

8 输入输出系统

提纲

8.1

外围设备的速度分级与信息交换方式

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

8.1 外围设备的速度 分级与信息交换方式

提纲

8.1.1

外围设备的速度分级

8.1.2

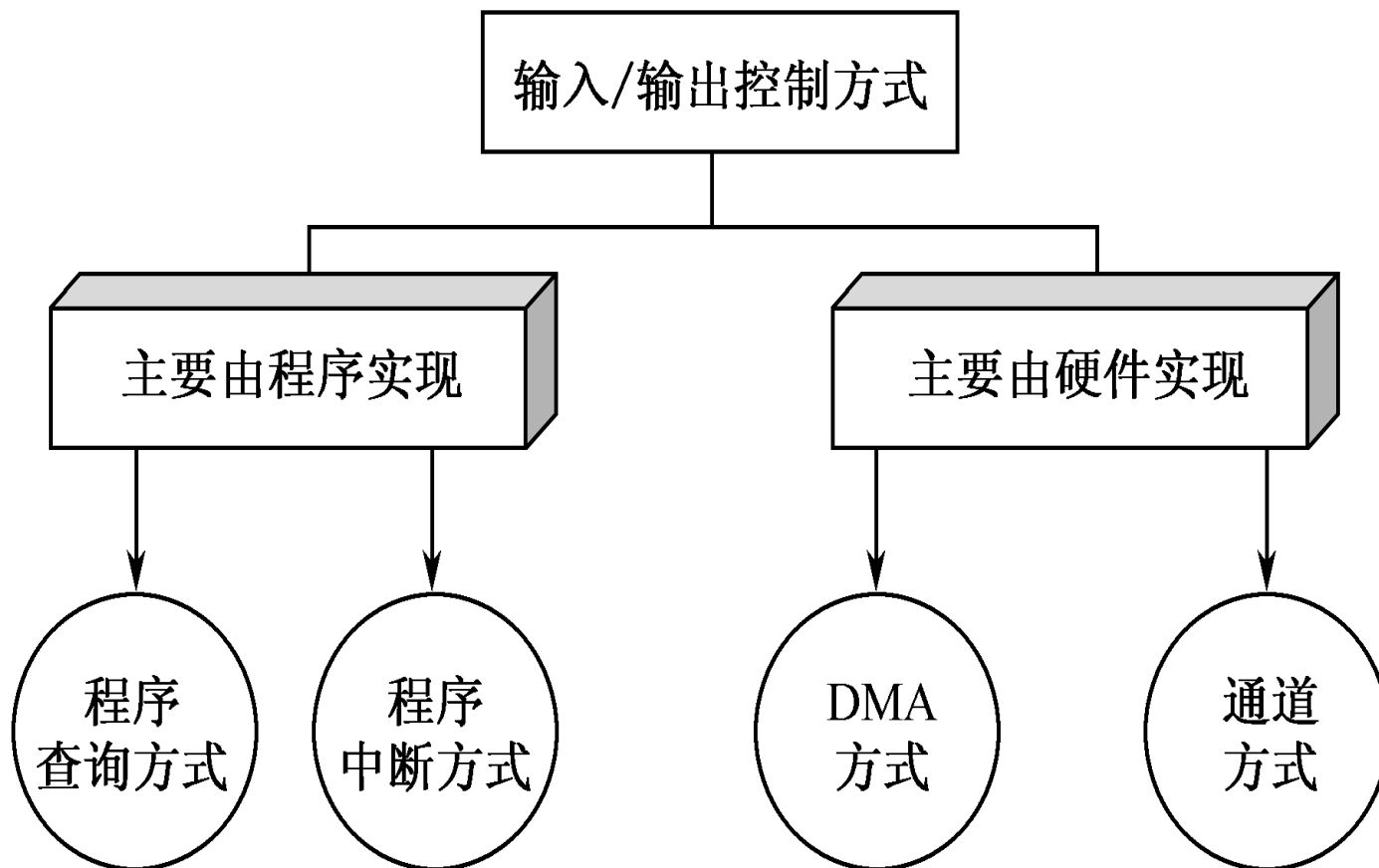
信息交换方式



8.1.1 外围设备的速度分级

- 外设种类繁多，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
- 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
 - 慢速或者中速的设备
 - 可以采用异步定时的方式（应答式数据交换）
 - 高速外设
 - 采用同步定时方式

8.1.2 信息交换方式





8.1.2 信息交换方式

■ 程序查询方式

- 程序查询方式又叫程序控制I/O方式
- 数据在CPU和外围设备之间的传送靠计算机程序控制，是在CPU主动控制下进行的
- 当输入/输出时，CPU暂停执行主程序，转去执行输入/输出的服务程序，根据服务程序中的I/O指令进行数据传送
- 最简单、最经济的输入/输出方式，只需要很少的硬件



8.1.2 信息交换方式

■ 程序中断方式

- 某一外设的数据准备就绪后，它“主动”向CPU发出请求中断的信号，请求CPU暂时中断目前的工作而进行数据交换
- 当CPU响应这个中断时，便暂停运行主程序，并自动转移到该设备的中断服务程序，当中断服务程序结束后，CPU又回到原来的主程序
- 这种原理和调用子程序相仿，不过，这里要求转移到中断服务子程序的请求是由外部设备发出的
- 中断方式特别适合于随机出现的服务



8.1.2 信息交换方式

■ 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，在这种方式中，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
- 一般用于高速地传送成组的数据
- 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



8.1.2 信息交换方式

■ 通道方式

- 通道是一个特殊功能的处理器，它有自己的指令和程序专门负责数据输入输出的传输控制，而CPU将“传输控制”的功能下放给通道后只负责“数据处理”功能
- 这样，通道与CPU分时使用存储器，实现了CPU内部运算与I/O设备的并行工作

8.2 程序查询方式



8.2 程序查询方式

1、设备编址

- 统一编址：输入/输出设备中的控制寄存器、数据寄存器、状态寄存器等也和内存单元一样看待，将它们和内存单元联合在一起编排地址，这样就可用访问内存的指令（读、写指令）去访问I/O设备的某个寄存器，因而不需要专门的I/O指令组
- 单独编址：内存地址和I/O设备的地址是分开的，访问内存和访问I/O设备使用不同操作码的指令，即访问I/O设备有专门的I/O指令组

8.2 程序查询方式

2、输入输出指令

■ I/O指令一般具有如下功能：

- 置 “1” 或置 “0” 设备接口的某些控制触发器，用于控制设备进行某些动作、如启动、关闭等
- 测试设备的某些状态，如 “忙” 、 “准备就绪” 等，以便决定下一步的操作
- 传送数据



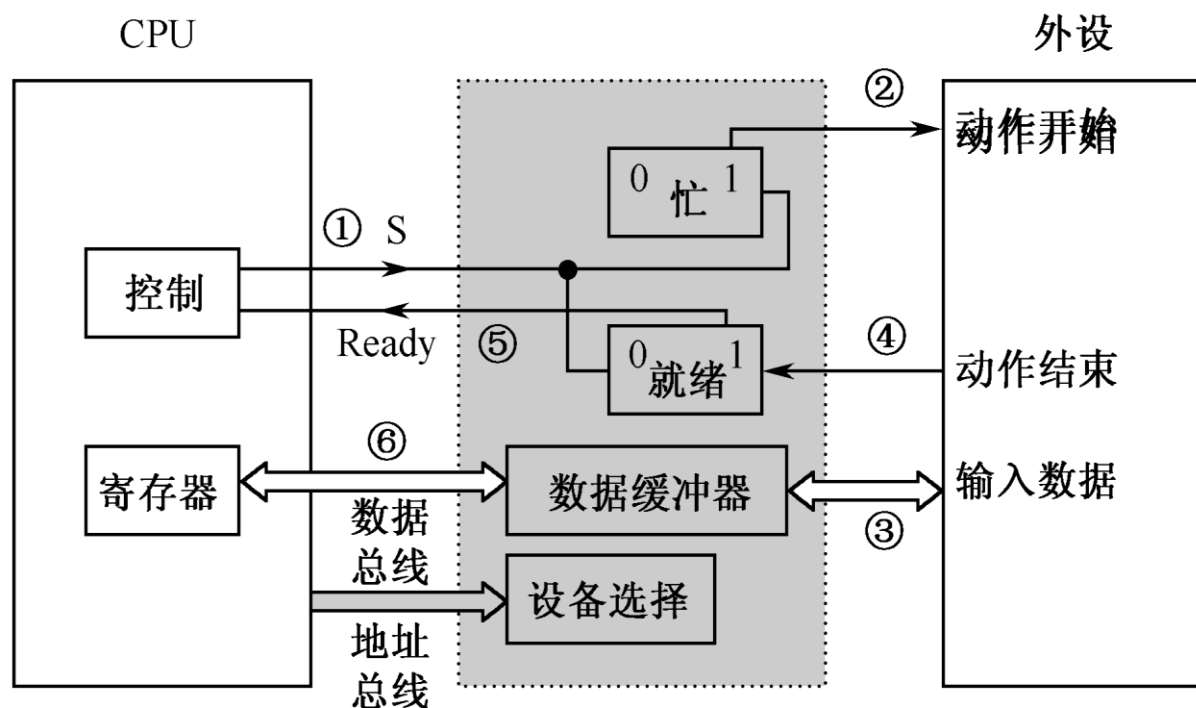
8.2 程序查询方式

3、程序查询接口

- 设备选择电路：接到总线上的每个设备预先都给定了设备地址码，设备选择电路判别地址总线上呼叫的设备是不是本设备，如果是，本设备就进入工作状态，否则不予理睬，实际上是设备地址的译码器

- 数据缓冲寄存器

- 设备状态寄存器：标志设备的工作状态





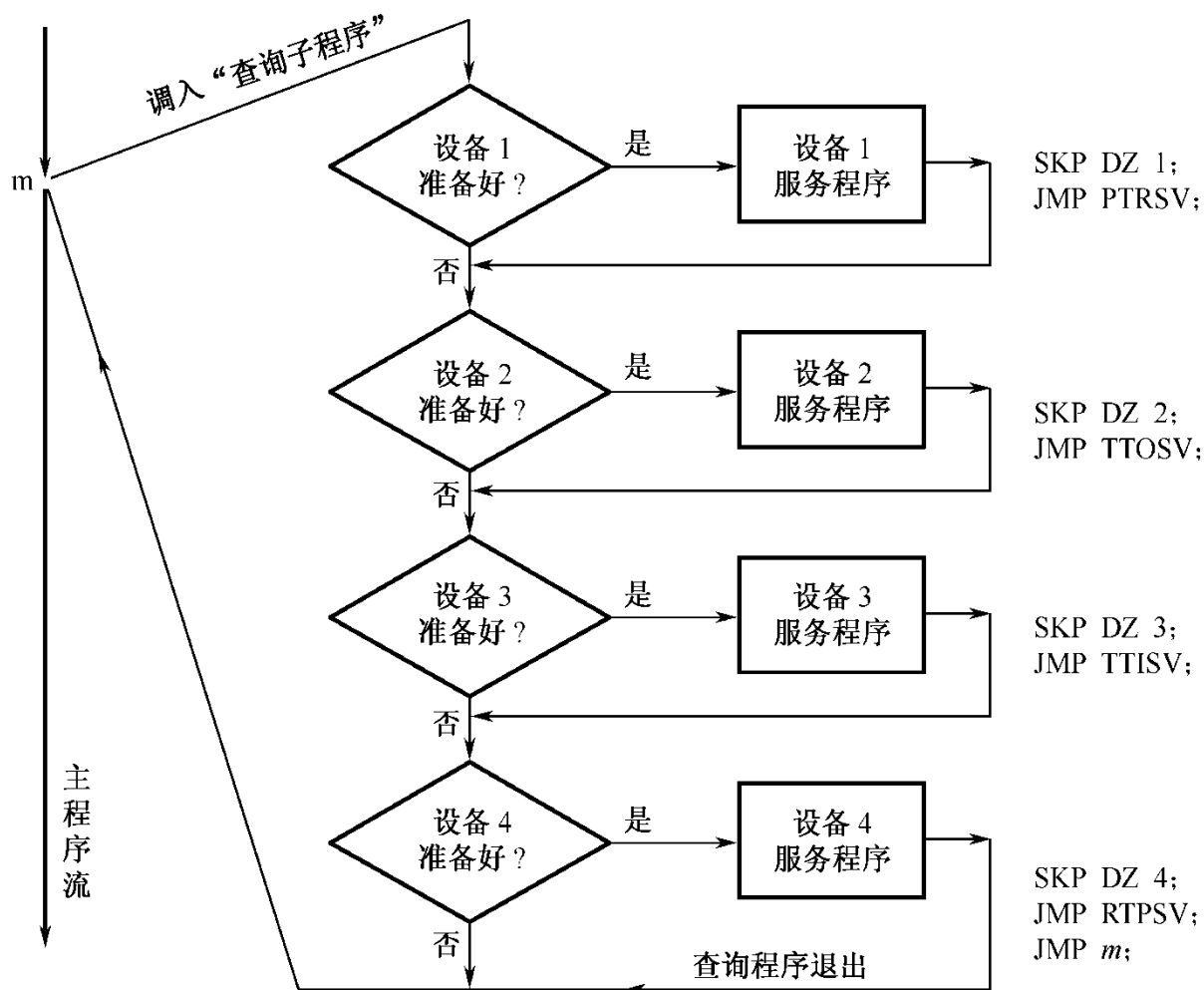
8.2 程序查询方式

4、程序查询I/O方式

- (1) 先向 I/O设备发出命令字，请求进行数据传送：
- (2) 从I/O接口读入状态字；
- (3) 检查状态字中的标志，看看数据交换是否可以进行
- (4) 假如这个设备没有准备就绪，则第（2）、第（3）步重复进行：一直到这个设备准备好交换数据、发出准备就绪信号 “Ready” 为止；
- (5) CPU从I/O 接口的数据缓冲寄存器输入数据，或者将数据从CPU输出至接口的数据缓冲寄存器，与此同时，CPU将接口中的状态标志复位

8.2 程序查询方式

■ 程序查询I/O设备流程图



8.3 程序中斷方式

提纲

8.3.1 中断的概念、功能

8.3.2 程序中断方式的基本接口

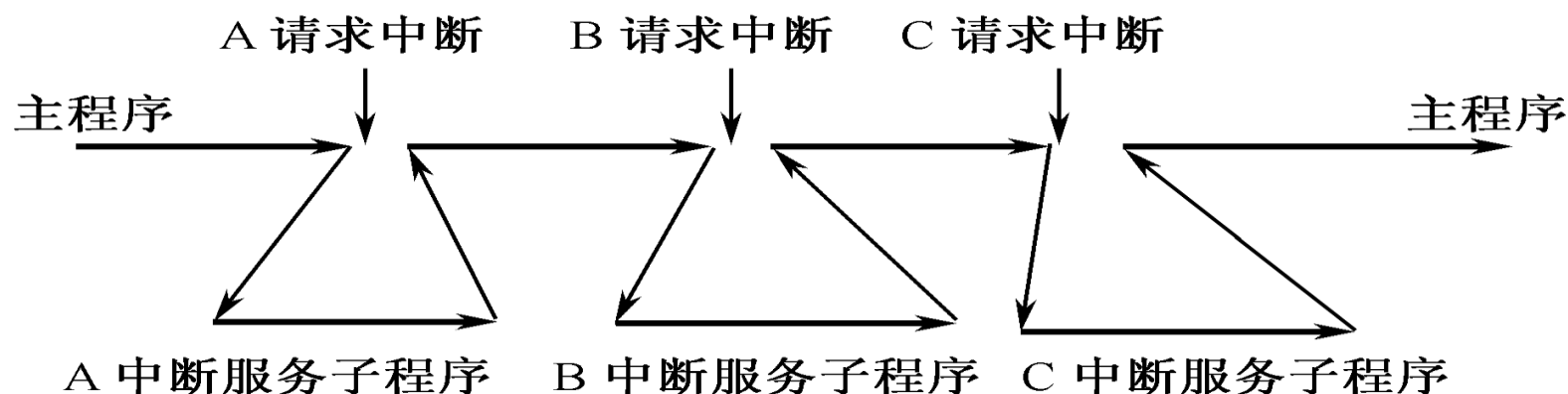
8.3.3 单级中断

8.3.4 多级中断

8.3.5 中断控制器

8.3.1 中断的概念、功能

- **中断** (Interrupt) 是指CPU暂时中止现执行程序，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
- 在速度较慢的外围设备准备自己的数据时，CPU照常执行自己的主程序。CPU和外围设备的一些操作是并行地进行的，因而同程序查询方式相比，计算机系统的效率大大提高了
- **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





8.3.1 中断的概念、功能

■ 中断系统的功能包括：

- (1) 实现主机和外设的并行工作
- (2) 处理故障
- (3) 实现多道程序和分时操作
- (4) 实时控制
- (5) 实现人机联系
- (6) 实现多机通信

8.3.1 中断的概念、功能

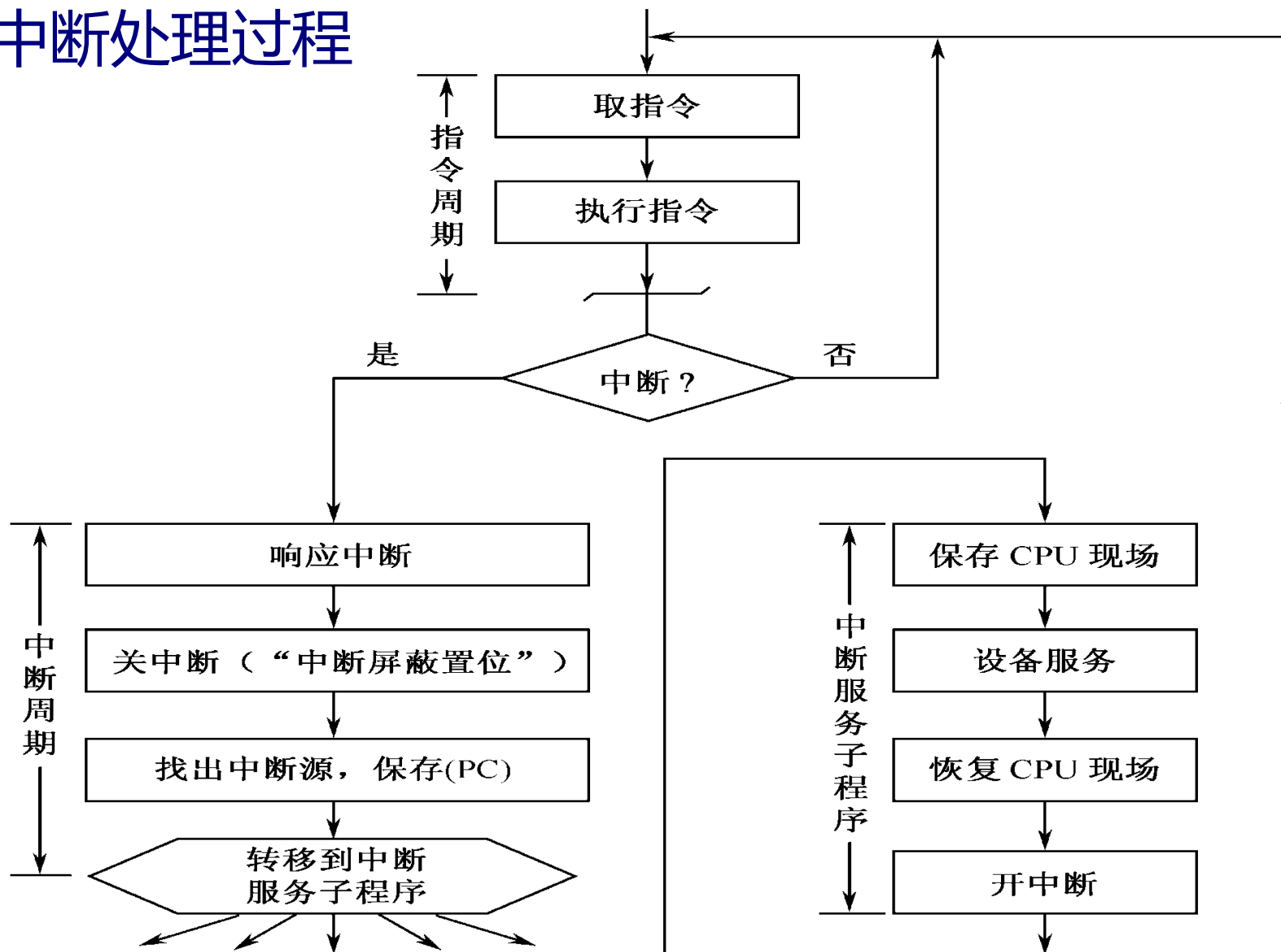
- **中断源**：能够向CPU发出中断请求的事件
- 常见中断源有：
 - 输入、输出设备中断。如键盘、打印机等工作过程中已做好接收或发送准备
 - 数据通道中断。如磁盘、磁带等要同主机进行数据交换等
 - 实时时钟中断
 - 故障中断。例如电源掉电、设备故障等要求CPU进行紧急处理等
 - 系统中断。如运算过程出现溢出、数据格式非法，数据传送过程出现校验错，控制器遇到非法指令等
 - 为了调试程序而设置的中断

8.3.1 中断的概念、功能

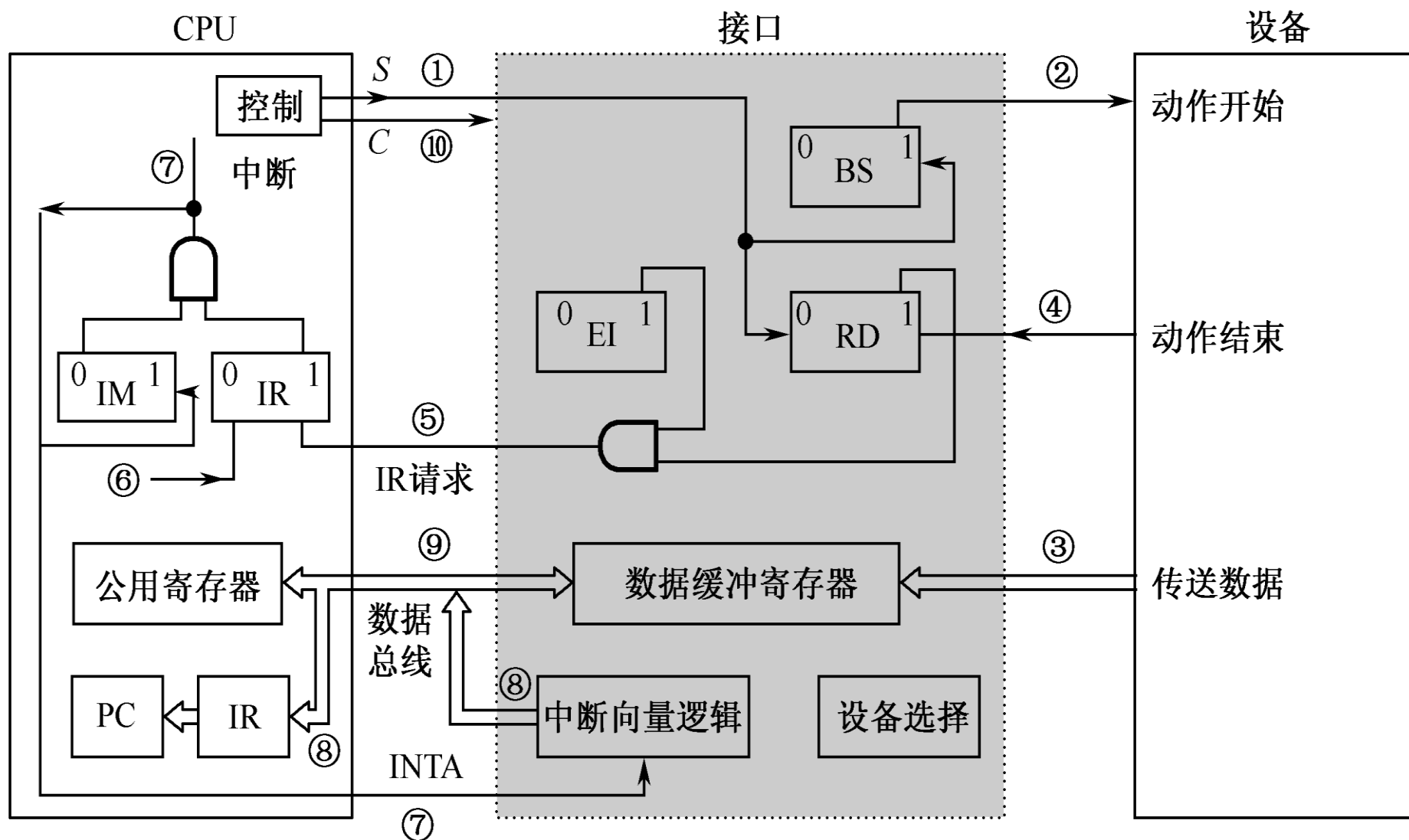
- 当CPU执行完一条现行指令时（即转入公操作时才受理设备的中断请求），如果外设向CPU发出中断请求，那么CPU在满足响应条件的情况下，将发出中断响应信号，与此同时关闭中断（“中断屏蔽”触发器置“1”），表示CPU不再受理另外一个设备的中断
- 这时，CPU将寻找中断请求源是哪一个设备，并保存CPU自己的程序计数器（PC）的内容
- 然后，它将转移到处理该中断源的中断服务程序，CPU在保存现场信息，设备服务（如交换数据）以后，将恢复现场信息
- 在这些动作完成以后，开放中断（“中断屏蔽”触发器置“0”），并返回到原来被中断的主程序的下一条指令

8.3.1 中断的概念、功能

■ 中断处理过程



8.3.2 程序中中断方式的基本接口



8.3.2 程序中断方式的基本接口

- RD，就绪标志触发器：一旦设备做好一次数据的接收或发送，便发出一个设备动作完毕信号，使RD标志为“1”，该标志用作中断源触发器，简称中断触发器
- EI，允许中断触发器：可以用程序指令来置位。EI为“1”时，某设备可以向CPU发出中断请求。目的是可以通过程序来控制是否允许某设备发出中断请求
- IR，中断请求触发器：暂存中断请求线上由设备发出的中断请求信号，当IR标志为“1”时，表示设备发出了中断请求
- IM，中断屏蔽触发器：CPU是否受理中断或批准中断的标志。IM标志为“0”时，CPU可以受理外界的中断请求

8.3.2 程序中断方式的基本接口

- 标号①-⑩表示由某一外设输入数据的控制过程
- ①表示由程序启动外设，将该外设接口的“忙”标志BS置“1”，“准备就绪”标志RD清“0”；
- ②表示接口向外设发出启动信号；
- ③表示数据由外设传送到接口的缓冲寄存器；
- ④表示当设备动作结束或缓冲寄存器数据填满时，设备向接口送出一控制信号，将数据“准备就绪”标志RD置“1”；
- ⑤表示允许中断标志EI为“1”时，接口向CPU发出中断请求信号；

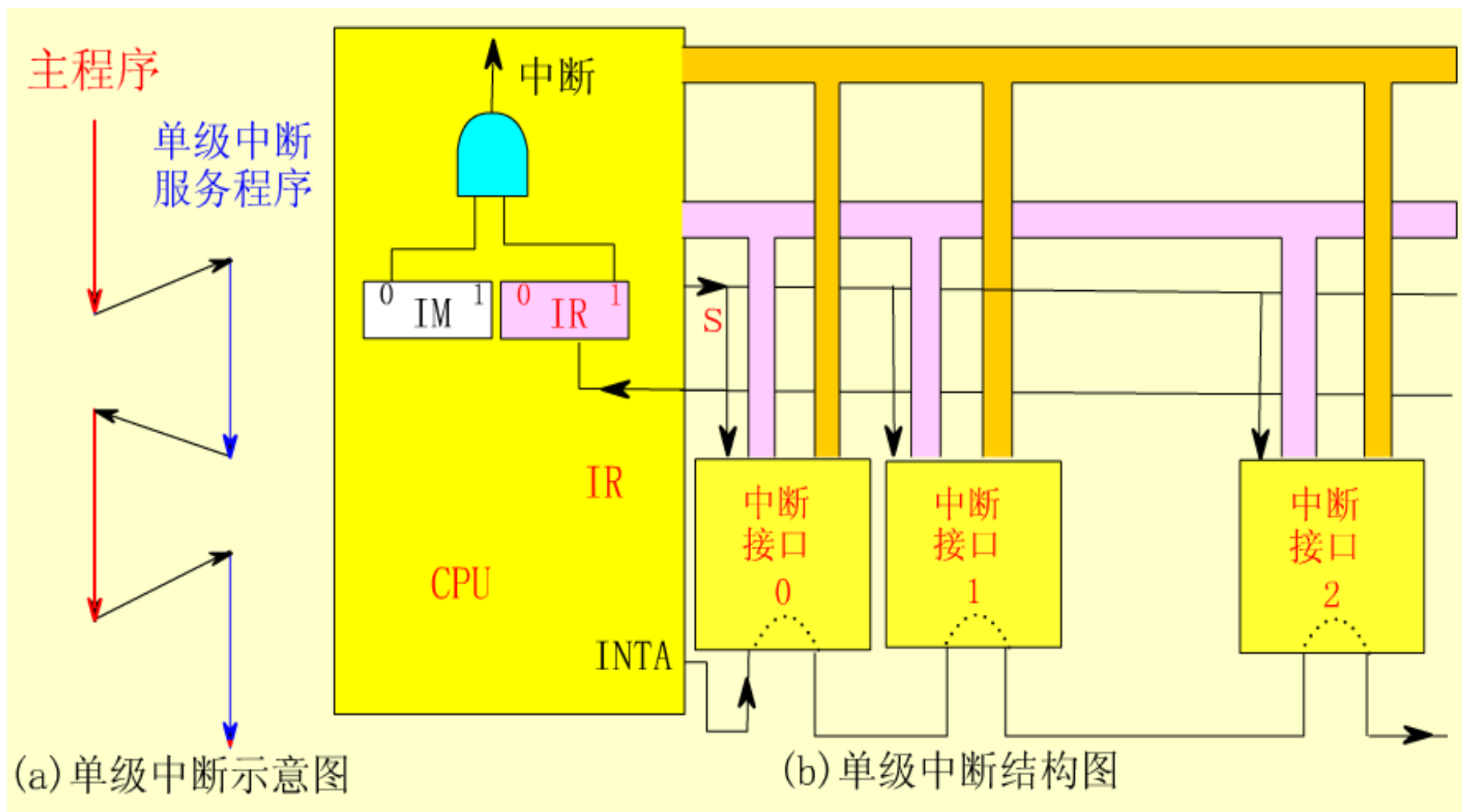
8.3.2 程序中断方式的基本接口

- ⑥表示在一条指令执行末尾CPU 检查中断请求线，将中断请求线的请求信号接收到“中断请求”标志IR；
- ⑦表示如果“中断屏蔽”标志IM为“0”时，CPU在一条指令执行结束后受理外设的中断请求，向外设发出响应中断信号并关闭中断：
- ⑧表示转向该设备的中断服务程序入口；
- ⑨表示在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读至CPU中的累加器或寄存器；
- ⑩表示 CPU 发出控制信号C将接口中的 BS和RD标志复位

8.3.3 单级中断

- 所有中断源属于同一级，离CPU越近，优先级越高，类似于第六章的链式查询方式
- 当响应某一中断请求时，执行该中断源的中断服务程序，在此过程中，不允许其它中断源再打断中断服务程序，即使优先权比它高的中断源也不能再打断
- 只有该中断服务程序执行完毕之后才能响应其他中断
- 所有的I/O 设备通过一条线向CPU发出中断请求信号，CPU响应中断请求后，发出中断响应信号INTA，以链式查询方式识别中断源

8.3.3 单级中断

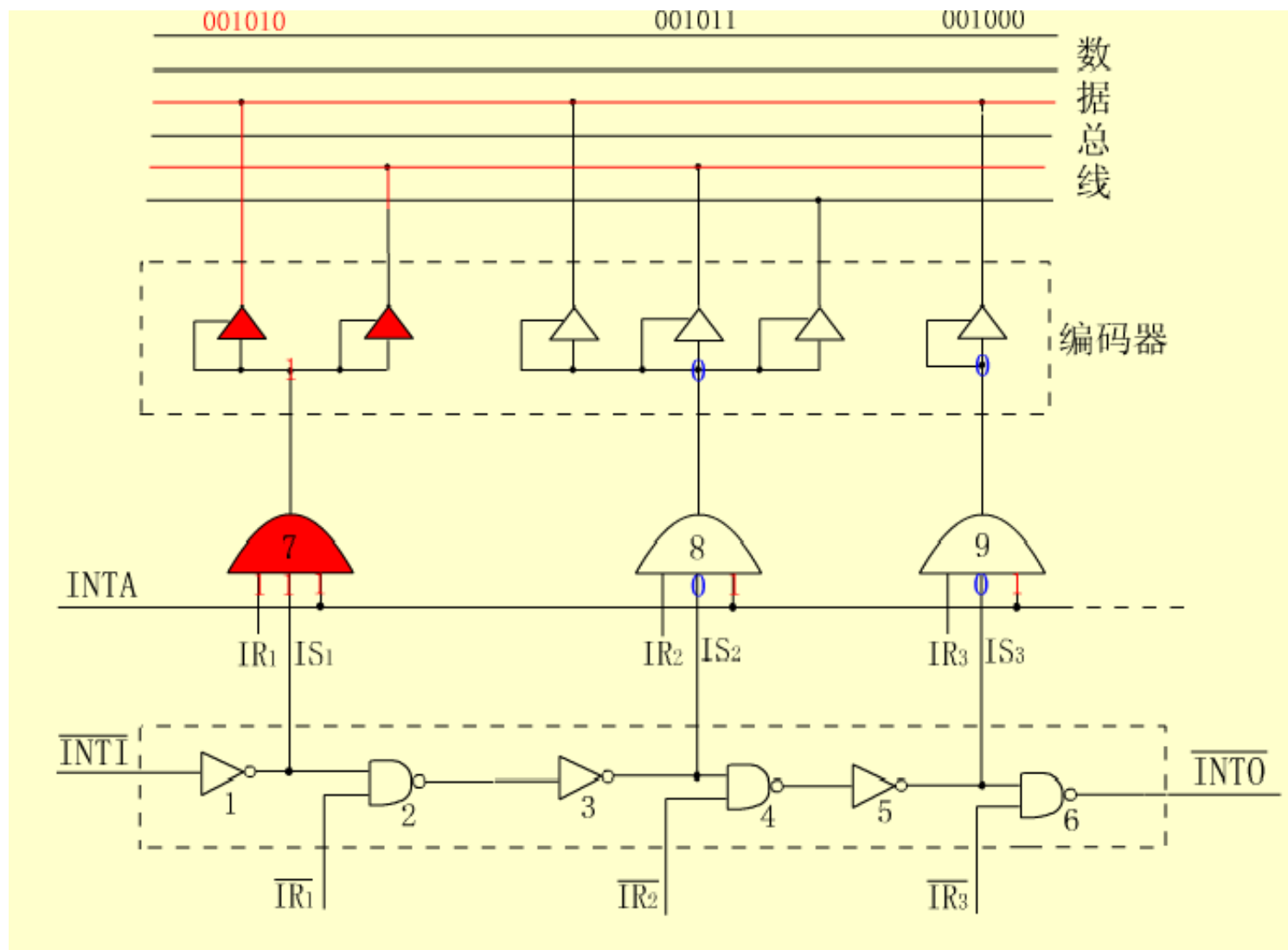


8.3.3 单级中断

- 中断源的识别：串行排队链法
 - IR_1, IR_2, IR_3 为中断请求信号
 - IS_1, IS_2, IS_3 为中断选中信号
 - $INTI$ 为中断排队输入
 - $INTO$ 为中断排队输出

8.3.3 单级中断

■ 串行排队链判优识别逻辑及中断向量的产生



8.3.3 单级中断

- $\overline{\text{INTI}}=0$ ，门1输出高电平，即 $\text{IS}_1=1$
- 若此时中断请求 $\text{IR}_1=1$ （有中断请求），当CPU发来中断识别信号 $\text{INTA}=1$ 时，发出IR请求的中断源被选中，选中信号经门7送入编码电路，产生一个唯一对应的设备地址，并经数据总线送往CPU的主存地址寄存器，然后执行该中断源设备的中断服务程序
- 另一方面、电于此时 IR_1 为0，封锁门2，使 IS_2 ， IS_3 全为低电平，即排队识别工作不再向下进行
- 若 IR_1 无请求，则 $\text{IR}_1=0$ ，门7被封锁，不会向编码电路送入选中信号，与此同时因 $\overline{\text{IR}}_1=0$ ，经门2和门3，使 $\text{IS}_2=1$ ，如果 $\text{IR}_2=1$ ，则被选中，否则查询链继续向下查询直至找到发出中断请求信号 IR_i 的中断源设备为止

8.3.3 单级中断

■ 中断向量:

- 当CPU响应中断时，由硬件直接产生一个固定的地址（即向量地址）
- 由向量地址指出每个中断源设备的中断服务程序入口，这种方法通常称为向量中断

8.3.4 多级中断

■ 概念

- 每级有一个中断优先权，一般说来，优先权高的中断级可以打断优先权低的中断服务程序
- 根据系统的配置不同，多级中断又可分为一维多级中断和二维多级中断
 - 一维多级中断是指每一级中断里只有一个中断源
 - 二维多级中断是指每级中断里又有多个中断源

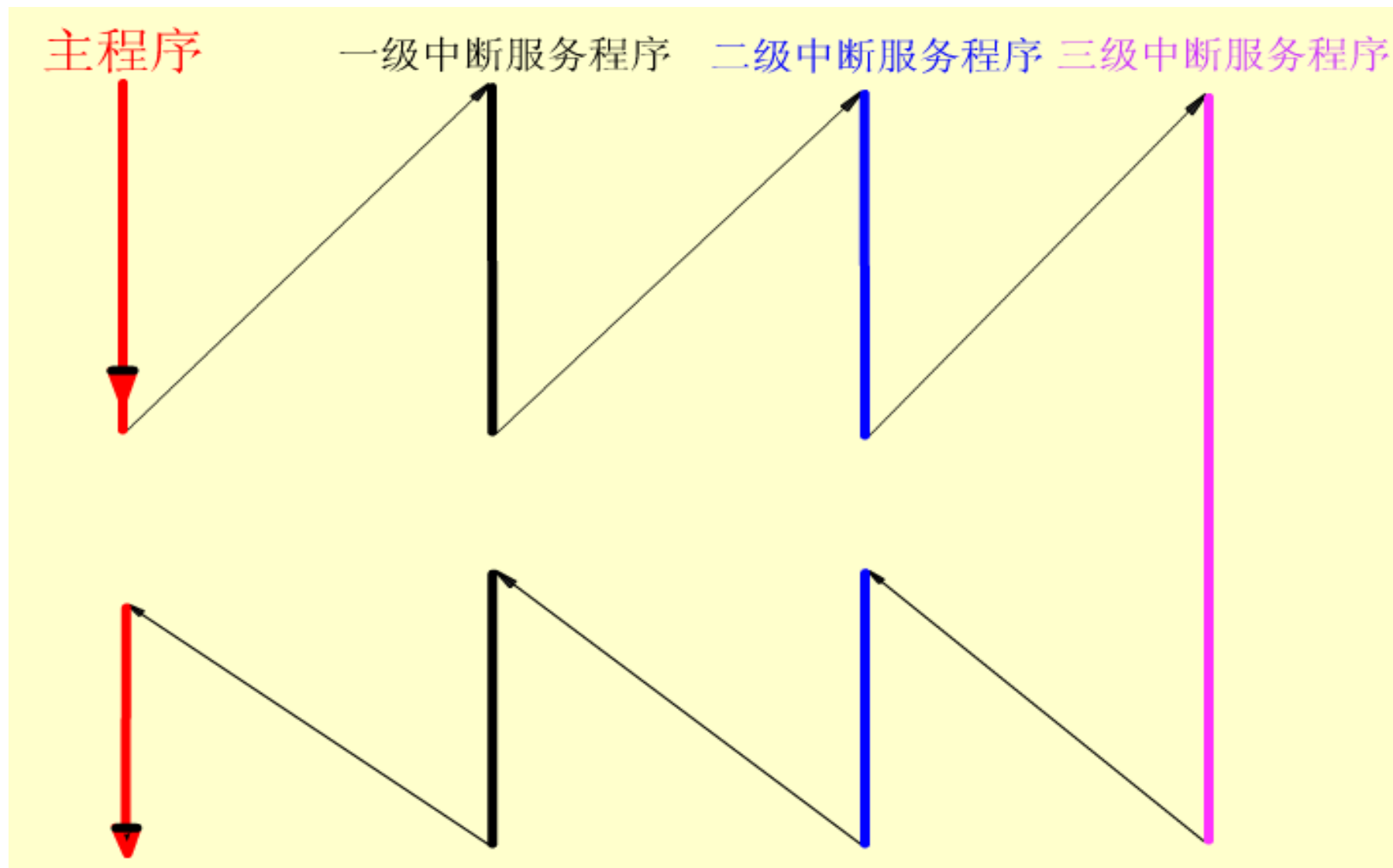


8.3.4 多级中断

■ 说明

- 一个系统有 n 级中断，则CPU中有 n 个IR， n 个IM
- 在响应中断时，需要把中断屏蔽寄存器的内容保存起来，并设置新的中断屏蔽状态，一般在某一级中断被响应后，要置“1”（关闭）本级和优先权低于本级的中断屏蔽触发器，置“0”（开放）更高级的中断屏蔽触发器，以此来实现正常的中断嵌套
- 多级中断可以嵌套，但同一级的中断不允许嵌套，必须是处理完一个中断后再响应和处理同一级内其它中断源
- 中断响应时，确定哪一级中断和中断源采用硬件实现，以获得较快的中断响应时间
- 使用多级堆栈保存现场（包括IM）

8.3.4 多级中断



8.3.4 多级中断

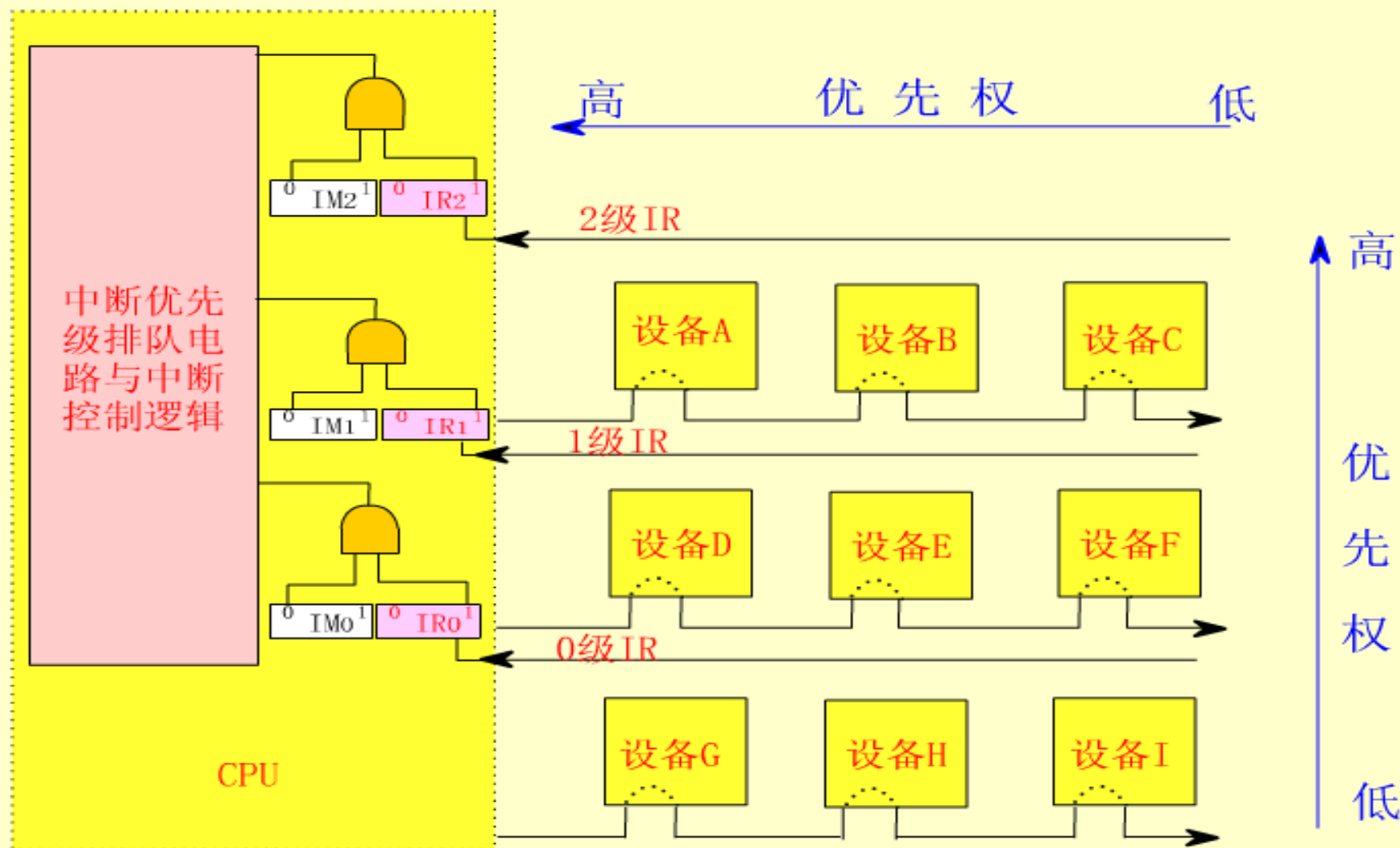
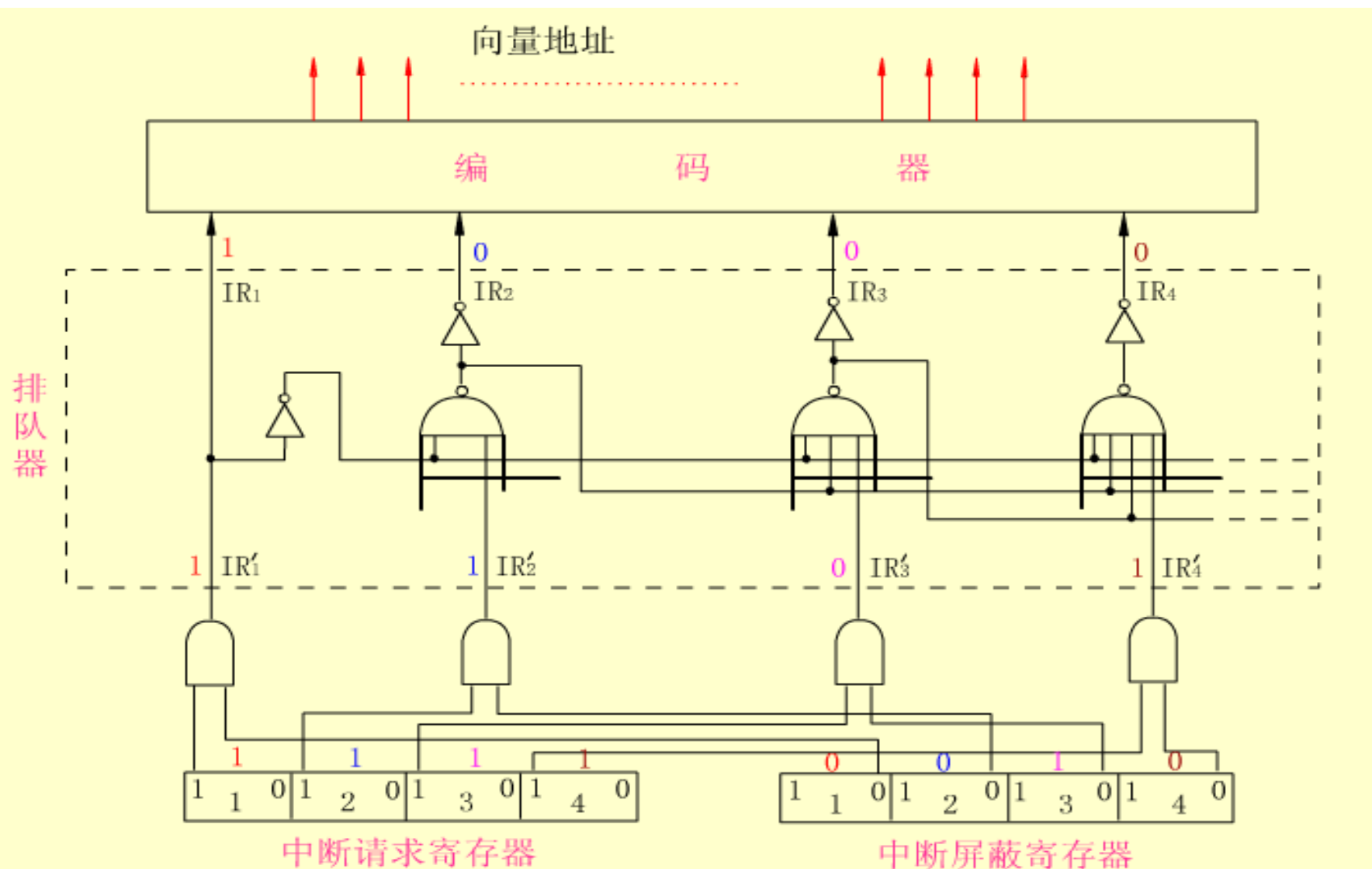


图8.9 多级中断 (b) 一维、二维多级中断结构

8.3.4 多级中断

■ 多级中断源的识别

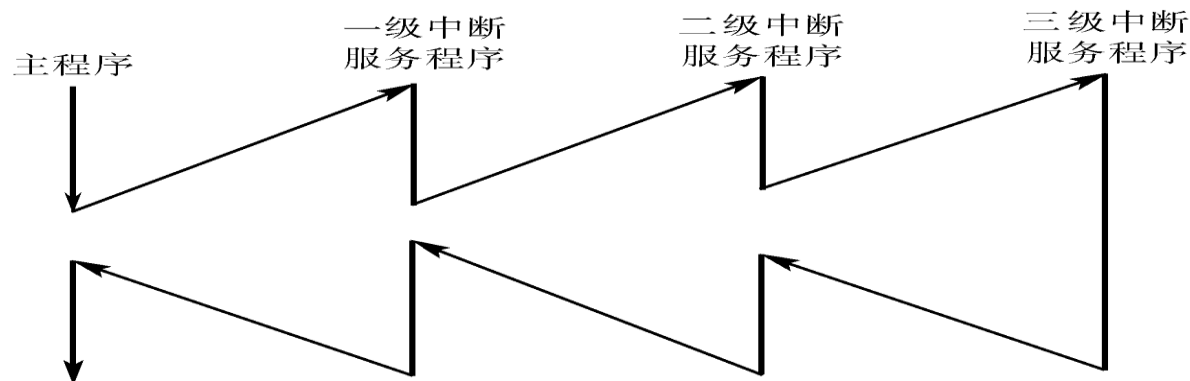


8.3.4 多级中断

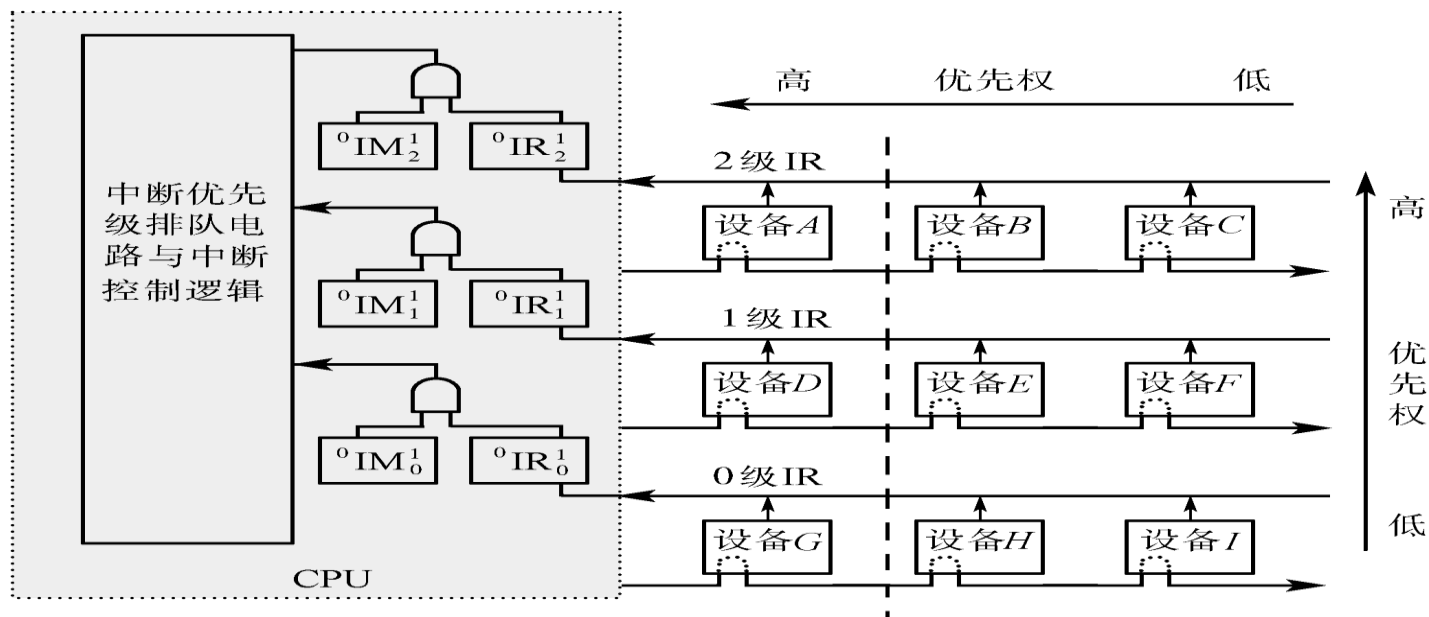
- 多级中断源的识别
 - 中断优先排队电路
 - 中断向量产生电路
- 在多级中断中，每一级均有一根中断请求线送往CPU的中断优先级排队电路，对每一级赋予了不同的优先级
- 每个中断请求信号保存在“中断请求”触发器中，经“中断屏蔽”触发器控制后，可能有若干中断请求信号 IR_i 进入排队电路
- 排队电路在若干中断源中决定首先响应哪个中断源，并在其对应的输出线 IR_i 上给出“1”信号，而其它各线为“0”信号（ IR_1 - IR_4 中只有一个信号有效）。之后，编码电路根据排上队的中断源输出信号 IR_i ，产生一个预定的地址码，转向中断服务程序入口地址

8.3.4 多级中断

■ 举例



(a) 多级中断示意图



(b) 一维、二维多级中断结构

8.3.4 多级中断

- (1) 在中断情况下, CPU和设备的优先级如何考虑? 请按降序排列各设备的中断优先级
- (2) 若CPU现执行设备B的中断服务程序, IM_2 , IM_1 , IM_0 的状态是什么? 如果CPU执行设备D的中断服务程序, IM_2 , IM_1 , IM_0 的状态又是什么?
- (3) 每一级的IM能否对某个优先级的个别设备单独进行屏蔽? 如果不能、采取什么办法可达到目的?
- (4) 假如设备C一提出中断请求, CPU 立即进行响应, 如何调整才能满足此要求?

8.3.4 多级中断

■ 分析

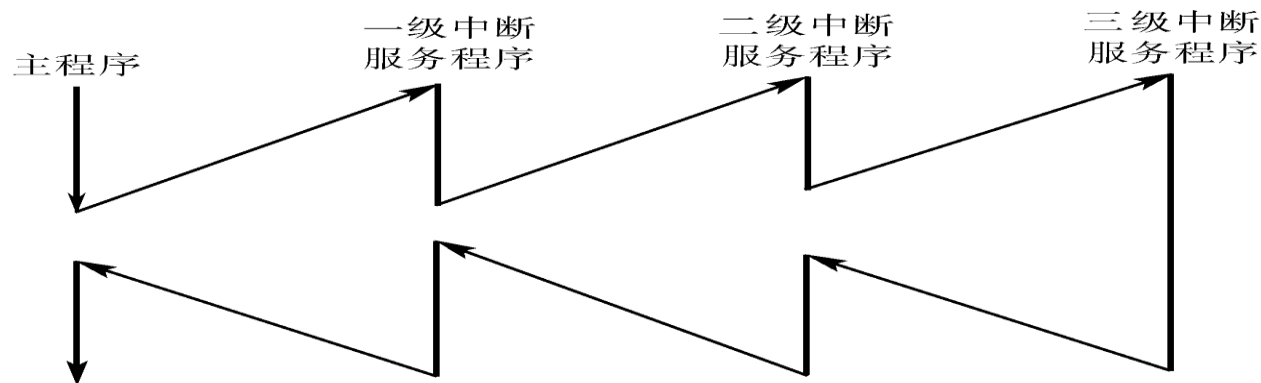
- (1) 在中断情况下, CPU的优先级最低。各设备的优先次序是: $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow I \rightarrow \text{CPU}$
- (2) 执行设备B的中断服务程序时 $IM_2IM_1IM_0=111$; 执行设备D的中断服务程序时, $IM_2IM_1IM_0=011$
- (3) 每一级的IM标志不能对某个优先级的个别设备进行单独屏蔽。可将接口中的EI(中断允许)标志清“0”, 它禁止设备发出中断请求
- (4) 要使设备C的中断请求及时得到响应, 可将设备C从第2级取出来, 单独放在第3级上, 使第3级的优先级最高, 即令 $IM_3=0$ 即可



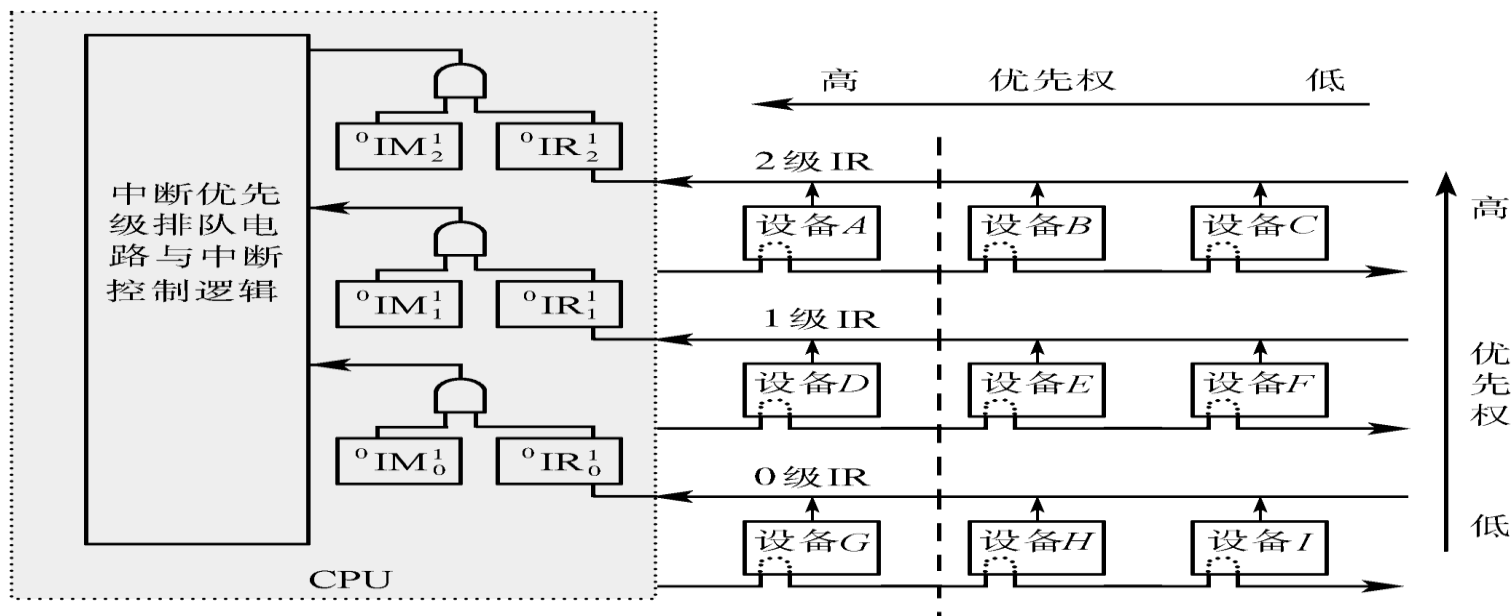
8.3.4 多级中断

- 只考虑A, B, C三个设备组成的单级中断结构, 它要求CPU在执行完当前指令时对中断请求进行服务。假设:
- (1) CPU “中断批准” 机构在响应一个新的中断之前, 先要让被中断的程序的一条指令一定要执行完毕;
- (2) T_{DC} 为查询链中每个设备的延迟时间;
- (3) T_A , T_B , T_C 分别为设备A, B, C的服务程序所需的执行时间;
- (4) T_S , T_R 为保存现场和恢复现场所需的时间;
- (5) 主存工作周期为 T_M 。
- 试问: 就这个中断请求环境来说, 系统在什么情况下达到中断饱和 (在全部的设备同时发出中断请求的情况下, 要把全部的中断请求按顺序响应一遍, 那就叫饱和) ?

8.3.4 多级中断



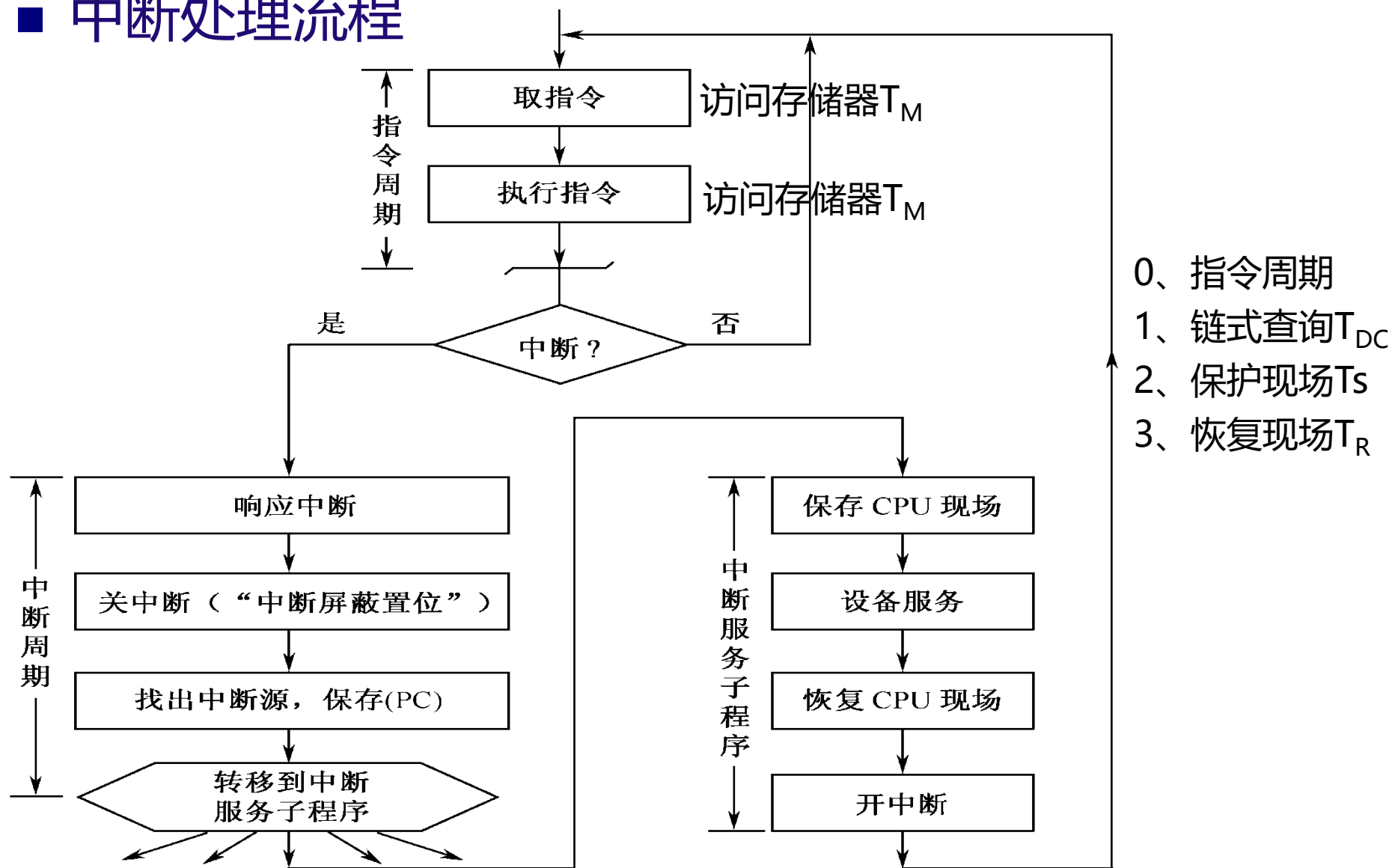
(a) 多级中断示意图



(b) 一维、二维多级中断结构

8.3.4 多级中断

■ 中断处理流程



8.3.4 多级中断

- 中断处理流程，并假设执行一条指令的时间也为 T_M 。如果三个设备同时发出中断请求，那么依次分别处理设备A、设备B、设备C的时间如下：

$$t_A = 2T_M + T_{DC} + T_S + T_A + T_R$$

$$t_B = 2T_M + 2T_{DC} + T_S + T_B + T_R$$

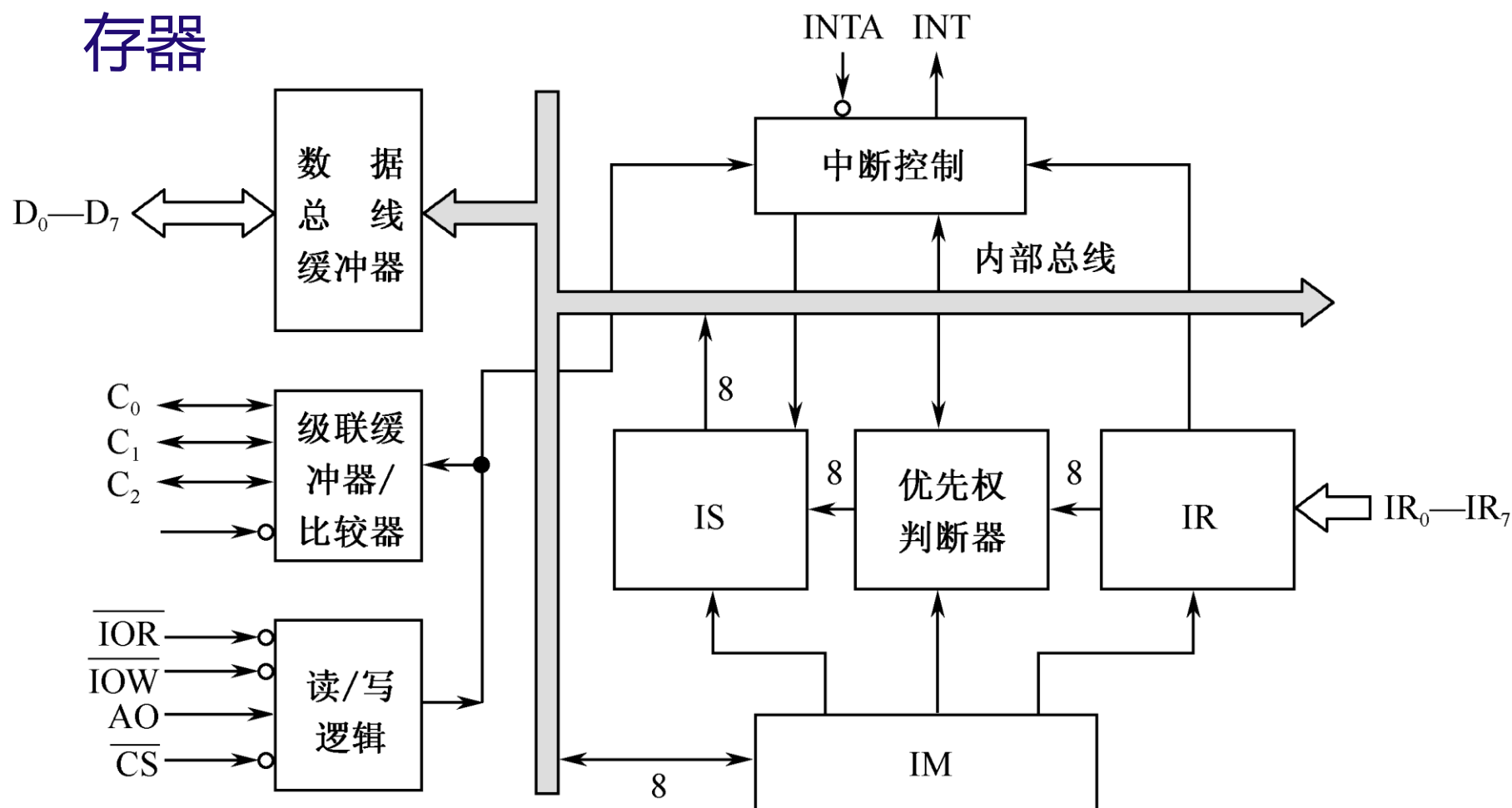
$$t_C = 2T_M + 3T_{DC} + T_S + T_C + T_R$$

- 处理三个设备所需的总时间为： $T = t_A + t_B + t_C$
- T 是达到中断饱和的最小时间，即中断极限频率为：

$$f = 1/T$$

8.3.5 中断控制器

- 8259中断控制器，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。IS：中断状态寄存器



8.4 DMA 方式

提纲

8.4.1 DMA方式的一般概念

8.4.2 DMA传送方式

8.4.3 基本的DMA控制器

8.4.4 选择型和多路型DMA控制器

8.4.1 DMA方式的一般概念

- 直接存储器访问 (Direct Memory Address)
DMA方式是为了在主存储器与I/O设备间高速交换批量数据而设置的
- 基本思想是：通过硬件控制实现主存与I/O设备间的直接数据传送，在传送过程中无需CPU的干预
- 数据传送是在DMA控制器控制下进行的
- 优点：速度快，有利于发挥CPU的效率

8.4.1 DMA方式的一般概念

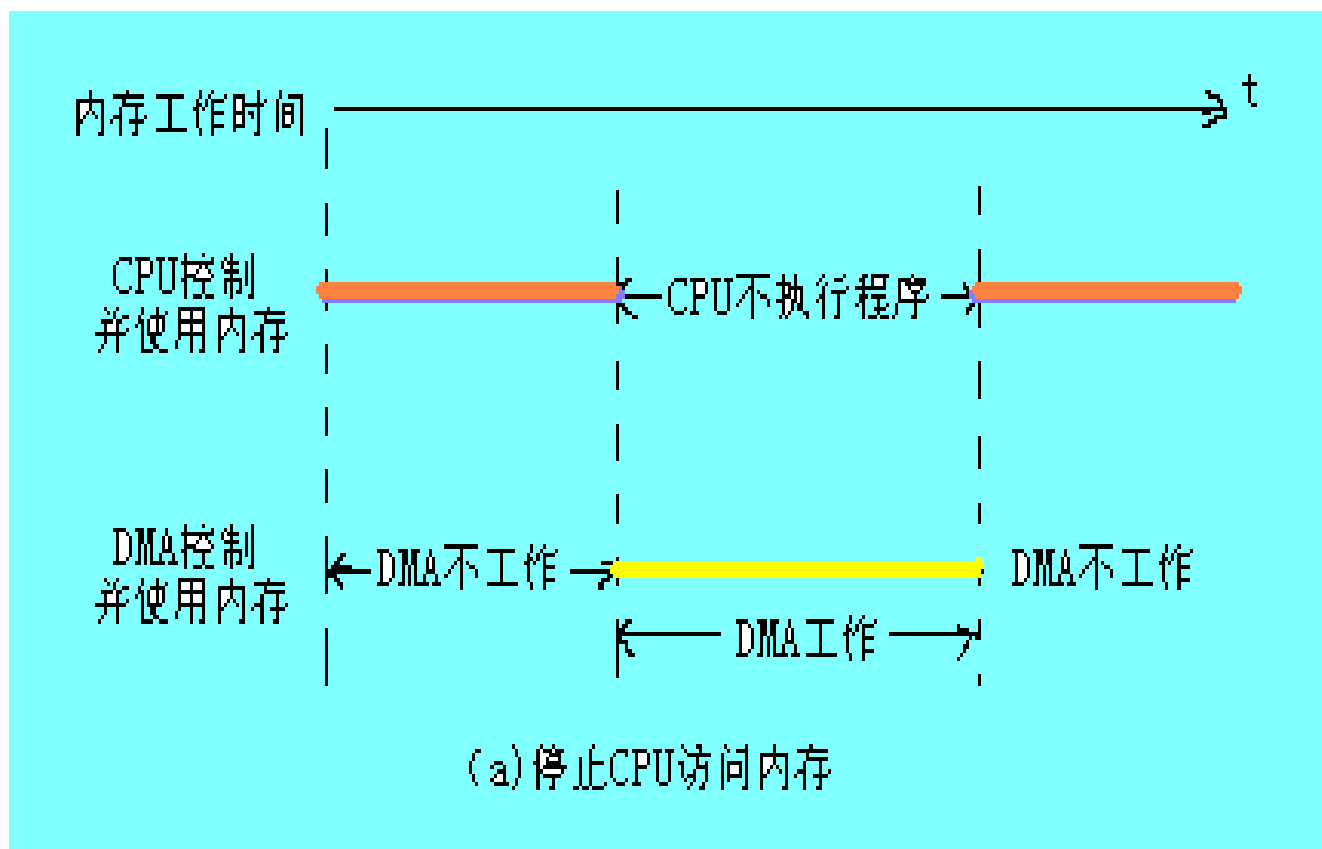
■ 过程描述:

- 由DMA控制器给出当前正在传送的数据的主存地址, 并统计传送数据的个数以确定一组数据的传送是否已结束
- 在主存中要开辟连续地址的专用缓冲器, 用来提供或接收传送的数据
- 在数据传送前和结束后要通过程序或中断方式对缓冲器和DMA控制器进行预处理和后处理

8.4.2 DMA 传送方式

- DMA控制器与CPU分时使用内存的三种方法：（1）停止CPU访存（2）周期挪用（3）DMA与CPU交替访存

1、停止CPU访存





8.4.2 DMA 传送方式

1、停止CPU访存

- 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，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
- 优点：
 - 控制简单，它适用于数据传输率很高的设备进行成组传送
- 缺点：
 - 在DMA控制器访内阶段，内存的效能没有充分发挥，相当一部分内存工作周期是空闲的。这是因为，外围设备传送两个数据之间的间隔一般总是大于内存存储周期，即使高速 I/O设备也是如此



8.4.2 DMA 传送方式

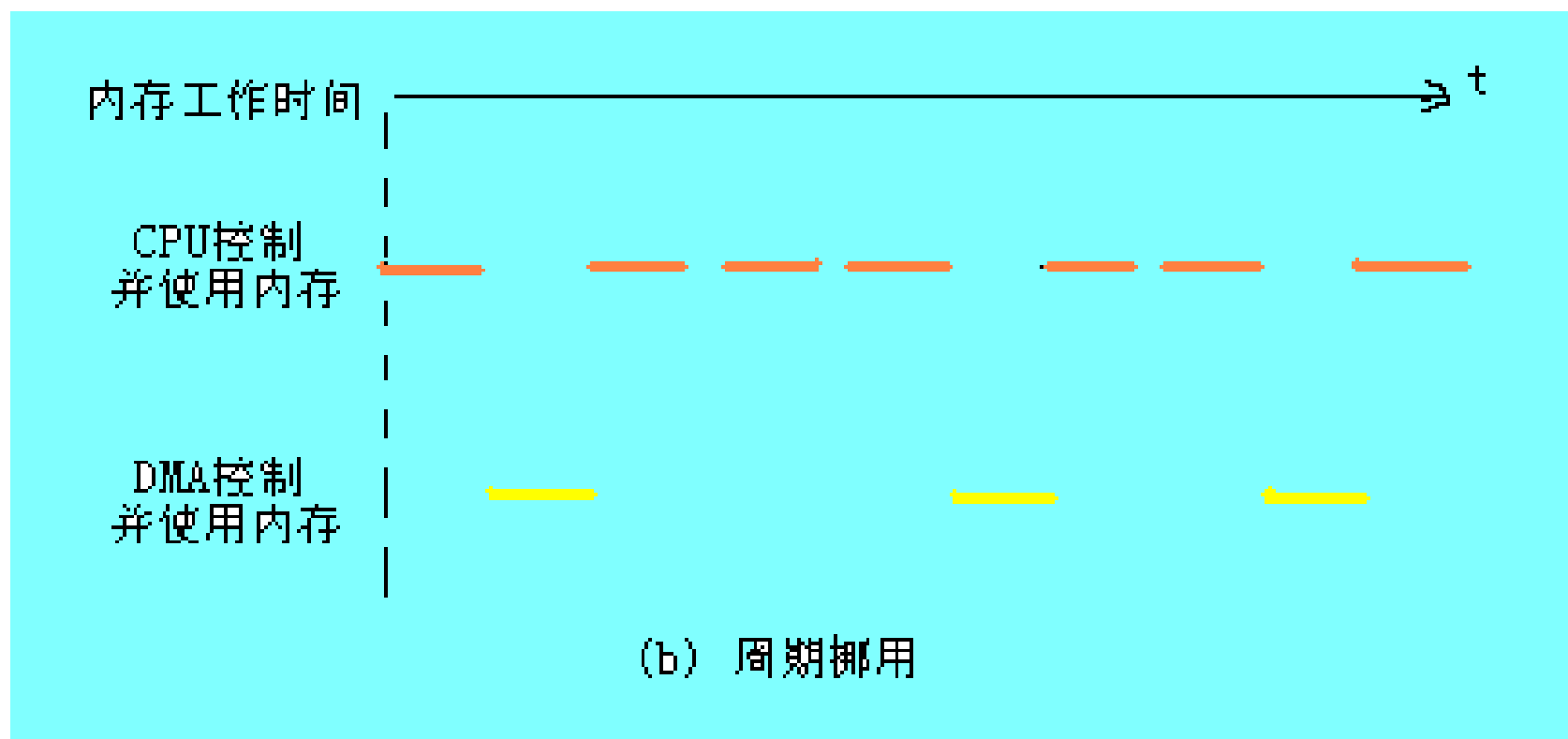
2、周期挪用方式

- 当I/O设备有DMA请求时，则由I/O设备挪用一个或几个内存周期I/O设备要求
- DMA传送时可能遇到两种情况：一种是此时 CPU 不需要访内，如CPU正在执行乘法指令，由于乘法指令执行时间较长，此时I/O访内与CPU访内没有冲突，即I/O设备挪用一二个内存周期对CPU执行程序没有任何影响，另一种情况是I/O设备要求访内时CPU也要求访内，这就产生了访内冲突，在这种情况下I/O设备访内优先，因为I/O访内有时间要求，前一个I/O数据必须在下一个访内请求到来之前存取完毕。在这种情况下I/O设备挪用一二个内存周期，意味着 CPU 延缓了对指令的执行

8.4.2 DMA 传送方式

2、周期挪用方式

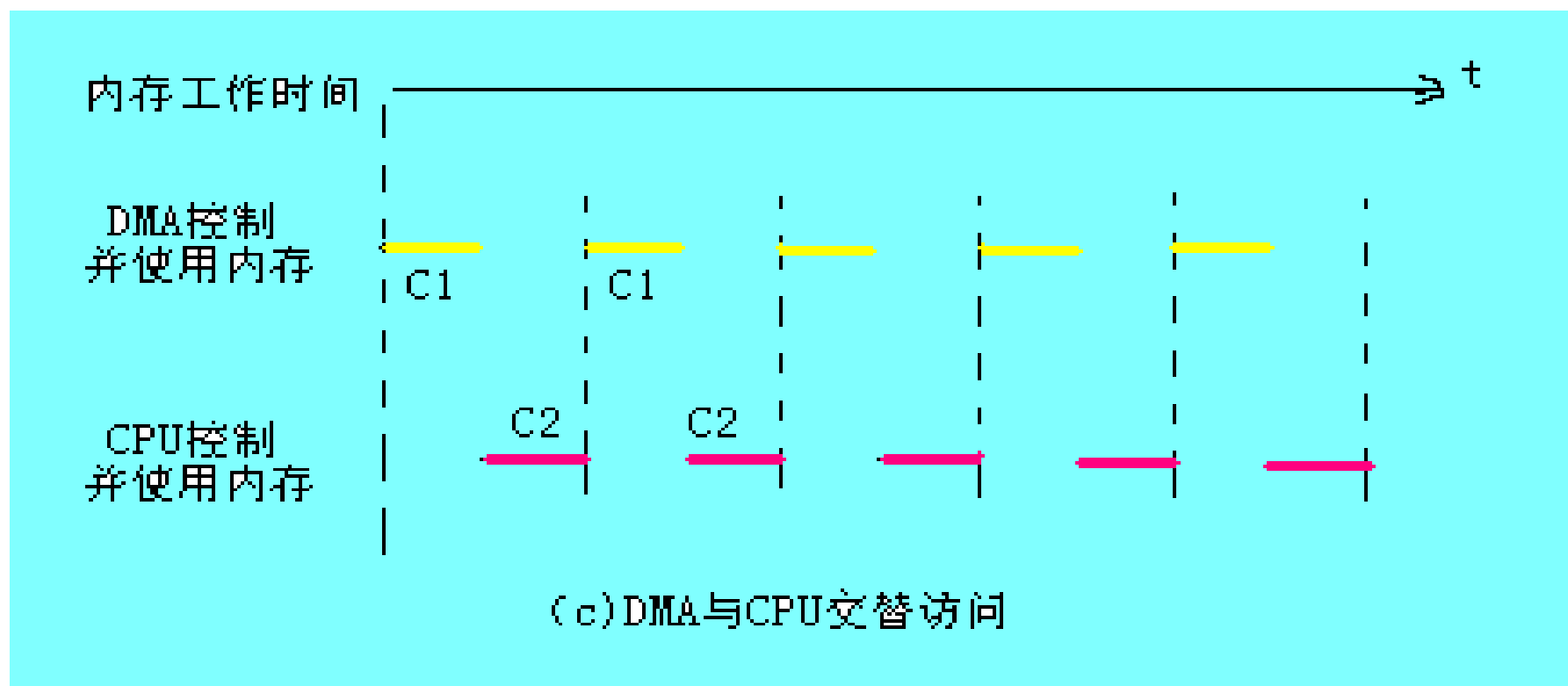
- DMA控制器与主存储器之间传送一个数据，占用（窃取）一个CPU周期，即CPU暂停工作一个周期，然后继续执行程序



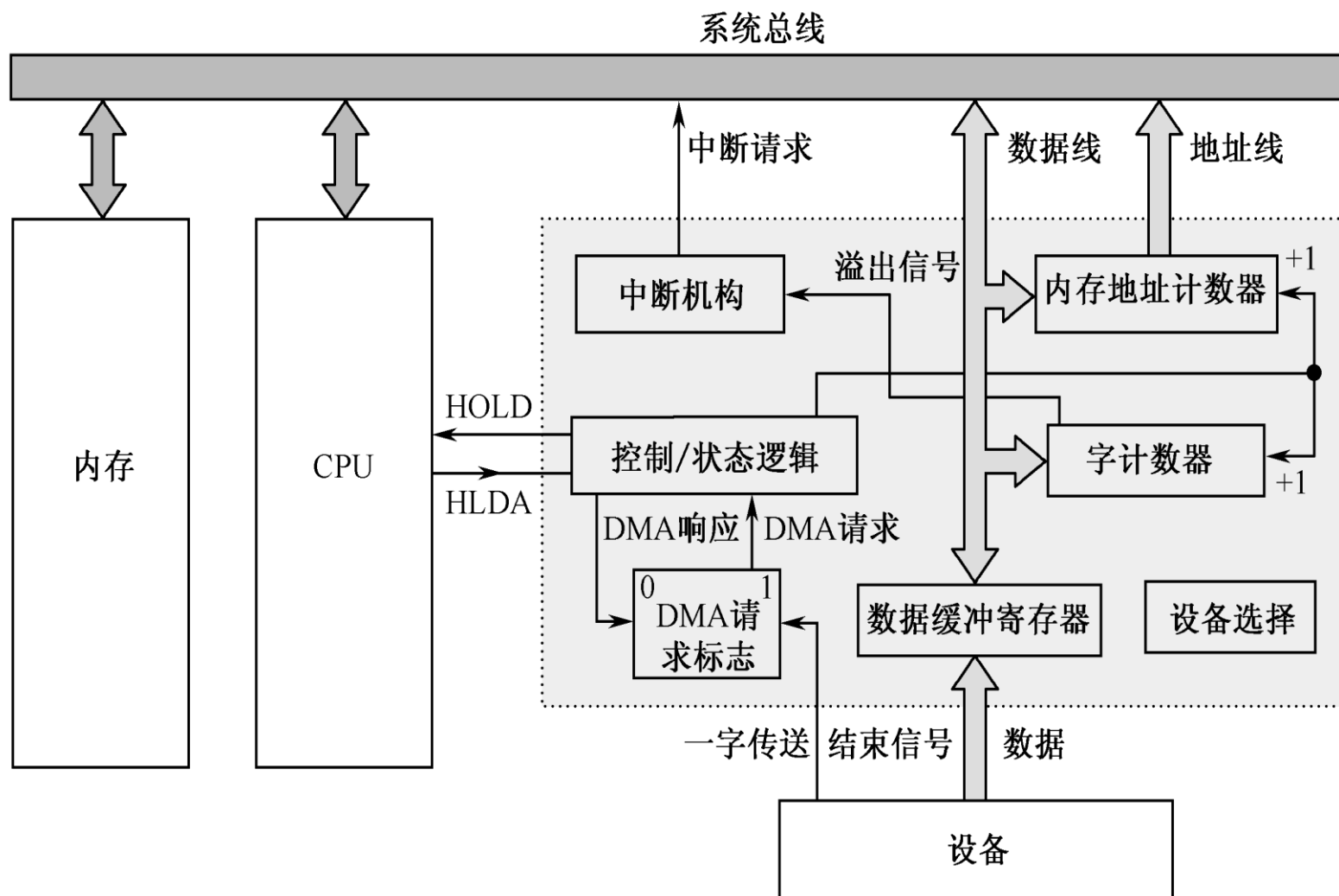
8.4.2 DMA 传送方式

3、DMA和CPU交替访问内存工作方式

- 如果CPU的工作周期比内存存取周期长很多，可以采用该方法
- 总线控制权的转移速度快，DMA效率高。



8.4.3 基本的DMA控制器



8.4.3 基本的DMA控制器

1、DMA控制器的基本组成

- (1) 内存地址计数器：用于存放内存中要交换的数据的地址，在DMA传送前，须通过程序将数据在内存中的起始位置（首地址）送到内存地址计数器。而当DMA传送时每交换一次数据，将地址计数器加“1”，从而以增量方式给出内存中要交换的一批数据的地址
- (2) 字计数器：用于记录传送数据块的长度（多少字数）
- (3) 数据缓冲寄存器：用于暂存每次传送的数据（一个字）



8.4.3 基本的DMA控制器

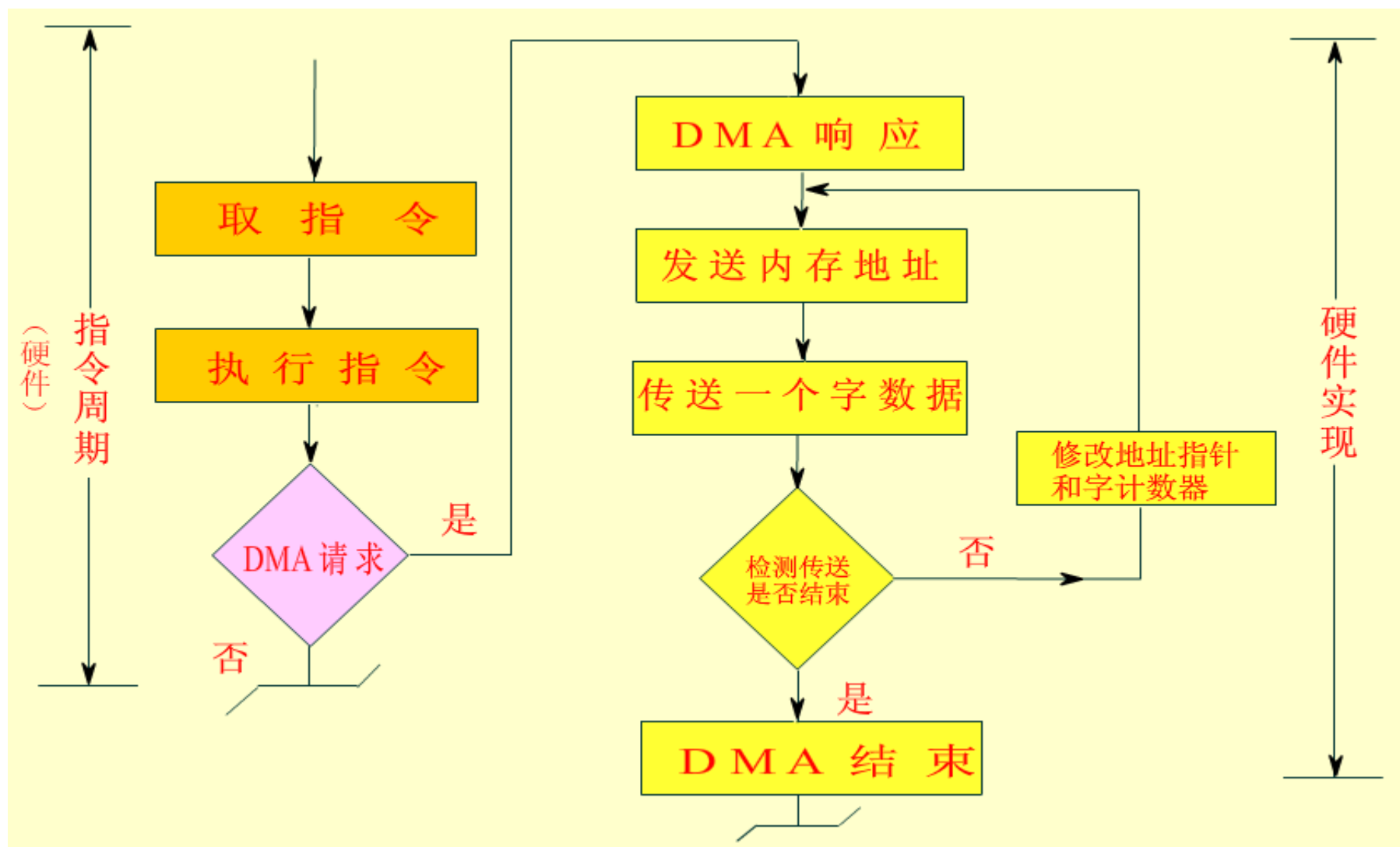
- (4) “DMA请求”标志：当设备准备好一个数据字后给出一个控制信号，使“DMA请求”标志置“1”。该标志置位后向“控制/状态”逻辑发出DMA请求，后者又向CPU发出总线使用权的请求（HOLD），CPU响应此请求后发回响应信号HLDA
- (5) “控制/状态”逻辑：由控制和时序电路，以及状态标志等组成，用于修改内存地址计数和字计数器，指定传送类型（输入或输出），并对“DMA请求”信号和CPU响应信号进行协调和同步
- (6) 中断机构：当字计数器溢出时（全0），意味着一组数据交换完毕，由溢出信号触发中断机构，向CPU提出中断报告

8.4.3 基本的DMA控制器

2、传送过程

- 当外设有DMA请求时，通常CPU在本指令周期结束后，响应DMA请求

8.4.3 基本的DMA控制器



8.4.3 基本的DMA控制器

- 传送一个数据块可以分为三个阶段。
 - 预处理阶段：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数，在这些工作完成后，CPU 继续执行原来的主程序



8.4.3 基本的DMA控制器

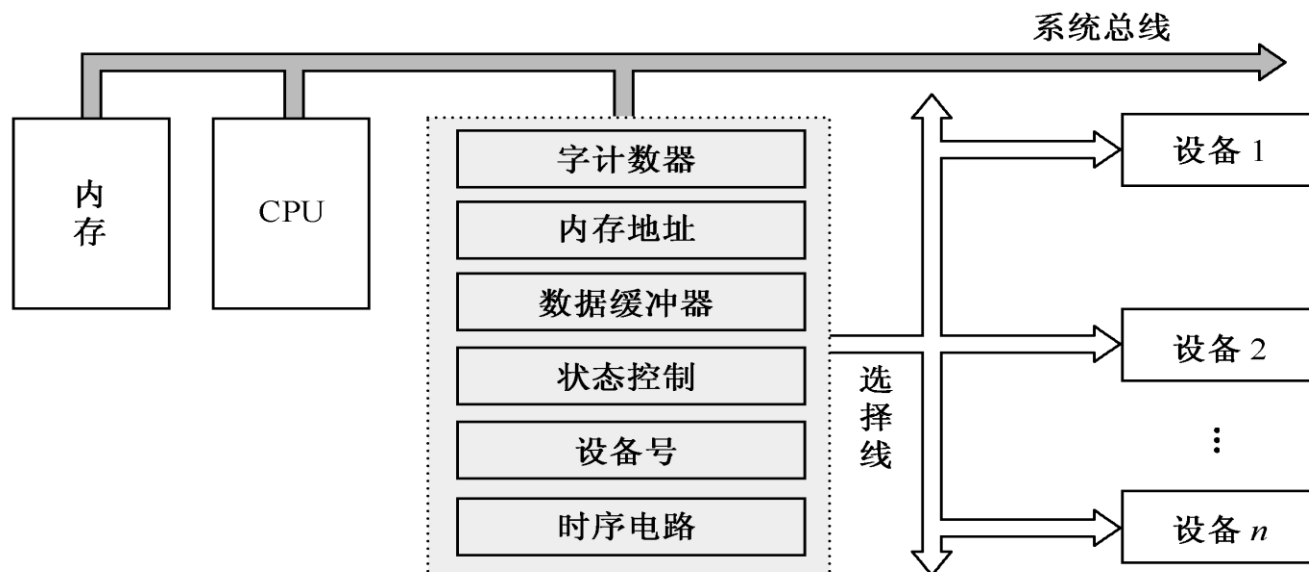
- 数据传送阶段：当外设准备好发送数据（输入）或接受数据（输出）时，它发出DMA请求，DMA控制器向CPU发出总线使用权的请求（HOLD），CPU在本指令周期执行结束后响应该请求，并使CPU的总线驱动器处于第三态（高阻状态）之后，CPU与系统总线相脱离，而DMA控制器接管数据总线与地址总线的控制，并向内存提供地址，于是在内存和外围设备之间进行数据交换，每交换一个字，则地址计数器和字计数器加“1”
- 当计数值到达零时，DMA操作结束，DMA控制器向CPU提出中断报告

8.4.3 基本的DMA控制器

- 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

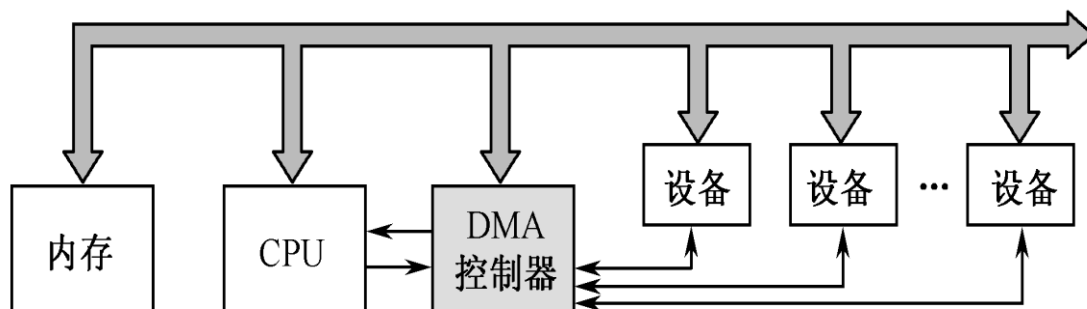
8.4.4 选择型和多路型DMA控制器

- 选择型：在物理上可以连接多个设备，而在逻辑上只允许接一个设备，换句话说，在某一段时间内只能为一个设备服务
- 在每个数据块传送之前的预置阶段，除了用程序中I/O指令给出数据块的传送个数、起始地址、传送命令外，还要给出所选择的设备号，从预置开始，一直到这个数据块传送结束DMA控制器只为所选设备服务



8.4.4 选择型和多路型DMA控制器

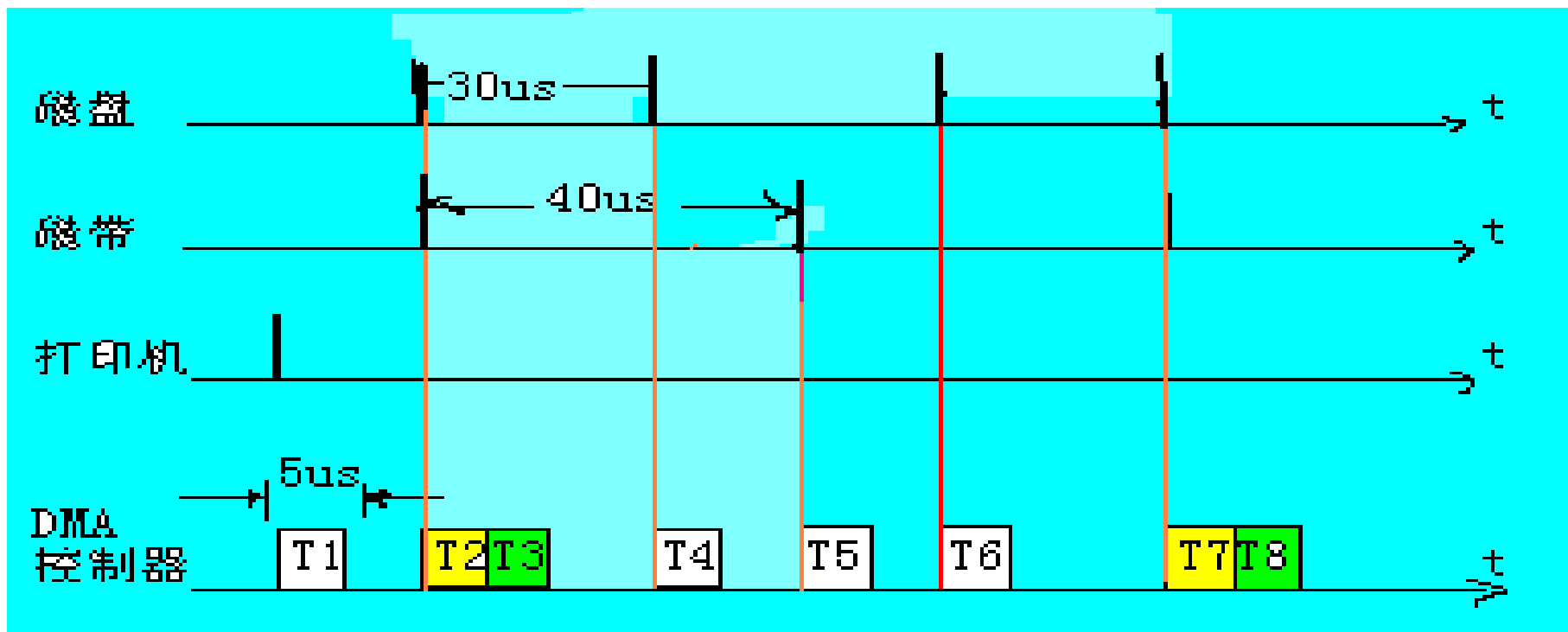
- 选择型DMA控制器不适用于慢速设备（否则有些设备需要等待较长时间），但是多路型DMA控制器却适合于同时为多个慢速外围设备服务
- 多路型DMA控制器不仅在物理上可以连接多个外围设备，而且在逻辑上也允许这些外围设备同时工作，各设备以字节交叉方式通过DMA控制器进行数据传送
- 由于多路型DMA同时要为多个设备服务，因此对应多少个DMA通路（设备），在控制器内部就有多少组寄存器用于存放各自的传送参数



8.4.3 基本的DMA控制器

- 例：下图中假设有磁盘、磁带、打印机三个设备同时工作。磁盘以 $30\mu\text{s}$ 的间隔向控制器发DMA请求，磁带以 $45\mu\text{s}$ 的间隔发DMA请求，打印机以 $150\mu\text{s}$ 间隔发DMA请求。根据传输速率，磁盘优先权最高，磁带次之，打印机最低，图中假设DMA控制器每完成一次DMA传送所需的时间是 $5\mu\text{s}$ 。若采用多路型DMA控制器，请画出DMA控制器服务三个设备的工作时间图

8.4.3 基本的DMA控制器



- T_1 间隔中控制器首先为打印机服务，因为此时只有打印机有请求。 T_2 间隔前沿磁盘、磁带同时有请求，首先为优先权高的磁盘服务，然后为磁带服务，每次服务传送一个字节。在 $150\mu s$ 时间阶段中，为打印机服务只有一次(T_1)，为磁盘服务四次(T_2, T_4, T_6, T_7)，为磁带服务三次(T_3, T_5, T_8)。从图上看到，在这种情况下DMA尚有空闲时间，说明控制器还可以容纳更多设备。

8.5 通道方式

提纲

8.5.1

通道的基本概念

8.5.2

通道的类型

8.5.3

通道的发展



8.5.1 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。这种输入输出系统增强了主机与通道操作的并行能力以及各通道之间、同一通道的各设备之间的并行操作能力。同时也为用户提供了增减外围设备的灵活性



8.5.1 通道的基本概念

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

8.5.1 通道的基本概念

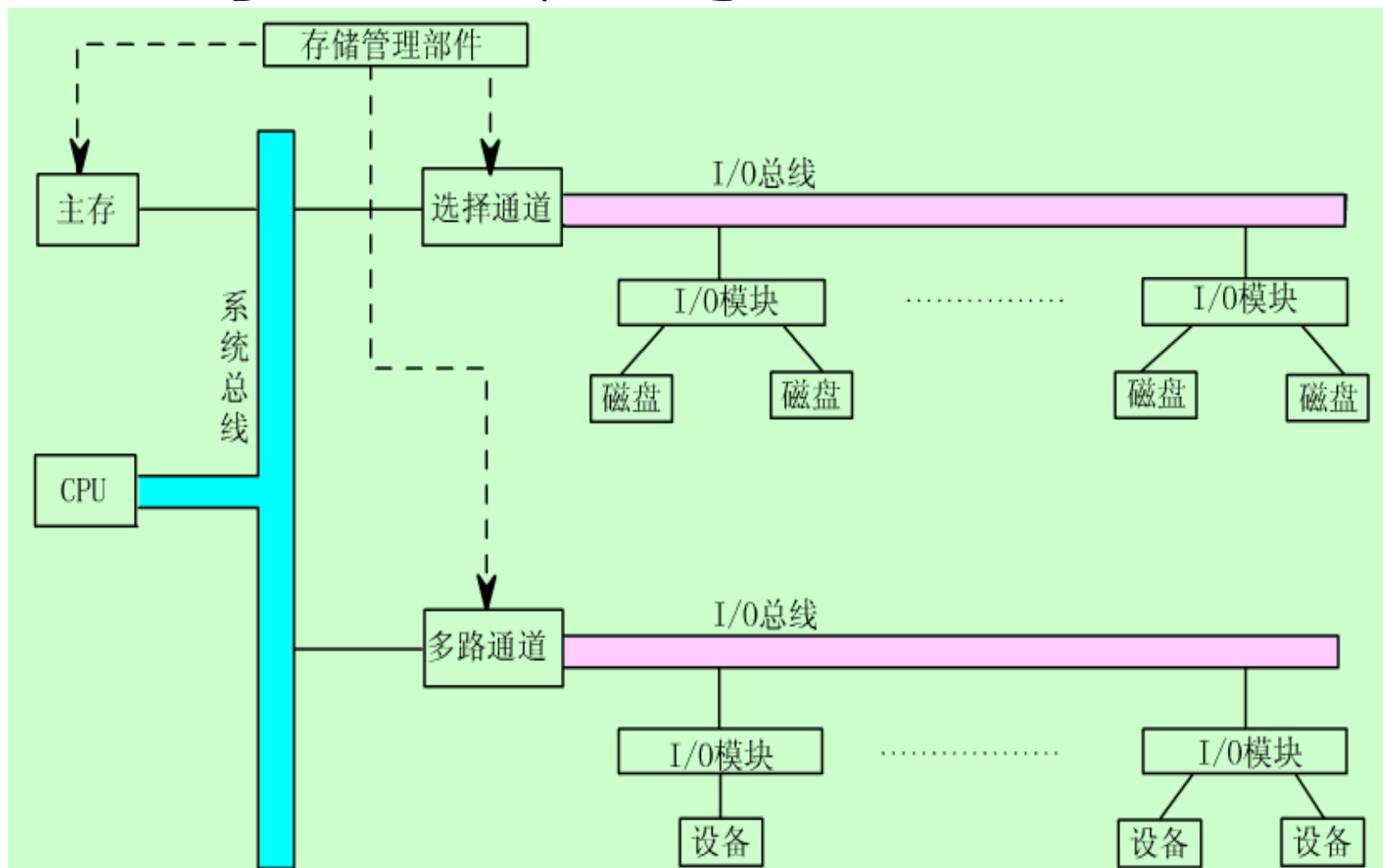


图8.20 通道结构



8.5.2 通道的类型

■ 选择通道

- 选择通道又称高速通道，在物理上它可以连接多个设备，但是这些设备不能同时工作，在某一段时间内通道只能选择一个设备进行工作

■ 多路通道

- 数组多路通道：当某设备进行数据传送时，通道只为该设备服务；当设备在执行寻址等控制性动作时，通道暂时断开与这个设备的连接，挂起该设备的通道程序，去为其他设备服务，即执行其他设备的通道程序。所以数组多路通道很像是一个多道程序的处理器
- 字节多路通道：字节多路通道主要用于连接大量的低速设备，如键盘、打印机等等

8.5.2 通道的类型

- 字节多路通道和数组多路通道的共同之处：
 - 都是多路通道，在一段时间内能交替执行多个设备的通道程序，使这些设备同时工作

8.5.2 通道的类型

- 字节多路通道和数组多路通道的不同之处：
 - 数组多路通道允许多个设备同时工作，但只允许一个设备进行传输型操作，其他设备进行控制型操作。而字节多路通道不仅允许多个设备同时操作，而且也允许它们同时进行传输型操作。
 - 数组多路通道与设备之间数据传送的基本单位是数据块，通道必须为一个设备传送完一个数据块以后，才能为别的设备传送数据块。而字节多路通道与设备之间数据传送的基本单位是字节，通道为一个设备传送一个字节后，又可以为另一个设备传送一个字节，因此各设备与通道之间的数据传送是以字节为单位交替进行

8.5.2 通道结构的发展

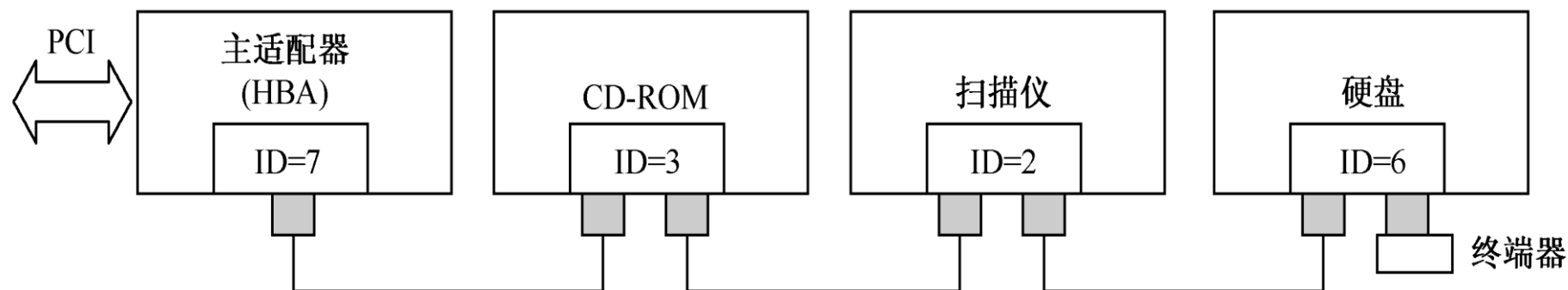
- 通道结构的进一步发展，出现了两种计算机I/O系统结构：
 - 输入输出处理器(IOP) 是通道结构的I/O处理器。IOP可以和CPU并行工作，提供高速的DMA处理能力，实现数据的高速传送。但是它不是独立于CPU工作的，而是主机的一个部件。有些IOP例如Intel 8089 IOP，还提供数据的变换、搜索以及字装配/拆卸能力。这类IOP广泛应用于中小型及微型计算机中
 - 外围处理机(PPU) PPU基本上是独立于主机工作的，它有自己的指令系统，完成算术/逻辑运算，读/写主存储器，与外设交换信息等。有的外围处理机干脆就选用已有的通用机。外围处理机I/O方式一般应用于大型高效率的计算机系统中

8.6 通用I/O接口标准

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



(HBA 内也有终端器)



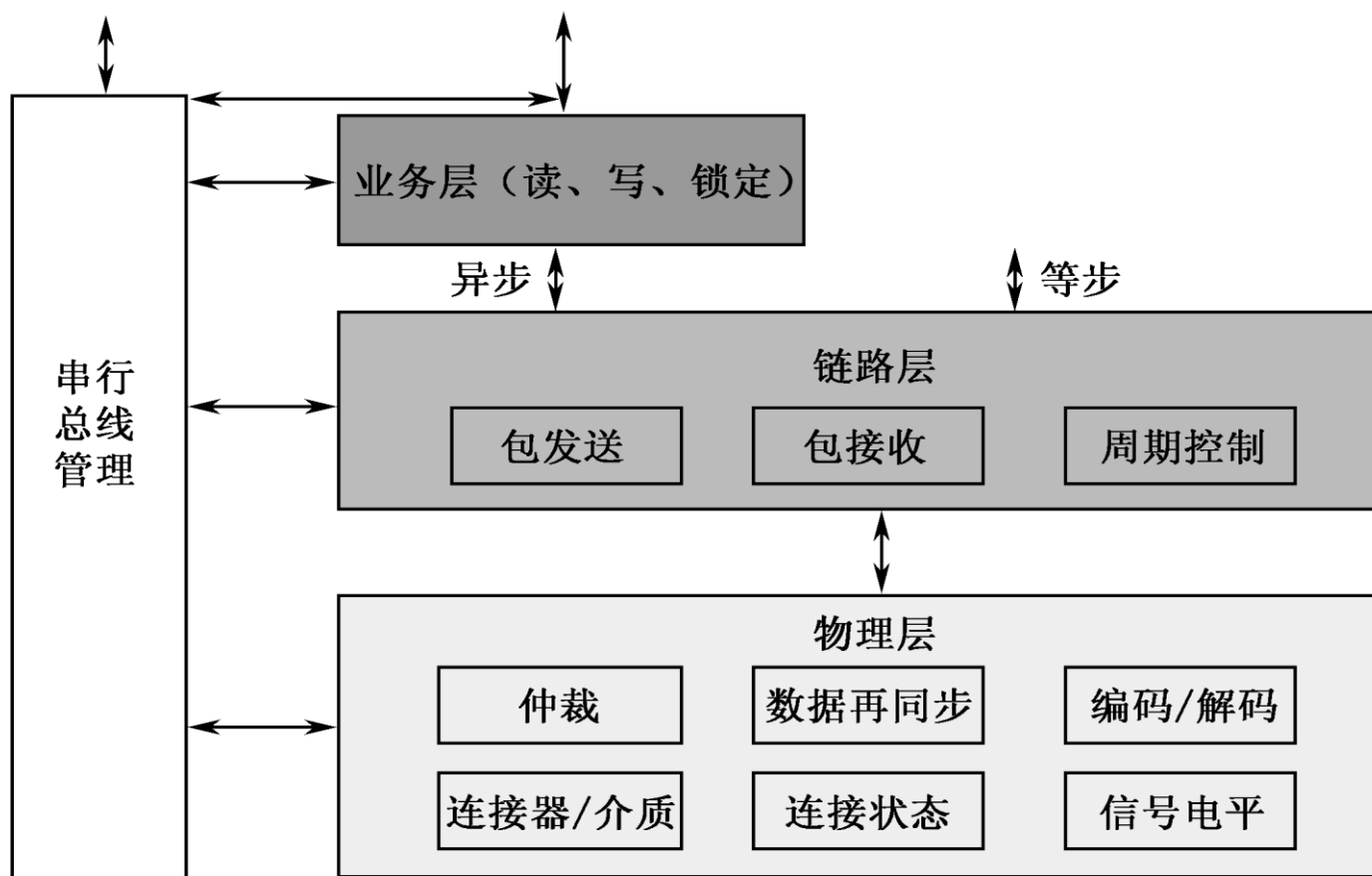
8.6 通用I/O接口标准

■ 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。
- 数据传送的高速性
- 数据传送的实时性
- 体积小易安装，连接方便

8.6 通用I/O接口标准

■ IEEE1394协议集：业务层、链路层和物理层协议集





本章小结

- 在计算机系统中，CPU对外围设备的管理方式有：
 - 程序查询方式；
 - 程序中断方式；
 - DMA方式
 - 通道方式
- 每种方式都需要硬件和软件结合起来进行

本章小结

- 程序查询方式是CPU管理I/O设备的最简单方式，CPU定期执行设备服务程序，主动来了解设备的工作状态。这种方式浪费CPU的宝贵资源

本章小结

- 程序中断方式是各类计算机中广泛使用的一种数据交换方式。当某一外设的数据准备就绪后，它“主动”向CPU发出请求信号。CPU响应中断请求后，暂停运行主程序，自动转移到该设备的中断服务子程序，为该设备进行服务，结束时返回主程序。中断处理过程可以嵌套进行，优先级高的设备可以中断优先级低的中断服务程序

本章小结

- DMA技术的出现，使得外围设备可以通过DMA控制器直接访问内存，与此同时，CPU可以继续执行程序。DMA方式采用以下三种方法：
 - 停止CPU访内
 - 周期挪用
 - DMA与CPU交替访内
- DMA控制器按其组成结构，分为选择型和多路型两类

本章小结

- 通道是一个特殊功能的处理器。它有自己的指令和程序专门负责数据输入输出的传输控制，从而使CPU将“传输控制”的功能下放给通道，CPU只负责“数据处理”功能。这样，通道与CPU分时使用内存，实现了CPU内部的数据处理与I/O设备的平行工作。通道有两种类型
 - 选择通道
 - 多路通道



本章小结

- 标准化是建立开放式系统的基础。CPU、系统总线、I/O总线及标准接口技术近年来取得了重大进步。其中并行I/O接口SCSI与串行I/O接口IEEE 1394是两个最具权威性和发展前景的标准接口技术
- 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
- IEEE 1394是串行I/O标准接口。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