在计算机系统中，CPU与外部设备之间的数据交换可采用3种通信方式：

无条件传送、查询传送、中断方式。

另外，当CPU面对多项任务时，由于资源有限，可能出现资源竞争的现象，中断是解决这种竞争的最有效方法。

**5.2.1 中断概述**

**1． CPU与外设之间的数据传送方式**

三种方式：

1. **无条件传送方式：**CPU始终认为外设处于“待命”状态，不需要与外设交换状态信息，直接将信息送给外设。 ------缺点：容易出错，能应用的场合少。
2. **查询传送方式：**CPU首先查询外设的状态信息来看外设中的数据是否已准备就绪，准备好才与外设进行数据交换。

优点：通用性好，可靠性高；缺点：CPU等待时间太长，查询期间，CUP不能进行其他工作，因此CPU的效率降低。

1. **中断传送方式：**

中断：CPU在执行主程序时，转去执行突发程序，该程序执行完，再去执行主程序。

在此传送方式中，CPU不对外设进行查询，当外设需要进行数据传送且已准备就绪时才告知CPU。这是CPU中断原有程序，转去与外设传送数据，数据传送完毕后，CPU再回到原有程序继续执行。

优点：避免查询等待的时间，提高CPU的效率，应用价值极高。

**2. 引用中断的优点**

（1） 提高CPU的工作效率，实现了CPU与外部设备的并行工作

（2） 实现了实时控制

（3） 便于突发鼓掌的及时发现，提高系统的可靠性

（4）用户能通过发出中断请求，随时对运行中的计算机进行干预

**3． S51的中断嵌套与中断系统结构**

每一个中断源有一个优先级别，CPU总是最先响应级别最高的中断。

S51有两个中断优先级：高优先级和低优先级。可实现两级中断嵌套。

中断嵌套：CPU在处理一个中断请求时，发生了另一个优先级比他高的中断请求，CPU暂停原来的中断程序，转去处理优先级更高的中断请求，处理完之后，再回到原来的低级中断处理程序。

中断系统：为实现中断功能而设定的各种硬件和软件

S51的中断系统：5个中断入口，4个特殊功能寄存器（TCON、SCON的相关位作中断源的标志位，中断允许控制寄存器IE，中断优先级控制器IP）、中断顺序查询逻辑电路等组成。

**5.2.2中断源与中断标志**

**1.中断源与中断请求信号**

**中断源：向CPU发出中断请求的来源，表明中断来自何处**

S51有[5个可用的中断源](#五个中断源)，分别是外部中断2个（INT0非和INT1非）、定时器/计数器中断两个（T0,T1）和串行口中断一个。

1. 外部中断： ——通过设置定时器控制寄存器[TCON来控制](#TCON)

由外部信号引起的，一共两个中断源：外部中断0（INT0非）的中断信号引入INT0（P3.2）引脚，外部中断1（INT1非）的中断信号引入INT1（P3.3）引脚。触发方式有两种：

* 1. 低电平触发：静态方式，中断请求信号为低电平有效。CPU在每个机器周期的S5P2时刻都要对两个引脚的电平采样，若为低电平，则表示有效的中断请求信号。（采用该方式时，中断请求信号必须保持低电平知道CPU响应此中断请求位置，但在返回主程序前必须采取措施撤销此低电平）
  2. 下降沿触发：动态方式，中断请求信号负跳变有效。在S5P2时刻对两个引脚的电平采样，若相继出现两次为前高后低，则表示中断信号有效。

1. 定时器/计数器中断：

该中断发生在单片机内部，两个定时器/计数器T0和T1，以技术的方法来实现定时或计数的功能。作为定时器使用时，计数信号来自于CPU内部的机器周期脉冲；作为计数器使用时，计数信号来自于CPU的T0和T1引脚。

计数器的值从全1变为全0（计数溢出）时，就会置位溢出标志位TF0或TF1，CPU查询到后就知道有定时器/计数器的溢出中断申请。 [-----在TCON中](#TCON)

1. 串行口中断：

为串行数据传送的需要而设置的。内部自动发生，不需要芯片上设置引入端。

每当串行口发送完一组串行数据后，就会使串行发送中断标志位TI置1；接受一组数据，会使串行接收中断标志位RI置1，作为串行口中段请求标志，产生一个中断请求。  [----在SCON中](#SCON)

**2.中断标志------TCON和SCON**

S51单片机对每一个中断请求都对应有一个中断请求标志位，分别在TCON和SCON中。

CPU通过查询这些寄存器中的中断请求标志位的值来判断是否有相关的中断请求。

1. **定时器控制寄存器TCON**

TCON的字节地址为88H，可字节寻址也可位寻址

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TCON | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 位符号 | TF1 | TR1 | TF0 | TR0 | IE1 | IT1 | IE0 | IT0 |
| 位地址 | 8FH | 8EH | 8DH | 8CH | 8BH | 8AH | 89H | 88H |

1. **IE0和IE1**：外部中断请求标志位。当CPU在INT0非和INT1非引脚上采样到有效信号后，IE0或IE1置1.中断响应完成后转向中断服务后，再由硬件将该位自动清0。——可以不用理会。
2. **IT0和IT1**：外部中断请求触发方式控制位。为1时，下降沿触发，为0时，低电平触发。根据需要由软件来设置。——默认为0
3. **TF0和TF1**：定时器/计数器溢出中断请求标志位。为1时，表示对应计数器的计数溢出，相应的溢出标志位由硬件置1。计数溢出标志位的使用有两种情况：中断方式时，作为中断请求标志位来使用，在转向中断服务程序后，由硬件自动清0；查询方式时，作为查询状态位来使用，由软件清0。
4. **TR0和TR1**：定时器/计数器的运行控制位。为0时，停止T0/1的工作，为1时，启动工作。由软件设置。

1. **[串行口控制寄存器SCON](5.4%20S51的串行接口.docx" \l "SCON)**

字节地址为98H，可字节寻址，可位寻址

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SCON | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 位符号 | SM0 | SM1 | SM2 | SEN | TB8 | RB8 | TI | RI |
| 位地址 | 9FH | 9EH | 9DH | 9CH | 9BH | 9AH | 99H | 98H |

1. **TI**：串行口发送中断请求标志位。 ——发送：trasmit

发送完一帧串行数据，由硬件置1；转向中断服务程序后，需要由软件对该位清0。

1. **RI**：串行口接受中断请求标志位。 ——接收：recive

接受完一帧串行数据，由硬件置1；转向中断服务程序后，需要由软件对该位清0。

串行中断请求由TI和RI的逻辑或得到，所以，无论是发送标志还是接收标志有效，都会产生串行中断请求。

其余的含义在第十章介绍。

**5.2.1 中断向量与中断控制**

**1.中断向量**

定义：中断入口地址（中断服务程序的入口地址）。

当某个中断源的中断请求被CPU响应后，CPU便自动将该中断源的中断入口地址装入程序计数器PC，中断服务程序便从该地址开始执行，直到执行到RETI指令才重新回到主程序。

S51的各个中断源的中断入口地址是固定的。

|  |  |
| --- | --- |
| 中断源 | 中断入口地址 |
| 外部中断INT0非 | 0003H |
| 定时器/计数器T0溢出中断 | 000BH |
| 外部中断INT1非 | 0013H |
| 定时器/计数器T1溢出中断 | 001BH |
| 串行口中断 | 0023H |

**2.中断控制**

1）与中断控制有关的寄存器

S51通过设置一些特殊功能寄存器（TCON,SCON,IE,IP）来对中断信号进行锁存、屏蔽、优先级控制。

1. **中断允许控制寄存器IE。——总开关**

功能：当某一中断出现时，相应的中断请求标志位被置1，但该中断请求能否被CPU识别，则由IE的相应位的值来决定，IE对各中断源进行开放和关闭的两级控制。

IE的字节地址为A8，可字节寻址，也可以位寻址

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IE | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 位符号 | EA | — | — | ES | ET1 | EX1 | ET0 | EX0 |
| 位地址 | A8H | A9H | AAH | ABH | ACH | ADH | AEH | AFH |

* 1. **EA**：中断允许/禁止位。——**中断请求的总开关，0为禁止，1为允许**
  2. **ES**：允许/禁止串行口中断。0为禁止
  3. **ET1**：允许/禁止定时器T1中断。
  4. **EX1**：允许/禁止INT1非中断。
  5. **ET0**：允许/禁止定时器T0中断。
  6. **EX0**：允许/禁止INT0非中断。

在S51复位后，将IE寄存器清零，单片机处于关中断状态。**若要开放中断，必须使EA=1，**且相应的中断允许位为1。

例：MOV IE, # 82H -------开放定时器T0的溢出中断（下两条语句功能与该条相同）

SETB EA

SETB ET0

1. **中断优先级控制寄存器IP。**

当几个中断源同时向CPU发出中断请求时，C**PU总是先响应优先级别最高的中断。**

S51有两个中断优先级：高优先级和低优先级，均可用软件设置。IP=0时，为低优先级；=1时，为高优先级。高优先级可以中断一个正在执行的低优先级程序，实现两级嵌套，但同级或低优先级中断源不能中断正在执行的中断服务程序。

IP的字节地址为B8，可按字节寻址，也可按位寻址

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| IP | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
| 位符号 | — | — | — | PS | PT1 | PX1 | PT0 | PX0 |
| 位地址 | BFH | BEH | BDH | BCH | BBH | BAH | B9H | B8H |

1. **PS**：串行口中断优先级控制位。=1时，串行口中断指定为高优先级
2. **PT1**：定时器/计数器T1中断优先级控制位
3. **PX1**：外部中断INT1非中断优先级控制位
4. **PT0**：定时器/计数器T0中断优先级控制位
5. **PX0**：外部中断INT0非中断优先级控制位

注：当几个同级的中断源提出中断请求，CPU同时收到几个同一优先级的中断请求时，谁先被执行，取决于单片机内部的硬件查询顺序，其硬件查询顺序形成了中断的自然优先级。顺序为：INT0非，T0，INT1非，T1，串行口中断

**2）中断处理过程**

三个阶段：中断响应、中断处理、中断返回

1. **中断响应**。

S51中断响应的条件如下：

* + - * 1. 当前不处于同级或更高级中断响应中
        2. 当前机器周期必须处于当前指令的最后一个机器周期，否则等待。（把该指令执行完，才能响应中断）
        3. 如果当前指令是中断返回指令RETI，或读/写 IE、IP，则必须**再执行一条指令后**才能响应中断请求。

若不满足以上条件，则忽略该机器周期对中断标志的查询结构，下一个机器周期继续查询。若满足，则下一个机器周期的S1状态进入中断响应过程。CPU首先将断点压入堆栈，然后执行一条由中断系统提供的硬件LCALL指令，自动转至相应的中断服务程序入口。

注：由于中断服务程序入口地址仅隔8B，难以容纳中断服务程序，所以在入口处安排一条跳转指令LJMP或AJMP

1. **中断处理**:CPU响应中断后，转至中断服务程序的入口，开始执行程序，直到RETI为止。编程要注意一下事项：

各中断程序的入口地址系统有统一的规定，用户不能自定义。

在进行中断服务前，必须保护现场。做法：将服务程序中用到的有关寄存器的内容压入栈，待中断完成返回后，再弹出。

中断服务程序必须以RETI结束。执行RETI之前，SP的值应该与中断入口时SP的值相同（保证PUSH和POP成对使用）

在编译内部串行口中断服务程序时，测试RI或TI标志后，应用软件将它们复位（清零）

1. **中断返回**：中断处理程序的最后一条指令是中断返回指令RETI。

RETI的功能：将断点弹出送回PC中，使程序能返回原来被中断的主程序继续执行。

**注：S51的RETI指令出了弹出断点之外，还通知中断系统：已经完成响应的中断处理。**