计算机组成原理

Principle of Computer Organization

第八章 输入输出系统

I/O System

北京邮电大学
计算机学院
戴志涛

I/O系统的组成

- I/O设备（外围设备）
- I/O接口（I/O适配器）
- I/O设备控制器
- I/O管理软件（驱动程序）

本章内容

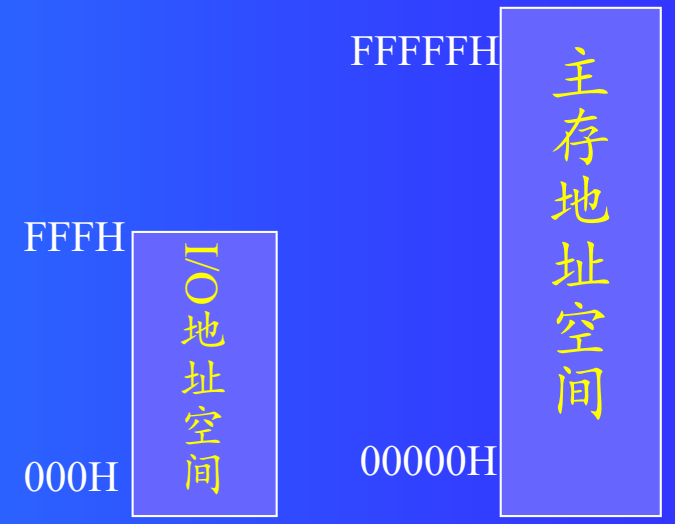
- I/O接口与外设间的数据传送方式
- CPU与I/O接口之间的数据传送方式
- 程序中断方式
- DMA传送方式
- 通道方式与I/O处理机方式

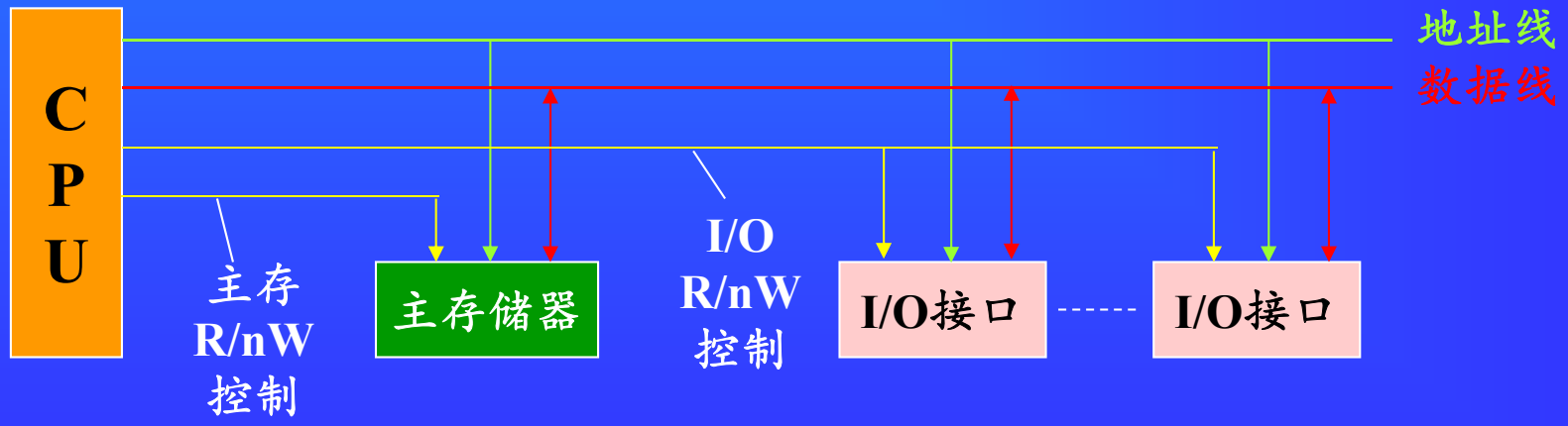
接口和端口

- 端口（Port）：接口电路中可以被CPU直接访问的寄存器
  - 命令口：接口内用于接收来自CPU等主控设备的控制命令的寄存器
  - 状态口：接口内向CPU报告I/O设备的工作状态的寄存器
  - 数据口：接口内在外设和总线间交换数据的缓冲寄存器
- 接口（Interface）：若干端口加上相应的控制逻辑电路

I/O系统的编址方式

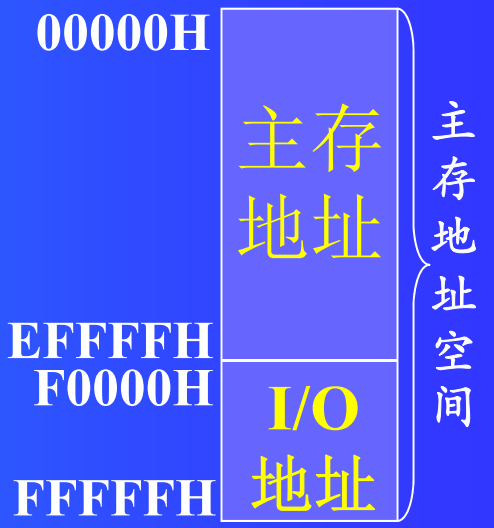
- 独立编址方式
  - I/O与主存分别编址，各自有独立的地址空间
  - 在低地址区，I/O和主存存在相同的地址编码
    - 通过不同的指令访问
    - 通过不同的控制信号区分

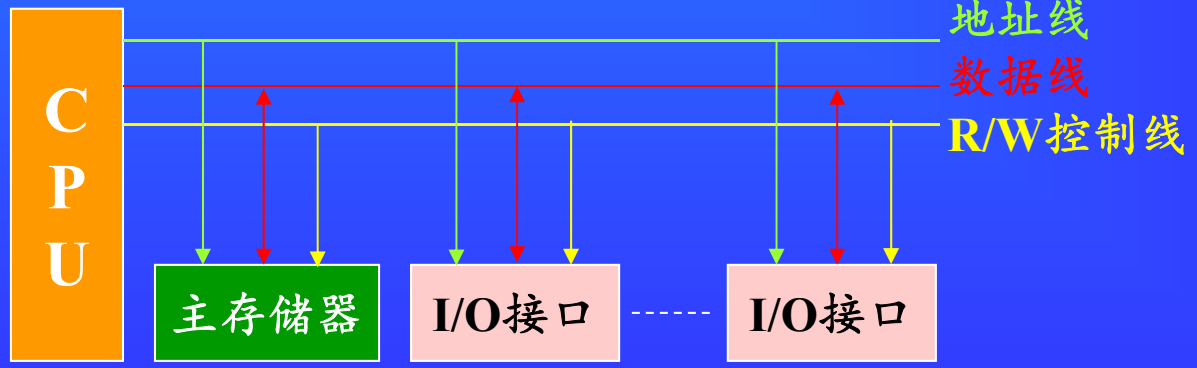




I/O系统的编址方式

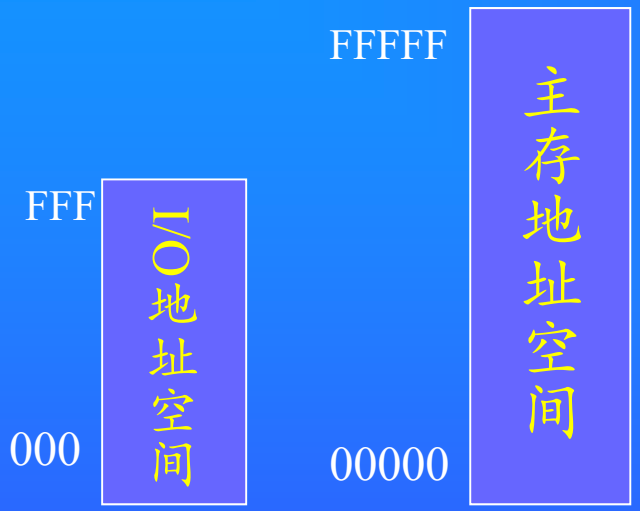
- 统一编址方式（存储器I/O映射）
  - 在主存储器的地址空间中划出某一区域专门作为外设接口地址区使用
  - 外设接口寄存器的地址包含在主存储器的地址空间内
  - 划给外设的这部分区域不能配置存储器芯片





I/O系统的编址方式：对比

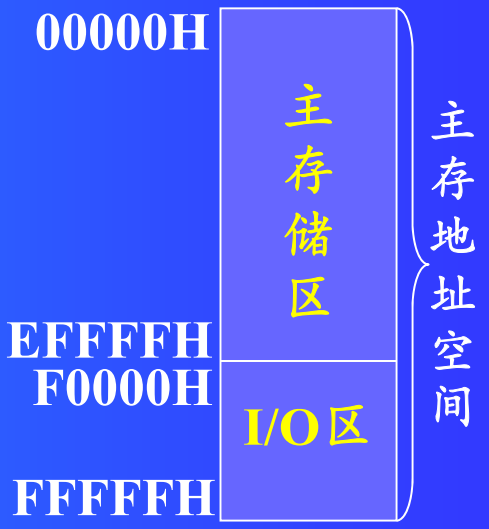
独立编址方式：



- ✓ 优点：I/O不占用主存空间
- ✓ 缺点：需增加I/O指令

实例： Intel 64

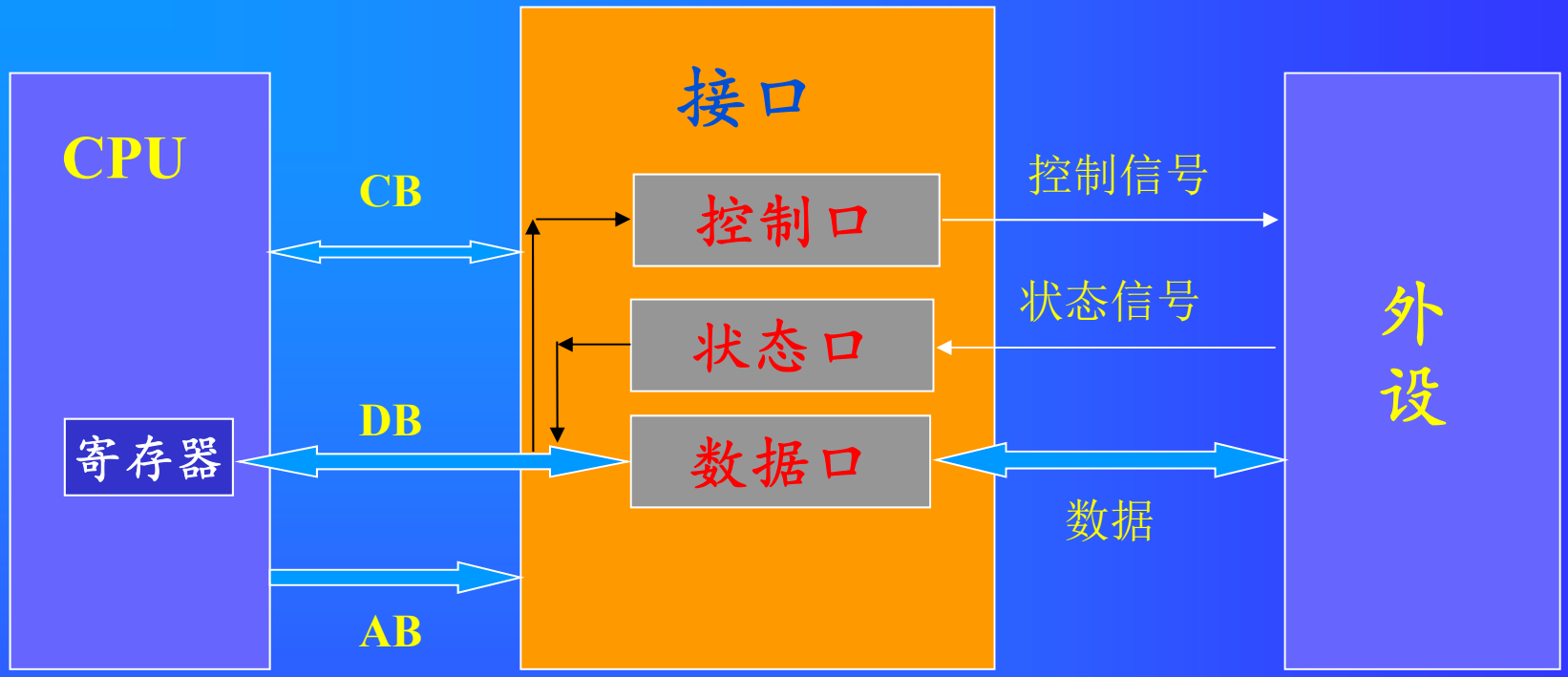
统一编址方式（存储器I/O映射）



- ✓ 优点：提高了CPU访问外设的灵活性
- ✓ 缺点：需要特别关注访存与访问I/O的差异

实例：ARM、PowerPC……

输入/输出操作的两个传输阶段



- I/O接口与外设间的数据传送
- CPU与I/O接口之间的数据传送

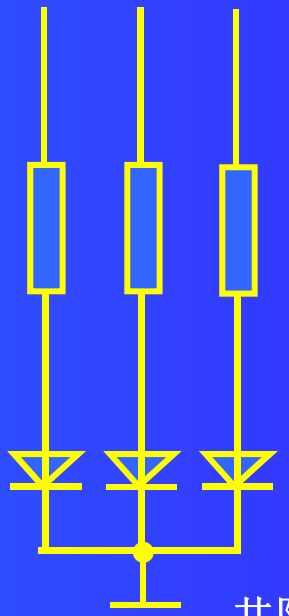
I/O接口与外设间的数据传送

- 无条件传送方式（零线握手联络方式）
- 同步传送方式
- 应答方式（异步传送方式）
  - 双线握手
  - 单线握手
  - 三线握手

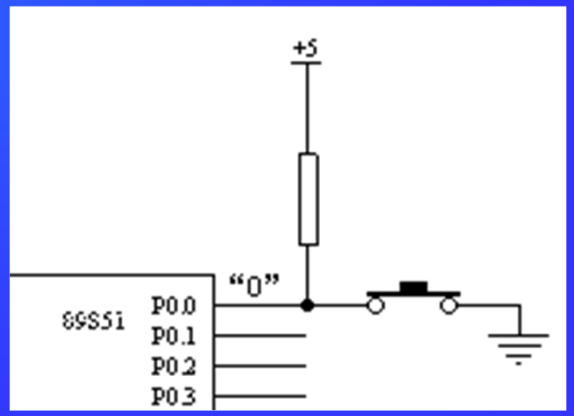
I/O接口与外设间的数据传送

无条件传送方式

- 前提：在任何一次数据交换之前，外设无需进行准备操作
- 通常用于简单的慢速设备
- 无需握手信号线，接口只需实现数据缓冲和寻址功能
- 又称为零线握手联络方式



共阴极



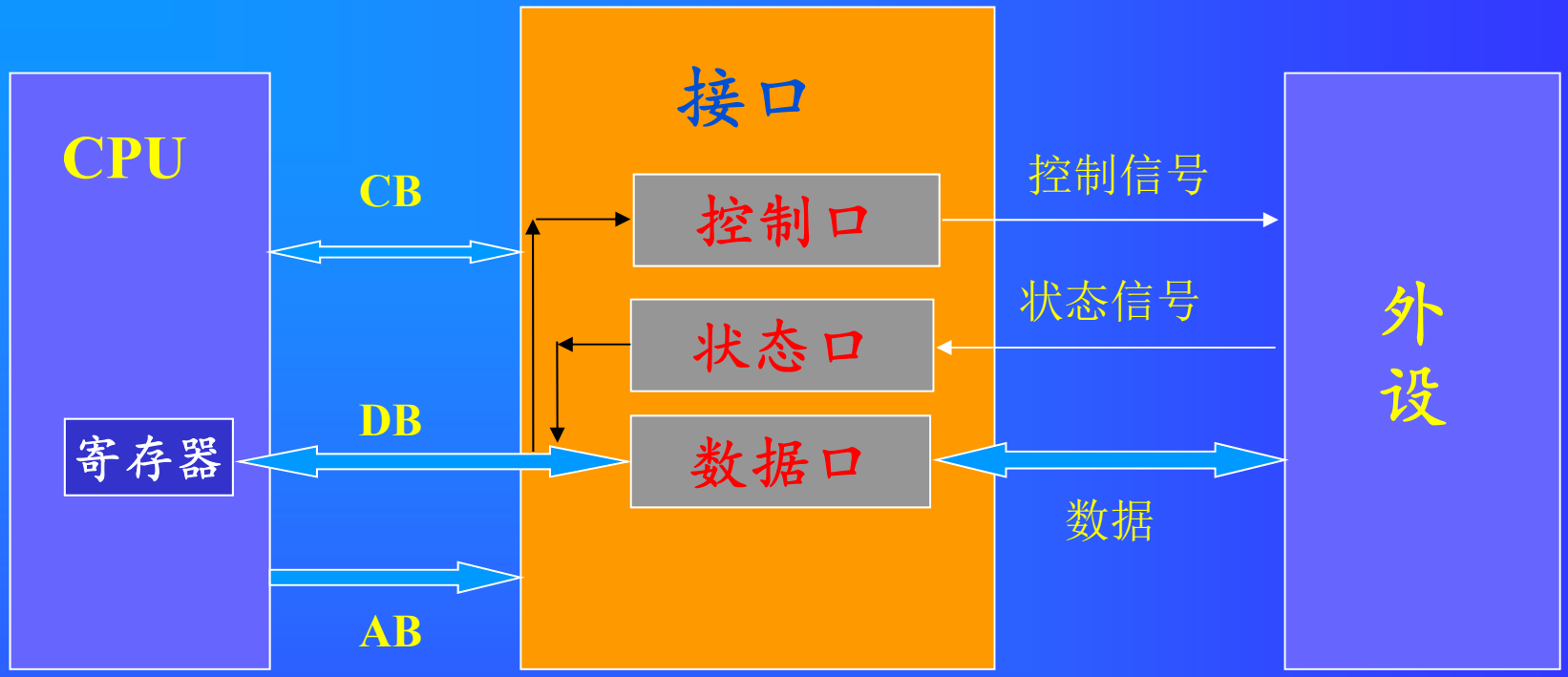
I/O接口与外设间的数据传送

- 同步传送方式
  - 接口以某一确定的时钟速率和外设交换信息
  - 适用于中等以上数据传送速率和按规则间隔工作的外部设备

I/O接口与外设间的数据传送

- 应答方式（异步传送方式）
  - 在数据传送信号线之外另外安排若干条握手（联络、挂钩）信号线，用以在收发双方之间传递控制信息，指明何时能够交换数据
  - 根据握手信号线的条数，可以分为
    - 双线握手
      - 发方向收方发出选通信号或请求信号，指明数据是否有效
      - 收方向发送方发出应答信号，指明数据是否已经被取走
    - 单线握手
    - 三线握手

输入/输出操作的两个传输阶段



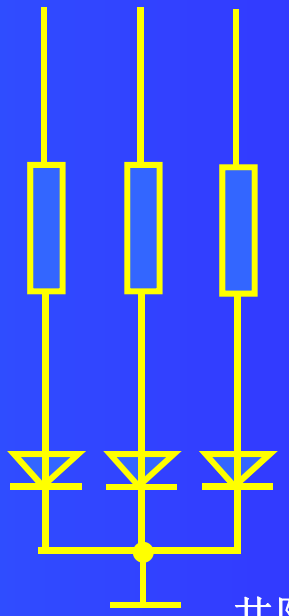
- I/O接口与外设间的数据传送
- CPU与I/O接口之间的数据传送

CPU与I/O接口之间的数据传送

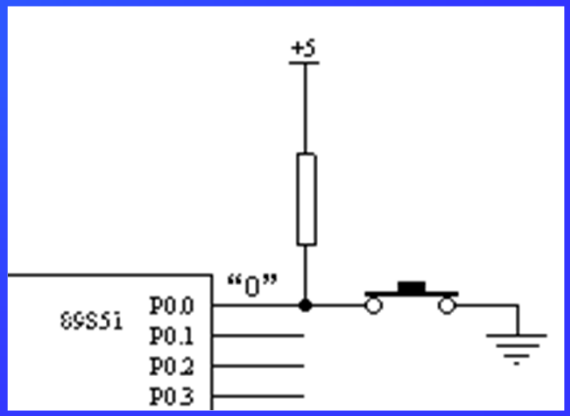
- 简单I/O方式（无条件传送方式）
- 查询（轮询，polling）传送方式
- 中断驱动方式
- 直接存储器存取（DMA）方式
- 通道【输入输出处理机（IOP）】方式
- 外围处理机（PPU）方式

CPU与I/O接口之间的数据传送

- 简单I/O方式（无条件传送方式）
  - 只有当接口与外设之间采用无条件传送方式时，CPU与接口之间才能采用无条件传送方式
  - 在读、写操作之前对目标设备的状态不作任何检测



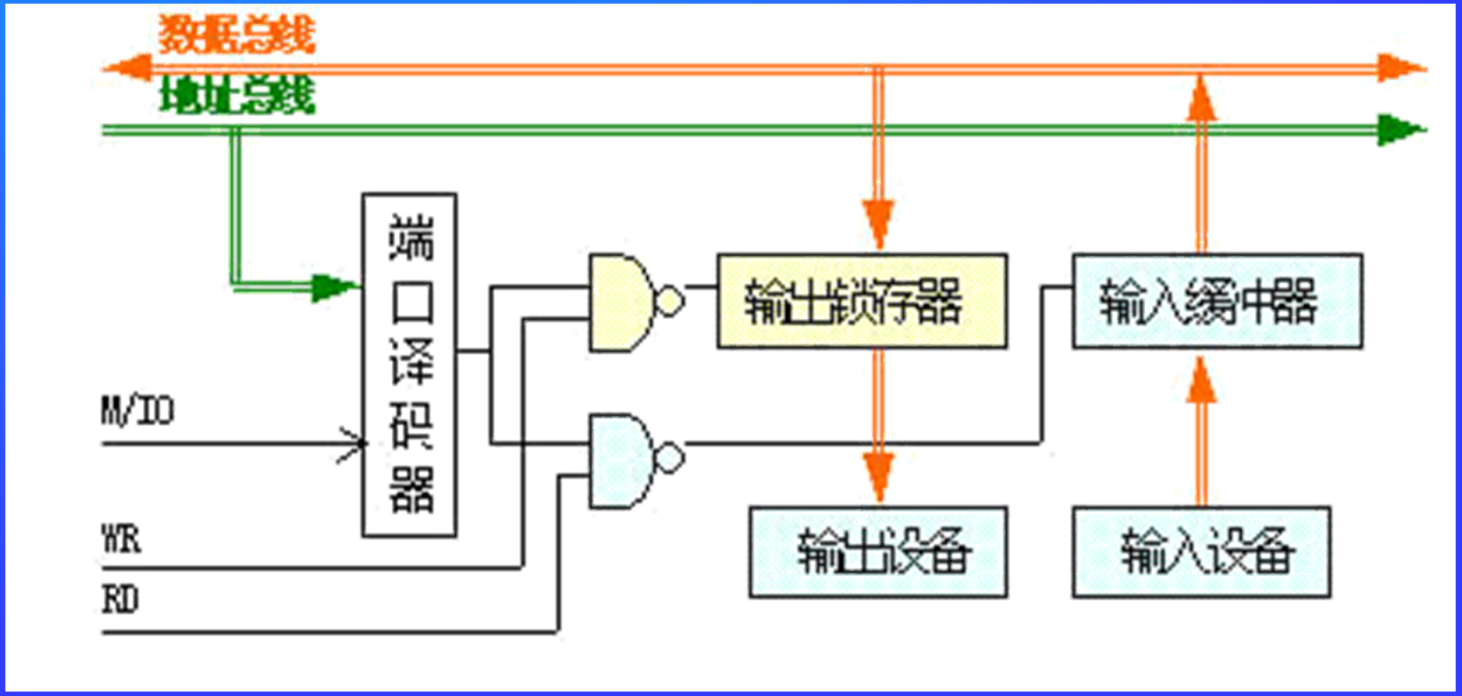
共阴极



CPU与I/O接口之间的数据传送

简单I/O方式（无条件传送方式）

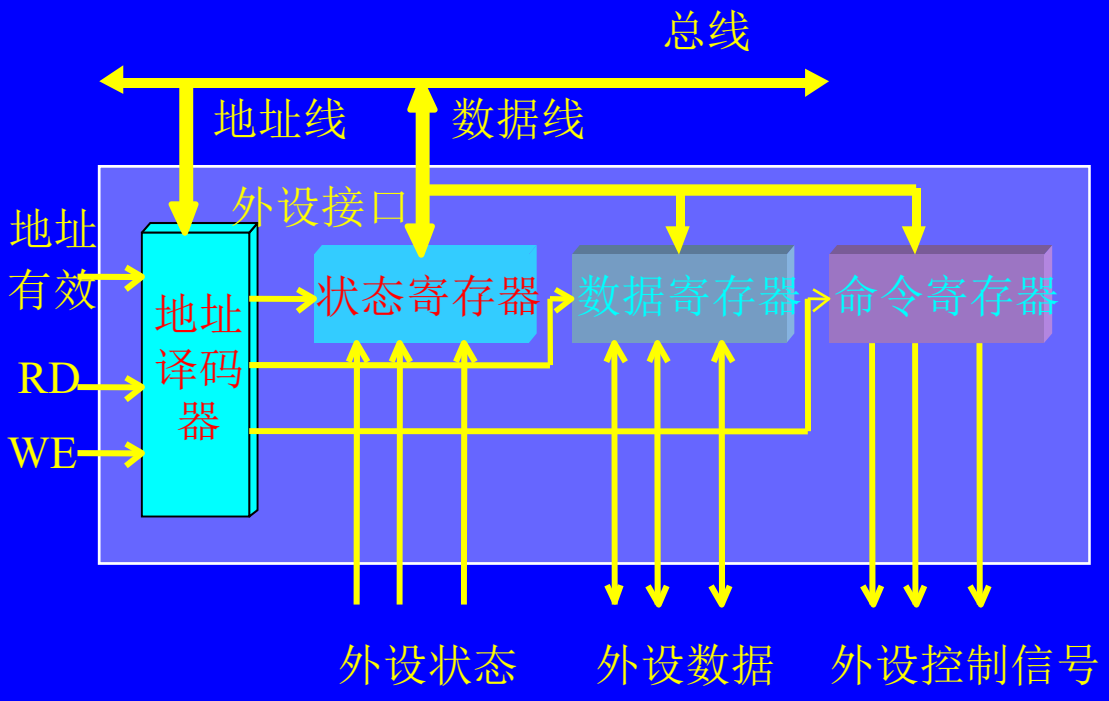
- 当简单外设作为输入设备时，可使用三态缓冲器与数据总线相连
- 当简单外设作为输出设备时，输出须采用锁存器

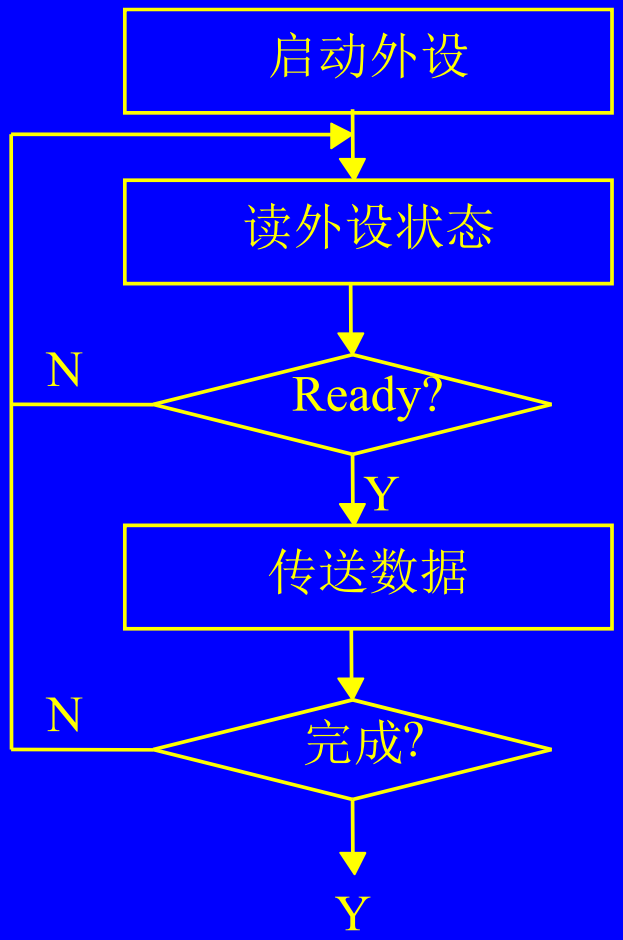


CPU与I/O接口之间的数据传送

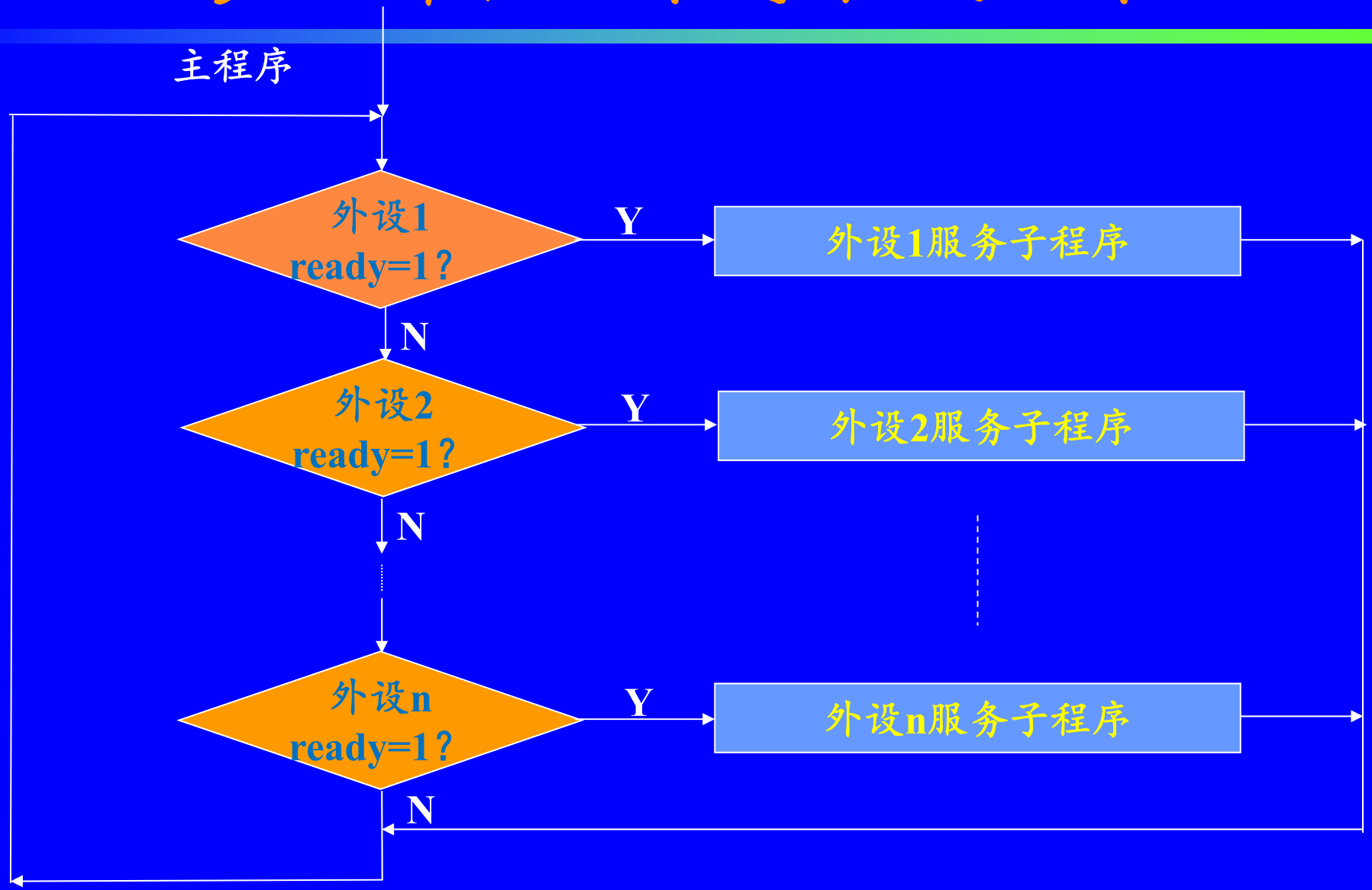
- 查询（轮询，polling）传送方式
  - 多数外设每传送完一次数据总要进行一段时间的处理或准备才能传送下一个数据
  - 在数据传送之前，对目标设备的状态进行查询：
    - 外设已准备好传送数据： 进行数据传送
    - 外设未准备好传送数据： CPU不断地查询并等待
  - 特点：I/O操作由CPU发起，CPU处于主动地位
  - 实现：
    - 外设的状态通过接口反映给CPU
    - 接口电路中需要设置传送外设状态的端口

查询（轮询，polling）传送方式

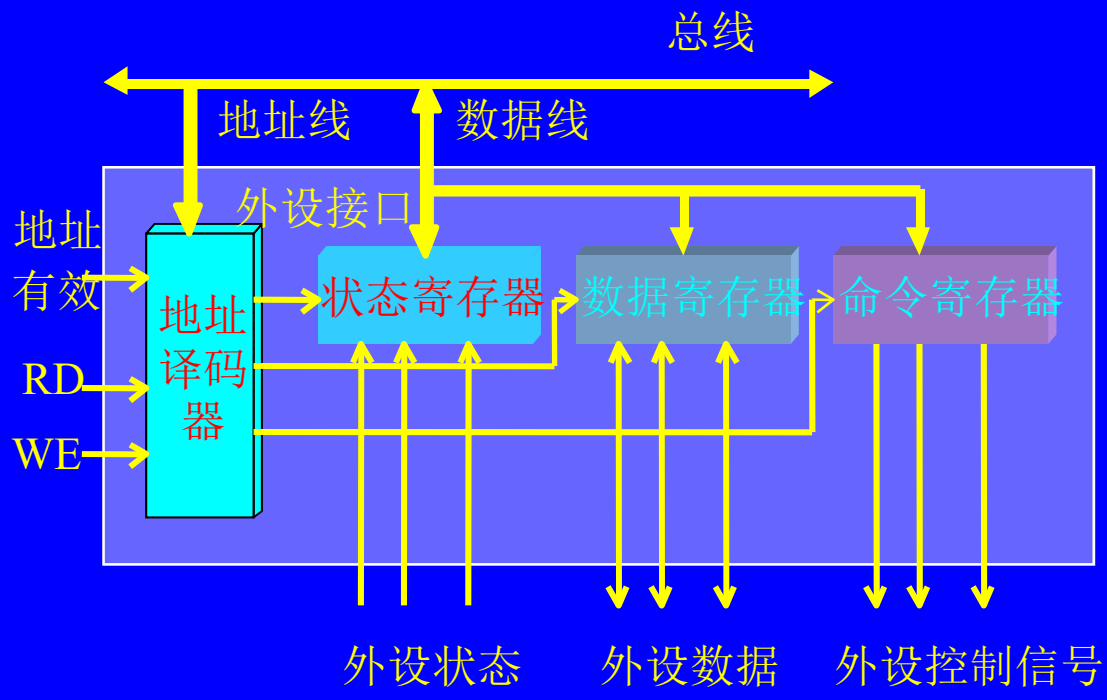


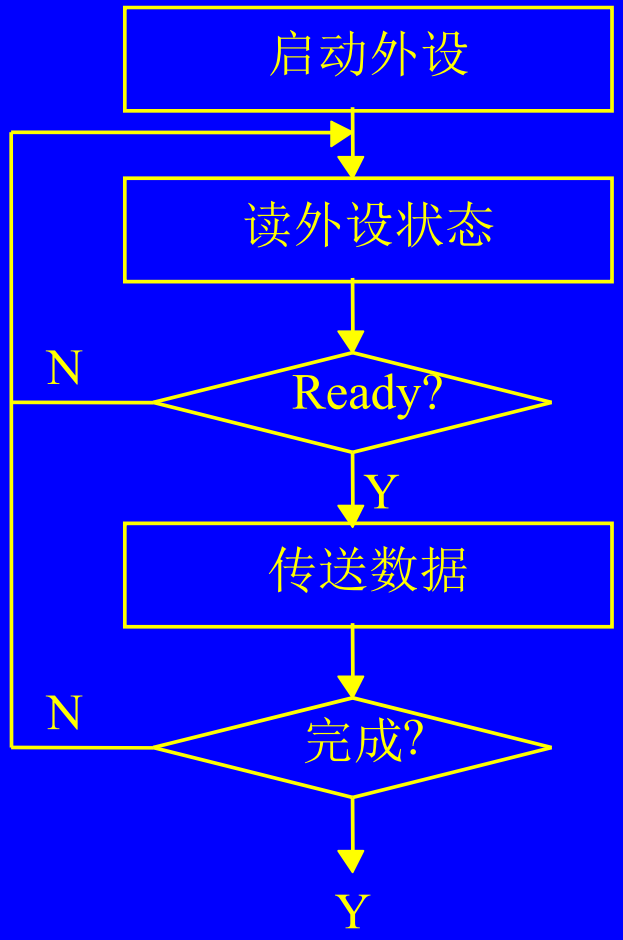


多台外设的程序轮询过程



查询（轮询，polling）传送方式

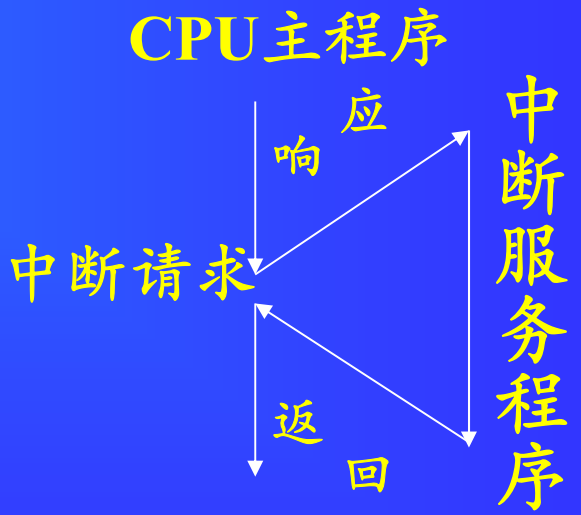




- **特点：**
  - **软硬件简单**
  - **CPU效率低**
  - **用于连接低速外设**

CPU与I/O接口之间的数据传送

- 中断驱动方式
  - 当外设需要CPU服务时，通过接口向CPU发出中断请求
  - CPU在当前机器指令执行完毕后暂停正在执行的主程序，转去执行中断服务程序
  - 待中断服务程序执行完毕，再返回到原程序继续执行

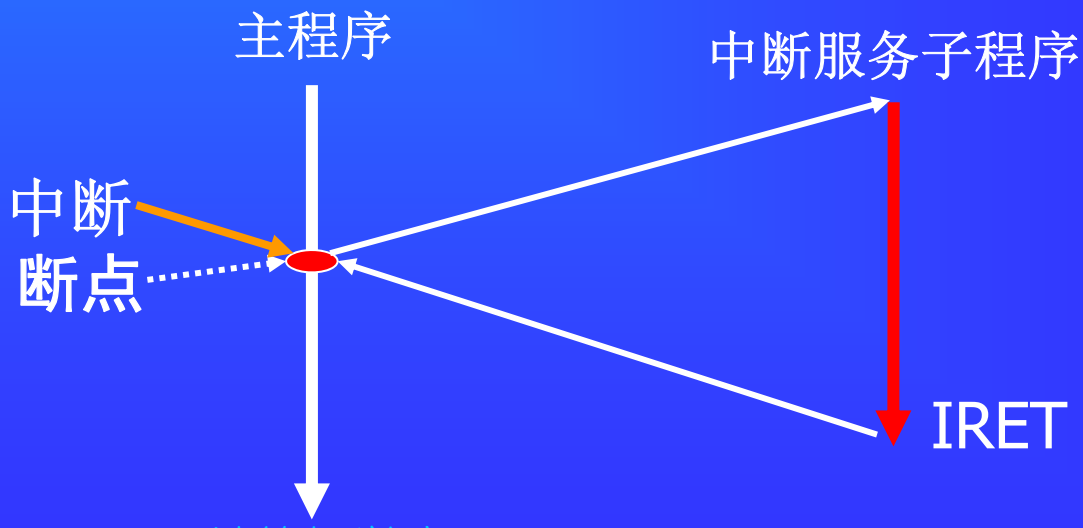


CPU与I/O接口之间的数据传送

- 中断驱动方式
  - 优点
    - 节省CPU的时间，提高计算机的效率
    - 能使I/O设备的服务请求得到及时响应
  - 适合于计算机工作量十分饱满、而I/O处理的实时性要求又很高的系统
  - 缺点
    - 硬件复杂度提高
      - 增加中断控制器等逻辑电路
    - 软件复杂度提高
      - 软件的开发和调试比程序查询式复杂和困难

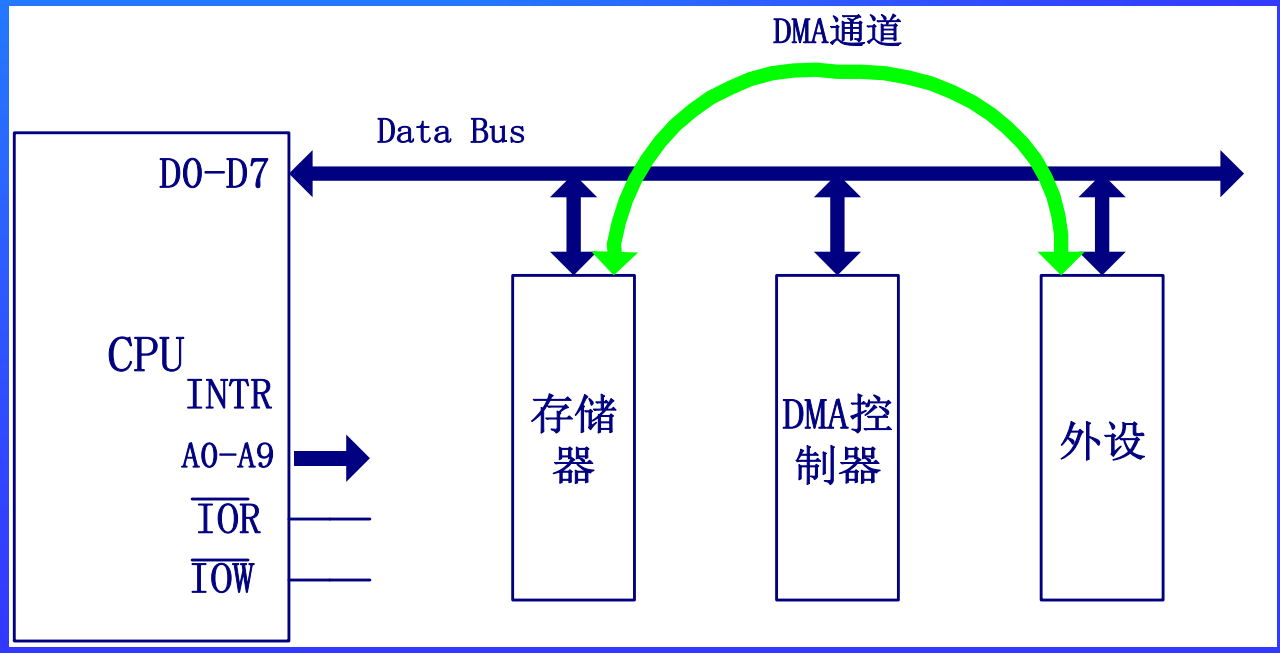
CPU与I/O接口之间的数据传送

- 中断方式的局限性：在高速、成批传送数据时，中断方式不能满足速度要求
  - 通过CPU执行程序来实现数据传送
    - 每进行一次传送，CPU必须执行一遍中断处理程序，完成一系列取指令、分析指令、执行指令的过程
  - 每进入一次中断处理程序，CPU都要保护断点和状态条件寄存器
  - 在中断处理程序中，通常要保护及恢复通用数据寄存器
  - 在指令流水方式中，中断发生或从中断返回时，指令队列预取的指令全部作废

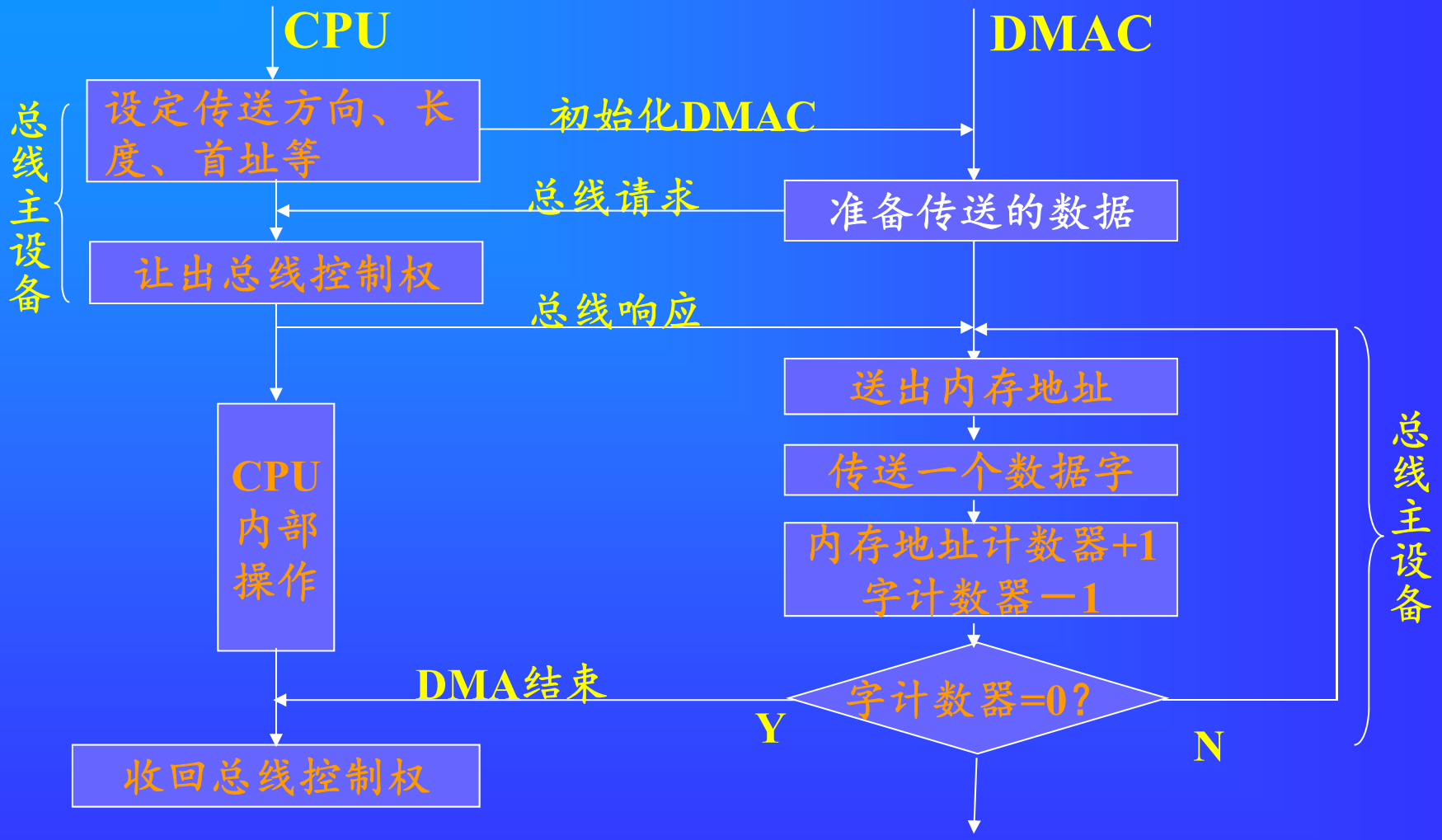


CPU与I/O接口之间的数据传送

- DMA（Direct Memory Access，**直接存储器存取**）**方式**
  - **在外设与主存之间开设直接的数据通路，由硬件（DMA控制器，DMAC）执行I/O交换**
  - **DMAC从CPU接管对总线的控制权**



CPU与I/O接口之间的数据传送



CPU与I/O接口之间的数据传送

- DMA（Direct Memory Access，**直接存储器存取**）**方式**
  - **特点：**
    - 数据交换不经过CPU
    - 速度快，适用于高速批量数据传送
      - **在批量数据传送过程中，无需CPU干预**
    - 硬件线路复杂

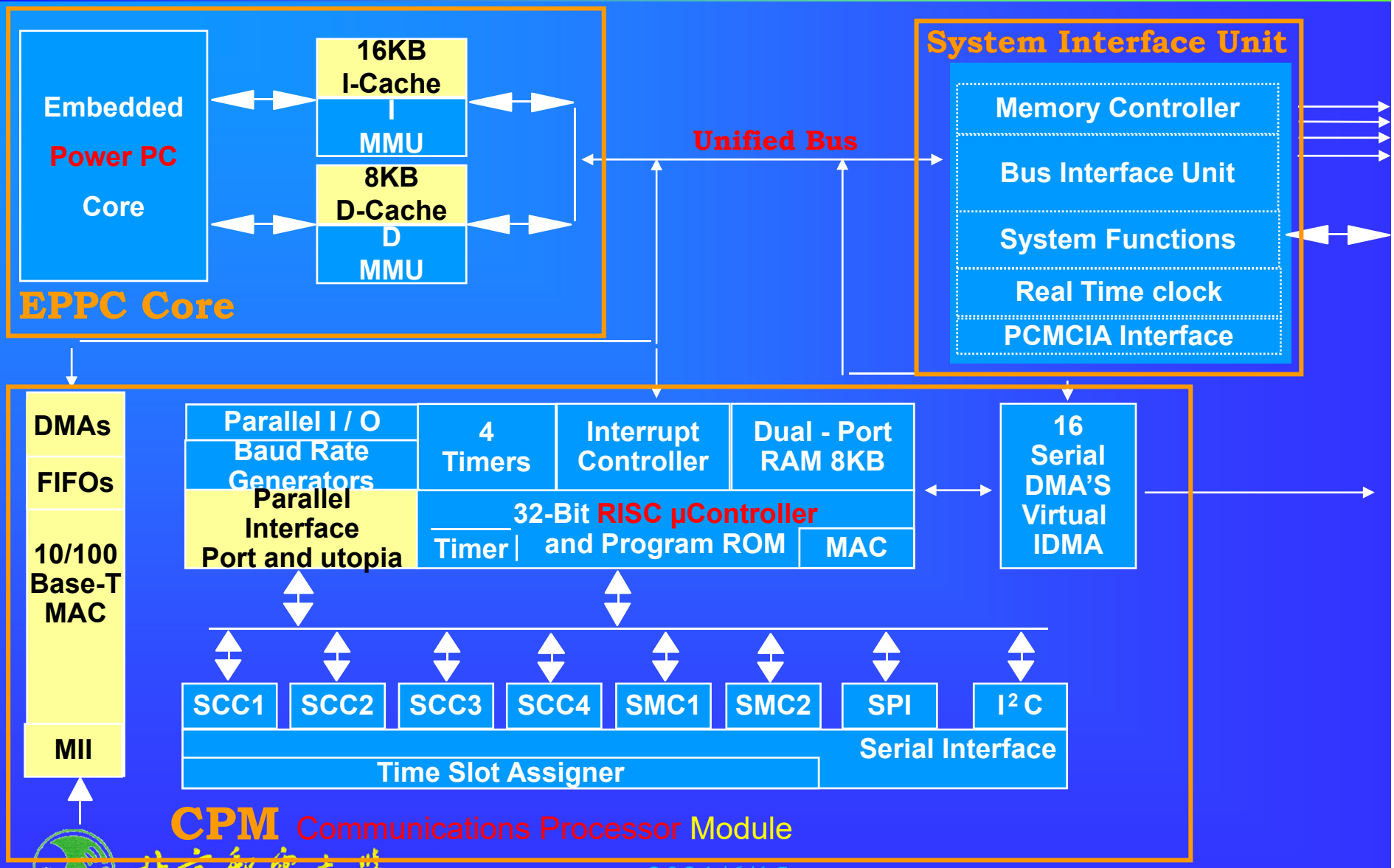
CPU与I/O接口之间的数据传送

- 通道方式（I/O Channel Control）
  - 通道（通道控制器）：
    - 大、中型系统中专门进行I/O控制及数据传送的硬件
    - 具有特定功能的处理器（IOP）
    - 有自己的指令和程序
    - 代替CPU管理、调度外设与主机交换信息
  - 通道方式进一步减轻了CPU的负担
  - 一个通道可以连接控制多台外设，系统效率高

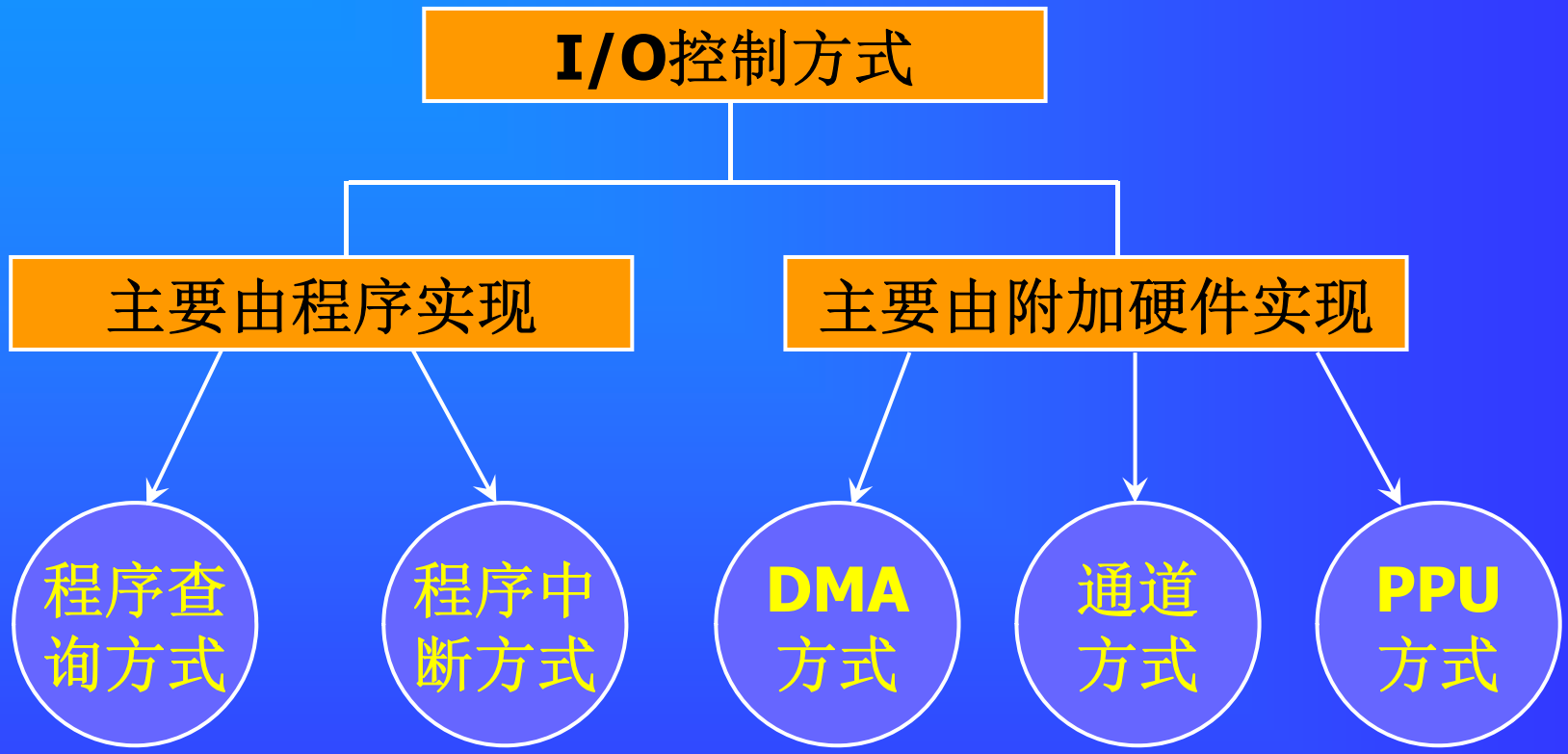
通道结构的发展

- 通道与CPU间并行度的提高
  - 早期的通道：通道与CPU共用主存
  - 后来的通道：为通道配置自己的局部存储器
- 输入输出处理机IOP
  - 功能与一般CPU类似，但专用于对I/O过程的控制处理
  - 细分：
    - I/O处理器（IOP）
      - 主机中的一个部件，分担对I/O设备与I/O过程的管理
    - 外围处理机（PPU，Peripheral Process Unit）
      - 具有更强的处理功能，基本上完全独立于主机，结构更接近一般处理机
      - 有自己的指令系统，可以完成算/逻运算、读/写主存等

Freescale/NXP MPC860P PowerQUICC Integrated Communications Processor



输入/输出方式的比较



中断（异常）

interrupt (exption)

中断的本质：实现程序随机切换

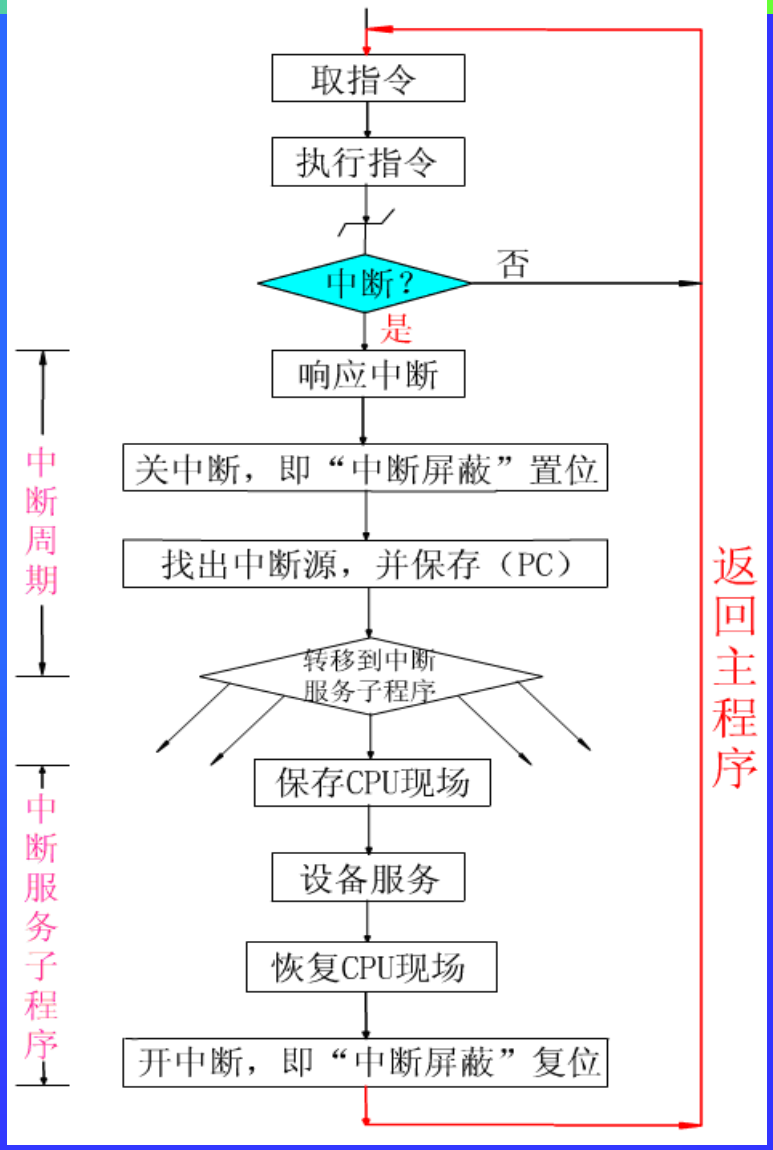
中断（异常）方式的典型应用

- 实现CPU与外界进行信息交换的握手联络
  - 实现CPU与外设的并行工作
  - 对于慢速I/O设备，使用中断方式可以有效提高CPU的效率
- 故障处理
  - 常见的硬件故障：掉电、校验错、运算出错等
  - 常见的软件故障：溢出、地址越界、非法指令等
- 实时处理
  - 在事件出现的实际时间内及时地进行处理
- 程序调度
- 软中断（程序自愿中断、自陷）
  - 与子程序调用的功能相似，但调用接口简单，不依赖于程序入口地址，便于软件的升级维护和调用

中断处理的典型过程

- 中断请求
- 中断响应（中断处理的隐操作）
  - 暂时中止现行程序，并发出中断响应信号（进入中断周期）
  - 关中断
  - 保护断点和PSWR
  - *查找中断源，中断排队与判优*
  - *获取中断服务程序入口地址*
- 中断处理
  - 保护现场
  - 中断服务
  - 恢复现场
- 开中断
- 中断返回

也可由软件完成

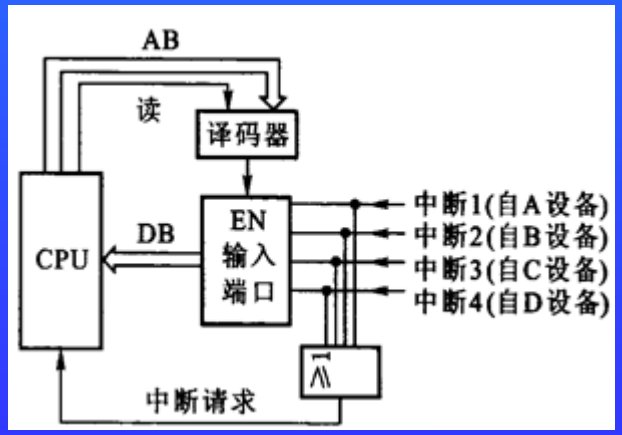


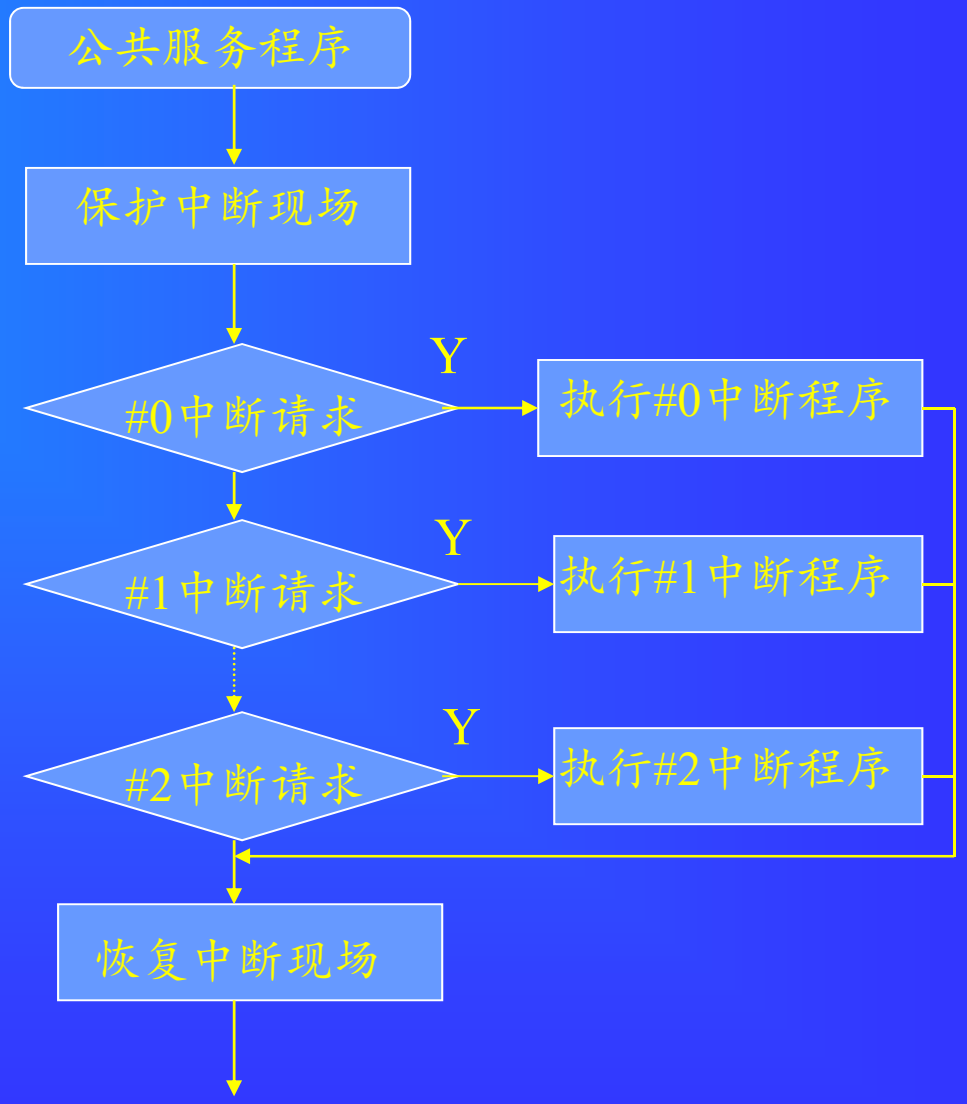
中断周期由硬件实现

中断服务程序由机器指令序列实现

中断服务程序入口的获取

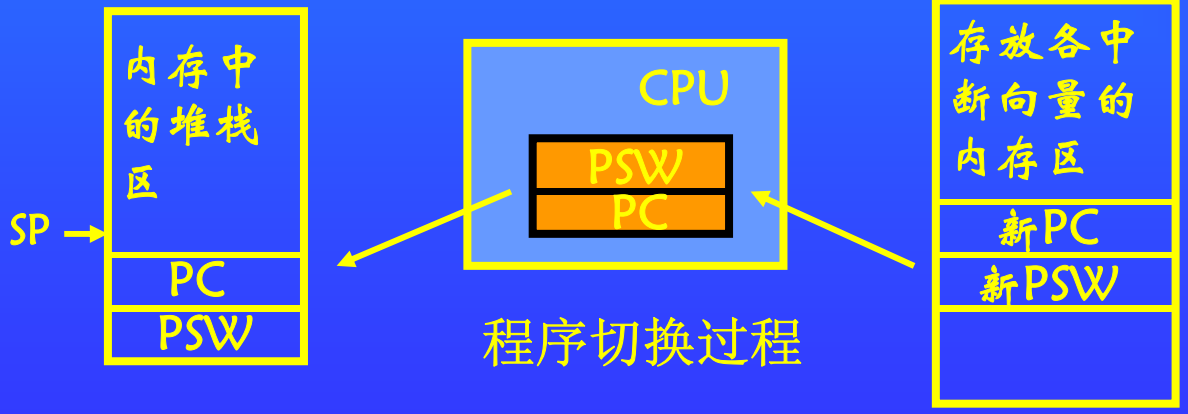
- 查询中断：
  - 硬件不直接提供中断服务程序的入口地址，由软件查询中断源并跳转至相应中断服务程序入口





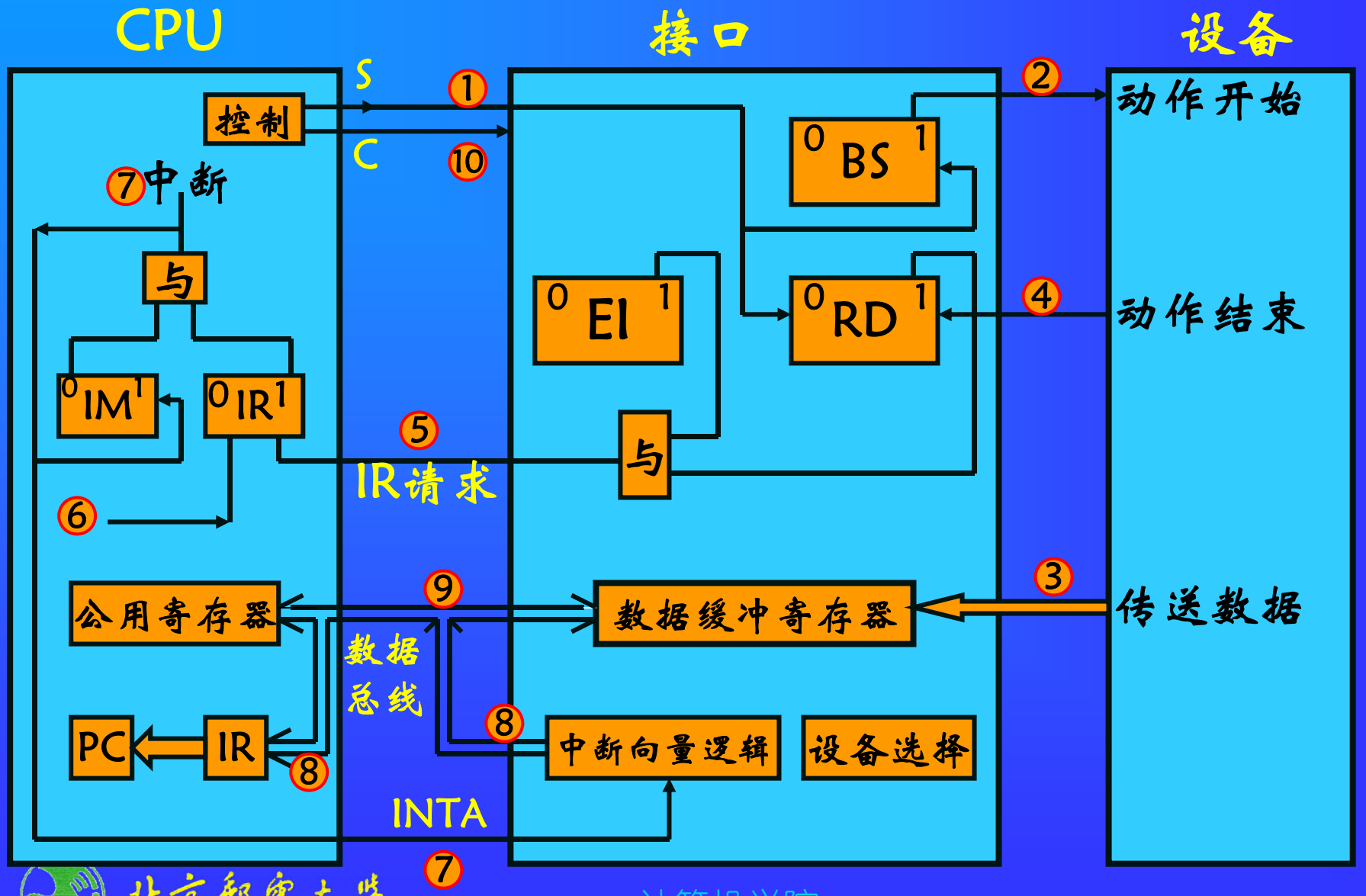
中断服务程序入口的获取

- **向量中断：**
  - CPU响应中断后，由中断机构自动将相应中断源的中断向量送入CPU，由其指明中断服务程序入口地址并实现切换
  - 中断向量：与中断服务程序入口地址（及初始PSW）相关的参数

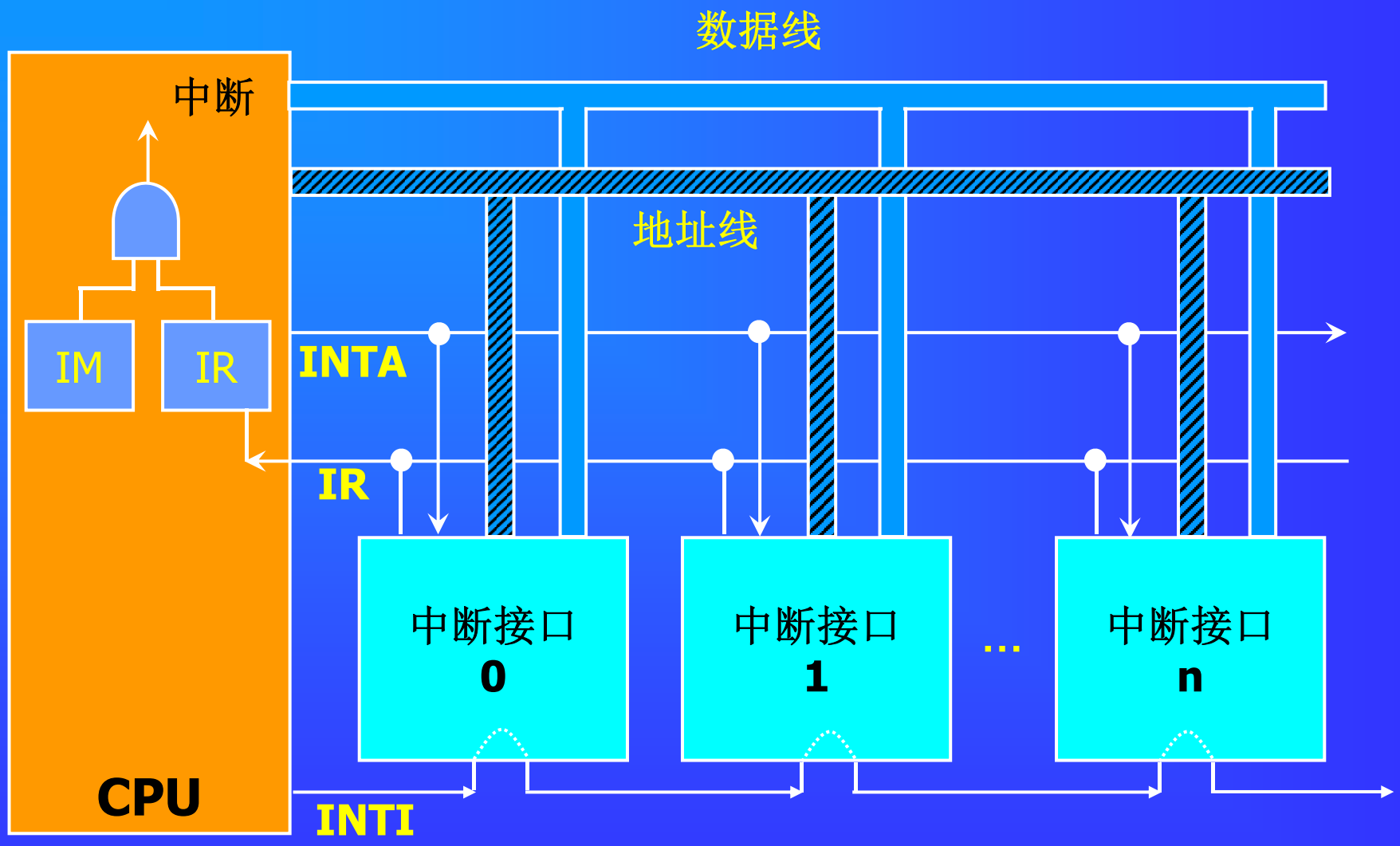


程序切换过程

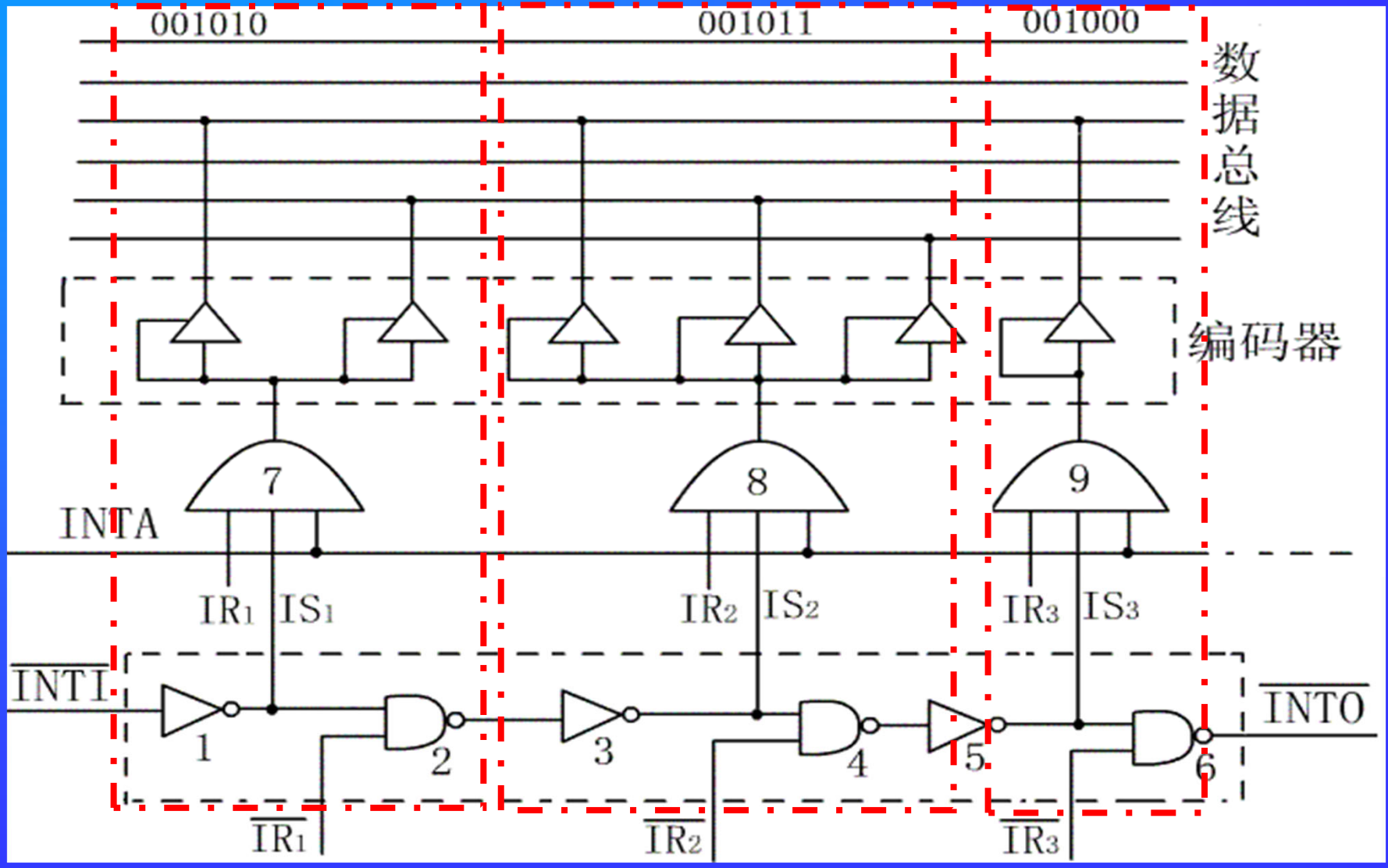
程序中断方式的基本接口



中断判优与排队：菊花链式优先排队逻辑



菊花链式优先排队逻辑及中断向量的产生



中断判优与排队：并行优先排队逻辑

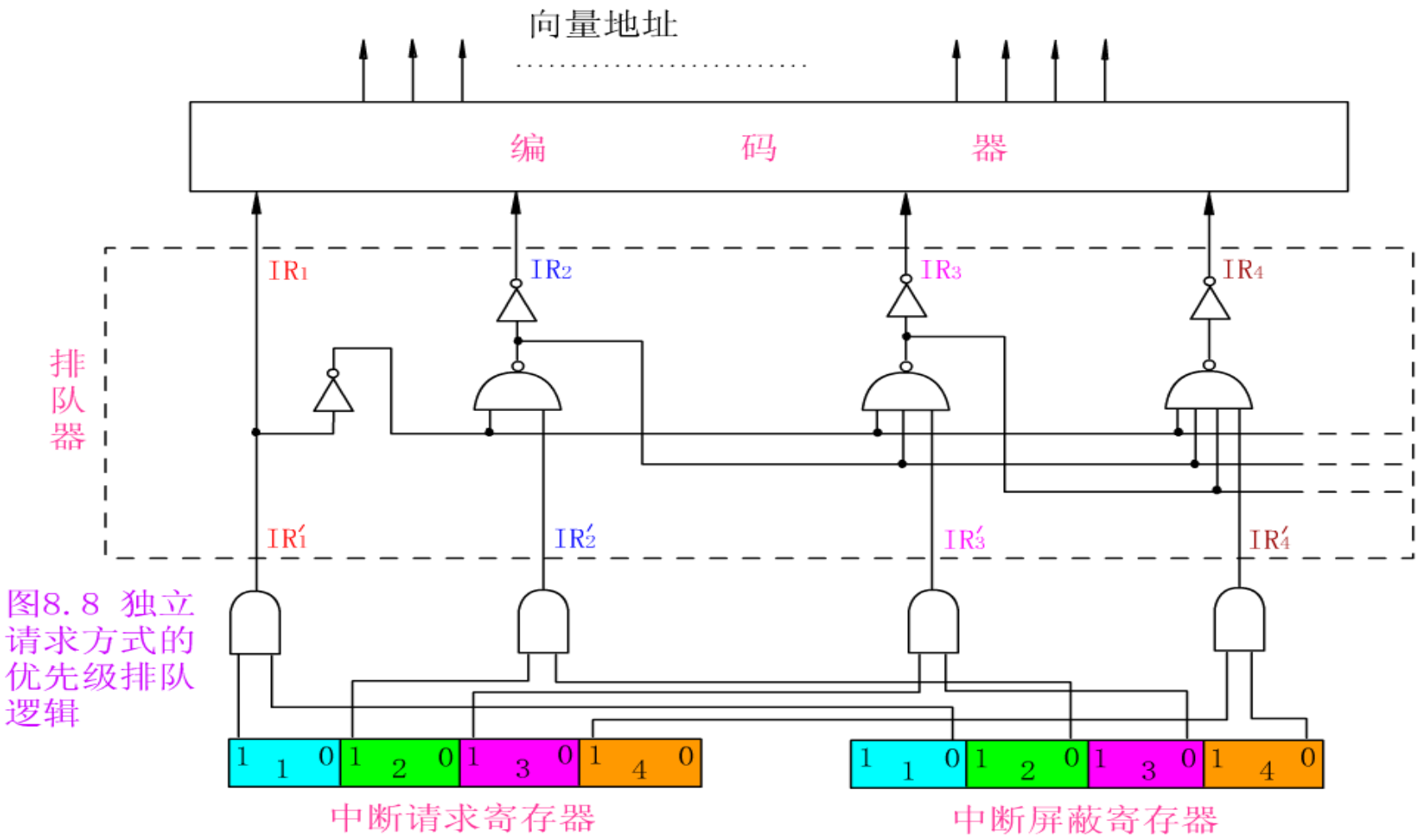
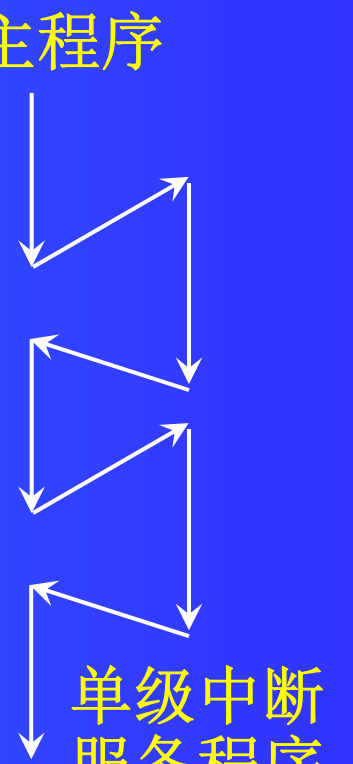


图8.8 独立请求方式的优先级排队逻辑

中断处理：单级中断与多级中断

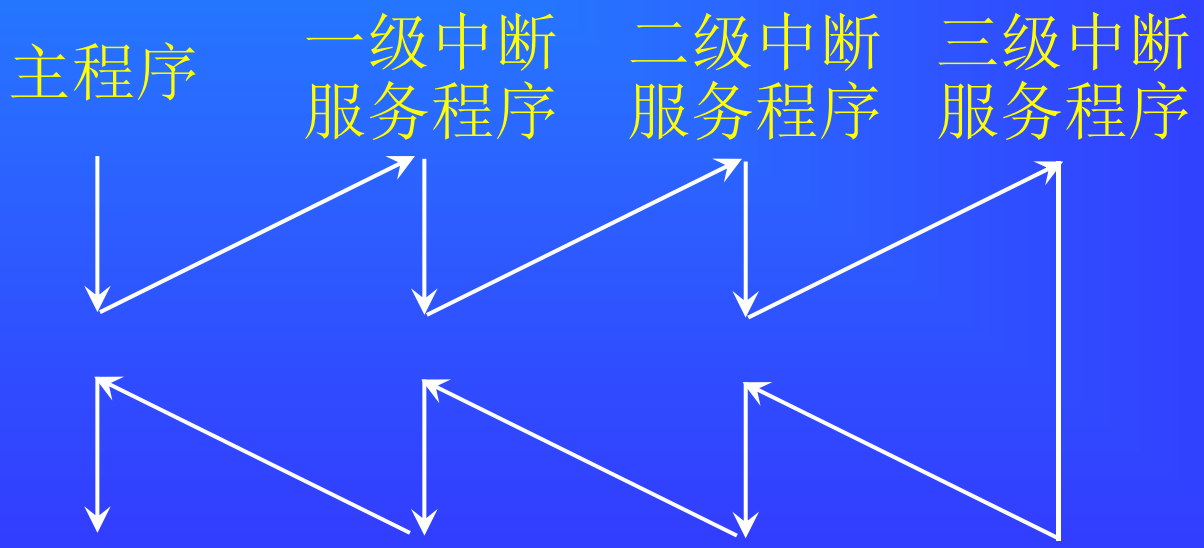
单级中断系统

- CPU响应某个中断请求后，只能为该请求源服务，不允许被其它中断请求打断
- 只有本次中断服务全部完成并返回原程序后，CPU才能响应新的中断请求
- 非强占式优先，不允许中断嵌套
- 在中断周期中已由硬件关中断，所以在本次服务过程中不再响应新的中断请求；直到本次服务完毕、中断返回之前才开中断



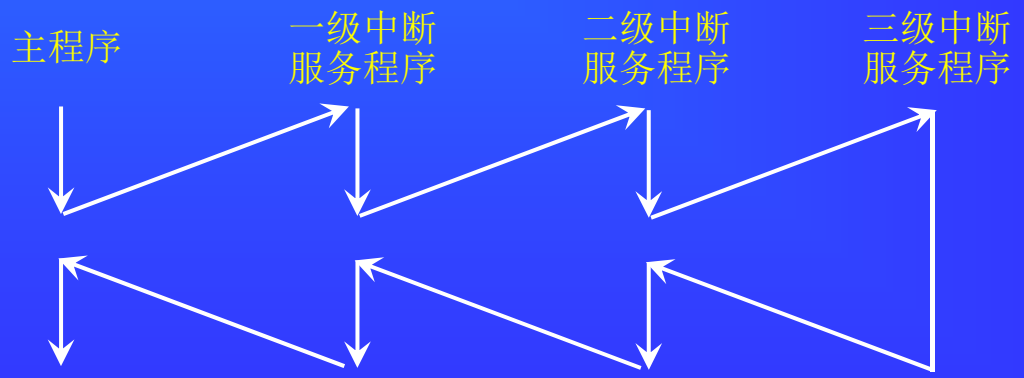
中断处理：单级中断与多级中断

- 多级中断系统
  - 中断处理过程中，允许响应并处理优先级别更高的中断请求
  - 优先权高的中断级可以打断优先权低的中断服务程序
  - 强占式优先，允许中断嵌套



中断处理：单级中断与多级中断

- 多级中断系统的实现
  - 在保护现场后，送出新的屏蔽字，屏蔽掉与本请求同一优先级别以及更低级别的其它请求
  - 软件开中断，再开始本请求源所要求的服务处理
  - 如果在处理过程中，CPU又接到优先级更高的新请求，就可以暂停正在执行的服务程序，保存其断点，转去响应新的中断请求



一维多级中断

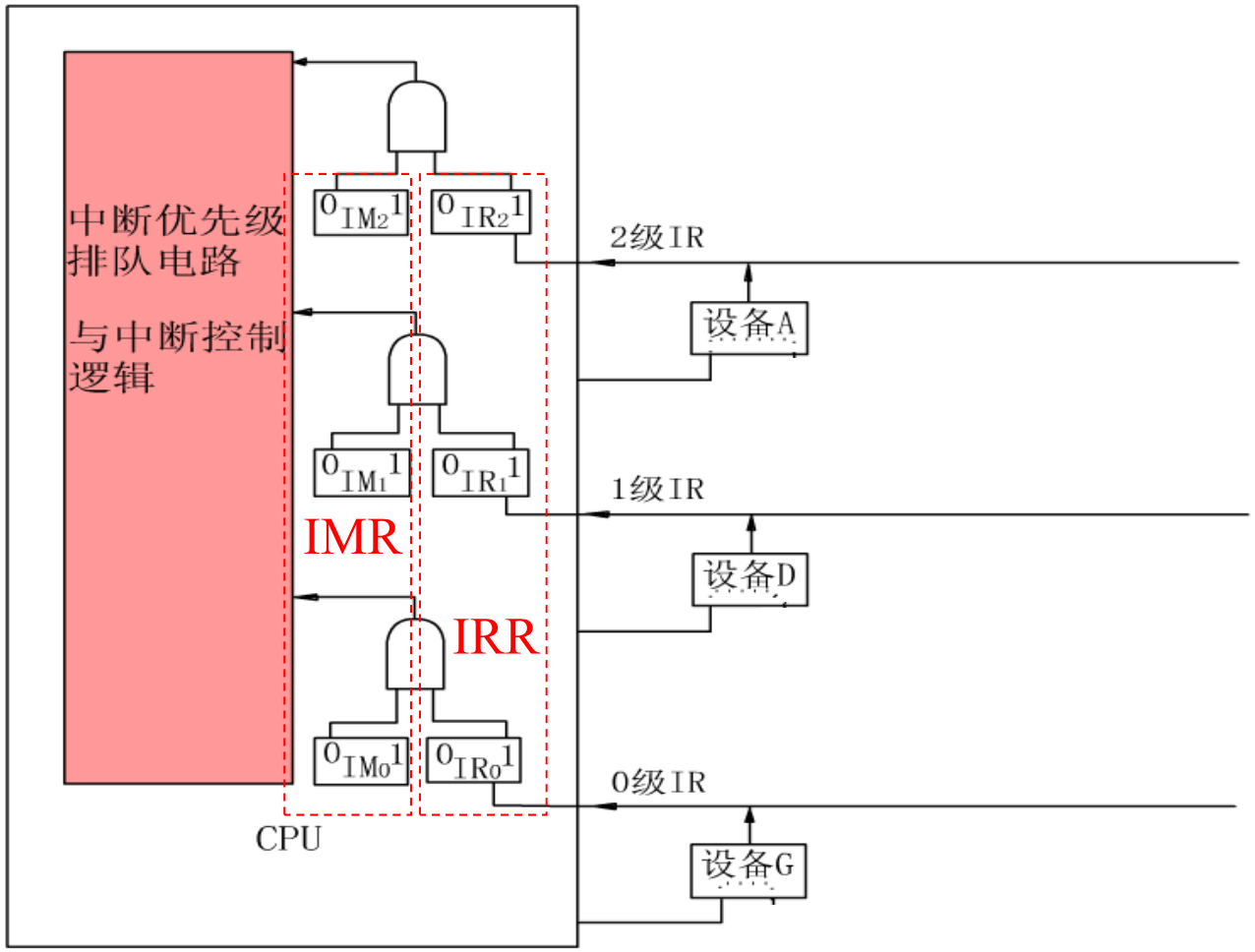
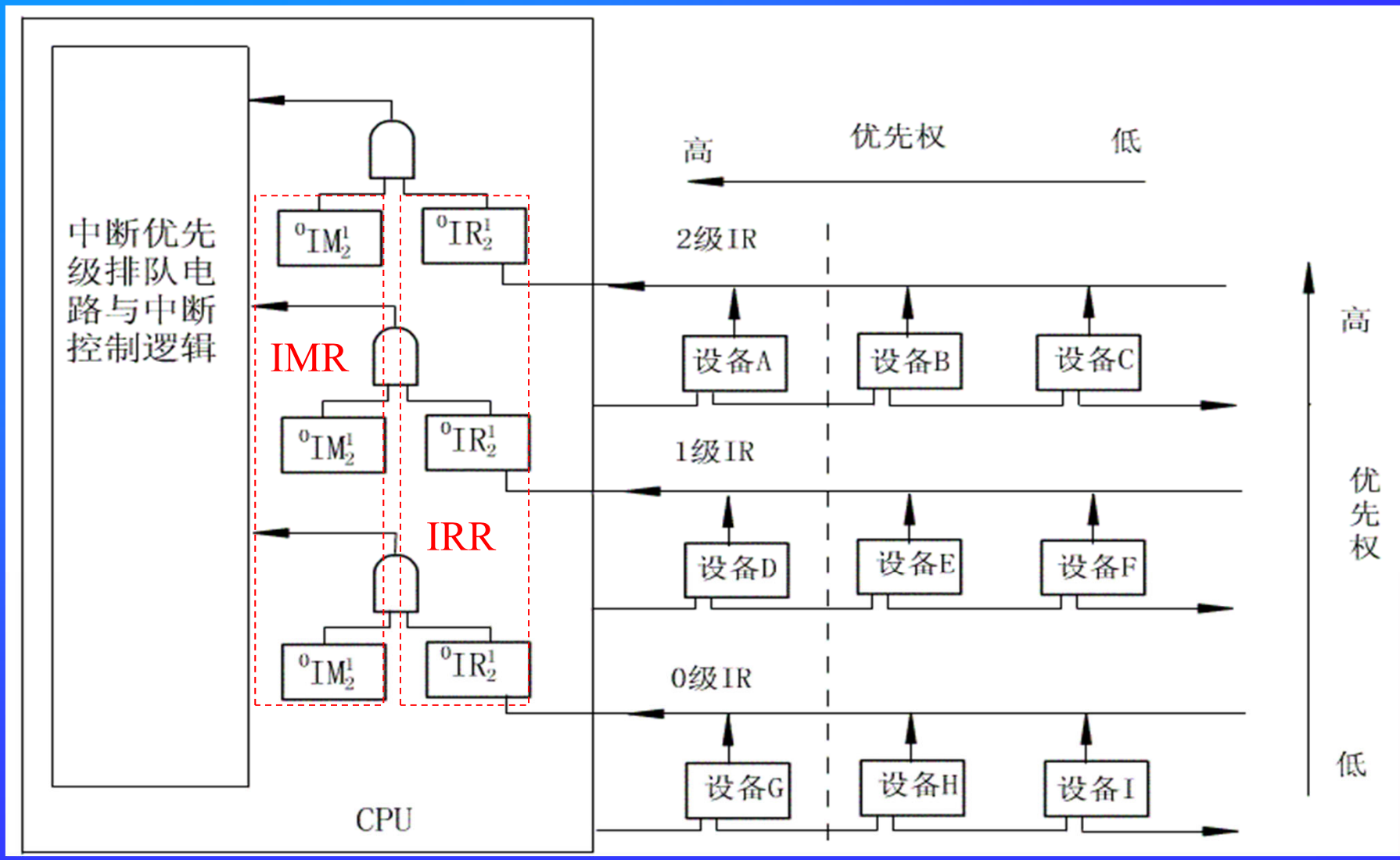


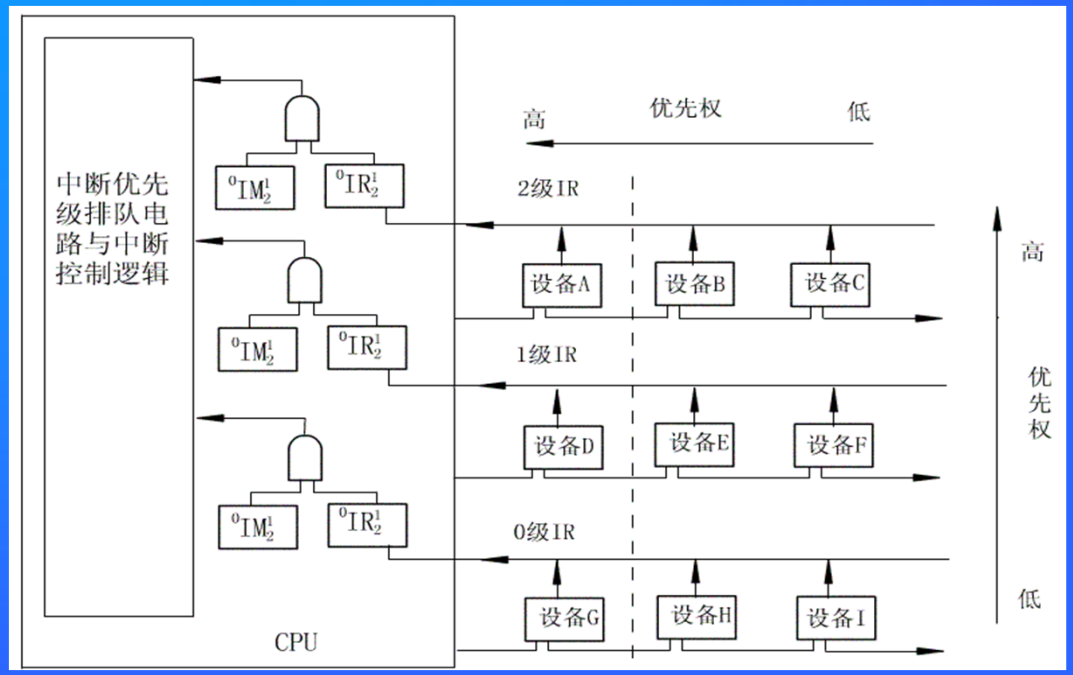
图8.7 多级中断

一维多级中断结构

二维多级中断


中断优先级排队电路与中断控制逻辑
CPU
IMR
IRR
高
优先权
低
2级IR
1级IR
0级IR
设备A
设备B
设备C
设备D
设备E
设备F
设备G
设备H
设备I
高
优先权
低

二维多级中断实例



【例1】如图所示的二维中断系统：

(1)在开中断情况下，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。

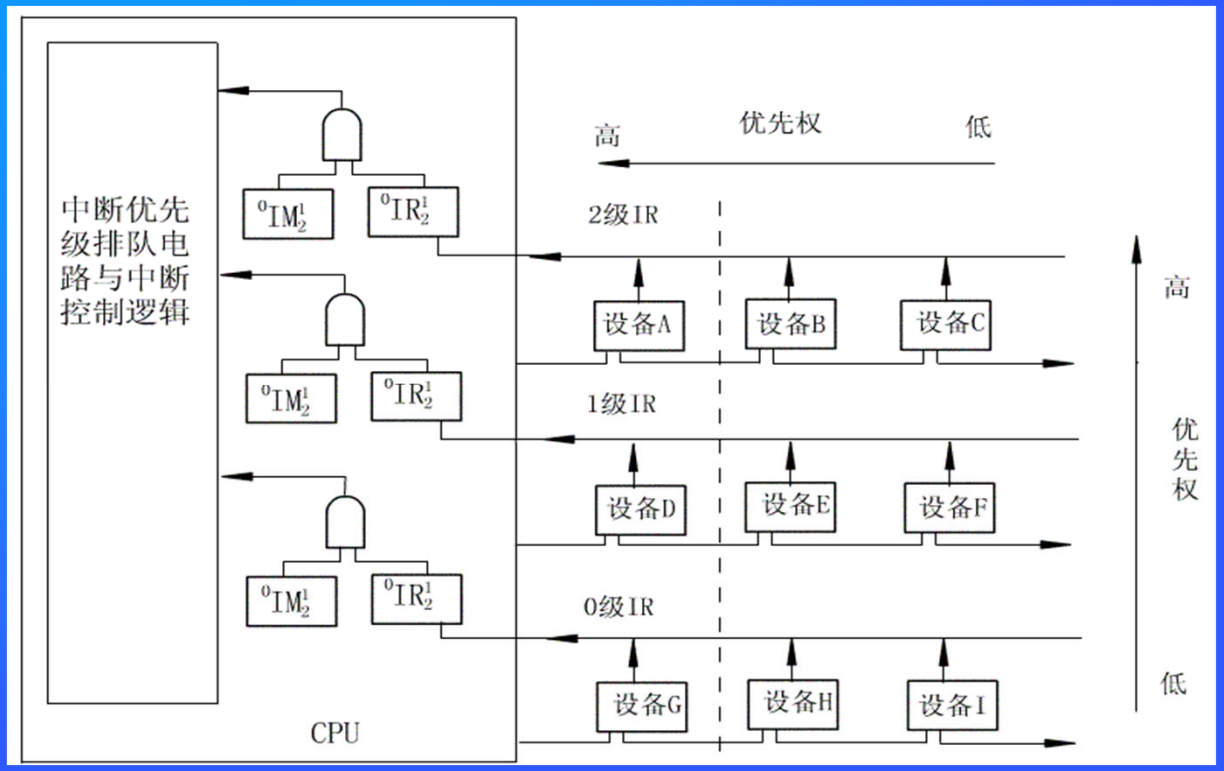
【解】

1. 在开中断情况下，CPU的优先级最低。

   各设备的优先次序是：

   A→B→C→D→E→F→G→H→I→CPU

二维多级中断实例



【例1】如图所示的二维中断系统：

(2)若CPU现执行设备B的中断服务程序，IM2、IM1、IM0的状态是什么？

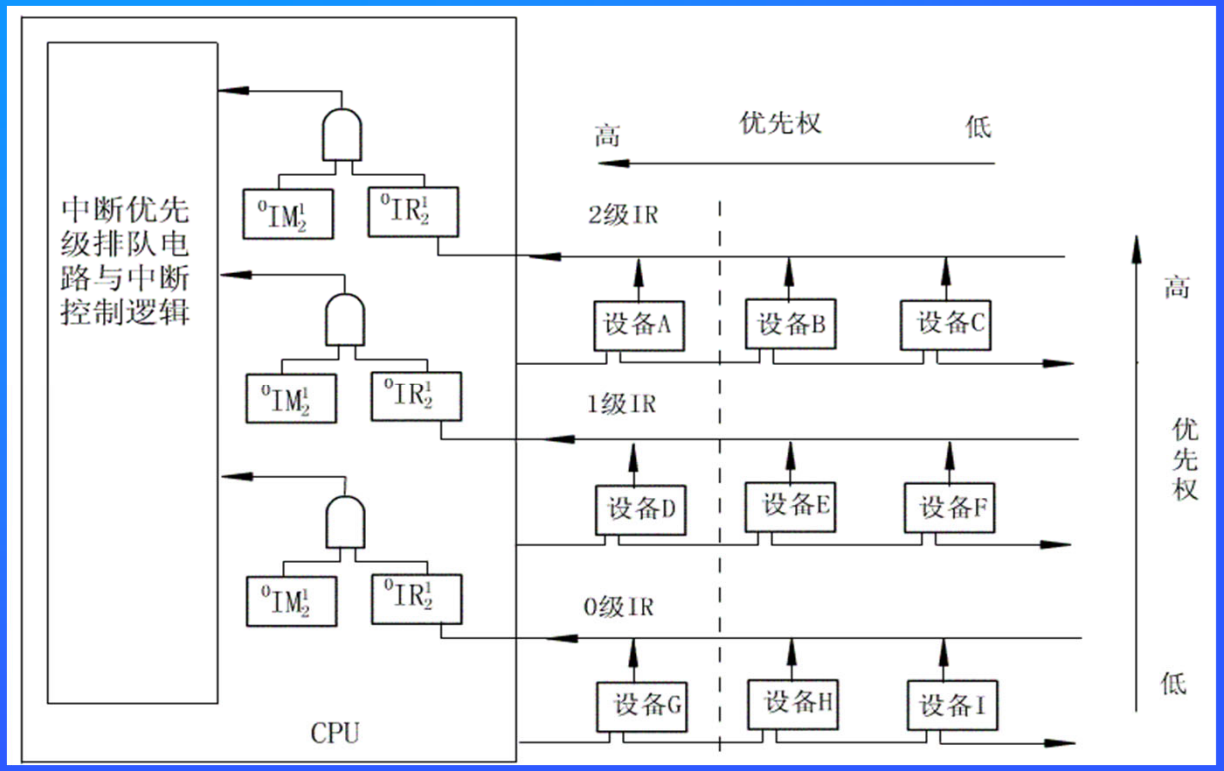
如果CPU执行设备D的中断服务程序，IM2、IM1、IM0的状态又是什么？

【解】

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：$IM_2IM_1IM_0=011$

二维多级中断实例



【例1】如图所示的二维中断系统：

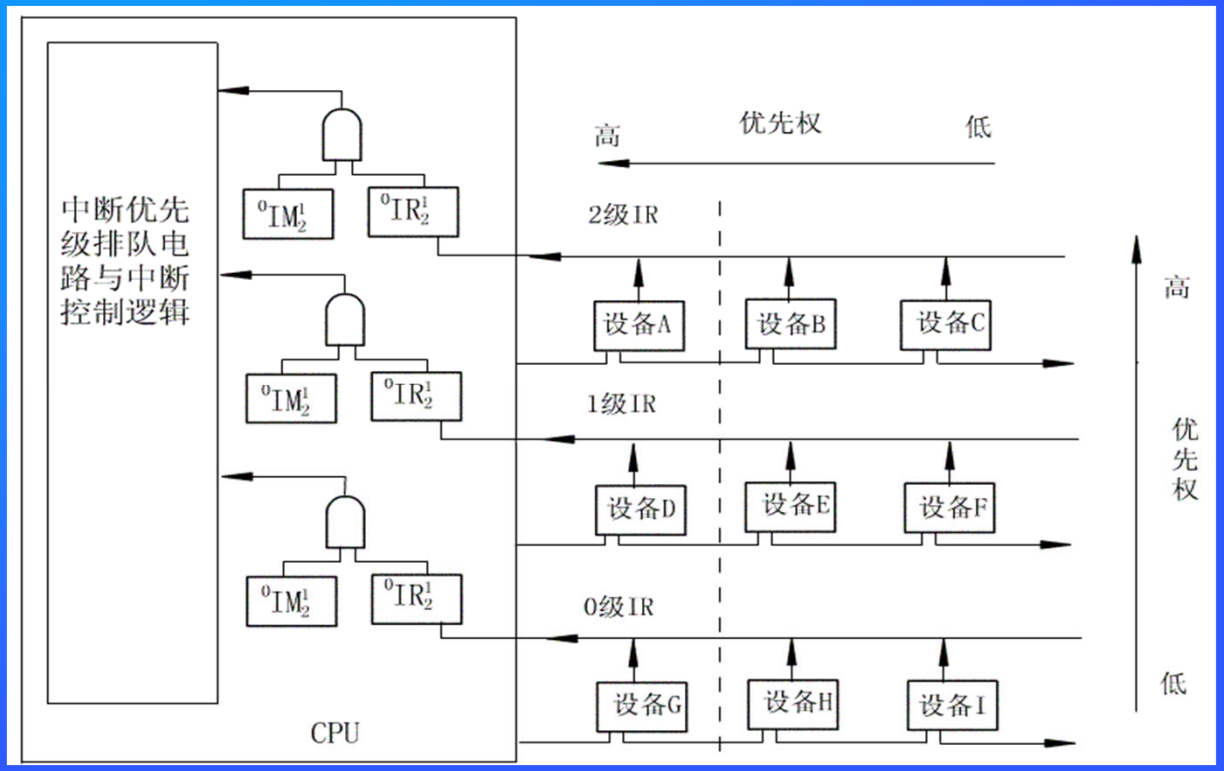
(3)每一级的IM能否对某个优先级的个别设备单独进行屏蔽?如果不能，采取什么办法可达到目的?

【解】

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”，禁止设备发出中断请求

二维多级中断实例



【例1】如图所示的二维中断系统：

(4)假如设备C一提出中断请求，CPU需立即进行响应，如何调整才能满足此要求?

【解】

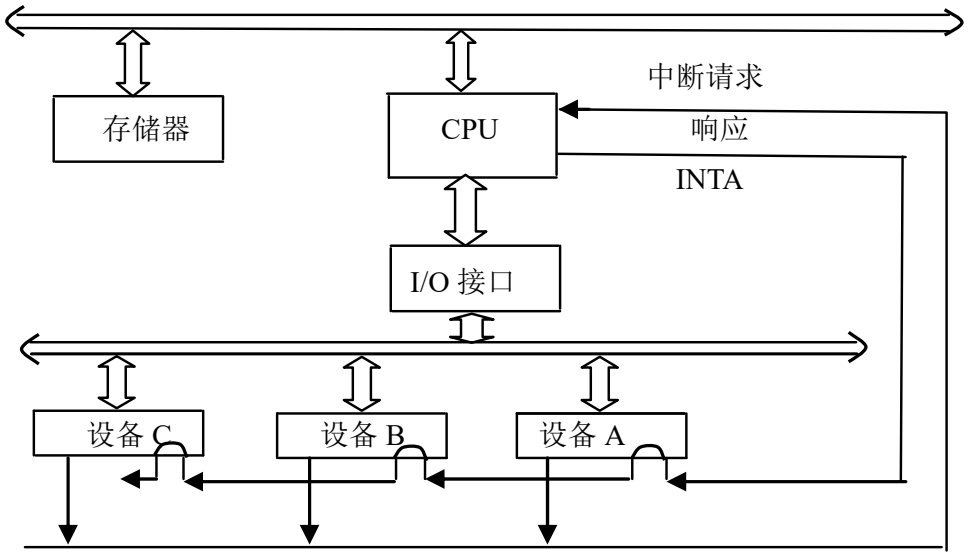
4。要使设备C的中断请求及时得到响应，可将设备C从第2级取出来，单独放在第3级上，使第3级的优先级最高，即令$IM_3=0$即可。

二维多级中断实例

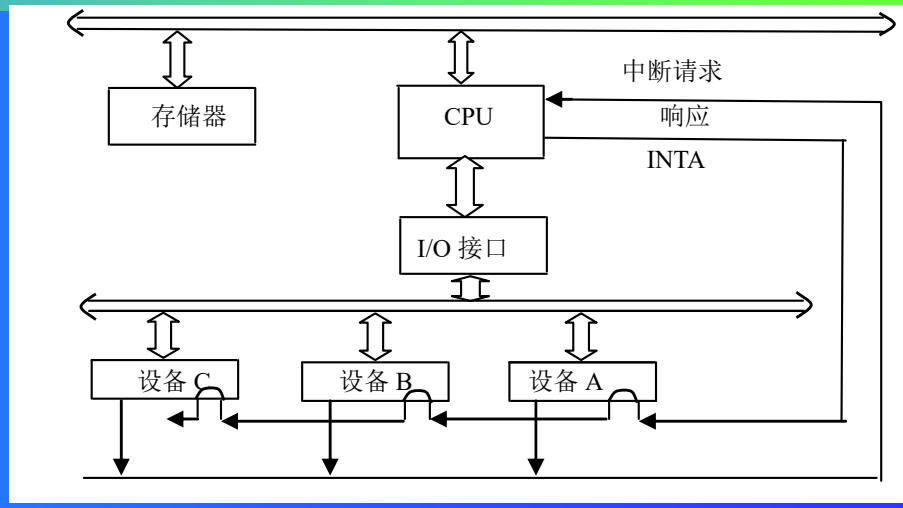
【例2】如图所示的系统，A、B、C三个设备组成单级中断结构，它要求CPU在执行完当前指令时对中断请求进行服务。假设：

1. CPU"中断批准"机构在响应一个新的中断之前，一定要让被中断的程序的一条指令执行完毕；
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**？



中断处理流程


存储器
CPU
中断请求
响应
INTA
I/O 接口
设备 C
设备 B
设备 A

指令
周期
取指令
T_M
执行指令
T_M
中断周期
中断
$n \times T_{DC}$
硬件实现
T_S
保护现场
中断服务
程序
T_A, T_B, T_C
设备服务程序
T_R
恢复现场

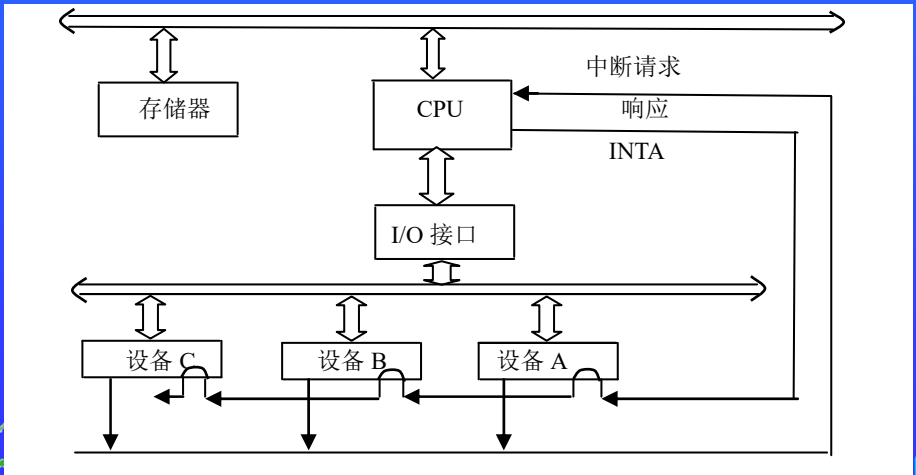
二维多级中断实例

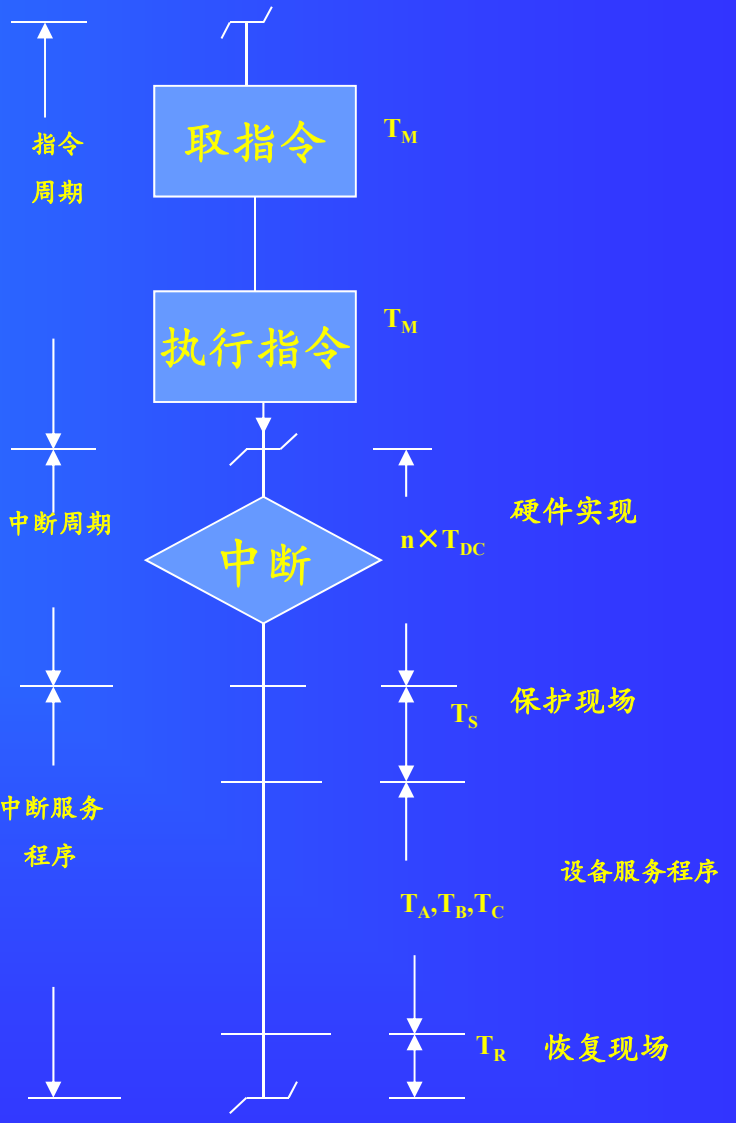
- 答：
  - $t_A = 2T_M + T_{DC} + T_S + T_A + T_R$
  - $t_B = 2T_M + 2T_{DC} + T_S + T_B + T_R$
  - $t_C = 2T_M + 3T_{DC} + T_S + T_C + T_R$

三个设备中断响应所需时间为：

$T = t_A + t_B + t_C$

中断极限频率为：$f = 1 / T$





DMA
（直接存储器存取）

DMA（直接存储器存取）的特点

- 以响应随机请求的方式，实现主存与I/O设备间的快速数据传送
- 不影响CPU的程序执行状态
  - 只要不存在访存冲突，CPU就可以继续执行自己的程序
- 只能处理简单的数据传送
- 与查询方式相比：
  - CPU不必等待查询，可以执行自身的程序
  - 直接由硬件（DMA控制器）控制，CPU不必执行指令
- 与中断方式相比：
  - 仅需占用系统总线，不切换程序
  - CPU可与DMA传送并行工作
  - 可以实现简单的数据传送，难以识别和处理复杂事态

DMA的应用

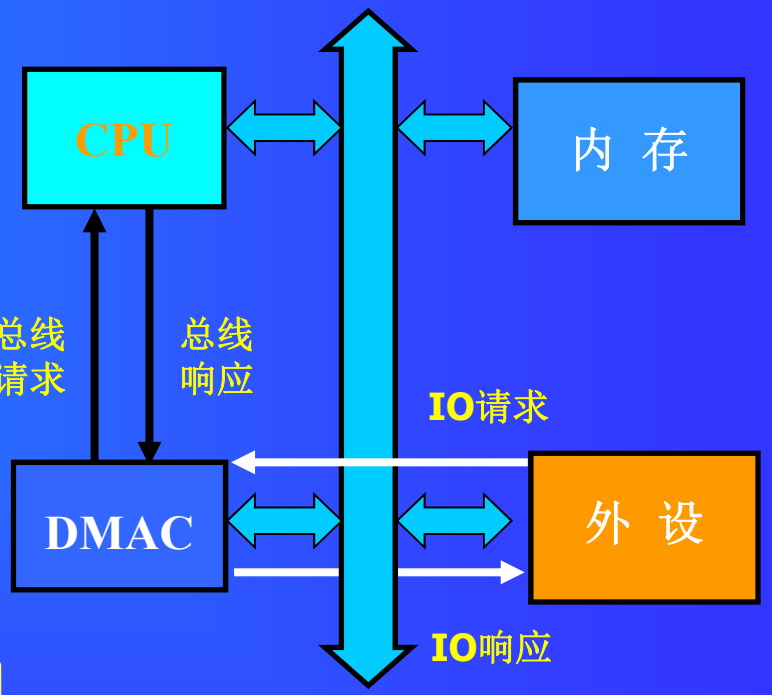
- 应用：主存与高速I/O设备间的简单、批量数据传送
- 特点：传送开始的时间是随机的，但开始传送后需要进行连续批量数据交换
- 实例
  - 磁盘读/写——以数据块为单位
  - 与外部通信——以数据帧为单位
  - 大批量数据采集

DMA方式的数据传送过程

- DMA传送前预处理阶段：
  - CPU在每次DMA传送前，通过指令对DMA控制器进行初始化，告知DMAC本次传送的参数：
    - 向DMAC的操作方式寄存器送入控制字，设定数据传送方向
    - 设置需要进行数据交换的外设的设备号
    - 向DMAC的字计数器送入要传送的字数
    - 向DMAC的内存地址计数器送入内存数据区首地址
  - CPU是主设备，DMA控制器是从设备

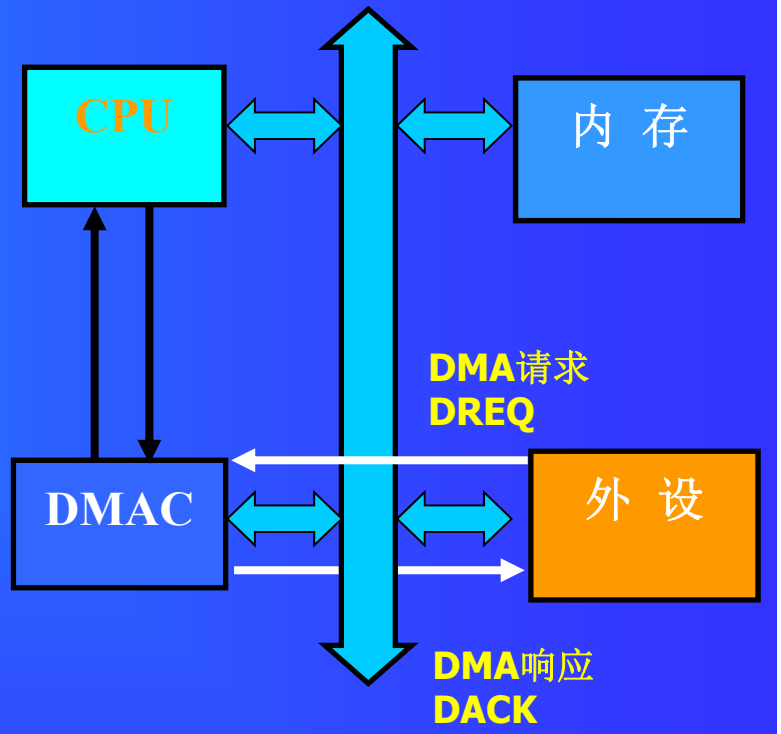
DMA方式的数据传送过程

- DMA传送前预处理阶段：
  - 外设准备好传送后，向DMA控制器发送DMA请求
  - DMA控制器向CPU发总线请求
    - 当有多个DMA请求同时产生时，可用硬件排队器择优响应
  - CPU响应总线请求，发回总线响应信号给DMA控制器，即可开始数据传送



DMA方式的数据传送过程

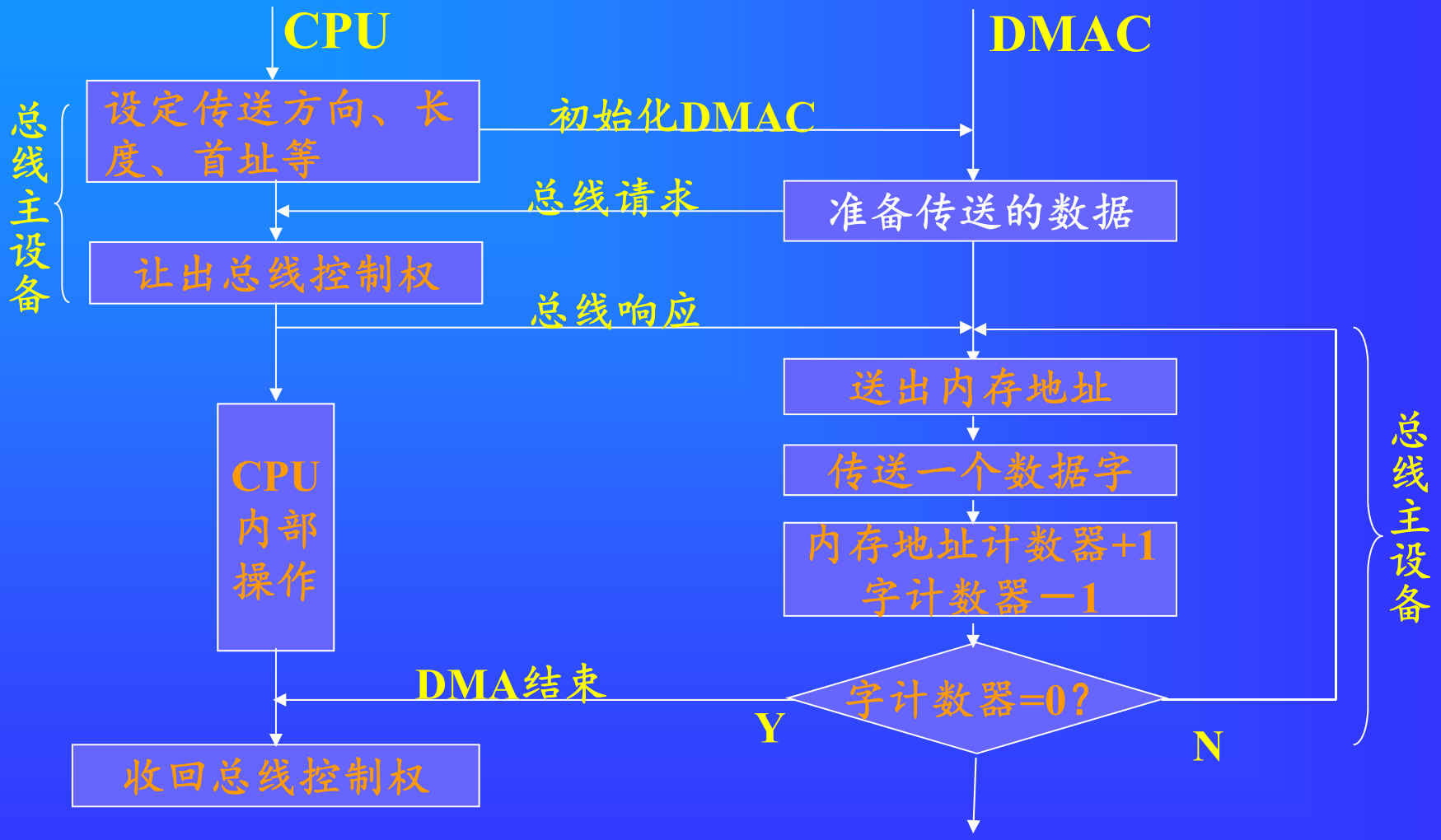
- 数据传送阶段（以成组传送方式为例）
  - DMA控制器作为主设备控制数据传送过程
    - DMAC发送内存地址和内存读/写信号
    - DMAC向外设发送DMA响应信号和I/O读写信号
    - DMAC将字计数器减1
      - 若字计数器不等于零，则修改内存地址，并进行下一字的传送
      - 若字计数器等于零，则归还总线控制权，并报告DMA操作结束



DMA方式的数据传送过程

- 传送结束后的处理阶段
  - CPU是主设备，DMA控制器是从设备
  - 若DMA控制器以中断方式向CPU报告DMA传送结束，则当CPU响应中断时，转而执行“DMA传送结束中断服务程序”
  - DMA结束处理工作：
    - 校验送入内存的数据是否正确
    - 检查DMA过程是否有错误
    - 如果无新的数据传送，则结束DMA操作
    - 如果有新的数据传送，则启动DMAC进行下一次DMA传输

DMA方式的数据传送流程

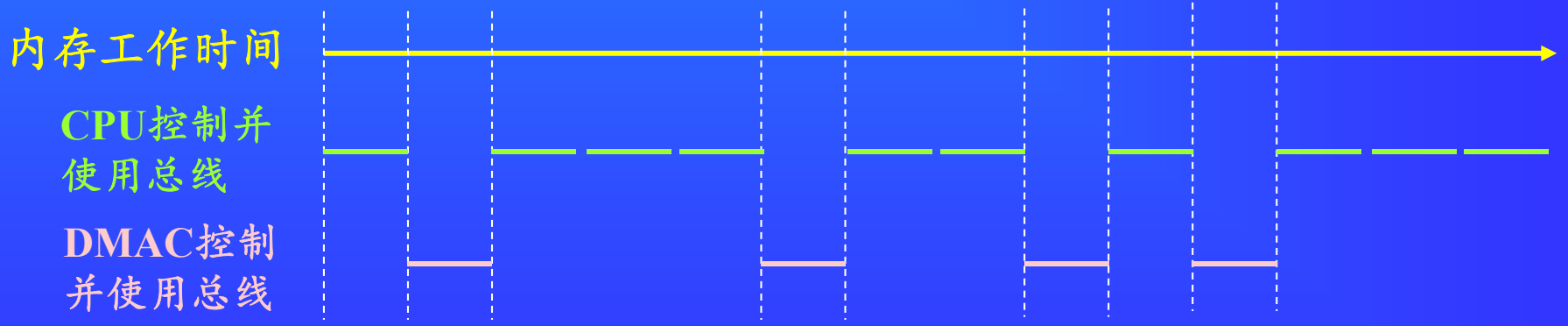


DMA传送方式

- 根据每提出一次DMA请求，DMAC将占用多少个总线周期，可以将DMA传送分成以下几种方式：
  - 单字传送方式/周期挪用方式/周期窃取方式（cycle stealing）
  - 成组连续传送方式
  - 透明DMA方式（交替操作方式，总线周期分时方式）

DMA传送方式：周期挪用方式

- 每次DMA请求获得批准后
  - CPU让出一两个总线周期的总线控制权
  - 由DMA控制器控制系统总线，以DMA方式传送一个数据字
  - 然后DMA控制器将系统总线控制权交回CPU继续执行程序
- DMAC如需继续传送，需重新申请总线使用权

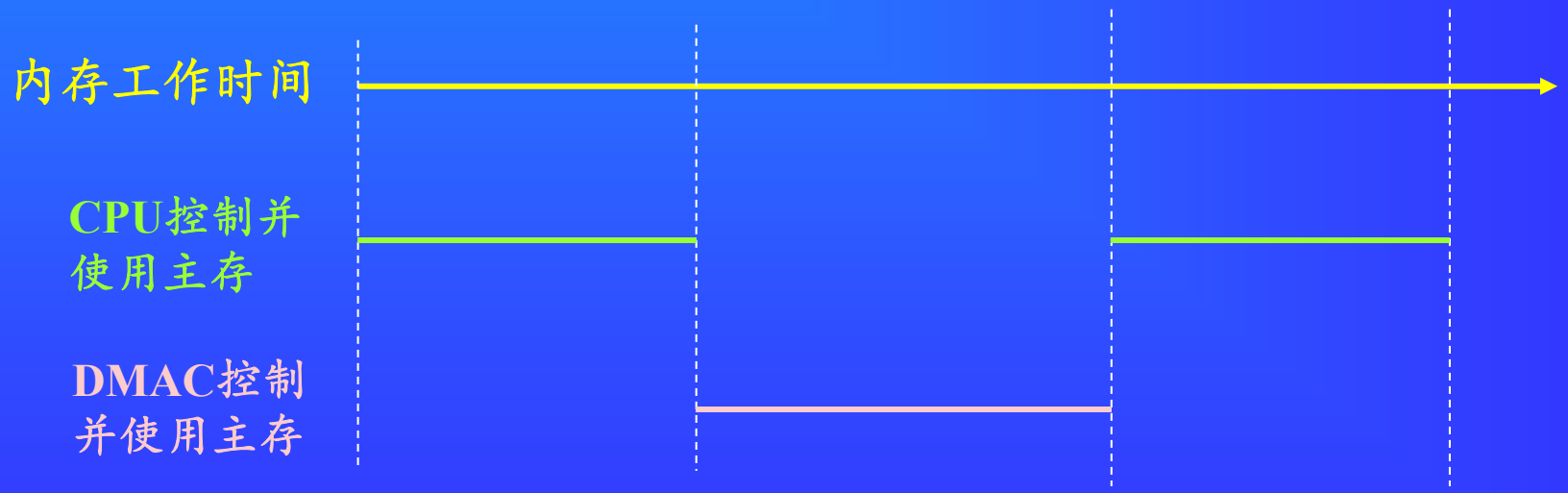


DMA传送方式：周期挪用方式

- I/O设备要求DMA传送时可能遇到两种情况：
  - CPU正好不使用总线，I/O访存与CPU无冲突（“窃取”）
  - CPU也要使用总线，产生总线冲突（“挪用”）
    - DMAC有优先权而CPU被延迟
- 适用：当主存储器工作速度高出I/O设备较多时
  - 提高主存利用率
  - 对CPU程序执行的影响较小

DMA传送方式：成组连续传送方式

- 每次DMA请求获得批准后，DMA控制器掌管总线控制权，连续占用若干个总线周期，进行成组连续批量传送，直到批量传送结束，才将总线控制权交还给CPU
- 在传送期间CPU处于保持状态，停止访问主存

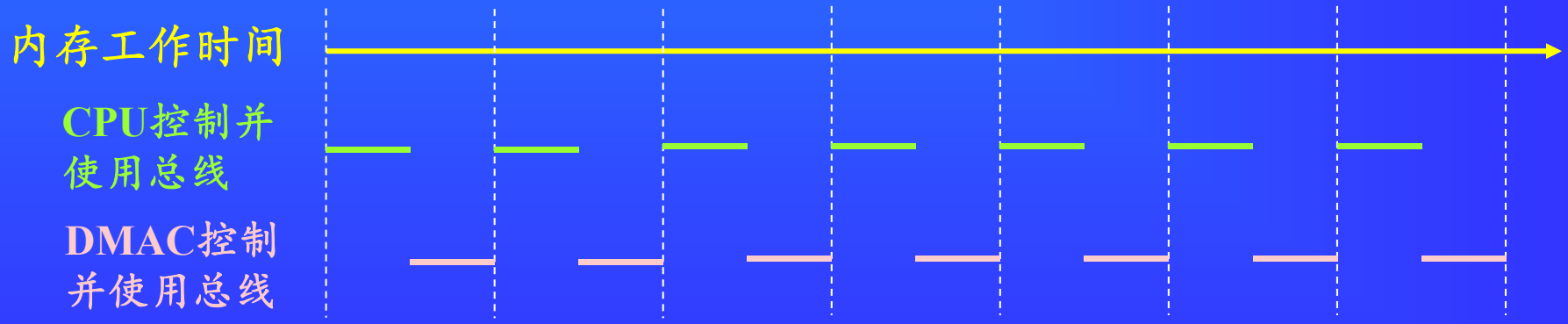


DMA传送方式：成组连续传送方式

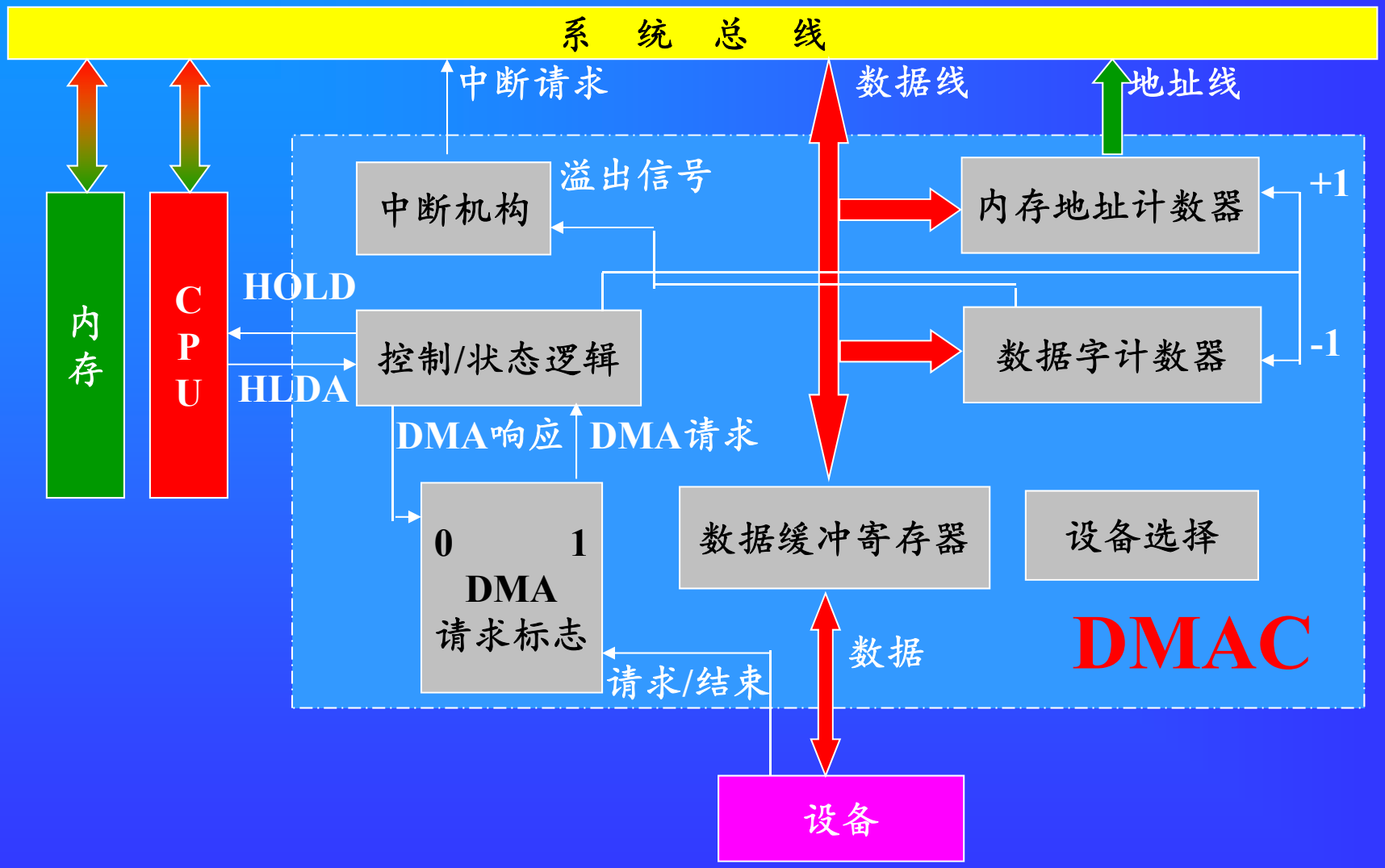
- 适用：当I/O设备的数据传输率接近于主存工作速度时，或者CPU除了等待DMA传送结束并无它事可干时
  - 可以减少系统总线控制权的交换次数，有利于提高输入/输出速度
  - 由于CPU长时间不能使用总线，有可能造成有些事件不能及时处理

DMA传送方式：透明DMA方式

- 将一个CPU周期分为两个子周期
  - C1时间内DMAC可使用总线
  - C2时间内CPU可使用总线
  - 根据C1和C2时间分配总线使用权而无需每次申请
- 不需要总线使用权的申请、建立和归还过程



DMA控制器的基本组成



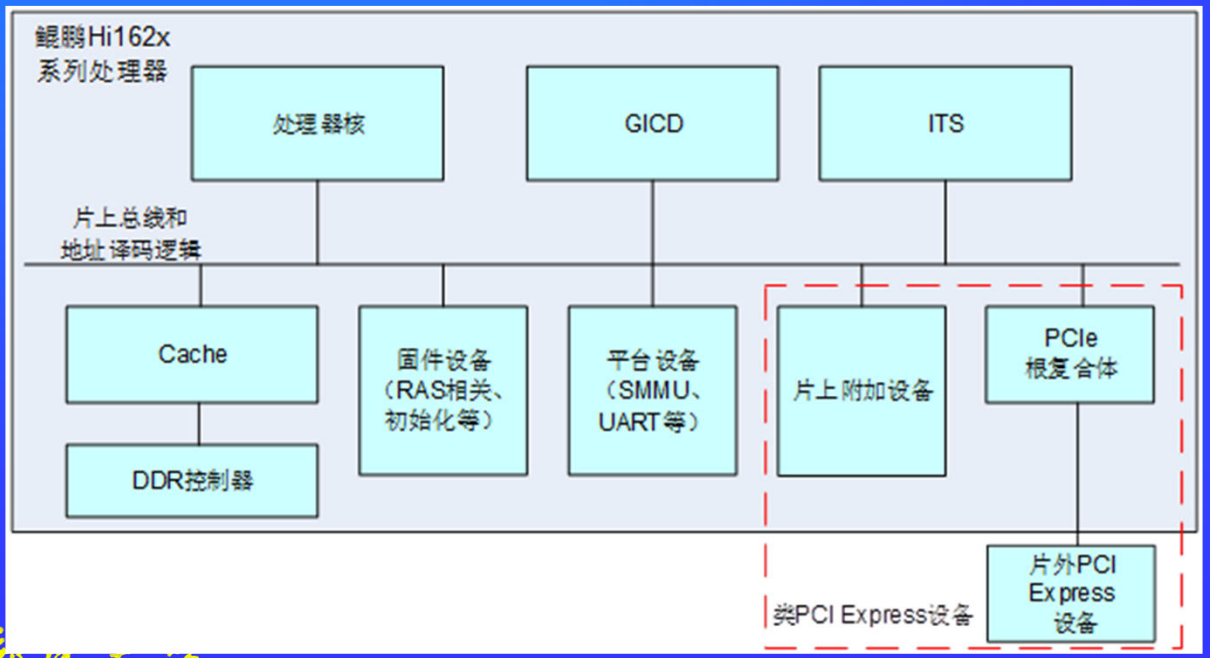
鲲鹏920处理器片上系统的设备与输入输出

鲲鹏920处理器片上系统的I/O概述

- 设备：除内核外的物理部件
  - I/O集群
  - 片上加速设备
  - 管理设备
- 基本的设备管理机制：PCIe系统拓扑
  - 片内集成设备+PCIe总线扩展片外附加设备

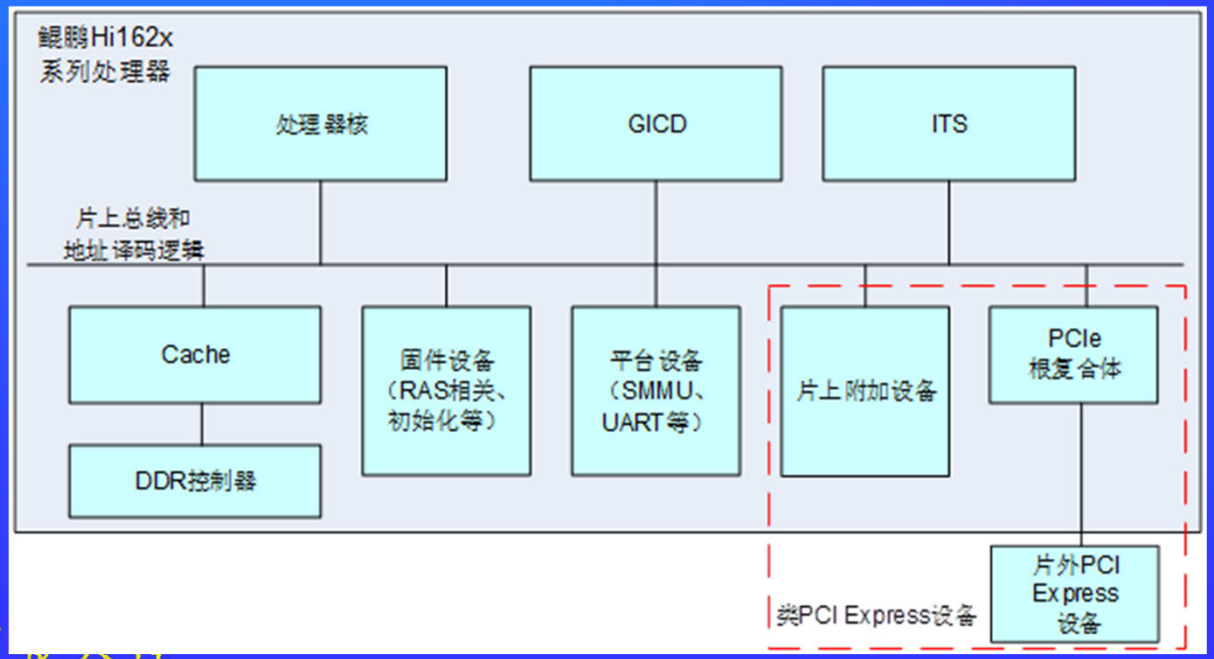
鲲鹏920处理器片上系统的片上设备类型

- 固件设备（Firmware Devices）
  - 多种功能模式和系统参数配置，引导加载初始化
    - DDR存储器初始化
    - 系统地址译码和SERDES部件初始化
  - BIOS和UEFI的运行时服务



鲲鹏920处理器片上系统的片上设备类型

- 平台设备（Platform Devices）
  - 为系统服务的标准设备
    - 管理系统所必须的系统设备控制器
      - SMMU、GIC、看门狗定时器……
    - 大多数系统都需要用到的对外通信接口
      - UART……

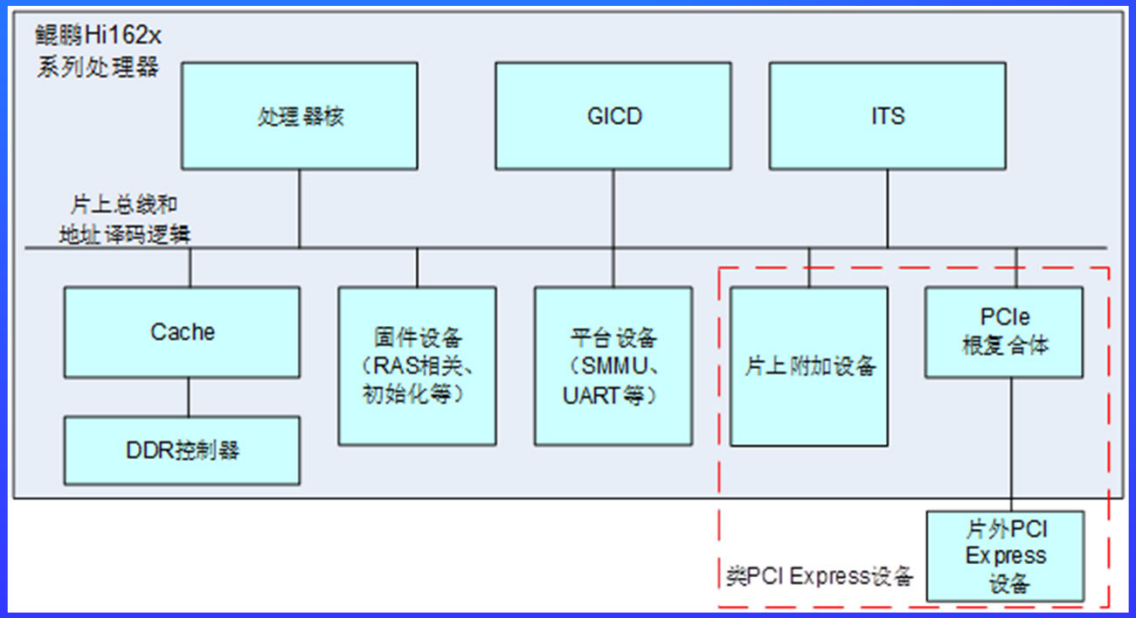


鲲鹏920处理器片上系统的平台设备

平台设备	功能	数量
SMMU	实现地址变换和存储访问保护功能。在PCI Express I/O集群中配置的系统存储管理单元还支持ATS/PRI虚拟化功能	每个I/O集群配置一个私有的系统存储管理单元部件
中断翻译服务部件ITS	将MSI/MSI-X设备中断翻译为系统LPI中断并路由至不同的通用中断控制器转发器GICR	每个芯片配置了三个ITS部件
通用中断控制器分发器GICD	实现ARM通用中断控制器中断分发器（Interrupt Distributer）功能	从软件的角度看，全系统只有一个GICD，物理上由分布在每个晶片上的GICD部件组合而成
通用看门狗（Watchdog）	实现看门狗定时器功能	对称多处理器系统只会看到一个看门狗
UART	实现通用异步收发器接口功能	每个芯片配置一个物理UART接口 软件可见的UART接口数量由固件定义
IMU消息通道	在应用（AP）和智能管理单元（IMU）之间交换信息	一组智能管理单元消息通道

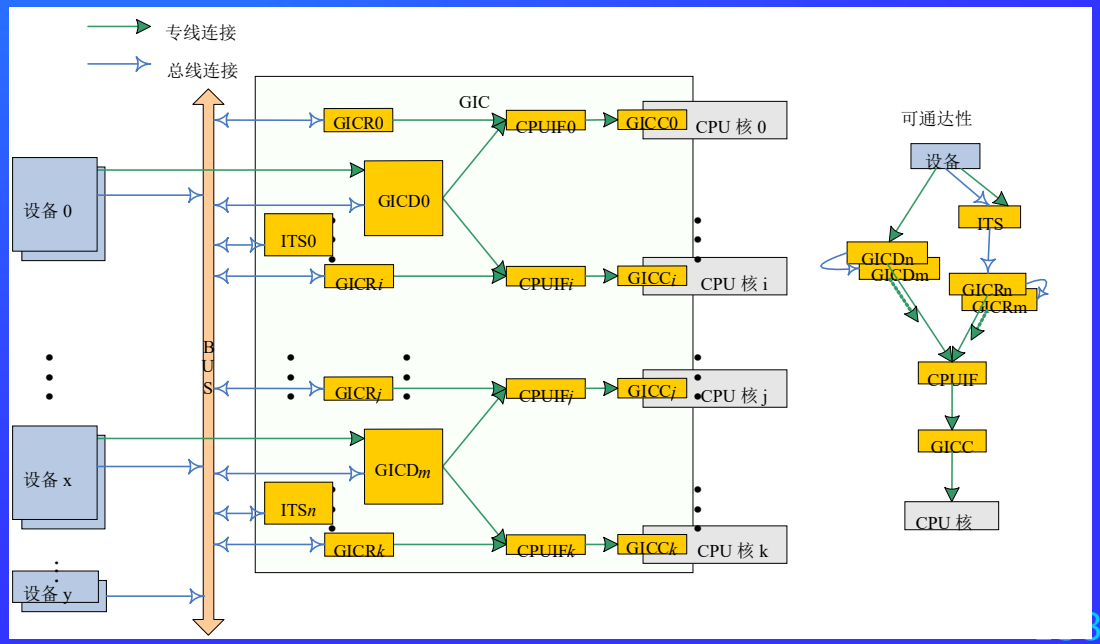
鲲鹏920处理器片上系统的片上设备类型

- 片上附加设备（Add-on Devices）
  - SOC内可以被类似功能的设备替换的部件
    - 通常使用设备提供商设计的驱动程序
    - 实现系统与外部互连通信
      - 网络接口卡控制器、USB控制器……



鲲鹏920的通用中断控制器（GIC）

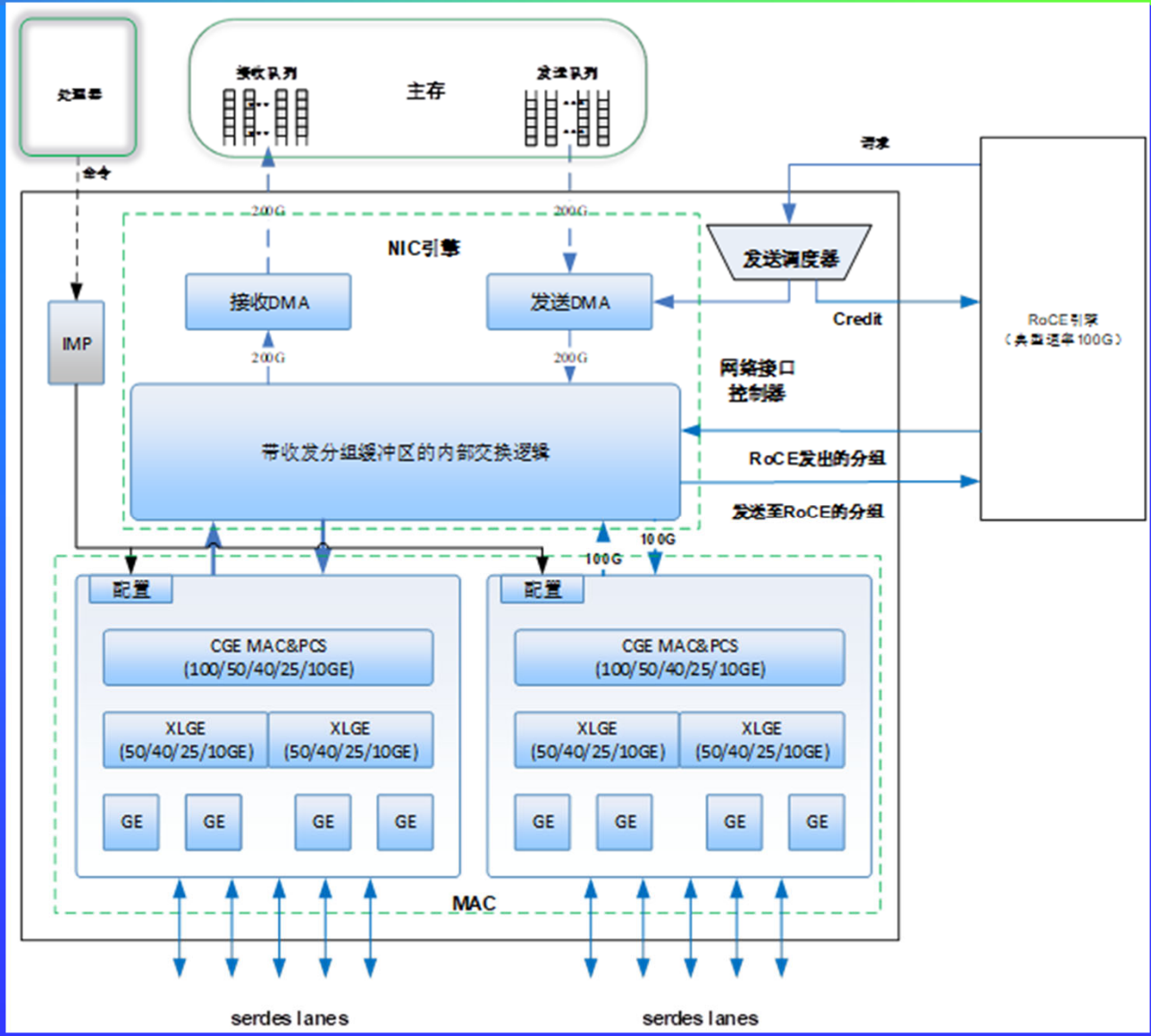
- 鲲鹏920 Hyper Interrupt Controller
  - 基于ARM GICv3的全局中断管理部件
- 组成：由系统中分布的多个部件组成的逻辑部件
  - GIC分发器GICD（GIC Distributer）
  - 中断翻译服务部件ITS
  - GIC转发器GICR（GIC Redistributer）
  - CPUIF（CPU Interface）
  - GIC CPU Interface



北京郵電大學

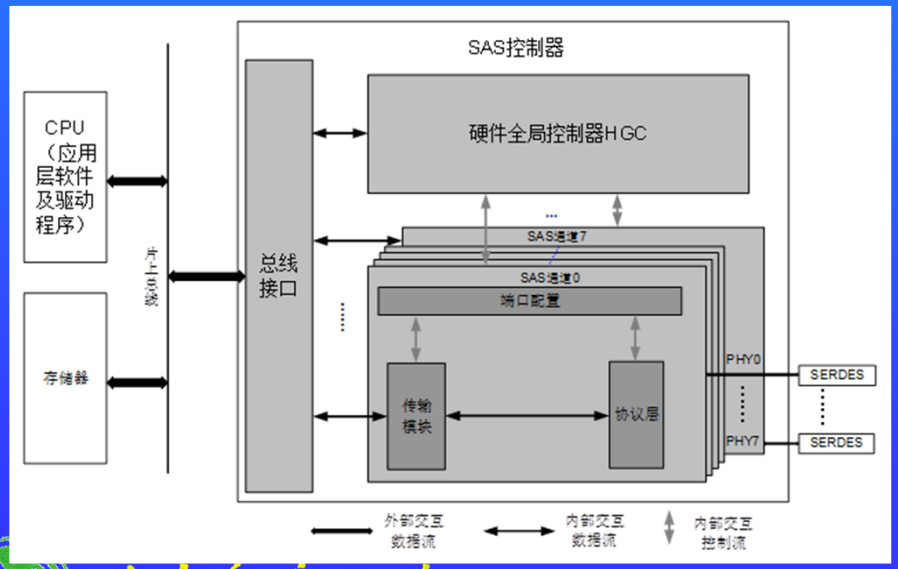
鲲鹏920的网络子系统

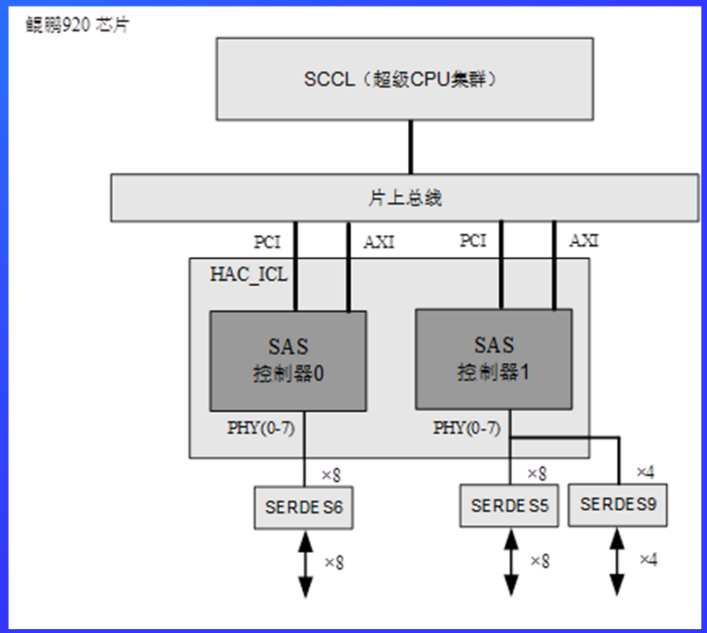
- 鲲鹏920网络子系统组成
  - Network Interface Controller（NIC）
  - RoCE引擎 RoCEE



鲲鹏920的外存储子系统

- 鲲鹏920外存储子系统
  - 外部存储设备
  - 内置的SAS控制器和SATA控制器
    - Serial Attached SCSI（SAS）控制器
    - 对外接口侧：通过片内SERDES驱动的模拟差分信号线对与远端SAS/SATA设备交换数据





本章内容

- I/O接口与外设间的数据传送方式
- CPU与I/O接口之间的数据传送方式
- 程序中断方式
- DMA传送方式
- 通道方式与I/O处理机方式

本章重点

- I/O系统的地位、作用及其组成
- 外设与主机交换信息的几种主要方式
- 程序中断方式
- DMA方式

作业

1. 单级中断系统中，中断服务程序执行顺序是（　　）。

I. 保护现场　II. 开中断　III. 关中断　IV. 保存断点

V. 中断事件处理　VI. 恢复现场　VII. 中断返回

A. I、V、VI、II、VII

B. III、I、V、VII

C. III、IV、V、VI、VII

D. IV、I、V、VI、VII

2. 下列选项中，能引起外部中断的事件是

A. 键盘输入　　B. 除数为 0

C. 浮点运算下溢　　D. 访存缺页

作业

3. CPU主频为500MHz ，CPI为5。假定某外设的数据传输率为0.5MB/s，采用中断方式与主机进行数据传送，以32位为传输单位，对应的中断服务程序包含18条指令，中断服务的其他开销相当于 2 条指令的执行时间。

（1）在中断方式下，CPU 用于该外设I/O的时间占整个CPU时间的百分比是多少?

（2）当该外设的数据传输率达到 5 MB/s时，改用DMA方式传送数据。假定每次DMA 传送块大小为5000B，且DMA预处理和后处理的总开销为500个时钟周期，则CPU用于该外设I/O的时间占整个CPU时间的百分比是多少？（假设DMA与CPU之间没有访存冲突）

计算机组成原理

Principle of Computer Organization

第八章 输入输出系统

本章结束