TKS B系列专业仿真器“Trace”功能的妙用

（2004/8/6 V1.0）

本章节主要阐述了在Keil IDE（μVision2/μVision3）集成开发环境下， TKS-B系列专业仿真器中“Trace”功能的特点和使用方法。

1. “Trace”的定义解释

“trace”在英文中是“踪迹”的意思，在调试环境中，跟踪已经被执行过的指令的运行轨迹就叫做“trace”。本文中所阐述的“Trace”是仿真器中使用的先进技术，主要用于查看程序运行中各种状态的连续变化，其主要用途是跟踪程序指针的运行轨迹，便于用户分析程序的流向。

由于“Trace”需要连续记录程序的运行状态，因此需要一个高速缓冲区来进行实时数据记录。高档仿真器一般具备大的缓冲区，价格也高。然而，无论“Trace”的高速缓冲区有多大，在程序实时运行后将很快溢出。因此，仿真器中的“Trace”一般都采用“向后记录”的方式，溢出后最前面的数据（旧数据）将被新数据覆盖。所以，程序运行中“Trace”缓冲区域中总是记录程序最新的状态数据。当程序运行被中断（断点/夭折/单步）后，用户可以查看缓冲区域中的状态数据，用户可以根据这些数据来分析/排除故障。

2. TKS-B中“Trace”的特点

TKS仿真器B系列内部有64K的“Trace”缓冲区，记录用户程序的运行流程，最大可以记录64K条用户指令。当程序运行被中断仿真器进入监控状态时，“Trace”缓冲区中记录了前64K条指令的运行轨迹情况；如果当前实际运行的程序没有超过64K条，则按照实际记录的有效条数计算。**注意**：64K的“Trace”记录在复位后将全部清除。

3. 使用方法

“Trace”功能是在硬件仿真环境中，通过使能跟踪记录和打开“Trace”窗口实现的。**注意**：用户只能在反汇编窗口观察跟踪记录，这是因为“Trace”只能以汇编指令记录，所以只能以汇编的形式显示，因而在uV2中，跟踪窗口会在反汇编窗口显示出来。

下面举一个例子具体说明“Trace”的使用方法和功能。

以下的例程“trace.c”使用了定时器T0中断，在主程序中会不断进入定时中断服务程序。使用TKS-58B对其进行硬件仿真，并使用“Trace”功能，可以清楚地看出程序的运行流向。

```
/*********************************************************************
 函数：trace.c
 功能：使用了定时器/计数器0的模式0，在主程序中会不断地进入定时中断服务程序
 说明：在硬件仿真中使用"Trace"功能，可以清楚地看出程序的运行流向
*********************************************************************/
#include"reg51.h"
```

```
#define uchar unsigned char
uchar a1=0,a2=0,a3=0;
void t0_int(void) interrupt 1
    {
        TH0=0xff;                          //设置定时器初值为0xfffc
        TL0=0xfc;
        a3++;
    }
void main(void)
    {
        TMOD=0x01;                         //设置T0的M1=0，M0=1，C/T=0，GATE=0
        TH0=0xff;                          //设置定时器初值为FFFCH
        TL0=0xfc;
        EA=1;
        ET0=1;
        TR0=1;                             //启动定时器0，开始计数
        while(1)
            {
                a1++;
                while(!TF0);
                a2++;
            }
    }
```

使用“Trace”功能的步骤：

1.仿真环境设置完成后，点击菜单“Debug→Start/Stop Debug Session”或快捷图标 进入硬件仿真环境。

2.点击菜单“Debug→Go”或快捷图标 全速运行，等待数秒后点击菜单“Debug→Stop Running”或快捷图标 停止运行，当仿真器进入监控状态后，点击菜单“Debug→Enable/Disable Trace Recording” 或快捷图标 开启/关闭跟踪记录。

3.在反汇编窗口中，点击菜单“Debug→View Trace Records” 或快捷图标 观察存在的跟踪记录。

按照上述步骤，进入“Trace”功能，其源程序和“Trace”窗口，如图1所示。

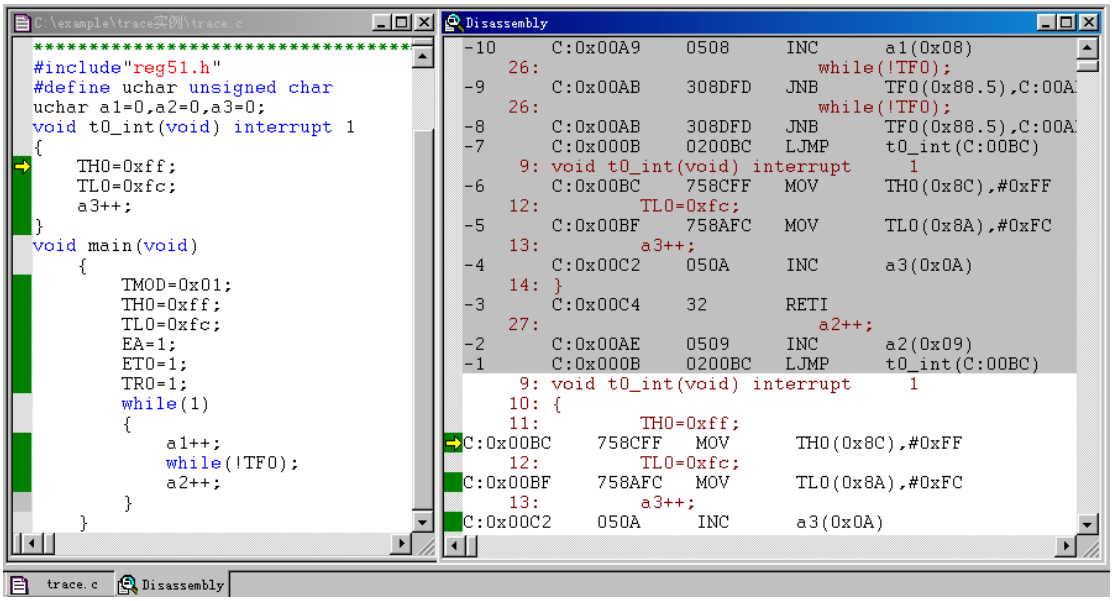

图1 硬件仿真环境中源程序和“Trace”窗口

从“Trace”窗口中，用户可以非常清楚地看出程序的运行流向。被覆盖的区域中每一个反汇编程序行的前方都有一个“－n”（n为1、2、3…），它表明了程序执行的先后次序。“－1”表明该行是刚执行过的上一个程序行，“－2”表明是在“－1”前执行的那个程序行，依此类推。

在本例程中，通过观察“Trace”窗口，程序运行的先后顺序一目了然——程序从主函数void main(void)运行到定时器中断服务函数void t0_int(void)，然后又返回到主函数，如此往复。查看跟踪记录，用户可以很清楚地了解程序的运行，方便用户的查错与纠错。

另外，从图1中可以看出，将要运行的代码和“Trace”记录的已运行的代码总是相互紧密连接在一起的。这是因为在Keil IDE（μVision）**集成开发**环境下，“Trace”使用了**覆盖嵌接技术**。**注意：**由于程序中存在跳转和源程序位置的变动，这种覆盖嵌接有时会发生错位或引起“Trace”窗口的关闭，此时需要用户再次点击菜单“Debug→View Trace Records” 或快捷图标  来调整“Trace”窗口的位置。

由于窗口尺寸有限制，当前反汇编窗口只能显示64K跟踪信息的一小部分。用户可以使用计算机键盘上“Page Up/Page Down”键来变换“Trace”窗口的显示内容，也可以使用鼠标点击反汇编窗口中右上角/右下角的上移/下移箭头。**注意**：反汇编窗口右边的滑动块不能用来移动“Trace”的显示内容，否则会退出“Trace”窗口。

4. 局限性

由于产品价格/定位的原因，“Trace”的触发/停止依赖于用户程序的运行/停止，因此不能实时对跟踪记录进行观察，用户需要在硬件仿真环境中停止程序运行之后才能观察。