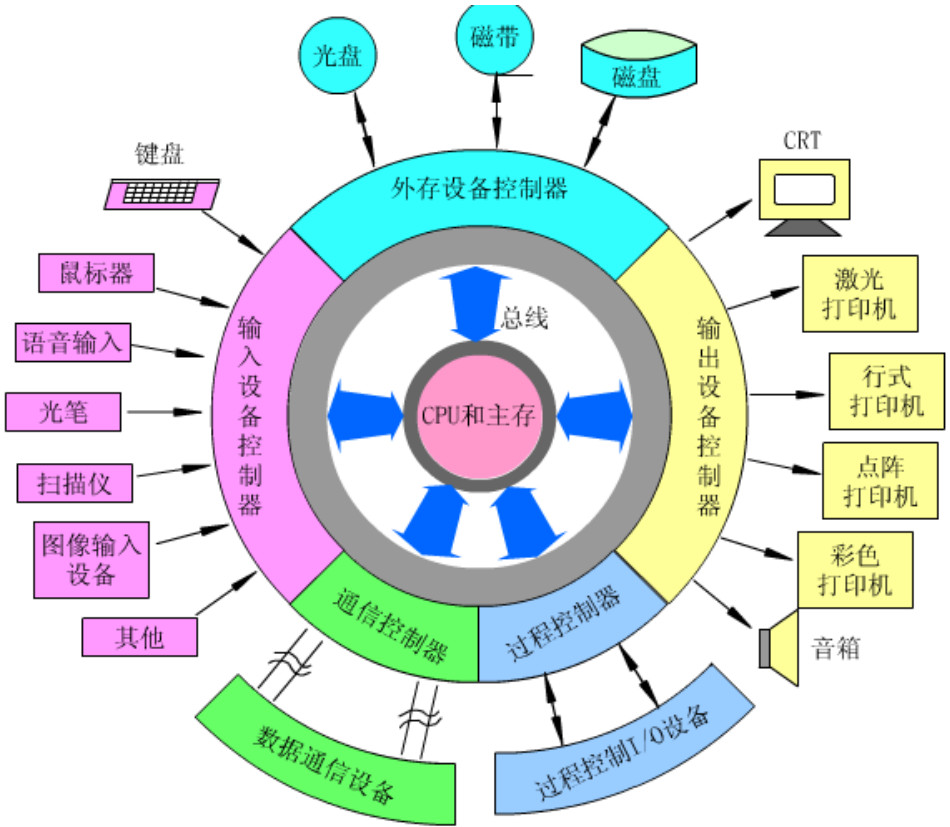
第八章 输入输出系统

- 8.1 外围设备的定时方式和信息交换方式
- 8.2 程序查询方式
- 8.3 程序中断方式
- 8.4 DMA方式
- 8.5 通道方式
- 8.6 通用I/O标准接口
- 输入输出系统是计算机硬件系统中除CPU和存储器外的第三个关键部分；
- I/O设备的数量和种类多，与主机的联络方式及信息交换方式也各不相同，输入输出系统涉及I/O设备和各种I/O如何与主机交换信息；
- 重点分析I/O设备与主机交换信息的三种控制方式：程序直接控制方式、中断方式、DMA方式

8.1 外围设备的速度分级与信息交换方式

- **8.1.1 外围设备的速度分级：**
  - 外设种类繁多，在CPU与外设信息交换时，存在以下几种情况：
    - 不同种类的外设数据传输速率差别很大，同一种设备在不同时刻传输速率也可能不同
    - 不同外设输入/输出信号的形式多样：电，光，数字式，模拟式

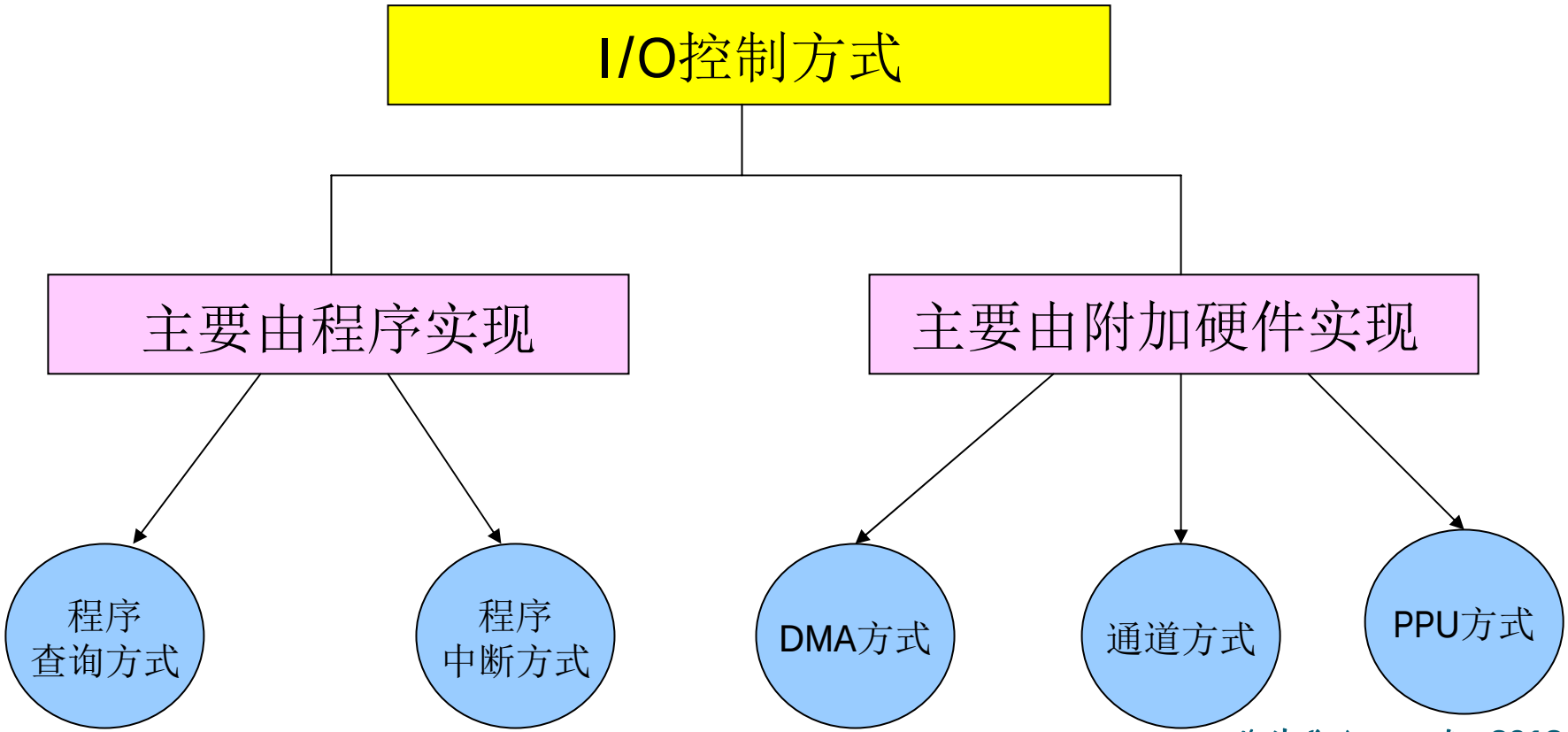


8.1.1 外围设备的速度分级

- 输入/输出设备同CPU交换数据的过程：
  - 输入过程：
    - (1)CPU把地址值放在地址总线上，选择某设备；
    - (2)CPU等候输入设备的数据成为有效；
    - (3)CPU从数据总线读入数据，并放入寄存器。
  - 输出过程：
    - (1)CPU把地址值放在地址总线上，选择设备；
    - (2)CPU把数据放在数据总线上；
    - (3)输出设备认为数据有效，从而把数据取走。
- 高速的CPU与速度参差不齐的外设怎样在时间上同步呢？
  - 解决办法是在CPU和外设之间数据传送时加以定时。
    - 速度极慢或简单的外设：CPU只需要接受或者发送数据即可，不需定时，比如机械开关，显示二极管等；
    - 慢速或者中速的设备：可以采用异步定时的方式/应答式定时；
    - 高速外设：采用同步定时方式，直接使用时钟周期定时；

8.1.2 信息交换方式

- **输入输出控制**：实现CPU与外围设备进行信息交换方式(幼儿园吃糖问题提示P236：分类管理)
  - 程序查询方式　　程序中断方式　直接内存访问（DMA）方式
    通道方式　　外围处理机方式



8.1.2 信息交换方式

- 程序查询方式：
  - 数据在CPU和外围设备之间的传送完全靠计算机程序控制；
  - 优点
    - CPU的操作和外围设备的操作能够同步，而且硬件结构比较简单。
  - 缺点
    - 外围设备动作很慢，程序进入查询循环等待时，将白白浪费掉CPU很多时间，CPU此时只能等待，不能处理其他业务。
    - 即使CPU采用**定期地由主程序转向查询设备状态的子程序**，进行扫描查询的办法，CPU宝贵资源的浪费也是可观的。因此当前除单片机外，很少使用程序查询方式。

8.1.2 信息交换方式

- 程序中断方式：
  - 中断：外围设备准备好，“主动”通知CPU，准备送出输入数据或接收输出数据的一种方法。
  - 中断过程：当一个中断发生时，CPU暂停它的现行程序，而转向中断处理程序，从而可以输入或输出一个数据。当中断处理完毕后，CPU又返回到它原来的任务，并从它停止的地方开始执行程序。
  - 中断方式节省CPU宝贵时间，是管理I/O操作的一个比较有效的方法。
    - 中断方式一般适用于随机出现的服务，并且一旦提出要求，即可立即进行。同程序查询方式相比，硬件结构相对复杂一些，服务开销时间较大。
    - 中断方式交换数据时，每处理一次I/O交换，约需几十微秒到几百微秒。对于一些高速的外围设备，以及成组交换数据的情况，仍然显得速度太慢。

8.1.2 信息交换方式

- 直接内存访问（DMA）方式：
  - 直接内存访问(DMA)方式是一种完全由硬件执行I/O交换的工作方式。
  - DMA既考虑到中断响应，同时又要节约中断开销。数据交换时，DMA控制器从CPU完全接管对总线的控制，数据交换不经过CPU，而直接在内存和外围设备之间进行，以高速传送数据。
  - DMA方式优点是数据传送速度很高，传送速率仅受到内存访问时间的限制。
  - 与中断方式相比，DMA方式需要更多的硬件，其主要适用于内存和高速外围设备之间大批数据交换的场合。

8.1.2 信息交换方式

- 通道方式：
  - 通道：是一个具有特殊功能的处理器，某些应用中称为输入输出处理器(IOP)，它可以实现对外围设备的统一管理和外围设备与内存之间的数据传送。
  - 与DMA方式相比，通道方式进一步提高了CPU的效率。然而这种提高CPU效率的办法是以花费更多硬件为代价的。
- 外围处理机方式：
  - 外围处理机(PPU)方式是通道方式的进一步发展。由于PPU基本上独立于主机工作，它的结构更接近一般处理机，甚至就是微小型计算机。在一些系统中，设置了多台PPU，分别承担I/O控制、通信、维护诊断等任务。从某种意义上说，这种系统已变成分布式的多机系统。

8.1.2 信息交换方式

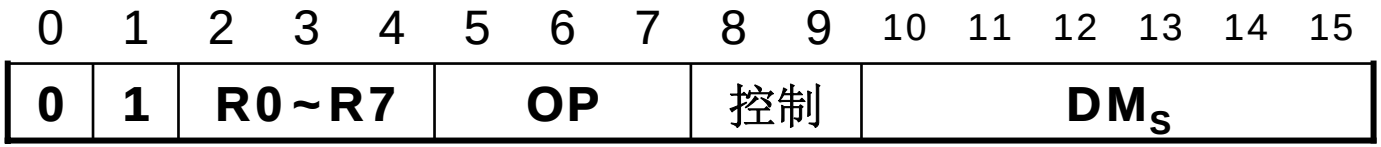
- 程序查询方式和程序中断方式适用于数据传输率比较低的外围设备，主要由软件方式完成；
- DMA方式、通道方式和PPU方式则适用于数据传输率比较高的设备，其功能则更侧重于由附加硬件电路来完成。
- 目前，单片机和微型机中多采用程序查询方式、程序中断方式和DMA方式。通道方式和PPU方式大都用在中、大型计算机中。

8.2 程序查询方式

- 程序控制I/O方式：
  - 通过由I/O指令所编的程序，来控制主机与外设之间的信息传送。
  - 特点：CPU与外围设备的信息交换过程完全由计算机程序控制，所有过程在CPU的主动控制下进行。
  - 过程：当需要输入/输出时，CPU暂停执行主程序，转去执行设备输入/输出服务程序，根据服务程序中的I/O指令进行信息交换。
    - 先由主机通过启动指令启动外设工作，启动后主机用测试指令不断查询外设工作是否完成，一旦外设工作完成，就可进行数据传送了。

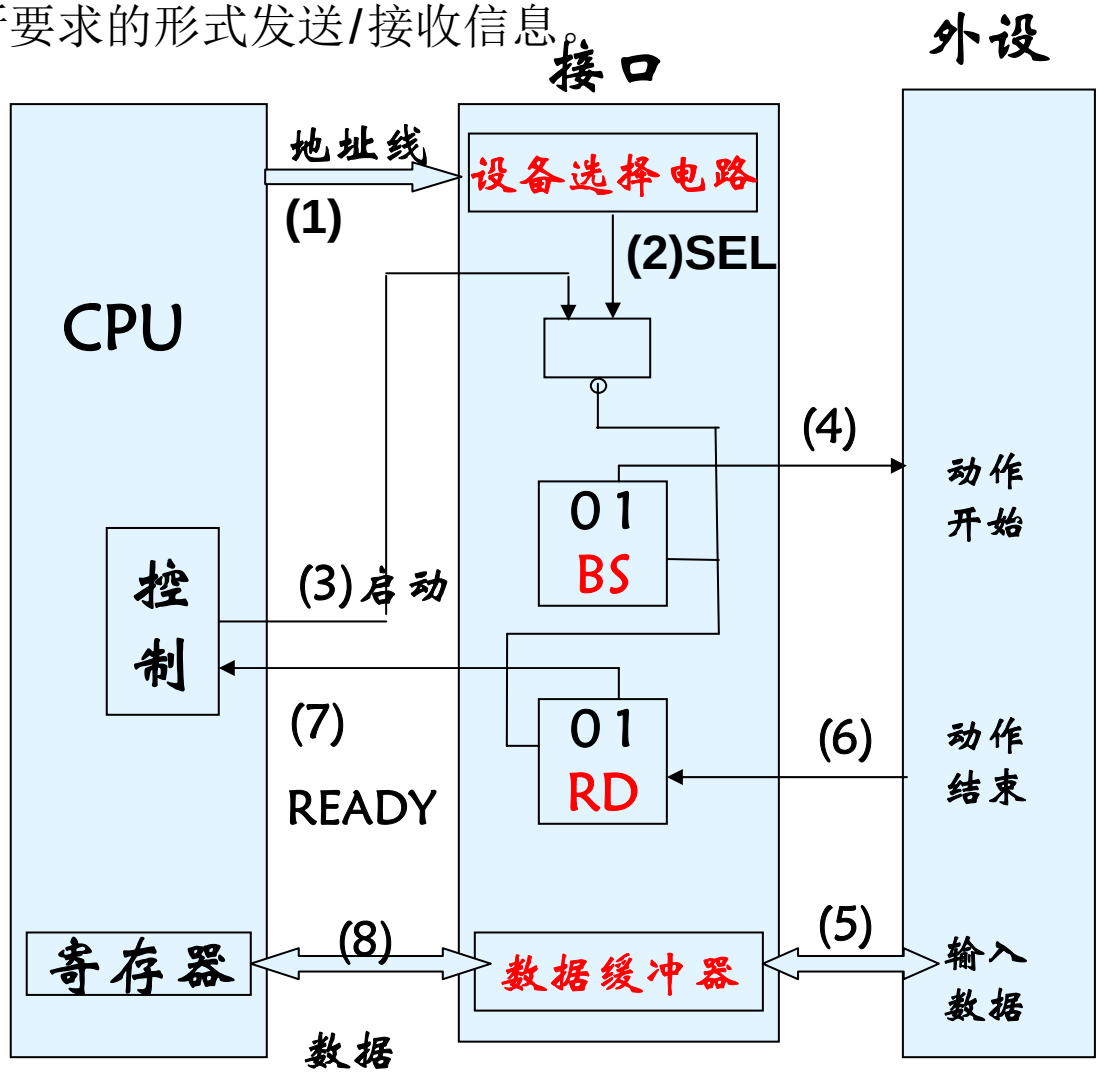
8.2 程序查询方式

- 设备编址：I/O程序识别进行信息交换的外围设备。
  - 统一编址：I/O设备中的控制寄存器，数据寄存器，状态寄存器等和内存单元一起进行编址，可以使用访问内存的指令访问外围设备。
  - 单独编址： I/O设备和内存单元分开单独编址，使用不同的指令访问I/O设备和内存单元。
- 输入输出指令：I/O指令编码
  - **0~1**：访问I/O或内存，01访问I/O；
  - **2~4**：选择通用寄存器；
  - **5~7**：OP，提供读，写，测试等功能；
  - **8~9**：控制功能，01：启动；11：停止；等
  - **10~15**：外部设备地址；



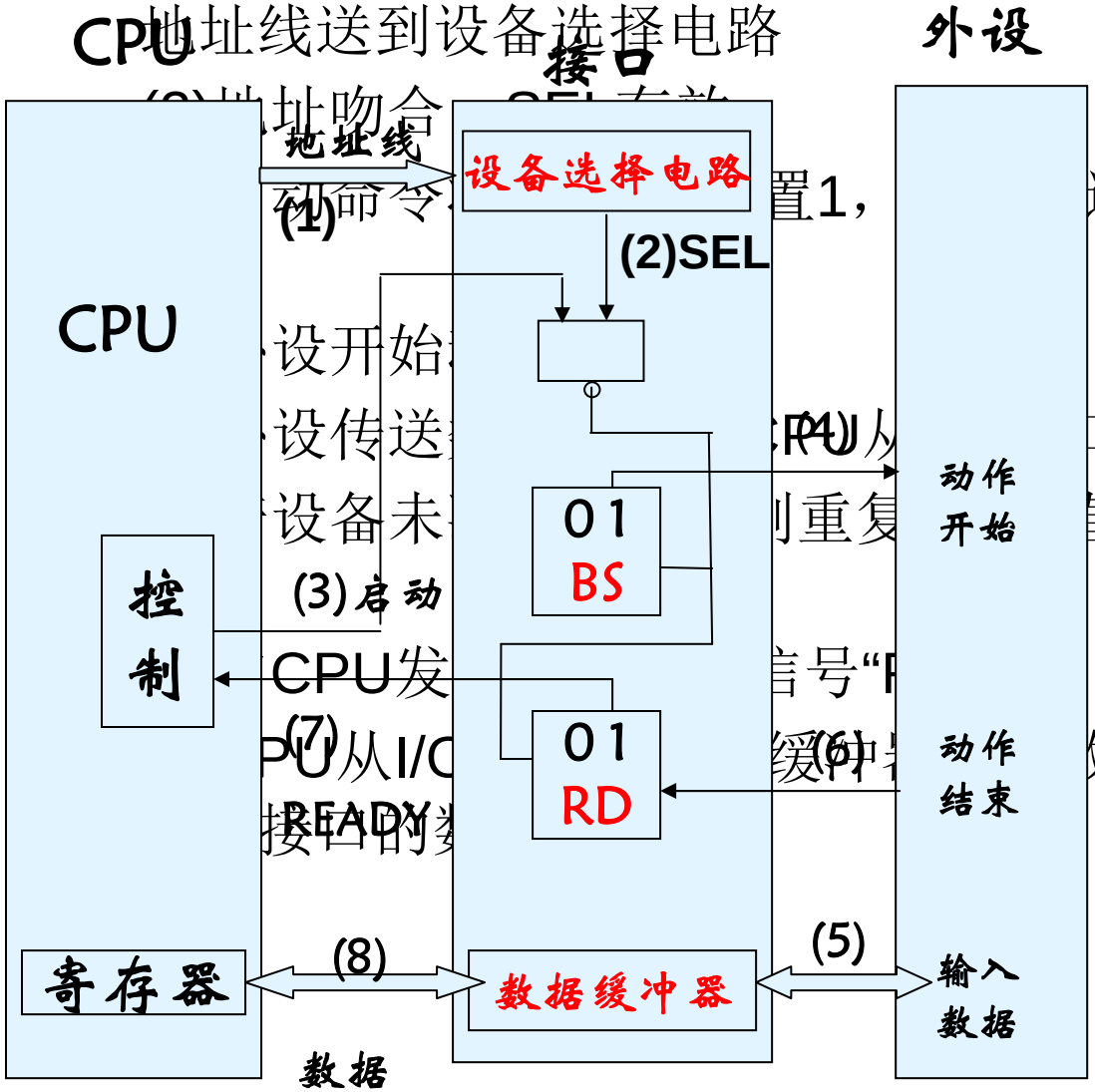
8.2 程序查询方式

- 程序查询方式的接口：组成：设备选择电路，数据缓冲器，状态设备状态标志。
  - 接口功能：使用接口，连接外部设备和总线，保证外部设备用计算机系统特性所要求的形式发送/接收信息。



8.2 程序查询方式

- 程序查询方式的接口：工作过程
  - (1)当CPU通过I/O指令启动设备时，指令的设备地址码字段通过地址线送到设备选择电路
  - (2)地址吻合，SEL有效
  - (3)启动命令将BS触发器置1，
  - (4)外设开始动作
  - (5)外设传送数据，同时
  - (6)若设备将RD触发器
  - (7)向CPU发出准备就绪
  - (8)CPU从I/O接口的数据



(1)从I/O接口读入状态字令启动设备时，指令的设备地址码字段通过地址线送到设备选择电路好，将RD触发器置1

(2)地址吻合，SEL有效

(3)启动命令将BS触发器置1，请求传送输入数据，同时CPU撤销数据从CPU输出至接口的数据缓冲器

(4)外设开始动作

(5)外设传送数据，同时CPU从I/O接口读入状态字令启动设备时，指令的设备地址码字段通过地址线送到设备选择电路好，将RD触发器置1

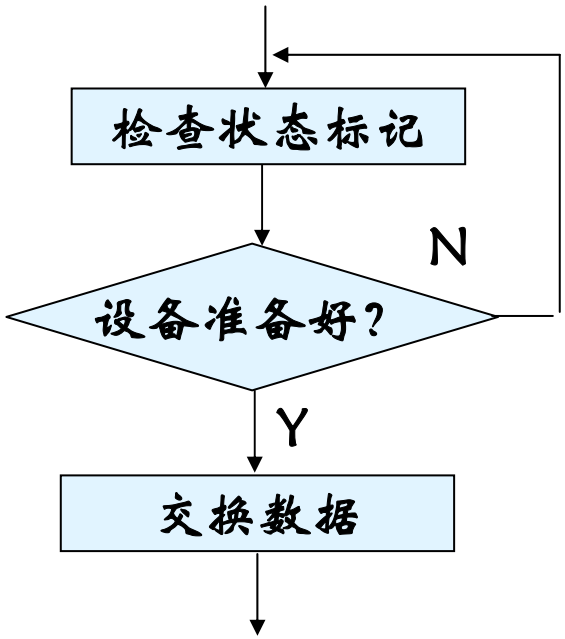
(6)若设备将RD触发器则重复(5)，直至准备好

(7)向CPU发出准备就绪信号“Ready”

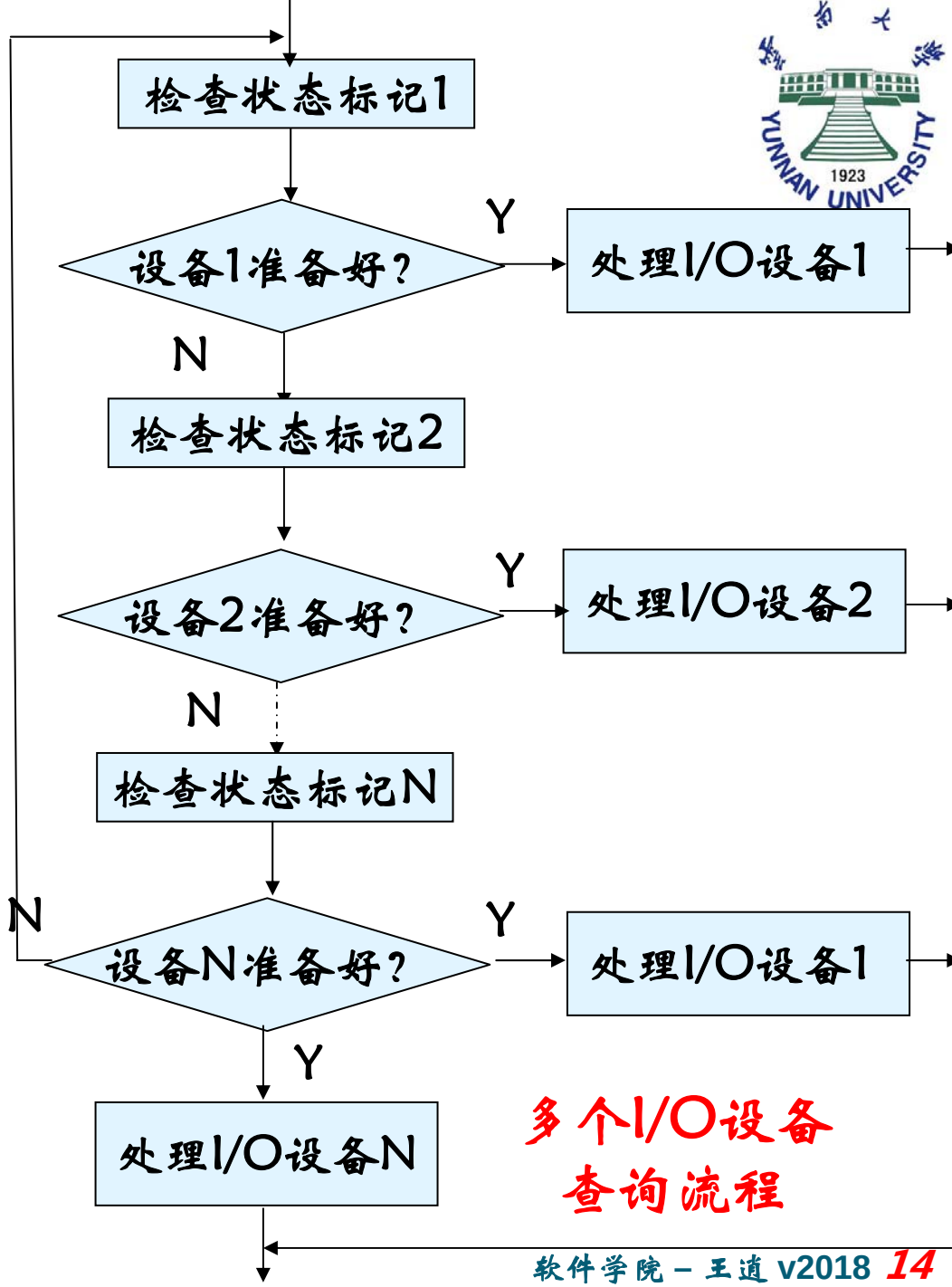
(8)CPU从I/O接口的数据缓冲器读取数据，或者将数据从CPU输出至接口的数据缓冲器置0

8.2 程序查询方式

- 程序查询I/O设备流程图：



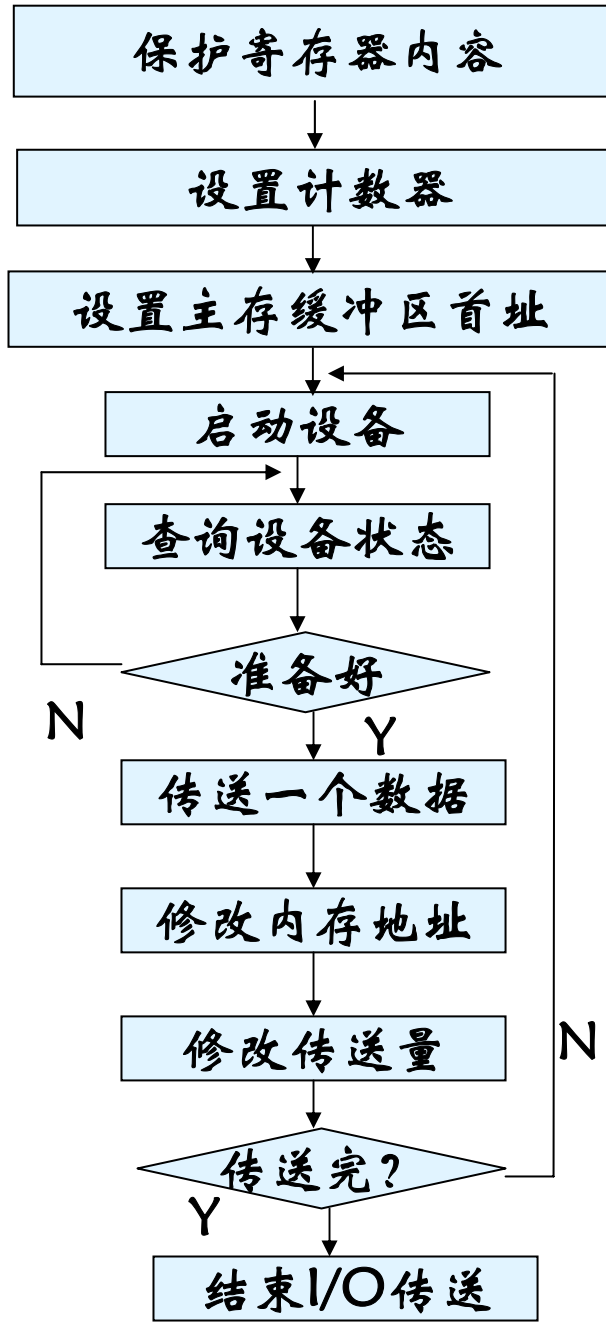
单个I/O设备查询流程



多个I/O设备查询流程

8.2 程序查询方式

- I/O服务子程序流程图：



8.2 程序查询方式

- 程序查询方式的特点：
  - CPU能控制何时对何设备进行输入输出操作
  - 外围设备和CPU处于异步工作关系
  - 数据的输入输出要经过CPU
  - 用于连接低速外围设备
  - 接口设计简单，灵活性好
  - CPU在信息传送过程中要花很多时间用于查询和等待，效率低

8.3 程序中断方式

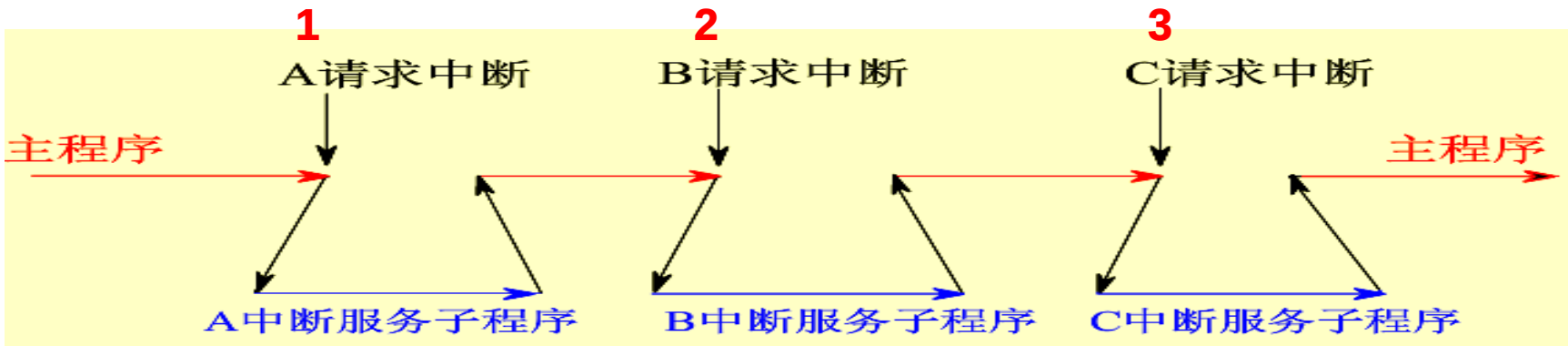
- 信息交换方式的优化：
  - CPU在信息交换中是主方，具有决策权，正常做它应该做的事，不需要浪费时间浪费资源去测试是否有信息交换出现；
  - 当有信息交换出现的时候，CPU表态是否同意或不同意。
  - 信息交换出现应该使用某一种主动去通知CPU，然后等待CPU的决策处理。
  - 中断是主动去通知CPU有某个需要处理的事情出现的技术。
- 中断技术的出现，提供了一种以响应外部异常事件而改变状态流程的有效手段，它支持了多重程序的运行及多个用户同时共享整个计算机资源，充分发挥了计算机的高速处理和实时处理能力，以及自动处理机内部故障的能力。
- 中断方式：CPU与外设并行工作方式。程序查询方式：串行工作方式。

8.3.1 中断的基本概念

- 基本概念：
  - 中断（Interrupt）：指CPU暂时中止现行程序，转去处理随机发生的紧急事件，处理完后自动返回原程序的功能和技术。
  - 中断系统：是计算机实现中断功能的软硬件总称。
  - 一般在CPU中设置中断机构，在外设接口中设置中断控制器，在软件上设置相应的中断服务程序。
  - 中断技术：实现中断所需的软硬件技术。
  - 中断服务程序：中断发生时，需要执行的程序代码。
  - 中断控制器：管理中断请求的硬件，8255。

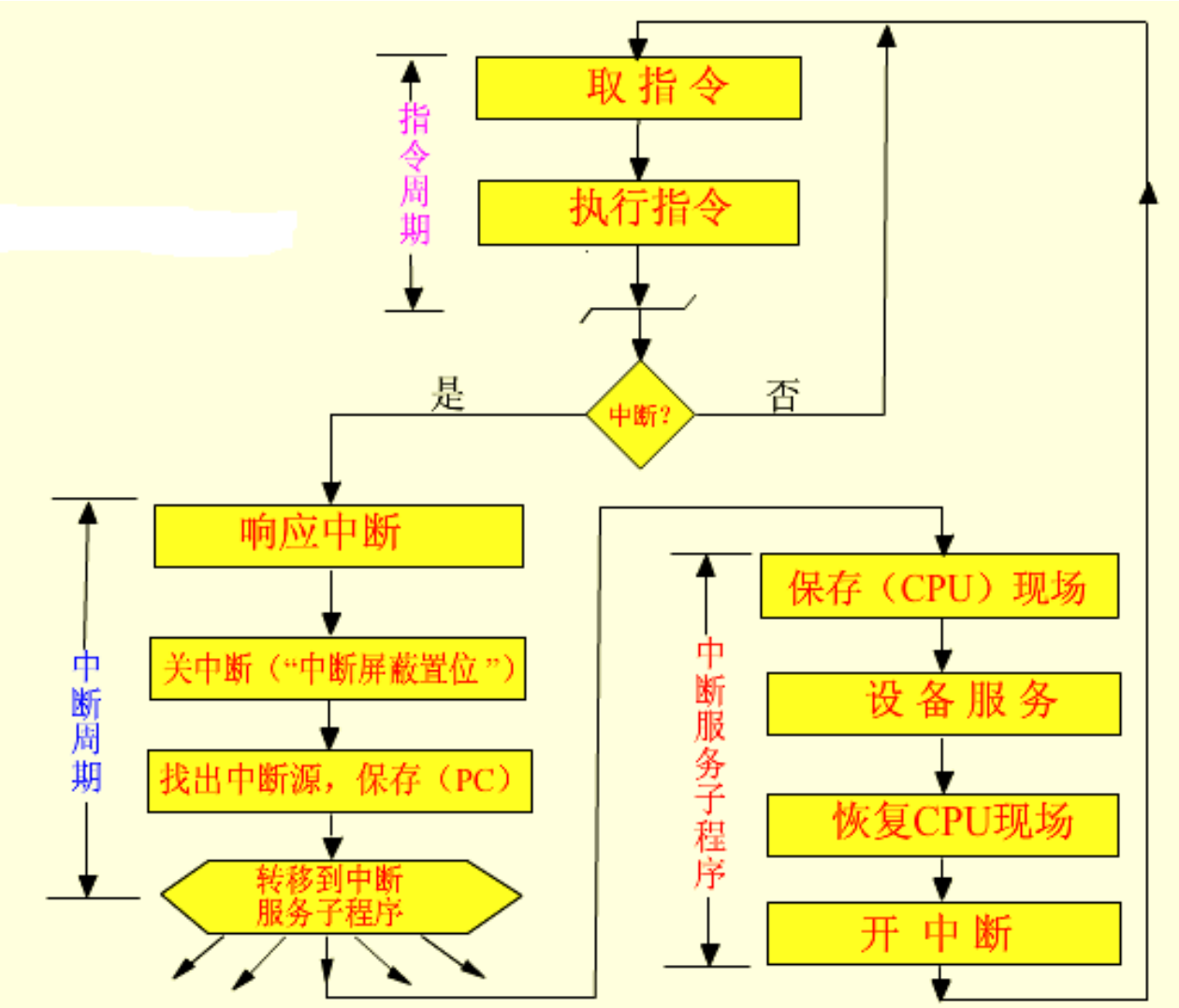
8.3.1 中断的基本概念

- 中断处理示意图：
  - CPU正常执行主程序
    - 在时刻1，A请求中断，如果响应A的中断请求，暂停执行主程序，转入执行A中断服务子程序；否则，屏蔽A中断请求；
    - 在时刻2，……；在时刻3，……； ……；



8.3.1 中断的基本概念

- 中断处理流程图：
  - CPU 现场：CPU中寄存器的取值；中断是一种公操作。



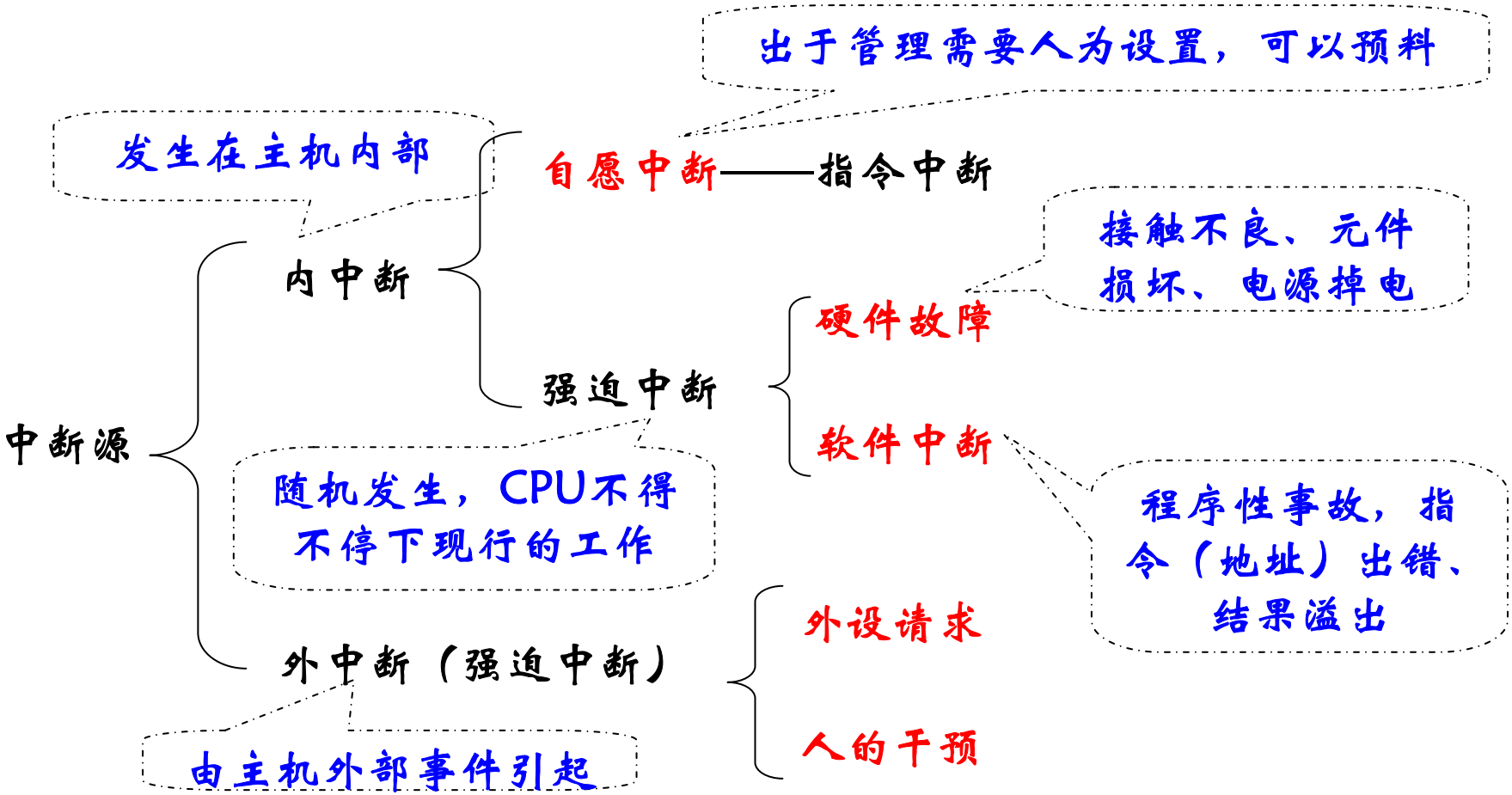
8.3.1 中断的基本概念

- 中断处理几个问题：
  - 响应中断时机：外界中断请求时间是随机的，但CPU只有在当前指令执行完毕后，才转至中断公操作。
  - 断点/现场保护问题：为了保证中断响应后CPU能正常继续执行主程序，需要在进入中断服务程序前保存PC寄存器和CPU中标志寄存器的取值，以便在执行完中断服务程序返回后可以以原来的环境继续执行主程序；
    - 断点：PC寄存器值；现场：标志寄存器的值；
  - 中断屏蔽：使用IF标志位的取值来控制是否响应中断，避免中断嵌套；
    - **IF=1**，开中断，响应。
    - **IF=1**，关中断，不响应。
  - 中断是由软硬件结合起来实现的。

8.3.1 中断的基本概念

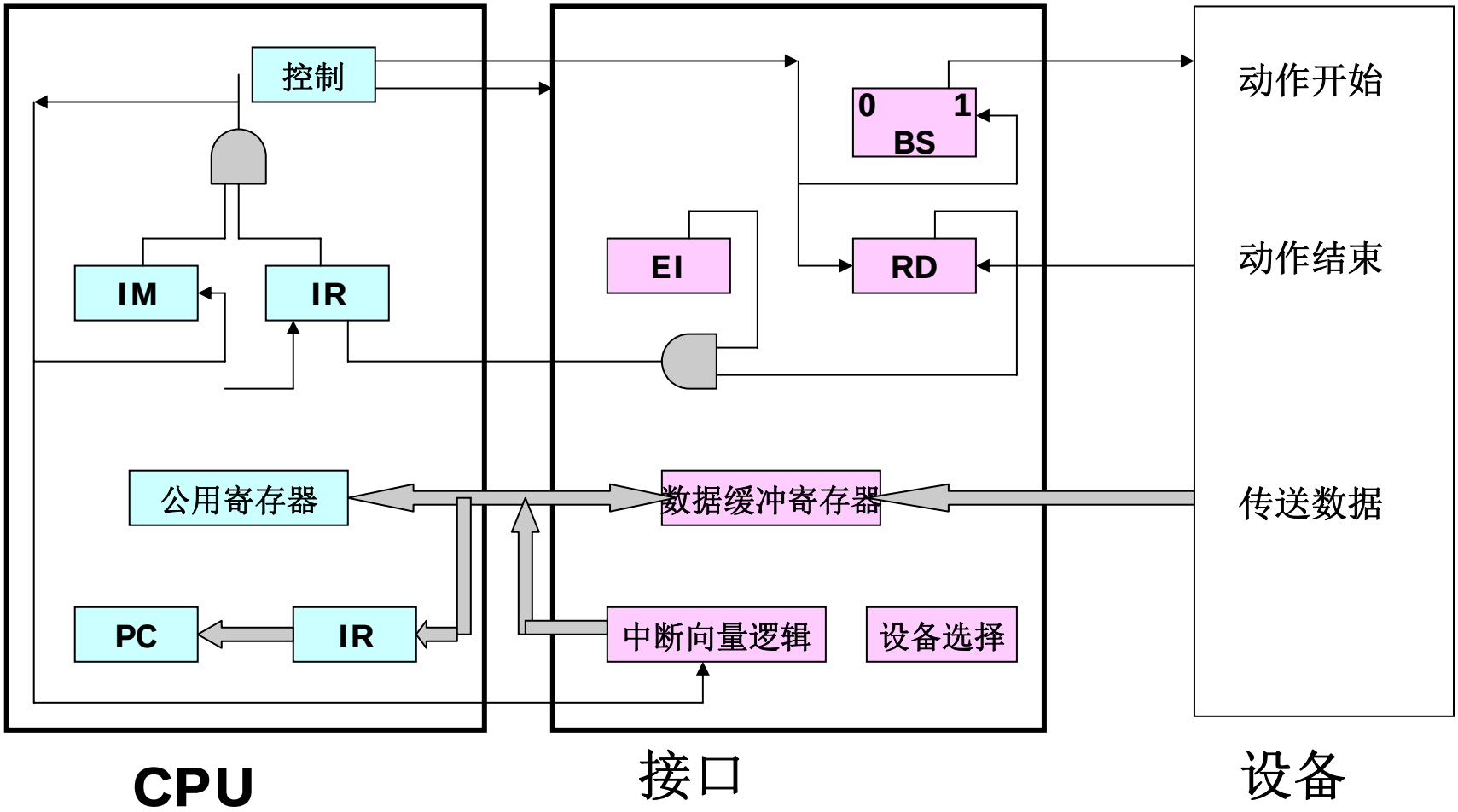
- 中断的类型：
  - 按中断处理方法分类：
    - 程序中断：主机响应中断请求后，通过执行一段程序来处理有关的事宜。
    - 简单中断：主机响应中断请求后，不需要执行服务程序，而是让出一个或几个主存周期，使I/O设备和主存直接交换数据。
  - 按中断源分类：中断源是请求CPU中断的设备或事件
  - 强迫中断：是随机产生的。产生强迫中断的中断源有四个方面：
    - 内中断：由硬件故障及程序故障引起的中断。
    - 外中断：是由系统配置的外部设备引起的中断。
    - 正在执行着的现行程序所引起的中断。
    - 处理机之间的中断。
  - 自愿中断（程序自中断）：事先在程序某初设置断点，并借用中断处理方式保护现场，引出一段服务程序。

- 中断的类型：


中断源
内中断
发生在主机内部
外中断（强迫中断）
由主机外部事件引起
自愿中断——指令中断
出于管理需要人为设置，可以预料
强迫中断
随机发生，CPU不得不停下现行的工作
硬件故障
接触不良、元件损坏、电源掉电
软件中断
程序性事故，指令（地址）出错、结果溢出
外设请求
人的干预

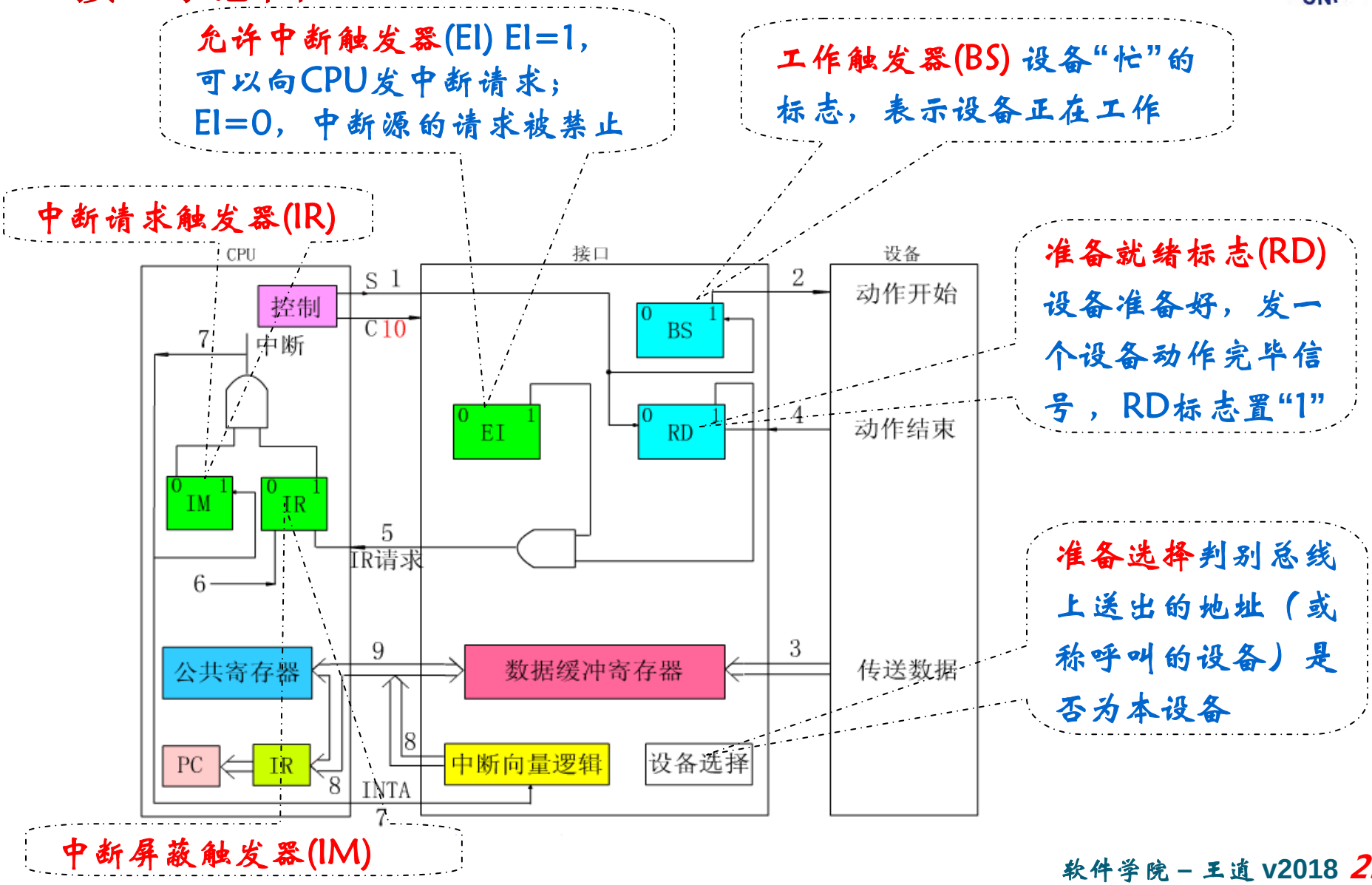
8.3.2 程序中断方式的基本I/O接口

- 接口示意图：



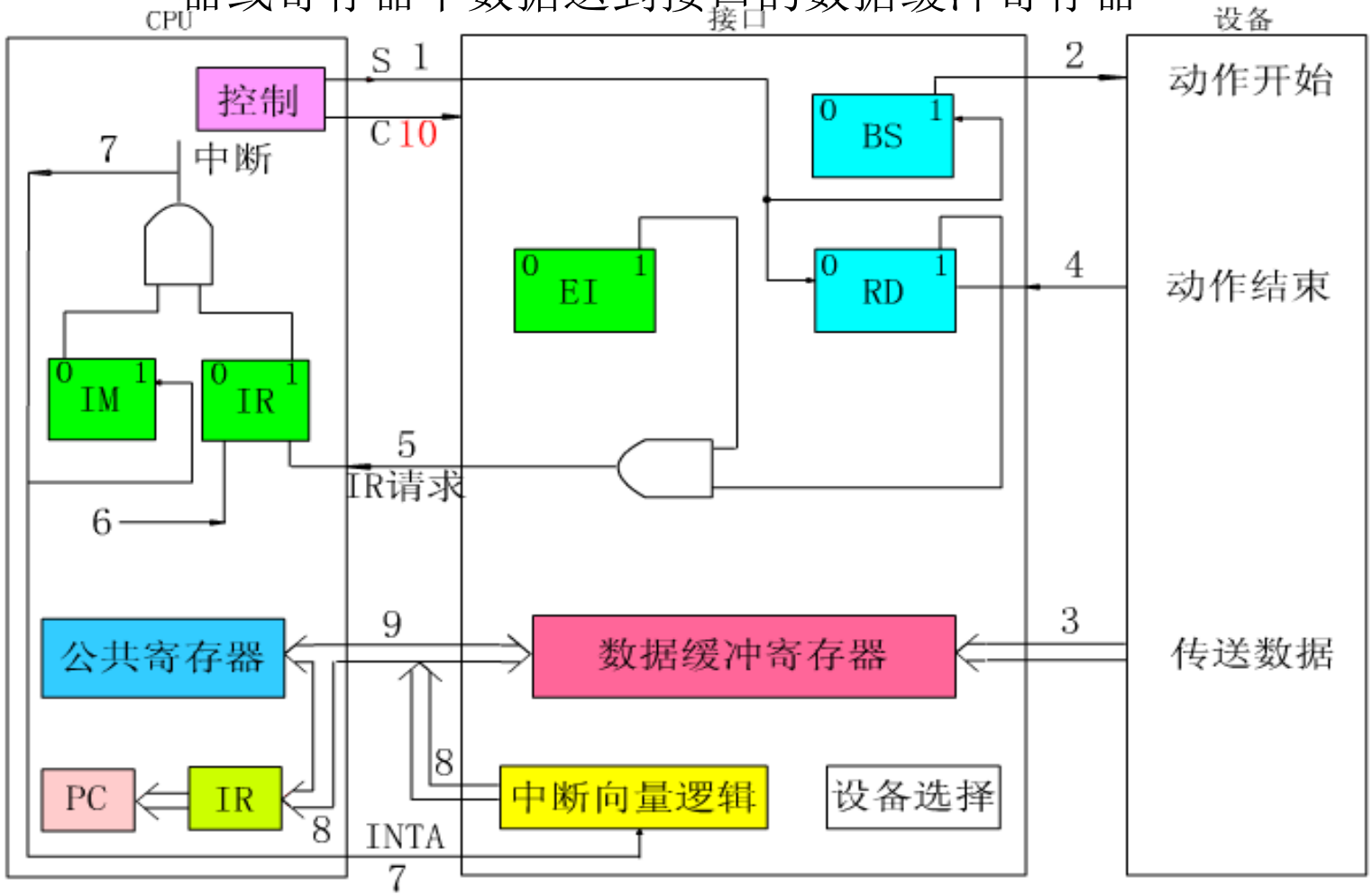
8.3.2 程序中断方式的基本I/O接口

- 接口示意图：



8.3.2 程序中断方式的基本I/O接口

- 工作过程：(9)中断服务程序通过输入指令把接口中数据缓冲寄存器的数据送CPU的累加器或寄存器，或通过输出指令把CPU的累加器或寄存器中数据送到接口的数据缓冲寄存器



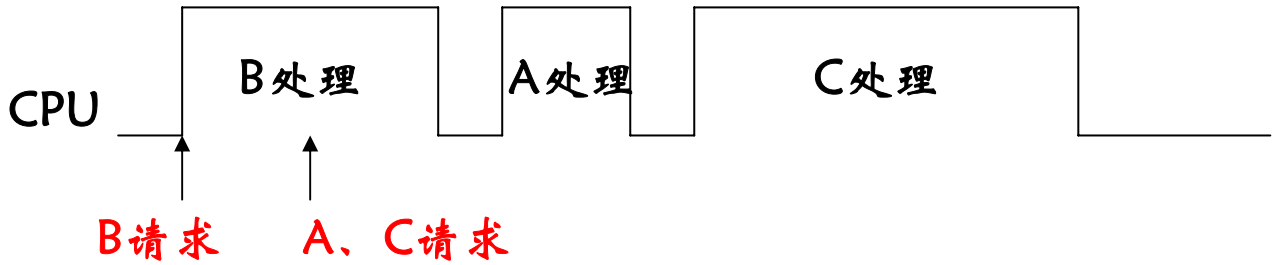
8.3.3 / 4 单级中断与多级中断

- 当几个设备同时要求中断时，CPU响应并处理的原则是优先级高的优先处理，当CPU正在处理低优先级设备时，出现了高优先级设备的中断请求，有两种解决方式：
  - 单级中断
    - 不同优先级的设备同时提出请求，CPU按优先级一个一个处理；
    - 当CPU正在处理某个中断时，不允许其他设备再中断CPU的程序；即使优先级高的设备也不能打断，只能等到中断处理完毕后，CPU才响应其他中断；
  - 多重中断
    - 在处理某一个中断时又发生了新的中断请求，从而中断该服务程序的执行，转去进行新的中断处理，这种重叠处理中断的现象又称为中断嵌套
    - 一般，在处理某级中的某个中断时，同级的或低级的新中断请求应不能中断它的处理，需处理完后再响应新的中断
    - 优先级高的新中断请求能中断它的处理

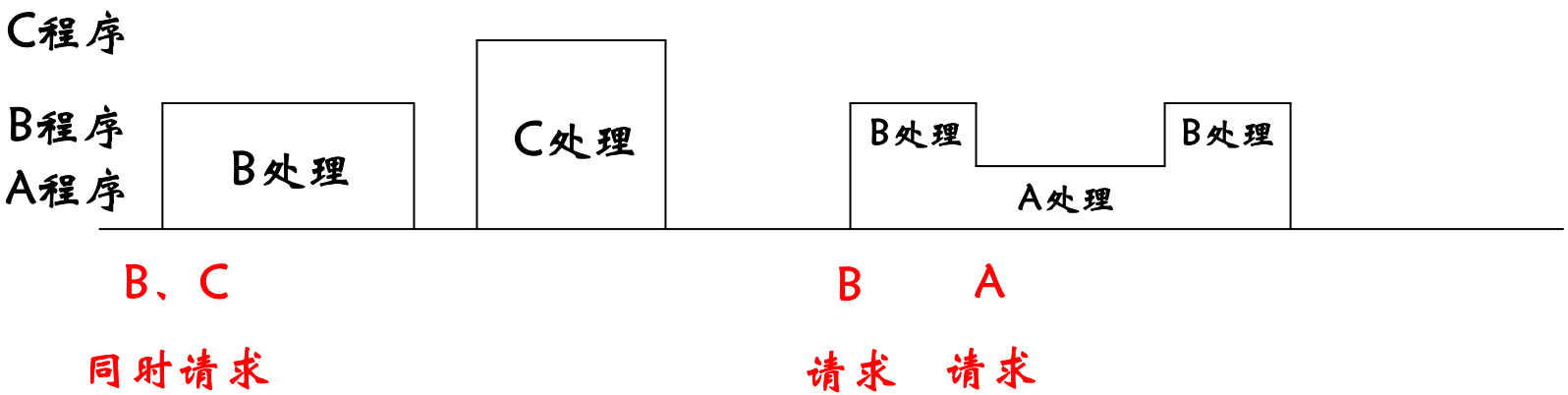
8.3.3 / 4 单级中断与多级中断

例如，3个中断优先等级是：A→B →C →CPU

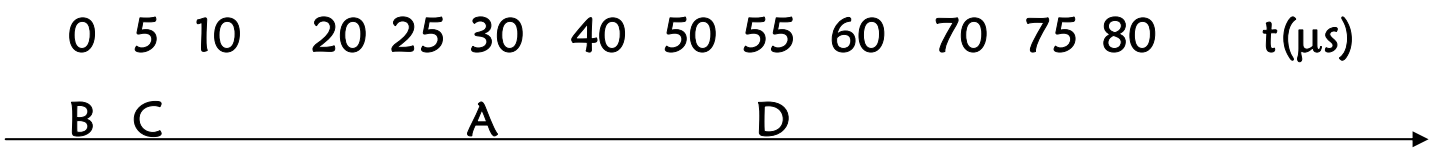
单级中断CPU运行轨迹

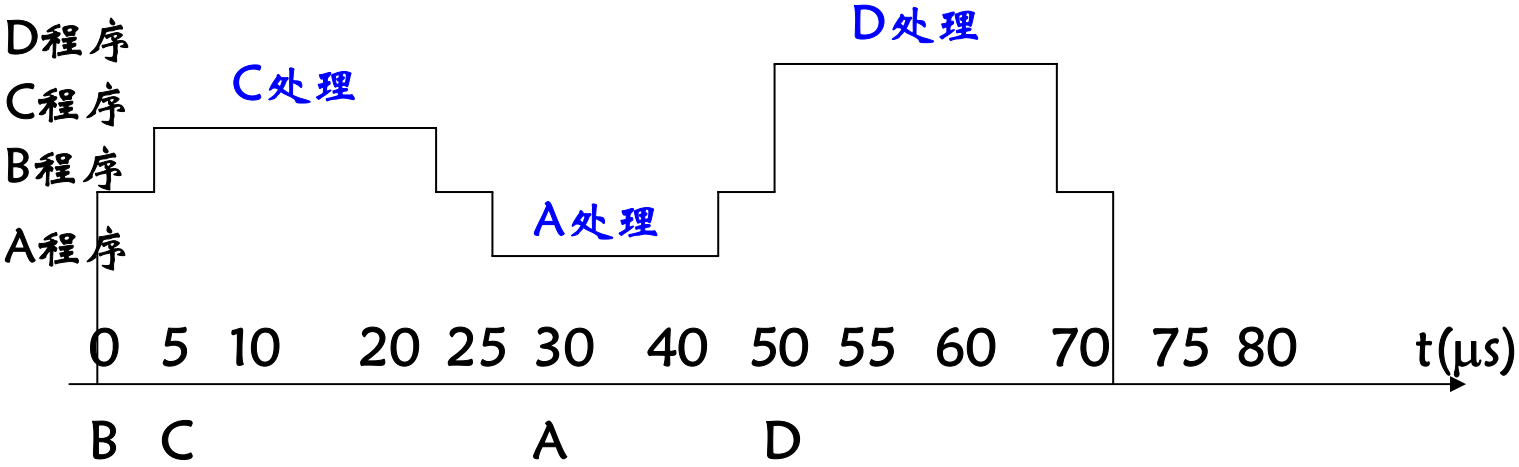


多重中断CPU运行轨迹



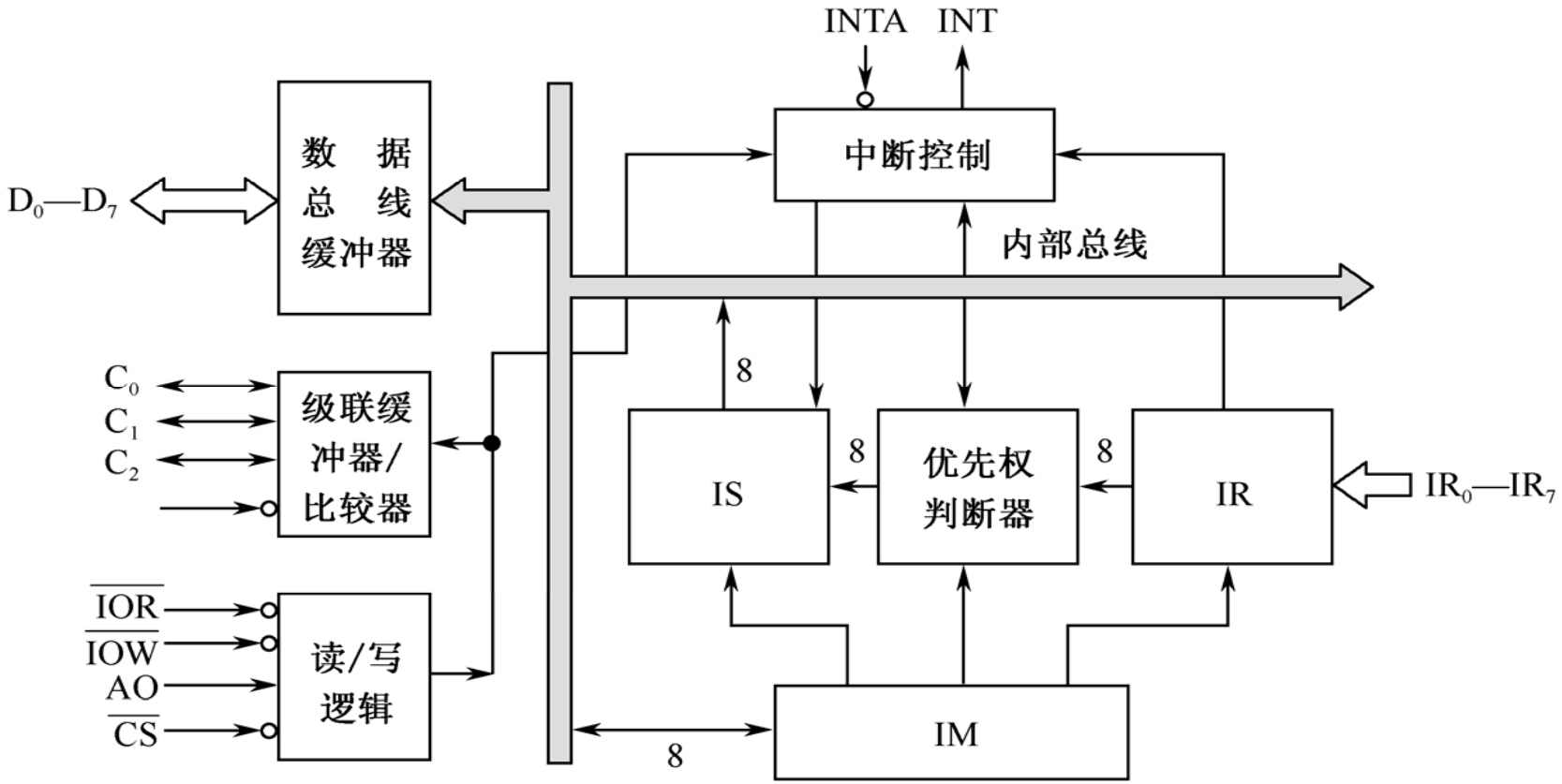
- 例：设某机4级中断A、B、C、D，优先次序为C>A>D>B。请按如下所示时间轴给出的设备中断请求时刻，画出CPU执行程序的轨迹， A、B、C、D中断服务程序的时间宽度均为20μs
- 解：





8.3.5 中断控制器

- 8259中断控制器是一个集成电路芯片，它将中断接口与优先级判断等功能汇集于一身，常用于微型机系统。



8.3.5 中断控制器

- 8259屏蔽方式：
  - ①简单屏蔽方式，提供8位屏蔽字，每位对应着各自的IR线。被置位的任一位则禁止了对应IR线上的中断。
  - ②特殊屏蔽方式，允许CPU让来自低优先级的外设中断请求去中断高优先级的服务程序。例如屏蔽字为11001111，说明IR4和IR5线上的中断请求可中断任何高级别的中断服务程序。
- 8259优先级选择方式：
  - ①完全嵌套方式：是一种固定优先级方式，连至IR存0的设备优先级最高，IR存7的优先级最低。这种固定优先级方式对级别低的中断不利，在有些情况下最低级别的中断请求可能一直不能被处理。
  - ②轮换优先级方式A：每个级别的中断保证有机会被处理，将给定的中断级别处理完后，立即把它放到最低级别的位置上去。
  - ③轮换优先级方式B：要求CPU可在任何时间规定最优优先级，然后顺序地规定其他IR线上的优先级。
  - ④查询方式：由CPU访问8259的中断状态寄存器，一个状态字能表示出正在请求中断的最高优先级IR线，并能表示出中断请求是否有效。

8.3.6 Pentium中断机制

- 中断类型
- 中断源是引起中断的事件及发生中断请求的来源，Pentium中有两种中断源：**中断**和异常。
  - 中断 通常称为外部中断，它是由CPU的**外部硬件**信号引发的。有两种情况 ：
    - 可屏蔽中断：CPU的**INTR**引脚收到中断请求信号，如果CPU中标志寄存器**IF=1**时，可引发中断；**IF=0**时，中断请求信号在CPU内部被禁止。
    - 非屏蔽中断：CPU的**NMI**引脚收到的中断请求信号而引发的中断，这类中断不能被禁止(不管IF=0或IF=1)。

8.3.6 Pentium中断机制

- 中断类型
  - 异常 通常称为异常中断，它是由**指令执行**引发的。有两种情况：
    - 执行异常： CPU执行一条指令过程中出现错误、故障等不正常条件引发的中断；
    - 执行软件中断指令： 如执行INT 0，INT 3，INT n等指令，执行时产生异常中断。
    - **INT 类型号** ；中断调用指令
- 优先级：
  - 当有一个以上的异常或中断发生时，CPU以一个预先确定的优先顺序为它们先后进行服务。
  - 中断优先级分为5级。
    - 异常中断的优先级高于外部中断的级，这是因为异常中断发生在取一条指令或译码一条指令或执行一条指令时出现故障的情况下，情况更为紧急。

8.3.6 Pentium中断机制

- 中断类型
  - Pentium共有256种中断和异常。
  - 每种中断给予一个编号，称为中断向量号(0—255)/类型号；
  - 使用类型号，可以计算一个入口地址/中断向量，入口地址指定的位置保存一个子程序：中断服务子程序。
    - 总共有256个入口地址，指定256个中断服务子程序。

8.3.6 Pentium中断机制

- 中断服务子程序进入过程
  - 中断发生的时候，要执行的代码就是中断服务子程序。
  - 类型号/中断向量号：每个中断对应的1个8位编码；
  - 中断向量：是一个地址/指针，指定的位置保存有1个中断服务子程序。
  - 中断向量表IVT/中断描述符表IDT：256个中断向量组成的表。
    - 实模式中是IVT表，保护模式中是IDT表。
    - IVT表每个表项32位；IDT表每个表项64位；
  - 中断服务子程序入口地址：即中断向量，指向中断服务子程序的第1条指令。
  - 中断发生时,CPU将：
    - 获取类型号n→在IVT/IDT表中获取n号中断向量→跳转执行n号中断服务子程序。

8.3.6 Pentium中断机制

- 获取中断向量号/类型号
  - **指令给出**：如软件中断指令INT n 中的n即为中断向量号。
  - **外部提供**：可屏蔽中断是在CPU接收到INTR信号时产生一个中断识别周期，接收外部中断控制器由数据总线送来的中断向量号；非屏蔽中断是在接收到NMI信号时中断向量号固定为2。
  - **固定类型号**：CPU识别错误、故障现象，根据异常和中断产生的条件自动指定向量号。**除数=0，类型号0**。

8.3.6 Pentium中断机制

- 获取中断向量/入口地址
  - **实模式**：中断向量保存在IVT表。IVT表保存在内存地址0开始的1KB空间中。
    - IVT表中每个表项是1个中断向量，占4个字节，共32位。
  - 根据**段寻址**规定：
    - 类型号n的中断向量保存在起始地址=(0：n×4)指定的连续4个内存单元中。
    - 地址小的两个位置提供入口偏移地址，地址大的两个位置提供如何段基址。

8.3.6 Pentium中断机制

- 获取中断向量/入口地址
  - **保护模式**：中断向量保存在IDT表。IDT表在内存地址中保存起始位置由IDTR/**中断描述符表寄存器** 提供。
    - IDT表中每个表项是一个中断描述符，每个描述符占8个字节，64位，IDT表共2KB。
  - 根据**保护模式寻址**规定：
    - 类型号n的中断向量保存在起始地址=((IDTG)：n×8)指定的连续8个内存单元中。

8.3.6 Pentium中断机制

- 中断处理过程

(1) 当中断处理的CPU控制权转移涉及到特权级改变时，必须把当前的SS和ESP两个寄存器的内容压入系统堆栈予以保存。

(2) 标志寄存器EFLAGS的内容也压入堆栈。/保护现场

(3) 清除标志触发器TF和IF。

(4) 当前的代码段寄存器CS和指令指针EIP也压入此堆栈。 /保护断点

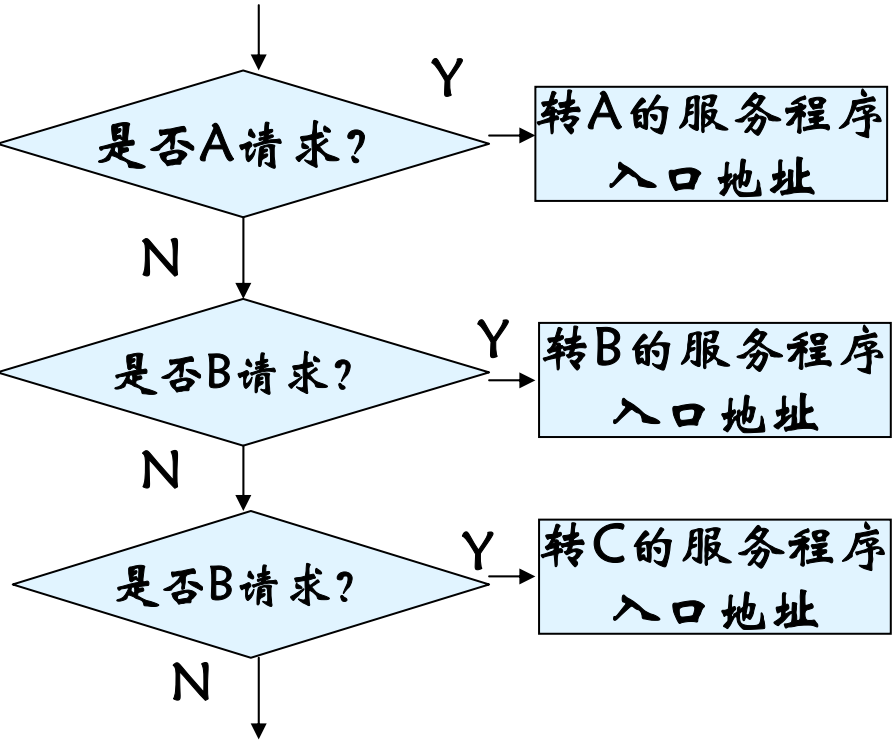
(5) 如果中断发生伴随有错误码，则错误码也压入此堆栈。

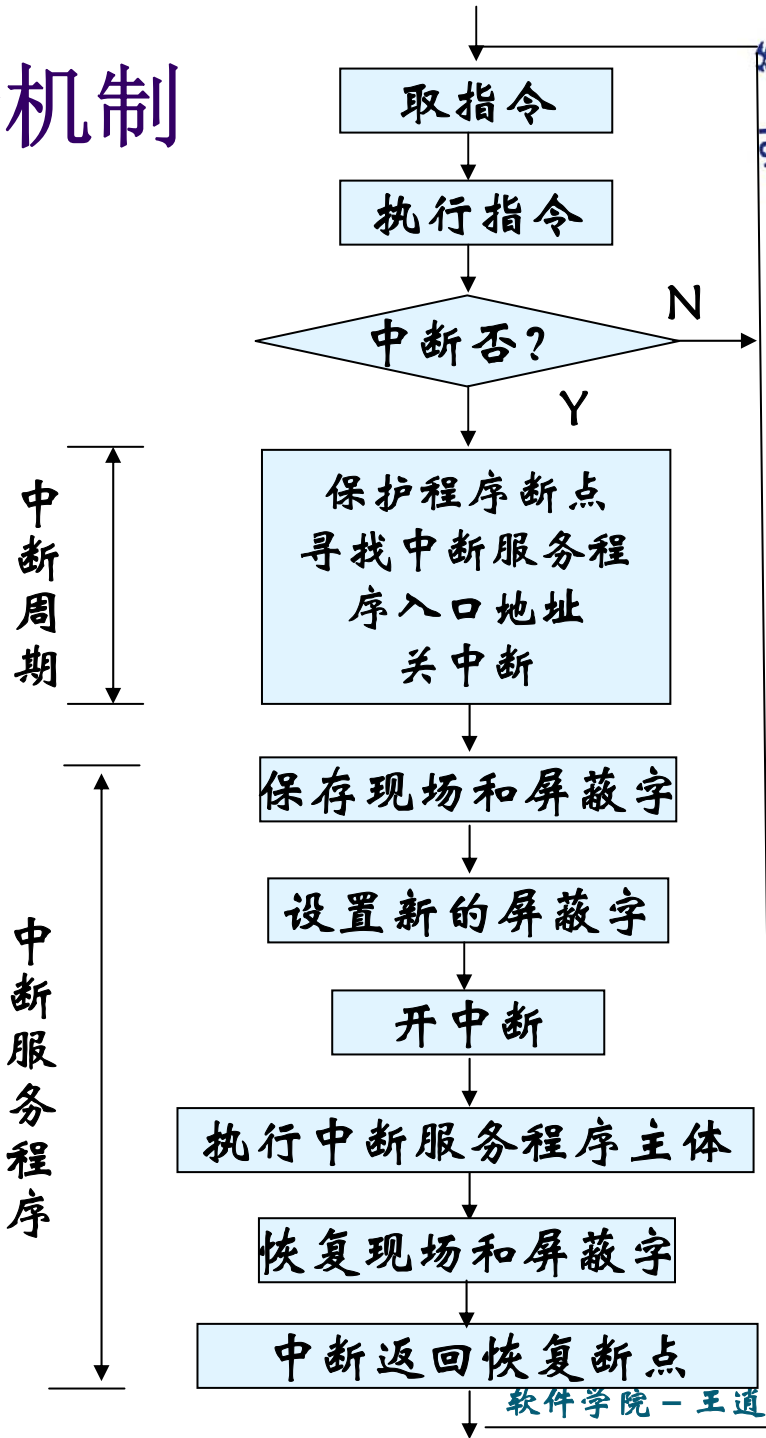
(6) 完成上述中断现场保护后，从中断向量号获取的中断服务子程序入口地址(段，偏移)分别装入CS和EIP，开始执行中断服务子程序。

(7) 中断服务子程序最后的IRET指令使中断返回。保存在堆栈中的中断现场信息被恢复，并由断点继续执行原程序。

- ,

8.3.6 Pentium中断机制





中断方式和程序查询方式的比较

- 相同点：
  - 数据的输入输出都经过CPU，适用低速设备
- 不同点：

(1) 在程序查询中，何时对何设备进行操作受CPU控制；在I/O中断中，由外围设备主动通知CPU

(2) 程序查询方式中，CPU与外围设备不能并行工作；但中断方式中，它们可以并行操作

(3) 程序查询方式无法处理异常事件，中断I/O方式可以

(4) 程序查询方式硬件结构较简单，CPU效率低且只能进行数据传送，中断方式硬件结构相对复杂一些

8.4 DMA方式

- 在中断方式下，CPU在处理中断服务程序时需暂停原程序的正常运行,当高速I/O设备频繁地、成批地与主存交换信息时， CPU需不断地暂停执行主程序而执行中断服务程序。为了提高CPU的效率，提出了MDA方式。
- 直接内存访问方式——完全由硬件执行I/O交换的工作方式
  - 在数据传送时，DMA控制器从CPU完全接管对总线的控制，数据交换不经过CPU，而直接在内存和I/O设备之间进行。
  - DMA方式一般用于高速传送成组数据。DMA控制器将向内存发出地址和控制信号，修改地址，对传送的字的个数计数，并且以中断方式向CPU报告传送操作的结束。
  - 最大优点是数据交换的速度高，适用于高速成组传送数据

8.4.1 DMA方式的基本概念

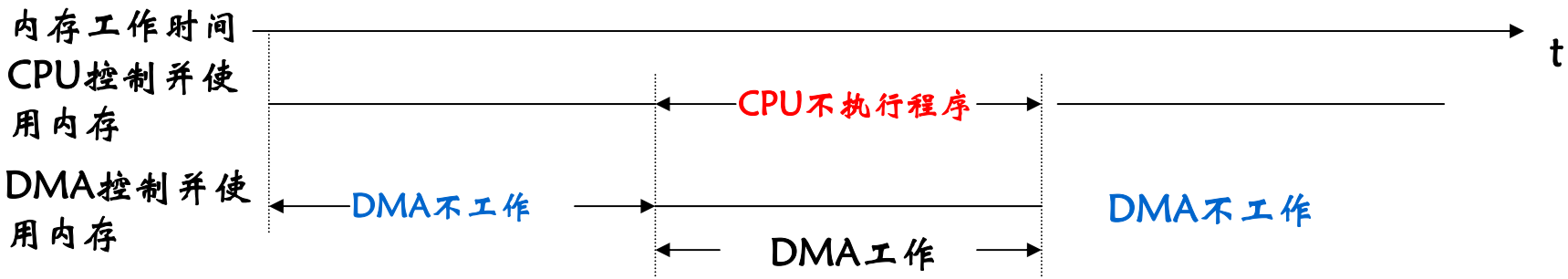
- **DMA控制器能执行的基本操作包括：**
  - 向CPU发出DMA请求；
  - CPU响应请求，把CPU工作改成DMA操作方式，DMA控制器从CPU接管总线的控制；
  - 由DMA控制器对内存寻址，即决定数据传送的内存单元地址及数据传送个数的计数，并执行数据传送的操作；
  - 向CPU报告DMA操作的结束。
  - 说明：
    - 在DMA方式中，一批数据传送前的准备工作，以及传送结束后的处理工作，均由管理程序承担，而DMA控制器仅负责数据传送的工作。

8.4.2 DMA传送方式

- 在DMA方式中，DMA控制器直接访问内存，与此同时，CPU可以继续执行程序。DMA接口与CPU共享主存，有可能出现两者争用主存的冲突。
- 为了有效地分时使用主存，通常DMA与主存交换数据时采用采用以下3种方法：
  - 停止CPU访问内存；
  - 周期挪用；
  - DMA与CPU交替访问；
  - 说明：
    - 若采用多总线结构、双端口存储器等，即可避免这种情况

8.4.2 DMA传送方式

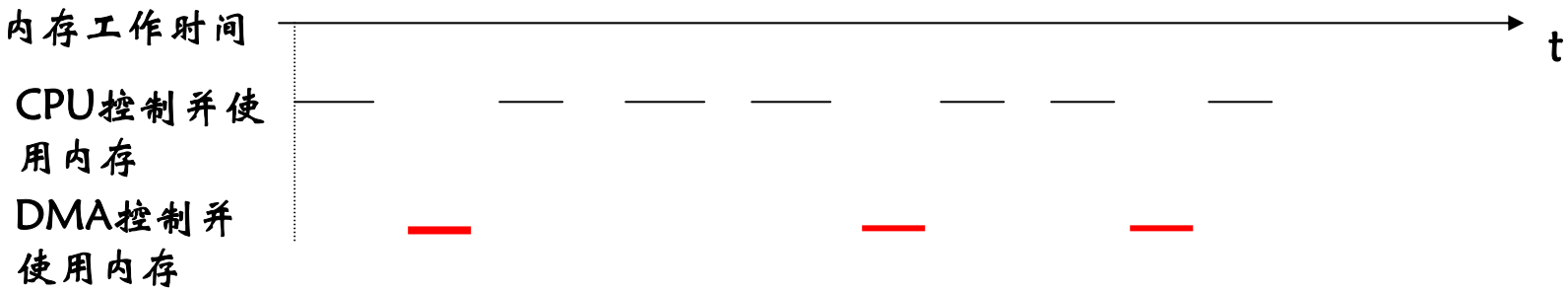
- 停止CPU访问内存
  - 当外围设备要求传送一批数据时，由DMA控制器发一个停止信号给CPU，要求CPU放弃对地址总线、数据总线和有关控制总线的使用权。
  - DMA控制器获得总线控制权以后，开始进行数据传送。
  - 在一批数据传送完毕后，DMA控制器通知CPU可以使用内存，并把总线控制权交还给CPU。
- 控制简单，适用于高速设备进行成组传送
- DMA传送过程中，主存的存取速度高于外设的工作速度，主存的效能没有充分发挥。



8.4.2 DMA传送方式

- 周期挪用
  - 设备没有DMA请求时，CPU按程序要求访问内存；设备有DMA请求，由I/O设备挪用一个或几个内存周期；
  - 与停止CPU访内相比，既实现了I/O传送，又较好地发挥了内存和CPU的效率。

周期挪用的时间示意图



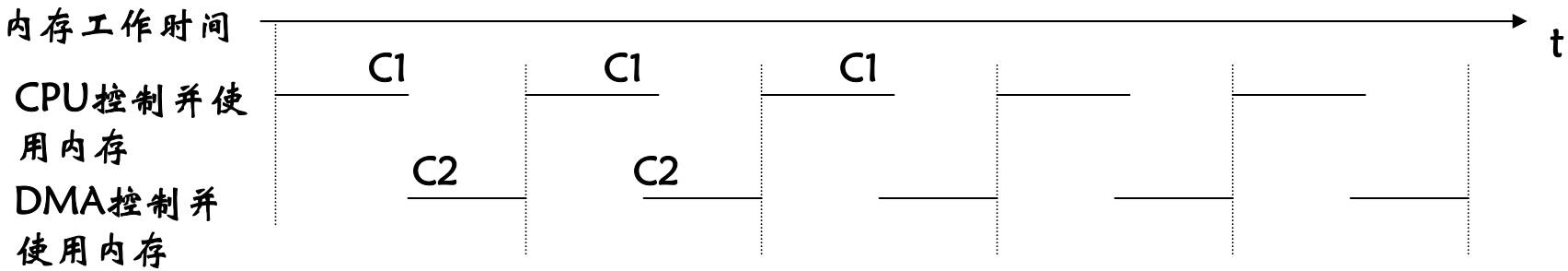
8.4.2 DMA传送方式

- 周期挪用
  - 两种访内情况：
    - 访内正常：在挪用周期中CPU执行程序没有访内，则DMA可以正常访内；
    - 访内冲突：在挪用周期中，CPU与DMA同时访内，冲突发生；
      - 规定**DMA访内优先**。
  - I/O设备每一次挪用都有申请、建立和归还总线控制权的过程，传送一个字对内存来说要占用一个周期，但对DMA控制器来说需要多个内存周期，适用于I/O设备读写周期大于内存存储周期的情况。

8.4.2 DMA传送方式

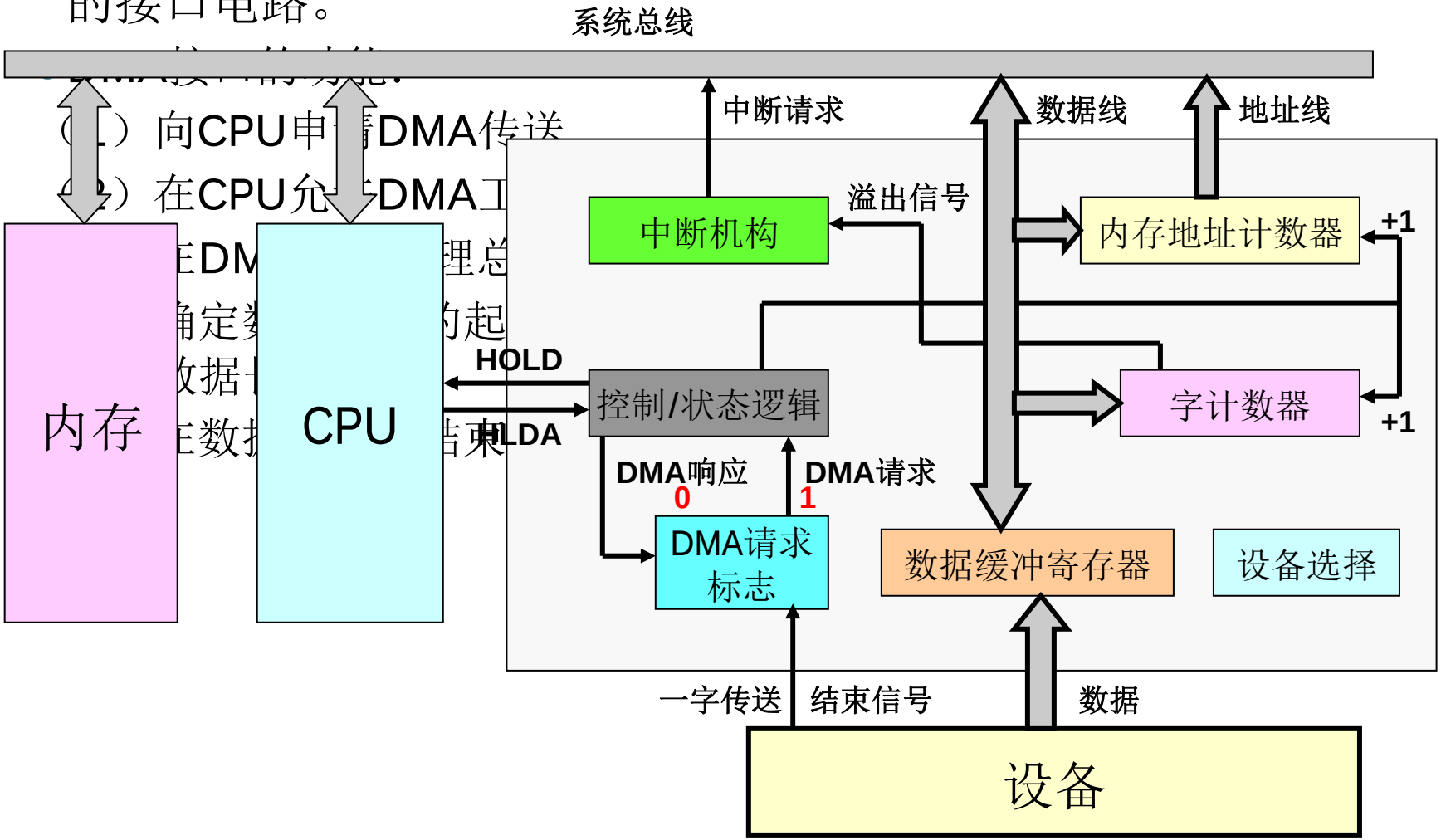
- **DMA与CPU交替访内 / 透明的DMA方式**
  - CPU工作周期比主存存取周期长，将CPU工作周期一分为二，一半DMA访内，一半CPU访内。
  - 不需总线使用权的申请、建立和归还过程，只需要总线控制权转移；
  - 总线使用权分时控制，硬件逻辑更加复杂

DMA与CPU交替访问时间示意图



8.4.3 基本的DMA控制器

- **DMA**控制器的基本组成
  - DMA控制器实际上是采用DMA方式的外围设备与系统总线之间的接口电路。



8.4.3 基本的DMA控制器

- DMA控制器的基本组成
  - 内存地址计数器
    - 用于存放内存中要交换的数据的地址，DMA传送前，用程序将数据在内存中的起始位置送内存地址计数器。
    - DMA传送时，每交换一次数据，地址计数器加“1”。
  - 字计数器
    - 用于记录传送数据块的长度(多少字数)。其内容也是在数据传送之前由程序预置，交换的字数通常以补码形式表示。
    - 在DMA传送时，每传送一个字，字计数器就加“1”，当计数器溢出即最高位产生进位时，表示这批数据传送完毕，于是引起DMA控制器向CPU发中断信号。
  - 数据缓冲寄存器
    - 用于暂存每次传送的数据(一个字)。
    - 当输入时，由设备(如磁盘)送往数据缓冲寄存器，再由缓冲寄存器通过数据总线送到内存。
    - 输出时，由内存通过数据总线送到数据缓冲寄存器，然后再送到设备。

8.4.3 基本的DMA控制器

- DMA控制器的基本组成
  - DMA请求标志
    - 设备准备好→ “DMA请求” 标志置“1” →发DMA请求→发总线使用权请求(HOLD) → 发响应信号HLDA → 发DMA响应信号→复位“DMA请求”标志。
  - 控制 / 状态逻辑
    - 由控制和时序电路以及状态标志等组成，用于修改内存地址计数器和字计数器，指定传送类型(输入或输出)，并对“DMA请求”信号和CPU响应信号进行协调和同步。
  - 中断机构
    - 一组数据交换完毕，由溢出信号触发中断机构，向CPU提出中断请求报告一组数据传送结束。

8.4.3 基本的DMA控制器

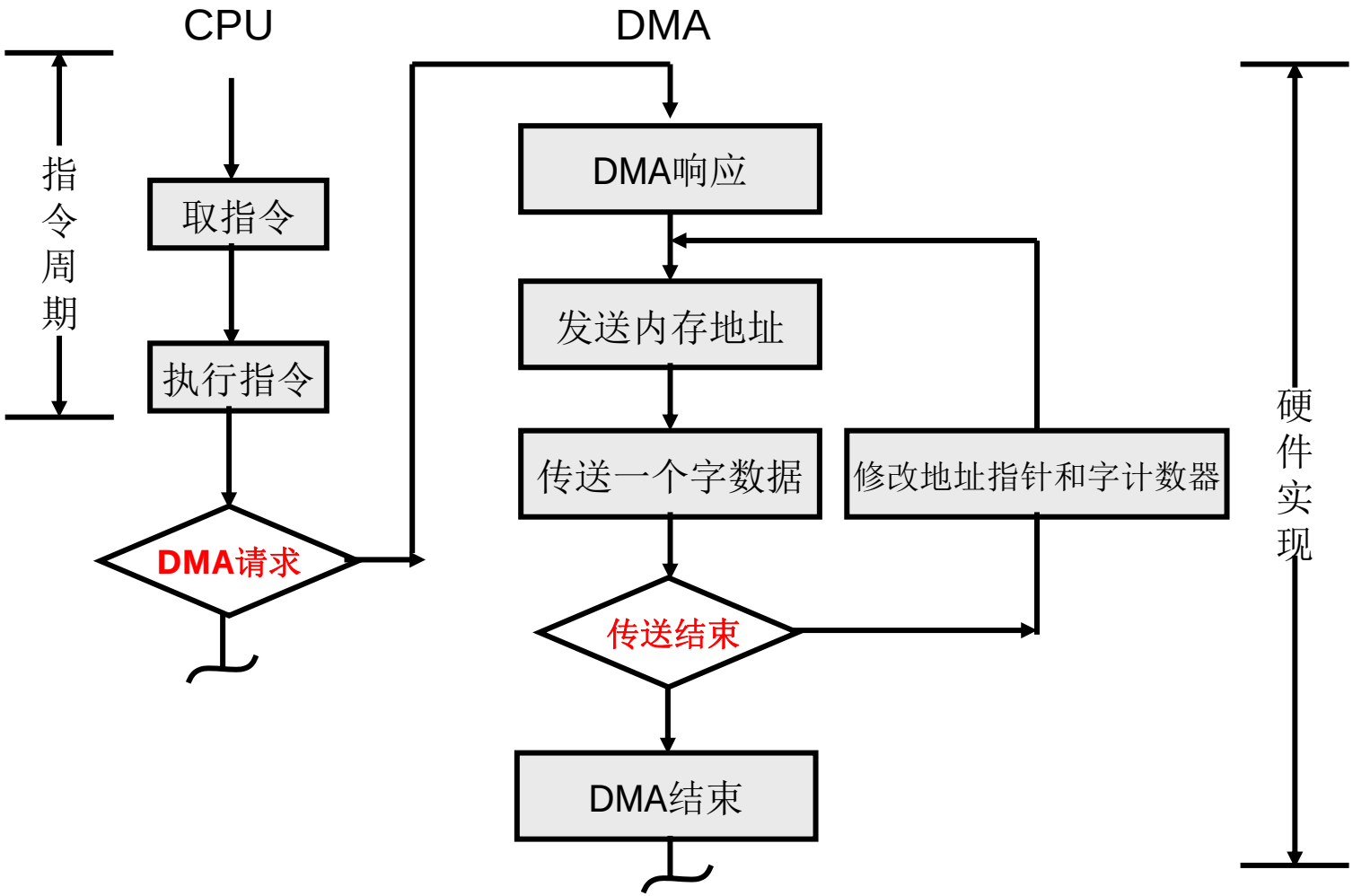
- DMA数据传送过程
  - 分3个阶段：传送前预处理，正式传送，传送后处理。
  - 传送前预处理
    - 由CPU执行几条输入输出指令，测试设备状态，向DMA控制器的设备地址寄存器中送入设备号并启动设备，向内存地址计数器中送入起始地址，向字计数器中送入交换的数据字个数。在这些工作完成后，CPU继续执行原来的主程序。
  - 正式传送
    - 当外设准备好发送数据(输入)或接受数据(输出)时，它发出DMA请求，由DMA控制器向CPU发出总线使用权的请求(HOLD)。
    - DMA的数据传送是以数据块为基本单位进行的，因此，每次DMA控制器占用总线后，无论是数据输入操作，还是输出操作，都是通过循环来实现的。当进行输入操作时，外围设备的数据(一次一个字或一个字节)传向内存；当进行输出操作时，内存的数据传向外围设备。

8.4.3 基本的DMA控制器

- DMA数据传送过程
  - 传送后处理
    - 一旦DMA的中断请求得到响应，CPU停止主程序的执行，转去执行中断服务程序做一些DMA的结束处理工作。这些工作包括校验送入内存的数据是否正确；决定继续用DMA方式传送下去，还是结束传送；测试在传送过程中是否发生了错误等等。
  - DMA控制器仅负责数据块的传送控制。

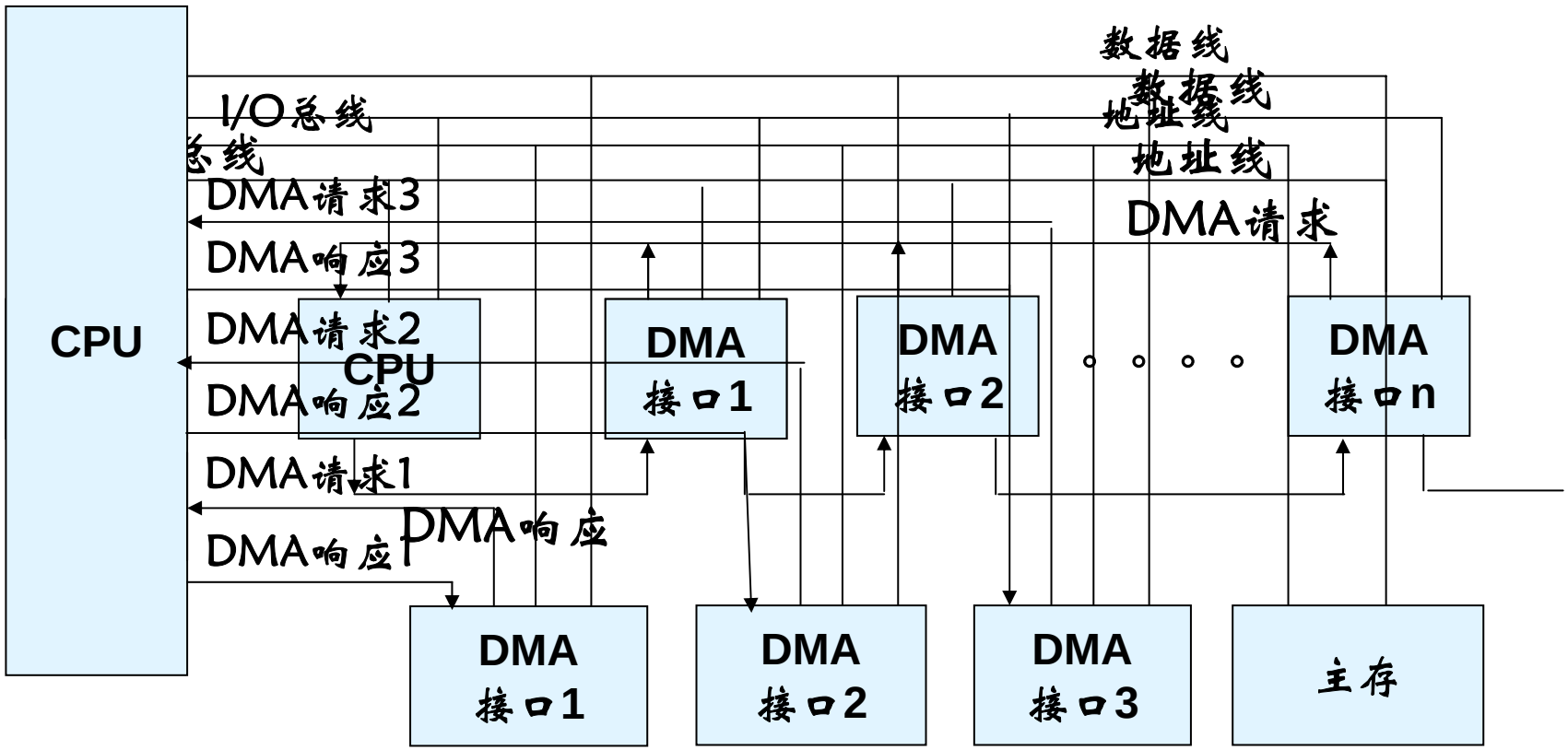
8.4.3 基本的DMA控制器

- DMA传送数据流程图
  - 停止CPU访内的DMA流程图



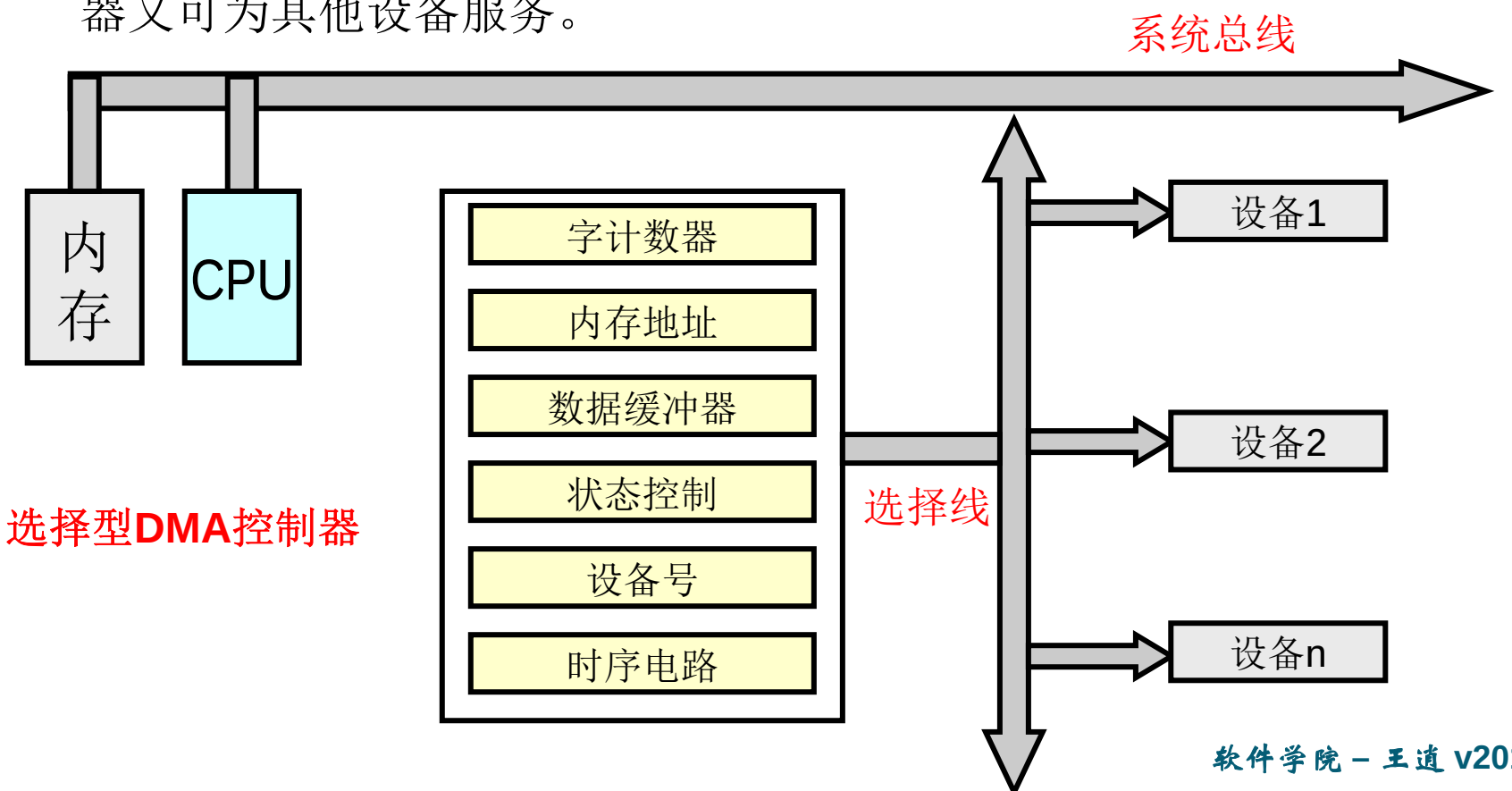
8.4.3 基本的DMA控制器

- 基本DMA控制器与系统的连接方式：
  - 公用的DMA请求方式：同一条DMA请求线。
  - 独立的DMA请求方式，这与中断方式类似：独立的DMA请求线。



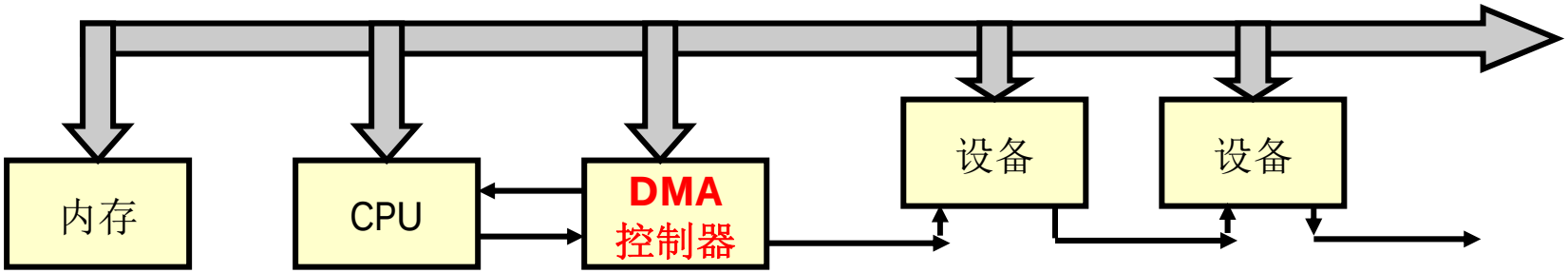
8.4.4 选择型和多路型DMA控制器

- **DMA控制器连接多个DMA设备**
  - 物理上可以连接多个设备，逻辑上只允许连接一个设备。在某一段时间内只能为一个设备服务，预处理时将所选设备的设备号送人设备地址寄存器。
  - 适用于数据传输率很高的设备，在很快地传送完一个数据块后，控制器又可为其他设备服务。

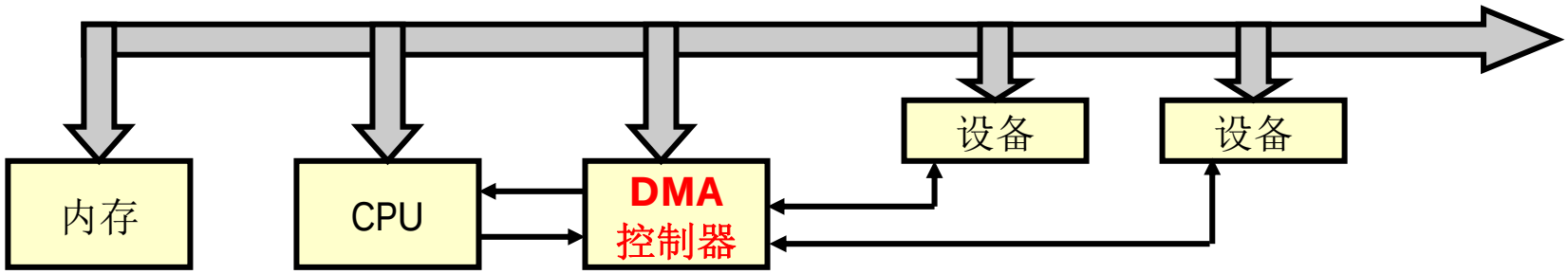


8.4.4 选择型和多路型DMA控制器

- **DMA控制器连接多个DMA设备**
  - 在物理上可以连接多个外围设备，逻辑上也允许这些外围设备同时工作。各设备以字节交叉方式通过DMA控制器进行数据传送，适合于同时为多个传输率不太高的外围设备服务

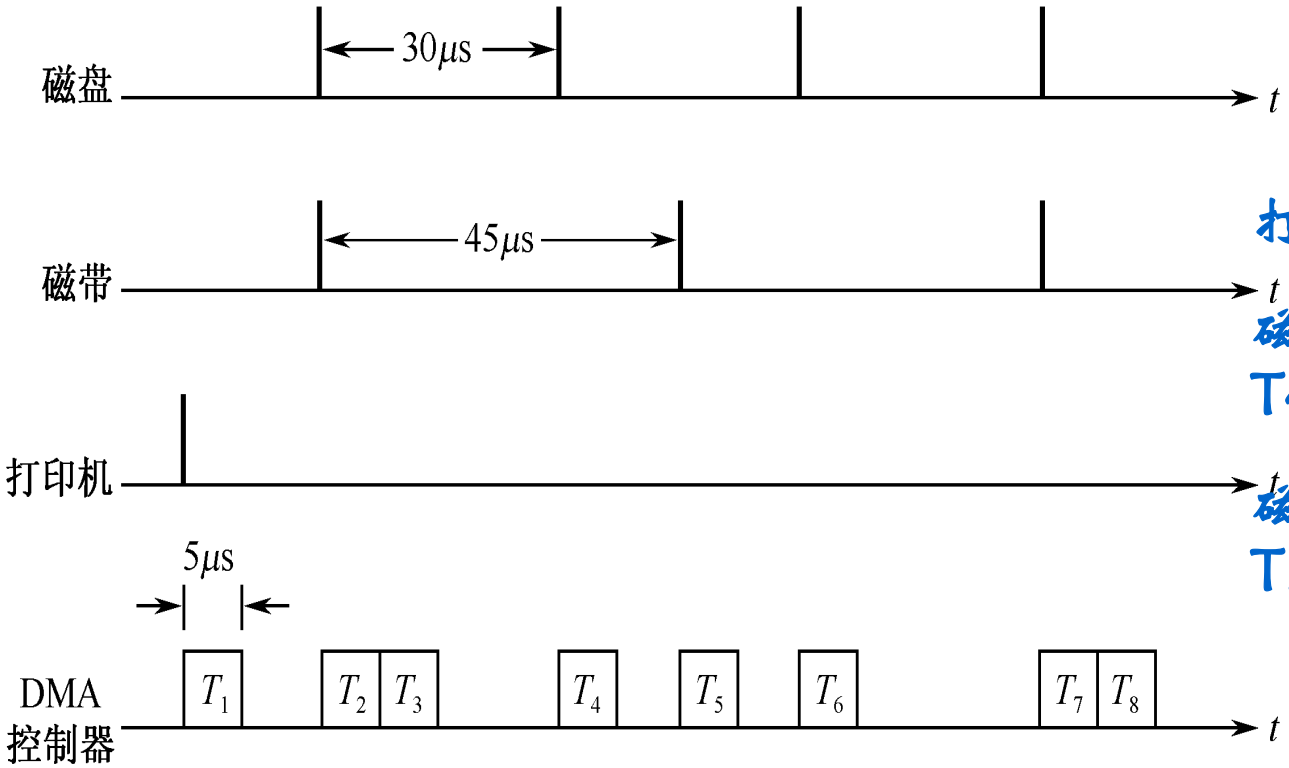


链式多路型DMA



独立请求多路型DMA

- 例：以DMA方式实现打印机，磁盘，磁带与主机信息交换。
  - 磁盘每30us发一次DMA请求。磁带每45us发一次DMA请求。打印机每150us发一次DMA请求。
  - 磁盘优先级最高，打印机最低。
  - 完成一次DMA传送的时间是5us。
  - 画出DMA服务3个设备的工作时间图。



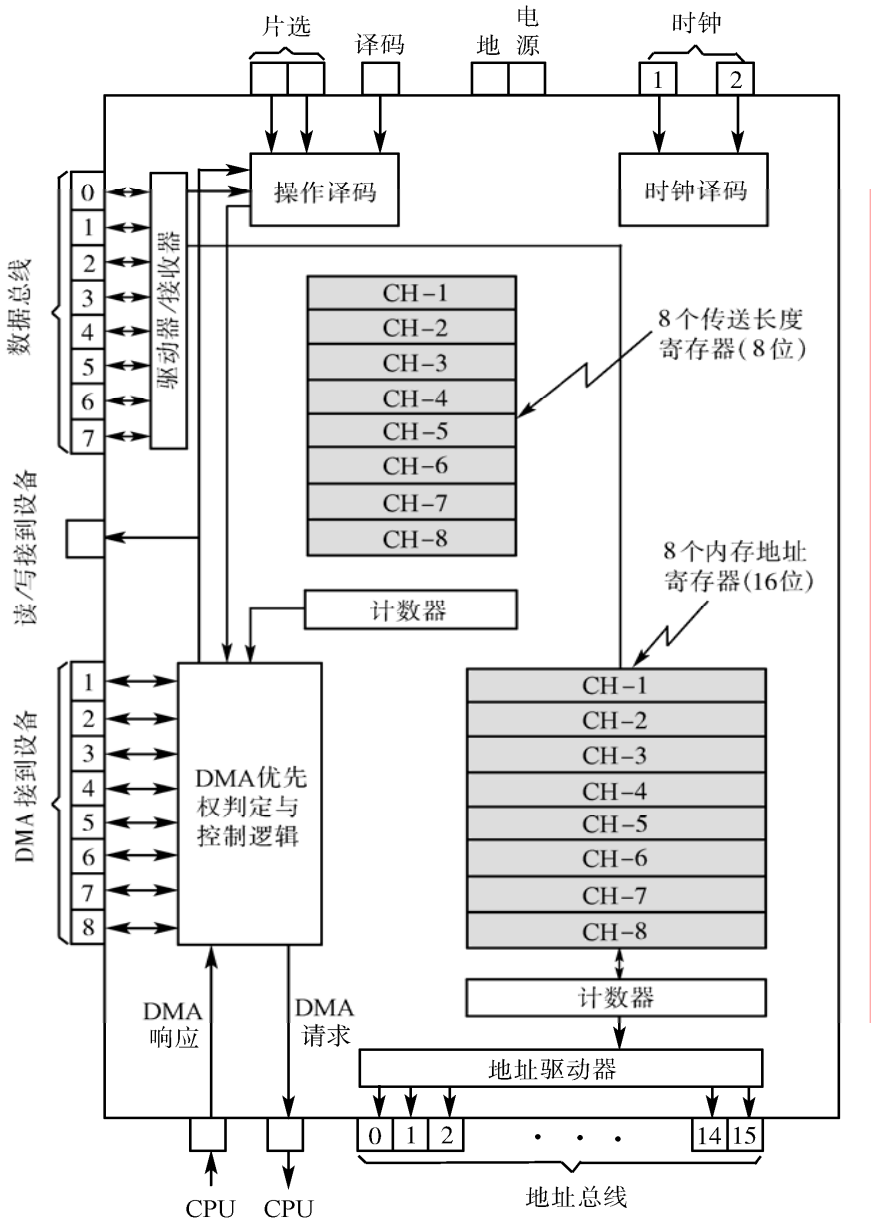
打印机服务一次(T1)

磁盘服务四次(T2，T4，T6，T7)

磁带服务三次(T3，T5，T8)

8.4.4 选择型和多路型DMA控制器

- DMA芯片



- 例：一个DMA接口可采用周期挪用方式传送字符的最大批量为400B。若存取周期为100ns，每处理一次中断需5us，字符设备的传输率是9600bps字符间的传输无间隙，忽略预处理时间。采用DMA方式每秒因数据传输需占处理器多少时间？完全采用中断方式，又占处理器多少时间？

每秒传送9600bps/8=1200B

每秒中断次数=1200B/400B=3次

1200个字符需1200个存取周期

DMA方式每秒占用处理器的时间

=1200×0.1us+3×5us=135us

完全采用中断方式占用处理器的时间

=1200×5us=6000us

- 例：磁盘采用DMA方式与主机交换信息的传输率为2MBps。DMA的预处理需1000个时钟周期，传送后处理中断需500个时钟周期，平均传输的数据长度为4KB。在硬盘工作时，50MHz的处理器需用多少时间比率进行DMA辅助操作（预处理和后处理）？

解：每秒传输次数=2MBp/4KB=512次

512次传送需DMA辅助操作的时钟周期数

=512×(1000+500)=512×1500个时钟周期

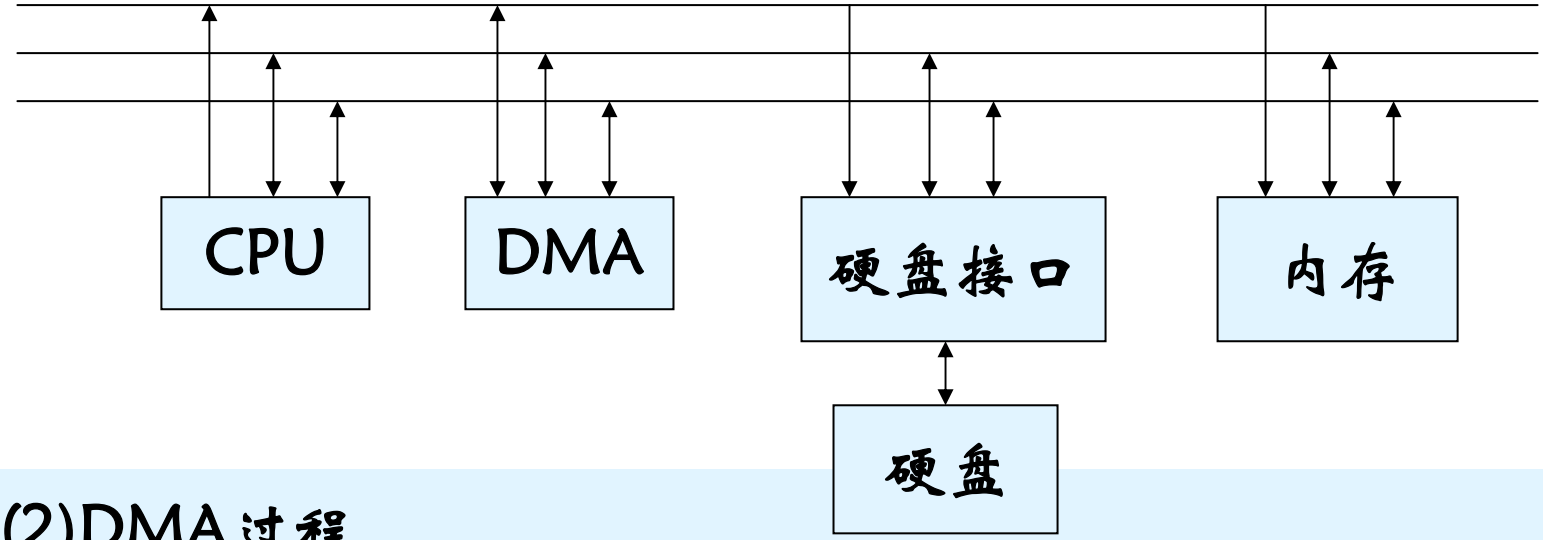
=512×1500/(50×10^6)s=1.536×10^{-2}s

DMA辅助操作占用CPU的时间比率为

(1.536×10^{-2}/1)*100%=1.5%

- 例：已知一些模块：CPU、DMA控制器、硬盘、硬盘接口、内存。
  - (1)请将这几个模块连成一个单总线结构计算机，分别画出3类总线
  - (2)该计算机用DMA方式将硬盘上的多块数据读入内存。CPU通过中断初始化DMA控制器，DMA每次初始化只能读一块数据。请简要说明这一过程。
  - (3)为什么要用程序中断方式进行后处理？

- 解：



(2)DMA过程

DMA操作可以分3个主要步骤——预处理、数据传送和后处理

因为每次只能传送一块数据，所以要反复重复以上三个步骤，直到多块数据都被传送完。

(3) DMA的数据传送过程完全由DMA控制器控制完成，CPU并不知道什么时候结束数据传送，所以当DMA控制器完成数据传送过程后必须主动通知CPU，CPU随即进行后处理。

8.5 通道方式

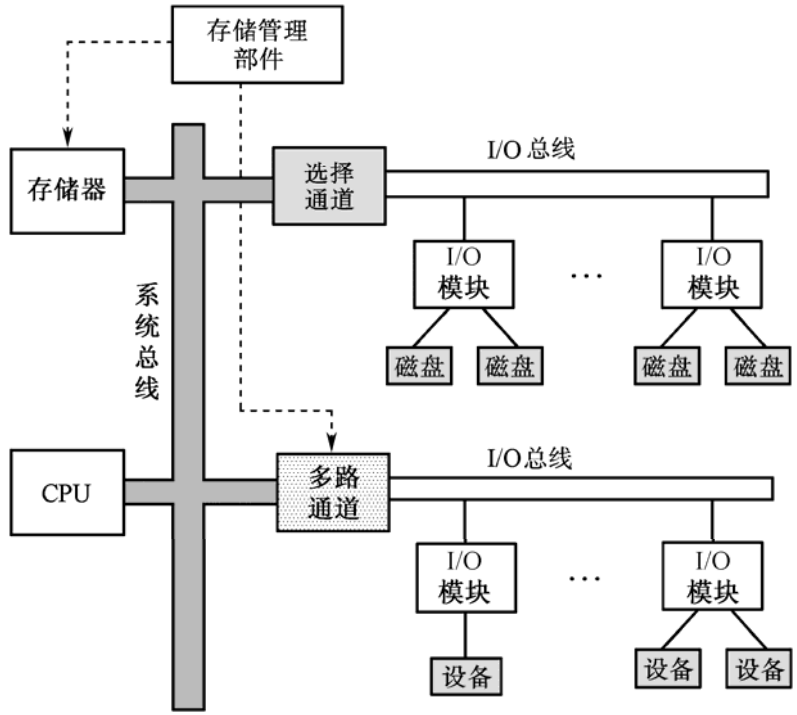
- 通道的基本概念
  - 通道”是计算机系统中代替CPU管理控制外设的独立部件，是一种能执行有限I/O指令集合—通道命令的I/O处理机。
  - 在通道控制方式中，一个主机可以连接几个通道。每个通道又可连接多台I/O设备，这些设备可具有不同速度，可以是不同种类。
  - **系统总线：**

    通道与存储器、

    CPU与存储器之间数据传输
  - **I/O总线/通道总线：**

    外围设备与通道之间数据传送

采用通道方式组织输入输出系统，多使用主机/主存—通道—I/O模块—I/O设备四级连接方式。



8.5.1 通道的功能

- 通道数据传送过程
  - 在CPU启动通道后，通道自动地去内存取出通道指令并执行指令。直到数据交换过程结束向CPU发出中断请求，进行通道结束处理工作。
  - 主要过程分为如下三步进行：

    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，完成指定的数据输入输出工作。

    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，调用管理程序对输入输出中断请求进行处理。

8.5.1 通道的功能

- 通道的基本功能
  - 执行通道指令，组织外围设备和内存进行数据传输，按I/O指令要求启动外围设备，向CPU报告中断等，具体有以下五项任务：

(1)接受CPU的I/O指令，按指令要求与指定的外围设备进行通信。

(2)从内存选取属于该通道程序的通道指令，经译码后向设备控制器和设备发送各种命令。

(3)组织外围设备和内存之间进行数据传送，并根据需要提供数据缓存的空间，以及提供数据存入内存的地址和传送的数据量。

(4)从外围设备得到设备的状态信息，形成并保存通道本身的状态信息，根据要求将这些状态信息送到内存的指定单元，供CPU使用。

(5)将外围设备的中断请求和通道本身的中断请求，按次序及时报告CPU。

8.5.1 通道的功能

采用通道方式组织输入输出系统，多使用主机/主存—通道—I/O模块—I/O设备四级连接方式。

- **CPU对通道的管理**
  - 执行用户程序中的I/O指令
  - 处理来自通道的中断
    - 传送结束中断与故障中断
  - 管态：CPU执行操作系统的管理程序的状态；
  - 目态：CPU执行目的程序的状态；
    - I/O指令是管态指令，要执行I/O指令，CPU要处于管态。目态不能执行I/O指令。
- **通道对I/O模块的管理**
  - 使用通道指令控制I/O模块进行数据传送操作，并以通道状态字接收I/O模块反映的I/O设备的工作状态。
  - I/O模块是通道对I/O设备实现传送控制的执行机构：
  - 从通道接受通道指令，控制I/O设备的操作；
  - 以通道状态字向通道反映I/O设备的状态；
  - 进行信号类型转换；

8.5.2 通道的类型

- 根据工作方式分：选择通道和多路通道。
- 选择通道 / 高速通道
  - 选择通道每次只能从所连接的设备中选择一台I/O设备的通道程序，此刻该通道程序独占了整个通道。连接在选择通道上的若干设备，只能依次使用通道与主存传送数据。
  - 数据传送以成组（数据块）方式进行，每次传送一个数据块，因此，传送速率很高。
  - 选择通道多适合于快速设备（磁盘），这些设备相邻字之间的传送空闲时间极短。

8.5.2 通道的类型

- 多路通道分 / 多路转换通道：数组多路通道和字节多路通道。
- 数组多路通道（Block Multiplexor Channel）
  - 数组多路通道把字节多路通道和选择通道的特点结合起来。它有多个子通道，既可以执行多路通道程序，象字节多路通道那样，所有子通道分时共享总通道；又可以用选择通道那样的方式传送数据。

8.5.2 通道的类型

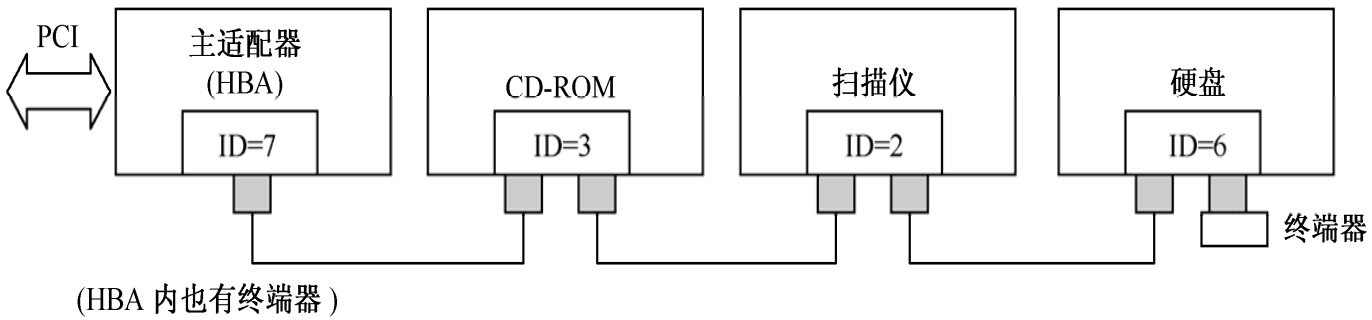
- 多路通道分 / 多路转换通道：数组多路通道和字节多路通道。
- 字节多路通道（ Byte Multiplexor Channel）
  - 是一种简单的共享通道，在时间分割的基础上，服务于多台低速和中速面向字符的外围设备。
  - 字节多路通道包括多个子通道，每个子通道服务于一个设备控制器，可以独立地执行通道指令。每个子通道都需要有字符缓冲寄存器、I/O请求标志 / 控制寄存器、主存地址寄存器和字节计数寄存器。而所有于通道的控制部分是公共的，由所有子通道所共享。通常，每个通道的有关指令和参量存放在主存固定单元中。当通道在逻辑上与某一设备连通时，将这些指令和参量取出来，送入公共控制部分的寄存器中使用。
  - 字节多路通道要求每种设备分时占用一个很短的时间片，不同的设备在各自分得的时间片内与通道建立传输连接，实现数据的传送。

8.5.3 通道结构的发展

- 输入输出处理机（IOP）
  - 输入输出处理机（IOP）不是一台独立的计算机，而是计算机系统中的一个部件。IOP可以和CPU并行工作，提供高速的DMA处理能力，实现数据的高速传送。此外，有些IOP还提供数据的变换、搜索和字装配／分拆能力。
  - 8位和16位微机中使用的Intel 8089 I/O处理器就是这种通道型I/O处理器
- 外围处理机
  - 外围处理机结构更接近于一般处理机，或者就是选用已有的通用机。外围机基本上是独立于主处理机工作的，应用于大型高效率的计算机系统中。

8.6 通用I/O接口标准

- 并行I/O接口SCSI
  - 小型计算机系统接口的简称，它是一个高速智能接口，可以混接各种磁盘、光盘、磁 带机、打印机、扫描仪、条码阅读器以及通信设备。
  - SCSI是系统级接口，是处于主适配器和智能设备控制器之间的并行I/O接口，改进的SCSI可允许连接1～15台不同类型的高速外围设备。SCSI的不足处在于硬件较昂贵，并需要通用设备驱动程序和各类设备的驱动程序模块的支持。



8.6 通用I/O接口标准

- 串行接口标准IEEE1394
  - IEEE 1394是一种高速串行I/O标准接口。各被连接装置的关系是平等的，不用PC介入也能自成系统。这意味着1394在家电等消费类设备的连接应用方面有很好的前景。
  - 与SCSI并行I/O接口相比，它具有更高的数据传输速率和数据传送的实时性，具有更小的体积和连接的方便性。IEEE 1394的一个重大特点是，各被连接的设备的关系是平等的，不用PC介入也能自成系统。因此IEEE 1394已成为Intel、Microsoft等公司联手制定的新标准。