

TKS 仿真器 B 系列快速入门

(2004/10/20 V2.0)

目录

- 一、 TKS 仿真器 B 系列技术特点和背景资料
- 二、 TKS 仿真器 B 系列的技术优势
- 三、 TKS 仿真器 B 系列型号和功能分类
- 四、 TKS 仿真器 B 系列使用前应该了解的知识
- 五、 TKS 仿真器 B 系列在 Keil 中的快速操作
- 六、 TKS 仿真器 B 系列涉及的技术参数及处理
- 七、 TKS 仿真器 B 系列的物理结构
- 八、 仿真头组件的使用
- 九、 TKS 仿真器 B 系列的限制
- 十、 TKS 仿真器 B 系列使用中的常见问题
- 十一、 TKS 仿真器 B 系列的升级
- 十二、 结束语

警告：该文档的内容可能会在以后发生改变，TKS 的用户需要参照随机提供的电子文档为标准。

一、TKS 仿真器 B 系列技术特点和背景资料

TKS 仿真器 B 系列是广州致远电子有限公司在 Philips 和 Keil 公司支持下推出业界领先的系列仿真器，采用当前最先进的 HOOKS 仿真技术，设计独到的仿真性能处于全球的全面领先水准。

硬件上采用 Philips 公司 MCU 设计部门的经验，具备**高度运行稳定性/芯片兼容性**。运行频率突破 HOOKS 技术的极限，达到前所未有的 **32MHz**。低电压仿真方面性能卓越，可以**稳定运行在 2.0V** 以下。内部的部件经过全面优化后，能以较低的价格支持多项高级仿真功能。每种型号的仿真器均能支持多种 80C51 单片机的仿真，并能直接或通过简单的升级支持仿真 Philips 陆续推出的全系列 80C51 内核单片机。

软件上支持 TKStudio/Keil 中英文双平台，并首次在 Keil 公司的 uVision2/uVision3 上稳定实现 64K 超大容量 **Trace 接口/4x64K 代码数据覆盖/加彩运行轨迹显示/4x64K 运行断点/超精密运行时间显示**等多项激动人心的超级仿真功能，使 uVision 用户体验以往只能在软件仿真下实现的仿真功能。B 系列在 TKStudio 环境中表现更佳，并能实现数据覆盖分析/加彩显示/高级语法分析等更多附加功能，其中 TKSsitant 智能导向技术将简化用户设计难度。

- 真正无缝嵌接 Keil/TKStudio 中英文双开发/调试平台，实现多种高级调试功能。
- 采用 Philips 授权的 HOOKS 仿真技术，可靠仿真标准 P0/P2 口特性。
- 采用超高稳定 IO/总线设计，仿真性能异常稳定。
- 内部线形可调 32KHz-50MHz **任意时钟发生器**，点精度 0.001，满足所有特殊时钟要求。
- 高达 115200bps 高速串口通讯，提高仿真运行/代码下载速度。
- 仿真绝对不占用任何用户资源，包括堆栈/内部 RAM/SFR 等。
- 业界领先的超低电压仿真，稳定仿真至 **2.0V** 以下。
- 业界领先的超高速仿真设计，实时运行频率高达 **0-32MHz[®]12Clock / 0-26MHz[®]6Clock**。
- 超精密运行时间/周期显示，0-32MHz 范围内最高时间精度为 1 clock。
- 内部代码空间/数据空间均支持 1byte 精度的 ReMap 功能，方便用户仿真不同资源的芯片。
- 支持运行仿真器外部/外部 0-64K 的 ROM 和 RAM。
- 支持 64K 全范围的冯诺曼 **Von Neumann** 结构。
- 64K 的仿真器内部程序存储器/64K 的数据存储器。
- 64K 超大容量实时 **Trace** 功能，协助用户分析程序运行轨迹。
- **4×64K** 用户实时断点，方便用户特殊仿真要求。
- 64K 全地址范围内的代码覆盖分析，**加彩运行轨迹**显示。
- 64K 全地址范围内的代码读取覆盖显示。
- 64K 全地址范围内的数据读覆盖显示。
- 64K 全地址范围内的数据写覆盖显示。
- 采用高速信号跟踪技术，真正支持 ALE 静态关闭或动态关闭，不限制对 ALE 信号的非常规切换。
- 自动感知 6Clock/12Clock 时钟，并支持动态切换和静态切换，不限制用户对时钟信号的非常规切换。
- 所有仿真器进出口线 100% 保护，避免使用中误操作引起仿真器的损坏，保护用户投资。
- 支持用户目标板的时钟输入和用户目标板的振荡晶体，并进行时钟有效性检查。
- 支持外部复位信号，并在全速运行中有效(可选择为关闭)，方便用户调试外部看门狗。
- 丰富的提示信息，自动感知当前仿真任何异常，帮助用户准确了解当前 MCU 运行状态。
- 采用自主知识产权的 HOOKS 时序同步技术，有效减小干扰引起的程序跑飞，适合于长时间程序运行。
- 采用与业界不同的 I/O 口重造技术，再现标准 MCU 的 I/O 特性而无任何限制，用户可以任意在 IO/总线 2 种状态中切换。
- TKSsistant 智能设计导向工具，帮助用户轻松完成设计。
- 单机支持仿真芯片最多。支持全系列标准 8051 芯片仿真（包括最新推出的低电压 V 系列）。

P8xC5x:	P80C31、P80C32 P87C51、P87C52、P87C54、P87C58 P89C51、P89C52、P89C54、P89C58
P8xC5xX2:	P80C31X2、P80C32X2 P87C51X2、P87C52X2、P87C54X2、P87C58X2 P89C51X2、P89C52X2、P89C54X2、P89C58X2
P8xC5xFx:	P80C51FA P87C51FA、P87C51FB、P87C51FC
P87CL5xX2:	P87CL52X2、P87CL54X2
P89C6x:	P89C60X2、P89C61X2
P8xC51Rx+:	P80C51RA+ P87C51RA+、P87C51RB+、P87C51RC+、P87C51RD+
P8xC51Rx2:	P87C51RA2、P87C51RB2、P87C51RC2、P87C51RD2 P89C51RA2、P89C51RB2、P89C51RC2、P89C51RD2
P89V51Rx2:	P89V51RD2 (看门狗/SPI/时钟切换/工作频率不能完全仿真)
P89LV51Rx2:	P89LV51RD2 (看门狗/SPI/时钟切换/工作频率不能完全仿真)
P8xC66x:	P87C660X2、P87C661X2 P89C660、P89C662、P89C664、P89C668
P8xC591:	P83C591、P87C591
AT89C5x:	AT89C55、AT89C1051/2051/4051
AT89S5x:	AT89S51、AT89S52 (不能仿真 WDT)
W78E5x:	W78E51、W78E52、W78E54、W78E58
HY97C5x:	HY97C51、HY97C52、HY97C54、HY97C58

以及其它所有公司的标准 80C51 系列单片机。并直接支持 PHILIPS 公司陆续推出的标准 44 引脚（或 40 脚）单片机的精确仿真。

注：TKS 仿真器 B 系列根据型号的不同，支持仿真的 MCU 种类不同。

二、TKS 仿真器 B 系列的技术优势

● 为什么要选用 HOOKS 技术的仿真器？

HOOKS 仿真技术是 PHILIPS 公司的专利仿真技术，HOOKS 本身技术复杂，但却能同时仿真不同的 MCU 而不需要制作众多的仿真芯片。同当前流行的 BondOut 仿真技术相比，在制作仿真器时成本和难度要增加很多。在国内和国外有很多仿真器厂家采用 HOOKS 仿真技术设计仿真器，但是在解决性能稳定性，芯片兼容性，I/O 口重造的真实性上遇到了较多的问题。

10 年前，PHILIPS 半导体公司将 80C51 系列高性能单片机介绍给国内用户的同时，也痛心国内众多的仿真器厂家竟无一能推出质优价廉的 HOOKS 仿真器。在面对国内用户着急的询问 PHILIPS 芯片仿真器的解决方案时，只能无可奈何的告诉用户使用普通的仿真器替代，或者采用国外 1000 多美金的高档仿真器。同时，广州致远电子有限公司投入大量的人力和资金开始研制 HOOKS 仿真器，并试图全面提高 HOOKS 仿真器的性能和减低成本。

在长达几年的研制中，逐渐掌握了 HOOKS 仿真的关键技术，并同 PHILIPS 的技术支持部门配合，针对了现在 HOOKS 技术中的限制，提出了 HOOKS 的自同步技术并率先在 TKS 系列仿真器中采用，HOOKS 仿真器的稳定性问题得到彻底解决；

在以往的 HOOKS 仿真器中，仿真差异较大的单片机时，如 P89C51 和 P89C51RD2 需要更换仿真头来解决或者购买另外一种型号，由于 HOOKS 仿真器本身价格较高，因此即便是更换一个仿真头也会给用户带来较大的经济负担。在多年的研制中，降低成本始终也是研制的重要目标。在当前 TKS 的仿真器中，根据价格的不同都能兼容一个系列的单片机。例如性能最好的 TKS-668B，能兼容几乎所有 PHILIPS 的 80C51 内核单片机，并能够仿真 PHILIPS 不断推出的新型号。这种兼容的性能将给用户带来的方便和利益是巨大的。

在 HOOKS 技术中的难点 I/O 口重造上，TKS 系列将标准 80C51 内核的 I/O 口部分完全移植过来，在使用中对用户没有任何限制。

经过精心设计的 TKS 系列 HOOKS 仿真器，在较低的价格下保持了很高的仿真性能。用户可以根据自己的仿真范围选择相应型号。如果用户使用的 PHILIPS 单片机型号较多，则可以选择当前支持器件最多的 TKS-668，该款仿真器可直接支持目前几乎所有标准 80C51 的仿真，也是采用 HOOKS 技术的 TKS 系列中性价比最高的一款。根据 PHILIPS 的 51 内核单片机系列的衍生标准，TKS-668 也将能支持以后陆续推出的其它型号。

PHILIPS 是全球最大的 80C51 单片机供应商，它的 80C51 内核产品除了标准的 MCS-51 外，绝大多数属于增强型的 80C51 单片机。它的增强性能，例如 6 CLK 的机器周期，ALE 的可关闭，软件复位，看门狗，PCA，内部附加的 Xram、I²C 总线、AD/DA、PWM 及 SPI 等，这些功能使用标准的 80C51 仿真器是不可能仿真的，如果使用专用的仿真器也不可能制作如此众多的仿真芯片。因此，采用 HOOKS 技术的仿真器在仿真不断推出的增强型 80C51 芯片就有其独特的优势。TKS 的 HOOKS 仿真器在设计时充分考虑的将来的兼容问题，在最大程度上保证用户可以不用升级或经过简单的升级即可仿真 PHILIPS 不断推出的新型 80C51 芯片，在最大程度上保证用户的投资利益。

● 为什么要选用 TKS 仿真器 B 系列？

TKS 仿真器 B 系列不但采用了比较先进的 HOOKS 仿真技术，在经过精心设计后，仿真的性能和稳定性都得到很大的提高。同国内普遍采用的通用 BondOut 技术相比较，优点非常明显。TKS 采用的仿真技术在以下几个方面表现优异，代表全球 HOOKS 仿真技术的领先水平，其技术的先进性得到 HOOKS 技术提供厂商 PHILIPS 公司技术部门的认可。

■ **超低电压仿真性能 2.0-5.5V。**

经过 PHILIPS 技术部门的认定 TKS 的低电压仿真性能是 HOOKS 技术仿真器中最优异的，在最坏的电源输入情况下能忍受外部电源的异常波动而不丢失用户的仿真状态。TKS 内部设有电压跟踪电路，能自动跟踪外部输入电源变化，用户电源电压的变化能及时从 I/O 反映出来。

■ **超高速仿真设计，实时运行频率高达 0-32MHz@12Clock / 0-26MHz@6Clock。**

B 系列仿真器内部使用超高速器件设计，实时仿真运行频率达到 HOOKS 技术的极限频率。由于内部使用了独特的 HOOKS 同步技术，在极限运行频率下，同样能得到稳定可靠的仿真效果。

■ **内部线形可调 20KHz-50MHz 任意时钟发生器，点精度 0.001，满足所有特殊时钟要求。**

B 系列仿真器内置一颗精密时钟发生器，可以产生范围在 20KHz-50MHz 的任意时钟频率。这种功能能给用户在仿真时带来极大的方便性，可线性调整的精密时钟可以帮助用户形成特殊要求的时钟频率，也可以方便用户测试整个系统的频率性能曲线。

■ **内部代码空间/数据空间均支持 1byte 精度的 ReMap 功能，方便用户仿真不同资源的芯片。**

B 系列仿真器真正支持仿真不同资源的芯片。仿真器内部预置 64K 的 ROM 空间和 64K 的 RAM 空间，用户可以根据要仿真芯片的种类自己定义空间资源分配的情况。例如，用户可以把 ROM 空间选择为 89C52 模式，内部 4K ROM 为仿真器提供，外部的 60K 则由用户板提供，**B 系列仿真器具备运行仿真器外部程序的能力**。对于 64K 的 RAM 空间，用户也可以根据自己的需要设置成仿真器内部或外部，仿真器将根据用户的配置自动从内部或外部读取数据。

■ **支持 64K 全范围的冯诺曼 Von Neumann 结构。**

B 系列仿真器支持 64K 地址范围的冯诺曼结构。用户可以把程序的全部或部分设置成冯诺曼机构，则该空间既可以作为程序空间运行代码，也可以作为数据空间来读写。这种功能在仿真用户的动态加载程序时非常方便。

■ **4×64K 用户实时断点，方便用户特殊仿真要求。**

B 系列仿真器具备 4×64K 的断点，用户可以根据自己的需要设置不同的断点来调试程序。

- 64K 程序运行断点：程序运行到该位置时停止运行。
- 64K 代码读取断点：MOVC 指令读取该地址数据时停止运行。
- 64K 数据读取断点：MOVX 指令读取该地址数据时停止运行。
- 64K 数据写入断点：MOVX 指令写入该地址数据时停止运行。

用户可以单独或混合使用上述的 4 种断点，每种断点的使用个数最大为 64K。

■ **64K 地址范围内的加彩运行轨迹显示。**

B 系列仿真器具有 64K 地址范围内的加彩运行轨迹显示，用户程序运行过的位置将以彩色轨迹显示，方便用户了解程序运行的分支流程。

■ **4×64K 全地址范围内的覆盖。**

B 系列仿真器具有 4×64K 地址范围内的覆盖显示，分别是：

- 代码运行覆盖：代码运行过的覆盖显示。
- 代码读取覆盖：MOVC 读取过的覆盖显示。
- 数据读取覆盖：MOVX 读取过的覆盖显示。
- 数据写入覆盖：MOVX 写入过的覆盖显示。

■ ALE 信号的切换。

TKS 是全球首家支持 ALE 关闭的仿真器厂商，用户在监控状态/运行状态中都可以任意施加对 ALE 的关闭/开启而不会影响当前的运行状态。

■ 6/12Clock 时钟信号的切换。

TKS 是全球首家/国内唯一支持高速时钟切换的仿真器厂商，用户在监控状态/运行状态中都可以任意施加 6/12Clock 时钟信号的切换而不会影响当前的运行状态。6/12Clock 时钟信号切换功能被更多的 MCU 厂家接受并采用后，TKS 的这种高性能仿真特性给用户带来更多的方便。

■ 准确的 I/O 口特性。

HOOKS 技术的“瓶颈”在于 I/O 口的准确重造，采用通常的技术不能支持 I/O 和总线的动态交替使用，在这种情况下用户只能限制选择 I/O 或总线的一种状态。另外一些厂家的折衷措施将引起总线方式下驱动能力的不足和 I/O 口方式下的驱动过强。TKS 内部采用时序监控技术，有效解决 IO/总线的分时复用问题，真正对用户 P0/P2 总线口的使用没有任何限制。

● 选用 HOOKS 仿真器如何仿真其它公司的 80C51 芯片？

HOOKS 是 PHILIPS 公司开发仿真技术，不但能精确仿真 PHILIPS 的全系列 80C51 单片机，而且也能直接或完全精确仿真其它公司的 80C51 单片机。这是由于 PHILIPS 是全球最大的 80C51 单片机供应商，80C51 的产品线能覆盖绝大多数的应用领域。在仿真其它公司的 80C51 单片机时，除了特殊型号或用途的单片机，一般在 PHILIPS 的 80C51 单片机中都能找到完全相同或基本类似的仿真型号。

我们建议用户在系统设计期间尽量选择 PHILIPS 的 80C51 单片机系列，而不要选择一些 80C51 产品线很短的系列。这是因为 PHILIPS 的各种型号的 80C51 单片机都有仿真器支持，这在设计大系统中是非常关键的。在大系统设计验证后期，系统任何一个问题的排除都变得非常复杂。如果没有类似仿真器的工具支持，用户将耗费大量的精力和物力去排除一个细小的问题。

如果用户确实想使用 HOOKS 仿真器来仿真其它公司的 80C51 单片机型号，请致电我们的技术支持，我们可以根据您的实际要求为您提供相应的 PHILIPS 单片机型号以及开发工具的解决方案。

三、TKS 仿真器 B 系列型号和功能分类

采用 PHILIPS 授权 HOOKS 仿真技术的 TKS 系列仿真器 B 系列目前的型号见表 1，广州致远电子有限公司将陆续推出采用 HOOKS 技术的仿真器，全面支持 PHILIPS 和其它公司芯片的仿真。

表 1 TKS 仿真器 B 系列型号列表

型号	TKS-52B	TKS-58B	TKS-61B	TKS-668B	TKS-591B
仿真技术	HOOKS	HOOKS	HOOKS	HOOKS	HOOKS
仿真程序空间 64K	内外任意配置	内外任意配置	内外任意配置	内外任意配置	内外任意配置
仿真数据空间 64K	内外任意配置	内外任意配置	内外任意配置	内外任意配置	内外任意配置
任意时钟发生器	√	√	√	√	√
彩色运行轨迹显示	√	√	√	√	√
4×64K 代码/数据覆盖	√	√	√	√	√
4×64K 任意断点	4×64K	4×64K	4×64K	4×64K	4×64K
冯·诺曼结构支持	√	√	√	√	√
64K 实时跟踪支持	-	√	√	√	√
外部电源输入支持	2.0-5.5V	2.0-5.5V	2.0-5.5V	2.0-5.5V	2.0-5.5V
外部复位输入支持	√	√	√	√	√
仿真芯片可更换性	√	√	√	√	√
实时仿真频率	0-32MHz	0-32MHz	0-32MHz	0-32MHz	0-32MHz
支持仿真范围	仿真标准 51	仿真标准 51	除 87C591 外	除 87C591 外	仿真芯片无限制
6/12 clk 动态切换	√	√	√	√	√
ALE 动态切换	√	√	√	√	√
后续芯片支持特性	-	√	√	√	√
仿真器升级特性	-	√	√	√	√
芯片支持	8xC5x 以下	8xC5x 以下	8xC61 以下	8xC668 以下	8xC591 以下
出厂内置仿真芯片	P87C52X2	P87C52X2	P87C52X2	P87C52X2	P87C52X2

- **TKS-52B:** 经济型，仿真 8xC5x 以下，无跟踪功能，不可以升级。
- **TKS-58B:** 标准型，仿真 8xC5x 以下，有跟踪功能，可以升级。
- **TKS-61B:** 标准型，仿真 8xC61 以下，有跟踪功能，可以升级。
- **TKS-668B:** 高级型，仿真 8xC668 以下，有跟踪功能，可以升级。
- **TKS-591B:** 高级型，仿真 8xC591 以下，有跟踪功能，可以升级。

注：TKS-591B 除仿真 P8xC591 外，还兼容仿真所有的 Philips 的 44 脚以下的标准 51 芯片，它具备 TKS-668B 的所有性能。

四、TKS 仿真器 B 系列使用前应该了解的知识

TKS 仿真器 B 系列是 TKS 系列仿真器的一个系列品种，用户在使用前须仔细阅读《TKS 系列仿真器用户使用手册》的有关事项，在具备以下条件后可以开始熟悉使用 TKS 仿真器 B 系列。

- 熟悉 Windows 的系统操作以及一般应用软件的操作技能。了解 80C51 系列单片机的原理。
- 了解一般的电子电路知识。
- 准备一台符合要求的计算机和操作系统。
- 安装并熟悉 Keil 的 μ Vision2 或 μ Vision3 集成开发环境。

另外，在开始使用 TKS 仿真器 B 系列前，您首先应该熟悉一下 TKS 仿真器 B 系列的硬件。在您购买回来的仿真器套件中，有几样东西是必须使用的：

- **仿真器主机：**完成仿真功能的最主要部分。
- **仿真器电源：**为仿真器主机提供工作电源。它的输入电源为交流 220V。
- **仿真头：**带有插头，可以插入用户目标板的插座中。它的另外一端通过仿真电缆连接到仿真主机，这样把仿真主机和用户目标板连接起来。
- **仿真电缆：**用于连接仿真器主机和仿真头。随机提供的仿真电缆有两种长度规格。较长的一种便于用户连接用户目标板；当用于较高频率或工作于干扰较大的环境时可使用较短的仿真电缆。
- **RS-232 串口通讯电缆：**用于连接仿真器主机和您的计算机，长度为 1.5m 。

与 TKS 仿真器 B 系列配合使用的可选附件有：

- 44 脚插座仿真头，用于仿真 PLCC44 封装的器件。
- 40 脚 DIP 针状插座仿真头，用于仿真 DIP40 封装的器件。

五、TKS 仿真器 B 系列在 Keil 中的快速操作

Keil 软件是德国 Keil Software/Keil Elektronik 开发，近几年在国内得到迅速普及，国内使用的一般是比较稳定的 6.2 版本以及最新的 7.2 版本，用户可以向 Keil 公司的国内代理商广州周立功单片机发展有限公司购买正式软件或索取 Demo 版本，也可以直接到 Keil 公司的网站下载该软件。Keil 公司的网址是：<http://www.keil.com/>。

当您得到该软件的安装软件后，您可以按照提示进行安装。安装完毕后，您就可以在这套功能强大的开发平台上完成您所有的设计工作，包括源程序的编写、程序的编译和连接、软件模拟仿真的验证和排错，最后您可以使用 TKS 仿真器在 Keil 中进行硬件的仿真和调试，直到最终完成您的设计。

在安装完毕后，您就可以编写一个简单的程序来熟悉 Keil 的 IDE 环境，也可以使用 Keil 软件本身提供的范例程序，下面我们主要结合 Keil 的范例程序 Hello 来讲解 Keil 的具体使用。

当您安装完毕后，Hello 程序位于\Keil\C51\EXAMPLES，具体驱动器的位置取决与您的安装选择，如果您选择的是缺省安装，Keil 将会安装在 C 盘。

5.1 打开 Hello 工程

在 Keil 中文件的管理使用的工程的方法，而不是以前的单一文件的模式。您的所有的文件包括源程序（包括 C 程序、汇编程序）、头文件和一般的文本文件都将放在工程里统一管理。在使用 Keil 前，您应该习惯这种工程的管理方式。

点击 Keil 的图标启动 Keil 程序，然后您就可以看到 Keil 的主界面。在有的版本中 Hello 工程会自动打开。如果您没有看到 Hello 工程或看到的不是 Hello 工程，您可以点击 Project->Open Project 按照前面提示的路径找到该工程文件。

Hello 工程是一个简单的 C 语言单模块程序，完成的主要就是通过串口发送“Hello World”字符串。

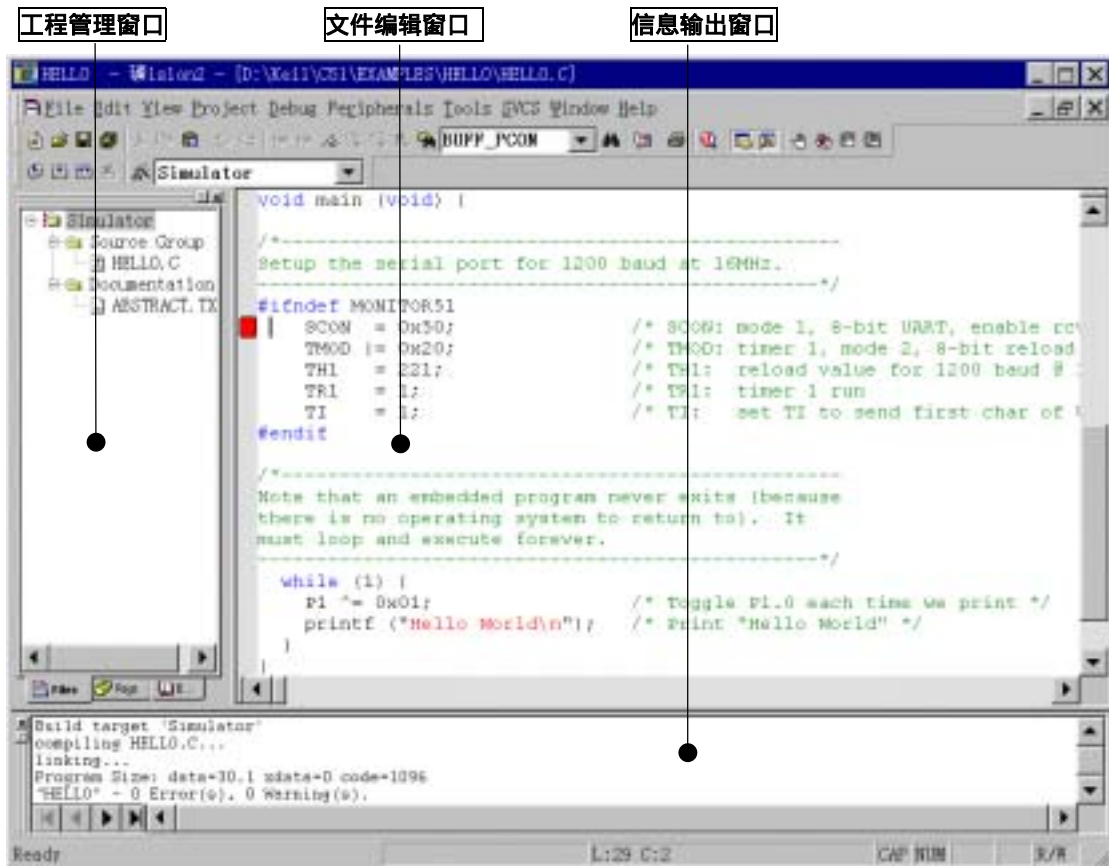


图 5.1 标准编译界面

5.2 为工程选择目标器件

在打开工程以后,还应该为您的工程选择合适的目标器件。选择方法为: 点击菜单 Project->Select Device for Target...后出现下面的对话框, 见图 5.2。

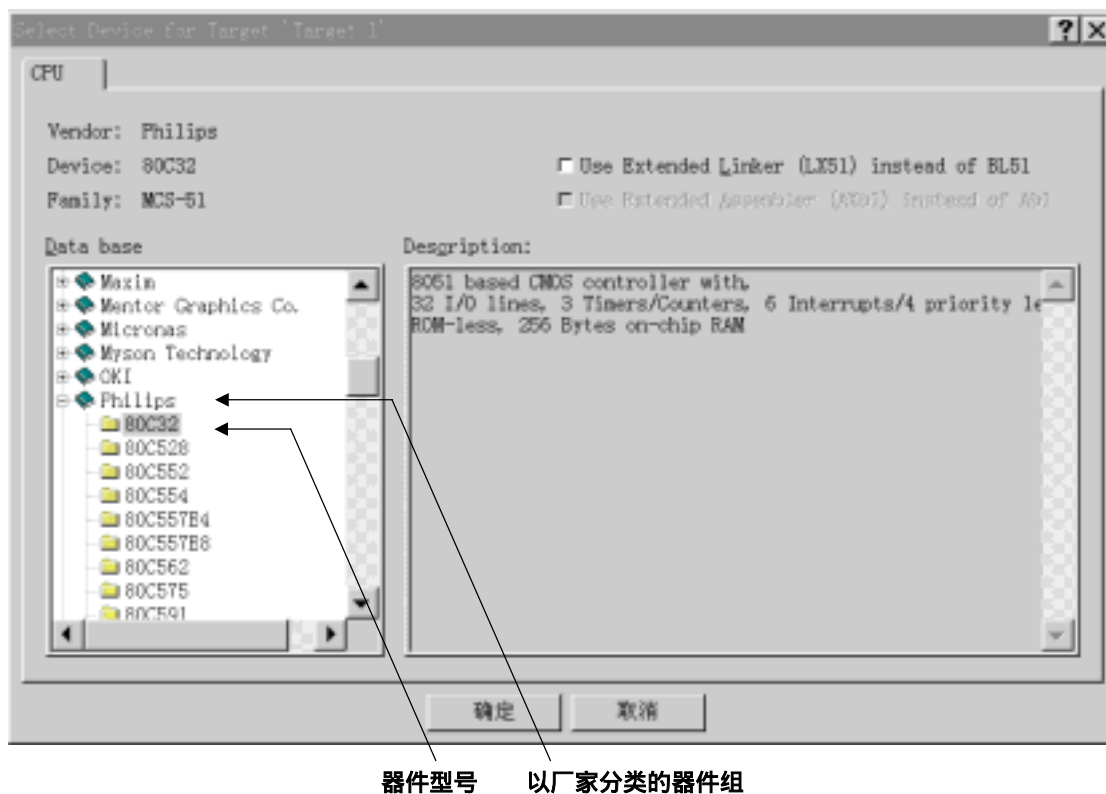


图 5.2 目标器件选择界面

在此对话框中,左边的 Data base 窗口中把 Keil 所支持的器件按照生产厂商进行分组列表。您可以点开 PHILIPS 组,找到并选择您想仿真的目标器件,例如图中选择了 PHILIPS 公司的 80C591,按动对话框上的确定按钮。选择正确的仿真器件非常重要,因为仿真时系统将根据所选定的器件为您提供一些特定的功能菜单。例如,如果您选择了 80C591,进入仿真时在 Peripherals 菜单中将会增加很多外设观察选择,例如 I2C Interface 浮动窗口、PWM unit 等。

5.3 编译和连接 Hello 工程

对于已经打开的 Hello 工程, 里面已经为您设置好了所有的系统配置, 因此您可以直接进行编译和连接。对于您自己新建的工程文件, 需要您自己加入文件和进行编译环境等设置, 有关 Keil 的设置在下面叙述。

选择菜单 Project->Rebuilt All Target Files, Keil 将对 Hello 工程进行连接和编译, 操作结果的信息显示在输出窗口。如果用户没有对 Hello 工程做任何的改动, 编译和连接将会正确完成。编译和连接的结果在信息输出窗口输出。



```
Output Window
Build target 'Simulator'
compiling HELLO.C...
linking...
Program Size: data=30.1 xdata=0 code=1096
"HELLO" - 0 Error(s), 0 Warning(s)
```

图 5.3 编译结果在信息输出窗口的输出

经过编译和连接后, Hello 工程已经可以进行模拟仿真或硬件仿真。模拟仿真是利用 PC 的资源模仿单片机的运行, 优点是简单, 不需要动用硬件, 缺点是对于外部信号无法进行正确的仿真; 硬件仿真正好相反, 但硬件仿真是调试的必须步骤, 下面我们主要讲述硬件仿真。

5.4 仿真 Hello 工程

Hello 经过编译和连接后，已经可以进行硬件仿真，也就是说可以通过 TKS 仿真器 B 系列进行实际的硬件仿真。在仿真前，还必须配置硬件仿真参数，否则可能引起仿真失败！

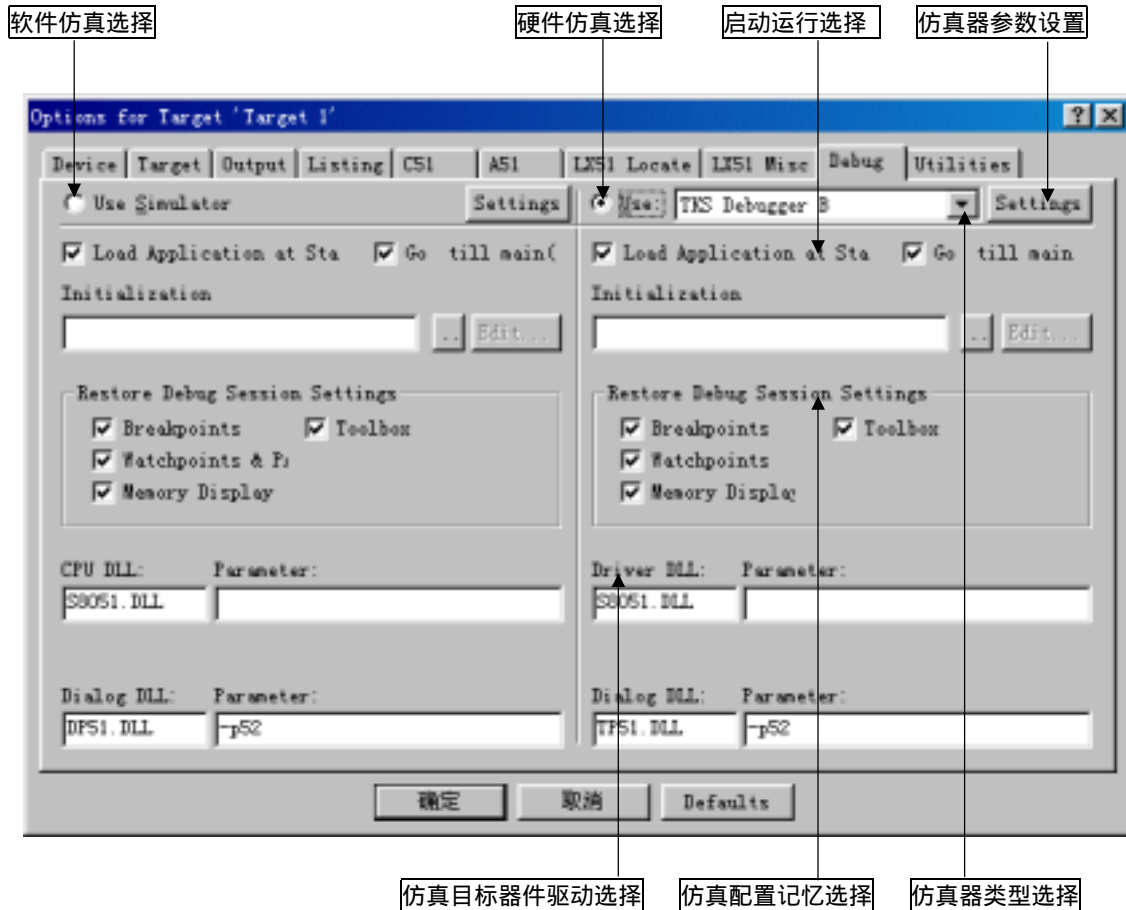


图 5.4 仿真配置界面

5.4.1 仿真器的设置

① 仿真器驱动程序声明:

在配置仿真器前，请将随机提供的 TKS 仿真器 B 系列驱动文件 TKS_DEB_B.dll 拷贝到 Keil 的安装目录中。例如，如果您的 Keil 的安装目录为 C:\Keil，则将 TKS_DEB_B.dll 拷贝到 C:\Keil\C51\bin 目录下。然后打开 C:\Keil 目录下的 Tools.ini 文件，在几个分类中找到[C51]类别，加入下列描述：

TDRV3=C:\Keil\C51\bin\TKS_DEB_B.dll ("TKS Debugger B") 其中：

- TDRV3 是驱动 DLL 的序号，如果您前面已经安装了多个驱动 DLL 以致占用了 TDRV3，则将 TKS 的驱动程序序号向后顺延，例如 TDRV5。在随机软件中，有一个示范 Tools.ini 文件，可以用来参考如何加入驱动程序。

- C:\Keil\C51\bin\TKS_DEB_B.dll 是驱动程序放置的路径，该路径下必须有 TKS_DEB_B.dll 文件。

注意：该驱动程序说明行所指出的路径可能和您的不同，这要根据您的 Keil 软件在硬盘中的安装位置。

② 仿真环境的设置:

击菜单 Project->Option for Target..进入工程配置对话框, 选择 Debug, 出现下面的对话框。在这个对话框中, 主要供用户选择仿真时需要的一些设置。在这些设置中, 请用户仔细按照下面的要求选择, 不正确的选择将回造成仿真的困难和不准确, 甚至是难以理解的现象!

- **软件仿真选择:** 使用软件仿真当前的设计。
- **硬件仿真选择:** 使用硬件仿真当前的设计。使用 TKS 仿真器 B 系列必须选择硬件仿真。
- **仿真器驱动程序选择:** 用于选择当前的硬件仿真驱动。如果用户在 C:\KeilTools.ini 文件中正确填写了 TKS 的驱动说明, 而且该说明行所指向的路径也有对应的 DLL 文件, 则用户可以选中 TKS 驱动项目。
- **启动运行选择**
Load Application at Start: 进入仿真后调入程序代码进入仿真器。
Go till main: 在使用 C 语言设计后直接运行到 main () 函数。
- **仿真配置记忆选择**
Breakpoints: 记忆当前设置的断点, 下次进入仿真后该断点设置存在并有效。
Watchpoints: 记忆当前设置的观察项目, 下次进入仿真仍有效。
Memory Display: 记忆当前存储器区域的显示, 下次进入仿真仍有效。
Toolbar: 记忆当前的工具条设置, 下次进入后仍有效,
- **仿真目标器件驱动程序选择**
如果用户在目标器件选择中选择了相应的器件, Keil 将自动选择相应的仿真目标器件驱动程序。在图 5.4 中, Keil 选择了标准 S80C51.DLL 驱动文件, Dialog.DLL 选择 p591。Keil 根据不同的器件选择不同的仿真驱动 DLL, 这样在仿真时就会有该器件相应的外设菜单。用户进入仿真(硬件仿真和软件仿真)后, 在 Peripherals 菜单中会添加该器件的外设观察菜单, 用户点击后会出现浮动的观察窗口, 方便用户观察和修改。
- **硬件仿真参数设置:** 该配置项主要设置硬件仿真的重要选项, 用户要仔细选择。点击 Settings 进入对话框, 如图 5.5 所示。

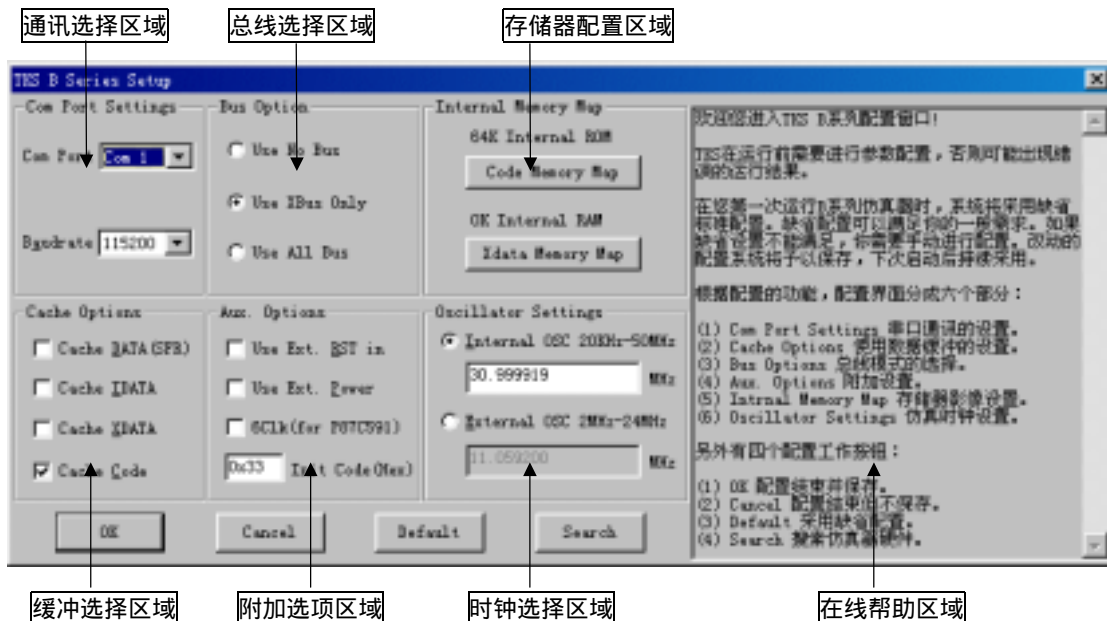


图 5.5 仿真硬件配置界面

Com Port Settings 串口设置区域

- **Port:** 选择硬件仿真器使用的串口。TKS 使用串口同 Keil 进行通讯。
- **Baudrate:** 串口使用的波特率。TKS 使用自动适应波特率，也就是该选项在一定选择范围内 TKS 都能正常工作。选择快的波特率能加快运行速度，但是低的波特率在环境恶劣的条件下更可靠。

TKS 仿真器 B 系列适应的波特率范围在 1200-115200 之间，但根据用户的实际情况略有差异。

注意：选定波特率后，必须先接通 TKS 系列仿真器电源，仿真器主机面板上的指示灯闪烁结束后，才可以运行 Keil 的硬件仿真程序。在硬件仿真环境中用户可以在 Peripherals 菜单中打开 Target Settings 窗口（图 5.5）来更改波特率，更改后整个系统重新建立连接；也可以更改串口，但是用户必须在更改后的串口上连接仿真器。

Catch Option 选择区域

使用存储器缓冲区域，这样在一般的操作中仿真软件不用频繁的读取仿真器中的内容，而使用缓冲区域。使用缓冲可以加快仿真速度。使用缓冲的缺点是屏幕的数据刷新慢，但是在单步或运行后所有在屏幕上显示的信息将全部刷新一次。

Select Oscillator 选择区域:

- **Internal OSC 20KHz-50MHz:** 选择仿真器内部提供的精密线性时钟

B 系列内部设有一个精密时钟发生器，可以连续可调产生 20KHz-50MHz 的稳定时钟，精度为 0.001。用户只需要在频率输入框中输入想要的频率数值（以 MHz 为单位）即可。需要注意的是，用户输入的时钟数值并不能 100% 准确地产生，一般会有一定的误差，在指定的 20KHz-50MHz 范围内误差不超过 0.001。用户输入频率数值后，可以用鼠标点击窗口其它部分，频率输入框的频率会自动计算并显示出最终产生的频率数值。

- **External OSC 2MHz-24MHz:** 选择仿真器外部时钟

选择外部时钟用户可以根据自己的要求来选择时钟频率，只需要在仿真头上插入希望的晶振或使用目标系统板上的晶振。但是由于时钟信号从外部引入到仿真器内部，中间要经过仿真电缆和接插件，因此稳定性和抗干扰性能不如使用内部时钟。

用户需要在外部时钟频率输入框输入实际的频率，这个频率将作为精密时间显示的基准。

Aux. 选择区域

- **Use Ext. RST in** 使用外部复位输入选择

如果用户选择该项，则仿真运行动作（单步、全速或全速断点）会根据外部复位信号的电平来动作。如果在开始运行前发现有效的外部复位电平，仿真器将提示用户，并停止运行，等待用户解除外部复位；如果在运行当中发现有外部复位信号，系统也将提示用户，并复位系统，复位信号解除后重新运行。

- **Use Ext. Power 使用外部电源选择**

使用外部电源作为仿真电源。在不使用外部电源时，使用的是内部的+5V 电源，要求用户板上的电源也必须是+5V，但是允许有 0.7V 的上下偏差。如果用户需要仿真非+5V 供电的系统，则可以选择该项。使用的外部电源电压必须在 2.0V-5.5V 之间。外部输入电源从用户板得到，经过仿真头电源脚进入仿真器内部。外部输入电源最低电压为 2.0V，小于 2.0 伏内部电压将维持在 2.0 伏不变；外部输入电压最大为 5.5V，超过 5.5V 将引起内部保护装置的启动。需要注意的是，如果仿真器内部的仿真芯片不支持低电压运行，请不要选择该项。TKS 系列仿真器内部有检查装置，如果仿真器件不支持低电压运行，将不会使用外部电源。

- **6Clk (for P87C591) 选择时钟模式**

为 P87C591 选择时钟模式。在使用 TKS-591B 仿真器且内部的仿真芯片为 P87C591 时，如果选中，则内部的 591 仿真芯片工作于 6Clk 模式下；反之则工作于 12Clk 模式下。注意：对于除 591 以外的芯片该选择项无任何作用。

- **InitCode (Hex) 设置仿真器初始化程序代码**

选择仿真器的初始化代码。仿真器复位后（包括上电），将内部程序空间按照该区域设定的数值进行初始化。这样在用户调入用户代码后，用户没有使用的区域仍保持初始化的数值。当用户在程序中的没使用部分设置软件陷阱时，这个功能非常有用。用户可以直接在该栏文本区域中填入数值，注意必须为十六进制 Hex 数值。

Bus Option 总线选择区域

为了取得更好的仿真效果，B 系列仿真器将总线分成 3 种模式，用户可以根据实际需要加以选择，选择不同的总线模式只影响 P0/P2 口的状态。

- **Use No Bus 不使用任何总线**

在仿真时，对于操作总线的情况都加以制止，主要针对 P0/P2 口。P0/P2 口在这种情况下是标准的 I/O 模式，遇到 MOVX/MOVC 指令时 P0/P2 仍然表现出 I/O 特性而不输出总线信号。如果用户没有使用总线则选择这种模式较好。另外，如果用户仅使用了 MOVX 指令来操作一些单片机的内部 xdata 空间，则也可以选择这种模式。

- **Use XBus Only 仅使用数据总线**

仿真时，P0/P2 口将根据指令的运行情况来决定工作模式。遇到 MOVX 指令时 P0/P2 口将输出总线地址/数据，MOVC 指令结束后将恢复 I/O 模式。一个特殊的情况是，如果用户使用 MOVX @Ri 指令，P2 口仍将表现出 I/O 特性，这种情况下 P2 口仍然可以作为 I/O 口使用，B 系列支持这种功能。

- **Use All Bus 使用数据总线和程序总线**

仿真时，P0/P2 口完全作为总线使用，用户可以通过 MOVX 指令操作仿真器外部数据空间，也可

以通过 MOVC 指令读取外部代码数据，甚至可以运行外部的程序。在这种模式下，P0/P2 不能再作为 I/O 使用。

注意：总线模式的选择需要同 Internal Memory Map 配合使用。下列的情况必须注意，当需要运行仿真器外部程序时，一定要选择 Use All Bus。

Internal Memory Map 内部存储器影像设置

TKS 内部有 64K 的代码空间/64K 的数据空间供用户使用，用户也可以使用全部/部分外部的程序空间/数据空间，影像的分辨率为 1byte，用户可以把任一地址的程序/数据分配到仿真器内部或外部。

● Code Memory Map 代码空间影像

允许用户设置内部代码空间的影像，Code Memory Map 窗口中已经为用户设置了几个标准配置，分别是仿真器内部 64K/32K/16K/8K/4K/0K，则仿真器外部代码空间则为 0K/32K/48K/56K/60K/64K。如果这些标准设置不能满足你的要求，用户可以添加自己的影像范围。需要注意的是添加的影像是内部代码空间，没有涉及到的空间地址则影像到仿真器外部。

按钮 **Code Memory Map** 上方的文字表明当前代码分配的情况。例如，**64K Internal ROM** 表明 ROM 状态为仿真器内部 64K，外部 ROM 空间为 64K-64K=0K。如果用户需要改变当前的 ROM 分配情况，可以点击 **Code Memory Map** 按钮，进入 ROM 空间分配界面：

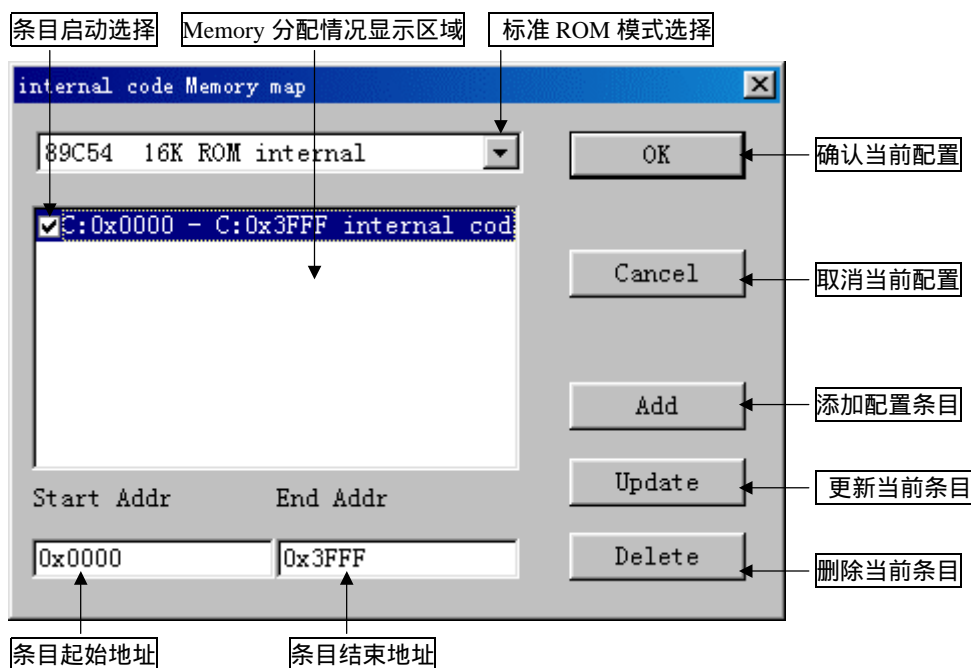


图 5.6 仿真器内部代码空间分配界面

Memory 分配情况显示区域： 该区域显示当前用户的内部 ROM 空间的分配情况。

标准 ROM 模式选择： 供用户选择几种标准的 ROM 模式。

条目启动选择： 启动或关闭该条目。

条目起始地址： 用户输入增加条目的起始地址。

条目结束地址:	用户输入增加条目的结束地址。
添加配置条目:	增加一个 Memory 分配条目。
更新当前条目:	更新当前选中的 Memory 分配条目。
删除当前条目:	删除当前选中的 Memory 分配条目。
确认当前配置:	确认当前所有的 Memory 分配条目。
取消当前配置:	取消当前所做的 Memory 分配改动。

标准 ROM 模式的选择

如果用户选择的模式属于以下的标准模式，可以直接使用**标准 ROM 模式选择**下拉列表选择，用户可以选择的 ROM 模式为：

- 89C516 仿真器内部 ROM 大小为 64K，仿真器外部 ROM 大小为 0K。
- 89C58 仿真器内部 ROM 大小为 32K，仿真器外部 ROM 大小为 32K。
- 89C54 仿真器内部 ROM 大小为 16K，仿真器外部 ROM 大小为 48K。
- 89C52 仿真器内部 ROM 大小为 8K，仿真器外部 ROM 大小为 56K。
- 89C51 仿真器内部 ROM 大小为 4K，仿真器外部 ROM 大小为 60K。
- 89C31 仿真器内部 ROM 大小为 0K，仿真器外部 ROM 大小为 64K。

用户选择相应的模式后，在 **Memory 分配情况显示区域** 将显示该模式的地址分配情况。

用户添加 Memory 分配条目

在用户添加 Memory 分配条目前，需要在**条目起始地址**和**条目结束地址**中填入正确的起始地址和结束地址，地址的表示方法是十六进制，要求结束地址必须大于或等于起始地址；另外，地址范围不能同已经存在的地址范围有重叠。点击 **Add** 按钮后，该分配条目在 **Memory 分配情况显示区域**中显示出来。

用户删除 Memory 分配条目

用户可以在 **Memory 分配情况显示区域**中选中一个 Memory 分配条目，点击 **Delete**。

用户更新 Memory 分配条目

如果用户需要更改已经存在的 Memory 分配条目，可以在 **Memory 分配情况显示区域** 双击该条目，然后在**条目起始地址**和**条目结束地址**中重新写入希望的地址，然后点击按钮 **Update** 进行更新。

● Xdata Memory Map 数据空间影像

允许用户设置内部数据空间的影像，Xdata Memory Map 窗口中已经为用户设置了几个标准配置，分别是仿真器内部 64K/32K/16K/8K/4K/0K，则仿真器外部数据空间则为 0K/32K/48K/56K/60K/64K。如果这些标准设置不能满足你的要求，用户可以添加自己的影像范围。需要注意的是添加的影像是仿真器内部数据空间，没有涉及到的空间地址则影像到仿真器外部。

按钮 **Xdata Memory Map** 上方的文字表明当前 Xdata 分配的情况。例如，**64K Internal RAM** 表明 RAM 状态为仿真器内部 64K，外部 RAM 空间为 64K-64K=0K。如果用户需要改变当前的 RAM 分配情况，可以点击 **Xdata Memory Map** 按钮，进入 RAM 空间分配界面：

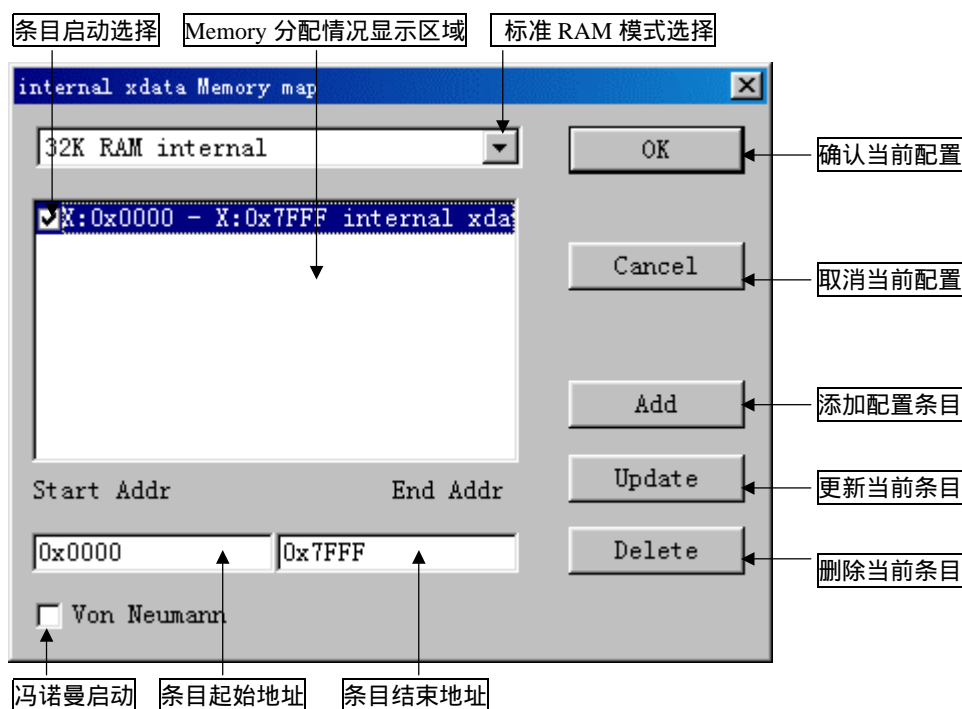


图 5.7 仿真器内部数据空间分配界面

Memory 分配情况显示区域:	该区域显示当前用户的内部 RAM 空间的分配情况。
标准 RAM 模式选择:	供用户选择几种标准的 RAM 模式。
条目启动选择:	启动或关闭该条目。
条目起始地址:	用户输入增加条目的起始地址。
条目结束地址:	用户输入增加条目的结束地址。
添加配置条目:	增加一个 Memory 分配条目。
更新当前条目:	更新当前选中的 Memory 分配条目。
删除当前条目:	删除当前选中的 Memory 分配条目。
确认当前配置:	确认当前所有的 Memory 分配条目。
取消当前配置:	取消当前所做的 Memory 分配改动。
冯诺曼结构启动:	启动冯诺曼结构。

标准 RAM 模式的选择

如果用户选择的模式属于以下的标准模式，可以直接使用**标准 RAM 模式选择**下拉列表选择，用户可以选择的 RAM 模式为：

- 64K 模式 仿真器内部 RAM 大小为 64K，仿真器外部 RAM 大小为 0K。
- 32K 模式 仿真器内部 RAM 大小为 32K，仿真器外部 RAM 大小为 32K。
- 16K 模式 仿真器内部 RAM 大小为 16K，仿真器外部 RAM 大小为 48K。
- 8K 模式 仿真器内部 RAM 大小为 8K，仿真器外部 RAM 大小为 56K。
- 4K 模式 仿真器内部 RAM 大小为 4K，仿真器外部 RAM 大小为 60K。
- 0K 模式 仿真器内部 RAM 大小为 0K，仿真器外部 RAM 大小为 64K。

用户选择相应的模式后，在 **Memory 分配情况显示区域** 将显示该模式的地址分配情况。

用户添加 Memory 分配条目

在用户添加 Memory 分配条目前，需要在**条目起始地址**和**条目结束地址**中填入正确的起始地址和结束地址，地址的表示方法是十六进制，要求结束地址必须大于或等于起始地址；另外，地址范围不能同已经存在的地址范围有重叠。点击 Add 按钮后，该分配条目将在**Memory 分配情况显示区域**中显示出来。

用户删除 Memory 分配条目

用户可以在**Memory 分配情况显示区域**中选中一个 Memory 分配条目，点击 Delete。

用户更新 Memory 分配条目

如果用户需要更改已经存在的 Memory 分配条目，可以在**Memory 分配情况显示区域**双击该条目，然后在**条目起始地址**和**条目结束地址**中重新写入希望的地址，然后点击按钮 Update 进行更新。

注：

- (1) 用户必须注意：如果某一个地址的 xdata 影像到仿真器内部，则该地址外部的 xdata 则被忽略。如果用户外部有一些 xdata 的外设，不要将该外设地址映射到仿真器内部。
- (2) 内部数据空间同外部数据空间在使用上使完全相同的。但是如果在 Use No Bus 下使用内部数据空间，由于 P0/P2 口不输出地址/数据，因此内部的 64K 数据空间类似于片上 xdata，但是 wr/rd 信号线将正常输出信号。
- (3) TKS 仿真器内部的 64K 数据空间支持冯·诺曼结构，该结构既可以当作程序运行又可以当作数据空间进行操作。在添加数据影像时，选中 Von Neumann 将使添加的项目具有冯·诺曼特性。

5.4.2 使用 TKS 仿真器 B 系列进行硬件仿真

5.4.2.1 硬件仿真的基本步骤

- (1) 用随机提供的串口电缆连接 TKS 仿真器 B 系列和 PC 机。注意选择的串口必须同 Keil 硬件仿真设置中的选择一致。
- (2) 将随机提供的电源适配器的交流端子插入 220V 交流电源，另一端接入 TKS 仿真器 B 系列的电源插座。这时，TKS 仿真器 B 系列的指示灯将闪烁，闪烁停止后仿真器进入命令等待。
- (3) 点击菜单 Debug->Start/Stop Debug Session，Keil 将进入硬件仿真状态。
- (4) 再次点击菜单 Debug->Start/Stop Debug Session，Keil 将退出硬件仿真状态。

关于更为详细的资料，点击图 5.6 中的**帮助文件启动**，参考 Keil 的随机文档。根据经验，用户在使用 Keil 的仿真环境时，最为困难的是具体的操作问题，下面就用户提问最多的问题加以描述：

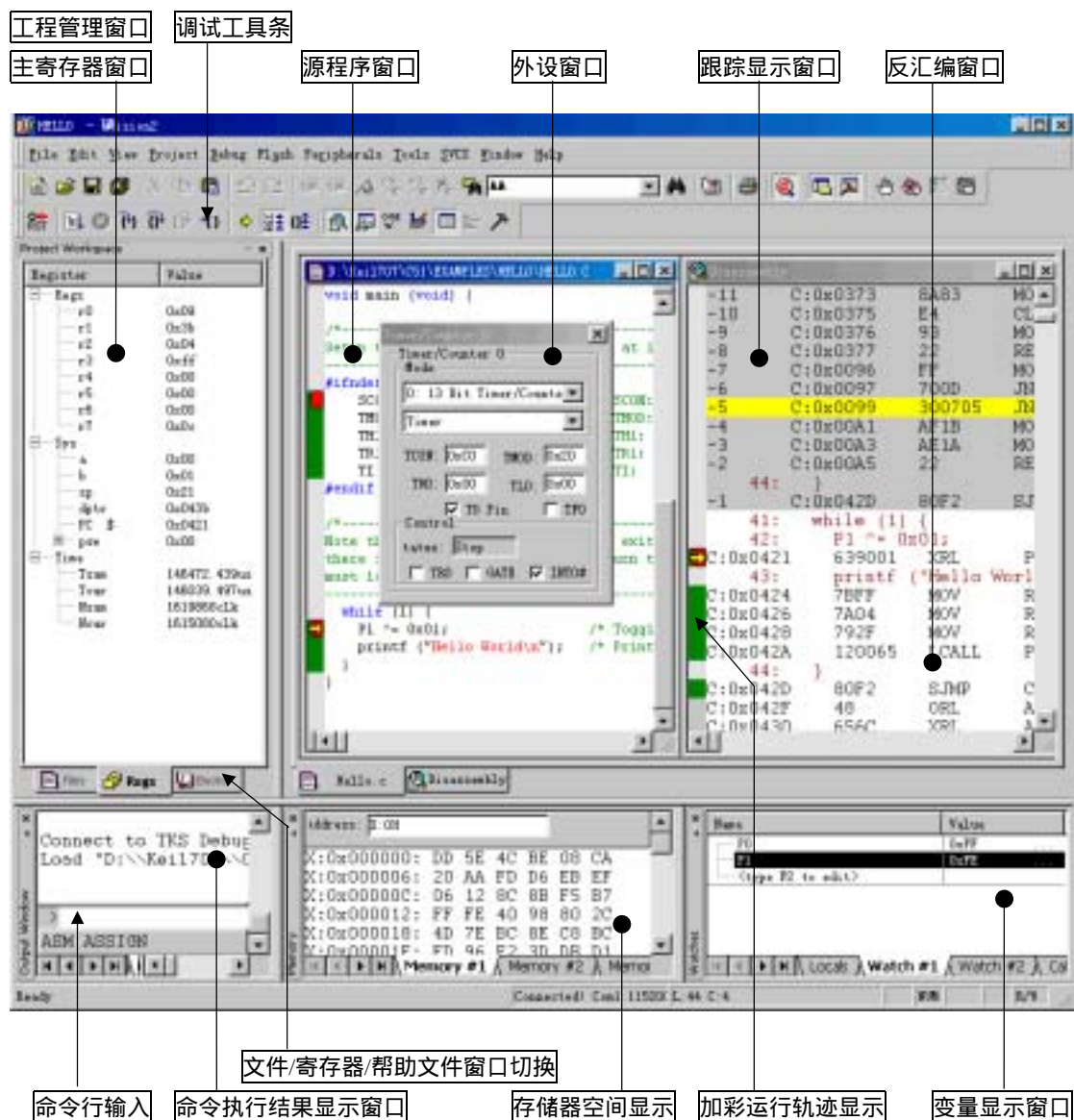


图 5.8 标准硬件仿真界面

5.4.2.2 硬件仿真的基本操作

工程管理窗口

用于工程的管理，包含文件窗口、寄存器窗口、帮助文件窗口三部分。用户可以使用文件/寄存器/帮助文件窗口切换标签进行切换。

- **文件窗口：** 显示当前工程的组织结构，用户可以添加/删除文件和文件夹。
- **主寄存器窗口：** 显示/修改主寄存器内容，该窗口只在进入仿真后有效。
- **帮助文件窗口：** 显示当前 Vu2 所有的帮助文件，用户可以点击某文件进行阅读。

主寄存器窗口

用于显示和修改主寄存器得内容，B 系列窗口中有精密时间显示。

- **查看和修改主寄存器的内容**

仿真时主寄存器的内容在主寄存器窗口显示，用户除了可以观察以外，还可以修改。用鼠标点击选中一个单元，例如 SP，然后再单击 SP 的数值位置，出现文字框后输入希望数值按回车即可。另外的输入方法是使用命令行窗口。例如，输入 A=0X34，将把 A 的数值设置为 0X34。

- **观察和修改变量**

点击 View->Watch & Call stack Window，该窗口出现后，选择 Watch1-3 中的任一个窗口，按动 F2 后在 Name 栏填入变量的名称，如 DPTR、TL0 等；也可以输入用户自己定义的变量名称，如 Temp1、Counter 等，但必须是存在的变量。如果想修改数值，可单击 Value 栏，出现文本框后输入希望的数值。

- **运行时间的使用**

B 系列新增了一个极为先进的运行时间的显示功能。显示的内容分为总运行时间/当前操作的运行时间/总运行时钟数/当前操作的运行时钟数。

总运行时间 Tsum

为仿真系统从上电或复位后到当前状态经历的有效运行总时间。在监控状态并不会使该时间增大，只有有效的运行操作例如单步/全速运行才能使之增加。

当前运行时间 Tcur

记录了前一次有效运行操作经历的时间。例如，运行一个单步经历了 1us 则该时间显示为 1us；再运行一个单步经历了 2us，则该时间显示为 2us。与**总运行时间**的不断累加不同，该时间是一个时间差，便于用户观察本次操作经历的时间。

总运行时钟数 Nsum

为仿真系统从上电或复位后到当前状态经历的振荡周期数。在监控状态并不会使该时间增大，只有有效的运行操作例如单步/全速运行才能使之增加。

当前运行时钟数 Ncur

记录了前一次有效运行操作经历的振荡周期数。例如，运行一个单步经历了 12Clk 则显示为 12；再运行一个单步经历了 12Clk，则该时间显示为 12Clk。与**Nsum**的不断累加不同，该时间是一个差值，便于用户观察本次操作经历的时钟数。

注意事项：

- **Tsum/Tcur** 显示的数值跟用户选择的时钟有关系。在 TKS 仿真器 B 系列中，用户可以有 4 种时钟选择，见图 5.5 仿真硬件配置界面。
- 如果用户选择仿真器内部提供的时钟，仿真器能完全准确的显示 **Tsum/Tcur**。
- 如果用户选择外部时钟（External OSC），仿真器需要知道用户从外部输入时钟的频率。如果用户填写的时钟频率不准确，**Tsum/Tcur 显示的数值将是错误的！**用户在选择 External OSC 后，需要在晶振频率设置栏中填入频率数值，注意是以 MHz 为单位的。

Nsum/Ncur 显示的是运行中经过的时钟周期数，显示的数值是独立的，即使在选择外部时钟时不依赖与用户设置的时钟频率！**Ncur** 在观察当前的时钟模式非常有用。例如，在

仿真 87C52X2 芯片时，该芯片可以动态的改写当前的时钟模式 6/12Cik。如果当前的时钟模式为 12Cik，则执行一个 NOP 操作后 Ncur=12；如果当前的时钟模式为 6Cik，则执行一个 NOP 操作后 Ncur=6。这种功能在使用有双时钟模式的仿真芯片如：P89C668/P87C591/P89CRD2/P89C6X 等，可以非常方便的看出当前的时钟模式。

- 复位后 Tsum/Tcur/Nsum/Ncur 的数值全部为 0，且不能改动。

源程序窗口 用户编写并加入工程的 c/asm 程序代码。

● 源程序窗口的开启

源程序必须是用户编写的 C 语言或汇编语言程序，并已经加入了当前的工程中。如果某一个源文件只是在 uV2 中使用主菜单 File->Open 打开，但是没有加入工程中，则文件在 uV2 只是可以编辑，不能进行调试，因此不是我们这里所说的源文件。

如果用户需要的源文件没有显示，用户可以在主菜单 Window 中查看该文件是否存在，如果存在则点击该文件，使之成为当前窗口；如果该文件没有存在，用户可以在 Project WorkSpace 窗口中，选择 File 窗口，查看文件结构并点击打开需要的源文件。

● 源程序窗口的操作

在源程序窗口中点击鼠标右键，可以看到多种针对源程序的操作，如图 5.9 所示。

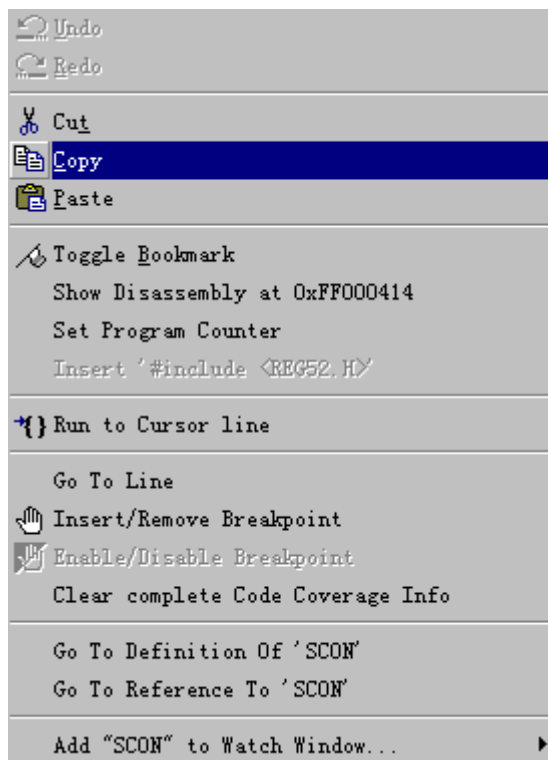


图 5.9 源程序窗口的操作菜单

Undo: 取消操作。
Redo: 重复操作。

Cut:	剪切。
Copy:	拷贝。
Paste:	粘贴。
Toggle Bookmark:	在当前行插入/取消书签。
Set Show Disassembly at:	在汇编窗口显示汇编内容。
Program Counter:	设置当前 PC 指针。
Insert/Remove BreakPoint:	插入/取消断点。
Enable/Disable Breakpoint:	启动/关闭断点。
Clear complete Code Coverage Info:	清除全部覆盖信息。
Goto Definition of :	跳转到定义位置。
Goto Reference of:	跳转到引用位置。
Add “ ” to Watch Windows:	将变量加到 观察窗口 。

反汇编窗口 以反汇编的形式显示当前的用户程序。

- **反汇编窗口的开启**

如果用户的源程序是用 C 语言写成的，进入调试环境中可能会不出现反汇编窗口。用户可以从 View->Disassembly Window 打开反汇编窗口。在反汇编窗口中点击鼠标右键，可以看到多种针对反汇编的操作，包括反汇编显示模式、在线汇编和程序代码的调入。

- **反汇编窗口的操作**

在反汇编窗口中，点击鼠标的右键，将弹出操作菜单，用户可以选择需要的操作。

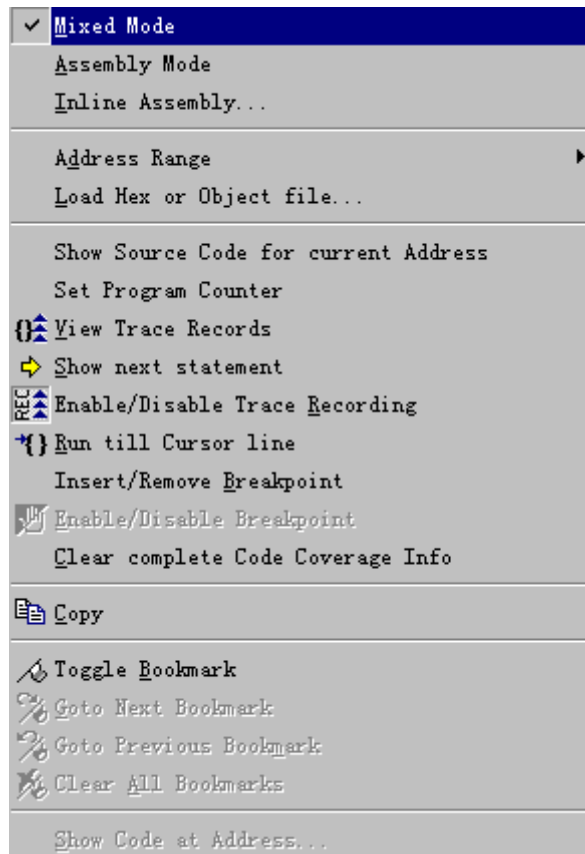


图 5.9 反汇编窗口的操作菜单

Mixed Mode:	混合反汇编模式，插入源程序语句。
Assembly Mode:	反汇编模式，不源程序语句。
Inline Assembly:	对当前行的反汇编语句进行在线修改。
Address Range:	调整当前窗口的地址范围（不支持）。
Load Hex or Object file:	下载用户程序代码（Hex 或 Object）。
Show source Code for current Addrss:	显示该汇编行在源程序窗口对应的代码。
Set Program Counter:	将当前的 PC 指针设置到当前的汇编行。
View Trace Records:	显示跟踪汇编行。
Show next statement:	显示当前 PC 指针位置。
Enable/Disable Trace Recording:	开启/关闭跟踪记录。
Run till Cursor line:	全速运行到当前汇编行。
Insert/Remove Breakpoint:	插入/取消断点。
Enable/Disable Breakpoint:	启动/关闭一个断点
Clear complete Code Coverage Info:	清除全部的代码覆盖信息。
Copy:	拷贝当前行的反汇编内容。
Toggle Bookmark:	在当前行插入/取消书签
Goto Next Bookmark:	跳转到下一个书签位置。
Goto Previous Bookmark:	跳转到上一个书签位置。
Clear All Bookmark:	清除全部书签。

● Trace 窗口的操作

B 系列有 64K 的 Trace 缓冲区，可以记录 64K 条指令的运行。在任意时刻进入监控后，Trace 缓冲区中记录了前 64K 条指令的运行轨迹情况；如果当前程序实际运行没有超过 64K 条，则按照实际记录的有效条数计算。64K 的 Trace 记录在复位后将全部清除。

由于 Trace 只能以汇编指令记录，所以只能以汇编的形式显示。在 uV2 中，跟踪窗口在反汇编窗口显示。使用主菜单 **Debug->Enable/Disable Trace Recording** 开启/关闭跟踪记录，然后用主菜单 **Debug->View Trace Records** 观察存在的跟踪记录。

uV2 中 Trace 在汇编窗口中使用的是**覆盖嵌接技术**，未运行的代码和 Trace 记录的已运行的代码总是相互紧密连接在一起，用户非常清楚的看出程序的运行流向。但是由于程序存在跳转和源程序位置的变动，这种覆盖嵌接有时候会发生错位或引起 Trace 窗口关闭，需要用户再次点击 **Debug->View Trace Records** 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 PageUp/PageDown 键变换显示内容，也可以用鼠标点击反汇编窗口右上角/右下角的上移/下移箭头。反汇编窗口右边的滑动块不能用来移动 Trace 显示内容。

调试工具条 用于调试中的快速操作。

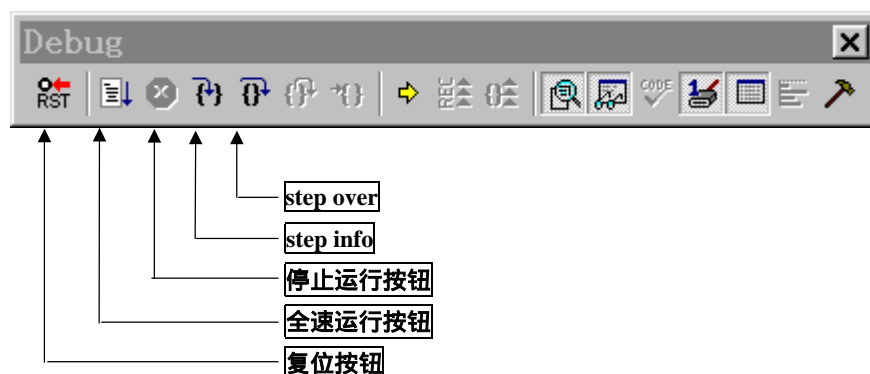


图 5.10 调试工具条

● 运行程序

在 Keil 的仿真环境中，有几种运行方式：

1. Run 全速运行，遇到断点停下或用户按动 Stop 按钮停止。
2. Step info 单步跟踪运行一条指令。如果该语句为 C 中的调用子程序语句或汇编中的 CALL 指令，Step info 指令将跟踪进入子程序内部。
3. Step over 单步运行完一条指令。如果该语句为 C 中的调用子程序语句或汇编中的 CALL 指令，Step over 指令将全速完成该子程序的运行，停在下一指令处。
4. Run till Cursor Line 从当前位置运行到光标处。
5. 连续单步运行。

TKS 仿真器 B 系列有连续单步运行的功能，每一步运行完后将刷新屏幕，然后执行下一步运行。将当前屏幕转到反汇编串口，在命令窗口输入例如 T 1000 的命令（1000 表示运行 1000 个汇编单步），然后就可以看到程序快速的执行单步操作，PC 指针不断的移动，屏幕上的数据也会发生变化。**注意：**C 语言的源程序窗口下不支持连续单步运行方式。

另外，用户还必须注意当前的运行背景，即是在源程序背景下还是在反汇编背景下，源程序分汇编程序和 C 语言程序。在不同的背景下，使用上述的运行方式会有所不同。在汇编源程序和反汇编背景下，Step Info 每次只执行一条汇编指令，遇到 CALL 指令会跟踪到调用程序内部；而 Step over 是在执行 CALL 指令后完成执行被调用的指令，停在 CALL 的下一条指令上。在 C 语言源程序背景下，Step Info 是执行一条 C 语言程序，这条 C 语言程序可能包含多条汇编程序，但是不包含调用函数。如果该语句中包含调用函数，经过 Step Info 运行后将跟踪到调用函数的内部。Step over 则是执行完被调用的函数，但是简单语句的作用同 Step Info 一样。

● 复位按钮

在 Keil 的仿真环境中，工具条中有 Reset 按钮，表示系统的复位。

● 退出仿真状态

在仿真状态中，用户不可以再对工程进行编译和连接。退出仿真的方法同进入一样，点击菜单 Debug->Start/Stop Debug Session，Keil 将退出硬件仿真状态。**建议用户在 Keil 退出仿真环境后，然后关闭 TKS 仿真器！**

断点的设置

B 系列仿真器有 4×64K 断点，分别是程序运行断点/MOVC 读取断点/MOVX 读取断点/MOVX 写入断点等四种。这四种断点用户可以单独分别使用，也可以混合使用。

● 断点种类

- 程序运行断点：** 程序运行到该指令位置时停止，进入仿真监控状态。
- MOVC 读取断点：** 当 MOVC 指令读取该地址时，程序在完成操作后停止进入仿真监控状态。
- MOVX 读取断点：** 当 MOVX 指令读取该地址时，程序在完成操作后停止进入仿真监控状态。
- MOVX 写入断点：** 当 MOVX 指令写入该地址时，程序在完成操作后停止进入仿真监控状态。

● 断点定义

程序运行断点的定义

程序运行断点有非常简单的定义方法。用户可以在源程序窗口（包括 C 语言和汇编）和反汇编窗口中，双击想要设置断点的程序行，则在窗口左边的状态条中出现红色的断点标志块。再次双击程序行该程序行，则为取消断点，取消断点后窗口左边的红色的断点标志块消失。

在源程序窗口中，按照这种方法有的时候不能在某些程序行中设置断点。出现这种情况一般是该源程序行无对应的程序代码，原因是该程序没有被编译或者由于语言优化而没有产生最终的程序代码。在这种情况下不可能在该源程序行设置断点。

用户也可以使用断点菜单和断点工具条。在源程序窗口（包括 C 语言和汇编）和反汇编窗口中，用户用鼠标或键盘上的上移键或下移键移动光标到想要的程序行，点击主菜单 **Debug->Insert/Remove Breakpoint** 来插入/删除一个断点。

● 程序运行断点的关闭/启动（Enable/Disable）

断点的操作有插入断点/关闭断点/取消断点三种，其中关闭断点和取消断点是不同的。取消断点是把该断点取消，当需要时需要再次设置；关闭断点是对已经插入的断点暂时关闭，可以在以后再次简单的启动。对于已经定义的断点，用户可以点击主菜单 **Debug->Enable/Disable Breakpoint** 来启动/关闭一个断点。

● 用断点管理器管理断点

由于**程序运行断点**直接和程序相对应，用户可以使用简单的方法来设置。对于 B 系列中的其它三种断点，用户需要启动**断点管理器**来进行断点的处理。点击主菜单 **Debug->Breakpoints** 打开断点管理器，如图 5.11 所示。

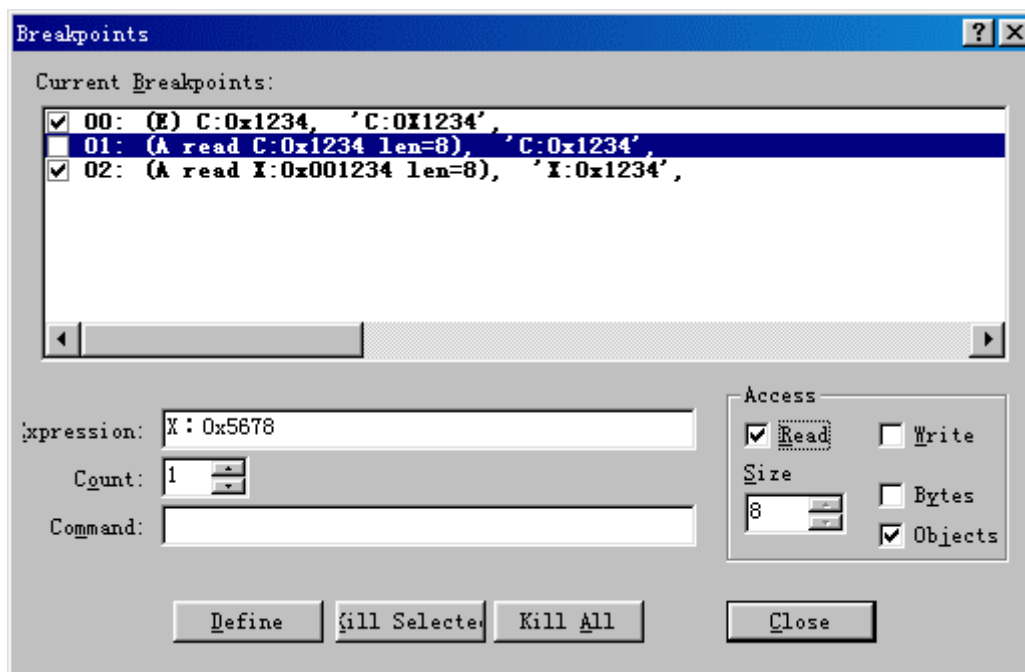


图 5.11 断点管理窗口

断点管理器主要有下面的功能部件：

- Current Breakpoints:** 当前所有的断点显示窗口。
- Expression:** 断点的表达式。
- Count:** 触发断点的次数（B 系列不支持）。
- Command:** 断点命令输入（B 系列不支持）。
- Read:** 断点操作属性为读。
- Write:** 断点操作属性为写。
- Size:** 断点的尺寸，以 Byte 为单位。， B 系列只支持 8bits 尺寸（Size=1）。
- Byte:** B 系列不支持选择。
- Objects:** B 系列不支持选择。
- Define:** 定义一个断点。
- Kill Selected:** 删除选择的断点。
- Kill All:** 删除全部的断点。

定义程序运行断点

用户在 **Expression** 中写入例如 C:0x1234 的表达式，C 表示是程序属性；0x1234 表示为程序代码地址。因为是程序运行断点，Access 中的 Read 和 Write 属性均不选择。然后点击 **Define** 按钮，在 **Current Breakpoints** 窗口中将出现该断点的表达式（断点 0），E C: 0x1234, "C:0x1234" 。

Expression 输入栏也可以接收符号，例如 main，表示在 main 函数位置添加断点。如果你有一个名字叫 tmp() 的函数，在 **Expression** 中写入 tmp，则可以在函数 tmp() 入口处添加一个断点。使用符号不需要在前面添加类似 C:0x1234 的空间表达式，因为符号已经可以提供空间信息。

定义 MOVC 读取断点

用户在 **Expression** 中写入例如 C:0x1234 的表达式，C 表示是程序属性；0x1234 表示为程序代码地址。因为是 MOVC 读取断点，Access 中的 Read 需要选中，Write 属性不选中。Size 选择为 1，选择其它数值仍将按照 8 处理。然后点击 **Define** 按钮，在 **Current Breakpoints** 窗口中将出现该断点的表达式。（断点 1），（A Read C: 0x1234 Len=8），“C:0x1234”。

注意：在定义 MOVC 读取断点时，Read 属性一定要选中，否则将成为程序运行断点。Write 不要选中，因为程序代码无法进行写操作。

Expression 输入栏也可以接收符号。例如，如果你定义了一个 code unsigned char tmp=0x55 的变量，你想在 MOVC 指令读取 tmp 时设置一个断点，可以在 **Expression** 加入 tmp，其它按照上面的设置，按动 Define 按钮后可以在 tmp 添加一个 MOVC 读取断点。

定义 MOVX 读取/写入断点

用户在 **Expression** 中写入例如 X:0x1234 的表达式，X 表示是数据空间属性；0x1234 表示为数据地址。因为是 MOVX 操作断点，Access 中的 Read/Write 至少必须有一项选中。Size 选择为 1，选择其它数值仍将按照 1 处理。然后点击 **Define** 按钮，在 **Current Breakpoints** 窗口中将出现该断点的表达式。（断点 1），（A Read X: 0x1234 Len=8），“X:0x1234”。

Expression 输入栏也可以接收符号。例如，如果你定义了一个 xdata unsigned char tmp 的外部数据空间变量，你想在 MOVX 指令操作 tmp 时设置一个断点，可以在 **Expression** 加入 tmp，其它按照上面的设置，按动 Define 按钮后可以在 tmp 添加一个 MOVX 读取断点。

注意：在定义 MOVX 读取断点时，Read/Write 至少必须有一项选中，否则将不能形成有效断点。

关闭已经存在的断点

对于 **Current Breakpoints** 窗口中存在的每个断点，用户都可以暂时关闭，使之断点作用失效。由于该失效的断点仍然存在于 **Current Breakpoints** 窗口中，因此用户可以在以后很方便重新起用它，这比删除这个断点然后重新增加一个断点要方便的多。如图 5.11 **Current Breakpoints** 窗口中，断点 2 处于关闭的状态。

B 系列仿真器不支持的断点设置

Count:	触发断点的次数（B 系列不支持）。
Command:	断点命令输入（B 系列不支持）。
Size:	断点的尺寸，以 Byte 为单位。B 系列只支持 8bits 尺寸（Size=1）。
Byte:	B 系列不支持选择。
Objects:	B 系列不支持选择。

六、TKS 仿真器 B 系列涉及的技术参数及处理

6.1 PSEN 输出

TKS 仿真器 B 系列全部支持内部/外部 64K 程序空间的仿真，程序运行时可能在仿真器内部/外部取指，因此 PSEN 的输出可能为高电平（仿真器内部取指）或低电平（仿真器外部取指）。

- **Use All Bus 选中：** 根据用户空间分配的情况而定，读取仿真器外部程序空间时为低，其余为高。
- **Use All Bus 不选中：** PSEN 永远为高电平。

6.2 ALE 信号输出

- **Use All Bus 选中：** ALE 信号在监控和运行状态都有输出。
- **Use All Bus 不选中：** ALE 信号在单步运行和全速运行时严格按照单片机的标准进行输出，但是在监控状态下 ALE 信号将输出固定电平，因为用户程序没有运行而处于冻结状态。用户可以采用 ALE 信号作为外部的时钟基准或其它用途，因为 ALE 信号的表现同实际的单片机机完全一致。

6.3 复位信号输入

TKS 仿真器 B 系列接受从用户板上进入的复位信号，并参与程序的运作。用户可以在仿真器设置时打开或关闭外部复位输入（见图 5.5 仿真器的设置）。

- 选择了关闭复位输入，则复位信号对于仿真器没有任何影响；
- 选择了开启复位输入且复位为无效复位信号，则外部复位信号对于仿真器没有任何影响；
- 选择了开启复位输入且复位为有效复位信号，则外部复位信号电平的变化将影响仿真器的运行：
 - 在单步或全速运行前发现复位信号，退出运行并提出警告。
 - 在全速运行期间发现复位信号，复位仿真器内部时序并重新开始运行，并提出警告。

6.4 使用外部电压的问题

TKS 仿真器在缺省时为使用仿真器内部的+5V 电源，这样处理时 TKS-HOOKS 最为稳定可靠。但是 TKS 仿真器 B 系列也支持外部电源，电压范围为 2.0V-5.5V。需要指出的是，使用外部电源对仿真器来讲具有很多不可预知的因素，可能会带来一些不稳定的情况，因此建议用户尽量使用仿真器内部提供的电源。需要注意的是，为了保证内部仿真芯片可靠工作，外部电源必须在 2.0V 到 5.5V 之间，小于 2.0 伏外部电源将不被使用，大于 5.5V 将会引起内部保护电路的启动。如果用户在调试过程中切换电源，当前仿真器芯片内部的状态将会被破坏，电源转换结束后仿真器内部仿真芯片处于复位后的状态。另外在切换电源过程中，用户已经调入的程序代码在一些极特殊的条件下可能会改变，转换结束后最好重新装载代码。注意：如果用户外部电源不稳定，仿真器可能会出现不正常的现象。

用户对电源的使用情况将作为配置保存下来，下次进入仿真环境后自动使用前一次的使用配置。

七、TKS-HOOKS 仿真器 B 系列的物理结构

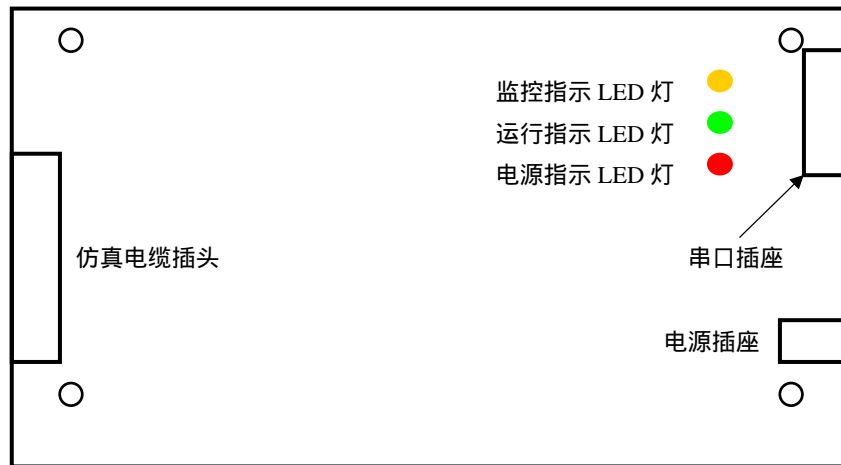


图 7.1 TKS-HOOKS 仿真器平面图（顶部俯视）

- 监控指示 LED 灯： 点亮表示处于监控状态。
- 运行指示 LED 灯： 点亮表示进入运行状态。
- 电源指示 LED 灯： 点亮表示系统电源正常。
- 仿真电缆插头： 插入仿真电缆连接到仿真头。
- 串口插座： 插入串口通讯电缆连接到计算机。
- 电源插座： 输入仿真器主机工作需要的电源。

八、仿真头组件的使用

TKS-B 系列仿真器通过仿真电缆与仿真头组件连接，然后再通过仿真头上的连接器插入到用户目标板上进行仿真。

8.1 仿真头组件结构

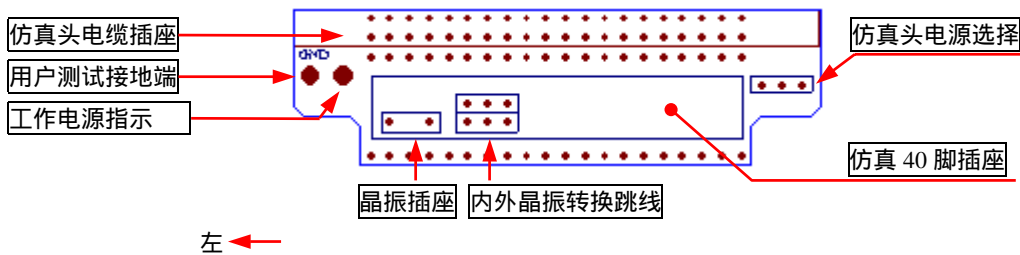


图 8.1 仿真头结构示意图（正面俯视）

(1) 仿真头主板

- 仿真头电缆插座:** 使用时用仿真电缆连接仿真器和仿真头。
- 用户测试接地端:** 用户在进行测试时，可以使用该端作为地参考点。
- 仿真 40 脚插座:** 用于连接仿真头和用户目标板。
- 工作电源指示:** 如果仿真头有电源接入（仿真器内部或外部用户板），则该 LED 点亮。
- 内外晶振转换跳线:** 用于转换当前使用的晶振。当该跳线位于右选择时，选择用户目标板上的晶振；当该跳线位于左选择时，选择仿真头**晶振插座**的晶振。用户板上的晶振和仿真头上的晶振可以同时存在。用户板上必须有 10-50P 的补偿电容，并根据用户使用振荡频率而变化。
- 晶振插座:** 用于插入晶振。
- 仿真头电源选择:** 用于选择仿真头的工作电源。当向左选择时，使用仿真器提供的+5V 电源；当向右选择时，使用外部用户板上输入的电源。当左右都不选择时，仿真头上的电路不工作。具体使用方法如下：
- 若用户使用仿真器内部时钟，可以左右都不选择。由于仿真头上无电源，所以时钟电路不工作，这样可以减少干扰，提高稳定性。
 - 若用户使用仿真头上产生的时钟且仿真器使用内部电源，一定向左选择使用仿真器内部电源。
 - 若用户使用仿真头上产生的时钟且仿真器使用用户板上提供的电源，在外部电源在 4V-5V 的稳定电压下可以考虑向右选择使用用户板上提供的电源，如果用户板上的电压小于 4V 建议仍然向左选择使用仿真器内部电源。
- 注意：尽量不要使用用户板上的电源对仿真头振荡电路供电。**

(2) 仿真电缆

仿真电缆一般为 40/50 芯（根据仿真器型号）扁平电缆。

8.2 仿真头的使用方法和注意事项

1. 用户在使用仿真器内部时钟时，仿真头上的时钟电路没有必要使用，因此可将**仿真头电源选择**的短路跳线取下，这样仿真头上的电路将失去工作电压而不工作。这在仿真一些对干扰敏感的用户系统中是有好处的。
2. 如果用户不使用仿真器内部提供的时钟，用户可以选择外部时钟（见图 5.5 **仿真硬件配置界面**）。使用外部时钟用户可以有 2 种选择方案：
 - 使用仿真头上的时钟振荡电路，晶振需要用户自己接入。晶振接入的位置有两种选择：
 - 在仿真头上的**晶振插座** 插入，**内外晶振转换跳线** 需要向左选择。
 - 在用户目标板上提供接入(XTAL1 和 XTAL2 间)。**内外晶振转换跳线**需向右选择。用户应注意晶振的品质，以及同仿真头之间能形成良好的接触。
 - 使用用户目标板上提供的时钟（不是晶振），必须从 XTAL1 接入时钟信号。用户提供的时钟要求有 50% 的占空比，如果差异太大在高频时钟仿真时可能达不到规定的最高频率。**内外晶振转换跳线** 需要向右选择。

注意：在使用仿真器外部时钟（包括从用户板上直接输入时钟）时都要使用仿真头上的时钟振荡电路，因此**仿真头电源选择** 必须向左或向右选择，不能悬空。
3. 在仿真一些用户系统时，如果发生一些很难解释的现象，在某些情况下是由于仿真器和用户目标系统的地线连接不是很理想。用户可以尝试增加另外的地线连接，从仿真头上的接地端子**用户测试接地端**上连接另外一根地线到用户系统地上，并且可以尝试用户目标板上不同位置的接地点。
4. 如果用户选择了使用外部电源，则仿真芯片工作电源将由用户的目标板提供（电压范围为 2.0V-5.5V），从**仿真 40 脚插座** 的第 40 脚输入（或 PLCC44 仿真头的 44 脚）。用户提供的电源除了有电压要求外，还要求稳定，不能有电压的瞬间起伏，否则会引起仿真状态的破坏。

九、TKS 仿真器 B 系列的限制

看门狗的仿真

TKS 仿真器 B 系列不全部支持看门狗的仿真。用户如果需要打开看门狗，只能在程序中运行看门狗启动程序，并且全速运行。如果用户在程序全速运行后打开了看门狗，并又终止程序运行并返回到监控状态，则系统将复位。因为看门狗打开后只能作全速运行，返回到仿真器监控状态后看门狗还将继续运作，最终很快溢出并引起复位。

如果在全速运行中用户通过运行程序打开看门狗，并且在看门狗溢出前程序正常喂狗，用户程序将正常的运行；如果用户没能够及时喂狗而发生了溢出，仿真芯片将发生复位。仿真器监控系统能够发现这种现象并复位仿真器的运行时序重新开始运行，在信息输出窗口提示用户。

P0、P2 口差异

在 HOOKS 仿真技术中，P0、P2 口是重新构造的，这种做法是因为 HOOKS 仿真原理所要求的。在 TKS-HOOKS 的 B 系列仿真器中，P0、P2 口完全是根据 80C51 单片机的内部结构仿照 IC 设计重新构造出的，无论是内部的时序，I/O 口的驱动能力等参数指标都是正确的，对用户在使用中没有任何的限制。

十、TKS 仿真器 B 系列使用中的常见问题

1. TKS 同计算机通讯失败

如果出现通讯失败将无法进入硬件仿真，出现这种现象的原因一般是：

- 仿真器同 PC 的通讯电缆连接不良。
- 仿真器没有电源。
- 错误的选择了仿真器硬件驱动程序。
- 串口选择错误，或者该串口已经被其它设备占用。
- 计算机串口已经损坏。

2. 不能使用用户板提供的电源

TKS 仿真器 B 系列可以使用外部用户板提供的电源，但在以下情况下无法使用外部电源：

- 外部电源没有从仿真头的电源输入脚引入，或因为接触不好而无法引入。
- 用户外部输入的电源电压没有在 2.0V-5.5V 之间，或不稳定。
- 当前 TKS 系列仿真器内部的仿真芯片无法支持低电压仿真，需要更换可以低电压操作的仿真芯片。
- 用户没有选择使用外部电源（见图 5.5 仿真硬件配置界面）。

3. 单步运行后仿真器会进入“死机”状态，且运行和监控指示灯全部点亮

这种现象一般出现在 C 语言调试中，或者 C 语言和汇编语言混合调试中。如果当用户当前的源程序窗口是 C 源程序，但是当前 PC 的位置却并没有在 C 源程序的有效范围内，单步运行实际上是要求仿真器单步运行一条 C 程序语句，到下一条 C 程序语句位置停止。由于当前的 PC 位置并没有处于 C 的有效位置，因此仿真器只能单步运行每一条汇编程序到下一条 C 程序语句位置。如果这期间的汇编语句较多，可能会耗费较多的时间。仿真器连续的单步运行，不再接收其它命令，好象进入死机状态。连续快速的在单步运行和监控不断切换，所以监控和运行指示灯会全亮。

例如，如果用户当前的 PC=0，这时在 C 源程序中不会有程序运行位置指示，但是在汇编窗口中运行指示在 PC=0000H 位置。如果用户把当前的窗口点击到 C 程序窗口，则仿真器认为下一个单步将运行到 main() 函数位置，在用户要求单步运行后将单步运行 PC=0000H 到 main() 函数期间的所有汇编指令，可能要耗费较多的时间。

采用全速运行加断点的方式可以解决这个问题。

4. 无法使用外部时钟

- 仿真头/用户板上没有晶振，或者晶振损坏不能正常工作。
- 仿真头上的晶振选择跳线没有正确设置。
- 仿真电缆同仿真器主机和仿真头之间接触不良。
- 用户没有选择使用外部时钟（见图 5.5 仿真硬件配置界面）。

十一、TKS 仿真器 B 系列的升级

TKS 仿真器 B 系列出厂时内部放置一个固定的仿真芯片（表 1），由于 80C51 系列的兼容性所以可以仿真向下的 80C51 芯片。另外，有些仿真器在出厂时备有多个其它品种的仿真芯片，这些备用仿真芯片可以替换仿真器中的原有芯片以便改变现有仿真器的仿真特性。备用仿真芯片的品种和数量跟仿真器的型号有关系，其中 TKS-591B 支持的仿真芯片最多；当 PHILIPS 推出新型号的 80C51 单片机后，TKS 仿真器的 B 系列用户可以申请新型号的仿真芯片。

用户可以请当地经销商来更换备用仿真芯片，也可以通过寄回更换，或者在我们的技术支持下自己更换。TKS 仿真器 B 系列用户享受定期的升级服务，其中包含免费的软件升级。请定期留意我们的主页上的升级信息，或直接使用技术支持电话询问升级事项。

十二、结束语

TKS 仿真器 B 系列是经过精心设计的 80C51 系列仿真器, 采用了 PHILIPS 公司授权的 HOOKS 仿真技术, 并内嵌独创的 HOOKS 时序同步技术。它的 2.0V-5.5V 外部电压仿真, 任意内部/外部 64K 程序仿真, 外部时钟输入仿真, 真正外部复位仿真都使 HOOKS 技术达到了一个新的水准。在仿真芯片的更换上更是采用了复杂而先进的自适应技术, 基本适应全部的不同封装芯片仿真。用户不需要更换昂贵的仿真头, 直接插入廉价的仿真芯片, 极大地降低了用户的开发费用。

如果您遇到关于 PHILIPS 80C51 单片机系列的任何问题, 请致电我们的技术支持, 也可以访问我们的主页 <http://www.zlgmcu.com>。从产品的设计思想、原器件的选型、开发工具的选择、调试的方法、可靠性的设计, 我们的技术支持工程师都能为您提供全套的解决方案或良好的建议。