计 算 机 组 成 原 理

输入输出系统

2023年5月15日

本讲安排

1、概述

主机与外围设备的连接模式及组织管理　总线类型与标准

2、总线操作与时序

总线的基本概念 分类，信息传输方式，通信同步方式

3、总线控制

链式查询方式，计数器定时查询方式，独立请求方式

4、输入输出接口

功能和分类　串行总线接口，并行总线接口

5、程序查询方式

6、中断方式

7、DMA传送方式

总线系统

总线的概念和结构形态

总线接口

总线仲裁、定时、数据传送模式

总线实例

1. 总线的基本概念

(1) 什么是总线

现代计算机系统多采用模块结构，一个模块就是一个功能部件，如主机板、显示适配器、解压卡、声卡、A/D板等。各模块之间进行信息传送的公共通路称为总线。

借助于总线连接，计算机在各功能部件间实现地址、数据和控制信息的交换，并在争用资源的基础上进行工作。

(2) 单处理器系统中总线分类

一个单处理器系统中的总线，大致分为三类：

内部总线：CPU内部连接各寄存器及运算部件之间的总线。

系统总线：CPU同计算机系统的其他高速功能部件，如存储器、通道等互相连接的总线。

I/O总线： 中、低速I/O设备之间互相连接的总线。

总线有多种分类方法

按相对于CPU与其他芯片的位置可分为

片内总线和片外总线。

按总线传送信息的类别，可把总线分为

地址总线、数据总线和控制总线。

按照总线传送信息的方向，可把总线分为

单向总线和双向总线。

按总线的层次结构可分为

CPU总线、存储总线、系统总线和外部总线。

(3) 总线的特性

物理特性：

指总线的物理连接方式，包括总线的根数，总线的插头、插座的形状，引脚线的排列方式等。

功能特性：

描述总线中每一根线的功能。如：

- 地址总线的宽度指明了总线能够直接访问的存储器的地址空间范围；
- 数据总线的宽度指明了访问一次存储器或外设时能够交换的数据的位数；
- 控制总线包括了CPU发出的各种命令，请求信号和仲裁信号，中断信号等。

电气特性：

定义每一根线上信号的传递方向及有效电平范围。

- 送入CPU的信号叫输入信号(IN)，
- 从CPU发出的信号叫输出信号(OUT)。
- 地址总线是输出线，高电平有效；
- 数据总线是双向传输线，高电平有效；
- 控制总线中各条线一般是单向的，有CPU发出的，也有进入CPU的，有高电平有效的，也有低电平有效的；
- 总线的电平都符合TTL电平的定义。

时间特性： 定义了每根线在什么时间有效。

规定了总线上各信号有效的时序关系，CPU才能正确无误地使用。

(4) 总线的标准化

相同的指令系统，相同的功能，不同厂家生产的各功能部件在实现方法上几乎没有相同的，但各厂家生产的相同功能部件却可以互换使用，其原因在于它们都遵守了相同的系统总线的要求，这就是系统总线的标准化问题。

通常有两类标准：

正式公布的标准和实际存在的工业标准。

正式公布的标准由IEEE（国际电气电子工程师学会）或CCITT（国际电报电话咨询委员会）等国际组织正式确定和承认。

实际的工业标准首先由某一公司提出，而又得到其它公司广泛使用，有可能经过一段时间后提交有关组织讨论从而成为正式标准。

在标准中，对插件引线的几何尺寸、引线数、各引线的定义、时序及电气参数等都做出明确规定，据此，人们可方便的进行系统设计和功能扩充。

(5) 微机结构与系统总线的发展

PC/XT结构与PC总线　针对IBM PC/XT机(8086)

PC总线时钟频率4.77MHz，总线宽度8位，寻址能力1MB，最快存储器访问周期由4个时钟周期组成——带宽约1MBps

PC/AT结构与AT总线　针对IBM PC/AT机(286)

AT总线时钟频率8MHz，总线宽度16位，寻址能力16MB，最快存储器访问周期由3个时钟周期组成——带宽约5MBps

PC总线与AT总线后来经过标准化，称为 *ISA总线*

Industry Standard Architecture——工业标准体系结构

EISA总线 针对 386、486

Extended ISA——扩展工业标准体系结构

总线时钟频率8.33MHz，总线宽度32位，寻址能力4GB，支持突发传送——带宽约33MBps

VESA总线 针对 486

Video Electronic Standard Association——视频电子标准协会。VESA总线也称为VL-bus(VESA Local Bus)。

总线时钟频率33MHz，总线宽度32位，寻址能力4GB，支持突发传送——带宽约132MBps

PCI总线 针对 Pentium以上处理器

PCI: Peripheral Component Interconnect——外部设备互连

(6) 总线的性能指标

总线宽度

数据总线的根数。

16位总线，指其数据总线为16根。

寻址能力

取决于地址总线的根数。

PCI总线的地址总线为32位，寻址能力达4GB。

传输率

也称为总线带宽，通常指总线所能达到的最高数据传输率，单位是Bps（每秒传送字节数）

计算公式：$D_r=D\times f/N$

D——数据宽度；

f——总线时钟频率；

N——完成一次数据传送所需的时钟周期数。

PCI总线1.0版的总线带宽132MBps

是否支持突发传送

总线上数据传送方式：

正常传送——每个传送周期先传送数据的地址，再传送数据。

突发传送——支持成块连续数据的传送，只需给出数据块的首地址，后续数据地址自动生成。

PCI总线支持突发传送，ISA不支持。

负载能力

总线上能够连接的设备数。

例: (1)某总线在一个总线周期中并行传送4个字节的数据，假设一个总线周期等于一个总线时钟周期，总线时钟频率为33MHz，则总线带宽是多少?

(2)如果一个总线周期中并行传送64位数据，总线时钟频率升为66MHz，则总线带宽是多少?

解： (1) $D_r=D\times f/N=4B\times33\times10^6/s=132MB/s$

(2) $D_r=D\times f/N=8B\times66\times10^6/s=528MB/s$

2. 总线系统的连接方式

在计算机系统中，总线的排列布置与其他各功能部件的连接方式对计算机系统的性能有重要影响。

总线的组织方法很多，单机系统中采用的总线结构基本上可分成三类：

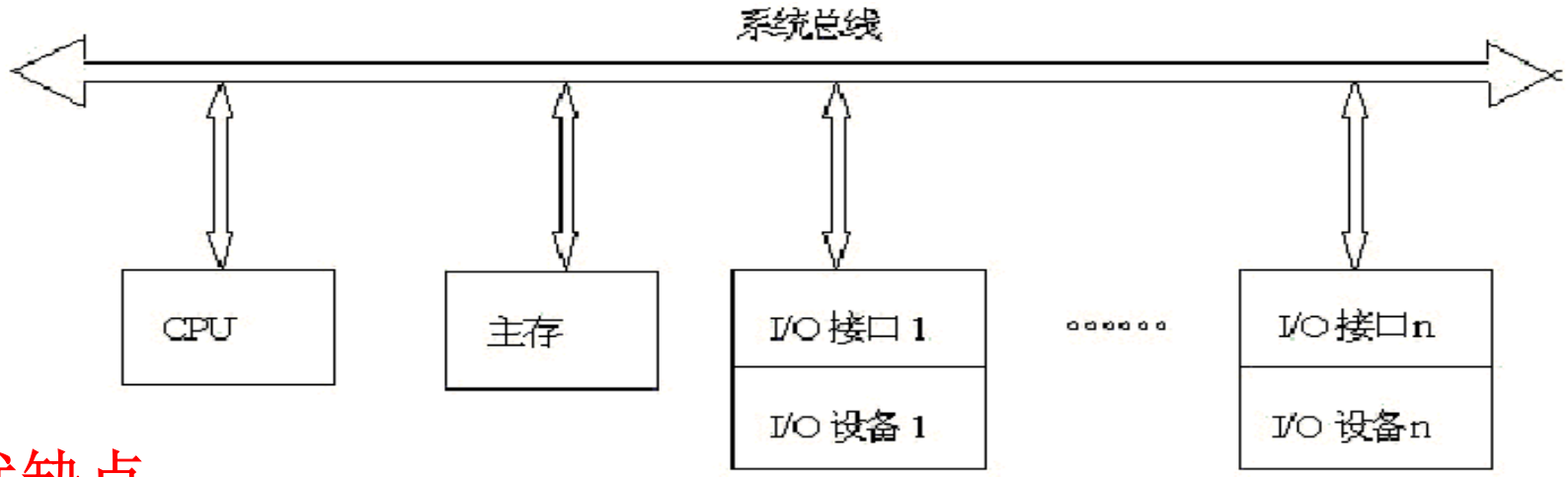
单总线结构

双总线结构

三总线结构

(1) 单总线结构

在许多单处理器的计算机中，使用一条单一的系统总线来连接CPU、主存和I/O设备，叫做单总线结构。



优缺点：

单总线具有结构简单便于扩充等优点，但由于所有数据的传送都通过这一共享的总线，因此总线可能成为系统的瓶颈。所以单总线结构多在对速度要求不高的微型机和小型机中。

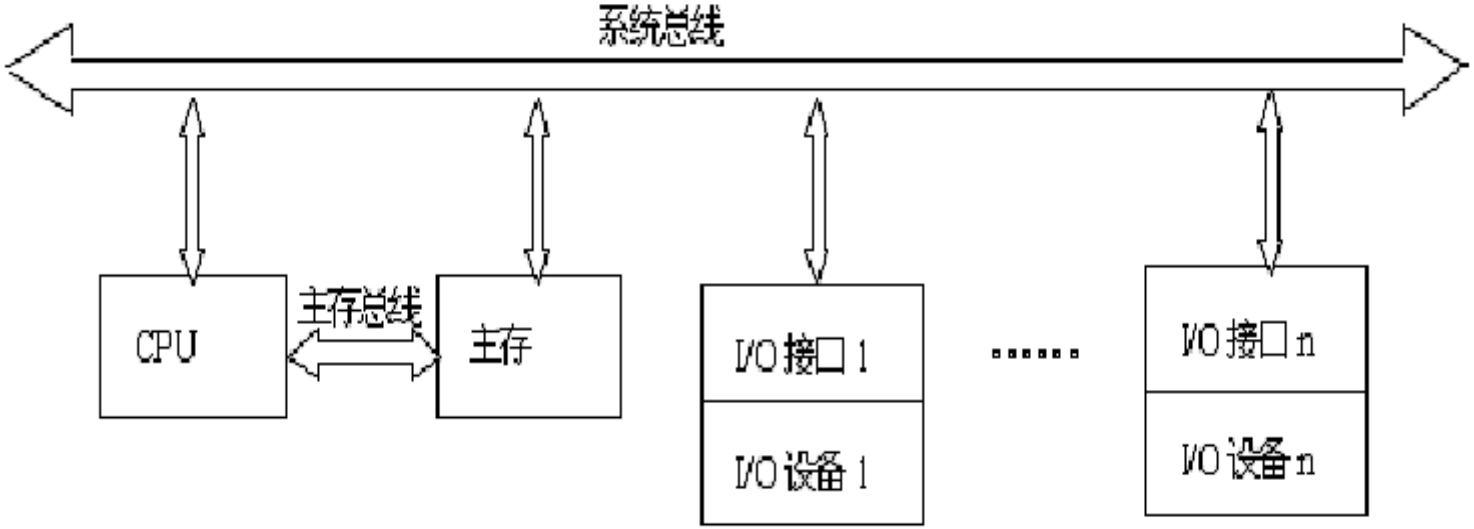
控制简单、易扩充、使用率高

分时工作，导致很大的时间延迟，总效率低

(2) 双总线结构

单总线系统中，由于所有功能部件都连在同一组总线上，总线只能分时工作，即某一时间只允许在两个部件之间传送信息。

由于CPU频繁的访问主存，在高档微型机和一些小型机中专门设置了主存总线，形成了双总线结构：



在CPU和主存之间专门设置了一组高速的存储总线

CPU与存储器可通过专用总线交换信息

- CPU可通过专用总线与存储器交换信息，减轻了系统总线的负担；
- 高速外设与主存之间仍可通过系统总线实现DMA操作；
- CPU通过系统总线与中低速外部设备交换信息。

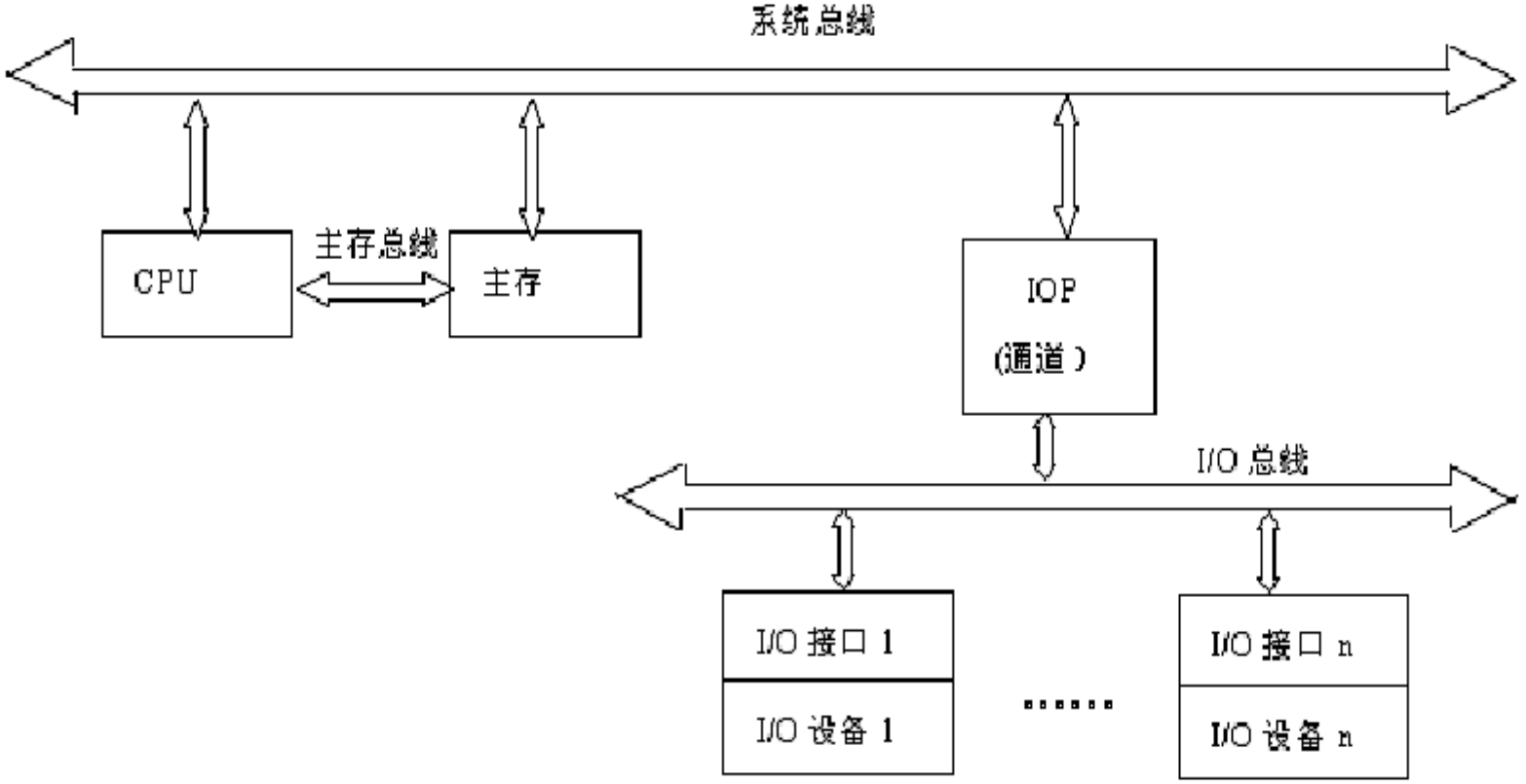
优缺点：

双总线结构保持了单总线系统简单、易于扩充的优点，又提高了信息传送的吞吐量。但这是以增加硬件为代价的。

(3) 三总线结构

在双总线系统的基础上增加I/O总线，便形成了三总线系统结构。其中：

- 系统总线是CPU、内存和通道之间进行信息传送的公共通路，
- I/O总线是多个外部设备与通道之间进行信息传送的公共通路。



通道：

通道实际上是一台具有特殊功能的处理器，又称为IOP(I/O处理器)，它分担了一部分CPU的功能，以实现对外设的统一管理及外设与主存之间的数据传送。

显然，由于增加了IOP，使整个系统的效率大大提高。然而这是以增加更多的硬件代价换来的。

3. 总线结构对计算机系统性能的影响

(1) 最大存储容量

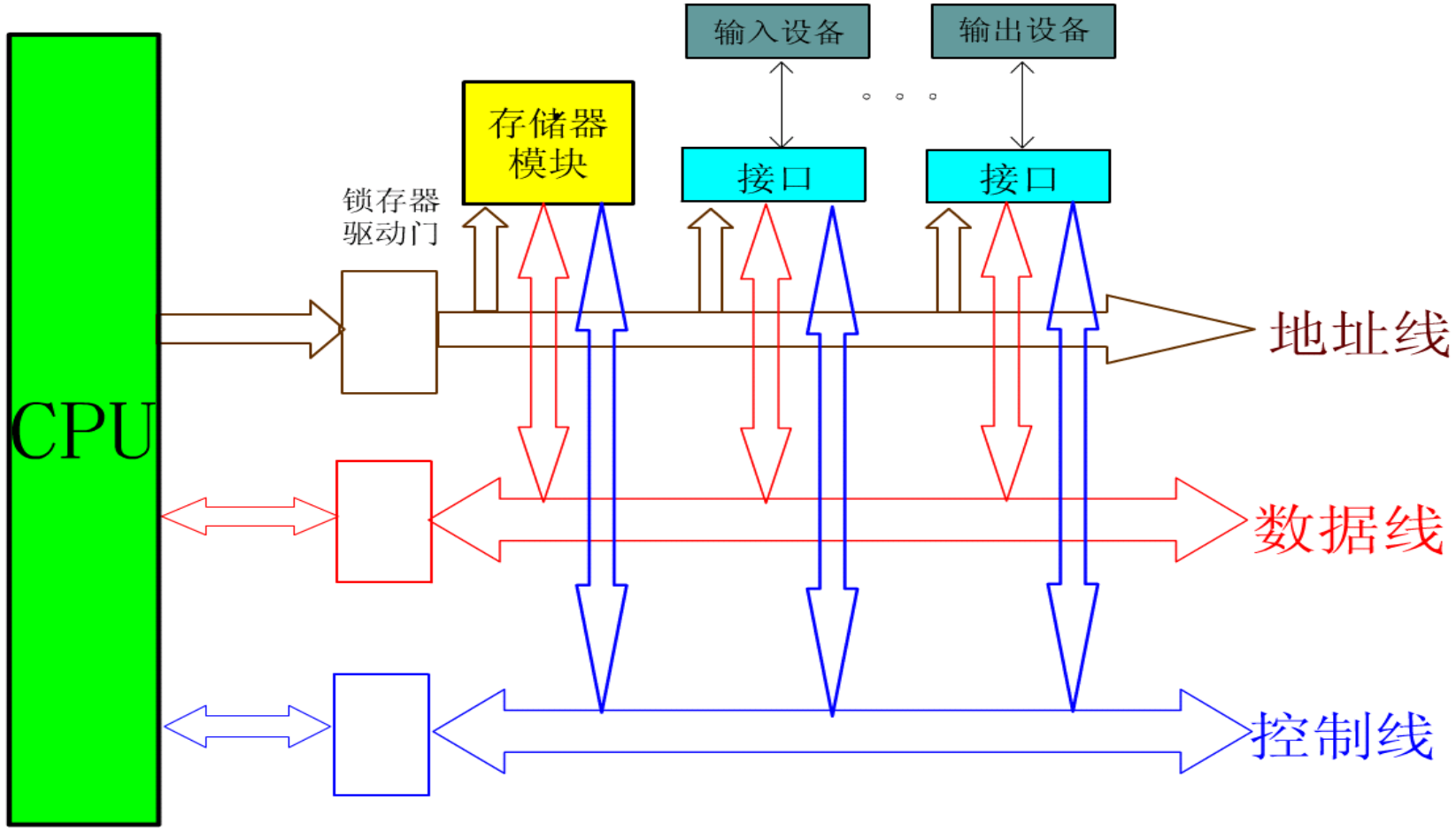
(2) 指令系统

(3) 吞吐量

4. 总线的内部结构

(1) 早期总线的内部结构

早期的总线实际上是处理器芯片引脚的延伸，是处理器与I/O设备适配器的通道。这种简单的总线一般由50~100条线组成，这些线按其功能可分为三类：地址线、数据线和控制线。



简单总线结构的不足之处在于：

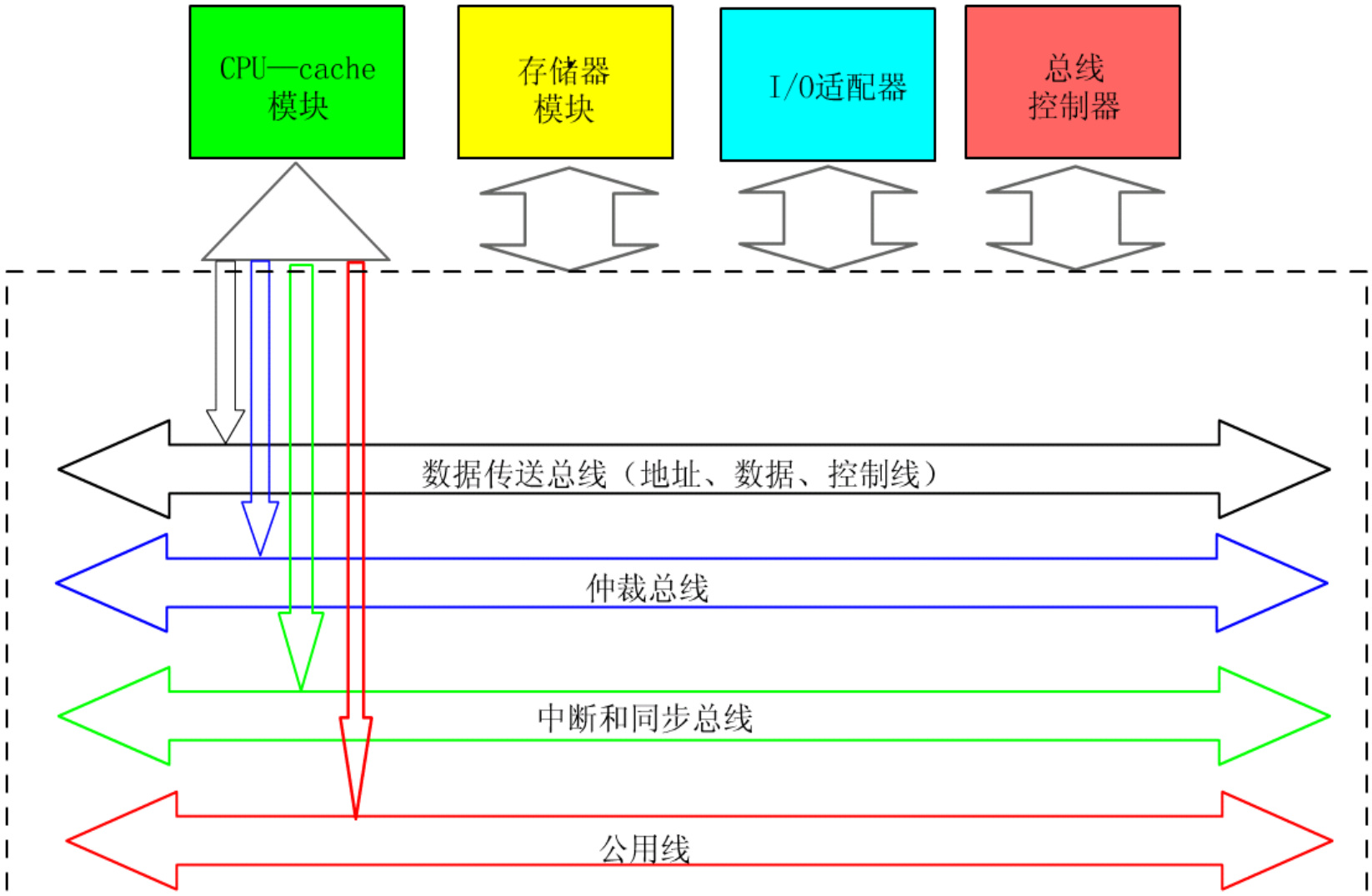
第一 **CPU是总线上的唯一主控者。**

第二 **总线信号是CPU引脚信号的延伸，故总线结构紧密与CPU相关，通用性较差。**

(2) 当代流行的总线的内部结构

它是一些标准总线，追求与结构、CPU、技术无关的开发标准，并满足包括多个CPU在内的主控者环境需求。

在当代总线结构中，CPU和它私有的cache一起作为一个模块与总线相连。系统中允许有多个这样的处理器模块。而总线控制器完成几个总线请求者之间的协调与仲裁。


CPU—cache
模块
存储器
模块
I/O适配器
总线
控制器
数据传送总线（地址、数据、控制线）
仲裁总线
中断和同步总线
公用线

整个总线分成如下四部分：

数据传送总线： 由地址线、数据线、控制线组成。

仲裁总线： 包括总线请求线和总线授权线。

中断和同步总线：用于处理带优先级的中断操作，包括中断请求线和中断认可线。

公用线： 包括时钟信号线、电源线、地线、系统复位线以及加电或断电的时序信号线等。

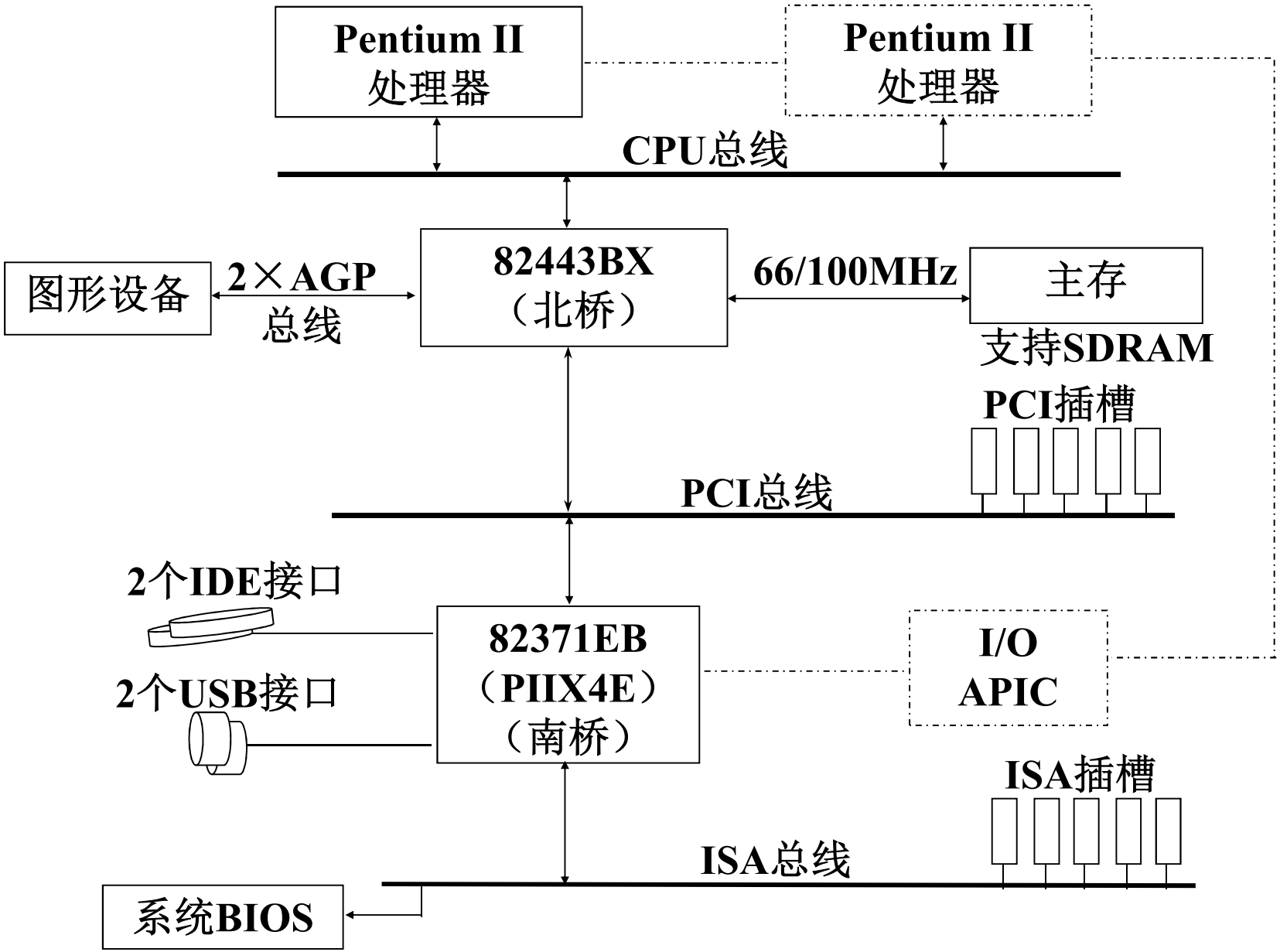
5. 总线结构实例 —分层的多总线结构

大多数计算机采用了分层次的多总线结构。

在这种结构中：

速度差异较大的设备模块使用不同速度的总线；

速度相近的设备模块使用同一类总线。


Pentium II
处理器
Pentium II
处理器
CPU总线
图形设备
2×AGP
总线
82443BX
（北桥）
66/100MHz
主存
支持SDRAM
PCI插槽
PCI总线
2个IDE接口
2个USB接口
82371EB
（PIIX4E）
（南桥）
I/O
APIC
ISA插槽
ISA总线
系统BIOS

南北桥芯片

CPU总线、PCI总线、ISA总线通过两个“桥”芯片连成整体。

桥芯片在此起到了信号速度缓冲、电平转换和控制协议的转换作用。通过桥将两类不同的总线“粘合”在一起的技术特别适合于系统的升级换代。

pentium个人机总线系统中有一个核心逻辑芯片组，简称PCI芯片组，它包括主存控制器和cache控制器芯片、北桥芯片和南桥芯片。这个芯片组叫Intel 430系列、440系列，它们在系统中起着至关重要的作用。

总线系统

总线的概念和结构形态

总线接口

总线仲裁、定时、数据传送模式

总线实例

总线接口

1. 信息的传送方式

数字计算机使用二进制数，它们或用电位的高、低来表示，或用脉冲的有、无来表示。习惯上用：

高电位表示“1”，低电位表示“0”；
或用有脉冲表示“1”，无脉冲表示"0"

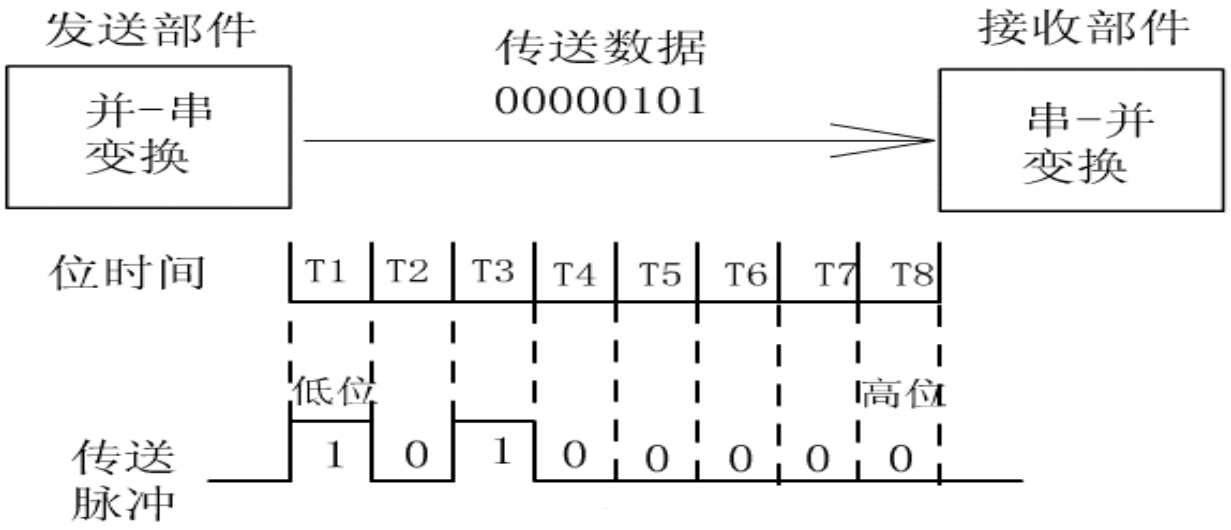
计算机系统中，传输信息采用三种方式：

串行传送、并行传送和分时传送。

但是出于速度和效率上的考虑，系统总线上传送的信息必须采用并行传送方式。

(1) 串行传送

- 当信息以串行方式传送时，只有一条传输线且采用脉冲传送。
- 在串行传送时，按顺序来传送表示一个数码的所有二进制位(bit)的脉冲信号，每次一位，通常以第一个脉冲信号表示数码的最低有效位，最后一个脉冲信号表示数码的最高有效位。
- 传送时低位在前，高位在后。



拆卸

在串行传送时，被传送的数据需要在发送部件进行并—串变换，这称为拆卸；

装配

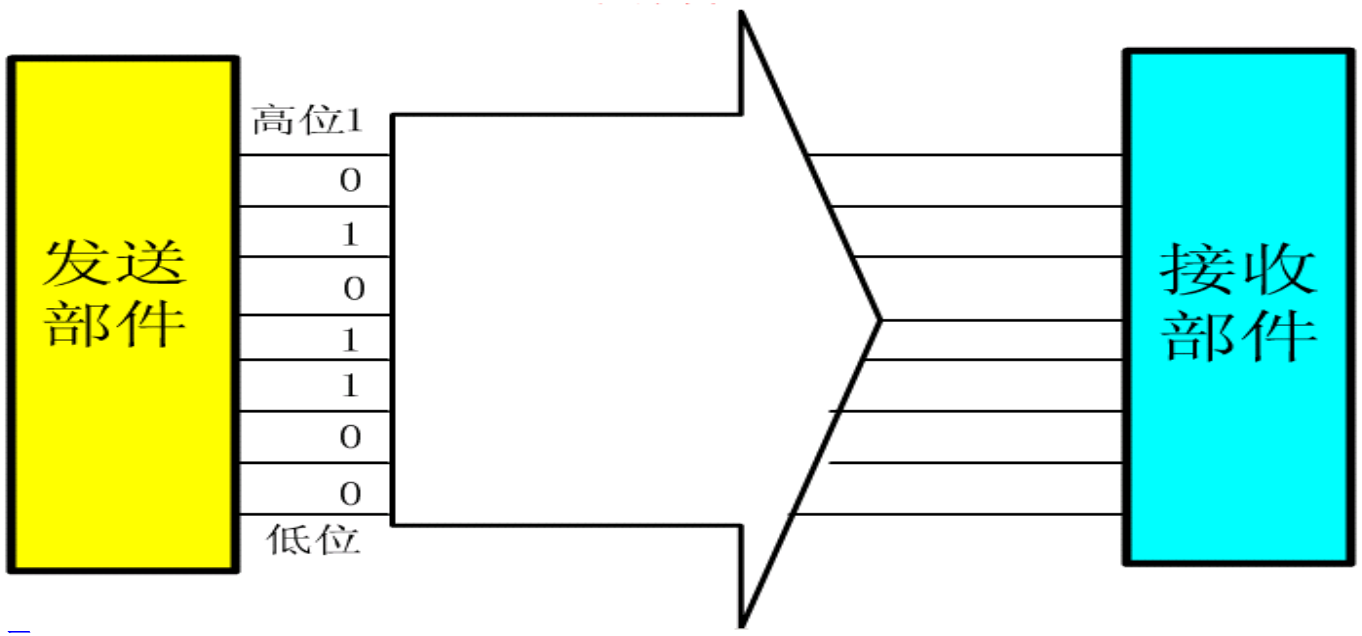
而在接收部件又需要进行串—并变换，这称为装配。

位时间

指一个二进制位在传输线上占用的时间长度。“位时间”是由同步脉冲来体现的。

(2) 并行传送

用并行方式传送二进制信息时，对每个数据位都需要单独一条传输线。信息有多少二进制位组成，就需要多少条传输线，从而使得二进制数“0”或“1”在不同的线上同时进行传送。并行传送一般采用电位传送。



(3) 分时传送

分时传送有两种概念。

① 总线复用方式

某个传输线上既传送地址信息，又传送数据信息。为此必须划分时间片，以便在不同的时间间隔中完成传送地址和传送数据的任务。

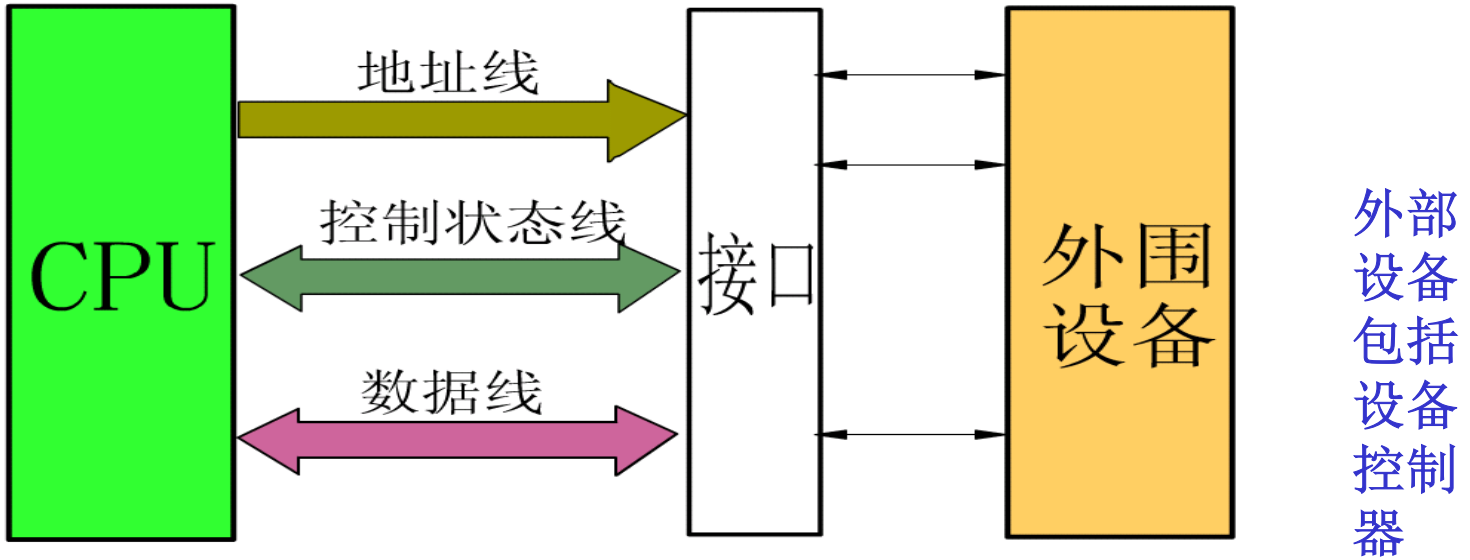
② 共享总线的部件分时使用总线

2. 接口的基本概念（重点掌握）

I/O设备适配器简称为接口。

接口定义：

接口指CPU和主存、外围设备之间通过总线进行连接的逻辑部件。



外部设备包括设备控制器

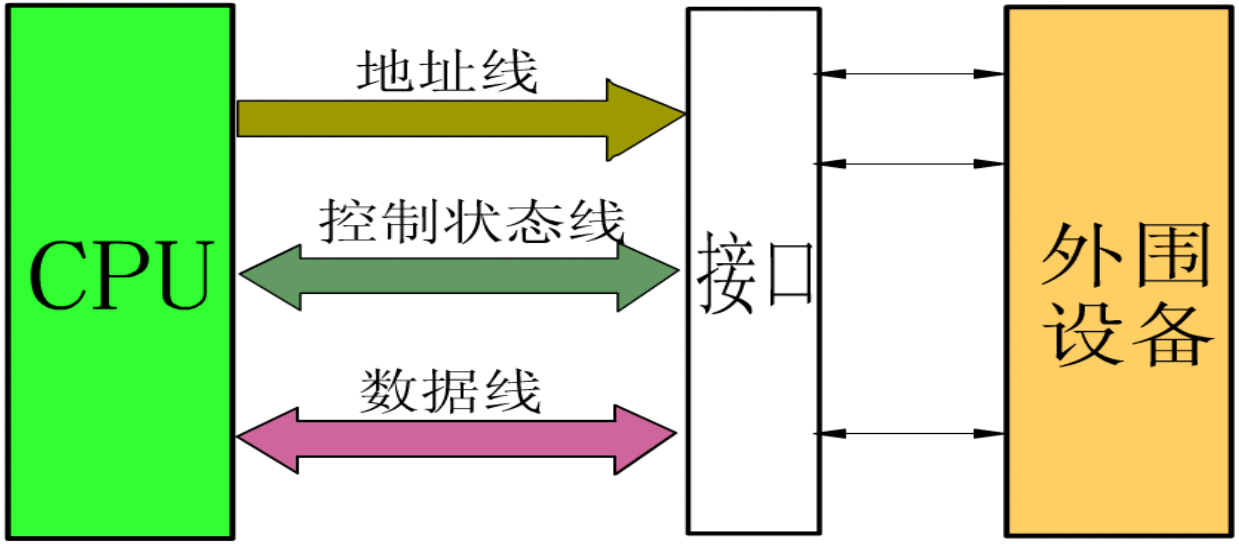
接口部件在它动态连接的两个部件之间起着“转换器”的作用，以便实现彼此之间的信息传送。

为了使所有的外围设备能够兼容，并能在一起正确地工作，CPU规定了不同的信息传送控制方法。

接口使外部设备用计算机系统特性所要求的形式发送和接收信息。

接口逻辑通常做成标准化。

CPU、接口和外围设备之间的连接关系



接口功能

(1) 控制

接口靠程序的指令信息来控制外围设备的动作，如启动、关闭设备等。

(2) 缓冲

接口在外围设备和计算机系统其他部件之间用作为一个缓冲器，以补偿各种设备在速度上的差异。

(3) 状态

接口监视外围设备的工作状态并保存状态信息。状态信息包括数据“准备就绪”、“忙”、“错误”等等，供CPU询问外围设备时进行分析之用。

(4) 转换

接口可以完成任何要求的数据转换, 例如并－串转换或串－并转换，因此数据能在外围设备和CPU之间正确地进行传送。

(5) 整理

接口可以完成一些特别的功能，例如在需要时可以修改字计数器或当前内存地址寄存器。

(6) 程序中断

每当外围设备向CPU请求某种动作时，接口即发送一个中断请求信号到CPU。

事实上，一个适配器必有两个接口：

一是和系统总线的接口，CPU和适配器的数据交换一定的是并行方式；

二是和外设的接口，适配器和外设的数据交换可能是并行方式，也可能是串行方式。根据外围设备供求串行数据或并行数据的方式不同，适配器分为串行数据接口和并行数据接口两大类。

外设的寻址

统一编址法

存储器映像的外设寻址:

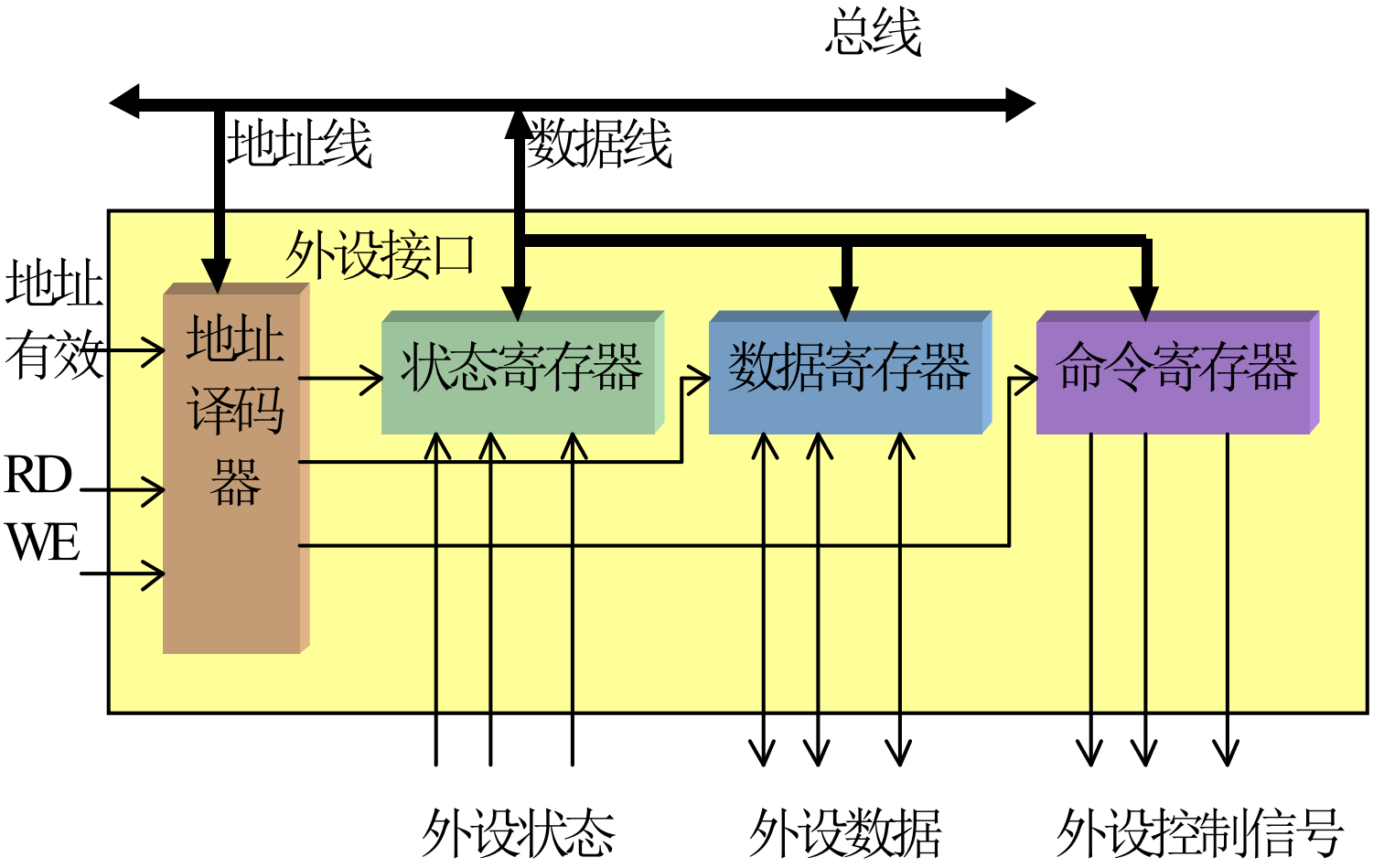
将接口中的控制寄存器、数据寄存器、状态寄存器和内存单元一样看待,可以利用访存指令进行输入输出操作

单独编址法

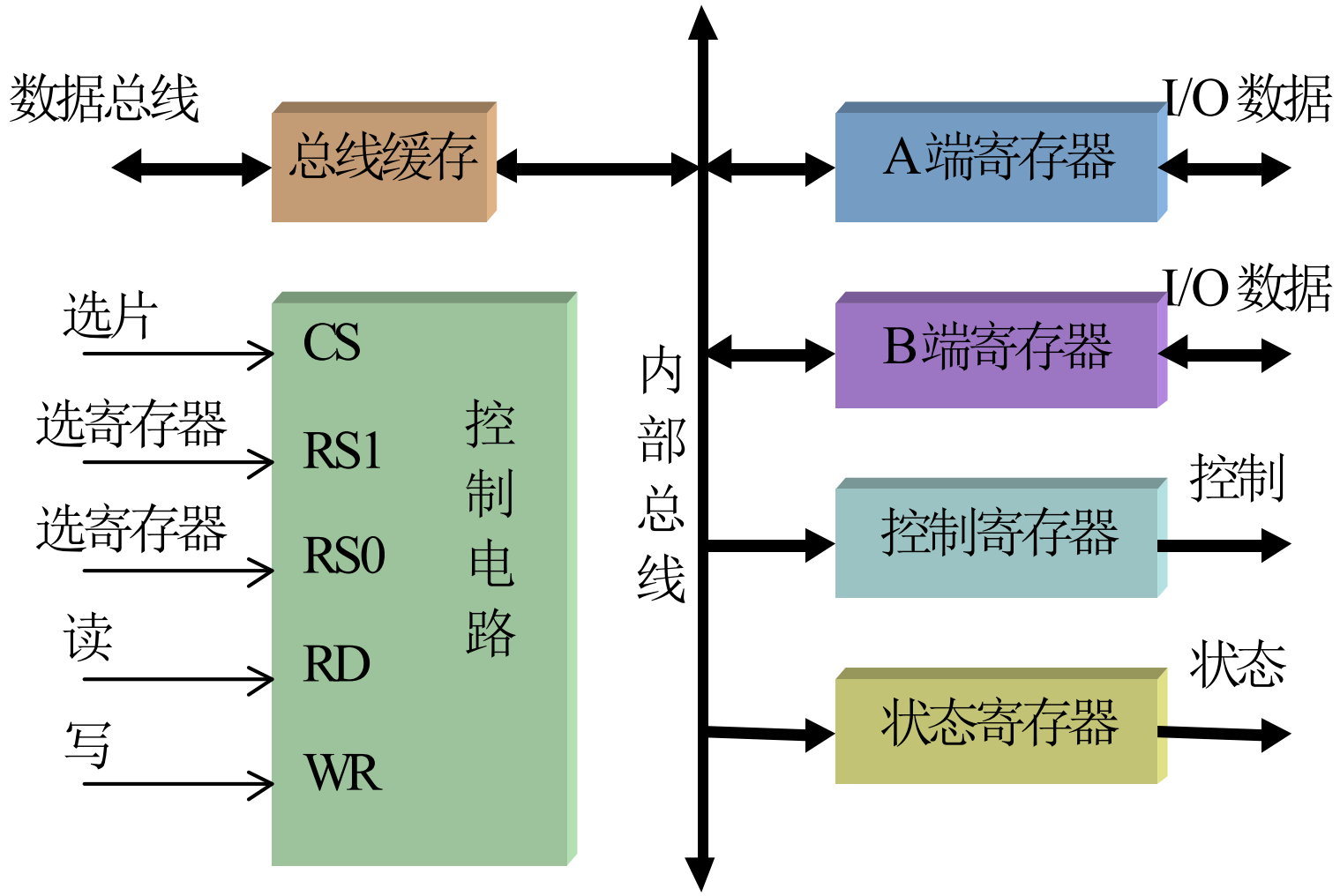
接口与存储器采用不同的两个地址空间

访问存储器和访问外围设备采用不同的指令

外围设备接口的结构


总线
地址线
数据线
外设接口
地址
有效
地址
译码
器
RD
WE
状态寄存器
数据寄存器
命令寄存器
外设状态
外设数据
外设控制信号

并行接口的例子


数据总线
总线缓存
A端寄存器
I/O 数据
选片
CS
B端寄存器
I/O 数据
选寄存器
RS1
控制电路
内部总线
选寄存器
RS0
控制寄存器
控制
读
RD
状态
写
WR
状态寄存器

例：利用串行方式传送字符，每秒钟传送的数据位数常称为波特率。假设数据传送速率是120个字符/秒，每一个字符格式规定包含10个数据位(起始位、停止位、8个数据位)，问传送的波特数是多少？每个数据位占用的时间是多少？

解：

波特数为：10位×120/秒=1200波特

每个数据位占用的时间T_d是波特数的倒数：

$$T_d=1/1200=0.833\times0.001s=0.833ms$$

总线系统

总线的概念和结构形态

总线接口

总线仲裁、定时、数据传送模式

总线实例

总线仲裁、定时及数据传送模式（了解实现硬件的原理）

1. 总线的仲裁

(1) 为什么要进行总线仲裁

连接到总线上的功能模块有主动和被动两种形态。

主方：可以启动一个总线周期；

从方：只能响应主方的请求。

- 每次总线操作，只能有一个主方占用总线控制权，但可以有多个从方。
- 除CPU外，I/O模块也可提出总线请求。

为了解决多个主设备同时竞争总线控制权，必须具有总线仲裁部件，以某种方式选择其中一个主设备作为总线的下一次主方。

(2) 总线仲裁方式

按照总线仲裁电路的位置不同，仲裁方式分为集中式仲裁和分布式仲裁两类。

① 集中式仲裁

若总线仲裁逻辑集中于一个单元，称为集中式仲裁。

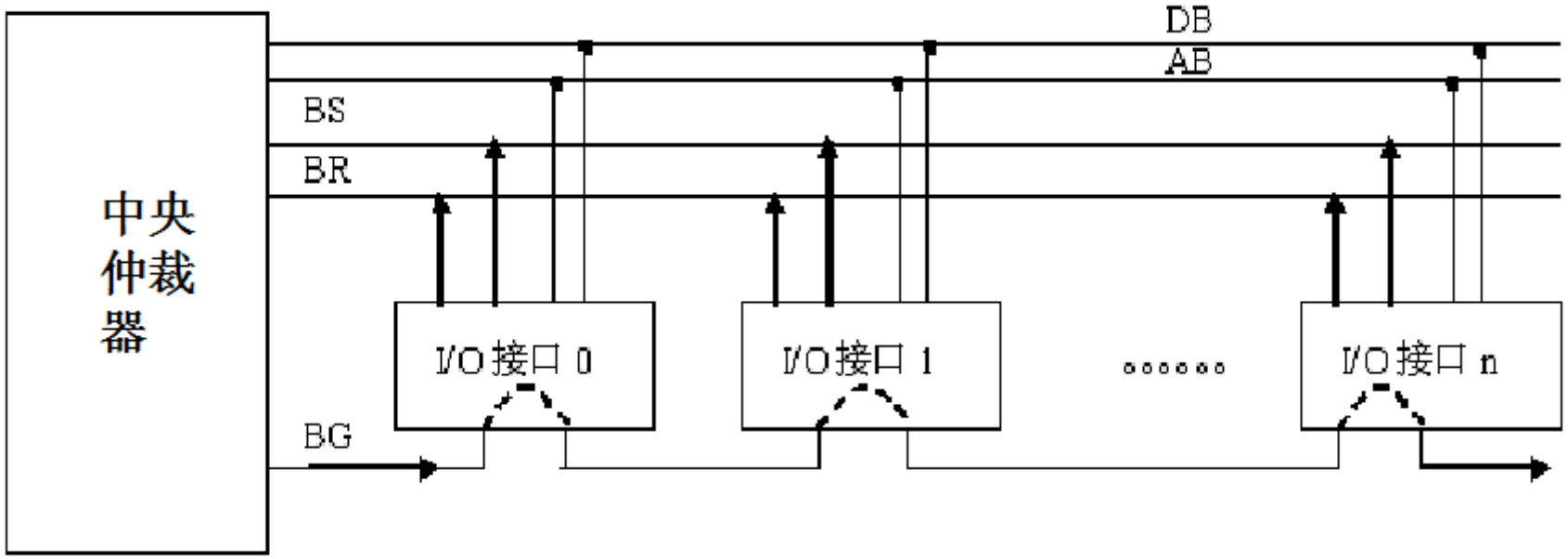
集中式仲裁中每个功能模块有两条线连到中央仲裁器：一条是送往仲裁器的总线请求信号线**BR**，一条是仲裁器送出的总线授权信号线**BG**。

BS（总线忙）：当某外设正使用总线时，**BS**=“1”。

集中控制是单总线、双总线和三总线结构机器中主要采用的方式。

链式查询方式、定时器查询方式、独立请求方式

链式查询方式



链式查询方式的主要特点：

总线授权信号**BG**串行地从一个**I/O**接口传送到下一个**I/O**接口。

假如**BG**到达的接口无总线请求，则继续往下查询；假如**BG**到达的接口有总线请求，**BG**信号便不再往下查询，该**I/O**接口获得了总线控制权。

离中央仲裁器最近的设备具有最高优先级，通过接口的优先级排队电路来实现。

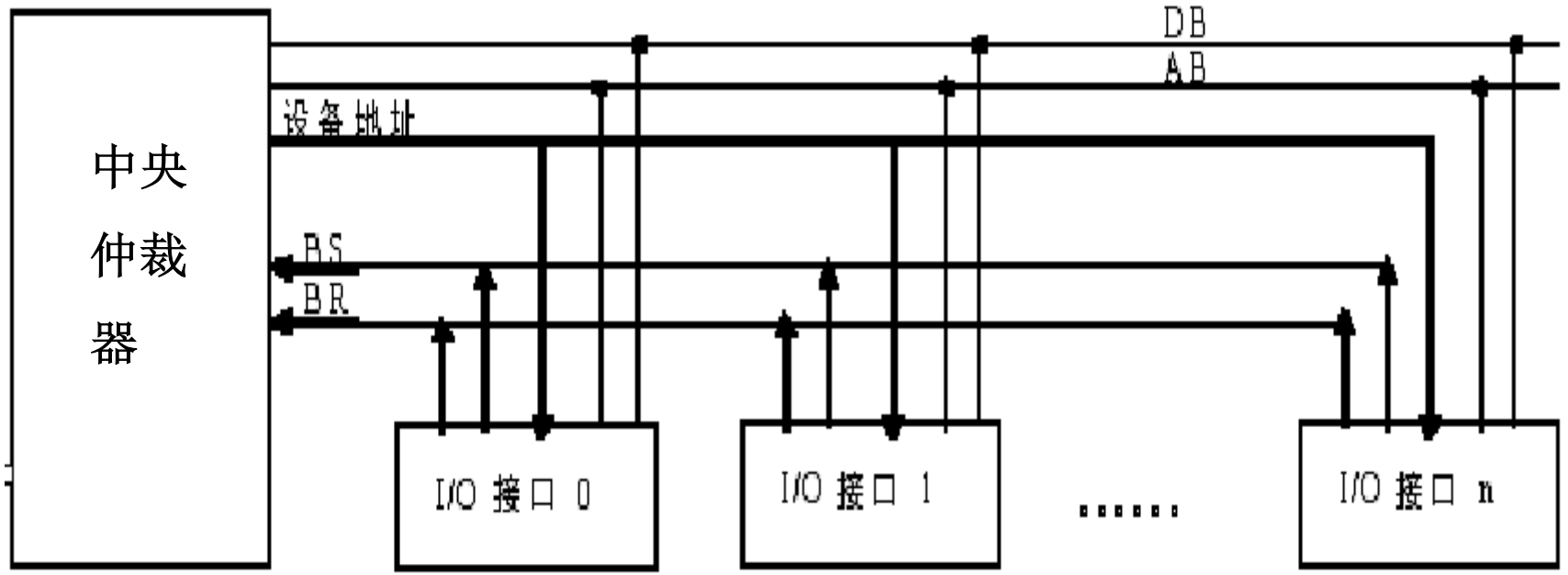
链式查询方式的优点：

只用很少几根线就能按一定优先次序实现总线仲裁，很容易扩充设备。

链式查询方式的缺点：

对询问链的电路故障很敏感，如果第i个设备的接口中有关链的电路有故障，那么第i个以后的设备都不能进行工作。查询链的优先级是固定的，如果优先级高的设备出现频繁的请求时，优先级较低的设备可能长期不能使用总线。

定时器查询方式

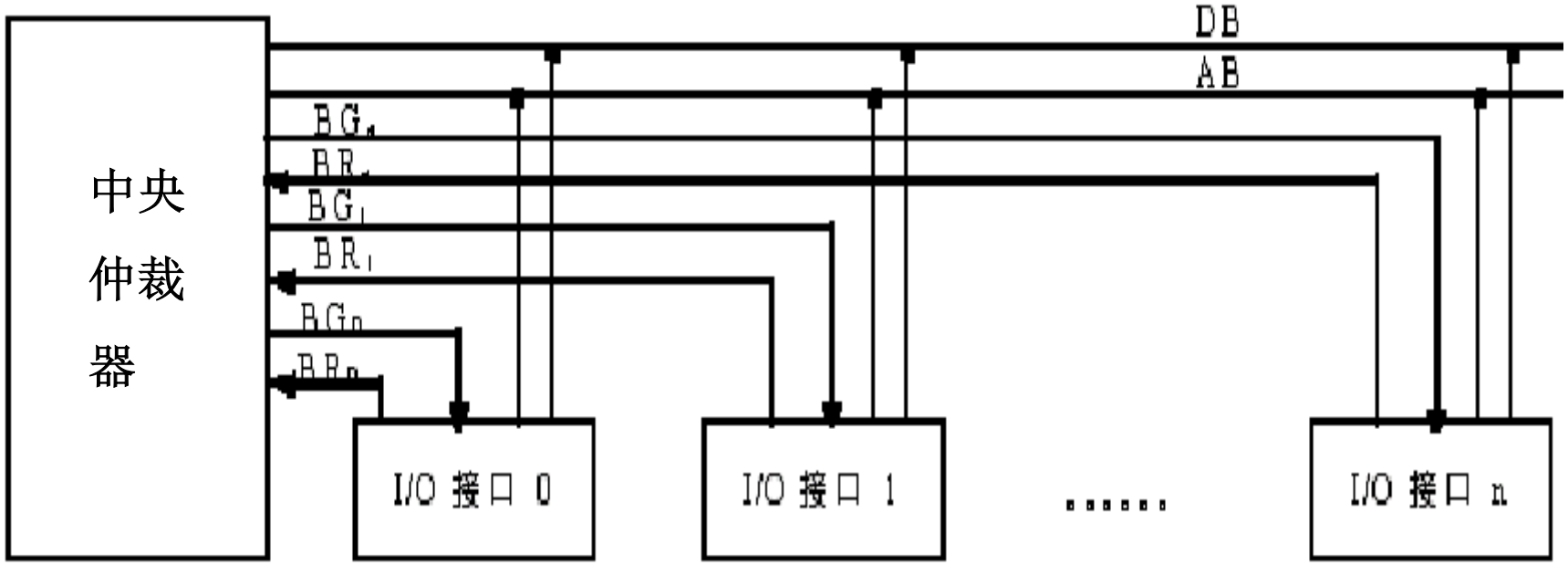


总线上的任一设备要求使用总线时，通过**BR**线发出总线请求。中央仲裁器接到请求信号以后，在**BS**线为“**0**”的情况下让计数器开始计数，计数值通过一组地址线发向各设备。每个设备接口都有一个设备地址判别电路，当地址线上的计数值与请求总线的设备地址相一致时，该设备置“**1**”**BS**线，获得了总线使用权，此时中止计数查询。

每次计数可以从“0”开始，也可以从中止点开始。如果从“0”开始，各设备的优先次序与链式查询法相同，优先级的顺序是固定的。如果从中止点开始，则每个设备使用总线的优先级相等。

计数器的初值也可用程序来设置，这可以方便地改变优先次序，但这种灵活性是以增加线数为代价的。

独立请求方式



每一个共享总线的设备均有一对总线请求线**BRi**和总线授权线**BGi**。当设备要求使用总线时，便发出该设备的请求信号。中央仲裁器中的排队电路决定首先响应哪个设备的请求，给设备以授权信号**BGi**。

独立请求方式的优点：

首先响应时间快，确定优先响应的设备所花费的时间少，用不着一个设备接一个设备地查询。

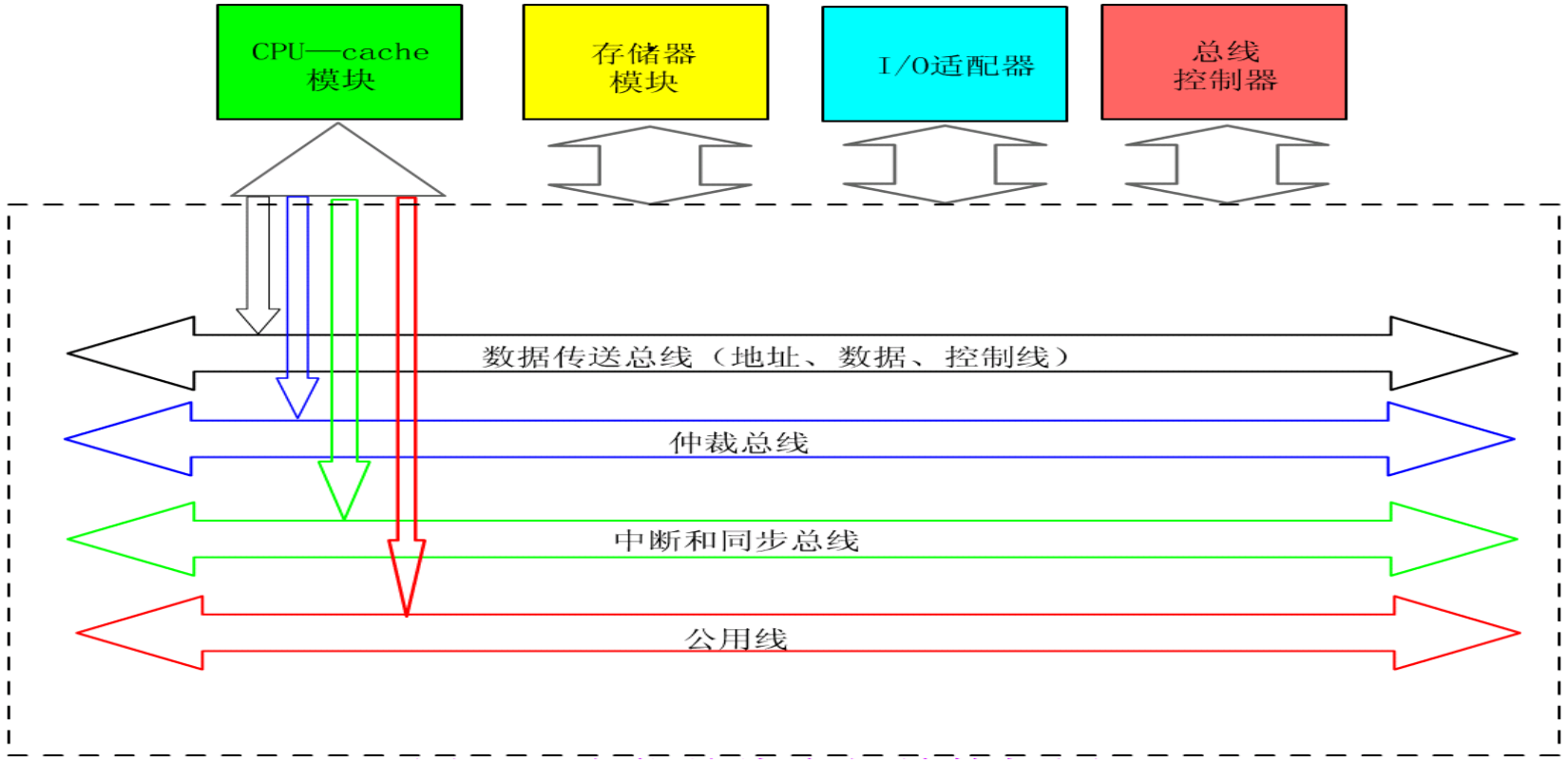
其次，对优先次序的控制相当灵活，可以预先固定也可以通过程序来改变优先次序；

还可以用屏蔽(禁止)某个请求的办法，不响应来自无效设备的请求。

因此，当代总线标准普遍采用独立请求方式。

中央仲裁器

对单处理器系统总线来说，中央仲裁器又称总线控制器，它是CPU的一部分。按照目前的总线标准，中央仲裁器一般是一个单独的功能模块。



② 分布式仲裁

- 分布式仲裁不需要中央仲裁器，每个潜在的主方功能模块都有自己的仲裁号和仲裁器。
- 当它们有总线请求时，把它们唯一的仲裁号发送到共享的仲裁总线上，每个仲裁器将仲裁总线上得到的号与自己的号进行比较。
- 如果仲裁总线上的号大，则它的总线请求不予响应，并撤消它的仲裁号。
- 最后，获胜者的仲裁号保留在仲裁总线上。

分布式仲裁是以优先级仲裁策略为基础。

2. 总线的定时

总线的一次信息传送过程，大致可分为如下五个阶段：

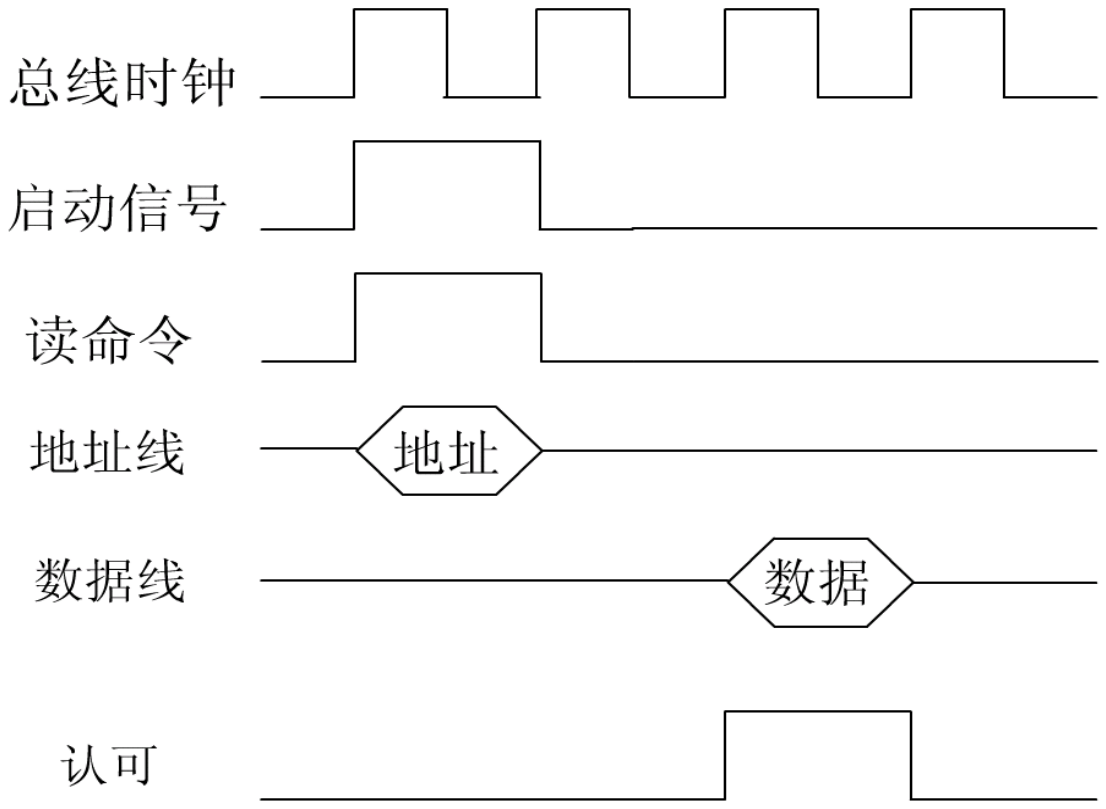
请求总线，总线仲裁，寻址(目的地址)，信息传送，状态返回(或错误报告)

为了同步主方、从方的操作，必须制订定时协议。

定时：事件出现在总线上的时序关系。

(1) 同步定时

在同步定时协议中，事件出现在总线上的时刻由总线时钟信号来确定。



读数据的同步时序

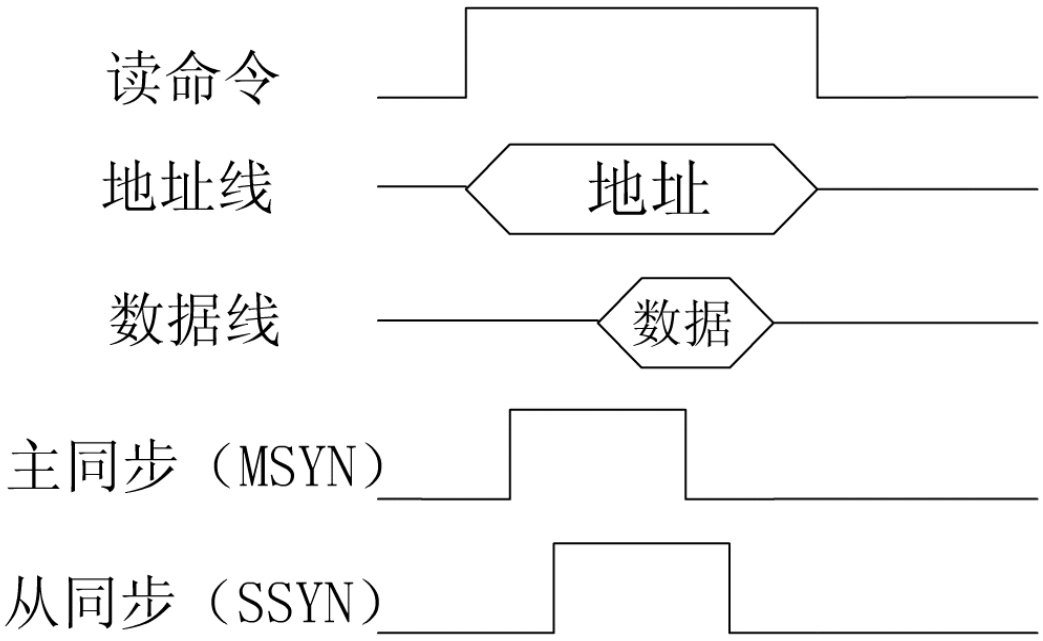
所有事件都出现在时钟信号的前沿。

由于采用了公共时钟，每个功能模块什么时候发送或接收信息都由统一时钟规定，因此具有较高的传送频率。

同步定时适用于总线长度较短、各功能模块存取时间比较接近的情况。

(2) 异步定时

在异步定时协议中，后一事件出现在总线上的时刻取决于前一事件的出现，即建立在应答式或互锁机制基础上。



在这种系统中，不需要统一的共公时钟信号。总线周期的长度是可变的。

异步定时的优点是总线周期长度可变，不把响应时间强加到功能模块上，因而允许快速和慢速的功能模块都能连接到同一总线上。但这以增加总线的复杂性和成本为代价。

总线的同步方式

1. 同步通信

串行同步：信号编码

并行同步：专用时钟信号线

优点：时序关系简单，实现简单。

缺点：在设备速度不一致时按最坏情况确定，不能太长。

2. 异步通信

串行异步通信：起始检测

并行异步通信：握手信号

非互锁

全互锁：（四边沿协议）

3. 总线数据传送方式

当代的总线标准大都能支持以下四类模式的数据传送：

(1) 读、写操作

读操作是由从方到主方的数据传送；
写操作是由主方到从方的数据传送。

- 一般，主方先以一个总线周期发出命令和从方地址，经过一定的延时再开始数据传送总线周期。
- 为了提高总线利用率，减少延时损失，主方完成寻址总线周期后可让出总线控制权，以使其他主方完成更紧迫的操作。然后再重新竞争总线，完成数据传送总线周期。

(2) 块传送操作

只需给出块的起始地址，然后对固定块长度的数据一个接一个地读出或写入。

对于CPU(主方)、存储器(从方)而言的块传送，常称为猝发式传送，其块长一般固定为数据线宽度(存储器字长)的4倍。

例如： 64位数据线的总线，一次猝发式传送可达256位。

(3) 写后读、读修改写操作

只给出地址一次，或进行先写后读操作，或进行先读后写操作。

- 先写后读操作：用于校验；
- 先读后写操作：用于多道程序系统中对共享存储资源的保护。

这两种操作和猝发式操作一样，主方掌管总线直到整个操作完成。

(4) 广播、广集操作

广播：

一般而言，数据传送只在一个主方和一个从方之间进行。但有的总线允许一个主方对多个从方进行写操作，这种操作称为广播。

广集：

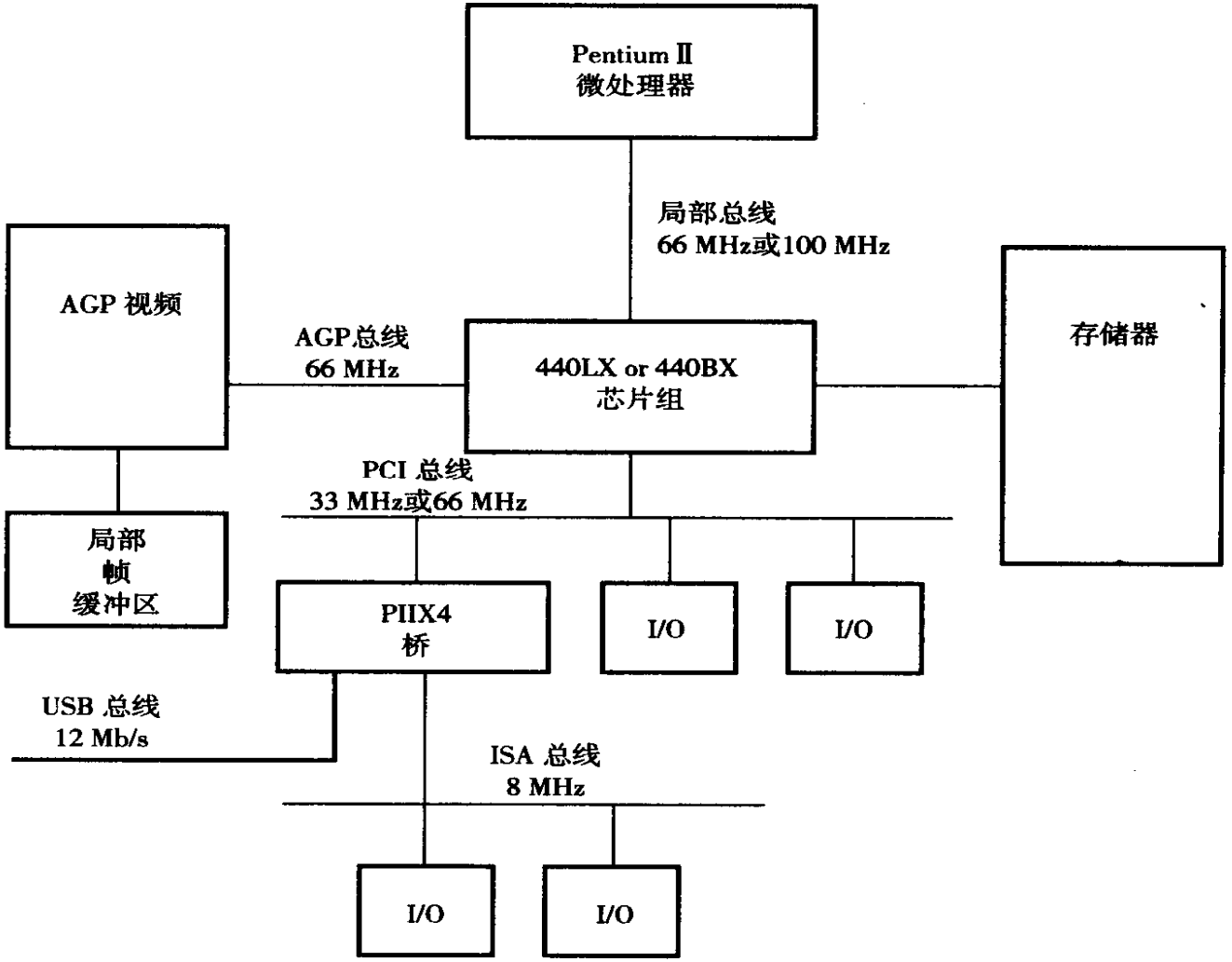
与广播相反的操作称为广集，它将选定的多个从方数据在总线上完成AND或OR操作，用以检测多个中断源。

总线标准

- ISA
- PCI
- AGP
- PCI-E

总线——AGP总线

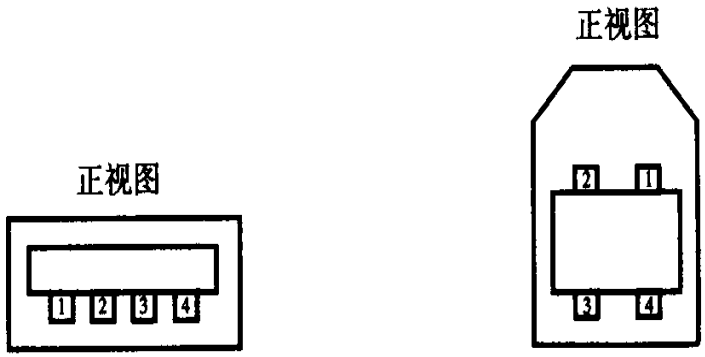
AGP——加速图形端口



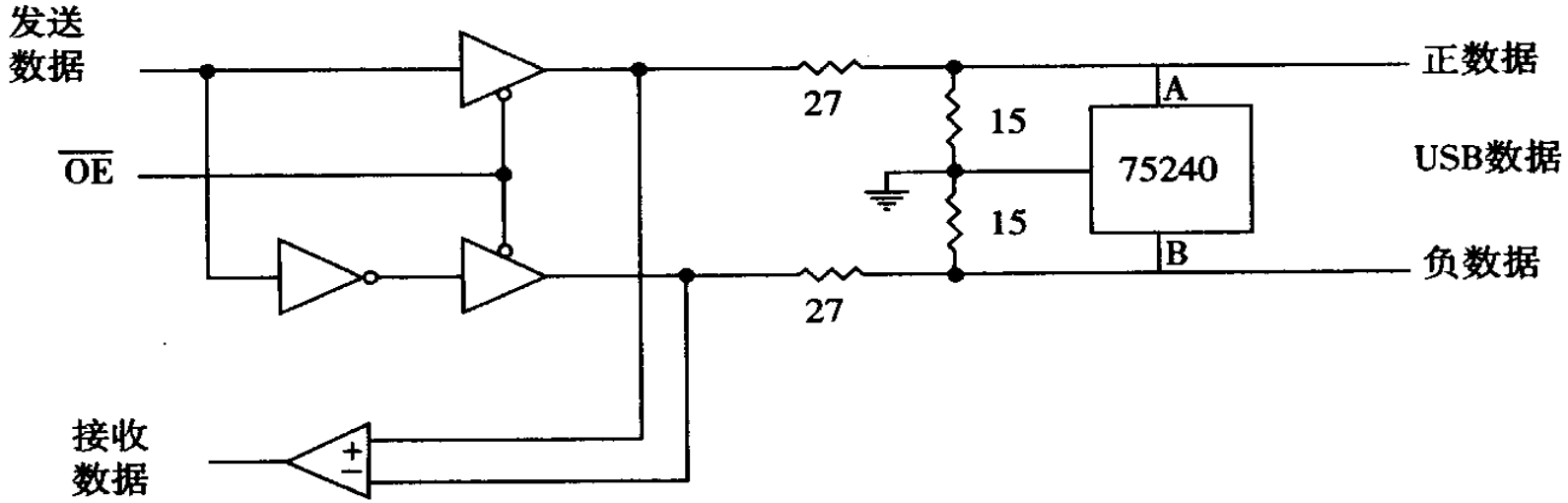
总线——USB总线

USB——通用串行总线

全速12Mb/s(<=5米)，慢速1.5Mb/s(<=3米)



引脚号	信号
1	5.0V
2	负数据
3	正数据
4	地



外围设备的定时方式与信息交换方式

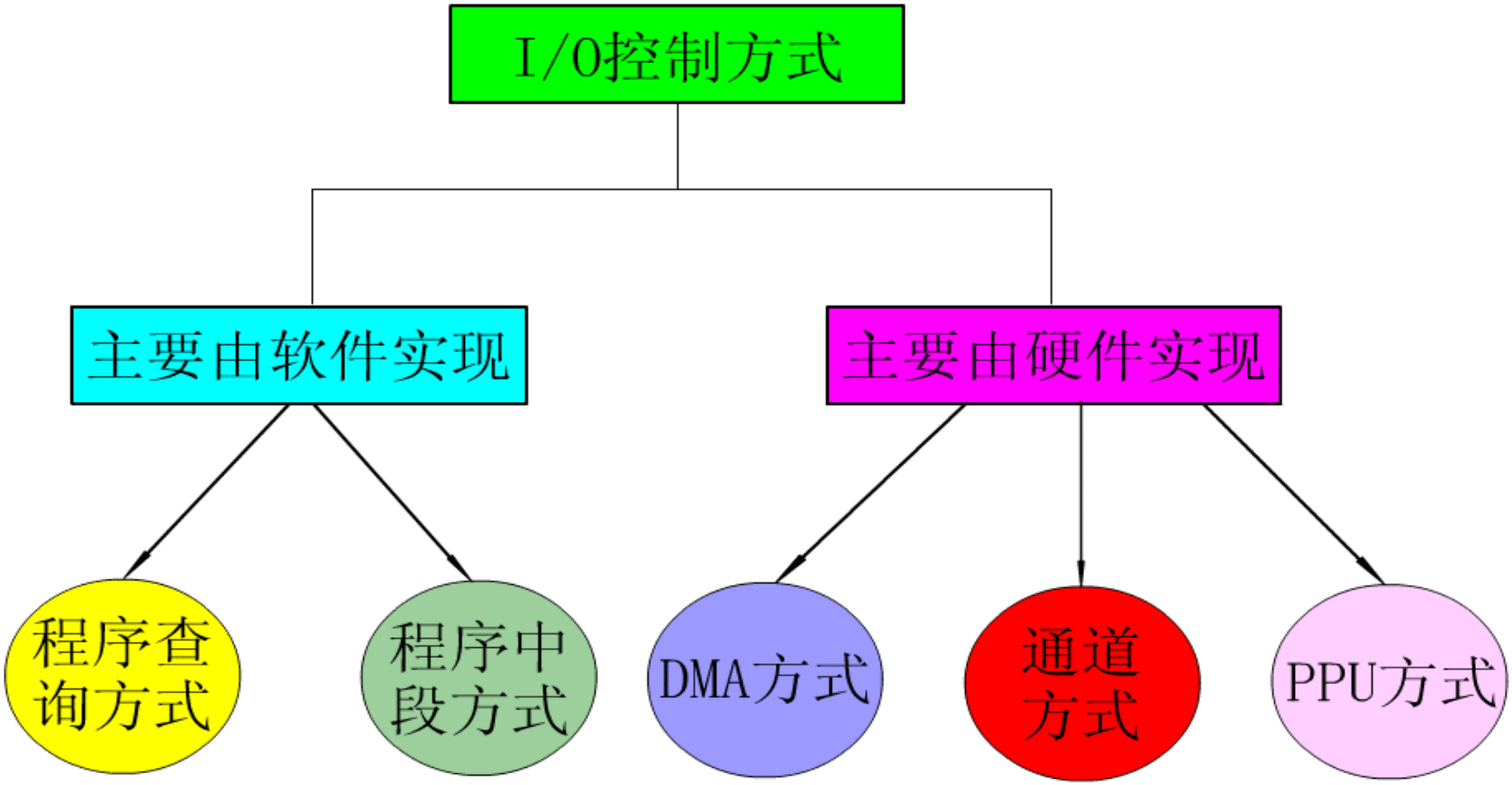
程序方式

中断方式

DMA方式

通道处理机

2. 信息交换方式



程序查询方式

无条件传送方式

条件传送方式

(1)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。

(2)程序中断方式

中断是外围设备“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。

(3)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。

(4) 通道方式

DMA方式的出现已经减轻了CPU对I/O操作的控制，使得CPU的效率有显著的提高，而通道的出现则进一步提高了CPU的效率。这是因为，CPU将部分权力下放给通道。

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。

这种提高CPU效率的办法是以花费更多硬件为代价的。

(5) 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。

(6) 各种方法比较：

- 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，
- **DMA**方式、通道方式 和**PPU**方式适用于数据传输率比较高的设备。

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。

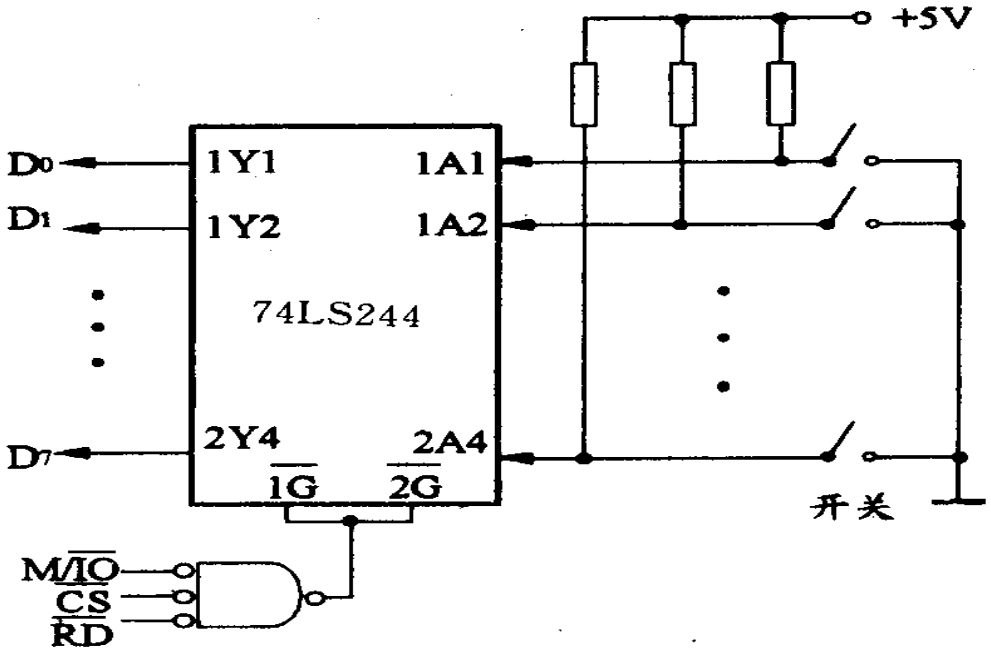
程序查询方式——无条件传送

外设总是准备好

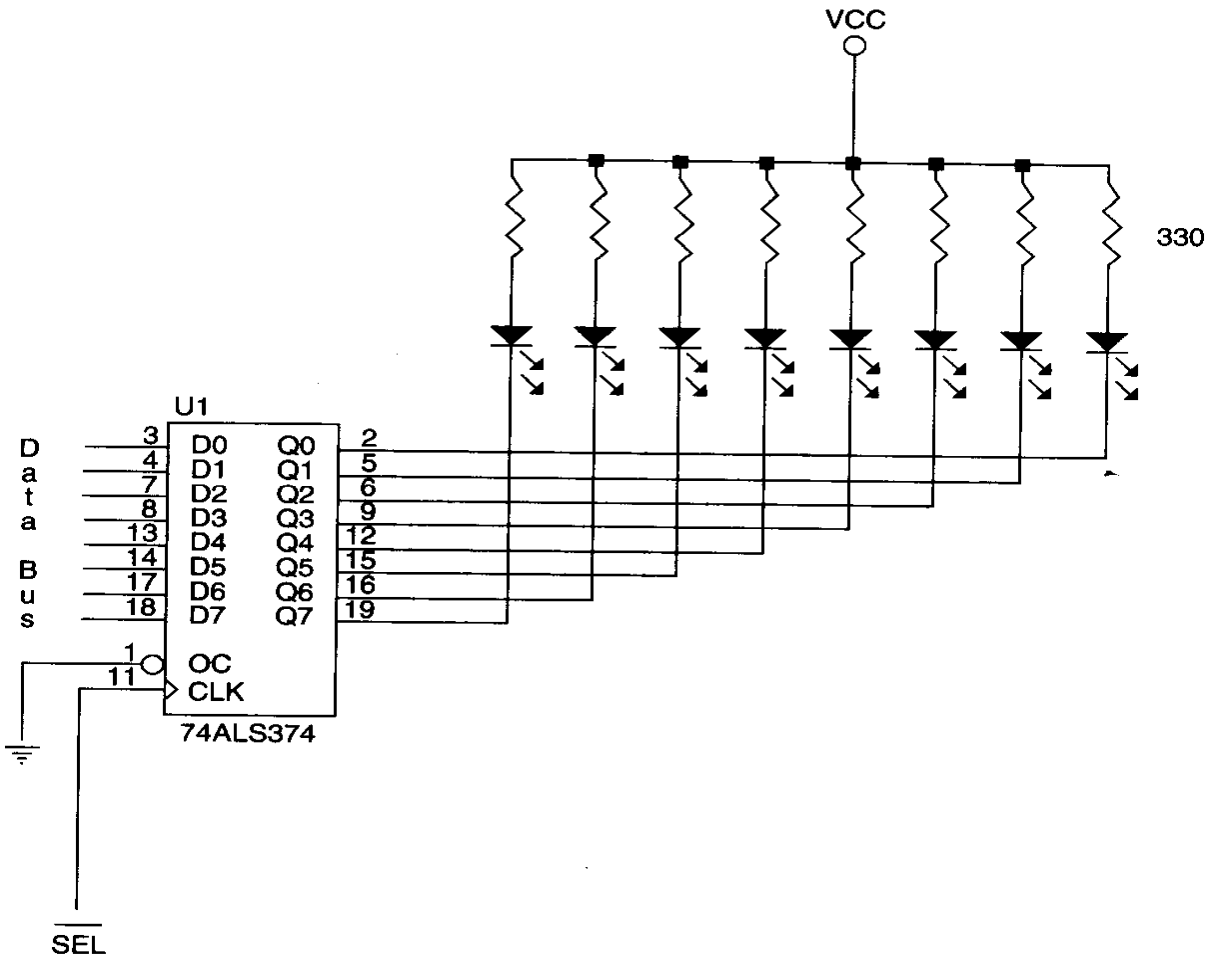
- 输入——数据已经准备好
- 输出——已准备好接收

只有数据，没有状态，同步方式

不需要过多的程序处理，在需要与外设交换信息时，随时访问I/O端口

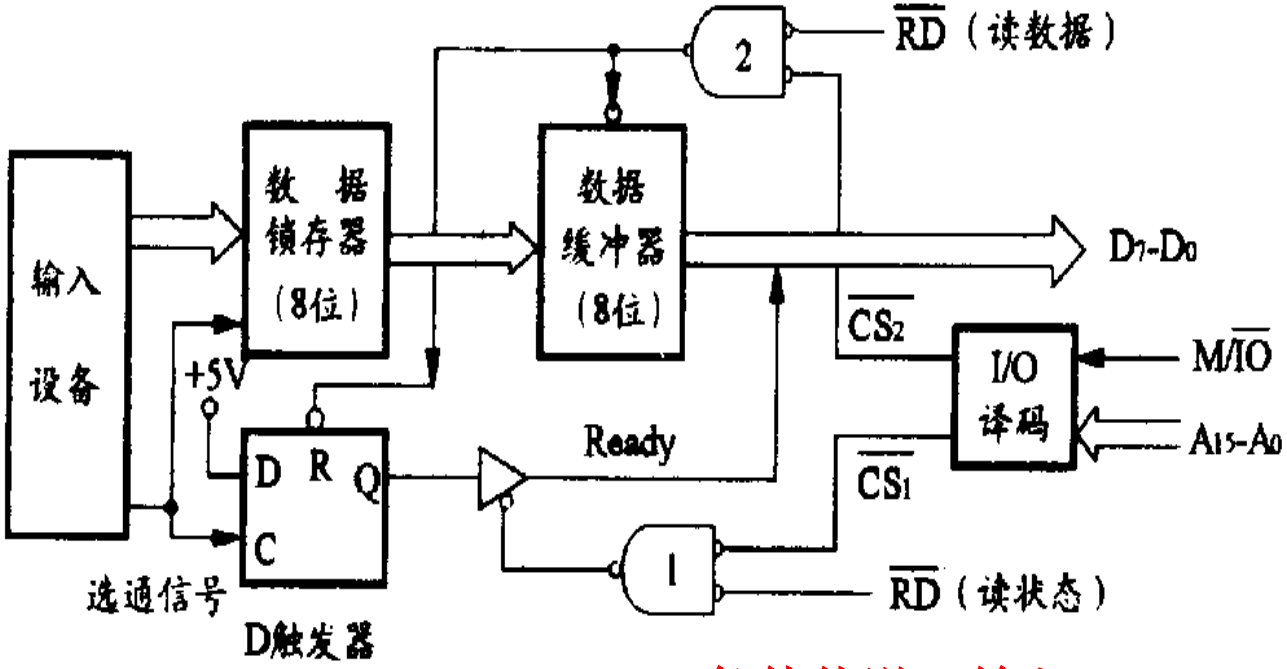


程序查询方式——无条件传送



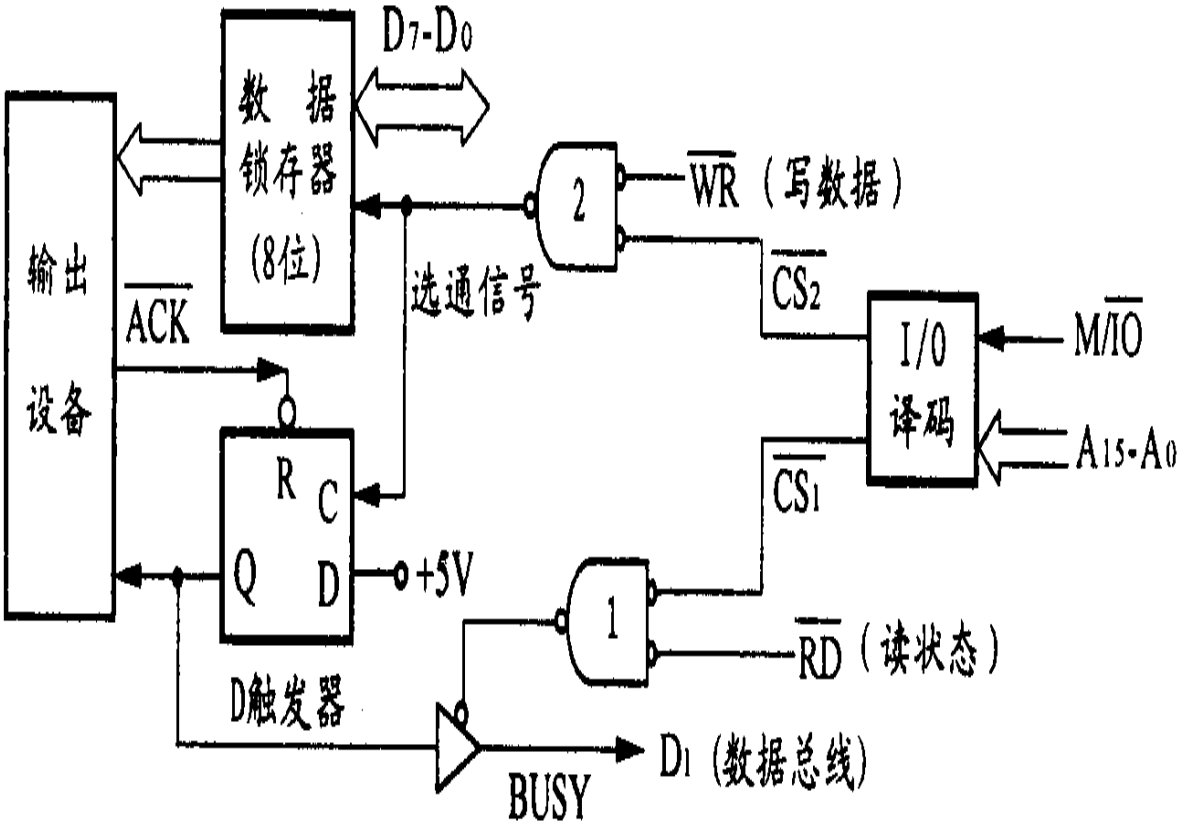
程序查询方式——条件传送

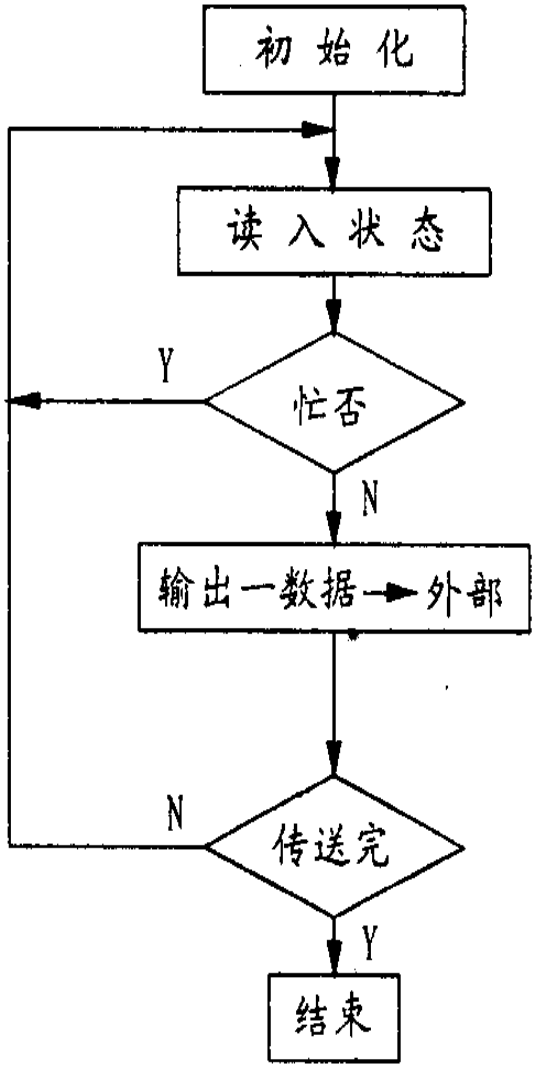
- 查询传送方式
- 查询外设的状态信息
  - 输入——数据已准备好
  - 输出——接收装置已准备好
- 状态端口、数据端口



条件传送---输入

条件传送---输出





- 特点：
  - 控制简单,CPU和外设只能串行工作，系统效率低
  - CPU在一个时间内只能和一个外设交换信息。

程序查询方式特点：

1、何时对何设备输入/输出操作完全由　CPU控制

2、外设与CPU处于异步工作方式

3、外设与CPU处于异步工作方式

4、数据的输入/输出要经过CPU，至少要几条指令

5、CPU利用率低，但控制简单

6、实时性？

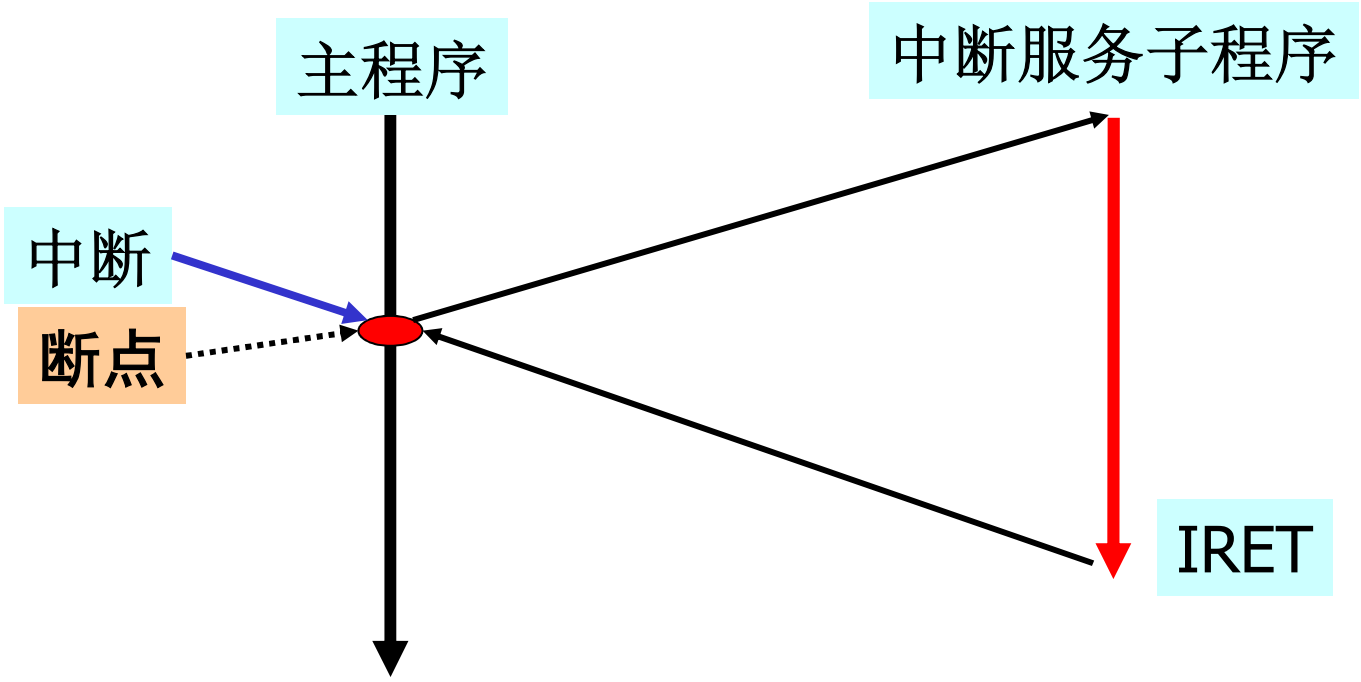
某计算机处理器主频为50MHz，采用定时查询方式控制设备A的I/O，查询程序运行一次所用的时钟周期数至少为500。在设备A工作期间，设备A每秒最多与CPU进行10次数据交换，其交换数据的最小间隔为0.5ms,为保证数据不丢失,则CPU用于设备A的I/O的时间占整个CPU时间的百分比至少是多少（　）

A．0.2%　B．0.05%　C．2.0%　D．0.50%

程序中断方式

1. 中断的基本概念

当CPU正常运行程序时，由于内部事件或外设请求(随机的)，引起CPU暂时中止正在运行的程序，转去执行发出请求的外设（或内部事件）的服务子程序，待该服务程序执行完毕，再返回被中止的程序，这一过程称为中断。



2. 中断的作用

“中断”是由I/O设备或其他非预期的急需处理的事件引起的，它使CPU暂时中断现在正在执行的程序，而转至另一服务程序去处理这些事件，处理完后再返回原程序。

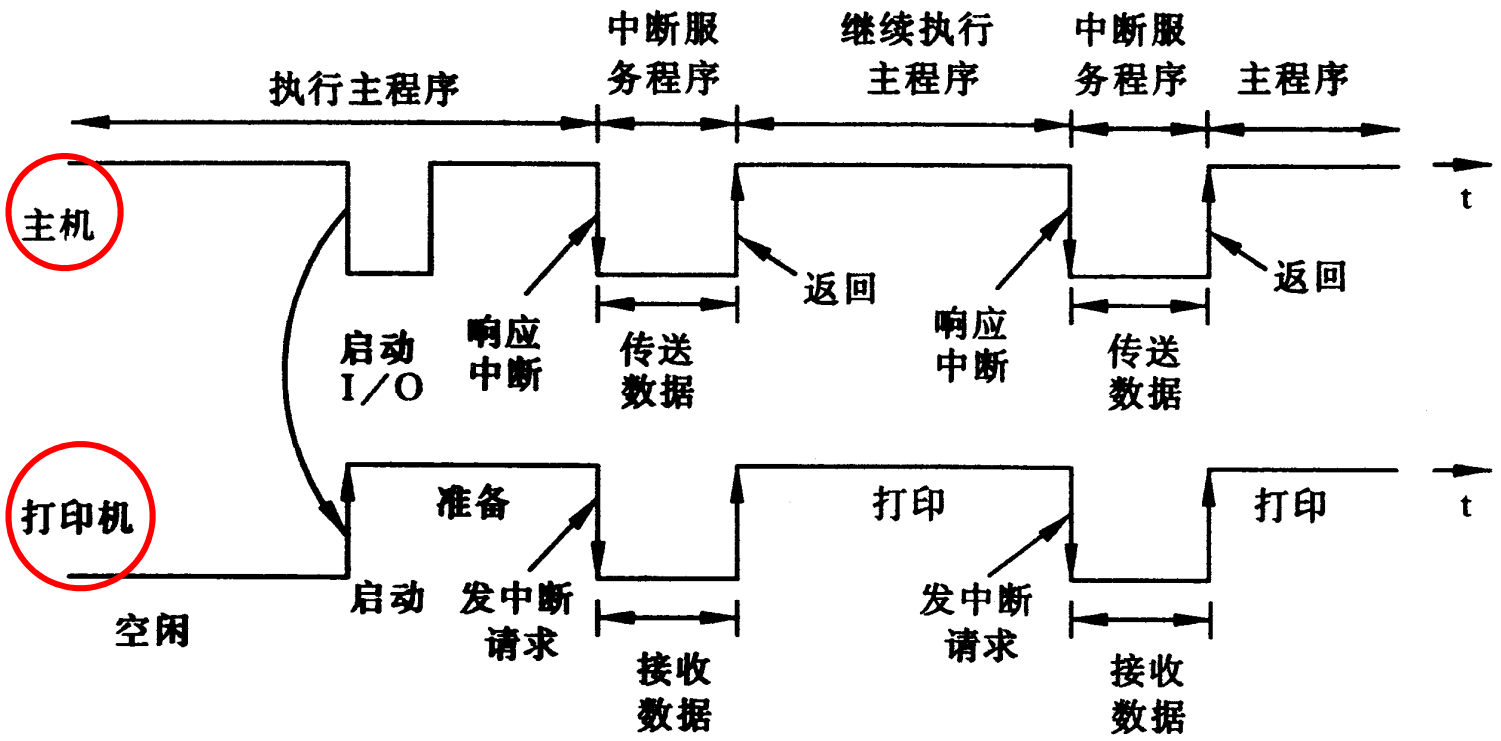
中断有下列一些作用：

(1) CPU与I/O设备并行工作。

例CPU与针式打印机并行工作的时间安排。 当打印机完成一行打印后，向CPU发中断信号，若CPU响应中断，则停止正在执行的程序转入打印中断服务程序，将要打印的下一行字传送到打印机控制器并启动打印机工作。然后CPU又继续执行原来的程序，此时打印机开始了新一行字的打印过程。

CPU与打印机并行工作

打印机打印一行字需要几毫秒到几十毫秒，而中断处理时间一般是微秒级



(2)硬件故障处理。 计算机运行时，如硬件出现某些故障，机器中断系统发出中断请求，CPU响应中断后自动进行处理。

(3)实现人机联系。 在计算机工作过程中，如果用户要干预机器，如抽查计算中间结果，了解机器的工作状态，给机器下达临时性的命令等。在没有中断系统的机器里这些功能几乎是无法实现的。

(4)实现多道程序和分时操作。 计算机实现多道程序运行是提高机器效率的有效手段。多道程序的切换运行需借助于中断系统。在一道程序的运行中，由I/O中断系统切换到另外一道程序运行。 也可以通过分时分配每道程序一个固定时间片，利用时钟定时发中断进行程序切换。

(5)实现实时处理。 这是指在某个事件或现象出现时，及时地进行处理，而不是集中起来再进行批处理。例如，在某个计算机过程控制系统中，当出现压力过大，温度过高等情况时，必须及时输入到计算机进行处理。 这些事件出现的时刻是随机的，而不是程序本身所能预见的，因此要求计算机中断正在执行的程序，转而去执行中断服务程序

(6)实现应用程序和操作系统(管态程序)的联系。可以在用户程序中安排一条“Trap”指令进入操作系统，称之为“软中断”。其中断处理过程与其他中断类似。

(7)多处理机系统中各处理机间的联系。 在多处理机系统中，处理机和处理机之间的信息交流和任务切换可以通过中断来实现。

3. 中断与调用子程序的区别

从表面上看起来，计算机的中断处理过程有点类似于调用子程序的过程，这里现行程序相当于主程序，中断服务程序相当于子程序。

但有本质上的区别：

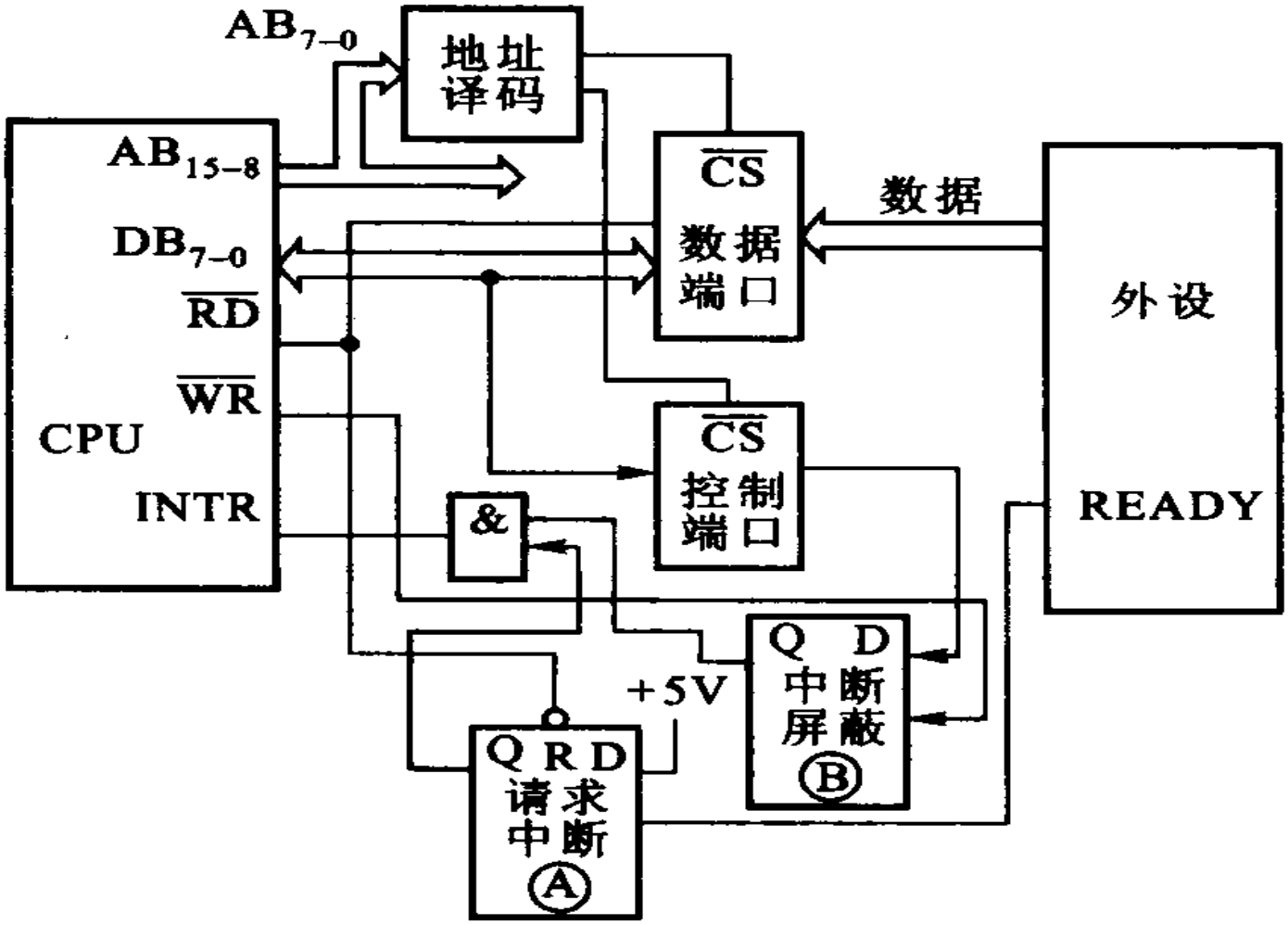
(1)子程序的执行是由程序员事先安排好的，而中断服务程序的执行则是由随机的中断事件引起的。

(2)子程序的执行受到主程序或上层子程序的控制，而中断服务程序一般与被中断的现行程序毫无关系。

(3)不存在同时调用多个子程序的情况，而可能发生多个外设同时请求**CPU**为自己服务的情况。

特点：CPU的利用率高

中断接口


AB7–0
地址译码
AB15–8
DB7–0
RD
WR
CPU
INTR
CS
数据端口
数据
外设
CS
控制端口
READY
&
Q D
中断屏蔽
B
+5V
Q R D
请求中断
A

单选题 1分

设置

某计算机处理器主频为50MHz，采用定时查询方式控制设备A的I/O，如果采用中断方式，中断服务程序运行一次所用的时钟周期数至少为500。在设备A工作期间，设备A每秒最多与CPU进行10次数据交换，其交换数据的最小间隔为0.5ms,则CPU用于设备A的I/O的时间占整个CPU时间的百分比至少是多少（ ）

A 0.01%

B 0.05%

C 2.0%

D 0.5%

提交

某计算机处理器主频为50MHz，采用定时查询方式控制设备A的I/O，如果采用中断方式，中断服务程序运行一次所用的时钟周期数至少为500。在设备A工作期间，设备A每秒最多与CPU进行10次数据交换，其交换数据的最小间隔为0.5ms,则CPU用于设备A的I/O的时间占整个CPU时间的百分比至少是多少（　）

A．0.01%　B．0.05%　C．2.0%　D．0.50%

4.中断的产生和响应

(1)中断源---引起中断的事件。

①中断源的种类

外中断--I/O设备，定时时钟等来自处理机外部设备的中断。

内中断--处理机硬件故障或程序“出错”引起的中断。

例如，电源故障，算术溢出，除数为零，校验错，指令非法，用户程序执行特权指令以及虚拟存储器页面失效等。

软中断--由“Trap”指令产生的中断。

②中断触发器

当中断源发生引起中断的事件时，先将它保存在设备控制器的“中断触发器”中，即置“1”，当中断触发器为“1”时，向CPU发出“中断请求”信号

每个中断源有一个中断触发器。多个中断触发器构成中断寄存器。其内容称为中断字或中断码。 CPU进行中断处理时，根据中断字确定中断源，转入相应的服务程序。

(2)中断的分级与中断优先权

在设计中断系统时，要把全部中断源按中断性质和处理的轻重缓急进行排队并给予优先权。优先权是指有多个中断同时发生时，对各个中断响应的优先次序。

中断源数量很多时，中断字就会很长，一般把所有中断按不同的类别分为若干级，称为中断级，在同一级中还可以有多个中断源。首先按中断级确定优先次序，然后在同一级内再确定各个中断源的优先权。

对设备分配优先权时，必须考虑数据的传输率和服务程序的要求。数据的有效时间短，则该设备的优先权高

(3)禁止中断和中断屏蔽

①禁止中断：产生中断源后，由于某种条件的存在，**CPU**不能中止现行程序的执行，称为禁止中断。

- 一般在**CPU**内部设有一个“中断允许”触发器。
- 只有该触发器为“**1**”时，才允许处理机响应中断;
- 该触发器为“**0**”，则不响应所有中断源申请的中断。
- 前者叫允许中断，后者叫禁止中断。
- “中断允许”触发器是通过“开中断”或“关中断”指令来置位，复位。

进入中断服务程序后自动“关中断”

②中断屏蔽：当产生中断请求后，用程序方式有选择地封锁部分中断，而允许其余部分中断仍得到响应，称为中断屏蔽。实现方法是为每个中断源设置一个中断屏蔽触发器来屏蔽该设备的中断请求。

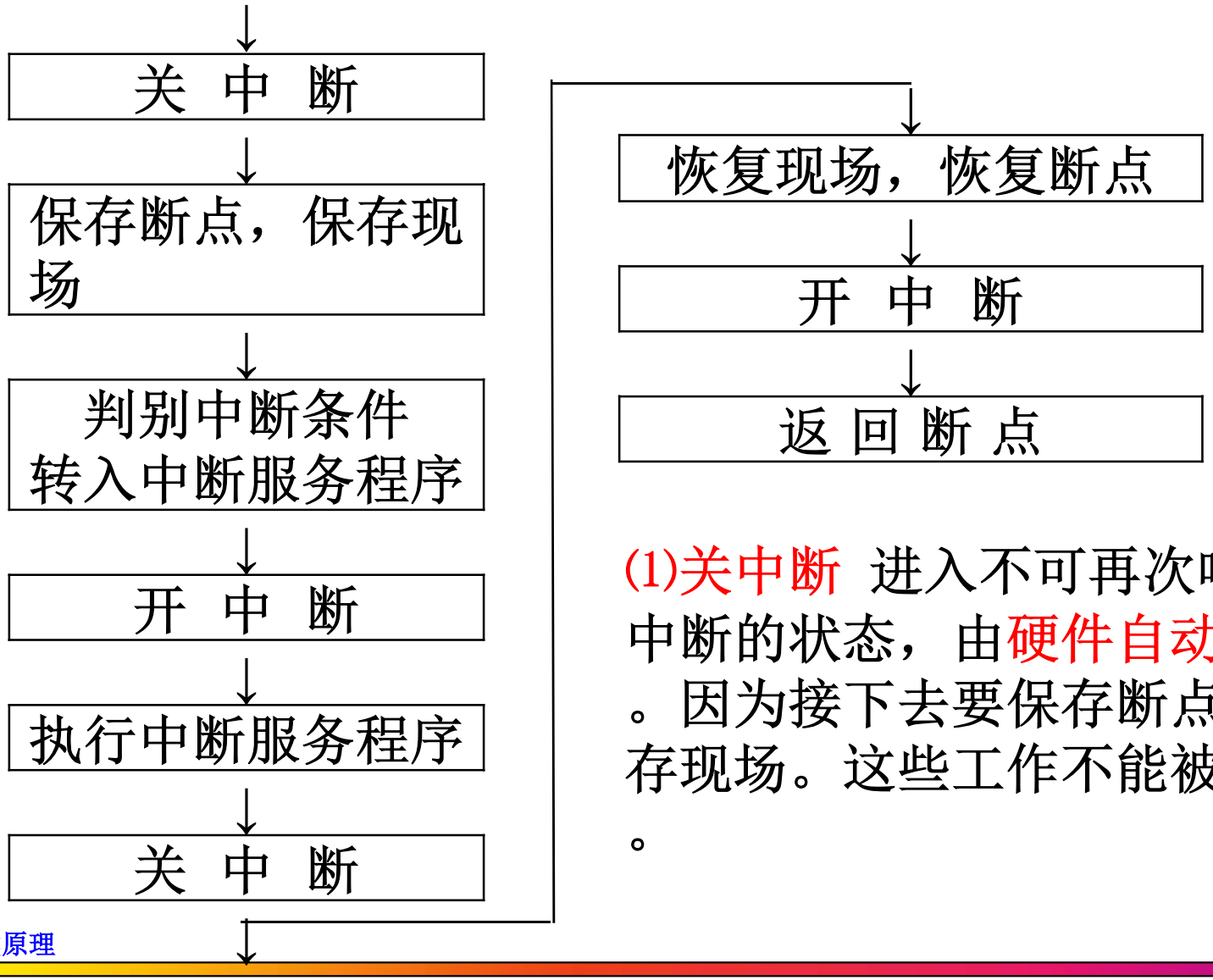
具体说，用程序方法将该触发器置“1”，则对应的设备中断被封锁;若将其置“0”，才允许该设备的中断请求得到响应。由各设备的中断屏蔽触发器组成中断屏蔽寄存器。

有些中断请求是不可屏蔽的，也就是说，不管中断系统是否开中断，这些中断源的中断请求一旦提出，CPU必须立即响应。例如电源掉电就是不可屏蔽中断。 所以，中断又可分为可屏蔽中断和非屏蔽中断。非屏蔽中断具有最高优先权。

中断的产生和响应(小结)

- 中断源
  - 种类：内部中断/外部中断
  - 建立：中断触发器
  - 分级：中断禁止/开放，中断优先级
- 中断响应的条件
  - 有中断请求；
  - **CPU**允许中断
  - 在一条指令执行完后响应。

中断处理过程



(1)关中断 进入不可再次响应中断的状态，由硬件自动实现。因为接下去要保存断点，保存现场。这些工作不能被打断。

(2)保存断点和现场　为了在中断处理结束后能正确地返回到中断点，在响应中断时，必须把当前的程序计数器PC中的内容(即断点)保存起来。

现场信息一般指的是程序状态字，中断屏蔽寄存器和CPU中某些寄存器的内容

对现场信息的处理有两种方式：一种是由硬件对现场信息进行保存和恢复；另一种是由软件即中断服务程序对现场信息保存和恢复。

对硬件保存现场信息的方式　有的机器把断点等保存在主存固定的单元；有的机器每次响应中断后把处理机状态字和程序计数器内容相继压入堆栈，再从指定的两个主存单元分别取出新的程序计数器内容和处理机状态字来代替，称为交换新，旧状态字方式。

(3)判别中断源转向中断服务程序　在多个中断源同时请求中断的情况下，本次实际响应的只能是优先权最高的那个中断源。所以需进一步判别中断源，并转入相应的中断服务程序入口。

(4)开中断　因为接下去就要执行中断服务程序，开中断将允许更高级中断请求得到响应，实现中断嵌套。

(5)执行中断服务程序　不同中断源的中断服务程序是不同的，实际有效的中断处理工作是在此程序段中实现的。

(6)退出中断　在退出时，又应进入不可中断状态，即关中断，恢复现场，恢复断点，然后开中断，返回原程序执行进入中断时执行的关中断，保存断点等操作一般是由硬件实现的，它类似于一条指令，但它与一般的指令不同，不能被编写在程序中。因此，常常称为“中断隐指令”。

5. 单级中断与多级中断

根据计算机系统对中断处理的策略不同，可分为单级中断系统和多级中断系统。

(1) 单级中断的概念

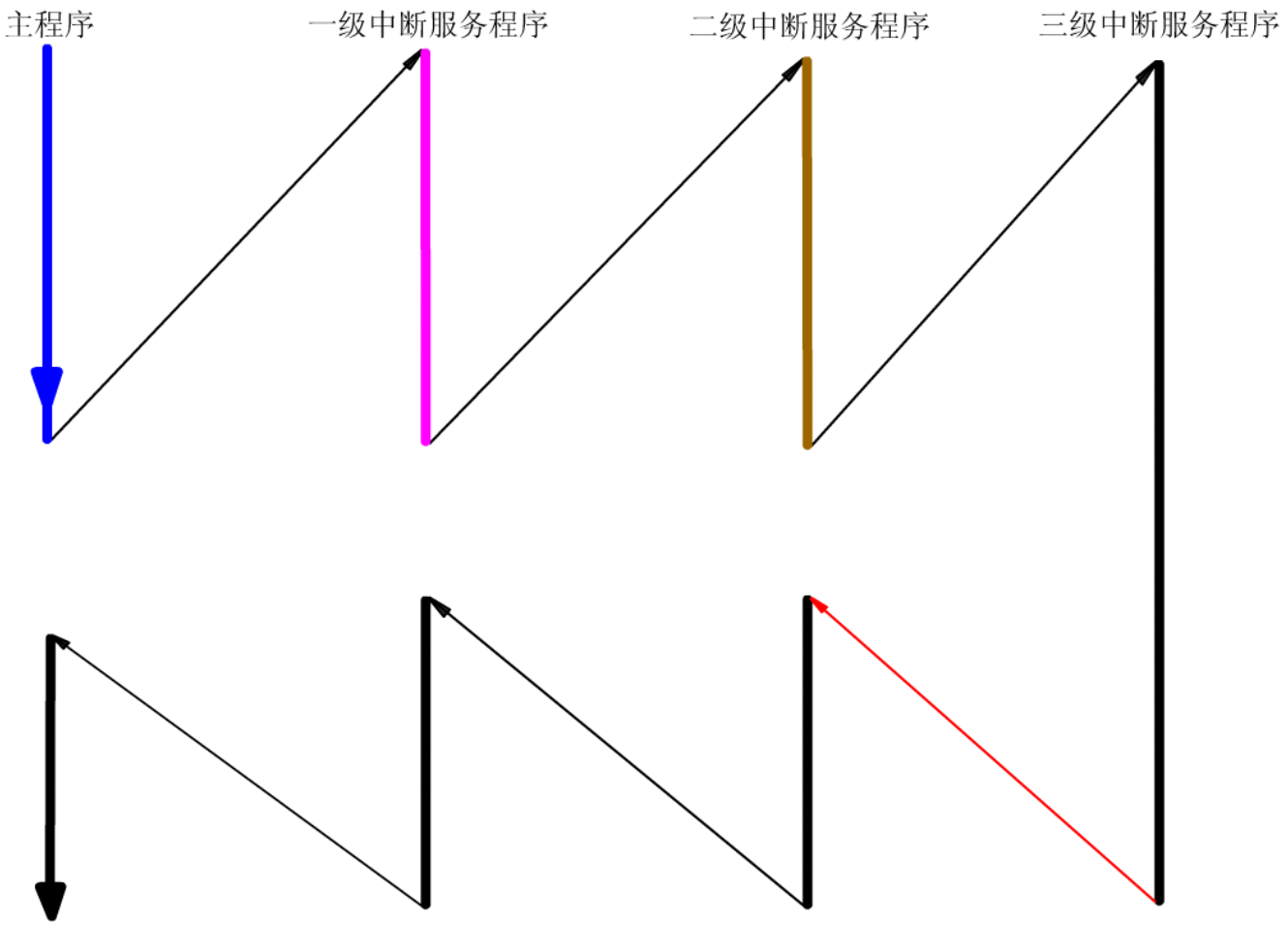
- 在单级中断系统中，所有的中断源都属于同一级，所有中断源触发器排成一行，其优先次序是离CPU近的优先权高。
- 当响应某一中断请求时，执行该中断源的中断服务程序。在此过程中，不允许其他中断源再打断中断服务程序，即使优先权比它高的中断源也不能再打断。

(2) 多级中断的概念

多级中断系统是指计算机系统中有相当多的中断源，根据各中断事件的轻重缓急程度不同而分成若干级别，每一中断级分配给一个优先权。优先权高的中断级可以打断优先权低的中断服务程序，以程序嵌套方式工作。

- 中断嵌套
  - 多个中断源，有可能同时发出中断请求
  - 多重中断响应（服务）
- 中断优先级
  - 给每个中断源规定优先级别，CPU先响应高级中断的请求
  - 一般情况下，在允许中断嵌套时，高级中断可以打断低级中断，同级或低级中断不能打断高级或同级中断

多级中断示意图

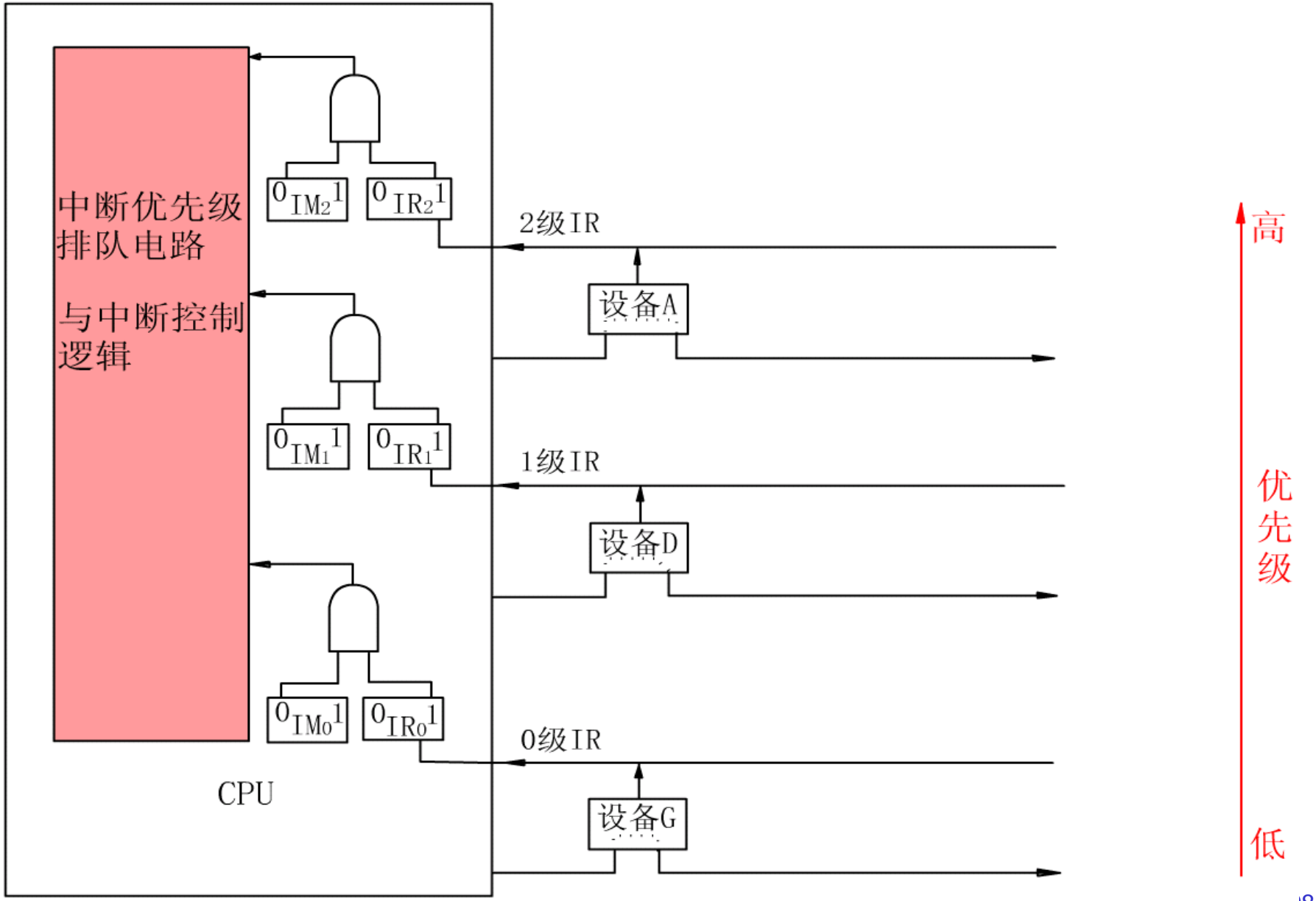


根据系统的配置不同，多级中断可分为一维多级中断和二维多级中断：

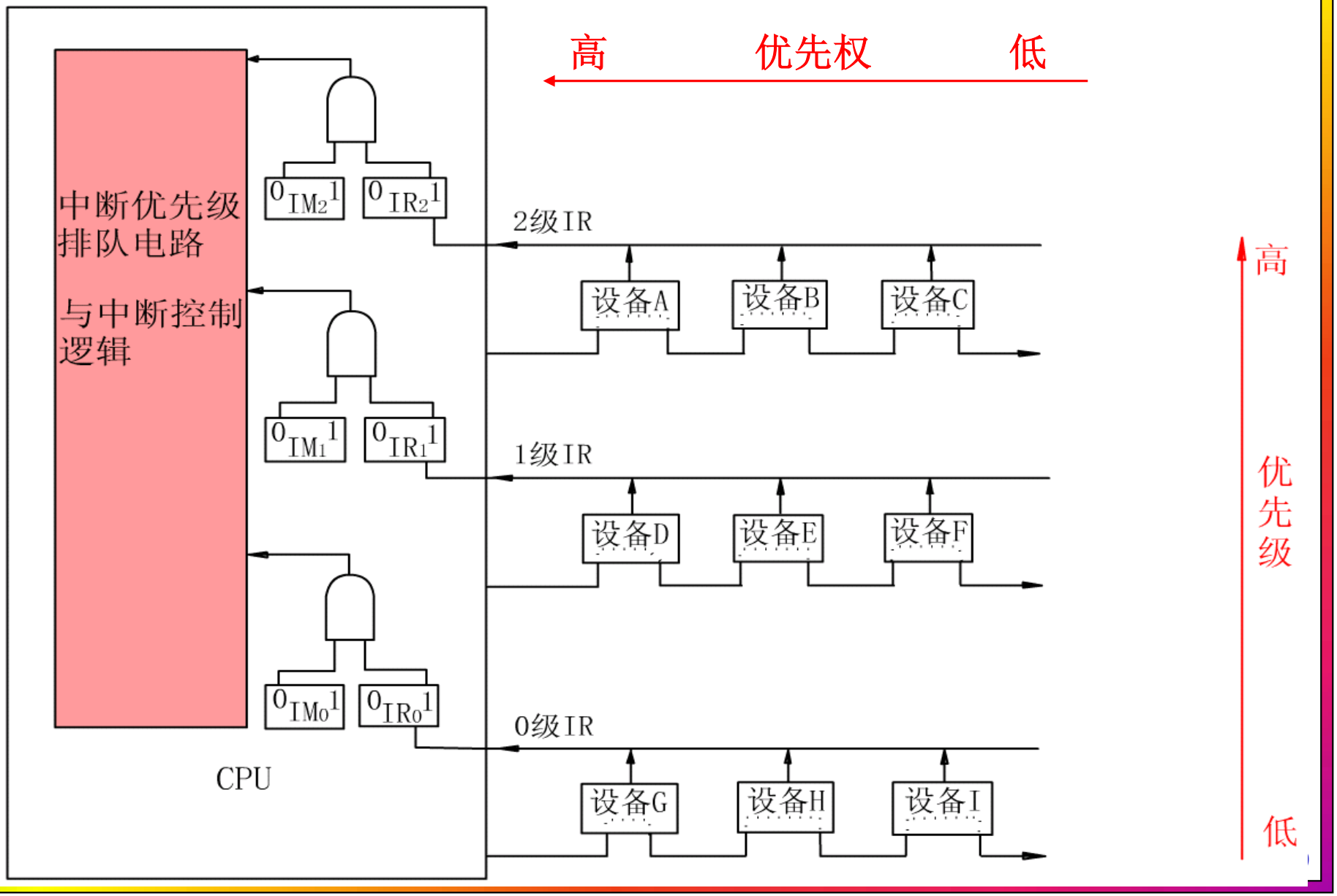
一维多级中断是指每一级中断里只有一个中断源，

二维多级中断是指每一级中断里又有多个中断源。

一维多级中断结构


中断优先级
排队电路
与中断控制
逻辑
CPU
$0_{IM_2}1$
$0_{IR_2}1$
$0_{IM_1}1$
$0_{IR_1}1$
$0_{IM_0}1$
$0_{IR_0}1$
2级IR
1级IR
0级IR
设备A
设备D
设备G
高
优先级
低

二维多级中断结构


高
优先权
低
中断优先级
排队电路
与中断控制
逻辑
$0_{IM_2}1$
$0_{IR_2}1$
$0_{IM_1}1$
$0_{IR_1}1$
$0_{IM_0}1$
$0_{IR_0}1$
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
CPU
高
优先级
低

几个问题：

① 一个系统若有n级中断，在CPU中就有n个中断请求触发器，总称为中断请求寄存器；与之对应的有n个中断屏蔽触发器，总称为中断屏蔽寄存器。

与单级中断不同，在多级中断中，中断屏蔽寄存器的内容是一个很重要的程序现场，因此在响应中断时，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。

② 多级中断中的每一级可以只有一个中断源，也可以有多个中断源。在多级中断之间可以实现中断嵌套，但是同一级内有不同中断源的中断是不能嵌套的，必须是处理完一个中断后再响应和处理同一级内其他中断源。

③ 设置多级中断的系统一般都希望有较快的中断响应时间，因此首先响应哪一级中断和哪一个中断源，都是由硬件逻辑实现，而不是用程序实现。

另外，在二维中断结构中，除了有中断优先级排队电路确定优先响应中断级外，还要确定优先响应的中断源，一般通过链式查询的硬件逻辑来实现。显然，这里采用了独立请求方式与链式查询方式相结合的方法决定首先响应哪个中断源。

④ 和单级中断情况类似，在多级中断中也使用中断堆栈保存现场信息。使用堆栈保存现场的好处是：

a. 控制逻辑简单，保存和恢复现场的过程按先进后出顺序进行。

b. 每一级中断不必单独设置现场保护区，各级中断现场可按其顺序放在同一个栈里。

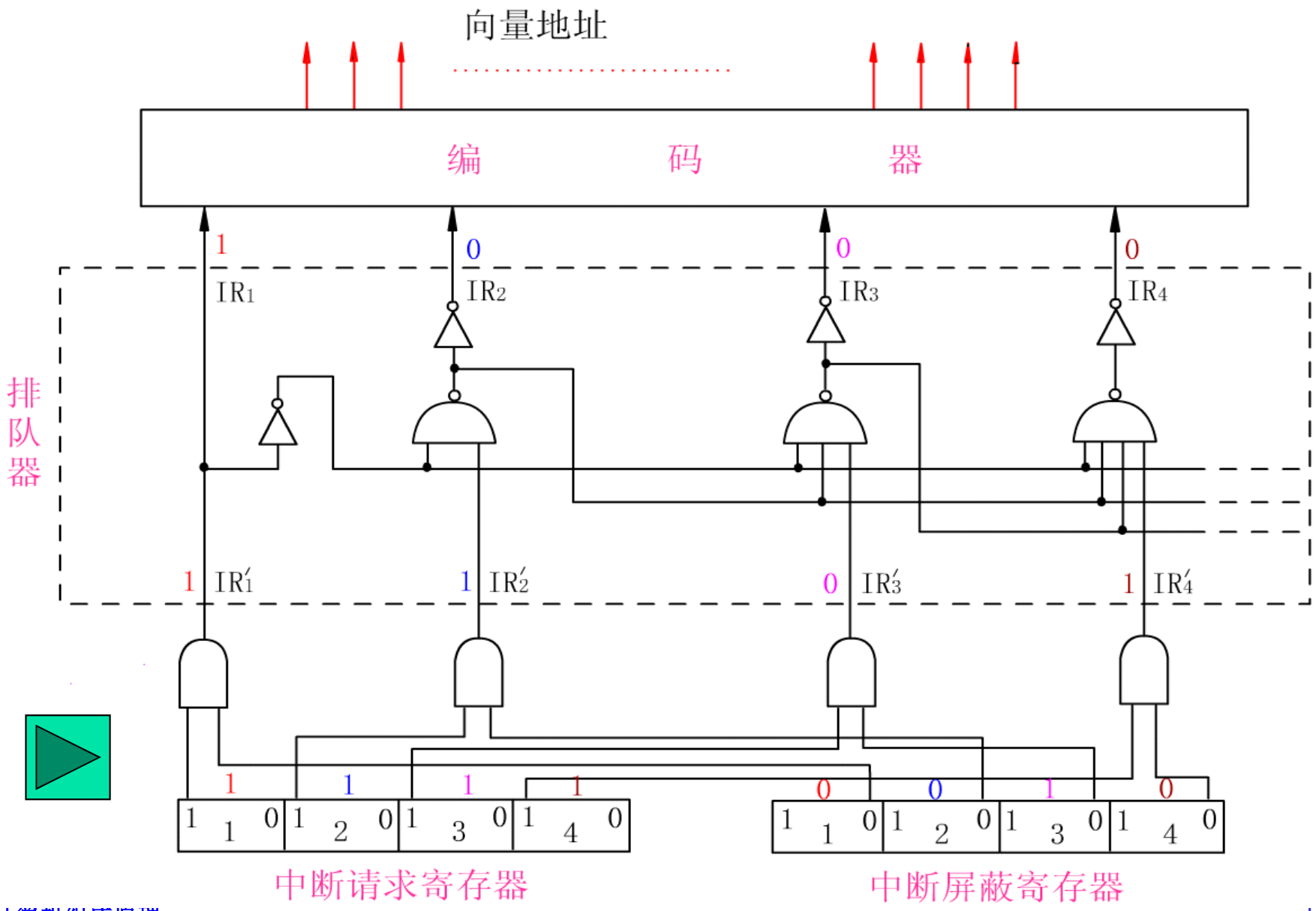
(3) 多级中断源的识别

- 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。显然这种结构就是独立请求方式的逻辑结构。

- 在多级中断中，如果每一级请求线上还连接有多个中断源设备，那么在识别中断源时，还需要进一步用串行链式方式查询。

  这意味着要用二维方式来设计中断排队逻辑。

独立请求方式的中断优先级排队与中断向量产生的逻辑结构



处理一个中断的过程，就是妥善处理以下一些基本问题的过程：

1）何时检查中断输入信号及其处理办法。

2）如何把控制转给中断服务程序。

3）如何保护和恢复中断的现场。

4）如何识别中断源。

5）如何识别优先级较高的中断。

6）如何开放和关闭中断。

中断方式的特点：

1、CPU与外设能并行工作

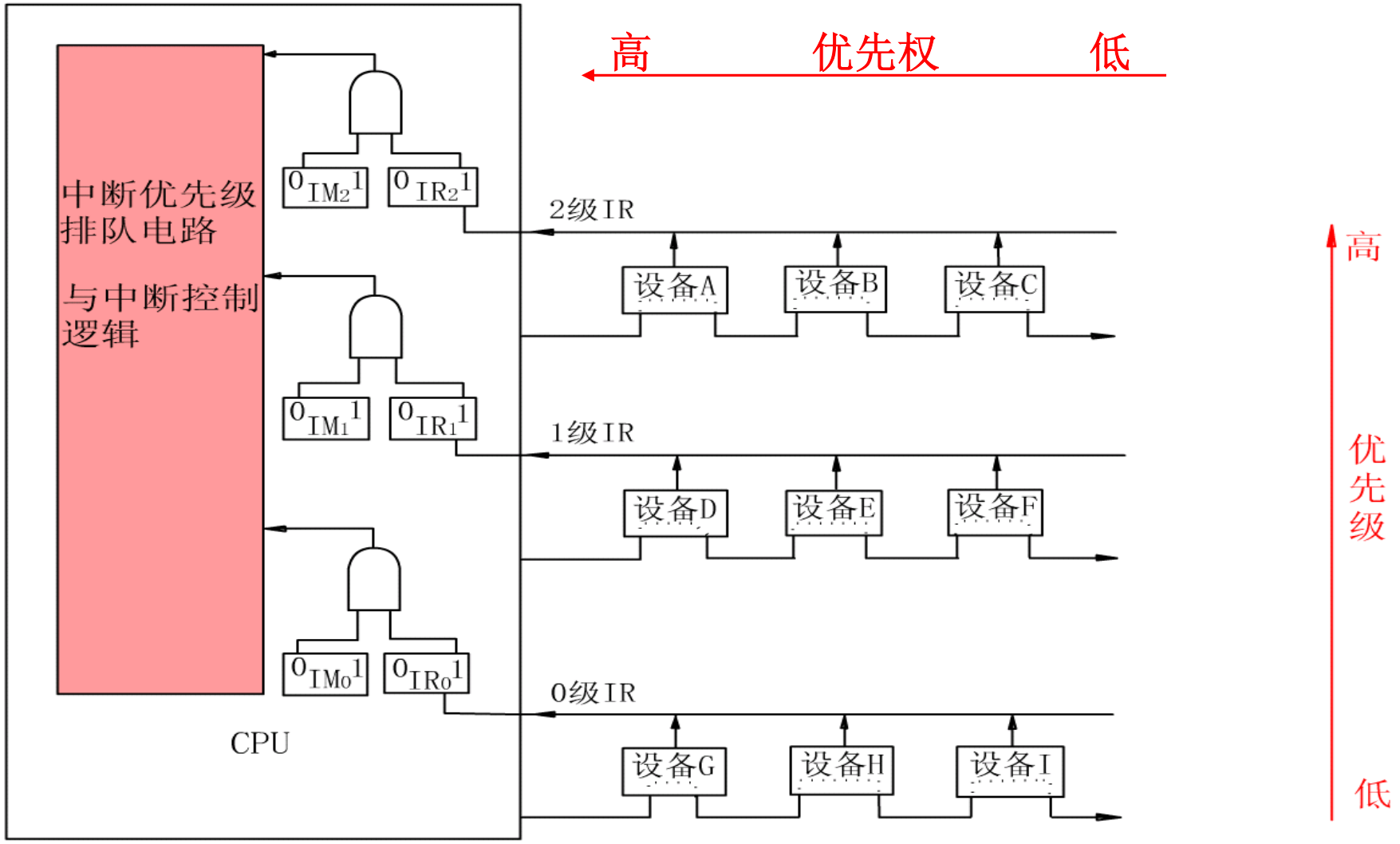
2、能处理异常事件

3、I/O操作仍然经过CPU，在程序控制下完成I/O

4、实时性好

5、CPU利用率仍然不太好

例：多级中断系统的连接方式如下图所示：



请问：

(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?

解：

(1)在中断情况下，CPU的优先级最低。各设备的优先次序是：

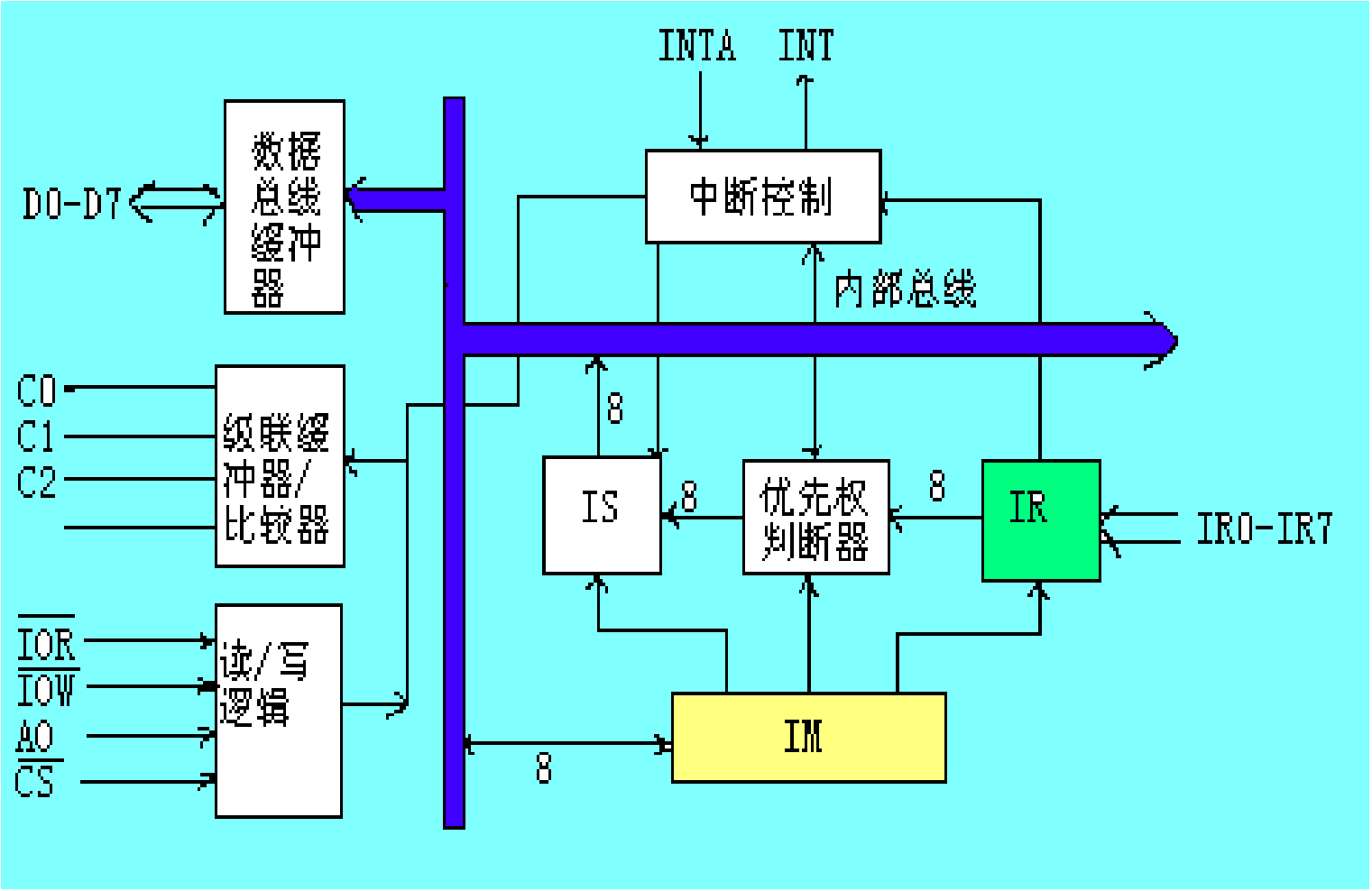
A→B→C→ D→E→F→G→H→I→CPU。

(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。

6. 中断控制器(8259 -简介)



8259的中断优先级选择方式有四种：

(1)完全嵌套方式：是一种固定优先级方式，连至**IR0**设备优先级最高，**IR7**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。

8259提供了两种屏蔽方式：

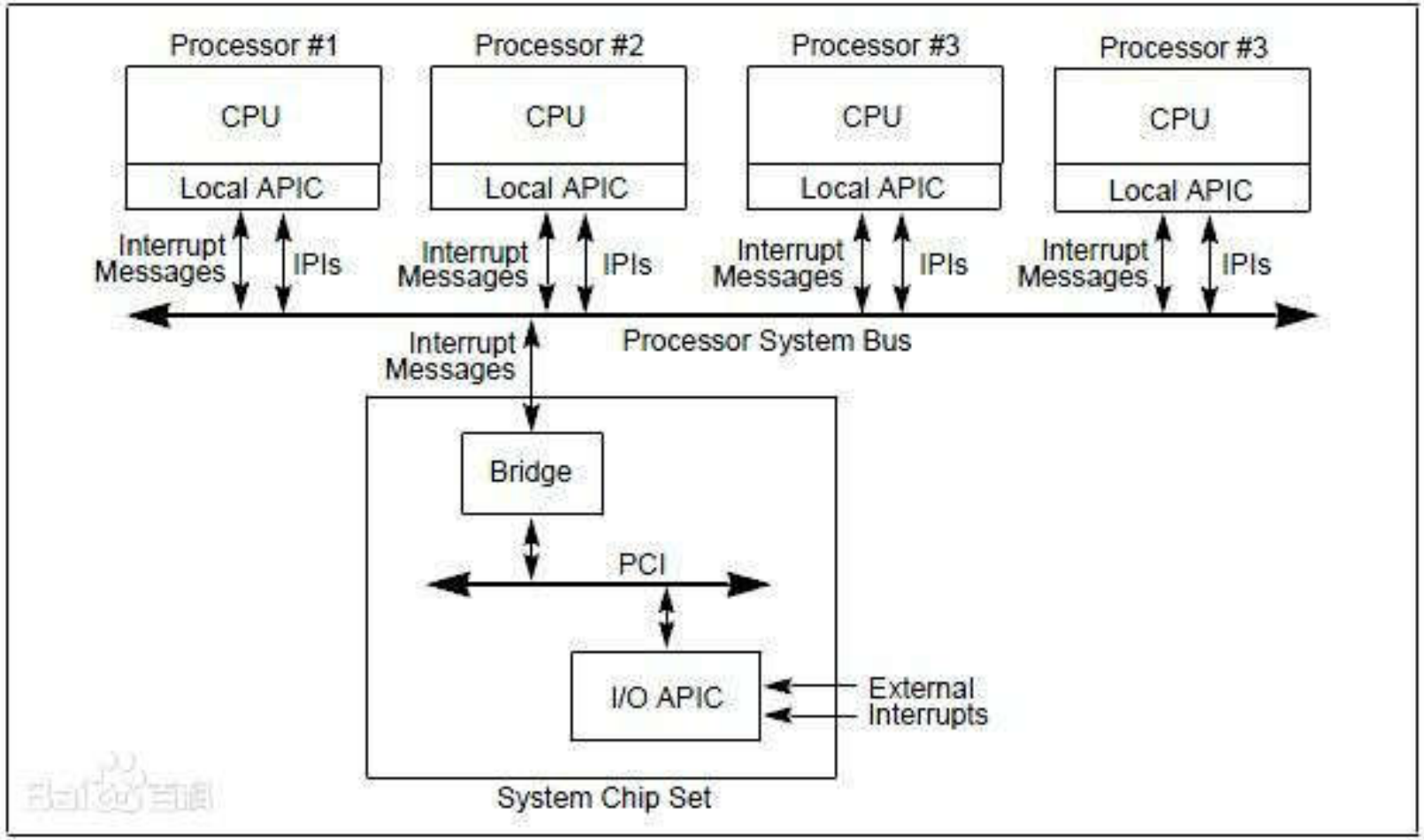
(1)简单屏蔽方式：提供8位屏蔽字，每位对应着各自的IR线。被置位的任一位则禁止了对应IR线上的中断。

(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。当8位屏蔽位的某位置“0”时，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中断控制器的不同工作方式是通过编程来实现的。CPU送出一系列的初始化控制字和操作控制字来执行选定的操作。

对称多处理机系统中

南桥芯片中的APIC 取代了8259A



7. 奔腾中断机制

(1) 中断类型

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：

①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。

②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。

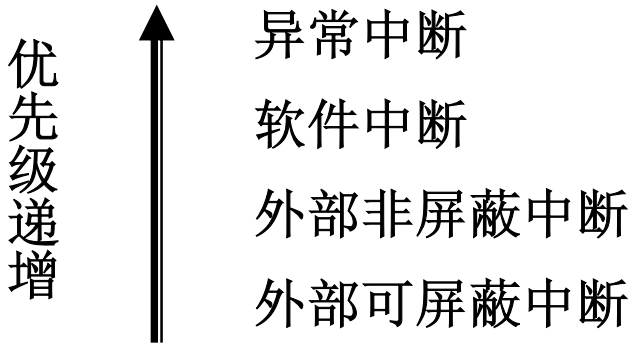
异常 通常称为异常中断，它是由指令执行引发的。有两种情况：

①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；

②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。



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)，以便发生中断时，程序转向相应的中断服务子程序入口地址。

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。

(2) 中断服务子程序进入过程

中断服务子程序的入口地址信息存于中断向量号检索表内。实模式为中断向量表IVT，保护模式为中断描述符表IDT。

CPU识别中断类型取得中断向量号的途径有三种：

①指令给出：如软件中断指令INT n 中的n即为中断向量号。

②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。

③CPU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。

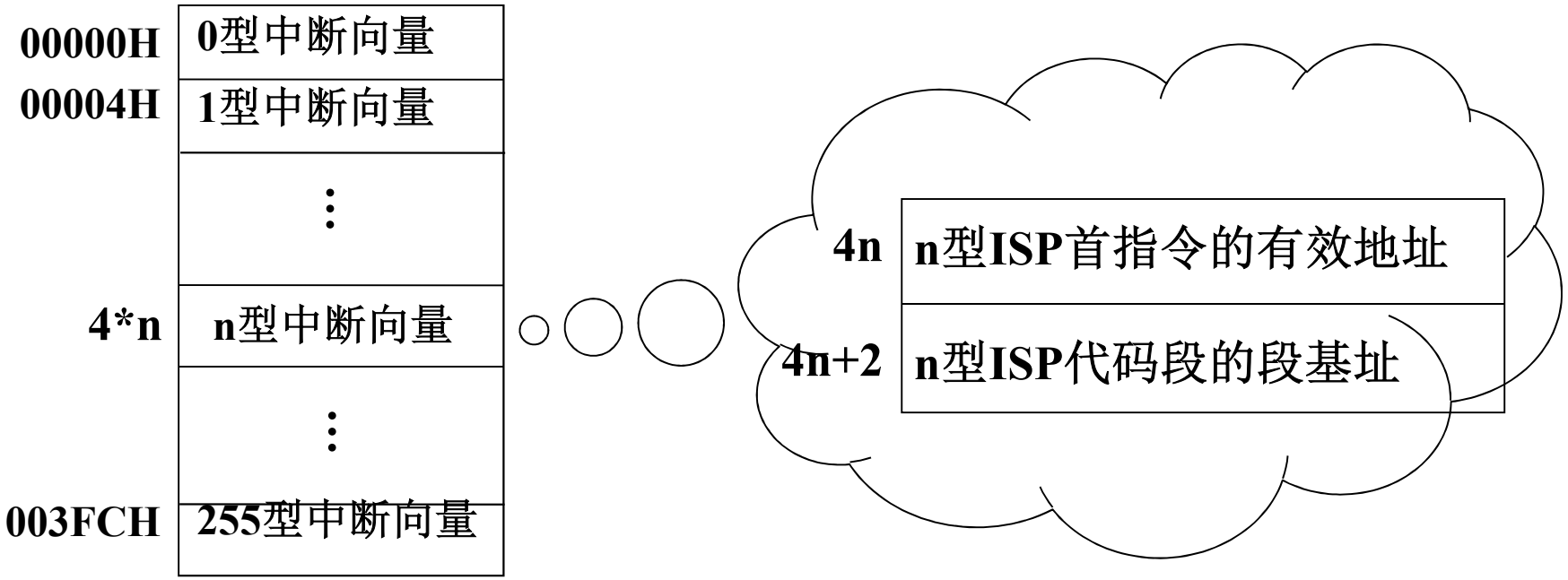
CPU依据中断向量号获取中断服务子程序入口地址，但在实模式下和保护模式下采用不同的途径:

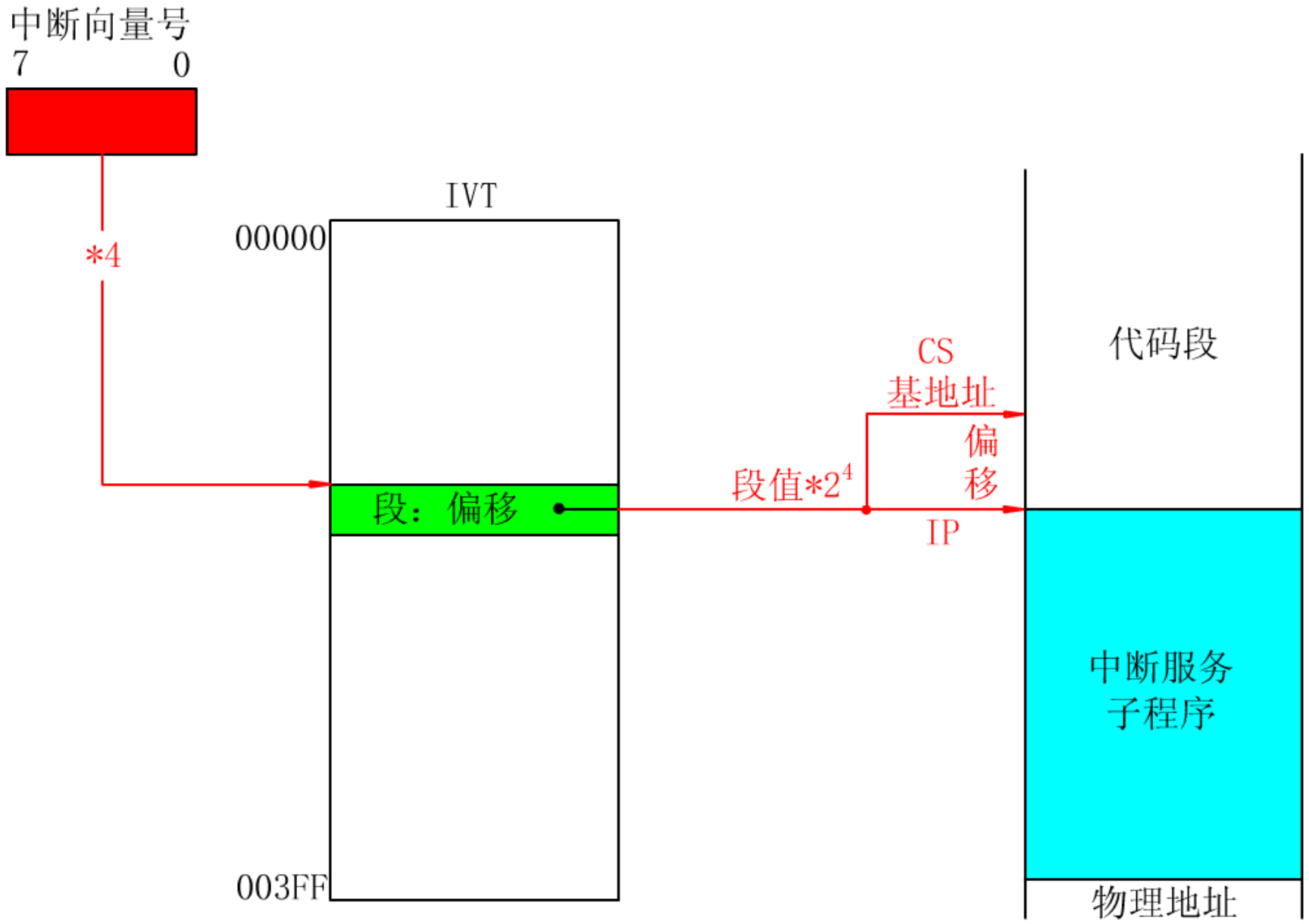
实模式下：

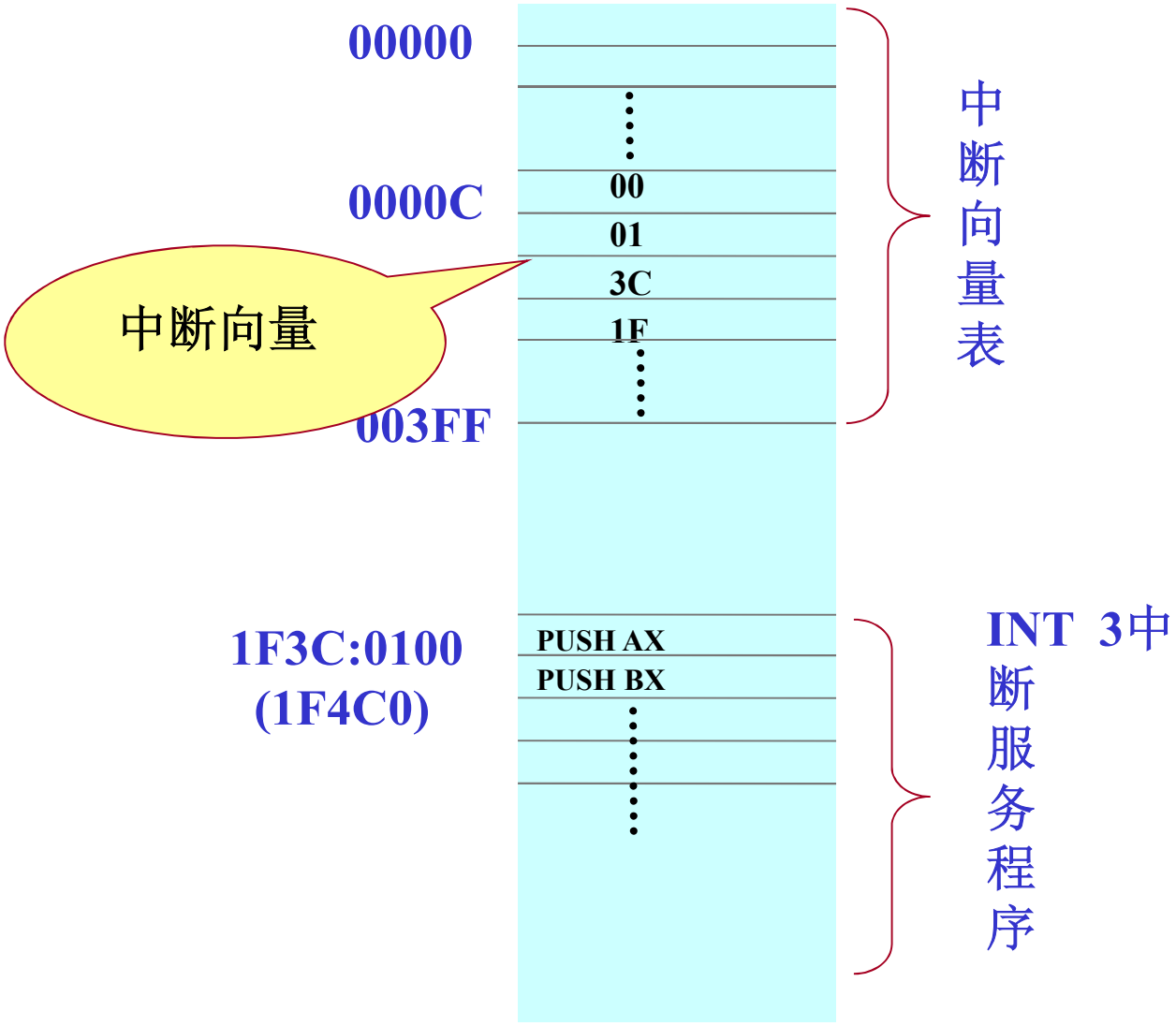
使用中断向量表

中断向量表IVT位于内存地址0开始的1KB空间。

实模式是16位寻址，中断服务子程序入口地址(段，偏移)的段寄存器和段内偏移量各为16位。


00000H
0型中断向量
00004H
1型中断向量
4*n
n型中断向量
003FCH
255型中断向量
4n
n型ISP首指令的有效地址
4n+2
n型ISP代码段的段基址


中断向量号
7
0
*4
IVT
00000
003FF
段：偏移
段值*2^4
CS
基地址
偏
移
IP
代码段
中断服务
子程序
物理地址


00000
0000C
003FF
00
01
3C
1F
中断向量
中断向量表
1F3C:0100
(1F4C0)
PUSH AX
PUSH BX
INT 3中断服务程序

保护模式下

使用中断描述符表

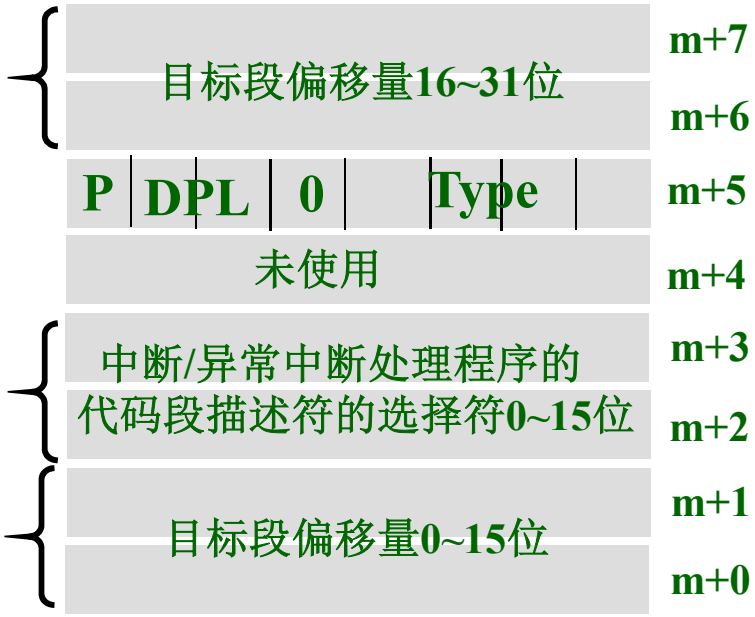
概述

- 保护模式下采用中断描述符表IDT管理各级中断；
- IDT中最多可以有256个描述符，对应于256个中断/异常源；
- IDT表中的描述符包括了中断服务程序的入口地址信息；
- 这些门描述符为8字节长，对应256个中断向量号，IDT表长为2KB。

IDT可置于内存的任意区域，其起始地址由中断描述符表寄存器(IDTR)设置；

- 中断门和陷阱门必须设在IDT表中，中断门对应外部硬件中断，陷阱门对应内部软件中断或异常。

中断门、陷阱门描述符



访问权限字节

P位： P=0，该段不在内存中

P=1，该段在内存中

DPL： 取值0~3，确定段的特权级

00为0级；

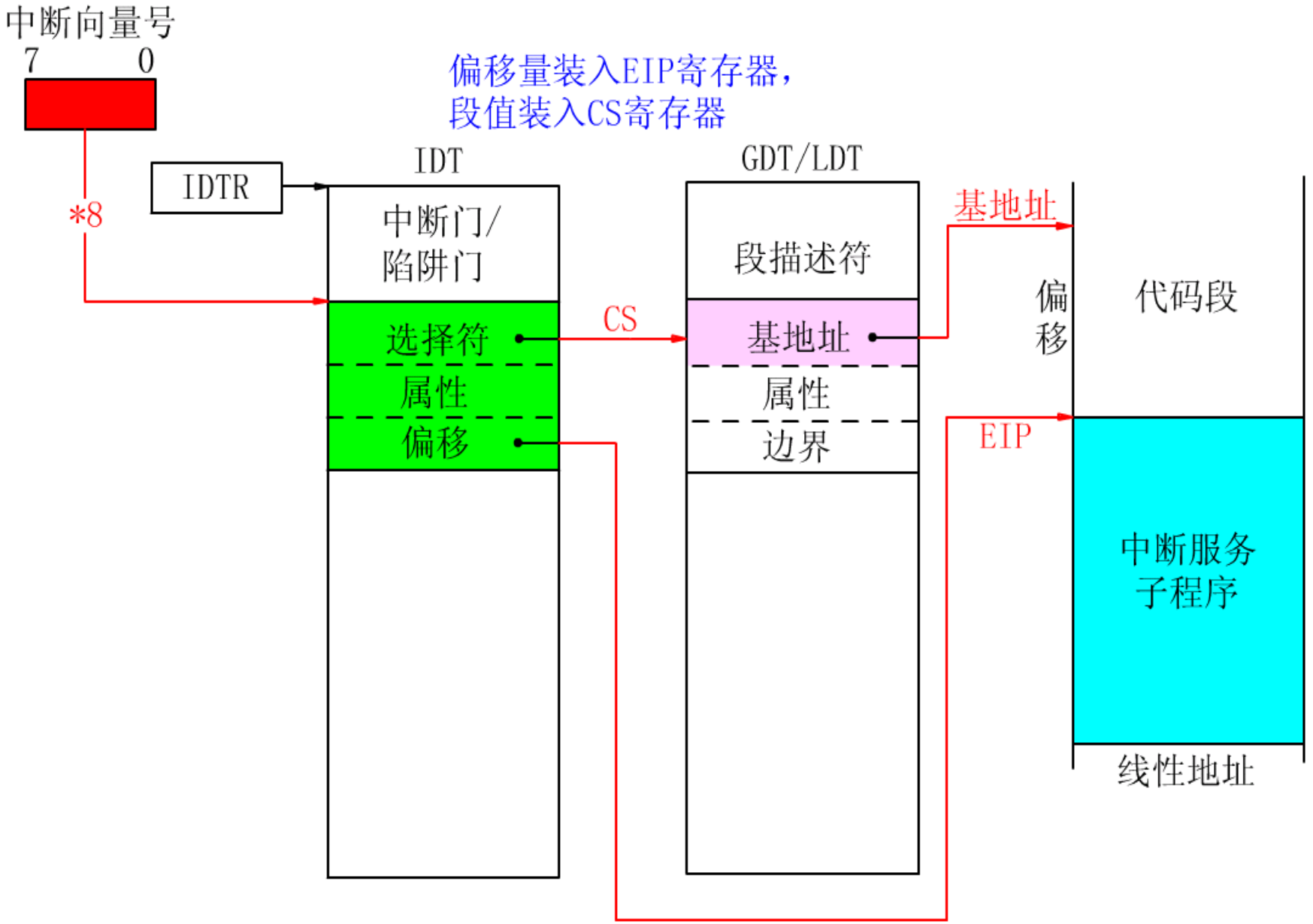
01为1级；

10为2级；

11位3级。

Type : 1110 —中断门；

1111— 陷阱门。


中断向量号
7
0
偏移量装入EIP寄存器，
段值装入CS寄存器
IDT
GDT/LDT
IDTR
*8
中断门/
陷阱门
选择符
属性
偏移
CS
段描述符
基地址
属性
边界
基地址
偏
移
代码段
EIP
中断服务
子程序
线性地址

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。

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。

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，并由中断点继续执行原程序。

DMA方式

1. DMA的基本概念

直接内存访问(DMA)是一种完全由硬件执行数据(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 。

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。

DMA能执行以下一些基本操作：

(1) 从外围设备发出DMA请求；

(2) 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；

(3) 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；

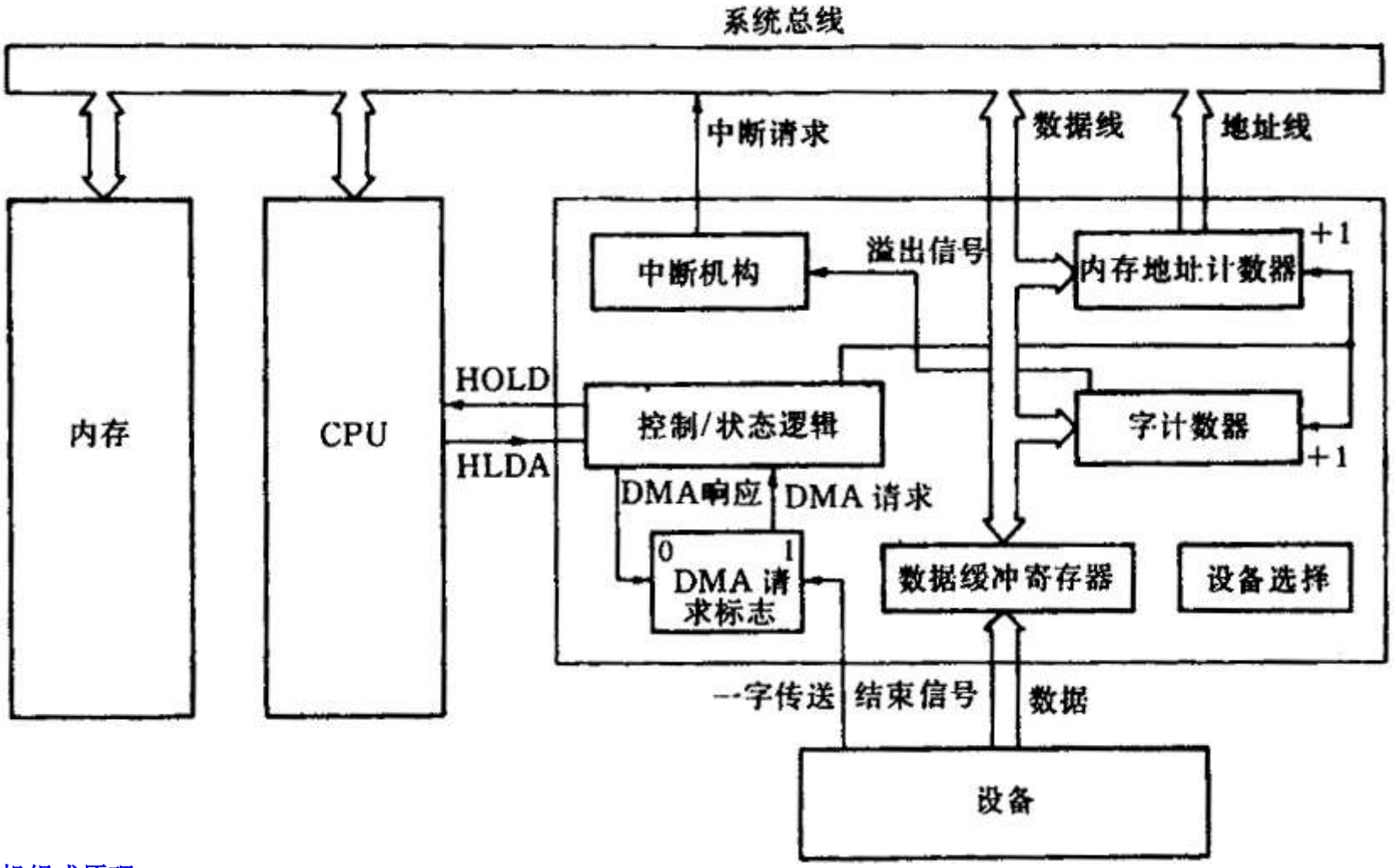
(4) 向CPU报告DMA操作的结束。

注意

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。

2. 基本的DMA控制器

(1) DMA控制器的基本组成



DMA控制器中的6个主要部件

（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)。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。

（3）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。反之，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。

（4）“DMA请求”标志

每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，“控制/状态”逻辑接收此信号后发出DMA响应信号，使“DMA请求”标志复位，为交换下一个字做好准备。

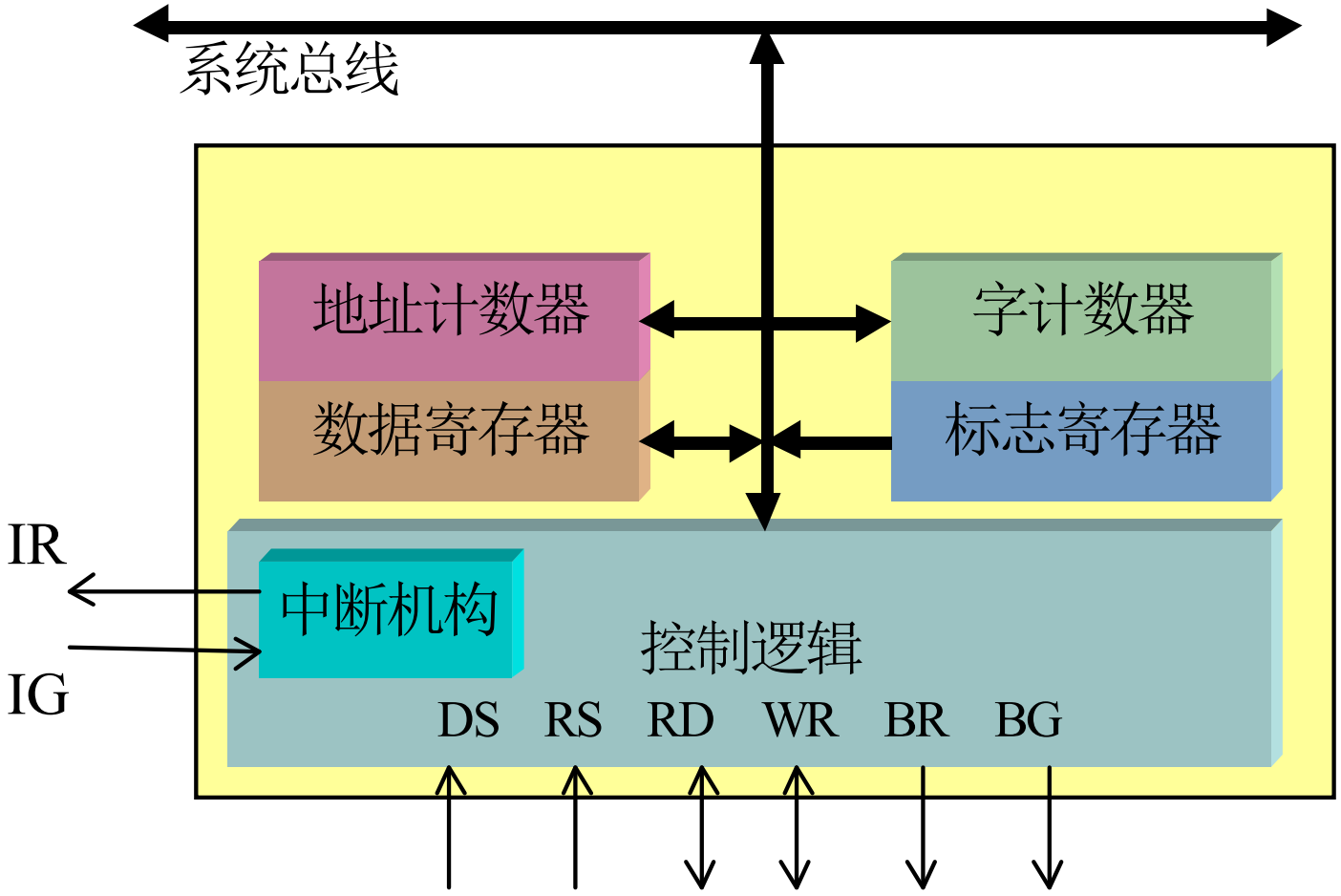
（5）“控制/状态”逻辑

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。

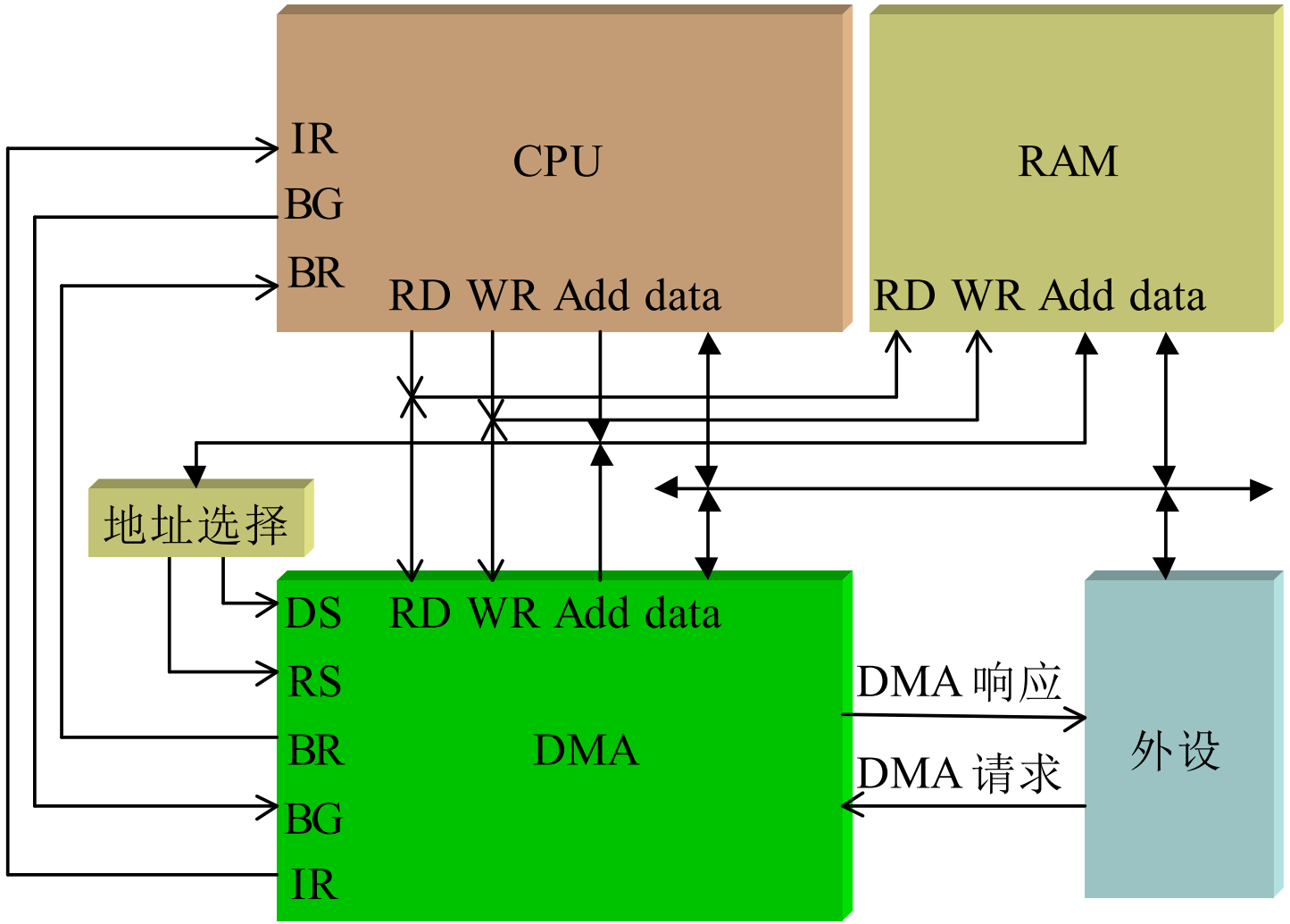
（6）中断机构

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。这里的中断与上一节介绍的I/O中断所采用的技术相同，但中断的目的不同，前面是为了数据的输入或输出，而这里是为了报告一组数据传送结束。因此它们是I/O系统中不同的中断事件。

DMA控制器结构

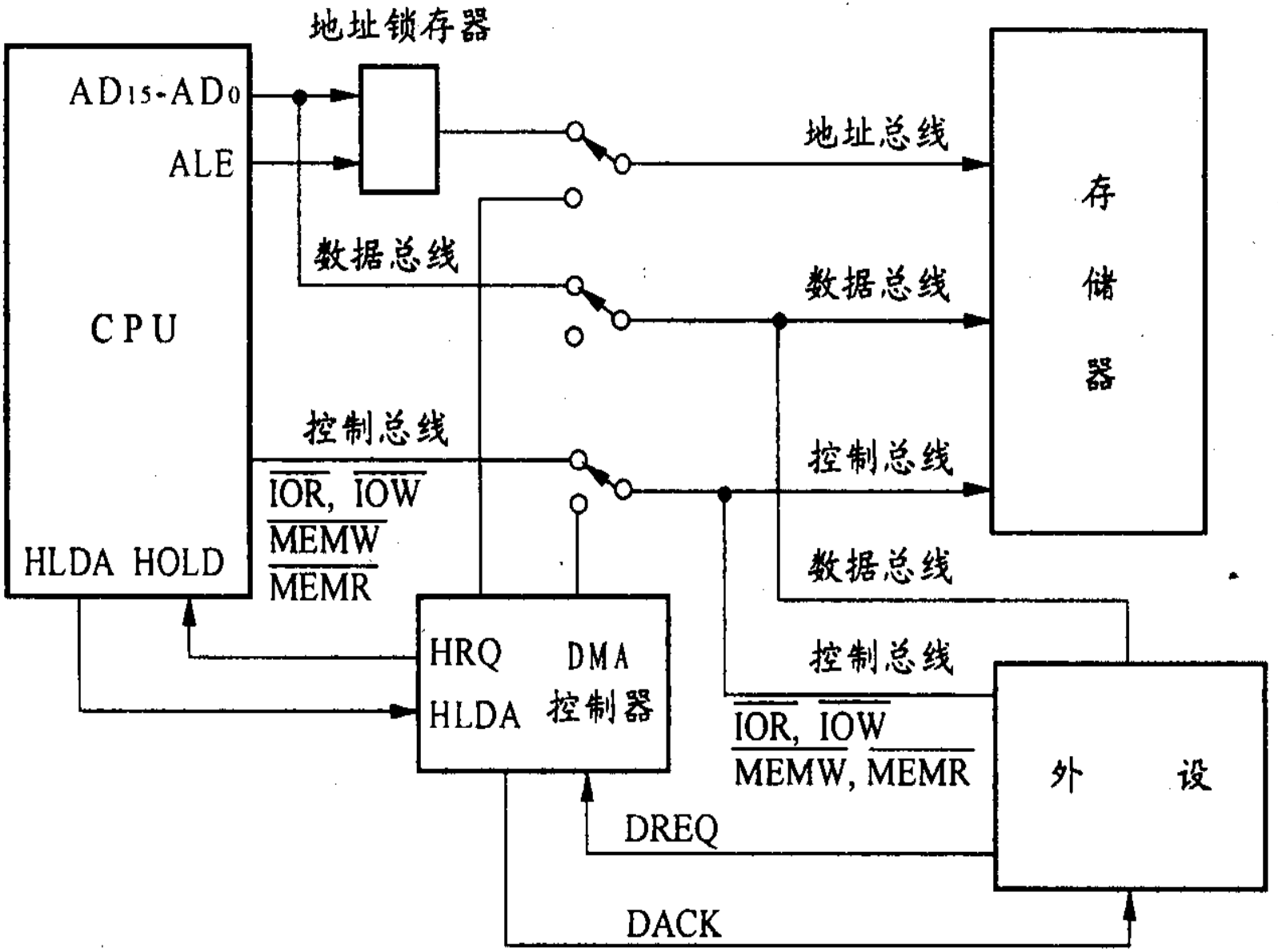


DMA与CPU的连接


IR
BG
BR
CPU
RD WR Add data
RAM
RD WR Add data
地址选择
DS
RS
BR
BG
IR
RD WR Add data
DMA
DMA 响应
DMA 请求
外设

8237A内部寄存器的功能、端口地址等信息

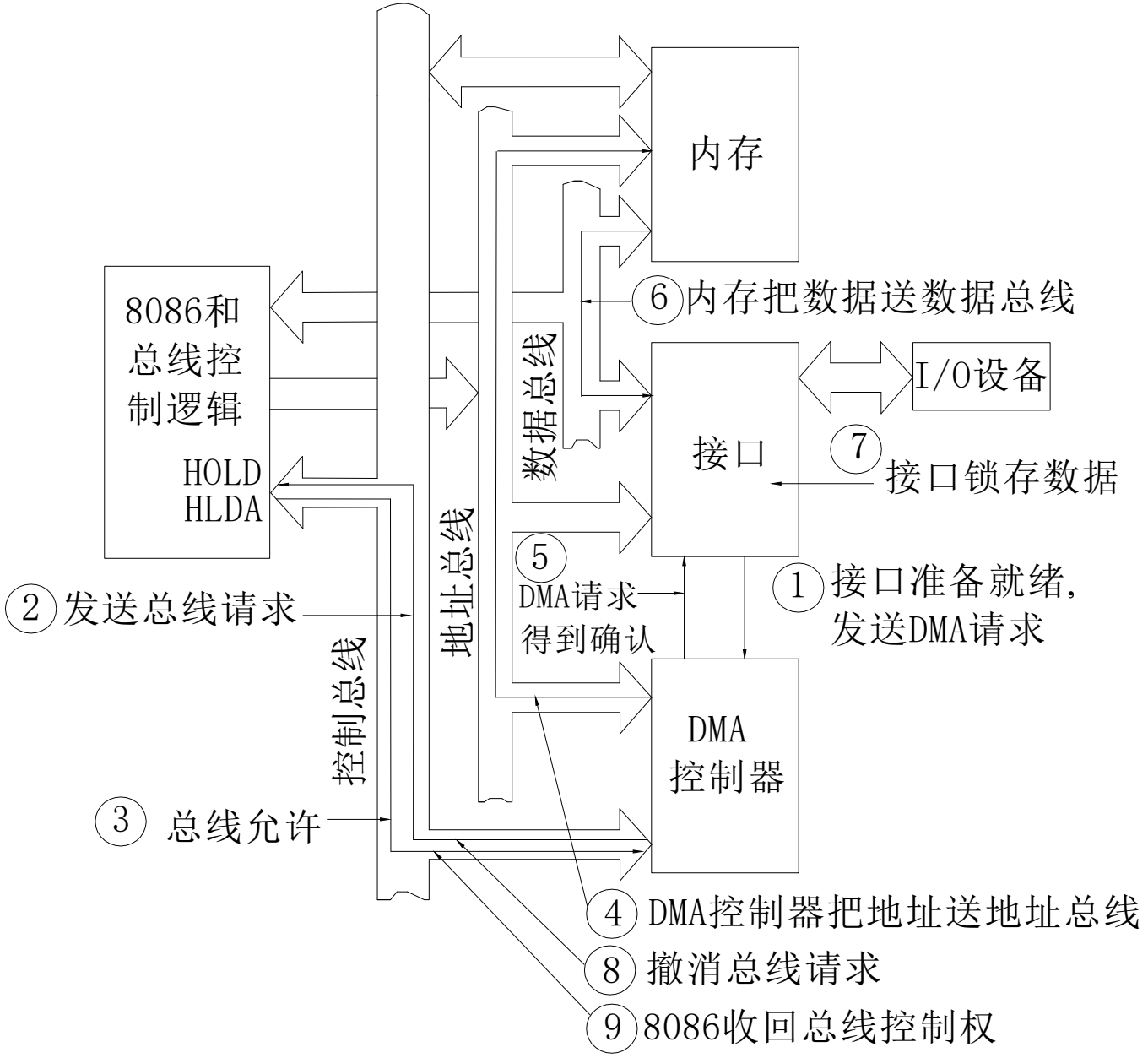
寄存器名称	功能	位数	数量	所属	CPU 访问方式	端口地址低 4 位
控制寄存器	存放命令字	8	1	四通道共用	只写	1000
状态寄存器	存放状态字	8	1	四通道共用	只读	1000
工作模式寄存器	存放模式字	8	4	每通道一个	只写	1011
基地址寄存器	存放存贮器起始地址	16	4	每通道一个	只写	0000~0111 中的偶地址
当前地址寄存器	存放存贮器当前地址	16	4	每通道一个	可读/写	0000~0111 中的偶地址
基字节数计数器	存放传送字节总数	16	4	每通道一个	只写	0000~0111 中的奇地址
当前字节计数器	存放尚未传送的字节总数	16	4	每通道一个	可读/写	0000~0111 中的偶地址
请求触发器	设置 DMA 请求标志	1	4	每通道一个	只写	1001
屏蔽触发器	设置通道屏蔽标志	1	4	每通道一个	只写	1010 或 1111
暂存寄存器	用于两存贮区之间的传送	8	1	四通道共用	只读	1101


地址锁存器
AD15-AD0
ALE
CPU
HLDA HOLD
数据总线
控制总线
IOR, IOW
MEMW
MEMR
HRQ
HLDA
DMA
控制器
地址总线
数据总线
控制总线
存
储
器
数据总线
控制总线
IOR, IOW
MEMW, MEMR
外
设
DREQ
DACK

当某个外围设备请求DMA服务时，操作过程如下：

1. DMA控制器接到设备发出的DMA请求时，将请求转送到CPU。
2. CPU在适当的时刻响应DMA请求。若CPU不需要占用总线则继续执行指令；若CPU需要占用总线，则CPU进入等待状态。
3. DMA控制器接到CPU的响应信号后，进行以下工作：
   - ① 对现有DMA请求中优先权最高的请求给予DMA响应；
   - ② 选择相应的地址寄存器的内容驱动地址总线；
   - ③ 根据所选设备操作寄存器的内容，向总线发读、写信号；
   - ④ 外围设备向数据总线传送数据，或从数据总线接收数据；
   - ⑤ 每个字节传送完毕后，DMA控制器使相应的地址寄存器和长度寄存器加“1”或减“1”。

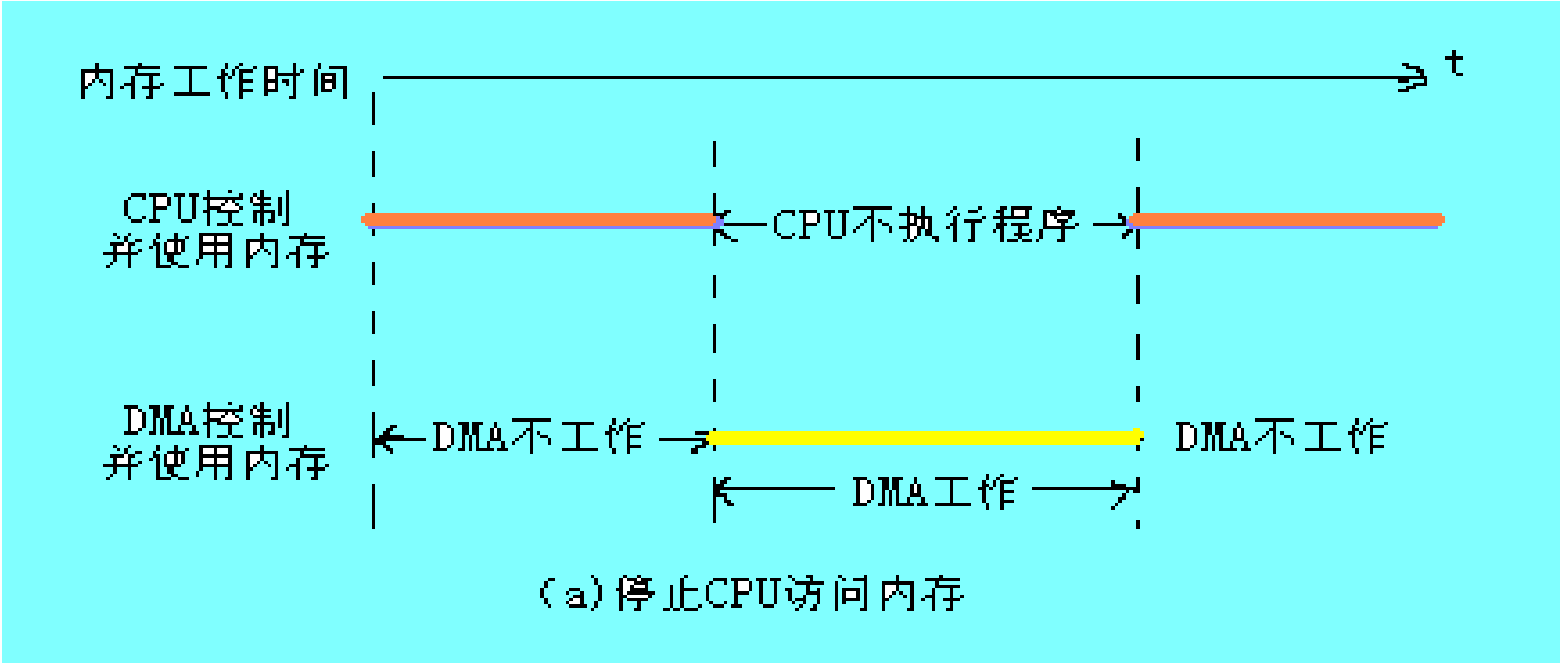
以DMA方式输出一个字节数据的工作过程



3. DMA数据传送模式（3种）

(1)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。在这种DMA传送过程中，CPU基本处于不工作状态或者说保持状态。

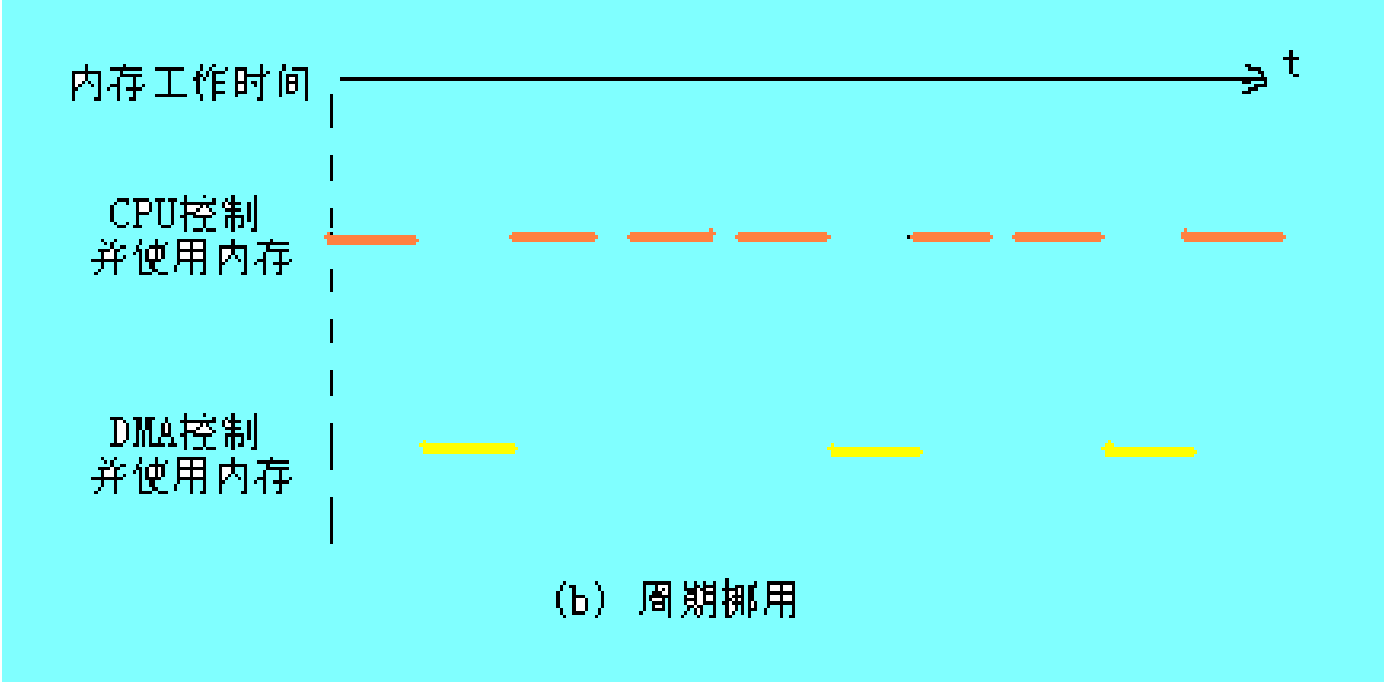


优点：控制简单，它适用于数据传输率很高的设备进行成组传送。

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。

(2) 周期挪用

当I/O设备没有DMA请求时，CPU按程序要求访问内存；一旦I/O设备有DMA请求，则由I/O设备挪用一个或几个内存周期。



(b) 周期挪用

I/O设备要求DMA传送时可能遇到两种情况：

① 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。

② 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。显然，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，或者更明确地说，在CPU执行访内指令的过程中插入DMA请求，挪用了一二个内存周期。

优缺点：

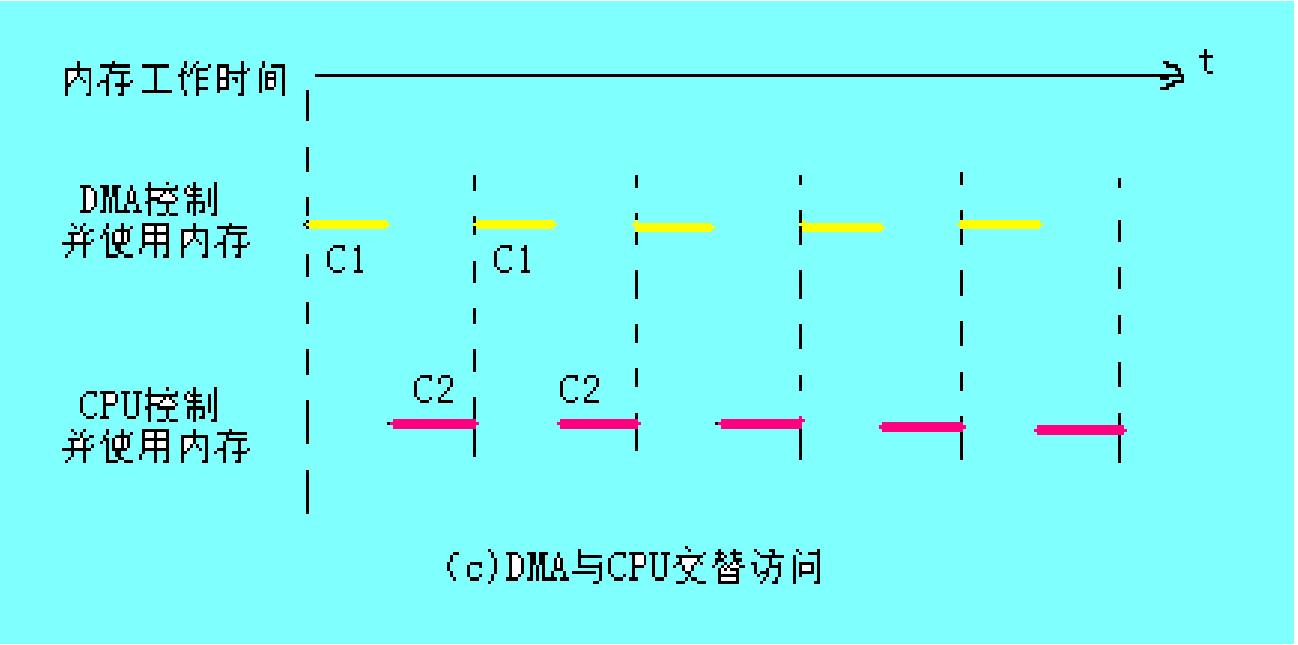
周期挪用的方法既实现了I/O传送，又较好地发挥了内存和CPU的效率，是一种广泛采用的方法。

但是I/O设备每一次周期挪用都有申请总线控制权、建立总线控制权和归还总线控制权的过程，所以传送一个字对内存来说要占用一个周期，但对DMA控制器来说一般要2—5个内存周期(视逻辑线路的延迟而定)。

因此，周期挪用的方法适用于I/O设备读写周期大于内存存储周期的情况。

(3)DMA与CPU交替访内（透明DMA方式）

如果CPU的工作周期比内存存取周期长很多，此时采用交替访内的方法可以使DMA传送和CPU同时发挥最高的效率。



(c)DMA与CPU交替访问

假设CPU工作周期为 1.2μs，内存存取周期小于0.6μs，那么一个CPU周期可分为C1和C2两个分周期，其中C1供DMA控制器访内，C2专供CPU访内。

优缺点：

这种方式不需要总线使用权的申请、建立和归还过程，总线使用权是通过C1和C2分时进行的。CPU和DMA控制器各自有自己的访内存地址寄存器、数据寄存器和读/写信号等控制寄存器。在C1周期中，如果DMA控制器有访内请求，可将地址、数据等信号送到总线上。在C2周期中，如CPU有访内请求，同样传送地址、数据等信号。

事实上，对于总线，这是用C1，C2控制的一个多路转换器，这种总线控制权的转移几乎不需要什么时间，所以对DMA传送来讲效率是很高的。

硬件逻辑复杂。

(2)DMA数据传送过程

DMA的数据块传送过程可分为三个阶段：

a)传送前预处理；　b)正式传送；　c) 传送后处理。

传送前处理通常由操作系统或者应用程序来处理。

传送后处理则由中断服务程序来处理（操作系统和应用程序设定）

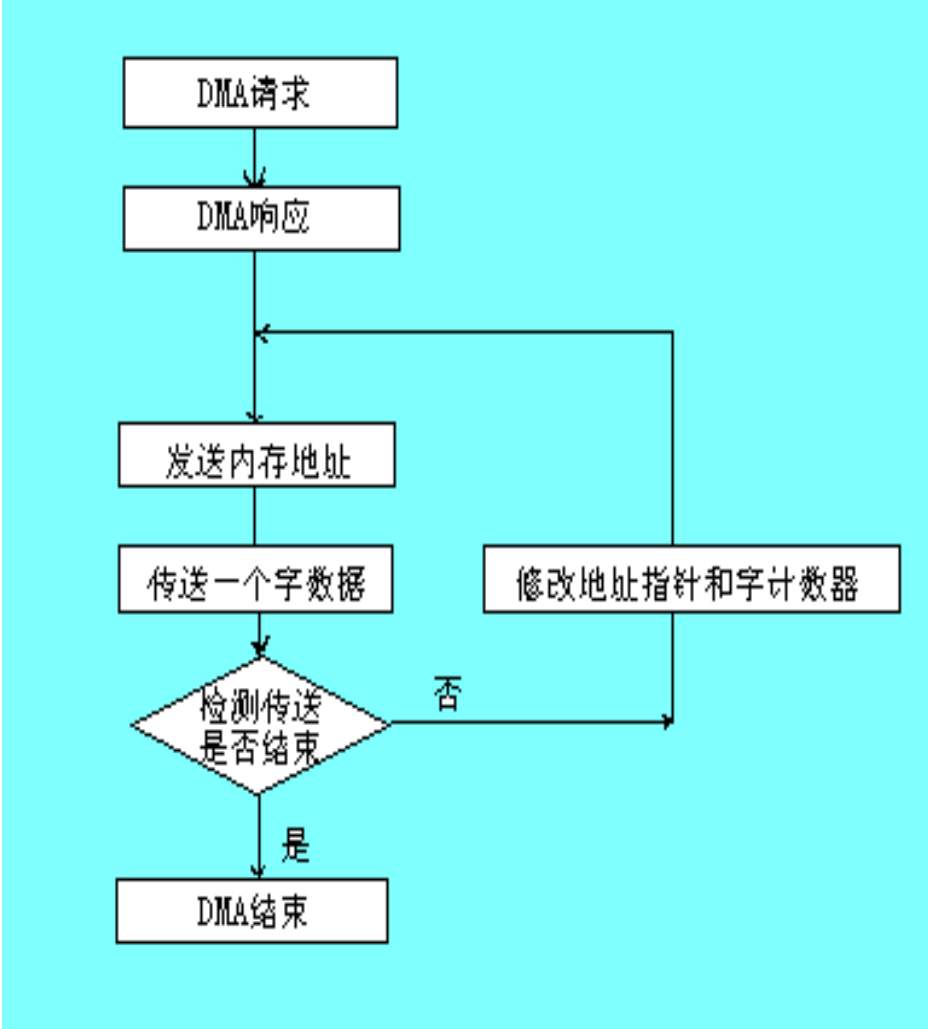
正式传送则由 DMA控制器（硬件）来完成

a)传送前预处理阶段

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个数。

在这些工作完成后，CPU继续执行原来的主程序。

b)正式传送阶段



停止CPU访内方式的DMA传送数据的流程图

c) 传送后处理阶段

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。

(3)DMA控制器与系统的连接

基本DMA控制器与系统的连接方式：

（1）公用的DMA请求方式；

（2）独立的DMA请求方式。

这与中断方式类似。

中断方式与DMA方式哪些不同？

1、实现方式不同：

中断主动通知，硬件、软件结合，信息交换在软件控制下完成

DMA中CPU不介入交换过程，不影响CPU的状态

2、CPU响应中断和DMA请求的时机不同：

中断：一条指令的最后一个机器周期最后一个T状态

DMA：CPU在每一个机器周期的最后一个T状态

3、二者优先权不同：

CPU首先检测有无DMA请求，再检测有无中断请求

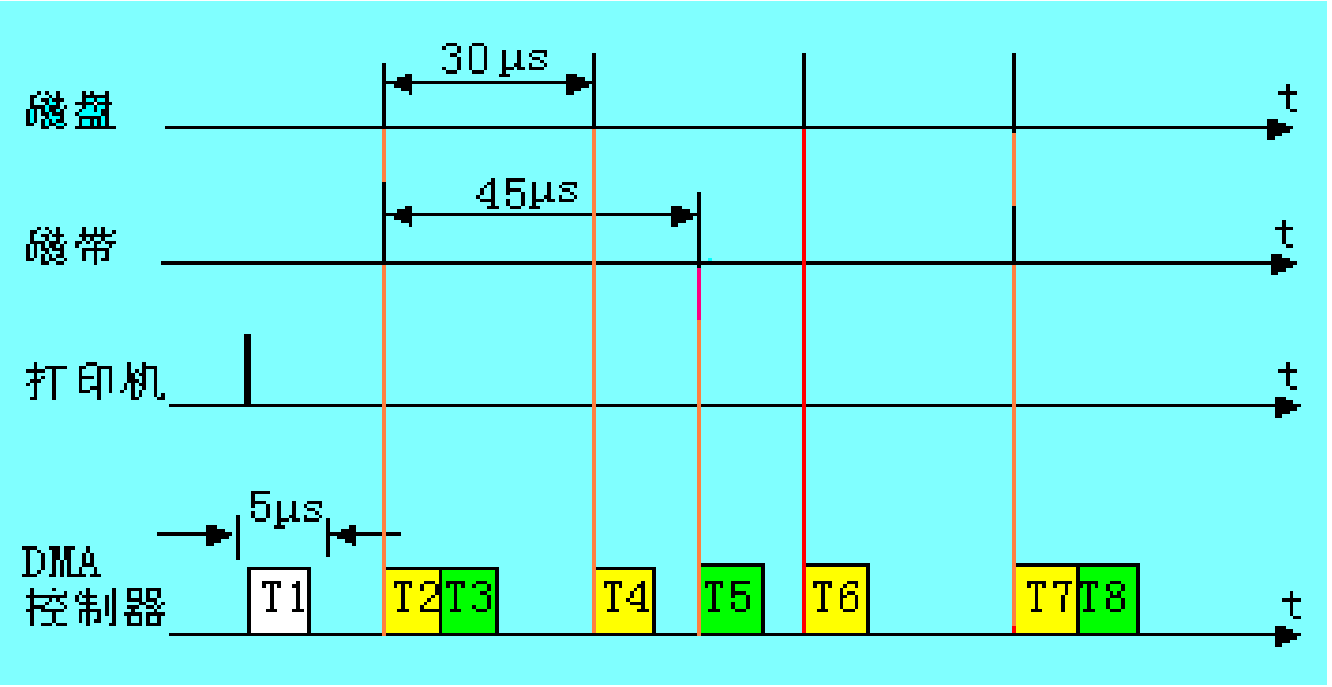
4、应用不同：

中断适用于随机出现的服务，具有处理异常事件的能力

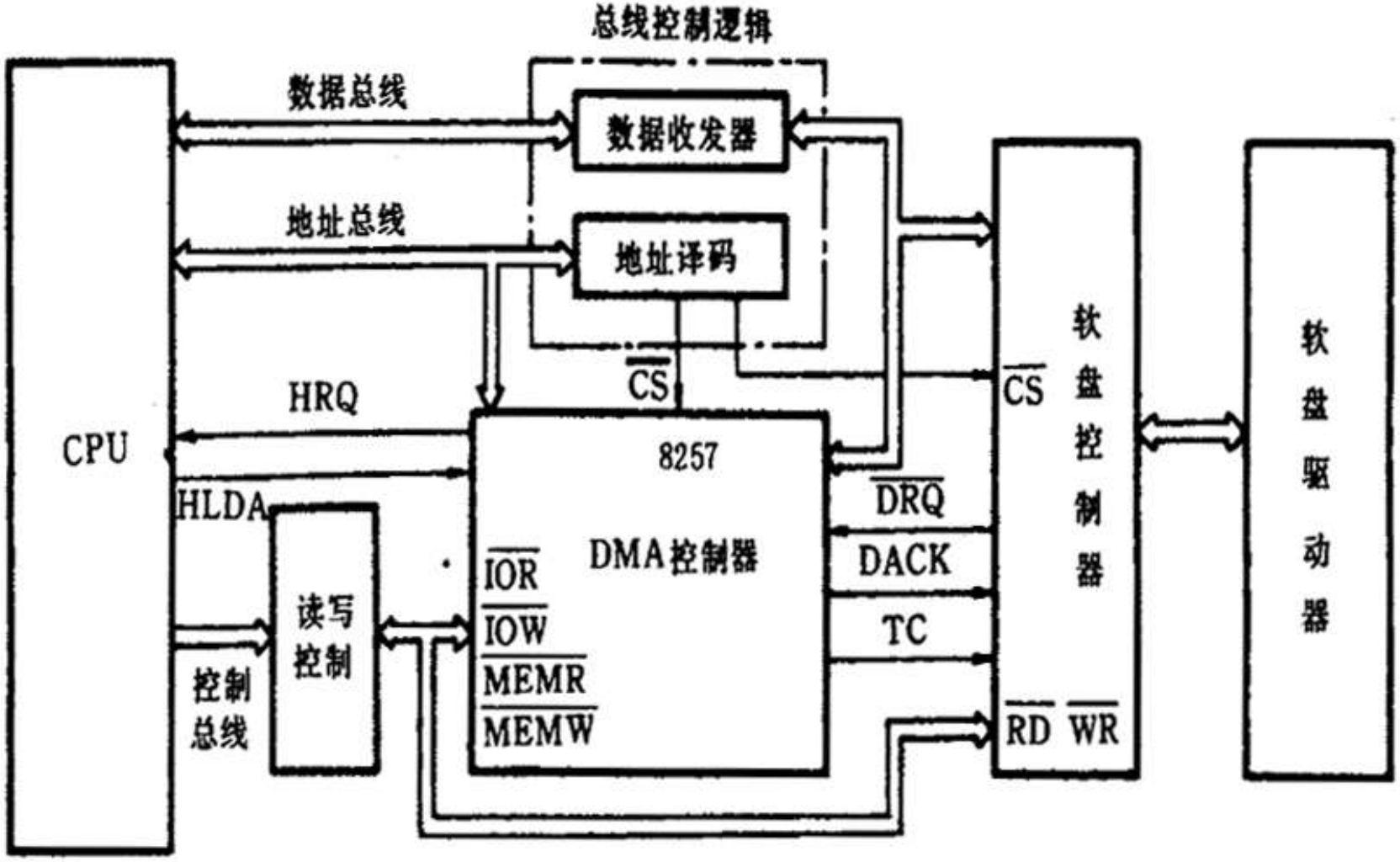
DMA适用于高速外设间成组数据交换

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30μs的间隔向控制器发DMA请求，磁带以45 μs的间隔发DMA请求，打印机以150 μs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 μs。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。

解：



例下图所示为微型机中软盘控制器的系统接口电路，请进行分析说明。



解：

CPU和软盘控制器之间的接口电路包括DMA控制和总线控制两部分。

- 8257DMA控制器提供4个独立的DMA通路(CH_0, CH_1, CH_2, CH3)。

  每个通路各有2个16位寄存器(DMA地址寄存器、字节计数寄存器)，它们必须在通路使用前加以预置：

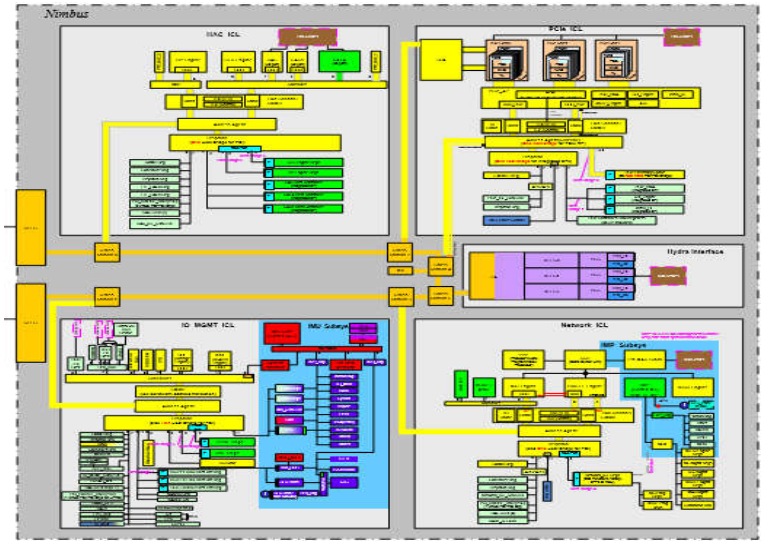
  DMA地址寄存器存放被寻址的主存首地址；

  字节计数寄存器存放本次DMA传送的字节数；

  此外还包含工作方式(读、写、校验)和状态寄存器。

芯片架构- IO子系统

- 丰富的IO，PCIE化设计。
- 各子系统 PCIE（含CCIX），Hydra（多片互联），Network，Storage，HAC，ME。各自遵循行业标准，兼容软件接口规范，满足开源及演进要求。



SBSA	GIC: GICv3/v4； SMMU： SMMUv3.1 UART/Watchdog:
PCI-E	36 lanes of PCIe G4.0, 20 Root ports at max, Peer2Peer, ATS/PRI, CCIX
Networking（NIC、ROCE) ROCE)	8 lanes of ETH, Combo MACs support 2x 100GE, 2x 50GE, 4x 25GE, 8x 10GE, 8x GE 8x 10GE, 8x GE RoCEv2/RoCEvl with programmable DC-QCN， long/normal Atomic, SR-IOV SR-IOV
Storage IO	x4 USB 3.0 EHCI/UHCI x16 SAS 3.0(STP supported) x2 SATA 3.0 AHCI
Crypto Engine	AES, DES/3DES, MD5,SHA1,SHA2,HMAC, CMAC Up to 50Gbps
Compression/ Decompression	GZIP, LZS, LZ4 Up to 40Gbps(compress)/100Gbps(decompress) Statefull and stateless
RAID	XOR/PQ/EC/in-line DIF acceleration
ME（IMU）	Isolated management subsystem. Co-works with BMC and provides firmware configuration of the server chip
Scale up	Coherent SMP interface for 2P/4P Up to 240Gbps per port

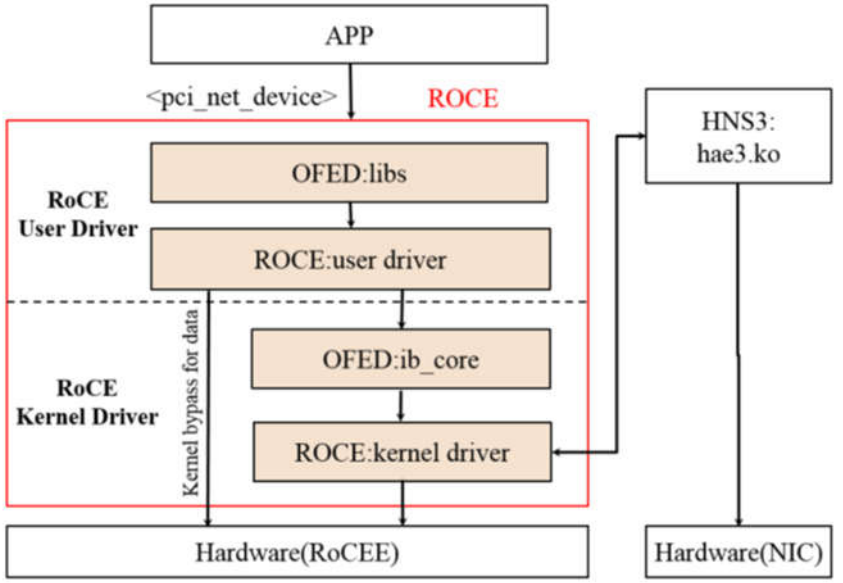
HUAWEI

鲲鹏920系列芯片架构(6)—网络子系统

- 网络子系统包括Network ICL和RoCE引擎两大部分。
- Network ICL包括多个1Gbps~100Gbps以太网控制器，二层DCB、MAC地址表，多播表，VLAN过滤表，流表，中断，PCIe化，具有完整的NIC引擎，可以在RoCE引擎关闭的情况下单独工作。
- RoCE(RDMA over Converge Ethernet)是一种在以太网上采用RDMA(Remote Direct Memory Access，远程直接内存访问)的网络互联技术。
- 华为鲲鹏920处理器使用的RoCE v2协议是由InfiniBand(IB)协议演进而来，既具有InfiniBand网络的低时延、低CPU利用率等特点，又能够很好地兼容于Ethernet网络。

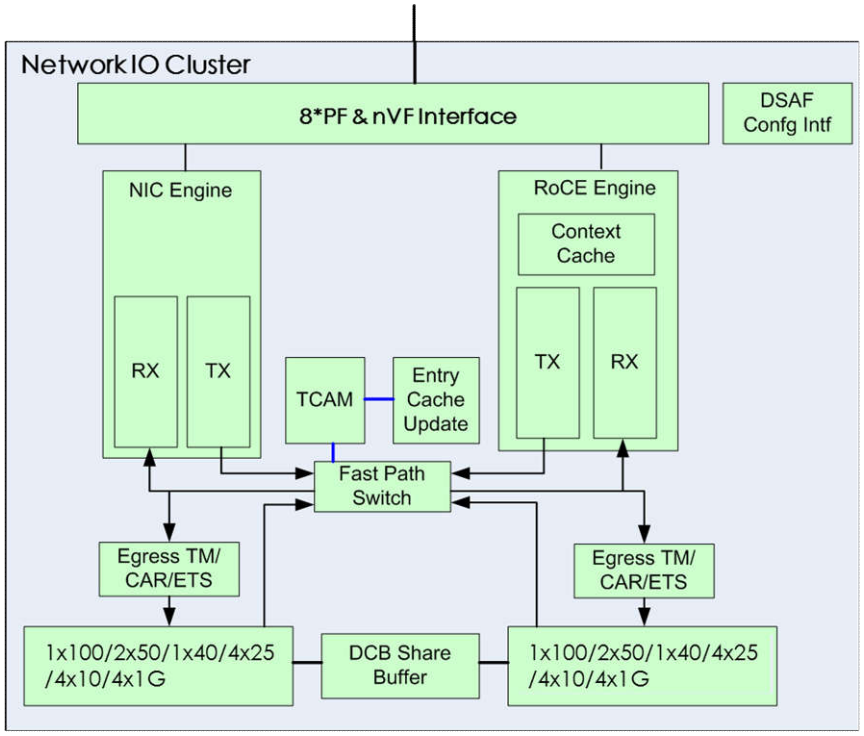
鲲鹏920系列芯片架构(6)—网络子系统

- 华为鲲鹏920处理器RoCE设备的软件呈现是一个PCI网络设备。
- RoCE驱动依赖于OFED的驱动框架，由用户态驱动和内核态驱动构成。
- 当业务建立后，在执行过程中，RoCE用户态驱动可以Kernel bypass将数据发给硬件。
- RoCE内核态驱动在初始化时从HNS3网卡驱动程序获取RoCE设备的一些信息。



HUAWEI

鲲鹏920系列芯片架构(6)—网络子系统

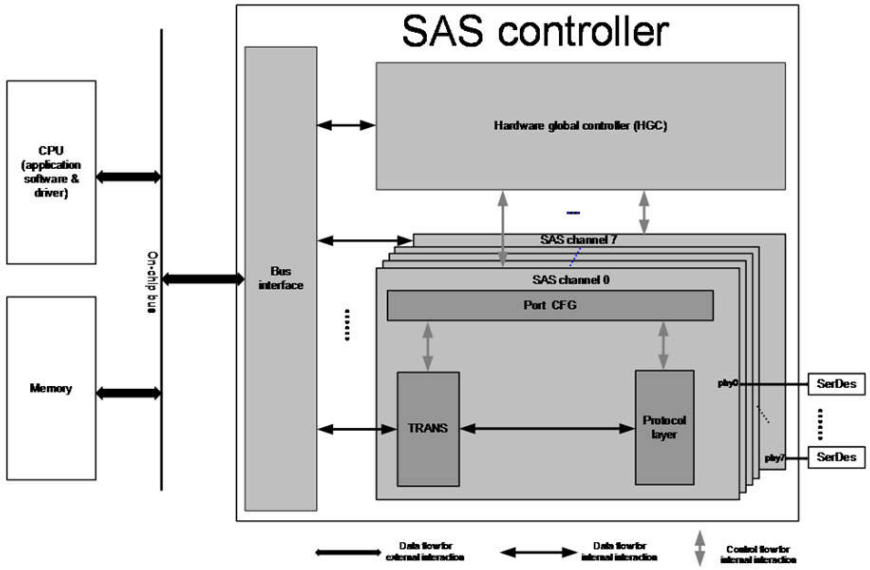


- Ethernet Physical Port Configuration
  - 2x100GE/50GE/40GE/25GE/10GE/GE + 2*GE
  - 4x50GE/25GE/10GE/GE +2*GE
  - 8x25GE/10GE/GE
- Speed Auto-Negotiation
  - MAC Speed can auto-negotiate between GE/XGE/25GE
- Support DCB(Data Center Bridge)
  - ETS(Enhanced Transmission Selection)
  - PFC(Priority-based Flow Control)
  - QCN(Quantized Congestion Notification)
- Support RoCEv1, RoCEv2
- Virtualization switch accelerator
  - generic flow table based switching
  - Support VEB, but not support EVB. [RFC]
- Shared resource between multiple Physical Ports
  - DCB Buffers
  - Queue Resource
  - Flow Table Entry

鲲鹏920系列芯片架构(7)—SAS子系统

- SAS(Serial Attached SCSI)即串行SCSI技术，一种磁盘连接技术。SAS控制器用于磁盘与内存之间进行交互。
- SAS控制器主要通过总线与CPU和内存进行交互，同时通过SERDES与硬盘进行连接。
- SAS控制器与设备连接方式有两种：直连和Expander连接。
  - 直连表示SAS控制器的PHY与设备直接连接，不经过中间转换或扩展；
  - Expander连接表示SAS控制器与设备之间通过扩展器进行连接
- SAS盘分为SAS机械盘和SAS SSD盘，SAS盘是为满足高性能、高可靠性而设计，在内部驱动电机的可靠性、转速以及基板方面都与SATA盘有差异。

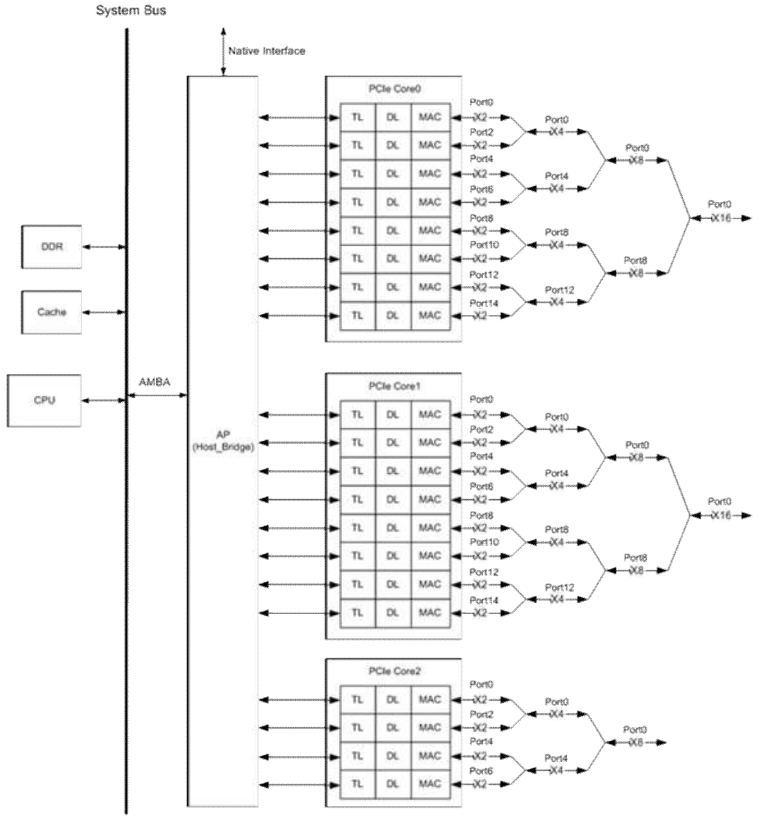
鲲鹏920系列芯片架构(7)—SAS子系统



- 提供2个X8 SAS 3.0控制器：
- 支持SAS 3.0，向下兼容SAS2.0和SAS1.0；
- 支持SATA3.0，向下兼容SATA2.0和SATA1.0；
- SAS支持12G/6G/3G/1.5G四种速率，SATA支持6G/3G/1.5G速率，同时可以实现速率的自协商；
- 可以直接不经过Expander最大连接8个SAS盘或者SATA盘，两者可以混插；
- 可以连接SAS Expander扩展更多磁盘。
- 提供1个X2 SATA控制器：
- 支持SATA 3.0，向下兼容SATA 2.5；
- 支持AHCI 1.3，向下兼容 AHCI 1.2；
- 支持6G/3G/1.5G速率自协商；
- 支持直连两个SATA盘。
- 支持NOR Flash控制器，4个片选，NOR FLASH最大支持512K。
- 支持SPI Flash控制器，2个片选，SPI Flash最大支持32M。
- 支持NAND FLASH接口，4个片选。

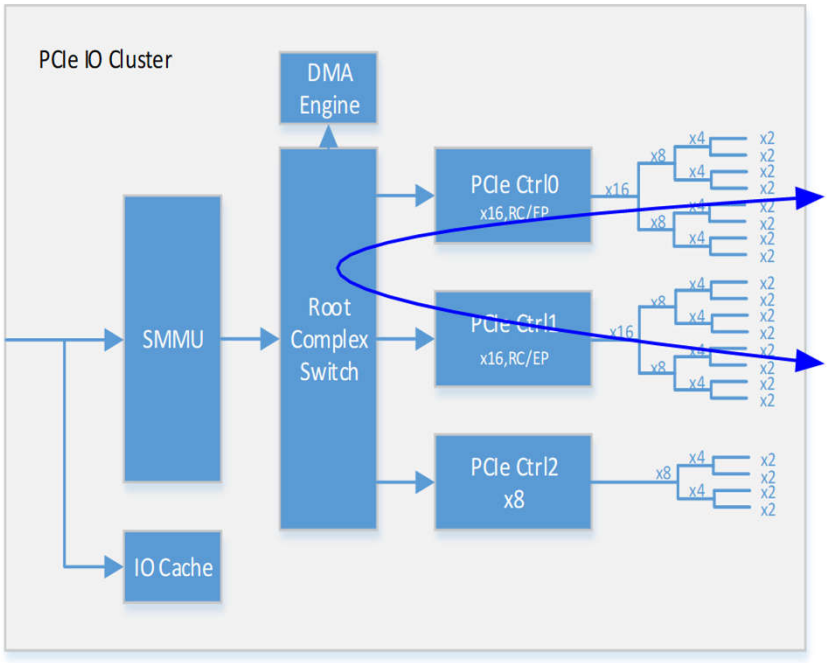
鲲鹏920系列芯片架构(8)—PCIe子系统

- PCIe是一种高性能、通用的I/O互连接口，适用于各种计算和通信平台。鲲鹏920 PCIe子系统提供了实现PCIe根联合体(Root Complex，RC)或端点(Endpoint，EP)应用程序的解决方案。
- 鲲鹏920 PCIe子系统包含3个PCIe Core，最多支持40个PCIe Lane。每个PCIe Core包括多个PCIe端口。PCIe Core0共享16个Lane。PCIe Core1共享16个Lane。PCIe Core2共享8个Lane。3个PCIe Core均可作为根端口(Root Port，RP)使用。只有PCIe Core1能作为EP端口。
- PCIe模块通过PIPe接口与PCS连接，连接速率支持最大16Gbps，兼容8Gbps、5Gbps和2.5Gbps。另一方面，PCIe模块通过AMBA总线与系统总线相连。



HUAWEI

鲲鹏920系列芯片架构(8)—PCIe子系统



- PCIe GEN1/2/3/4.0 Supported
  - Run at the 2.5G/5G/8G/16G
  - x16 PCIe Controller
  - Embedded DMA engine
- 40 Lanes support totally
  - 3 PCIe Controller
  - Support 20 Root Port
- Hardware features
  - SRIS(Separate Refclk Independent SSC)
- Support SR-IOV
- Support Shared Virtual Memory
- Support CCIX
- Support P2P(Peer to Peer)
  - Peer to Peer traffic between different controller

本章小结

1） 总线的概念、类型、多级总线、总线性能

2） 接口的作用和组成

3） I/O工作方式

4） 中断概念、响应的时间、工作过程、中断屏蔽、多级中断、优先级的处理

5） DMA的概念、工作过程、响应的时间

6） 通道的基本概念