

## 第8章 输入输出系统 作业参考题解

8.1 简述外设的编址方式，并比较它们的优缺点。

**参考答案：**

外设有两种编址方式：

- 外设与存储器统一编址：一个 I/O 端口等同于一个存储器单元。优点是指令系统中不设置专用的 I/O 指令，用功能很强的访存指令(如 LOAD/STORE 或者 MOV)来访问 I/O 端口，通过地址来区分访问的是存储器还是 I/O 端口；外设数目或 I/O 寄存器数几乎不受限制；微机的读写控制逻辑较为简单。缺点是 I/O 端口占用部分主存空间，可用主存空间减小；访存指令较长，执行速度较慢；I/O 端口地址译码电路复杂，译码时间较长。
- 外设独立编址：I/O 端口地址空间与存储器地址空间相互独立。指令系统中设置了专用的 I/O 指令，用 I/O 指令来访问 I/O 端口，用访存指令来访问存储器。优点是 I/O 端口地址不占用存储器地址空间；I/O 端口数量不多，占用地线少，地址译码简单，速度较快；使用专用 I/O 命令(IN/OUT)，指令短，执行速度快，可读性强。缺点是专用 I/O 指令增加指令系统复杂性，且 I/O 指令类型少，程序设计灵活性较差；要求处理器提供 MEMR/MEMW 和 IOR/IOW 两组控制信号，增加了控制逻辑的复杂性。

8.2 当控制器执行访存指令和 I/O 指令时，发出的读写控制信号有何不同？

**参考答案：**

当控制器执行访存指令时，发送的是存储器的读/写控制信号（ $\overline{\text{MEMR}}/\overline{\text{MEMW}}$ ）；执行 I/O 指令时，发送的是 I/O 设备的读/写控制信号（ $\overline{\text{IOR}}/\overline{\text{IOW}}$ ）。

8.3 为何要在 CPU 和外设之间设置一个接口？简述接口的功能和组成。

**参考答案：**

**设置外设接口的原因：** 外设是计算机系统的重要组成部分，计算机通过它与外部交换信息，完成实际工作任务。外设有以下特点：①种类繁多，结构、原理各异，有机械式、电子式、电磁式等；②各种外设的数据传输速率相差甚大；③外设的输入输出信号形式和信息传送方式也可能不同；④外设的工作具有一定的独立性，一般它具有自己独立的时序控制逻辑，CPU 不能像访问主存和寄存器那样直接存取。

因此，与其他计算机部件相比，外设具有工作速度差异大、结构原理差异大、时序独立、异步性明显等特点，处理的信息从数据格式到逻辑时序一般不可能直接与 CPU 兼容。鉴于此，计算机与 I/O 设备间的连接与信息交换不能直接进行，而必须设计一个“接口电路”作为两者之间的桥梁，使 CPU 和外设协调工作，这种 I/O 接口电路又叫“I/O 适配器”（I/O Adapter）。

**接口的功能包含：** 实现数据缓冲、执行 CPU 的命令、返回外设的状态、设备译码与选择、实现数据格式的转换、实现信号的转换、中断管理功能。

8.4 主机和外设交换信息的方式有哪几种？简述各自的工作原理和特点。

**参考答案：**

主机和外设交换信息的方式有 4 种方式：

- (1) **程序查询方式**：CPU 和外设（接口）之间的数据传送完全靠 CPU 执行计算机查询程序来实现——CPU 首先查询外设是否准备好数据交换（发送或者接收），若准备好，则在 CPU 和外设之间传送一个数据，否则，CPU 循环测试与等待，直至外设准备好。它的特点是 CPU 与外设间通过程序同步，CPU 被外设独占，CPU 效率低下。
- (2) **程序中断方式**：在外设准备数据时，CPU 执行与传送数据无关的工作；外设准备好数据后，主动向 CPU 发送一个中断请求，当 CPU 执行完当前指令后，停止当前程序的执行，自动转向中断服务程序，在中断服务程序中，完成一个数据的传送，之后中断返回至原来的断点处，继续执行。特点是在外设准备数据时，CPU 与外设并行工作，CPU 效率有所提高，并且 CPU 可以同时被多个外设占用。
- (3) **直接存储器访问（DMA）方式**：将 I/O 过程中，与内存交换数据的操作交由 DMA 控制器来控制，简化了 CPU 对输入输出的控制，进一步提高了 CPU 的效率。特点是数据的传送不经过 CPU（由 DMAC 控制），而 I/O 设备管理由 CPU 控制，简化了 CPU 对 I/O 的控制；虽然硬件开销大，结构复杂，但 CPU 的效率很高。
- (4) **通道与输入输出处理机方式**：通道是一个具有特殊功能的处理器，它可以实现对外围设备的统一管理和外围设备与内存之间的数据传送。特点是能独立地执行用通道指令编写的输入输出控制程序，产生相应的控制信号送给由它管辖的设备控制器，继而完成复杂的输入输出过程。

8.5 CPU 与外设接口交换信息的方式有以下几种，其中在 A 下，CPU 被外设独占；而在 B 下支持 CPU 与外设并行工作，并且可以用于突发事件的处理；C 用于高速、大批量的数据传送，并由硬件实现。

- A. 程序查询方式                      B. 程序中断方式                      C. DMA 方式  
D. 并行传送                              E. 串行传送

8.6 程序中断方式与 DMA 方式有何异同？

**参考答案：**

程序中断方式和 DMA 方式传送数据的**相同之处**：都必须先做好传送准备，前者初始化中断向量，后者初始化 DMAC；然后启动传送，前者开中断，后者启动传送。两种传送方式均在外设准备数据期间，CPU 能并行工作，效率较高。

程序中断方式和 DMA 方式传送数据的**不同之处**：在中断方式下，数据传送过程中的一些操作，如存数和取数、地址刷新和计数以及检测传送是否结束等，是由软件控制相应的指令实现的；时间开销很大，不适合高速的外设或者块设备（如磁盘）。在 DMA 方式下，这些操作都由 DMA 控制器的硬件实现；因此传送速率很高，适合高速大批量的数据传送。

8.7 何谓中断？简述中断的全部过程。

**参考答案：**

**中断**：在 CPU 执行程序的过程中，由于某种事件发生，CPU 暂时停止正在执行的程序而转向对所发生的事件进行处理，当对事件的处理结束后又能回到原来中止的地方，接着中止前的状态继续执行原来的程序，这一过程称为中断。

**中断的全部过程**：包含 4 个阶段：

- (1) **中断申请**：对于外中断，外设或其他中断源通过 CPU 的中断请求引脚向 CPU 发中断请求信号，CPU 在每条指令执行完后，监测是否有中断请求，有则转入中断响应阶

段。

- (2) 中断响应：CPU 首先通过硬件保存程序断点（PC）及标志寄存器，以便中断返回，即 CPU 执行中断隐指令。然后进入中断响应周期，或者通过向量方式或者通过软件查询方式得到中断服务程序入口，并置入 PC。
- (3) 中断服务：CPU 转入中断服务程序并执行，进行外设所需的数据交换。中断服务程序中，首先保护现场，将有关寄存器的内容压栈，然后进行 I/O 操作，实现数据传送。最后，恢复现场，并执行中断返回指令。
- (4) 中断返回：即恢复断点及标志。中断返回指令的功能，就是将中断隐指令保存的程序断点和标志读出并送入 PC 和标志寄存器，从而回到 CPU 原来的程序断点处继续执行。

8.8 简要说明中断请求线的几种传送方式，并对比它们的优缺点。

**参考答案：**

中断请求线有 3 种传送方式：

- (1) 公共请求线：各中断源共用一根公共请求线，当 CPU 接到中断请求后，必须通过软件或硬件的方法来查询是哪个设备提出的中断请求，从而得到中断服务程序的入口地址。其特点是在负载允许的情况下，中断源的数目可随意扩充。
- (2) 独立请求线：各个中断源单独设置中断请求线，将中断请求直接送往 CPU。其特点是 CPU 无需查询，即可确定中断源及其中断服务程序的入口地址，提高了中断的响应速度；但其硬件代价较大，且 CPU 所能连接的中断请求线数目有限，中断源难以扩充。
- (3) 二维结构：将中断请求线连成二维结构，同一优先级别的若干中断源共享一条中断输入线。当 CPU 接到中断请求信号时，仍然需要查询，以识别中断源，但只需对共享一条中断请求线上的几个设备进行查询，不同请求线上的中断源优先级别不同。它综合了前两种方法的优点，适合管理数目较多的中断源且具有较快的响应速度。

8.9 什么是中断屏蔽字？简述中断屏蔽技术的作用。

**参考答案：**

中断屏蔽字：是指中断屏蔽寄存器（IMR）中的编码。中断屏蔽寄存器中的每一位对应屏蔽控制一个中断源：该位为 1，CPU 屏蔽该中断源的中断请求信号，不予以响应；该位为 0，则 CPU 开放该中断源的中断请求信号。

中断屏蔽技术：CPU 通过指令向中断屏蔽寄存器写入屏蔽字，以开放或者屏蔽某一个或者某一些中断源；但是利用中断屏蔽技术也可以巧妙地改变各中断源的优先级，使计算机适应各种场合的需要。通常把屏蔽字看成软件排队，通过在中断服务程序中修改屏蔽字，可以方便地改变中断源得到 CPU 服务的先后次序，也即动态地改变中断处理的优先级。

8.10 什么是中断嵌套？它解决了什么问题？如何才能实现中断嵌套？

**参考答案：**

中断嵌套：是指 CPU 在执行某个中断服务程序的过程中允许再响应更高级别的中断请求，也称为多重中断。

中断嵌套解决的问题是：如果 CPU 在执行某个中断服务程序的过程中，又发生新的中断请求，那么 CPU 如何处理？对于单重中断，CPU 不响应新的中断请求；对于中断嵌套，如果新中断的优先级更高，则优先服务新中断；否则，不予响应。因此，中断嵌套实现了：当多个中断请求同时到达时能先为优先级更高的请求服务，而且当优先级高的中断请求到达得晚，也能尽快得到优先服务。

中断嵌套技术的实现：关键是在中断处理程序中必须适时开放中断（STI 指令），并且使用堆栈的“先进后出”特性保证中断的逐级返回。

#### 8.11 中断源可以分为哪几类？各自有何特点？

##### 参考答案：

中断源可以分为硬中断和软中断 2 类：

硬中断：是指由外部设备和其他 CPU 外部事件引起的中断，因此又叫外中断。常见的外部中断有输入输出请求、实时时钟、计时器、电源故障、设备故障、校验线路等等。外中断一般通过 CPU 的中断请求引脚引入，CPU 必须通过中断响应周期来获取中断源的中断服务程序的入口地址，再转入执行中断服务程序。

软中断：是指 CPU 内部的指令或程序执行中的突发事件所引起的中断，又叫内中断。常见的软中断主要包括指令中断（例如中断指令 INT n）和程序异常（例如除数为零，运算溢出、指令的单步运行、程序运行至断点处等等）。内中断因为是软件引起的，中断原因及中断类型号固定，因此，可直接转入执行对应的中断服务程序，无需经过中断响应周期。

#### 8.12 有哪几种中断优先级排队方法？比较它们的优缺点。

##### 参考答案：

中断请求的优先级排队方法有有兩種：

- （1） 软件查询：就是用程序来判断优先级。当存在中断请求时，通过程序查询连接在该中断请求线上的每一个设备（检测设备中断请求寄存器的对应位状态），根据其查询的先后次序确定其优先级。优点是可以灵活地修改中断源的优先级别，硬件电路实现简单；缺点是查询、判优完全靠程序实现，需要占用 CPU 时间，同时中断响应较慢，优先级较低的设备被响应的等待时间也较长。
- （2） 硬件排队电路：采用硬件排队判优电路来判断中断源的优先级。优先级别高的中断请求将自动封锁优先级别低的中断请求的处理。硬件排队电路来判别中断优先级，优点是判别和响应速度快；缺点是硬件电路复杂，一旦设计连接好之后，优先级固定，不够灵活。

#### 8.13 解释下列名词：

- |         |          |          |          |
|---------|----------|----------|----------|
| （1）中断向量 | （2）中断隐指令 | （3）DMA   | （4）通道    |
| （5）多重中断 | （6）单重中断  | （7）中断类型号 | （8）菊花链电路 |

##### 参考答案：

- （1）中断向量：是指某个中断源的中断服务程序的入口地址（首址）。
- （2）中断隐指令：CPU 在响应某中断请求时，必须要先通过硬件保存程序断点（PC）及标志寄存器，以便中断返回。这个过程因为是硬件完成的，对程序员透明，因此称作

中断隐指令。

- (3) **DMA**：是指直接存储器访问（Direct Memory Access），是一种 CPU 与外设交换信息的方式。在这种方式下，由 DMA 控制器（DMAC）控制实现内存与外设之间的直接快速传送，而无需经过 CPU。
- (4) **通道**：是一个具有特殊功能的处理器，它可以实现对外围设备的统一管理和外围设备与内存之间的数据传送；它能独立地执行用通道指令编写的输入输出控制程序，产生相应的控制信号送给由它管辖的设备控制器，继而完成复杂的输入输出过程。
- (5) **多重中断**：即中断嵌套，指 CPU 在执行某个中断服务程序的过程中允许再响应更高级别的中断请求。
- (6) **单重中断**：指 CPU 正在执行的中断服务程序中，禁止再响应其他中断请求。
- (7) **中断类型号**：在中断系统中，对所有的中断源编码，为其分配一个惟一的代号，称为中断类型号。
- (8) **菊花链电路**：是指判断中断优先级的一种串行排队链电路，它对接在一条公共中断请求线上的几个中断源进行判优，方法是 CPU 发送的中断响应信号送至优先级最高的中断源，级别高的中断源封锁了级别低的中断源的中断请求信号。串行排队的硬件电路上，距离 CPU 最近的中断源优先级最高，只有当它没有中断请求时，CPU 发送的中断响应信号才能传递到下一个中断源。

8.14 假设有 4 个中断源 A、B、C、D，硬件排队后，中断优先级从高到低依次为 A→B→C→D。现在需要将 4 个中断源得到 CPU 响应的次序更改为 C→A→D→B，写出各个中断源的中断服务程序中应该设置的中断屏蔽字。

**参考答案：**

在各个中断源的中断服务程序中设置的中断屏蔽字为：

|           | A | B | C | D |
|-----------|---|---|---|---|
| A 的中断屏蔽字： | 1 | 1 | 0 | 1 |
| B 的中断屏蔽字： | 1 | 1 | 1 | 0 |
| C 的中断屏蔽字： | 1 | 1 | 1 | 1 |
| D 的中断屏蔽字： | 1 | 1 | 1 | 1 |