

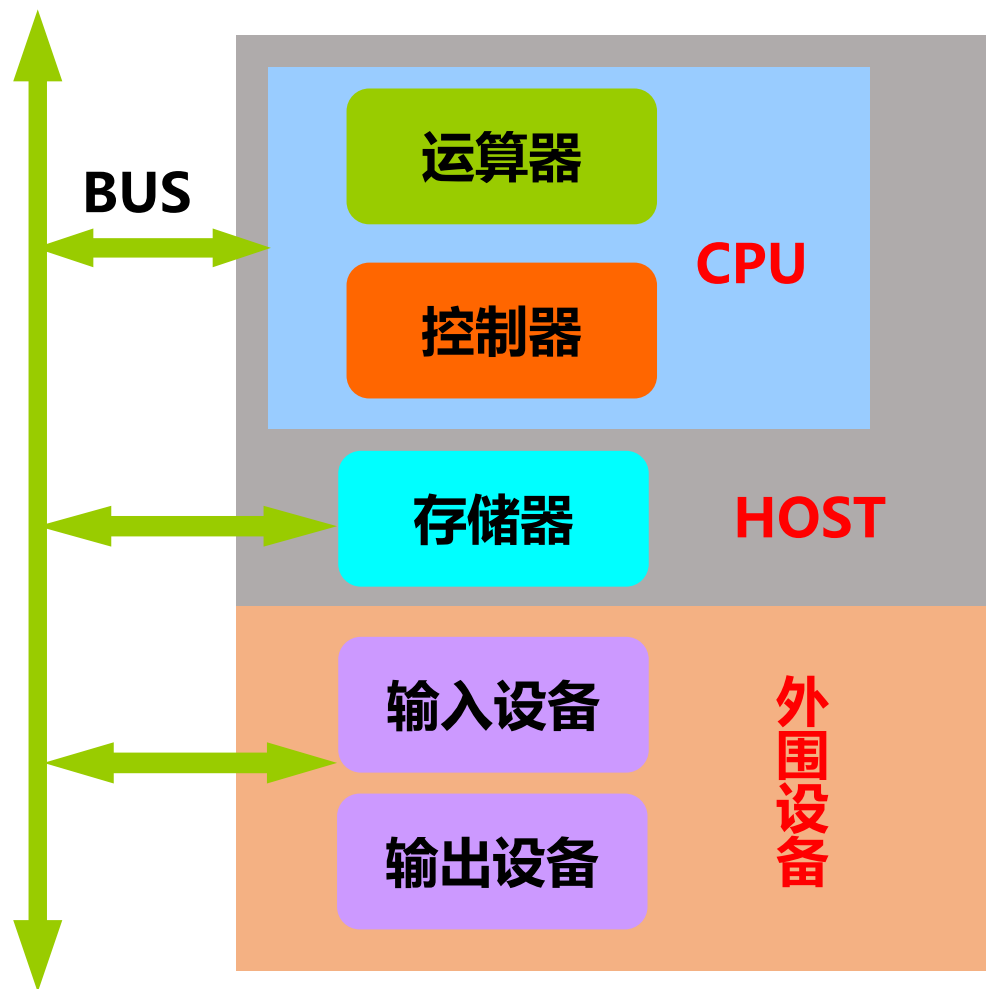


计算机组成原理



八、输入输出原理





■ 主机：

- 中央处理器(CPU)

- 主存储器(MM)

■ 外部设备或外围设备（外设）：

- 主机以外的大部分硬设备

- 包括输入输出设备，外存储器，脱机输入输出设备等

- 输入设备：把数据输入到主机的设备
- 输出设备：内容在主机处理完毕，接受主机内容的设备



本章主要内容

- 9.1 输入输出设备与特性
- 9.2 I/O接口
- 9.3 数据传送控制方式
- 9.4 程序控制方式
- 9.5 程序中断方式
- 9.6 DMA方式
- 9.7 通道方式
- 9.8 常见I/O设备



9.1 输入输出设备与特性

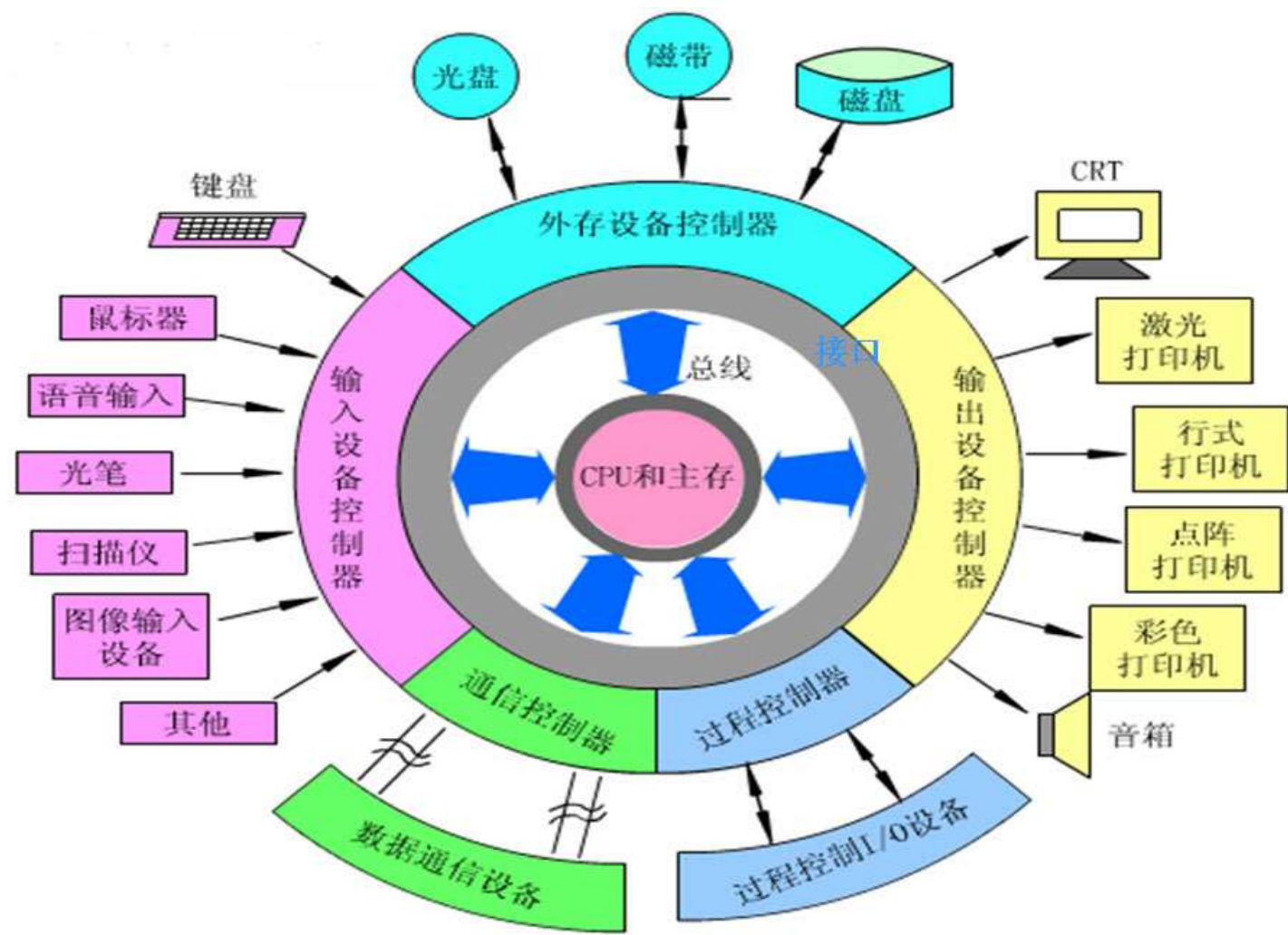
- **输入输出设备**是计算机与人或者机器系统进行数据交互的装置，用于实现计算机内部二进制信息与外部不同形式信息的转换，简称**外部设备或外设**
 - **输入设备**：负责将数据、文字、图像、声音、电信号等转换成计算机可以识别的二进制信息，如键盘、鼠标、扫描仪、摄像头等
 - **输出设备**：负责将计算机处理结果转换成数字、文字、图形、图像、声音或电信号，如显示器、打印机等；
 - **输入输出设备**：既能输入也能输出，如磁盘、网卡等。
- **输入输出设备特性**
 - **异步性、实时性、独立性**

输入/输出系统的组成与功能

- 外部设备、接口部件、总线以及相应的管理软件统称为计算机的输入/输出系统，简称**I/O系统**
 - 完成计算机内部二进制信息与外部多种信息形式间的交流
 - 保证CPU能够正确选择I/O设备并实现对其控制，与数据传输
 - 利用数据缓冲、合适的数据传送方式，实现主机外设间速度匹配
 - I/O硬件
 - ◆ 外设、控制器、I/O接口、总线
 - I/O软件
 - ◆ OS无关库，设备无关库，驱动



输入/输出系统



输入设备



键盘

- 按照结构原理分类

- 触点式键盘

- 无触点式键盘

- 从编码的功能上分类

- 全编码

- 非全编码

- 按照工作原理

- 机械键盘

- 塑料薄膜式键盘

- 导电橡胶式键盘

- 无接点静电电容键盘

键盘的工作原理

- 主机通过扫描键盘识别按下的键
- 扫描键盘
 - 硬件扫描键盘：由硬件逻辑实现
 - 软件扫描键盘：由键盘扫描程序实现

键盘的工作原理

- 键盘是由一组排列成阵列形式的按键开关组成的，每按下一个键，产生一个相应的字符代码(每个按键的位置码)，然后将它转换成ASCII码或其他码，送主机。
- 常用的标准键盘有101个键，它除了提供通常的ASCII字符以外，还有多个功能键(由软件系统定义功能)、光标控制键(上、下、左、右移动等)与编辑键(插入或消去字符)等。

光笔

- 光笔(light pen)的外形与钢笔相似，头部装有一个透镜系统，能把进入的光会聚为一个光点。
- 在光笔头部附有开关，当按下开关时，进行光的检测，光笔就可拾取显示器屏幕上的坐标。
- 光笔与屏幕上的光标配合，可使光标跟踪光笔移动，在屏幕上画出图形或修改图形，这个过程与人用钢笔画图的过程类似。



图形板和画笔（或游动标）

- 画笔(stylus)为笔状，用于图形板(tablet)。
- 图形板是一种二维的A / D变换器(数字化板)
- 当画笔接触到图形板上的某一位置时，画笔在图形板上的位置坐标就会自动传送到计算机中去，随着笔在板上的运动可以画出图形。
- 图形板和画笔结合构成二维坐标的输入系统，主要用于输入工程图等。
- 将图纸贴在图形板上，画笔沿着图纸上的图形移动，读取图形坐标，即可输入工程图。



游动标

- 为了提高读图精度，常用游动标(cursor)代替画笔与图形板配合使用。
- 游动标是一个手持的方形坐标读出器，游动标上有一块透明玻璃，玻璃上刻有十字标记。
- 十字标记的中心就是游动标的中心。
- 使用时将十字中心对准图形的坐标点上，它比画笔读取的坐标更精确。

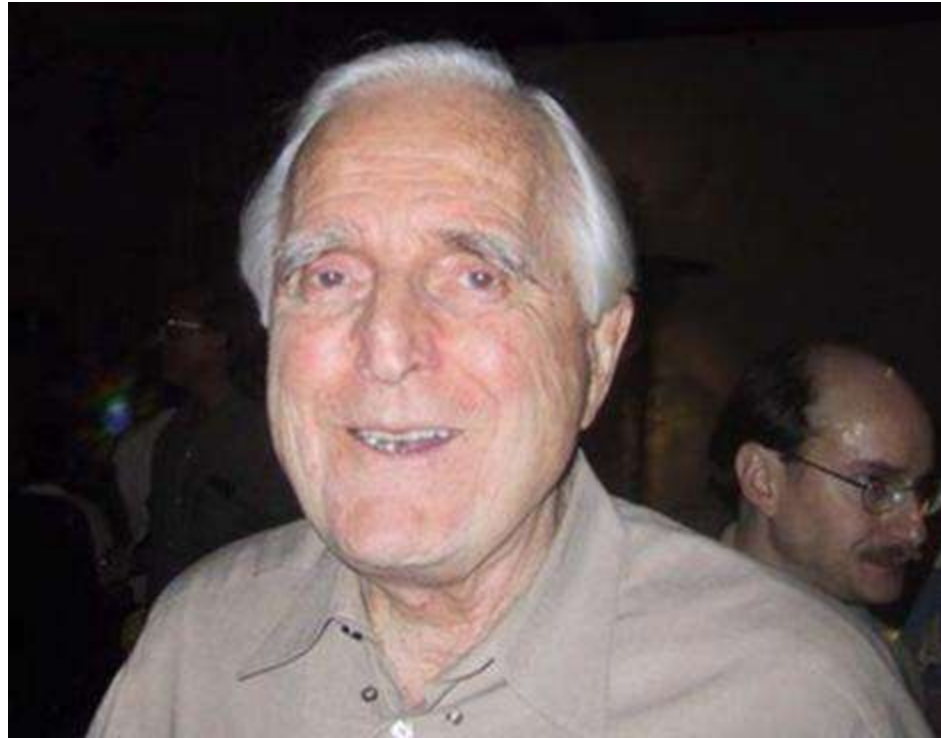
鼠标器、跟踪球和操作杆

- 鼠标器、跟踪球和操作杆输入相对坐标。
 - 它们必须和显示器的光标配合。
 - 计算机先要给定光标的初始位置，然后用读取的相对位移移动光标。
- 鼠标器、跟踪球和操作杆操作容易、制作简单而且造价低，但定位精度都比较差。



鼠标器

- 1964年美国科学家道格拉斯·恩格巴特(Douglas Eglebart)博士发明了鼠标。
- 直到1968年12月的美国秋季计算机会议上，他向与会者展示了他的新发明。



尊敬的鼠标之父——道格·恩格尔巴特博士

鼠标器

- 恩格尔巴特于**1925年1月30日**出生在俄勒冈州波特兰市附近的一个小农场，**1942年**，他在俄勒冈州立大学学习电气工程。
- 二战期间，恩格尔巴特中断了学业去参军，在**Phillipines**作为一名电子雷达兵服了两年兵役。
- **1948年**，拿到电气工程学士学位后，他留在旧金山半岛**NACA Ames** 实验室（**NASA**美国国家航天局的前身）做一名电子工程师。

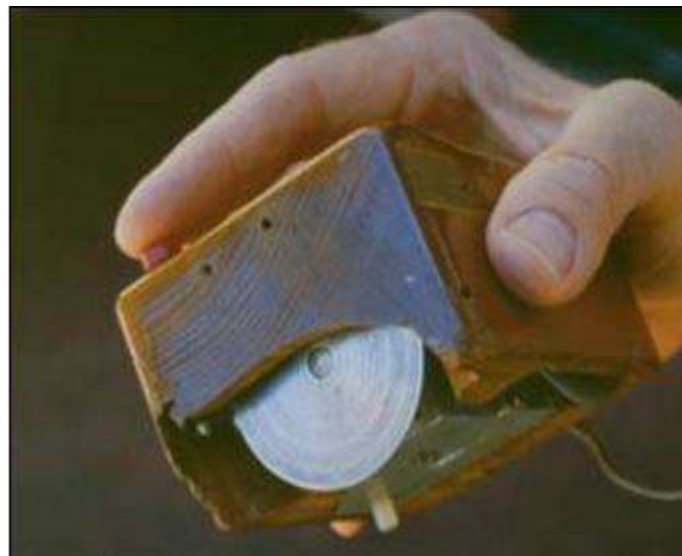
鼠标器

- 在1946年世界上第一台电子计算机诞生以后，那时电脑功能还远没有现在这么完善。这些价格昂贵的奢侈品主要用途也是用于科研、军事等领域。尽管如此，作为一个富有远见卓识的发明家，恩格尔巴特博士已经意识到，随着电脑的发明以及普及，在人类历史的发展进程中具有无法估量的重大意义，鼠标自然也有可能被广泛应用，所以他发明鼠标，**并且最初起名叫“显示系统X-Y位置指示器”**。这个像老鼠一样拖着一条长长尾巴连线的装置，被恩格尔巴特博士和他的同事戏称为“**Mouse**”，也许是“**Mouse**”这个名字简洁而且形象生动，所以“鼠标”的称呼一直被流传下来，这就是鼠标名称的来历。
- 1997年人们为了表彰恩格巴特在人机交互方面的卓越贡献，**授予他计算机界最高荣誉——图灵奖**。鼠标的发明，也曾被**IEEE（全球最大的专业技术学会——美国电气与电子工程师学会）**列为计算机诞生**50**年来最重大的事件之一，可见其对IT历程的重大影响作用。

鼠标器



最初的鼠标外观



最初的鼠标外观

触摸屏

- 透明的，安装在显示器屏幕的外面(表面)。
- 触摸屏系统包括 触摸屏控制器(卡)和触摸检测装置。
 - 触摸屏控制卡上有微处理器和固化的监控程序，将触摸检测装置送来的触摸信息转换成触点坐标，送给计算机；同时它能接收计算机送来的命令，并予以执行。
 - 触摸屏的控制卡及其电源可以安装在显示器内，称为内置式；也可放在显示器外面，称为外挂式。
- 技术分类：电阻式、电容式、红外线式、表面声波技术和底坐式矢量压力测力技术。



图像输入设备

- 摄像机与摄像头
 - 摄像机摄取的景物，经数字化后变成数字图像存入闪存、磁盘或磁带



图像输入设备

- 数字照相机
 - 数字照相机内置有存储器，并带用LCD预映屏幕。
 - 在拍照时观察屏幕可将最佳快照录入相机的存储器中，并即时删除不理想的照片。

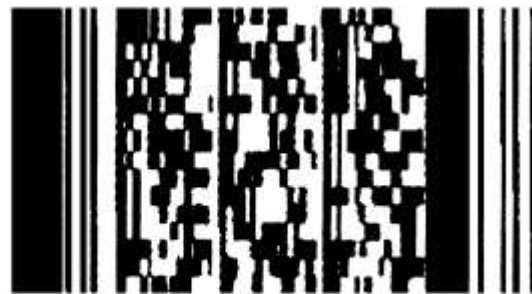


条形码

- 条形码又叫条码。条码的定义是：由一组宽度和反射率不同的平行相邻的“条”和“空”，按照预先规定的编码规则组合起来，用以表示一组数据的符号。这组数据可以是数字、字母或某些符号。



(a) 一维条码



(b) 二维条码

光学字符识别(OCR)技术

- 西方国家普遍使用打字机，因此计算机识别的对象多是印刷体英文、数字、符号。因此，在计算机中的扫描仪都配备了OCR（Optical Character Recognition,）软件，用户可以方便的将英文文本送入计算机。
- OCR识别是指对文本资料的图像文件进行分析识别处理，获取文字及版面信息的过程。亦即将图像中的文字进行识别，并以文本的形式返回。
- OCR系统是多项技术结合的产物，识别技术是其核心内容，还包括图形文本的扫描输入，光电信号变换，电信号的数字化处理，版面分析与理解，字的切分处理以及输入信息载体(页)的自动传送技术等。
- 典型的OCR的技术路线如下图所示



光学字符识别(OCR)技术

- 其中影响识别准确率的技术瓶颈是文字检测和文本识别，而这两部分也是OCR技术的重中之重。
- 在传统OCR技术中，图像预处理通常是针对图像的成像问题进行修正。常见的预处理过程包括：几何变换（透视、扭曲、旋转等）、畸变校正、去除模糊、图像增强和光线校正等
- 文字检测即检测文本的所在位置和范围及其布局。通常也包括版面分析和文字行检测等。文字检测主要解决的问题是哪里有文字，文字的范围有多大。
- 文本识别是在文本检测的基础上，对文本内容进行识别，将图像中的文本信息转化为文本信息。文字识别主要解决的问题是每个文字是什么。识别出的文本通常需要再次核对以保证其正确性。

语音与文字输入系统

目前计算机主要以键盘输入为主，未来研究如何让机器听懂话和识别文字。因此产生了一些对应的学科，包括文字识别、自然语言处理、计算机视觉等。

语音与文字输入系统的基本概念

- 语音与文字输入
 - 要让计算机从语音的声波和文字的形狀中领会到含义，并将它转换成计算机可以处理的代码。
 - 信息获取、预处理、特征提取、给出识别结果
- 汉字识别
 - 脱机手写汉字识别、联机手写汉字识别、印刷体汉字识别
- 语言识别
 - 自动语言识别、语言合成、语言理解、自动电话查询和翻译

显示设备

- 显示设备：以可见光的形式传递和处理信息的设备



显示技术中的相关术语

- 图形(graphics): 最初是指没有亮暗层次变化的线条图, 如建筑、机械所用的工程设计图、电路图等。早期的图形显示和处理只是局限在二值化的范围, 只能用线条的有无来表示简单的图形。
- 图像(image): 则最初就是指具有亮暗层次的图, 如自然景物、新闻照片等。经计算机处理后显示的图像称作数字图像, 就是将图片上连续的亮暗变化变换为离散的数字量, 并以点阵列的形式显示输出。

显示技术中的相关术语

- 在显示屏幕上，图形和图像都是由称作像素的光点组成的。
- 光点的多少称作分辨率，光点的深浅变化称作灰度级(在黑白显示器上表现为灰度级，在彩色显示器上表现为颜色)。
- 分辨率和灰度级决定了所显示图的质量。
- 高分辨率多灰度级的光栅扫描的显示器不仅可以显示图像，也可以显示图形。

显示技术中的相关术语

- 分辨率和灰度级是显示器的两个重要技术指标。
- 分辨率(resolution)指的是显示设备所能表示的像素个数。像素越密，分辨率越高，图像越清晰。分辨率取决于荧光粉的粒度，屏的尺寸和电子束的聚焦能力。
- 灰度级(gray level)指的是所显示像素点的亮暗差别，在彩色显示器中则表现为颜色的不同。灰度级越多，图像层次越清楚逼真。
 - 4位一个像素：16级灰度或颜色

显示技术中的相关术语

- 根据不同的分辨率，有不同的显示器接口(或称为适配器)与之配合，IBM公司制定了显示器分辨率的标准，并被业界所接受

表 9.2 各种显示模式的分辨率

显示模式	CGA	EGA	VGA	SVGA	XGA	SXGA
分辨率	640×200	640×350	640×480	800×600	1024×768	1280×1024

本章主要内容

- 9.1 输入输出设备与特性
- **9.2 I/O接口**
- 9.3 数据传送控制方式
- 9.4 程序控制方式
- 9.5 程序中断方式
- 9.6 DMA方式
- 9.7 通道方式
- 9.8 常见I/O设备



I/O接口定义与功能

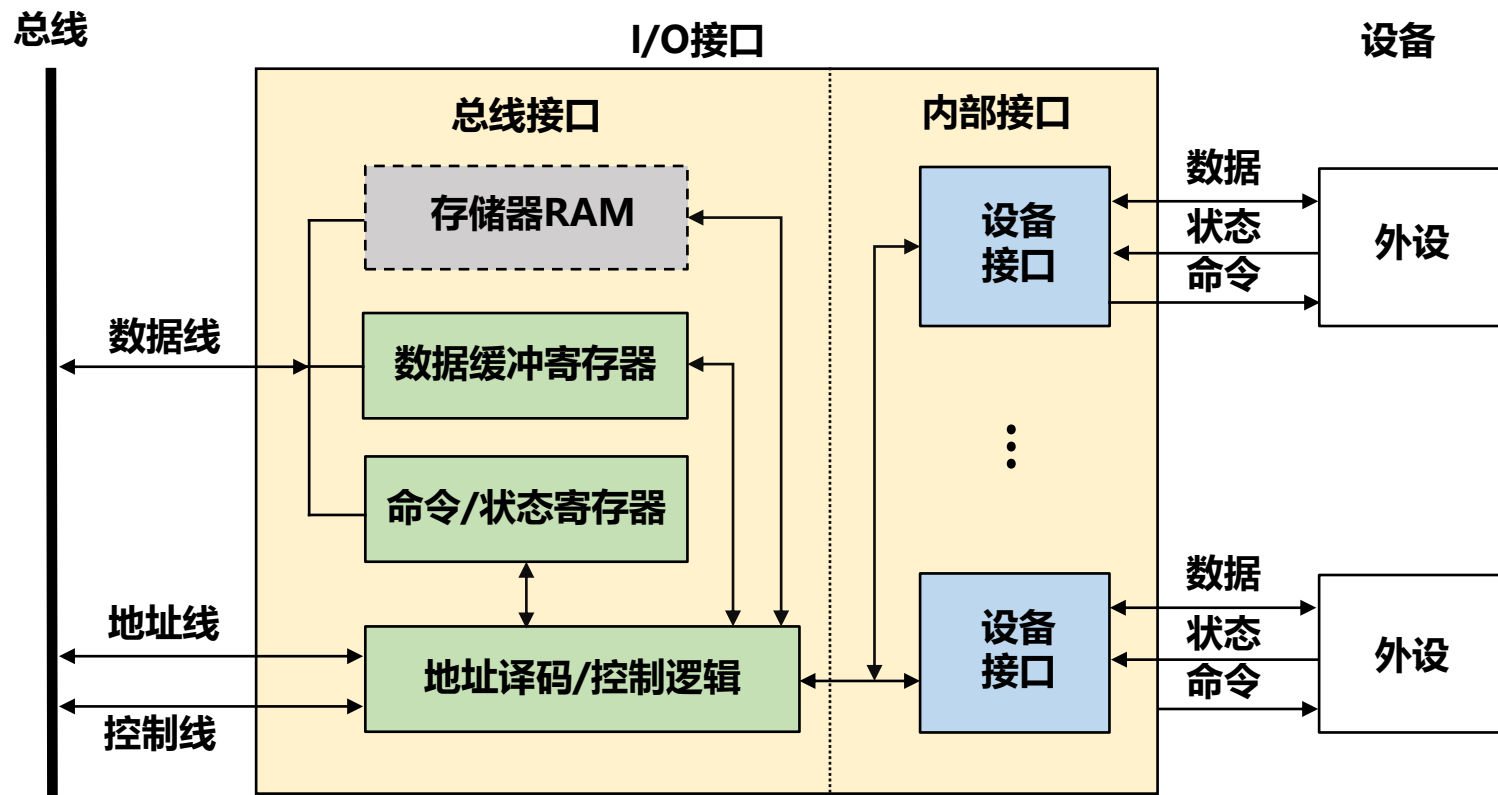
■ I/O接口: 连接总线与I/O设备的物理和逻辑界面

- 既包括**物理连接电路**，也包括**软件逻辑接口**
- 所有设备均通过 I/O接口（总线接口）与总线相连
- CPU使用设备地址经总线与I/O接口通信访问I/O设备
- **标准接口**有利于提升I/O系统的独立性，降低连接复杂度

■ I/O接口功能

- **设备寻址、数据交互、设备控制**
- **状态检测、数据缓冲、格式转换**

I/O接口结构



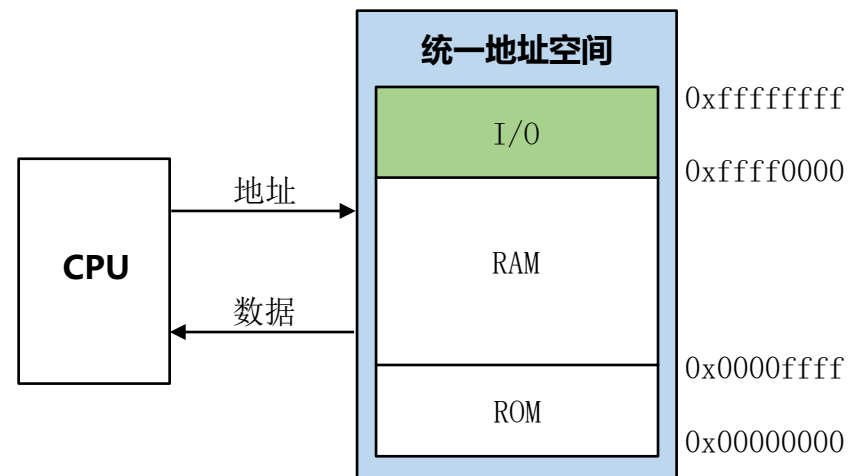
I/O接口的功能

- 设备寻址
- 数据交互
- 设备控制
- 状态检测
- 数据缓冲
- 格式转换

I/O接口编址

■ 统一编址

- 内存映射编址 (Memory-mapped)
- 外设地址与内存地址统一编址，同一个地址空间
- 不需要设置专用的I/O指令
- 采用访存指令访问外设，具体访问什么设备取决于地址



■ 独立编址

- 端口映射编址 (Port-mapped)
- I/O地址空间与主存地址空间相互独立
- I/O地址又称为I/O端口
- 不同设备中的不同寄存器和存储器都有唯一的端口地址
- 使用I/O指令访问外设

I/O接口的软件

- 现代计算机中用户必须**通过操作系统间接访问设备**，屏蔽设备细节，使用更方便

- **与OS无关的I/O库（用户态）**

如C语言中的标准I/O库stdio.h,

printf、scanf、getchar、putchar、fopen、fseek、fread、fwrite、fclose等

用户程序主要通过调用I/O库访问设备，方便程序在不同OS间移植

- **与设备无关的OS调用库（内核态）**

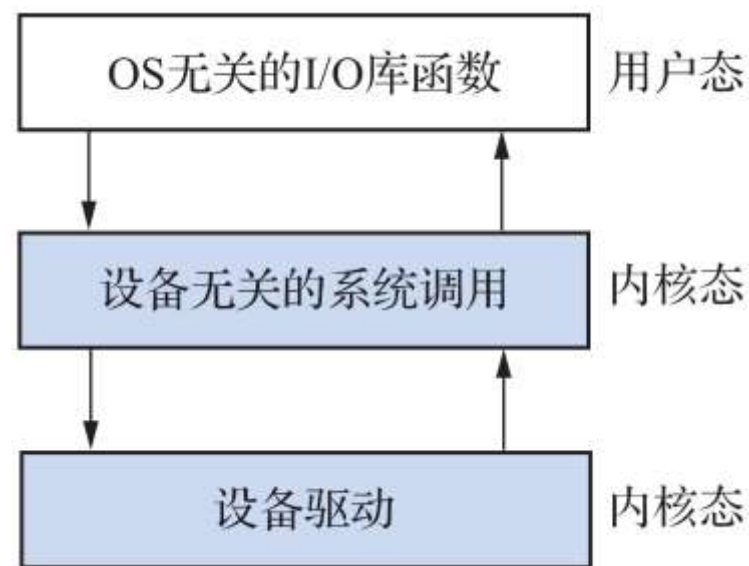
open、read、write、seek、ioctl、close

- **独立的设备驱动程序（内核态）**

设备驱动程序是与设备相关的I/O软件部分

不同设备对应不同的驱动程序

遵循具体设备的I/O接口约定，包含设备接口细节



I/O接口分类

- 按数据传送方式

 - 并行、串行接口

- 按接口的灵活性

 - 可编程接口、不可编程接口

- 按通用性： 通用、专用接口

- 按总线传输的通信方式： 同步、异步接口

- 按访问外设的方式

 - 直接传送方式、程序控制方式、程序中断方式、DMA及通道处理机接口

本章主要内容

- 9.1 输入输出设备与特性
- 9.2 I/O接口
- **9.3 数据传送控制方式**
- 9.4 程序控制方式
- 9.5 程序中断方式
- 9.6 DMA方式
- 9.7 通道方式
- 9.8 常见I/O设备



数据传输控制方式

■ 程序查询方式

- 程序控制方式是指输入输出完全依靠CPU执行程序实现

■ 程序中断方式

- 当外部设备准备好后主动向CPU发送中断请求，当前CPU进程放入等待队列并转去执行其他进程

■ 直接内存访问方式DMA

- 由DMA控制器临时代替CPU控制总线，控制设备和内存之间进行直接的数据交换，信息传送不再经过CPU寄存器中转

■ 通道方式

- 通道拥有独立的通道指令系统，可以通过执行通道程序来完成CPU 指定的I/O任务

■ 外围处理机方式

- 通道方式的进一步发展

本章主要内容

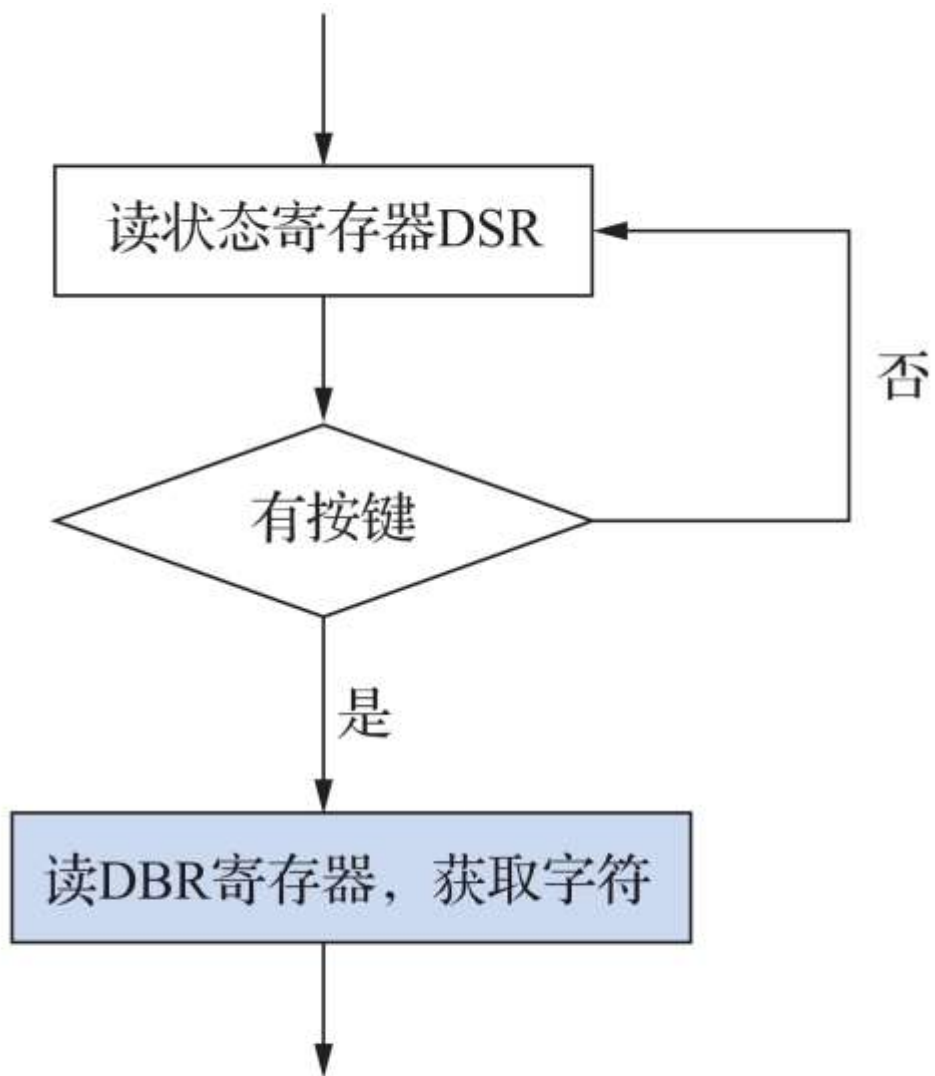
- 9.1 输入输出设备与特性
- 9.2 I/O接口
- 9.3 数据传送控制方式
- **9.4 程序控制方式**
- 9.5 程序中断方式
- 9.6 DMA方式
- 9.7 通道方式
- 9.8 常见I/O设备



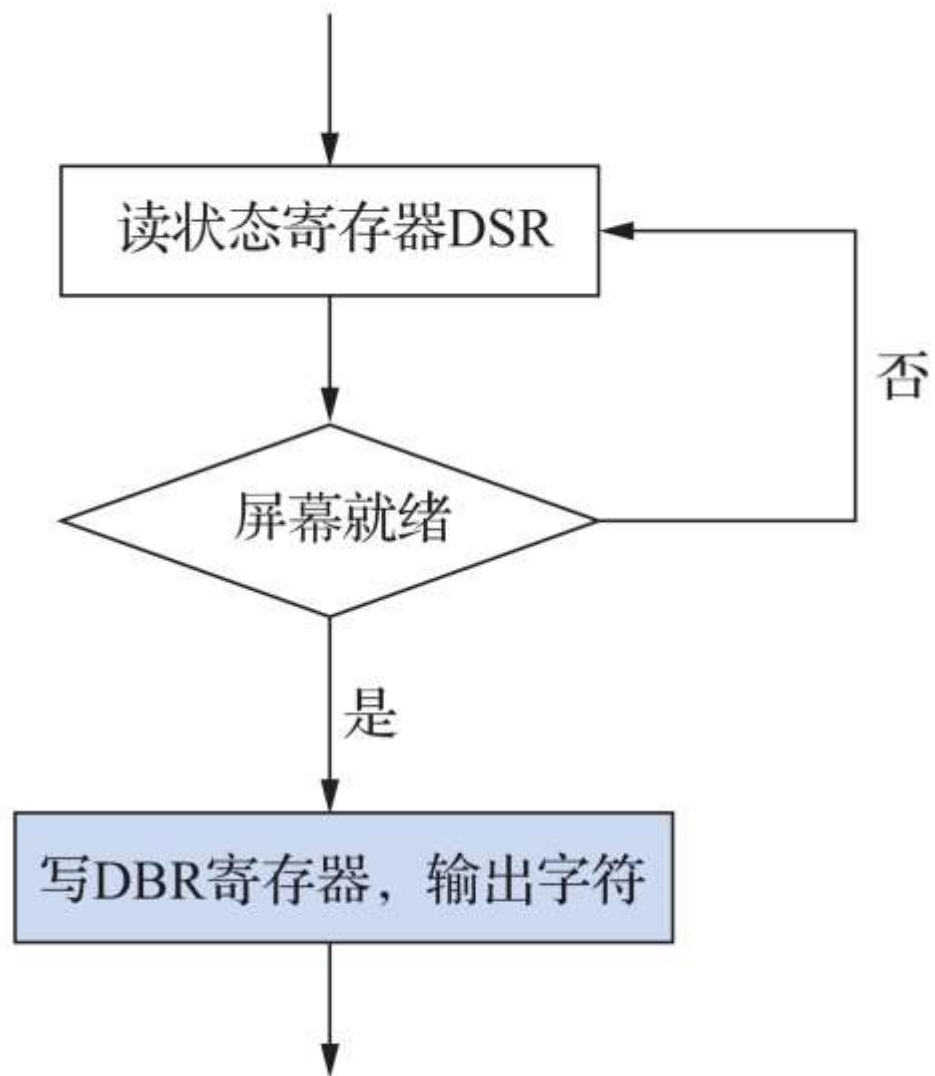
程序控制方式

- 执行一段输入输出程序来实现CPU与外部设备的数据交换
 - 程序查询：又称为轮询方式(Polling)，每次传送前都要查询设备状态，只有当设备准备就绪后才可进行后续操作；
 - ◆ 忙等待(Busy-waiting)
 - ◆ 定时轮询(Polling)
 - 直接传送：无须查询设备状态即可与设备进行数据交互

简单设备查询流程

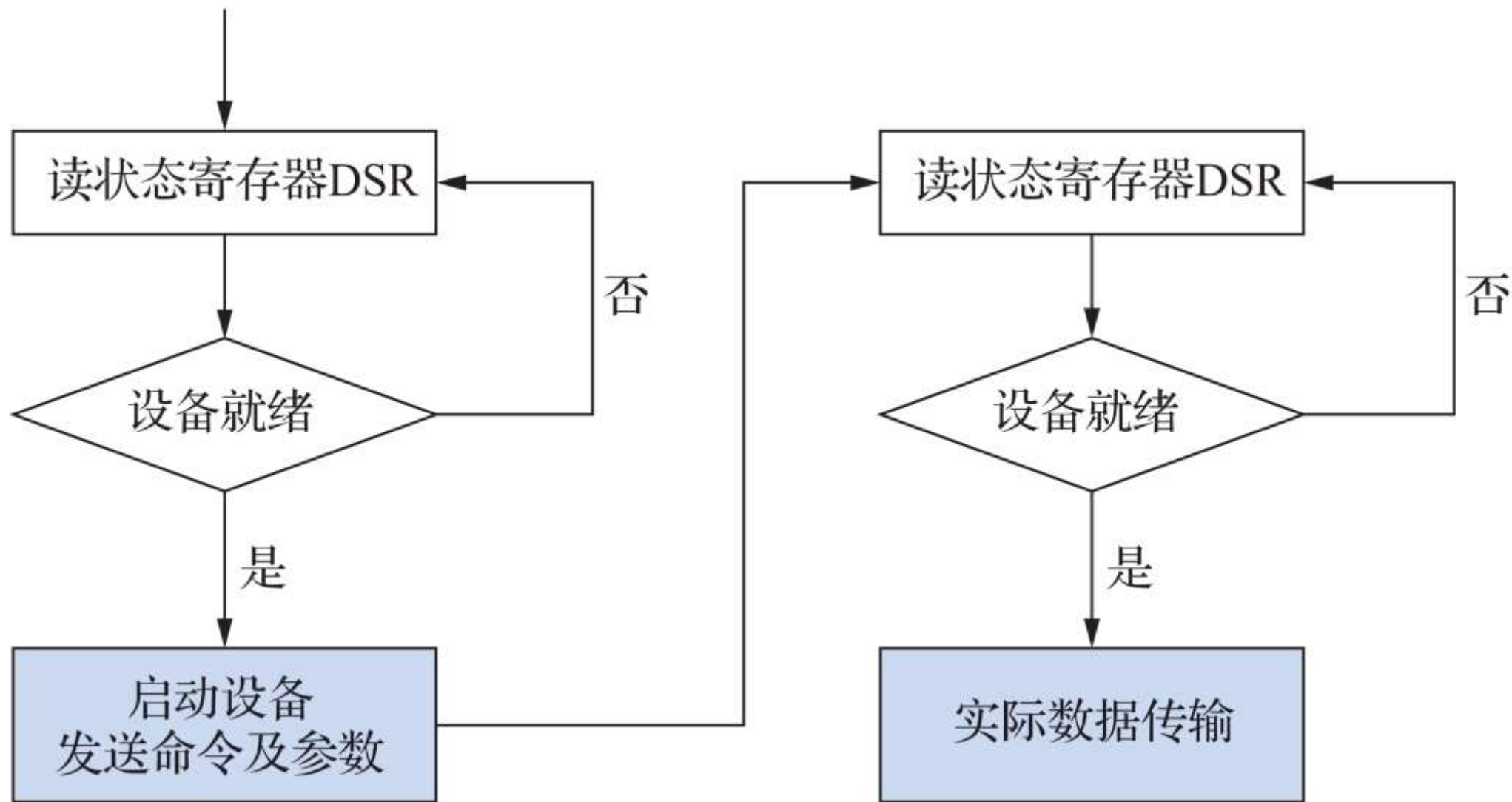


(a) 键盘程序查询流程

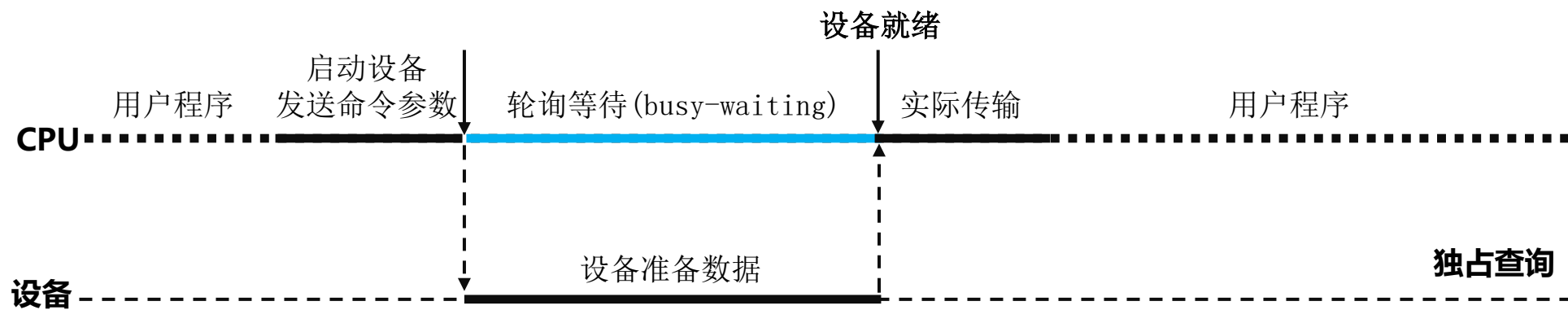


(b) 字符终端程序查询流程

复杂设备查询流程

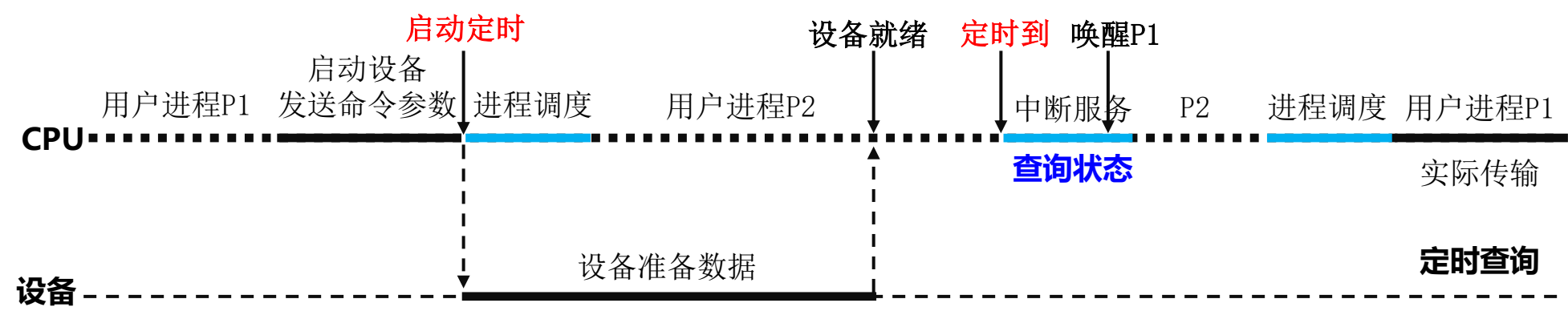


忙等待的程序查询方式运行轨迹



- CPU不停地反复查询设备状态直至设备就绪，这个阶段CPU 不能执行其他任务，一旦设备就绪后CPU即可查询感知。
- CPU 浪费了大量的时间进行轮询操作，通常在单任务操作系统中可采用这种方式

定时轮询的程序查询方式运行轨迹



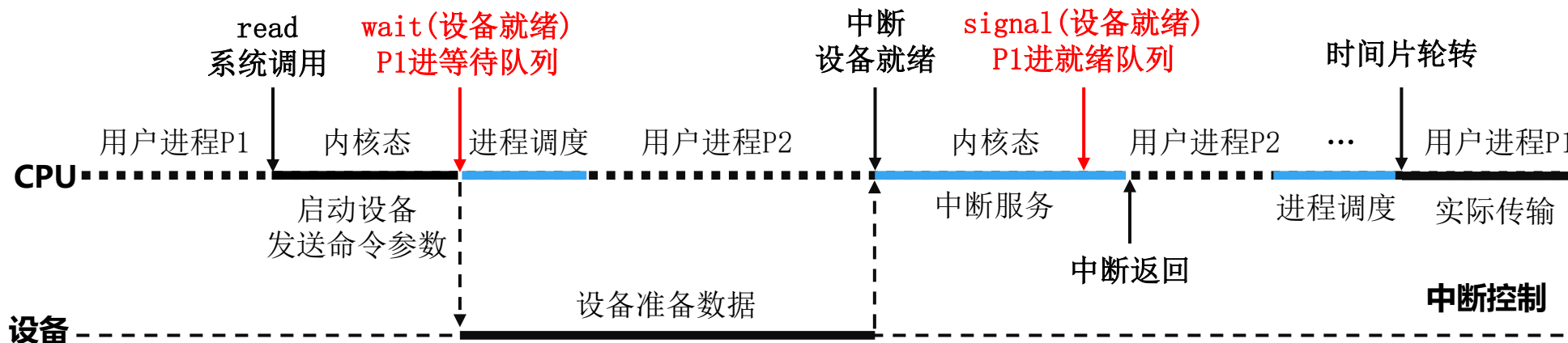
- CPU启动设备后会启动一个定时中断;然后挂起当前用户进程P1并放入I/O等待队列，调度用户进程P2运行

本章主要内容

- 9.1 输入输出设备与特性
- 9.2 I/O接口
- 9.3 数据传送控制方式
- 9.4 程序控制方式
- **9.5 程序中断方式**
- 9.6 DMA方式
- 9.7 通道方式
- 9.8 常见I/O设备



中断控制方式



- **用户系统调用:** 用户进程P1通过系统调用read()函数从设备读取数据
- **驱动启动设备:** 由设备驱动程序负责将读命令以及相关参数通过I/O接口发送给I/O设备
- **操作系统进程调度:** 进程P1放入I/O等待队列, 并通过进程调度进行上下文切换, 调度用户进程P2运行, P2的执行和设备准备阶段是并行
- **设备中断请求:** 当设备准备好数据后, 立刻主动向CPU 发出中断请求, 告知CPU数据准备就绪, 主动中断请求是中断控制方式的核心, 可以避免不必要的轮询
- **CPU 中断服务:** CPU暂停当前进程的执行, 转去执行设备中断服务程序
- **CPU恢复运行:** 返回断点继续执行

中断优势

- 提高了CPU的使用效率

- 主动告知机制避免了反复查询设备状态

- 仍需CPU占用（中断服务子程序运行时间+中断开销）

- 适合随机出现的服务

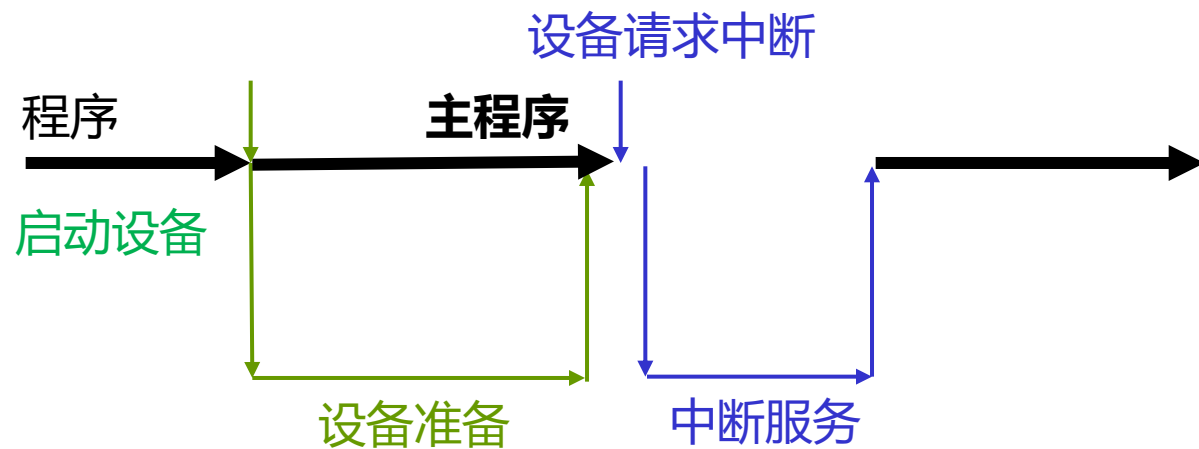
- 需要专门的硬件

程序中中断方式

- 中断基本概念
- 程序中中断基本接口
- 中断仲裁方式
- 中断控制器

中断基本概念

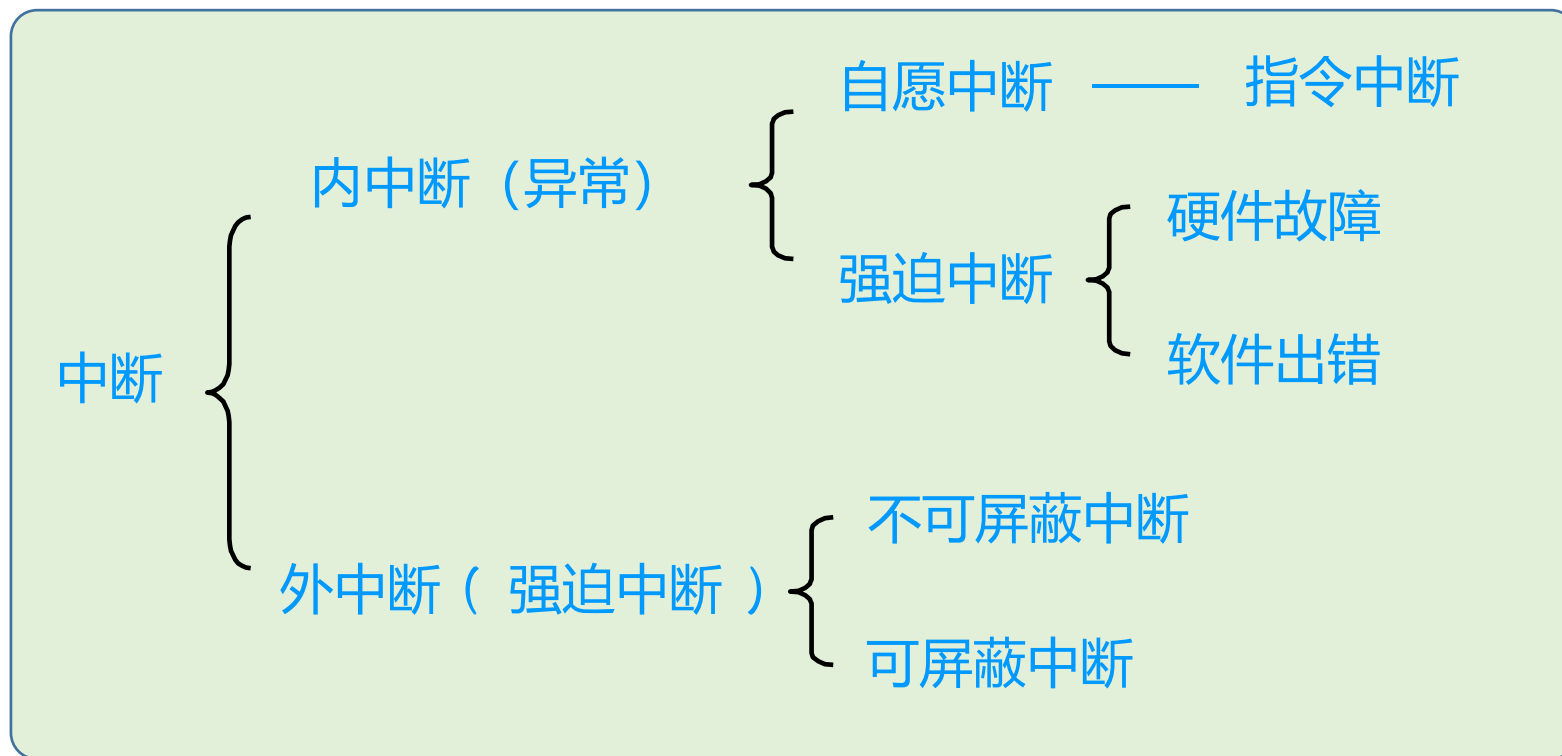
- CPU暂时中止现程序的执行，转去执行为某个**随机事件**服务的中断处理子程序，处理完后自动恢复原程序的执行
- 实现主机和外设准备阶段的并行工作
 - 避免重复查询外设状态、提升工作效率



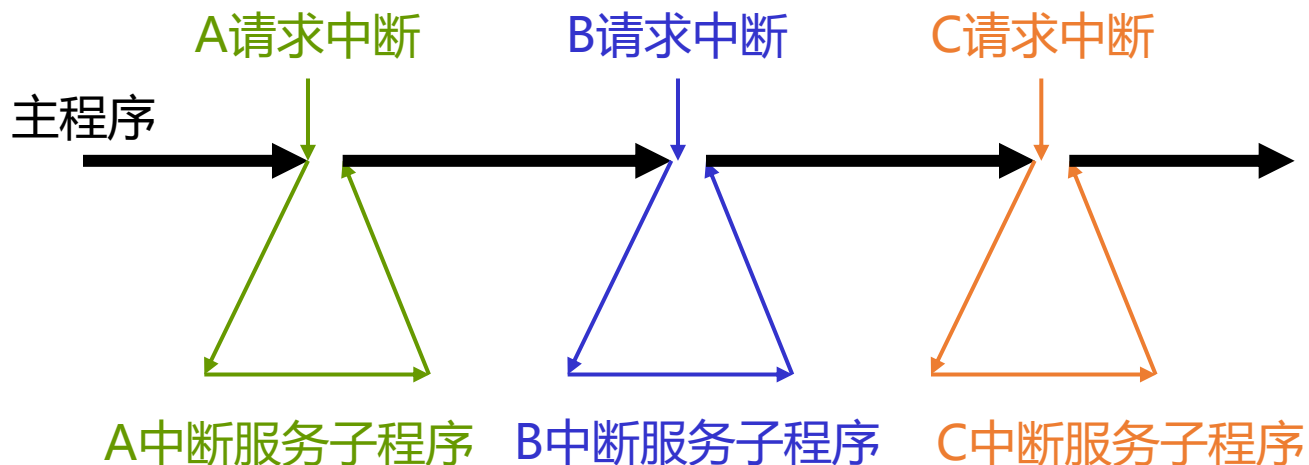
中断的分类与作用

- 中断技术赋予计算机应变能力，将有序的运行和无序的事件统一起来，大大增强了系统的处理能力

- 主机外设并行工作
- 程序调试
- 故障处理
- 实时处理
- 人机交互



程序中中断处理示意图

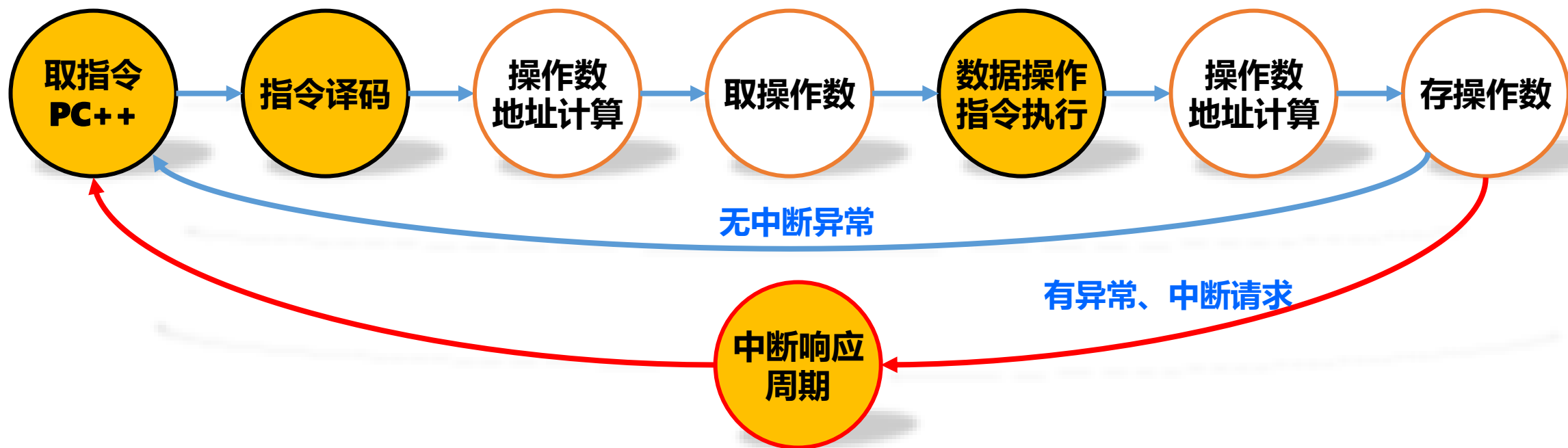


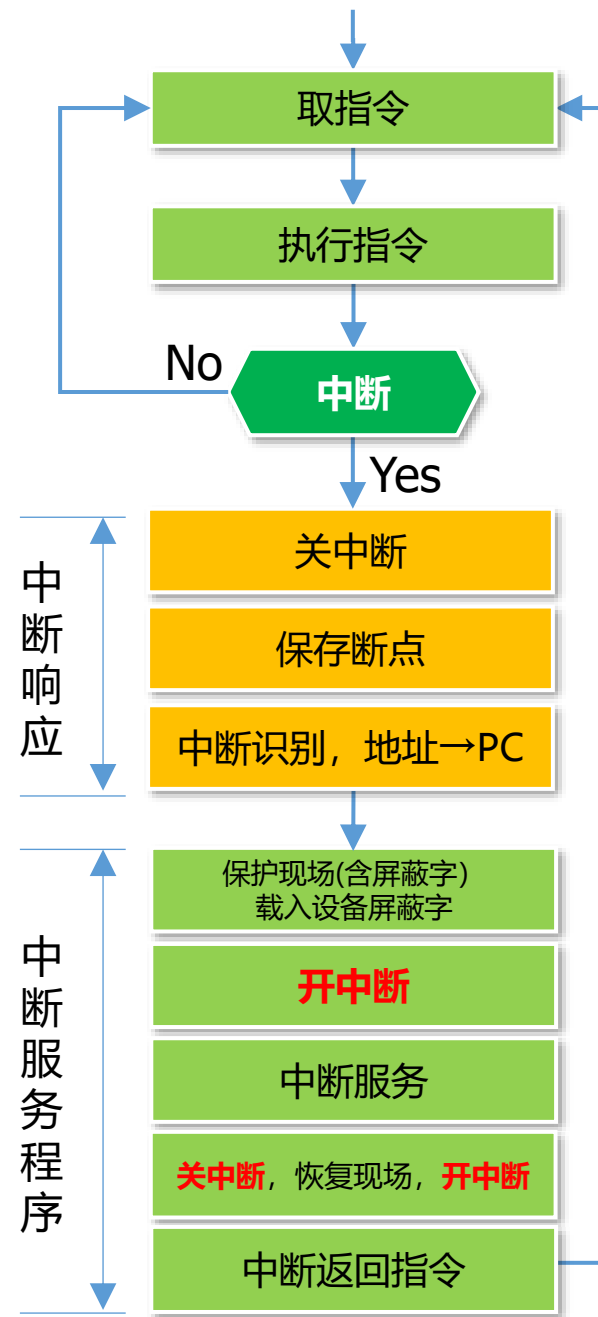
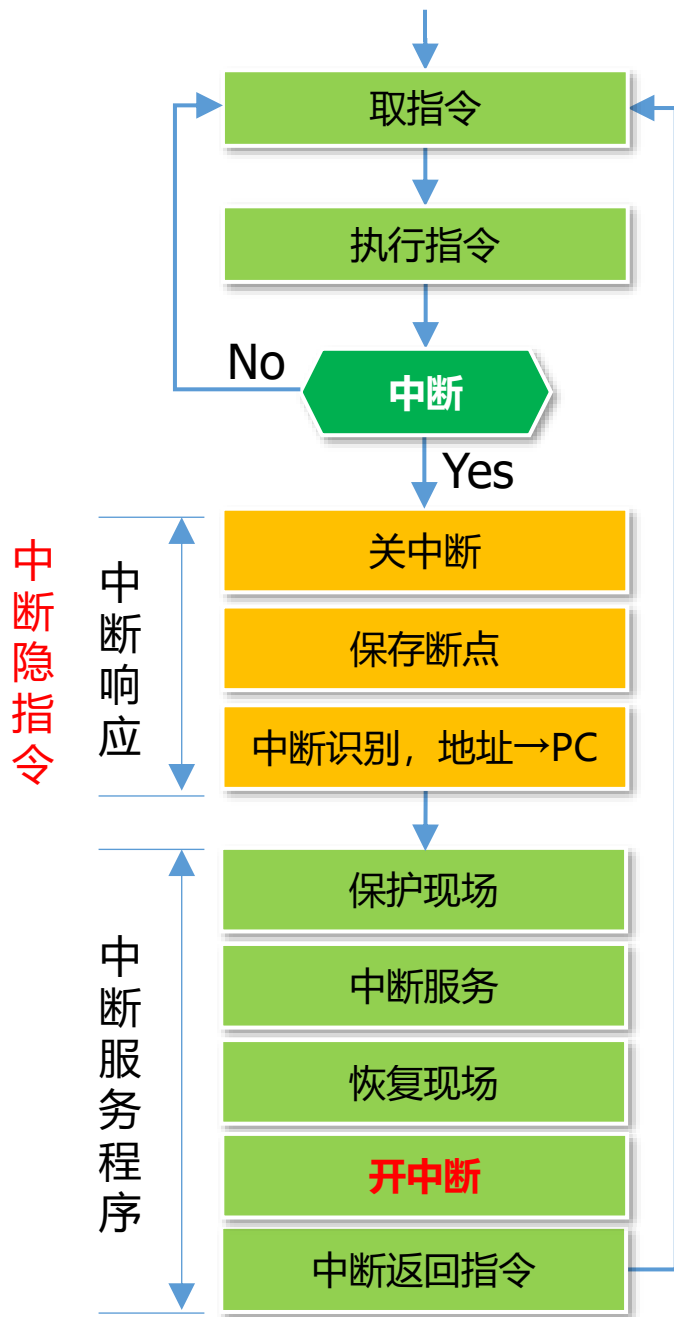
- 子程序与中断服务子程序的区别？
 - 子程序在特定位置显式调用，后者随机调用，现场不同？
- 如果A，B，C同时产生中断？
 - 中断优先级问题，中断仲裁
- 如果正在运行A中断服务子程序，又收到B中断？
 - 中断嵌套

什么是主程序？

指令执行一般流程

- 取指令、执行指令反复循环
- 指令功能、寻址方式不同，数据通路不同，执行时间不同，**如何安排时序？**
- **访存指令、寄存器运算指令、加法指令、除法指令**





中断优先级

- 多设备同时产生中断请求时，如何处理？
 - 优先级高的先响应，优先级低的后响应
 - CPU优先级随不同中断服务程序而改变
 - ◆ 执行某设备中断服务子程序
 - ◆ CPU优先级就与该设备的优先级一样

单级中断与多级中断

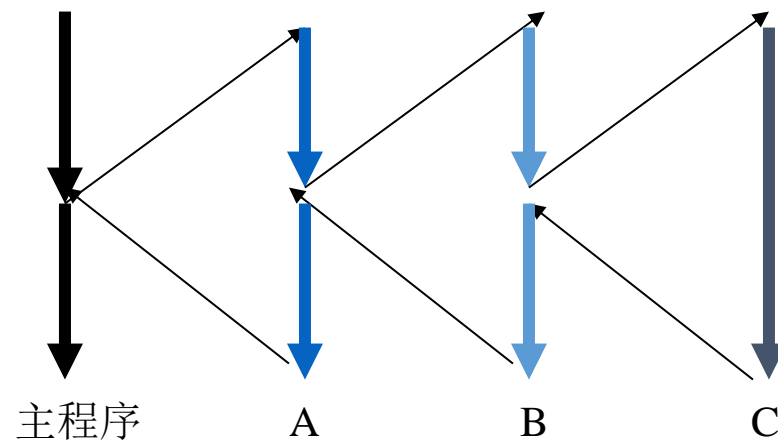
- 高优先级中断请求能否中断运行中的程序呢？
- 系统硬件、软件开销的权衡

□ 单级中断

- ◆ 所有中断源均属同一级，离CPU近的优先级高
- ◆ CPU处理某个中断时，不响应其他中断

□ 多重中断

- ◆ 优先级高的中断可以打断优先级低的中断服务程序
- ◆ 中断嵌套



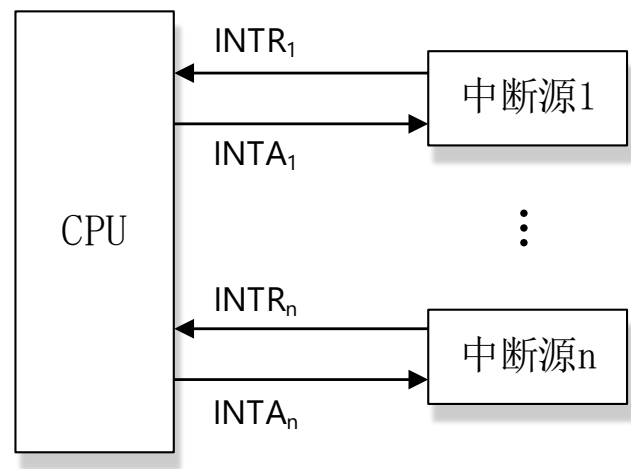
划分优先级的一般规律

- 硬件故障中断属于最高级，其次是程序错误中断
- 非屏蔽中断优于可屏蔽中断
- DMA请求优先于I/O设备传送的中断请求
- 高速设备优于低速设备
- 输入设备的中断优于输出设备
- 实时设备优先于普通设备

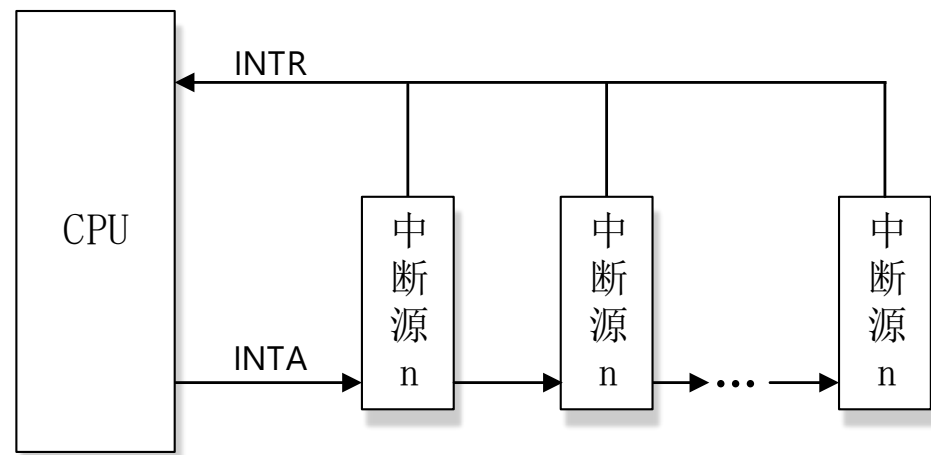
优先级实现---中断仲裁

- 同一时刻可能有多个设备同时发出中断请求，响应谁？
 - 独立请求
 - 链式查询
 - 中断控制器方式
 - 分组链式结构

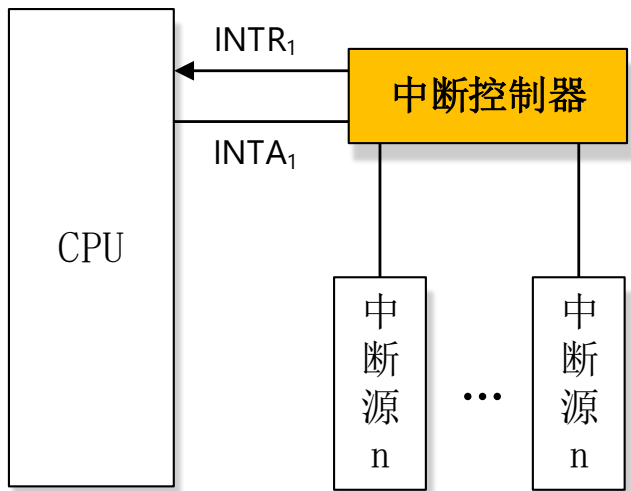
中断请求信号的传输方式



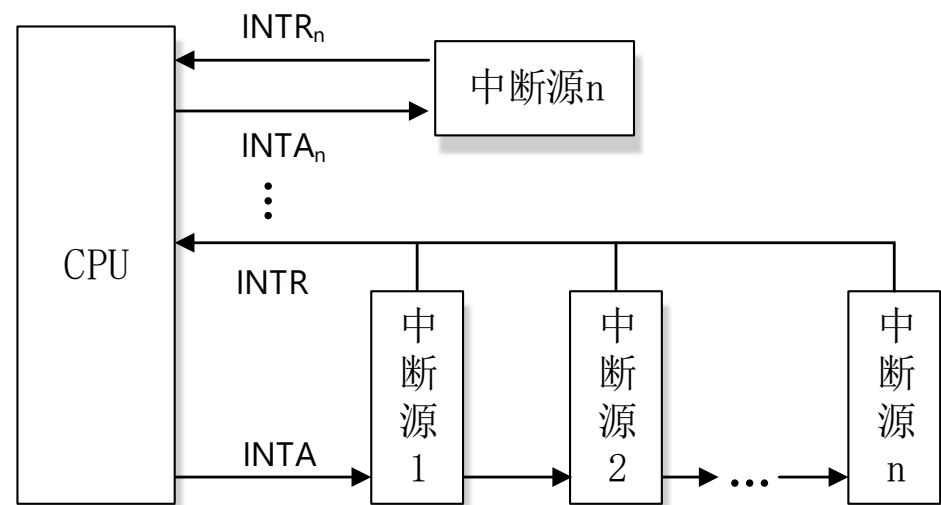
(a) 独立请求方式



(b) 链式请求方式



(c) 中断控制器方式



(d) 分组链式请求

本章主要内容

- 9.1 输入输出设备与特性
- 9.2 I/O接口
- 9.3 数据传送控制方式
- 9.4 程序控制方式
- 9.5 程序中断方式
- **9.6 DMA方式**
- 9.7 通道方式
- 9.8 常见I/O设备



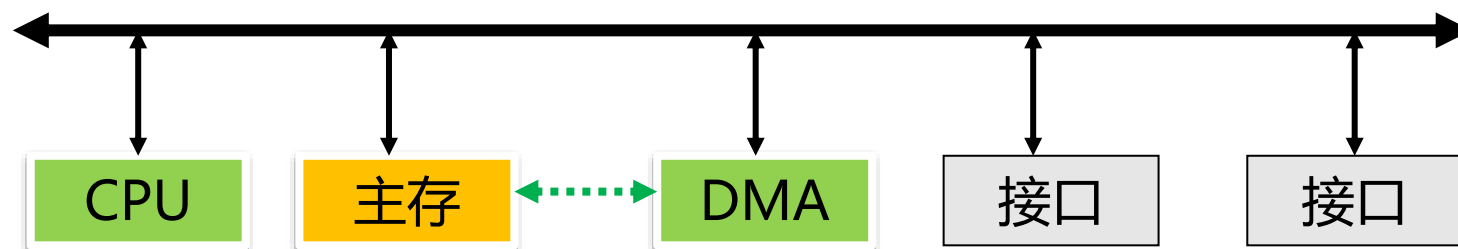
DMA基本概念

■ 中断方式

- 传送一个数据执行一次中断服务子程序（几十条指令）
- 效率低下，不适合于高速传输的系统

■ DMA方式

- 外设与主存间建立一个由硬件管理的数据通路（虚拟通路，还是通过系统总线）
- CPU不介入外设与主存的数据传送操作
- 减少CPU开销，提升效率



DMA方式

- DMA基本概念
- DMA传输方式
- 基本DMA控制器

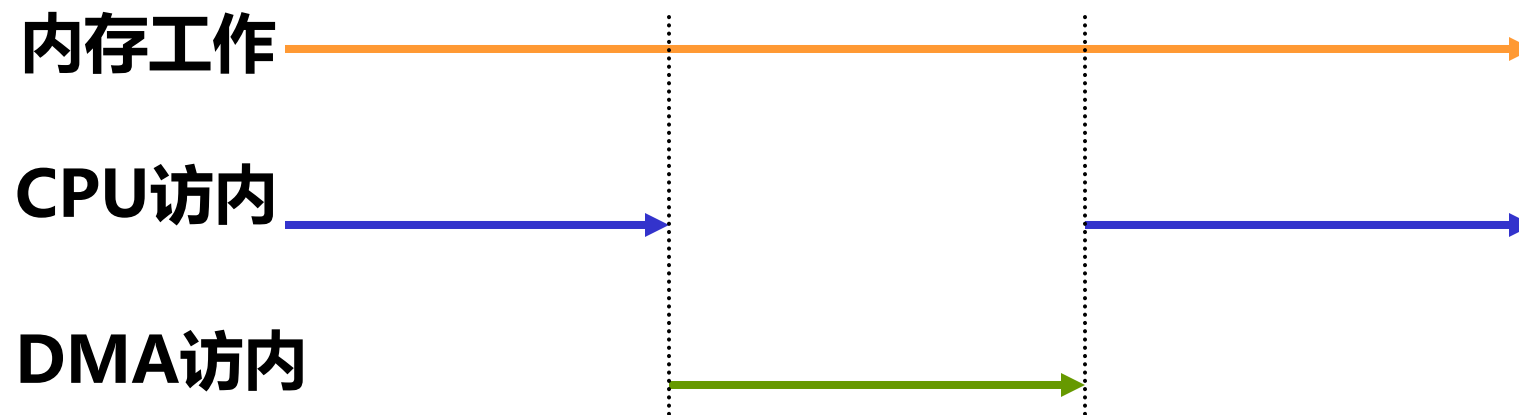
内存争用

- DMA方式进行数据传送时
 - DMA控制器直接访问内存
 - CPU执行主程序（需要访内）
 - 主存使用权的冲突（资源冲突）
- 如何处理这种冲突？
 - 停止CPU使用主存
 - DMA与CPU交替使用主存
 - 周期挪用法

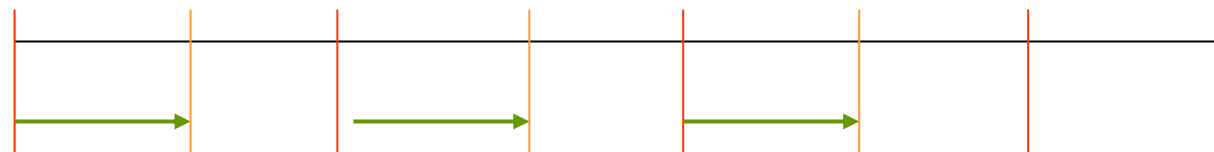
停止CPU使用主存

- DMA传送数据时，CPU停止使用主存
- 一批数据传送结束后，DMA再交还主存使用权
- DMA传送过程中，CPU处于等待状态

停止CPU访内

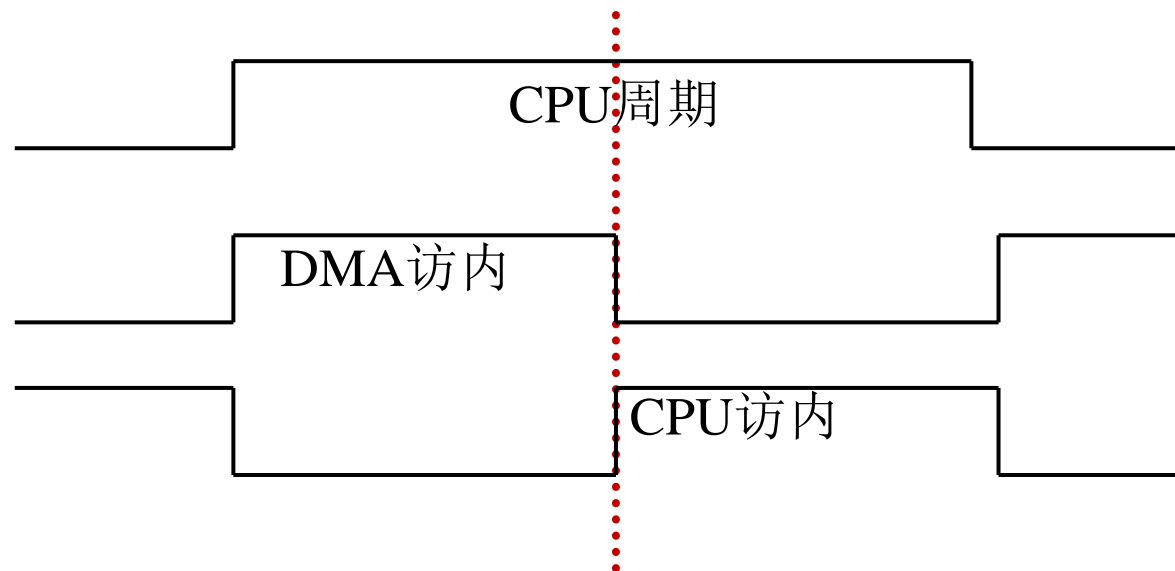


1. DMA批量数据传输周期过长，CPU长期无法访内
2. 外设传送两个数据的时间间隔大于存储周期，内存未充分利用



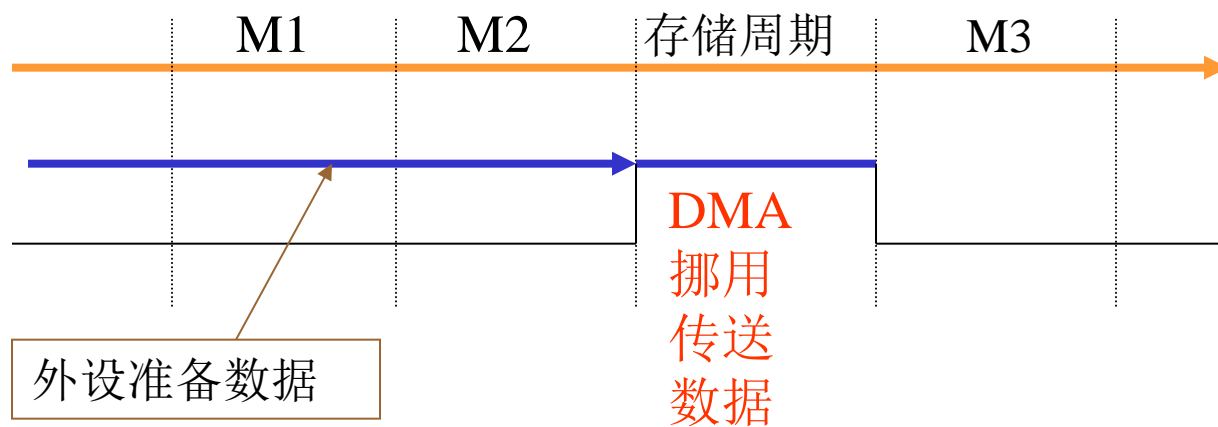
DMA与CPU交替使用主存

- 每个CPU工作周期分成两段
 - 一段用于 DMA访问主存
 - 一段用于CPU访问主存
- 无主存使用权移交过程



周期挪用法

- DMA要求访问主存时，CPU暂停**一个或多个存储周期**。一个数据传送结束后，CPU继续运行。
- CPU现场没有变动，仅延缓了指令的执行
 - **周期挪用**，或称**周期窃取**。
- 如发生访存冲突，则DMA优先访问。



DMA主要操作过程（准备阶段）

- 主机通过CPU指令向DMA接口发送必要的传送参数，并启动DMA工作。
 1. 数据传送的方向
 2. 数据块在主存的首地址
 3. 数据在外设存储介质上的地址
 4. 数据的传送量

DMA主要操作过程（传送阶段）

■ **宏观** DMA是连续传送一批数据 **微观** 每传送一个数据，发一次DMA请求

■ 传输过程（周期挪用）

1. 设备准备数据：当设备接收到 CPU 的 DMA 命令后就可以开始准备数据
2. 设备发送 DMA 请求：数据准备好后就通过 DREQ 控制线向 DMAC 发出 DMA 请求。
3. DMAC 申请总线：DMAC 收到 DMA 请求后立即将 HOLD 信号置“1”，向 CPU 申请总线控制权。
4. 总线授权：CPU 在机器周期结束后响应总线使用申请，让出总线控制权，并发出总线授权信号 HLDA 通知 DMAC。

DMA主要操作过程（传送阶段）

5. DMA 数据传输：收到 HLDA 信号将内存地址放置在地址总线上；设置控制总线读写命令控制信号，并向设备DMA 应答信号 DACK。设备收到 DACK 信号后会和内存完成一个机器字的数据交换
6. 传输控制：设备传输完一次数据后会继续重复第 1 步到第 5 步的工作，DMAC 在每次传输时还需要负责维护内存地址和传输计数器，并撤除 HOLD信号释放总线
7. 数据传输结束时，DMAC 会通过 INTR 信号线发送一个 EOP（End OfProcess）的 DMA 中断请求信号，告知 CPU 传输完成

DMA主要操作过程（结束阶段）

- DMA在两种情况下都进入结束阶段。
 - 正常结束，一批数据传送完毕
 - 非正常结束，DMA故障
- 结束阶段DMA向主机发出中断请求
- CPU执行中断服务程序
 - 查询DMA接口状态，根据状态进行不同处理

DMA与程序中断的区别

- 中断通过程序传送数据，DMA靠硬件来实现。
- 中断时机为两指令之间，DMA响应时机为两存储周期之间。
- 中断不仅具有数据传送能力，还能处理异常事件。DMA只能进行数据传送。
- DMA仅挪用了一个存储周期，不改变CPU现场。
- DMA请求的优先权比中断请求高。CPU优先响应DMA请求，是为了避免DMA所连接的高速外设丢失数据。
- DMA利用了中断技术

本章主要内容

- 9.1 输入输出设备与特性
- 9.2 I/O接口
- 9.3 数据传送控制方式
- 9.4 程序控制方式
- 9.5 程序中断方式
- 9.6 DMA方式
- **9.7 通道方式**
- 9.8 常见I/O设备



通道方式

- 通道的功能
- 通道类型
- 通道结构的发展

通道方式

- DMA方式依赖硬件逻辑支持，随着设备数量的增加，DMA控制器增加，成本也相应增加。必须找出一种方法使DMA技术被更多的设备共享。
- DMA接口的起始准备仍需CPU执行一段程序完成。高速设备的信息是成批传送的，一批数据包含了相当多的数据块，每一数据块都要使DMA接口初始化。数据块连续频繁地传送，其占用CPU的时间就不可忽视了。

通道方式

- DMA方式的进一步发展，数据的传送方向、内存起始地址及传送的数据块长度等都由独立于CPU的通道来进行控制，可进一步减少CPU的干预。
 - 通道是一个具有特殊功能的处理器IOP
 - 分担CPU的I/O 处理的功能
 - 可实现外设的统一管理和DMA操作
 - 大大提高CPU效率，更多的硬件
- 通道执行通道程序来完成CPU指定的I/O任务，通道程序是由一系列通道指令组成的。
- 当通道执行完通道程序后，就发出中断请求表示I/O结束，CPU响应中断请求，执行相应的中断处理程序实现与通道之间的数据传输。

通道方式

- 设置专用的**输入输出处理机（通道）**，分担输入输出管理的全部或大部分工作。
- 吸取了DMA技术，增加了软件管理，设有专用通道指令
- 层次性的I/O系统
 - 一个主机可以连接多个通道
 - 一个通道可以管理多个设备控制器
 - 一个设备控制器又可以控制多台设备。

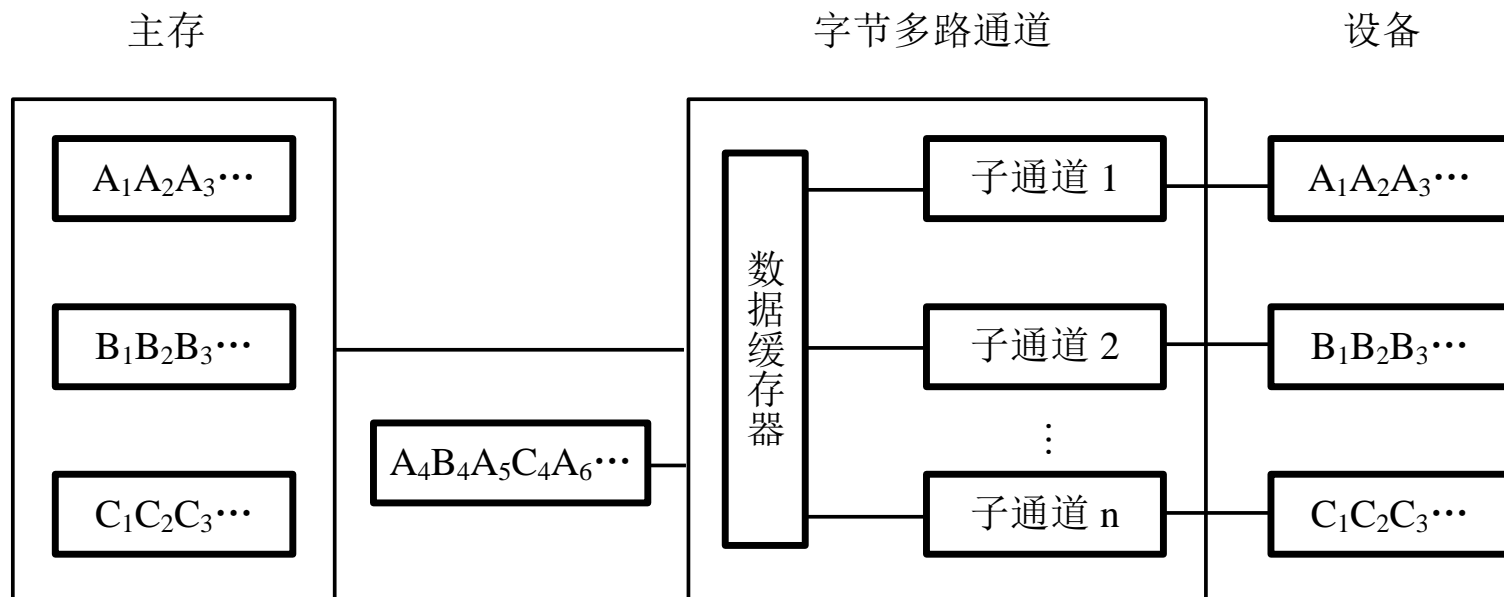
通道功能

- 根据CPU要求，组织设备与系统连接和通信；
- 选取通道指令，向设备发出操作命令；
- 指出数据在设备中的位置和主存缓冲区内的位置，组织设备与主存间的数据传输。
- 向CPU反映设备、设备控制器及通道本身的状态信息。
- 将外设和通道本身的中断请求，按次序及时报告CPU。
- 设备控制器介于通道与设备之间，是通道对外部设备实行具体控制的机构。
 - 将通道发送的命令转换为设备能接受的控制信号
 - 向通道反映设备的状态
 - 将设备的各种电平信号转换成通道能识别的标准逻辑信号。

通道分类

- 根据设备共享通道的情况及信息传送速度的要求分为3类
 - 字节多路通道
 - 选择通道
 - 数组多路通道

字节多路通道



- 包括若干子通道，每个子通道服务于一个慢速设备
- 在一段时间内交替执行多个设备的通道子程序
- 传输单位是字节
- 宏观上这些设备并行工作

选择通道

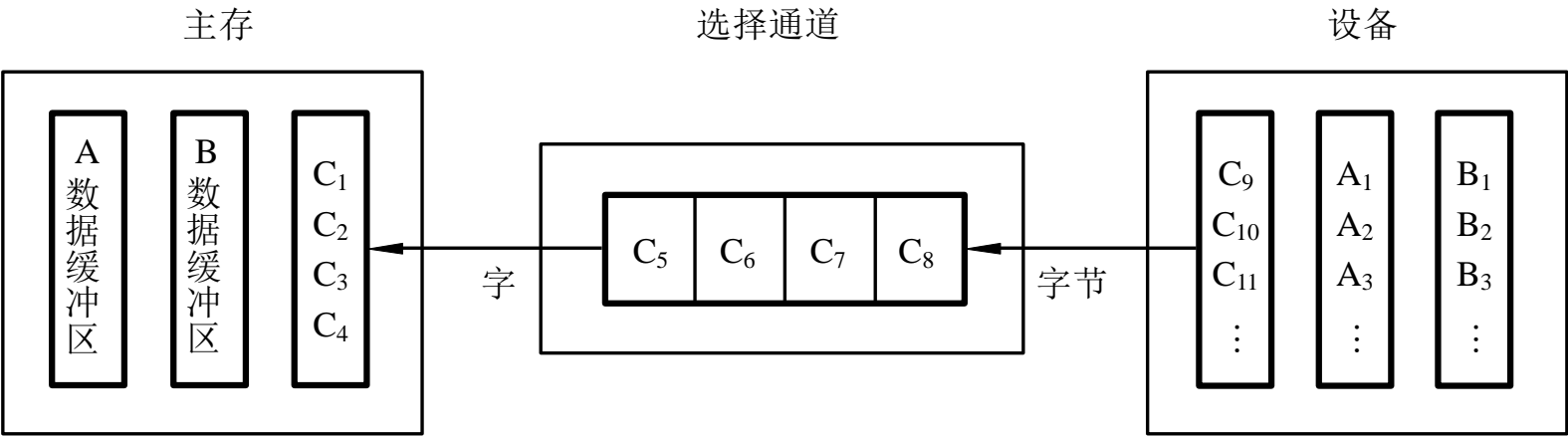
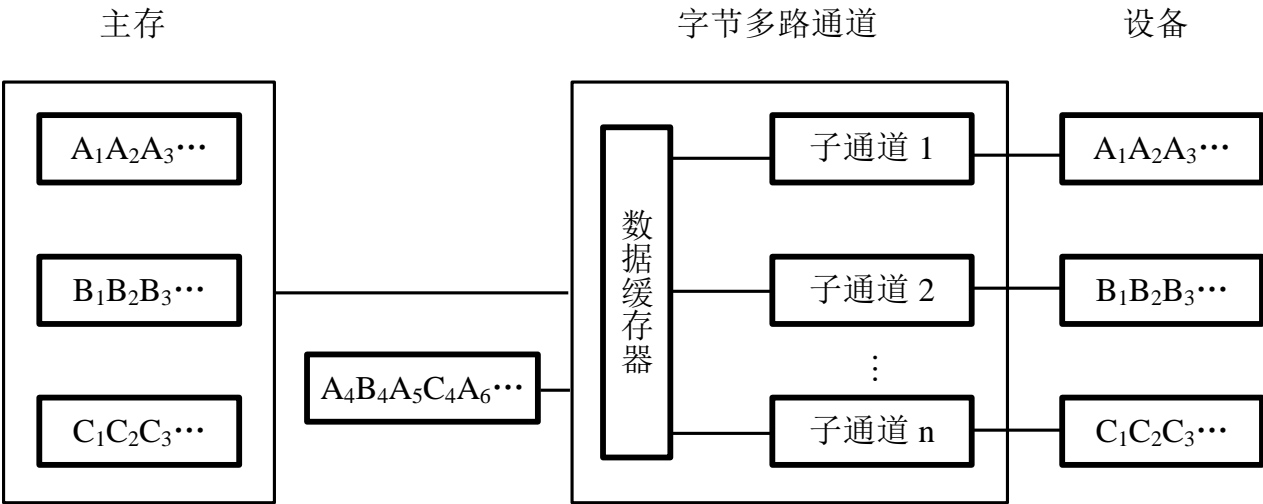
■ 字节多路通道

- 适合慢速设备，不适合高速设备
- 高速设备传送两个字节间的空闲很短

■ 选择通道

- 设备以成批数据连续传送方式占用通道，直到指定数量的数据全部传送完毕，通道才转为其它设备服务。
- 选择通道在物理上可以连接多个设备，但设备不能同时工作。
- 选择通道只有一个子通道，它适用于大批量数据的高速传送。

选择通道



数组多路通道

- 通道能高速传送数据，但设备辅助操作时间不能有效利用
 - 如硬盘启动后，平均定位时间较长，磁带机磁头定位时间更长，可达几分钟。导致通道处于等待状态
- 为利用这段时间，将字节多路 and 选择通道折中，称为数组多路通道。
 - 多个设备以数据组（块）为单位交叉使用通道。
 - 设备占用通道时，连续传送一组数据，然后将出让通道使用权
 - 数据组的大小因设备而异，有256B、512B或1KB等。

数组多路通道

- 数组多路通道也包含若干个子通道。
 - 数组多路通道适用于中、高速设备，如磁带机、磁盘等。
 - 传送的基本数据单位与字节通道不同。
 - 同一时刻只允许一个设备进行传输型的工作
- 某设备执行辅助操作时
 - 通道暂时断开与该设备的连接，挂起与该设备对应的通道程序
 - 转为其它设备服务，当设备完成了辅助操作，且通道空闲时，通道才重新转为该设备服务。
- 传送效率高，硬件复杂度高