9.1 主机与外设的连接

现代计算机中外设各种各种各样，各类外部设备不仅结构和工作原理不同，而且和主机的连接方式也是复杂多变的。

 9.1.1 输入输出接口

分析一下主机与外设之间需要交换的信息

   1.数据信息：可以通过输入设备送入计算机的，也可以通过输出设备输出结果。

   2.控制信息：CPU对外设的控制信息或管理命令。

   3.状态信息：这类信息用来标志外设的工作状态。

   4.联络信息：这个是主机和外设之间联络(也就是时间配合问题)的信息。

   5.外设识别信息:I/O寻址信息，使得CPU可以锁定与之交换信息的外设。

 9.1.2 接口的功能和基本组成

  1.接口的功能

  （1）.实现主机和外设的通信联络控制：也就是解决主机和外设时间分配问题的功能

  （2）.进行地址译码和设备选择：实现主机识别是哪个外设与之通信的功能

  （3）.实现数据缓冲：实现大容量软件的运行

  （4）.数据格式的变换：实现数模、模数、并串转换的功能

  （5）.传递控制命令和状态信息：实现主机对外设的控制以及对外设状态的了解的功能。

  2.接口的基本组成

  接口基本由数据端口、命令端口和状态端口组成。这些端口都是寄存器，用来暂时存储数据信息、控制信息以及状态信息的。CPU可以把信息写入这些端口，同时也可以从这些端口中读取信息。

  3.接口的类型

  （1）.按数据传送方式分类：串行接口和并行接口

  （2）.按主机访问I/O设备的控制方式分类：程序查询式接口、程序中断接口、DMA接口以及通道控制器。

  （3）.按功能选择的灵活性分类：可编程接口和不可编程接口

  （4）.按通用性分类：专用接口和通用接口

  （5）.按输入输出的信号分类：数字接口和模拟接口

  （6）.按应用来分类：运行辅助接口、用户交互接口、传感接口以及控制接口

 9.1.3 外设的识别与端口寻址

为了能在众多外设中找到与主机进行信息交换的外设，就必须要对外设进行编址了。首先外设识别是通过地址总线和接口电路中的识别电路来实现的。如果要想实现对这些端口的访问，就必须要对端口进行编址。

  1.端口编址方式

  （1）.独立编址：与主存分开进行编址

  （2）.统一编址：与主存放在一起编址

  2.独立编址方式的端口访问

  就是采用独立编地之后如何对端口进行访问

  可以将两个连续的8位端接口作为16位端口处理，同理，四个连续的8位端口做32位端口处理。

  假如最多有256个端口地址：

  可以指定为0~255的256个8位端口

  也可以指定为0、2、4、…、254的128个16位接口

  也可以指定为0、4、8、…、252的64个32位接口

 9.1.4 输入输出信息传送控制方式

按照主机与外设之间的信息传送控制方式的发展次序，可以分为四种

  1.程序查询方式

  这是主机和外设之间交换的最简单的方式，输入和输出完全由CPU执行程序来完成的。假如CPU选中一个外设，那么他就会查看这个外设是不是准备就绪，如果不是，就继续查看，直到他准备就绪了，就执行I/O操作。

  2.程序中断方式

  在主机启动外设之后，他就继续干他之前干的事，不去理会外设，然后外设在启动之后开始做准备，做好准备之后，给主机发送一个中断请求，主机收到了这个中断请求之后，才停下手里的活，去 中断服务程序 对外设进行处理，处理完之后，又一头栽回到自己原来的事情上去。

  3.DMA方式

  这方式就直接在主存和外设之间开一条数据通路，注意这里是数据通路，也就是说他只能输入输出数据了，这样做了，就不需要麻烦CPU了，从而保证CPU高速运转。

  4.通道控制方式

  这个技术是在DMA技术上进一步发展，他在系统中设置很多通道控制部件，每个通道控制部件上面都控制着很多外设。大体上：主机执行I/O指令，让有关通道打开，然后通道开始工作，执行相应的控制信号，控制要操作的外设，完成输入输出操作。这个通道控制器是从属于CPU的专用处理器，在输入输出全过程，只打扰CPU两次，也就是启动和停止的时候，这就能保证主机、外设以及通道可以并行同时工作。

  目前，小型、微型计算机用程序查询、程序中断和DMA方式，大、中型机采用通道方式。

9.2 程序查询方式及其接口

程序查询方式是主机和外设之间进行信息交换最简单的方式了，他的核心问题就是CPU要不断的查询外设是否准备好。

 9.2.1 程序查询方式

  1.程序查询的基本思想

  由CPU执行输入输出程序来实现主机和外设之间数据传送的方式叫做程序直接控制方式，这种方式分为两种：无条件传送方式和程序查询方式。

  （1）.无条件传送方式：外设(这样的外设一般贼简单，像一个灯泡一样)的一直给主机说：“我准备好了”,他也确实准备好了，然后CPU想用的时候，不用查询他的状态了，因为默认外设就是处于准备好的状态，CPU就直接利用I/O指令访问I/O端口，从而与外设之间进行信息交换。

  （2）.程序查询方式：这个在上面已经简单介绍过了，在这里我还就要再说一遍(← \_ ←)。和上面无条件传送方式不同的是，此时的外设并不都是准备好的，那么CPU就得对每个外设进行查询，看他准备好没有，没有，继续查询，直到外设准备好了，开始进行信息交换。

  2.程序查询方式的工作流程

  （1）.预置传送参数：包括主存缓冲区首地址和数据个数

  （2）.向外设接口发出命令字：给外设发送命令(控制信息),让外设知道他要干啥

  （3）.从外设接口中取回状态字：取回外设的状态字，判断数据传送是否可以进行

  （4）.查询外设标志：CPU不断查询状态标志，如果没有准备好就转到第（3）步，直到外设准备好

  （5）.传送数据：外设准备好之后，主机与外设之间进行数据传送。

  （6）.修改传送参数：将主存缓冲区地址+1(也就是指向下一个数据)，将程序计数器+1(也就是指向下一条命令)。

  （7）.判断传送是否结束：如果程序计数器不为0，就证明还有命令没有执行完毕，跳转到第（3）步，继续执行，直到程序计数器为0，表示传送结束。

 9.2.2 程序查询方式接口

  1.输入接口

  当输入设备准备好数据的时候，发送一个STB信号，将数据放入到锁存器中，此时将Ready(状态字)设置为1.

  CPU从外设输入数据时：

  （1）.先读取状态字，也就是Ready，假如Ready=1，也就是锁存器中有数据

  （2）.再读取数据，从锁存器中取出数据，同时将Ready置零，以准备接受下一个数据

  （3）.Ready置零了，CPU等待，如果有数据重新被输入到锁存器中，那么Ready就被置1了，又可以继续读取数据了

  2.输出接口

  CPU输出数据到外设中：

  （1）.CPU读取状态字Busy=1，表示锁存器中有数据，暂时不能输出，等待；如果Busy=0，代表可以输出，那么CPU将数据送入到锁存器中，并将Busy置为1。

  （2）.当外设把CPU送来的数据从锁存器中取出输出的时候，发送一个ACK非的命令，时Busy置0，准备下一次传送。

  假如有多个外设的时候，那么CPU就会巡回查询各个外设，发现哪个准备就绪了，就实施数据传送，然后对下一外设进行查询，依次循环。

9.3 中断系统和程序中断方式

中断是现代计算机发挥效能和提高效率的一个十分重要的功能。CPU中就设有处理中断的机构-中断系统。本章主要讲的是I/O中断。

 9.3.1 中断的基本概念

  1.中断的提出

  由于程序查询方式的缺点：

  （1）.查询过程中，CPU长时间处于等待状态

  （2）.CPU在一段时间内只能和一台外设交换信息

  （3）.不能发现和处理错误和异常情况

  所以，我们引进了程序中断方式。

  程序中断的思想：

  （1）.CPU先启动一个外设，然后CPU就去做自己的事情了，不需要像程序查询方式那样一直等，此时的外设也开始了自己的启动工作

  （2）.当外设启动好之后，给CPU发送一个中断请求

  （3）.CPU在可以响应这个请求的情况下，就会暂时中止自己手里的活，去 执行中断服务程序 (这是一个专门执行中断程序的程序，CPU执行的)为中断申请者服务，也就是完成一次主机与外设之间的数据传送。

  （4）.传送完成之后，CPU返回到之前自己执行的程序上继续执行。

  从上图我们可以看出：

  （1）.实现了CPU和外设的并行工作

  （2）.也可以实现多个外设并行工作

  （3）.可以检测到异常事件，在中断请求中，可以捕获到故障和错误

  （4）.这个过程实际上就是程序的切换过程：从现行程序切换到中断服务程序，再从中断服务程序切换到现行程序。在切换到中断服务程序的之前，CPU要保护断点、保护现场，在切换回现行程序之后，CPU还要恢复现场、恢复断点。这个过程会限制数据传送的速度。

  中断系统是计算机中实现中断功能的软硬件总策划呢。1.在CPU内配置中断机构 2.在外设接口中配置中断控制器 3.在软件上设计相应的中断服务程序。

  2.程序中断和调用子程序的区别

  很多人认为程序中断的本质就是调用子程序，但其实不是，他们之间的区别还是很多的：

  （1）.子程序调用一般都是事先安排好的，而中断的话由随机的中断事件引起的

  （2）.子程序的执行受到上层子程序或主程序的控制，但是中断服务程序和中断的程序没有关系。

  （3）.不存在同时调用多个子程序的情况，但是可以存在调用多个外设进行同时服务

  3.中断的基本类型

  （1）自愿中断和强迫中断

  （2）.程序中断和简单中断

  （3）.内中断和外中断

  （4）.向量中断和非向量中断：中断服务程序的入口地址是由中断事件提供的，这个地址称为向量地址。

  （5）.单重中断和多重中断：单重中断不可被打断，多重中断意思是可以中断嵌套，也就是可以被打断。

 9.3.2 中断请求和中断判优

  1.中断源和中断请求信号

  中断源是中断请求信号的来源，也就是引起计算机中中断的事件。通常计算机中每个中断源都会对应一个中断请求触发器(INTR)。当一个中断源有中断请求的时候，对应的中断请求触发器就会置1，表示向CPU发送中断信号。

  中断请求触发器 可以分散在各个中断源中，也可以集中到中断接口电路中。在中断接口电路中，多个中断请求触发器构成一个中断请求寄存器。中断请求寄存器的每一位对应着一个中断源。

  2.中断请求信号的传送

  （1）.独立请求线：每一个中断源单独设置中断请求线，将中断信号直接送到CPU(类似于集中仲裁的独立请求方式)。

  （2）.公共请求线

  多个中断源公用一根请求线。(类似于集中仲裁中的链式查询方式)

  （3）.二维结构

  同意优先级的用一根公共请求线，不同优先级的用不同的请求线。

  3.中断优先级与判优方法

  CPU虽然可以让外设实现并行工作，但是CPU在一瞬间只能接受一个中断源的请求，对那些提出中断请求后需要立即处理，否则就会造成严重的后果的规定为最高优先级；对那些可以延迟响应的规定较低的优先级。如故障的优先级一般较高，其次就是简单中断了，接着才是I/O设备中断。

  每个中断均有一个为其服务的中断服务程序，每个中断源也有与之对应的优先级别。另外CPU执行的程序也有优先级，只有中断源的优先级别高于执行程序的优先级别，才能中止CPU的执行程序。

  中断判优的方法有两种：

  （1）.软件判优：通过程序对中断寄存器各位的检测，检测顺序是按优先级大小排列的。

  （2）.硬件判优：采用硬件设计电路来判优

 9.3.3 中断响应和中断处理

  1.CPU响应中断的条件

  （1）.CPU接收到中断信号请求

  （2）.CPU允许中断：CPU有个允许中断触发器，等于1的时候才能相应中断。

  （3）.一条指令执行完毕：这是中断处理的时间限制，你总得让人家把一条指令执行完吧，总不能执行一半吧。

  2.中断隐指令

  这些指令是CPU在响应中断之后，在执行中断服务程序之前进行的一些隐蔽操作，之所以说他隐蔽，是因为他并不是以指令的形式存在的，他是由硬件直接实现的，所以叫他中断隐指令。他完成的操作主要有：

  （1）.保存断点

  （2）.暂不允许中断：为了保护CPU在主要寄存器中的内容，不让新的中断打断正在执行的中断，从而保证正在执行的中断结束之后，可以继续执行CPU当时在执行的程序。

  （3）.引出中断服务程序

  3.中断周期

  中断周期的设置是用来完成中断隐指令的。

  4.进入中断服务程序

  （1）.软件方面：就是上面提到的，由中断隐指令引出中断总服务程序，在那里判优、寻找中断源并且转入相应的中断服务程序(不同于向量中断)

  （2）.硬件方面：采用向量中断法，当CPU响应某一中断请求的时候，硬件能自动形成并找出与该中断源对应的中断服务程序的入口地址。

  向量中断的过程：

  （1）.中断源向CPU发送中断请求信号INTR非

  （2）.CPU进行一定的判优处理，若决定响应这个中断请求，就向中断源发送中断响应信号INTA。

  （3）.中断源接到INTA信号就通过自己的向量形成不见向CPU发送向量地址。

  （4）.CPU接受到向量地址之后就可转入响应的中断服务程序。

  向量地址通常有两种情况：

  （1）.向量地址就是中断服务程序的入口地址

  （2）.向量地址是中断向量表的指针：通过中断向量表中存储单元中取出中断服务程序的入口地址。

  5.中断现场的保护和恢复

  （1）.软件保护：将需要保存的参数送入主存

  .（2）.硬件保护：堆栈保存信息

  （3）.软、硬件保护(一般和向量中断法一起使用)

 9.3.4 多重中断与中断屏蔽

  1.中断嵌套

  中断嵌套也就是在执行一个中断的时候，接到了另一个优先级比当前中断还要高的中断的时候，确保这个时候系统必须处于开中断状态，这时就可以去执行优先级较高的中断了，对于这个中断采取先恢复后保护，而对于之前的中断采取先保护后恢复。

  2.允许和禁止中断

  CPU中又一个中断允许触发器，也就是上面说的开(关)中断。

  开中断情况：

  （1）.单重/多重中断的中断服务程序执行完毕，恢复中断现场之后。

  （2）.在多重中断的情况下，保护中断现场之后。

  关中断情况：

  （1）.当响应某一中断时，不希望被其他中断打断的时候。

  （2）.在保护和恢复现场之前

  3.中断屏蔽

  发出的中断并不一定都能到达CPU，可以用程序方式有选择地封锁部分中断，这就是中断屏蔽。

  屏蔽字的1越多，屏蔽能力越厉害，一般能屏蔽同级和比自己低级的。

  4.中断升级

  中断屏蔽字的另一个作用就是改变中断优先级，将原级别较低的中断源变成较高的级别，称为中断升级。

 9.3.5 中断全过程

  中断的全过程大致上分为5个阶段：中断请求、中断判优、中断响应、中断处理、中断返回。

  其中最重要的就是中断处理了，也就是执行中断服务程序。大多数计算机的中断处理分为三部分：

  （1）.准备部分

  保护线程，对于非向量中断方式需要确定中断源，开放中断，允许更高级的中大怒请求打断低级的中断服务程序。

  （2）.处理部分

  执行中断服务程序。

  （3）.结尾部分

  先关中断，恢复现场，然后再开中断，中断服务程序的最后一条指令一定是中断返回指令。

9.4 DMA方式及其接口

DMA方式是为了在主存与外设之间实现高速、批量数据交换设置的，DMA方式是直接靠硬件（DMA控制器）来实现的，不需要执行程序（节省CPU时间）。

 9.4.1 DMA方式的基本概念

  1.DMA方式的特点

  （1）.使得主存和CPU脱钩，主存既可以被CPU访问也可以被外设访问。

  （2）.在数据块传送时，主存地址的确定、传送数据的计数都是硬件电路直接实现的。

  （3）.主存中要开辟专用缓冲区，及时供给和接受外设的数据

  （4）.DMA传送速度快，CPU和外设并行工作，提高了系统效率

  （5）.DMA在传送之前要进行预处理，结束之后要通过中断方式进行后处理

  2.DMA和中断的区别

  （1）.中断方式是程序切换，需要保护和恢复现场；DMA除了开始和结尾，不占用CPU资源

  （2）.中断的发生时间限制的是指令执行完毕之后；而DMA则更小，可以发生在每个机器周期结束时。

  （3）.中断传送过程需要CPU干预；DMA则不需要。

  （4）.DMA的请求优先级高于中断

  （5）.中断可以处理异常；DMA只能局限于完成传送数据块的I/O操作。

  3.DMA方式的应用

  一般用于主存和高速外设(磁盘、磁带、光盘)之间的简单数据传送。

  在大批量数据采集系统(DRAM的刷新)中也采用DMA方式。

 9.4.2 DMA接口

  DMA接口相对前面两种比较复杂，习惯将DMA接口称为DMA控制器

  1.DMA控制器的功能

  （1）.接受外设发出的DMA请求，并向CPU发出总线请求

  （2）.当CPU响应总线请求后，发出响应信号，DMA控制器接管对总线的控制，进入DMA操作周期

  （3）.确定数据的主存单元地址和传送长度，能够自动修改主存地址计数值和传送长度计数值

  （4）.规定数据在主存和外设之间的传送方向，发出读写或其他控制信号，执行数据传送的操作

  （5）.向CPU报告DMA操作结束

  2.DMA控制器的基本组成

  （1）.主存地址计数器：用来存放减缓数据的主存地址

  （2）.传送长度计数器：用来记录传送数据块的长度

  （3）.数据缓冲寄存器：用来暂存每次传送的数据

  （4）.DMA请求触发器：当外设准备好之后给出一个控制信号，使DMA请求触发器置位

  （5）.控制/状态逻辑：指定传送方向，修改传送参数，对DMA请求信号和CPU响应信号进行协调和同步。

  （6）.中断机构：当一个数据块传送完毕后发触发中断机构向CPU提出中断请求，CPU将进行DMA传送的结尾处理。

  3.DMA控制器的引出线

  （1）.地址线

  （2）.数据线

  （3）.控制数据传送方式的信号线

  （4）.DMA控制器与外设之间的联络信号线

  （5）.DMA控制器与CPU之间的联络信号线

  4.DMA控制器的链接和传送

  写入主存的操作：

  （1）.由外设向DMA控制器发出DMA请求信号DREQ

  （2）.DMA控制器向CPU发送总线请求信号HRQ

  （3）.CPU向DMA发出总线响应信号HLDA，DMA控制器获得总线的控制权

  （4）.DMA控制器向外设发出DMA响应信号DACK，表示他已经控制了总线，允许外设与主存交换数据

  （5）.DMA控制器按主存地址计数器的内容发出地址信号作为主存地址的选择，同时主存地址计数器+1

  （6）.DMA控制器发出IOR非信号到外设，将外设数据读入到数据缓冲寄存器，发出MEMW非信号，将数据缓冲寄存器的内容写入选中的主存单元

  （7）.传送长度计数器-1

  重复（5）~（7）直到长度计数器为0，DMA控制器的HRQ降为低电平，总线控制权交还CPU

 9.4.3 DMA传送方法与传递过程

  1.DMA传送方法

  （1）.CPU停止访问主存法

  （2）.存储器分时法

  （3）.周期挪用法

  2.DMA传送过程

  （1）.DMA预处理

  （2）.数据传送

  （3）.DMA后处理

9.5 通道控制方式

在大型计算机系统中，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CPU采取中断和DMA等控制方式还是不能满足要求，于是通道控制方式被引入计算机系统

 9.5.1 通道的基本概念

  1.通道控制方式与DMA方式的区别

  （1）.DMA控制器是一个专门设计的硬件控制逻辑来实现对数据传送的控制；而通道则是一个特殊功能的处理器，类似于一个处理器，更像一个小CPU，通道具有更强的独立处理数据输入输出的功能。

  （2）.DMA控制器通常只能控制一台或者几台同类设备；而一个通道则可以同时控制许多同类或不同类的设备。

  2.通道功能

  （1）.接受CPU的I/O指令，按照指令要求与制定的外设进行联系。

  （2）.从主存中取出通道指令，经过译码后向设备控制器和设备发送各种命令。

  （3）.实施主存和外设间的数据传送。

  （4）.从外设获得设备的状态信息。

  （5）.将外设的中断请求和通道本身的中断请求按次序报告给CPU

 9.5.2 通道的类型与结构

  1.通道类型

  （1）.字节多路通道：是一种简单的共享通道，以字节交叉方式传送信息。

  （2）.选择通道：选择一个通道程序，执行完了之后执行下一个，在执行的时候独占整个通道，只有当前的执行完了之后才能执行下一个通道程序。缺点是当进行辅助操作的时候(比如磁盘机的寻道)，通道暂时不用，通道就会处于等待状态，利用率不高。

  （3）.数组多路通道：是前两种的结合，记性数据传送的时候，置为该设备服务，进行辅助操作的时候，先挂起该设备的通道程序，去为其他设备服务。

  2.通道结构

  CCWR：通道命令字寄存器

  CCW：控制I/O操作的关键参数，一条条命令字(通道指令)构成通道程序，放在主存中。

  CAWR：通道地址寄存器，指出了通道程序在主存的起始地址。

  CSWR：通道状态字寄存器：记录通道程序执行后本通道和响应设备的状态信息，这些信息称为通道状态字（CSW）

 9.5.3 通道程序

  1.通道指令

  IBM的通道指令格式：

  （1）.命令码

  （2）.数据地址

  （3）传送字节计数

  （4）.标识码：

    数据链特征：CD=1表示接下来的一条指令也是数据传送命令。

    命令链特这：CC=1表示本条通道指令执行完毕，只有当CC和CD全为0的时候，才表示这个指令是通道程序的最后一条指令，通道程序将结束。

    封锁错误长度特征：SLI=1，封锁了错误长度请求，就是不让这个请求出去，不让他告诉CPU，就当每发生，继续执行接下来的操作。

    封锁写入主存特征：SKIP=1，禁止外设读出的数据写入主存

    程序控制中断特征：PCI=1，表示本指令允许产生一个中断条件

  2.通道程序举例

 9.5.4 通道工作过程

  （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，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响应之后，第二次调用管理程序对中断请求进行处理