

图 5.1 I/O 接口的基本组成

5.2.4 I/O 与主机交换信息的控制方式之一——程序查询方式

1. 程序查询方式的特点

这种方式 CPU 一旦启动 I/O 设备,必须停止现行程序的运行,并在现行程序中插入一段程序。这段程序要时刻查询 I/O 设备的准备状况,等待 I/O 设备准备就绪时可实现 I/O 设备与主机交换信息,如图 5.2 所示。程序中要用到测试指令、转移指令和传送指令。可见这种方式的主要特点是 CPU 有“踏步”等待现象,CPU 与 I/O 设备处于串行工作状态。

2. 程序查询方式的接口电路

以输入设备为例,程序查询方式接口电路的基本组成如图 5.3 所示。图中 B 是工作触发器,D 是完成触发器。

当 CPU 通过 I/O 指令启动输入设备时,指令的设备码字段通过地址线送至设备选择电路,假设该接口的设备码与地址线上的代码吻合,其输出 SEL 有效,使 I/O 指令的启动命令经过与非门使 B 置“1”,使 D 置“0”,然后由 B 触发器启动设备工作。当输入设备将数据送至数据缓冲寄存器后,由设备发出工作结束信号,使 D 置“1”,B 置“0”,表示外设准备就绪。这样,当 CPU 通过测试指令测得完成触发器 D 的状态为“1”时,就可通过传送指令将数据缓冲寄存器的内容送至 CPU,再送至存储器。

这种方式由程序判断输入数据是否全部输入结束,其程序流程如图 5.4 所示。

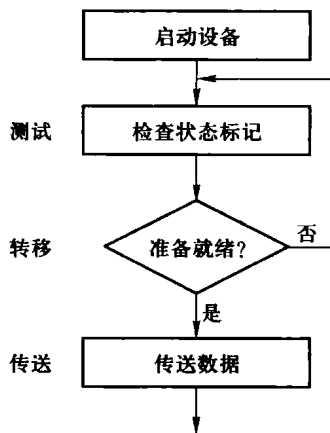


图 5.2 程序查询方式示意

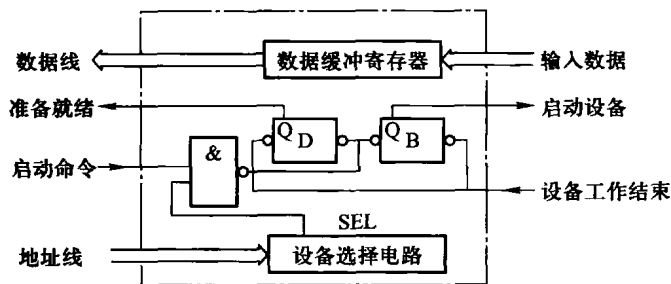


图 5.3 程序查询方式接口电路的基本组成

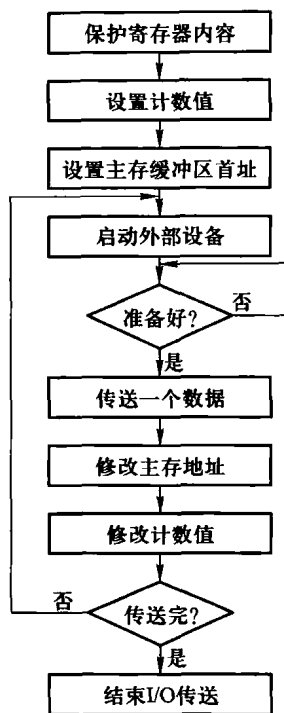


图 5.4 程序查询方式的程序流程

5.2.5 I/O 与主机交换信息的控制方式之二——程序中断方式

1. 程序中断方式的特点

这种方式 CPU 启动 I/O 设备后,不必停止现行程序的运行。而 I/O 设备接到启动命令后,进入自身准备阶段。当准备就绪时,向 CPU 提出请求,此时 CPU 即中断现执行程序,并保存原程序断点,转至执行中断服务程序,为 I/O 服务。中断服务程序结束后,CPU 又返回到程序的断点处,继续执行原程序。如图 5.5 所示。

可见这种方式 CPU 启动 I/O 设备后不必查询 I/O 设备的准备状况,I/O 设备作准备和 CPU 运行程序是并行的,仅当 I/O 设备准备就绪向 CPU 发出中断请求时,如果条件允许(允许中断触发器为“1”),CPU 才中断现执行程序。

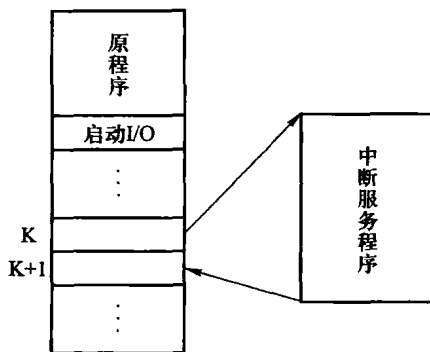


图 5.5 程序中断方式示意图

2. 程序中断方式的接口电路

以输入设备为例,程序中断方式接口电路的基本组成如图 5.6 所示。图中 INTR 为中断请求触发器, MASK 为屏蔽触发器, B 为工作触发器, D 为完成触发器。

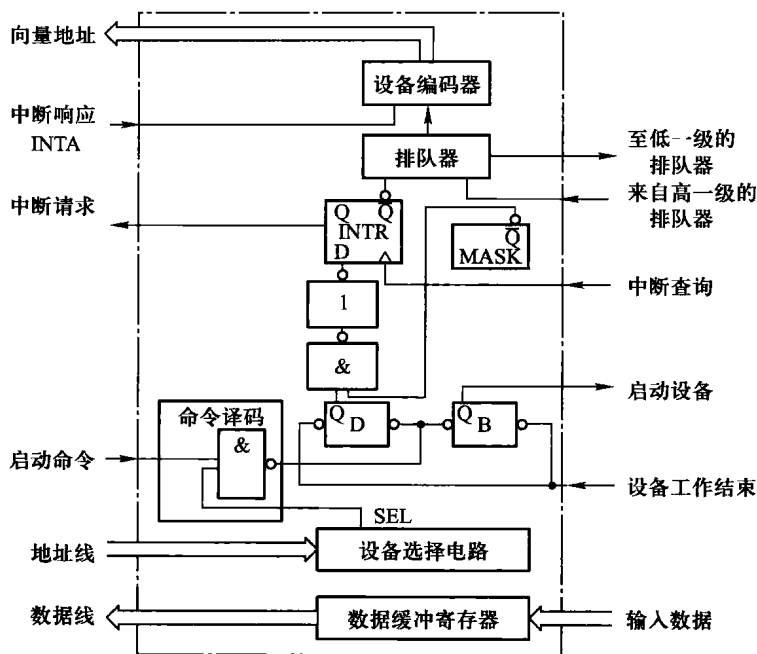


图 5.6 程序中断方式接口电路的基本组成

当 CPU 通过 I/O 指令启动输入设备时,指令的设备码字段通过地址线送至设备选择电路,假设该接口的设备码和地址线上的代码吻合,其输出 SEL 有效,使 I/O 指令的启动命令经过与非门使 B 置“1”,使 D 置“0”,然后由 B 触发器启动设备工作。当输入设备将数据送至数据缓冲寄存器后,由设备发出工作结束信号,使 D 置“1”,B 置“0”。此时若 CPU 内的允许中断触发器 EINT=1,且该接口对应的设备未被屏蔽(MASK=0),则在每条指令执行阶段结束时刻,由 CPU 发来的中断查询信号将 INTR 置“1”,接口向 CPU 发中断请求。与此同时,该请求信号经排队器送至设备编码器。在 CPU 响应中断时(INTA 有效),就可将该设备的向量地址通过数据线送至 CPU。CPU 通过向量地址找到中断服务程序的入口地址,便中断现执行程序,转入该设备的中断服务程序,将数据缓冲寄存器的数据送至 CPU,再送至存储器。中断服务程序的最后一条中断返回指令将 CPU 返回到原程序的断点处,继续执行原程序。

综上所述,一次中断处理过程可简单归纳为中断请求、中断判优、中断响应、中断服务和中断返回五个阶段。

3. 单重中断和多重中断

如果 CPU 在执行中断服务程序的过程中,又出现了新的中断请求,而 CPU 对新的中断请求不予响应,这种中断叫做单重中断,如图 5.7(a)所示。如果 CPU 在执行中断服务程序的过程中,又出现了新的中断请求,而且这个新的中断请求的级别比当前正在服务的中断请求级别更高,此时 CPU 再次中断现行的中断服务程序,转去处理新的中断请求,这种中断叫做多重中断,又叫中断嵌套,如图 5.7(b)所示。

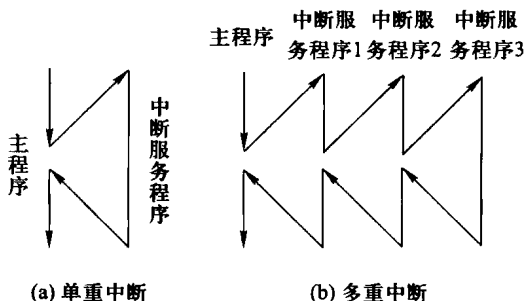


图 5.7 单重中断和多重中断示意图

从宏观上看,虽然程序中断方式克服了程序查询方式中的 CPU“踏步”现象,实现了 CPU 与 I/O 并行工作,提高了 CPU 的资源利用率。但从微观操作分析,CPU 在处理中断服务程序时,仍需暂停原程序的正常运行,尤其是当高速 I/O 设备或辅助存储器频繁地、成批地与主存交换信息时,需不断打断 CPU 执行现行程序而执行中断服务程序。如果设想在主存和设备之间有一条直接数据通路,它们之间的信息交换可以不通过 CPU,直接在这条数据通路上传送,就可以不中断现行程序,这就是 DMA 方式。

5.2.6 I/O 与主机交换信息的控制方式之三——DMA 方式

1. DMA 方式的特点

DMA 方式是直接存储器存储方式,其特点是主存和 DMA 接口之间有一条直接数据通路,图 5.8 所示为 DMA 和程序中断两种方式的数据通路。由于 DMA 方式传送数据不需经过 CPU,因此不必中断现行程序,I/O 与主机并行工作。但当 DMA 接口与 CPU 同时访存时,要求 CPU 将总线的控制权交给 DMA 使用,这叫做周期窃取或周期挪用。

2. DMA 的传送方式

(1) 停止 CPU 访问主存

这种方式当外设需传送一批数据时,由 DMA 接口向 CPU 发出一个信号,要求 CPU 放弃地址线、数据线和有关控制线的使用权,DMA 接口获得总线控制权后,开始进行数据传送。在数据传送结束后,DMA 接口通知 CPU 可以使用主存,并把总线控制权交回给 CPU。在这种传送过程中,CPU 基本处于不工作状态或保持原状态。

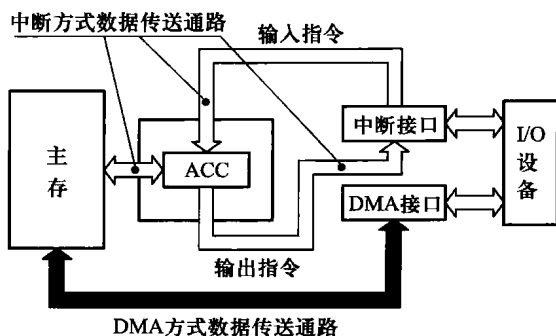


图 5.8 DMA 和程序中断两种方式的数据通路

这种传送方式控制简单,适用于数据传输率很高的设备成组传送。缺点是在访存阶段,主存的效能未充分发挥。这是因为设备在传送一批数据时,CPU 不能访问主存,而主存的速度远远高于设备的速度,即使是高速外部设备,在两个数据之间的准备间隔时间也总大于一个存取周期,使相当一部分主存周期是空闲的。为了提高主存的利用率,可采用周期挪用方式。

(2) 周期挪用

这种方式当 I/O 设备没有 DMA 请求时,CPU 按程序的要求访问主存,一旦 I/O 设备有 DMA 请求,会遇到三种情况,一种是此时 CPU 不在访存(如 CPU 正在执行乘法指令),故 I/O 设备的访存请求与 CPU 未发生冲突。第二种是 CPU 正在访存,则必须待存取周期结束后,CPU 再将总线占有权让出。第三种是 I/O 和 CPU 同时请求访存,出现了访存冲突,此刻 CPU 要暂时放弃总线占有权,由 I/O 设备挪用一個或几个存取周期。

与停止 CPU 访问主存方式相比,周期挪用方式既实现了 I/O 传送,又较好地发挥了主存与 CPU 的效率,是一种广泛采用的方法。

(3) DMA 与 CPU 交替访问主存

这种方式适用于 CPU 的工作周期比主存存取周期长的情况。例如 CPU 的工作周期是 $1.2\ \mu\text{s}$,主存的存取周期小于 $0.6\ \mu\text{s}$,那么可将一个 CPU 周期分为 C_1 和 C_2 两个周期,其中 C_1 专供 DMA 访存, C_2 专供 CPU 访存。

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 C_1 和 C_2 分时控制的。实际上总线变成了在 C_1 和 C_2 控制下的多路转换器,总线控制权的转移几乎不需要什么时间,具有很高的 DMA 传送效率。CPU 既不停止主程序的运行,也不进入等待状态,完成了 DMA 的数据传送。当然其相应的硬件逻辑变得更复杂。

3. DMA 方式的接口电路

图 5.9 是简单的 DMA 接口组成原理图。图中 AR 存放数据块在主存的首地址,有计数功能;DAR 为设备地址寄存器,用于存放设备号;WC 为字计数,存放交换数据的字数;BR 为数据缓冲寄存器,存放主存和设备之间交换的数据字。这些信息均在 DMA 传送的预处理阶段由 CPU

经数据线送至 DMA 接口内。在图 5.9 中, DMA 控制逻辑用于负责管理 DMA 的传送过程, 由控制电路、时序电路和命令状态寄存器等组成。图中的中断机构在 DMA 传送一批数据结束时, 可向 CPU 提出中断请求作为 DMA 传送的后处理。

以输出为例, 假设预处理已完成, 则当数据从 DMA 接口中的数据缓冲寄存器 BR 送至 I/O 设备后, 设备向 DMA 接口发出请求 (DREQ); DMA 控制逻辑收到该信号后即向 CPU 申请总线控制权 (HRQ); CPU 接到信号后, 若允许 DMA 接口占用总线, 即发回 HLDA 总线响应信号; DMA 接口收到 HLDA 信号后, 开始使用总线, 将 AR 的主存地址送地址总线, 命令存储器读, 并通知设备已被授予一个 DMA 周期 (DACK), 为接受下一个字作准备; 主存接到读命令后, 将对应 AR 地址的内容通过数据总线送至 DMA 接口电路中的 BR, 再送至设备。至此一个字传送结束, 修改主存地址 AR 和字计数 WC 之值, 然后根据 WC 的值判断数据块传送是否结束。若未结束, 继续传送; 若已结束, 则向 CPU 申请程序中断, 进行后处理。

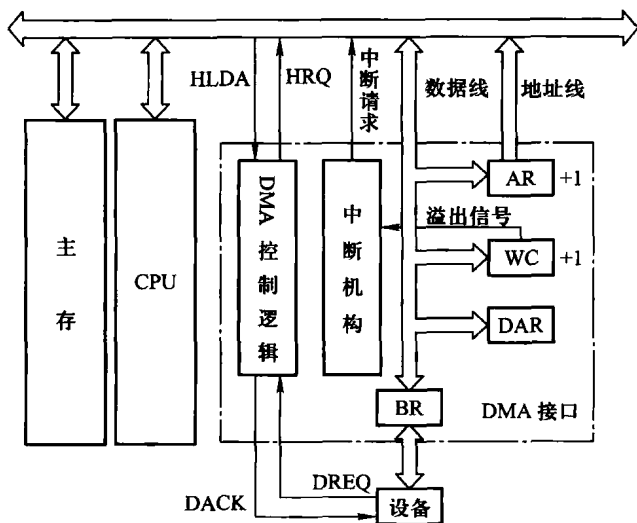


图 5.9 简单的 DMA 接口组成原理图

4. DMA 的传送过程

DMA 的数据传送过程分预处理、数据传送和后处理三个阶段。

(1) 预处理

- ① 指明数据传送方向是输入 (主存写) 还是输出 (主存读)。
- ② 设备地址送至 DMA 接口中的设备地址寄存器 DAR。
- ③ 主存首地址送至 DMA 接口中的主存地址计数器 AR。
- ④ 传送数据字数送至 DMA 接口中的字计数器 WC。
- ⑤ 启动设备。

(2) 数据传送

- ① 主存地址送总线。
- ② 数据送 I/O 设备(或主存)。
- ③ 修改主存地址。
- ④ 修改字计数器。
- ⑤ 直到数据块传送结束为止。

(3) 后处理

由中断服务程序作 DMA 结束处理,包括测试传送过程中是否出错,决定是否继续使用 DMA 传送其他数据块等。

5.3 例题精选

例 5.1 解释接口 (Interface) 和端口 (Port) 的概念。

【解】 如图 5.10 所示是一个简单的基本外部设备接口。图的中间部分是 I/O 接口,它位于 CPU 和外部设备之间,是一个连接部件。它一边通过地址线、数据线和控制线与 CPU 连接,另一边通过数据信息、控制信息和状态信息与外部设备连接。CPU 就是通过 I/O 接口与设备进行这三种信息的传送。

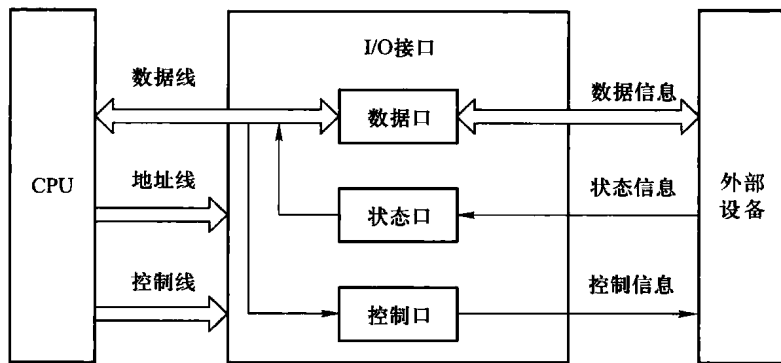


图 5.10 例 5.1 简单的外部设备接口

数据信息可以是数据量、模拟量和开关量三种。其中模拟量必须先经过接口电路中的“模/数”转换器(图中未画)转换为数字量后,才能输入 CPU 进行处理。状态信息表示外部设备当前所处的工作状态,如用 READY(就绪信号)表示输入设备已准备好,用 BUSY(忙信号)表示输出设备是否能接收。控制信息是由 CPU 发出的,用于控制外部设备接口工作方式,以及启动和停止外部设备。

数据信息、状态信息和控制信息(如设置外部设备接口的工作方式)通常以数据形式通过