

RX族 仿真/调试程序 V.1.01

用户手册

瑞萨单片机开发环境系统

本资料所记载的内容，均为本资料发行时的信息，瑞萨电子对于本资料所记载的产品或者规格可能会作改动，恕不另行通知。
请通过瑞萨电子的主页确认发布的最新信息。

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.
 - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.
 - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.
 - "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

前言

有关本手册

本手册说明附属在 High-performance Embedded Workshop (HEW) 的仿真 / 调试程序的调试功能。

在本手册中，没有说明 C/C++ 语言和汇编语言的编程方法、操作系统的使用方法以及适合各器件的程序编程方法等。有关这些内容，请参照各手册。

HEW 在安装时被定义为各种语言。

Microsoft、MS-DOS、Windows、Windows NT 是美国 Microsoft 公司在美国以及其他国家的注册商标。

Visual SourceSafe 是 Microsoft 公司在美国以及其他国家的商标。

IBM 是 International Business Machines Corporation 的注册商标。

其他记载的产品名是各公司的商标或者注册商标。

本手册的符号

以下说明本手册中使用的符号的含义。

表 2. 符号一览表

符号	含义
[Menu->Menu Option]	粗体字和“->”表示菜单选项（例如，[File->Save As..]（文件->另存为...））。
FILENAME.C	大写字母的名字表示文件名。
“字符串的输入”	下划线表示输入的字符串（省略“”）。
Key + Key	表示键输入。例如，CTRL+N 键是指同时按 CTRL 键和 N 键。
☞ （“操作方法”的记号）	此记号在左端时，其右边的文章表示某种操作方法。

有关产品内容和本手册的咨询处

瑞萨电子公司
网页

<http://www.renesas.com/en/tools>

目 录

1.	前言	1
2.	仿真 / 调试程序的功能	2
2.1	特点	2
2.2	调试对象程序	2
2.3	仿真范围	3
2.4	存储器管理	3
2.5	指令执行的复位处理	4
2.6	异常处理	4
2.7	字节序	4
2.7.1	CPU 的字节序	4
2.7.2	外部存储区的字节序	4
2.8	外围功能仿真	5
2.8.1	定时器	5
2.8.2	串行通信接口	6
2.8.3	中断控制器	10
2.8.4	时钟	11
2.8.5	外围功能的使用	11
2.9	跟踪	11
2.10	标准输入 / 输出和文件输入 / 输出的处理	12
2.11	暂停条件	12
2.12	浮点数据	15
2.13	函数调用履历的显示	15
2.14	性能测量	15
2.14.1	剖析功能	15
2.14.2	性能分析	15
2.15	伪中断	16
2.16	覆盖率	16
3.	调试	17
3.1	工作空间的创建	18
3.1.1	目标的选择	18
3.1.2	仿真程序工作空间的设置	19
3.2	仿真 / 调试程序的启动	20
3.3	仿真 / 调试程序的设置更改	20
3.3.1	CPU 字节序和工作频率的设置	20
3.3.2	仿真系统的设置更改	21
3.3.3	存储器映像和存储器资源的设置更改	22
3.3.4	存储器映像的设置对话框	23
3.3.5	存储器资源的设置对话框	24
3.4	设置外围功能仿真	25
3.4.1	外围功能仿真模块的注册	25
3.4.2	外围功能的地址更改	26
3.4.3	外围功能中断源信息的更改	27
3.4.4	控制寄存器的存储器资源	28
3.4.5	被连接的外围功能的确认	28
3.4.6	虚拟端口的文件输入 / 输出	29
3.5	存储器的操作	32
3.5.1	窗口显示内容的定期更新	32

3.5.2	I/O 区域内容的显示和更改	32
3.6	仿真 / 调试程序断点的使用	32
3.6.1	断点一览表的显示	33
3.6.2	断点的设置	34
3.6.3	断点设置内容的更改	41
3.6.4	允许断点	41
3.6.5	禁止断点	41
3.6.6	断点的删除	41
3.6.7	断点的全部删除	41
3.6.8	断点的源行显示	41
3.6.9	输入 / 输出文件的关闭	41
3.6.10	输入 / 输出文件的全部关闭	41
3.7	跟踪信息的查看	42
3.7.1	跟踪窗口的打开	42
3.7.2	跟踪信息获取条件的设置	42
3.7.3	跟踪事件的设置	43
3.7.4	跟踪信息的获取	44
3.7.5	跟踪信息的检索	47
3.7.6	跟踪信息的筛选	48
3.7.7	跟踪信息的清除	48
3.7.8	跟踪信息的文件保存	48
3.7.9	源文件的显示	49
3.7.10	时戳显示的切换	49
3.7.11	函数执行履历的显示	50
3.8	剖析信息的查看	51
3.8.1	堆栈信息文件	51
3.8.2	堆栈信息文件的加载	52
3.8.3	允许剖析	53
3.8.4	测量方法的指定	53
3.8.5	用户程序执行结果的确认	53
3.8.6	[List] 表	53
3.8.7	[Tree] 表	54
3.8.8	剖析图窗口	56
3.8.9	显示数据的种类和用途	57
3.8.10	剖析信息文件的创建	58
3.8.11	注意事项	58
3.9	性能分析	59
3.9.1	性能分析窗口的打开	59
3.9.2	评估函数的设置	60
3.9.3	数据收集的开始	60
3.9.4	数据的复位	60
3.9.5	评估函数的删除	60
3.9.6	评估函数的全部删除	60
3.9.7	当前显示内容的保存	60
3.10	代码覆盖率的测量	61
3.10.1	覆盖率窗口的打开	61
3.10.2	覆盖率信息的全部获取	62
3.10.3	覆盖率信息的全部清除	62
3.10.4	源文件的显示	63
3.10.5	新覆盖率测量范围的设置	63
3.10.6	覆盖率测量范围的更改	63
3.10.7	覆盖率测量范围的删除	64

3.10.8	覆盖率信息的获取	64
3.10.9	覆盖率信息的清除	64
3.10.10	覆盖率信息的文件保存	64
3.10.11	覆盖率信息文件的加载	64
3.10.12	最新信息的更新	65
3.10.13	[Confirmation Request] (确认请求) 对话框	65
3.10.14	覆盖率数据的保存对话框	65
3.10.15	编辑窗口的覆盖率结果显示	66
3.10.16	反汇编窗口的覆盖率结果显示	66
3.11	伪中断的手动产生	67
3.11.1	触发窗口	67
3.11.2	GUI I/O 窗口	69
3.12	标准输入 / 输出和文件输入 / 输出	71
3.12.1	仿真 I/O 窗口的打开	71
3.12.2	输入 / 输出功能	71
3.13	虚拟输入 / 输出面板的创建	73
3.13.1	GUI 输入 / 输出窗口的打开	73
3.13.2	创建按钮	74
3.13.3	创建标签	75
3.13.4	创建 LED	76
3.13.5	文本字符串的创建	78
3.13.6	项目大小 / 配置的更改	79
3.13.7	项目的复制	79
3.13.8	项目的删除	79
3.13.9	网格线的显示	79
3.13.10	输入 / 输出面板信息的保存	80
3.13.11	输入 / 输出面板信息的读取	80
4.	窗口	81
5.	命令行	82
5.1	命令一览表 (按功能分类)	82
5.1.1	执行相关	82
5.1.2	下载相关	82
5.1.3	寄存器操作相关	82
5.1.4	存储器操作相关	83
5.1.5	汇编 / 反汇编相关	83
5.1.6	暂停设置相关	83
5.1.7	跟踪相关	84
5.1.8	覆盖率测量相关	84
5.1.9	性能测量相关	84
5.1.10	监视相关	84
5.1.11	脚本 / 记录文件相关	85
5.1.12	映像相关	85
5.1.13	仿真设置相关	85
5.1.14	标准输入 / 输出和文件输入 / 输出相关	85
5.1.15	实用程序相关	85
5.1.16	工程 / 工作空间相关	86
5.1.17	测试支援功能相关	86
5.1.18	实时 OS 对应调试功能相关	86
5.1.19	虚拟端口文件输入 / 输出相关	87
5.2	命令一览表 (按字母顺序)	88

6.	信息一览表	92
6.1	信息	92
6.2	错误信息	93
7.	使用指南	94
7.1	调试的准备	94
7.1.1	示例程序	94
7.1.2	示例程序的创建	94
7.2	调试的设置	94
7.2.1	存储器资源的确保	94
7.2.2	示例程序的下载	96
7.2.3	源程序的显示	97
7.2.4	PC 断点的设置	98
7.2.5	剖析的设置	99
7.2.6	仿真 I/O 的设置	100
7.2.7	跟踪信息获取条件的设置	101
7.2.8	堆栈指针和程序计数器的设置	101
7.3	调试的开始	102
7.3.1	程序的执行	102
7.3.2	跟踪缓冲器的使用方法	104
7.3.3	跟踪检索的执行	105
7.3.4	仿真 I/O 的确认	106
7.3.5	断点的确认	106
7.3.6	变量的参照	107
7.3.7	程序的单步执行	108
7.3.8	剖析信息的确认	111

1. 前言

仿真 / 调试程序具有瑞萨电子单片机的 CPU 仿真功能和调试功能。能通过使用仿真 / 调试程序，高效率地调调用 C/C++ 语言或者汇编语言编写的程序。

仿真 / 调试程序和 High-performance Embedded Workshop(HEW) 一起运行。HEW 为瑞萨电子单片机提供图形用户界面，简单地开发和调调用 C/C++ 语言和汇编语言编写的应用程序，以高功能并且提供直观的手段，存取、测量以及改变要执行应用程序的仿真 / 调试程序。

在使用仿真 / 调试程序前，请参照仿真 / 调试程序和 HEW 的帮助。

2. 仿真 / 调试程序的功能

本章说明 RX600 系列仿真 / 调试程序的功能。

2.1 特点

本仿真 / 调试程序有以下特点：

1. 因为在主计算机上运行，即使没有实际的用户系统也能开始程序的调试，从而能缩短整个系统的开发时间。
2. 在仿真时，计算程序的指令执行周期数和指令执行时间。因此，即使没有实际的用户系统也能进行性能评价。
3. 具有伪中断功能和仿真 I/O 功能，能进行简单的系统级仿真。
4. 具有以下功能，能高效率地进行程序的测试和调试。
 - 对应 RX600 系列的各 CPU。
 - 在执行用户程序过程中发生异常时，具有能控制忽视异常而继续执行或者停止执行的功能。
 - 获取剖析数据以及测量函数单位的性能。
 - 具有丰富的暂停功能。
 - 设置和编辑存储器映像。
 - 显示函数调用履历。
 - 显示 C/C++ 和汇编源级的覆盖率。
 - 具有通过图像显示和波形显示的视觉调试功能。
5. 在 Windows® 上运行，能在对话框中设置断点、存储器映像、性能和跟踪，也能在对话框中设置对应 RX600 单片机各种存储器映像的环境。

还有以下特点：

- 直观的用户界面
- 联机帮助
- 共同的显示和操作性

2.2 调试对象程序

仿真 / 调试程序可以对 ELF/DWARF2 格式的装入模块进行符号调试。对于其他格式的装入模块，能下载装入模块并且执行该模块的指令，但是不能进行符号调试。详细内容请参照《High-performance Embedded Workshop User's Manual》。

2.3 仿真范围

1. 支持RX600系列的仿真。
2. 仿真/调试程序支持RX600系列单片机的下述功能。
 - 全执行指令
 - 异常处理
 - 寄存器
 - 全地址空间
3. 仿真/调试程序不支持RX600系列单片机的下述功能。使用下述功能的程序必须使用RX600系列的仿真器进行调试。

No.	项目	备注
1	低功耗状态	如果执行 WAIT 指令就停止仿真。
2	非屏蔽中断 (NMI)	
3	下述指令执行中途的中断接受 (RMPA、SCMPU、SMOVF、SMOVB、SMOVU、SSTR、SUNTIL、SWHILE)	在指令执行结束时接受中断。
4	指令执行结束后不确定的数据和寄存器的值	
5	累加器 (ACC) 的低 16 位	

2.4 存储器管理

(1) 存储器映像的设置

存储器映像用于计算仿真时的存储器的存取周期数。能设置的项目如下：

- 存储器类型
- 存储区的起始位置和结束位置
- 存储器的存取周期数
- 存储器的数据总线宽度
- 字节序

在存储器映像中，只能对外部区域指定字节序。内部 ROM 区域和内部 RAM 区域的字节序使用在启动仿真 / 调试程序时显示的 [Set Simulator]（设置仿真程序）对话框的 [CPU Configuration]（CPU 结构）选项卡中指定的 [Endian]（字节序）。

详细内容请参照“3.3.3 存储器映像和存储器资源的设置更改”。

(2) 存储器资源的设置

为了装入和执行用户程序，需要设置存储器资源。能设置的项目如下：

- 起始地址
- 结束地址
- 存取类型

存取类型有可读写、只读和只写。

在用户程序中写只读存储器等的非法存取时会出错，因此能检测到错误的存储器存取。

有关存储器资源的设置的详细内容，请参照“3.3.3 存储器映像和存储器资源的设置更改”。

2.5 指令执行的复位处理

仿真 / 调试程序在以下情况下对指令执行数、指令执行周期数和指令执行时间进行复位。

- ① 在指令仿真停止后重新执行前更改了 PC。
- ② 执行了指定起始地址的 Run 命令。
- ③ 初始化或者加载了程序。

2.6 异常处理

仿真 / 调试程序检测 RX600 系列单片机发生的异常并仿真异常处理，因此能进行发生异常时的仿真。

异常处理的仿真按以下步骤进行：

- (a) 在执行指令过程中检测异常的发生。
- (b) 将 PC 和 PSW 保存到专用寄存器（高速中断的情况）或者堆栈区（一般中断的情况）。如果在保存处理中发生错误，就中止异常处理并且在显示发生的异常处理错误后返回到仿真 / 调试程序的命令等待状态。
- (c) 设置 PSW 的以下的位。
U=0
I=0
PM=0
- (d) 从对应向量号的向量地址开始读起始地址。如果在读处理中发生错误，就中止异常处理并且在显示发生的异常处理错误后返回到仿真 / 调试程序的命令等待状态。
- (e) 从起始地址开始执行指令。

2.7 字节序

2.7.1 CPU 的字节序

在启动仿真 / 调试程序时显示的 [Set Simulator]（设置仿真程序）对话框的 [CPU Configuration]（CPU 结构）选项卡中，指定 CPU 的字节序。

CPU 的字节序适用于内部 ROM 和内部 RAM。

指定的详细内容请参照“3.3.1 CPU 字节序和工作频率的设置”。

2.7.2 外部存储区的字节序

在 [Set Memory Map]（设置存储器映像）对话框中设置外部存储区的字节序。

指定的详细内容请参照“3.3.4 存储器映像的设置对话框”。

2.8 外围功能仿真

2.8.1 定时器

(1) 支持范围

RX600 系列仿真 / 调试程序支持由 2 个通道的 16 位定时器构成的比较匹配定时器（CMT）（2 个单元（单元 0、单元 1），共 4 个通道）。

(2) 控制寄存器

仿真 / 调试程序支持的 CMT 控制寄存器如表 2.1 所示。

表中支持状况的“○”表示支持。

必须以寄存器长度为单位存取控制寄存器。

表 2.1 仿真 / 调试程序支持的 CMT 控制寄存器

外围功能模块名	单元	控制寄存器	支持状况
CMT	单元 0	CMSTR0	○
		CMCR0	○
		CMCNT0	○
		CMCOR0	○
		CMCR1	○
		CMCNT1	○
		CMCOR1	○
	单元 1	CMSTR1	○
		CMCR2	○
		CMCNT2	○
		CMCOR2	○
		CMCR3	○
		CMCNT3	○
		CMCOR3	○

能在 [Peripheral Module Configuration]（外围模块结构）对话框中参照和更改控制寄存器的地址。有关 [Peripheral Module Configuration]（外围模块结构）对话框的详细内容，请参照“3.4 外围功能仿真的设置”。

2.8.2 串行通信接口

(1) 支持范围

RX600 系列仿真 / 调试程序支持 7 个通道的串行通信接口（SCI）。仿真 / 调试程序支持的 SCI 功能如表 2.2 所示。

表中支持状况的“○”表示支持，“—”表示不支持。

表 2.2 仿真 / 调试程序支持的 SCI 功能

项目		支持状况	
串行通信方式	异步 / 时钟同步	○	
	智能卡接口	—	
内部波特率发生器的时钟源	PCLK 时钟	○	
	PCLK/4 时钟、PCLK/16 时钟和 PCLK/64 时钟	—	
全双工通信		○	
中断源	发送结束、发送数据字节序、接收数据满、接收错误	○	
异步模式	数据长	7 位 / 8 位	○
	发送停止位	1 位 / 2 位	○
	奇偶校验功能	偶数 / 奇数 / 无	○
	接收错误检测功能	奇偶校验错误、溢出错误、帧错误	○
	暂停检测		—
	时钟源	内部时钟	○
	外部时钟、TMR 的传送率时钟	—	
时钟同步模式	数据长	8 位	○
	接收错误的检测	溢出错误	○

(2) 控制寄存器

仿真 / 调试程序支持的 SCI 控制寄存器如表 2.3 所示。

表中支持状况的“○”表示支持，“Δ”表示只支持“2.8.1(1) 支持范围”中说明的功能的相关位。必须以寄存器长度为单位存取控制寄存器。

表 2.3 仿真 / 调试程序支持的 SCI 控制寄存器

外围功能模块名	通道	控制寄存器	支持状况
SCI	0 ~ 6	SMR	Δ
		BRR	○
		SCR	Δ
		TDR	○
		SSR	Δ
		RDR	○
		SCMR	Δ
		SEMR	Δ

能在 [Peripheral Module Configuration]（外围模块结构）对话框中参照和更改控制寄存器的地址。有关 [Peripheral Module Configuration]（外围模块结构）对话框的详细内容，请参照“3.4 外围功能仿真的设置”。

(3) 数据的输入 / 输出

仿真 / 调试程序将部分引脚作为虚拟端口分配到存储器，调试对象程序和调试程序能通过此虚拟端口存取引脚。

SCI 的虚拟端口地址如表 2.4 所示。

表 2.4 SCI 的虚拟端口地址

通道	虚拟端口名	地址	存取长度	功能
0	RxD0	H'00088224	16	通道 0 接收数据
	TxD0	H'00088226	16	通道 0 发送数据
1	RxD1	H'00088228	16	通道 1 接收数据
	TxD1	H'0008822A	16	通道 1 发送数据
2	RxD2	H'0008822C	16	通道 2 接收数据
	TxD2	H'0008822E	16	通道 2 发送数据
3	RxD3	H'00088230	16	通道 3 接收数据
	TxD3	H'00088232	16	通道 3 发送数据
4	RxD4	H'00088234	16	通道 4 接收数据
	TxD4	H'00088236	16	通道 4 发送数据
5	RxD5	H'00088238	16	通道 5 接收数据
	TxD5	H'0008823A	16	通道 5 发送数据
6	RxD6	H'0008823C	16	通道 6 接收数据
	TxD6	H'0008823E	16	通道 6 发送数据

虚拟端口 RxD 的结构、虚拟端口 TxD 的结构以及 RxD 和 TxD 的位功能分别如表 2.5、表 2.6 和表 2.7 所示。

表 2.5 RxD 的结构

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SB	PE	FE	—	—	—	—	—	D7	D6	D5	D4	D3	D2	D1	D0

表 2.6 TxD 的结构

位	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SB	—	—	—	—	—	—	—	D7	D6	D5	D4	D3	D2	D1	D0

表 2.7 RxD 和 TxD 的位功能

位	位名	初始值	R/W	说明
0	D0	0	R/W	数据位 当 8 位数据时，使用 D7 ~ D0。 当 7 位数据时，使用 D6 ~ D0。
1	D1	0	R/W	
2	D2	0	R/W	
3	D3	0	R/W	
4	D4	0	R/W	
5	D5	0	R/W	
6	D6	0	R/W	
7	D7	0	R/W	
12 ~ 8	—	全 0	—	保留位 读写值都为“0”。
13	FE	0	R/W	帧错误位 当此位被置“1”时，SCI 发生帧错误。
14	PE	0	R/W	奇偶校验错误位 当此位被置“1”时，SCI 发生奇偶校验错误。
15	SB	1	R/W	开始位 发送侧在发送开始时从“1”变为“0”；在发送结束时从“0”变为“1”。

仿真 / 调试程序将数据的发送和接收抽象化，同时进行数据的发送和接收。仿真 / 调试程序的接收和发送分别如图 2.1 和图 2.2 所示。

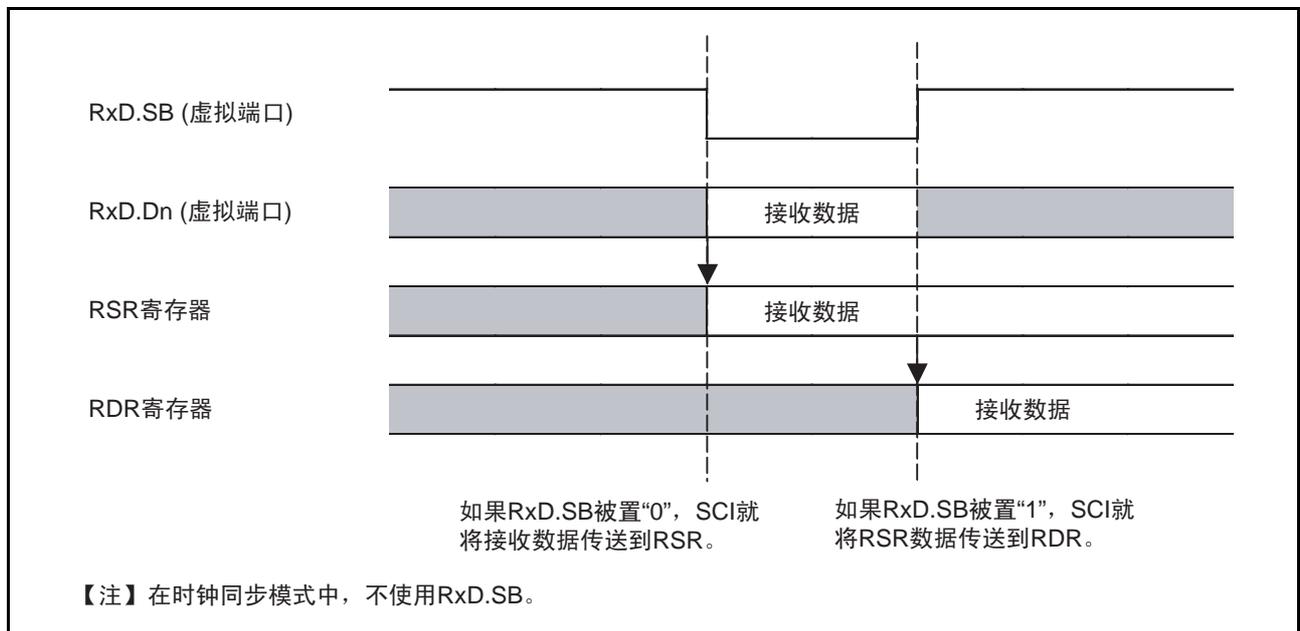


图 2.1 仿真 / 调试程序的接收

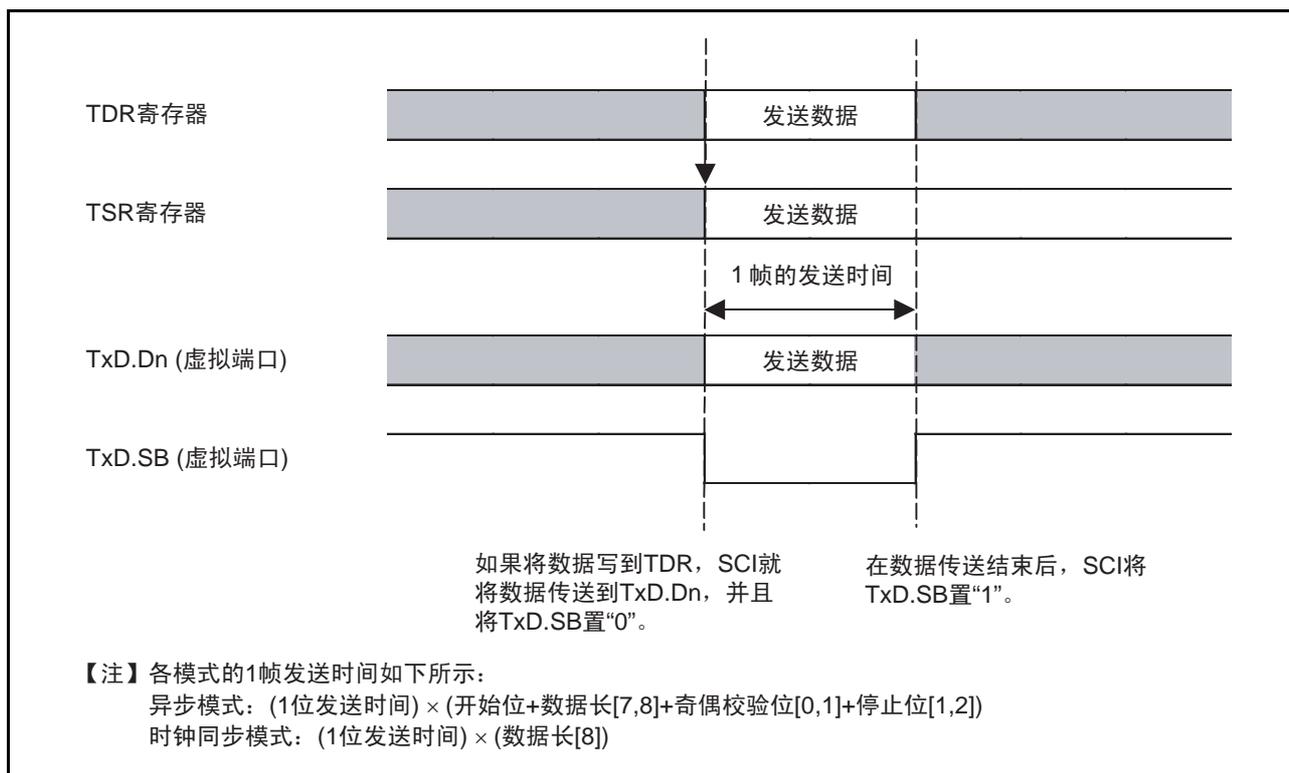


图 2.2 仿真 / 调试程序的发送

本仿真 / 调试程序能对虚拟端口进行文件的输入 / 输出，详细内容请参照“3.4.6 虚拟端口的文件输入 / 输出”。

2.8.3 中断控制器

(1) 支持范围

RX600 系列仿真 / 调试程序支持 CMT 和 SCI 的相关中断控制器 (ICU)。
只支持向 CPU 请求的中断，而不支持 DTC 和 DMAC 的启动。

(2) 控制寄存器

仿真 / 调试程序支持的 ICU 控制寄存器如表 2.8 所示。
表中支持状况的“○”表示支持，“△”表示只支持“2.8.1(1) 支持范围”中说明的功能的相关位。
必须以寄存器长度为单位存取控制寄存器。

表 2.8 仿真 / 调试程序支持的 ICU 控制寄存器

外围功能模块名	控制寄存器	支持状况
ICU	IRn (n=028 ~ 029、214 ~ 241)	○
	ISELR028	△
	ISELR029	△
	ISELR030	△
	ISELR031	△
	ISELR215	△
	ISELR216	△
	ISELR219	△
	ISELR220	△
	ISELR223	△
	ISELR224	△
	ISELR227	△
	ISELR228	△
	ISELR231	△
	ISELR232	△
	ISELR235	△
	ISELR236	△
	ISELR239	△
	ISELR240	△
	IER03	△
	IER1A	△
	IER1B	○
	IER1C	○
	IER1D	○
	IER1E	△
	IPRm (m=04 ~ 07、80 ~ 86)	○
FIR	○	

能在 [Peripheral Module Configuration] (外围模块结构) 对话框中参照和更改控制寄存器的地址、中断向量号和中断优先级寄存器的位置。有关 [Peripheral Module Configuration] (外围模块结构) 对话框的详细内容，请参照“3.4 外围功能仿真的设置”。

(3) 使用 ICU 时的注意事项

能选择在发生中断时是否暂停。

请通过 [Simulator System]（仿真系统）对话框或者 EXEC_STOP_SET 命令进行设置。

2.8.4 时钟

仿真 / 调试程序支持与存储器存取有关的系统时钟、外围功能时钟和定时器运行时钟。

通过存储器映像指定的周期数为内部时钟。请在 [Set Peripheral Function Simulation]（设置外围功能仿真）对话框中设置系统时钟和外围功能时钟的比。

必须通过定时器的控制寄存器指定定时器运行时钟的分频率。

2.8.5 外围功能的使用

要使用外围功能时，需要在仿真 / 调试程序启动时显示的 [Set Peripheral Function Simulation]（设置外围功能仿真）对话框中注册要使用的模块。

模块注册的详细内容请参照“3.4 外围功能仿真的设置”。

2.9 跟踪

仿真 / 调试程序将执行结果写到跟踪缓冲器。在 [Trace Acquisition]（跟踪获取）对话框中指定跟踪信息的获取条件。[Trace Acquisition]（跟踪获取）对话框能通过 [Trace]（跟踪）窗口上单击鼠标的右键显示弹出式菜单，然后选择 [Acquisition...]（获取 ...）进行显示。在 [Trace]（跟踪）窗口中显示获取的跟踪信息。

能查找跟踪信息。在 [Find]（查找）对话框中设置查找的条件。[Find]（查找）对话框能通过 [Trace]（跟踪）窗口上单击鼠标的右键显示弹出式菜单，然后选择 [Find -> Find...]（查找 -> 查找 ...）进行显示。

详细内容请参照“3.7 跟踪信息的查看”。

2.10 标准输入 / 输出和文件输入 / 输出的处理

仿真 / 调试程序能从用户程序进行标准输入 / 输出和文件输入 / 输出。在使用输入 / 输出功能时，必须打开 [Simulated I/O]（仿真 I/O）窗口。

支持的输入 / 输出处理如下：

表 2.9 输入 / 输出功能一览表

No.	功能码	功能名	内容
1	H'21	GETC	从标准输入进行 1 字节的输入。
2	H'22	PUTC	给标准输出进行 1 字节的输出。
3	H'23	GETS	从标准输入进行 1 行的输入。
4	H'24	PUTS	给标准输出进行 1 行的输出。
5	H'25	FOPEN	打开文件。
6	H'06	FCLOSE	关闭文件。
7	H'27	FGETC	从文件进行 1 字节的输入。
8	H'28	FPUTC	给文件进行 1 字节的输出。
9	H'29	FGETS	从文件进行 1 行的输入。
10	H'2A	FPUTS	给文件进行 1 行的输出。
11	H'0B	FEOF	检查文件的结束。
12	H'0C	FSEEK	移动文件的指针。
13	H'0D	FTELL	获取文件指针的当前位置。

输入 / 输出功能的详细内容请参照“3.12 标准输入 / 输出和文件输入 / 输出”。

2.11 暂停条件

用户程序仿真的暂停条件如下：

- 暂停命令的条件成立引起的暂停
- 在执行用户程序时检测到错误而引起的暂停
- 跟踪缓冲器满引起的暂停
- 执行 WAIT 指令引起的暂停
- [STOP]（停止）按钮引起的暂停

(1) 暂停命令的条件成立引起的暂停

设置暂停条件的命令有以下 9 种：

- BREAKPOINT : 指令的执行位置引起的暂停
- BREAK_ACCESS : 存储器的存取范围引起的暂停
- BREAK_CYCLE : 执行周期数引起的暂停
- BREAK_DATA : 给存储器写数值引起的暂停
- BREAK_DATA_DIFFERENCE : 存储器的数值变化量（差分）引起的暂停
- BREAK_DATA_INVERSE : 存储器的数值符号取反引起的暂停
- BREAK_DATA_RANGE : 存储器的数据范围引起的暂停
- BREAK_REGISTER : 给寄存器写数值引起的暂停
- BREAK_SEQUENCE : 指定执行顺序的暂停

如果将暂停条件成立时的运行指定为 [Stop]（停止），就在该暂停条件成立时中止程序的执行。详细内容请参照“3.6 仿真 / 调试程序断点的使用”。

在用户程序执行过程中，暂停条件成立并且中止程序的执行时的处理（是在执行断点指令前停止还是在执行断点指令后停止）如表 2.10 所示。

表 2.10 暂停条件成立时的处理

命令名	暂停条件成立指令	
	执行	不执行
BREAKPOINT		○
BREAK_ACCESS	○	
BREAK_CYCLE	○	
BREAK_DATA	○	
BREAK_DATA_DIFFERENCE	○	
BREAK_DATA_INVERSE	○	
BREAK_DATA_RANGE	○	
BREAK_REGISTER	○	
BREAK_SEQUENCE		○

对于 BREAKPOINT 和 BREAK_SEQUENCE，如果将断点设置在执行指令的起始位置以外的位置，就无法检测暂停。

如果在用户程序执行过程中暂停条件成立，就在 [Output]（输出）窗口中显示暂停条件成立的信息后中止指令的执行。

(2) 在执行用户程序时检测到错误而引起的暂停

为了检测 CPU 异常发生功能无法检测的程序错误，仿真 / 调试程序设置了仿真错误。能通过 [Simulator System]（仿真系统）对话框选择在发生错误时是停止还是继续仿真。错误类型、错误信息、错误发生原因以及继续模式时的仿真 / 调试程序的运行如表 2.11 所示。

表 2.11 仿真错误一览表

错误类型 / 信息	错误发生的原因	继续模式时的处理
存储器的存取错误 / Memory Access Error (Address:H'nnnnnnnn)	<ul style="list-style-type: none"> 存取没有确保的存储区。 写只读存储器。 读只写存储器。 存取没有存储器的区域。 	写存储器时，什么也不写。 读存储器时，读到全位“1”。

在停止模式中，如果发生仿真错误，仿真 / 调试程序就在中止指令的执行后显示错误信息，然后返回到命令等待状态。仿真错误停止后的 PC 状态如表 2.12 所示，而且仿真错误停止后的 PSW 内容不变。

表 2.12 仿真错误停止时的寄存器

错误的类型	PC 的内容
存储器的存取错误	当读指令时： 发生错误的指令的起始地址 当执行指令时： 发生错误的指令的下一条指令地址

当程序中有发生仿真错误的指令时，必须按以下顺序进行调试：

- (a) 先通过停止模式执行程序，然后确认预料之外的位置是否有错误。
- (b) 确认结束后，通过继续模式执行。

【注】 在停止模式中发生错误并且停止执行的状态下，如果将模式改为继续模式后重新开始仿真，就可能无法正确地进行仿真。要重新开始仿真时，必须将寄存器和存储器的内容恢复到发生错误前的状态，然后重新执行。

(3) 跟踪缓冲器满引起的暂停

通过 [Trace Acquisition]（跟踪获取）对话框的 [Trace Buffer Full Handling]（跟踪缓冲器满时的处理）指定 [Stop]（停止），如果在指令执行过程中跟踪缓冲器满，仿真 / 调试程序就中止指令的执行。在中止时，[Output]（输出）窗口显示以下信息：

Trace Buffer Full

(4) 执行 WAIT 指令引起的暂停

如果在执行指令时执行 WAIT 指令，仿真 / 调试程序就中止指令的执行。在中止时，[Output]（输出）窗口显示以下信息：

WAIT Instruction

【注】 要重新开始执行时，必须将 PC 值改为重新开始位置的指令地址。

(5) [STOP]（停止）按钮引起的暂停

用户能在指令执行过程中能强制中止指令的执行。在中止时，状态栏显示以下信息：

Stop

能通过 Go、Step 命令重新开始执行。

2.12 浮点数据

因为能将浮点数指定为实数数据，所以用数值等处理浮点时的操作就很容易。能指定浮点的项目如下：

- 在 [Select Break Type]（选择暂停类型）对话框中将暂停类型指定为 [Break Data]（数据暂停）或者 [Break Register]（寄存器暂停）时的数据
- [Memory]（存储器）窗口中的数据
- [Fill Memory]（填充存储器）对话框中的数据
- [Search Memory]（检索存储器）对话框中的数据
- 寄存器值编辑对话框中的输入值

浮点数据格式符合 ANSI C 的浮点数据格式。

对于在从浮点数的 10 进制数转换到 2 进制数时发生的舍入模式，仿真 / 调试程序使用 RN（舍入到最近值）。

另外，如果在从 10 进制数转换到 2 进制数或者从 2 进制数转换到 10 进制数时指定非规范化数，就以非规范化数进行处理。如果在从 10 进制数转换到 2 进制数时发生上溢，就设置为无穷大。

2.13 函数调用履历的显示

在中止仿真时，[Stack Trace]（堆栈跟踪）窗口显示函数的调用履历，因此能确认程序的运行流程。另外，通过在 [Stack Trace]（堆栈跟踪）窗口中选择函数名，[Editor]（编辑）窗口显示相应的源程序，因此除了中止的函数外，还能检查调用该函数的原函数。

在以下情况下，更新函数的调用履历。

- 根据“2.11 暂停条件”所示的条件中止了仿真时
- 在上述中止的状态下更改了寄存器的值时
- 正在单步执行仿真时

详细内容请参照《High-performance Embedded Workshop User's Manual》。

2.14 性能测量

为了测量用户程序的性能，仿真 / 调试程序提供剖析功能和性能分析功能。

2.14.1 剖析功能

剖析功能显示用户程序的全部函数以及全局变量的地址、容量、函数的调用次数和剖析数据。剖析数据因 CPU 而不同。

以列表格式、树的格式和图表格式显示剖析信息。

能通过剖析信息进行优化的探讨，如将容量小并且调用次数多的函数作为内嵌函数处理等。

详细内容请参照“3.8 剖析信息的查看”。

2.14.2 性能分析

性能分析显示用户程序内指定函数的执行周期数和调用次数。由于只取指定函数的性能数据，所以能通过剖析进行高速的仿真。详细内容请参照“3.9 性能分析”。

2.15 伪中断

仿真 / 调试程序能在仿真过程中产生伪中断。产生伪中断的方法有以下 2 种：

(1) 由暂停条件成立时的运行产生的伪中断

能通过暂停命令将暂停条件成立时的运行指定为 [Interrupt]（中断）来产生伪中断。
详细内容请参照“3.6 仿真 / 调试程序断点的使用”。

(2) 通过窗口产生的伪中断

能通过单击 [Trigger]（触发）窗口或者 [GUI I/O]（GUI 输入 / 输出）窗口的按钮来产生伪中断。
详细内容请参照“3.11 伪中断的手动产生”。

如果在发生伪中断到接受中断之间发生下一个伪中断，就只接受优先级高的中断。

(3) 发生伪中断时的暂停

能选择在发生伪中断时是否暂停。

请通过 [Simulator System]（仿真系统）对话框或者 EXEC_STOP_SET 命令进行设置。

【注】 给伪中断指定中断向量号和中断优先级。中断优先级能指定 0 ~ 8 或者 0 ~ H'10。如果在 0 ~ 8 的情况下指定 8 或者在 0 ~ H'10 的情况下指定 H'10，就作为高速中断处理。

2.16 覆盖率

对于用户指定的测量范围，仿真 / 调试程序能在指令执行过程中收集指令覆盖率信息。

除了通过指定直接地址来设置测量范围以外，还能通过指定源文件名来设置该文件包含的全部函数。

能通过利用指令覆盖率信息来观察各指令的执行状态，并能容易地确定程序的哪个部分是否还没有执行。

[Coverage]（覆盖率）窗口显示收集的指令覆盖率信息。

指令覆盖率信息在 [Editor]（编辑）窗口中强调显示与指令执行完的源行对应的列。

另外，对于测量对象的地址范围或者函数，用百分率格式显示覆盖率统计信息，因此能定量掌握程序的执行量。

将指令覆盖率信息保存到文件或者从文件装入指令覆盖率信息。能装入的文件只限于“.COV”文件。

详细内容请参照“3.10 代码覆盖率的测量”。

3. 调试

本章说明仿真 / 调试程序特有的操作以及相关的窗口和对话框。

有关下述 HEW 的共同功能，请参照 HEW 的帮助。

- 调试的准备
- 程序的显示
- 存储器内容的参照和设置
- 存储器内容的显示（波形格式）
- 存储器内容的显示（图像格式）
- 任意地址的存储器内容的参照和设置
- I/O寄存器的参照和设置
- 寄存器的参照和设置
- 程序的执行、停止和复位
- 当前状态的参照
- 多个调试平台的同步
- 命令行界面的调试
- Elf / Dwarf2的支持
- 标签的参照和设置

3.1 工作空间的创建

为了使用仿真 / 调试程序，首先需要通过 HEW 创建工作空间。在此，只进行仿真 / 调试程序特有的说明。创建工作空间的详细内容请参照 HEW 的用户手册。

3.1.1 目标的选择

启动 HEW，在新建工程工作空间时显示以下的对话框。
在此，请选择仿真 / 调试程序的目标。

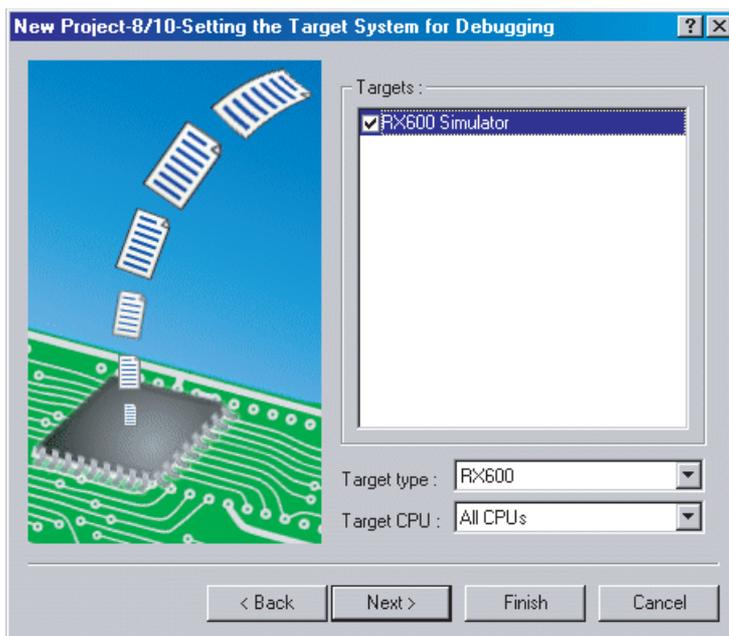


图 3.1 调试程序的目标设置画面（8/10）

- | | |
|----------------------|---|
| [Targets]（目标） | 设置调试程序的目标。必须选择（选定）调试程序的目标，但是可以不选择或者选择多个调试程序的目标。 |
| [Target type]（目标类型） | 指定 [Targets]（目标）显示的目标类型。 |
| [Target CPU]（目标 CPU） | 指定 [Targets]（目标）显示的 CPU 类型。 |

3.1.2 仿真程序工作空间的设置

在 [New Project-9/10]（新工程 -9/10）中，能设置仿真程序的工作空间。

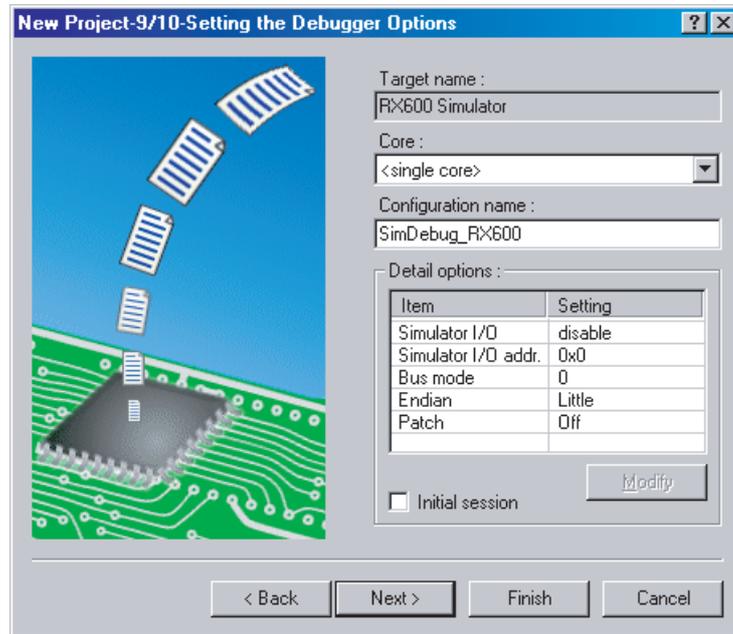


图 3.2 调试程序的选项设置画面（9/10）

- [Detail options]（详细选项） 设置调试程序的目标选项。要更改时，请选择 [Item]（项目）后单击 [Modify]（修改）。对于不能更改的项目，即使选择 [Item]（项目），[Modify]（修改）也是灰色。
- [Simulator I/O] 从用户程序进行标准输入 / 输出或者文件输入 / 输出的仿真 I/O 为 [Enable]（允许）或者 [Disable]（禁止）。
- [Simulator I/O addr.] 上述仿真 I/O 的起始地址
- [Bus mode] 现在不使用。
- [Endian] 显示 CPU 的字节序。
- [Patch] 允许或者禁止中断优先级、MVTIPL 指令
 Off 中断优先级为 0 ~ 15，允许 MVTIPL 指令。
 RX610 中断优先级为 0 ~ 7，禁止 MVTIPL 指令。

有关 [Detail options]（详细选项）以外的项目，请参照《High-performance Embedded Workshop User's Manual》。

3.2 仿真 / 调试程序的启动

通过将仿真 / 调试程序的使用设置切换到预先注册的会话文件，能连接仿真 / 调试程序。如果在创建工程时选择了目标，就创建与所选目标相同个数的会话文件。请从图 3.3 所示的工具栏的下拉式列表中选择与连接的目标对应的会话文件。



图 3.3 会话文件的选择

当选择了仿真 / 调试程序注册的会话文件并且仿真 / 调试程序处于断开连接的状态时，请选择 [Debug -> Connect]（调试 -> 连接）或者单击连接工具栏按钮 。

要断开仿真 / 调试程序的连接时，请选择 [Debug -> Disconnect]（调试 -> 断开连接）或者单击连接断开工具栏按钮 。

3.3 仿真 / 调试程序的设置更改

本节说明在启动仿真 / 调试程序后更改仿真程序的设置方法。

3.3.1 CPU 字节序和工作频率的设置

在启动仿真 / 调试程序时显示的 [Set Simulator]（设置仿真程序）对话框的 [CPU Configuration]（CPU 结构）选项卡中，设置 CPU 字节序和工作频率。

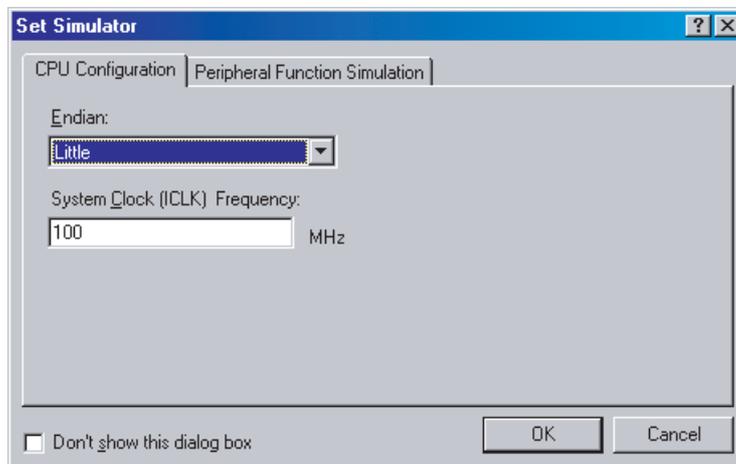


图 3.4 [Set Simulator]（设置仿真程序）对话框（[CPU Configuration]（CPU 结构）选项卡）

在此对话框中设置以下项目：

[Endian]（字节序）

设置 CPU 的字节序。

[Big] 大端法

[Little] 小端法

[System Clock (ICLK) Frequency]（系统时钟（ICLK）频率）

设置 CPU 的工作频率（单位：MHz）。

指定范围：1 ~ 1000

如果选定 [Don't show this dialog box]（不显示此对话框）的复选按钮，就在下次启动仿真 / 调试程序时不显示此对话框。

3.3.2 仿真系统的设置更改

在 [Simulator System]（仿真系统）对话框的 [System]（系统）选项卡中，更改仿真 I/O 的起始位置和执行模式等的设置。

要打开 [Simulator System]（仿真系统）对话框的 [System]（系统）选项卡时，请选择 [Setup -> Simulator -> System...]（设置 -> 仿真程序 -> 系统 ...）或者单击 [Simulator System]（仿真系统）工具栏按钮 。

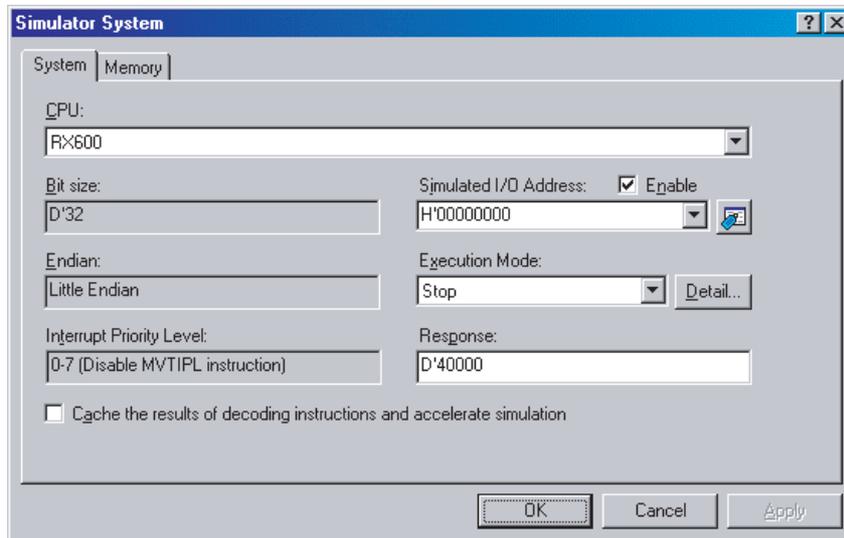


图 3.5 [Simulator System]（仿真系统）对话框（[System]（系统）选项卡）

在此对话框中设置或者显示以下项目：

[CPU]	当前设置的 CPU
[Bit Size]（位的长度）	CPU 的地址空间容量（位数）
[Endian]（字节序）	CPU 的字节序
[Priority Level of Interrupts]（中断优先级）	允许或者禁止显示中断优先级、MVTIPL 指令。 0-7（禁止 MVTIPL 指令） 中断优先级为 0～7。 0-15（允许 MVTIPL 指令） 中断优先级为 0～15。
[Simulated I/O Address]（仿真 I/O 地址）	指定从用户程序进行标准输入 / 输出或者文件输入 / 输出的仿真 I/O 的起始位置。 [Enable]（允许） 选定后就允许仿真 I/O。
[Response]（响应）	指定是按几条指令刷新窗口。 (1～D'2147483647，默认值为 D'40000。)
[Execution Mode]（执行模式）	规定发生仿真错误时的仿真 / 调试程序的运行。 中断属于仿真错误，服从此设置。 另外，能通过 [Detail...]（详细信息 ...）按钮规定发生中断时的运行。 [Stop]（停止） 停止仿真。 [Continue]（继续） 继续仿真。
[Cache the results of decoding instructions and accelerate simulation]（将指令译码结果进行高速缓存，提高仿真速度）	在执行指令时保持译码结果并且 + 设置在执行同一地址时利用译码结果的功能是否有效。 一旦选定，就允许指令译码高速缓存功能，提高仿真速度。

通过单击 [OK] 按钮或者 [Apply]（应用）按钮，设置更改的内容。如果单击 [Cancel]（取消）按钮，就不设置而关闭对话框。

【注】 因为指令译码高速缓存功能再次利用译码结果，所以不能用于使用自改写代码的程序。如果因预料之外的程序运行而改写指令，就可能无法正确地检测错误。

3.3.3 存储器映像和存储器资源的设置更改

在 [Simulator System]（仿真系统）对话框的 [Memory]（存储器）选项卡中，进行存储器映像和存储器资源的设置和更改。

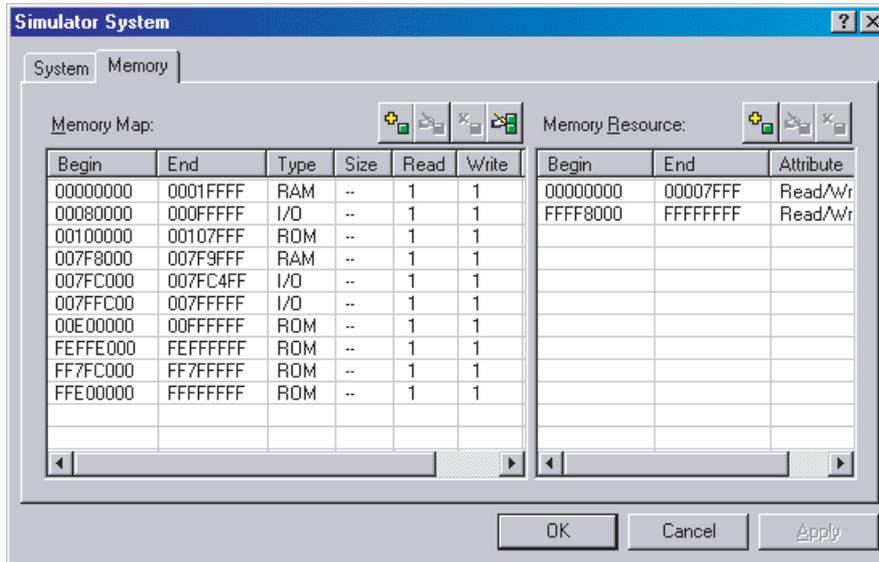


图 3.6 [Simulator System]（仿真系统）对话框（[Memory]（存储器）选项卡）

在此对话框中设置以下项目：

[Memory Map]（存储器映像） 作为存储器信息，显示存储器类型、起始地址、结束地址、数据总线宽度和存取状态数。

[Memory Resource]（存储器资源） 显示当前设置的存储器资源的存取类型、起始地址和结束地址。

通过以下各按钮，能添加、更改和删除 [Memory Resource]（存储器资源）。



添加 [Memory Resource]（存储器资源）的项目。单击此按钮，能打开 [Set Memory Resource]（设置存储器资源）对话框进行添加。



更改 [Memory Resource]（存储器资源）的项目。在列表框上选择要更改的项目后单击按钮，能打开 [Set Memory Resource]（设置存储器资源）对话框进行更改。



删除 [Memory Resource]（存储器资源）的项目。在列表框上选择要删除的项目后单击按钮。

通过以下各按钮，能添加、更改和删除 [Memory Map]（存储器映像）。



添加 [Memory Map]（存储器映像）的项目。单击此按钮，能打开 [Set Memory Map]（设置存储器映像）对话框（参照图 3.7）进行添加。



更改 [Memory Map]（存储器映像）的项目。在列表框上选择要更改的项目后单击按钮，能打开 [Set Memory Map]（设置存储器映像）对话框（参照图 3.7）进行更改。



删除 [Memory Map]（存储器映像）的项目。在列表框上选择要删除的项目后单击按钮。

能通过按钮  将 [Memory Map]（存储器映像）复位到默认值。

通过单击 [OK] 按钮或者 [Apply]（应用）按钮，设置更改的内容。如果单击 [Cancel]（取消）按钮，就不设置而关闭对话框。

如果有优化链接编辑程序输出的链接清单文件（.map），就能自动确保基于存储器映像和链接映像信息的存储器资源。

详细内容请参照《High-performance Embedded Workshop User's Manual》的“Automatically Allocating the Memory Resource”。

3.3.4 存储器映像的设置对话框

在 [Set Memory Map]（设置存储器映像）对话框中，设置对象 CPU 的存储器映像。
各项目显示的内容因对象 CPU 而不同，仿真 / 调试程序将这些值用于存储器存取的仿真。

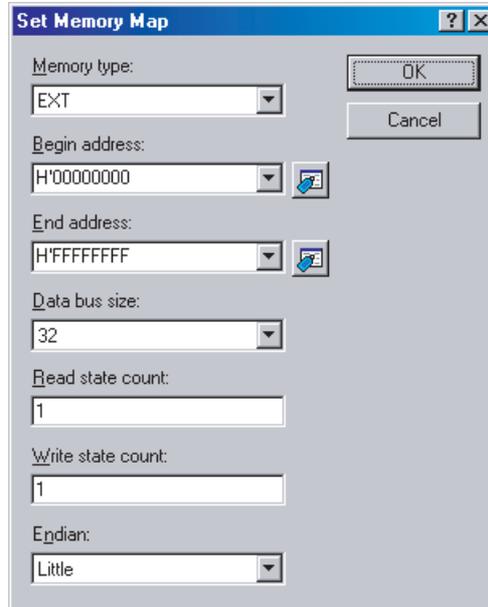


图 3.7 [Set Memory Map]（设置存储器映像）对话框

在此对话框中设置以下项目：

[Memory type]（存储器类型）	存储器类型
	[ROM] 内部 ROM
	[RAM] 内部 RAM
	[EXT] 外部存储器
	[IO] 内部 I/O
[Begin address]（起始地址）	对应存储器类型的存储器的起始地址
[End address]（结束地址）	对应存储器类型的存储器的结束地址
[Data bus size]（数据总线长度）	存储器的数据总线宽度
[Read state count]（读状态数）	存储器的读存取状态数
[Write state count]（写状态数）	存储器的写存取状态数
[Endian]（字节序）	对象存储区的字节序

通过单击 [OK] 按钮，设置更改的内容。如果单击 [Cancel]（取消）按钮，就不设置而关闭对话框。

【注】

- (1) 不能删除和更改已确存储器资源的区域的存储器映像。请先通过 [Simulator System]（仿真系统）对话框的 [Memory]（存储器）选项卡删除存储器资源，然后删除或者更改存储器映像。
- (2) 对于外部存储器以外的存储器类型，不能显示和更改数据总线长度。
- (3) 数据总线长度、读状态数和写状态数不影响指令的仿真。存储器的存取状态数总是为“1”。
- (4) 只能以 16 字节为边界设置存储器映像。对于 16 字节边界以外的设置，补正为包括所设存储器映像在内的 16 字节边界。
- (5) 不能显示和更改内部 I/O 的字节序。
- (6) 不能在此对话框中而能在 [Set Simulator]（设置仿真程序）对话框中，更改内部 ROM 和内部 RAM 的字节序。有关 [Set Simulator]（设置仿真程序）对话框的详细内容，请参照“3.3.1 CPU 字节序和工作频率的设置”。

3.3.5 存储器资源的设置对话框

在 [Set Memory Resource]（设置存储器资源）对话框中，设置和更改存储器资源。

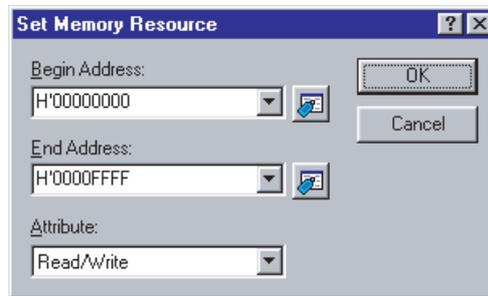


图 3.8 [Set Memory Resource]（设置存储器资源）对话框

在此对话框中设置以下项目：

[Begin Address]（起始地址）	确保的存储区的起始地址
[End Address]（结束地址）	确保的存储区的结束地址
[Attribute]（属性）	存取类型
[Read]	只读
[Write]	只写
[Read/Write]	可读写

在指定各项目后，通过单击 [OK] 按钮，设置和更改存储器资源。如果单击 [Cancel]（取消）按钮，就不设置而关闭对话框。

【注】

- (1) 一旦设置存储器资源，就能使用 PC 的存储器。因此，如果获取过多的存储器资源，PC 的运行就可能非常慢。
- (2) 只能以 16 字节为边界设置存储器资源。对于 16 字节边界以外的设置，补正为包括所设存储器资源在内的 16 字节边界。另外，存取类型也为 16 字节边界。
当以少于 16 字节边界使用时，必须在硬件手册规定的范围内使用存储器。
- (3) 在通过指令写只读存储器或者读只写存储器时，发生存储器的存取错误。

3.4 外围功能仿真的设置

仿真 / 调试程序通过 DLL 格式的模块实现外围功能的仿真。

在此说明用于使外围功能仿真有效的外围功能仿真模块的注册方法以及外围功能仿真结构的设置方法。

3.4.1 外围功能仿真模块的注册

在 [Set Simulator]（设置仿真程序）对话框的 [Peripheral Function Simulation]（外围功能仿真）选项卡中，注册外围功能的仿真模块。在启动仿真 / 调试程序时显示 [Set Simulator]（设置仿真程序）对话框。

如果在此对话框中注册外围功能的仿真模块，就能利用该外围功能仿真模块提供的外围功能仿真。在启动仿真 / 调试程序后，不能更改外围功能仿真模块的注册内容。要更改利用的外围功能仿真模块时，请重新启动仿真 / 调试程序来显示此对话框。

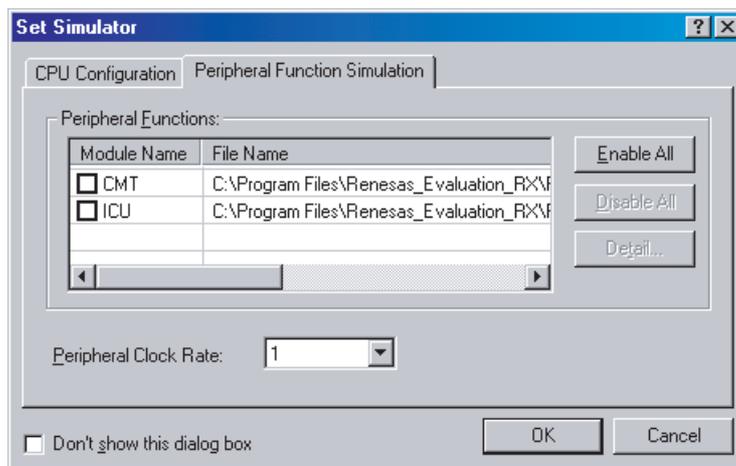


图 3.9 [Set Simulator]（设置仿真程序）对话框
（[Peripheral Function Simulation]（外围功能仿真）选项卡）

在此对话框中设置以下项目：

[Peripheral Functions]（外围功能）

显示外围功能仿真模块的信息。

[Module Name] 要仿真的外围功能名

[File Name] 外围功能仿真模块的文件名

注册并能利用选定了 [Module name] 栏的复选框的外围功能仿真模块。

[Enable All]（允许全部）

允许全部的外围功能仿真模块。

[Disable All]（禁止全部）

禁止全部的外围功能仿真模块。

[Detail...]（详细信息...）

显示 [Peripheral Module Configuration]（外围模块结构）对话框，以便显示外围功能信息、更改外围功能的起始地址和中断源信息。

[Peripheral Clock Rate]（外围时钟比）

指定外围时钟和系统时钟的比（1 个外围时钟相当于几个系统时钟）。

（选择 1、2、3、4、6、8、12、16、24、32。）

在指定各项目后，通过单击 [OK] 按钮，设置和更改外围功能的仿真。如果单击 [Cancel]（取消）按钮，就不设置而关闭对话框。

如果选定 [Don't show this dialog box]（不显示此对话框）的复选按钮，就在下次启动仿真 / 调试程序时不显示此对话框。

3.4.2 外围功能的地址更改

在 [Peripheral Module Configuration]（外围模块结构）对话框中，更改外围功能的地址。在 [Peripheral Module Configuration]（外围模块结构）对话框的 [Address]（地址）选项卡中，更改有中断源信息的外围功能的地址。

要打开 [Peripheral Module Configuration]（外围模块结构）对话框时，请在 [Set Simulator]（设置仿真程序）对话框的 [Peripheral Function Simulation]（外围功能仿真）选项卡的 [Peripheral Functions]（外围功能）栏中，选择外围功能后单击 [Detail...]（详细信息 ...）按钮。

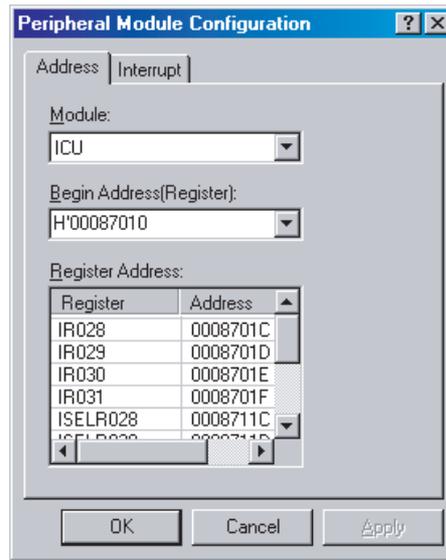


图 3.10 [Peripheral Module Configuration]（外围模块结构）对话框
（[Address]（地址）选项卡）

在此对话框中显示和设置以下项目：

- | | |
|---------------------------|---|
| [Module]（模块） | 所选外围功能仿真模块支持的外围功能名 |
| [Start Address]（起始地址） | [Module]（模块）选择的外围功能的起始地址 |
| [Register Address]（寄存器地址） | 显示 [Module]（模块）指定的外围功能的寄存器名和寄存器地址。不能更改各寄存器地址。 |

在指定各项目后，通过单击 [OK] 按钮，设置外围功能的地址。如果单击 [Cancel]（取消）按钮，就不设置而关闭对话框。

3.4.3 外围功能中断源信息的更改

在 [Peripheral Module Configuration]（外围模块结构）对话框的 [Interrupt]（中断）选项卡中，能参照外围功能的中断源信息。

要打开 [Peripheral Module Configuration]（外围模块结构）对话框时，请在 [Set Simulator]（设置仿真程序）对话框的 [Peripheral Function Simulation]（外围功能仿真）选项卡的 [Peripheral Functions]（外围功能）栏中，选择外围功能后单击 [Detail...]（详细信息 ...）按钮。

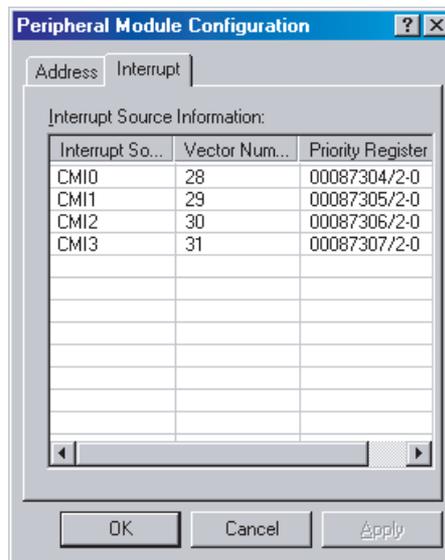


图 3.11 [Peripheral Module Configuration]（外围模块结构）对话框
（[Interrupt]（中断）选项卡）

在此对话框中显示以下项目：

中断源信息	[Interrupt Source]	外围功能支持的中断源名
	[Vector Number]	中断向量号
	[Priority Register Address/ Bit Field Position]	中断优先级寄存器地址和寄存器内的位的位置

要更改中断源信息时，请双击要更改的中断源。显示 [Set Interrupt Source Information]（设置中断源信息）对话框。

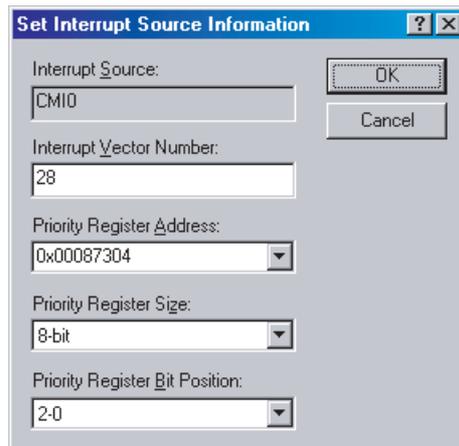


图 3.12 [Set Interrupt Source Information]（设置中断源信息）对话框

在此对话框中显示和设置以下项目：

[Interrupt Source]（中断源）	中断源名
[Interrupt Vector Number]	中断向量号 （省略前缀时为 10 进制输入和 10 进制显示）
[Priority Register Address]	中断优先级寄存器的地址
[Priority Register Size]	中断优先级寄存器的长度
[Priority Register Bit Position]	中断优先级寄存器位的位置

在指定各项目后，通过单击 [OK] 按钮，设置中断源信息。如果单击 [Cancel]（取消）按钮，就不设置而关闭对话框。

3.4.4 控制寄存器的存储器资源

外围功能仿真模块确保控制寄存器区域的存储器资源。不能删除和更改已确保的控制寄存器的存储器资源。有关存储器资源设置的详细内容，请参照“3.3.3 存储器映像和存储器资源的设置更改”。

3.4.5 被连接的外围功能的确认

在启动仿真 / 调试程序后，在 [Status]（状态）窗口的 [Platform]（平台）表的 [Peripheral Modules]（外围模块）项目中显示被连接的外围功能名。

3.4.6 虚拟端口的文件输入 / 输出

仿真 / 调试程序将部分引脚作为虚拟端口分配到存储器，这些虚拟端口支持文件的输入 / 输出。有关本仿真 / 调试程序支持的虚拟端口，请参照“2.8.2(3) 数据的输入 / 输出”。

(1) 被设置的文件输入 / 输出一览表的显示

要显示当前设置的文件输入 / 输出一览表时，请打开 [Simulator System]（仿真系统）对话框的 [Port I/O]（端口输入 / 输出）选项卡。

要打开 [Simulator System]（仿真系统）对话框的 [Port I/O]（端口输入 / 输出）选项卡时，请选择 [Setup -> Simulator -> System...]（设置 -> 仿真程序 -> 系统 ...），然后选择 [Simulator System]（仿真系统）对话框的 [Port I/O]（端口输入 / 输出）选项卡。

如果有虚拟端口的模块尚未被注册，就不显示 [Port I/O]（端口输入 / 输出）选项卡。

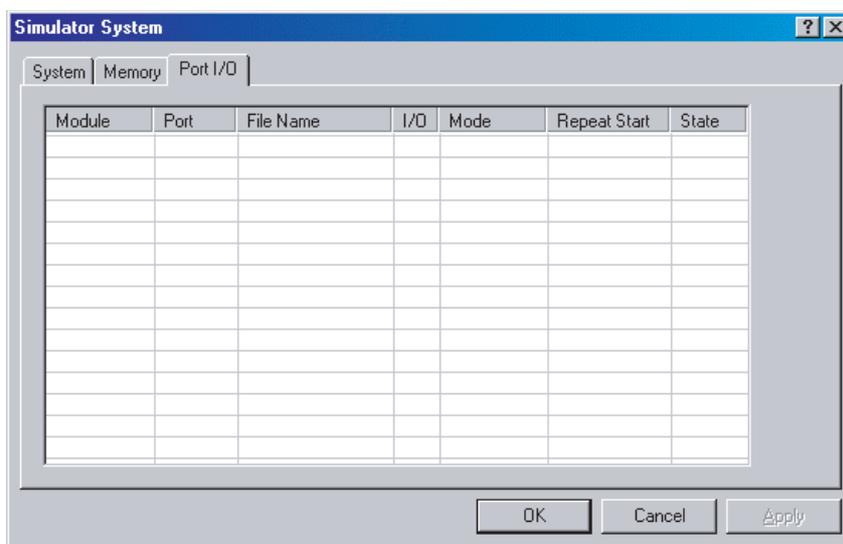


图 3.13 [Simulator System]（仿真系统）对话框（[Port I/O]（端口输入 / 输出）选项卡）

显示的项目如下：

[Module]（模块）	显示模块名。
[Port]（端口）	显示端口名。
[File Name]（文件名）	显示文件名。
[I/O]（输入 / 输出）	显示文件的输入 / 输出。 [In]（输入）：文件输入 [Out]（输出）：文件输出
[Mode]（模式）	显示文件的数据输入 / 输出模式。 [Repeat]（重复）：重复输入 [Once]（1 次）：只输入 1 次 [Overwrite]（盖写）：盖写输出 [Append]（追加）：追加输出
[Repeat Start]（重复开始）	显示重复输入模式中的重复开始号。
[State]（状态）	显示文件的打开 / 关闭状态。 [Open]（打开）：文件打开状态 [Close]（关闭）：文件关闭状态

(2) 文件的添加

在 [Port I/O]（端口输入 / 输出）选项卡上单击鼠标的右键后，从弹出式菜单中选择 [Add]（添加）或者双击列表项目。打开 [Set Port I/O]（设置端口输入 / 输出）对话框。

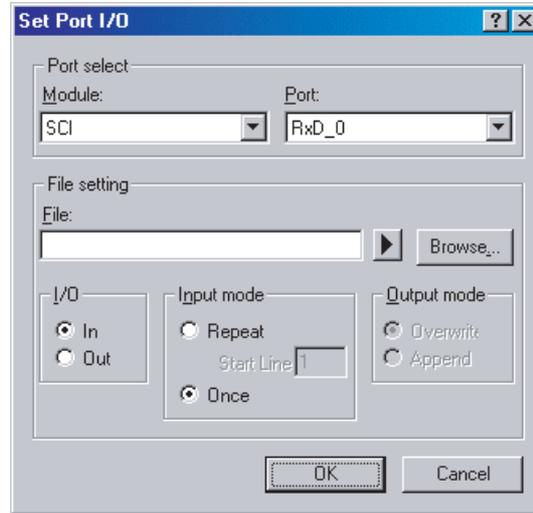


图 3.14 [Set Port I/O]（设置端口输入 / 输出）对话框

设置的项目如下：

[Port select] (选择端口)	[Module] (模块) [Port] (端口)	从列表中选择有输入 / 输出对象端口的模块名。 从列表中选择输入 / 输出的对象端口名。
[File setting] (文件设置)	[File] (文件)	指定输入 / 输出的文件名。 如果省略文件扩展名，就附加 “.csv”。
	[I/O] (输入 / 输出)	[Input] (输入) 输入文件。 [Output] (输出) 输出文件。
	[Input mode] (输入模式)	[Repeat] (重复) 当到达输入文件的最后位置时，返回到起始位置重复输入。 [Start Line] (开始行) 指定在重复输入模式中开始重复的行号。(1 ~ 65535)
		[Once] (1 次) 当到达输入文件的最后位置时，结束文件的输入。
	[Output mode] (输出模式)	[Overwrite] (盖写) 当输出文件已经存在时，进行盖写。 [Append] (追加) 当输出文件已经存在时，从文件的最后位置开始追加。

能给 1 个端口分配的文件数是输入 / 输出各 1 个文件，能将同一个文件指定给多个输入端口。

(3) 文件的打开

在 [Port I/O]（端口输入 / 输出）选项卡上选择要打开的文件的行后，如果从弹出式菜单中选择 [Open]（打开），就打开所选择的文件。

(4) 全部文件的打开

如果在 [Port I/O]（端口输入 / 输出）选项卡上从弹出式菜单中选择 [Open All]（全部打开），就打开全部文件。

(5) 关闭文件

在 [Port I/O]（端口输入 / 输出）选项卡上选择要关闭的文件的行后，如果从弹出式菜单中选择 [Close]（关闭），就关闭所选择的文件。

(6) 关闭全部文件

如果在 [Port I/O]（端口输入 / 输出）选项卡上从弹出式菜单中选择 [Close All]（全部关闭），就关闭全部文件。

(7) 文件指定的编辑

在 [Port I/O]（端口输入 / 输出）选项卡上选择要编辑的文件的行后，如果从弹出式菜单中选择 [Edit]（编辑）或者双击，就打开 [Set Port I/O]（设置端口输入 / 输出）对话框，能编辑文件的指定。

(8) 文件的删除

在 [Port I/O]（端口输入 / 输出）选项卡上选择要删除的文件的行后，如果从弹出式菜单中选择 [Delete]（删除），就删除所选择的文件。

(9) 虚拟端口的文件格式

虚拟端口文件为 CSV 格式。

输入文件的格式如下所示：

```
< 时间 >, < 数据 >
:
```

输入文件指定要输入的时间和数据。时间是用皮秒（整数值）指定与前一个数据的差分，必须指定大于等于 1 的值。必须用 16 进制的整数值指定数据。

输出文件的格式如下所示：

```
[Module]
< 模块名 >
[Port]
< 端口名 >
[Length]
< 数据位长 >
[Data]
< 时间 >, < 数据 >
:
```

在输出文件中输出模块名、端口名、数据位长、时间和数据。时间是用皮秒（整数值）输出的从仿真开始到输出数据为止的时间。

3.5 存储器的操作

3.5.1 窗口显示内容的定期更新

如果从 [Memory]（存储器）窗口的弹出式菜单中选择 [Auto Refresh]（自动更新），仿真 / 调试程序就能在用户程序执行过程中定期地更新 [Memory]（存储器）窗口的显示内容。

更新间隔的默认值以及能指定的更新间隔的范围如下：

更新间隔的默认值：100 毫秒

能指定的更新间隔的范围：10 毫秒～ 10000 毫秒

3.5.2 I/O 区域内容的显示和更改

要在 [Memory]（存储器）窗口中显示和更改 I/O 区域时，必须按照硬件手册记载的存取长度切换显示后进行显示和更改。

如果与硬件手册记载的存取长度不同，就可能无法正确地显示和更改。

3.6 仿真 / 调试程序断点的使用

除了 HEW 的标准 PC 断点以外，仿真 / 调试程序还有更高的断点功能。

能对这些断点设置暂停条件、指定暂停条件成立时的运行以及显示被设置的断点。

3.6.1 断点一览表的显示

要显示当前设置的断点一览表时，请打开 [Eventpoints]（事件点）窗口。

要打开 [Eventpoints]（事件点）窗口时，请选择 [View -> Code -> Eventpoints]（视图 -> 代码 -> 事件点）或者单击 [Eventpoints]（事件点）工具栏按钮 。

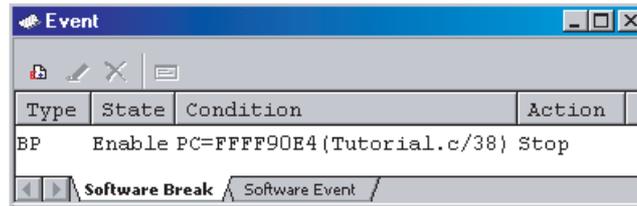


图 3.15 [Event]（事件）窗口

显示的项目如下：

[Type] 显示暂停类型。

- [BP] : PC 暂停
- [BA] : 存取暂停
- [BD] : 数据暂停
- [BR] : 寄存器暂停
- [BS] : 顺序暂停
- [BCY] : 周期暂停

[State] 显示允许 / 禁止断点。

- [Enable] : 允许
- [Disable] : 禁止

[Condition] 显示 Break 成立条件。显示内容因暂停类型而不同。

暂停类型为 BR 时显示寄存器名，为 BCY 时显示周期数。

- BP 时 : PC= 程序计数器（对应的文件名 / 行、符号名）
- BA 时 : Address= 地址（符号名）
- BD 时 : Address= 地址（符号名）
- BR 时 : Register= 寄存器名
- BS 时 : PC= 程序计数器（对应的文件名 / 行、符号名）
- BCY 时 : Cycle= 周期数（16 进制显示）

[Action] 显示暂停条件成立时的运行。

- [Stop] : 停止执行。
- [File Input]（文件名）[文件的状态] : 从文件读存储器数据。
- [File Output]（文件名）[文件的状态] : 将存储器数据写到文件。
- [Interrupt]（中断类型 / 优先级） : 中断处理
- [Trace Trigger] : 开始取跟踪信息。

通过 [Action] 的设置，[Software Break] 选项卡显示指定 [Stop] 的条件，[Software Event] 选项卡显示指定 [Stop] 以外的条件。

3.6.2 断点的设置

如果在 [Eventpoints] (事件点) 窗口的弹出式菜单中选择 [Add...] (添加 ...), 就能打开 [Select Break Type] (选择暂停类型) 对话框进行断点的设置。

从 [Select Break Type] (选择暂停类型) 对话框打开设置暂停条件的 [Set xx Condition] (设置 xx 条件) 对话框以及设置在暂停成立时运行的 [Set xx Action] (设置 xx 运行) 对话框。作为 [Action type] (运行类型), 当指定 [Stop] (停止) 时, 在 [Software Break] 选项卡中选择弹出式菜单; 当不指定 [Stop] (停止) 时, 在 [Software Event] 选项卡中选择弹出式菜单。

(1) 暂停类型的选择

如果在 [Eventpoints] (事件点) 窗口的弹出式菜单中选择 [Add...] (添加 ...), 就打开 [Select Break Type] (选择暂停类型) 对话框。

在 [Select Break Type] (选择暂停类型) 对话框的 [Break Type] (暂停类型) 域中选择暂停类型。



图 3.16 [Select Break Type] (选择暂停类型) 对话框

选择的内容如下:

[Break type] (暂停类型)	内容
[PC Breakpoint] (PC 断点)	指令的执行位置引起的断点
[Break Access] (存取暂停)	存储器的存取范围引起的暂停
[Break Data] (数据暂停)	存储器的数值引起的暂停
[Break Register] (寄存器暂停)	寄存器的数值引起的暂停
[Break Sequence] (顺序暂停)	指定执行顺序的断点
[Break Cycle] (周期暂停)	周期数引起的暂停

(2) 暂停条件的设置

在 [Select Break Type] (选择暂停类型) 对话框中选择暂停类型后, 如果单击 [Detail...] (详细信息), 就显示设置各暂停类型条件的对话框。

(a) PC 断点

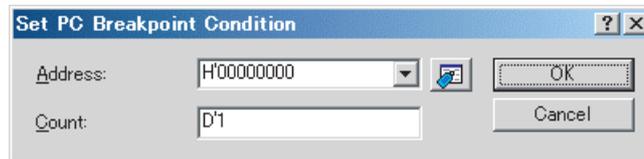


图 3.17 [Set PC Breakpoint Condition] (设置 PC 断点条件) 对话框

设置 PC 断点条件:

[PC Breakpoint] (PC 断点)	最多能指定 1024 个。
[Address] (地址)	暂停的指令位置
[Count] (次数)	指定位置的取指令次数 (省略前缀时为 10 进制输入和 10 进制显示) (1 ~ 16383, 省略时为 1。)

(b) 存取暂停

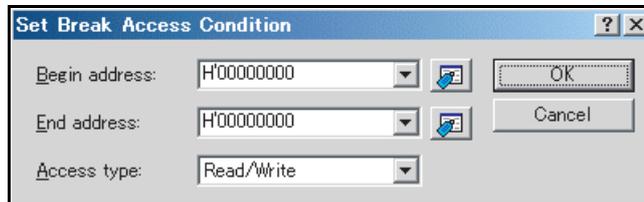


图 3.18 [Set Break Access Condition] (设置存取暂停条件) 对话框

设置存取暂停的条件:

[存取暂停]	最多能指定 1024 个。
[Begin address] (起始地址)	存取和暂停的存储器的起始位置
[End address] (结束地址)	存取和暂停的存储器的结束位置 (省略时只有起始位置。)
[Access type] (存取类型)	存取类型

【注】字符串指令和乘加运算指令只以最后的数据存取为存取暂停的检查对象。

(c) 寄存器暂停

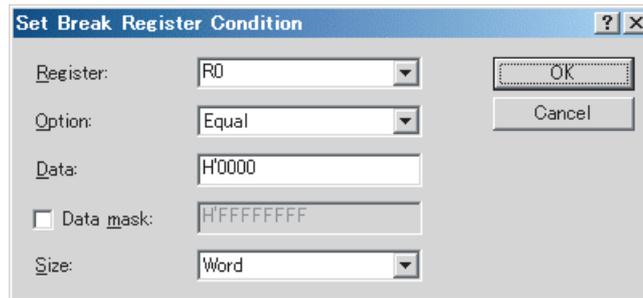


图 3.19 [Set Break Register Condition] (设置寄存器暂停条件) 对话框

设置寄存器暂停的条件:

[寄存器暂停] 最多能指定 1024 个。

[Register] (寄存器)

设置暂停条件的寄存器名

[Option] (选项)

数据相同 / 不相同

[Data] (数据)

暂停条件的数据值

(如果省略, 就在每次写寄存器时暂停。)

[Data mask] (数据屏蔽)

屏蔽条件 (指定“0”的位被屏蔽。)

[Size] (长度)

数据长度

【注】

(1) 字符串操作指令和乘加运算指令只以最后的数据存取为存取暂停的检查对象。

(2) 将堆栈指针寄存器设置为寄存器暂停时的检查对象寄存器如下:

寄存器暂停的指定寄存器	存取寄存器	
	ISP	USP
“R0”	○	○
“ISP”	○	×
“USP”	×	○

○: 检查暂停 ×: 不检查暂停

(d) 顺序暂停

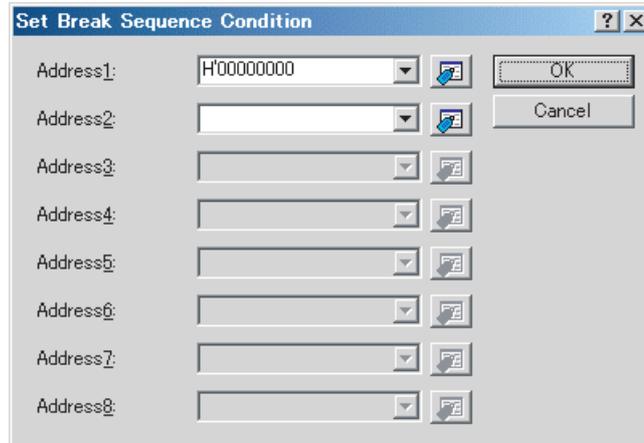


图 3.20 [Set Break Sequence Condition] (设置顺序暂停条件) 对话框

设置顺序暂停的条件:

- [顺序暂停] 只能指定 1 组。
 [Address1] (地址 (1)) ~ [Address8] (地址 (8)) 暂停发生条件的通过地址
 (8 点不需要全部设置。)

(e) 周期暂停

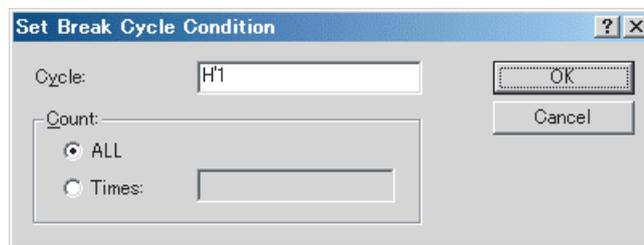


图 3.21 [Set Break Cycle Condition] (设置周期暂停条件) 对话框

设置周期暂停的条件:

- [周期暂停] 最多能指定 1024 个。
 [Cycle] (周期) 判断暂停的周期数 (1 ~ H'FFFFFFF)。
 [周期] × n 的条件相同。
 但是, 指定的周期和实际条件相同的周期可能有偏差。
- [Count] (次数) 暂停成立的次数
 [ALL] (全部) 每当条件相同时暂停成立。
 [Times] (次) 只在条件相同的次数 ≤ [Times]
 (次) 时暂停成立。
 (省略前缀时为 16 进制输入和
 16 进制显示) (1 ~ 65535)

(f) 数据暂停

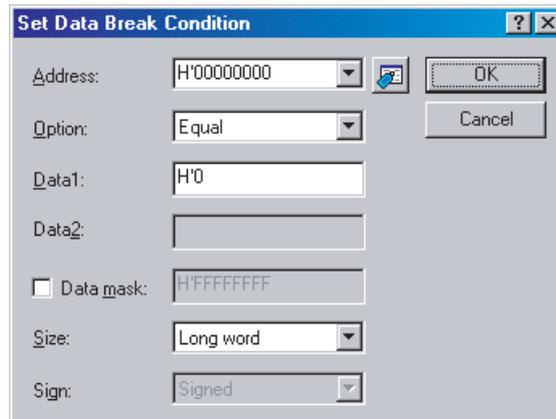


图 3.22 [Set Data Break Condition] (设置数据暂停条件) 对话框

设置数据暂停条件:

[数据暂停] 最多能指定 1024 个。

[Address] (地址) 判断暂停的存储器位置

[Option] (选项) 判断方法

相同 在存储器的数据和指定值 ([Data] (数据)) 相同时成立。

不相同 在存储器的数据和指定值 ([Data] (数据)) 不相同成立。

符号取反 *1 在上次存储器的写入值和本次存储器的写入值的符号取反时成立。

差分 *1 在上次存储器的写入值和本次存储器的写入值的差分超过指定值 ([Data] (数据)) 时成立。

GT(>) 在存储器的写入值大于指定值 ([Data] (数据)) 时成立。

LT(<) 在存储器的写入值小于指定值 ([Data] (数据)) 时成立。

GE(>=) 在存储器的写入值大于等于指定值 ([Data] (数据)) 时成立。

LE(<=) 在存储器的写入值小于等于指定值 ([Data] (数据)) 时成立。

范围内 在存储器的写入值在指定值 ([Data1] (数据 1) 和 [Data2] (数据 2)) 的范围内时成立。

([Data1] (数据 1) ≤ 存储器的写入值 ≤ [Data2] (数据 2))

范围外 在存储器的写入值在指定值 ([Data1] (数据 1) 和 [Data2] (数据 2)) 的范围外时成立。

(存储器的写入值 < [Data1 数据 1] || [Data2] (数据 2) < 存储器的写入值)

[Data 1] (数据 1) 暂停条件的数据值
在数据比较方法为 [范围内] 或者 [范围外] 时, 为暂停条件的开始数据值。

[Data 2] (数据 2) 暂停条件的结束数据值
在数据比较方法为 [范围内] 或者 [范围外] 时有效。

[Data mask] (数据屏蔽) 屏蔽条件 (指定“0”的位被屏蔽)
在 [符号取反] 和 [差分] 除外的数据比较方法选项时有效。

[Size] (长度) 数据长度

[Sign] (符号) 数据的符号

在以下情况下有效:

- 数据比较方法为 [差分] 时

- 数据比较方法为 [GT(>)]、[LT(<)]、[GE(>=)]、[LE(<=)]、[范围内]、[范围外] 并且数据长度为 [Byte]、[Word]、[Long word] 时

*1 : 因为 [符号取反] 和 [差分] 与上次的写入值比较, 所以在复位后或者在暂停成立后第 1 次的判断总是不成立。

【注】字符串指令和乘加运算指令只以最后的数据存取为存取暂停的检查对象。

(3) 运行类型的选择

如果在各暂停条件的设置对话框中设置条件后单击 [OK]，就再次打开 [Select Break Type]（选择暂停类型）对话框。

在 [Select Break Type]（选择暂停类型）对话框的 [Action type]（运行类型）域中选择运行类型。

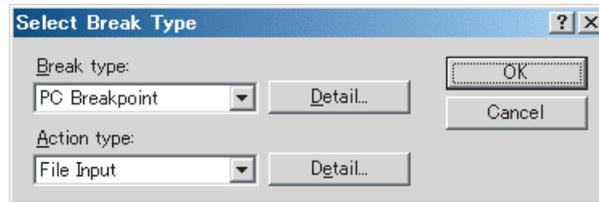


图 3.23 [Select Break Type]（选择暂停类型）对话框

选择的内容如下：

[运行类型]	内容
[Stop]（停止）	在条件成立时停止用户程序的执行。
[File Input]（文件输入）	在条件成立时将从指定的文件读取的数据写到指定的存储器。
[File Output]（文件输出）	在条件成立时将指定的存储器内容写到指定的文件。
[Interrupt]（中断）	在条件成立时进行中断处理。
[Trace Trigger]（跟踪触发）	在条件成立时开始获取跟踪信息。

(4) 运行内容的设置

如果在 [Select Break Type]（选择暂停类型）对话框中选择运行类型后单击 [Detail...]（详细信息），就显示设置各运行类型（[Stop]（停止）和 [Trace Trigger]（跟踪触发）除外）内容的对话框。

(a) 文件输入

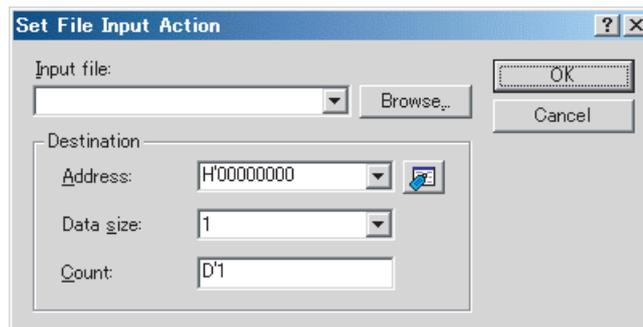


图 3.24 [Set File Input Action]（设置文件输入运行）对话框

设置文件输入运行的内容：

[文件输入]	在条件成立时将从指定的文件读取的数据写到指定的存储器。
[Input file]（输入文件）	指定要读的数据文件。 如果读到文件的结束，就从文件的起始位置重复读取。
[Address]（地址）	指定要写数据的存储器地址。
[Data size]（数据长度）	指定 1 个读数据的长度（字节数）（1/2/4/8）。
[Count]（数据个数）	指定读数据的个数（省略前缀时为 10 进制输入和 10 进制显示）（1 ~ H'FFFFFFFF）。

(b) 文件输出

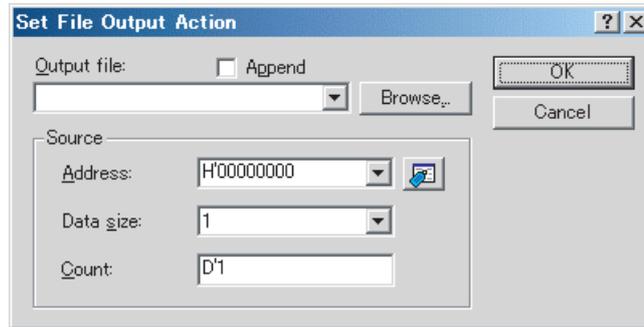


图 3.25 [Set File Output Action] (设置文件输出运行) 对话框

设置文件输出运行的内容:

- | | |
|----------------------|--|
| [文件输出] | 在条件成立时将指定的存储器内容写到指定的文件。 |
| [Output file] (输出文件) | 指定要写数据的文件。 |
| [Append] (追加) | 如果在 [Output file] (输出文件) 中指定了已经存在的文件, 就需要指定是否在文件的最后追加输出。 |
| [Address] (地址) | 指定要读数据的存储器地址。 |
| [Data size] (数据长度) | 指定 1 个写数据的长度 (字节数) (1/2/4/8)。 |
| [Count] (数据个数) | 指定写数据的个数 (省略前缀时为 10 进制输入和 10 进制显示) (1 ~ H'FFFFFFF)。 |

(c) 中断

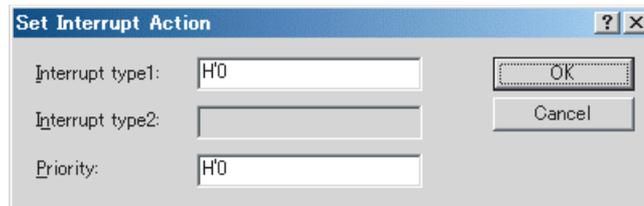


图 3.26 [Set Interrupt Action] (设置中断运行) 对话框

设置中断运行的内容:

- | | |
|-----------------------------|---|
| [中断] | 在条件成立时进行中断处理, 详细内容请参照“2.15 伪中断”。 |
| [Interrupt type 1] (中断类型 1) | 指定中断向量号 (省略前缀时为 16 进制输入和 16 进制显示)。 |
| [Priority] (优先级) | 指定中断优先级 (省略前缀时为 16 进制输入和 16 进制显示) (0 ~ 8 或者 0 ~ H'10)。
如果在 0 ~ 8 时指定 8 或者在 0 ~ H'10 时指定 H'10, 就作为高速中断处理。 |

(d) 注意事项

如果在多个 [File Input] (文件输入) 中指定同一个文件, 就按暂停成立顺序从文件读数据; 如果在多个 [File Output] (文件输出) 中指定同一个文件, 就按暂停成立顺序给文件写数据。但是, 如果在 [File Input] (文件输入) 和 [File Output] (文件输出) 中指定同一个文件, 就只有最初成立的运行有效。

3.6.3 断点设置内容的更改

如果在选择要更改的断点后从弹出式菜单中选择 [Edit...] (编辑 ...), 就打开 [Select Break Type] (选择暂停类型) 对话框, 能更改暂停条件。[Edit...] (编辑 ...) 菜单只在选择 1 个断点时有效。

3.6.4 允许断点

如果在选择断点后从弹出式菜单中选择 [Enable] (允许), 就允许所选择的断点。

3.6.5 禁止断点

如果在选择断点后从弹出式菜单中选择 [Disable] (禁止), 就禁止所选择的断点。在允许设置后, 尽管断点还留在列表中, 但是即使指定的条件相同, 暂停也不成立。

3.6.6 断点的删除

如果在选择断点后从弹出式菜单中选择 [Delete] (删除), 就删除所选择的断点。不想删除断点而在保持详细信息的情况下即使条件相同也不使暂停成立时, 使用 [Disable] (禁止) 选项 (参照 “3.6.5 禁止断点”)。

3.6.7 断点的全部删除

如果从弹出式菜单中选择 [Delete All] (全部删除), 就删除全部断点。

3.6.8 断点的源行显示

如果在选择断点后从弹出式菜单中选择 [Go to Source] (转至源), 就打开有断点的 [Source] (源) 或者 [Disassembly] (反汇编) 窗口。[Go to Source] (转至源) 菜单只在选择 1 个断点时有效。

3.6.9 输入 / 输出文件的关闭

如果在选择断点后从弹出式菜单中选择 [Close File] (关闭文件), 就关闭所选择的 [File Input] (文件输入) 或者 [File Output] (文件输出) 的数据文件, 对文件的读位置进行复位。

3.6.10 输入 / 输出文件的全部关闭

如果从弹出式菜单中选择 [Close All Files] (关闭全部文件), 就关闭全部 [File Input] (文件输入) 和 [File Output] (文件输出) 的数据文件, 对文件的读位置进行复位。

3.7 跟踪信息的查看

作为跟踪信息，仿真 / 调试程序能获取和显示指令的执行结果。

[Trace]（跟踪）窗口显示跟踪信息。在 [Trace Acquisition]（跟踪获取）对话框中设置跟踪信息的获取条件。

3.7.1 跟踪窗口的打开

要打开 [Trace]（跟踪）窗口时，请选择 [View -> Code -> Trace]（视图 -> 代码 -> 跟踪）或者单击 [Trace]（跟踪）工具栏按钮 。

3.7.2 跟踪信息获取条件的设置

在打开 [Trace]（跟踪）窗口后，设置跟踪信息的获取条件。

在 [Trace Acquisition]（跟踪获取）对话框中设置跟踪信息的获取条件。

要打开 [Trace Acquisition]（跟踪获取）对话框时，请从弹出式菜单中选择 [Acquisition...]（获取 ...）。

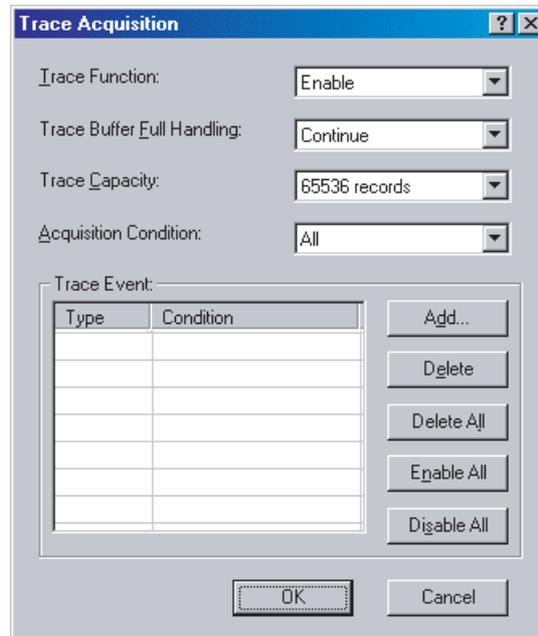


图 3.27 [Trace Acquisition]（跟踪获取）对话框

在此对话框中设置跟踪信息的获取条件。

[Trace Function]（跟踪功能）

- | | |
|---------------|-----------|
| [Disable]（禁止） | 停止获取跟踪信息。 |
| [Enable]（允许） | 开始获取跟踪信息。 |

[Trace Buffer Full Handling]（跟踪缓冲器满时的处理）

- | | |
|----------------|--------------------|
| [Continue]（继续） | 即使跟踪信息获取缓冲器满也继续获取。 |
| [Break]（停止） | 在跟踪信息获取缓冲器满时停止执行。 |

[Trace Capacity]（跟踪容量）

- | | |
|-------------|--------------------|
| [65536] 记录 | 跟踪缓冲器的容量为 64K 记录。 |
| [131072] 记录 | 跟踪缓冲器的容量为 128K 记录。 |
| [262144] 记录 | 跟踪缓冲器的容量为 256K 记录。 |

[524288] 记录 跟踪缓冲器的容量为 512K 记录。

[1048576] 记录 跟踪缓冲器的容量为 1M 记录。

[Acquisition Condition] (获取条件)

[All] (全部) 在程序停止执行前, 获取跟踪信息。

[Event Trigger] (事件触发) 在每次触发事件成立时, 获取触发发生前的 255 个记录、触发发生行以及触发发生后的 256 个记录, 共计 512 个记录的跟踪信息。

[Trace Event] (跟踪事件)

显示开始获取跟踪信息的事件信息。

显示的项目如下:

[Type] 事件类型

[Condition] 事件条件

选定 [Type] 栏的复选框的跟踪事件有效。

[Add...] (添加 ...) 显示指定事件的对话框。

[Delete] (删除) 删除指定的事件。

[Delete All] (全部删除) 删除全部事件。

[Enable All] (允许全部) 允许全部事件。

[Disable All] (禁止全部) 禁止全部事件。

如果更改跟踪获取对话框的设置, 就清除跟踪信息。

通过单击 [OK] 按钮, 设置所指定的内容。如果单击 [Cancel] (取消) 按钮, 就不设置而关闭对话框。

3.7.3 跟踪事件的设置

跟踪事件使用暂停条件, 作为跟踪事件成立时的运行, 获取事件成立前后的跟踪信息。

在 [Select Break Type] (选择暂停类型) 对话框中设置跟踪事件。

要打开 [Select Break Type] (选择暂停类型) 对话框时, 请单击 [Trace Acquisition] (跟踪获取) 对话框的 [Add] (添加) 按钮, 或者在 [Event] (事件) 窗口的 [Software event] 选项卡上从弹出式菜单中选择 [Add...] (添加...)。

有关跟踪事件的条件、条件成立时的运行设置, 请参照 “3.6 仿真 / 调试程序断点的使用”。

要更改跟踪事件的条件时, 请在 [Trace Event] (跟踪事件) 双击要更改的事件条件, 打开 [Select Break Type] (选择暂停类型) 对话框。

3.7.4 跟踪信息的获取

如果在开始获取跟踪信息的状态下执行指令，就能获取跟踪。

在 [Trace]（跟踪）窗口中显示获取的跟踪信息。

跟踪信息的显示能进行总线显示、反汇编显示、源显示及其混合显示。

(1) 总线显示模式

从弹出式菜单中选择 [Display Mode -> BUS]（显示模式 -> BUS）。

(a) 全部获取模式

显示从开始仿真到停止仿真的跟踪信息。

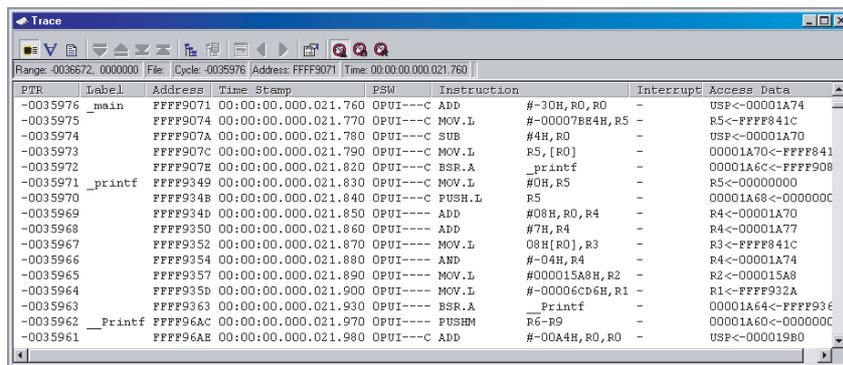


图 3.28 全部获取模式的 [Trace]（跟踪）窗口（总线显示模式）

显示的项目如下：

[PTR]	跟踪缓冲器内的指针（最后执行的指令为“0”。）
[Label]	对应地址的标签（只在设置了标签时才显示。）
[Address]	指令地址
[Time Stamp]	累计的指令执行时间（时：分：秒.毫秒.微秒.毫微秒）
[PSW]	用助记符显示处理器状态字（PSW）的值。
[Instruction]	指令助记符
[Interrupt]	中断（发生中断：“Interrupt”，没有发生中断：“—”）
[Access Data]	数据存取（用传送目标<- 传送数据的格式进行显示。）*1

*1: 字符串操作指令和乘加运算指令的数据存取显示只限于最后的数据。

(b) 事件触发模式

显示事件成立前后的信息。1 次显示 1 个事件所获取的 512 个记录，能在跟踪窗口的弹出式菜单 [Trace Point -> Trace Point Previous]（跟踪点 -> 前一个跟踪点）和 [Trace Point -> Trace Point Next]（跟踪点 -> 下一个跟踪点）中更改显示的信息。在停止仿真后，显示最早的事件跟踪信息。

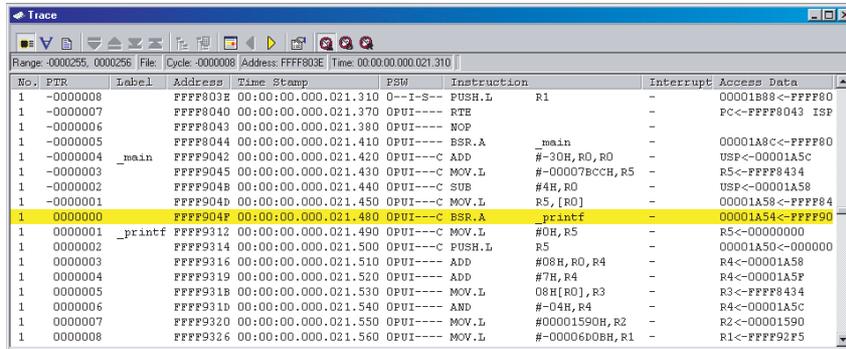


图 3.29 事件触发模式的 [Trace]（跟踪）窗口（总线显示模式）

显示的项目如下：

- [No.] 跟踪点的序号（显示在开始执行后第几次成立的跟踪点。）
- [PTR] 跟踪缓冲器内的指针（显示的信息触发的行是“0”。）
- [Label] 对应地址的标签（只在设置了标签时才显示。）
- [Address] 指令地址
- [Time Stamp] 累计的指令执行时间（时：分：秒.毫秒.微秒.毫微秒）
- [PSW] 用助记符显示处理器状态字（PSW）的值。
- [Instruction] 指令助记符
- [Interrupt] 中断（发生中断：“Interrupt”，没有发生中断：“—”）
- [Access Data] 数据存取（用传送目标<- 传送数据的格式进行显示。）*1

*1: 字符串操作指令和乘加运算指令的数据存取显示只限于最后的数据。

(2) 反汇编显示模式

从弹出式菜单中选择 [Display Mode -> DIS]（显示模式 -> DIS）。
能参照执行的指令。

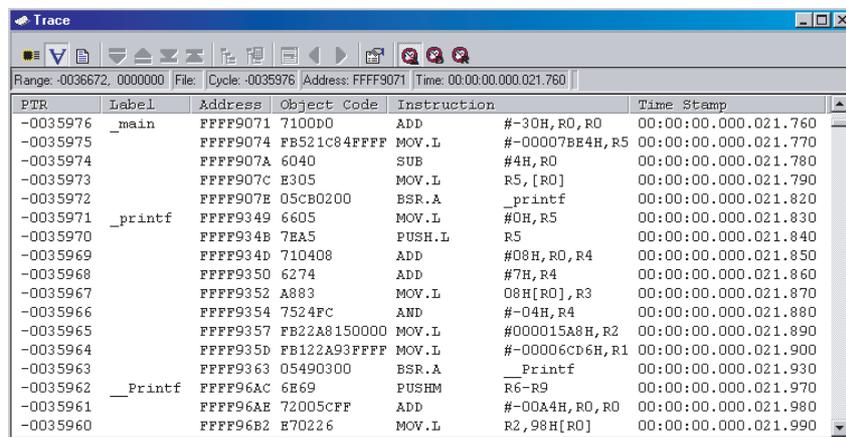


图 3.30 [Trace]（跟踪）窗口（反汇编模式）

(3) 源显示模式

从弹出式菜单中选择 [Display Mode -> SRC] (显示模式 -> SRC)。

能参照源程序的执行路径。

从当前的跟踪周期开始, 按正向或者反向在跟踪数据内进行源的单步执行, 能确认执行路径。

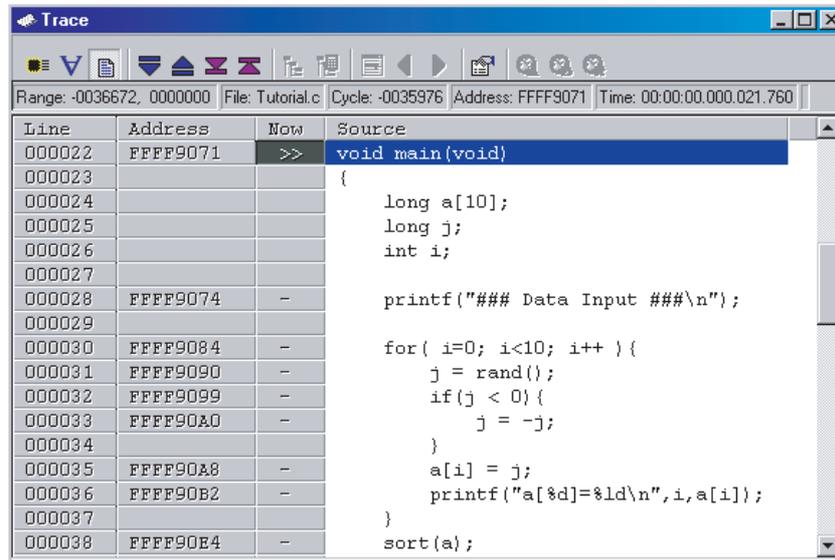


图 3.31 [Trace] (跟踪) 窗口 (源显示模式)

(4) 混合显示模式

能进行总线显示、反汇编显示、源显示的混合显示。

在选择弹出式菜单的 [Display Mode -> BUS] (显示模式 -> BUS) 后, 如果选择 [Display Mode -> DIS] (显示模式 -> DIS), 就能进行总线和反汇编的混合显示。

用同样的方法, 能进行总线和源、反汇编和源以及总线、反汇编和源的混合显示。

在设置为总线和反汇编的混合显示后, 要返回到只进行总线显示时, 请再次选择弹出式菜单的 [Display Mode -> DIS] (显示 -> DIS) 就能变为总线显示。

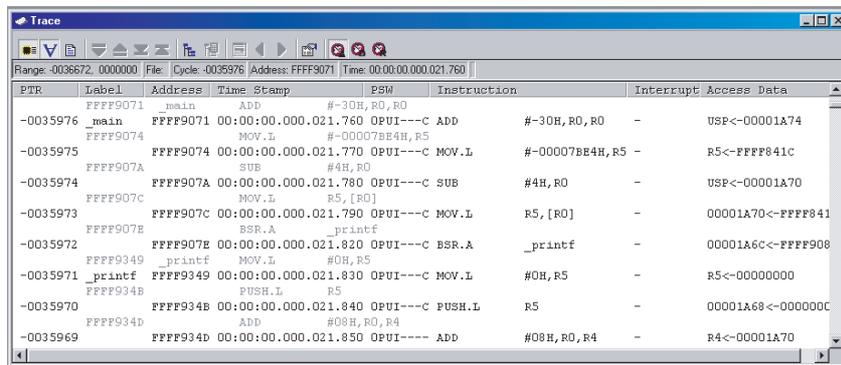


图 3.32 [Trace] (跟踪) 窗口 (混合显示模式)

3.7.5 跟踪信息的检索

要检索跟踪信息时，请使用 [Find]（查找）对话框。

要打开 [Find]（查找）对话框时，请选择弹出式菜单的 [Find -> Find...]（查找 -> 查找 ...）。

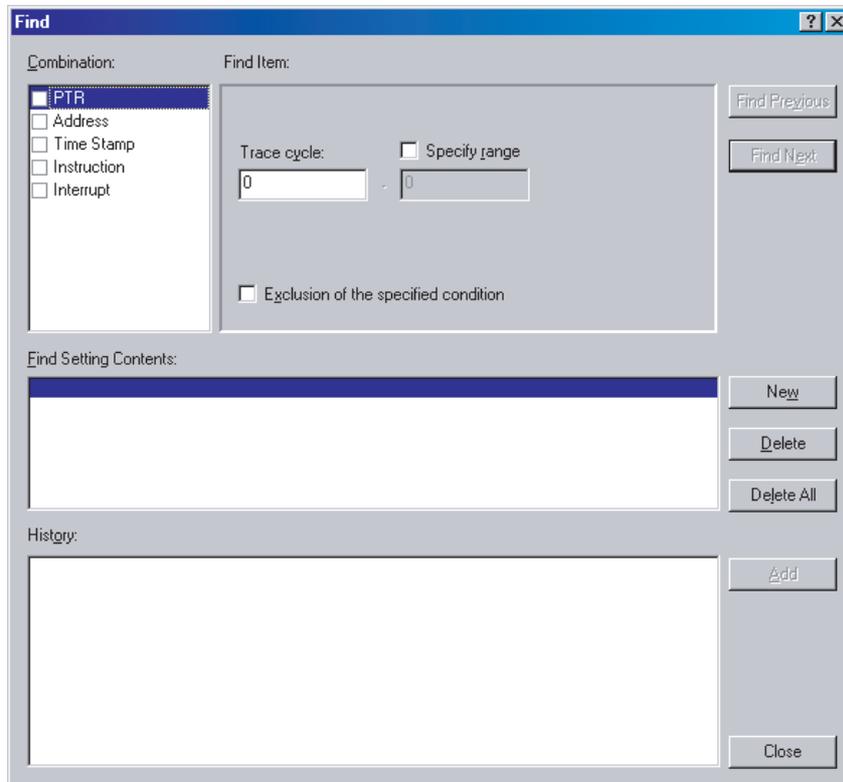


图 3.33 [Find]（查找）对话框

请在 [Combination]（组合）栏中选择查找对象的条件后选定复选框。

能在 [Find Item]（查找项目）栏中指定与选择条件对应的详细项目。

如果在 [Combination]（组合）栏中选择多个条件，就必须对各条件指定详细项目。查找对象为多个条件的 AND。

跟踪检索的对象项目如下所示：

项目	内容	检索条件
[PTR]	跟踪缓冲器内的指针	指定数值（10 进制数）。 可指定范围。 可进行指定条件以外的检索。
[Address]	指令地址	指定数值（16 进制数）。 可指定范围。 可进行指定条件以外的检索。
[Time Stamp]	累计的指令执行时间	通过各时间单位的编辑框进行指定。 可指定范围。 可进行指定条件以外的检索。
[Instruction]	指令助记符	指定字符串。 可进行指定条件以外的检索。
[Interrupt]	中断的发生	检索固定字符串“Interrupt”。 可进行指定条件以外的检索。

在 [Find Setting Contents]（查找设置内容）列表框中显示设置的检索条件。

在设置检索条件后，单击 [Find Previous]（向前查找）按钮或者 [Find Next]（向后查找）按钮开始检索。

如果找到与检索结果相同的记录，就强调显示该记录。

如果没有找到相同的记录，就显示信息对话框。

在找到相同记录的情况下，如果在弹出式菜单中选择 [Find Previous]（向后查找）或者 [Find Next]（向前查找），就能检索下一个记录。

3.7.6 跟踪信息的筛选

能利用筛选功能从获取的跟踪信息中抽出需要的记录。要使用筛选功能时，请从 [Trace]（跟踪）窗口的弹出式菜单中将 [Auto Filter]（自动筛选）置为有效。如果将 [Auto Filter]（自动筛选）置为有效，就在各列中显示自动筛选的箭头记号 。如果单击箭头记号  并且从下拉式列表中选择 [Option...]（选项），就显示选择筛选条件的选项对话框。

能筛选的项目和筛选条件与跟踪记录的检索对象项目和检索条件相同。

【注】 在指定事件触发时不能使用筛选功能的显示。

3.7.7 跟踪信息的清除

如果在获取跟踪信息后再次执行指令仿真，就清除跟踪信息。

3.7.8 跟踪信息的文件保存

以文本格式而不能以二进制格式保存 [Trace]（跟踪）窗口显示的跟踪信息。

要将跟踪信息保存到文件时，请从弹出式菜单中选择 [File -> Save...]（文件 -> 保存...）。

显示 [Save As]（另存为）文件对话框，将跟踪缓冲器的内容保存为文本文件，并且能通过 [Start - End Cycle]（开始 - 结束周期）指定保存的范围。必须注意：此文件不能重新加载到跟踪缓冲器。

3.7.9 源文件的显示

要在 [Editor]（编辑）窗口中显示对应跟踪记录的源文件时，请以源显示模式从弹出式菜单中选择 [File -> Edit Source]（文件 -> 编辑源）。

要以跟踪窗口的源显示模式来显示别的源文件时，请使用 [Display Source]（显示源）对话框。

要打开 [Display Source]（显示源）对话框时，请选择弹出式菜单的 [File -> Display Source]（文件 -> 显示源）。

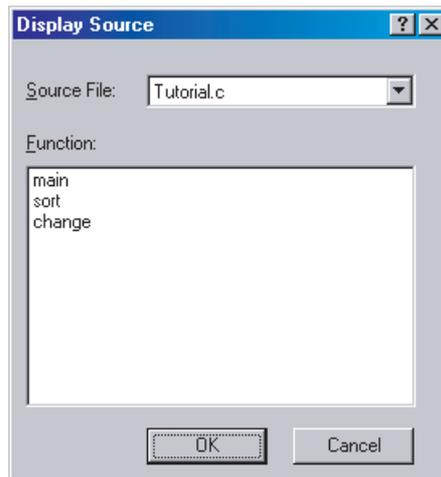


图 3.34 [Display Source]（显示源）对话框

在此对话框中选择跟踪窗口显示的源文件。如果在设置条件后单击 [OK] 按钮，就在跟踪窗口中显示源文件并且强调显示所选函数的起始部分。

3.7.10 时戳显示的切换

能将 [Trace]（跟踪）窗口显示的时戳切换为绝对时间、差分时间和相对时间。

初始状态显示绝对时间。

- 绝对时间
从弹出式菜单中选择 [Time -> Absolute Time]（时间 -> 绝对时间）或者单击 [Absolute Time]（绝对时间）工具栏按钮 。
- 差分时间
从弹出式菜单中选择 [Time -> Differences]（时间 -> 差分）或者单击 [Differences]（差分）工具栏按钮 。
- 相对时间
从弹出式菜单中选择 [Time -> Relative Time]（时间 -> 相对时间）或者单击 [Relative Time]（相对时间）工具栏按钮 。

3.7.11 函数执行履历的显示

能从获取的跟踪信息中显示函数执行履历。要显示函数执行履历时，请从弹出式菜单中选择 [Function Execution History -> Function Execution History]（函数执行履历 -> 函数执行履历）或者单击 [Function Execution History]（函数执行履历）工具栏按钮 。

如果从弹出式菜单中选择 [Analyze Execution History]（分析执行履历）或者单击 [Analyze Execution History]（分析执行履历）工具栏按钮 ，就从跟踪结果的最后开始分析并且以树结构显示结果。

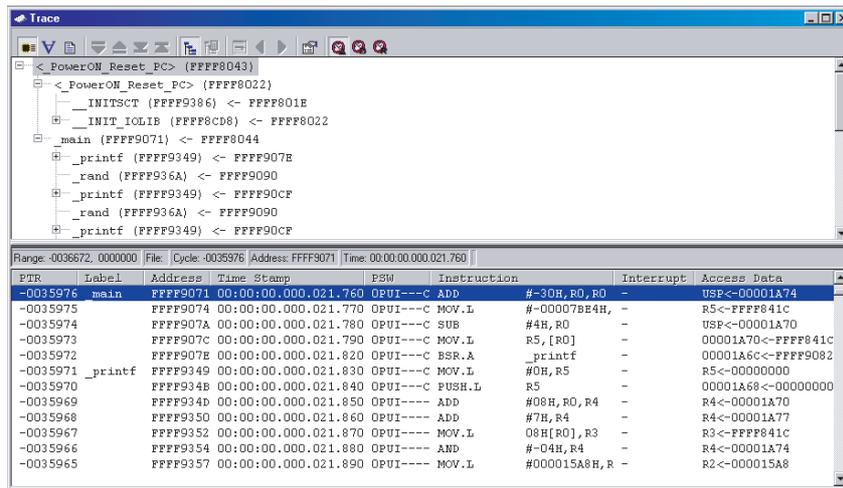


图 3.35 [Trace]（跟踪）窗口（函数执行履历的显示）

在下段的窗口中，从上段窗口中所选函数的记录开始显示跟踪结果。

【注】 在指定事件触发时不能使用函数执行履历的显示。

3.8 剖析信息的查看

剖析功能以函数为单位测量应用程序的执行性能，能调查应用程序中性能恶化的位置和原因。
HEW 根据剖析数据的参照方法和参照目的，在 3 个窗口中显示剖析测量结果。

3.8.1 堆栈信息文件

剖析功能能读优化链接程序（Ver.7.0 以后）输出的堆栈信息文件（扩展名 “.SNI”），此文件中有源文件中的（静态）函数调用关系的信息。HEW 通过读堆栈信息文件，即使不执行用户应用程序（在进行剖析数据测量前）也能显示函数的调用关系（但是，在 [Profile]（剖析）窗口的弹出式菜单中选定了 [Setting -> Only Executed Functions]（设置 -> 仅限执行的函数）的情况除外）。

如果 HEW 不读堆栈信息文件，剖析功能显示的数据就只限于剖析数据测量中执行的函数。

要使用链接程序生成堆栈信息文件时，请在 [Standard Toolchain]（标准工具链）对话框的 [Link/Library]（链接/库）选项卡中将 [Category]（种类）列表框指定为 [Other]（其他），并且选定 [Stack information output]（堆栈信息输出）复选框。

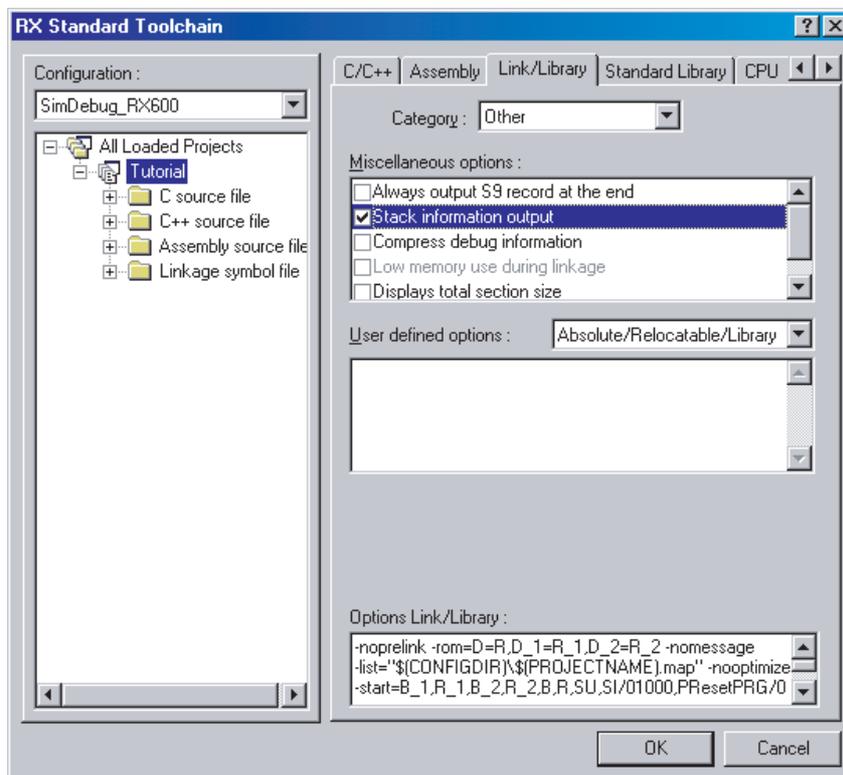


图 3.36 [Standard Toolchain]（标准工具链）对话框 (1)

3.8.2 堆栈信息文件的加载

能在加载装入模块时显示的确认信息框中指定是否读堆栈信息文件。如果单击信息框的 [OK] 按钮，就加载堆栈信息文件。

在以下情况下显示确认信息框：

- 在堆栈信息文件存在时
- 在 [Options] (选项) 对话框 (如果选择主菜单的 [Setup -> Options...] (设置 -> 选项...))，就打开对话框) 的 [Confirmation] (确认) 选项卡 (图 3.37) 中选定了 [Load Stack Information Files (SNI files)] (加载堆栈信息文件(SNI文件)) 复选框时

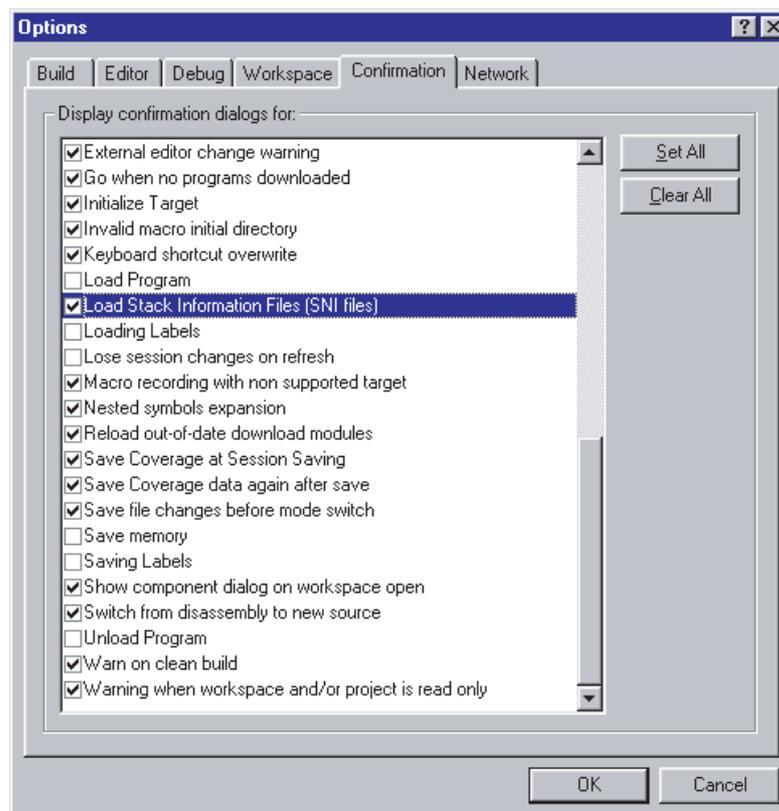


图 3.37 [Options] (选项) 对话框

3.8.3 允许剖析

选择 [View -> Performance -> Profile] (视图 -> 性能 -> 剖析), 打开 [Profile] (剖析) 窗口。

在 [Profile] (剖析) 窗口的弹出式菜单中选择 [Enable Profiler] (允许剖析器) 菜单选项 (菜单中有复选标记)。

3.8.4 测量方法的指定

在测量剖析数据时, 能指定是否跟踪函数的调用。如果跟踪函数的调用, 就能以树的格式显示用户程序执行时的函数调用关系。如果不跟踪函数调用, 就不能显示函数调用关系, 但是能缩短剖析数据的测量时间。

为了不跟踪函数调用, 请选择 [Profile] (剖析) 窗口的弹出式菜单 [Disable Tree (Not traces function call)] (禁止树 (不跟踪函数调用)) (菜单中有复选标记)。

对于通过 OS 的任务开关等非一般的方法调用函数的程序, 可能无法正确显示函数的调用, 因此在测量剖析数据时不能跟踪函数的调用。

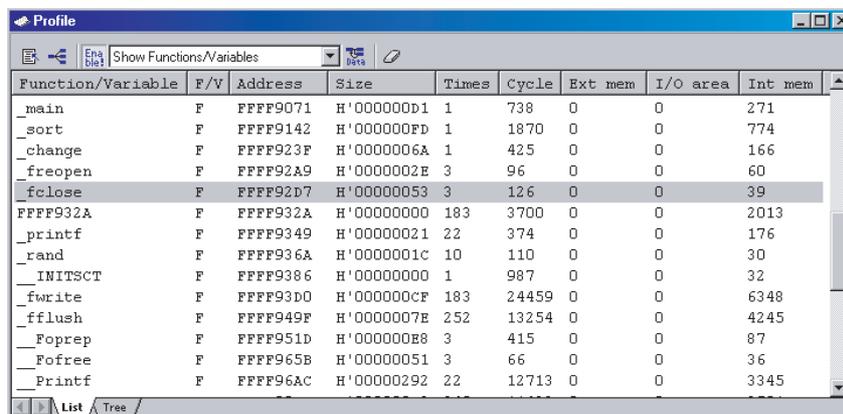
3.8.5 用户程序执行结果的确认

如果在执行后停止用户程序, 就在 [Profile] (剖析) 窗口中显示测量结果。

[Profile] (剖析) 窗口有 [List] 表和 [Tree] 表。

3.8.6 [List] 表

列表页显示函数和全局变量, 并且显示各函数 / 变量的剖析数据。



Function/Variable	F/V	Address	Size	Times	Cycle	Ext mem	I/O area	Int mem
_main	F	FFFF9071	H'000000D1	1	738	0	0	271
_sort	F	FFFF9142	H'000000FD	1	1870	0	0	774
_change	F	FFFF923F	H'0000006A	1	425	0	0	166
_freopen	F	FFFF92A9	H'0000002E	3	96	0	0	60
_fclose	F	FFFF92D7	H'00000053	3	126	0	0	39
FFFF932A	F	FFFF932A	H'00000000	183	3700	0	0	2013
_printf	F	FFFF9349	H'00000021	22	374	0	0	176
_rand	F	FFFF936A	H'0000001C	10	110	0	0	30
__INITSCT	F	FFFF9386	H'00000000	1	987	0	0	32
_fwrite	F	FFFF93D0	H'000000cF	183	24459	0	0	6348
_fflush	F	FFFF949F	H'0000007E	252	13254	0	0	4245
_fopen	F	FFFF951D	H'000000E8	3	415	0	0	87
_free	F	FFFF965B	H'00000051	3	66	0	0	36
_Printf	F	FFFF96AC	H'00000292	22	12713	0	0	3345

图 3.38 [List] 表

如果单击列首, 就按字母或者数值升降顺序进行排序和显示。

当单击 [Function/Variable] (函数 / 变量) 列或者 [Address] (地址) 列时, 显示与该地址对应的源程序。

如果在窗口内单击鼠标的右键, 就显示弹出式菜单。有关此弹出式菜单, 请参照 “3.8.7 [Tree] 表”。

3.8.7 [Tree] 表

显示函数的调用关系和各函数调用位置的剖析数据。

[Tree] 表只在没有选定 [Profile]（剖析）窗口的弹出式菜单 [Disable Tree (Not traces function call)]（禁止树（不跟踪函数调用））时有效。

Function	Address	Size	Stack Size	Times	Cycle	Ext mem	I/O area	Int mem
_main	FFFF9071	H'000000D1	H'00000000	1	738	0	0	271
_printf	FFFF9349	H'00000021	H'00000000	22	374	0	0	176
_rand	FFFF936A	H'0000001C	H'00000000	10	110	0	0	30
_change	FFFF923F	H'0000006A	H'00000000	1	425	0	0	166
_sort	FFFF9142	H'000000FD	H'00000000	1	1870	0	0	774
_CLOSEALL	FFFF8DF7	H'0000004F	H'00000000	1	510	0	0	144
_INIT_IOLIB	FFFF8CD8	H'0000011F	H'00000000	1	89	0	0	31
_freopen	FFFF92A9	H'0000002E	H'00000000	3	96	0	0	60
_fclose	FFFF92D7	H'00000053	H'00000000	3	126	0	0	39
_fflush	FFFF949F	H'0000007E	H'00000000	3	57	0	0	12
_fopen	FFFF965B	H'00000051	H'00000000	3	66	0	0	36
_close	FFFF8EE1	H'00000009	H'00000000	3	21	0	0	6
_Poprep	FFFF951D	H'000000E8	H'00000000	3	415	0	0	87

图 3.39 [Tree] 表

如果双击 [Function]（函数）列的函数，展开或者折叠显示树结构。也能用 '+'/'-' 键进行展开或者折叠显示。如果双击 [Address]（地址）列，就显示对应该地址的源程序。

如果在窗口内单击鼠标的右键，就显示弹出式菜单，此菜单包括以下的选项。

(1) [View Source]（查看源）

显示与所选行的该地址对应的源程序或者反汇编。

(2) [View Profile-Chart]（查看剖析图）

显示所进行函数的 [Profile-Chart]（剖析图）窗口。

(3) [Enable Profiler]（允许剖析器）

能切换剖析数据测量的 ON 或者 OFF。当剖析数据测量为 ON 时，在菜单文本的左边显示复选标记。

(4) [Not trace the function call]（不跟踪函数调用）

如果选定此菜单，就在测量剖析数据时不跟踪函数调用。例如，用于对通过 OS 任务开关的非一般方法调用函数的程序进行数据测量的情况。

为了在 [Profile]（剖析）窗口的 [Tree] 表中显示函数的调用关系，在测量剖析数据时不能选定此菜单。另外，在使用测量结果的剖析信息文件，并且通过优化链接编辑程序进行优化时，也不能选定此菜单。

(5) [Find...]（查找...）

显示要查找 [Function] 列的字符串的 [Find Text]（查找文本）对话框。如果在编辑框中输入要查找的字符串后单击 [Find Next]（查找下一个）按钮或者输入“Enter”键，就开始检索。

(6) [Find Data...] (查找数据 ...)

显示 [Find Data] (查找数据) 对话框。

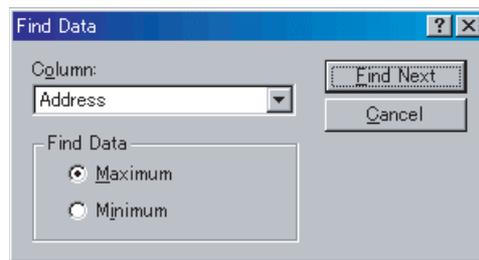


图 3.40 [Find Data] (查找数据) 对话框

在 [Column] (列) 组合框设置检索列并且在 [Find Data] (查找数据) 组设置检索方向后, 如果单击 [Find Next] (查找下一个) 按钮或者输入 “Enter”, 就开始检索。如果连续单击 [Find Next] (查找下一个) 按钮或者输入 “Enter”, 就检索下一个大的数据 (如果指定最小值就查找小的数据)。

(7) [Clear Data] (清除数据)

清除函数调用次数的计数和剖析数据, 并且清除 [Profile] (剖析) 窗口的 [List] 表和 [Profile-Chart] (剖析图) 窗口的数据。

(8) [Output Profile Information Files...] (输出剖析信息文件 ...)

显示 [Save Profile Information Files] (保存剖析信息文件) 对话框, 将剖析结果保存到剖析信息文件 (扩展名 “.pro”)。

(9) [Output Text File...] (输出文本文件 ...)

显示 [Save Text of Profile Data] (以文本格式保存剖析数据) 对话框, 将显示的状态保存到文本文件。

(10) [Setting] (设置)

此菜单有以下子菜单 (以下说明中也包括只有 [List] 表的菜单)。

(a) [Show Functions/Variable] (显示函数 / 变量)

在 [Function/Variable] (函数 / 变量) 列中显示函数和全局变量。

(b) [Show Functions] (显示函数)

在 [Function/Variable] (函数 / 变量) 列中只显示函数。

(c) [Show Variable] (显示变量)

在 [Function/Variable] (函数 / 变量) 列中只显示全局变量。

(d) [Only Executed Functions] (仅限执行的函数)

能只显示已执行的函数。当优化链接编辑程序输出的堆栈使用量信息文件 (扩展名: sni) 和装入模块不在同一个目录时, 与此复选框的设置无关, 只显示已执行的函数。

(e) [Include Data of Child Functions] (包括子函数的数据)

设置在显示的剖析数据中是否包括函数内调用的子函数的剖析数据。

(11) [Properties...] (属性 ...)

本仿真 / 调试程序不能使用属性。

3.8.8 剖析图窗口

[Profile-Chart]（剖析图）窗口显示着眼于特定函数的函数调用关系。此窗口以着眼函数为中心，左边显示该着眼函数的调用源函数，右边显示该着眼函数调用的函数，并且显示各调用次数。

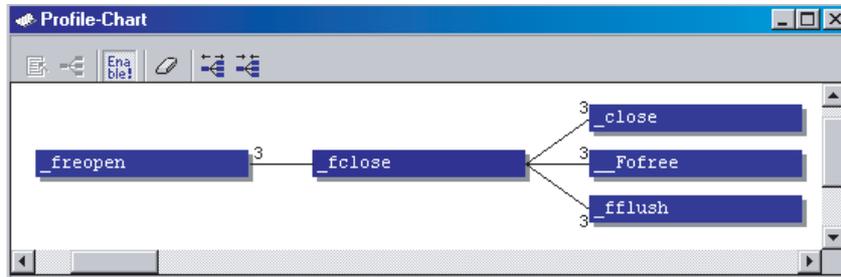


图 3.41 [Profile-Chart]（剖析图）窗口

如果在窗口内单击鼠标的右键，就显示弹出式菜单，此菜单包括以下选项。

(1) [View Source]（查看源）

对于单击鼠标右键时的位置的函数，显示与该函数的地址对应的源程序或者反汇编。如果单击鼠标右键时的位置不是函数，此菜单选项就显示为灰色。

(2) [View Profile-Chart]（查看剖析图）

对于单击鼠标右键时的位置的函数，显示着眼于该函数的 [Profile-Chart]（剖析图）窗口。如果单击鼠标右键时的位置不是函数，此菜单选项就显示为灰色。

(3) [Enable Profiler]（允许剖析器）

允许或者禁止切换剖析数据收集。当剖析数据的允许测量时，菜单文本的左边显示复选标记。

(4) [Clear Data]（清除数据）

清除函数调用次数的计数，并且清除 [Profile]（剖析）窗口的 [List] 表和 [Tree] 表的数据。

(5) [Multiple View]（多视图）

在显示 [Profile-Chart]（剖析图）窗口时，如果 [Profile-Chart]（剖析图）窗口已经打开，就设置是打开别的窗口还是在同窗口中显示。如果在菜单文本的左边显示了复选标记，就打开别的窗口。

(6) [Output Profile Information Files...]（输出剖析信息文件...）

显示 [Save Profile Information Files]（保存剖析信息）对话框。将剖析结果保存到剖析信息文件（扩展名“.pro”）。优化链接编辑程序能根据剖析信息优化用户程序。有关使用剖析信息进行优化的详细内容，请参照优化链接编辑程序的手册。

(7) [Expands Size] 展开

展开显示各函数的间隔。也能用“+”键进行展开显示。

(8) [Reduces Size] 折叠

折叠显示各函数的间隔。也能用“-”键进行折叠显示。

3.8.9 显示数据的种类和用途

能从剖析功能中获得以下信息。

(1) Address

能知道在存储器中配置函数的位置。通过按地址顺序排序显示，能以存储器中的配置图像排列函数和全局变量。

(2) Size

如果按大小顺序排序显示，就能找到小而频繁被调用的函数。如果存在这样的函数，通过将该函数置为 `inline` 函数，就可能降低函数调用的系统开销。

(3) Stack Size

如果函数调用层次很深，就需要通过追寻函数调用路径并且累计该路径上全部函数的堆栈大小，大致估算堆栈使用量。

(4) Times

如果按调用（存取）次数顺序排序显示，就能简单地调查频繁调用的函数和频繁存取的变量。

(5) 剖析数据

- [Cycle]（执行周期数）
- [Ext_mem]（外部存储器的存取次数）
- [I/O_area]（内部I/O的存取次数）
- [Int_mem]（内部存储器的存取次数）

能从执行该函数的调用指令时的累计执行周期数和执行该函数的返回指令时的累计执行周期数的差，计算执行周期数。

字符串操作指令和乘加运算指令的数据存取次数只限于最后数据的 1 次存取。

3.8.10 剖析信息文件的创建

要创建剖析信息文件时，请选择 Pop-up 菜单的 [Output Profile Information Files...]（输出剖析信息文件 ...）菜单选项。显示 [Output Profile Information Files...]（输出剖析信息文件 ...）对话框，如果在选择文件名后单击 [Save]（保存）按钮，就将剖析信息写到所选择的文件。如果单击 [Save All]（全部保存）按钮，就将剖析信息写到全部文件。

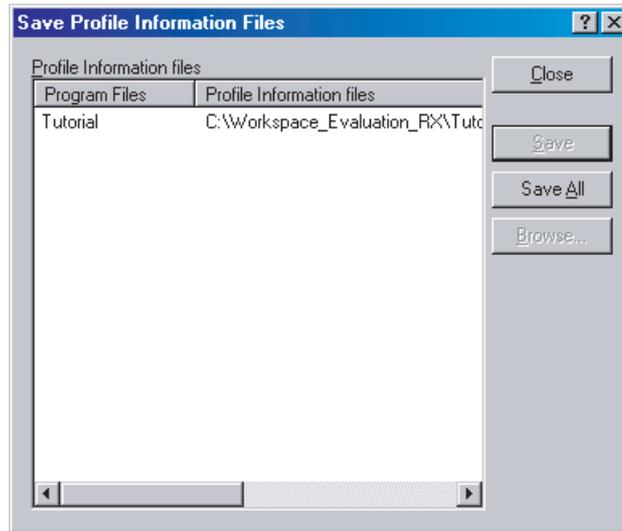


图 3.42 [Save Profile Information Files]（保存剖析信息文件）对话框

3.8.11 注意事项

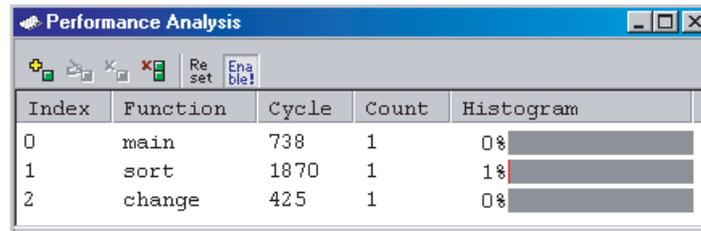
1. 通过剖析功能测量应用程序执行周期数时的数据可能有误差。剖析功能能调查整个应用程序中各函数所占执行时间的比率。但是，如果需要更严密地测量函数的执行周期数，就必须使用性能分析功能。
2. 如果通过没有调试信息的装入模块进行剖析信息的测量，就可能不显示函数名。
3. 堆栈信息文件（扩展名“.SNI”）需要与装入模块文件（扩展名“.ABS”）一起放在同一个目录。
4. 不能累积测量结果。
5. 不能编辑测量结果。

3.9 性能分析

要指定函数名进行性能分析时，请使用 [Performance Analysis]（性能分析）窗口。

3.9.1 性能分析窗口的打开

要打开 [Performance Analysis]（性能分析）窗口时，请选择 [View -> Performance -> Performance Analysis]（视图 -> 性能 -> 性能分析）或者单击 [Performance Analysis]（性能分析）工具栏按钮 。



Index	Function	Cycle	Count	Histogram
0	main	738	1	0%
1	sort	1870	1	1%
2	change	425	1	0%

图 3.43 [Performance Analysis]（性能分析）窗口

此窗口显示各指定函数的执行周期数。

用以下计算式求执行周期数。

执行周期数 = 执行指定函数的返回指令时的累计执行周期数 - 执行指定函数的调用指令时的累计执行周期数

显示的项目如下：

[Index]	设置条件的索引号
[Function]	测量对象的函数名（或者函数的起始地址）
[Cycle]	该函数的累计执行周期数
[Count]	该函数的累计调用次数
[Histogram]	该函数在整个程序的执行周期数中所占执行周期数的比例 以百分比和直方图的格式进行显示。

3.9.2 评估函数的设置

打开 [Performance Analysis]（性能分析）窗口，设置要评估的函数。如果从弹出式菜单中选择 [Add Range...]（添加范围...），就打开 [Performance Option]（性能选项）对话框。也能用“Insert”键打开 [Performance Option]（性能选项）对话框。



图 3.44 [Performance Option]（性能选项）对话框

在此对话框中设置要进行性能评估的函数（也可以是标签），最多能设置 255 个评估函数。

通过单击 [OK] 按钮设置评估函数。如果单击 [Cancel]（取消）按钮，就不设置而关闭对话框。

在选择所设置的评估函数后，如果选择弹出式菜单的 [Edit Range]（编辑范围），就显示 [Performance Option]（性能选项）对话框，能更改评估函数。也能用“Insert”键打开 [Performance Option]（性能选项）对话框。

3.9.3 数据收集的开始

如果在选定弹出式菜单的 [Enable Analysis]（允许分析）后执行程序，就能收集性能数据。

3.9.4 数据的复位

如果从弹出式菜单中选择 [Reset Counts/Times]（复位次数/次），就清除当前程序的性能数据。

3.9.5 评估函数的删除

如果在选择评估函数的状态下从弹出式菜单中选择 [Delete Range]（删除范围），就删除被选择的评估函数，重新计算其他范围的数据。也能用“Delete”键删除评估函数。

3.9.6 评估函数的全部删除

如果从弹出式菜单中选择 [Delete All Ranges]（删除全部范围），就删除全部的当前评估函数，并且清除性能数据。

3.9.7 当前显示内容的保存

能将当前窗口显示的内容保存到文本文件。请从弹出式菜单中选择 [Save to File...]（保存到文件...）。

3.10 代码覆盖率的测量

对用户指定的地址范围，[Coverage]（覆盖率）窗口收集代码覆盖率信息（C0 覆盖率和 C1 覆盖率）并且显示结果。

3.10.1 覆盖率窗口的打开

如果选择 [View -> Code -> Coverage...]（视图 -> 代码 -> 覆盖率 ...）或者单击 [Coverage]（覆盖率）工具栏按钮 ，就打开 [Open Coverage]（打开覆盖率）对话框。

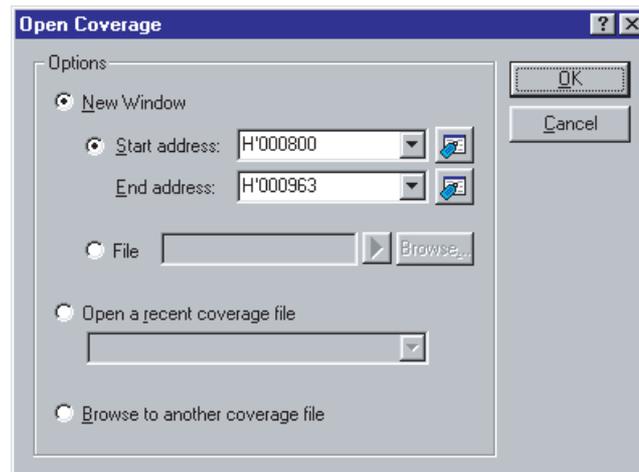


图 3.45 [Open Coverage]（打开覆盖率）对话框

在 [Open Coverage]（打开覆盖率）对话框中指定覆盖率测量范围。

要设置新的覆盖率范围时，有以下 2 种方法：

- 在 [新的覆盖率测量范围] 中指定起始地址和结束地址。

指定的项目如下：

[Start Address]（起始地址）	指定覆盖率信息显示的起始地址（省略前缀时为 16 进制输入）。
[End Address]（结束地址）	指定覆盖率信息显示的结束地址（省略前缀时为 16 进制输入）。

- 在 [新的覆盖率测量范围] 中指定文件。

指定的项目如下：

[File]（文件）	指定当前工程中有 “.C” 或者 “.CPP” 扩展名的源文件。 将指定文件内存在的函数设置为覆盖率范围。 如果省略扩展名，就附加 “.C”。不能指定 “.C” 和 “.CPP” 扩展名以外的文件。 能利用占位符或者 [Browse...]（浏览 ...）按钮。
------------	--

要重新设置已保存在覆盖率信息文件中的内容时，请从 [Open a recent coverage file]（打开最近使用过的覆盖率文件）中选择或者通过 [Browse to another coverage file]（浏览其他覆盖率文件）显示文件打开对话框进行选择。

在 [Open a recent coverage file]（打开最近使用过的覆盖率文件）中最多显示 4 个最近保存的文件。

如果单击 [OK] 按钮，就显示 [Coverage]（覆盖率）窗口。

如果因指定地址而 [Coverage]（覆盖率）窗口已处于显示状态，就在此窗口中添加设置。

(1) 覆盖率窗口（指定地址）

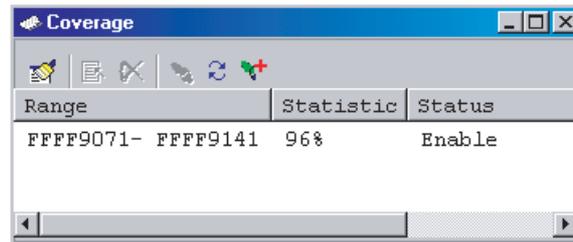


图 3.46 [Coverage]（覆盖率）窗口（指定地址）

显示覆盖率范围和覆盖率统计信息。

显示的项目如下：

- [Range]（范围） 地址范围
- [Statistic]（统计） C0 覆盖率值（百分比显示）
- [Status]（状态） 允许 / 禁止各覆盖率范围的状态

如果关闭 [Coverage]（覆盖率）窗口，就清除所获取的覆盖率信息以及设置的获取条件。

(2) 覆盖率窗口（指定源文件）

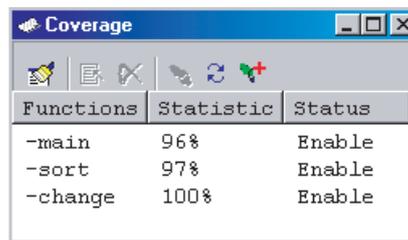


图 3.47 [Coverage]（覆盖率）窗口（指定源文件）

显示覆盖率范围和覆盖率统计信息。

显示的项目如下：

- [Functions]（函数） 覆盖率对象的函数
- [Statistic]（统计） C0 覆盖率值（百分比显示）
- [Status]（状态） 允许 / 禁止各函数的状态
- 通过单击列选项卡，能按函数名或者 C0 覆盖率值的降序或者升序重新排列。

如果关闭 [Coverage]（覆盖率）窗口，就清除所获取的覆盖率信息以及设置的获取条件。

3.10.2 覆盖率信息的全部获取

如果从弹出式菜单中选定 [Enable All]（允许全部）后执行指令，就获取全部覆盖率信息。默认值是选定 [Enable All]（允许全部）。

3.10.3 覆盖率信息的全部清除

如果从弹出式菜单中选择 [Clear All]（全部清除），就清除所获取的全部覆盖率信息。

3.10.4 源文件的显示

如果从弹出式菜单中选择 [View Source]（查看源），就打开 [Editor]（编辑）窗口，显示与 [Coverage]（覆盖率）窗口光标位置的地址对应的源文件。

3.10.5 新覆盖率测量范围的设置

如果从弹出式菜单中选择 [Add Range...]（添加范围...），就显示 [Open Coverage]（打开覆盖率）对话框（图 3.45）。有关 [Open Coverage]（打开覆盖率）对话框，请参照“3.10.1 覆盖率窗口的打开”。

3.10.6 覆盖率测量范围的更改

(1) 用地址指定覆盖率测量范围的情况

如果选择覆盖率测量范围并且从弹出式菜单中选择 [Edit Range...]（编辑范围...），就显示 [Coverage Range]（覆盖率范围）对话框。

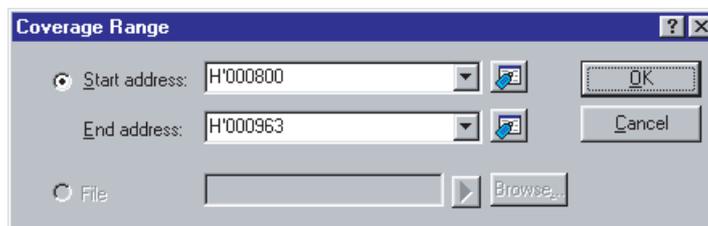


图 3.48 [Coverage Range]（覆盖率范围）对话框（指定地址）

在此对话框中更改指令执行信息获取的条件，能指定以下的项目：

[Start address]（起始地址） 起始地址（省略前缀时为 16 进制输入）
 [End address]（结束地址） 结束地址（省略前缀时为 16 进制输入）

如果单击 [OK] 按钮，就更改变覆盖率测量范围。

(2) 用源文件指定覆盖率测量范围的情况

如果从弹出式菜单中选择 [Edit Range...]（编辑范围...），就显示 [Coverage Range]（覆盖率范围）对话框。

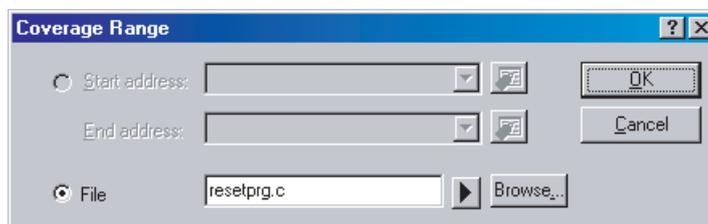


图 3.49 [Coverage Range]（覆盖率范围）对话框（指定源文件）

在此对话框中更改指令执行信息获取的条件，能指定以下的项目：

[File]（文件） 指定当前工程中有“.C”或者“.CPP”扩展名的源文件。
 将指定文件内存在的函数设置为覆盖率范围。
 如果省略扩展名，就附加“.C”。不能指定“.C”和“.CPP”扩展名以外的文件。
 能利用占位符或者 [Browse...]（浏览...）按钮。

如果单击 [OK] 按钮，就更改变覆盖率测量范围。

3.10.7 覆盖率测量范围的删除

如果选择覆盖率测量范围并且从弹出式菜单中选择 [Delete Range]（删除范围），就删除所选择的覆盖率测量范围。

3.10.8 覆盖率信息的获取

如果选择覆盖率测量范围并且从弹出式菜单选定 [Enable Coverage]（允许覆盖率）后执行指令，就获取覆盖率信息。默认值是选定 [Enable Coverage]（禁止覆盖率）。

3.10.9 覆盖率信息的清除

如果选择覆盖率测量范围并且从弹出式菜单中选择 [Clear Data]（清除数据），就清除所获取的覆盖率信息。

3.10.10 覆盖率信息的文件保存

如果从弹出式菜单中选择 [Save Data...]（保存数据 ...），就显示将覆盖率信息保存到文件的 [Save Data]（保存数据）对话框。



图 3.50 [Save Data]（保存数据）对话框

指定要保存的覆盖率信息文件的位置和名字。能使用占位符或者 [Browse...]（浏览 ...）按钮。

如果省略输入文件扩展名，就自动附加 “.COV” 文件扩展名。

如果输入的文件扩展名不是 “.COV” 或者 “.TXT”，就输出错误信息。

3.10.11 覆盖率信息文件的加载

如果从弹出式菜单中选择 [Load Data...]（加载数据 ...），就显示从文件加载覆盖率信息的 [Load Data]（加载数据）对话框。



图 3.51 [Load Data]（加载数据）对话框

指定要加载的覆盖率信息文件的位置和名字。能使用占位符或者 [Browse...]（浏览 ...）按钮。

能加载的文件扩展名只限于 “.COV”，如果输入其他文件扩展名，就输出错误信息。

3.10.12 最新信息的更新

如果从弹出式菜单中选择 [Refresh]（更新），就将 [Coverage]（覆盖率）窗口的内容更新为最新的内容。

3.10.13 [Confirmation Request]（确认请求）对话框

如果单击 [Clear All]（全部清除）、[Clear Data]（清除）、[Edit Range...]（编辑范围...）、[Delete Range]（删除范围）或者关闭 [Coverage]（覆盖率）窗口，就显示确认对话框。

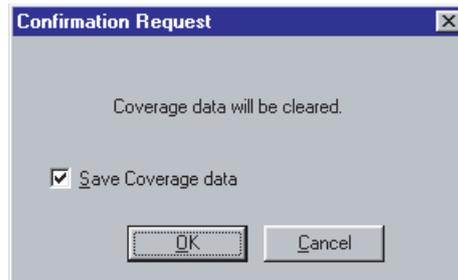


图 3.52 [Confirmation Request]（确认请求）对话框

如果单击 [OK] 按钮，就清除覆盖率数据。

如果选定 [Save Coverage data]（保存覆盖率数据），就显示图 3.50 所示的 [Save Data]（保存数据）对话框，能在清除前将覆盖率数据保存到文件。

3.10.14 覆盖率数据的保存对话框

如果单击 [File -> Save Session]（文件 -> 会话的保存）菜单选项，就显示 [Save Coverage data]（保存覆盖率数据）对话框，能分别或者统一保存 [Coverage]（覆盖率）窗口的数据。

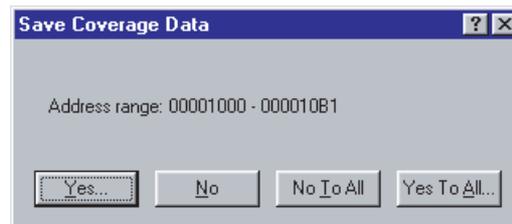


图 3.53 [Save Coverage Data]（保存覆盖率数据）对话框

- 如果打开了多个 [Coverage]（覆盖率）窗口，就打开窗口个数的 [Save Coverage Data]（保存覆盖率数据）对话框。
- 如果单击 [No To All]（全否）按钮，就不保存全部覆盖率数据而关闭对话框。
- 如果单击 [Yes To All]（全是）按钮，就将全部 [Coverage]（覆盖率）窗口的数据保存到 1 个文件。

3.10.15 编辑窗口的覆盖率结果显示

通过强调显示与指令执行后的源行对应的 [Coverage]（覆盖率）列，在 [Editor]（编辑）窗口中显示覆盖率结果。如果在 [Coverage]（覆盖率）窗口中更改覆盖率的相关设置，就更新对应的 [Coverage]（覆盖率）列的显示。

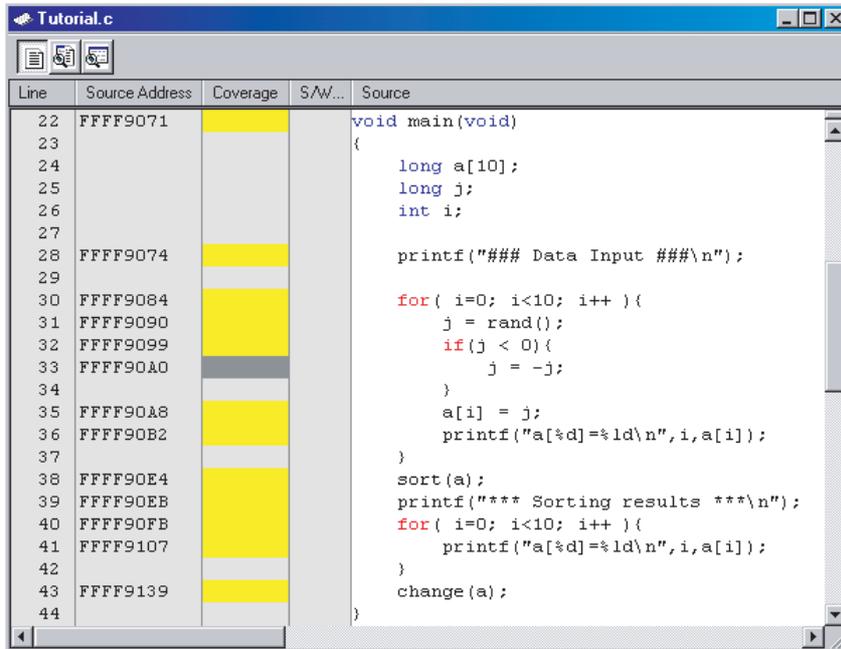


图 3.54 覆盖率列（源）

3.10.16 反汇编窗口的覆盖率结果显示

通过强调显示与指令执行后的反汇编行对应的 [Coverage-ASM]（覆盖率 ASM）列，在 [Disassembly]（反汇编）窗口中显示覆盖率结果。如果在 [Coverage]（覆盖率）窗口中更改覆盖率的相关设置，就更新对应的 [Coverage - ASM]（覆盖率 ASM）列的显示。

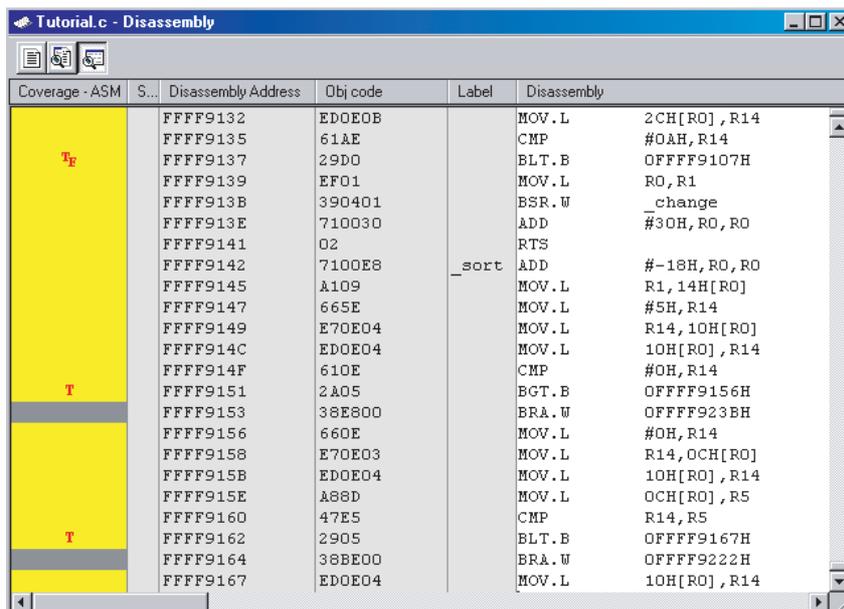


图 3.55 覆盖率列（反汇编）

3.11 伪中断的手动产生

要在窗口上单击按钮来手动产生伪中断时，请使用 [Trigger]（触发）窗口或者 [GUI I/O] 窗口。

3.11.1 触发窗口

要打开 [Trigger]（触发）窗口时，请选择 [View -> CPU -> Trigger]（视图 -> CPU -> 触发）或者单击 [Trigger]（触发）工具栏按钮 。

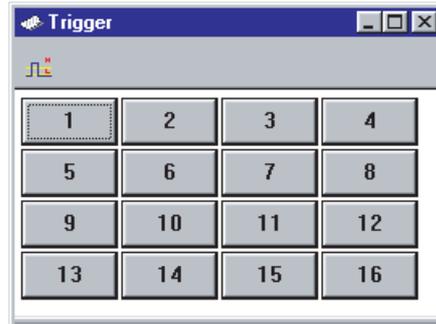


图 3.56 [Trigger]（触发）窗口

此窗口显示手动产生伪中断的触发按钮。在 [Trigger Setting]（触发设置）对话框中，设置按下触发按钮时产生的伪中断内容。

最多能设置 256 个触发按钮。

有关仿真 / 调试程序的伪中断处理，请参照“2.15 伪中断”。

(1) 触发按钮的设置

要设置按下各触发按钮时产生的伪中断内容时，请使用 [Trigger Setting]（触发设置）对话框。
要打开 [Trigger Setting]（触发设置）对话框时，请选择弹出式菜单的 [Setting...]（设置 ...）。

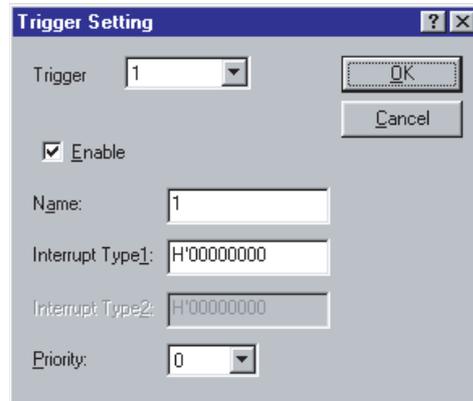


图 3.57 [Trigger Setting]（触发设置）对话框

在此对话框中设置按下触发按钮时产生的中断内容。

[Trigger]（触发）	选择要设置的触发按钮。
[Name]（名字）	指定 [Trigger]（触发）窗口显示的触发按钮名。
[Enable]（允许）	如果选定，就允许触发按钮。
[Interrupt type1]（中断条件 1）	指定中断向量号。
[Priority]（优先级）	指定中断优先级（省略前缀时为 16 进制输入和 16 进制显示）（0 ~ 8 或者 0 ~ H'10）。 如果在 0 ~ 8 时指定 8 或者在 0 ~ H'10 时指定 H'10，就作为高速中断处理。

通过单击 [OK] 按钮设置指定的内容。如果单击 [Cancel]（取消）按钮，就不设置而关闭对话框。

【注】 如果在更改多个触发按钮的设置后单击 [Cancel]（取消）按钮，全部触发按钮的设置更改就无效。

(2) 触发按钮个数的更改

要更改 [Trigger]（触发）窗口显示的触发按钮个数时，请从弹出式菜单 [Number of Buttons]（按钮数）的子菜单中选择 [4]、[16]、[64]、[256]。

(3) 触发按钮大小的更改

要更改 [Trigger]（触发）窗口显示的触发按钮大小时，请从弹出式菜单 [Size]（大小）的子菜单中选择 [Large]（大）、[Normal]（中）、[Small]（小）。

3.11.2 [GUI I/O] (GUI 输入 / 输出) 窗口

要打开 [GUI I/O] 窗口时, 请选择 [View -> Graphic -> GUI I/O] (视图 -> 图形 -> GUI I/O) 或者单击 [GUI I/O] 工具栏按钮 。

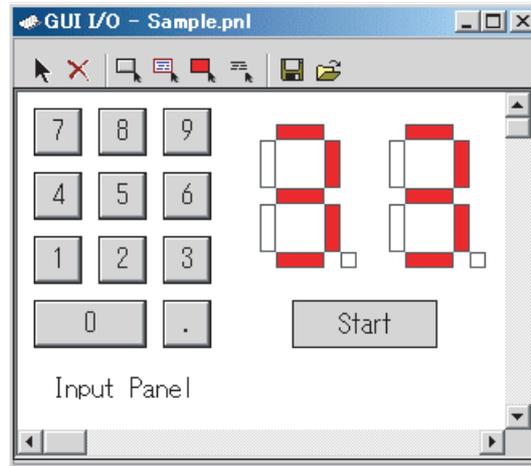


图 3.58 GUI I/O 窗口

在此窗口中设置手动产生伪中断的按钮。在 [Set Button] (设置按钮) 对话框中设置按下按钮时产生的伪中断内容。

有关仿真 / 调试程序的伪中断处理, 请参照 “2.15 伪中断”。

(1) 按钮的设置

要设置产生伪中断的按钮时, 请使用 [Set Button] (设置按钮) 对话框。

首先, 请选择弹出式菜单的 [Create Button] (创建按钮) 或者单击工具栏的 [Create Button] (创建按钮) 按钮 , 鼠标光标的形状变为 “+”。

请在此状态下将鼠标光标从要创建的按钮的左上位置拖拉到右下位置。

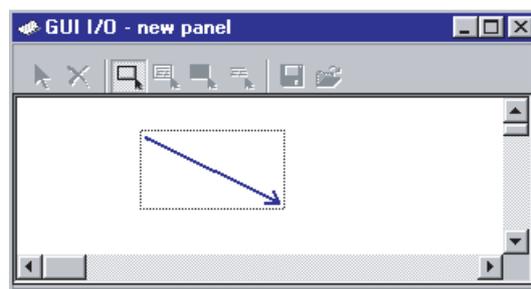


图 3.59 GUI I/O 窗口 (创建按钮)

然后, 双击此按钮, 打开 [Set Button] (设置按钮) 对话框。

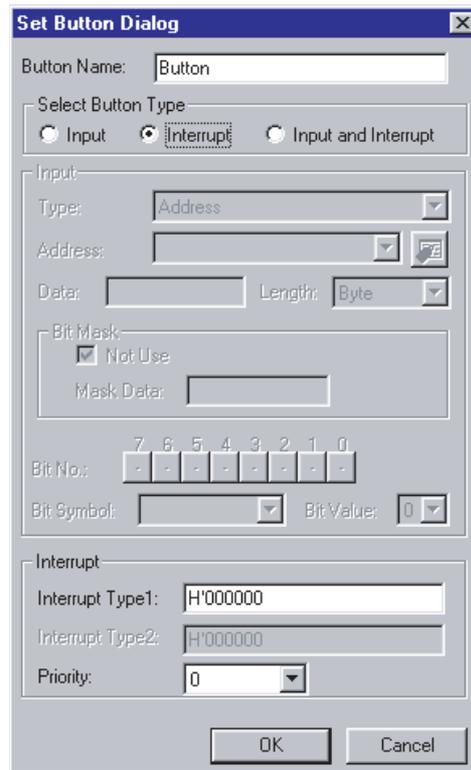


图 3.60 [Set Button Dialog] (设置按钮对话框)

在此对话框中设置按下按钮时产生的中断内容。

[Button Name] (按钮名)

[Select Button Type] (选择按钮类型)

[Interrupt] (中断) [Interrupt Type1] (中断条件 1)

[Priority] (优先级)

指定 [GUI I/O] 窗口显示的按钮名。

选择 [Input] (输入) 或者 [Input and Interrupt] (输入和中断)。

指定中断向量号。

指定中断优先级 (省略前缀时为 16 进制输入和 16 进制显示) (0 ~ 8 或者 0 ~ H'10)。

如果在 0 ~ 8 时指定 8 或者在 0 ~ H'10 时指定 H'10, 就作为高速中断处理。

3.12 标准输入 / 输出和文件输入 / 输出

要从用户程序进行标准输入 / 输出和文件输入 / 输出的仿真 I/O 时，请使用 [Simulated I/O]（仿真 I/O）窗口。

3.12.1 仿真 I/O 窗口的打开

要打开 [Simulated I/O]（仿真 I/O）窗口时，请选择 [View -> CPU -> Simulated I/O]（视图 -> CPU -> 仿真 I/O）或者单击 [Simulated I/O]（仿真 I/O）工具栏按钮 。

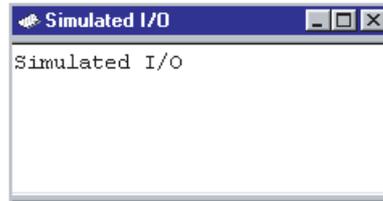


图 3.61 [Simulated I/O]（仿真 I/O）窗口

用户程序的标准输出被输出到此窗口，此窗口上的键输入为用户程序的标准输入。

3.12.2 输入 / 输出功能

支持的输入 / 输出处理如下：

表 3.1 输入 / 输出功能一览表

No.	功能码	功能名	内容
1	H'21	GETC	从标准输入进行 1 字节的输入。
2	H'22	PUTC	给标准输出进行 1 字节的输出。
3	H'23	GETS	从标准输入进行 1 行的输入。
4	H'24	PUTS	给标准输出进行 1 行的输出。
5	H'25	FOPEN	打开文件。
6	H'06	FCLOSE	关闭文件。
7	H'27	FGETC	从文件进行 1 字节的输入。
8	H'28	FPUTC	给文件进行 1 字节的输出。
9	H'29	FGETS	从文件进行 1 行的输入。
10	H'2A	FPUTS	给文件进行 1 行的输出。
11	H'0B	FEOF	检查文件的结束。
12	H'0C	FSEEK	移动文件的指针。
13	H'0D	FTELL	获取文件指针的当前位置。

为了实现此功能，首先在 [Simulator System]（仿真系统）对话框的 [Simulated I/O Address]（仿真 I/O 地址）中指定用于输入 / 输出的特定位置，在选定 [Enable]（允许）后执行用户程序。如果在用户程序的指令执行过程中检测到向指定位置转移的子程序转移指令（BSR、JSR）（即仿真 I/O 指令），仿真 / 调试程序就将 R1 和 R2 的内容作为参数进行输入 / 输出处理。

因此，在进行仿真 I/O 前，需要事先在用户程序中进行以下的设置。

- R1 寄存器：功能码



- R2 寄存器：参数块的地址
(参数块的内容请参照各功能说明。)



- 参数块和输入/输出缓冲区的确保

另外，在存取参数块各参数时，必须以该参数的长度进行存取。
如果输入/输出处理结束，就从仿真 I/O 指令的下一条指令重新开始仿真。
有关各输入/输出功能的详细内容，请参照仿真程序的帮助。

从标准输入（键盘）进行 1 个字母输入的例子如下所示。作为仿真 I/O 地址，指定标签 SYS_CALL。

```

MOV.L      #01210000h, R1
MOV.L      #PARM, R2
MOV.L      # SYS_CALL, R3
JSR        R3
STOP       NOP
SYS_CALL   NOP
PARM       .LWORD      INBUF
           .SECTION    B, DATA
INBUF      .BLKB       2
           .END

```

3.13 虚拟输入 / 输出面板的创建

仿真 / 调试程序具有在窗口上模拟用户目标系统的简单键输入面板和输出面板的 GUI 输入 / 输出功能。要创建此虚拟输入 / 输出面板时，请使用 GUI 输入 / 输出窗口。在 GUI 输入 / 输出窗口中，能配置虚拟按钮进行数据输入以及配置虚拟 LED 进行数据输出。

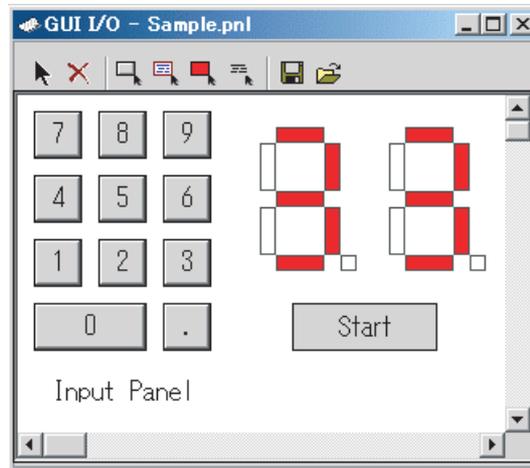


图 3.62 [GUI I/O] (GUI 输入 / 输出) 窗口例子

3.13.1 [GUI I/O] (GUI 输入 / 输出) 的打开

如果选择 [View -> Graphic -> GUI I/O] (视图 -> 图形 -> GUI 输入 / 输出) 或者单击 [GUI I/O] 工具栏按钮 ，就打开 GUI I/O 窗口。

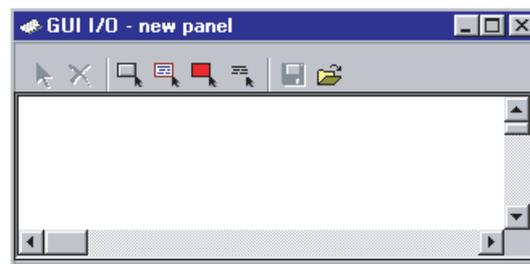


图 3.63 [GUI I/O] (GUI 输入 / 输出) 窗口

能在 GUI 输入 / 输出窗口中配置以下项目：

- [Button] (按钮)
通过按下按钮，给虚拟端口输入数据或者产生虚拟中断。
- [Label] (标签)
在给指定地址 (或者位) 写指定值时，显示或者擦除字符串。
- LED
在给指定地址 (或者位) 写指定值时，用指定的颜色显示 (替代LED点灯)。
- [Text] (文本)
显示文本字符串。

3.13.2 创建按钮

请单击工具栏按钮  或者选择弹出式菜单的 [Create Button]（创建按钮），鼠标光标的形状变为“+”。
请在此状态下将鼠标光标从要创建的按钮的左上位置拖拉到右下位置。

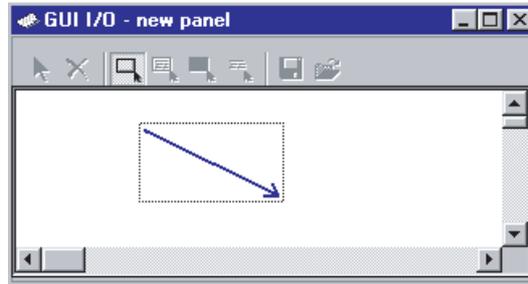


图 3.64 [GUI I/O]（GUI 输入 / 输出）窗口（创建按钮）

(1) 单击按钮时的事件指定

请在单击工具栏按钮  的状态下双击要创建的按钮，打开以下的对话框。

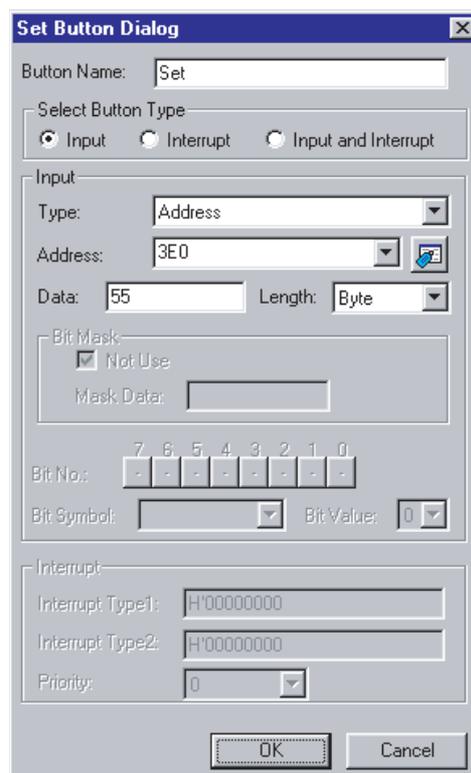


图 3.65 [Set Button Dialog]（设置按钮对话框）

请指定按钮名、输入端口地址和输入数据。
按钮名不能使用空格字符。

3.13.3 创建标签

请单击工具栏按钮  或者选择弹出式菜单的 [Create Label]（创建标签），鼠标光标的形状变为“+”。请在此状态下将鼠标光标从要创建的位置的左上位置拖拉到右下位置，显示标签框。

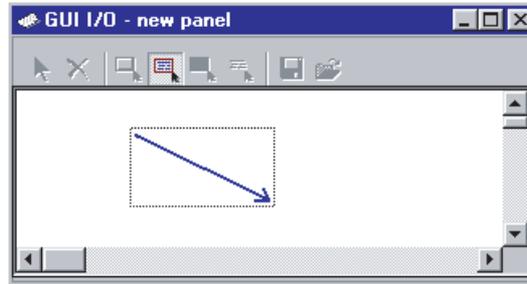


图 3.66 [GUI I/O]（GUI 输入 / 输出）窗口（创建标签）

请在单击工具栏按钮  或者选择弹出式菜单的 [Select Item]（选择项目）后双击标签框，打开设置标签内容的对话框。

请指定事件发生时的运行。标签名不能使用空格字符。

(1) 给指定位写任意值的情况

以下的设置为：当地址 0x3E0 的 bit3 为“0”时显示“印刷中”；当 bit3 为“1”时显示“可印刷”。

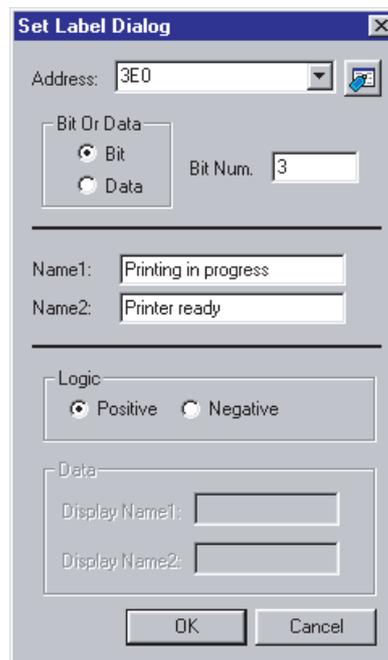


图 3.67 [Set Label Dialog]（设置标签对话框）（指定位）

(2) 给指定地址写任意值的情况

以下的设置为：当给地址 0x3E0 写了数据 0x10 时显示“印刷中”；当写了数据 0x20 时显示“可印刷”。

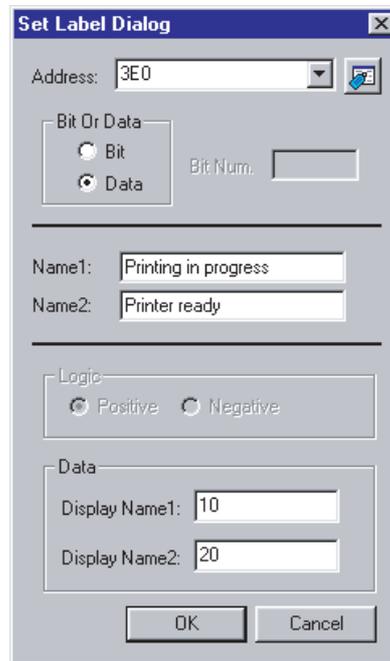


图 3.68 [Set Label Dialog]（设置标签对话框）（指定数据）

3.13.4 创建 LED

请单击工具栏按钮  或者选择弹出式菜单的 [Create LED]（创建 LED），鼠标光标的形状变为“+”。请在此状态下将鼠标光标从要创建的位置的左上位置拖拉到右下位置，显示 LED 输出框。

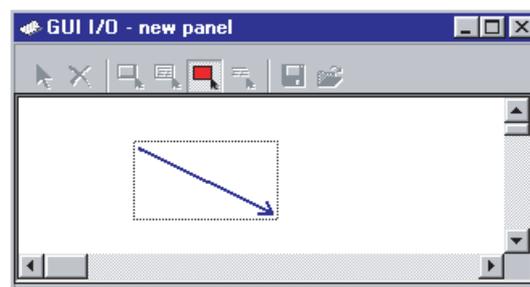


图 3.69 [GUI I/O]（GUI 输入 / 输出）窗口（创建 LED）

请在单击工具栏按钮  或者选择弹出式菜单的 [Select Item]（选择项目）后双击 LED 输出框，打开设置 LED 输出的对话框。请指定事件发生时的运行。

(1) 给指定位写任意值的情况

以下的设置为：当地址 0x3E0 的 bit2 为 “0” 时显示绿色；当 bit2 为 “1” 时显示红色。

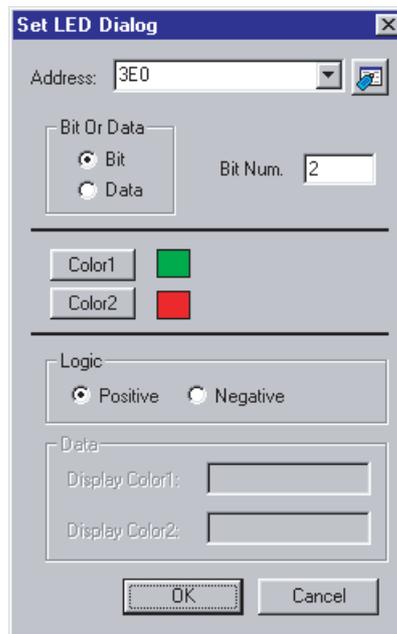


图 3.70 [Set LED Dialog] (设置 LED 对话框) (指定位)

(2) 给指定地址写任意值的情况

以下的设置为：当给地址 0x3E0 写了数据 0x10 时显示绿色；当写了数据 0x20 时显示红色。

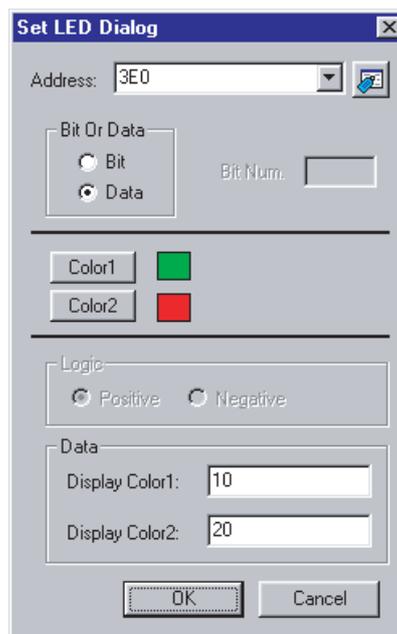


图 3.71 [Set LED Dialog] (设置 LED 设置对话框) (指定数据)

如果单击显示色 1 或者显示色 2 的按钮，就打开颜色选择对话框。

3.13.5 创建文本字符串

请单击工具栏按钮  或者选择弹出式菜单的 [Create Text]（文本的创建），鼠标光标的形状变为“+”。请在此状态下将鼠标光标从要创建的文本字符串的左上位置拖拉到右下位置。

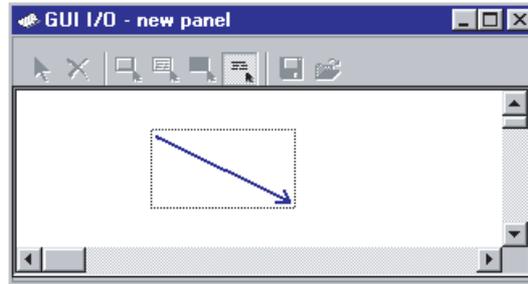


图 3.72 [GUI I/O]（GUI 输入 / 输出）窗口（创建字符串）

(1) 文本字符串显示格式的指定

请在单击了工具栏按钮  的状态下双击要创建的文本字符串。打开以下的设置文本字符串显示格式的对话框。

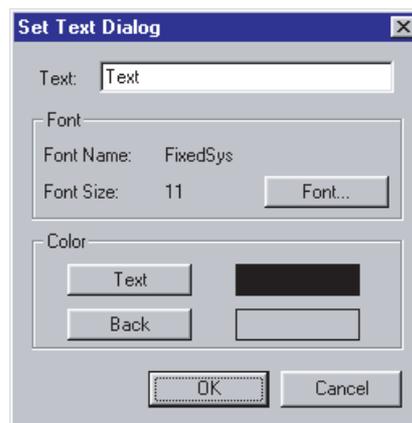


图 3.73 [Set Text Dialog]（设置文本对话框）

请按下 [Font...]（字体）按钮，设置要显示的文本字符串的字体和大小。
 请按下 [Text]（文本）按钮，设置要显示的文本字符串的字符颜色。
 请按下 [Back]（背景）按钮，设置要显示的文本字符串的背景颜色。

3.13.6 项目大小 / 配置的更改

请在单击了工具栏按钮  的状态下单击要更改的项目。项目被选择的状态如下所示。

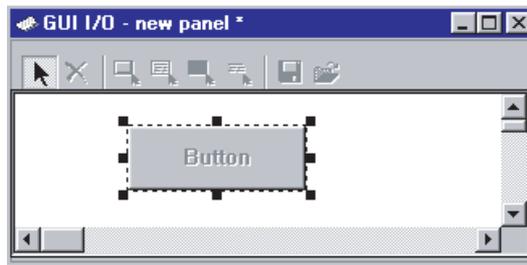


图 3.74 [GUI I/O] (GUI 输入 / 输出) 窗口 (选择项目)

如果在此状态下拖拉所选的项目，就能更改项目大小以及移动项目的位置。

3.13.7 项目的复制

请单击工具栏按钮  或者选择弹出式菜单的 [Copy] (复制)，鼠标光标的形状变为 “+”。

请在此状态下单击要复制的项目，然后单击工具栏按钮  或者选择弹出式菜单的 [Paste] (粘贴)，创建同样大小的项目。

3.13.8 项目的删除

请单击工具栏按钮  或者选择弹出式菜单的 [Delete] (删除)，鼠标光标的形状变为 “+”。

请在此状态下单击要删除的项目。

3.13.9 网格线的显示

请单击工具栏按钮  或者选择弹出式菜单的 [Display Grid] (显示网格)，在背景上显示网格。

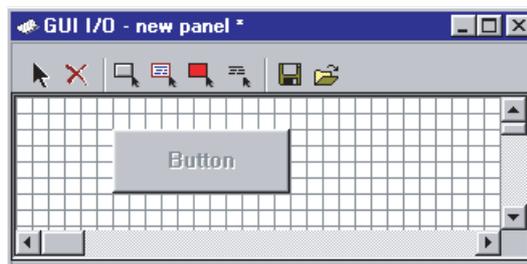


图 3.75 [GUI I/O] (GUI 输入 / 输出) 窗口 (显示网格)

如果再次单击按钮 ，就不显示网格。

3.13.10 输入 / 输出面板信息的保存

能通过将创建的输入 / 输出面板的信息保存到文件进行再利用。

请单击工具栏按钮  或者选择弹出式菜单的 [Save]（保存），打开 [Save GUI I/O Panel File]（保存 GUI 输入 / 输出面板文件）对话框，指定保存目录和文件名。

3.13.11 输入 / 输出面板信息的读取

请单击工具栏按钮  或者选择弹出式菜单的 [Load]（加载），打开 [Load GUI I/O Panel File]（加载 GUI 输入 / 输出面板文件）对话框。

指定要读取的文件，读取前的面板信息被删除。

4. 窗口

窗口一览表如表 4.1 所示。

有关工具栏按钮，请参照仿真 / 调试程序的帮助。

表 4.1 窗口一览表

窗口名	功能
[IO] (输入输出)	查看 I/O 寄存器的内容。
[Simulated I/O] (仿真 I/O)	进行标准输入 / 输出和文件输入 / 输出。
[Event] (事件)	使用仿真 / 调试程序的断点。
[Watch] (监视)	显示变量 (任意变量)。
[Editor] (编辑)	显示源码。
[Image] (图像)	以图像格式显示存储器的内容。
[Coverage] (覆盖率)	测量代码覆盖率。
[Disassembly] (反汇编)	显示汇编语言码。
[Command Line] (命令行)	通过命令行界面进行调试。
[Stack Trace] (堆栈跟踪)	查看函数的调用履历。
[Status] (状态)	参照当前的状态。
[Trigger] (触发)	手动产生伪中断。
[Trace] (跟踪)	查看跟踪信息。
[Wave] (波形)	以波形格式显示存储器的内容。
[Analyzing Performance Register] (性能分析寄存器)	分析性能。
[Profile/Profile-Chart] (剖析 / 剖析图)	查看剖析信息。
[Register] (存储器)	查看存储区。
[Label] (标签)	进行标签的参照和设置。
[Register] (寄存器)	查看寄存器的内容。
[Local] (局部)	显示变量 (局部变量)。
[GUI I/O] (GUI 输入 / 输出)	创建虚拟输入 / 输出面板。
[OS Object] (OS 目标)	显示任务和信号量等 OS 目标的状态。
[Task Trace] (任务跟踪)	测量使用实时 OS 的程序执行履历，并进行图形显示。
[Task Analyze] (任务分析)	显示 CPU 的占有状况。

5. 命令行

5.1 命令一览表（按功能分类）

表示按功能分类的命令一览表。
各命令的语法请参照帮助。

5.1.1 执行的相关命令

命令名	略称	说明
GO	GO	执行用户程序。
GO_RESET	GR	从复位向量开始执行用户程序。
GO_TILL	GT	将用户程序执行到临时断点为止。
HALT	HA	停止用户程序的执行。
RESET	RE	对 CPU 进行复位。
STEP	ST	单步执行（以指令为单位或者以源行为单位）。
STEP_MODE	SM	设置单步模式。
STEP_OUT	SP	单步执行到 PC 位置的函数结束为止。
STEP_OVER	SO	跨步执行。
STEP_RATE	SR	设置并显示单步执行的速度。

5.1.2 下载的相关命令

命令名	略称	说明
BUILD	BU	创建当前工程。
BUILD_ALL	BL	创建当前的全部工程。
BUILD_FILE	BF	编译文件。
BUILD_MULTIPLE	BM	创建多个工程。
CLEAN	CL	删除创建的中间文件和输出文件。
DEFAULT_OBJECT_FORMAT	DO	设置默认的目标（程序）格式。
FILE_LOAD	FL	加载目标（程序）文件。
FILE_LOAD_ALL	LA	加载全部目标（程序）文件。
FILE_SAVE	FS	将存储器的内容保存到文件。
FILE_UNLOAD	FU	卸载目标（程序）文件。
FILE_UNLOAD_ALL	UA	卸载全部目标（程序）文件。
FILE_VERIFY	FV	比较文件内容和存储器内容。
GENERATE_MAKE_FILE	GM	创建当前工作空间的 make 文件。

5.1.3 寄存器操作的相关命令

命令名	略称	说明
REGISTER_DISPLAY	RD	显示寄存器的值。
REGISTER_SET	RS	设置寄存器的值。

5.1.4 存储器操作的相关命令

命令名	略称	说明
CACHE	—	将存储器的高速缓存置为 on 或者 off。
MEMORY_COMPARE	MC	比较存储器的内容。
MEMORY_DISPLAY	MD	显示存储器的内容。
MEMORY_EDIT	ME	更改存储器的内容。
MEMORY_FILL	MF	通过指定的数据, 统一更改存储器的内容。
MEMORY_FIND	MI	查找存储器的内容。
MEMORY_MOVE	MV	移动存储器块。
MEMORY_TEST	MT	测试存储器块。

5.1.5 汇编 / 反汇编的相关命令

命令名	略称	说明
ASSEMBLE	AS	执行汇编。
DISASSEMBLE	DA	显示反汇编。
SYMBOL_ADD	SA	设置符号。
SYMBOL_CLEAR	SC	删除符号。
SYMBOL_LOAD	SL	加载符号信息文件。
SYMBOL_SAVE	SS	将符号信息保存到文件。
SYMBOL_VIEW	SV	显示符号。

5.1.6 暂停设置的相关命令

命令名	略称	说明
BREAKPOINT	BP	设置指令的执行位置引起的断点。
BREAK_ACCESS	BA	设置存储器的存取范围引起的暂停条件。
BREAK_CLEAR	BC	删除断点。
BREAK_CYCLE	BCY	设置周期引起的暂停条件。
BREAK_DATA	BD	设置存储器的数值引起的暂停条件。
BREAK_DATA_DIFFERENCE	BDD	设置存储器的数值变化量 (差分) 引起的暂停条件。
BREAK_DATA_INVERSE	BDI	设置存储器的数值符号取反引起的暂停条件。
BREAK_DATA_RANGE	BDR	设置存储器的数据范围引起的暂停条件。
BREAK_DISPLAY	BI	显示断点一览表。
BREAK_ENABLE	BE	允许或者禁止切换断点。
BREAK_REGISTER	BR	设置寄存器的数值引起的暂停条件。
BREAK_SEQUENCE	BS	设置指定执行顺序的断点。
SET_DISASSEMBLY_SOFT_BREAK	SDB	设置或者解除反汇编级的软件断点。
SET_SOURCE_SOFT_BREAK	SSB	设置或者解除源级的软件断点。
STATE_DISASSEMBLY_SOFT_BREAK	TDB	将反汇编级的软件断点置为允许或者禁止。
STATE_SOURCE_SOFT_BREAK	TSB	将源级的软件断点置为允许或者禁止。

5.1.7 跟踪的相关命令

命令名	略称	说明
TRACE	TR	显示跟踪信息。
TRACE_CONDITION_SET	TCS	设置跟踪信息的获取条件。
TRACE_SAVE	TV	将跟踪信息输出到文件。

5.1.8 覆盖率测量的相关命令

命令名	略称	说明
COVERAGE	CV	将覆盖率测量置为允许或者禁止。
COVERAGE_DISPLAY	CVD	显示覆盖率信息。
COVERAGE_LOAD	CVL	加载覆盖率信息。
COVERAGE_RANGE	CVR	设置覆盖率范围。
COVERAGE_SAVE	CVS	保存覆盖率信息。

5.1.9 性能测量的相关命令

命令名	略称	说明
ANALYSIS	AN	将性能分析功能置为允许或者禁止。
ANALYSIS_RANGE	AR	设置并显示性能评估函数。
ANALYSIS_RANGE_DELETE	AD	删除性能分析范围。
PROFILE	PR	将剖析置为允许或者禁止。
PROFILE_DISPLAY	PD	显示剖析信息。
PROFILE_SAVE	PS	将剖析信息输出到文件。

5.1.10 监视的相关命令

命令名	略称	说明
WATCH_ADD	WA	添加 Watch 项目。
WATCH_AUTO_UPDATE	WU	设置或者解除 Watch 项目的自动更新。
WATCH_DELETE	WD	删除 Watch 项目。
WATCH_DISPLAY	WI	显示监视窗口的内容。
WATCH_EDIT	WE	编辑 Watch 项目的值。
WATCH_EXPAND	WX	展开或者折叠 Watch 项目
WATCH_RADIX	WR	更改 Watch 项目的显示基数。
WATCH_RECORD	WO	将 Watch 项目的值的更新履历输出到文件。
WATCH_SAVE	WS	将监视窗口的显示内容保存到文件。

5.1.11 脚本 / 记录文件的相关命令

命令名	略称	说明
!	—	注释
ASSERT	—	检查条件。
AUTO_COMPLETE	AC	将自动完成功能置为允许或者禁止。
ERASE	ER	清除命令行窗口的内容。
EVALUATE	EV	计算表达式。
LOG	LO	操作记录文件。
SLEEP	—	延迟命令的执行。
SUBMIT	SU	执行命令文件。
TCL	—	显示 TCL 信息。

5.1.12 映像的相关命令

命令名	略称	说明
MAP_DISPLAY	MA	显示存储器资源的设置。
MAP_SET	MS	设置存储器资源。

5.1.13 仿真设置的相关命令

命令名	略称	说明
EXEC_MODE	EM	设置并显示执行模式。
EXEC_STOP_SET	ESS	设置并显示中断发生时的执行模式。

5.1.14 标准输入 / 输出和文件输入 / 输出的相关命令

命令名	略称	说明
SIMULATEDIO_CLEAR	SIOC	清除 [仿真 I/O] 窗口的内容。
TRAP_ADDRESS	TP	设置仿真 I/O 地址。
TRAP_ADDRESS_DISPLAY	TD	显示仿真 I/O 地址的设置。
TRAP_ADDRESS_ENABLE	TE	将仿真 I/O 置为允许或者禁止。

5.1.15 实用程序的相关命令

命令名	略称	说明
HELP	HE	显示命令行的帮助。
INITIALIZE	IN	进行调试平台的初始化。
QUIT	QU	结束 HEW。
RADIX	RA	设置输入的基数。
RESPONSE	RP	设置窗口的更新间隔。
STATUS	STA	显示调试平台的状况。
TOOL_INFORMATION	TO	将当前注册的工具信息输出到文件。

5.1.16 工程 / 工作空间的相关命令

命令名	略称	说明
ADD_FILE	AF	给当前工程添加文件。
CHANGE_CONFIGURATION	CC	设置配置。
CHANGE_PROJECT	CP	设置工程。
CHANGE_SESSION	CS	设置会话。
CHANGE_SUB_SESSION	CB	当允许同步调试时更改有效会话。
CLEAR_OUTPUT_WINDOW	COW	清除输出窗口的各选项卡的显示内容。
CLOSE_WORKSPACE	CW	关闭工作空间。
OPEN_WORKSPACE	OW	打开工作空间。
REFRESH_SESSION	RSE	重读会话信息。
REMOVE_FILE	REM	从当前工程删除文件。
SAVE_SESSION	SE	保存当前会话。
SAVE_WORKSPACE	SW	保存工作空间。
UPDATE_ALL_DEPENDENCIES	UD	更新当前工程的全部依存关系。

5.1.17 测试支援功能的相关命令

命令名	略称	说明
CLOSE_TEST_SUITE	CTS	关闭测试套具。
COMPARE_TEST_DATA	CTD	比较测试数据。
OPEN_TEST_SUITE	OTS	打开测试套具。
RUN_TEST	RT	执行测试。

5.1.18 实时 OS 对应调试功能的相关命令

命令名	略称	说明
OSOBJECT_ALL_ADD	OAA	添加 OS 目标（指定目标类型）。
OSOBJECT_ALL_DELETE	OAD	删除 OS 目标（指定表）。
OSOBJECT_AUTO_UPDATE	OAU	将显示更新更改为“执行中”和“停止时”。
OSOBJECT_DATA_LOWLINE	ODL	将 OS 目标向下移动 1 行（指定单体）。
OSOBJECT_DATA_SAVE	ODS	将 OS 目标的显示内容保存到文件。
OSOBJECT_DATA_UPLINE	ODU	将 OS 目标向上移动 1 行（指定单体）。
OSOBJECT_DISPLAY	OD	显示 OS 目标。
OSOBJECT_NO_UPDATE	ONU	将更新显示更改为“不更新”。
OSOBJECT_ONE_ADD	OOA	添加 OS 目标（指定单体）。
OSOBJECT_ONE_DELETE	OOD	删除 OS 目标（指定单体）。
OSOBJECT_ONE_EDIT	OOE	编辑 OS 目标（指定单体）。
OSOBJECT_SETTING_LOAD	OSL	读 OS 目标设置项目的文件。
OSOBJECT_SETTING_SAVE	OSS	将 OS 目标设置项目保存到文件。
OSOBJECT_STOP_UPDATE	OSU	将显示更新更改为“只限于停止时”。

5.1.19 虚拟端口文件输入 / 输出的相关命令

命令名	略称	说明
PORT_FILE_ADD	PFA	给虚拟端口添加数据输入 / 输出文件。
PORT_FILE_CLOSE	PFC	关闭虚拟端口的数据输入 / 输出文件。
PORT_FILE_DELETE	PFD	删除给虚拟端口指定的数据输入 / 输出文件。
PORT_FILE_OPEN	PFO	打开虚拟端口的数据输入 / 输出文件。
PORT_FILE_STATUS	PFS	显示虚拟端口的数据输入 / 输出文件的状态。

5.2 命令一览表（按字母顺序）

表示字母顺序命令一览表。
各命令的语法请参照帮助。

表 5.1 命令一览表

No.	命令名	略称	说明
1	!	—	注释
2	ADD_FILE	AF	给当前工程添加文件。
3	ANALYSIS	AN	将性能分析功能置为允许或者禁止。
4	ANALYSIS_RANGE	AR	设置并显示性能评估函数。
5	ANALYSIS_RANGE_DELETE	AD	删除性能分析范围。
6	ASSEMBLE	AS	执行汇编。
7	ASSERT	—	检查条件。
8	AUTO_COMPLETE	AC	将自动完成功能置为允许或者禁止。
9	BREAKPOINT	BP	设置指令的执行位置引起的断点。
10	BREAK_ACCESS	BA	设置存储器的存取范围引起的暂停条件。
11	BREAK_CLEAR	BC	删除断点。
12	BREAK_CYCLE	BCY	设置周期引起的暂停条件。
13	BREAK_DATA	BD	设置存储器的数值引起的暂停条件。
14	BREAK_DATA_DIFFERENCE	BDD	设置存储器的数值变化量（差分）引起的暂停条件。
15	BREAK_DATA_INVERSE	BDI	设置存储器的数值符号取反引起的暂停条件。
16	BREAK_DATA_RANGE	BDR	设置存储器的数据范围引起的暂停条件。
17	BREAK_DISPLAY	BI	显示断点一览表。
18	BREAK_ENABLE	BE	允许或者禁止切换断点。
19	BREAK_REGISTER	BR	设置寄存器的数值引起的暂停条件。
20	BREAK_SEQUENCE	BS	设置指定执行顺序的断点。
21	BUILD	BU	创建当前工程
22	BUILD_ALL	BL	创建全部当前工程。
23	BUILD_FILE	BF	编译文件。
24	BUILD_MULTIPLE	BM	创建多个工程。
25	CACHE	—	将存储器的高速缓存置为允许或者禁止。
26	CHANGE_CONFIGURATION	CC	设置配置。
27	CHANGE_PROJECT	CP	设置工程。
28	CHANGE_SESSION	CS	设置会话。
29	CHANGE_SUB_SESSION	CB	当允许同步调试时更改有效会话。
30	CLEAN	CL	删除创建的中间文件和输出文件。
31	CLEAR_OUTPUT_WINDOW	COW	清除输出窗口的各选项卡的显示内容。
32	CLOSE_TEST	CTS	关闭测试套具。
33	CLOSE_WORKSPACE	CW	关闭工作空间
34	COMPARE_TEST_DATA	CTD	比较测试数据。
35	COVERAGE	CV	将覆盖率测量置为允许或者禁止。
36	COVERAGE_DISPLAY	CVD	显示覆盖率信息。
37	COVERAGE_LOAD	CVL	加载覆盖率信息。

No.	命令名	略称	说明
38	COVERAGE_RANGE	CVR	设置覆盖率范围。
39	COVERAGE_SAVE	CVS	保存覆盖率信息。
40	DEFAULT_OBJECT_FORMAT	DO	设置默认的目标（程序）格式。
41	DISASSEMBLE	DA	显示反汇编。
42	ERASE	ER	清除 [Command Line]（命令行）窗口的内容。
43	EVALUATE	EV	计算表达式。
44	EXEC_MODE	EM	设置并显示执行模式。
45	EXEC_STOP_SET	ESS	设置并显示中断发生时的执行模式。
46	FILE_LOAD	FL	加载目标（程序）文件。
47	FILE_LOAD_ALL	LA	加载全部目标（程序）文件。
48	FILE_SAVE	FS	将存储器的内容保存到文件。
49	FILE_UNLOAD	FU	卸载目标（程序）文件。
50	FILE_UNLOAD_ALL	UA	卸载全部目标（程序）文件。
51	FILE_VERIFY	FV	比较文件内容和存储器内容。
52	GENERATE_MAKE_FILE	GM	创建当前工作空间的 make 文件。
53	GO	GO	执行用户程序。
54	GO_RESET	GR	从复位向量开始执行用户程序。
55	GO_TILL	GT	将用户程序执行到临时断点为止。
56	HALT	HA	停止用户程序的执行。
57	HELP	HE	显示命令行的帮助。
58	INITIALIZE	IN	进行调试平台的初始化。
59	LOG	LO	操作记录文件。
60	MAP_DISPLAY	MA	显示存储器资源的设置。
61	MAP_SET	MS	设置存储器资源。
62	MEMORY_COMPARE	MC	比较存储器内容。
63	MEMORY_DISPLAY	MD	显示存储器内容。
64	MEMORY_EDIT	ME	更改存储器内容。
65	MEMORY_FILL	MF	通过指定的数据，统一更改存储器的内容。
66	MEMORY_FIND	MI	查找存储器的内容。
67	MEMORY_MOVE	MV	移动存储器块。
68	MEMORY_TEST	MT	测试存储器块。
69	OPEN_TEST_SUITE	OTS	打开测试套具。
70	OPEN_WORKSPACE	OW	打开工作空间。
71	OSOBJECT_ALL_ADD	OAA	添加 OS 目标（指定目标类型）。
72	OSOBJECT_ALL_DELETE	OAD	删除 OS 目标（指定表）。
73	OSOBJECT_AUTO_UPDATE	OAU	将显示更新更改为“执行中”和“停止时”。
74	OSOBJECT_DATA_LOWLINE	ODL	将 OS 目标向下移动 1 行（指定单体）。
75	OSOBJECT_DATA_SAVE	ODS	将 OS 目标的显示内容保存到文件。
76	OSOBJECT_DATA_UPLINE	ODU	将 OS 目标向上移动 1 行（指定单体）。
77	OSOBJECT_DISPLAY	OD	显示 OS 目标。
78	OSOBJECT_NO_UPDATE	ONU	将更新显示更改为“不更新”。
79	OSOBJECT_ONE_ADD	OOA	添加 OS 目标（指定单体）。

No.	命令名	略称	说明
80	OSOBJECT_ONE_DELETE	OOD	删除 OS 目标（指定单体）。
81	OSOBJECT_ONE_EDIT	OOE	编辑 OS 目标（指定单体）。
82	OSOBJECT_SETTING_LOAD	OSL	读 OS 目标设置项目的文件。
83	OSOBJECT_SETTING_SAVE	OSS	将 OS 目标设置项目保存到文件。
84	OSOBJECT_STOP_UPDATE	OSU	将显示更新更改为“只限于停止时”。
85	PORT_FILE_ADD	PFA	给虚拟端口添加数据输入 / 输出文件。
86	PORT_FILE_CLOSE	PFC	关闭虚拟端口的数据输入 / 输出文件。
87	PORT_FILE_DELETE	PFD	删除虚拟端口的数据输入 / 输出文件。
88	PORT_FILE_OPEN	PFO	打开虚拟端口的数据输入 / 输出文件。
89	PORT_FILE_STATUS	PFS	显示虚拟端口的数据输入 / 输出文件的状态。
90	PROFILE	PR	将剖析置为允许或者禁止。
91	PROFILE_DISPLAY	PD	显示剖析信息。
92	PROFILE_SAVE	PS	将剖析信息输出到文件。
93	QUIT	QU	结束 HEW。
94	RADIX	RA	设置输入的基数。
95	REFRESH_SESSION	RSE	重读会话信息。
96	REGISTER_DISPLAY	RD	显示寄存器值。
97	REGISTER_SET	RS	设置寄存器值。
98	REMOVE_FILE	REM	从当前工程删除文件。
99	RESET	RE	对 CPU 进行复位。
100	RESPONSE	RP	设置窗口的更新间隔。
101	RUN_TEST	RT	执行测试。
102	SLEEP	—	延迟命令的执行。
103	SAVE_SESSION	SE	保存当前会话。
104	SAVE_WORKSPACE	SW	保存工作空间。
105	SET_DISASSEMBLY_SOFT_BREAK	SDB	设置或者解除反汇编级的软件断点。
106	SET_SOURCE_SOFT_BREAK	SSB	设置或者解除源级的软件断点。
107	SIMULATEDIO_CLEAR	SIOC	清除仿真 I/O 窗口的内容。
108	STATE_DISASSEMBLY_SOFT_BREAK	TDB	将反汇编级的软件断点置为允许或者禁止。
109	STATE_SOURCE_SOFT_BREAK	TSB	将源级的软件断点置为允许或者禁止。
110	STATUS	STA	显示调试平台的状况。
111	STEP	ST	单步执行（以指令为单位或者以源行为单位）。
112	STEP_MODE	SM	设置单步模式。
113	STEP_OUT	SP	单步执行到 PC 位置的函数结束为止。
114	STEP_OVER	SO	跨步执行。
115	STEP_RATE	SR	设置并显示单步执行的速度。
116	SUBMIT	SU	执行命令文件。
117	SYMBOL_ADD	SA	设置符号。
118	SYMBOL_CLEAR	SC	删除符号。
119	SYMBOL_LOAD	SL	加载符号信息文件。

No.	命令名	略称	说明
120	SYMBOL_SAVE	SS	将符号信息保存到文件。
121	SYMBOL_VIEW	SV	显示符号。
122	TCL	—	将 TCL 置为允许或者禁止。
123	TOOL_INFORMATION	TO	将当前注册的工具信息输出到文件。
124	TRACE	TR	显示跟踪信息。
125	TRACE_CONDITION_SET	TCS	设置跟踪信息的获取条件。
126	TRACE_SAVE	TV	将跟踪信息输出到文件。
127	TRACE_STATISTIC	TST	执行统计信息分析。
128	TRAP_ADDRESS	TP	设置仿真 I/O 地址。
129	TRAP_ADDRESS_DISPLAY	TD	显示仿真 I/O 地址的设置。
130	TRAP_ADDRESS_ENABLE	TE	将仿真 I/O 调用置为允许或者禁止。
131	UPDATE_ALL_DEPENDENCIES	UD	更新当前工程的全部依存关系。
132	WATCH_ADD	WA	添加 Watch 项目。
133	WATCH_AUTO_UPDATE	WU	设置或者解除 Watch 项目的自动更新。
134	WATCH_DELETE	WD	删除 Watch 项目。
135	WATCH_DISPLAY	WI	显示监视窗口的内容。
136	WATCH_EDIT	WE	编辑 Watch 项目的值。
137	WATCH_EXPAND	WX	展开或者折叠 Watch 项目。
138	WATCH_RADIX	WR	更改 Watch 项目的显示基数。
139	WATCH_RECORD	WO	将 Watch 项目的值的更新履历输出到文件。
140	WATCH_SAVE	WS	将监视窗口的显示内容保存到文件。

6. 信息一览表

6.1 信息

仿真 / 调试程序输出信息，将执行经过通知用户。输出的信息如表 6.1 所示。

表 6.1 信息一览表

信息	内容
Break Access (Access Address:H'nnnnnnnn, Type:xxxx, Access Size:yyyy)	因暂停存取条件成立而中止了执行。作为附加信息，显示成立的暂停存取条件（存取地址（Access Address）、存取类型（Type）和存取长度（Access Size））。
Break Cycle (Cycle:H'nnnnnnnn)	因暂停周期条件成立而中止了执行。作为附加信息，显示成立的暂停周期条件（周期数（Cycle））。
Break Data (Access Address:H'nnnnnnnn, Data:H'mmmm)	因暂停数据条件（符号取反和差分除外）成立而中止了执行。作为附加信息，显示成立的暂停数据条件（存取地址（Access Address）和值（Data））。
Break Data (Access Address:H'nnnnnnnn, Previous Data:H'mmmm, Current Data:H'mmmm)	因暂停数据条件（符号取反或者差分）成立而中止了执行。作为附加信息，显示成立的暂停数据条件（存取地址（Access Address）、上次的值（Previous Data）和本次的值（Current Data））。
Break Register (Register:XX, Value:H'mmmm)	因暂停寄存器条件成立而中止了执行。作为附加信息，显示成立的暂停寄存器条件（寄存器名（Register）和值（Value））。
Break Sequence (PC:H'nnnnnnnn)	因暂停顺序条件成立而中止了执行。作为附加信息，显示成立的暂停顺序条件（最后的指令地址（PC））。
I/O DLL Stop	外围功能停止。
PC Breakpoint (PC:H'nnnnnnnn)	因断点条件成立而中止了执行。作为附加信息，显示成立的断点条件（指令地址（PC））。
Step Normal End	单步执行正常结束。
Stop	因 [STOP] 按钮而中止了执行。
Trace Buffer Full	因在 [Trace buffer full handling]（跟踪缓冲器满时的处理）对话框的 [Trace Acquisition]（跟踪获取）中选择了 [Stop]（停止）模式并且跟踪缓冲器满而中止了执行。
WAIT Instruction	因 WAIT 指令而中止了指令的执行。

6.2 错误信息

仿真 / 调试程序输出错误信息，将用户程序和操作的错误通知用户。输出的错误信息如表 6.2 所示。

表 6.2 错误信息一览表

信息	内容和对策	
Undefined Instruction Exception	在未定义指令的异常处理中发生了错误。	
Privilege Instruction Exception	在特权指令的异常处理中发生了错误。	
Floating-point Exception	在浮点异常处理中发生了错误。	
Reset Exception	在复位异常处理中发生了错误。	
Interrupt Exception	在中断处理中发生了错误。	
INT Instruction Exception	在无条件的陷阱（INT 指令）的异常处理中发生了错误。	
BRK Instruction Exception	在无条件的陷阱（BRK 指令）的异常处理中发生了错误。	
I/O area not exist	删除了 I/O 区域。 必须设置 I/O 区域。	
I/O DLL Illegal Interrupt Information (errNum=2xx)	这是非法中断信息。[errNum] 表示详细信息。必须修改中断信息。	
	errNum	说明
	200	指定的向量超出范围。
	201	指定的优先级超出范围。
I/O DLL Memory Access Error (errNum=0xx, Address=0XXXXXXXX)	在存取外围功能的存储器时发生了错误。[errNum] 表示详细内容，[Address] 表示发生错误的地址。必须根据错误信息修改程序。	
	errNum	说明
	001	指定的地址超出范围。
	002	指定的区域没有存储器。
	003	不能确保需要的存储器。
	004	指定的数据长度超出范围。
	005	不能存取指定的地址。
I/O DLL Register Access Error (errNum=1xx, RegisterName=xxxx)	在存取外围功能的寄存器时发生了错误。[errNum] 表示详细内容，[RegisterName] 表示发生错误的寄存器。必须根据错误信息修改程序。	
	errNum	说明
	100	寄存器的记述有错。
	101	指定的数值非法。
Memory Access Error (Address:H'nnnnnnnn)	出现了以下某种状态。作为附加信息，显示发生错误的地址（Address）。 1. 存取了没有确保的存储区。 2. 写了只读存储器。 3. 读了只写存储器。 4. 存取了没有存储器的区域。 必须确保存储器和改变属性，或者修改用户程序，使之不存取该存储器。	
System Call Error	发生了仿真 I/O 错误。 必须修改寄存器 R1、寄存器 R2 和参数块内容的错误。	
The memory resource has not been set up	在存储器映像范围外设置了存储器资源。 必须修改存储器资源的设置，使之不发生错误。	

7. 使用指南

7.1 调试的准备

使用示例程序说明仿真 / 调试程序的主要特点。

注意

本章的使用例子（图）的内容可能因客户使用的编译程序版本而不同，请谅解。

7.1.1 示例程序

示例程序是用 C 程序编写的，将 10 个随机数据按升序进行排序后再按降序进行排序。

示例程序进行以下处理：

- 由 main 函数生成要排序的随机数据。
- 由 sort 函数输入数组（保存由 main 函数生成的随机数据的数组）并且按升序进行排序。
- 由 change 函数输入由 sort 函数生成的数组并且按降序进行排序。
- 用 printf 函数显示随机数据和被排序后的数据。

示例程序使用 HEW 的示范程序。

7.1.2 示例程序的创建

必须注意以下事项，创建 HEW 的示范程序。

- 必须在“创建新工作空间”中给 [Project Type]（工程类型）指定“Demonstration”。
- 必须给 [CPU Series:]（CPU 系列:）选择“RX600”，给 [Target:]（目标:）选择“RX600 Simulator”。
- 必须在创建前通过工具栏，给配置选择“SimDebug_RX600”。
- 必须通过工具栏给会话选择“SimSessionRX600”。
- 此示范程序不使用外围功能，因此在切换会话时显示的 [Set Peripheral Function Simulation]（设置外围功能仿真）对话框中，必须在选定 [Don't show this dialog box]（不显示此对话框）复选框后单击 [OK] 按钮。

为了说明调试功能，Demonstration 设置为不优化。请不要更改优化的设置。

7.2 调试的设置

7.2.1 存储器资源的确保

为了执行开发的应用程序，需要确保存储器资源。示范工程自动确保存储器资源，请确认设置。

- 请从 [Setup]（设置）菜单中选择 [Simulator->Memory Resource...]（仿真程序 -> 存储器资源...），显示当前的存储器资源。

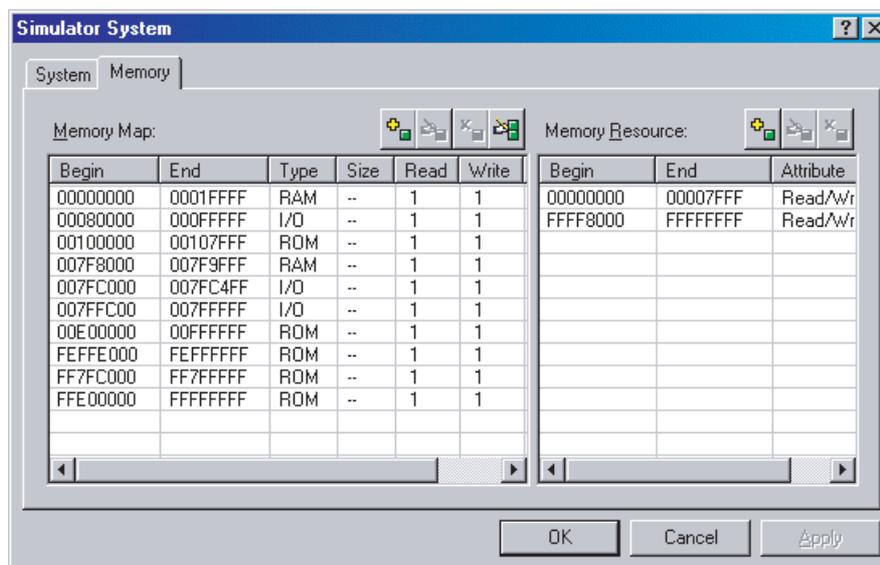


图 7.1 [Simulator System] (仿真系统) 对话框 (存储器页面)

将程序区的 H'FFFF8000 ~ H'FFFFFFF 和数据区的 H'00000000 ~ H'00007FFF 确保为可读写区。

- 请单击 [OK] 按钮，关闭对话框。

也能在 [RX Standard Toolchain] (RX 标准工具链) 对话框的 [Debugger] (调试程序) 页面中参照并更改存储器资源，反映两个页面中更改的内容。

7.2.2 示例程序的下载

示范工程自动设置要下载的程序，请确认设置。

- 请从[Debug]（调试）菜单中选择[Debug Settings...]（调试设置...），打开[Debug Settings...]（调试设置）对话框。

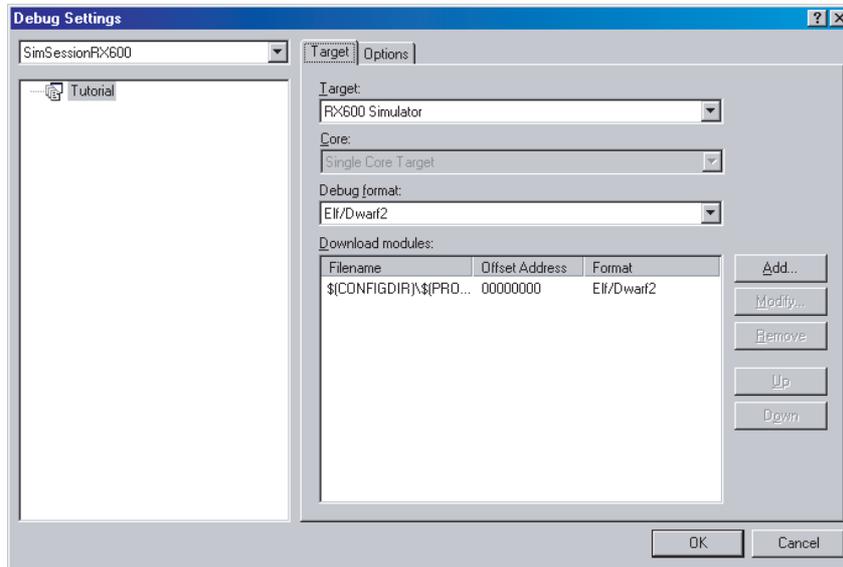


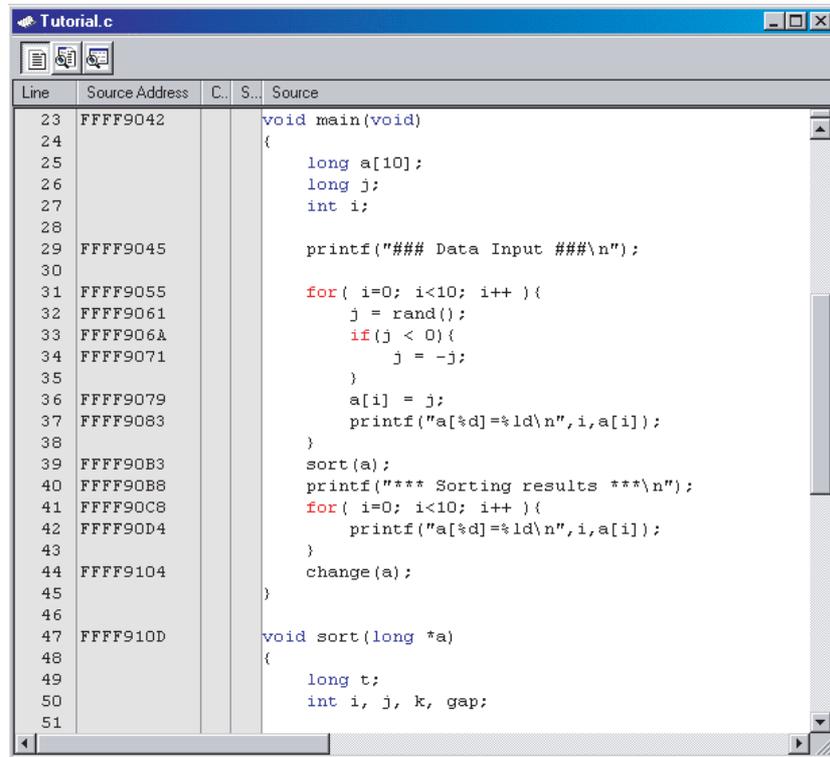
图 7.2 [Debug Setting]（调试设置）对话框

- [Download Modules]（下载模块）设置的文件是要下载的文件。
- 请单击[OK]按钮，关闭[Debug Settings...]（调试设置...）对话框。
- 请从[Debug]（调试）菜单中选择[Download Modules->All Download Modules]（下载模块 -> 下载全部模块），下载示例程序。

7.2.3 源程序的显示

HEW 能调试源级的程序。请显示源程序（“Tutorial.c”）。

- 请双击 [Workspace]（工作空间）窗口中的 Tutorial.c，打开 [Source]（源）窗口。



```
Tutorial.c
Line Source Address C. S... Source
23 FFFF9042 void main(void)
24 {
25     long a[10];
26     long j;
27     int i;
28
29 FFFF9045     printf("### Data Input ###\n");
30
31 FFFF9055     for( i=0; i<10; i++ ){
32 FFFF9061         j = rand();
33 FFFF906A         if(j < 0){
34 FFFF9071             j = -j;
35
36 FFFF9079             a[i] = j;
37 FFFF9083             printf("a[%d]=%ld\n",i,a[i]);
38
39 FFFF90B3         }
40 FFFF90B8         sort(a);
41 FFFF90C8         printf("*** Sorting results ***\n");
42 FFFF90D4         for( i=0; i<10; i++ ){
43
44 FFFF9104             printf("a[%d]=%ld\n",i,a[i]);
45
46
47 FFFF910D         }
48         change(a);
49     }
50
51 void sort(long *a)
52 {
53     long t;
54     int i, j, k, gap;
```

图 7.3 [Source]（源）窗口（显示源程序）

7.2.4 PC 断点的设置

能通过 [Source]（源）窗口简单地设置断点。用以下方法在调用 sort 函数的位置设置断点。

- 请将光标移动到 sort 函数调用的行后单击鼠标的右键，选择弹出式菜单的 [Toggle Breakpoint]（切换断点）。

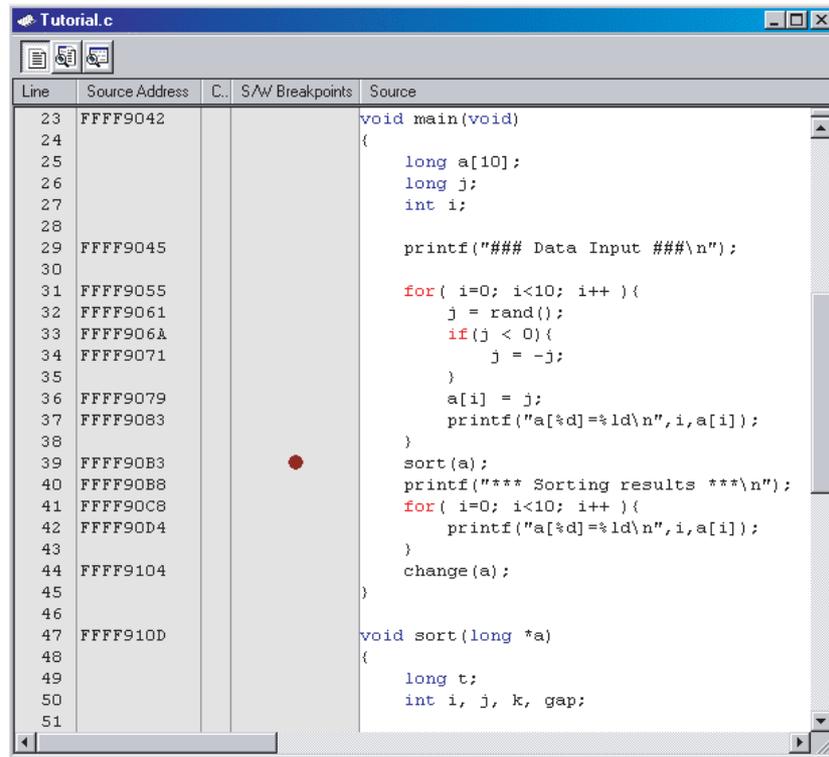


图 7.4 [Source]（源）窗口（设置断点）

在 sort 函数调用行上显示“●”，表示在该地址设置了 PC 断点。

7.2.5 剖析的设置

- 请从[View]（视图）菜单中选择[View->Performance]（视图 -> 性能），打开[Profile]（剖析）窗口。

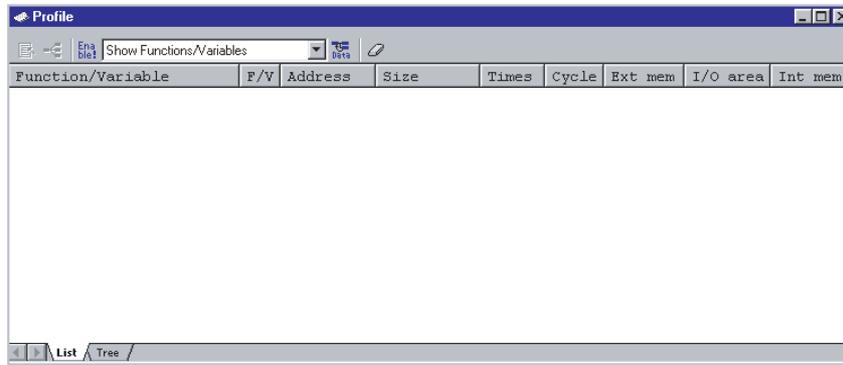


图 7.5 [Profile]（剖析）窗口

- 请在[Profile]（剖析）窗口上单击鼠标的右键，显示弹出式菜单，然后选择[Enable Profiler]（允许剖析器），允许获取剖析信息。

7.2.6 仿真 I/O 的设置

示范工程自动设置仿真 I/O，请确认设置。

- 请从[Setup]（设置）菜单中选择[Simulator->System]（仿真程序 -> 系统），打开[Simulator System]（仿真系统）对话框。

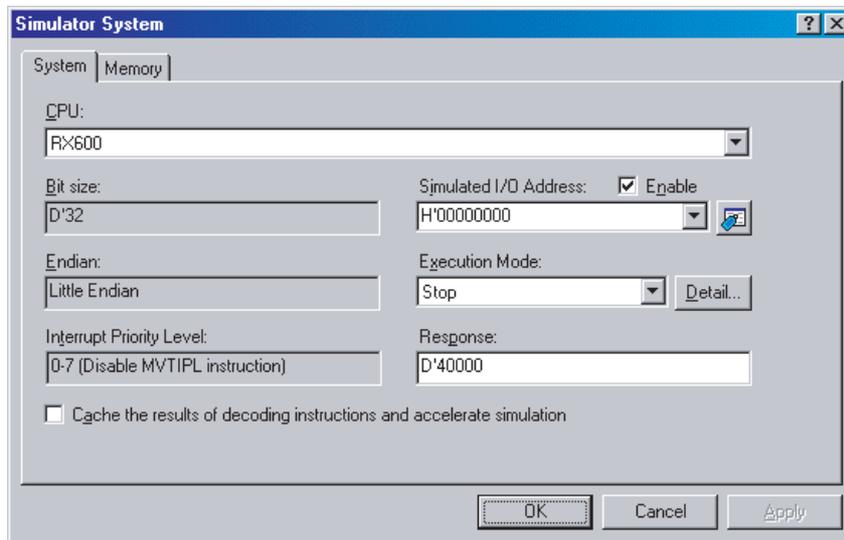


图 7.6 [Simulator System]（仿真系统）对话框（系统选项卡）

- 请确认是否选定了[Simulated I/O Address]（仿真 I/O 地址）的[Enable]（允许）。
- 请单击[OK]按钮，允许 I/O 仿真。
- 请从[View]（视图）菜单中选择[CPU->Simulated I/O]（CPU -> 仿真 I/O），打开[Simulated I/O]（仿真 I/O）窗口。如果不打开[Simulated I/O]（仿真 I/O）窗口，就禁止仿真 I/O。



图 7.7 [Simulated I/O]（仿真 I/O）窗口

7.2.7 跟踪信息获取条件的设置

- 请从[View]（视图）菜单中选择[Code->Trace]（代码 -> 跟踪），打开[Trace]（跟踪）窗口，然后在[Trace]（跟踪）窗口上单击鼠标的右键，显示弹出式菜单，选择[Acquisition...]（获取...）。

显示以下的 [Trace Acquisition]（跟踪获取）对话框。

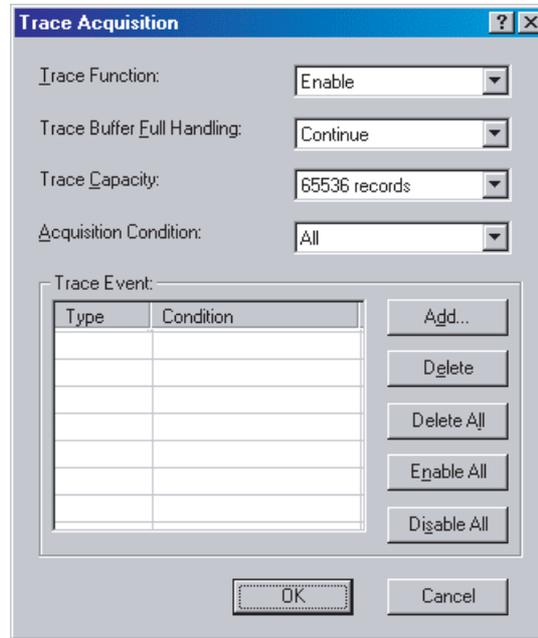


图 7.8 [Trace Acquisition]（跟踪获取）对话框

- 请将[Trace Acquisition]（跟踪获取）对话框的[Trace Function]（跟踪功能）设置为[Enable]（允许），并且单击[OK]按钮，将跟踪信息的获取置为允许。

7.2.8 堆栈指针和程序计数器的设置

必须通过复位向量设置程序计数器，以便执行程序。写在示例程序的复位向量中的 PC 值是 H'FFFF8000。

- 请从[Debug]（调试）菜单中选择[Reset CPU]（复位 CPU）或者单击工具栏上的[Reset CPU]（复位 CPU）按钮。

通过复位向量给程序计数器设置 H'FFFF8000。



图 7.9 CPU 的复位按钮

7.3 调试的开始

7.3.1 程序的执行

- 要执行程序时，请从[Debug]（调试）菜单中选择[Go]（执行）或者单击工具栏上的[Go]（执行）按钮。



图 7.10 执行按钮

程序执行到设有断点的位置为止。在 [Source]（源）窗口中显示 “=>”，表示程序停止的位置。另外，[Output]（输出）窗口显示 “PC Breakpoint (PC:H'FFFF90E4)”，表示停止的原因。

Line	Source Address	C.	SAW Breakpoints	Source
23	FFFF9042			void main(void)
24				{
25				long a[10];
26				long j;
27				int i;
28				
29	FFFF9045			printf("### Data Input ###\n");
30				
31	FFFF9055			for(i=0; i<10; i++){
32	FFFF9061			j = rand();
33	FFFF906A			if(j < 0){
34	FFFF9071			j = -j;
35				}
36	FFFF9079			a[i] = j;
37	FFFF9083			printf("a[%d]=%ld\n", i, a[i]);
38				}
39	FFFF90B3			sort(a);
40	FFFF90B8			printf("### Sorting results ###\n");
41	FFFF90C8			for(i=0; i<10; i++){
42	FFFF90D4			printf("a[%d]=%ld\n", i, a[i]);
43				}
44	FFFF9104			change(a);
45				}
46				
47	FFFF910D			void sort(long *a)
48				{
49				long t;
50				int i, j, k, gap;
51				

图 7.11 [Source]（源）窗口（暂停状态）

能在 [Status] (状态) 窗口中确认停止的原因。

- 请从 [View] (视图) 菜单中选择 [CPU->Status] (CPU -> 状态), 打开 [Status] (状态) 窗口, 然后在 [Status] (状态) 窗口中显示 [Platform] (平台) 表。

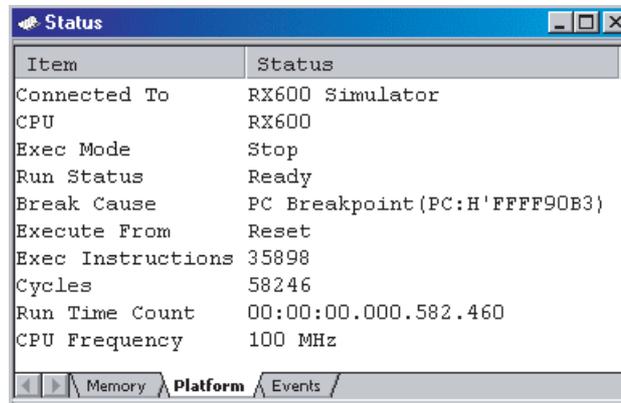


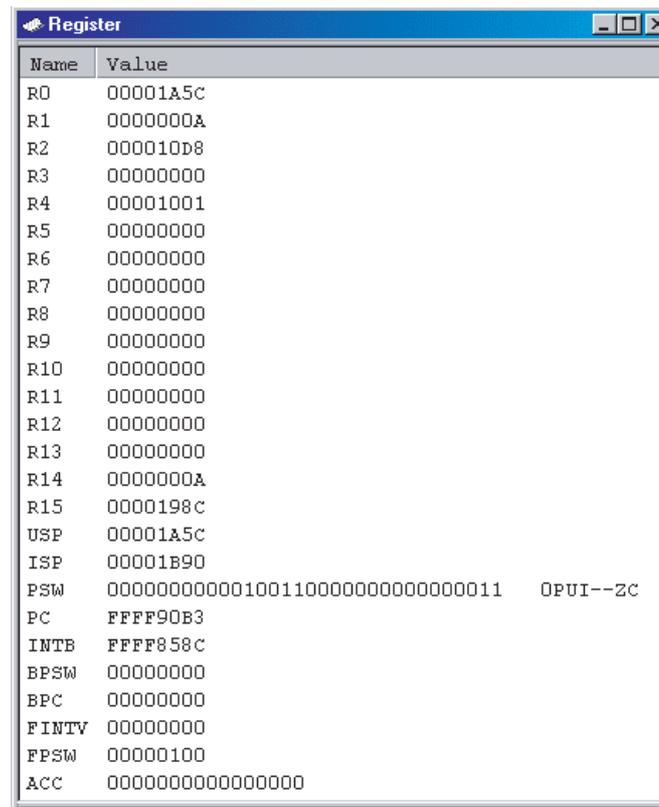
图 7.12 [Status] (状态) 窗口

显示以下的执行内容:

- ① 暂停的原因: PC 暂停
- ② 从复位开始的执行
- ③ 从复位开始的执行指令数: 35898 条指令
- ④ 从复位开始的执行周期数: 58246 个周期
- ⑤ 从复位开始的执行时间: 582.46 微秒
- ⑥ CPU 工作频率: 100MHz

能在 [Register]（寄存器）窗口中确认寄存器的值。

- 请从 [View]（视图）菜单中选择 [CPU->Register]（CPU -> 寄存器）。



Name	Value
R0	00001A5C
R1	0000000A
R2	000010D8
R3	00000000
R4	00001001
R5	00000000
R6	00000000
R7	00000000
R8	00000000
R9	00000000
R10	00000000
R11	00000000
R12	00000000
R13	00000000
R14	0000000A
R15	0000198C
USP	00001A5C
ISP	00001B90
PSW	00000000001001100000000000000011 OPUI--ZC
PC	FFFF90B3
INTB	FFFF858C
BPSW	00000000
BPC	00000000
FINTV	00000000
FPSW	00000100
ACC	0000000000000000

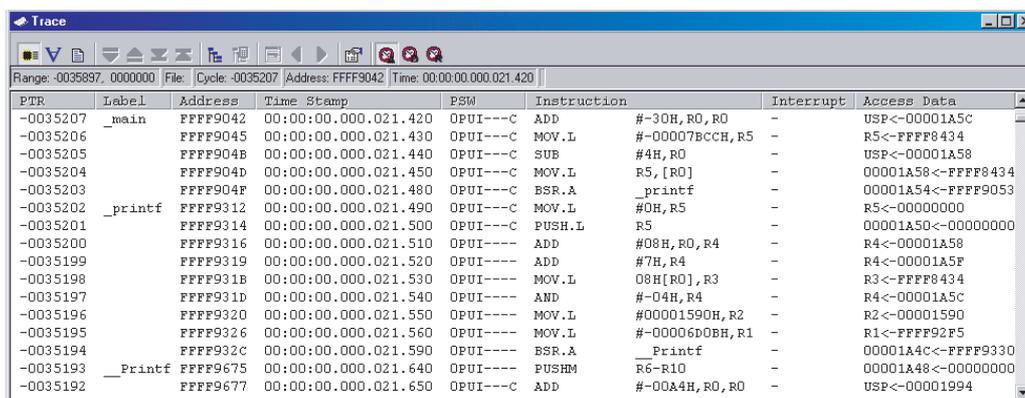
图 7.13 [Register]（寄存器）窗口

能确认程序停止时的各寄存器的值。

7.3.2 跟踪缓冲器的使用方法

能通过跟踪缓冲器知道指令的执行履历。

- 请从 [View]（视图）菜单中选择 [Code->Trace]（代码 -> 跟踪），打开 [Trace]（跟踪）窗口，然后将窗口向上滚动到 main() 函数的起始位置为止。



PTR	Label	Address	Time Stamp	PSW	Instruction	Interrupt	Access Data
-0035207	_main	FFFF9042	00:00:00.000.021.420	OPUI---C	ADD #30H, R0, R0	-	USP<-00001A5C
-0035206		FFFF9045	00:00:00.000.021.430	OPUI---C	MOV.L #00007BCC, R5	-	R5<-FFFF8434
-0035205		FFFF9048	00:00:00.000.021.440	OPUI---C	SUB #4H, R0	-	USP<-00001A58
-0035204		FFFF904D	00:00:00.000.021.450	OPUI---C	MOV.L R5, [R0]	-	00001A58<-FFFF8434
-0035203		FFFF904F	00:00:00.000.021.480	OPUI---C	BSR.A _printf	-	00001A54<-FFFF9053
-0035202	_printf	FFFF9312	00:00:00.000.021.490	OPUI---C	MOV.L #0H, R5	-	R5<-00000000
-0035201		FFFF9314	00:00:00.000.021.500	OPUI---C	PUSH.L R5	-	00001A50<-00000000
-0035200		FFFF9316	00:00:00.000.021.510	OPUI---C	ADD #08H, R0, R4	-	R4<-00001A58
-0035199		FFFF9319	00:00:00.000.021.520	OPUI---C	ADD #7H, R4	-	R4<-00001A5F
-0035198		FFFF931B	00:00:00.000.021.530	OPUI---C	MOV.L 08H[R0], R3	-	R3<-FFFF8434
-0035197		FFFF931D	00:00:00.000.021.540	OPUI---C	AND #-04H, R4	-	R4<-00001A5C
-0035196		FFFF9320	00:00:00.000.021.550	OPUI---C	MOV.L #00001590H, R2	-	R2<-00001590
-0035195		FFFF9326	00:00:00.000.021.560	OPUI---C	MOV.L #-00006D0BH, R1	-	R1<-FFFF92F5
-0035194		FFFF932C	00:00:00.000.021.590	OPUI---C	BSR.A _Printf	-	00001A4C<-FFFF9330
-0035193	_Printf	FFFF9675	00:00:00.000.021.640	OPUI---C	PUSHM R6-R10	-	00001A48<-00000000
-0035192		FFFF9677	00:00:00.000.021.650	OPUI---C	ADD #-00A4H, R0, R0	-	USP<-00001994

图 7.14 [Trace]（跟踪）窗口（显示跟踪信息）

7.3.3 跟踪检索的执行

首先，在 [Trace]（跟踪）窗口上单击鼠标的右键，显示弹出式菜单，然后选择 [Find -> Find...]（查找 -> 查找...），打开 [Find]（查找）对话框。

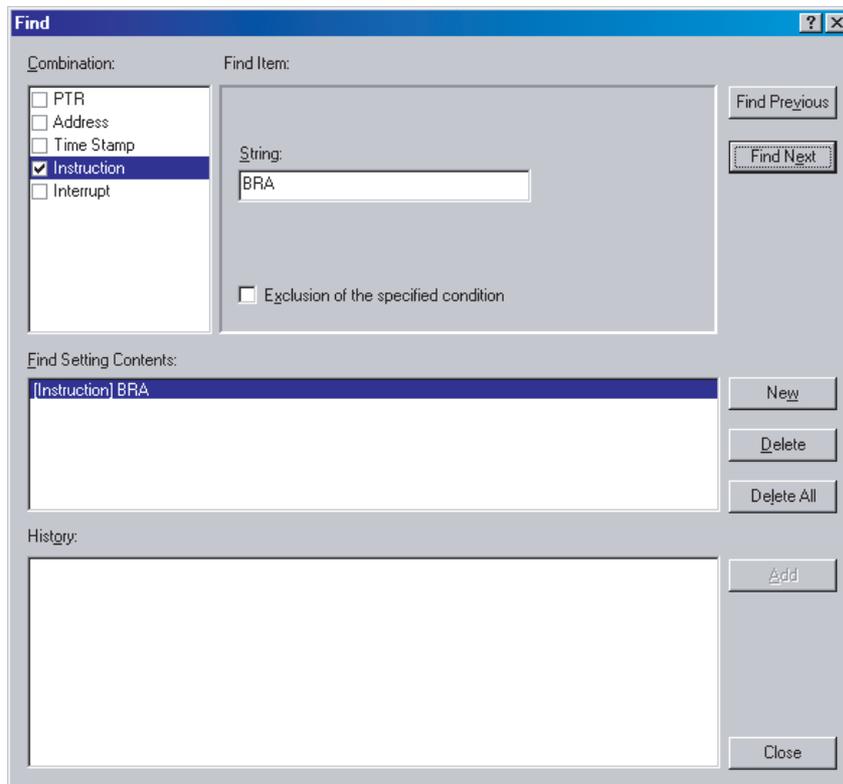


图 7.15 跟踪的检索对话框

请在 [Combination]（组合）栏中选择查找对象的条件后选定复选框。

在 [Find Item]（查找项目）栏中指定与选择条件对应的详细项目。

在 [Find Setting Contents]（查找设置内容）列表框中显示所设置的检索条件。

在设置检索条件后，单击 [Find Previous]（查找上一个）按钮或者 [Find Next]（查找下一个）按钮来开始检索。

如果找到与检索结果相同的记录，就强调显示该记录。

在找到相同记录的情况下，如果在弹出式菜单中选择 [Find Previous]（查找上一个）或者 [Find Next]（查找下一个），就能检索下一个记录。

PTR	Label	Address	Time Stamp	PSW	Instruction	Interrupt	Access Data
-0035886		FFFF935F	00:00:00.000.000.200	0-----	BRA.B next_loop1	-	PC<-FFFF936F
-0035885	next_loo	FFFF936F	00:00:00.000.000.210	0-----	CMP R4,R5	-	
-0035884		FFFF9371	00:00:00.000.000.240	0-----	BGTU.B loop1	-	PC<-FFFF9361
-0035883	loop1	FFFF9361	00:00:00.000.000.250	0-----	MOV.L [R4+],R1	-	R4<-FFFF8578 R1<-0
-0035882		FFFF9364	00:00:00.000.000.270	0-----	MOV.L [R4+],R3	-	R4<-FFFF857C R3<-0
-0035881		FFFF9367	00:00:00.000.000.280	0-----	CMP R1,R3	-	
-0035880		FFFF9369	00:00:00.000.000.290	0-----	BLEU.B next_loop1	-	
-0035879		FFFF936B	00:00:00.000.000.300	0-----	SUB R1,R3	-	R3<-00000458
-0035878		FFFF936D	00:00:00.000.003.100	0-----	SSTR.B	-	0000152F<-00 R1<-
-0035877	next_loo	FFFF936F	00:00:00.000.003.110	0-----	CMP R4,R5	-	
-0035876		FFFF9371	00:00:00.000.003.140	0-----	BGTU.B loop1	-	PC<-FFFF9361
-0035875	loop1	FFFF9361	00:00:00.000.003.150	0-----	MOV.L [R4+],R1	-	R4<-FFFF8580 R1<-0
-0035874		FFFF9364	00:00:00.000.003.170	0-----	MOV.L [R4+],R3	-	R4<-FFFF8584 R3<-0
-0035873		FFFF9367	00:00:00.000.003.180	0-----	ZC CMP R1,R3	-	
-0035872		FFFF9369	00:00:00.000.003.200	0-----	ZC BLEU.B next_loop1	-	PC<-FFFF936F
-0035871	next_loo	FFFF936F	00:00:00.000.003.210	0-----	CMP R4,R5	-	

图 7.16 [Trace]（跟踪）窗口（检索结果）

7.3.4 仿真 I/O 的确认

能在 [Simulated I/O] (仿真 I/O) 窗口中确认由 printf 函数显示的随机数据。

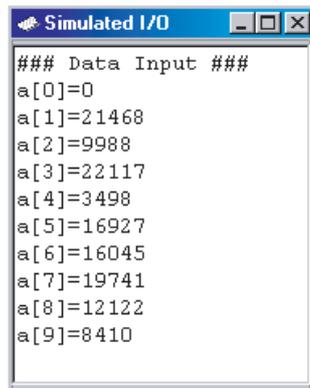


图 7.17 [Simulated I/O] (仿真 I/O) 窗口

- 请不要关闭 [Simulated I/O] (仿真 I/O) 窗口。

7.3.5 断点的确认

能在 [Event] (事件) 窗口中确认由程序设置的全部断点的清单。

- 请从 [View] (视图) 菜单中选择 [Code->Eventpoints] (代码 -> 事件点)。

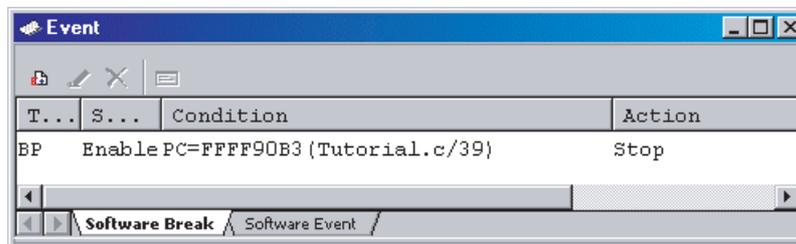


图 7.18 [Event] (事件) 窗口

能通过 [Event] (事件) 窗口设置断点、定义新的断点以及删除断点。
请关闭 [Event] (事件) 窗口。

7.3.6 变量的参照

能在 [Watch] (监视) 窗口中确认程序使用的变量值。例如, 要查看程序开始声明的 long 型的数组 a 的情况:

- 请从 [View] (视图) 菜单中选择 [Symbol->Watch] (符号->监视), 显示 [Watch] (监视) 窗口, 然后在 [Watch] (监视) 窗口上单击鼠标的右键, 显示弹出式菜单, 选择 [Add Watch...] (添加监视...).

显示以下的 [Add Watch] (添加监视) 对话框。

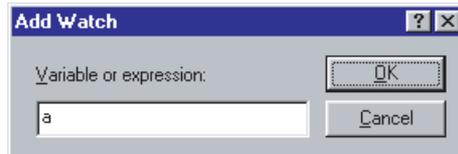


图 7.19 [Add Watch] (添加监视) 对话框

- 键入数组 “a”, 单击 [OK] 按钮。

在 [Watch] (监视) 窗口中显示 long 型的数组 a。

如果单击数组 a 前的符号 “+”, 就展开并显示数组。

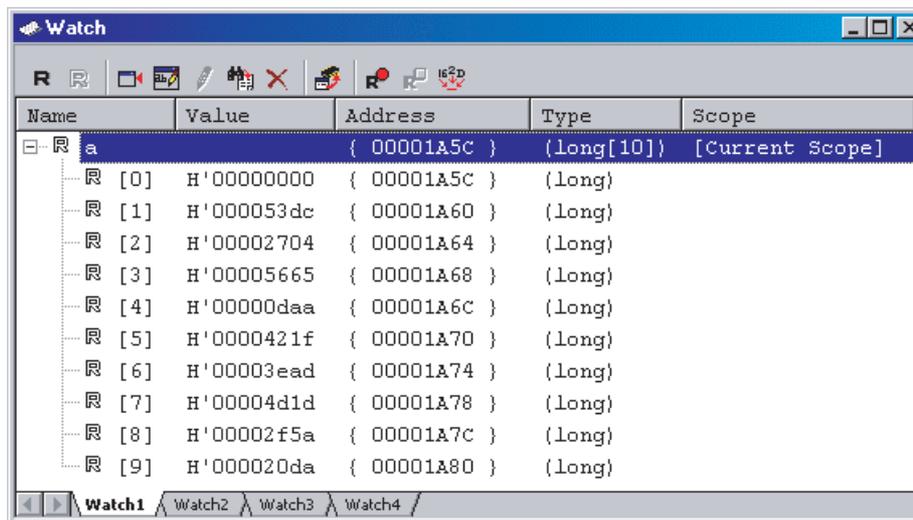


图 7.20 [Watch] (监视) 窗口

- 请关闭 [Watch] (监视) 窗口。

7.3.7 程序的单步执行

仿真 / 调试程序具有对调试程序有效的各种单步菜单。

菜单	说明
Step In (步进)	执行各语句 (包括函数内的语句)。
Step Over (跨步)	将函数调用作为 1 步进行单步执行。
Step Out (步出)	从函数退出, 停止在调用函数的程序中的下一个语句。
Step... (单步...)	以指定的速度进行指定次数的单步执行。

(1) [Step In] (步进) 的执行

[Step In] (步进) 进入调用函数, 停止在调用函数的起始位置。

- 请从 [Debug] (调试) 菜单中选择 [Step In] (步进) 或者单击工具栏的 [Step In] (步进) 按钮, 进入 sort 函数。



图 7.21 步进按钮

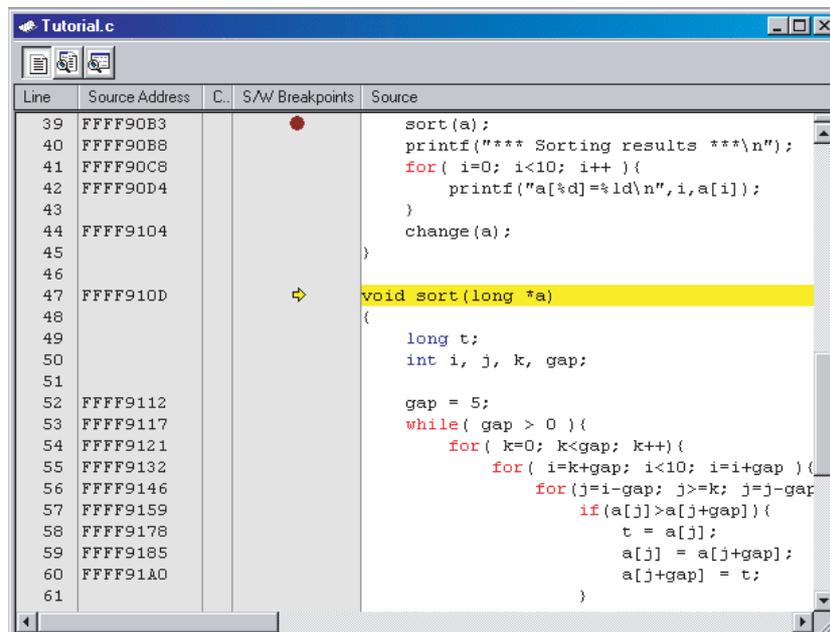


图 7.22 源窗口 (步进)

- [Source] (源) 窗口中指示 PC 位置的标签 “=>” 移动到 sort 函数的起始位置。

(2) [Step Out] (步出) 的执行

[Step Out] (步出) 从函数退出，停止在调用源程序的下一个语句。

- 请从[Debug] (调试) 菜单中选择[Step Out] (步出) 或者单击工具栏的[Step Out] (步出) 按钮，从 sort 函数退出。



图 7.23 步出按钮

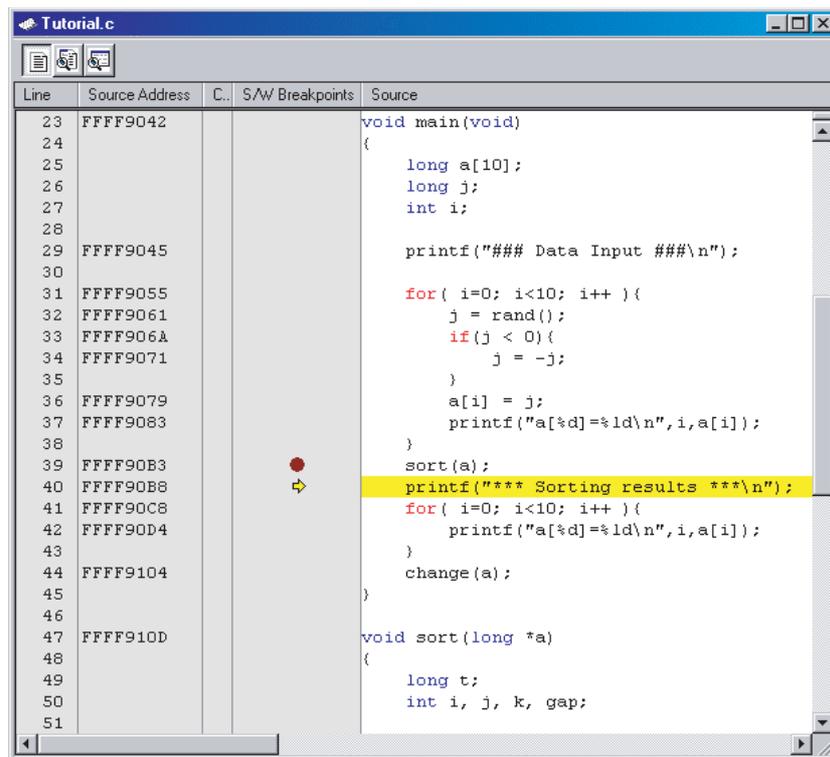


图 7.24 源窗口 (步出)

(3) [Step Over]（跨步）的执行

[Step Over]（跨步）将函数调用作为 1 步进行单步执行，停止在主程序中的下一个语句。

请从 [Debug]（调试）菜单中选择 [Step Over]（跨步）或者单击工具栏的 [Step Over]（跨步）按钮，单步执行一次 Printf 函数中的语句。



图 7.25 跨步按钮

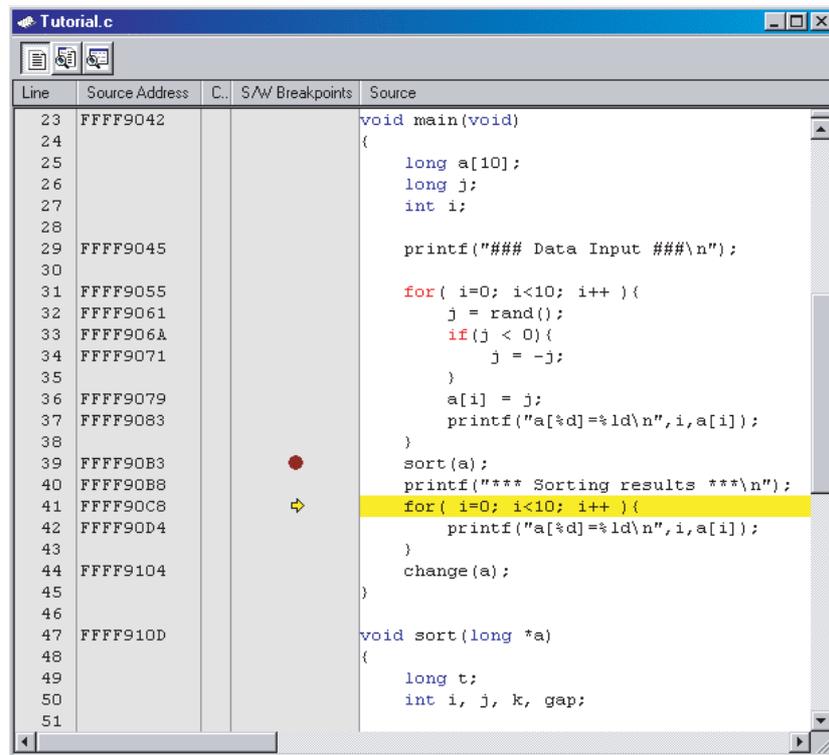


图 7.26 源窗口（跨步）

如果执行 printf 函数，就在 [Simulated I/O]（仿真 I/O）窗口中显示 “*** Sorting results ***”。

7.3.8 剖析信息的确认

能在 [Profile] (剖析) 窗口确认剖析信息。

- 如果单击 [Go] (执行) 按钮，从当前的 PC 继续执行，就在执行 BRK 指令后停止。

(1) [List] 表

以列表格式显示剖析信息。

- 请从 [View] (视图) 菜单中选择 [Performance->Profile] (性能 -> 剖析)，打开 [Profile] (剖析) 窗口，显示 [List] 表。

Function/Variable	F/V	Address	Size	Times	Cycle	Ext mem	I/O area	Int mem
_fwrite	F	FFFF9399	H'000000CF	183	24459	0	0	6348
_INIT\$CT	F	FFFF934F	H'00000000	1	969	0	0	32
_rand	F	FFFF9333	H'0000001C	10	110	0	0	30
_printf	F	FFFF9312	H'00000021	22	374	0	0	198
FFFF92F5	F	FFFF92F5	H'00000000	183	3477	0	0	2013
__fclose	F	FFFF92A2	H'00000053	3	120	0	0	39
_freopen	F	FFFF9274	H'0000002E	3	93	0	0	60
_change	F	FFFF920A	H'0000006A	1	424	0	0	166
_sort	F	FFFF910D	H'000000FD	1	1869	0	0	774
_main	F	FFFF9042	H'000000CB	1	717	0	0	271
_write	F	FFFF8EE2	H'0000008B	249	15438	0	0	5478
_close	F	FFFF8ED9	H'00000009	3	21	0	0	6
_open	F	FFFF8E44	H'00000095	3	192	0	0	39
_CLOSEALL	F	FFFF8DF7	H'0000004D	1	470	0	0	144
_INIT_IOLIB	F	FFFF8CD8	H'0000011F	1	89	0	0	31
_charput	F	FFFF8C98	H'00000000	249	2739	0	0	747

图 7.27 [List] 表 (剖析窗口)

从此窗口能知道调用了 3 次 __fclose 函数，执行周期为 120 个周期，存取了 39 次内部存储器。能找到调用次数多的函数和多次存取低速存储器的函数等的程序性能的临界路径。

(2) [Tree] 表

以树的格式显示剖析信息。

- 请选择 [Tree] 表。通过双击函数名，展开或者折叠树结构。

Function	Address	Size	Stack Size	Times	Cycle	Ext mem	I/O area	Int mem
_main	FFFF9042	H'000000CB	H'00000000	1	717	0	0	271
__printf	FFFF9312	H'00000021	H'00000000	22	374	0	0	198
__rand	FFFF9333	H'0000001C	H'00000000	10	110	0	0	30
__change	FFFF920A	H'0000006A	H'00000000	1	424	0	0	166
__sort	FFFF910D	H'000000FD	H'00000000	1	1869	0	0	774
__CLOSEALL	FFFF8DF7	H'0000004D	H'00000000	1	470	0	0	144
__INIT_IOLIB	FFFF8CD8	H'0000011F	H'00000000	1	89	0	0	31
__freopen	FFFF9274	H'0000002E	H'00000000	3	93	0	0	60
__fclose	FFFF92A2	H'00000053	H'00000000	3	120	0	0	39
__fflush	FFFF9468	H'0000007E	H'00000000	3	54	0	0	12
__fopen	FFFF9624	H'00000051	H'00000000	3	66	0	0	36
__close	FFFF8ED9	H'00000009	H'00000000	3	21	0	0	6
__fopen	FFFF94E6	H'000000E8	H'00000000	3	421	0	0	87

图 7.28 [Tree] 表 (剖析窗口)

从此窗口能知道 __fclose 函数调用了 3 次 __close 函数，此时的执行周期为 21 个周期，存取了 6 次内部存储器。

(3) 剖析图窗口

在 [Profile-Chart] (剖析图) 窗口中显示函数的调用关系。

- 请在 [Profile] (剖析) 窗口上选择 `__fclose` 函数，然后单击鼠标的右键，显示弹出式菜单并选择 [View Profile-Chart] (查看剖析图)，显示 [Profile-Chart] (剖析图) 窗口。

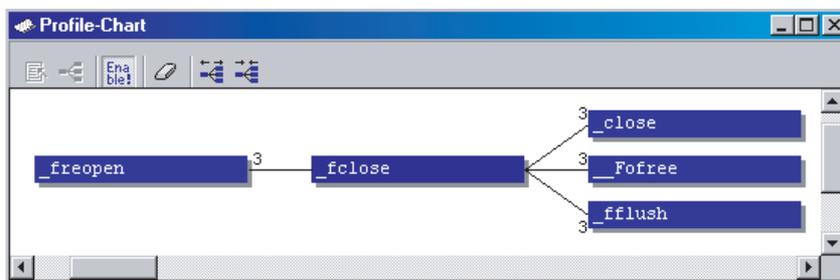


图 7.29 [Profile-Chart] (剖析图) 窗口

从此窗口能知道 `_freopen` 函数调用了 3 次 `__fclose` 函数，并且调用了 3 次 `_close` 函数。

以上是仿真 / 调试程序的使用指南。

修订记录	RX 族 仿真 / 调试程序 V.1.01 用户手册
------	----------------------------

Rev.	发行日	修订内容	
		页	修订处
1.00	2011.02.02	—	初版发行

RX 族 仿真 / 调试程序 V.1.01
用户手册

Publication Date: Rev.1.00 Feb 2, 2011

Published by: Renesas Electronics Corporation

**SALES OFFICES**

Renesas Electronics Corporation

<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
7F, No. 363 Fu Shing North Road Taipei, Taiwan, R.O.C.
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RX族 仿真/调试程序 V.1.01



瑞萨电子株式会社

R20UT0454CJ0100