

第 3 章 输入输出系统

本章主要讨论输入输出系统的基本概念、磁盘阵列的分级、总线的定时控制和总线仲裁以及 I/O 系统中的通道处理机的输入输出过程、通道类型和流量设计。

3.0 学习指南

1. 知识点和学习要求

- 输入输出系统的基本概念。

了解 I/O 系统的三种工作方式和 I/O 处理机的两种形式。

- 磁盘阵列。

了解磁盘阵列的基本概念。

领会磁盘阵列的分级以及各个级别的特点。

- 总线设计。

领会总线的特点。

领会数据宽度和数据通路宽度概念的区别,了解 5 种数据宽度的适用场合。

领会同步和异步通信方式的通信过程、优缺点以及适用场合。

了解总线的仲裁方式,领会三种集中总线控制方式的总线分配过程、优缺点以及辅助控制总线线数。

- 通道处理机。

了解通道方式的输入、输出全过程。

掌握通道的三种类型的特点、所采用的数据宽度以及适用场合。

掌握通道和 I/O 系统的流量设计与分析。

掌握字节多路通道流量的计算,掌握通道工作周期的设计,能画出通道响应和处理各台外设请求时刻的时间关系图。

2. 重点与难点

本章的重点: 磁盘阵列分级;总线的特点和数据宽度;通道方式工作过程及通道类型;通道流量的分析和设计。

本章的难点: 三种集中总线控制方式的特点;通道的流量设计;画字节多路通道响应和处理各外设请求的时间关系图。

3.1 输入输出系统概述

在计算机系统中,通常把处理机与主存储器之外的部分统称为输入输出系统,它包括输入输出设备、输入输出接口和输入输出软件等。

输入输出系统是计算机系统中最具多样性和复杂性的部分。它的多样性表现在广泛涉及机、光、电、磁、声、自动控制等多种学科,仅从计算机学科来看,输入输出系统最典型地反映着硬件与软件的结合。而输入输出系统的复杂性隐藏在操作系统之中,一般用户只需通过简单的命令或程序调用就能使用各种输入输出设备,无须了解输入输出设备的具体工作细节。

输入输出系统的特点集中反映在异步性、实时性和与设备无关性这三项基本要求上,它们对输入输出系统的组织产生决定性的影响。

1) 异步性

输入输出设备的工作在很大程度上独立于处理机之外,通常不使用统一的中央时钟,各个设备按照自己的时钟工作,但又要在某些时刻接受处理机的控制。

2) 实时性

对于一般外部设备,处理机必须按照不同设备所要求的传送方式和传输速率不失时机地为设备提供服务,包括从设备接收数据、向设备发送数据及对设备的控制等。如果错过了服务的时机,就可能丢失数据或造成外设工作的错误。

用于实时控制的计算机系统对时间性的要求更强,如果处理机提供的服务不及时,很可能造成巨大的损失,甚至造成人身伤害。

对于计算机系统本身的硬件或软件错误,如电源故障、数据校验错、页面失效、非法指令、地址越界等,CPU 也必须及时处理。

3) 与设备无关性

计算机系统为了能够适应各种外设的不同要求,规定了一些独立于具体设备的标准接口。例如,串行接口、并行接口、SCSI 接口、USB 接口等。各种外设必须根据自己的特点和要求,选择其中的一种标准接口与计算机连接。凡是连接到同一种标准接口上的不同类型的设备,它们之间的差异必须由设备本身的控制器通过硬件和软件来进行填补。这样,处理机本身就无须了解各种外设特定的具体工作细节,可以采用统一的硬件和软件对品种繁多的设备进行管理。

3.2 磁盘阵列

磁盘阵列(RAID)具有容量大、速度快、可靠性高、造价低廉的特点,它是目前解决计算机 I/O 瓶颈的有效方法之一,有着广阔的发展前景。

3.2.1 RAID 简介

RAID 是由美国加州大学伯克利分校的 D. A. Patterson 教授在 1988 年提出的。RAID 是 Redundant Array of Inexpensive Disks 的缩写,直译为“廉价冗余磁盘阵列”,简称“磁盘阵列”。后来 RAID 中的字母 I 被改为 Independent,RAID 就成了“独立冗余磁盘阵列”,但这只是名称的变化,实质性的内容并没有改变。可以把 RAID 理解成一种使用磁盘驱动器的方法,它将一组磁盘驱动器用某种逻辑方式联系起来,作为逻辑上的一个磁

盘驱动器来使用。一般情况下,组成的逻辑磁盘驱动器的容量要小于各个磁盘驱动器容量的总和。

RAID 的优点如下。

(1) 成本低,功耗小,传输速率高。在 RAID 中,可以让很多磁盘驱动器同时传输数据,而这些磁盘驱动器在逻辑上又是一个磁盘驱动器,所以使用 RAID 可以达到单个磁盘驱动器几倍、几十倍甚至上百倍的速率。

(2) 提供容错功能。这是使用 RAID 的第二个原因,因为如果不考虑磁盘上的循环冗余校验(CRC)码,普通磁盘驱动器就无法提供容错功能。RAID 的容错是建立在每个磁盘驱动器的硬件容错功能之上的,所以它提供更高的安全性。

(3) RAID 与传统的大直径磁盘驱动器相比,在同样的容量下,价格要低许多。

3.2.2 RAID 的分级

RAID 可以分为 7 个级别,即 RAID0~RAID6,如表 3-1 所示。在 RAID1~RAID6 的几种方案中,不论何时磁盘损坏,都可以随时拔出损坏的磁盘,再插入好的磁盘(需要在硬件上支持热插拔),数据不会受损,失效磁盘的内容可以很快地重建,重建的工作由 RAID 硬件或 RAID 软件来完成。但 RAID0 不提供错误校验功能,所以有人说它不能算作 RAID,其实这也是 RAID0 为什么被称为 0 级 RAID 的原因——0 本身就代表“没有”。

表 3-1 RAID 的分级

RAID 级别	名 称	数据 磁盘数	可正常工作的 最多失效磁盘数	检测 磁盘数
RAID0	无冗余无校验的磁盘阵列	8	0	0
RAID1	镜像磁盘阵列	8	1	8
RAID2	纠错海明码磁盘阵列	8	1	4
RAID3	位交叉奇偶校验的磁盘阵列	8	1	1
RAID4	块交叉奇偶校验的磁盘阵列	8	1	1
RAID5	无独立校验盘的奇偶校验磁盘阵列	8	1	1
RAID6	双维无独立校验盘的奇偶校验磁盘阵列	8	2	2

RAID0~RAID6 的组织方式,如图 3-1 所示。图中备份盘和校验盘用灰底色表示。

RAID0 为无冗余、无检验的磁盘阵列,见图 3-1(a),数据以条带(strip)的形式存放在磁盘表面上。当从磁盘阵列中按顺序读取这些数据时,所有的磁盘都可以并行工作,各自读出相应的部分。由于 RAID0 不提供数据冗余,只要有一个磁盘出现故障,整个系统将无法正常工作。

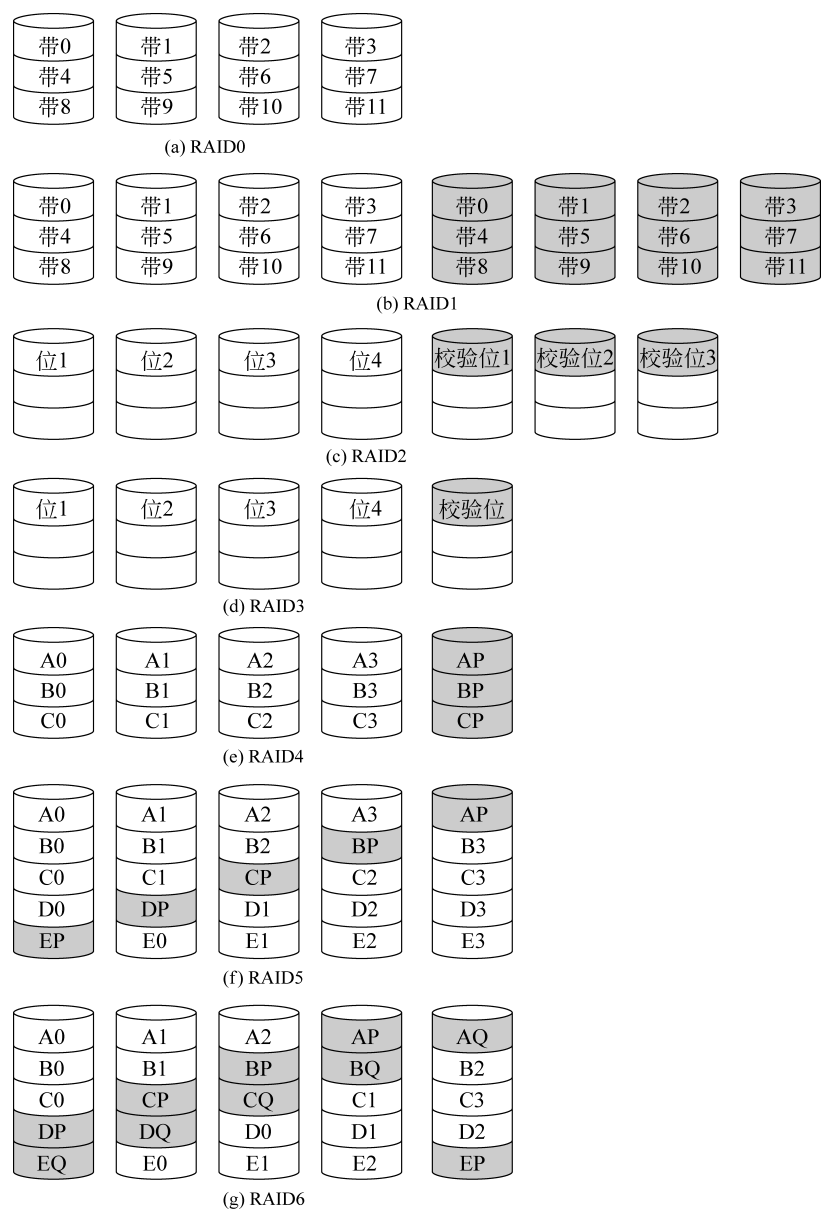


图 3-1 RAID0~RAID6

RAID1 称为镜像磁盘阵列,如图 3-1(b)所示。RAID1 在每次写入数据时,都会将数据复制到其他镜像盘上;当从该磁盘阵列中读取数据时,数据盘和镜像盘可独立同时工作,最后由最先读出数据的磁盘提供数据。如果某个磁盘出现故障,就由其镜像盘提供数据,系统仍能继续工作,只是降低了规格而已。

RAID2 为纠错海明码磁盘阵列,见图 3-1(c)。每个数据盘存放数据字的一位,还需要三个磁盘来存放海明检验位,图 3-1(c)中的每一行形成一个海明码。每当往数据盘写入数据时,就随之在校验盘上形成海明校验位。而当从数据盘上读出数据时,海明校验位

也被读出来,用于判断数据是否出错,如果出现了1位错误,则可以立即加以纠正。

RAID3 是 RAID2 的一个简化版本,称为位交叉奇偶检验磁盘阵列,如图 3-1(d)所示。图中的校验盘专门用于存放数据盘中相应数据的奇偶校验位。如果某个磁盘出故障,可以根据错误盘以外的所有其他盘中的正确信息恢复故障盘中的数据。

RAID4 是块交叉奇偶检验磁盘阵列,如图 3-1(e)所示。它采用比较大的条带,以块为单位进行交叉存储和计算奇偶检验。RAID4 的缺点是:它们必须访问同一个检验盘(只有一个),检验盘称为瓶颈。

RAID5 是无独立校验盘的奇偶校验磁盘阵列,又称为块分布奇偶检验磁盘阵列,如图 3-1(f)所示。这里每一行数据块的检验块被依次错开、循环地存放到不同的盘中,以达到均匀分布的目的。

RAID6 是在 RAID5 的基础上为了进一步加强数据保护而设计的一种 RAID 方式,实际上是一种扩展 RAID5 等级,如图 3-1(g)所示。与 RAID5 的不同之处在于,RAID6 增加了第二个独立的奇偶校验信息块。两个独立的奇偶系统使用不同的算法,数据的可靠性非常高,即使两块磁盘同时失效也不会影响数据的使用。但是,由于增加了一个校验,RAID6 需要分配给奇偶校验信息更大的磁盘空间,所以写入的效率较 RAID5 低,而且控制系统的设计也更为复杂,第二块的校验区也减少了有效存储空间。

目前理论上性能最高的 RAID 模式称为 RAID7,它具有全新的 RAID 架构,由于其自身就带有实时操作系统和用于存储管理的软件工具,可完全独立于主机运行,且不占用主机 CPU 资源,所以 RAID7 可以视为一种小型存储计算机,与其他 RAID 架构相比更为先进,但价格昂贵。

许多大型计算机系统并不局限于只使用一种类型的 RAID,可以使用多个 RAID 方案组合起来构建一种“新型”的 RAID。例如,可以将 RAID0 与 RAID1 组合起来,构成 RAID1+0 或者 RAID0+1。这两种 RAID 的区别在于组合方式上,RAID1+0 是先组织成镜像的 RAID1,再将两个 RAID1 组织成扩展容量的 RAID0,RAID0+1 则反之。4 个磁盘的两种复合 RAID 如图 3-2 所示,其中磁盘的并联表示数据镜像关系,磁盘的串联表示容量扩展关系。RAID1+0 或 RAID0+1 都至少使用 4 个硬盘。RAID0+1 比 RAID1+0 有着更快的读写速度,但可靠性上 RAID0+1 不及 RAID1+0,这是因为当 RAID1+0 有一个硬盘受损,其余三个硬盘会继续运作,而 RAID0+1 只要有一个硬盘受损,同组 RAID0 的另一个硬盘也会停止运作,只剩下两个硬盘运作。因此,RAID1+0 远比 RAID0+1 常用。

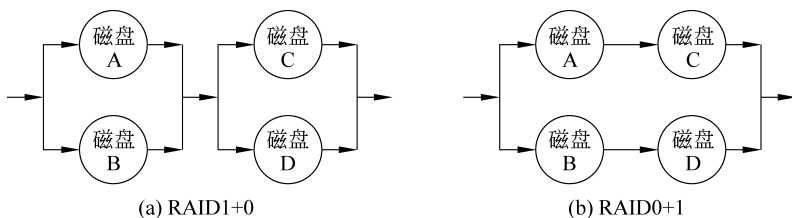


图 3-2 两种复合 RAID 的结构

3.3 总线设计

在大多数小型和微型计算机系统中,计算机的各子系统之间通过总线实现连接。

3.3.1 总线特点

总线(Bus)是一组能为多个部件分时共享的公共信息传送线路。共享是指总线上可以挂接多个部件,各个部件之间相互交换的信息都可以通过这组公共线路传送;分时是指同一时刻总线上只能传送一个部件发送的信息。总线的优点:成本低、简单。总线的缺点:总线的带宽形成了信息交换的瓶颈,从而限制了系统中总的 I/O 吞吐率。

1. 总线事务

通常把在总线上一对设备之间的一次信息交换过程称为一个“总线事务”,把发出总线事务请求的部件称为主设备,与主设备进行信息交换的对象称为从设备。例如,CPU 要求读取存储器中某单元的数据,则 CPU 是主设备,而存储器是从设备。总线事务类型通常根据它的操作性质来定义,典型的总线事务类型有“存储器读”“存储器写”“I/O 读”“I/O 写”“中断响应”等,一次总线事务简单来说包括两个阶段:地址阶段和数据阶段。

突发传送事务由一个地址阶段和多个数据阶段构成,用于传送多个连续单元的数据,地址阶段送出的是连续区域的首地址。因此,一次突发传送事务可以传送多个数据。

2. 总线使用权

总线是由多个部件和设备所共享的,为了正确地实现它们之间的通信,必须有一个总线控制机构,对总线的使用进行合理的分配和管理。

主设备发出总线请求并获得总线使用权后,就立即开始向从设备进行一次信息传送。这种以主设备为参考点,向从设备发送信息或接收从设备送来的信息的工作关系,称为主从关系。主设备负责控制和支配总线,向从设备发出命令来指定数据传送方式与数据传送地址信息。各设备之间的主从关系不是固定不变的,只有获得总线使用权的设备才是主设备,如 CPU 等。但主存总是从设备,因为它不会主动提出要与谁交换信息的要求。

在定义总线数据传送操作是“输入”或“输出”时,必须以主设备为参考点,即从设备将数据送往主设备称为“输入”;反之称为“输出”。

通常,将完成一次总线操作的时间称为总线周期。总线使用权的转让发生在总线进行一次数据传送的结束时刻。在一个总线周期开始时,对 CPU 或 I/O 设备的请求进行取样,并在这个总线周期进行数据传送的同时也进行判优,选择下一总线周期谁能获得总线使用权,然后在本周期结束时实现总线使用权的转移,开始新的总线周期。

3.3.2 总线的数据宽度

数据宽度是 I/O 设备取得 I/O 总线后所传送数据的总量,它不同于前述的数据通路宽度。数据通路宽度是数据总线的物理宽度,也就是数据总线的线数。而两次分配总线期间所传送的数据宽度可能要经过多个时钟周期分次传送才能完成。数据宽度有单字(单字节)、定长块、变长块、单字加定长块和单字加变长块等。

单字(单字节)宽度适合于低速设备。因为这些设备在每次传送一个字(字节)后的访问等待时间很长,在这段时间里让总线释放出来为别的设备服务,可大大提高总线利用率和系统效率。采用单字(单字节)宽度不用指明传送信息的长度,有利于减少辅助开销。

定长块宽度适合于高速设备,可以充分利用总线带宽。定长块也不用指明传送信息的长度,简化了控制。但由于块的大小固定,当它要比实际传送的信息块小得多时,仍要多次分配总线;而如果大于要传送的信息块,又会浪费总线带宽和缓冲器空间,也使得部件不能及时转入别的操作。

变长块宽度适合于高优先级的中高速设备,灵活性好,可按设备的特点动态地改变传送块的大小,使之与部件的物理或逻辑信息块的大小一致,以有效地利用总线的带宽,也使通信的部件能全速工作。但为此要增大缓冲器空间和增加指明传送信息块大小的辅助开销和控制。

单字加定长块宽度适合于速度较低而优先级较高的设备。这样,定长块的大小就不必选择过大,信息块超过定长块的部分可用单字处理,从而减少总线带宽、部件的缓冲器空间,减少部件可用能力的浪费。不过,若传送的信息块小于定长块的大小,但字数又不少时,设备或总线的利用率会降低。

单字加变长块宽度是一种灵活有效但复杂、费用高的方法。当要求传送单字时比只能成块传送的方法节省了不少起始辅助操作;而当成块传送时,块的大小又能调整到与部件和应用的要求相适应,从而优化了总线的使用。

3.3.3 总线定时控制

主机与外设通过总线进行信息交换时,必然存在时间上的配合和动作的协调问题,否则系统的工作将出现混乱。总线的定时控制方式一般分为同步定时方式和异步定时方式。

1. 同步定时方式

所谓同步定时方式,是指系统采用一个统一的时钟信号来协调发送和接收双方的传送定时关系。时钟产生相等的时间间隔,每个间隔构成一个总线周期。在一个总线周期中,发送和接收双方可以进行一次数据传送。由于是在规定的时间段内进行 I/O 操作的,所以发送者不必等待接收者有什么响应,当这个时间段结束后,就自动进行下一个操作。

同步定时方式中的时钟频率必须能适应在总线上最长的延迟和最慢的接口的需要。因此,同步定时方式的效率较低,时间利用也不够合理;同时,也没有办法知道被访问的外设是否已经真正地响应,故可靠性比较低。

2. 异步定时方式

异步定时方式也称为应答方式。在这种方式下,没有公用的时钟,也没有固定的时间间隔,完全依靠传送双方相互制约的“握手”信号来实现定时控制。

通常,把交换信息的两个部件或设备分为主设备和从设备,主设备提出交换信息的“请求”信号,经接口传送到从设备;从设备接到主设备的申请后,通过接口向主设备发出“回答”信号,整个“握手”过程就是一问一答地进行的。必须指出,从“请求”到“回答”的时间是由操作的实际时间决定的,而不是由 CPU 的节拍硬性规定的,所以具有很强的灵活性,而且对提高整个计算机系统的工作效率也是有好处的。

异步控制能保证两个工作速度相差很大的部件或设备间可靠地进行信息交换,自动完成时间的配合。但是,控制较同步方式稍复杂一些,成本也会高一些。

根据“请求”和“回答”信号的撤销是否互锁,异步定时方式有以下三种情况。

1) 不互锁

“请求”和“回答”信号都有一定的时间宽度,“请求”信号的结束和“回答”信号的结束不互锁,如图 3-3(a)所示。

2) 半互锁

“请求”信号的撤销取决于接收到“回答”信号,而“回答”的撤销由从设备自己决定,如图 3-3(b)所示。

3) 全互锁

“请求”信号的撤销取决于“回答”信号的来到,而“请求”信号的撤销又导致“回答”信号的撤销,如图 3-3(c)所示。全互锁方式给出了最高的灵活性和可靠性,当然也付出了增加接口电路复杂性的代价。

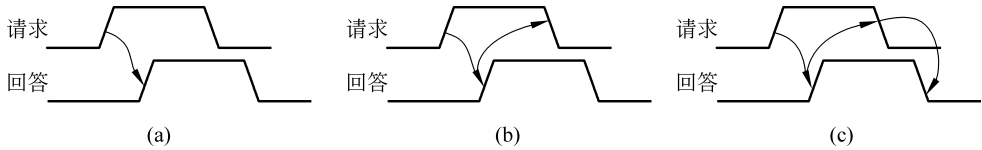


图 3-3 “请求”与“回答”信号的互锁

3.3.4 总线的集中仲裁方式

为了保证同一时刻只有一个申请者使用总线,总线控制机构中设置有总线判优和仲裁控制逻辑,即按照一定的优先次序来决定哪个部件首先使用总线,只有获得总线使用权的部件,才能开始数据传送。总线控制逻辑集中在一处(如在 CPU 中)的,称为集中式控制,就集中式控制而言,有三种常见的优先权仲裁方式:链式查询方式、计数器定时查询

方式和独立请求方式。

1. 链式查询方式

链式查询方式如图 3-4 所示,总线控制器使用三根控制线与所有部件和设备相连,而 AB 和 DB 分别代表地址总线和数据总线。三个控制信号如下。

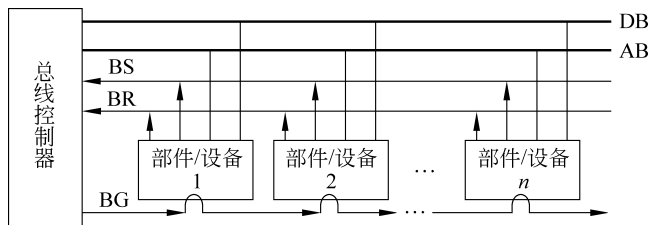


图 3-4 链式查询方式

- (1) 总线请求(BR): 该线有效,表示至少有一个部件或设备要求使用总线。
- (2) 总线忙(BS): 该线有效,表示总线正在被某部件或设备使用。
- (3) 总线批准(BG): 该线有效,表示总线控制器响应总线请求。

与总线相连的所有部件经公共的 BR 线发出总线请求,只有在 BS 信号未建立前,BR 才能被总线控制器响应,并送出 BG 回答信号。BG 信号串行地通过每个部件,如果某个部件本身没有总线请求,则将该信号传给下一个部件,如果这个部件有总线请求,就停止传送 BG 信号,获得总线使用权。这时该部件将建立 BS 信号,表示它占用了总线,并撤销 BR 信号,进行数据的传送。BS 信号在数据传送完后撤销,BG 信号也随之撤销。

显然,链式查询方式的优先次序是由 BG 线上串接部件的先后位置来确定的,在查询链中离总线控制器最近的设备具有最高优先权。

链式查询的优点: 只用很少几根线就能按一定的优先次序来实现总线控制,并很容易扩充。链式查询的缺点: 对查询链的故障很敏感,如果第 i 个部件中的查询链电路有故障,那么第 i 个以后的部件都不能工作。另外,因为查询的优先级是固定的,所以若优先级较高的部件出现频繁的总线请求时,优先级较低的部件就可能难以得到响应。

2. 计数器定时查询方式

计数器定时查询方式如图 3-5 所示。总线上的每个部件可以通过公共的 BR 线发出请求,总线控制器收到请求之后,在 BS 为“0”的情况下,让计数器开始计数,定时地查询各个部件以确定是谁发出的请求。当查询线上的计数值与发出请求的部件号一致时,该部件就使 BS 线置“1”,获得了总线使用权,并中止计数查询,直至该部件完成数据传送之后,撤销 BS 信号。

这种计数可以从“0”开始,也可以从中止点开始。如果从“0”开始,各部件的优先次序和链式查询方式相同,优先级的次序是固定的。如果从中止点开始,即为循环优先级,各个部件使用总线的机会将相等。计数器的初始值还可以由程序来设置,这就可以方便地改变优先次序,增加系统的灵活性。定时查询方式的控制线数较多,对于 n 个部件,共需

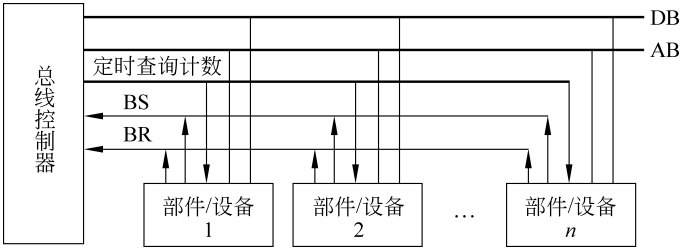


图 3-5 计数器定时查询方式

$2 + \lceil \log_2 n \rceil$ 根控制线。

3. 独立请求方式

图 3-6 表示独立请求方式。在这种方式中，每一个共享总线的部件均有一对控制线：总线请求(BR_i)和总线批准(BG_i)。当某个部件请求使用总线时，便发出 BR_i ，总线控制器中有一排队电路，根据一定的优先次序决定首先响应哪个部件的请求 BR_i ，然后给该部件送回批准信号 BG_i 。

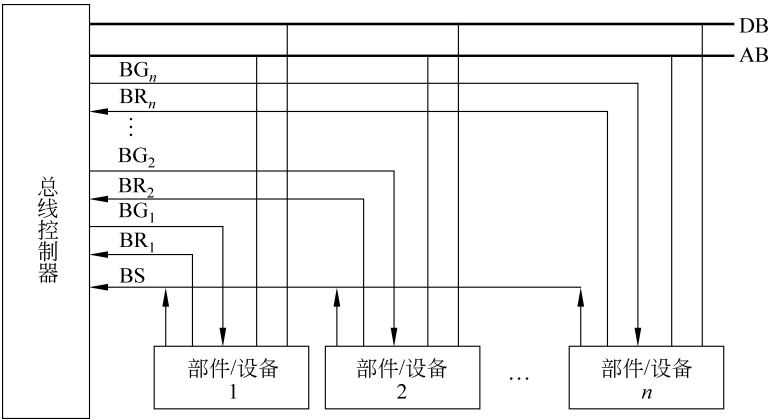


图 3-6 独立请求方式

独立请求方式的优点是响应时间快，然而这是以增加控制线数和硬件电路为代价的，对于 n 个部件，控制线的数目将达 $2n + 1$ 根。此方式对优先次序的控制也是相当灵活的，它可以预先固定，也可以通过程序来改变优先次序。

3.3.5 总线的分布仲裁方式

分布仲裁方式不需要中央仲裁器，即总线控制逻辑分散在连接于总线上的各个部件或设备中。连接到总线上的主方可以启动一个总线周期，而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使用权，但同一时间里可以有一个或多个从方。常见的分布式仲裁方式有三种：自举分布式、冲突检测分布式和并行竞争分布式。

1. 自举分布式

每个设备的优先级固定,需要请求总线控制权的设备在各自对应的总线请求线上送出请求信号。在总线仲裁期间,每个设备通过取回的信息能够检测出其他比自己优先级高的设备是否发出了总线请求,如果没有,则立即使用总线,并通过总线忙信号阻止其他设备使用总线;如果一个设备在发出总线请求的同时,检测到其他优先级更高的设备也请求使用总线,则本设备不能马上使用总线。也就是说,只有在查看到所有优先级比自己高的设备没有请求时,才能使用总线。

2. 冲突检测分布式

当某个设备要使用总线时,首先检查是否有其他设备正在使用总线,如果没有,则它就置总线忙,并直接使用总线。若两个设备同时检测到总线空闲,那它们可能都会立即使用总线,从而发生冲突。因此,每个设备在使用过程中,会侦听总线以检测是否发生冲突,当发生冲突时,两个设备都会停止传输,延迟一个随机时间之后再重新使用总线,以避免冲突。这种方案一般用在网络通信总线上,Ethernet 就是使用该方案进行总线裁决的。

3. 并行竞争分布式

这是一种较复杂但有效的裁决方案。其基本思想是:总线上的每个设备都有一个唯一的仲裁号,需要使用总线的主控设备把自己的仲裁号发送到仲裁线上,这个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时间段后占用总线还是撤销仲裁号。

3.4 通道处理机

在大型计算机系统中,所连接的 I/O 设备数量多,输入输出频繁,要求整体的速度快,单纯依靠主 CPU 采取程序查询、程序中断和 DMA 等控制方式已不能满足要求,于是计算机系统引入了通道控制方式。

3.4.1 通道的作用和功能

1. 通道的作用

在大型计算机系统中,如果仅采用程序查询、程序中断和 DMA 三种基本的输入输出方式来管理设备,将会带来以下两个问题。

1) CPU 负担重

对于程序查询方式,所有外设的输入输出工作均由 CPU 承担,不能充分发挥 CPU 的计算能力;对于程序中断方式,每传送一个字符都由 CPU 执行一段程序来完成(适用于低速外部设备);对于 DMA 方式,初始化、前处理和后处理等工作需要 CPU 来完成(适

用于高速外围设备)。

2) 接口数量多

大型机中的外围设备台数很多,其种类、工作方式和速度都有比较大的差别,但一般并不同时工作。如果为每一台设备都配置一个接口,不仅接口的数量很多,而且接口的利用率低。特别是 DMA 控制器,其硬件成本很高。

为了把对外设的管理工作从 CPU 中分离出来,使 CPU 摆脱繁重的输入输出负担,从 IBM 360 系列机开始,普遍采用了通道技术。

2. 通道的主要功能

一般来说,通道的功能包括:

(1) 接受 CPU 发来的 I/O 指令,并根据指令要求选择一台指定的外围设备与通道相连。

(2) 执行通道程序,即从主存中逐条取出通道指令,对通道指令进行译码,并根据需要向被选中的设备控制器发出各种操作命令。

(3) 给出外部设备的有关地址。例如,给出磁盘存储器的柱面号、磁头号、扇区号等。

(4) 给出主存缓冲区的首地址。这个缓冲区用来暂时存放从外设输入的数据或将要输出到外设的数据。

(5) 控制外设与主存缓冲区之间的数据交换的个数。

(6) 指定传送工作结束时要进行的操作。例如,将外设的中断请求和通道的中断请求送往 CPU 等。

(7) 检查外设的工作状态是否正常,并将该状态信息送往主存指定单元保存。

(8) 在数据传输过程中完成必要的格式变换。

3. 通道的硬件组成

通道的硬件包括相关的寄存器和控制逻辑。主要的寄存器有数据缓冲寄存器、主存地址计数器、传输字节数计数器、通道命令字寄存器、通道状态字寄存器、通道地址字寄存器等,其中通道命令字寄存器(CCWR)用来存放通道命令字(CCW)。CCW 是控制 I/O 操作的关键参数,一条条的通道命令字(通道指令)构成通道程序放在主存中。通道地址字寄存器(CAWR)指出了通道程序在主存中的起始地址,工作时通道就依照这个地址到主存中取出 CCW 并加以执行。通道状态字寄存器(CSW)记录了通道程序执行后本通道和相应设备的各种状态信息,这些信息称为通道状态字(CSW)。CSW 通常放在主存的固定单元中,此专用单元的内容在执行下一个 I/O 指令或中断之前是有效的,可供 CPU 了解通道、设备状态和操作结束的原因。

通道对外设的控制通过输入输出接口和设备控制器进行。通道与设备控制器之间一般采用标准的输入输出接口来连接。通道通过标准接口把操作命令送到设备控制器,设备控制器解释执行这些通道指令,完成命令指定的操作。设备控制器能够记录外设的状态,并把状态信息送往通道和 CPU。

3.4.2 通道工作过程

用户通过调用通道完成一次数据传输的过程如图 3-7 所示,CPU 执行用户程序和管理程序,通道执行通道程序的时间关系如图 3-8 所示。

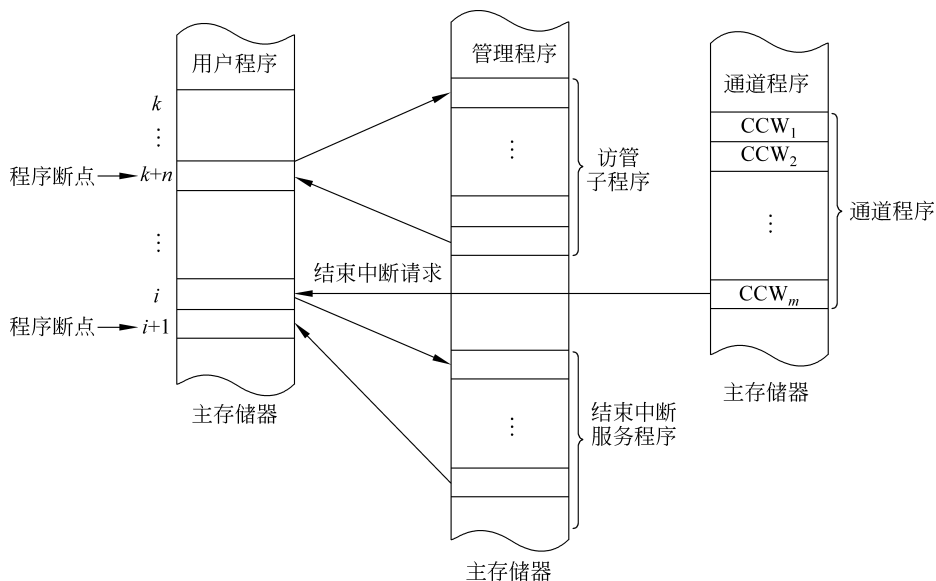


图 3-7 通道完成一次数据传输的过程

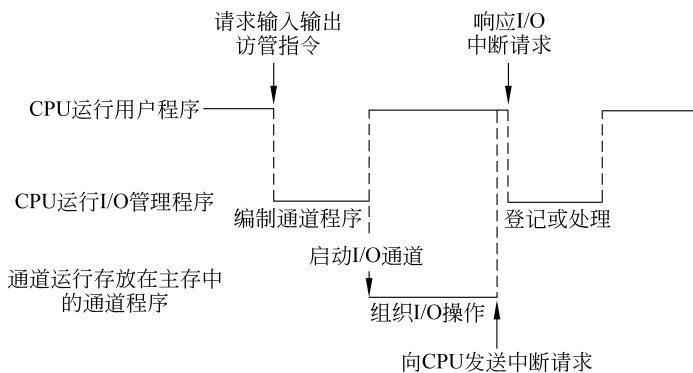


图 3-8 通道程序、管理程序 and 用户程序的执行时间关系

主要过程分为以下三步。

- (1) 在用户程序中使用访管指令进入管理程序,由 CPU 通过管理程序组织一个通道程序,并启动通道。
- (2) 通道执行 CPU 为它组织的通道程序,完成指定的数据输入输出工作。
- (3) 通道程序结束后向 CPU 发中断请求。CPU 响应这个中断请求后,第二次调用管理程序对中断请求进行处理。

需要指出的是,CPU 进行输入输出操作时,在用户程序中使用访管指令(地址为 K)迫使 CPU 由用户程序(目态)进入管理程序(管态),访管指令是一条广义指令,它除去给出访管子程序的入口地址外,还给出如设备号、交换长度、主存起始地址等参数,以便管理程序编制通道程序。管理程序根据访管指令给定的参数编写通道程序,写入主存的一片区域中,并将其首地址置入通道地址字(CAW)中,然后便可启动该通道开始工作,CPU 返回用户程序的断点 $K+n$ 继续工作。从此时开始,CPU 与通道处于并行工作状态。通道从 CAW 中获得通道程序的入口地址,逐条取出通道指令并执行它,待通道程序执行完毕后可向 CPU 发出中断请求,CPU 响应该中断请求,再次进入管理程序进行结束处理,本次输入输出操作完成。从图 3-8 中可以看出,整个输入输出操作的过程是在通道控制下完成的,而通道的控制是通过执行通道程序实现的。

这样,每完成一次输入输出工作,CPU 只需要两次调用管理程序,大大减少了对用户程序的打扰。

3.4.3 通道的类型

按照输入输出信息的传送方式,通道可分为三种类型,即字节多路通道、选择通道和数组多路通道。

1. 字节多路通道

字节多路通道是一种简单的共享通道,用于连接与管理多台低速设备,以字节交叉方式传送信息,其传送方式如图 3-9 所示。字节多路通道先选择设备 A,为其传送一个字节 A_1 ;然后选择设备 B,传送字节 B_1 ;再选择设备 C,传送字节 C_1 。后续通道再交叉地传送 $A_2, B_2, C_2, \dots$ 所以字节多路通道的功能好比一个多路开关,交叉(轮流)地接通各台设备。

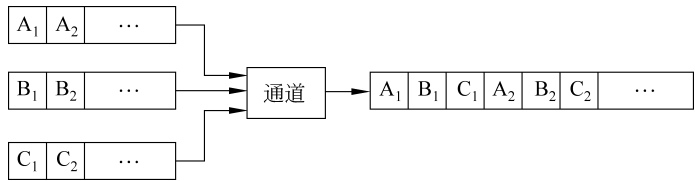


图 3-9 字节多路通道传送方式示意图

一个字节多路通道,包括多个按字节方式传送信息的子通道。每个子通道服务于一个设备控制器,每个子通道都可以独立地执行通道程序。各个子通道可以并行工作,但是,所有子通道的控制部分是公共的,各个子通道可以分时地使用。

通道不间断地、轮流地启动每个设备控制器,当通道为一个设备传送完一个字节后,就转去为另一个设备服务。当通道为某一设备传送时,其他设备可以并行地工作,准备需要传送的数据字节或处理收到的数据字节。这种轮流服务是建立在主机的速度比外设的速度高得多的基础之上的,它可以提高系统的工作效率。

2. 选择通道

对于高速设备,字节多路通道显然是不合适的。选择通道又称高速通道,在物理上它也可以连接多个设备,但这些设备不能同时工作,在一段时间内通道只能选择一台设备进行数据传送,此时该设备可以独占整个通道。因此,选择通道一次只能执行一个通道程序,只有当它与主存交换完信息后,才能再选择另一台外部设备并执行该设备的通道程序。如图 3-10 所示,选择通道先选择设备 A,成组连续地传送 $A_1 A_2 \dots$ 。当设备 A 传送完毕后,选择通道又选择设备 B,成组连续地传送 $B_1 B_2 \dots$ 。再选择设备 C,成组连续地传送 $C_1 C_2 \dots$ 。

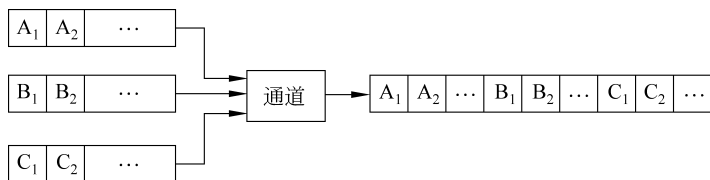


图 3-10 选择通道传送方式示意图

每个选择通道只有一个以成组方式工作的子通道,逐个为多台高速外围设备服务。选择通道主要用于连接高速外设,如磁盘、磁带等,信息以成组方式高速传送。但是,在数据传送过程中还有一些辅助操作(如磁盘机的寻道等),此时会使通道处于等待状态,所以虽然选择通道具有很高的数据传输速率,但整个通道的利用率并不高。

3. 数组多路通道

数组多路通道是把字节多路通道和选择通道的特点结合起来的一种通道结构。它的基本思想是:当某设备进行数据传送时,通道只为该设备服务;当设备在执行辅助操作时,通道暂时断开与这个设备的连接,挂起该设备的通道程序,去为其他设备服务。

数组多路通道有多个子通道,既可以执行多路通道程序,即像字节多路通道那样,所有子通道分时共享总通道,又可以用选择通道那样的方式成组地传送数据;既具有多路并行操作的能力,又具有很高的数据传输速率,使通道的效率充分得到发挥。

选择通道和数组多路通道都适用于连接高速外设,但前者的数据宽度是不定长的数据块,后者的数据宽度是定长的数据块。三种类型通道的比较见表 3-2。

表 3-2 三种类型通道的比较

性 能	字节多路通道	选 择 通 道	数组多路通道
数据宽度	单字节	不定长块	定长块
适用范围	大量低速设备	优先级高的高速设备	大量高速设备
工作方式	字节交叉	独占通道	成组交叉
共享性	分时共享	独占	分时共享
选择设备次数	多次	一次	多次

三种类型的通道组织在一起,可配置若干台不同种类、不同速度的 I/O 设备,使计算机的 I/O 组织更合理、功能更完善、管理更方便。

通道在单位时间内传送的位数或字节数,称为通道的数据传输率或流量,它标志着计算机系统系统中的系统吞吐率,也表明了通道对外设的控制能力和效率。在单位时间内允许传送的最大字节数或位数,称为通道的最大数据传输率或通道极限流量,它是设计通道的最大依据。

字节多路通道的实际流量是该通道上所有设备的数据传输率之和。而选择通道和数组多路通道由于在一段时间内只能为一台设备传送数据,此时的通道流量就等于这台设备的数据传输率,因此,这两种通道的实际流量等于连接在这个通道上的所有设备中流量最大的那一个。

3.4.4 通道中的数据传送过程

一个字节多路通路是分时为多台低速和中速外设服务的,在有 P 台设备同时连接到一个字节多路通道上时,它们的数据传送过程如图 3-11(a)所示。

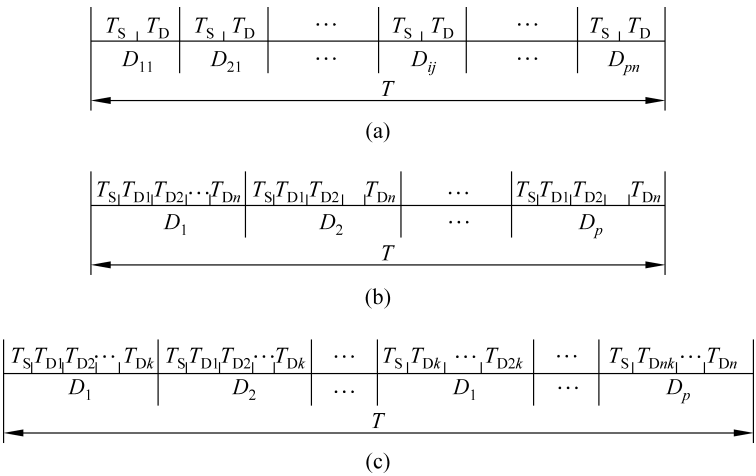


图 3-11 通道的数据传送过程

在图 3-11(a)中每一个参数的含义如下。

T_S : 设备选择时间。从通道需要设备发出数据传送请求开始,到通道实际为这台设备传送数据所需要的时间。

T_D : 传送一字节数据所用的时间,实际上就是通道执行一条通道指令,即数据传送指令所用的时间。

P : 在一个通道上连接的设备台数,且这些设备同时都在工作。

n : 每一个设备传送的字节数。这里,假设每一台设备传送的字节数都相同,都是 n 字节。

D_{ij} : 连接在通道上的第 i 台设备传送的第 j 个数据, 其中 $i = 1, 2, \dots, p; j = 1, 2, \dots, n$ 。

T : 通道完成全部数据传送工作所需要的时间。

在字节多路通道中, 通道每连接一台外设, 只传送一字节, 然后又与另一台设备相连, 因此, T_s 和 T_D 是间隔进行的。

当一字节多路通道上连接有 P 台设备, 每台设备都传送 n 字节时, 所需要的时间为

$$T_{\text{Byte}} = (T_s + T_D) \times P \times n$$

选择通道在一段时间内只能单独为一台高速外设服务, 当这台设备的数据传送工作全部完成后, 通道才能为另一台设备服务。选择通道的数据传送过程如图 3-11(b) 所示, 图中除去与字节多路通路相同的参数以外, 还有以下参数。

T_{Di} : 通道传送第 i 个数据所用的时间, 其中 $i = 1, 2, \dots, n$ 。

D_i : 通道正在为第 i 台设备服务, 其中 $i = 1, 2, \dots, p$ 。

在选择通道中, 通道每连接一台外设, 就把这个设备的 n 字节全部传送完成, 然后再与另一台设备相连接, 因此, 在一个 T_s 之后, 有连续 n 个数据传送时间 T_D 。

当一个选择通道连接 P 台设备, 每台设备都传送 n 字节时, 所需要的时间为

$$T_{\text{select}} = \left(\frac{T_s}{n} + T_D \right) \times P \times n$$

数组多路通道在一段时间内只能为一台高速设备传送数据, 但同时可以有多台高速设备在寻址, 包括定位和找扇区。数组多路通道的数据传送过程如图 3-11(c) 所示, 图中所用参数与前两种类型相同, 另外还有以下参数。

k : 一个数据块中的字节数。在一般情况下, $k < n$ 。

数据多路通道每连接一台高速设备, 一般传送一个数据块, 传送完成后, 又与另一台高速设备连接, 再传送一个数据块, 因此, 在一个 T_s 之后, 连续 k 个数据的传送时间 T_D 。

当一台数组多路通道连接 P 台设备, 每台设备都传送 n 字节时, 所需要的时间为

$$T_{\text{block}} = \left(\frac{T_s}{k} + T_D \right) \times P \times n$$

3.4.5 通道的流量分析

通道流量是指通道在数据传送期内, 单位时间里传送的字节数。它能达到的最大流量称为通道极限流量。

假设通道选择一次设备的时间为 T_s , 每传送一个字节的时间为 T_D , 通道工作时的极限流量分别如下。

(1) 字节多路通道的极限流量为

$$f_{\max \cdot \text{Byte}} = \frac{P \times n}{(T_s + T_D) \times P \times n} = \frac{1}{T_s + T_D}$$

每选择一台设备只传送一字节。

(2) 选择通道的极限流量为

$$f_{\max \cdot \text{select}} = \frac{P \times n}{\left(\frac{T_s}{n} + T_D\right) \times P \times n} = \frac{1}{\frac{T_s}{n} + T_D} = \frac{n}{T_s + nT_D}$$

每选择一台设备就把 n 字节全部传送完。

(3) 数组多路通道的极限流量为

$$f_{\max \cdot \text{block}} = \frac{P \times n}{\left(\frac{T_s}{k} + T_D\right) \times P \times n} = \frac{1}{\frac{T_s}{k} + T_D} = \frac{k}{T_s + kT_D}$$

每选择一台设备传送定长 k 字节。

若通道上接 P 台设备,则通道要求的实际流量分别如下。

(1) 字节多路通道的实际流量为

$$f_{\text{byte}} = \sum_{i=1}^P f_i$$

即所接 P 台设备的速率之和。

(2) 选择通道的实际流量为

$$f_{\text{select}} = \max_{i=1}^P f_i$$

即所接 P 台设备中最高速率。

(3) 数组多路通道的实际流量为

$$f_{\text{block}} = \max_{i=1}^P f_i$$

即所接 P 台设备中最高速率。

为使通道所接外部设备在满负荷工作时仍不丢失信息,应使通道的实际最大流量不能超过通道的极限流量。

如果在 I/O 系统中有多通道,各个通道是并行工作的,则 I/O 系统的极限流量应当是各通道或各子通道工作时的极限流量之和。

例 3-1 一个字节多路通道连接 D_1 、 D_2 、 D_3 、 D_4 、 D_5 共 5 台设备,这些设备分别每 $10\mu\text{s}$ 、 $30\mu\text{s}$ 、 $30\mu\text{s}$ 、 $50\mu\text{s}$ 和 $75\mu\text{s}$ 向通道发出一次数据传送的服务请求,请回答下列问题:

(1) 计算这个字节多路通道的实际流量和工作周期。

(2) 如果设计字节多路通道的最大流量正好等于通道的实际流量,并假设对数据传输率高的设备,通道响应它的数据传送请求的优先级也高。5 台设备在 0 时刻同时向通道发出第一次传送数据的请求,并在以后的时间里按照各自的数据传输率连续工作。画出通道分时为每台设备服务的时间关系图,并计算这个字节多路通道处理完各台设备的第一次数据传送请求的时刻。

(3) 从时间关系图上可以发现什么问题? 如何解决这个问题?

解: (1) 这个字节多路通道的实际流量为

$$f_{\text{Byte}} = \left(\frac{1}{10} + \frac{1}{30} + \frac{1}{30} + \frac{1}{50} + \frac{1}{75} \right) \text{MB/s} = 0.2 \text{MB/s}$$

通道的工作周期为

$$T = \frac{1}{f_{\text{Byte}}} = 5 \mu\text{s}$$

包括设备选择时间 T_s 和传送一字节的时间 T_D 。

(2) 5 台设备向通道请求传送和通道为它们服务的时间关系如图 3-12 所示, 向上的箭头表示设备的数据传送请求, 有阴影的长方形表示通道响应设备的请求并为设备服务所用的工作周期。

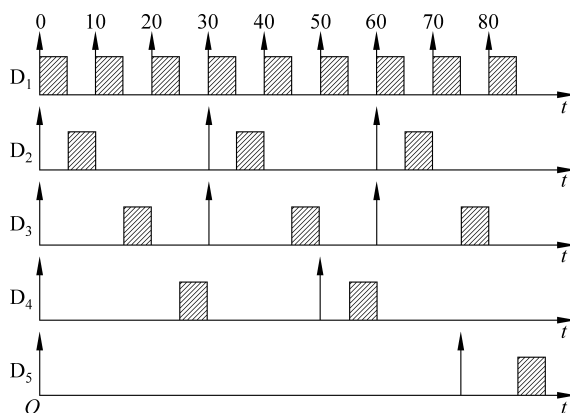


图 3-12 字节多路通道响应设备请求和为设备服务的时间关系图

在图 3-12 中, 5 台设备在 0 时刻同时向字节多路通道发出第一次传送时间的请求, 通道处理完各设备第一次请求的时间分别是:

处理完设备 D_1 的第一次请求的时刻为 $5 \mu\text{s}$ 。

处理完设备 D_2 的第一次请求的时刻为 $10 \mu\text{s}$ 。

处理完设备 D_3 的第一次请求的时刻为 $20 \mu\text{s}$ 。

处理完设备 D_4 的第一次请求的时刻为 $30 \mu\text{s}$ 。

设备 D_5 的第一次请求没有得到通道的响应, 直到第 $85 \mu\text{s}$ 通道才开始响应设备 D_5 的服务请求, 这时设备已经发出了两个传送数据的服务请求, 因此第一次传送的数据有可能丢失。

(3) 当字节多路通道的最大流量与连接在这个通道上的所有设备的数据流量之和非常接近时, 虽然能够保证在宏观上通道不丢失设备的信息, 但不能保证在某个局部时刻不丢失信息。由于高速设备在频繁地发出要求传送数据的请求时, 总是被优先得到响应和处理, 这就可能使低速设备的信息一时得不到处理而丢失, 如本例中的设备 D_5 。

为了保证本例中的字节多路通道能正常工作, 可以采取以下措施来解决。

① 增加通道的最大流量, 保证连接在通道上的所有设备的数据传送请求能够及时得

到通道的响应。

② 动态改变设备的优先级。例如,在图 3-12 中,只要在 $30\sim70\mu\text{s}$ 临时提高设备 D_5 的优先级,就可使设备 D_5 的第一次传送请求及时得到通道的响应,其他设备的数据传送请求也能正常得到通道的响应。

③ 增加一定数量的数据缓冲器,特别是对优先级比较低的设备。例如,只要为设备 D_5 增加一个数据缓冲器,它的第一次数据传送请求可在 $85\mu\text{s}$ 处得到通道的响应,第二次数据传送请求可以在 $145\mu\text{s}$ 处得到通道的响应,所有设备的数据都不会丢失。

习 题 3

3-1 总线控制方式有哪三种? 各需要增加几根用于总线控制的控制线? 总线控制优先级可否由程序改变?

3-2 简要列出集中式串行链接、定时查询和独立请求三种总线控制方式的优缺点。

3-3 如果通道在数据传送期中,选择设备需 $9.8\mu\text{s}$,传送一字节数据需 $0.2\mu\text{s}$ 。某低速设备每隔 $500\mu\text{s}$ 发出一字节数据传送请求,至多可接几台这种低速设备? 对于以下 A~F 6 种高速设备,一次通信传送的字节数不少于 1024 字节,哪些设备可以挂在此通道上? 哪些设备则不能挂在此通道上? 其中 A~F 设备每发一字节数据传送请求的时间间隔分别如表 3-3 所示。

表 3-3 题 3-3 中 6 种设备每发一字节数据传送请求的时间间隔

设 备	时间间隔/ μs	设 备	时间间隔/ μs	设 备	时间间隔/ μs
A	0.2	C	0.5	E	0.4
B	0.25	D	0.19	F	0.21

3-4 某字节多路通道连接 6 台外设,其数据传输率分别如表 3-4 所示。

表 3-4 题 3-4 中 6 台设备的数据传输率

设备号	传输率/(kB/s)	设备号	传输率/(kB/s)	设备号	传输率/(kB/s)
1	50	3	100	5	40
2	15	4	25	6	20

(1) 计算所有设备都工作时的通道实际最大流量。

(2) 若设计的通道实际工作周期使通道极限流量恰好与通道实际最大流量相等,以满足流量设计的基本要求,同时让速率越高的设备被响应的优先级越高。当 6 台设备同时发出请求开始,画出此通道在数据传送期内响应和处理各外设请求的时间示意图。能发现什么问题?

(3) 在(2)的基础上,在哪台设备内设置多少字节的缓冲器就可以避免设备信息丢失? 那么是否说关于流量设计的基本要求是没有必要的了? 为什么?

3-5 通道型 I/O 系统由一字节多路通道 A(其中包括两个子通道 A1 和 A2)、两个