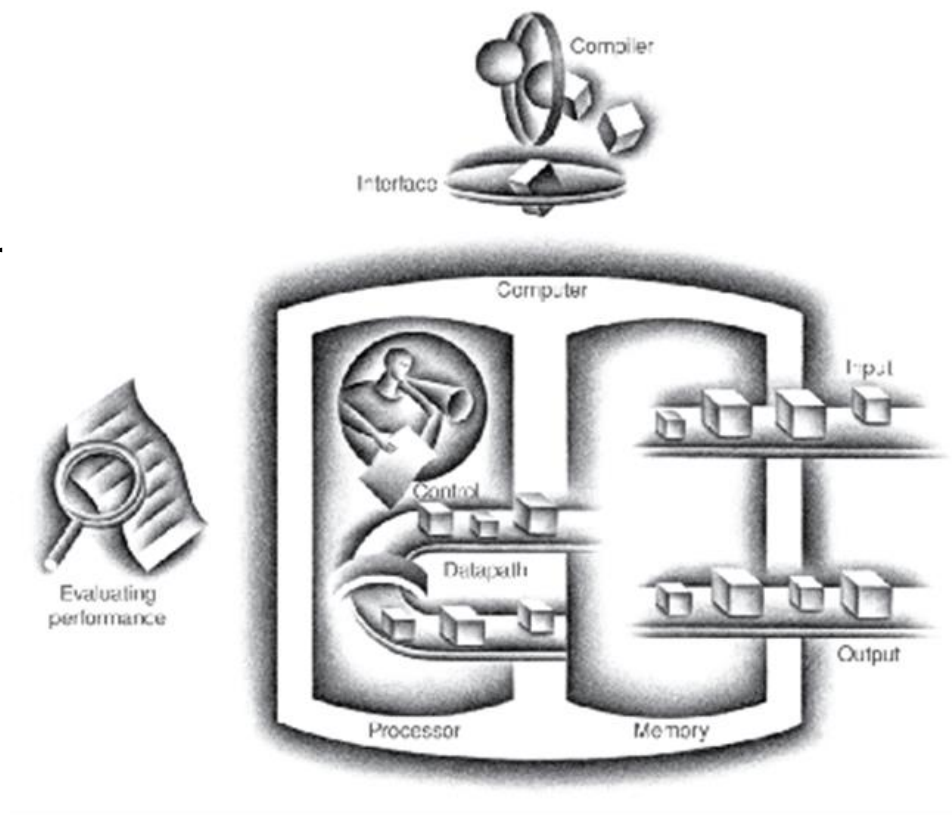


计算机组成与体系结构

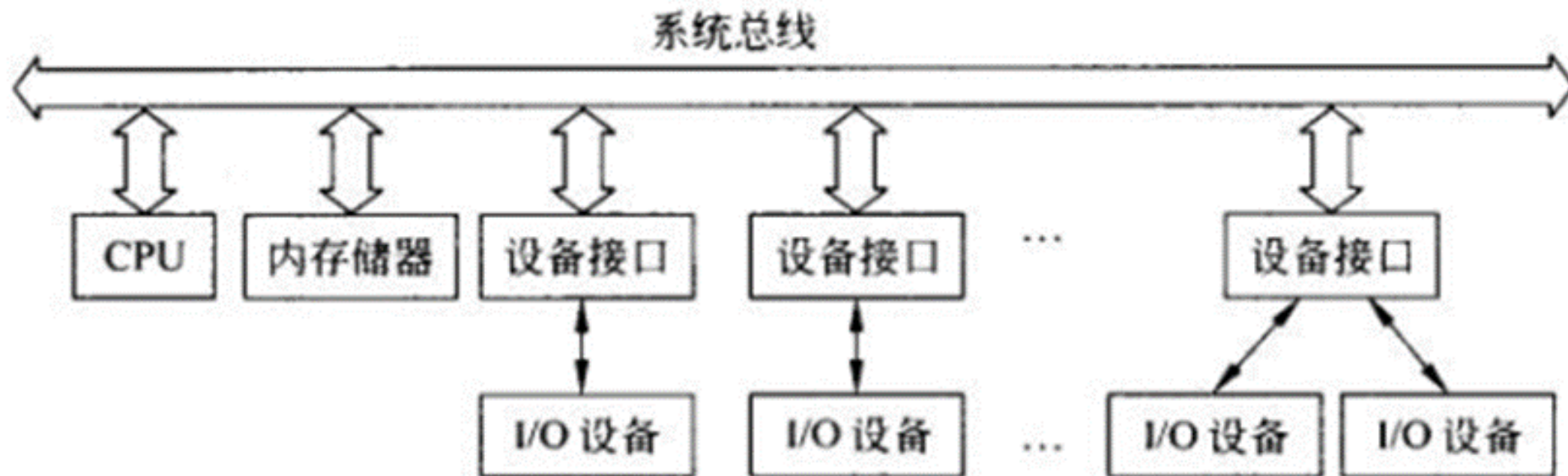
Chapter 6 输入输出系统

6.1 输入输出方式 6.1.1 输入输出系统及其作用

- 计算机系统中分布在主机周围的
- 各种I/O设备及其相关电路和控制软件
- 构成了计算机系统的I/O系统,
- 其作用就是实现数据的输入、输出。
- 基本的I/O设备包括:
- 键盘、显示器、鼠标
- 各种外存储器、移动存储器
- 网络接口设备等



输入输出系统典型结构



系统总线是系统中需要用到的各种信号线的集合

输入输出方式

- 为满足不同应用的需要，主机和I/O设备之间的数据传送方式可以分为：
- 无条件传送方式
- 查询方式
- 中断方式
- DMA方式等
- 各有各的特点、优缺点；使用场景

6.1.2 输入输出系统设计的重要原则

- 抽象原则
- I/O设备虽然因为应用领域的不同而具有多样性和复杂性，但是它们和主机之间交换数据时的数据格式、传输方式等却不会太多。
- 抽象原则就是要对不同I/O设备的工作机制进行分析，从多样性中发现同一性，找到具有共性的东西，经过概括、总结后，得出系统设计需要解决的基本问题，建立相应的抽象模型。

6.1.2 输入输出系统设计的重要原则

- **分治原则**

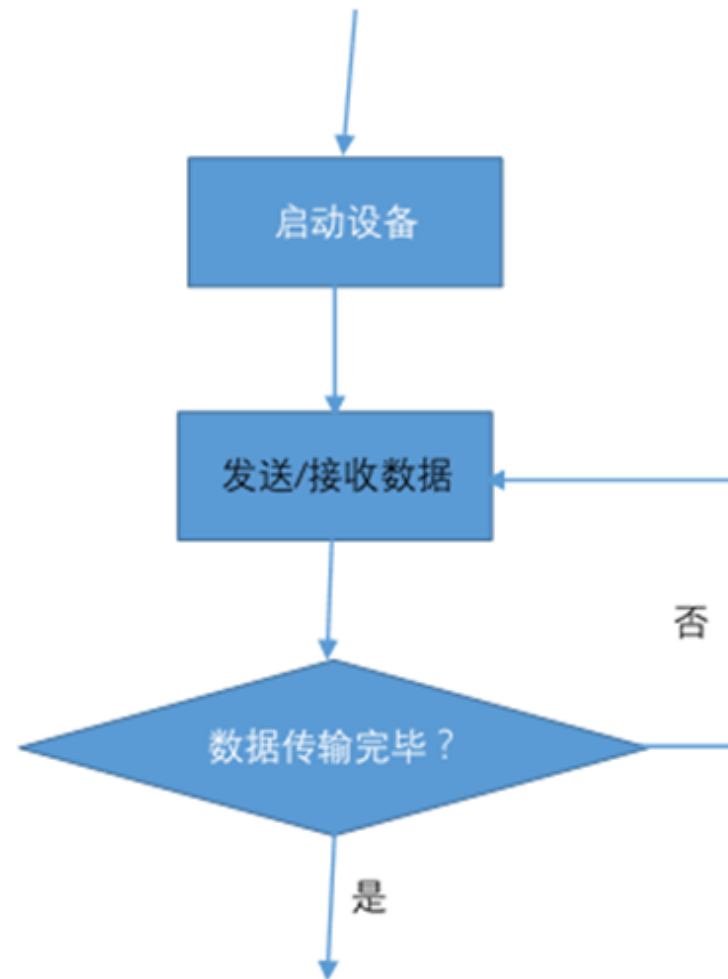
- 输入输出是主机和I/O设备双方的事。分治原则最重要的含义就是利用I/O接口在一定程度上将双方隔离开来，任意一方直接打交道的对象是接口，接口负责沟通双方。双方协作、互相配合通过I/O接口实现数据交换，各司其职。
- 在设计时，主要解决好：1) 主机一侧和接口之间的数据交换问题；2) I/O设备一侧和主机之间的数据交换问题；3) 利用接口实现双方的同步问题。

6.1.2 输入输出系统设计的重要原则

- **标准原则**
- 设计输入输出系统应该遵循一定的标准，基本要求是：可靠、可用、可信。
- 同时，设计成果也应该进行标准化处理，形成各种标准或规范，供开发新的I/O设备时遵照执行。

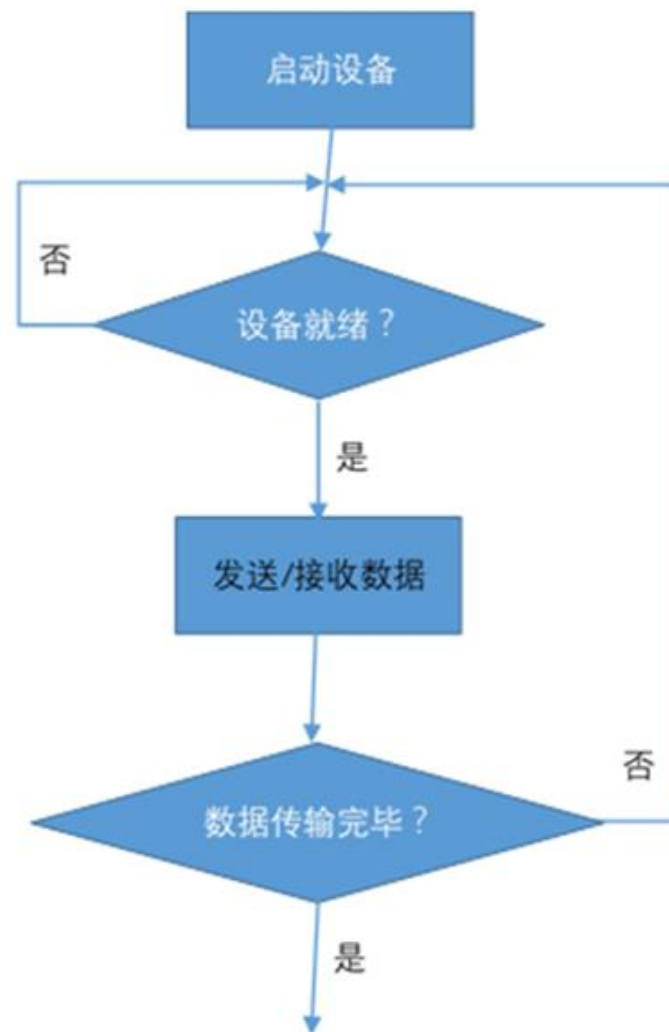
6.1.3 无条件传送方式

- 数据传送方式：无条件、查询
- 中断、DMA等
- 无条件传送方式是指
- 双发发送/接收数据是无条件的
- 任何时候想发就发，想收就收。
- 无条件传送方式的
- 数据发送/接收流程如图



6.1.4 查询方式

- 双方发送/接收数据是有条件的,
- 在发送/接收数据之前必须检测
- 是否满足发送/接收数据的条件,
- 只有具备发送/接收数据的条件
- 才能发送/接收数据。
- 对主机来说, 每次检测发送/接收数据的条件
- 就是通过读取接口内部的状态位
- (位于状态端口内) 并且对其进行分析实现的,



同步问题

- 一般来说，双方通过接口交换数据时，需要做好同步工作，基本规则是：先发数据，再收数据，再发数据，再收数据，……。
- 为此，发送方在发送数据前需要知道接收方是否已经准备好接收数据，而接收方需要知道是否有数据需要接收。
- 确保双方按此规则发送/接收数据的基本手段就是设置专门的控制联络信号供双方使用，

握手与握手协议

- 双方利用控制联络信号实现同步也叫握手，
- 控制联络信号也叫握手信号。
- 握手需要遵守的规则也叫握手协议，
- 换言之，握手协议就是关于
- “谁先做什么，然后谁再做什么；如果怎么样，就怎么样；否则又该怎么样；……”的一套规则。
- 双方必须严格遵守握手协议，否则可能导致错误。

6.1.5 中断方式

- 查询方式的缺点
- 主机发送/接收数据之前都要查询条件，
- 如果条件不满足，就得继续查询，
- 直到条件满足才能发送/接收一个数据。
- 为发送/接收下一个数据，还得先查询条件。
- 这样一来，查询条件会耗费大量的CPU时间。

中断

- 所谓中断，就是指由于系统内外发生了什么事情，
- CPU暂停正在执行的程序转而去处理所发生的事，
- 等处理完后，恢复执行原来暂停的程序。
- 换言之，原来正常执行的程序被发生的事情暂时中断了。
- 中断是I/O设备和主机交互的一种很重要的方式
- 与中断相联系的另一个重要概念是**异常**。
- **异常**是指异常事件引起的中断，

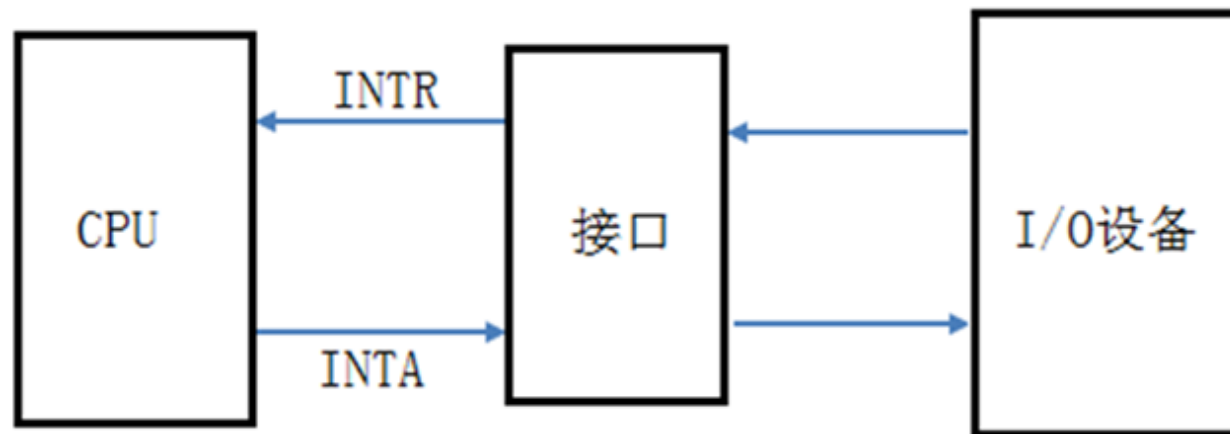
中断的基本问题： 中断请求、中断检测、中断响应、中断服务、中断返回

- **中断请求**：以中断方式和主机交互的I/O设备在适当的时候，比如输入设备准备好数据后，会通过中断请求信号线向CPU发送中断请求信号，通知CPU来取数据。
- **中断检测**：CPU在正常执行程序的过程中会定期对中断请求信号线进行检测。
- **中断响应**：CPU如果检测到中断请求信号，在系统允许中断的情况下，会通过中断响应信号线向发出中断请求的I/O设备发送中断响应信号，通知I/O设备其中断请求得到响应，并暂停正在执行的程序，转而进入中断服务程序。
- **中断服务**：执行中断服务程序，处理所发生的事，比如发送/接收数据。
- **中断返回**：结束执行中断服务程序，转而回到原来暂停执行的程序。

中断服务程序与中断源

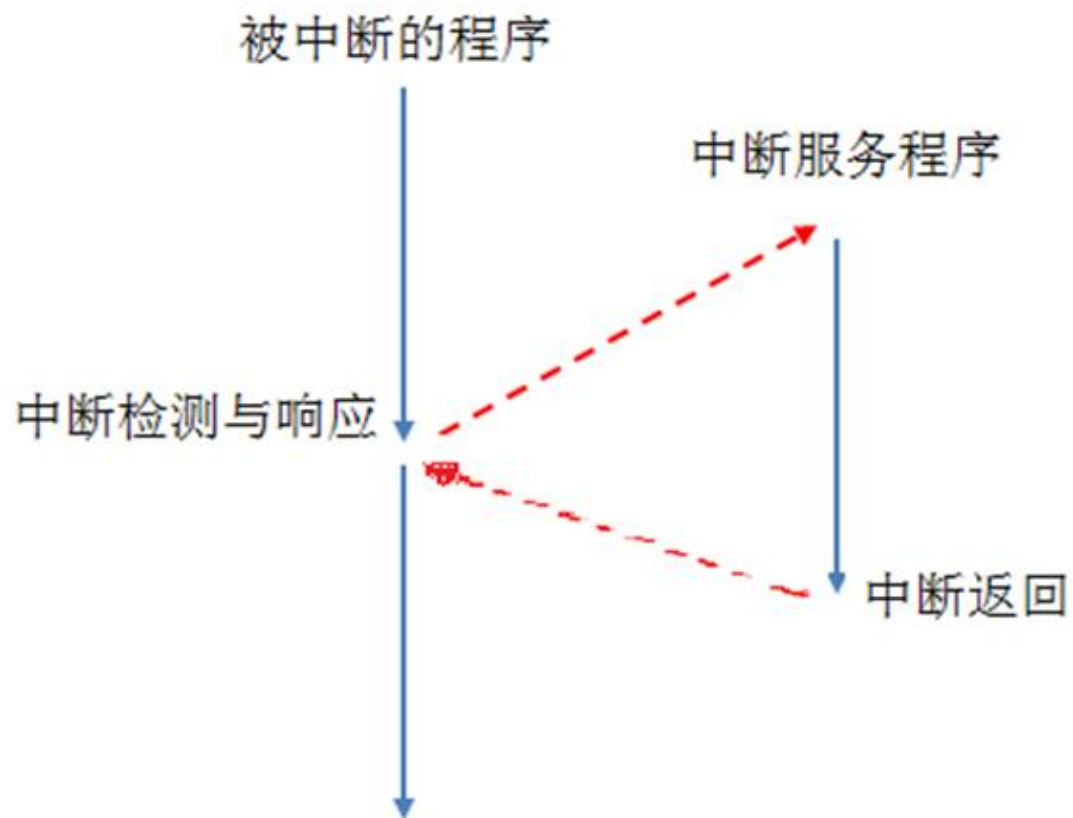
- 中断服务程序：事先编写好的一段程序，CPU通过执行中断服务程序来处理所发生的事，比如发送/接收数据。
- 中断源：系统中能够引起中断的各种原因（事件），即中断的来源。

单中断系统



- INTR和INTA分别为中断请求信号线和中断响应信号线。

单中断处理的一般过程



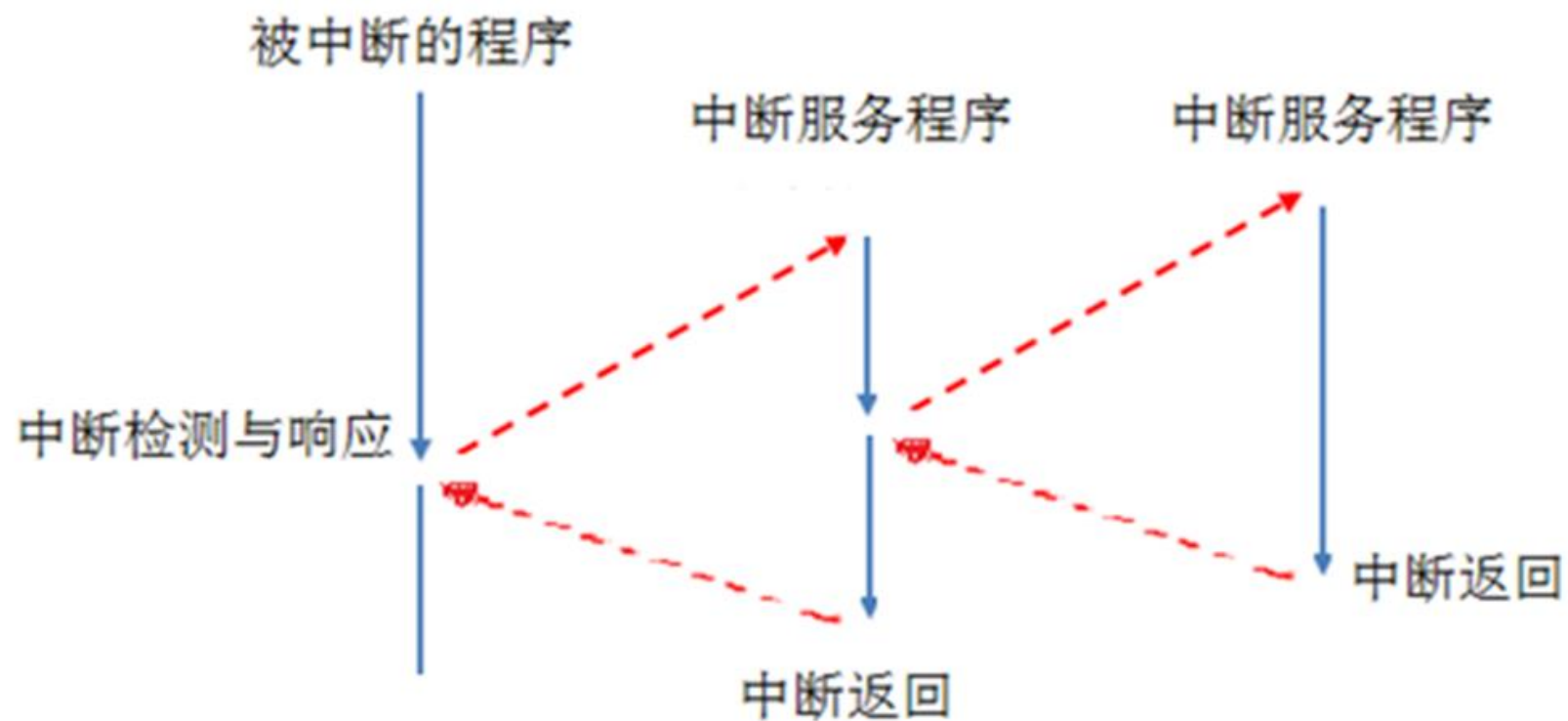
多中断系统

- 系统中有多多个中断源
- 同时有多多个中断请求时，该优先处理谁呢？
- 解决的办法就是给不同的中断源赋予不同的优先级。当不同优先级的中断源同时产生中断请求时，优先处理级别最高的中断请求。
- **中断优先级**：通常用无符号数来区分不同的中断源的优先级别。

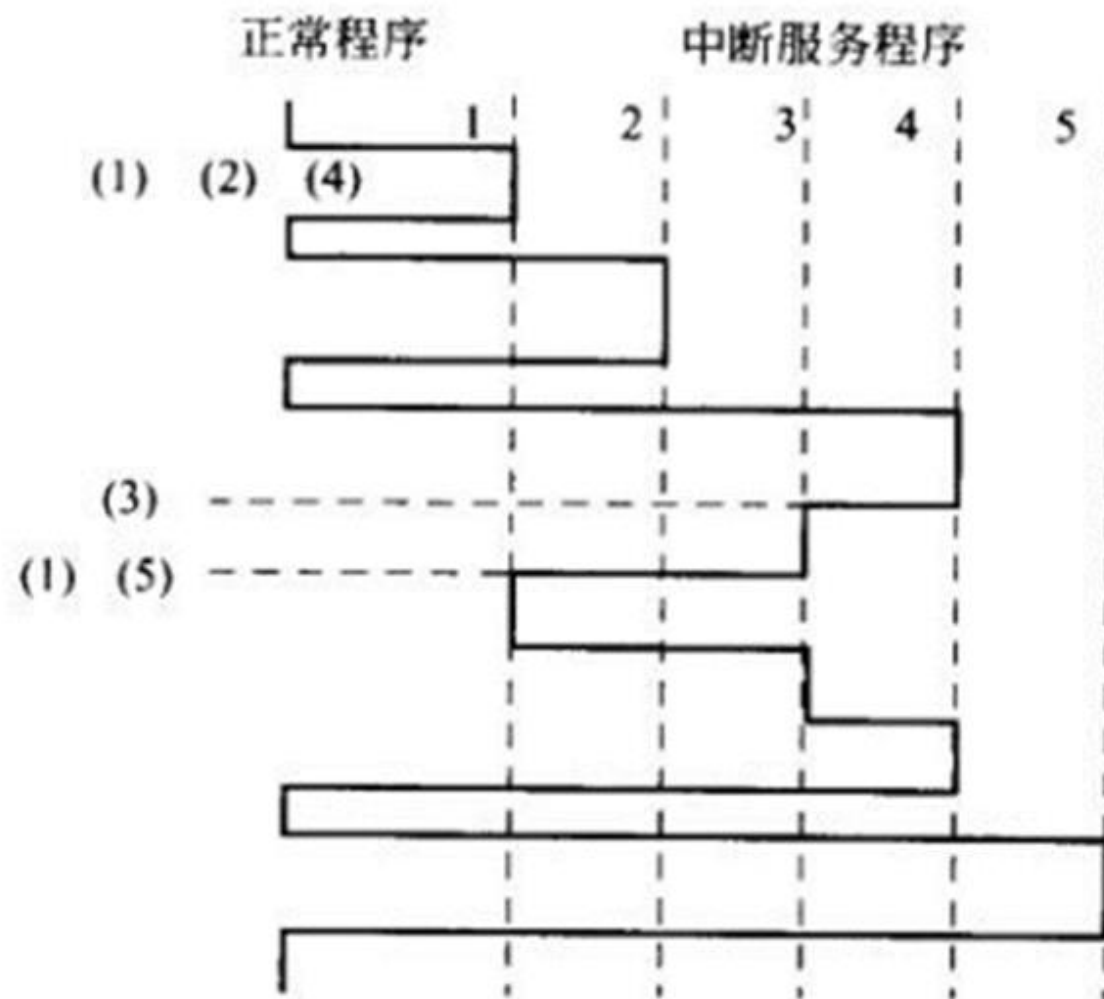
中断嵌套

- 在处理当前级别最高的中断请求期间，如果来了级别更高的新的中断请求，又该如何处理呢？
- 如果允许中断处理又被中断，正在处理的中断就会被新来的更高级别的中断中断，从而形成嵌套。
- **中断嵌套**：在执行某个中断服务程序的过程中，由于某种原因，需要暂停正在执行的中断服务程序，转而去处理新的中断，等新中断处理完毕后，又恢复执行被暂停的中断

中断嵌套原理

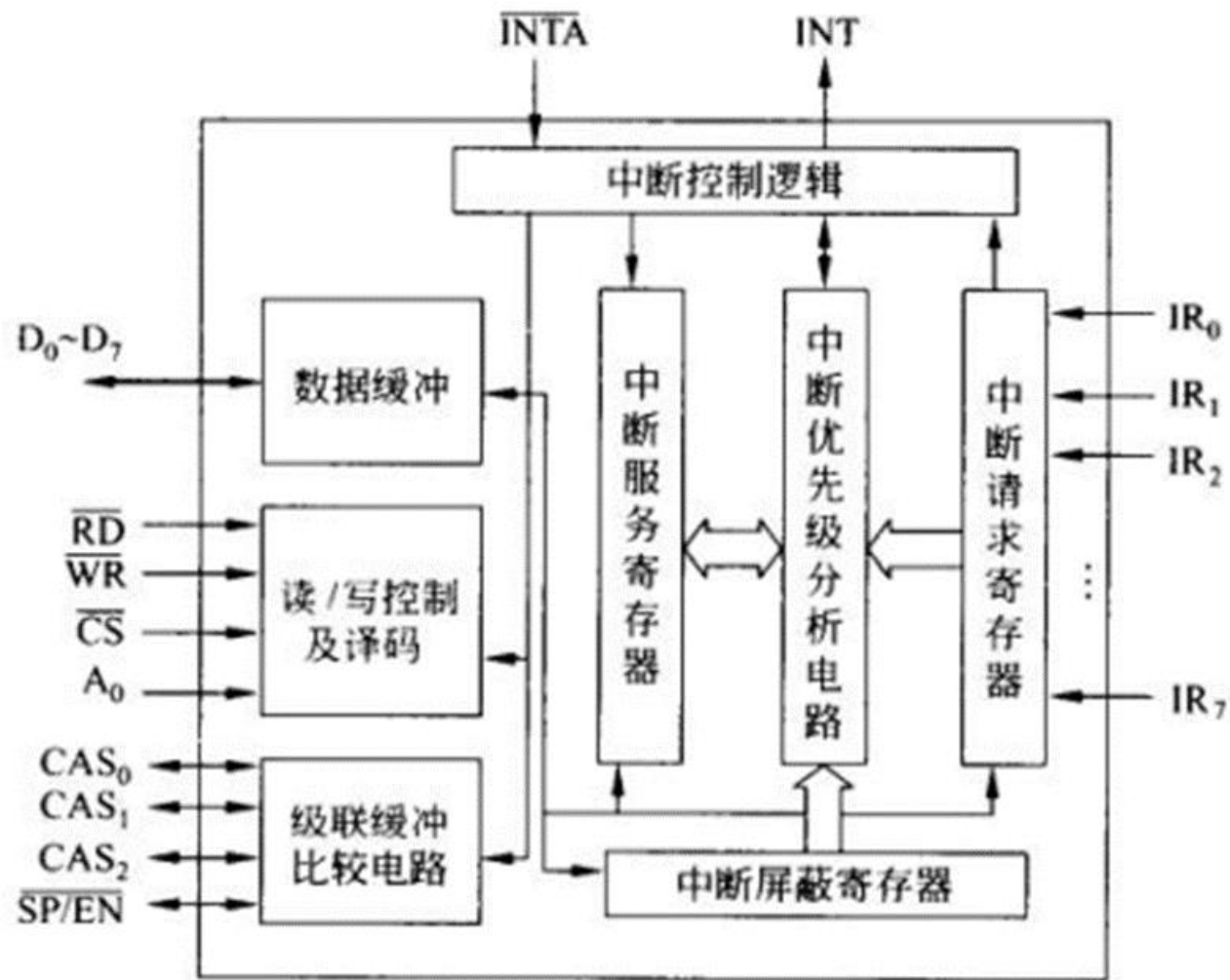


多中断处理



中断管理器

- 系统中专门负责集中
- 管理各个中断源的部件



- **中断请求寄存器**

- 用于接收并记录各个中断源的
- 中断请求;

- **中断屏蔽寄存器**

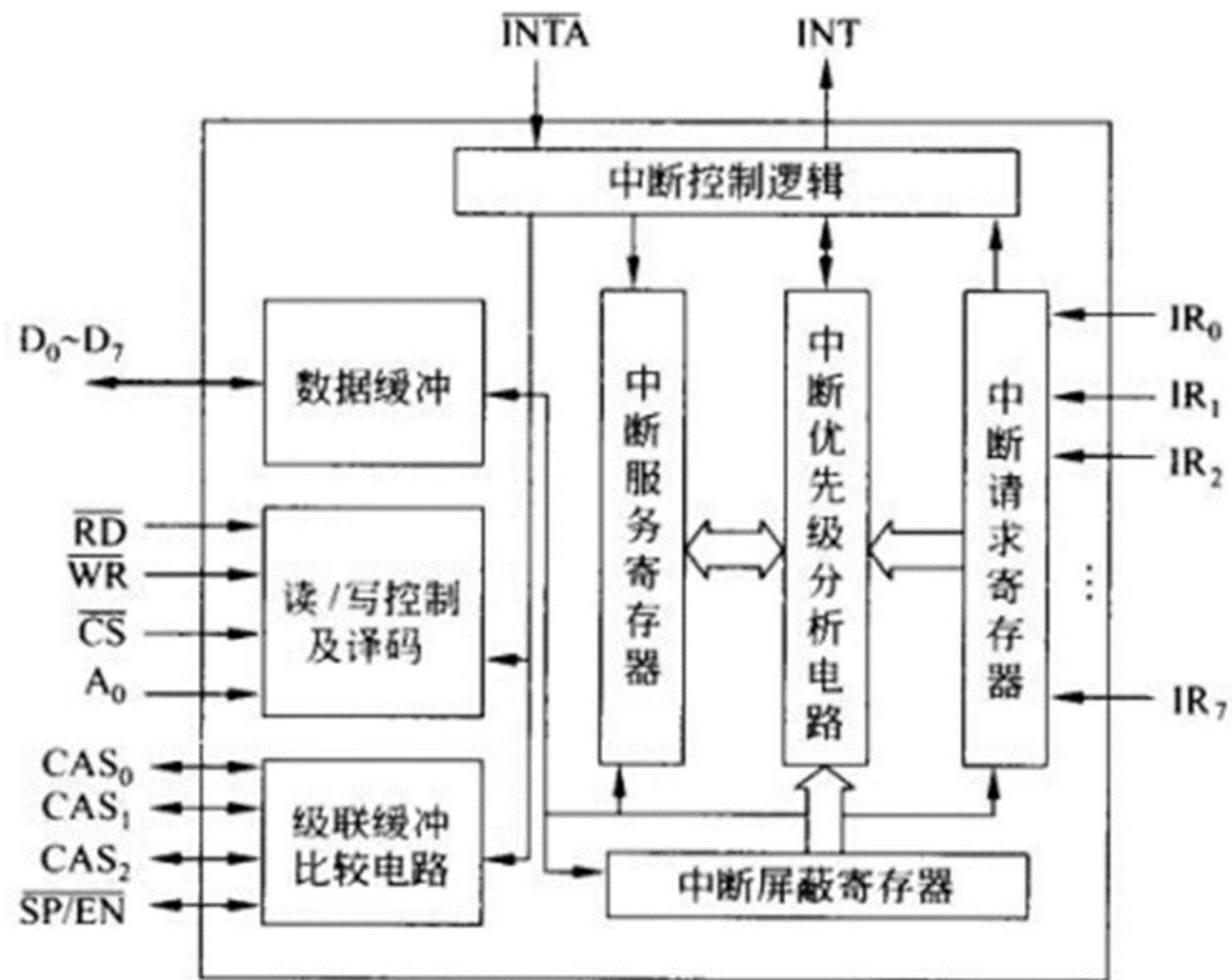
- 用于对各个中断源的中断请求
- 进行屏蔽管理,
- 被屏蔽的中断请求不会被处理;

- **中断服务寄存器**用于记录

- 目前正在处理以及被再次中断而
- 尚未处理完毕的中断请求;

- **中断优先级分析电路**用于

- 对未被屏蔽的中断请求进行
- 优先级比较分析。



中断管理器和CPU协作处理中断的过程

- 1) 各个中断源向中断管理器发中断请求;
- 2) 中断管理器记录大家的请求
- 3) 如果CPU目前没有处理任何中断, 中断管理器向CPU发中断请求; 如果CPU目前正在处理某个中断且新来的中断中有比正在处理的中断级别更高的中断, 中断管理器也向CPU发中断请求
- 4) 在允许中断 (包括中断正在处理的中断) 的情况下, CPU适时检测中断管理器发送的中断请求
- 5) CPU检测到中断管理器发送的中断请求后, 向中断管理器发送响应信号并开始中断响应过程: 暂停正在执行的程序或者中断服务程序, 保护相关的重要信息。

中断管理器和CPU协作处理中断的过程

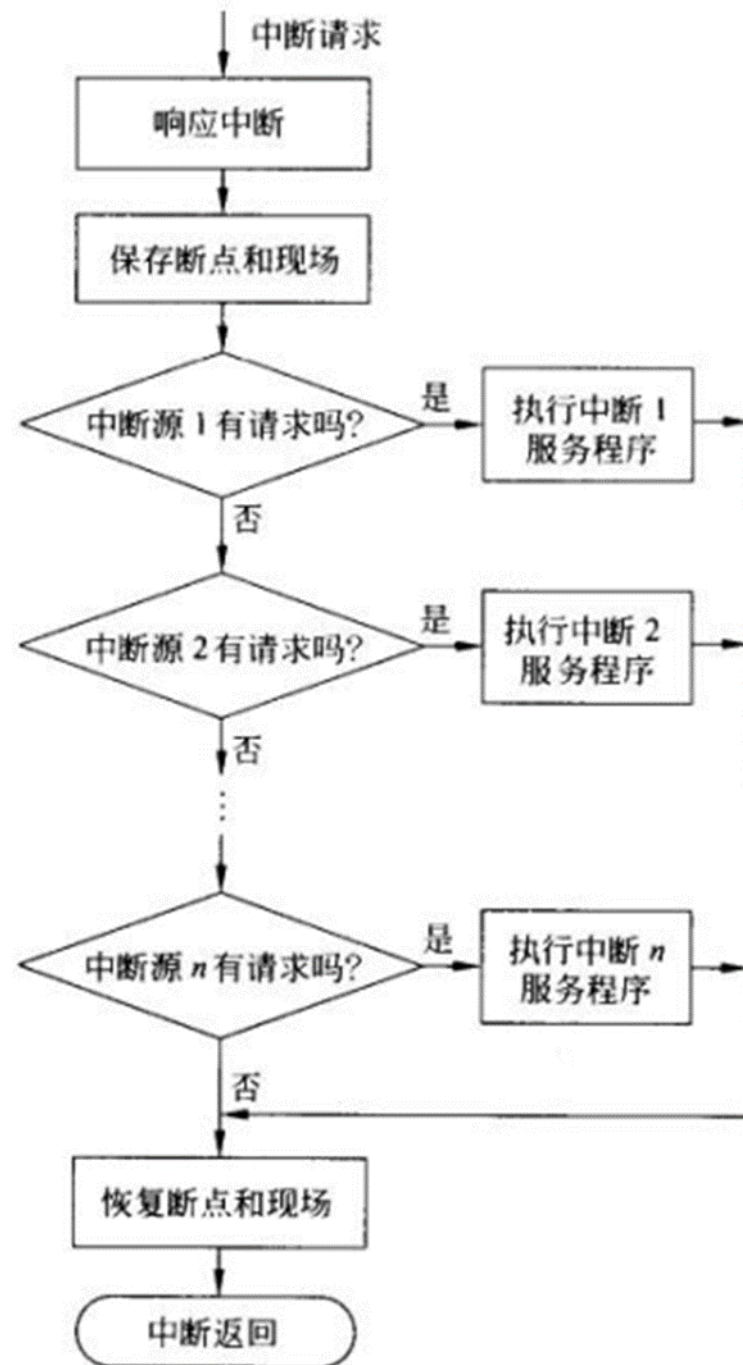
- 6) 中断管理器接收到CPU发送的中断响应信号后，向CPU提供级别最高或者新来的级别最高的中断请求的相关信息
- 7) CPU利用中断管理器提供的级别最高的中断请求的有关信息找到其中断服务程序的起始位置
- 8) 开始执行中断服务程序
- 9) 在中断服务程序的最后通过执行专门的中断返回指令回到被暂停的程序或中断服务程序

向量中断和查询中断

- 中断类型码：用来区分不同中断源及其中断服务程序的无符号数。
- 中断向量：中断服务程序在内存中的起始位置，也就是入口地址。
- 中断向量表：严格按中断类型码由小到大的顺序集中存放所有中断向量的地址表，通常位于内存某固定区域。
- **向量中断**：中断管理器在收到CPU的响应信号后，适时向CPU提供级别最高或者新来的级别最高的中断请求的中断类型码。CPU在响应期间收到此中断类型码后据此查询中断向量表，从中找到所需的中断向量，然后利用找到的中断向量转入相应的中断服务程序

查询中断

- 这种方式的特点是在CPU
- 检测到有中断请求产生时,
- 在中断响应期间通过执行查询程序
- 确定其中级别最高的中断源,
- 并转入响应的中断服务程序
- 查询中断的处理过程如图所示,
- 查询顺序决定了优先级



6.1.6 DMA方式

- 中断方式的缺点：
- 虽然不需要CPU查询发送/接收数据的条件，
- 但是最终的数据传送还是要靠CPU执行相应的程序
- （中断服务程序）来完成

DMA方式

- 在内存和外设之间增设一个专门部件，由它来代替CPU负责在内存和外设之间传送数据，这种数据传送方式就是**DMA方式**。
- DMA控制器：代替CPU负责在内存和外设之间传送数据的专门部件。
- 计算机系统软件（包括程序和数据）平时都存储在外存上。在程序执行过程中，需要用到时再将它们复制到内存。
- 指令及数据的局部性：程序在执行过程中，在一段时间内要执行的指令或要访问的数据往往集中在一起。

局部性带来的好处

- 需要用到的程序和数据没必要一次性全部复制到内存，只需要将一段时间需要用到的部分在其第一次需要用到时复制到内存即可。
- 这样做有诸多好处，比如：可以利用有限的内存空间执行大规模程序，处理大规模数据；可以在不同任务间共享内存空间；等等。

DMA的基本问题

- DMA请求：DMA控制器负责接收外设的DMA请求，并及时向CPU提出请求。
- DMA检测：CPU需要定期检测DMA控制器发出的请求
- DMA响应：CPU检测到DMA控制器发出的请求后，需要对DMA控制器做出应答并让出总线的控制权
- DMA传送：DMA控制器接管总线控制权，并利用总线传输数据
- DMA结束：数据传输结束后，DMA控制器归还总线给CPU，并通知CPU数据传送结束。

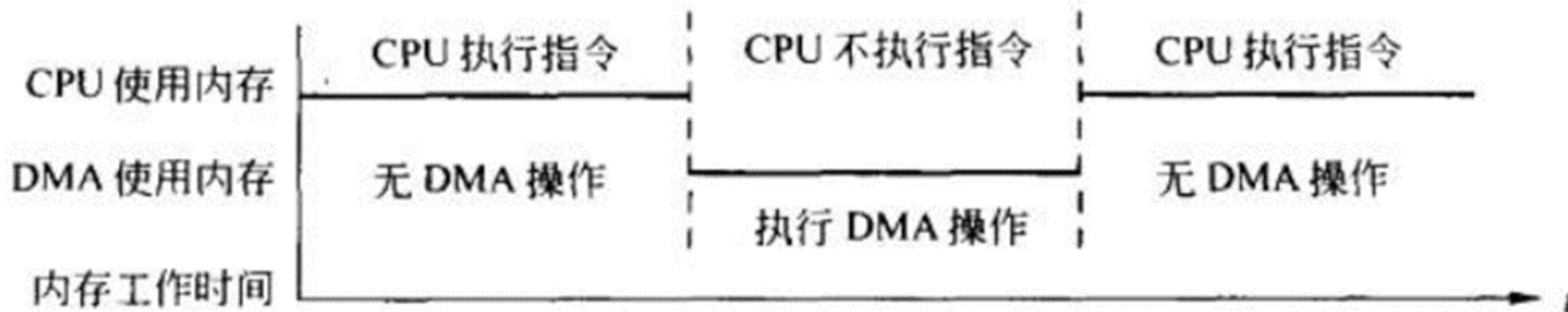
与中断方式的差别

- 系统处理DMA请求的过程与处理中断的过程也基本类似，主要差别体现在：
 - 1) 中断是通过CPU执行程序完成数据传输（是一种软件办法），而DMA是利用DMA控制器完成数据传输（是一种硬件办法）；
 - 2) 中断一般允许嵌套，而DMA一般不允许嵌套

DMA控制器与CPU共享总线的方式

- 通常DMA控制器和CPU共享系统总线,
- 在DMA控制器控制传输数据时,
- CPU必须放弃对系统总线的控制,
- 而由DMA控制器来控制系统总线.
- 不同的计算机系统会采用不同的方法来
- 解决CPU与DMA控制器共享总线的问题。
- CPU与DMA控制器共享总线大致有3种方式

CPU暂停方式



采用这种工作方式的I/O设备需要在其接口控制器中设置一定容量的存储器作为**数据缓冲存储器**使用，I/O设备与数据缓冲存储器交换数据，主存也只与数据缓冲存储器交换数据，由于数据缓冲存储器的存取速度较快，这样可以减少由于执行DMA数据传送而占用系统总线的时间，从而减少了CPU暂停的时间。

数据缓冲技术

计算机系统中不同的部件（如主机和外设）之间、不同的进程之间、同一进程的不同线程之间等等，它们之间经常需要交换数据。

实现双方数据交换的最简单高效的方式就是设立数据缓冲区，一方适时往里放数据，另一方适时从中取数据。缓冲区是双方交换数据的中转站，不仅可以暂时存放数据，还可以在在一定程度上缓解双方速度不匹配带来的问题。

缓冲区的容量可以综合考虑双方速度上的差异以及数据交换方式、应用场景等灵活确定。

周期挪用方式



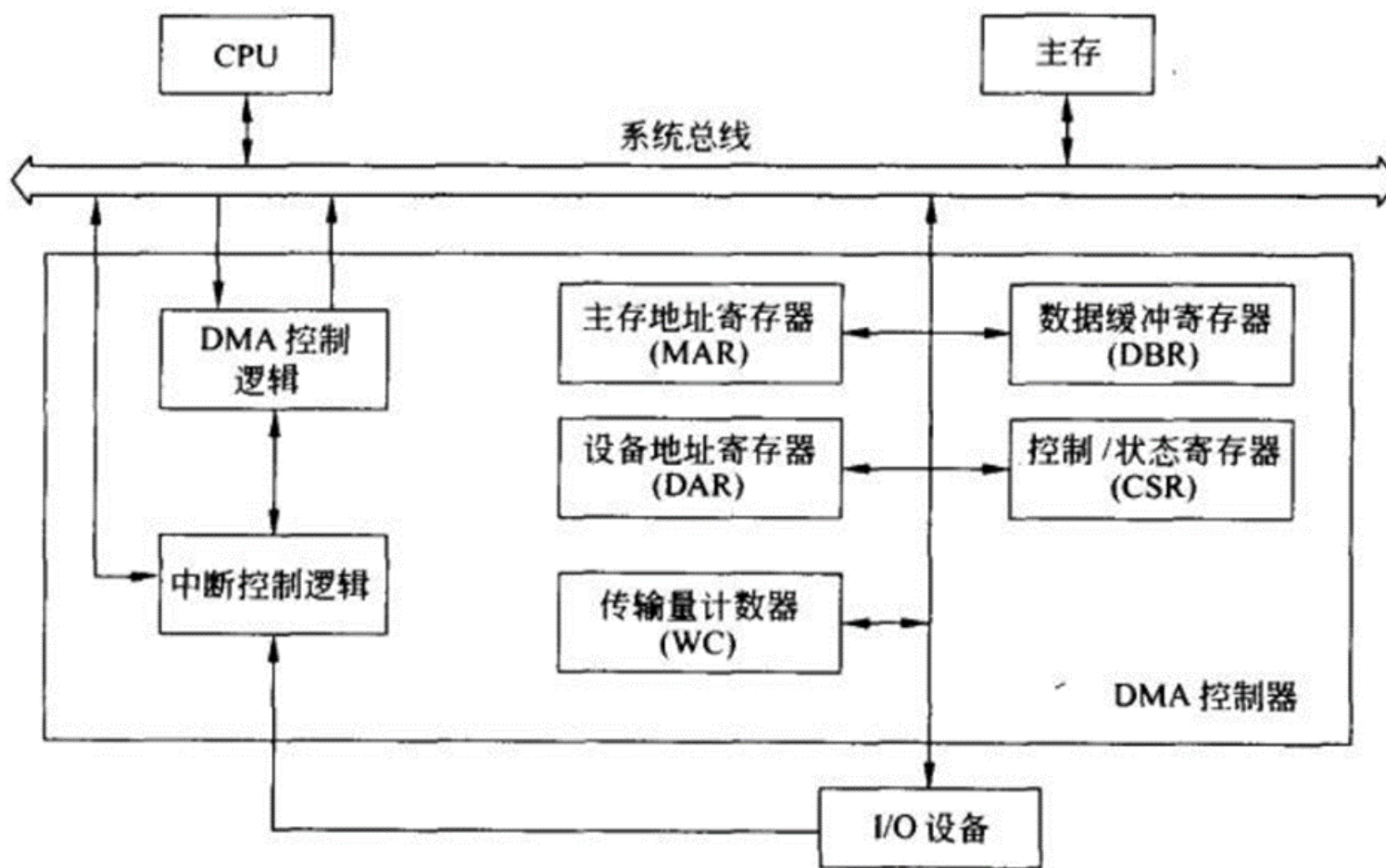
当I/O设备无DMA传送请求时，CPU正常访问主存。当I/O设备需要使用总线传送数据时，产生DMA请求。DMA控制器把总线请求发给CPU，此时若CPU本身无使用总线的要求，CPU就可把总线交给DMA控制器，由DMA控制器控制I/O设备使用总线，这样的情形当然最为理想；如果此时CPU也要使用总线，则CPU自身进入一个空闲总线周期状态，即CPU让出一个总线周期给DMA控制器（也称DMA控制器挪用一個总线周期）。DMA控制器利用此总线周期控制传送一个数据字后，再把总线交还给CPU，以便CPU可以执行总线操作。可见当I/O设备与CPU同时都要访问主存而出现访问主存冲突时，I/O设备访问的优先权高于CPU访问的优先权

交替访问方式

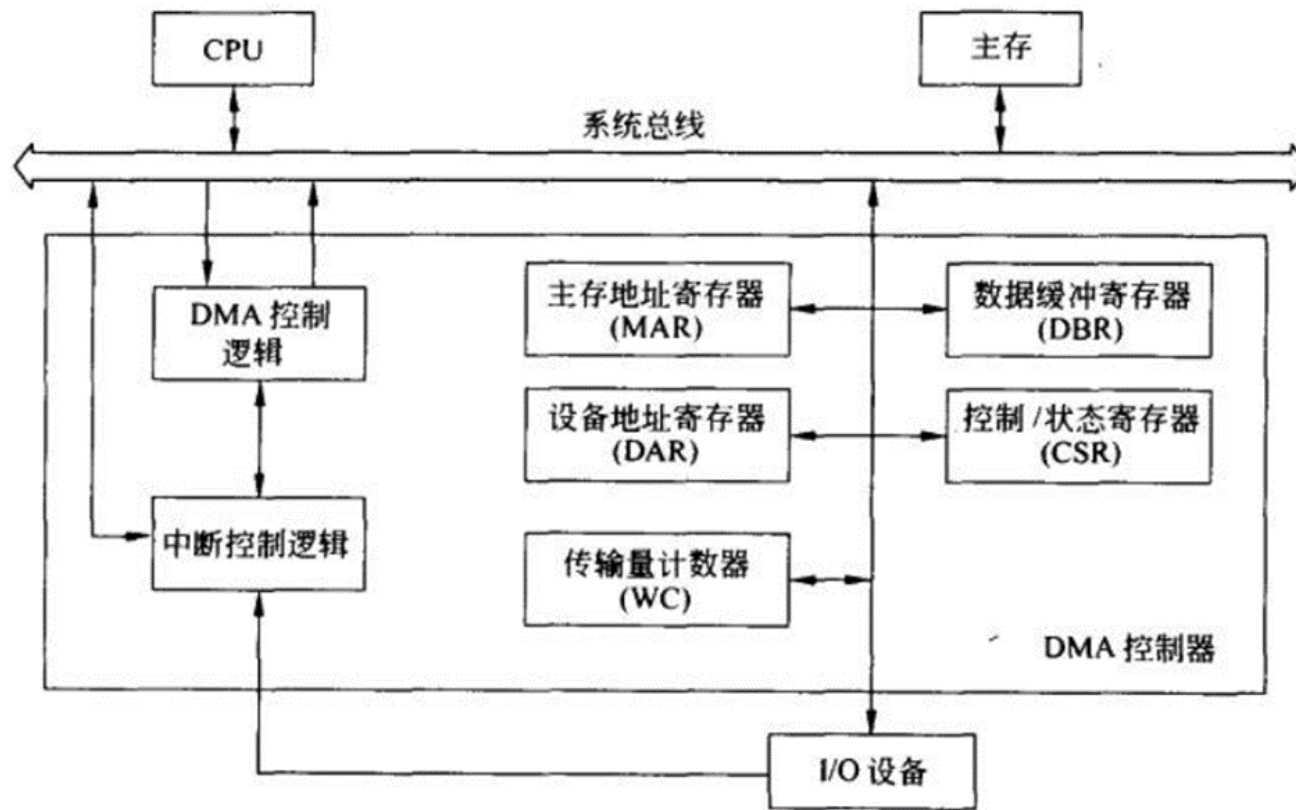


使用这种方式的前提是 CPU 的工作速度相对较慢，而内存的工作速度较快，或者人为拉长 CPU 执行指令的时间。如主存的存取周期为 Δt ，而 CPU 每隔 $2\Delta t$ 才产生一次访存请求，那么在 $2\Delta t$ 内，一个 Δt 供 CPU 访问主存，另一个 Δt 供 DMA 访问主存

DMA控制器的基本结构

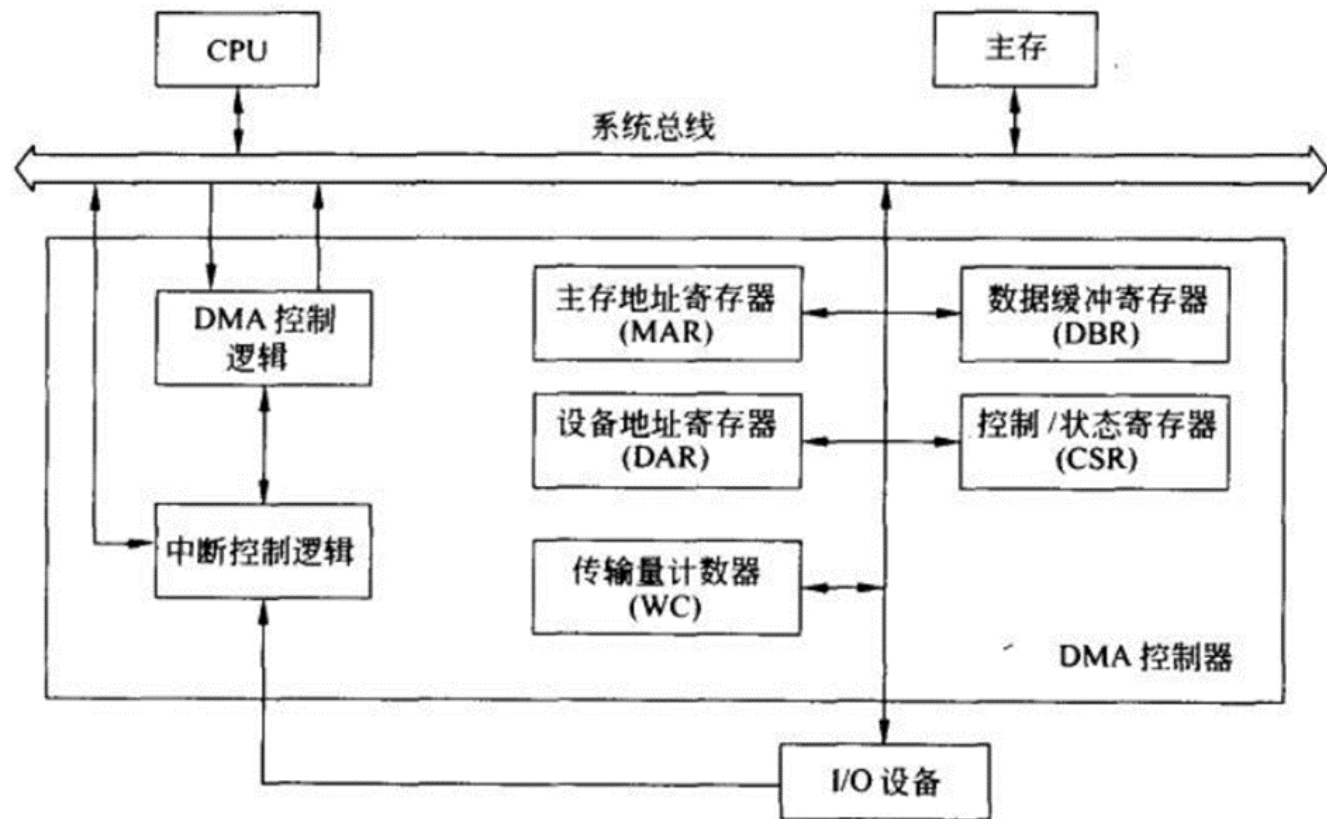


寄存器组



- 寄存器组用来提供DMA过程中需要用到的各种参数，其中：主存地址寄存器为访问内存提供地址，每访问一个单位的数据会自动修改；设备地址寄存器为访问外围设备提供地址；传输量计数器用于控制需要传输的数据总量，传输一个单位的数据会自动修改，通常采用倒数方式；数据缓冲寄存器在需要时用来暂时存放数据。

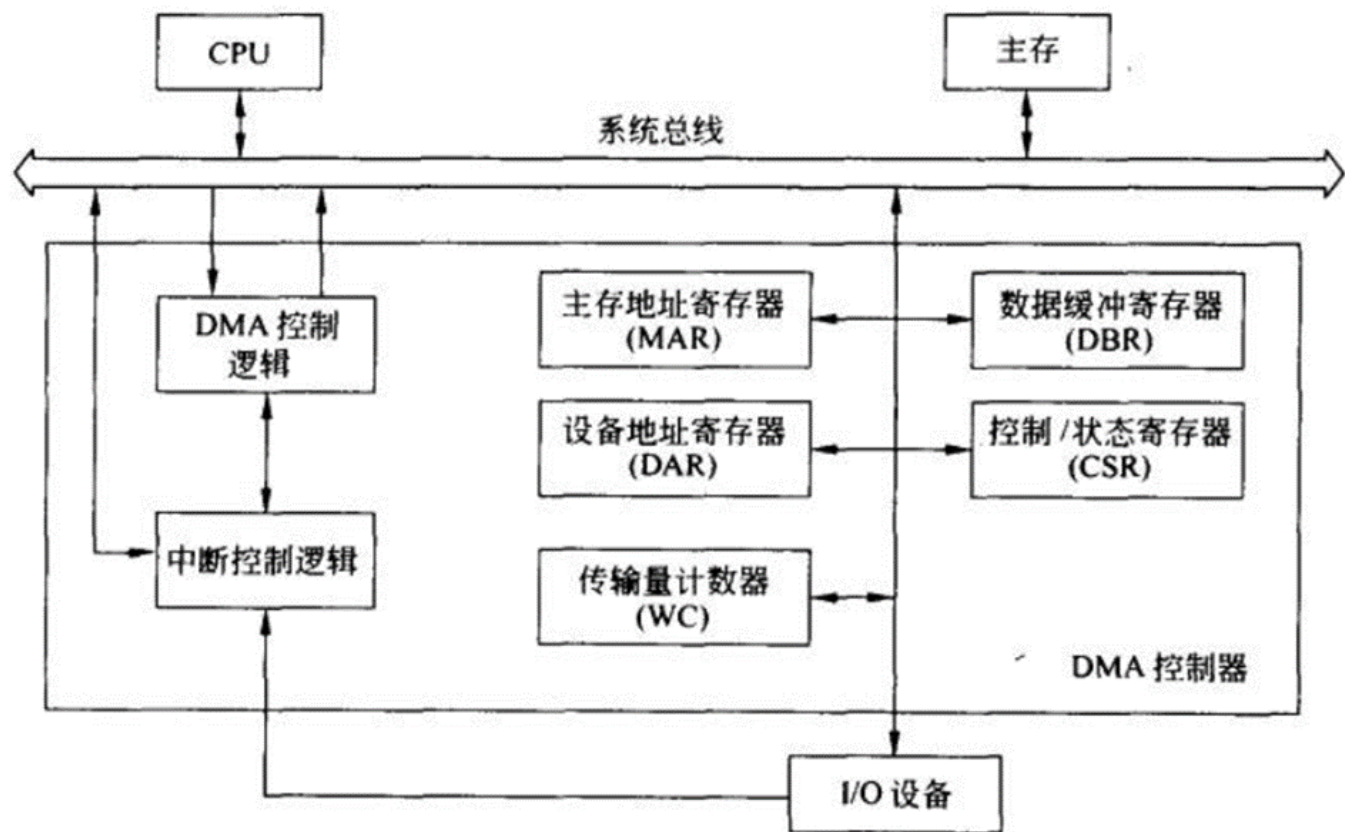
DMA控制逻辑



- DMA控制逻辑负责完成DMA的预处理（初始化各类寄存器）、接收设备控制器送来的DMA请求信号、向设备控制器回答DMA允许（应答）信号、向系统申请总线以及控制总线实现DMA传输控制等工作

中断控制逻辑

DMA中断控制逻辑负责在DMA操作完成后向CPU发出中断请求，申请CPU对DMA操作进行后处理或进行下一次DMA传送的预处理。

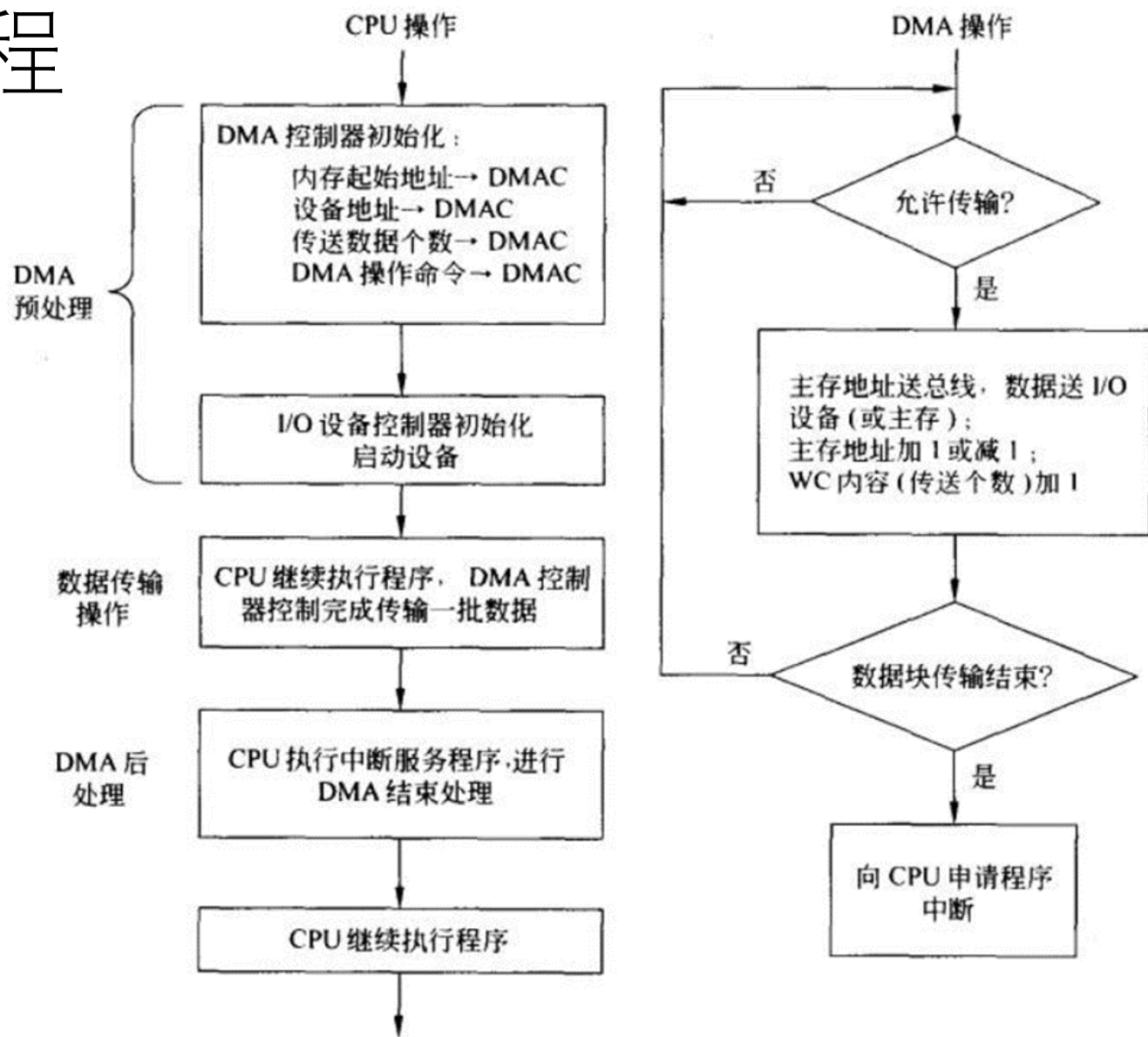


DMA数据传输过程

预处理阶段：对DMA控制器进行一些初始设置，比如：内存首地址、数据长度、数据传送方式等。

数据传送阶段：DMA控制器接管总线后进行的真正数据传输。

后处理阶段：数据传输结束后，DMA控制器通过中断机制通知CPU，然后CPU通过中断做该做的事。



6.2 总线

- 计算机系统中不同部件之间需要通过**信号线**相互连接起来，在彼此之间传送各种信息。
- 部件之间需要传递的信息主要是3种：**地址信息、数据信息和控制信息**。

6.2.1 总线

- 各部件之间传送信息的公共通路称为总线。实质上总线是一个共享的传输媒介。当多个设备连接到总线上，其中任何一个通过总线传输的信号都能被连接到总线上的其它设备所接收。如果两个设备同时向总线发送信号，总线上的信号就会叠加而发生混乱，因此要对各功能部件使用总线的方式进行一定的限制，保证任何时候只有一个设备向总线发送信号。

总线的分类

- 按总线所承担的任务可分为**内部总线**和**外部总线**。
- **内部总线**用于实现主机系统内部各功能模块（部件、板卡）之间的互联
- **外部总线**用于实现主机系统与外部设备或其他主机系统之间的互联。
- 其中，专门用于主机系统与外设之间互连的总线称为**设备总线**。

总线分类

- 按总线所处的物理位置可分为**片内总线**、**板内总线**、**板间总线**和**外部总线**。
- **片内总线**实现芯片内部功能部件之间的连接，例如微处理器内部使用的总线，
- **板内总线**实现该电路板上各个集成电路芯片之间的互联，
- **板间总线**则用于把各个功能模块（如CPU主存储器、I/O接口适配器等）连接到一起，构成主机系统，所以也称它为系统总线。

总线分类

- 按总线所传送的信息类型可分为**地址总线**、**数据总线**和**控制总线**等。
- 按总线一次传送数据的位数又分为**串行总线**和**并行总线**。
- 按总线操作的定时方式，又有**同步总线**和**异步总线**之分

总线标准

- 主要从以下几个方面来描述总线的功能和特性
- (1) **逻辑规范**: 引脚信号的功能描述, 包括信号的含义、信号的传送方向 (发送接收或双向)、有效信号所采用的电平极性 (高电平/低电平, 正脉冲/负脉冲) 及是否具有三态能力等。
- (2) **时序规范**: 描述各信号有效/无效的发生时间以及不同信号之间相互配合的时间关系。例如当地址信号有效后, 至少需要多长时间的延迟才能使读/写信号有效。

总线标准

- (3) **电器规范**：总线上各个信号所采用的电平标准和负载能力。负载能力定义了总线理论上最多可以连接模块的数量。
- (4) **机械规范**：它定义了总线包括插槽/插头或插板的结构、形状、大小方面的物理尺寸、接插件机械强度；总线信号的布局、引脚信号的长度、宽度以及间距等。
- (5) **通信协议**：定义数据通过总线传输时采用的连接方法数据格式、发送速度等方面的规定。对串行总线而言会有这方面的规范。通信协议通常还要分为若干层次。

总线的性能

- 总线的性能水平主要由以下几个因素决定：
- (1) **总线带宽**：总线带宽表示在单位时间内，总线所能传输的最大数据量，一般用兆字节/秒 (MB/S) 来表示。
- (2) **总线宽度**：笼统地说，一个总线所设置的通信线路（或线缆）的数目称为该总线的宽度。具体来说，在一个总线内设置的用于传送数据的信号线的数目，称为数据总线宽度。同样也存在一个地址总线的宽度。

总线性能

- **数据总线的宽度**决定了一次可以同时传送的二进制信息的位数。
在总线工作频率一定的条件下，数据总线单位时间内的数据传输量与数据总线的宽度成正比关系，因此数据总线的宽度是决定计算机系统性能的一个关键特性。
- **地址总线的宽度**决定计算机系统的寻址能力。

总线性能

- (3) **总线的时钟频率**：对于同步总线来说，由于采用统一的时钟脉冲作为定时基准，因此总线的时钟频率越高，总线上的操作就越快。显然，在数据总线宽度相同的情况下，较高的总线时钟频率会带来较大的数据吞吐量。
- (4) **总线的负载能力**：限定在总线上可以连接模块的最大数目。一般来说，都希望总线具有较高的带宽和较强的负载能力。

6.2.2 总线的控制与裁决

- 由于总线属于共享资源，为确保各个功能模块正确使用总线，需要有一个专门的部件来负责对总线的使用权进行管理和分配，这个专门部件就是**总线控制器**，也叫**总线仲裁器**
- **主设备**：经过裁决后，获得总线使用权的设备成为主设备
- **从设备**：与主设备交换数据的设备为从设备。
- 由此一来，通过总线相互连接的各个功能模块在需要利用总线传输数据时，需要经历**四个阶段**：申请、仲裁、传输、结束

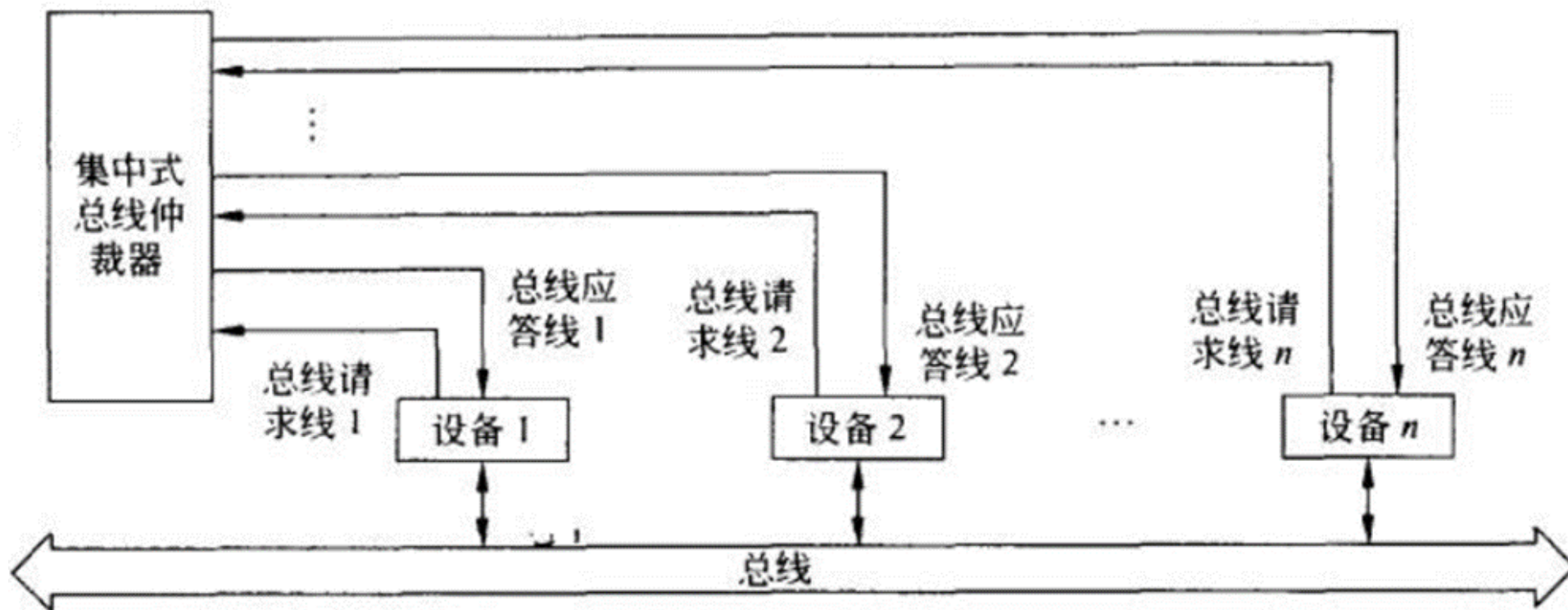
总线事务：总线传输过程

- **申请阶段**：当某个模块需要利用总线和别的模块传输数据时，首先向总线控制器提出请求。
- **仲裁阶段**：总线控制器接收到各个模块的请求后，根据一定的优先级规则进行裁决，确定谁将获得总线的使用权并且对获得使用权的模块作出回应。
- **传输阶段**：获得使用权的模块接管总线并利用总线传输自己的数据。
- **结束阶段**：获得总线使用权的模块利用总线传输完数据后，释放总线。

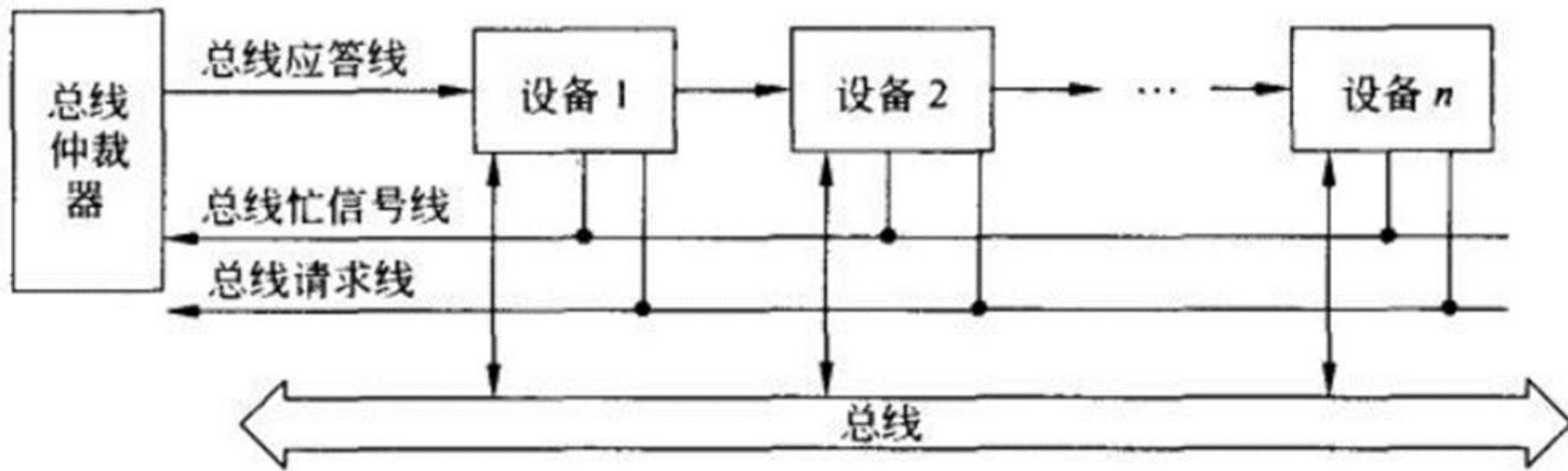
总线仲裁方式

- 总线仲裁的方式可以采用**集中式**，也可以采用**分布式**。
- **集中式**仲裁由单一总线仲裁器负责管理各个模块的总线请求并作出裁决，又可以细分为**并行裁决**和**串行裁决**

集中式并行仲裁



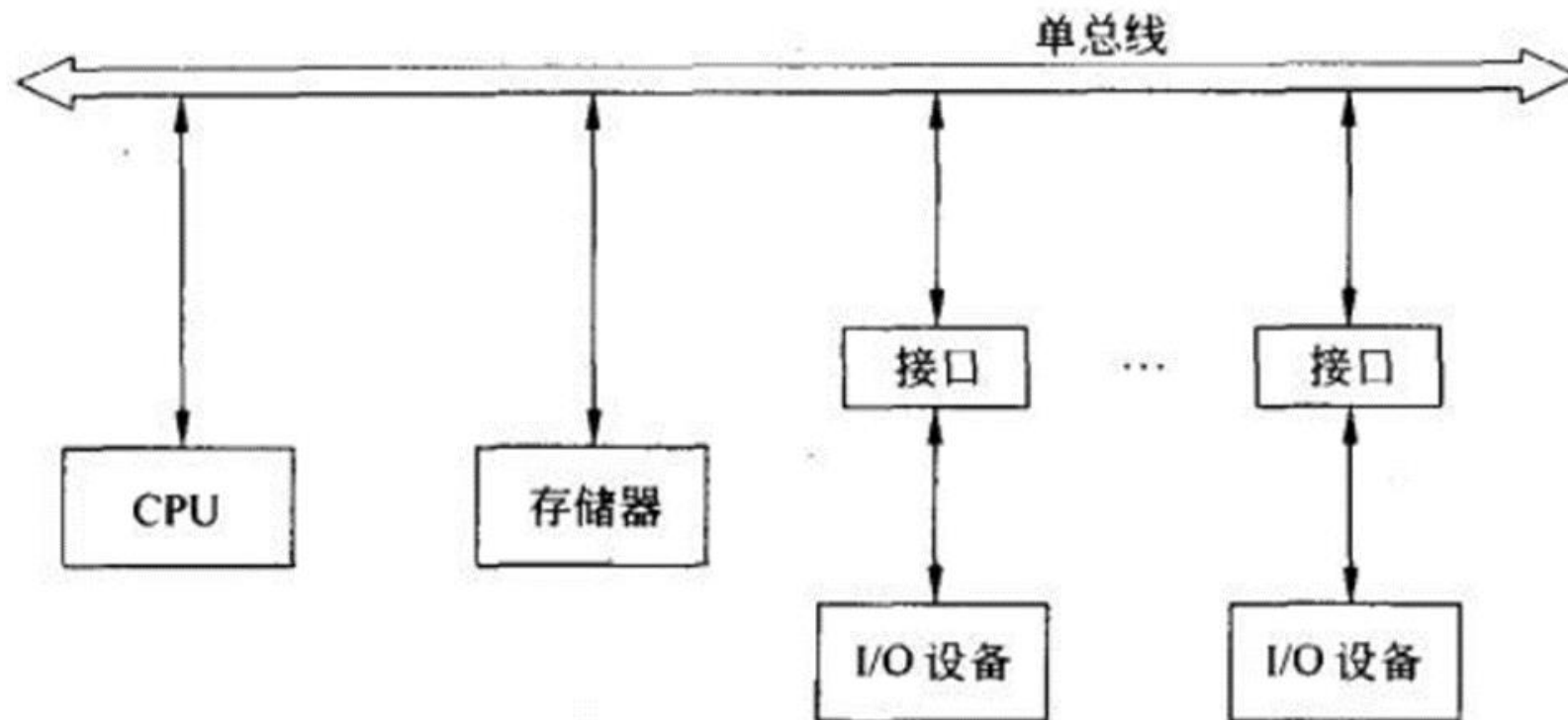
集中式串行仲裁



6.2.3 计算机系统的总线结构

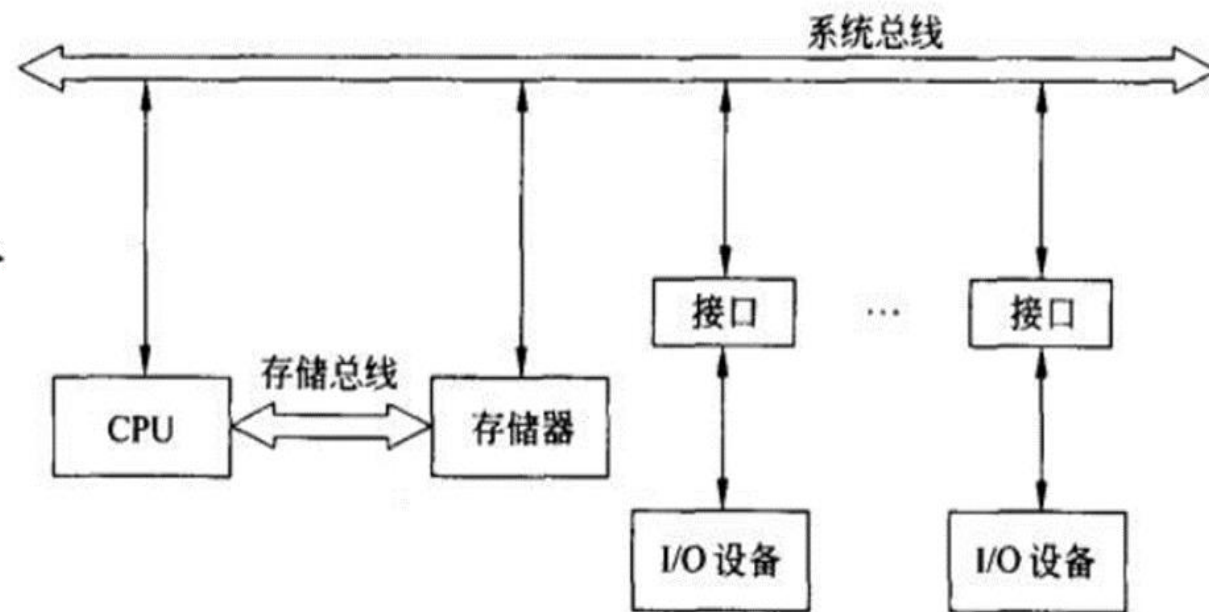
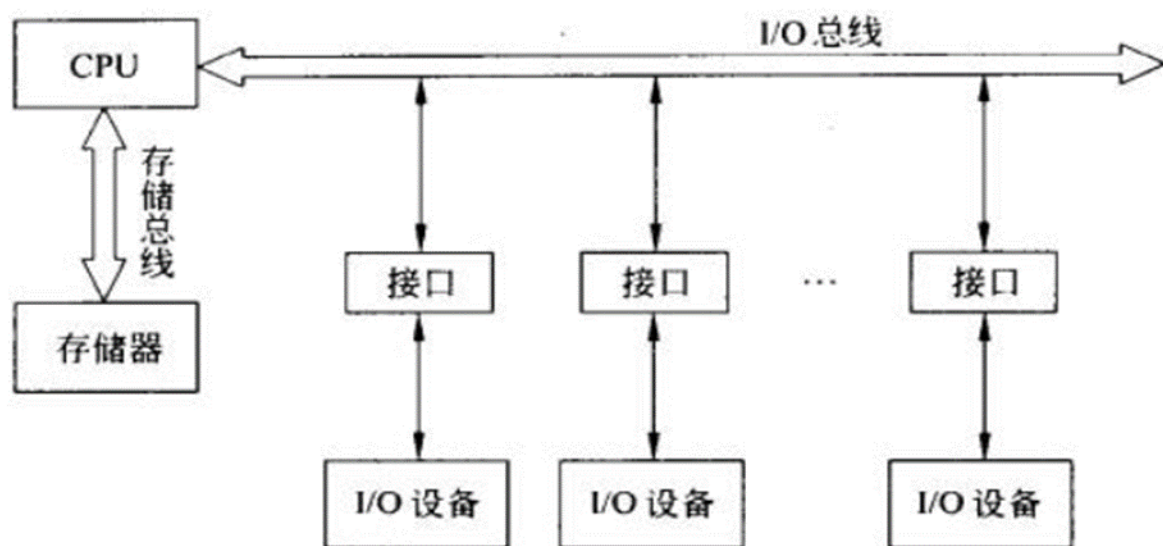
- 计算机系统总线组织方法很多，
- 按照总线组织结构不同，可以在大体上分成
- **单总线结构**的计算机系统
- **双总线结构**的计算机系统
- **多总线结构**的计算机系统。

单总线结构



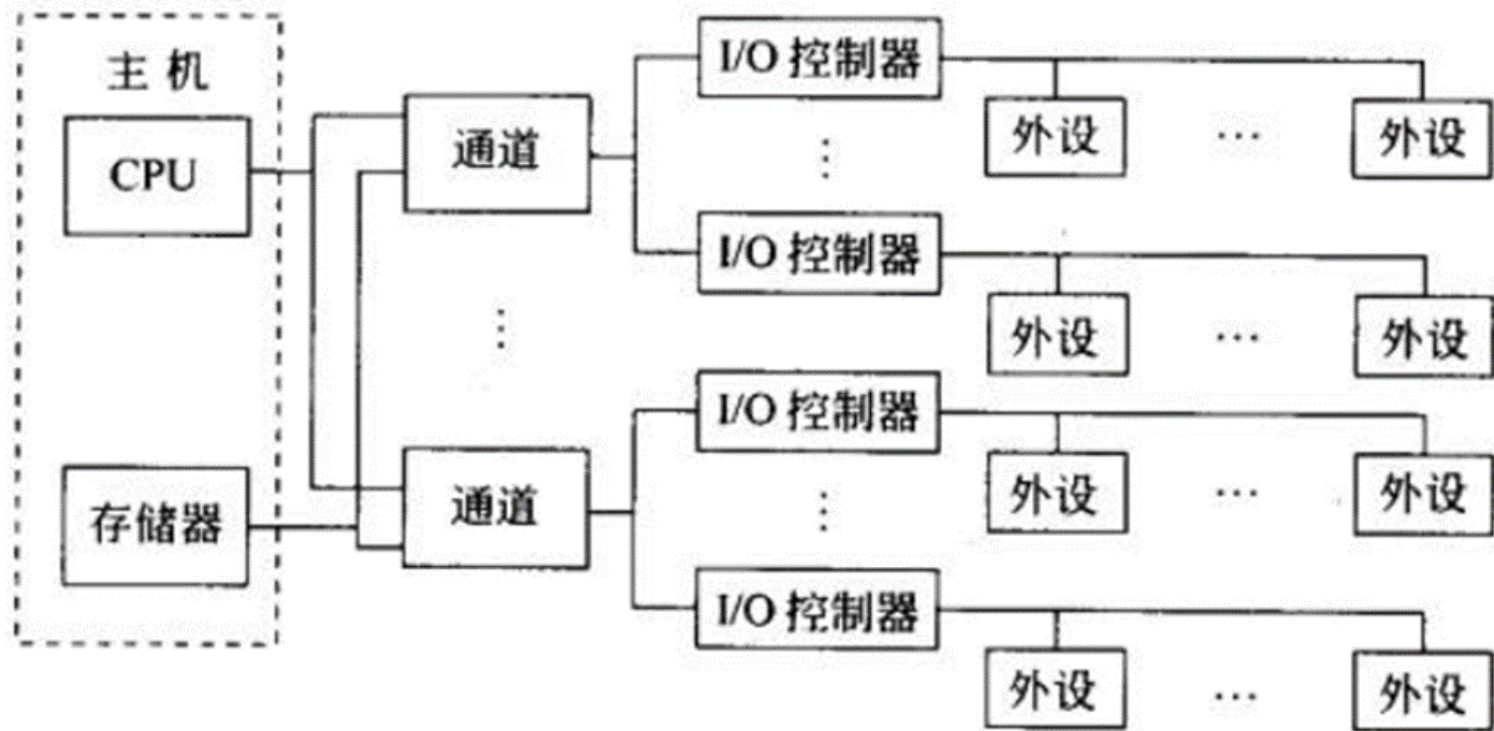
采用称为系统总线的一组总线来连接整个计算机系统各个功能部件，例如处理机模块、主存储器模块、I/O设备控制器模块等，计算机系统的所有外部设备通过设备控制器也挂在这条总线上。在这种结构的计算机系统内部，各个功能部件之间的所有信息传送都通过系统总线来实现。

双总线结构



- 增加了一条内存总线，CPU访问主存单元时通过内存总线来实现，原有的系统总线则用来实现CPU与外设以及内存与外设之间的数据通信，

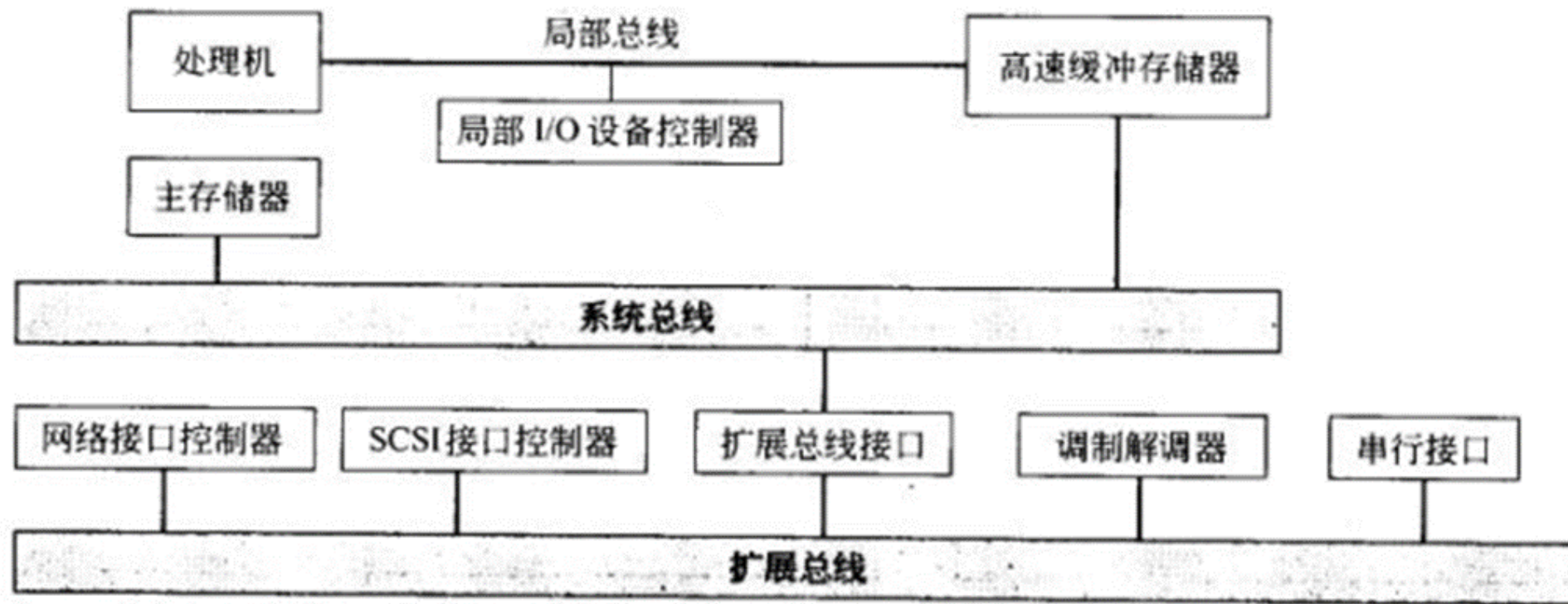
多总线结构



多总线结构的计算机系统是在双总线结构基础上增加I/O总线实现的一种计算机系统结构，目的是进一步提高计算机系统的工作效率。

这种总线结构是在计算机系统的各部件之间采用多条各自独立的总线来构成分层次的信息通路

多层次总线结构



根据系统功能模块性能上的要求，设置不同层次、不同种类的总线，不会完全拘泥于上述三种总线结构

6.2.4 总线定时

- 功能模块获得总线使用权后利用总线传输数据是由一序列的操作构成的，这些操作在时间上是有先后的。例如处理机要从内存中读数据时，需要先通过地址线向内存发送存储单元的地址，然后通过控制线向内存发出读控制，最后将内存发送到数据线上的数据读入。
- **总线定时**指的就是协调总线上发生的事件的时间顺序和时间长度，换言之，就是协调各种信号在总线上出现和撤销的时机
- 总线定时的方法分为**同步定时**和**异步定时**，由此总线又可分为**同步总线**和**异步总线**。

同步定时： 同步总线

- 总线上所有事件的发生都由同一个时钟脉冲序列来定时。在这种定时方式下，总线应包含一条时钟信号线，该时钟信号线负责传送一个固定频率的方波信号。一个时钟周期就是一个最基本的总线操作的时间单位，一个总线周期则由一个或多个时钟周期构成。
- 总线周期： 利用总线完成一次数据交换所花费的时间
- 连接到总线上的所有设备都通过时钟信号线获取用于事件同步的时钟脉冲信号，所有的总线事件都应在一个时钟周期的开始时启动动作。

异步定时： 异步总线

- 在采用异步定时的总线中，总线上一个事件发生与否，依赖于前一个事件的执行情况。

6.2.5 常用总线

- **(1) PCI局部总线：内部总线**
- 20世纪90年代后，随着图形处理技术和多媒体技术的广泛应用，特别是在以Windows为代表的图形用户界面应用得到普及之后，要求计算机系统具有对图形/图像数据的高速处理以及对显示数据的快速传输能力，这对总线技术提出了前所未有的挑战。原有的各类总线已远远不能满足应用的需求，总线成为整个计算机系统性能提升的主要瓶颈。
- 在此背景下，英特尔公司于1991年首先提出了PCI（外围组件互连）总线的概念，并联合Compaq、AST、HP等100多家公司成立了PCI集团兴趣小组，负责起草、制定PCI局部总线标准。

6.2.5 常用总线

- **(2) PCI Express总线：内部总线**
- PCI Express总线并不是PCI技术的延续，即PCI Express总线不是原有PCI总线的升级，而是一个全新的第三代总线。

设备总线

- (1) **小型计算机系统接口 (SCSI接口)**
- 小型计算机系统接口 (Small Computer System Interface, SCSI) 是用于小型、微型计算机和外围设备连接的一种接口标准，它可以支持包括磁盘驱动器、磁带机、光盘驱动器以及扫描仪在内的多种外部设备。虽然称它为接口，但SCSI接口实际上是一种外部并行总线
- (2) **ATA接口**
- ATA接口 (AT附件) 是微型计算机主板与硬盘等外部存储器之间的一种接口或总线。ATA接口的前身是IDE接口，即集成设备电子部件。

设备总线

- (3) **USB接口**
- USB是**通用串行总线**的简称，是由英特尔、康柏、微软、NEC、北方电信等多家世界著名的计算机和通信公司联合开发的一种新型串行接口总线标准。

6.3 接口和设备

- I/O设备需要和主机进行有效连接，才能实现数据的输入、输出。由于各方面的原因，I/O设备通常又不能直接和主机相连，解决办法就是在I/O设备和主机双方之间增设一个中间电路，双方通过增设的中间电路交换数据，这个中间电路（中间装置）就是**I/O接口**。

6.3.1 I/O接口的功能

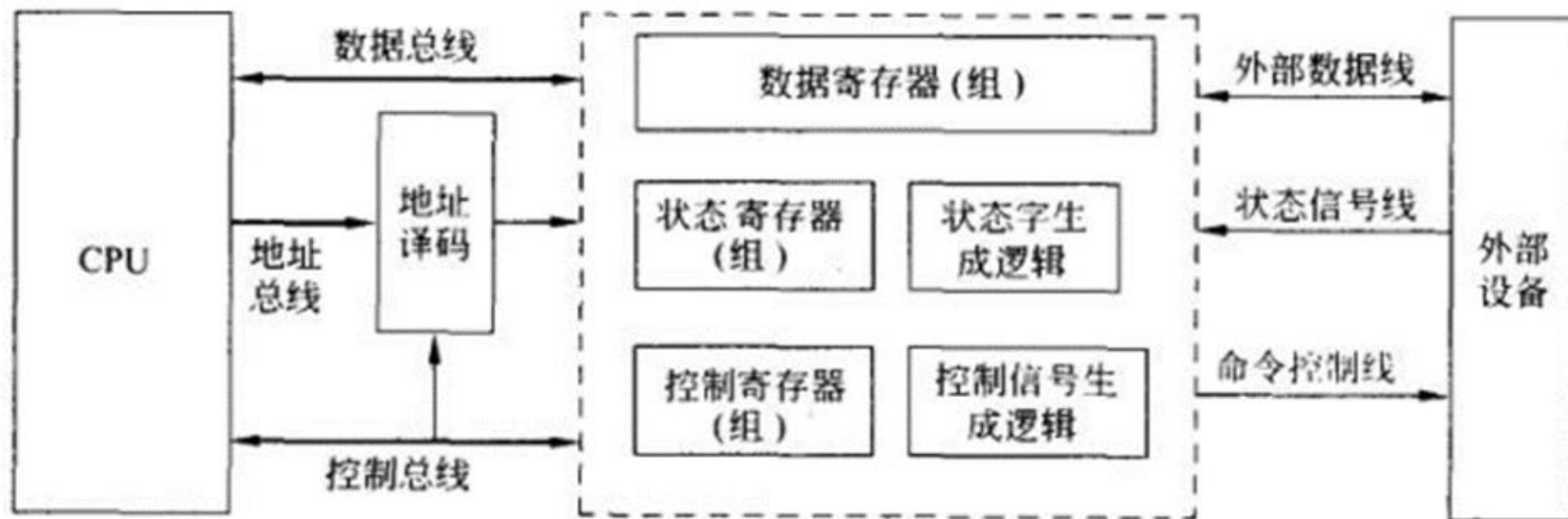
- I/O接口存在的原因是多方面的，这和I/O设备的多样化有关。不同的I/O设备在输入、输出的数据格式、速度以及输入、输出数据的具体方式等方面存在一定的差异。
- 此外，为了实现更灵活的输入、输出，需要对I/O设备进行相应的管理与控制，可以通过I/O接口来管理、控制I/O设备，而且一个I/O接口还可以同时管理多个I/O设备。

6.3.1 I/O接口的基本功能

- 数据格式转换
- 信号电平转换
- 并行/串行数字转换
- 数字/模拟信号转换
- 端口寻址
- 数据缓冲
- 各种管理、控制功能等

6.3.2 I/O接口的基本结构

- I/O接口的典型结构如下图所示：



接口内部的寄存器

- 数据寄存器：供双方交换数据使用，CPU可以读或写。输入数据时，I/O接口接收输入设备送过来的数据，存入数据寄存器后，供CPU读取；输出数据时，CPU写入数据，供I/O接口发送给输出设备。
- 状态寄存器：记录I/O接口及其控制的I/O设备的工作状态，供CPU读取。CPU通过读状态寄存器可以了解I/O接口及其控制的I/O设备的工作状态，从而决定接下来该做什么。
- 控制寄存器：接收CPU送过来的控制命令，按命令要求控制I/O接口及其控制的I/O设备的工作

6.3.3 I/O接口的分类

- 并行接口和串行接口
- 同步接口和异步接口
- 数字接口和模拟接口

6.3.4 常用外部设备（自学）

- 输入设备
- 输出设备
- 外存储器
- 终端设备
- 其它设备

输入设备

- 目前广泛使用的输入设备主要有：
- 用于字符输入的设备，键盘、联机手写识别器等。
- 用于图形输入的设备，数字化仪、鼠标器、跟踪球、操纵杆、图形输入板等。
- 用于图像输入的设备，摄像机、扫描仪等。
- 其它类型的设备，如数模转换、声音输入等。
- 特殊的输入设备，磁盘、磁带及光盘等。

输出设备

- 常用的输出设备主要有：
- 以输出字符为主的设备，行式打印机、点阵式打印机、喷墨和激光打印机、显示器。
- 以输出图形为主的设备，绘图仪、显示器、喷墨及激光打印机。
- 以输出图像为主的设备，显示器、喷墨及激光打印机。
- 其它类型的设备，如声音输出设备等。
- 特殊的输出设备，磁盘、磁带等。

外存储器

- 磁带
- 磁盘
- 光盘
- 各种移动存储设备
- 各种智能终端设备

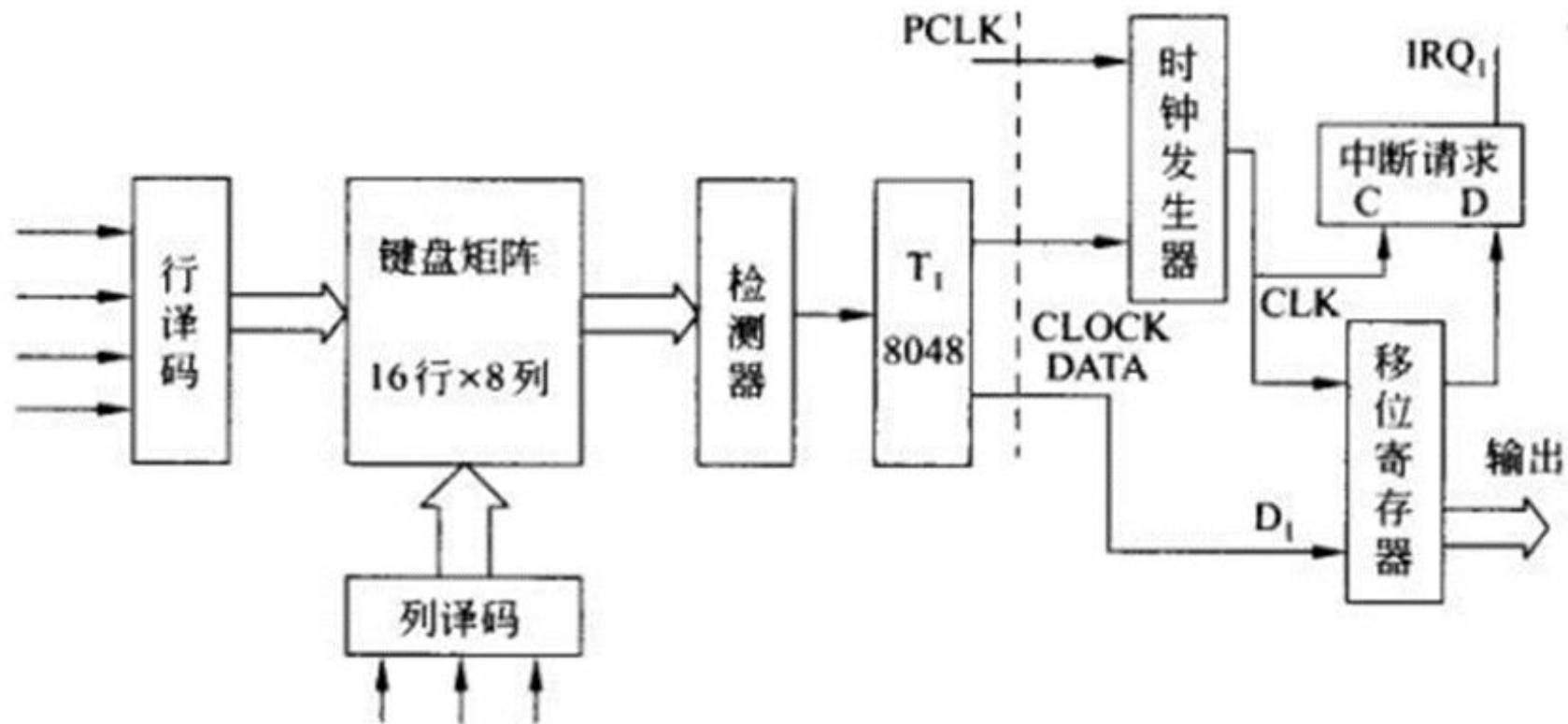
终端设备

- 与计算机网络的用户端相连接的设备，称之为终端设备。另外在大型计算机系统中，通过通信线路连接到主机的输入输出装置，也是一种终端设备。用户通常通过终端设备在一定距离之外操作计算机，通过终端输入信息或获得结果。利用终端设备，可使多个用户同时共享计算机系统资源。
- 具有一定的数据处理能力的终端称为智能终端，而只负责输入输出的终端称为哑终端。与主机距离较近的终端称为本地终端，例如在一个计算中心机房的终端；与主机距离较远的终端称为远程终端。远程终端往往要通过公共通信线路，利用调制解调器等通信设备与主机交换信息。

其它

- 在某些特定应用领域中，要用到一些特殊的I/O设备，如在工业控制应用中的数据采集设备、仪表、传感器、A/D和D/A转换器等。
- 还有一类所谓的脱机设备，即数据制备设备，如软磁盘数据站。它是一种数据录入装置，为了不让数据录入占用大、巨型主机的宝贵运行时间，大批数据录入往往采取脱机录入方式，即先在专门的录入装置上人工按键录入，结果存入磁盘或磁带之中，然后将利用磁盘或磁带联机输入主机。

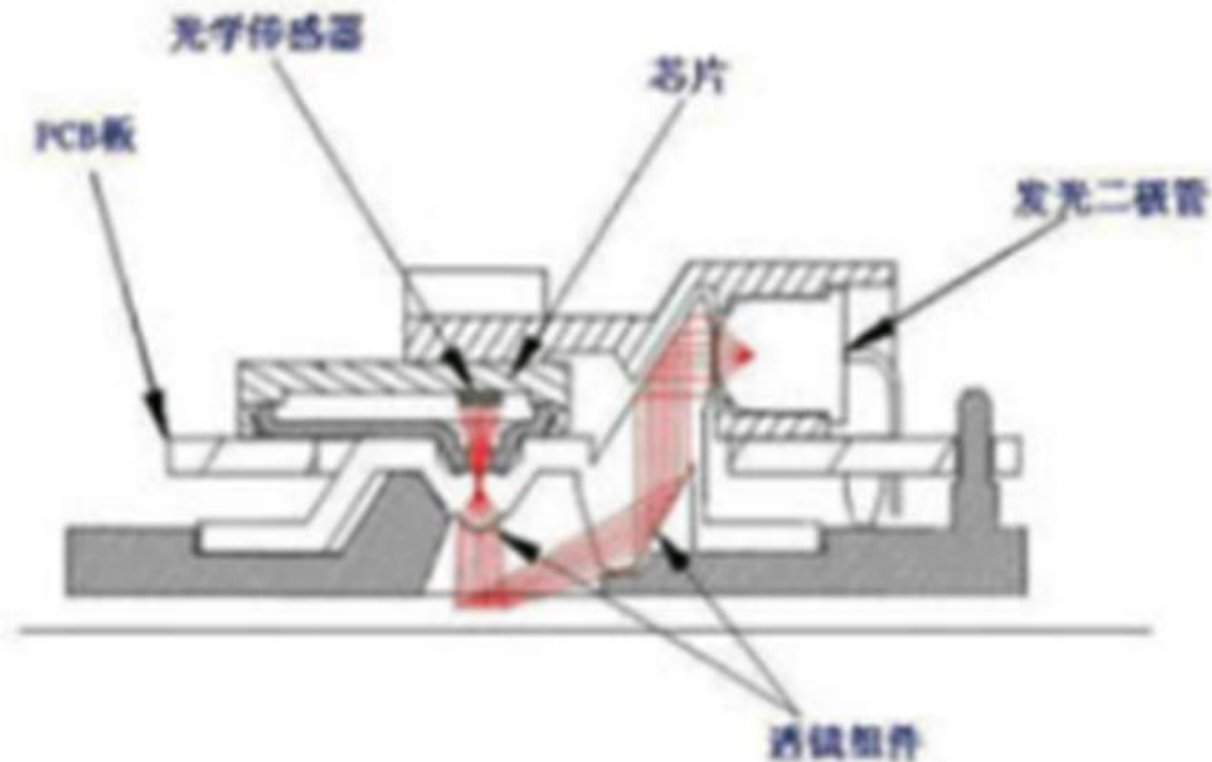
键盘



以**行列扫描法**获得按键**扫描码**。键盘通过电缆与主机板上的键盘接口相连，以串行方式将扫描码送往接口，由移位寄存器组装，然后向CPU请求中断。CPU以并行方式从接口中读取按键扫描码。虚线左边是键盘逻辑，右边是位于主机板上的接口逻辑。

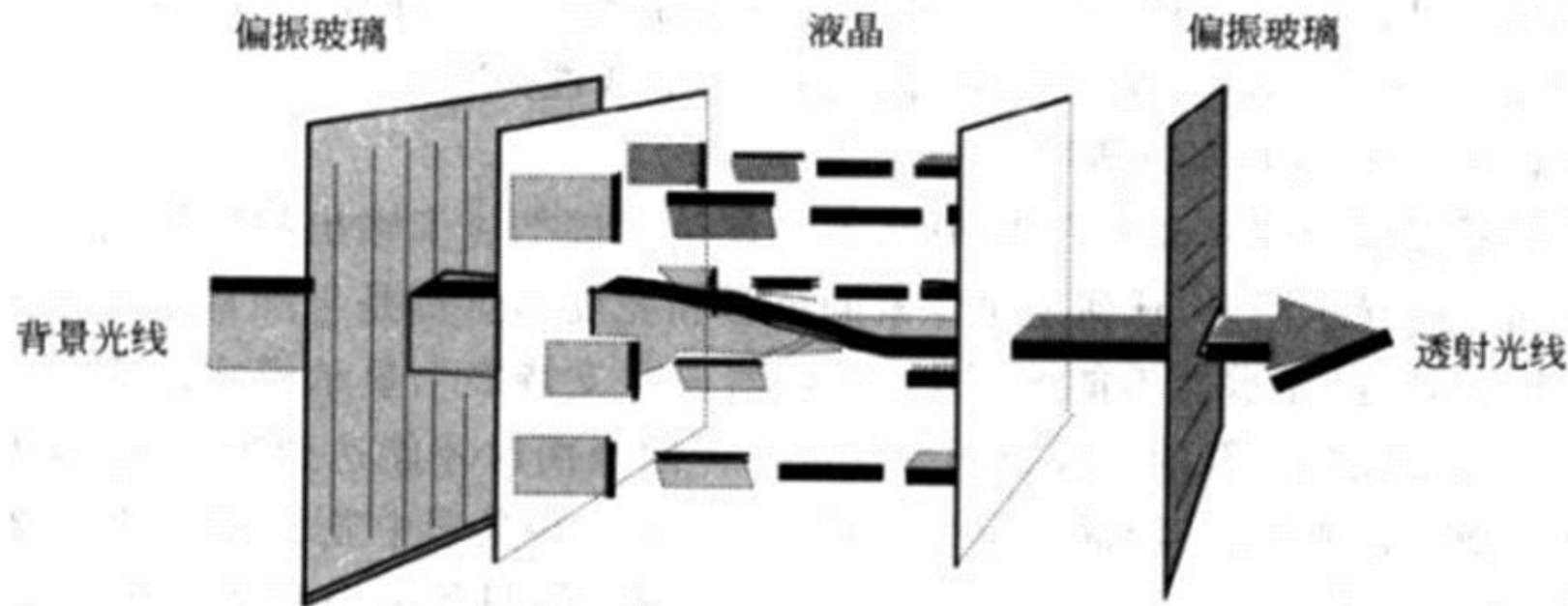
鼠标

在光电鼠标内部有一个发光二极管，通过该发光二极管发出的光线，照亮光电鼠标底部表面，然后将光电鼠标底部表面反射回的一部分光线，经过一组光学透镜，传输到一个光感应器件内成像。



这样，当光电鼠标移动时，其移动轨迹便会被记录为一组高速拍摄的连贯图像，最后利用光电鼠标内部的一块专用图像分析芯片对移动轨迹上摄取的一系列图像进行分析处理，通过对这些图像上特征点位置的变化进行分析，来判断鼠标的移动方向和移动距离，从而完成光标的定位。

显示器



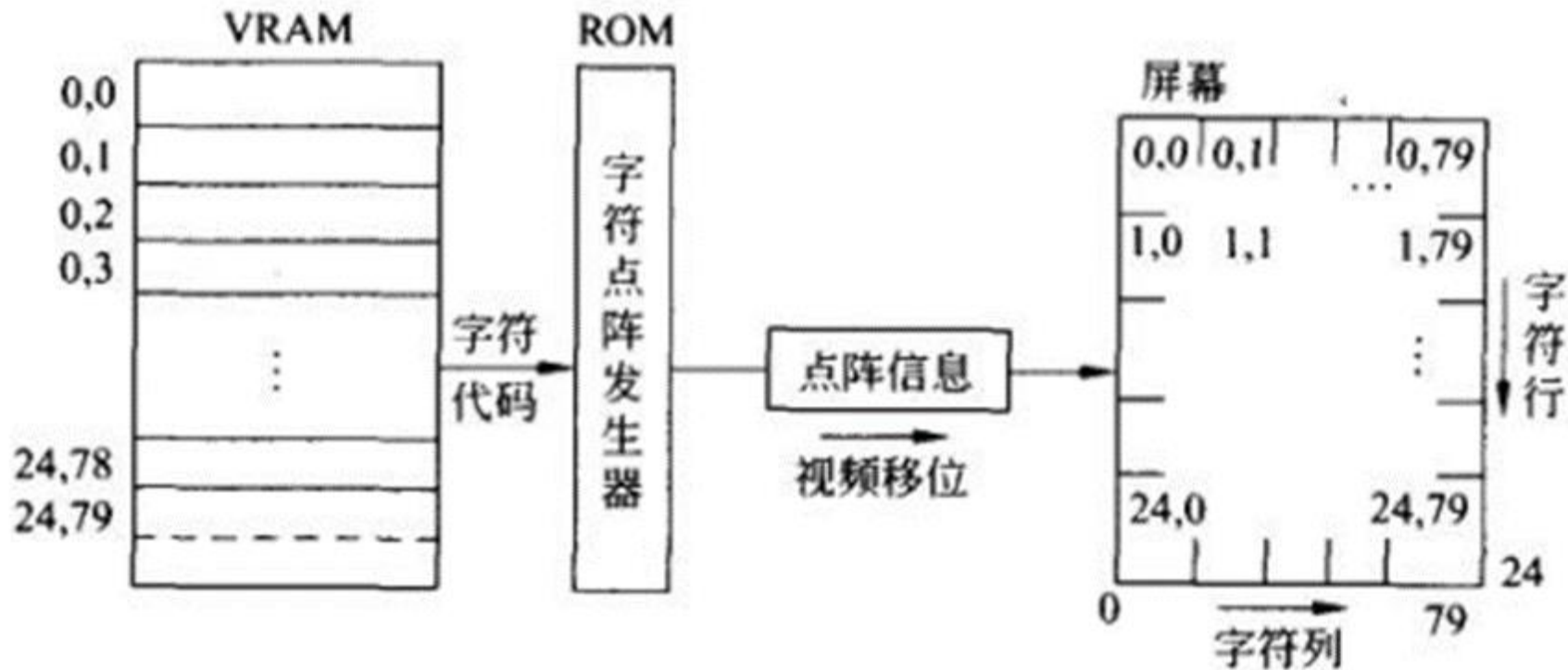
在玻璃板与液晶材料之间还有透明的电极（图中未画出），电极分为若干行和若干列，它与像素对应，电极的多少决定了液晶显示器的分辨率。在行与列的交叉点上，电荷使一些液晶分子发生扭曲，使光波经过时也发生了弯曲。当光线到达第二块偏振玻璃时，有的彩色光透过，而有的光线没有透过，从而在显示屏幕上形成图像。

显示适配卡与显示缓存

- 显示适配卡（也叫显卡）是显示器与主机之间的接口电路，负责将主机发送的待显示的信号送给显示器。
- 工业上为了便于生产，制定了一些显卡标准。早期的显卡标准有CGA、EGA等，现在常用的标准有VGA、TVGA、XGA等
- 在显示器上显示图像，实际上是把一帧图像存放在显示缓存中，以帧频的速率用缓存的内容刷新屏幕。
- 显示器一般可实现字符显示和图形显示两种功能。不论哪种显示，一帧图像的信息要存入显示缓存VRAM中。

字符显示原理

VRAM存放一帧待显示的字符代码，这些字符排列次序与在屏幕上的显示位置密切相关。一个屏幕被划分成若干字符行，每个字符行又划分成若干字符列。

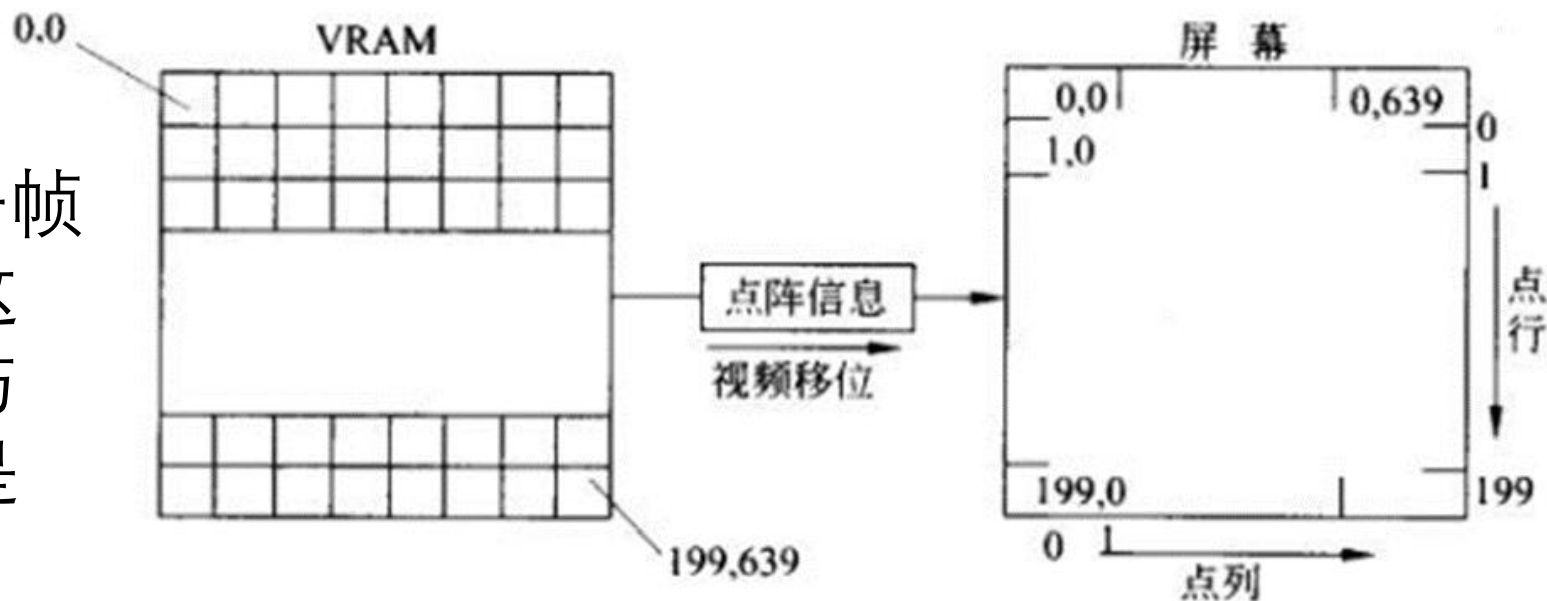


于是，屏幕显示字符的位置由字符行地址与字符列地址确定，而两者的组成——字符行列地址即是访问VRAM的实际地址。

由VRAM中取出的一个字符代码作为字符发生器ROM的地址，即可从ROM获取相应的字符点阵信息。这些信息按当前扫描的位置被送到移位寄存器，然后串行输出视频信号

图形显示原理

VRAM存入的是一帧待显示的图形信息，这些图形信息排列次序与在屏幕上显示位置也是密切相关的。



- 一个屏幕被划分为几百至几千个水平点和几百至几千个垂直点，于是屏幕上显示的一个点位置由点行地址与点列地址确定，两者的组合——点行列地址即对应VRAM中的一个字节或字节中的相应位。
- 由VRAM中取出的一个图形字节不需要访问ROM，而直接送入移位寄存器，然后串行输出视频信号

硬盘

- 硬盘由驱动器、控制器和盘片三部分组成
- 硬盘驱动器内部包含有旋转轴驱动部件、磁头定位部件、读写电路和数据传送电路等。
- 硬盘控制器的任务是接收主机发送的命令和数据，并转换成驱动器的控制命令和驱动器所要求的数据格式，控制驱动器的读写操作。一个控制器可以控制一台或多台驱动器。
- 盘片是存储信息的介质,硬盘的盘片一般以铝合金为基体，由多片组合在一起，形成盘片组。

硬盘结构和扇区

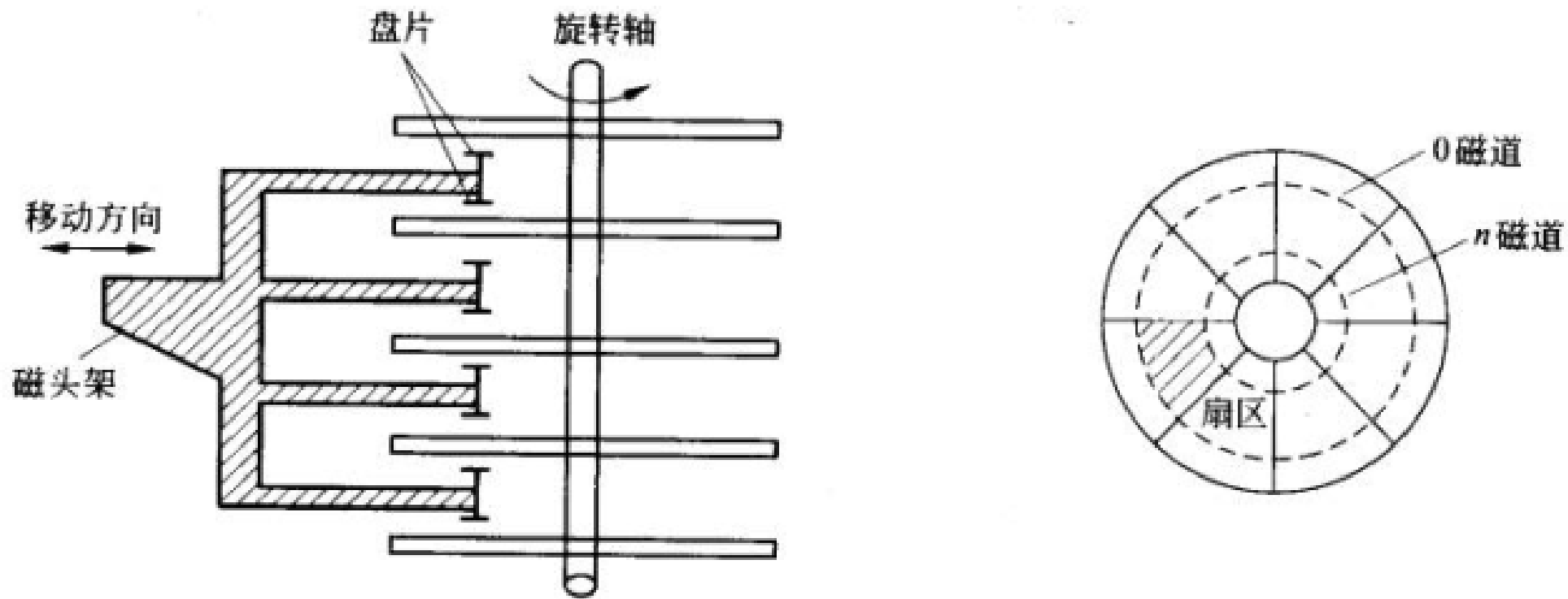
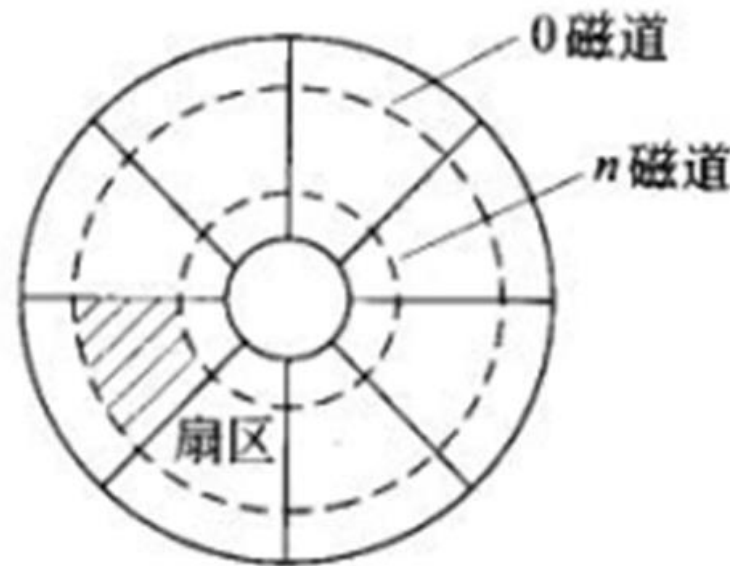


图 6.41 活动头硬盘结构和扇区

硬盘地址： 柱面号+盘面号+扇区号

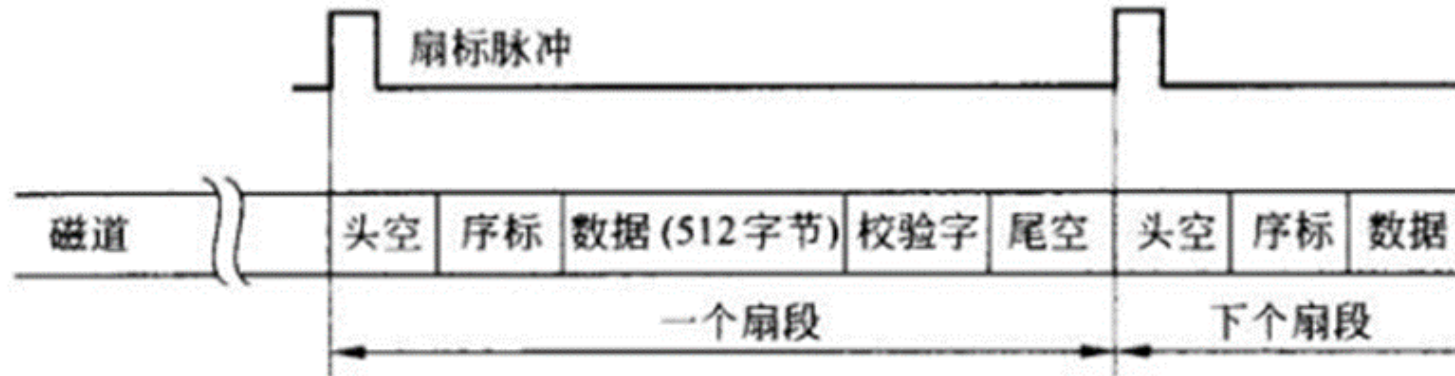
- 盘面
- 磁道
- 扇区
- 扇段
- 柱面
- 如果主机配有几台硬盘驱动器，则还应给出驱动器编号，用来选择所需的驱动器，此时硬盘信息的**地址格式**为：
- **驱动器号、柱面号、盘面号、扇区号**



硬盘和内存间的数据交换

- 硬盘和主存间的数据交换可通过DMA或通道控制完成。为了保证写入时数据的可靠性，通常在写操作以后启动一个读操作，把从硬盘读出的内容与从主存相应的单元读出的内容进行比较，如果不一致，则经中断系统向CPU送一个出错信息。
- 硬盘信息访问的基本单位是一个磁道的扇区部分，即扇段。

扇段的信息记录格式



由硬盘控制器产生的扇标脉冲标志着一个扇区的开始。每个扇区由头部空白、序标、数据、校验字、尾部空白等字段组成，其中空白段用来作为地址定位缓冲，便于硬盘控制器做好读写准备；序标部分指出本扇区的地址，以及作为硬盘控制器的同步定时信号；之后即为本扇区记录的数据；校验字用来校验读出数据是否正确，一般采用循环校验码。

寻址时间

- 在活动头系统中，当访问硬盘中某一扇段时，必须由磁道定位机构把读写磁头沿硬盘半径方向移到相应的磁道位置上，这一时间称为**定位时间**。定位时间取决于磁头的起始位置与所要求磁道间的距离。定位以后寻找所需扇区的时间称为**等待时间**，或称旋转延迟，平均值为硬盘旋转半圈的时间，可为几个毫秒。上述两个延时之和称为硬盘的**寻址时间**。
- 读写操作总是从扇区的边界开始，每次交换一个扇段的信息。如果写入的内容不满一个扇段，则在该扇段的余下部分重复数据的最后一位。