

计算机组成 (2022秋)

计算机组成课程组
(刘旭东、高小鹏、肖利民、栾钟治、万寒)

北京航空航天大学计算机学院中德所
栾钟治

北京航空航天大学

➤1

习题7——Cache与虚存

- ❖ 已发布
 - Spoc平台
- ❖ 12月09日截止
 - 23:55
- ❖ 在sopc提交
 - 电子版，可手写

北京航空航天大学

➤2

习题8——总线与IO

- ❖ 已发布
 - Spoc平台
- ❖ 12月13日截止
 - 7:55
- ❖ 在sopc提交
 - 电子版，可手写

北京航空航天大学

➤3

回顾：虚拟存储和Cache的交互

The diagram illustrates three scenarios of memory access flow:

- 物理 cache (Physical Cache):** CPU → TLB (VA/PA) → cache → 更底层 (Lower Level).
- 虚拟 (L1) cache (Virtual (L1) Cache):** CPU → cache → tlb (VA/PA) → 更底层.
- 虚拟-物理 cache (Virtual-Physical Cache):** CPU → (cache, tlb) (VA/PA) → 更底层.

Legend: 物理 cache (Physical Cache), 虚拟 (L1) cache (Virtual (L1) Cache), 虚拟-物理 cache (Virtual-Physical Cache).

北京航空航天大学

➤4

回顾：TLB，页表，Cache三种缺失的可能性

TLB	Page table	Cache	Possible? If so, under what circumstance?
hit	hit	miss	可能，TLB命中则页表一定命中，但实际上不会查页表
miss	hit	hit	可能，TLB缺失但页表可能命中，信息在主存，就可能在Cache
miss	hit	miss	可能，TLB缺失但页表可能命中，信息在主存，但可能不在Cache
miss	miss	miss	可能，TLB缺失页表可能缺失，信息不在主存，一定也不在Cache
hit	miss	miss	不可能，页表缺失，说明信息不在主存，TLB中一定没有该页表项
hit	miss	hit	同上
miss	miss	hit	不可能，页表缺失，说明信息不在主存，Cache中一定也没有该信息

最好的情况应该是hit、hit、hit，此时，访问主存几次？不需要访问主存！

以上组合中，最好的情况是什么？hit、hit、miss和miss、hit、hit 只需访问主存1次

以上组合中，最坏的情况是什么？miss、miss、miss 需访问磁盘、并访问至少2次
介于最坏和最好之间的是什么？miss、hit、miss 不需访问磁盘、但访问至少2次

回顾：总线的一般概念和结构

- ❖ 计算机部件的互连方式：分散连接 → 总线连接
- ❖ 总线：连接两个或多个功能部件的一组公共的信号传输线
- ❖ 总线特性：机械特性；电气特性；功能特性；时间特性
- ❖ 总线的设计要素：类型；仲裁方式；时序；总线宽度；标准传输率；信号线数

1.1 总线的一般概念

❖ 总线的分类

➢ 片内总线：芯片内部连接各元件的总线，如CPU内部的总线，是CPU内各寄存器、寄存器与ALU之间传递信息的公共通道

➢ 系统总线：CPU、主存、I/O部件（I/O接口）之间传递信息的公共通道。一般分为数据总线、地址总线和控制总线三部分

◆ 数据线：传输数据；

◆ 地址线：传输存储器地址和I/O地址；

◆ 控制线：

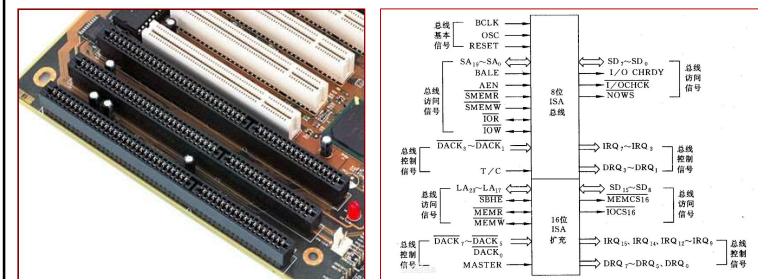
- 数据传输控制信号：存储器读写控制信号、I/O读写控制信号，应答信号等。
- 总线请求和交换信号：总线请求、总线允许，中断请求与响应信号等。
- 其他控制信号：时钟、复位、电源线等

➢ 通信总线：用于计算机系统之间或计算机系统与其他系统之间的通信

1.1 总线的一般概念

❖ ISA (Industrial Standard Architecture, 工业标准体系结构)

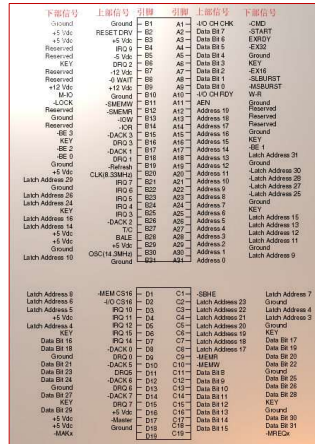
- IBM公司为PC/AT电脑而制定的总线标准，最开始是8位总线；
- 1984年推出IBM-PC/AT系统，ISA从8位扩充到16位；
- 16位数据总线，24位地址总线；
- 总线时钟频率8MHz，最大数据传输率16MB/s。



1.1 总线的一般概念

❖EISA(Extended Industrial Standard Architecture, 扩展的ISA)

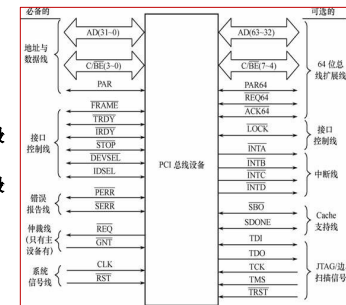
- 1988年, 康柏、HP、NEC等9个厂商协同把ISA扩展到32位, 即EISA总线(Extended ISA)。
- 32位数据总线, 32位地址总线, 总线时钟频率8MHz, 最大数据传输率33MB/s。
- 与ISA兼容, 连接器是一个两层槽设计, 既能接受ISA卡, 又能接受EISA卡。顶层与ISA卡相连, 底层则与EISA卡相连。



1.1 总线的一般概念

❖PCI (Peripheral Component Interconnect, 外部设备互连)

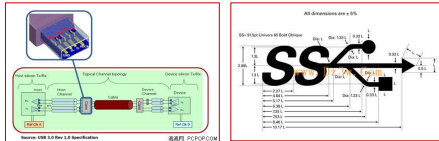
- Intel 1991年, 局部总线
 - 32位或64位的总线位宽
 - 33MHz频率下, 133MB/s~266MB/s的最大数据传输率
 - 66MHz频率下, 266MB/s~533MB/s的最大数据传输率
 - 64位的存储器和I/O寻址能力
 - 完全的多总线主控器
 - 无限突发读/写方式
 - CPU和存储器子系统或PCI设备并发工作
 - 地址线和数据线多路复用
 - 自动配置, 即插即用
 - PCI信号线: 必备的和可选的。作为从设备为最少47条, 作为主设备为最少49条。
- ❖后续发展: PCI-X, PCI-E



1.1 总线的一般概念

❖USB (Universal Serial Bus, 通用串行总线)

- 1995年, Intel、Compaq、Digital、IBM、Microsoft、NEC等7家世界著名的计算机和通信公司共同推出;
- USB采用主从结构, 主机叫Host, 从机叫Device。外观上Host一侧为4针公插, Device一侧为4针母插。可为外设提供电源;
- 允许外设开机状态下热插拔, 最多可串接下来127个外设
- 管脚定义: VCC (5V)、D-、D+、GND
- USB 1.0: 1.5Mbps ~ 12Mbps
- USB 2.0: 数据传输率最高可达480Mbps
- USB 3.0: SuperSpeed USB, 最大传输带宽高达5.0Gbps, 也就是625MB/s



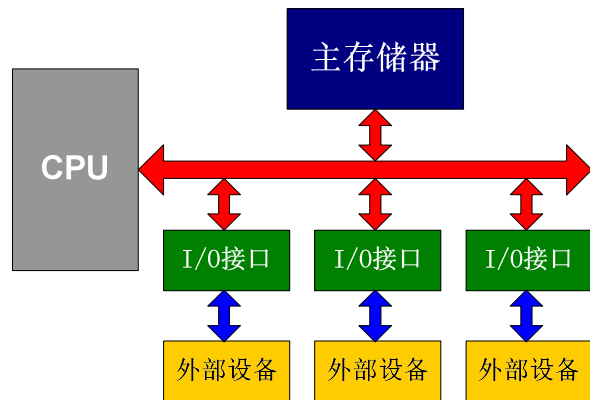
1.1 总线的一般概念

五种主要总线标准的关键特性

Characteristic	Firewire (1394)	USB 2.0	PCI Express	Serial ATA	Serial Attached SCSI
Intended use	External	External	Internal	Internal	External
Devices per channel	63	127	1	1	4
Basic data width (signals)	4	2	2 per lane	4	4
Theoretical peak bandwidth	50 MB/sec (Firewire 400) or 100 MB/sec (Firewire 800)	0.2 MB/sec (low speed), 1.5 MB/sec (full speed), or 60 MB/sec (high speed)	250 MB/sec per lane (1x); PCIe cards come as 1x, 2x, 4x, 8x, 16x, or 32x	300 MB/sec	300 MB/sec
Hot pluggable	Yes	Yes	Depends on form factor	Yes	Yes
Maximum bus length (copper wire)	4.5 meters	5 meters	0.5 meters	1 meter	8 meters
Standard name	IEEE 1394, 1394b	USB Implementers Forum	PCI-SIG	SATA-IO	T10 committee

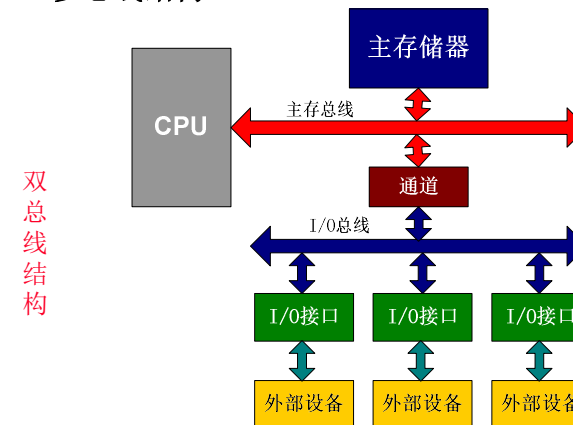
1.2 总线结构

❖单总线结构



1.2 总线结构

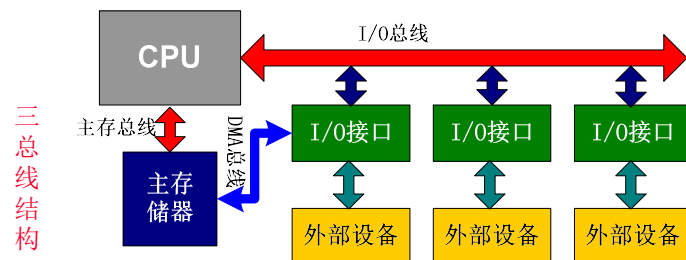
❖多总线结构



双总线结构

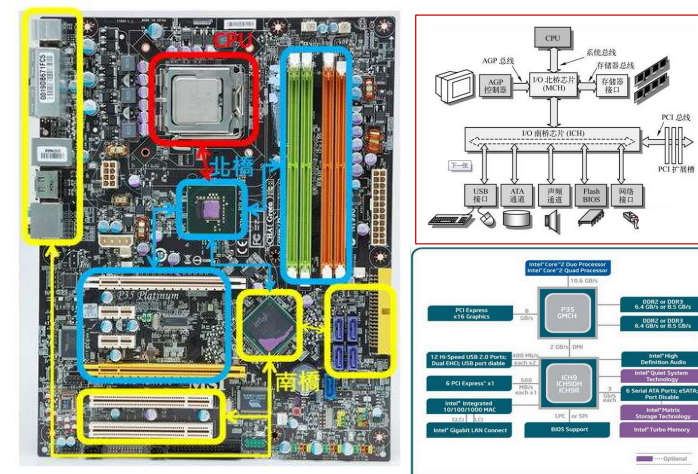
1.2 总线结构

❖多总线结构



三总线结构

X86设备的I/O互联



1.3 总线的仲裁（控制）方式

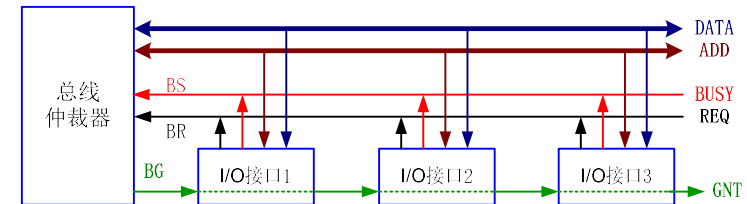
❖ 总线仲裁（控制）方式

- 总线仲裁的策略：优先级或公平
- 集中式的控制方式：总线仲裁逻辑集中在一处（如在CPU中）
 - 链式查询控制方式
 - 计数器定时查询方式
 - 独立请求方式
- 分布式的控制方式：总线仲裁逻辑分散在与总线连接的各部件上
 - 自举分布式仲裁
 - 冲突检测分布式仲裁
 - 并行竞争分布式仲裁

1.3 总线的仲裁（控制）方式

❖ 链式查询方式

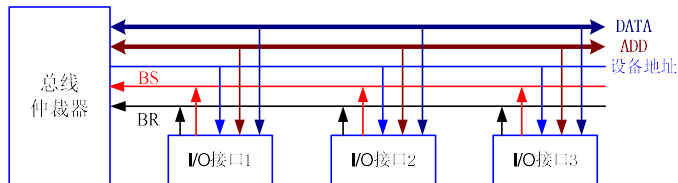
- 总线控制器（仲裁器）收到总线申请BR，BG（总线同意信号）逐个往下传；
- 遇到某接口有总线申请（BR：总线申请信号），BG停止往下传；
- 该接口获得总线使用权，并建立总线忙信号BS。



1.3 总线的仲裁（控制）方式

❖ 计数器定时查询方式

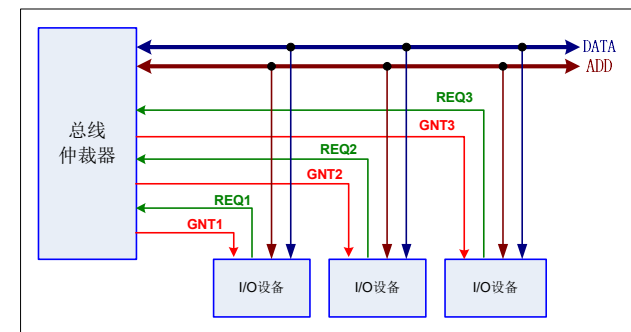
- 总线控制器（仲裁器）收到总线申请BR，计数器开始计数；
- 当某个有总线申请的设备地址与计数器一致，便获得总线使用权，并建立总线忙信号BS。



1.3 总线的仲裁（控制）方式

❖ 独立请求方式

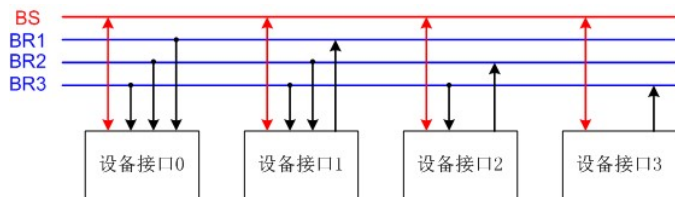
- 每个设备有独立的请求信号和总线同意信号；
- 总线控制器根据设备的优先级决定将总线的使用权交给哪个设备。



1.3 总线的仲裁（控制）方式

❖ 自举分布式仲裁方式

- 不需要集中的总线仲裁器
- 每个设备优先级固定，各设备根据优先级使用总线
- 图：BR1、BR2、BR3分别是设备1、设备2和设备3的总线申请，BS是总线忙信号，设备0只有在BR1、BR2、BR3都没有申请并且BS表示不忙时才能使用总线（将BS置成有效）



1.3 总线的仲裁（控制）方式

❖ 冲突检测分布式仲裁方式

- 设备先查总线是否空闲，若是，立即使用总线（置总线忙）
- 冲突：两个设备同时检查到总线空闲并同时使用总线的现象
- 传输流程
 1. 首先侦听总线，以检测是否发生冲突
 2. 若无冲突，开始传输
 3. 如发生冲突，两个设备都停止传输，延迟一个随机时间后再重新侦听
- 一般用于网络通信，如以太网

1.3 总线的仲裁（控制）方式

❖ 并行竞争分布式仲裁方式

- 基本思想：
 - 每个设备都有唯一的仲裁号
 - 设备申请总线时，主设备将仲裁号发送到仲裁线上；
 - 仲裁号将用在并行竞争算法中
 - 每个设备根据仲裁算法决定在一定的时间段后占用总线还是撤销仲裁号
- 较为复杂但有效的总线仲裁

1.4 总线的通信控制方式

❖ 总线的一次信息传送过程，大致可分五个阶段：

- 请求总线：由需要使用总线的部件或设备提出总线使用申请
- 总线仲裁：仲裁器决定下一传输周期的总线使用权是否授予该部件或设备
- 寻址：获得总线使用权的部件或设备，发出地址和有关命令
- 信息传送：进行数据传输
- 状态返回：该部件或设备有关信息从总线上撤除，让出总线使用权

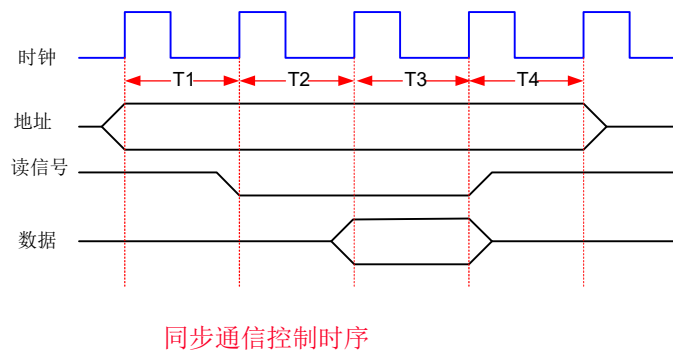
❖ 为协调通信双方，需进行通信控制，常见的方式有：

- 同步通信控制方式
- 异步通信控制方式

1.4 总线的通信控制方式

❖ 同步通信控制方式

- 数据传输在一个统一的时钟同步信号的控制下进行

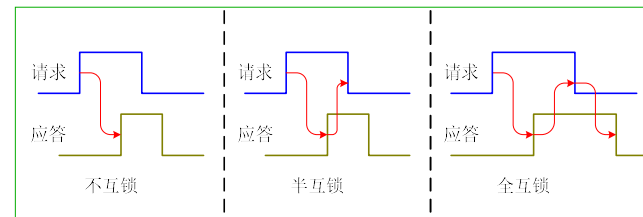


同步通信控制时序

1.4 总线通信的控制方式

❖ 异步通信控制方式

- 没有固定的时钟周期;
- 采用应答方式完成数据传输 (握手协议)
- 有全互锁 (三次握手)、半互锁 (两次握手) 和不互锁 (一次握手) 三种时序。

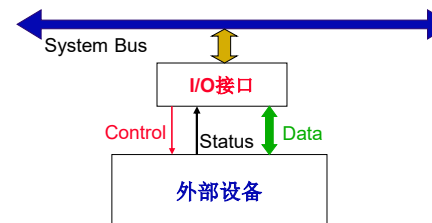


第六部分 总线与I/O

- 一、总线
- 二、I/O接口
- 三、程序查询I/O方式
- 四、中断与中断I/O方式
- 五、DMA I/O方式
- 六、I/O通道

2.1 I/O接口

- ❖ 外部设备并不直接挂载在系统总线上,而是通过I/O接口为桥梁实现与系统总线的连接
- 各种外设使用不同的操作方法, 由CPU来直接控制不同的外设不切实际。
- 外设的数据传送速度比存储器和处理器的速度慢得多, 使用高速的系统总线与慢速的外设直接连接, 不切实际。
- 外设经常使用与处理器不同的数据格式和字长度。



2.1 I/O接口

❖ I/O接口的功能

- 识别I/O地址，即地址译码；
- 实现主机与I/O设备的数据交换、控制命令的传递和状态检测与传递；
- 提供缓冲、暂存和驱动能力；
- 进行数据格式、类型方面的转换（串并行转换，电平转换等）；
- 支持一定的I/O方式（程序查询、程序中断、DMA等）；
- I/O控制与定时

2.1 I/O接口

❖ I/O接口的分类

- 按传送数据格式：串行接口，并行接口
 - 串行接口适合速度低、传输距离长的环境
 - 并行接口适合速度高、传输距离短的环境
- 按I/O方式：程序查询接口、中断接口、DMA接口、通道控制接口
- 按时序控制方式：同步接口、异步接口
 - 同步接口：数据传送由一个统一的时钟信号同步控制
 - 异步接口：数据传送采用异步应答方式控制

2.1 I/O接口

❖ I/O操作的过程

- 处理器查询I/O接口状态，以检查连接设备的状态；
- I/O接口回送设备状态；
- 如果设备可用，并准备好，CPU向I/O接口发出命令，请求传送；
- I/O接口获得来自外设的数据（字或字节）；
- 数据从I/O接口传送自CPU。

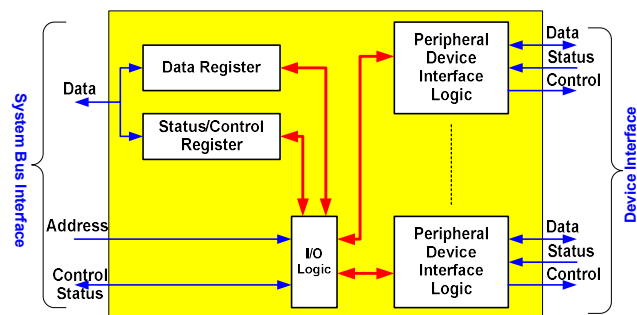
2.1 I/O接口

❖ I/O设备的编址

- I/O接口的编址
- I/O地址（I/O接口地址，I/O端口地址）：实际上是I/O接口电路中寄存器的地址。
- 编址方式
 - 独立编址方式：存储器地址与I/O地址分开，CPU具有专用的I/O指令，系统总线中具有区别存储器读写和I/O操作的控制信号，并以此区别地址总线上的地址是存储器地址还是I/O地址。
 - 统一编址方式：存储器地址与I/O地址统一考虑，地址空间的一部分是存储器，另一部分是I/O，支持存储器操作的指令都可用于I/O操作。

2.1 I/O接口

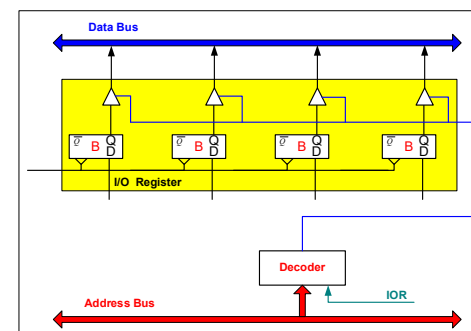
❖ I/O接口的通用结构



2.1 I/O接口

❖ I/O接口地址选择（译码）

- I/O接口地址是I/O接口电路中寄存器的地址
- 独立编址方式下的I/O地址选择电路



第九部分 总线与I/O

- 一、总线
- 二、I/O接口
- 三、程序查询I/O方式
- 四、中断与中断I/O方式
- 五、DMA I/O方式
- 六、I/O通道

I/O与主机信息交换的控制方式

- ❖ [例] 幼儿园老师带9个孩子，要给每个孩子分3块水果糖。并且要孩子们把3块糖都吃完，那么她可以采用什么方法呢？
- ❖ 方法1：她先给孩子甲一块糖，盯着甲吃完，然后再给第二块，等吃完第二块又给第三块。接着给孩子乙，其过程与孩子甲完全一样。依此类推，直到给第9个孩子发完3块糖。
- ❖ 方法2：每个孩子发1块糖各自去吃，并约定谁吃完后就向她举手报告，再发第2块。
- ❖ 方法3：进行批处理：每个孩子拿3块糖各自去吃，吃完3块糖后再向她报告。
- ❖ 方法4：权力下放，把发糖的事交给另一个孩子分管，只是必要时她才过问一下。

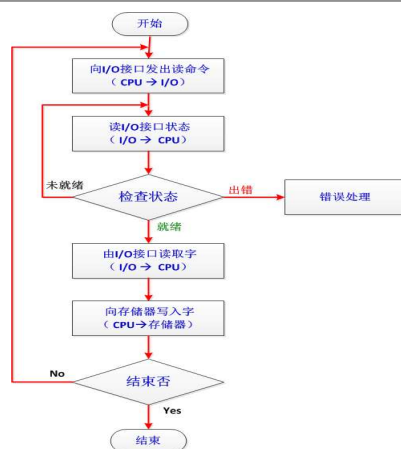
I/O与主机信息交换的控制方式

- ❖ 程序查询方式
- ❖ 程序中中断方式
- ❖ 直接内存访问(DMA)方式
- ❖ 通道方式

3.1 程序查询I/O方式

- ❖ 也称编程式I/O，处理器执行程序直接控制I/O，包括：
 - 检测设备状态
 - 发送读写命令（处理器发送I/O命令后，必须等待，直到I/O操作完成）
 - 传送数据
- ❖ I/O命令
 - 控制命令：激活外设完成动作。如指示磁带机快进或快退，控制命令与设备类型相关；
 - 测试命令：测试与I/O接口及其外部设备的各种状态条件；
 - 读命令：使I/O接口从外设获得一个数据项，存入内部缓冲区；
 - 写命令：使I/O接口从数据总线获得一个数据项，然后传送到外设。

3.1 程序查询I/O方式



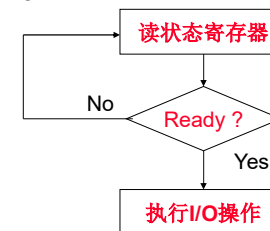
3.1 程序查询I/O方式

- ❖ I/O接口设置状态寄存器以表示外部设备的工作状态
- ❖ CPU通过不断读取状态寄存器以查询外部设备的状态
- ❖ 在外部设备准备就绪的时候，CPU通过I/O接口中的数据寄存器与外设完成数据交换。

示例（串行接口COM1（RS-232C串行接口））：

状态寄存器端口地址：3FD；状态寄存器：61H表示数据准备就绪；数据寄存器端口地址：3F8

```
RdSta: MOV DX,3FDH
        IN AL,DX
        CMP AL,61H
        JNE RdSta
        MOV DX,3F8H
        IN AL,DX
```



3.1 程序查询I/O方式

❖ 程序查询I/O接口的基本组成

- 状态寄存器
- 数据寄存器（输入寄存器，输出寄存器）
- 地址选择逻辑
- 总线接口逻辑

❖ 程序查询I/O方式的特点

- I/O操作由CPU直接完成（通过执行I/O指令完成）
- 外设速度慢，CPU速度快，在外设准备过程中，CPU处在不断的查询之中，CPU的效率得到了极大的浪费
- 外设与CPU完全串行工作

例子

- ❖ 在程序查询方式的I/O系统中，假设不考虑处理时间，每一次查询操作需要100个时钟周期，CPU的时钟频率为50Mhz。现有鼠标和硬盘两个设备，CPU必须每秒对鼠标进行30次查询；硬盘以32位字长为单位传输数据，即每32位被CPU查询一次，传输率为2MBps。求CPU对这两个设备查询所花费的时间比率，由此可得到什么结论？
- ❖ 每秒内CPU的时钟周期数： $1/(1/50\text{Mhz}) = 50 \times 10^6$ 个
- ❖ 对鼠标查询，每秒所需时钟周期数： 30×100 个
相应的时间比率： $(30 \times 100) / (50 \times 10^6) = 0.006\%$
- ❖ 对硬盘查询，每秒所需时钟周期数： $(2\text{MB}/4\text{B}) \times 100$ 个
相应的时间比率： $((2\text{MB}/4\text{B}) \times 100) / (50 \times 10^6) \approx 100\%$

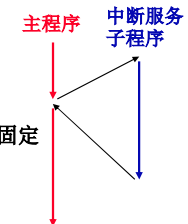
第九部分 总线与I/O

- 一、总线
- 二、I/O接口
- 三、程序查询I/O方式
- 四、中断与中断I/O方式
- 五、DMA I/O方式
- 六、I/O通道

4.1 中断与中断I/O

❖ 中断的概念

- 概念：机器出现了一些紧急事务，CPU不得不停下当前正在执行的程序，转去处理紧急事务，当紧急事务处理完后，继续执行被中断的程序
- 作用：主机与外设并行、实时处理和过程控制、硬件故障处理、多道程序和分时操作
- 一般情况下，中断是随机的
- 主程序：被中断的程序
- 中断服务子程序：处理中断事务的程序
- 中断向量：中断服务子程序的入口地址
- 中断向量表：保存所有中断向量的内存区域，一般固定



4.1 中断与中断I/O

❖引起中断的因素（中断源）

- 人为设置的中断：自愿中断，可重复
- 程序性事故：如溢出、除以“零”等
- 硬件故障：如电源掉电、磁盘损坏
- I/O操作：I/O设备准备就绪，请求操作
- 外部事件：如键盘操作

❖ 中断源分类

- 不可屏蔽中断：CPU不能不响应；
- 可屏蔽中断：若中断源被屏蔽，CPU不响应

❖ 中断的分类

- 非屏蔽中断与可屏蔽中断
- 程序中断与简单中断
- 硬中断与软中断（软中断不是真正的中断）

4.1 中断与中断I/O

❖ 中断系统需要解决的问题

- 中断源如何向**CPU**提出中断申请;
- 多个中断同时申请时, 中断系统如何响应;
- **CPU**响应中断的时间、条件和方式;
- **CPU**响应中断后如何保护现场;
- **CPU**响应中断后, 如何转向中断服务子程序;
- 中断处理结束后, **CPU**如何恢复现场返回主程序断点位置;
- 中断处理过程中出现新的中断申请怎么处理

4.1 中断与中断I/O

❖ 中断请求

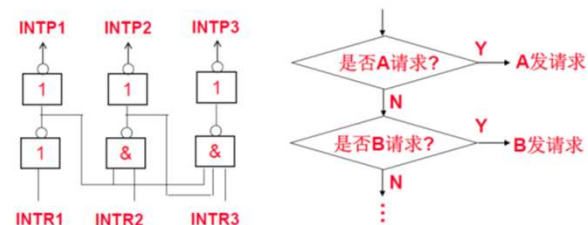
- **中断请求触发器（INTR）：**
 - 每个中断源配置一个中断请求触发器
 - 中断源可通过设置中断请求触发器来提出中断申请
- **中断请求标记寄存器：**各中断源请求触发器组成中断请求标记寄存器

1	2	3	4			n
掉电	过热	主存读写检验错	溢出		键盘输入	打印机输出

4.1 中断与中断I/O

◆ 中断判优逻辑

- 中断系统任何时刻最多只能响应一个中断源的请求
- 硬件排队判优
- 软件排队判优



4.1 中断与中断I/O

❖ 中断响应

- 条件：当前执行的程序允许被中断（即中断允许标志位为允许中断），非屏蔽中断不受中断允许标志位的限制
- 时机：当前指令执行完后，才能响应中断
- 方式：在允许中断的前提下，每条机器指令的执行周期中实际上包含一个中断周期，执行中断隐指令

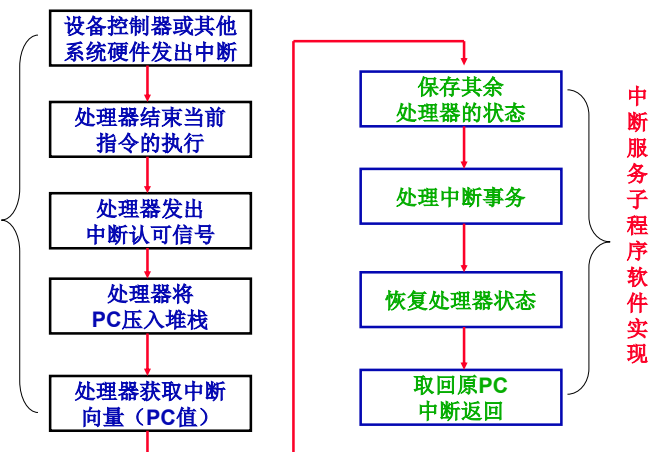
4.1 中断与中断I/O

❖ 中断处理

- 硬件自动执行中断隐指令
 - 保护程序断点：程序计数器内容入栈；
 - 查找中断服务子程序入口地址（中断向量）送PC，转向中断服务子程序；
 - 关中断。
- CPU执行中断服务子程序
 - 执行中断服务处理功能
 - 从中断服务子程序中返回：恢复程序断点，即把保存在堆栈中的PC内容弹出送PC，接下来继续执行主程序。

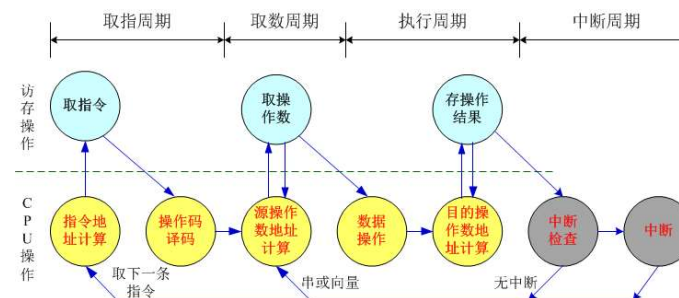
4.1 中断与中断I/O

硬件实现



4.1 中断与中断I/O

❖ 有中断的指令周期状态图



4.1 中断与中断I/O

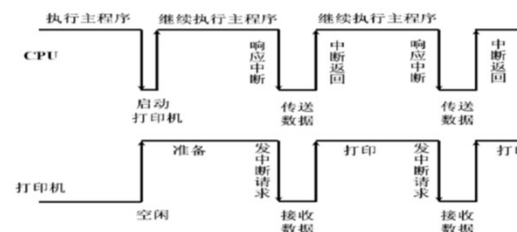
❖ 中断I/O接口的基本组成

- 数据输入寄存器
- 数据输出寄存器
- 状态寄存器
- 控制寄存器
- 地址选择逻辑
- 中断控制逻辑
- 总线接口逻辑

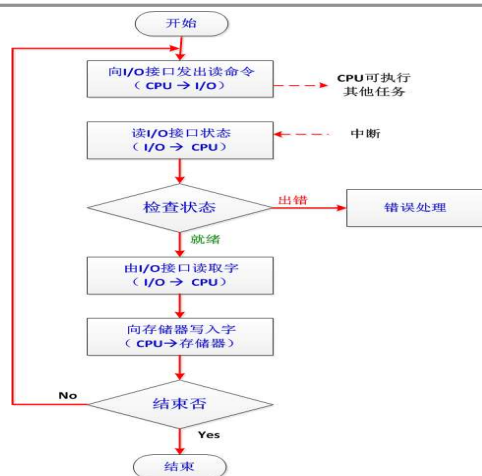
4.1 中断与中断I/O

❖ 中断I/O方式的特点

- I/O操作仍然由CPU通过I/O指令完成
- 在外设准备阶段，CPU可以执行其他程序，仅在外设准备就绪后，CPU才中断正在执行的程序，处理I/O事务
- 在外设准备阶段，CPU与外设的工作可以认为是并行的
- 中断I/O方式是目前最主要的I/O方式



4.1 中断与中断I/O



4.1 中断与中断I/O

❖ 多重中断

- 多重中断的概念
- 实现多重中断的条件
- 中断屏蔽触发器
- 中断屏蔽字
- 中断屏蔽字与中断优先级的关系
- 中断处理次序与中断屏蔽字的关系
- 多重中断的断点保护

思考

❖在程序中中断方式中，磁盘申请中断的优先级高于打印机。当打印机正在打印时，磁盘申请中断，试问是否要将打印机的打印操作停下来，等磁盘操作结束后，打印机才能继续打印？为什么？

- 打印操作与磁盘中断的关系
- 通常，打印机的打印动作只受打印机本身控制，与CPU无直接关系，因此，当打印机正在打印时，即使有优先级更高的磁盘请求中断，打印机也不会停止打印
- 但是，如果CPU正在执行打印机的中断服务程序，即打印机可能正在接收数据，此时，若磁盘请求中断，CPU就要中断正在运行的打印机中断服务程序，向打印机的数据传送会受到影响

第九部分 总线与I/O

- 一、总线
- 二、I/O接口
- 三、程序查询I/O方式
- 四、中断与中断I/O方式
- 五、DMA I/O方式
- 六、I/O通道

5.1 DMA的一般概念

❖程序I/O与中断I/O的不足

- I/O传送速度受处理器测试和服务设备速度的限制
- 处理器直接负责管理I/O，对于每一次I/O传送，处理器必须执行一些指令

❖DMA (Direct Memory Access)

- CPU对总线的控制被临时禁止。DMA控制器接管总线控制权，控制数据直接在存储器与外设之间高速交换
- CPU不再介入具体的I/O操作，由DMA控制器来负责提供存储器地址信号、读写控制信号等
- CPU与I/O设备在更大的程度上并行工作，效率更高
- DMA方式适合高速批量的数据传输，如视频显示刷新、磁盘存储系统的读写，存储器到存储器的传输等

5.2 DMA过程

❖CPU的工作：初始化DMA控制器

- 设置数据传送方向：是请求读还是请求写（对存储器而言）
- 设置I/O接口地址：DMA操作所涉及的I/O接口的地址
- 设置存储器起始地址：读或写存储器的起始单元地址
- 设置传送的数据数量：传送数据的字数
- 有关中断方式的设置：DMA结束后通过中断方式请求CPU处理

❖DMA请求

- 当接口做好数据传输的准备，通过有关逻辑向CPU发出DMA请求信号

❖DMA响应

- CPU接到DMA请求，在当前总线周期操作结束后，暂停CPU对系统总线的控制和使用，发出DMA响应信号，并交出系统总线的控制权

5.2 DMA过程

❖ DMA操作

- DMA控制器接到DMA应答信号后，通过控制逻辑向系统总线发送存储器地址信号、存储器读写控制信号、I/O接口读写控制信号等，完成一次数据传送。这些操作完全由硬件控制，一般仅需要一个总线周期，所以这种方式称为**周期窃用 (cycle-stealing)方式**
- 所有数据传送结束后，通过中断方式告知CPU进行善后处理
- CPU仅在开始DMA操作之前和完成DMA操作之后参与I/O处理，在DMA过程中，CPU可以运行原来的程序

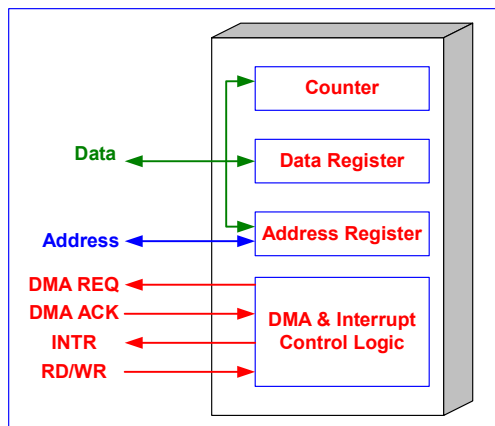
5.2 DMA过程

❖ DMA方式

- 周期窃取方式（单字传送方式）
 - 每次DMA请求得到响应后，DMA控制器窃取一个总线周期完成一次数据传送，然后释放总线
 - 一般适应存储器速度远高于I/O设备速度的情况
- 停止CPU访问内存（成组传送方式）
 - 一次DMA请求得到响应后，DMA控制器完全占用总线，进行多次DMA传送，直到所有数据传送完毕才释放总线，这段时间完全停止CPU访问内存
 - 适应高速外设与存储器交换数据的情况

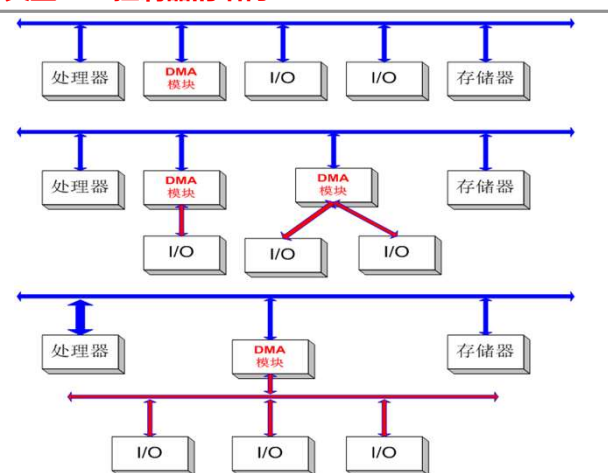
5.3 典型DMA控制器的结构

- Counter: 长度计数器，保存传送数据的字数。
- Data Reg: 数据寄存器。
- Address Reg: 地址寄存器，向地址总线提供存储器地址。
- DMA控制逻辑
- DMA状态逻辑
- 中断控制逻辑



5.3 典型DMA控制器的结构

可能的
DMA
结构



5.3 典型DMA控制器的类型

❖ 选择型DMA控制器

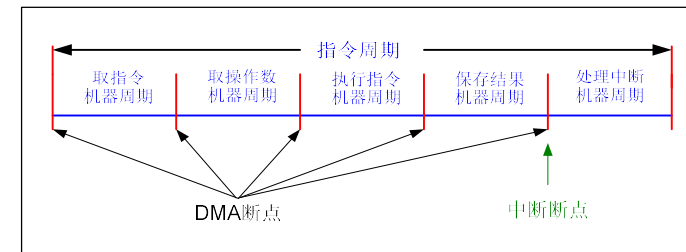
- 物理上可以连接多个I/O接口（外设）；
- 逻辑上只能连接一个设备，即在某一时间段只能为其中一台外设服务。
- 适应于数据传输率很高（接近于内存）的外设数据传输服务。

❖ 多路型DMA控制器

- 物理上可以连接多个I/O接口（外设）；
- 逻辑上也可连接多个设备，可通过交叉服务的方式为多台外设服务；
- 多路型DMA控制器内部应包括多个DMA通道；
- 适应于多个慢速（相对）外设的数据传输服务。

5.4 DMA vs. 中断

指令周期中的DMA和中断断点



- 响应时机：中断是在一条指令结束后响应；而DMA可在指令周期内的任一存取周期结束时响应
- 现场保护：中断要中断现行程序，需保护现场；而DMA不中断现行程序，无须保护现场
- 适应场合：中断适于处理紧急或异常事件；而DMA适于传送大批数据
- 传送方式：中断需要靠程序传送数据；而DMA靠硬件传送

第九部分 总线与I/O

- 一、总线
- 二、I/O接口
- 三、程序查询I/O方式
- 四、中断与中断I/O方式
- 五、DMA I/O方式
- 六、I/O通道

6.1 通道I/O方式及其特点

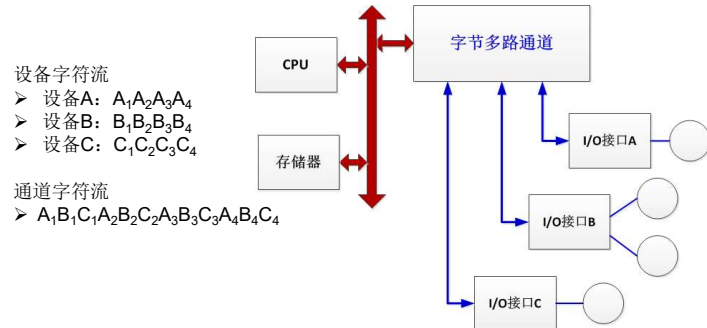
❖ 通道I/O的特点

- 通道I/O是一种专用的I/O控制器，具有自己的指令系统（基本上都是I/O指令）和执行这些I/O指令的专用处理器
- 通道I/O执行通道程序来实现和管理I/O，CPU基本上不需要管理I/O，CPU的效率得到更大的提高
- 通道程序由操作系统根据I/O任务的需求自动生成，存放在存储器中，通道程序由操作系统管理，用户程序执行和访问通道程序
- CPU通过请求I/O通道执行存储器中的通道程序来启动一次I/O数据传送，通道程序将指定一个或几个设备、一块或几块存储区域、优先级以及出错时的处理行为等，I/O通道通过执行这些指令来控制数据传送

6.2 通道分类

❖ 字节多路通道

- 通道连接多台慢速外设，通道可以同时为多台设备服务；
 - 以字节为单位交叉传送各外设的数据；
 - 通道的数据传输率 = 各外设的数据传输率之和。
- 不仅允许多个设备同时操作，且允许它们同时进行传输型操作

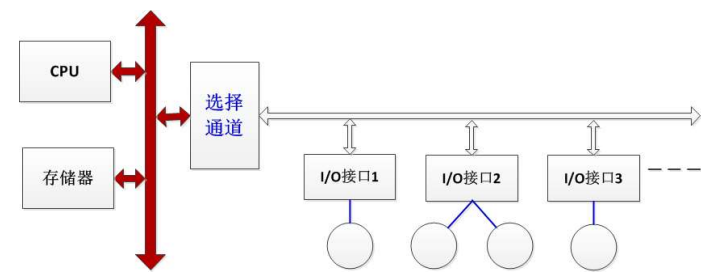


69

6.2 通道分类

❖ 选择通道

- 通道可以连接多台高速设备，但一次只能为其中一台设备服务；
 - 与一台设备的成组数据传送结束后，才能选择另一台设备；
 - 通道数据传输率 = 一台设备的数据传输率。
 - 一旦选择了一个外设，即使该外设没有准备好，也只能等待。
- 一般用于高速外部设备的数据传输

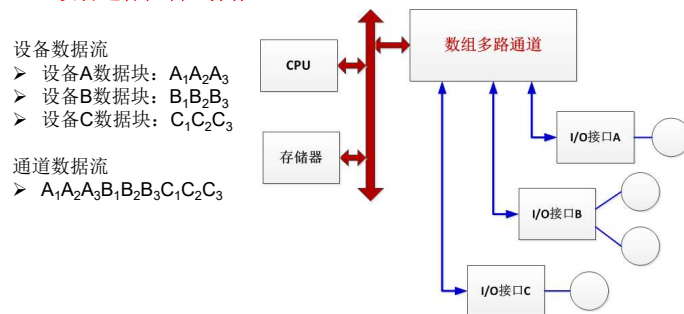


70

6.2 通道分类

❖ 数组多路通道

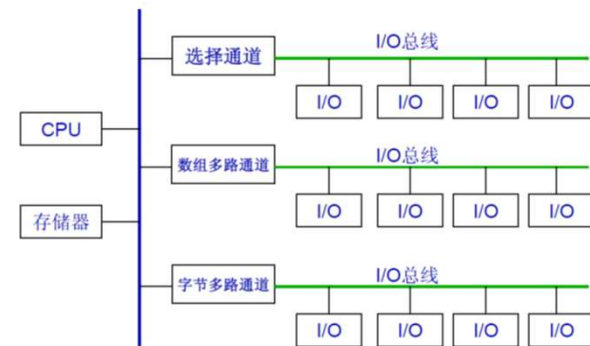
- 通道可以连接多台高速外设，通道可以同时为多台设备服务；
 - 以数据块为单位交叉传送各外设数据。
 - 通道数据传输率 = 各设备数据传输率之和
- 允许多个设备同时工作，但只允许一个设备进行传输型操作，其他设备进行控制型操作



71

6.2 通道分类

❖ 通道结构



72

I/O方式小结

❖ I/O方式的演变（CPU从I/O事务中的解放）

- ① **直接控制方式**：CPU直接控制外设，主要用于简单的微处理器控制设备；
- ② **程序I/O方式**：增加控制器和I/O模块，处理器使用编程I/O，使处理器从外设的I/O细节中解脱出来；
- ③ **中断I/O方式**：增加控制器和I/O模块，采用中断I/O方式，处理器不需要浪费时间等待I/O操作完成，提高了处理器的效率；
- ④ **DMA方式**：I/O模块通过DMA直接存取存储器，除在传输开始和结束时，传输数据不需要处理器参与；
- ⑤ **I/O通道方式**：I/O模块成为有自主控制权的处理器，有处理I/O的专用指令集。CPU指示I/O处理器执行存储器中的I/O程序，I/O处理器不需要CPU干预就能获取并执行I/O指令。这允许CPU指派一系列的I/O活动，并只在整个活动执行完成后才中断CPU；
- ⑥ **I/O处理器方式**：I/O模块带局部存储器，成为自治的计算机。这种结构可以控制大量的I/O设备而最小化CPU的干预。