

Ch 8: System Bus

系统总线

总线基本概念
总线设计要素
总线标准
总线互连结构

总线的分类

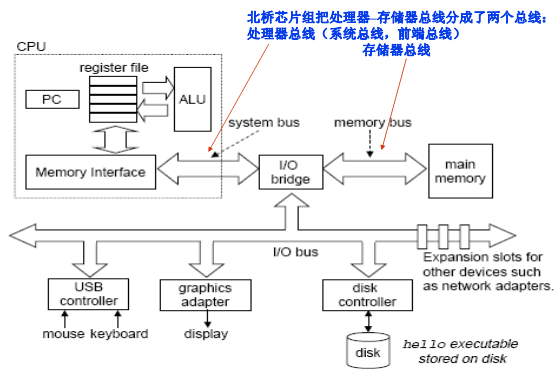
- 总线在各层次上提供部件之间的连接和交换信息通路
- 分为以下几类：
 - 芯片内总线：在芯片内部各元件之间提供连接
 - 例如，CPU芯片内部，各寄存器、ALU、指令部件等之间有总线相连
 - 系统总线：在系统主要功能部件（CPU、MM和各种I/O控制器）间提供连接
 - 单总线结构
 - 将CPU、MM和各种I/O适配卡通过底板总线(Backplane Bus)互连，底板总线为标准总线(Industry standard)
 - 多总线结构
 - 将CPU、Cache、MM和各种I/O适配卡用局部总线、处理器-主存总线、高速I/O总线、扩充I/O总线等互连。主要有两大类：
 - Processor-Memory Bus (Design specific or proprietary)
 - 短而快，仅需与内存匹配，使CPU-MM之间达最大带宽
 - I/O Bus (Industry standard)
 - 长而慢，需适应多种设备，一侧连接到Processor-Memory Bus 或 Backplane Bus，另一侧连到I/O控制器
- (注：Intel公司在推出845、850等芯片组时，对“System Bus”有专门的定义，将处理器总线称为前端总线(Front Bus)或系统总线)
- 通信总线：在主机和I/O设备之间或计算机系统之间提供连接

SKIP

io.2

2010年05月20日星期二

Intel 体系结构中特指的“系统总线”



io.3

BACK

2010年05月20日星期二

系统总线的组成

- 系统总线通常由一组控制线、一组数据线和一组地址线构成。也有些总线没有单独的地址线，地址信息通过数据线来传送，这种情况称为数据/地址复用。
- 数据线 (Data Bus)：承载在源和目的部件之间传输的信息。数据线的宽度反映一次能传送的数据的位数。
- 地址线 (Address Bus)：给出源数据或目的数据所在的主存单元或I/O端口的地址。地址线的宽度反映最大的寻址空间。
- 控制线 (Control Bus)：控制对数据线和地址线的访问和使用。用来传输定时信号和命令信息。典型的控制信号包括：
 - 时钟 (Clock)：用于总线同步。
 - 复位 (Reset)：初始化所有设备。
 - 总线请求 (Bus Request)：表明发出该请求信号的设备要使用总线。
 - 总线允许 (Bus Grant)：表明接收到该允许信号的设备可以使用总线。
 - 中断请求 (Interrupt Request)：表明某个中断正在请求。
 - 中断回答 (Interrupt Acknowledge)：表明某个中断请求已被接受。
 - 存储器读 (memory read)：从指定的主存单元中读数据到数据总线上。
 - 存储器写 (memory write)：将数据总线上的数据写到指定的主存单元中。
 - I/O读 (I/O read)：从指定的I/O端口中读数据到数据总线上。
 - I/O写 (I/O Write)：将数据总线上的数据写到指定的I/O端口中。
 - 传输确认 (transmission Acknowledge)：表示数据已被接收或已被送到总线

io.4

2010年05月20日星期二

总线设计要素

- 总线设计要考虑的基本要素
- 尽管有许多不同的总线实现方式，但总线设计的基本要素和考察的性能指标一样
- ①信号线类型(Signal line type):
 - 专用(Separate) / 复用(Multiplexed)
- ②仲裁方法(Arbitrating):
 - 集中式(Center) / 分布式(distributed)
- ③定时方式(Timing):
 - 同步通信 (Synchronous) / 异步通信 (Asynchronous)
- ④事务类型(Bus Transaction):
 - 总线所支持的各种数据传输类型和其他总线操作类型，如：
 - 存储器读、存储器写、I/O读、I/O写、读指令、中断响应等
- ⑤总线带宽(Bus Bandwidth):
 - 单位时间内在总线上传输的最大数据量（是一种传输能力）
 - 相当于公路的最大载客量。例如，沪宁高速每车道最多每5分钟发一列车，每辆车最多50人，共有6个车道，则最大流量为多少（?人/小时）？
 - 最大载客量：6道x12车/小时x50人/车= 3600人/小时

io.5

2010年05月20日星期二

信号线类型

总线的信号线类型有：专用、复用

- 专用信号线：
 - 信号线专用来传送某一种信息。
 - 例如，使用分立的数据线和地址线，使得数据信息专门由数据线传输，地址信息专门由地址线传输。
- 复用信号线：
 - 信号线在不同的时间传输不同的信息。
 - 例如，许多总线采用数据/地址线分时复用方式，用一组数据线在总线事务的地址阶段传送地址信息，在数据阶段传送数据信息。这样就使得地址和数据通过同一组数据线进行传输。
- 信号分时复用的优缺点：
 - 优：减少总线条数，缩小体积、降低成本。
 - 缺：总线模块的电路变复杂，且不能并行。

io.6

2010年05月20日星期二

总线裁决（总线控制/使用/访问权的获得）

总线被多个设备共享，但每一时刻只能有一对设备使用总线传输信息。

- 什么是总线裁决？
当多个设备需要使用总线进行通信时，采用某种策略选择一个设备使用总线
- 为什么要进行总线裁决？
总线被连接在其上的所有设备共享，如果没有任何控制，那么当多个设备需要进行通信时，每个设备都试图为各自的传输将信号送到总线上，这样就会产生混乱。所以必须进行总线裁决
- 如何避免上述混乱？
 - 在总线中引入一个或多个总线主控设备，只能主控设备控制总线
 - 主控设备：能发起总线请求并控制总线。（如：处理器）
 - 从设备：只能响应从主控设备发来的总线命令。（如：主存）
 - 利用总线裁决决定哪个总线主控设备将在下次得到总线使用权

io.7

2009年5月20日星期二

总线裁决（总线控制/使用/访问权的获得）

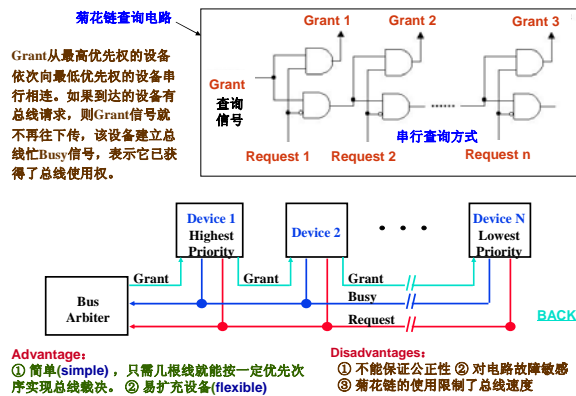
- ①总线裁决信号：总线请求线和总线许可线
信号线专用 / 信号线复用
如：数据线和总线请求线复用时，总线裁决和数据传输不能同时进行
- ②总线裁决有两种方式：集中式和分布式
集中式：将控制逻辑放在一个专门的总线控制器或总线裁决器中，通过将所有的总线请求集中起来利用一个特定的裁决算法进行裁决
菊花链（Daisy chain）
集中并行（Centralized, Parallel）
分布式：没有专门的总线控制器，其控制逻辑分散在各个部件或设备中
自举式（Self-selection）
冲突检测（Collision detection）
- ③裁决方案应在以下两个因素间进行平衡
等级性(Priority)—具有高优先级的设备应该先被服务
公平性(Fairness)—即使具有最低优先权的设备也不能永远得不到总线使用权

io.8

2009年5月20日星期二

SKIP

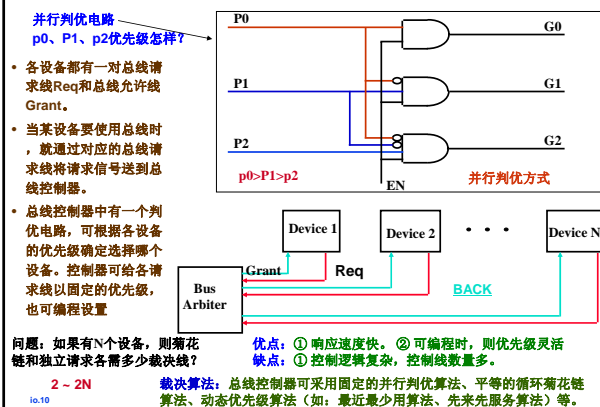
菊花链总线裁决



io.9

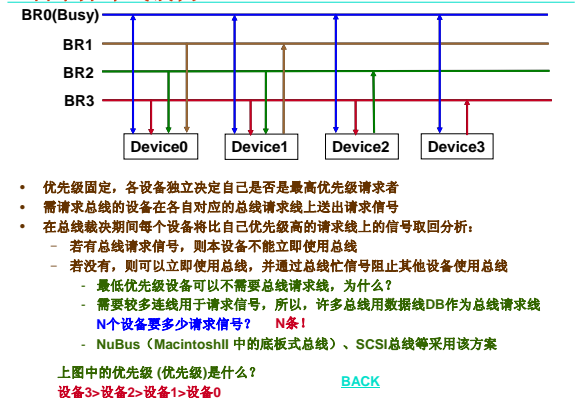
2009年5月20日星期二

独立请求方式裁决



io.10

自举分布式裁决



io.11

2009年5月20日星期二

冲突检测方式裁决

基本思想：

当某个设备要使用总线时，它首先检查一下是否有其他设备正在使用总线

如果没有，那它就置总线忙，然后使用总线；

若两个设备同时检测到总线空闲，则可能会同时使用总线，此时发生冲突；

一个设备在传输过程中，它会侦听总线以检测是否发生了冲突；

当冲突发生时，两个设备都会停止传输，延迟一个随机时间后再重新使用总线

- 该方案一般用在网络通信总线上，如：Ethernet总线等。

BACK

io.12

2009年5月20日星期二

总线定时方式

什么是总线的定时

通过总线裁决确定了哪个设备可以使用总线，那么一个取得了总线控制权的设备如何控制总线进行总线操作呢？也即如何来定义总线事务中的每一步何时开始、何时结束呢？这就是总线通信的定时问题。

总线通信的定时方式

- **Synchronous (同步)**：用时钟来同步定时
- **Asynchronous (异步)**：用握手信号定时
- **Semi-Synchronous (半同步)**：同步(时钟)和异步(握手信号)结合
- **Split transaction (拆分事务)**：在从设备准备数据时，释放总线

CPU-处理器总线都采用同步方式

异步方式只有I/O总线才会使用

I/O总线大多采用半同步方式

拆分事务方式可以提高总线的有效带宽

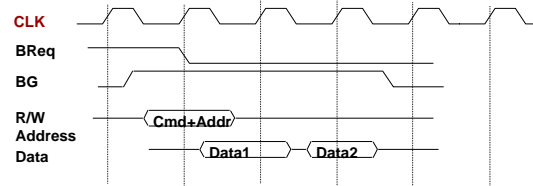
[SKIP](#)

io.13

2009年05月20日星期二

同步总线 (Synchronous Bus)

简单的同步协议如下图：一个总线事务：地址阶段 + 数据阶段 + ... + 数据阶段



控制线上有一个时钟信号进行定时，有确定的通信协议

[BACK](#)

Advantage(优点): 控制逻辑少而速度快

Disadvantages(缺点):

- (1) 所有设备在同一个时钟速率下运行，故以最慢速设备为准
- (2) 由于时钟偏移问题，同步总线不能很长

实际上，存储器总线比这种协议的总线复杂得多

存储器（从设备）响应需要一段时间，并不能在随后的时钟周期就准备好数据

io.14

2009年05月20日星期二

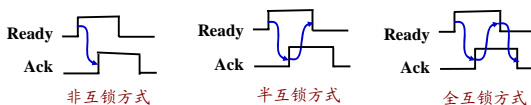
异步总线 (Asynchronous Bus)

非时钟定时，没有一个公共的时钟标准。因此，能够连接带宽范围很大的各种设备。总线能够加长而不用担心时钟偏移 (clock skew) 问题

采用**握手协议 (handshaking protocol)** 即：应答方式。

- 只有当双方都同意时，发送者或接收者才会进入到下一步，协议通过一对附加的“握手”信号线 (Ready、Ack) 来实现

异步通信有非互锁、半互锁和全互锁三种方式



• 优点：灵活，可挂接各种具有不同工作速度的设备

- 缺点：① 对噪声较敏感（任何时候都可能接收到对方的应答信号）
② 接口逻辑较复杂

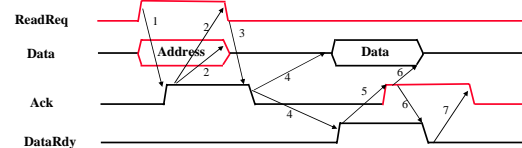
[BACK](#)

io.15

2009年05月20日星期二

Handshaking Protocol(握手协议)

一个总线事务：地址阶段 + 数据阶段 + ... + 数据阶段



Three control lines

- **ReadReq**: 请求读内存单元
(地址信息同时送到地址/数据线上)
- **DataRdy**: 表示已准备好数据
(数据同时送到地址/数据线上)
- **Ack**: ReadReq or DataRdy 的应答信号

上述为read过程, 但write操作基本类似

ReadReq和Ack之间的握手过程
完成地址信息的传输
DataRdy和Ack之间的握手过程
完成数据信息的传输
一共有多少次握手？ 7次
是全互锁方式！

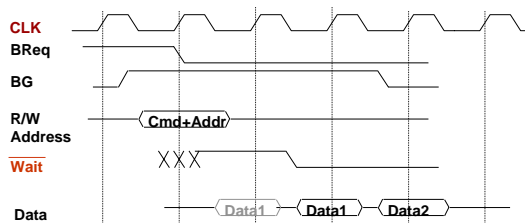
[BACK](#)

io.16

2009年05月20日星期二

半同步总线

为解决异步方式对噪声敏感的问题，在异步总线中引入时钟信号就绪和应答等握手信号（如：Wait信号、TRDY和IRDY信号等）都在时钟的上升沿有效信号的有效时间限制在时钟到达的时刻，而不受其他时间的信号干扰



通过“Wait”信号从设备告知主设备何时数据有效

结合了同步和异步的优点。既保持了“所有信号都由时钟定时”的特点，又允许“不同速度设备共享于总线”

[BACK](#)

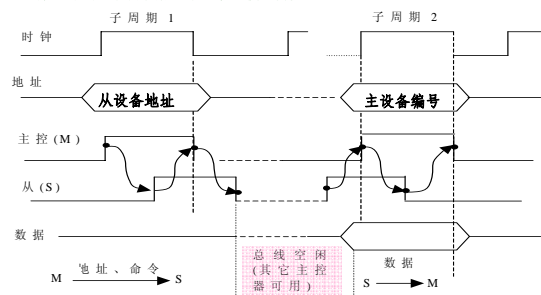
io.17

2009年05月20日星期二

Split Bus Transaction(拆分总线事务)

将一个事务分成两个子过程：

- 过程1：主控设备A获得总线使用权后，将请求的事务类型、地址及其他信息（如A的标识等）发到总线，从设备B记下这些信息。A发完信息后便立即释放总线，其他设备便可使用总线
- 过程2：B收到A发来的信息后，按照A的要求准备数据，准备好后，B便请求使用总线，获使用权后，B将A的编号及所需数据送到总线，A便可接收



io.18

2009年05月20日星期二

Split Bus Transaction(拆分总线事务)

- 请求- 回答方式 (Request-Reply)
 - CPU启动一次读或写事务
 - 传送信息: address, data, and command
 - 然后等待存储器回答
 - 分离总线事务方式 (Split Bus Transaction)
 - CPU启动一次读/写事务后, 释放总线
 - 传送信息: address, data (Write) , and command
 - 存储器启动一次回答事务, 请求使用总线
 - 传送信息: data (read) or acknowledge (write)
- 优点: **系统总效率改善**
(例如, 在存储器存取数据时可以释放总线, 以被其他设备使用)
- 缺点: **单独的事务响应时间变长**
增加复杂性

io.19

2009年9月26日星期二

例1: 同步和异步总线的最大带宽比较

举例: 假定同步总线的时钟周期为50ns, 每次总线传输花1个时钟周期, 异步总线每次握手需要40ns, 两种总线的数据都是32位宽, 存储器的取数时间为200ns。要求求出从该存储器中读出一个字时两种总线的带宽。

分析如下:

同步总线的步骤和时间为:

- (1) 发送地址和读命令到存储器: 50ns
 - (2) 存储器读数据: 200ns
 - (3) 传送数据到设备: 50ns
- 如果存储器读为230ns, 则结果是多少?
总时间为350ns, 4B/350ns=11.4MB/s
- 所以总时间为300ns, 故最大总线带宽为4B/300ns, 即: 13.3MB/s。

异步总线的步骤和时间为:

- 第1步为: 40ns;
第2、3、4步为: $\text{Max}(3 \times 40\text{ns}, 200\text{ns}) = 200\text{ns}$;
(第2、3、4步都和存储器访问时间重叠)
第5、6、7步为: $3 \times 40\text{ns} = 120\text{ns}$ 。
总时间为360ns, 故最大带宽为4B/360ns=11.1MB/s

由此可知: 同步总线仅比异步快大约20%。要获得这样的速度, 异步总线上的设备和存储器必须足够快, 以便每次在40 ns内能完成一个子过程

io.20

2009年9月26日星期二

例2: 数据块大小对带宽的影响

假定有一个系统具有下列特性:

- (1) 系统支持4~16个32位字的块访问。
- (2) 64位同步总线, 时钟频率为200MHz, 每个64位数据传输需一个时钟周期, 地址发送到存储器需1个时钟周期。
- (3) 在每次总线操作(事务)间有两个空闲时钟周期。
- (4) 存储器访问时间对于开始的4个字是200ns, 随后每4个字是20ns。

假定读出数据的总线传送和随后4个字的存储器读操作可重叠进行
一个总线事务是由一个地址传送后跟一个数据块传送组成的

请求出分别用4-字块和16-字块方式读取256个字时的持续带宽和等待时间。并且求出两种情况下每秒钟内的有效总线事务数。

io.21

2009年9月26日星期二

举例-数据块大小对带宽的影响

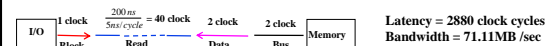
分析 4-字块传送情况:

对于4-字块传送方式, 一次总线事务由一个地址传送后跟一个4-字块的数据传送组成。也即每个总线事务传送一个4个字的数据块。

每个数据块所花时间为:

- (1) 发送一个地址到主存花1个时钟周期
- (2) 从主存读4个字花: $200\text{ns} / (5\text{ns}/\text{Cycle}) = 40$ 个时钟周期
(一个周期是 $10^9\text{ns} / 200\text{MHz} = 1000 / 200 = 5\text{ns}$)
- (3) 4个字 (128位) 的传输需2个时钟周期
(一个64位数据传输需1个时钟周期)
- (4) 在这次传送和下次之间有2个空闲时钟周期

所以一次总线事务总共需45个周期, 256个字需 $256/4=64$ 个事务, 所以整个传送需 $45 \times 64 = 2880$ 个时钟周期, 因而总等待时间为: $2880 \text{ 周期} \times 5\text{ns}/\text{周期} = 14400\text{ns}$ 。每秒钟的总线事务数为: $64 \times (1\text{s} / 14400\text{ns}) = 4.44\text{M}$ 个。总线带宽为: $(256 \times 4\text{B}) / 14400\text{ns} = 71.11\text{MB/s}$ 。



io.22

2009年9月26日星期二

举例-数据块大小对带宽的影响

分析 16-字块传送情况:

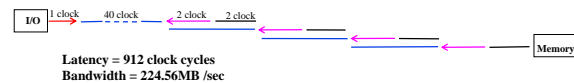
对于16-字块传送, 一次总线事务由一个地址传送后跟一个16-字块的数据传送组成。也即每个总线事务传送一个16个字的数据块。

第一个4-字所花时间为:

- (1) 发送一个地址到主存花1个时钟周期
- (2) 从主存读开始的4字花: $200\text{ns} / (5\text{ns}/\text{Cycle}) = 40$ 个时钟周期
- (3) 4个字需2个时钟周期, 在传输期间存储器开始读取下一个4字
- (4) 在本次和下次之间有2个空闲时钟, 此期间下一个4字已读完

所以, 16字中其余三个4字只要重复上述最后两步。因此对于16-字块传送, 一次总线事务共需花费的周期数为: $1 + 40 + 4 \times (2 + 2) = 57$ 个周期, 256个字需 $256 / 16 = 16$ 个事务, 因此整个传送需 $57 \times 16 = 912$ 个时钟周期。故总等待时间为: $912 \text{ 周期} \times 5\text{ns} / \text{周期} = 4560\text{ns}$ 。几乎仅是前者的1/3。每秒钟的总线事务个数为: $16 \times (1\text{s} / 4560\text{ns}) = 3.51\text{M}$ 个。总线带宽为: $(256 \times 4\text{B}) \times (1\text{s} / 4560\text{ns}) = 224.56\text{MB/s}$, 比前者高3.6倍。

由此可见, 大数据块传输的优势非常明显。



io.23

2009年9月26日星期二

增加同步总线带宽的措施

- 提高时钟频率
- Data bus width(增加数据线宽度)
 - 能同时传送更多位
 - Example: SPARCstation 20's memory bus 有 128 bit
 - Cost: more bus lines
- Block transfers(允许大数据块传送)
 - 背对背总线周期, 也称为突发(Burst)传输方式)
 - 只要开始送一次地址, 后面连续送数据
 - Cost: (a)增加复杂性
(b)延长响应时间
- Split Bus Transaction(拆分总线事务)
 - 一次总线事务时间延长, 但整个系统带宽增加
 - Cost: (a) 增加复杂性
(b) 延长响应时间
- 不采用分时复用方式
 - 地址和数据可以同时送出
 - Cost(代价): (a) more bus lines, (b) 增加复杂性

io.24

2009年9月26日星期二

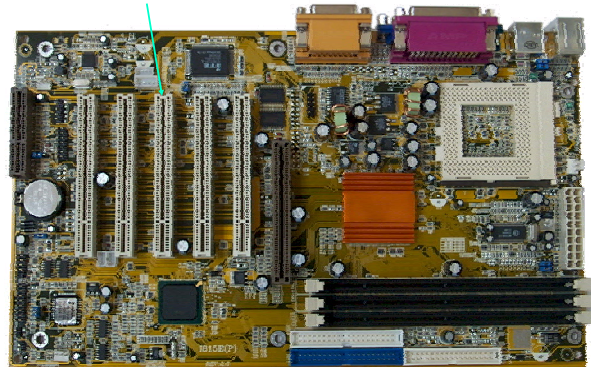
关于I/O总线标准

- ° I/O总线是各类I/O控制器与CPU、内存之间传输数据的一组公用信号线，这些信号线在物理上与主板扩展槽中插入的扩展卡（I/O控制器）直接连接。
- ° I/O总线是标准总线，I/O总线标准有：
 - ISA / EISA总线：（已逐步被淘汰）
 - Multibus总线：（已逐步被淘汰）
 - PCI总线：目前PC机所用的主流标准
 - PCI-Express(高速PCI总线)：目前PC机所用的主流标准
- ° I/O总线的带宽
 - 总线的数据传输速率(MB/s) =
数据位数/8 × 总线工作频率（MHz）× 每个总线周期的传输次数

10.25

2009年09月26日星期二

PCI总线扩展槽



10.26

2009年09月26日星期二

（自学）PCI总线标准

(1) 信号线

PCI有50根必须的信号线。按功能可分为以下几组：

- 系统信号：包括时钟和复位线。
- 地址和数据信号：包含32根分时复用的地址/数据线、4根分时复用的总线命令/字节使能线以及对这36根信号线进行奇偶校验的一根校验信号线。
- 接口控制信号：对总线事务进行定时控制，用于在事务的发起者和响应者之间进行协调。
- 裁决信号：它不同于其他信号，不是所有设备共享同一根信号线，而是每个总线主控设备都有一对仲裁线：总线请求和总线允许。PCI采用集中式裁决，所有设备的仲裁线都连接到一个总线裁决器中。
- 错误报告信号：用于报告奇偶校验错误以及其他错误。

10.27

2009年09月26日星期二

（自学）PCI总线标准

PCI的总线命令（事务类型）有：

- 中断响应：用于对PCI总线上的中断控制器提出的中断请求进行响应。地址线不起作用，在数据周期从中断控制器读取一个中断向量，此时C/BE信号线表示读取的中断向量的长度
- 特殊周期：用于总线主设备向一个或多个目标广播一条消息。
- I/O读和I/O写：I/O读/写命令用于在发起者和一个I/O控制器之间进行数据传输
- 存储器读、存储器行读、存储器多行读：用于总线主控设备从存储器中读取数据。PCI支持突发传送，所以它将占用一个或多个数据周期。这些命令的解释依赖于总线上的存储控制器是否支持PCI的高速缓存协议。如果支持的话，那么，与存储器之间的数据传送以Cache行的方式进行
- 存储器写、存储器写并无效：这两种存储器写命令用于总线主控设备向存储器写数据，它们将占用一个或多个数据周期。其中存储器写并无效命令用于回写Cache行到存储器，所以它必须保证至少有一个Cache行被写回
- 配置读、配置写：用于一个总线主控设备对连接到PCI总线上的设备中的配置参数进行读或更新。每个PCI设备都有一个寄存器组（最多可有256个寄存器），这个寄存器用于系统初始化时对本设备进行配置
- 双地址周期：由一个事务发起者用来表明它将使用64位地址来寻址

10.28

2009年09月26日星期二

（自学）PCI总线标准

PCI总线上存储器读命令的含义

读命令类型	支持 Cache 的内存	不支持 Cache 的内存
存储器读	突发传送半个或不到一个 Cache 行	突发传送两个数据周期或更少
存储器行读	突发传送半个以上到 3 个 Cache 行	突发传送 3 个到 12 个数据周期
存储器多行读	突发传送 3 个以上 Cache 行	突发传送 12 个以上数据周期

存储器读、存储器行读、存储器多行读：用于总线主控设备从存储器中读取数据。PCI支持突发传送，所以它将占用一个或多个数据周期。这些命令的解释依赖于总线上的存储控制器是否支持PCI的高速缓存协议。如果支持的话，那么，与存储器之间的数据传送以Cache行的方式进行

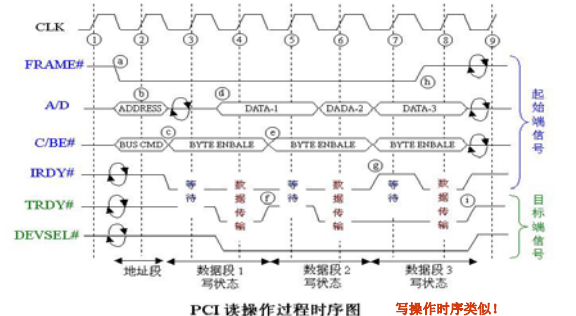
10.29

2009年09月26日星期二

（自学）PCI总线标准

(3) PCI数据传送过程

PCI总线上的数据传送由一个地址周期和一个或多个数据周期组成。所有事件在时钟下降沿(即在时钟周期中间)同步。总线设备在时钟上升沿采样总线信号

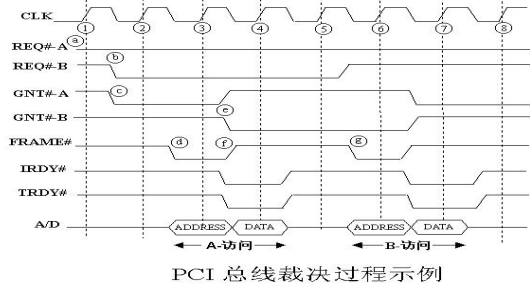


PCI读操作过程时序图 写操作时序类似！

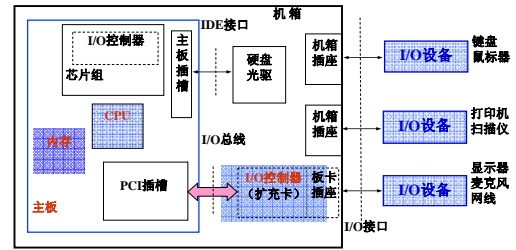
（自学）PCI总线标准

(4) PCI总线裁决

采用独立请求方式，有两个独立的裁决线：请求线REQ#和允许线GNT#。总线仲裁器可使用静态的固定优先级法、循环优先级法或先来先服务法等仲裁算法。采用隐式仲裁方式，在总线进行数据传送时进行总线仲裁，仲裁不会浪费总线周期。



回顾：I/O总线、I/O控制器、I/O接口与I/O设备的关系

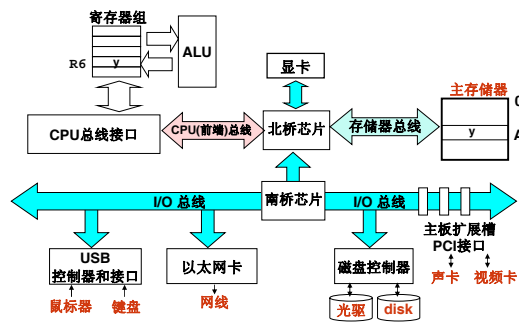


- I/O设备通常都是物理上相互独立的设备，它们一般通过I/O接口与I/O控制器连接
- I/O控制器通过扩展卡或者南桥芯片与I/O总线连接
- I/O总线经过北桥芯片与内存、CPU连接

io.32

2010年10月20日星期二

回顾：I/O总线、I/O控制器、I/O接口与I/O设备的关系



io.33

2010年10月20日星期二

本章小结

- 总线是共享的传输介质和传输控制部件，用于在部件或设备间传输数据
- 总线可能在芯片内、芯片之间、板卡之间和计算机系统之间连接
- I/O总线是I/O控制器与主机之间传输数据的一组公用信号线，它们在物理上与主板扩展槽中插入的扩展卡（I/O控制器）直接连接。
- 总线可以采用“同步”或“异步”方式进行定时。
 - 同步总线用“时钟”信号定时；异步总线用“握手信号”定时
 - 可以结合同步和异步方式进行半同步定时通信
 - 可以把一个总线事务分离成两个事务，在从设备准备数据时释放总线（总线事务分离方式）
- 总线的裁决：有集中和分布两类裁决方式
 - 分布裁决：自举裁决、冲突检测
 - 集中裁决：菊花链、独立请求并行判优
- 总线标准（PCI总线）
 - 总线互连结构
 - 单总线结构（早期计算机采用）
 - 多总线结构（现代计算机采用）

io.34

2010年10月20日星期二

第三次作业

- 8.29
- 8.34
- 8.39（假设条件与8.18相同改为与8.38相同）
- 8.40（假设条件与8.18相同改为与8.38相同）

io.35

2010年10月20日星期二