



计算机组成与系统结构

第八章 输入输出系统

吕昕晨

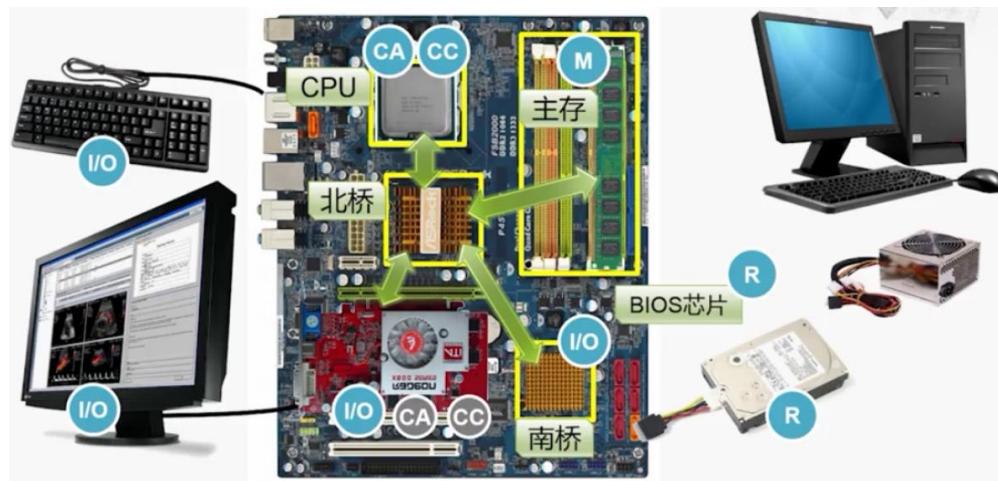
lvxinchen@bupt.edu.cn

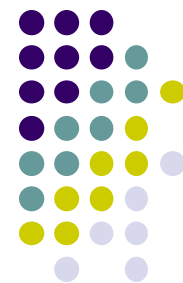
网络空间安全学院

输入输出系统

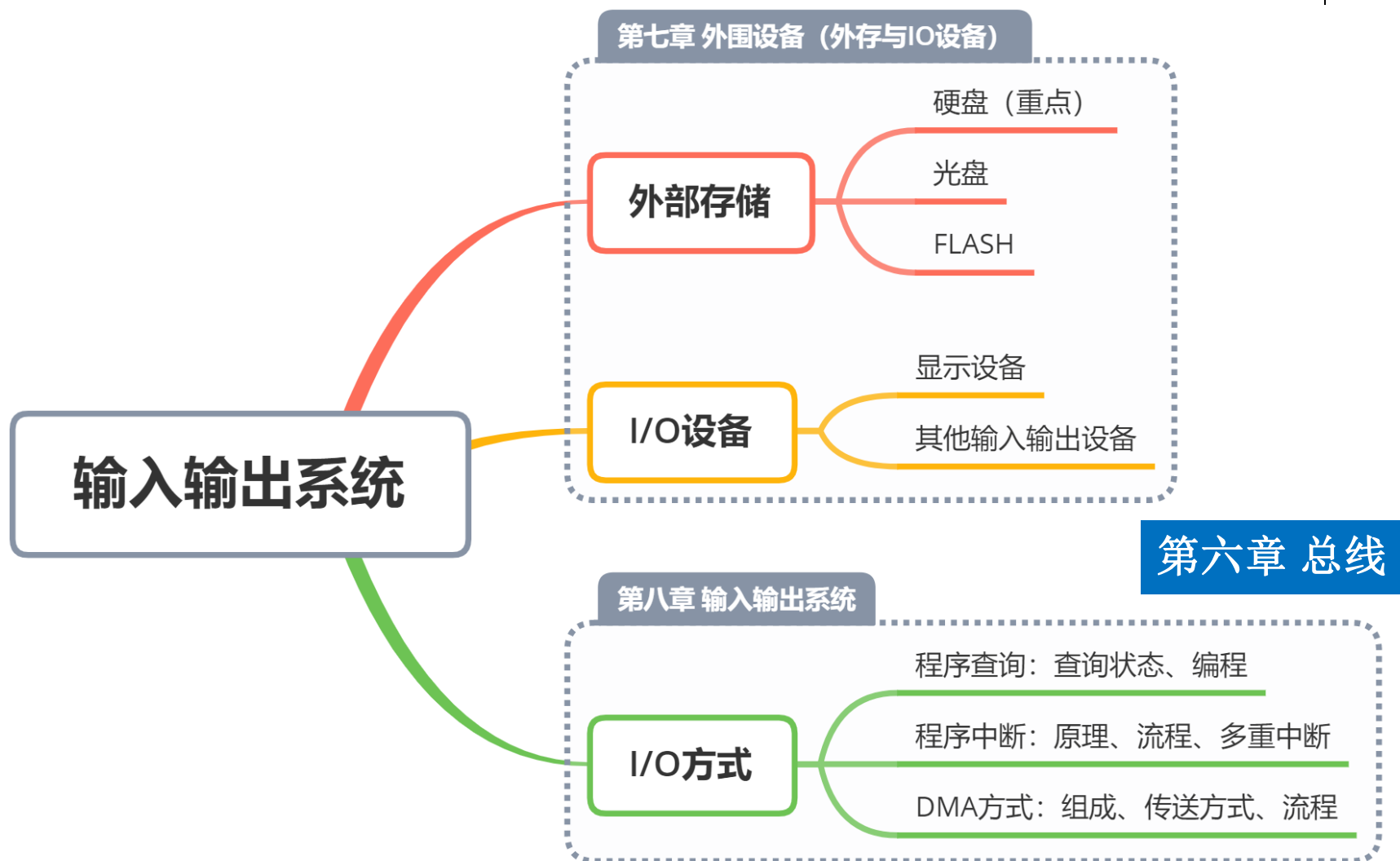


- 上周内容
 - 外存与I/O设备
 - 外存
 - 数据存储原理
 - 逻辑结构与实现
 - 输入输出设备
 - 关键：工作（存储）原理、逻辑结构、工作方式
- 本章内容——输入输出系统
 - 关键：CPU与外设如何进行信息交互
 - 输入输出控制方式
 - 程序控制、程序中断
 - DMA、通道方式



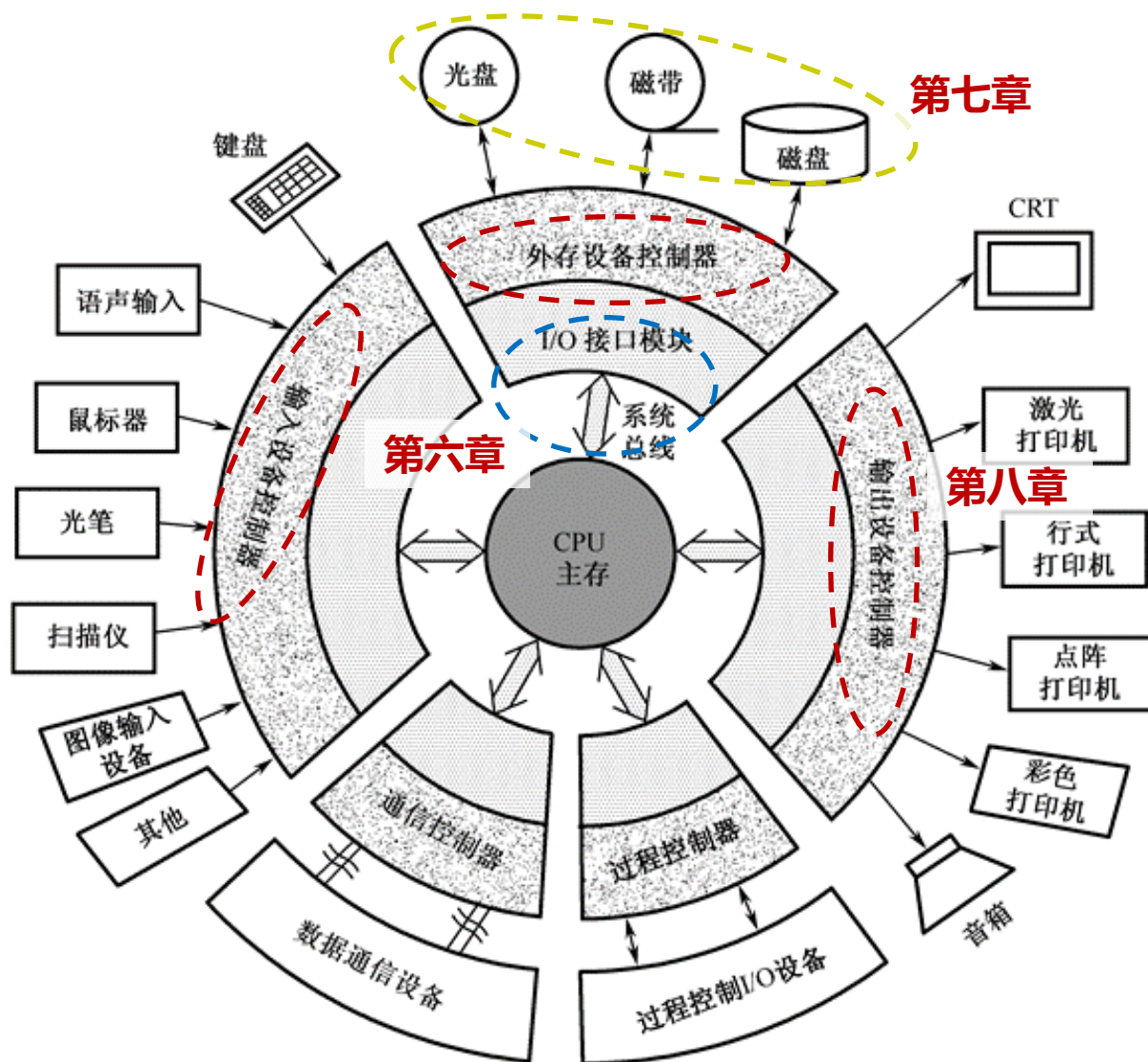


输入输出系统—总结





外存与I/O设备——结构图/分类





第八章 输入输出系统

- 外围设备的速度分级
- 信息交换方式概述
 - 程序查询方式
 - 程序中断方式
 - DMA方式



回顾：外围设备功能与速率

- 外围设备的功能是在计算机和其他机器之间，以及计算机与用户之间提供联系

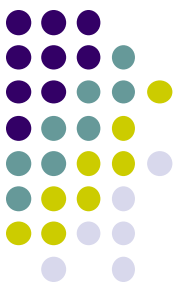
- > 安全设备
- > 处理器
- > 磁盘驱动器
- > 存储控制器
- > 打印队列
- > 电池
- > 固件
- > 计算机
- > 监视器
- > 键盘
- > 蓝牙
- > 其他设备
- > 人体学输入设备
- > 软件设备
- > 软件组件
- > 声音、视频和游戏控制器
- > 鼠标和其他指针设备
- > 通用串行总线控制器
- > 图像设备
- > 网络适配器
- > 系统设备
- > 显示适配器
- > 音频输入和输出

设备名称	功能	人/机接口	数据速率 (Mbit/s)
键盘	输入	人	0.000 1
鼠标	输入	人	0.003 8
语音输入设备	输入	人	0.264 0
声音输入设备	输入	机器	3.000 0
扫描仪	输入	人	3.200 0
语音输出设备	输出	人	0.264 0
声音输出设备	输出	人	8.000 0
激光打印机	输出	人	3.200 0
图形显示器	输出	人	800.000 0~8 000.000 0
调制解调器	输入或输出	机器	0.016 0~0.064 0
网络/局域网	输入或输出	机器	100.000 0~1 000.000 0
网络/无线局域网	输入或输出	机器	11.000 0~54.000 0
光盘	存储	机器	80.000 0
磁带	存储	机器	32.000 0
磁盘	存储	机器	240.000 0~2560.000 0



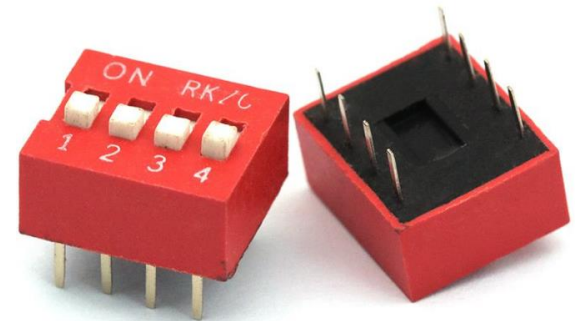
外围设备的速度分级

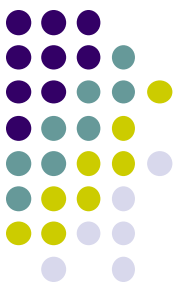
- 外设种类繁多，存在以下几种情况：
 - 不同种类的外设数据传输速率差别很大
 - 同一种设备在不同时刻传输速率也可能不同
- 问题
 - 高速的CPU与速度参差不齐的外设如何时间上同步？
- 解决办法
 - 在CPU和外设之间数据传送时加以定时
 - 对于不同速度的外围设备，需要有不同的定时方式
- 分类
 - 简单外围设备、慢速/中速外围设备、高速外围设备



速度极慢或简单的外围设备

- 速度极慢或简单的外围设备
 - 如机械开关、显示二极管等
 - 特点
 - CPU总是能足够快地作出响应
 - 对机械开关来讲，CPU可以认为输入的数据一直有效，因为机械开关的动作相对CPU的速度来讲是非常慢的，对显示二极管来讲，CPU可以认为输出一定准备就绪
 - 在这种情况下，CPU只要接收或发送数据就可以了
- 输入输出特点
 - 外围设备可以等价于一个存储器，可以直接进行数据传输
 - 无需进行查询与确认等交互操作

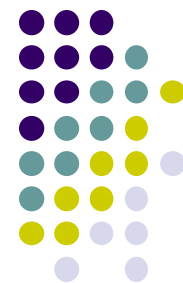




慢速或中速的外围设备

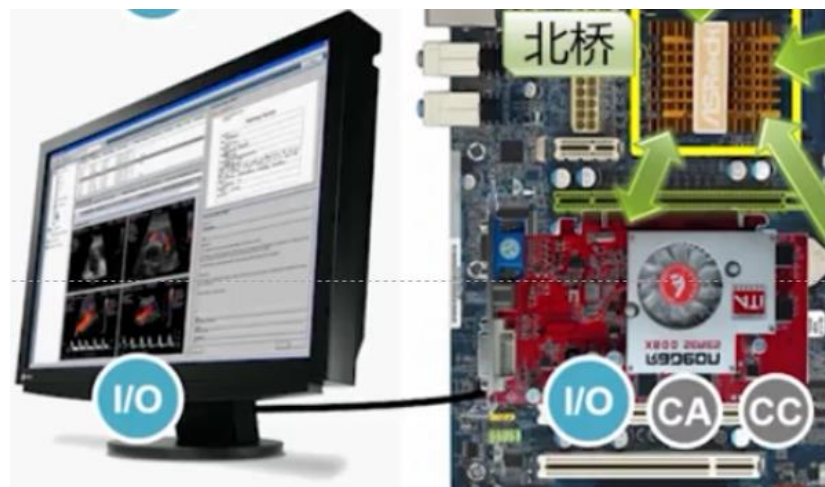
- 慢速或中速的外围设备
 - 速度和CPU的**速度并不在一个数量级**
 - 由于设备(如键盘)本身是在**不规则时间间隔**下操作的
 - CPU与这类设备的数据交换**可采用异步定时方式**
- 异步定时（交互）方式流程
 - **应答式数据交换**（读取数据）
 - CPU首先询问外设的状态，如果该外设的状态标志表明设备已**准备就绪**，从总线上接收数据；否则“**忙/未就绪**”标志
 - CPU在接收数据以后，发出**输入响应**信号，告诉外设已经把数据总线上的数据取走
 - 外设把**准备就绪**的状态标志复位，并准备下一个字的交换

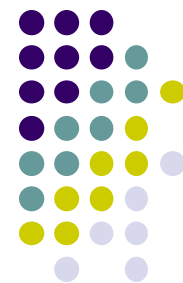




高速的外围设备

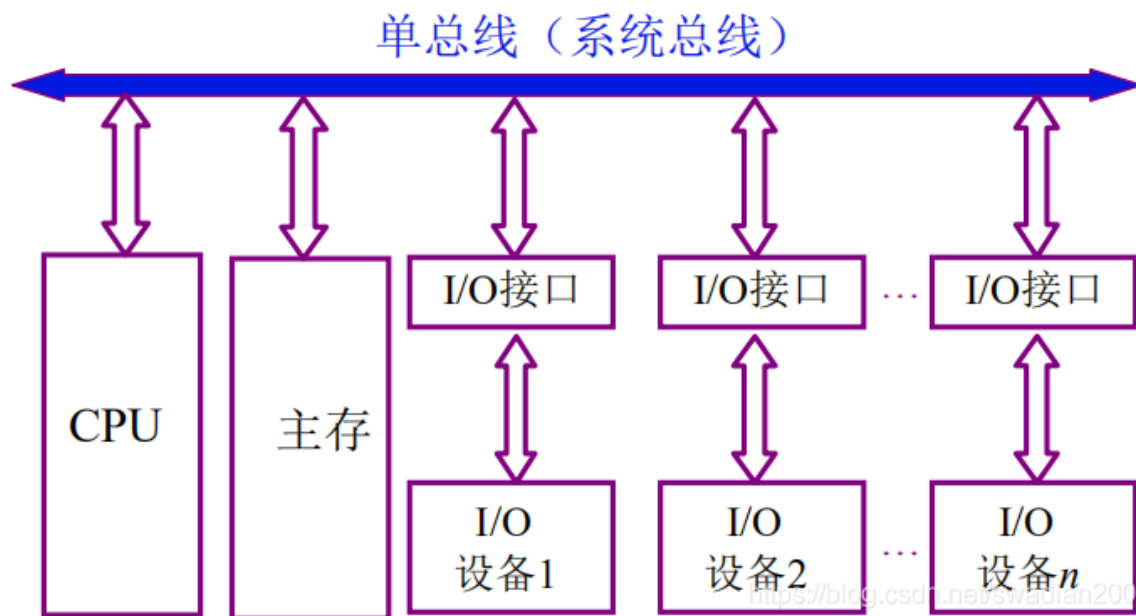
- 这类外设是以相等的时间间隔操作的，而CPU也是以等间隔的速率执行输入/输出指令的
- 连续不断的数据传输（恒定高速率）
- 一般采用同步定时方式





设备编址—统一编址（总线）

- 单总线结构
 - 使用一条单一的系统总线来连接CPU、内存和I/O设备
- 适配器（接口）
 - 实现高速CPU与低速外设之间工作速度上的匹配和同步，并完成计算机和外设之间的所有数据传送和控制

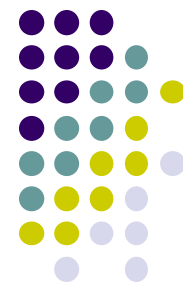




设备编址—统一编址（指令）

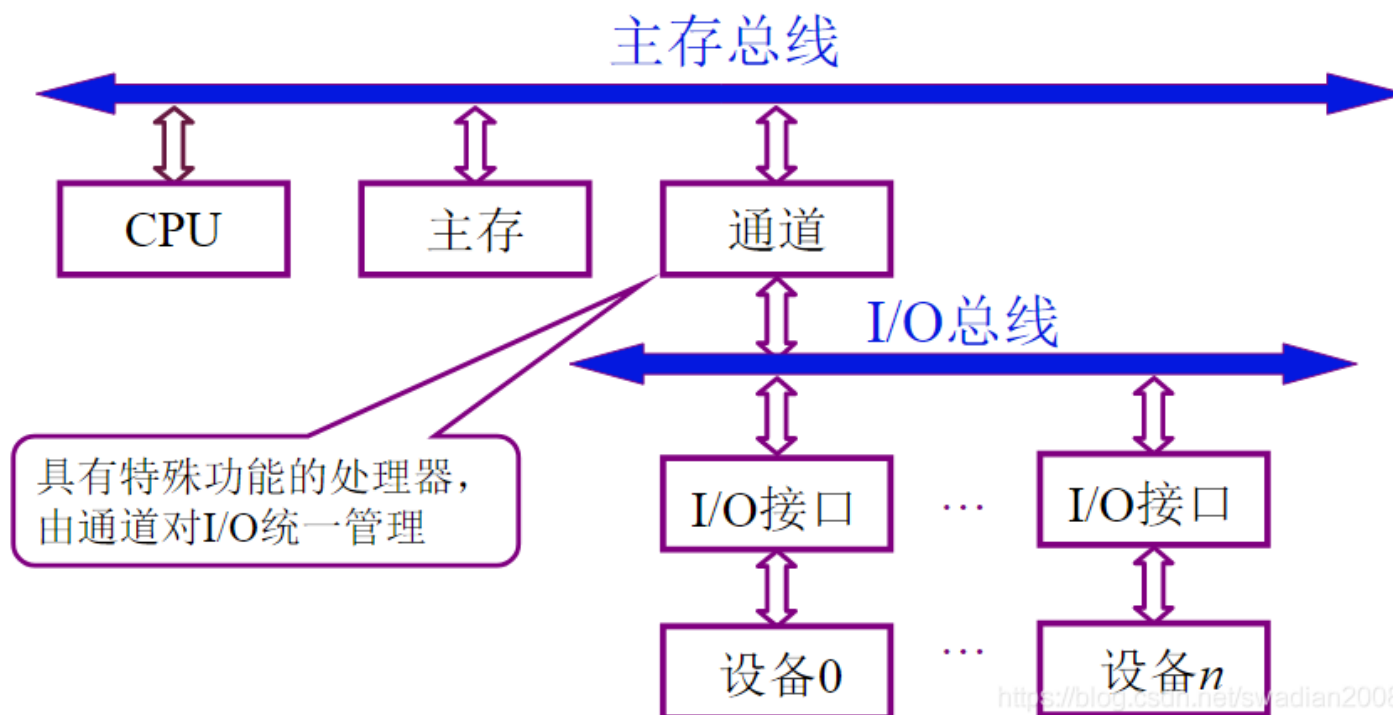
- 统一编址
 - 输入/输出设备中的控制寄存器、数据寄存器、状态寄存器等和内存单元一样看待，和内存单元联合在一起编排地址，使用相同指令
 - 无需单独命令，通过地址区分内存单元与外围设备
 - 又名：存储器映射方式
 - 优点：编程灵活、无需访存指令
 - 缺点：内存空间小，访问速度慢

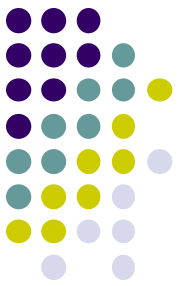




设备编址—独立编址（总线）

- 主存总线：用于CPU、主存和通道间的数据传送
- I/O总线：用于多个外部设备与通道的数据传送
- 通道：特殊功能处理器，管理IO设备，提高效率的目的





设备编址—独立编址（指令）

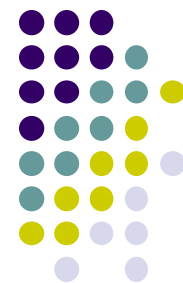
- 单独（独立）编址
 - 内存地址和I/O设备地址是分开的，访问内存和访问I/O设备使用不同操作码的指令，访问I/O设备有专门的I/O指令组（指令区分内存与外围设备）
 - 可扩展内存实际可使用空间（地址线位数固定）
 - 优点：译码速度快，内存空间大
 - 缺点：指令集更为复杂





第八章 输入输出系统

- 外围设备的速度分级
- 信息交换方式概述
 - 程序查询方式
 - 程序中断方式
 - DMA方式



程序查询方式

- **程序查询方式**是早期计算机中使用的一种方式，数据在CPU和外围设备之间的传送**完全靠计算机程序控制**
- 流程
 - CPU（**定期**）查询外围设备接口状态
 - 若发现设备有待传输数据（准备就绪）
 - CPU通过总线取走数据进行处理
 - 向设备发送传送成功信息（恢复状态）
- 全部**编程控制**，**灵活配置**，**无需硬件配合**



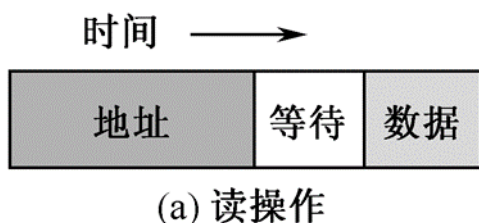


程序查询方式

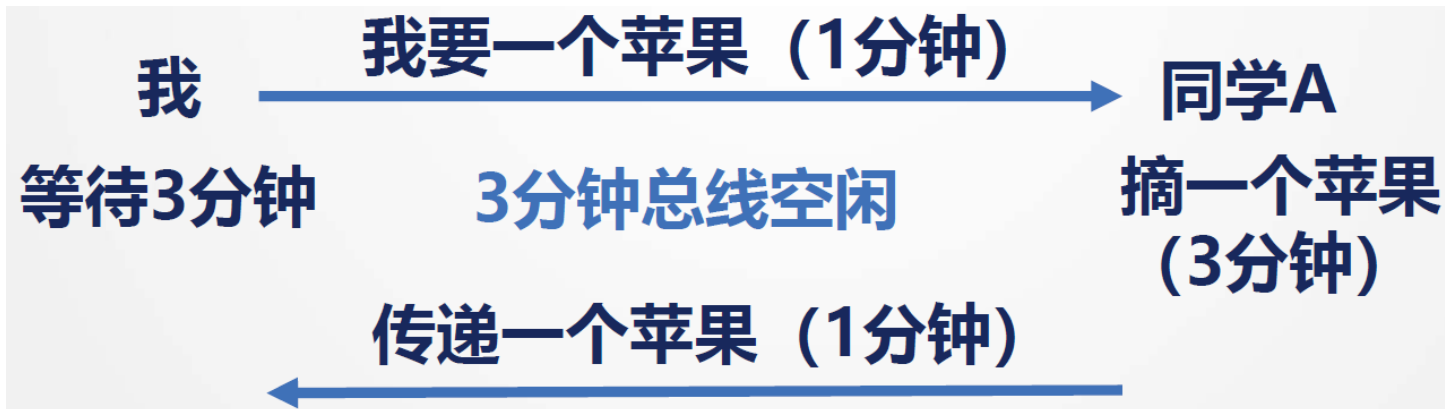
- 数据在CPU和外围设备之间的传送完全靠计算机程序控制
- 优点
 - CPU的操作和外围设备的操作能够同步，灵活配置
- 缺点
 - 外围设备动作很慢，程序进入查询循环时将白白浪费掉CPU很多时间。这种情况下，CPU此时只能等待，不能处理其他业务。
 - 即使CPU采用定期地由主程序转向查询设备状态的子程序进行扫描轮询的办法，CPU宝贵资源的浪费也是可观的。

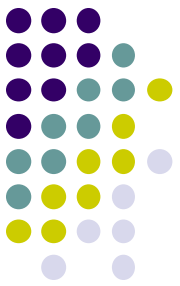


示例：摘苹果—分离式通信



主模块发出地址、命令	占用总线
从模块准备数据	不占用总线
从模块发出数据	占用总线

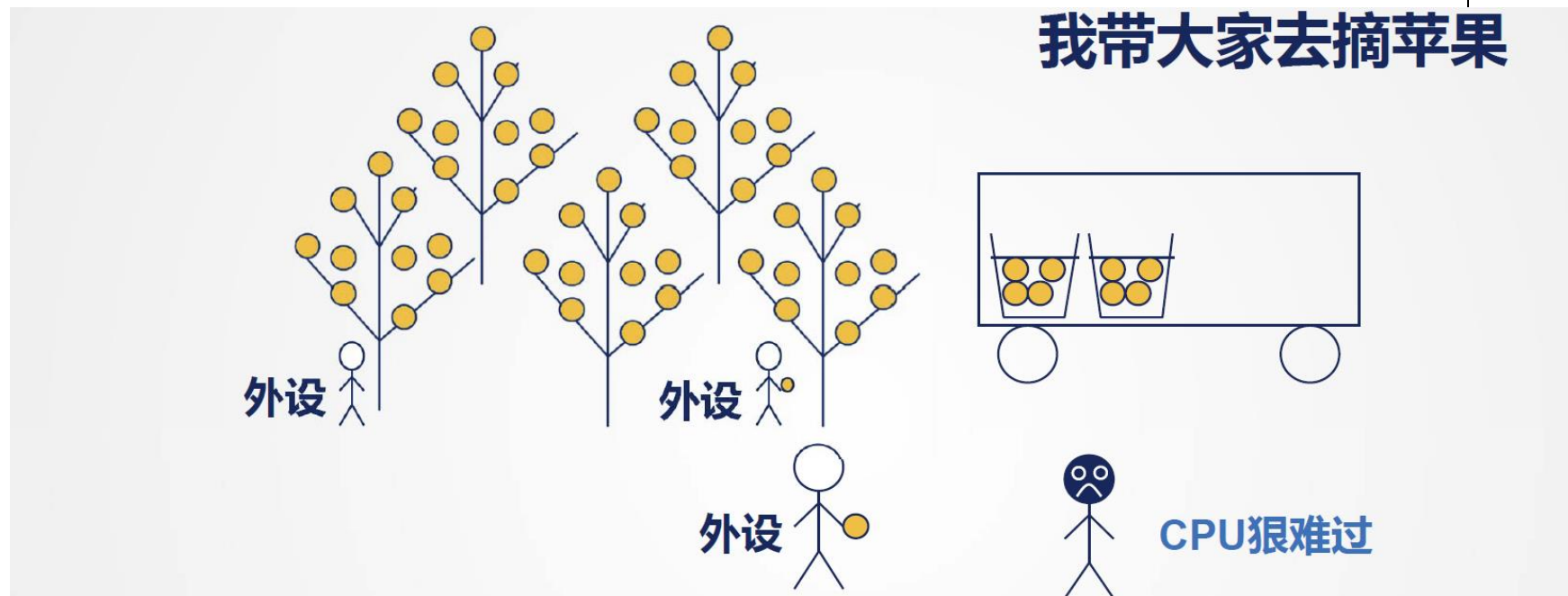




示例：摘苹果—分离式通信



输入输出方式类比—查询



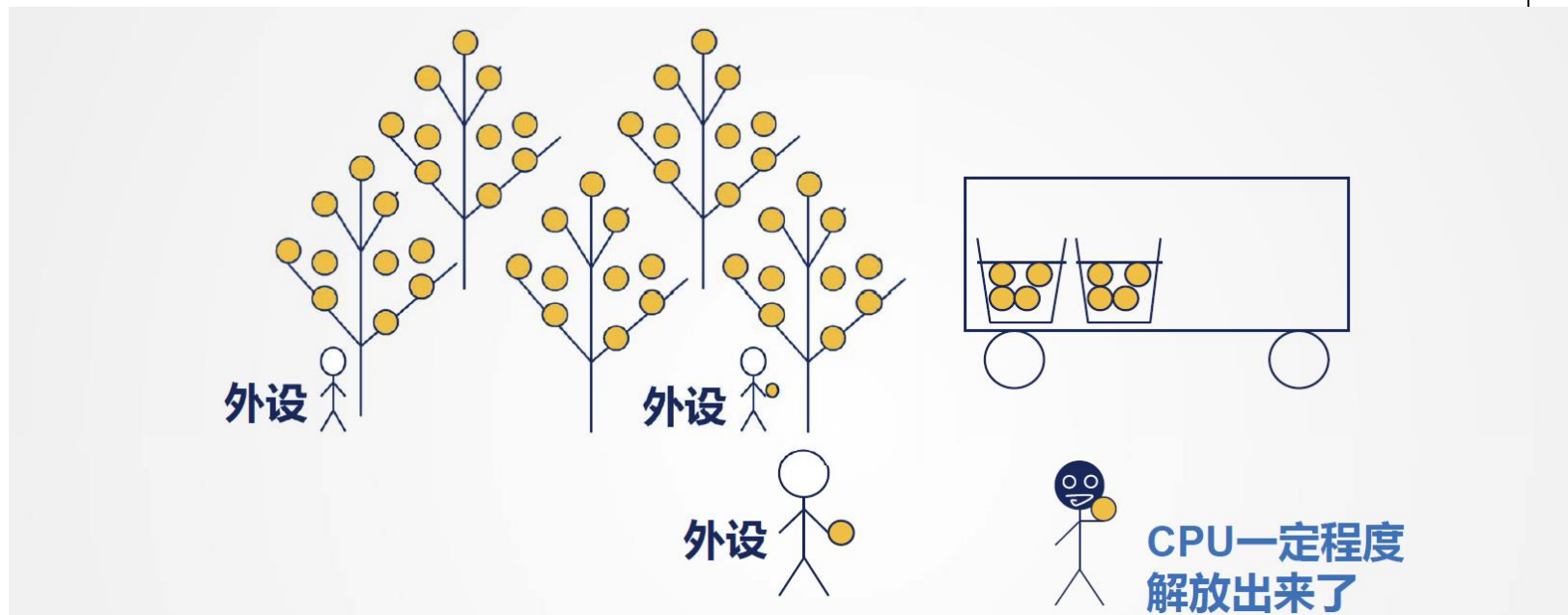
- 每一位同学都是一个外设；我是CPU，彼此有任务分工在查询模式下，我需要不断轮流问每一位同学“摘完了吗”，导致我不能做自己的事情



第八章 输入输出系统

- 外围设备的速度分级
- 信息交换方式概述
 - 程序查询方式
 - 程序中断方式
 - DMA方式

输入输出方式类比—中断

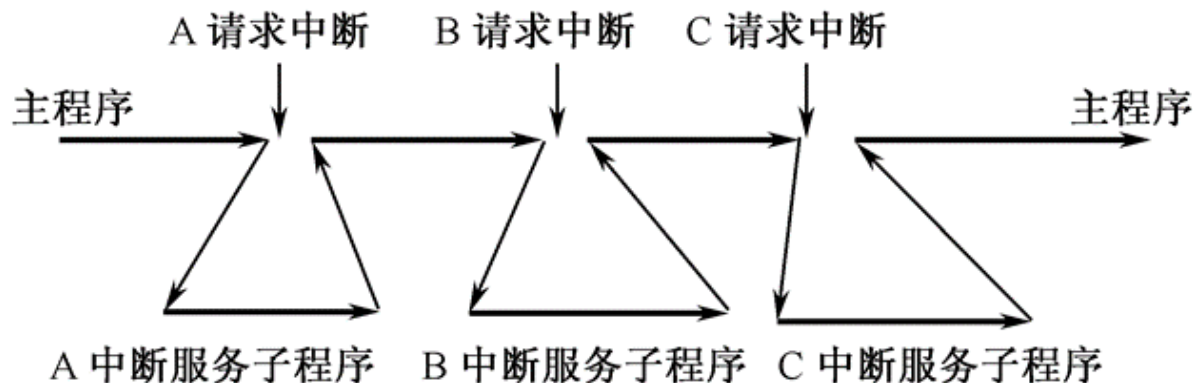


- 在中断模式下，我不需要轮询了。每位同学摘到苹果会主动来找我，可大幅节省我的时间



程序中断方式

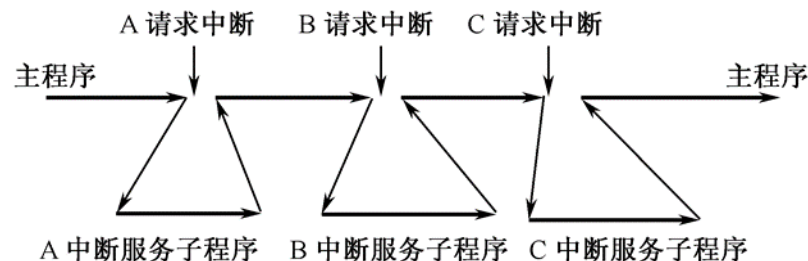
- 思路：增加硬件，使得外围设备可以通知CPU，避免查询
- 定义：中断是外围设备用来“主动”通知CPU，准备送出输入数据或接收输出数据的一种方法



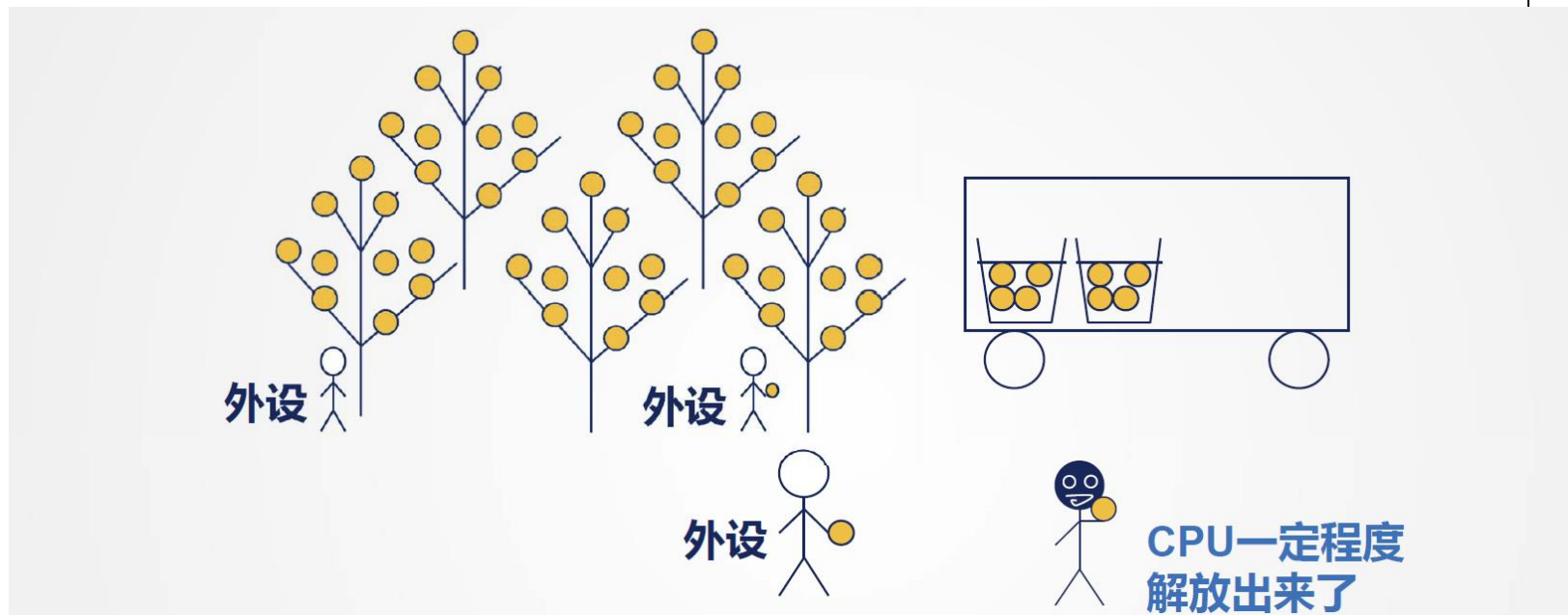
程序中断方式



- 特征
 - CPU不需要查询外设状态，等待外设中断请求
- 优缺点
 - 节省了CPU宝贵的时间，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，应立即进行
 - 不适合频繁数据传输（保存、恢复现场开销），时延较大
- 问题
 - 需要额外硬件电路
 - 中断请求总线（请求、响应）
 - 中断控制器
 - 中断处理程序（如何寻找对应程序）



输入输出方式类比—中断



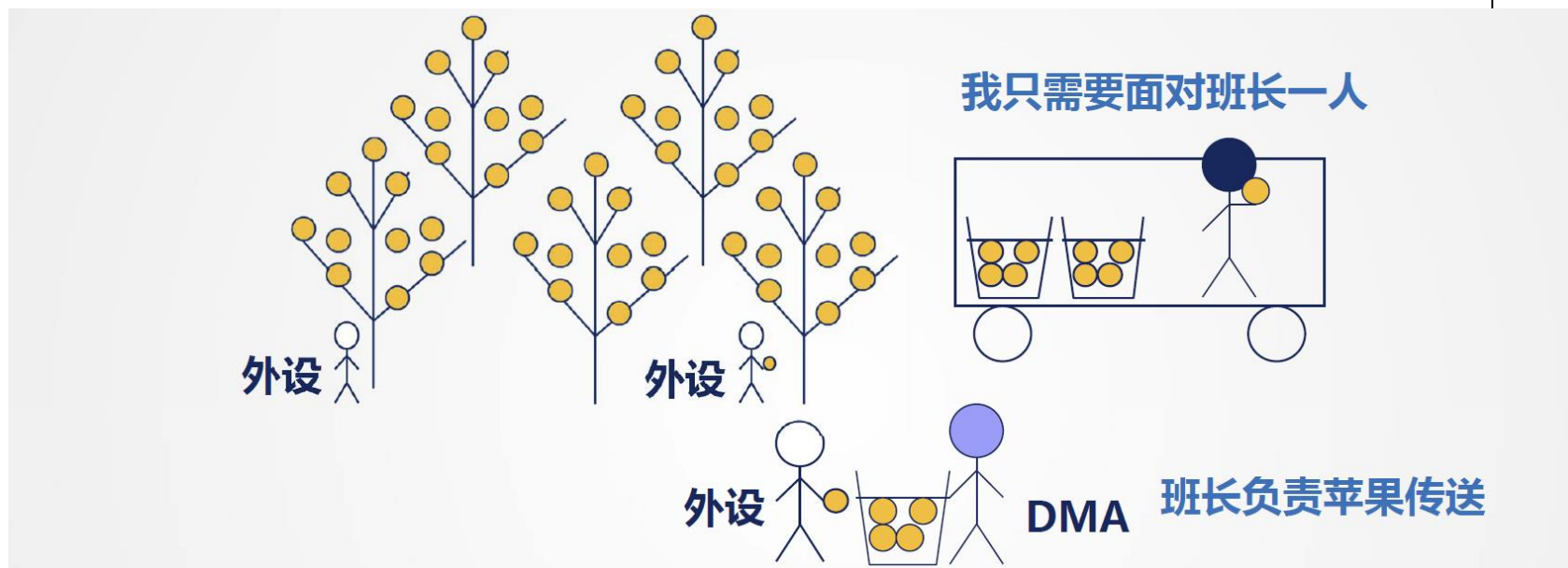
- 在中断模式下，我不需要轮询了。每位同学摘到苹果会主动来找我，可大幅节省我的时间
- **但是同学数量很多，而且我需要亲自把苹果运到车上，仍会占用我较多时间**



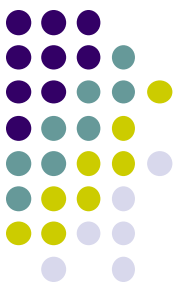
第八章 输入输出系统

- 外围设备的速度分级
- 信息交换方式概述
 - 程序查询方式
 - 程序中断方式
 - DMA方式

输入输出方式类比—DMA



- 在DMA模式下，我把苹果传送的任务交给班长，CPU被进一步解放



直接内存访问 (DMA)

- 问题
 - 如果外设**随机出现数据**，并与存储器进行通信
 - **通信频率高，数据量大**
- 现有方案是否使用？（**CPU控制数据传输**）
 - 程序查询方式
 - 随机数据出现，浪费CPU资源
 - 与存储器进行通信，不应影响CPU工作
 - 程序中断方式
 - 通信数据量很大，频繁交互，中断程序开销大
 - 频繁保存现场会影响系统性能
- 解决方案（**其他硬件负责数据搬运**）
 - 新增DMAC，具有总线控制权限，辅助完成数据传输



直接内存访问 (DMA)

- 直接内存访问(DMA)方式
 - 一种完全由硬件执行I/O交换的工作方式
 - 这种方式既考虑到中断响应，同时又要节约中断开销。
 - DMA控制器从CPU完全接管对总线的控制，数据交换不经过CPU，直接在内存和外围设备之间进行，以高速传送数据
- 优缺点
 - 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，需要更多的硬件
 - DMA方式适用于内存和高速外围设备之间大批数据交换的场合



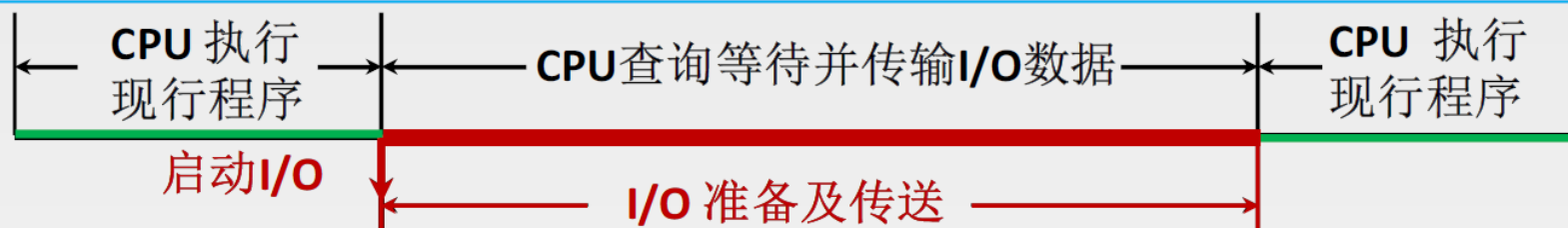
DMA方式的主要优点

- DMA方式的主要优点是速度快
 - 由于CPU根本不参加传送操作
 - 省去了CPU取指令、取数、送数等操作，速度更快
 - 在数据传送过程中，没有保存现场、恢复现场的工作
 - 内存地址修改、传送字个数的计数等，也不是由软件实现，而是用硬件线路直接实现的
 - 速度更快，避免占用CPU资源
- 总结
 - DMA方式能满足高速I/O设备的要求，有利于CPU效率的发挥

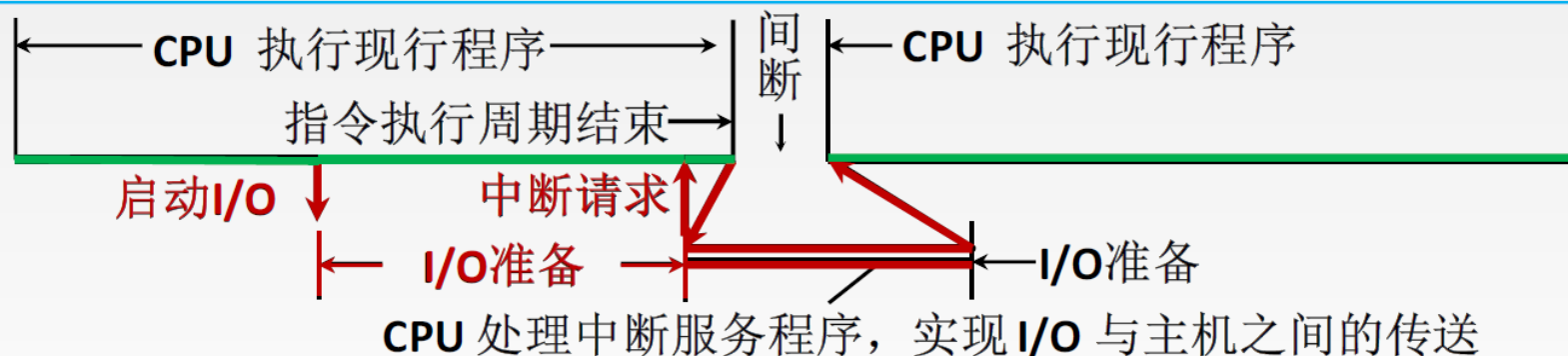


输入输出方式总结

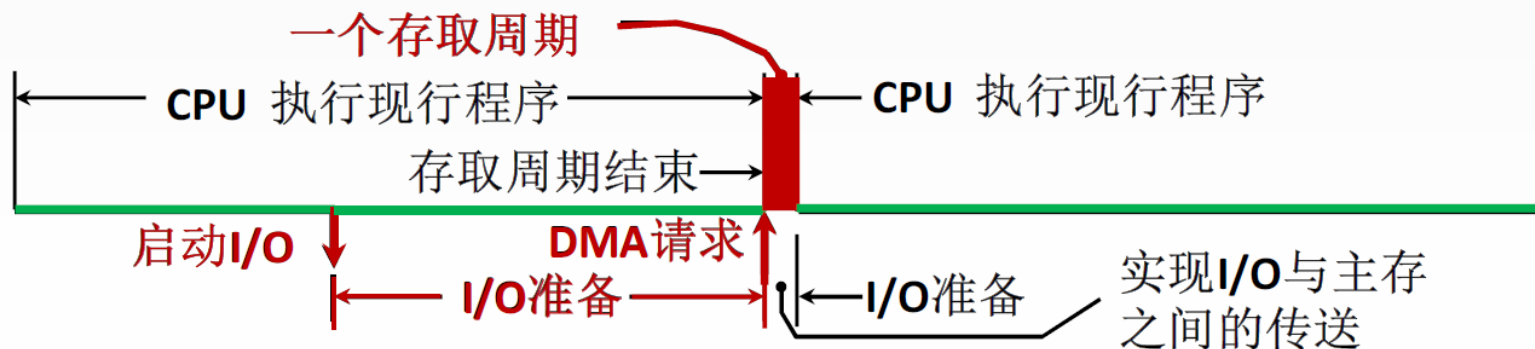
程序
查询
方式



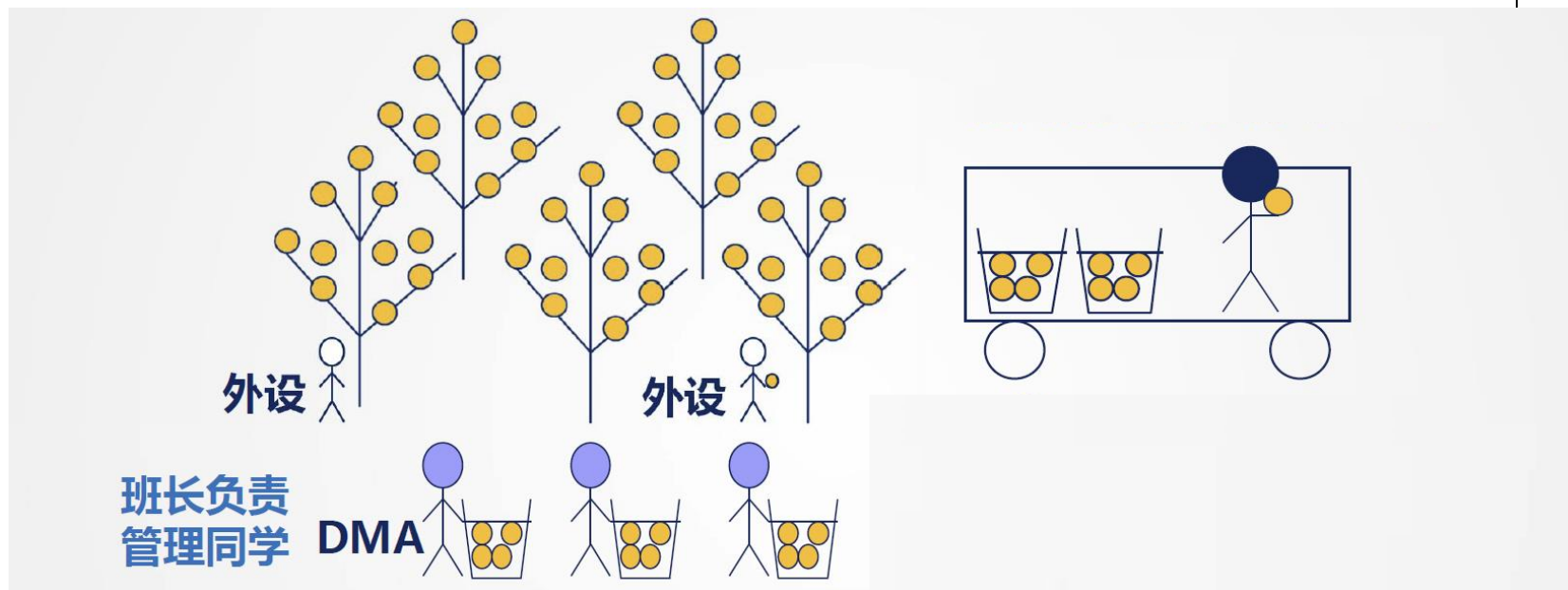
程序
中断
方式



DMA
方式

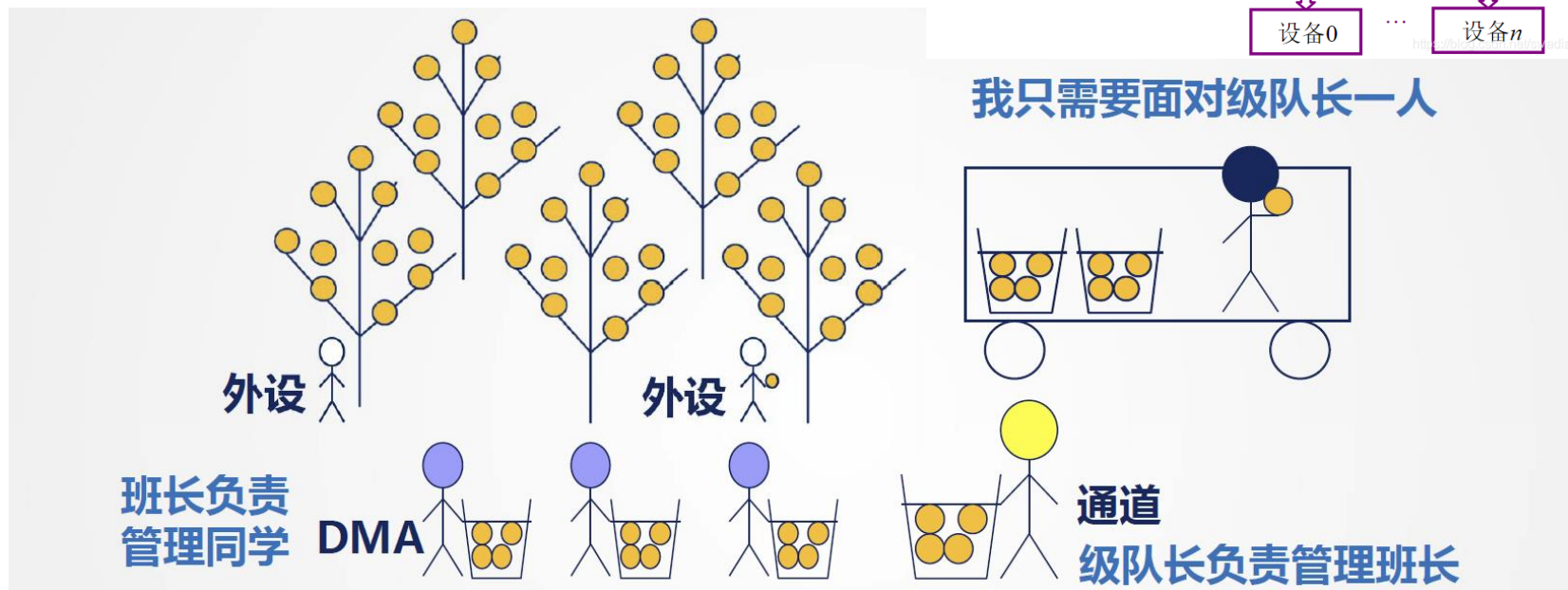
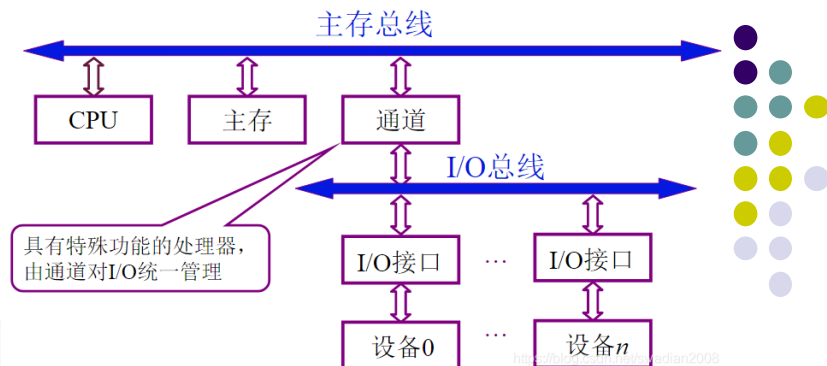


I/O演进—DMA问题



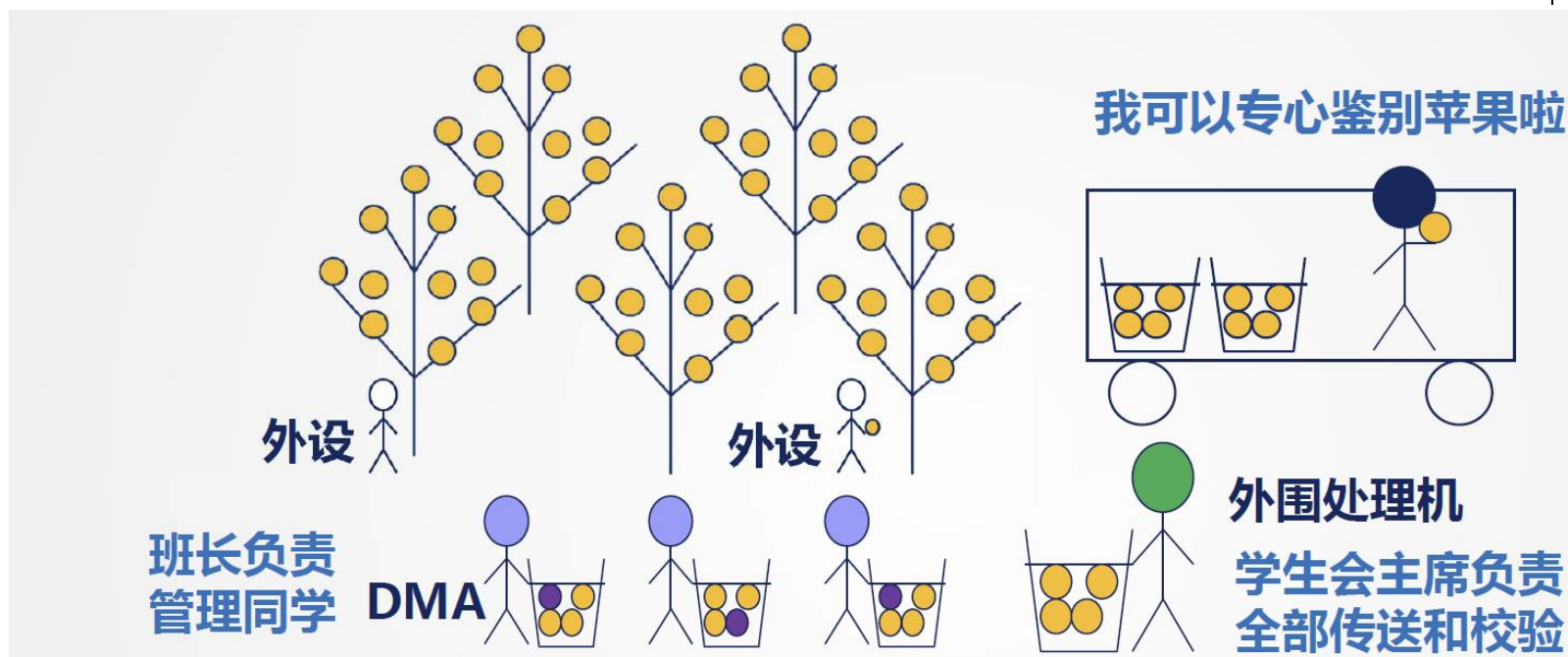
- 在DMA模式下，我把苹果传送的任务交给班长，CPU被进一步解放
- 但是班长数目还是比较多，能不能进一步减少开销

I/O演进—通道



- 在通道模式下，我让级队长负责所有苹果传送事宜，因此可更专心于自身工作
- 但是，如果传来的苹果有虫子 (数据校验错)，还得CPU检查

I/O—外围处理机



- 通道+检查功能 = 外围处理机
- CPU终于被解放了



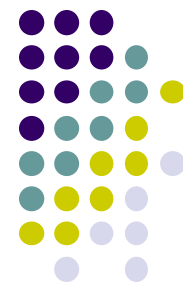
简单题—输入输出方式对比

- 比较通道、DMA、中断三种基本I/O方式的异同点

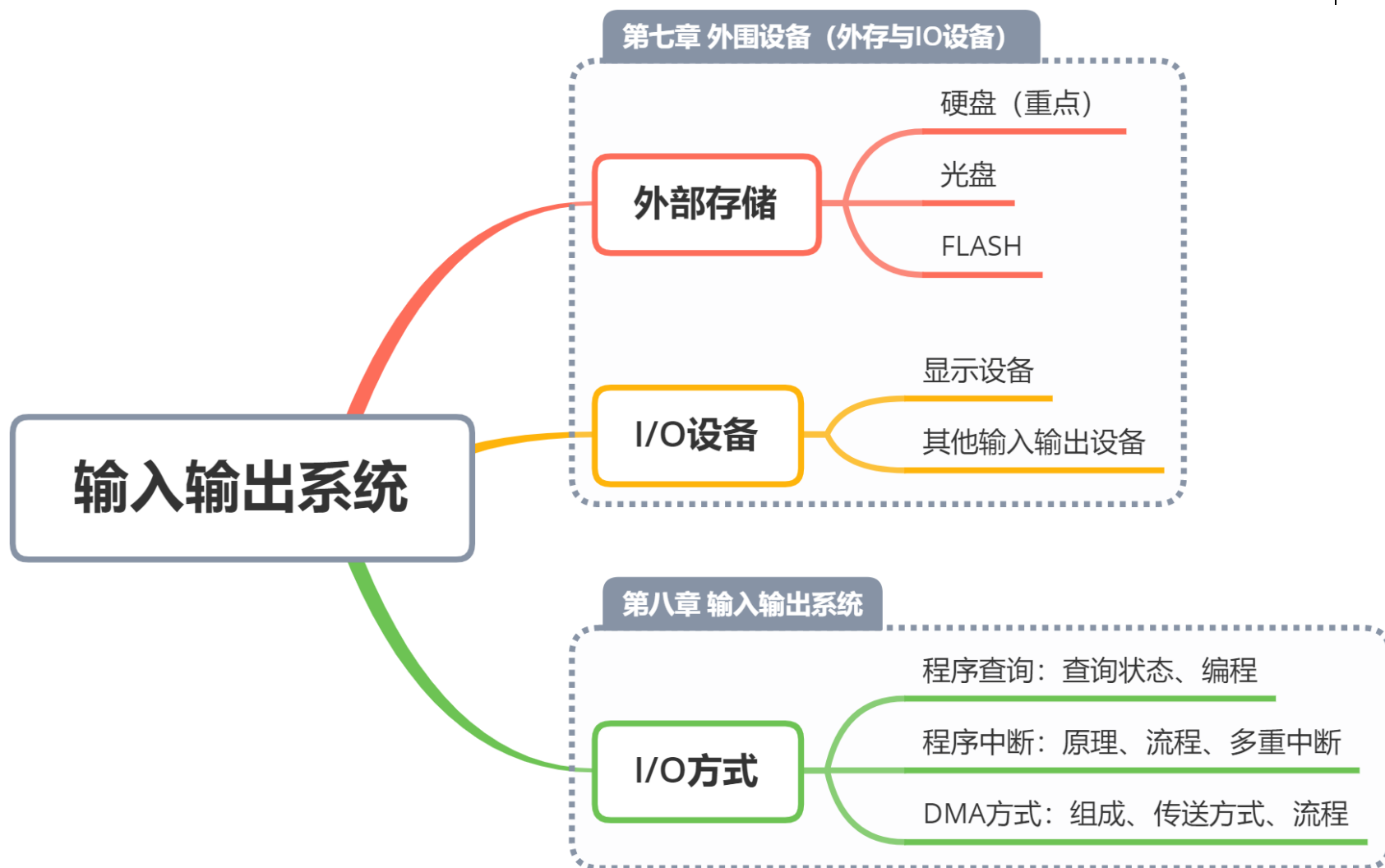


简单题—输入输出方式对比

- 比较通道、DMA、中断三种基本I/O方式的异同点
- 中断
 - 允许I/O设备主动打断CPU的运行并请求服务，从而“解放”CPU，使得其向I/O控制器发送读命令后可以继续做其他有用的工作；
- DMA
 - 直接存储器存取方式的基本思想是在I/O设备和内存之间开辟直接的数据交换通路，彻底“解放”CPU；
- I/O通道方式
 - DMA方式的发展，它可以进一步减少CPU的干预，进一步减少CPU资源消耗比例。



输入输出系统内容



I/O方式

程序查询：查询状态、编程

流程：查询接口/传输/恢复状态

设备编址：统一编址/独立编址

查询接口：状态寄存/数据缓冲/设备选择

程序中断：原理、流程、多重中断

流程：开关中断、保存现场

接口：IR、IM（屏蔽）

中断向量号：中断服务程序入口

单级、多级中断

DMA方式：组成、传送方式、流程

特征：DMA控制器，无需CPU

优点：高速外设、大批量交互

传送分配方式

停止CPU访内

周期挪用

交替访内

数据传送流程：预处理、正式传送、后处理

组成：对比中断