authors](#)
[Editors](#)
[Reviewers](#)


Refine by Publications

[Publication Year](#)
[Publication Names](#)
[ACM Publications](#)
[All Publications](#)
[Content Formats](#)
[Publishers](#)


Refine by Conferences

[Sponsors](#)
[Events](#)
[Proceeding Series](#)

ADVANCED SEARCH

 [Advanced Search](#)

CONTACT US

 [Contact The DL Team](#)

Found 3,145 of 437,879

Search Results

Related Journals

Related Magazines

Related SIGs

Related Conferences

Results 1 - 20 of 3,145

Sort by [relevance](#) in [expanded form](#)

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#) [>>](#)


- 1 [A new intra-disk redundancy scheme for high-reliability RAID storage systems in the presence of unrecoverable errors](#)

 [Ajay Dholakia](#), [Evangelos Eleftheriou](#), [Xiao-Yu Hu](#), [Ilias Iliadis](#), [Jai Menon](#), [K.K. Rao](#)

May 2008

Transactions on Storage (TOS), Volume 4 Issue 1

Publisher: ACM  [Request Permissions](#)


Full text available:  [PDF](#) (1.32 MB)

Bibliometrics: Downloads (6 Weeks): 8, Downloads (12 Months): 65, Downloads (Overall): 881, Citation Count: 21

Today's data storage systems are increasingly adopting low-cost disk drives that have higher capacity but lower reliability, leading to more frequent rebuilds and to a higher risk of unrecoverable media errors. We propose an efficient intradisk redundancy ...

Keywords: File and I/O systems, RAID, reliability analysis, stochastic modeling


- 2 [Disk Scrubbing Versus Intradisk Redundancy for RAID Storage Systems](#)

 [Ilias Iliadis](#), [Robert Haas](#), [Xiao-Yu Hu](#), [Evangelos Eleftheriou](#)

July 2011

Transactions on Storage (TOS), Volume 7 Issue 2

Publisher: ACM  [Request Permissions](#)

Full text available:  [PDF](#) (643.38 KB)

Bibliometrics: Downloads (6 Weeks): 5, Downloads (12 Months): 54, Downloads (Overall): 335, Citation Count: 7

Two schemes proposed to cope with unrecoverable or latent media errors and enhance the reliability of RAID systems are examined. The first scheme is the established, widely used, disk scrubbing scheme, which operates by periodically accessing disk drives ...

Keywords: MTTDL, RAID, Unrecoverable or latent sector errors, reliability analysis, stochastic modeling

- 3 [Disk scrubbing versus intra-disk redundancy for high-reliability raid storage systems](#)

内容概要

6.1 引言

6.2 外部存储设备

6.3 可靠性、可用性和可信性

6.4 廉价磁盘冗余阵列RAID

6.5 I/O设备与CPU和存储器的连接

6.6 I/O与操作系统

6.5 I/O设备与CPU和存储器的连接

- 通过接口将许多子系统连接起来
- 接口使用的最多的就是总线
- 输入输出方式：
 - 程序查询
 - 中断
 - DMA
 - 通道

总线 (bus)

□ 优点：低成本、多样性

- BitBus, MultiBus, UniBus, PC-Bus(8), AT-Bus(16), PCI, Compact PCI, PCI-E, PCI-X, VXI, PXI, SCXI, RS-232C, RS-422, RS-485, USB, 1394, SATA, PATA, SCSI, AMBA ...

□ 缺点：必须独占使用，造成了设备信息交换的瓶颈，从而限制了系统中总的I/O吞吐量。

- 多核系统逐渐转到基于NoC的架构上的主要原因

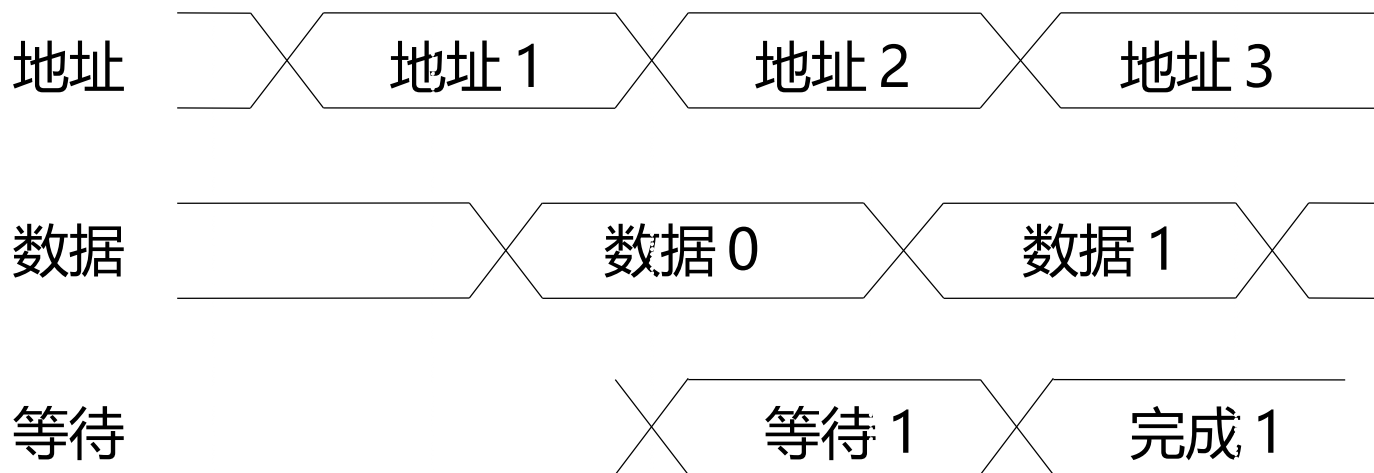
总线的设计

| 选择 | 高性能 | 低价格 |
|--------|-------------------|-------------------|
| 总线宽度 | 独立的地址和数据总线 | 分时复用数据和地址总线 |
| 数据总线宽度 | 越宽越快 (例如: 64位) | 越窄越便宜 (例如: 8位) |
| 传输块大小 | 块越大总线开销越小 | 每次传送单字 |
| 总线主设备 | 多个 (需要仲裁) | 单个 (无需仲裁) |
| 分离事务 | 采用 | 不用 |
| 定时方式 | 同步 | 异步 |

总线的设计

□ 分离事务总线

- 在有多个主设备时，总线通过数据打包来提高总线带宽（将总线事务分为请求和应答），而不必在整个传输过程中都占有总线。



总线的设计

□ 同步总线

- 同步总线上所有设备使用统一的总线系统时钟。
- 优点：成本低，因为它不需要设备之间互相确定时序的逻辑。
- 缺点：总线操作必须以相同的速度运行。

□ 异步总线

- 异步总线上的设备之间没有统一的系统时钟，设备自己内部定时。

几种常用的并行总线

| | IDE / Ultra ATA | SCSI | PCI | PCI-X |
|-------------|--------------------|---|------------|-------------------|
| 数据宽度 | 16bit | 8 / 16bit | 32 / 64bit | 32 / 64bit |
| 时钟频率 MHz | 100 | 10 (Fast) 20 (Ultra) 40 (Ultra2) 80 (Ultra3) 160 (Ultra4) | 33 / 66 | 66 / 100 / 133 |
| 主设备数量 | 一个 | 多个 | 多个 | 多个 |
| 峰值带宽 | 200MB/s | 320MB/s | 533MB/s | 1066MB/s |
| 同步方式 | 异步 | 异步 | 同步 | 同步 |
| 标准 | 无 | ANSI X3.131 | 无 | 无 |

并行总线宽度大，适合于速度要求高的场合。

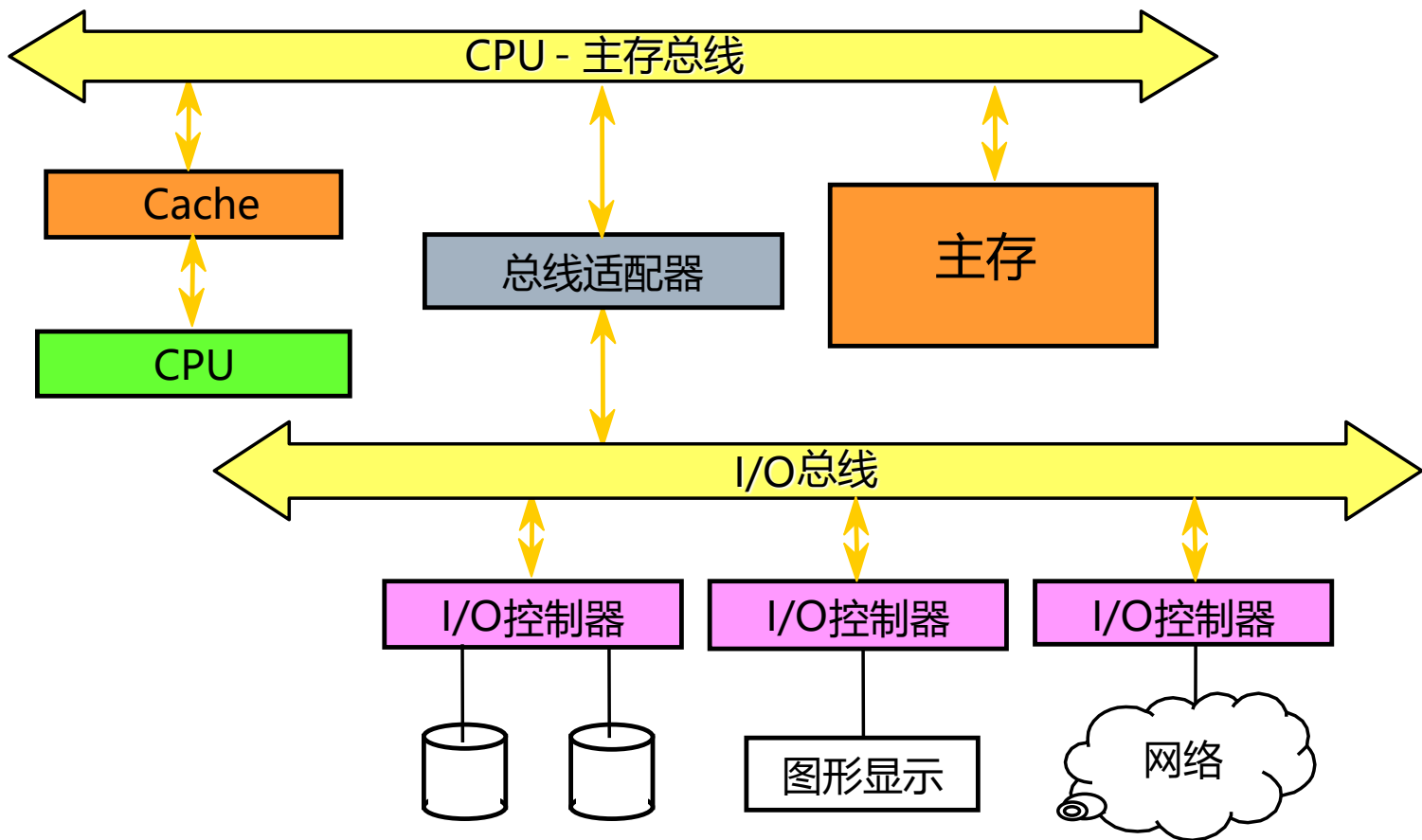
几种常用的串行总线

| | I ² C | 1-wire | RS-232 | SPI |
|-------------|------------------|--------|--------------------|-----|
| 数据宽度 (bit) | 1 | 1 | 2 | 1 |
| 信号线数量 | 2 | 1 | 9 / 25 | 3 |
| 时钟频率 (MHz) | 0.4 ~ 10 | 异步 | 0.04或异步 | 异步 |
| 总线主设备数量 | 多个 | 多个 | 多个 | 多个 |
| 峰值带宽 (Mb/s) | 0.4 ~ 3.4 | 0.014 | 0.192 | 1 |
| 同步方式 | 异步 | 异步 | 异步 | 异步 |
| 标准 | 无 | 无 | EIA, ITU-T V.21 | 无 |

串行总线宽度小，适用于嵌入式系统。如基于RS-232的COM1 COM2接口。

设备的连接

□ 典型的总线连接



设备的连接

□ CPU寻址I/O设备的方式

- 存储器映射I/O或统一编址
- I/O设备单独编址
- 无论选择哪一种编址方法，每个I/O设备都提供状态寄存器和控制寄存器。

□ 设备的连接和工作方式

- 程序查询、中断、DMA、通道处理机

设备的连接

□ 程序查询、中断和DMA方式管理外围设备会

引起两个问题：

- 所有外围设备的I/O工作都要涉及CPU，CPU的I/O负担很重，不能专心于用户程序的计算。
- 大型计算机系统中的外围设备台数众多，CPU管理的开销巨大。

□ 解决上述问题的方法：

- 采用通道处理机

通道

- 通道处理机能够负担外围设备的大部分I/O工作。
- 通道处理机：**能够执行有限I/O指令，并且能够被多台外围设备共享的小型DMA专用处理机。**
- 通道的功能：
 - ① 接受CPU发来的I/O指令，从而指定外围设备与通道相连接；
 - ② 执行通道程序，从主存中取出通道指令，对通道指令进行译码，向被选中的设备控制器发出操作命令；
 - ③ 给出外围设备的有关地址，即进行读 / 写操作的数据所在的位置。如磁盘存储器的柱面号、磁头号、扇区号等；

通道

□ 通道的功能:

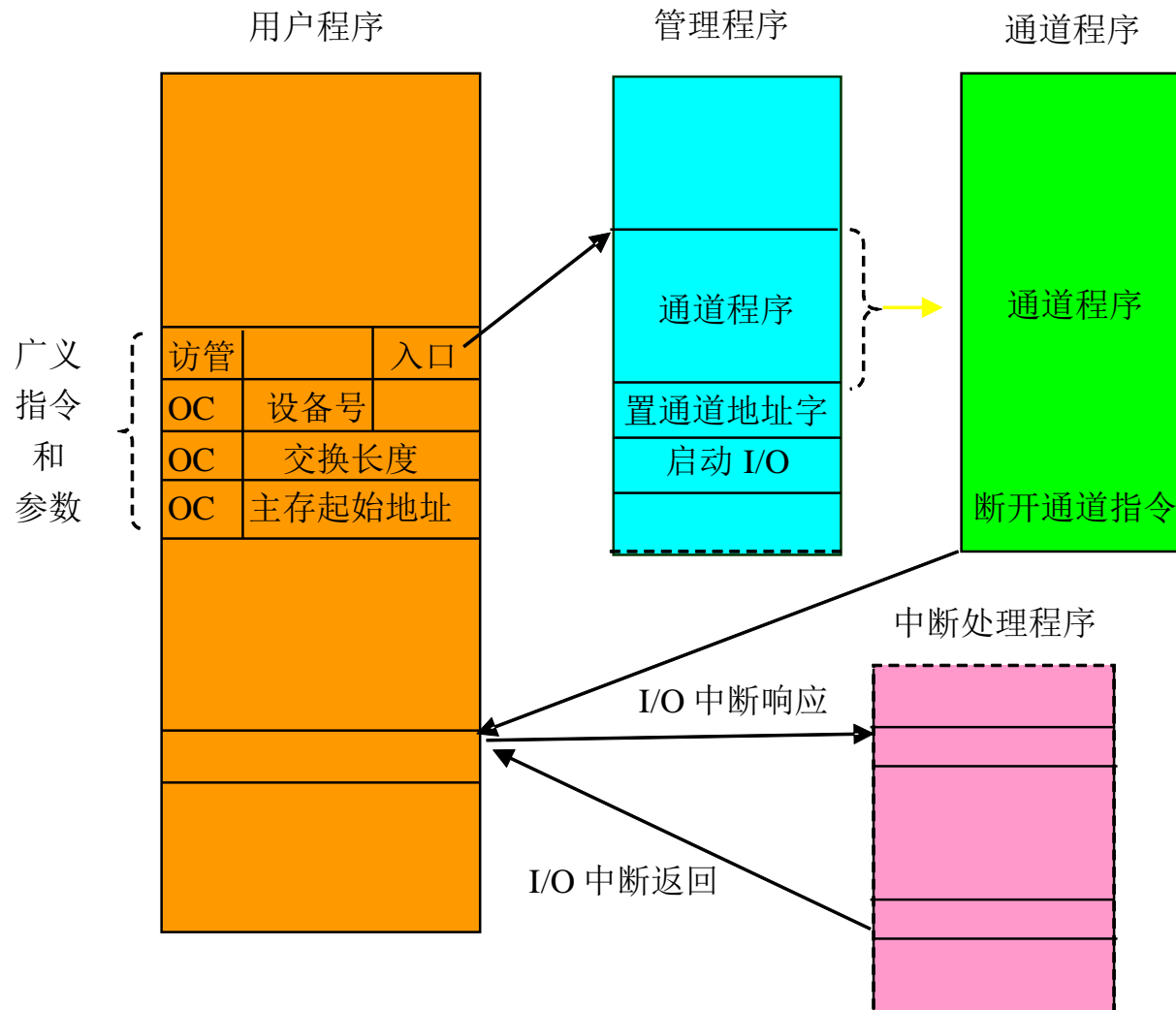
- ④ 给出主存缓冲区的首地址，存放从外围设备上输入/将要输出到外围设备中去的数据。
- ⑤ 控制外围设备与主存缓冲区之间数据交换的个数，并进行计数，并判断数据传送工作是否结束。
- ⑥ 指定传送工作结束时要进行的操作。
- ⑦ 检查外围设备的工作状态，是正常或故障。根据需要将设备的状态信息送往主存指定单元保存。
- ⑧ 在数据传输过程中完成必要的格式变换。例如：把字拆卸为字节，或者把字节装配成字。

通道

□ 通道对外围设备的控制通过I/O接口和设备控制器进行，工作过程如下：

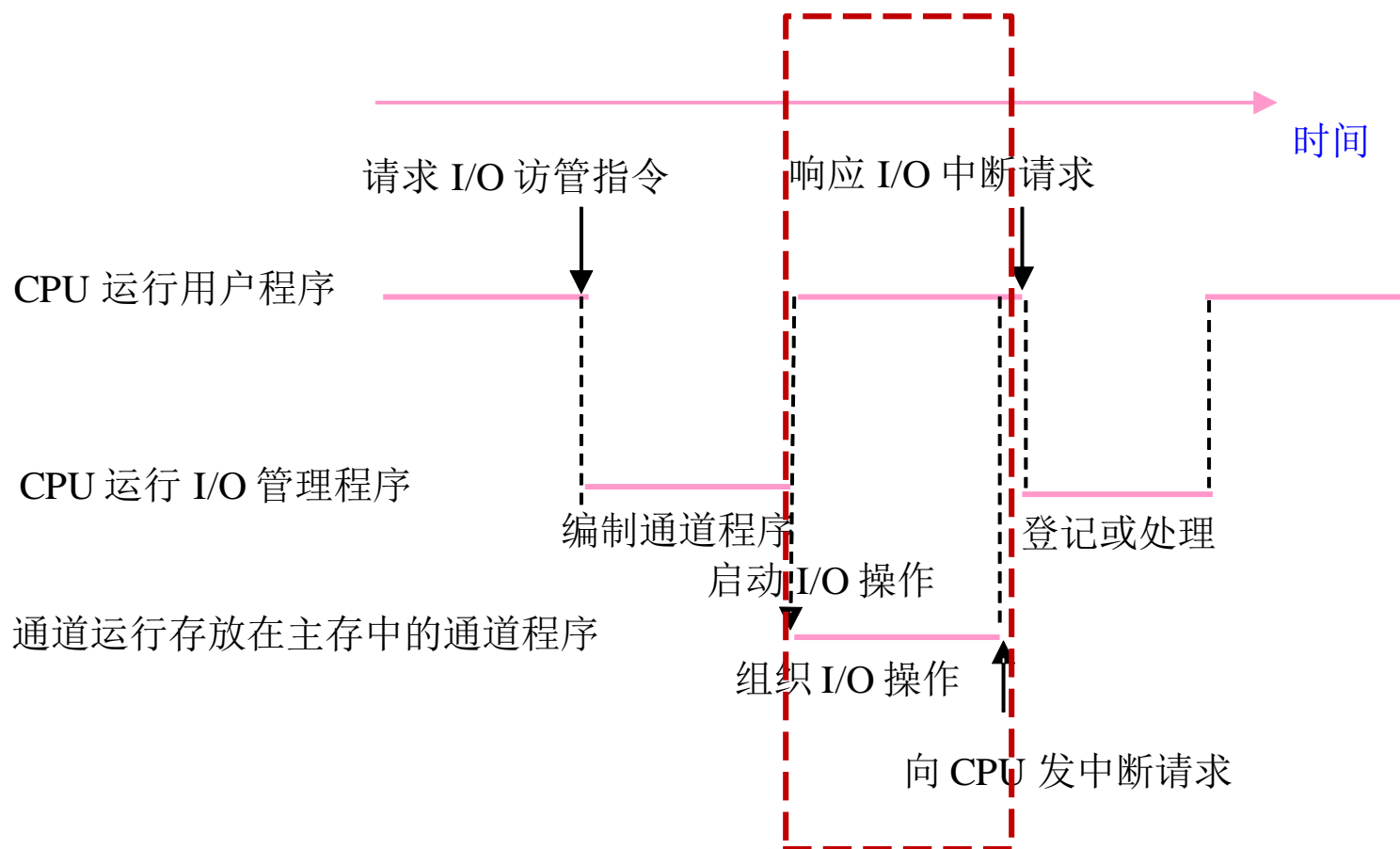
1. 在用户程序中使用**访管指令**进入管理程序，由CPU通过管理程序组织一个**通道程序**，并启动通道。
2. 通道处理机执行CPU为它组织的通道程序，完成指定的数据I/O工作。
 - **通道处理机执行通道程序是与CPU执行用户程序并行的。**
3. 通道程序结束后向CPU发**中断请求**。CPU响应这个中断请求后，第二次进入操作系统，调用管理程序对I/O中断请求进行处理。

通道



通道

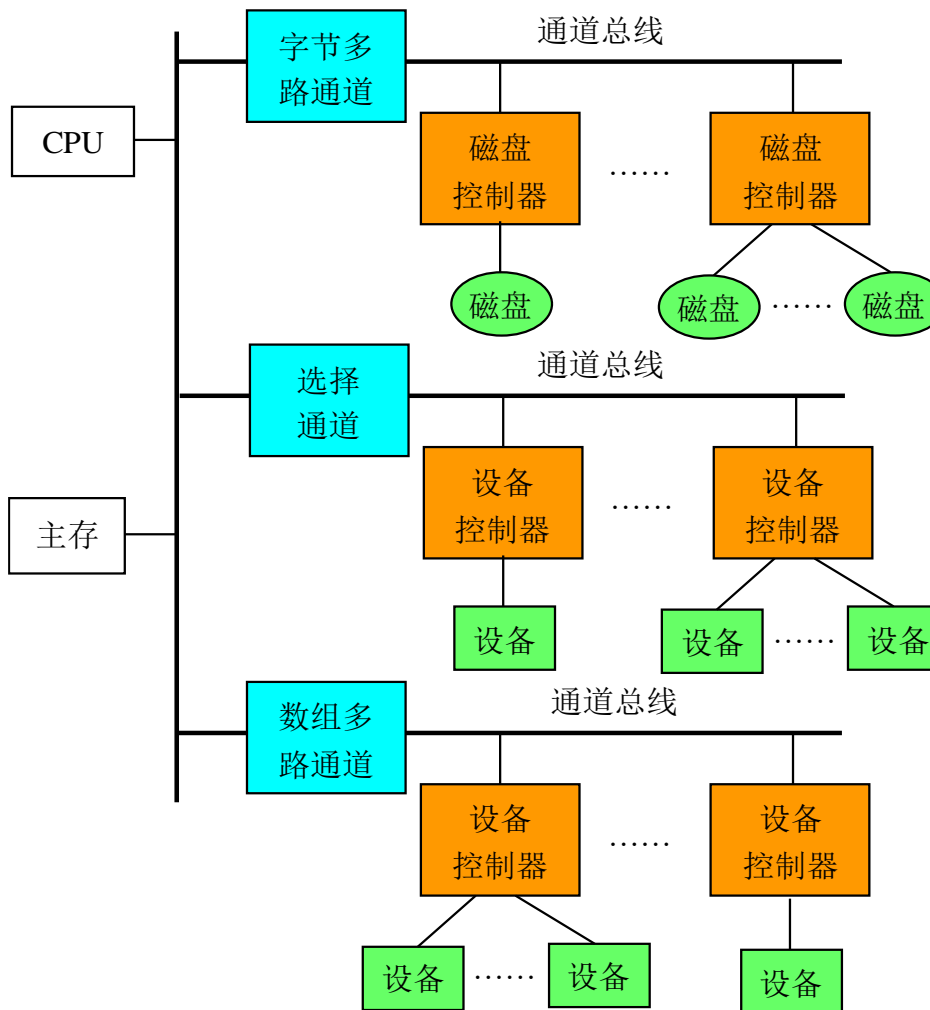
□ 通道程序、管理程序和用户程序的执行时间



通道

□ 通道种类，通常分为3类：

- 字节多路通道
- 选择通道
- 数组多路通道



通道

□ 字节多路通道

- 为多台**低速或中速**的外围设备服务;
- 采用**分时方式**工作, 依靠它与CPU之间的高速数据通路分时为多台设备服务。

□ 选择通道

- 为多台**高速**外围设备服务;
- 传送数据期间, 通道**只能为一台高速外围设备服务**, 在不同时间内可以选择不同设备。

通道

□ 数组多路通道

- 数组多路通道适于为**高速**设备服务；
- 每次选择一个高速设备后**传送一个数据块**，并轮流为多台外围设备服务；
- 数组多路通道之所以能够并行地为多个高速外围设备服务，是因为这些高速外围设备并不能在整个数据输入输出时间内单独利用通道的全部传输能力。

通道

□ 通道中的数据传送过程

■ 字节多路通道的数据传送过程

- 通道每连接一个外围设备，只传送一个字节，然后又与另一台设备连接，并传送一个字节。

■ 数组多路通道的数据传送过程

- 每连接一台高速设备，传送一个数据块，传送完成后，又与另一台高速设备连接，再传送一个数据块。

■ 选择通道的工作过程

- 每连接一个外围设备，就把这个设备的 n 个字节全部传送完成，然后再与另一台设备相连接。

通道

□ 通道中数据的传送过程与流量分析

- 通道流量：一个通道在数据传送期间，单位时间内能够传送的最大数据量，一般用**字节个数**来表示。又称为通道吞吐率，通道数据传输率等。
- 通道最大流量：一个通道在满负荷工作状态下的流量。

通道

□ 流量计算公式

- T_S : 设备选择时间。
- T_D : 传送一个字节所用的时间。
- p : 在一个通道上连接的设备台数, 且**这些设备同时都在工作**。
- n : 每台设备传送的字节数, **这里为了方便分析假设每台设备传送的字节数都相同**。
- k : 数组多路通道传输的一个数据块中的包含的字节数。在一般情况下, $k < n$ 。对于磁盘、磁带等磁表面存储器, 通常 $k = 512$ 。
- T : 通道完成全部数据传送工作所需时间。

通道

□ 传输时间计算公式：

■ 字节多路通道：

$$T_{\text{BYTE}} = (T_S + T_D) \times p \times n$$

■ 选择通道：

$$T_{\text{SELECT}} = \left(\frac{T_S}{n} + T_D \right) \times p \times n$$

■ 数组多路通道：

$$T_{\text{BLOCK}} = \left(\frac{T_S}{k} + T_D \right) \times p \times n$$

通道

□ 最大流量

■ 字节多路通道

$$f_{\text{MAX-BYTE}} = \frac{pn}{(T_S + T_D)pn} = \frac{1}{T_S + T_D}$$

■ 选择通道

$$f_{\text{MAX-SELECT}} = \frac{pn}{(\frac{T_S}{n} + T_D)pn} = \frac{1}{\frac{T_S}{n} + T_D}$$

■ 数组多路通道

$$f_{\text{MAX-BLOCK}} = \frac{pn}{(\frac{T_S}{k} + T_D)pn} = \frac{1}{\frac{T_S}{k} + T_D}$$

通道

□ 实际流量小于最大流量

■ 字节多路通道

$$f_{\text{BYTE}} \leq f_{\text{MAX-BYTE}}$$

■ 选择通道

$$f_{\text{SELECT}} \leq f_{\text{MAX-SELECT}}$$

■ 数组多路通道

$$f_{\text{BLOCK}} \leq f_{\text{MAX-BLOCK}}$$

Next Topic

Interconnection Network