中断优先级与中断响应时间

渤海三叠浪　*lch_29@163.com*　2009-12-2

Keil C51.exe V8.18　晶振 22.1184Mhz

看如下程序调试：（图 1）

```
01 #include <reg52.h> #include <string.h>
02 #include "user.h"    INT8U i;
03 void Timer0_Server(void) interrupt 1
04 {
05     RI = TRUE;
06     TH0 = 0x4C; TL0 = 0x00;//每4ms进中断一次
07 }
08 void UART_Server(void) interrupt 4
09 {
10     if (RI) {
11         RI = FALSE;
12     }
13 }
14 void main(void)
15 {
16     INT8U i;
17     TMOD |= 0x01;TH0  = 0x4C;TL0  = 0x00;TR0  = 1;ET0  = 1;
18     TMOD |= 0x20;TH1  = 250;TL1  = 250;SCON = 0x50;
19     PCON = 0x00; TR1  = 1;ES   = 1;
20 //   PS = 1;//将串口设置为高优先级别
21
22
23     RI = TRUE;
24     TF0 = TRUE; //定时器0溢出
25     EA = 1;
26     while (1) {
27         i = 1;
28         i = 1;
29         i = 1;
30         i = 1;
31     }
32 }
```

要注意到：（1）程序中将“串口设置为高优先级别”这句话注销了

（2）定时器是 16 位手动装载，每 4ms 进入中断一次！事实上啥方式、啥时间都无所谓，因为下面所讨论的问题都是以 us（机器周期）为单位的。

(3)调试时，要用软件调试，因为硬件调试时可能会不准确，比如说晶振本身的误差。要用 stepinto 进行单步调试，并且软件优化级别设置为 0

分析：main 里面令 EA = 1,也就是将串口中断和定时器 1 中断同时打开(哪个先打开与 RI = TRUE 和 TF0 = TRUE 这两句话的语句次序无关，可以交换位置看看就知道了)，那么首先进入的是定时器中断。因为在相同的优先级下（系统复位后，IP = 0xx00000,也就是都是低优先级），先响应优先次序高的中断（定时器 0 的优先顺序序号是 2，而串口的序号是 5）。

在执行完 EA = 1 后，又执行了 I = 1；那么为什么不是执行 E = 1 后马上执行中断服务程序 Timer_Serve 里面的语句呢？这就引出了另一个概念——中断响应时间。就是说硬件置位中断请求标志位后（当然咱这里是用软件置位的标志位来模拟硬件置位标志位），需要一定的时间（最快 3 个机器周期，一般小于 8 个机器周期）才会进入中断程序。（孙育才《MCS-51 系列单片微型计算机及其应用

（第三版）P145 页》（第一版 1986 年/第二版 1989 年/第三版 1997 年）），或者看哈工大的张毅刚的书，书中也讲了相同的内容。

接着继续单步调试（图 2）

```
01 #include <reg52.h> #include <string.h>
02 #include "user.h"    INT8U i;
03 void Timer0_Server(void) interrupt 1
04 {
05      RI = TRUE;
06      TH0 = 0x4C; TL0 = 0x00;//每4ms进中断一次
07 }
08 void UART_Server(void) interrupt 4
09 {
10      if (RI) {
11           RI = FALSE;
12      }
13 }
14 void main(void)
15 {
16      INT8U i;
17      TMOD |= 0x01;TH0   = 0x4C;TL0   = 0x00;TR0   = 1;E
18      TMOD |= 0x20;TH1   = 250;TL1   = 250;SCON = 0x50;
19      PCON = 0x00; TR1   = 1;ES    = 1;
20 //   PS = 1;//将串口设置为高优先级别
21
22
23      RI = TRUE;
24      TF0 = TRUE; //定时器0溢出
25      EA = 1;
26      while (1) {
27           i = 1;
28           i = 1;
29           i = 1;
30           i = 1;
31      }
32 }
```

执行完后定时器 0 中断后，没有直接进入串口中断，而是返回了 while 里面。为什么呢？当然还是中断响应需要时间了！

预计再执行几个 i = 1 就会进入串口中断了！

继续单步调试（图 3）

```
01 #include <reg52.h> #include <string.h>
02 #include "user.h"    INT8U i;
03 void Timer0_Server(void) interrupt 1
04 {
05      RI = TRUE;
06      TH0 = 0x4C; TL0 = 0x00;//每4ms进中断一次
07 }
08 void UART_Server(void) interrupt 4
09 {
10      if (RI) {
11           RI = FALSE;
12      }
13 }
14 void main(void)
15 {
16      INT8U i;
17      TMOD |= 0x01;TH0   = 0x4C;TL0   = 0x00;TR0   = 1;ET0
18      TMOD |= 0x20;TH1   = 250;TL1   = 250;SCON = 0x50;
19      PCON = 0x00; TR1   = 1;ES    = 1;
20 //   PS = 1;//将串口设置为高优先级别
21
22
23      RI = TRUE;
24      TF0 = TRUE; //定时器0溢出
25      EA = 1;
26      while (1) {
27           i = 1;
28           i = 1;
29           i = 1;
30           i = 1;
31      }
32 }
```

果然又执行了两个 i = 1 就进入了串口中断

继续单步调试 （图 4）

```
#include <reg52.h> #include <string.h>
#include "user.h"    INT8U i;
void Timer0_Server(void) interrupt 1
{
     RI = TRUE;
     TH0 = 0x4C; TL0 = 0x00;//每4ms进中断一次
}
void UART_Server(void) interrupt 4
{
     if (RI) {
          RI = FALSE;
     }
}
void main(void)
{
     INT8U i;
     TMOD |= 0x01;TH0   = 0x4C;TL0   = 0x00;TR0   =
     TMOD |= 0x20;TH1   = 250;TL1   = 250;SCON = 0
     PCON = 0x00; TR1   = 1;ES    = 1;
//   PS = 1;//将串口设置为高优先级别


     RI = TRUE;
     TF0 = TRUE; //定时器0溢出
     EA = 1;
     while (1) {
          i = 1;
          i = 1;
          i = 1;
          i = 1;
     }
}
```

现在设置不同的优先级别看看调试情况

将串口中断优先级别设置为高（图 5）

```
#include <reg52.h> #include <string.h>
#include "user.h"    INT8U i;
void Timer0_Server(void) interrupt 1
{
     RI = TRUE;
     TH0 = 0x4C; TL0 = 0x00;//每4ms进中断一次
}
void UART_Server(void) interrupt 4
{
     if (RI) {
          RI = FALSE;
     }
}
void main(void)
{
     INT8U i;
     TMOD |= 0x01;TH0   = 0x4C;TL0   = 0x00;TR0   = 1;ET0   = 1;
     TMOD |= 0x20;TH1   = 250;TL1   = 250;SCON = 0x50;
     PCON = 0x00; TR1   = 1;ES    = 1;
     PS = 1;//将串口设置为高优先级别

     RI = TRUE;
     TF0 = TRUE; //定时器0溢出
     EA = 1;
     while (1) {
          i = 1;
          i = 1;
          i = 1;
          i = 1;
     }
}
```

继续（图 6）

```
#include <reg52.h> #include <string.h>
#include "user.h"    INT8U i;
void Timer0_Server(void) interrupt 1
{
    RI = TRUE;
    TH0 = 0x4C; TL0 = 0x00;//每4ms进中断一次
}
void UART_Server(void) interrupt 4
{
    if (RI) {
        RI = FALSE;
    }
}
void main(void)
{
    INT8U i;
    TMOD |= 0x01;TH0  = 0x4C;TL0  = 0x00;TR0  = 1;
    TMOD |= 0x20;TH1  = 250;TL1  = 250;SCON = 0x5
    PCON = 0x00; TR1  = 1;ES   = 1;
    PS = 1;//将串口设置为高优先级别

    RI = TRUE;
    TF0 = TRUE; //定时器0溢出
    EA = 1;
    while (1) {
        i = 1;
        i = 1;
        i = 1;
        i = 1;
    }
}
```

----------下图是图 7--------------------

```
#include <reg52.h> #include <string.h>
#include "user.h"    INT8U i;
void Timer0_Server(void) interrupt 1
{
    RI = TRUE;
    TH0 = 0x4C; TL0 = 0x00;//每4ms进中断一次
}
void UART_Server(void) interrupt 4
{
    if (RI) {
        RI = FALSE;
    }
}
void main(void)
{
    INT8U i;
    TMOD |= 0x01;TH0  = 0x4C;TL0  = 0x00;TR0  = 1;ET0  = 1;
    TMOD |= 0x20;TH1  = 250;TL1  = 250;SCON = 0x50;
    PCON = 0x00; TR1  = 1;ES   = 1;
    PS = 1;//将串口设置为高优先级别

    RI = TRUE;
    TF0 = TRUE; //定时器0溢出
    EA = 1;
    while (1) {
        i = 1;
        i = 1;
        i = 1;
        i = 1;
    }
}
```

仔细看看下一步执行的是啥？图 8

```
#include <reg52.h> #include <string.h>
#include "user.h"   INT8U i;
void Timer0_Server(void) interrupt 1
{
    RI = TRUE;
    TH0 = 0x4C; TL0 = 0x00;//每4ms进中断一次
}
void UART_Server(void) interrupt 4
{
    if (RI) {
        RI = FALSE;
    }
}
void main(void)
{
    INT8U i;
    TMOD |= 0x01;TH0  = 0x4C;TL0  = 0x00;TR0  = 1;ET0  = 1
    TMOD |= 0x20;TH1  = 250;TL1  = 250;SCON = 0x50;
    PCON = 0x00; TR1  = 1;ES   = 1;
    PS = 1;//将串口设置为高优先级别

    RI = TRUE;
    TF0 = TRUE; //定时器0溢出
    EA = 1;
    while (1) {
        i = 1;
        i = 1;
        i = 1;
        i = 1;
    }
}
```

看出高优先级的威力的吧！

执行了定时器中断程序的 **RI = TRUE** 后，没有执行 **TH0** 和 **TL0** 的赋值就迫不及待的进入了串口中断了。这说明了：

（1） 串口优先级别是高级别，而定时器中断是低级别。假如两个都是低级别或者定时器高级别，串口低级别，那么肯定是执行了 **TH0** 和 **TH1** 赋值后，再可能经过更长的响应时间（就是说可能会跳到 **while** 里面执行 **i=1**），再进入串口中断服务程序的。

（2）由于串口设置了高优先级，串口中断响应时间变短了（当然响应时间和编译器的编译能力有关，如果编译器的优化级别设置更高，响应时间可能更短，因为响应时间和汇编指令有关）。因为：如果响应时间长，那么肯定会先执行 **TH0** 和 **TH1** 的赋值语句的。

[注]：要将本文中的 while 里面的 i=1 换成_nop_,并根据生成的反汇编指令和每条指令执行的时间就会得出更精确的调试结论，也可以看出软件仿真是多么的强大-----竟然精确到 1us。

渤海三叠浪 2009-12-2