

致尊敬的顾客

关于产品目录等资料中的旧公司名称

NEC电子公司与株式会社瑞萨科技于2010年4月1日进行业务整合（合并），整合后的新公司暨“瑞萨电子公司”继承两家公司的所有业务。因此，本资料中虽还保留有旧公司名称等标识，但是并不妨碍本资料的有效性，敬请谅解。

瑞萨电子公司网址：<http://www.renesas.com>

2010年4月1日
瑞萨电子公司

【发行】瑞萨电子公司（<http://www.renesas.com>）

【业务咨询】<http://www.renesas.com/inquiry>

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.

R0E530640MCU00

M16/64组的E100仿真器MCU单元

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
 - (1) artificial life support devices or systems
 - (2) surgical implantations
 - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
 - (4) any other purposes that pose a direct threat to human lifeRenesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

注意

本文只是参考译文，前页所载英文版“Cautions”具有正式效力。

关于利用本资料时的注意事项

1. 本资料是为了让用户根据用途选择合适的本公司产品的参考资料，对于本资料中所记载的技术信息，并非意味着对本公司或者第三者的知识产权及其他权利做出保证或对实施权力进行的承诺。
2. 对于因使用本资料所记载的产品数据、图、表、程序、算法及其他应用电路例而引起的损害或者对第三者的知识产权及其他权利造成侵犯，本公司不承担任何责任。
3. 不能将本资料所记载的产品和技术用于大规模破坏性武器的开发等目的、军事目的或其他的军需用途方面。另外，在出口时必须遵守日本的《外汇及外国贸易法》及其他出口的相关法令并履行这些法令中规定的必要手续。
4. 本资料所记载的产品数据、图、表、程序、算法以及其他应用电路例等所有信息均为本资料发行时的内容，本公司有可能在未做事先通知的情况下，对本资料所记载的产品或者产品规格进行更改。所以在购买和使用本公司的半导体产品之前，请事先向本公司的营业窗口确认最新的信息并经常留意本公司通过公司主页 (<http://www.renesas.com>)等公开的最新信息。
5. 对于本资料中所记载的信息，制作时我们尽力保证出版时的精确性，但不承担因本资料的叙述不当而致使顾客遭受损失等的任何相关责任。
6. 在使用本资料所记载的产品数据、图、表等所示的技术内容、程序、算法及其他应用电路例时，不仅要对所使用的技术信息进行单独评价，还要对整个系统进行充分的评价。请顾客自行负责，进行是否适用的判断。本公司对于是否适用不负任何责任。
7. 本资料中所记载的产品并非针对万一出现故障或是错误运行就会威胁到人的生命或给人体带来危害的机器、系统(如各种安全装置或者运输交通用的、医疗、燃烧控制、航天器械、核能、海底中继用的机器和系统等)而设计和制造的,特别是对于品质和可靠性要求极高的机器和系统等(将本公司指定用于汽车方面的产品用于汽车时除外)。如果要用于上述的目的，请务必事先向本公司的营业窗口咨询。另外，对于用于上述目的而造成的损失等，本公司概不负责。
8. 除上述第7项内容外，不能将本资料中记载的产品用于以下用途。如果用于以下用途而造成的损失，本公司概不负责。
 - 1) 生命维持装置。
 - 2) 植埋于人体使用的装置。
 - 3) 用于治疗(切除患部、给药等)的装置。
 - 4) 其他直接影响到人的生命的装置。
9. 在使用本资料所记载的产品时，对于最大额定值、工作电源电压的范围、放热特性、安装条件及其他条件请在本公司规定的保证范围内使用。如果超出了本公司规定的保证范围使用时，对于由此而造成的故障和出现的事故，本公司将不承担任何责任。
10. 本公司一直致力于提高产品的质量和可靠性，但一般来说，半导体产品总会以一定的概率发生故障、或者由于使用条件不同而出现错误运行等。为了避免因本公司的产品发生故障或者错误运行而导致人身事故和火灾或造成社会性的损失，希望客户能自行负责进行冗余设计、采取延烧对策及进行防止错误运行等的安全设计(包括硬件和软件两方面的设计)以及老化处理等，这是作为机器和系统的出厂保证。特别是单片机的软件，由于单独进行验证很困难，所以要求在顾客制造的最终的机器及系统上进行安全检验工作。
11. 如果把本资料所记载的产品从其载体设备上卸下，有可能造成婴儿误吞的危险。顾客在将本公司产品安装到顾客的设备上时，请顾客自行负责将本公司产品设置为不容易剥落的安全设计。如果从顾客的设备上剥落而造成事故时，本公司将不承担任何责任。
12. 在未得到本公司的事先书面认可时，不可将本资料的一部分或者全部转载或者复制。
13. 如果需要了解关于本资料的详细内容，或者有其他关心的问题，请向本公司的营业窗口咨询。

前言

R0E530640MCU00 是 M16C/60 系列 M16C/64 组 MCU 的全规格仿真器。本用户手册主要介绍 R0E530640MCU00 的规格以及如何设置它。有关本产品中包括的仿真调试器的详细信息，请参考本产品的用户手册。

表 1.1 套装组件（第 1 页）中显示了本产品的所有组件。如果对本产品有任何问题或疑问，请联系当地的经销商。

下面列出了与使用本产品相关的手册。可以从瑞萨工具主页 (<http://www.renesas.com/tools>) 下载最新手册。

相关手册

项目	手册
附件	R0E0100TNPJ00 用户手册
	R0E0100TNPJK00 用户手册
集成开发环境	HEW 用户手册
仿真调试器	R0E530640MCU00 用户手册
C 编译器	用于 R8C/Tiny、M16C/60、30、Tiny、20 和 10 系列的 C 编译器套件 C 编译器用户手册
汇编器	用于 R8C/Tiny、M16C/60、30、Tiny、20 和 10 系列的 C 编译器套件 汇编器用户手册

重要事项

使用本产品前，请务必仔细阅读本用户手册。

务必保管好用户手册，在使用中如有不明之处，请再次阅读。

仿真器：

本资料中的仿真器是指瑞萨科技公司生产的以下产品：

- (1) E100 仿真器主部件
- (2) MCU 单元
- (3) 用于连接用户系统的节距转换器板

不包含客户的用户系统和主机

仿真器的使用目的：

本仿真器是用来进行系统开发的器件，用于开发使用瑞萨 16 位单片 MCU 的 M16C 族 M16C/60 系列 M16C/64 组的系统。它从软件和硬件两方面为系统开发提供支持。

请遵循该使用目的正确使用本产品，不要将其用于其他目的。

仿真器的使用对象：

仔细阅读并理解用户手册后，才能使用本仿真器。

用户需了解电子电路、逻辑电路和 MCU 的基本知识。

使用仿真器时：

- (1) 本产品是在程序开发阶段和评估阶段使用的开发支持装置。批量生产所开发的程序时，用户必须事先进行全面测试、评估，或通过其他试验自行判断该程序是否适用。
- (2) 对于使用本仿真器所产生的任何后果，瑞萨科技公司概不负责。
- (3) 如有产品缺陷，瑞萨科技公司将努力改进产品或提供有偿或无偿的解决办法。但瑞萨科技公司不保证在所有情况下都能改进产品或提供解决办法。
- (4) 本仿真器是为在实验室进行的程序开发和评估而设计的。因此，在日本使用时，本产品不在电气设备及材料安全法 (Electrical Appliance and Material Safety Law) 管辖范围之内，也不保证能防电磁干扰。
- (5) 瑞萨科技公司无法预测所有可能存在危险的状况或使用不当的情况。因此，本用户手册中的警告以及仿真器上所贴的警告标签不能涵盖所有危险情况。请客户正确安全使用本产品并自行承担相关责任。
- (6) 本产品尚未取得 UL 等安全规格和 IEC 等行业规格方面的认证。因此，如果将本产品从日本国内携带到海外，请了解这一点。

使用限制：

本产品是作为系统开发支持工具而开发。因此，请不要将其用于设备中内嵌的应用程序，也不要用于如下所示的系统或设备开发用途：

- (1) 运输和交通车辆
- (2) 医疗（用于涉及生命安全的装置）
- (3) 航空航天
- (4) 原子能控制
- (5) 海底中继器

因上述目的而考虑使用本产品的客户，请向当地的经销商咨询。

关于产品的变更:

本公司一直在不断改良本产品的设计和性能。因此，在更改规格、设计和用户手册时，恕不另行通知。

关于权利:

- (1) 如因使用本用户手册中所述的任何信息、产品或者电路而引起损害，或导致侵犯专利权等其他权利，本公司概不承担任何责任。
- (2) 本用户手册中的任何信息或资料不以暗示手段或其他方式允许用户使用本公司或第三方的专利权或任何其他权利。
- (3) 本用户手册和仿真器产品享有著作权保护，瑞萨保留所有权利。事先未经本公司的书面许可，不得翻印、复制和转载本用户手册的部分或者全部内容。

关于图示:

本用户手册的部分图示可能和实物有差异。

安全事项

图标的定义

在用户手册和产品自身中，使用了若干图标来确保正确操作本产品，并确保避免对用户或其他人员造成人身伤害或财产损失。

本章介绍为安全、正确地使用本产品而应注意的事项。使用本产品前，请务必阅读本章。



该符号表示安全警告，用于提醒用户：存在可能造成人身伤害的潜在危险。为了避免可能的伤害或死亡，请务必查看该符号旁边的安全消息。



危险

“危险”表示紧急危险情况，如果不避免，会导致死亡或重伤。但是，对于本用户手册中介绍的产品，还没有这种危险情况。



警告

“警告”表示潜在危险情况，如果不避免，会导致死亡或重伤。



注意

“注意”表示潜在危险情况，如果不避免，会导致轻伤或中等伤害。


注意

不带安全警告符号的“注意”表示潜在危险情况，如果不避免，将导致财产损失。


重要事项

该图标在操作过程或说明性描述中使用，用于向用户传达异常情况或注意信息。


除了上面五种图标，还会根据情况适当使用以下图标。


 表示“警告”或“注意”。

例如： “注意电击”

 表示“禁止”。

例如： “禁止拆卸”

 表示“强制操作”。

例如： “从插座拔下电源电缆”

警告

AC 电源警告：



- 如果附带的 AC 电源电缆与插座不匹配，请不要改变 AC 电源电缆，也不要强行插入，否则，可能导致电击和 / 或火灾。
- 应使用遵从所在国家或地区的安全标准的 AC 电源电缆。
- 手湿的时候不要接触 AC 电源电缆的插头，否则可能导致电击。
- 本产品的信号接地和机架接地相连。如果正在开发的产品是不变压的（没有 AC 电源的隔离变压器），这可能导致电击。另外，这可能对本产品和用户正在开发的产品造成无法修复的损害。在开发过程中，请通过隔离变压器连接产品的 AC 电源与市电，以避免这些危险。
- 如果将其他设备连接到相同的分支电路，应小心，不要使电路过载。
- 安装本设备时，请务必保持可靠的接地连接。
- 该电缆的额定电压是 125V。连接到大于 125V 的电源时，请使用适合该电压的电缆。



- 如果闻到奇怪的气味、听到异常声音，或看到本产品冒烟，请立即从插座中拔出 AC 电源电缆，以断开电源。
此时不能继续使用，因为存在电击和 / 或失火的危险。发生这种情况时，请联系当地的经销商。
- 设置本仿真器并将其连接到其他设备之前，请关闭电源或拔下电源电缆，以避免伤害用户或损坏产品。

关于本产品的警告：



- 不要拆卸或修改本产品。如果拆卸或更改本产品，可能会发生电击而导致人身伤害。如果拆卸或更改本产品，将不予以保修。
- 确保没有任何东西掉在上面板上的冷却风扇中，特别是液体、金属物品或任何易燃物。

安装时的警告：



- 不要将本产品放在水中或湿度很高的地方。请务必不要将本产品弄湿。产品中溅入水或其他液体可能导致无法修复的损坏。

关于使用环境的警告：



- 使用本设备时，最高环境温度为 35 °C。请小心不要超过该温度。

 **注意**

打开电源时的注意事项：



- 尽可能同时打开 / 关闭仿真器和用户系统的电源。
- 在关闭电源之后再次打开电源之前，请等待约 10 秒钟。

操作本产品时的注意事项：



- 操作主部件时应小心，不要使其受到机械冲击。
- 不要直接触摸仿真器主部件的连接器引脚和目标 MCU 连接器引脚。静电可能会损坏内部电路。
- 不要借助通信接口电缆或柔性电缆来拉拽本仿真器。而且，过度弯曲或用力可能会折断导体。
- 不要过度弯曲柔性电缆，否则，电缆可能折断。
- 不要在本设备中使用英制尺寸螺钉。本设备中使用的螺钉都是 ISO（公制尺寸）类型的螺钉。更换螺钉时，请使用与以前配备类型相同的螺钉。

关于系统故障的注意事项：



- 如果仿真器由于外部噪音等干扰而出现功能异常，请立即关闭仿真器，然后重新启动。

目 录

第 1 章	概要	1
1.1	套装组件	1
1.2	进行开发所需的其他工具产品	1
1.3	系统配置	2
1.3.1	系统配置	2
1.3.2	仿真器各部分的名称和功能	3
1.4	规格	5
1.5	操作环境	6
第 2 章	设置	7
2.1	启动仿真器的流程图	7
2.2	安装附带的软件	9
2.3	将 MCU 单元连接到 E100 仿真器主部件或断开它们之间的连接	10
2.4	连接主机	11
2.5	连接仿真器电源	12
2.6	打开电源	13
2.6.1	检查仿真器系统的连接	13
2.6.2	打开 / 关闭电源	13
2.7	自检	14
2.8	选择时钟提供	15
2.8.1	时钟	15
2.8.2	使用内部振荡器电路板	16
2.8.3	在用户系统中使用振荡器电路	17
2.8.4	使用内部生成器电路	18
2.9	连接用户系统	18
2.9.1	连接到 100 引脚 0.65mm 节距支脚模式	19
2.9.2	连接到 100 引脚 0.5mm 节距支脚模式	20
第 3 章	教程	21
3.1	简介	21
3.2	启动 HEW	22
3.3	连接仿真器	22
3.4	下载教程程序	23
3.4.1	下载教程程序	23
3.4.2	显示源程序	24
3.5	设置软件断点	25
3.6	执行程序	26
3.6.1	复位 CPU	26
3.6.2	执行程序	26
3.7	检查断点	27
3.7.1	检查断点	27
3.8	更改寄存器内容	28
3.9	引用符号	29
3.10	检查存储器内容	30
3.11	引用变量	31
3.12	显示局部变量	33
3.13	单步执行程序	33
3.13.1	执行 [Step In] (跳入) 命令	34
3.13.2	执行 [Step Out] (跳出) 命令	35

3.13.3	执行 [Step Over] (跳过) 命令	36
3.14	强行中断程序	37
3.15	硬件断点工具	37
3.15.1	在程序执行指定地址时停止程序	37
3.16	在程序存取存储器时停止程序	38
3.17	跟踪工具	39
3.17.1	显示由 Fill Until Stop (填充直至停止) 获取的跟踪信息	40
3.17.2	显示由 Fill around TP (围绕 TP 填充) 获取的跟踪信息	43
3.17.3	显示函数执行历史记录	45
3.17.4	筛选工具	47
3.18	堆栈跟踪工具	49
3.19	后续内容	50
第 4 章	准备调试	51
4.1	启动 HEW	51
4.2	建立新的工作空间 (不使用工具链)	52
4.3	建立新的工作空间 (使用工具链)	55
4.4	打开现有工作空间	58
4.5	连接仿真器	59
4.5.1	连接仿真器	59
4.5.2	重新连接仿真器	59
4.6	断开仿真器连接	59
4.6.1	断开仿真器连接	59
4.7	退出 HEW	59
4.8	设置调试	60
4.8.1	指定下载模块	60
4.8.2	设置命令行批文件的自动执行	61
第 5 章	调试功能	62
5.1	设置仿真环境	65
5.1.1	启动时设置仿真器	65
5.1.2	设置目标 MCU	65
5.1.3	设置系统	67
5.1.4	创建存储器映像	69
5.1.5	设置闪存 ROM 盖写	70
5.1.6	设置异常事件的警告	71
5.1.7	设置可选板	71
5.1.8	显示引导处理的进度	72
5.2	下载程序	73
5.2.1	下载程序	73
5.2.2	显示源代码	73
5.2.3	关闭所有源文件中的列	75
5.2.4	关闭一个源文件中的列	75
5.2.5	显示汇编语言代码	76
5.2.6	更正汇编语言代码	78
5.3	实时显示存储器内容	78
5.3.1	实时显示存储器内容	78
5.3.2	设置 RAM 监控更新间隔	80
5.3.3	清除 RAM 监控存取历史记录	80
5.3.4	清除 RAM 监控错误检测数据	80
5.4	显示当前状态	81
5.4.1	显示仿真器状态	81

5.4.2	在状态栏中显示仿真器状态	82
5.5	定期读出和显示仿真器状态	83
5.5.1	定期读出和显示仿真器信息	83
5.5.2	选择要显示的项目	84
5.6	使用软件断点	84
5.6.1	使用软件断点	84
5.6.2	添加 / 移除软件断点	85
5.6.3	允许 / 禁止软件断点	86
5.7	使用事件	87
5.7.1	使用事件	87
5.7.2	添加事件	88
5.7.3	移除事件	93
5.7.4	注册事件	95
5.7.5	每次输入事件或重复使用事件	97
5.7.6	应用事件	98
5.8	设置硬件断点条件	99
5.8.1	设置硬件断点条件	99
5.8.2	设置硬件断点	99
5.8.3	保存 / 加载硬件断点的设置内容	101
5.9	查看跟踪信息	102
5.9.1	查看跟踪信息	102
5.9.2	获取跟踪信息	102
5.9.3	设置跟踪信息获取条件	104
5.9.4	设置跟踪模式	106
5.9.5	设置跟踪点	108
5.9.6	设置捕捉 / 不捕捉条件	112
5.9.7	选择跟踪获取的内容	114
5.9.8	显示跟踪结果	115
5.9.9	过滤跟踪信息	117
5.9.10	搜索跟踪记录	118
5.9.11	将跟踪信息保存到文件	119
5.9.12	从文件加载跟踪信息	120
5.9.13	临时停止跟踪信息获取	120
5.9.14	重新启动跟踪信息获取	120
5.9.15	切换时间戳显示	120
5.9.16	显示函数执行历史记录	121
5.9.17	显示任务执行历史记录	122
5.10	测量性能	123
5.10.1	测量性能	123
5.10.2	显示性能测量的结果	123
5.10.3	设置性能测量条件	125
5.10.4	开始性能测量	127
5.10.5	清除性能测量条件	128
5.10.6	清除性能测量结果	128
5.10.7	关于性能的最长测量时间	128
5.11	测量代码覆盖	129
5.11.1	测量代码覆盖	129
5.11.2	打开 [Code Coverage] (代码覆盖) 窗口	129
5.11.3	分配代码覆盖存储器 (硬件资源)	130
5.11.4	测量地址范围	132
5.11.5	添加地址范围	133
5.11.6	更改地址范围	135

5.11.7	移除地址范围	136
5.11.8	测量源文件	138
5.11.9	添加源文件	139
5.11.10	移除源文件	140
5.11.11	显示百分比和图形	142
5.11.12	使用 Sort 函数	142
5.11.13	搜索未执行的行	144
5.11.14	清除代码覆盖信息	144
5.11.15	更新覆盖信息	145
5.11.16	禁止信息更新	145
5.11.17	将代码覆盖信息保存到文件	145
5.11.18	从文件加载代码覆盖信息	146
5.11.19	关于覆盖信息文件加载模式	147
5.11.20	在 [Editor] (编辑器) 窗口中显示代码覆盖结果	149
5.12	测量数据覆盖	149
5.12.1	测量数据覆盖	149
5.12.2	打开 [Data Coverage] (数据覆盖) 窗口	150
5.12.3	分配数据覆盖存储器 (硬件资源)	151
5.12.4	测量地址范围	153
5.12.5	添加地址范围	154
5.12.6	更改地址范围	155
5.12.7	移除地址范围	157
5.12.8	测量段	159
5.12.9	添加段	160
5.12.10	移除段	161
5.12.11	测量任务堆栈	163
5.12.12	清除 [Data Coverage] (数据覆盖) 信息	164
5.12.13	更新覆盖信息	164
5.12.14	禁止信息更新	164
5.12.15	将数据覆盖信息保存至文件	164
5.12.16	从文件加载数据覆盖信息	165
5.13	查看实时分析信息	166
5.13.1	查看实时分析信息	166
5.13.2	设置实时分析测量模式	168
5.13.3	测量函数分析	168
5.13.4	设置函数分析测量范围	169
5.13.5	保存函数分析测量范围	169
5.13.6	加载函数分析测量范围	170
5.13.7	测量任务分析	170
5.13.8	设置任务分析测量范围	171
5.13.9	保存任务分析测量任务	172
5.13.10	加载任务分析测量任务	172
5.13.11	清除实时分析测量结果	172
5.13.12	保存实时分析测量结果	173
5.13.13	设置测量单位	173
5.13.14	实时分析的最长测量时间	174
5.14	检测异常事件	174
5.14.1	检测异常事件	174
5.14.2	检测存取保护的违规操作	175
5.14.3	设置存取保护区域	176
5.14.4	检测忽略初始化	181
5.14.5	检测性能上溢	181

5.14.6	检测实时分析上溢:	182
5.14.7	检测跟踪存储器上溢	182
5.14.8	检测任务堆栈存取的违规操作:	182
5.14.9	设置任务堆栈区域	183
5.14.10	检测操作系统调度	186
5.15	使用启动 / 停止功能	187
5.15.1	打开 [Start/Stop Function Setting] (启动 / 停止功能设置) 对话框	187
5.15.2	指定要执行的例行程序	187
5.15.3	启动 / 停止功能的限制	188
5.15.4	对指定例行程序中所写语句的限制	189
第 6 章	故障排除 (针对错误采取操作)	190
6.1	故障解决流程图	190
6.2	自检错误	191
6.3	调试器启动时的错误	192
6.4	如何请求支持	193
第 7 章	硬件规格	194
7.1	目标 MCU 规格	194
7.2	实际 MCU 与仿真器的不同点	194
7.3	连接图	196
7.3.1	R0E530640MCU00 的连接图	196
7.4	外部尺寸	197
7.4.1	E100 仿真器的外部尺寸	197
7.4.2	转换器板 R0E0100TNPFJ00 的外部尺寸	198
7.4.3	转换器板 R0E0100TNPFK00 的外部尺寸	198
7.5	关于使用本产品的注意事项	199
第 8 章	维修和保证	203
8.1	用户注册	203
8.2	维修	203
8.3	保证	203
8.4	维修提供	203
8.5	如何提出维修申请	204

用户注册

本手册包括客户注册表，请填写该客户注册表，然后将其传真或通过电子邮件发送给本地分销商。如果通过电子邮件注册，可以使用在以下文件夹中安装软件时创建的用户注册表的文本格式。注册的信息仅用于售后服务，不用于其他任何目的。如果不进行用户注册，将无法收到如现场变更或故障信息通知等维护服务。因此，务必进行用户注册。

有关用户注册的详细信息，请与当地经销商联系。

用户注册表的文本格式 **C:\Program Files\Renesas\Hew\Support**

术语表

本用户手册中使用的一些特定字词定义如下：

MCU 单元 R0E530640MCU00

是指 M16C/64 组的 E100 仿真器。

仿真器系统

是指围绕 MCU 单元 R0E530640MCU00 创建的仿真器系统。该仿真器系统配置如下：一个仿真器主部件 R0E001000EMU00、MCU 单元 R0E530640MCU00、仿真器电源、USB 电缆、仿真调试器以及主机。

集成开发环境高性能嵌入式工作区（High-performance Embedded Workshop，简称 HEW）

此工具为开发用于瑞萨单片机的嵌入式应用系统提供强有力的支持。该工具还具有仿真调试器功能，允许从主机通过接口控制仿真器。另外，使用该工具还可在同一应用程序中执行从编辑工程到创建和调试工程这一系列的操作。而且，它还支持版本管理。

仿真调试器

是指一种软件工具，该软件工具从 HEW 启动来控制此产品并启用调试。

固件

是指存储在仿真器中的控制程序。此程序分析与仿真调试器的通信内容并控制仿真器硬件。要升级固件，请通过仿真调试器下载程序。

主机

是指用于控制仿真器的个人计算机。

目标 MCU

是指要调试的 MCU。

用户系统

是指使用要调试的 MCU 的用户应用程序系统。

用户程序

是指要调试的程序。

评估 MCU

是指在工具的特定模式下运行的仿真器上装载的 MCU。

#

此符号用于显示低电平有效。（例如，RESET#：复位信号）

第 1 章 概要

本章介绍套装组件、系统配置、仿真器功能规格和操作环境。

1.1 套装组件

R0E530640MCU00 套件包括以下物品。打开包装时，请清点 R0E530640MCU00 是否包括所有这些物品。

表 1.1 套装组件

物品	数量
R0E530640MCU00 MCU 安装板	1
安装在 IC17 插座中的振荡器模块 (20MHz)	1
R0E001000FLX10 柔性电缆	2
R0E530640MCU00 发行说明 (英文)	1
R0E530640MCU00 发行说明 (日文)	1
维修申请表 (英文)	1
维修申请表 (日文)	1
CD-ROM	1
- M16C R8C E100 仿真调试程序 - 用户手册	

- 注意：
1. 请保管好 R0E530640MCU00 的包装箱和衬垫材料，以便以后因修理或其他目的需要运送产品时重新使用。运送本产品时，请始终使用该包装盒和衬垫材料。
 2. 如果对包装的产品有任何问题或疑问，请联系当地的经销商。

1.2 进行开发所需的其他工具产品

要在 M16C/60 系列 M16C/64 组 MCU 上进行程序开发，除了上述套装中包含的物品，还需要下面所列的产品。请另外获取它们。

表 1.2 进行开发所需的其他工具产品

产品	部件号
仿真器主部件 E100	R0E001000EMU00
100 引脚 0.65mm 节距 QFP (PRQP0100JD-B 以前的代码: 100P6F-A)	R0E0100TNPFFJ00
100 引脚 0.5mm 节距 LQFP (PLQP0100KB-A 以前的代码: 100P6Q-A)	R0E0100TNPFFK00

注意： 要购买这些产品，请联系当地的经销商。

1.3 系统配置

1.3.1 系统配置

图 1.1 显示了仿真器系统的一种配置。

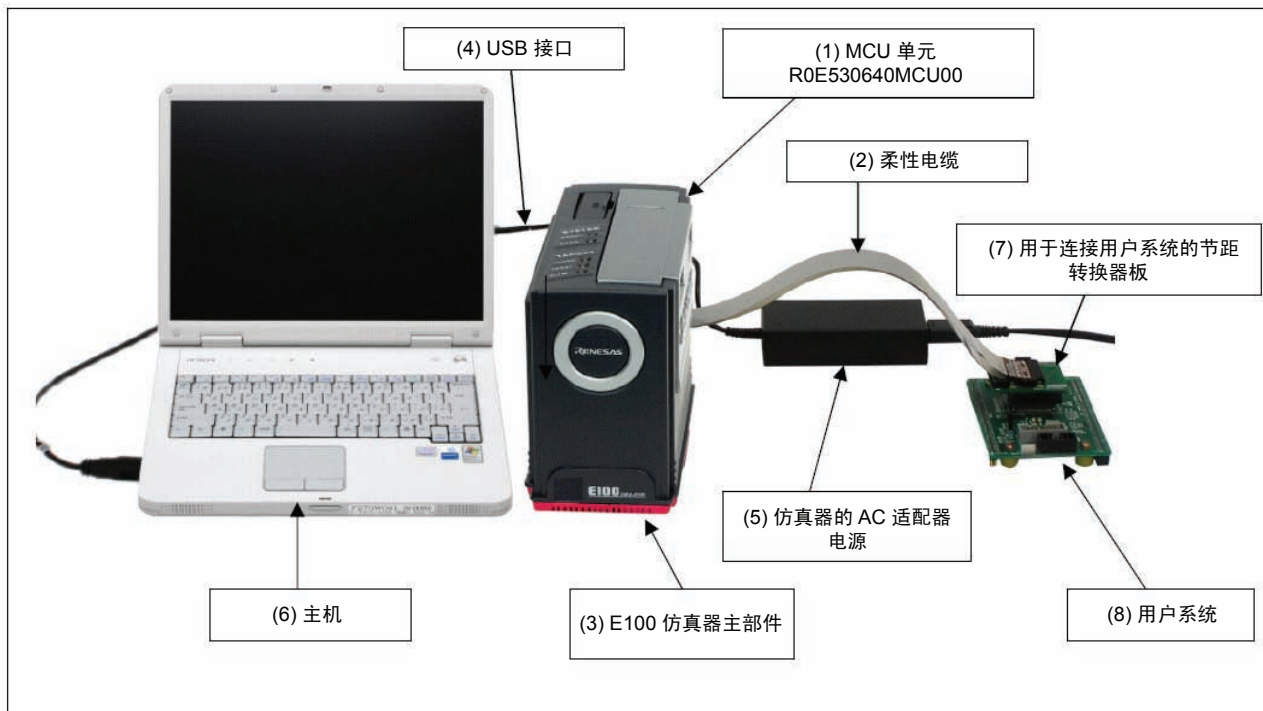


图 1.1 系统配置

- (1) MCU 单元 R0E530640MCU00（本产品）
这是带有 512 KB ROM 并包含一个评估 MCU 的 M16C/60 系列 M16C/64 组 MCU 的 MCU 安装板。
- (2) 柔性电缆 R0E001000FLX10（包含）
- (3) E100 仿真器主部件 R0E001000EMU00
这是 E100 仿真器主部件。
- (4) USB 接口电缆
这是主机和仿真器的接口电缆。
- (5) 仿真器的 AC 适配器电源
- (6) 主机
用于控制仿真器的个人计算机。
- (7) 用于连接 R0E0100TNPJ00 等用户系统的节距转换器板

(8) 用户系统和用户系统电源

用户系统是用户的应用程序系统。如果没有用户系统，仍可使用仿真器。

用户系统电源是向用户系统供电的电源。本仿真器无法向用户系统供电。请另外获取电源。

1.3.2 仿真器各部分的名称和功能

图 1.2 显示了仿真器各部分的名称。



图 1.2 仿真器各部分的名称

(1) 电源开关

这是用于打开 / 关闭仿真器的开关。

(2) USB 电缆连接器

这是用于连接仿真器的 USB 电缆的连接器。

(3) 电源连接器

这是用于连接仿真器 AC 适配器电源的 DC 电缆的连接器。

(4) 外部触发器连接器

这是用于连接仿真器的外部触发器电缆的连接器。

(5) 系统状态 LED

系统状态 LED 指示仿真器 E100 的电源、固件工作状况等。表 1.3 列出了每个系统状态 LED 的定义。

表 1.3 系统状态 LED 的定义

名称	状态	含义
POWER (电源)	ON (开)	仿真器系统电源打开。
	OFF (关)	仿真器系统电源关闭。
SAFE (安全)	ON (开)	仿真器系统运行正常。
	闪烁	仿真器系统无法与主机通信。
	闪烁 (每 2 秒闪烁一次)	正在执行自检。
	OFF (关)	仿真器系统运行不正常 (系统状态错误)。

(6) 目标状态 LED

目标状态 LED 指示目标 MCU 和用户系统电源的工作状况。表 1.4 列出了每个目标状态 LED 的定义。

表 1.4 目标状态 LED 的定义

名称	状态	含义
POWER (电源)	ON (开)	正在向用户系统供电。
	OFF (关)	未向用户系统供电。
RESET (复位)	ON (开)	正在复位目标 MCU, 或用户系统的复位信号保持在 low (低) 水平。
	OFF (关)	目标 MCU 不在复位。
RUN (运行)	ON (开)	正在执行用户程序。
	OFF (关)	用户程序已停止。

重要事项

关于目标状态“POWER”(电源) LED 注意事项:

- 如果 MCU 有两个或更多 Vcc 引脚, 则除非向所有引脚供电, 否则该 LED 不会亮起。

1.4 规格

表 1.5 列出了 R0E530640MCU00 的规格。

表 1.5 R0E530640MCU00 的规格

适用 MCU	M16C/60 系列 M16C/64 组 MCU，带有 512 KB ROM	
适用 MCU 模式	单片模式、存储器扩展模式、微处理器模式	
最大 ROM/RAM 容量	1. 内部闪存 ROM: 8KB+16KB+512KB 0E000h~0FFFFh、10000h~13FFFh、80000h~FFFFFFh 2. 内部 RAM: 31KB 00400h~07FFFh	
最高工作频率	电源电压: 2.7~5.5V 25MHz (带 PLL) (仿真存储器 0 个等待时: 12MHz, 1 个等待或 1 个等待以上时: 25MHz)	
软件断点	4096 点 (执行前使用 RAM 完成断点功能)	
硬件断点	16 点 (执行地址、总线检测、中断、外部触发器信号)	
组合, 通过计数	- 累积 AND/OR/ 状态转换 - 255 通过计数	
异常事件检测	存取保护的违规操作 / 任务堆栈存取的违规操作 / 操作系统调度 / 忽略初始化	
实时跟踪	192 位 × 4M 周期 (地址、数据、状态、CPU 状态、总线状态、目标状态、任务 ID、时间戳、32 外部触发输入)	
跟踪模式	Fill until stop (填充直至停止) /full (已满) /point and delay (点和延迟) /repeat (free) (重复 (空闲)) /repeat (full) (重复 (已满))	
跟踪提取 / 删除	通过事件捕捉 / 不通过事件捕捉 或 指定的数据存取指令提取 / 点之前和之后的跟踪提取	
实时 RAM 监控	- 16384 字节 (512 字节 × 32 个块) - 数据 / 上次存取	
时间测量	- 程序启动和停止之间的执行时间 - 指定的八个段的最多 / 最少 / 平均执行时间和通过计数 - 计数时钟: 等于 MCU 时钟或 10ns~1.6us	
覆盖范围测量	2 MB (256 KB × 8 个块)	
配置文件	1 MB (128 KB × 8 个块)	
与用户系统的连接	100 引脚 0.65mm 节距 QFP	R0E0100TNPJ00
	100 引脚 0.5mm 节距 LQFP	R0E0100TNPJK00
仿真器电源	从附带的 AC 适配器提供 (电源电压: 100~240 V, 50/60 Hz)	

1.5 操作环境

使用本仿真器时，请务必遵循表 1.6 和表 1.7 中所列的操作环境。

表 1.6 操作环境条件

项目	描述
工作温度	5 到 35 °C（不结露）
存储温度	-10 到 60 °C（不结露）

表 1.7 主机操作环境

项目	描述
主机	IBM PC/AT 兼容
OS	Windows XP [*1] Windows 2000
CPU	推荐使用 Pentium IV 1.6 GHz 或更高
接口	USB 2.0 [*2]
存储器	推荐使用 768 MB 或更高（超过加载模块的文件大小的 10 倍）
鼠标等点击器件	鼠标或任何其他可用于上述操作系统的可连接到主机的点击器件
CD 光驱	用于安装仿真调试器或参阅用户手册

- 注意：
1. Windows 和 Windows NT 是 Microsoft Corporation 在美国和 / 或其他国家（地区）的注册商标或商标。
 2. USB 接口并不保证可用于主机、USB 器件和 USB 集线器的所有组合。

第 2 章 设置

本章介绍使用该产品的准备工作、启动仿真器的过程，以及如何更改设置。

2.1 启动仿真器的流程图

图 2.1 和图 2.2 中显示了启动仿真器的过程。有关详细信息，请参阅后面各节的内容。如果仿真器无法正常启动，请参阅第 6 章“故障排除（针对错误采取操作）”（第 190 页）。

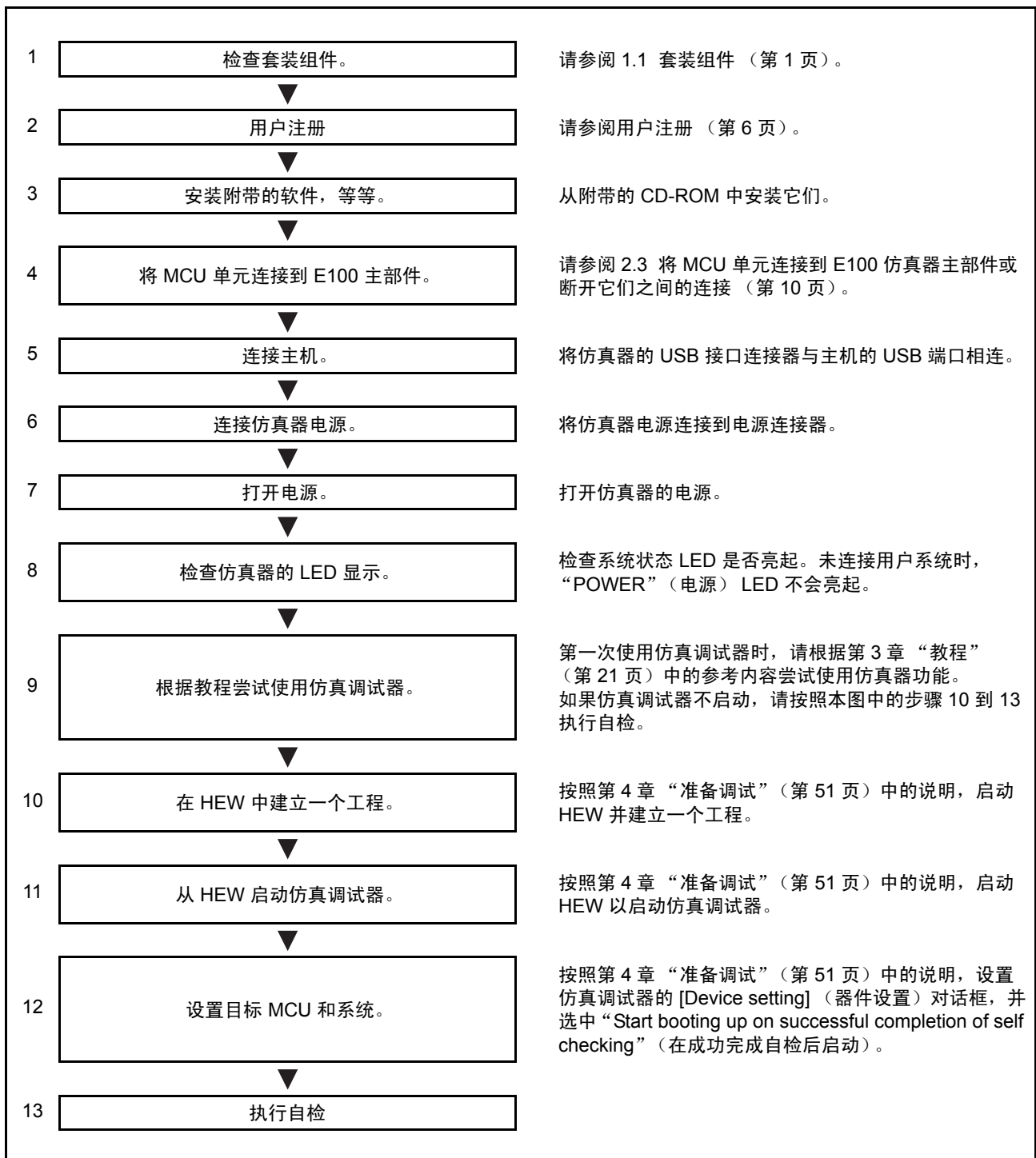


图 2.1 （第一次）启动仿真器的流程图

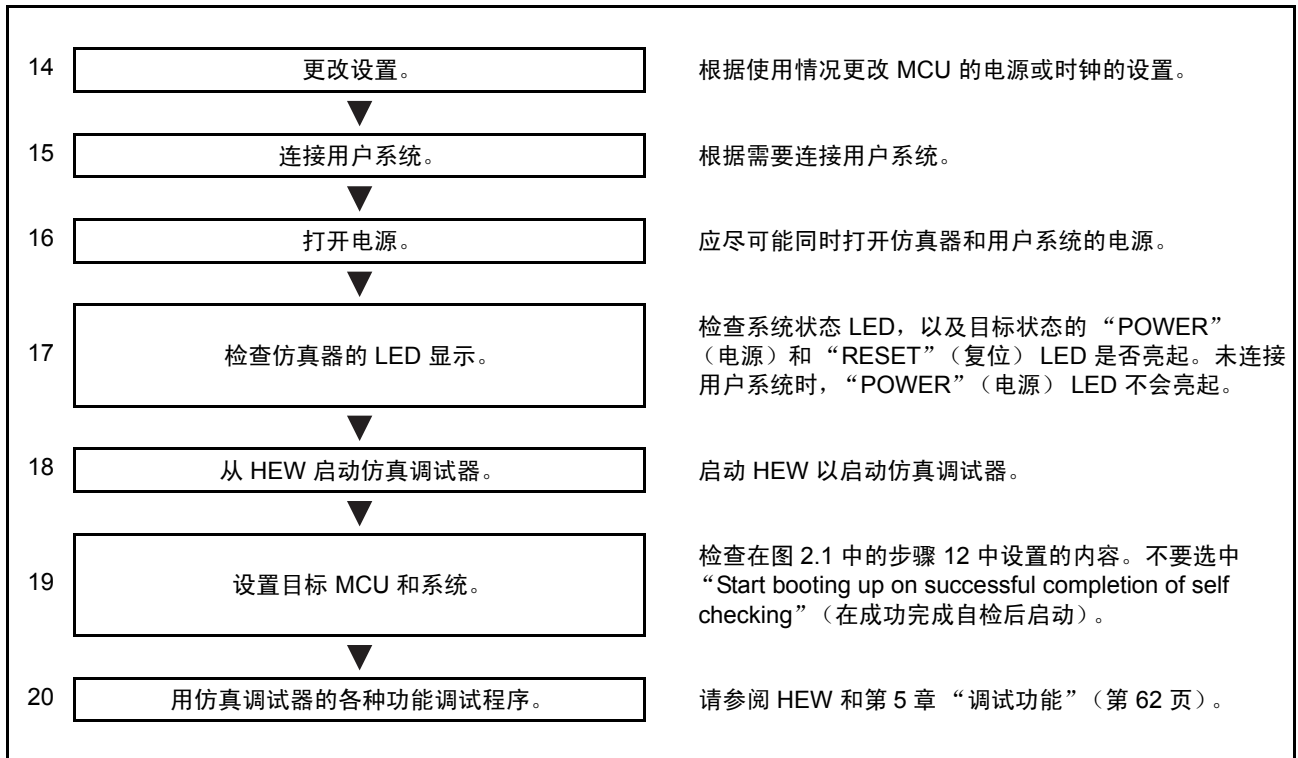


图 2.2 （在自检后）启动仿真器的流程图

2.2 安装附带的软件

如果主机上有 Windows XP 或 2000，则该安装必须由具有管理员权限的用户执行。请注意，没有管理员权限的用户无法完成该安装。

将附带的 CD-ROM 插入 CD-ROM 驱动器之后，将显示一条消息。根据所显示的消息安装软件。

将显示一个对话框，用于输入用户信息（用户、公司名称、地址、安装目的地）。所输入的信息的格式将在通过邮件发送的用户注册表中进行设置。

2.3 将 MCU 单元连接到 E100 仿真器主部件或断开它们之间的连接

图 2.3 显示了将 MCU 单元连接到 E100 仿真器主部件的过程。

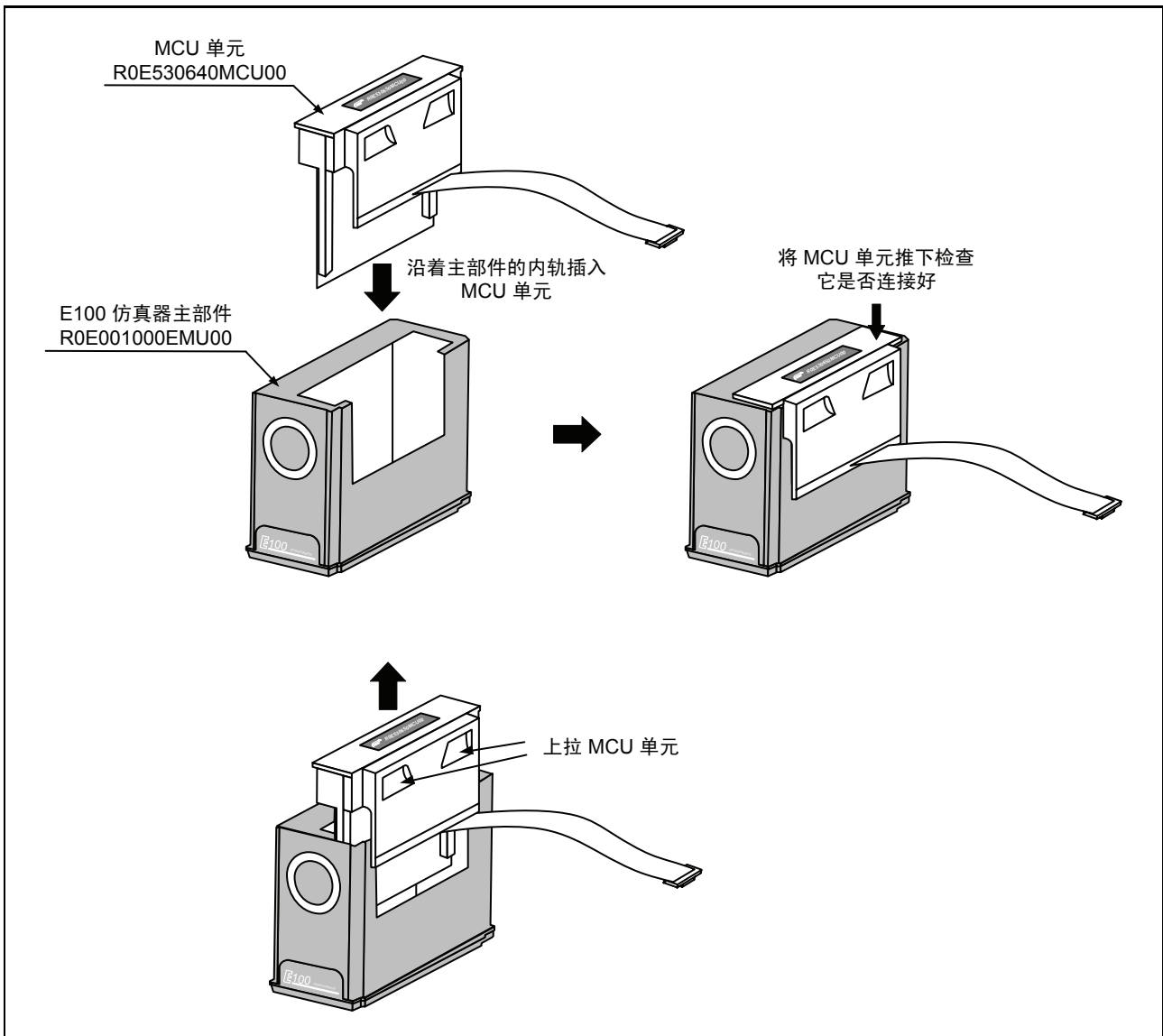


图 2.3 将 MCU 单元连接到 E100 仿真器主部件或断开它们之间的连接

⚠ 注意

关于将 MCU 单元连接到 E100 仿真器主部件的注意事项:

- 将 MCU 单元连接到 E100 仿真器主部件时, 应始终关闭电源, 否则可能会损坏内部电路。



2.4 连接主机

USB 接口用于将仿真器连接到主机。USB 电缆连接到仿真器的 USB 电缆连接器和主机的 USB 端口。

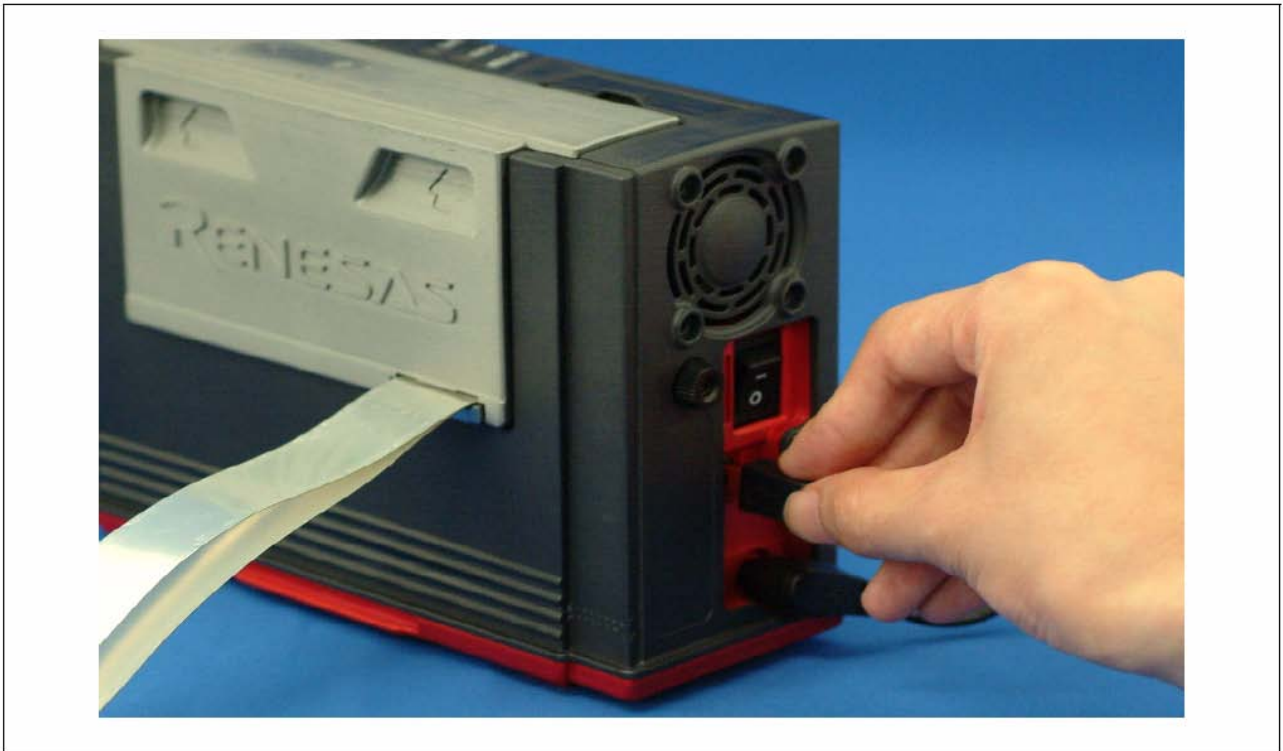


图 2.4 连接主机

2.5 连接仿真器电源

从附带的 AC 适配器向仿真器供电。下面显示了如何连接 AC 适配器。

- (1) 关闭仿真器。
- (2) 将 AC 适配器的 DC 电缆连接到仿真器。
- (3) 将 AC 电源电缆连接到 AC 适配器。
- (4) 将 AC 电源电缆连接到插座。



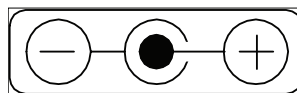
图 2.5 连接仿真器电源

⚠ 注意

AC 适配器注意事项:



- 仅使用 E100 套装中附带的 AC 适配器。
- 附带的 AC 适配器仅供 E100 仿真器主部件使用，不要将它用于其他产品。
- 安装本产品或将其连接到其他设备之前，请将 AC 电源电缆从插座中拔出，以避免伤害或意外事故。
- 附带的 AC 适配器的 DC 插头的极性如下。



- 附带的 AC 适配器没有电源开关。AC 适配器在连接到 AC 电源电缆时始终保持打开状态。

2.6 打开电源

2.6.1 检查仿真器系统的连接

打开电源之前，检查接口电缆与主机、仿真器和用户系统的连接。

2.6.2 打开 / 关闭电源

- 尽可能同时打开 / 关闭仿真器和用户系统的电源。
- 系统 LED 的“SAFE”（安全）LED 闪烁时，检查 USB 电缆是否已连接到主机。每个目标状态 LED 闪烁时，检查是否已连接 MCU 单元。
- 在关闭电源之后再次打开电源之前，请等待约 10 秒钟

重要事项

关于电源的注意事项：

- 仿真器引脚 Vcc 连接到用户系统是为了监控用户系统电压。因此，仿真器无法向用户系统供电。应单独为用户系统供电。
用户系统的电压应当如下所示。
$$2.7\text{ V} \leq V_{cc1} = V_{cc2} \leq 5.5\text{ V}$$
- 在不使用用户系统的情况下启动仿真器时，不要连接转换器板。在连接了转换器板的情况下启动时，MCU 将处于复位状态。
- 在不使用用户系统的情况下启动仿真器时，请小心，不要让金属片接触柔性电缆头部的连接器。
- 不要让仿真器或用户系统接通电源，否则，可能会由于泄漏电流而损坏内部电路。

2.7 自检

自检是为了检查仿真器功能是否正常工作。要运行仿真器的自检功能，请按照以下步骤操作。正在进行自检时，LED 将按图 2.6 中所示的那样变化。在发生错误时，由于目标状态 LED 将根据错误进行变化，因此请检查系统状态 LED。

- (1) 如果连接了用户系统，请断开转换器板与用户系统的连接。
- (2) 打开仿真器。
- (3) 启动仿真调试器，并选中 [Device setting]（器件设置）对话框中的 [Start booting up on successful completion of self checking]（在成功完成自检后启动）复选框。
- (4) 单击 [OK]（确定）后，自检将开始。如果在约 60 秒内显示正常结果，则自检正常结束。

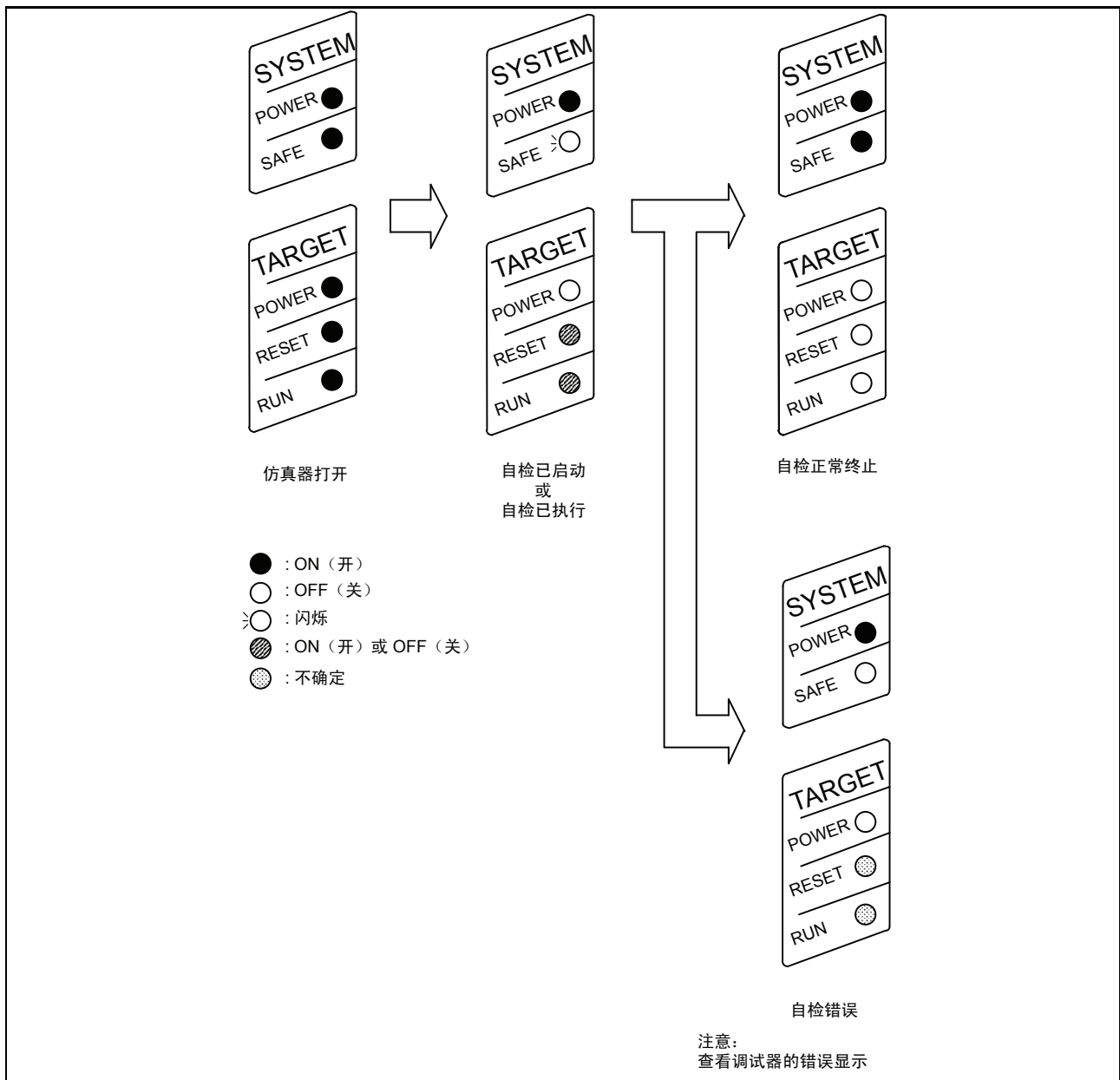


图 2.6 自检过程中的 LED 显示

2.8 选择时钟提供

2.8.1 时钟

可以在仿真调试器的 [Configuration properties]（配置属性）对话框的 [Emulator]（仿真器）选项卡中选择向评估 MCU 提供的时钟。表 2.1 显示了时钟及其初始设置。

表 2.1 向 MCU 提供时钟

时钟	仿真调试器的显示	描述	默认设置
Main (X _{IN} -X _{OUT})	Emulator（仿真器）	IC17 安装振荡器模块	是
	User（用户）	用户系统中的振荡器电路	-
	Generate（生成）	内部生成器电路 (1.0~20.0 MHz)	-
Sub (X _{CIN} -X _{COU} T)	Emulator（仿真器）	内部振荡器电路 (32.768 kHz)	是
	User（用户）	用户系统中的振荡器电路	-

重要事项

关于更改时钟提供的注意事项：

- 时钟提供可以在启动仿真调试器时通过 [Configuration properties]（配置属性）对话框设置，或通过 [Command Line]（命令行）窗口中输入 emulator_clock 命令设置。

2.8.2 使用内部振荡器电路板

振荡器电路板的种类

在出厂设置中，在 IC17 上安装了一个振荡器模块 (20MHz)。如果更改该频率，请更换振荡器模块。

(1) 更换振荡器模块

从 E100 仿真器主部件中取出 MCU 单元，然后更换 IC17 的振荡器模块（参见图 2.7）。

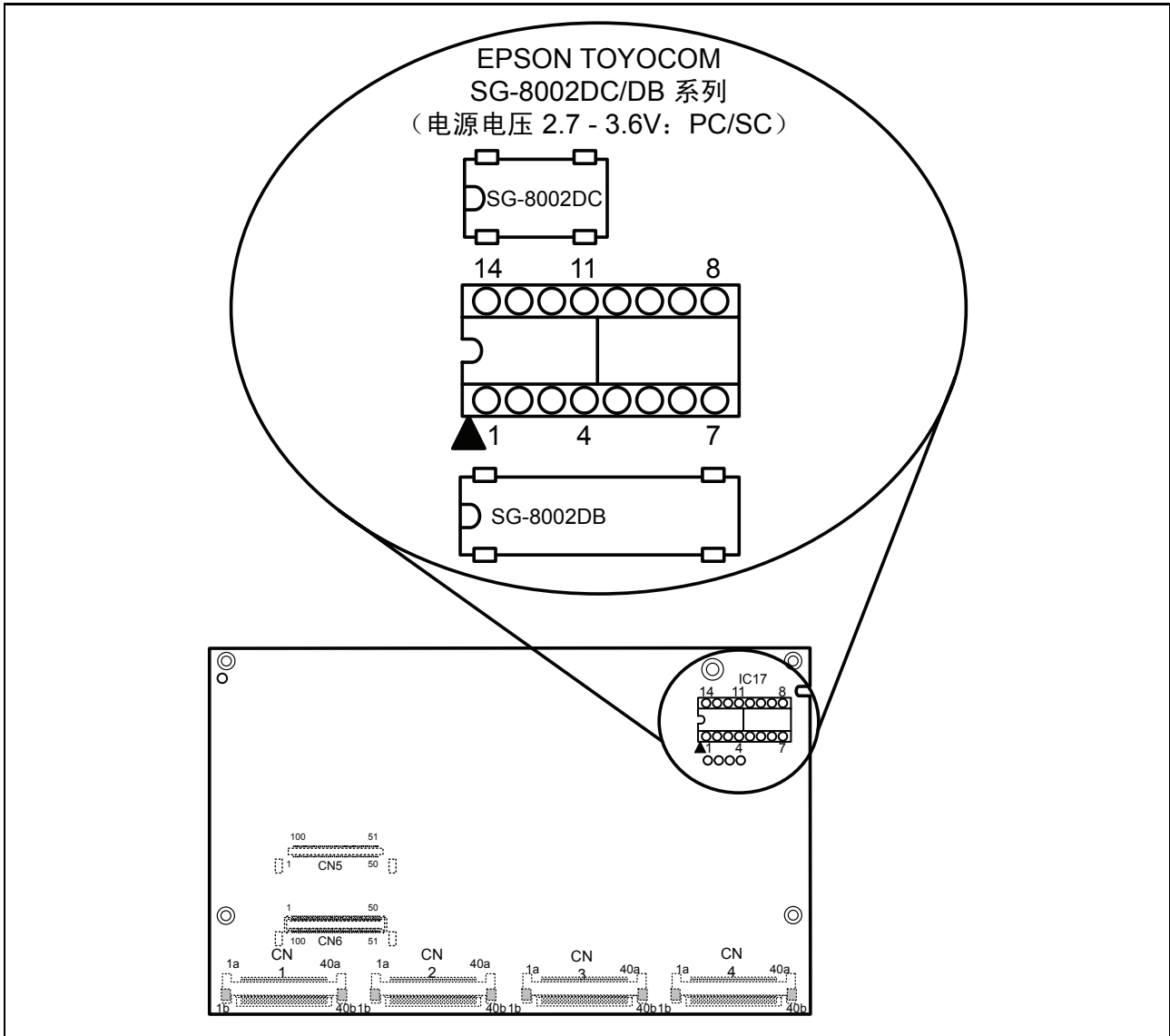


图 2.7 更换振荡器模块

注意

关于更换振荡器模块和振荡器电路板的注意事项：

- 更换振荡器模块时应始终关闭电源，否则可能会损坏内部电路。



2.8.3 在用户系统中使用振荡器电路

要将本产品与外部时钟一起使用，请按图 2.8 中所示在用户系统中构建振荡器电路，并将处于 50% 占空比（在评估 MCU 工作范围内）的振荡器输出输入到引脚 X_{IN} 中。并且，引脚 X_{OUT} 应打开。在仿真调试器中选择 [User]（用户）以使用该时钟。

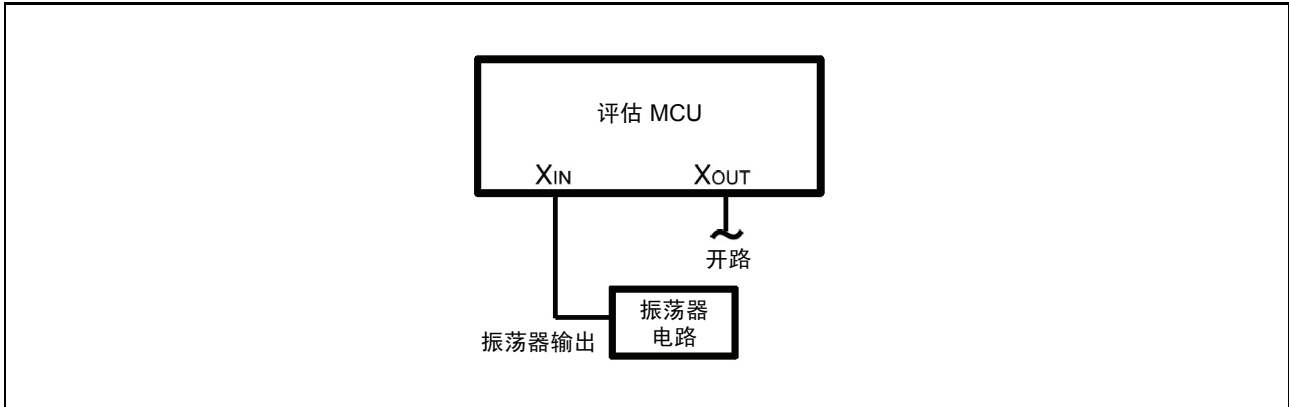


图 2.8 外部振荡器电路

请注意，在图 2.9（该图显示引脚 X_{IN} 和 X_{OUT} 之间连接了一个振荡器）中显示的振荡器电路中，由于在评估 MCU 和用户系统之间使用了一个转换器板和其他设备，因此不发生振荡。对于副时钟振荡器电路（X_{CIN} 和 X_{COU}T）也是如此。

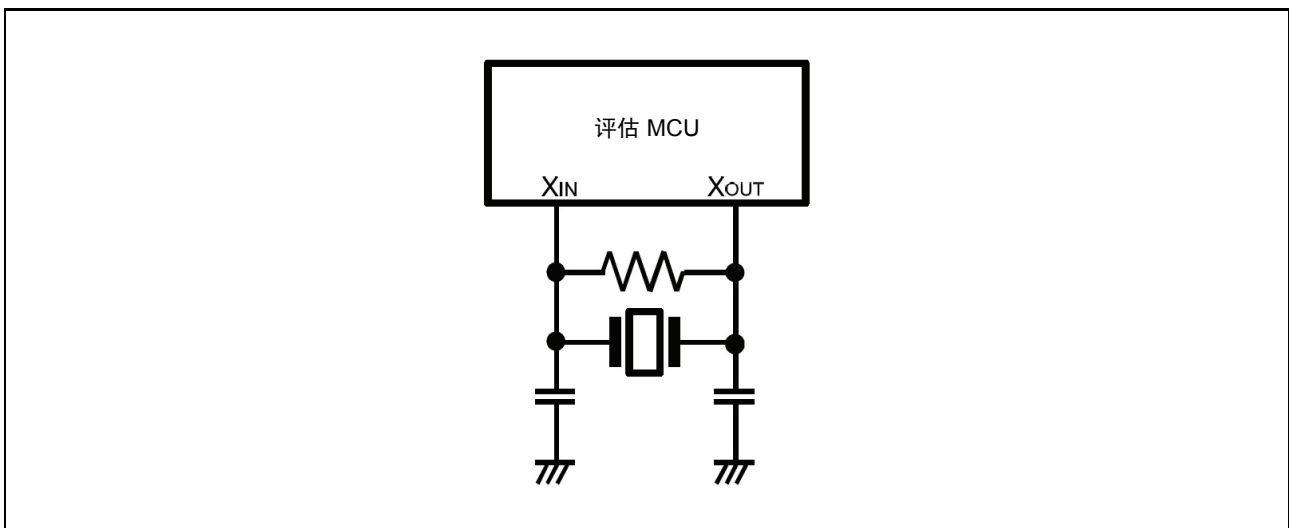


图 2.9 不发生振荡的电路

2.8.4 使用内部生成器电路

E100 中的专用电路可以生成仿真调试器指定的任意频率，并且可以作为主时钟提供。它不依赖 MCU 单元中的振荡器电路板或用户系统中的振荡器电路。如果希望在没有用户系统的情况下调试程序，或要暂时更改频率，可以在购买振荡器之前检查其工作情况。如果要在 E100 中使用内部生成器电路作为主时钟，请在仿真调试器中选择 [Generate]（生成），并指定要使用该时钟的频率。

虽然用户可以在 1.0 ~ 99.9 MHz 之间以 0.1 MHz 为单位更改 E100 的频率，但指定的值不应超过 MCU 的 X_{IN} 的最高输入频率 20 MHz。

重要事项

关于使用内部生成器电路的注意事项：

- 为了进行临时调试而配备了内部生成器电路。不保证频率的温度特性。
- 请务必使用与用于最终评估的振荡器模块或振荡器电路（仿真器）频率相同的振荡器来评估系统。

2.9 连接用户系统

图 2.10 显示了如何将本产品连接到用户系统。

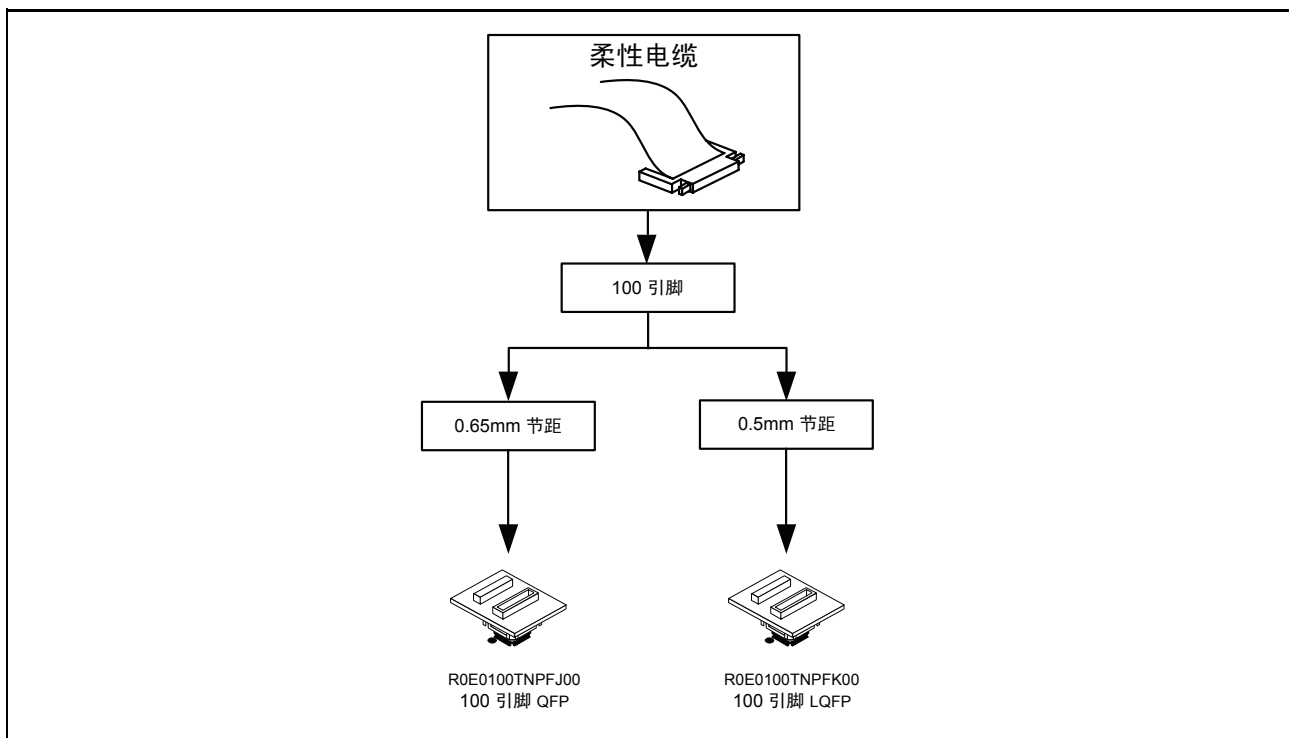


图 2.10 将本产品连接到用户系统

注意

关于连接用户系统的注意事项：



- 切勿以错误的方向连接转换器板，否则可能严重损坏仿真器和用户系统。

2.9.1 连接到 100 引脚 0.65mm 节距支脚模式

以下是使用 R0E0100TNPJ00（不附带）连接到用户系统中的 100 引脚 0.65mm 节距支脚模式的过程。有关 R0E0100TNPJ00（不附带）的详细信息，请参阅其用户手册。

- (1) 将随 R0E0100TNPJ00 提供的 NQPACK100RB 连接到用户系统。
- (2) 将随 R0E0100TNPJ00 提供的 YQPACK100RB 连接到 NQPACK100RB，并用 YQ-GUIDE 将其固定。
- (3) 将 R0E0100TNPJ00 连接到 YQPACK100RB。
- (4) 将 R0E0100TNPJ00 的 CN2 端连接到柔性电缆的 CN2 端。
- (5) 将 R0E0100TNPJ00 的 CN1 端连接到柔性电缆的 CN1 端。

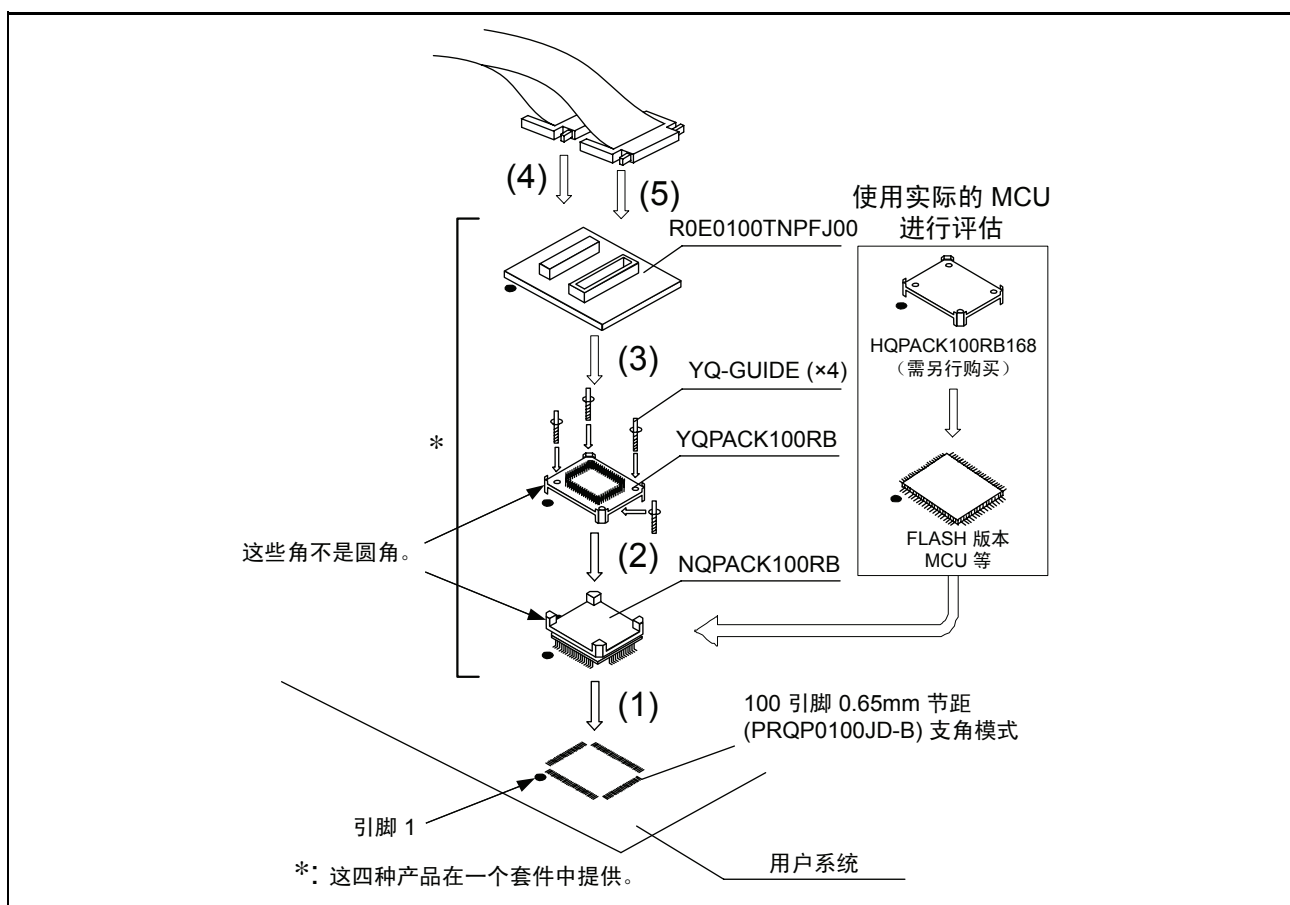


图 2.11 连接到 100 引脚 0.65mm 节距支脚模式

注意

关于连接用户系统的注意事项:



- 切勿以错误的方向连接转换器板，否则可能严重损坏仿真器和用户系统。
- R0E0100TNPJ00 的连接器只能保证 50 次插 / 拔不损坏。
- 要购买 HQPACK100RB168，请联系:

Tokyo Eletech Corporation

http://www.tetc.co.jp/e_index.htm

2.9.2 连接到 100 引脚 0.5mm 节距支脚模式

以下是使用 R0E0100TNPFK00（不附带）连接到用户系统中的 100 引脚 0.5mm 节距支脚模式的过程。有关 R0E0100TNPFK00（不附带）的详细信息，请参阅其用户手册。

- (1) 将随 R0E0100TNPFK00 提供的 NQPACK100SD-ND 连接到用户系统。
- (2) 将随 R0E0100TNPFK00 提供的 YQPACK100SD 连接到 NQPACK100SD-ND，并用 YQ-GUIDE 将其固定。
- (3) 将 R0E0100TNPFK00 连接到 YQPACK100SD。
- (4) 将 R0E0100TNPFK00 的 CN2 端连接到柔性电缆的 CN2 端。
- (5) 将 R0E0100TNPFK00 的 CN1 端连接到柔性电缆的 CN1 端。

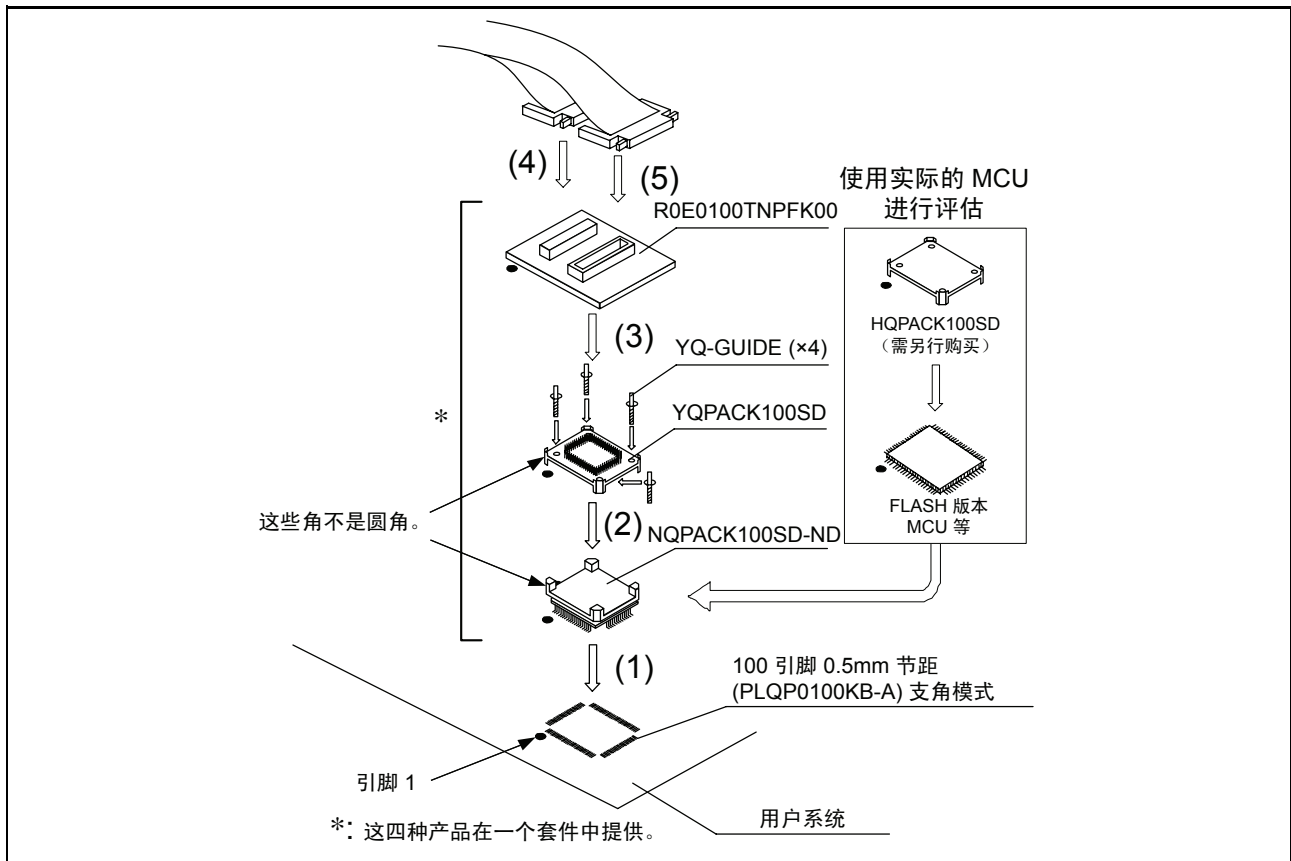


图 2.12 连接到 100 引脚 0.5mm 节距支脚模式

注意

关于连接用户系统的注意事项:



- 切勿以错误的方向连接转换器板，否则可能严重损坏仿真器和用户系统。
- R0E0100TNPFK00 的连接器只能保证 50 次插 / 拔不损坏。
- 要购买 HQPACK100SD，请联系：

Tokyo Eletech Corporation

http://www.tetc.co.jp/e_index.htm

第 3 章 教程

3.1 简介

E100 仿真器提供一个教程程序。该程序可作为向用户提供仿真器主要功能的方法，本文中将其进行说明。

该教程程序用 C 语言编写，可用升 / 降序来对 10 条随机数据进行排序。
下面略述该教程程序执行的处理。

为了重复执行排序过程，main 函数重复调用 tutorial 函数。

tutorial 函数生成要排序的随机数据并按顺序调用 sort 和 change 函数。

sort 函数接受一个包含 tutorial 函数生成的随机数据的数组作为其输入，并将输入数据按升序排序。

change 函数接受一个由 sort 函数按升序排序的数组作为其输入，并将输入数据按降序排序。

该教程程序是一个旨在帮助用户理解如何使用仿真器及仿真调试器的功能的程序。开发用户系统和用户程序时，请参考目标 MCU 的用户手册。

注意： 如果重新编译该教程程序，则编译后的程序中的地址可能与本章中描述的地址不同。

3.2 启动 HEW

按 4.4 “打开现有工作空间” 描述的步骤打开一个工作空间。

指定下面给出的目录。

装有操作系统的驱动器 \Workspace\Tutorial\E100\M16C

指定如下所示的文件。

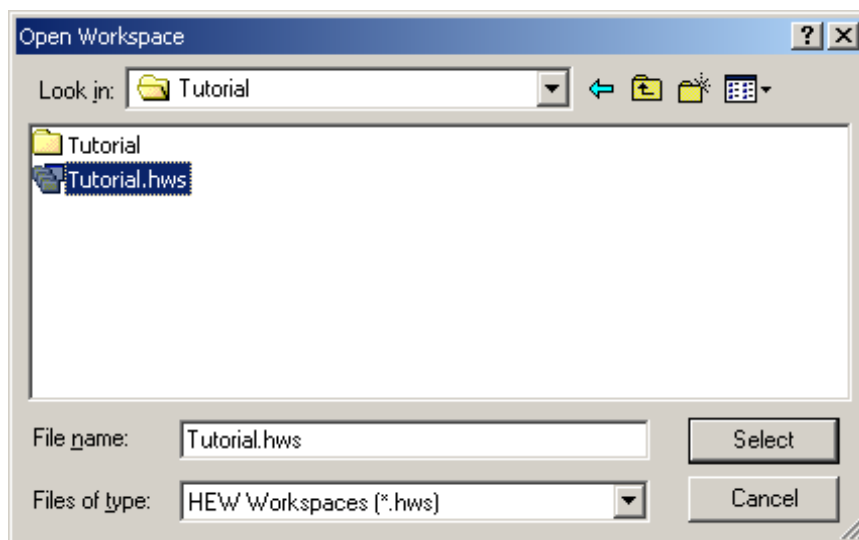


图 3.1 [Open Workspace]（打开工作空间）对话框

3.3 连接仿真器

调试器连接到仿真器后，会显示一个用于设置调试器的对话框。在此对话框中，进行调试器的初始设置。调试器设置完成后，可以准备调试了。

3.4 下载教程程序

3.4.1 下载教程程序

下载要调试的目标程序。但是请注意，对于使用的每个微型计算机而言，要下载的程序和下载到微型计算机中的地址会有所不同。请读取屏幕上适合用户所使用的微型计算机的字符串等内容。

从 [Download modules]（下载模块）中的 [Tutorial.x30] 下选择 [Download]（下载）。

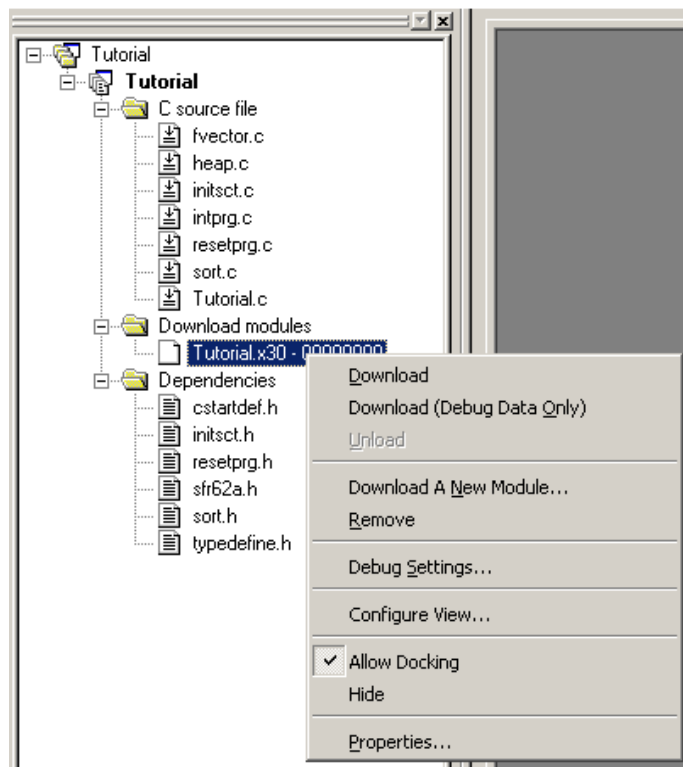


图 3.2 教程程序的 [Download]（下载）显示

3.4.2 显示源程序

在 HEW 中，可在源级调试程序。

双击 [C source file]（C 源文件）下的 [Tutorial.c]。

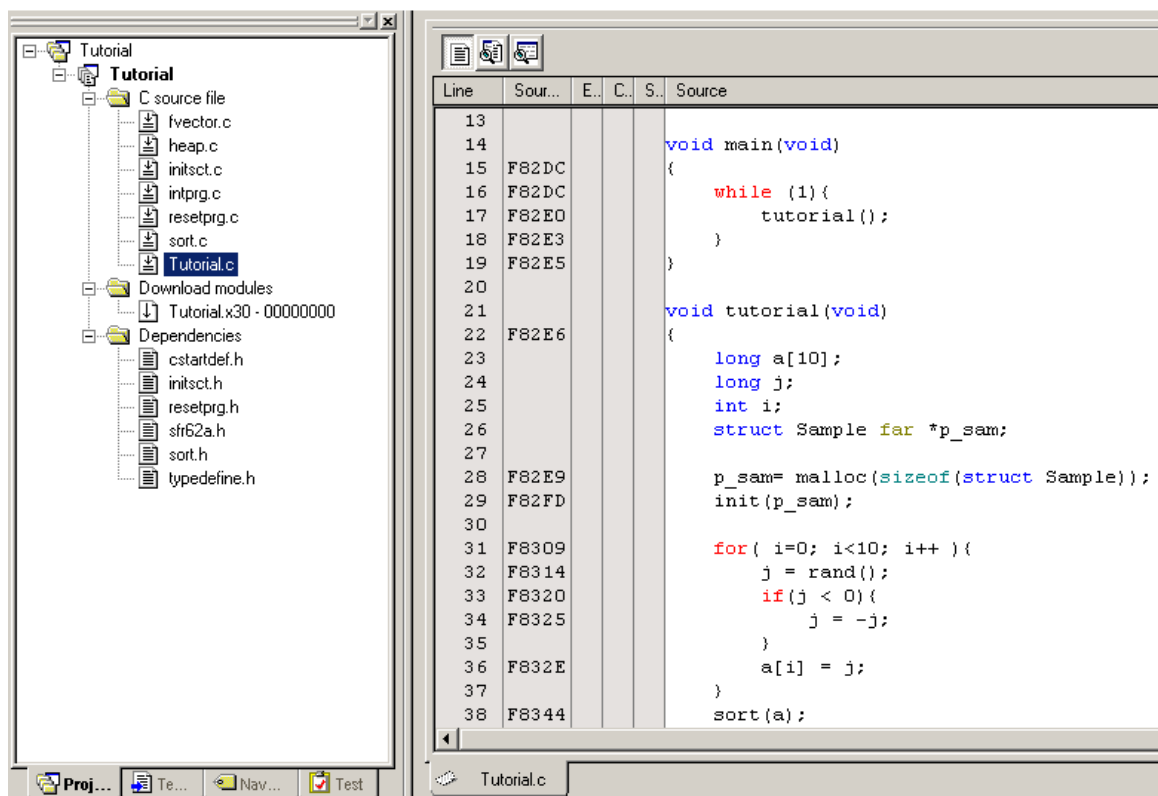


图 3.3 [Editor]（编辑器）窗口（显示源程序）

如有必要，可以更改字体和字号以使文本更便于阅读。有关如何更改的详细信息，请参考“HEW 用户手册”。

[Editor]（编辑器）窗口初始显示程序的起始部分。用户可以使用滚动栏查看程序的另一部分。

3.5 设置软件断点

软件断点是一个简单的调试工具。

[Editor]（编辑器）窗口使用户可以轻松地设置软件断点。例如，用户可以在 `sort` 函数调用处设置一个软件断点。

在 [S/W Breakpoints]（软件断点）列中，双击与包含 `sort` 函数调用的源行对应的行。

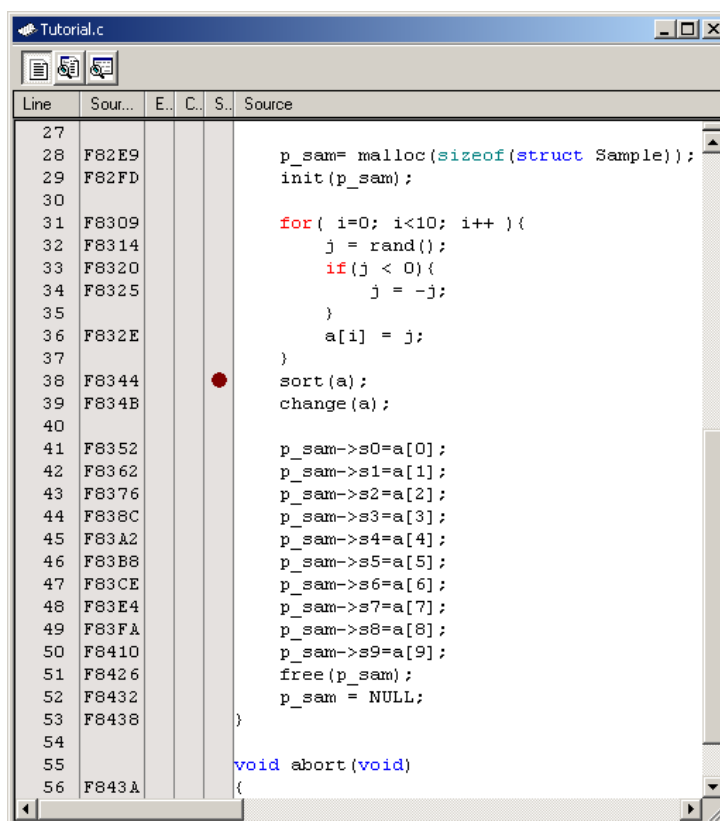


图 3.4 [Editor]（编辑器）窗口（设置软件断点）

包含 `sort` 函数的源行将用一个红圈标记，指示在该位置设置了一个软件断点。

3.6 执行程序

下面介绍如何运行程序。

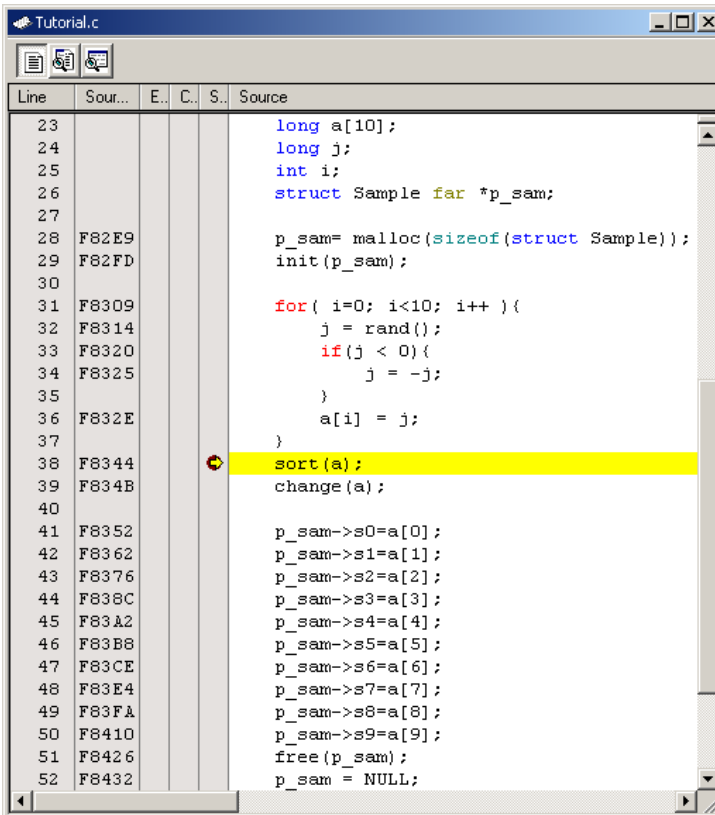
3.6.1 复位 CPU

要复位 CPU，请从 [Debug]（调试）菜单选择 [Reset CPU]（复位 CPU），或者单击工具栏上的 [Reset CPU]（复位 CPU）按钮。

3.6.2 执行程序

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


程序将一直执行，直到到达断点为止。在 [S/W Breakpoints]（软件断点）列中将显示一个箭头，指示程序停止的位置。



Line	Sour...	E..	C..	S..	Source
23					long a[10];
24					long j;
25					int i;
26					struct Sample far *p_sam;
27					
28	F82E9				p_sam= malloc(sizeof(struct Sample));
29	F82FD				init(p_sam);
30					
31	F8309				for(i=0; i<10; i++){
32	F8314				j = rand();
33	F8320				if(j < 0){
34	F8325				j = -j;
35					}
36	F832E				a[i] = j;
37					}
38	F8344			●	sort(a);
39	F834B				change(a);
40					
41	F8352				p_sam->s0=a[0];
42	F8362				p_sam->s1=a[1];
43	F8376				p_sam->s2=a[2];
44	F838C				p_sam->s3=a[3];
45	F83A2				p_sam->s4=a[4];
46	F83B8				p_sam->s5=a[5];
47	F83CE				p_sam->s6=a[6];
48	F83E4				p_sam->s7=a[7];
49	F83FA				p_sam->s8=a[8];
50	F8410				p_sam->s9=a[9];
51	F8426				free(p_sam);
52	F8432				p_sam = NULL;

图 3.5 [Editor]（编辑器）窗口（程序位于断点处）

使用 [Status]（状态）窗口，用户可以检查上次发生中断的原因。

从 [View]（视图）菜单选择 [CPU] → [Status]（状态），或者单击 [View Status]（视图状态）工具栏按钮 。显示 [Status]（状态）窗口后，打开其中的 [Target]（目标）表，并进行检查。

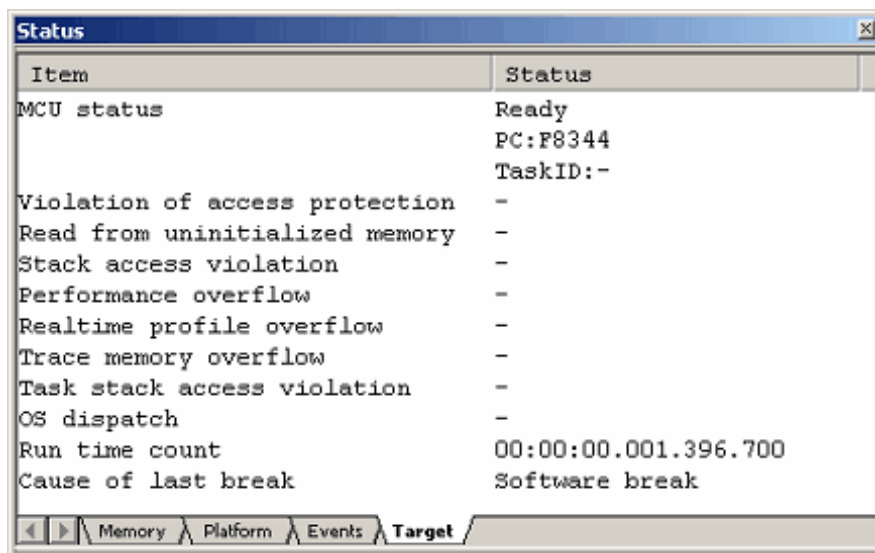


图 3.6 [Status]（状态）窗口

注意：对于每个产品，该窗口中显示的内容会有所不同。有关为每个产品显示的内容的详细信息，请参考第 4 章“准备调试”或在线帮助。

3.7 检查断点

使用 [Breakpoints]（断点）对话框可检查所有软件断点设置。

3.7.1 检查断点

按下 PC 键盘上的 Ctrl + B 键。此时将显示如下所示的 [Breakpoints]（断点）对话框。

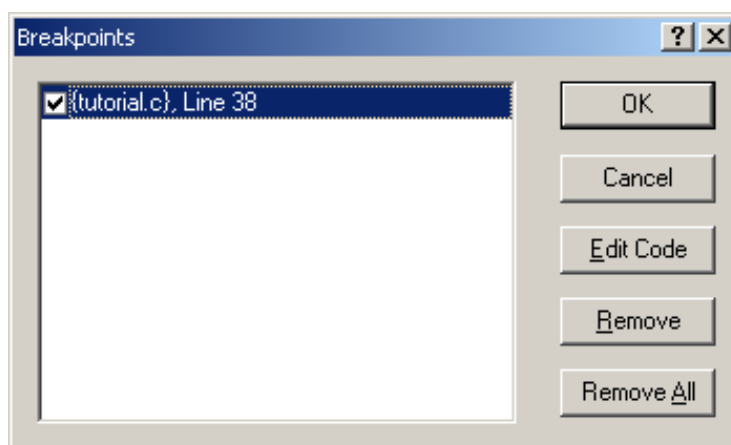

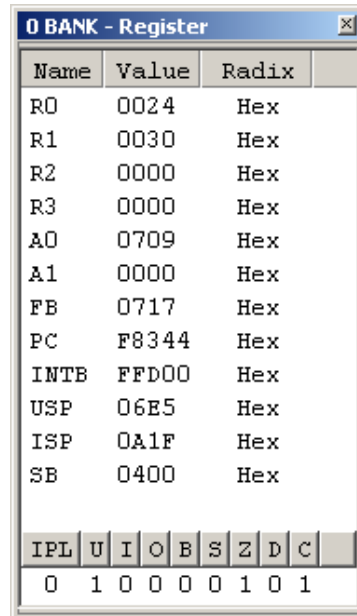


图 3.7 [Breakpoints]（断点）对话框

使用该对话框可以删除某个断点或者启用 / 禁用某个断点。

3.8 更改寄存器内容

从 [View] (视图) 菜单选择 [CPU] → [Registers] (寄存器), 或者单击 [Registers] (寄存器) 工具栏按钮 。将显示下面所示的 [Register] (寄存器) 窗口。



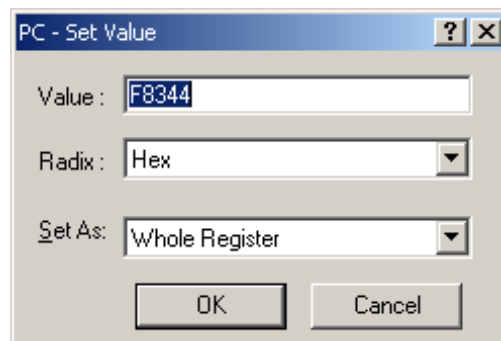
Name	Value	Radix
R0	0024	Hex
R1	0030	Hex
R2	0000	Hex
R3	0000	Hex
A0	0709	Hex
A1	0000	Hex
FB	0717	Hex
PC	F8344	Hex
INTB	FFD00	Hex
USP	06E5	Hex
ISP	0A1F	Hex
SB	0400	Hex

Below the table, there are control buttons: IPL, U, I, O, B, S, Z, D, C. Below these buttons is a row of binary digits: 0 1 0 0 0 0 1 0 1.

图 3.8 [Register] (寄存器) 窗口

可以更改任何寄存器的内容。

双击要更改的寄存器所在的行。此时将显示如下所示的对话框, 以使用户输入更改寄存器所使用的新值。




The dialog box 'PC - Set Value' contains the following fields and controls:

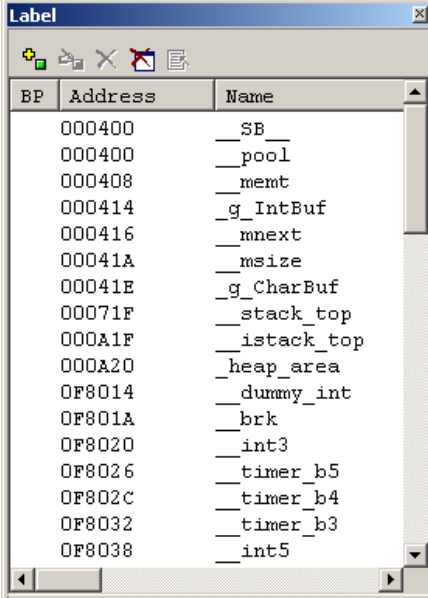
- Value:** A text input field containing 'F8344'.
- Radix:** A dropdown menu currently set to 'Hex'.
- Set As:** A dropdown menu currently set to 'Whole Register'.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

图 3.9 [Set value] (设置值) 对话框 (PC)

3.9 引用符号

使用 [Label]（标签）窗口，可以显示包含在模块中的符号信息。

从 [View]（视图）菜单选择 [Symbol]（符号）→ [Label]（标签），或者单击 [Label]（标签）工具栏按钮 。将显示如下所示的 [Label]（标签）窗口。使用该窗口可以查看包含在模块中的符号信息。




BP	Address	Name
	000400	__SB__
	000400	__pool
	000408	__ment
	000414	__g_IntBuf
	000416	__mnext
	00041A	__msize
	00041E	__g_CharBuf
	00071F	__stack_top
	000A1F	__istack_top
	000A20	__heap_area
	0F8014	__dummy_int
	0F801A	__brk
	0F8020	__int3
	0F8026	__timer_b5
	0F802C	__timer_b4
	0F8032	__timer_b3
	0F8038	__int5

图 3.10 [Label]（标签）窗口

3.10 检查存储器内容

通过指定一个标签名，可以在 [Memory]（存储器）窗口中检查寄存标签的存储器的内容。例如，用户可以以字节大小的形式检查与 `_main` 相应的存储器内容，如下所示。

从 [View]（视图）菜单选择 [CPU] → [Memory]（存储器）或者单击 [Memory]（存储器）工具栏按钮  以显示 [Display Address]（显示地址）对话框。

在 [Display Address]（显示地址）对话框的编辑框中输入 “`_main`”。

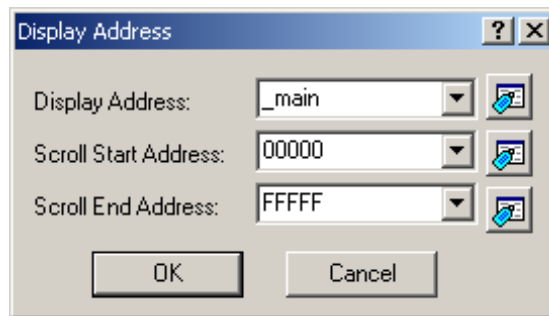
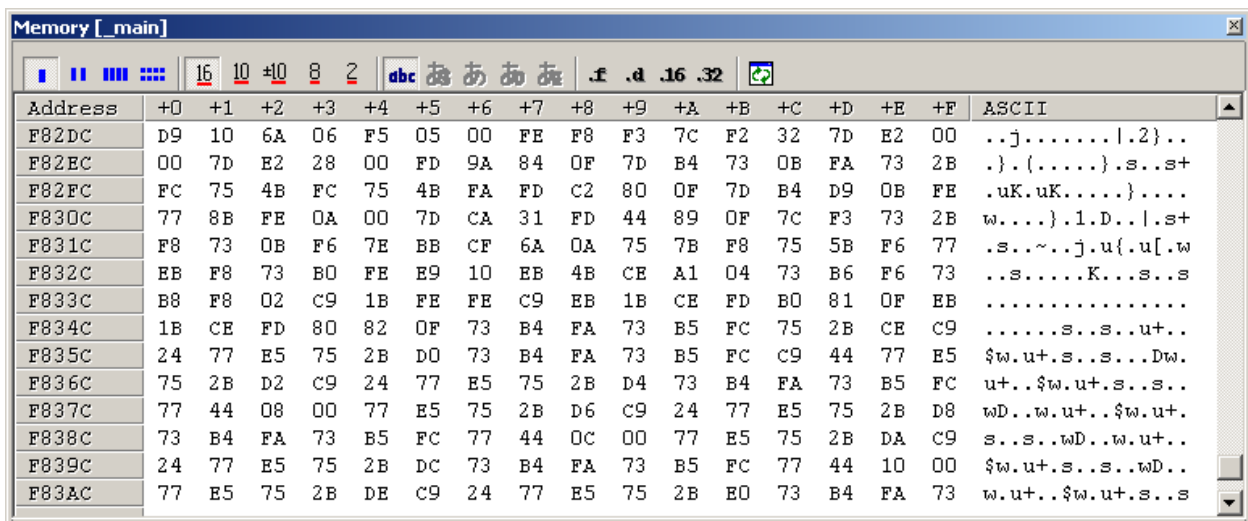


图 3.11 [Display Address]（显示地址）对话框

单击 [OK]（确定）按钮。此时将显示 [Memory]（存储器）窗口，窗口中显示了指定的存储区。



Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F	ASCII
F82DC	D9	10	6A	06	F5	05	00	FE	F8	F3	7C	F2	32	7D	E2	00	..j..... .2}..
F82EC	00	7D	E2	28	00	FD	9A	84	0F	7D	B4	73	0B	FA	73	2B	..}{.....}.s..s+
F82FC	FC	75	4B	FC	75	4B	FA	FD	C2	80	0F	7D	B4	D9	0B	FE	.uK.uK.....}....
F830C	77	8B	FE	0A	00	7D	CA	31	FD	44	89	0F	7C	F3	73	2B	w....}.1.D.. .s+
F831C	F8	73	0B	F6	7E	BB	CF	6A	0A	75	7B	F8	75	5B	F6	77	.s...~.j.u{.u[.w
F832C	EB	F8	73	B0	FE	E9	10	EB	4B	CE	A1	04	73	B6	F6	73	..s.....K.....s
F833C	B8	F8	02	C9	1B	FE	FE	C9	EB	1B	CE	FD	B0	81	0F	EBs.....s
F834C	1B	CE	FD	80	82	0F	73	B4	FA	73	B5	FC	75	2B	CE	C9s.....s+..
F835C	24	77	E5	75	2B	D0	73	B4	FA	73	B5	FC	C9	44	77	E5	\$w.u+.s.....Dw.
F836C	75	2B	D2	C9	24	77	E5	75	2B	D4	73	B4	FA	73	B5	FC	u+..\$w.u+.s.....
F837C	77	44	08	00	77	E5	75	2B	D6	C9	24	77	E5	75	2B	D8	wD..w.u+..\$w.u+
F838C	73	B4	FA	73	B5	FC	77	44	0C	00	77	E5	75	2B	DA	C9	s...wD..w.u+..
F839C	24	77	E5	75	2B	DC	73	B4	FA	73	B5	FC	77	44	10	00	\$w.u+.s.....wD..
F83AC	77	E5	75	2B	DE	C9	24	77	E5	75	2B	E0	73	B4	FA	73	w.u+..\$w.u+.s...s

图 3.12 [Memory]（存储器）窗口

3.11 引用变量

在单步执行程序时，用户可以看到逐步执行源行或指令时在程序中使用的变量的值将如何变化。例如，从下面描述的步骤，可以看到在程序起始处声明的 long 类型的数组 “a”。

单击 [Editor]（编辑器）窗口中显示的数组 “a” 的左侧，并将光标放置在那里。用鼠标右键选择 [Instant Watch]（即时监视）。此时将显示如下所示的对话框。

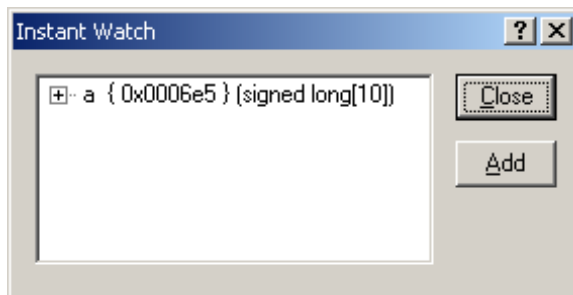


图 3.13 [Instant Watch]（即时监视）对话框

单击 [Add]（添加）按钮向 [Watch]（监视）窗口中添加一个变量。

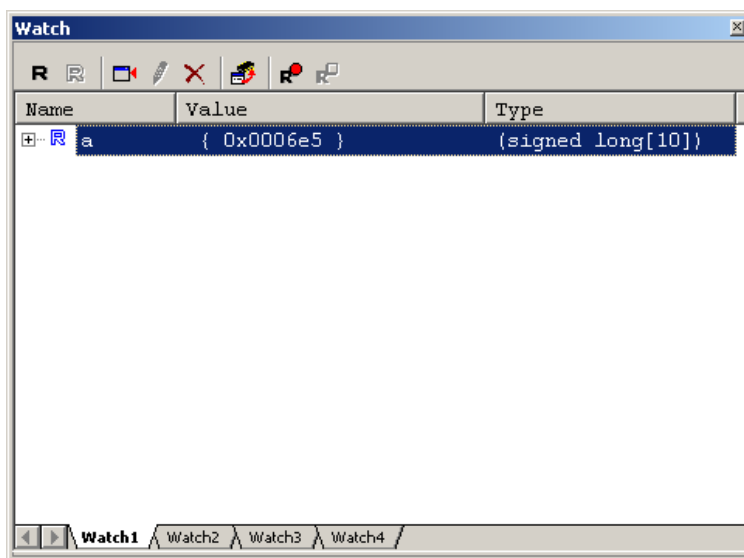


图 3.14 [Watch]（监视）窗口（显示数组）

或者可以指定一个变量名，向 [Watch]（监视）窗口中添加一个变量。在 [Watch]（监视）窗口中单击鼠标右键，从上下文菜单中选择 [Add Watch]（添加监视）。此时将显示如下所示的对话框。

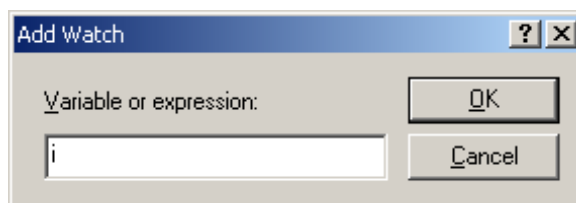


图 3.15 [Add Watch]（添加监视）对话框

在 [Variable or Expression]（变量或表达式）编辑框中输入变量“i”并单击 [OK]（确定）按钮。此时 [Watch]（监视）窗口中将显示一个 int 类型的变量“i”。

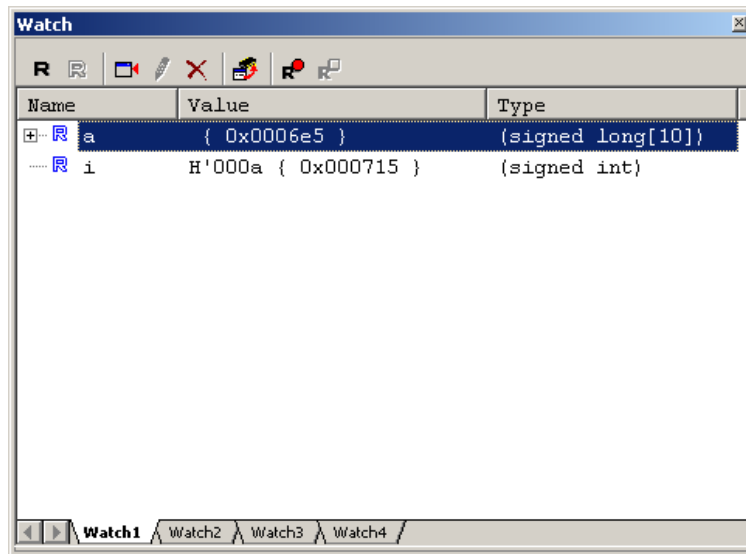


图 3.16 [Watch]（监视）窗口（显示一个变量）

在 [watch]（监视）窗口中单击数组“a”左侧显示的“+”标记，可以查看数组“a”的每一个元素。

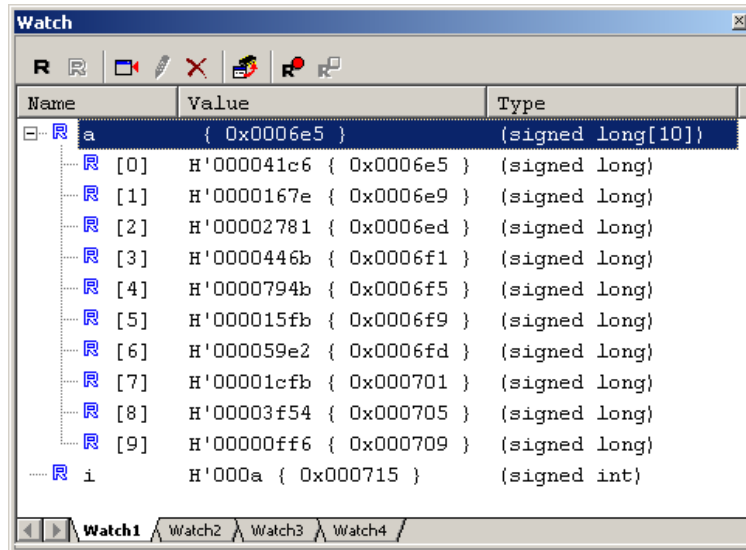



图 3.17 [Watch]（监视）窗口（显示数组元素）

3.12 显示局部变量

使用 [Local]（局部变量）窗口可以显示函数中包含的局部变量。以检查 tutorial 函数的局部变量为例。该函数声明三个局部变量 “j”、“i” 和 “p_sam”。

从 [View]（视图）菜单选择 [Symbol]（符号）→ [Local]（局部变量）或者单击 [Locals]（局部变量）工具栏按钮  以显示 [Locals]（局部变量）窗口。

[Locals]（局部变量）窗口显示当前程序计数器（PC）指示的函数中的局部变量及其值。

如果该函数中不存在变量，则 [Locals]（局部变量）窗口不显示任何信息。

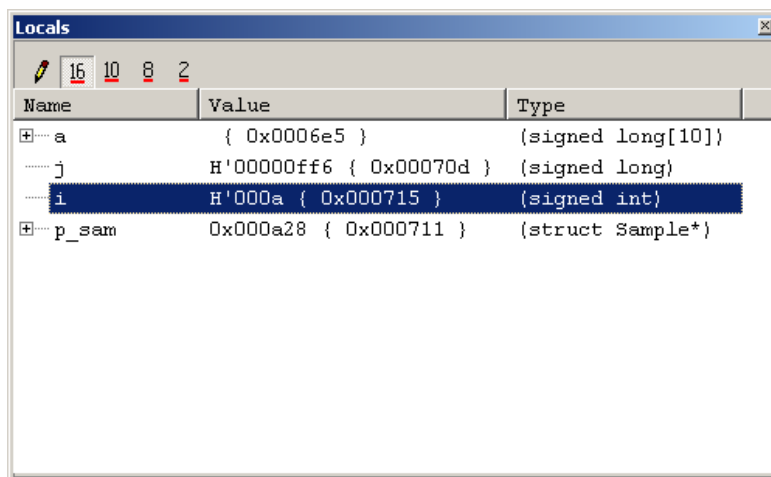


图 3.18 [Locals]（局部变量）窗口

在 [Local]（局部变量）窗口中单击类实例 p_sam 左侧的 “+” 标记来显示组成类实例 p_sam 的元素。执行 sort 函数之前和之后查看类实例 p_sam 的元素，以确认随机数据是否按降序排序。

3.13 单步执行程序

HEW 提供了多种在调试程序时非常有用的步进命令。

表 3.1 步进选项

项目编号	命令	描述
1	Step In（跳入）	每次执行程序的一个语句（包括函数中的语句）。
2	Step Over（跳过）	通过“跳过”函数调用（如果有），每次执行程序的一个语句。
3	Step Out（跳出）	退出函数后，在程序中调用该函数的下一语句处停止。
4	Step...（单步...）	以指定速度，单步执行程序指定次数。

3.13.1 执行 [Step In]（跳入）命令

[Step In]（跳入）命令“单步执行”被调用函数，并在被调用函数的第一个语句处停止。

要进入 sort 函数，请从 [Debug]（调试）菜单选择 [Step In]（跳入），或者单击工具栏上的 [Step In]（跳入）按钮。



图 3.19 [Step In]（跳入）按钮

```

Line Sour... E. C. S. Source
8      init(struct Sample *p_sam)
9      {
10     p_sam->s0 = 0;
11     p_sam->s1 = 0;
12     p_sam->s2 = 0;
13     p_sam->s3 = 0;
14     p_sam->s4 = 0;
15     p_sam->s5 = 0;
16     p_sam->s6 = 0;
17     p_sam->s7 = 0;
18     p_sam->s8 = 0;
19     p_sam->s9 = 0;
20     }
21
22     //-----
23     int g_IntBuf;
24     char g_CharBuf;
25     //-----
26
27     sort(long *a)
28     {
29         long t;
30         int i, j, k, gap;
31
32         gap = 5;
33         while( gap > 0 ){
34             for( k=0; k<gap; k++){
35                 for( i=k+gap; i<10; i=i+gap ){
36                     for(j=i-gap; j>=k; j=j-gap){
37                         g_IntBuf = j;
38                         if(a[j]>a[j+gap]){
39                             t = a[j];
40                             a[j] = a[j+gap];
41                             a[j+gap] = t;
42                         }
43                         else
44                             break;
45                     }
46                 }
47             }
48             gap = gap/2;
49         }
50         g_CharBuf = (char)g_IntBuf & 0x00FF;
51     }

```

图 3.20 [Editor]（编辑器）窗口（[Step In]（跳入））

[Editor]（编辑器）窗口中的突出显示移至 sort 函数的第一个语句。

3.13.2 执行 [Step Out]（跳出）命令

[Step Out]（跳出）命令通过快速执行来退出被调用函数，在调用该函数的程序的下一语句处停止。

要退出 sort 函数，请从 [Debug]（调试）菜单选择 [Step Out]（跳出），或者单击工具栏上的 [Step Out]（跳出）按钮。



图 3.21 [Step Out]（跳出）按钮

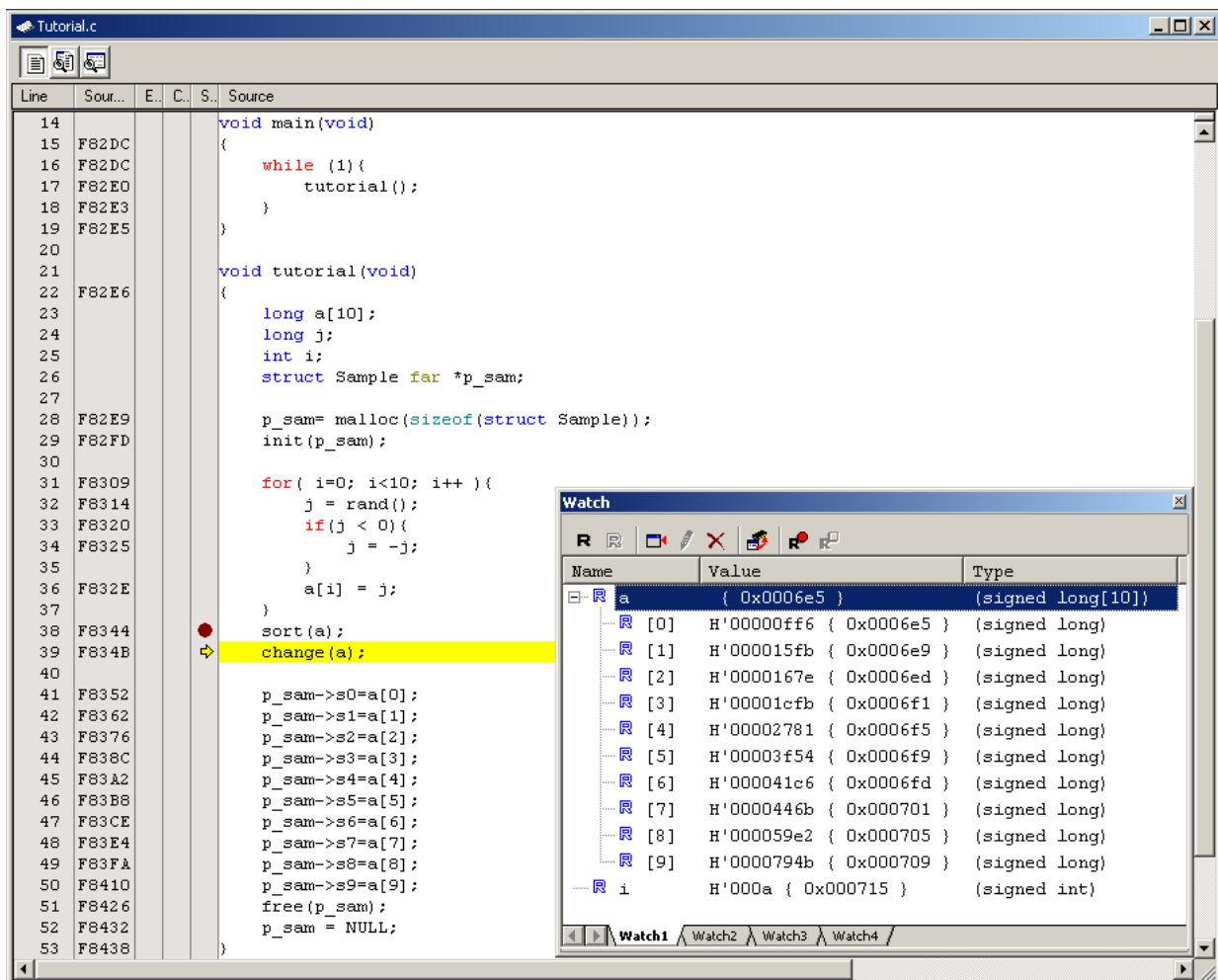


图 3.22 [Editor]（编辑器）窗口（[Step Out]（跳出））

[Watch]（监视）窗口中显示的变量“a”的数据将以升序排序。

3.13.3 执行 [Step Over]（跳过）命令

[Step Over]（跳过）命令将整个函数调用作为单步执行，然后在主程序的下一语句处停止。

要一次执行 change 函数中的所有语句，请从 [Debug]（调试）菜单选择 [Step Over]（跳过），或者单击工具栏上的 [Step Over]（跳过）按钮。



图 3.23 [Step Over]（跳过）按钮

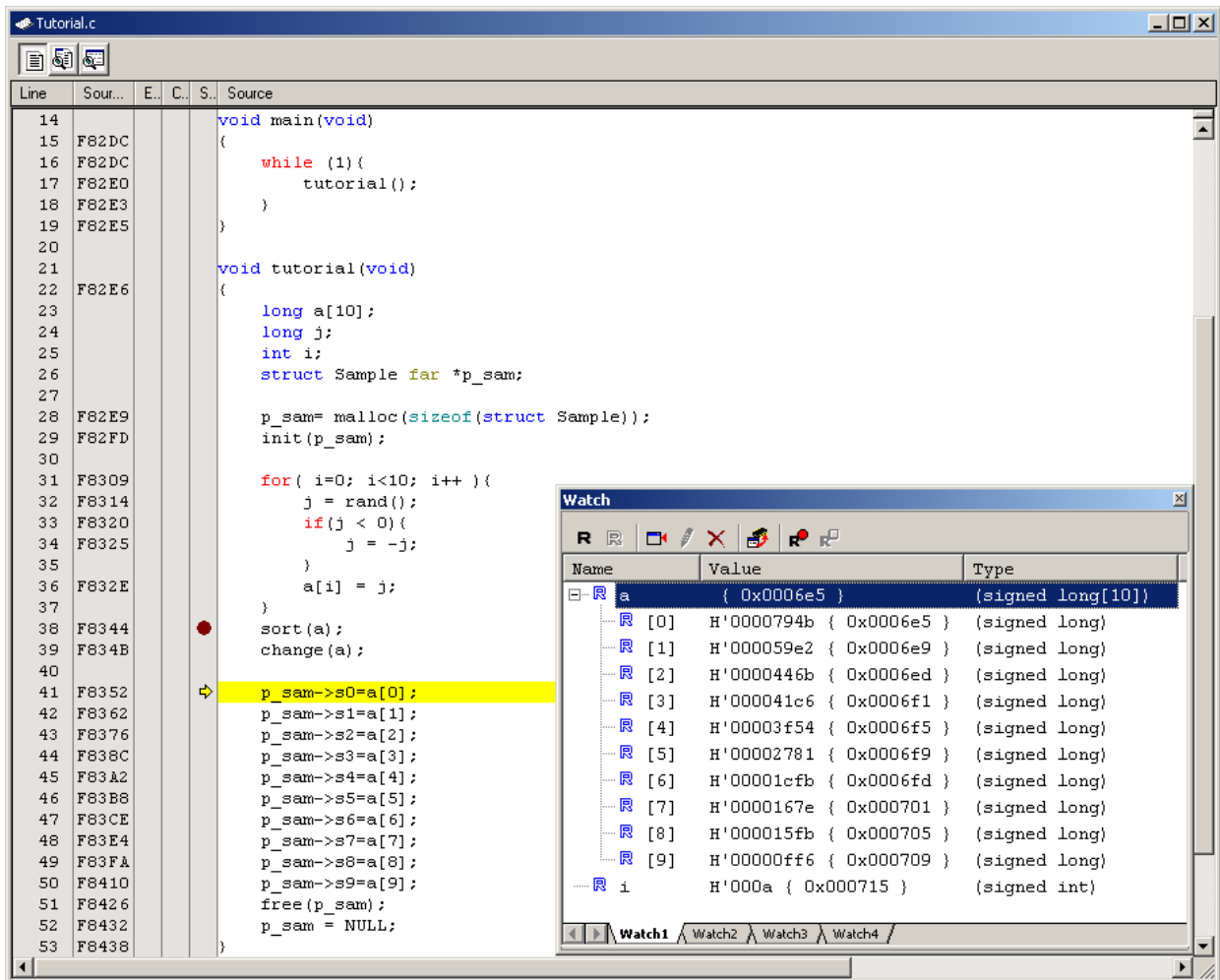


图 3.24 [Editor]（编辑器）窗口（[Step Over]（跳过））

[Watch]（监视）窗口中显示的变量“a”的数据将以降序排序。

3.14 强行中断程序

HEW 允许用户强行中断程序。

清除所有断点。

要执行 tutorial 函数的其余部分，请从 [Debug]（调试）菜单选择 [Go]（执行），或者单击工具栏上的 [Go]（执行）按钮。



图 3.25 [Go]（执行）按钮

由于程序正在执行一个无限循环过程，请从 [Debug]（调试）菜单选择 [Stop Program]（停止程序），或者单击工具栏上的 [Halt]（停止）按钮。



图 3.26 [Halt]（停止）按钮

3.15 硬件断点工具

在程序执行指定的地址（指令取回）或者读取或写入指定的内存位置（数据存取）时，硬件断点会导致程序停止。

3.15.1 在程序执行指定地址时停止程序

使用 [Editor]（编辑器）窗口，用户可以轻松地设置指令取回事件。例如，可以在 sort 函数调用处设置一个指令取回事件。

在 [Event]（事件）列中，双击与包含 sort 函数调用的源行对应的行。

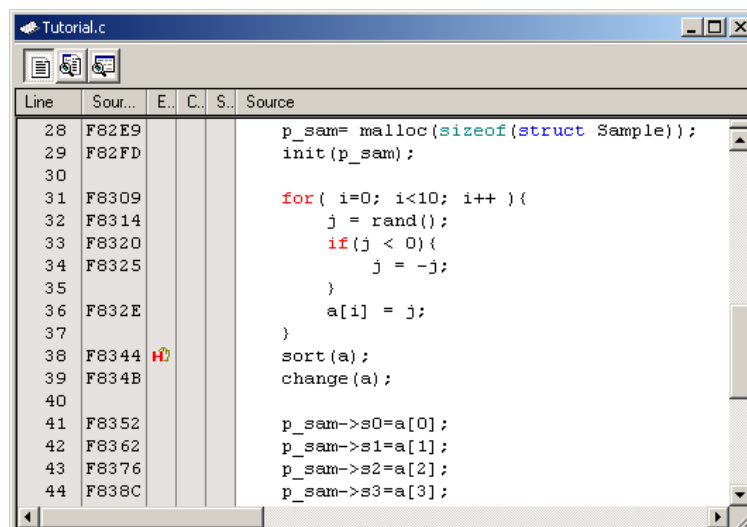


图 3.27 [Editor]（编辑器）窗口（设置硬件断点）

包含 sort 函数的源行将用  标记，指示此处设置了在程序取回指令时将导致程序停止的硬件断点。

3.16 在程序存取存储器时停止程序

要在程序读取全局变量或向全局变量写入值时停止程序，请按如下所述设置硬件断点。

从 [View]（视图）菜单选择 [Event]（事件）→ [Hardware Break]（硬件断点）以显示 [Hardware Break]（硬件断点）对话框。

打开 [Hardware Break]（硬件断点）对话框的 [OR] 页。在 [Editor]（编辑器）窗口中，选择一个要作为硬件断点的对象的全局变量，以便在程序读取变量或向变量写入值时停止程序，并将选中的变量拖放到 [OR] 页。

然后单击 [Apply]（应用）按钮。

运行程序后，该程序将在读取用户设置的全局变量的值或向该变量写入值时停止运行。

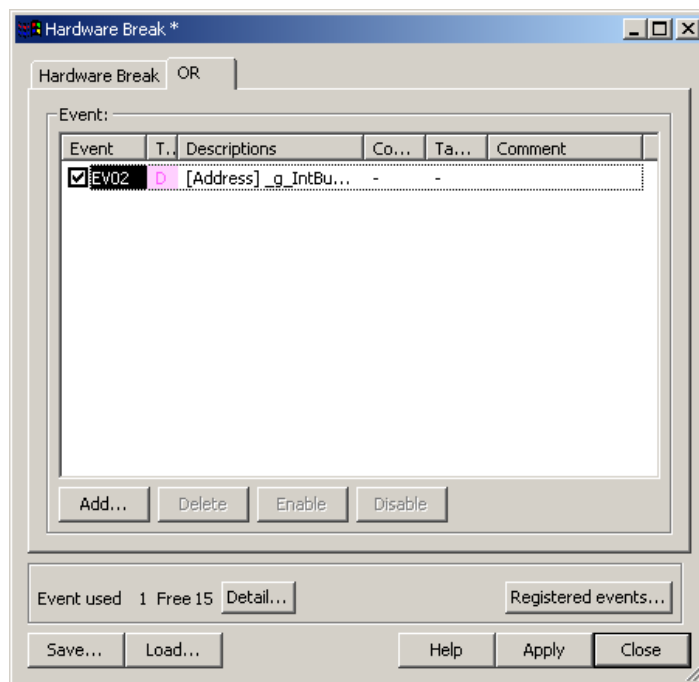



图 3.28 [Hardware Break]（硬件断点）对话框

- 注意：
1. 只能设置大小为 1 字节或 2 字节的全局变量。
 2. 不能设置局部变量。

3.17 跟踪工具

E100 仿真器的跟踪工具拥有被称作“跟踪存储器”的特殊存储器，该存储器最大可容纳 4M 总线周期的执行记录，并且总是在程序执行期间更新。跟踪存储器的内容显示在 [Trace]（跟踪）窗口中。

从 [View]（视图）菜单选择 [Code]（代码）→ [Trace]（跟踪），或者单击 [Trace]（跟踪）工具栏按钮 。

将显示如下所示的 [Trace]（跟踪）窗口。

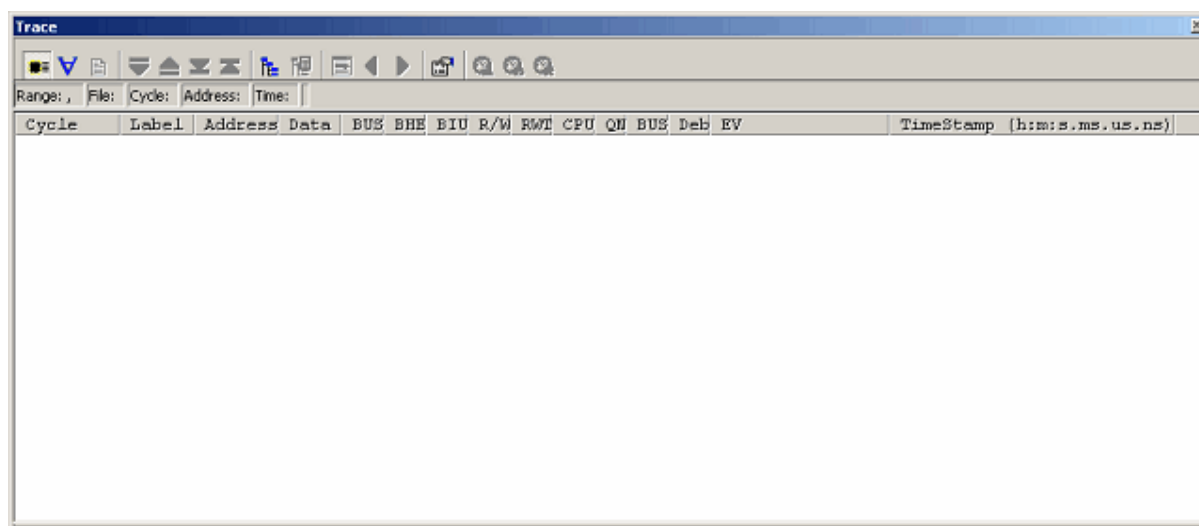


图 3.29 [Trace]（跟踪）窗口

下面略述跟踪工具并介绍如何设置。

3.17.1 显示由 Fill Until Stop（填充直至停止）获取的跟踪信息

空闲跟踪工具将逐次获取从用户程序开始运行到中断时的跟踪信息。

- (1) 清除所有断点条件。在 [Trace]（跟踪）窗口中的任意位置单击鼠标右键，从显示的上下文菜单中选择 [Acquisition]（获取）。将显示如下所示的 [Trace conditions]（跟踪条件）对话框。查看选中的跟踪模式是否为 [Fill until stop]（填充直至停止）。单击 [Close]（关闭）按钮。

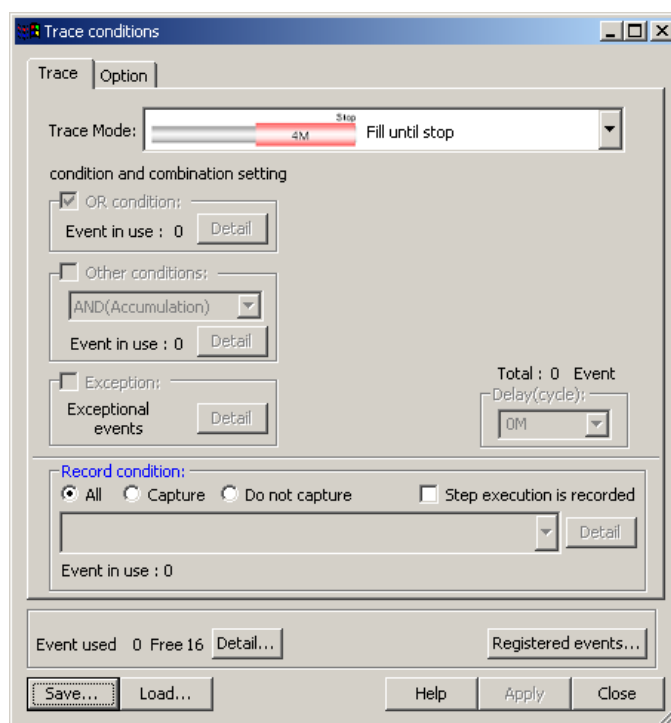


图 3.30 [Trace conditions]（跟踪条件）对话框（空闲跟踪）

- (2) 在 tutorial 函数中写入 `p_sam ->s0=a[0];` 的行处设置一个软件断点。
- (3) 从 [Debug]（调试）菜单选择 [Reset Go]（复位执行）。断点将会中断处理，从开始到断点的跟踪信息将显示在 [Trace]（跟踪）窗口中。

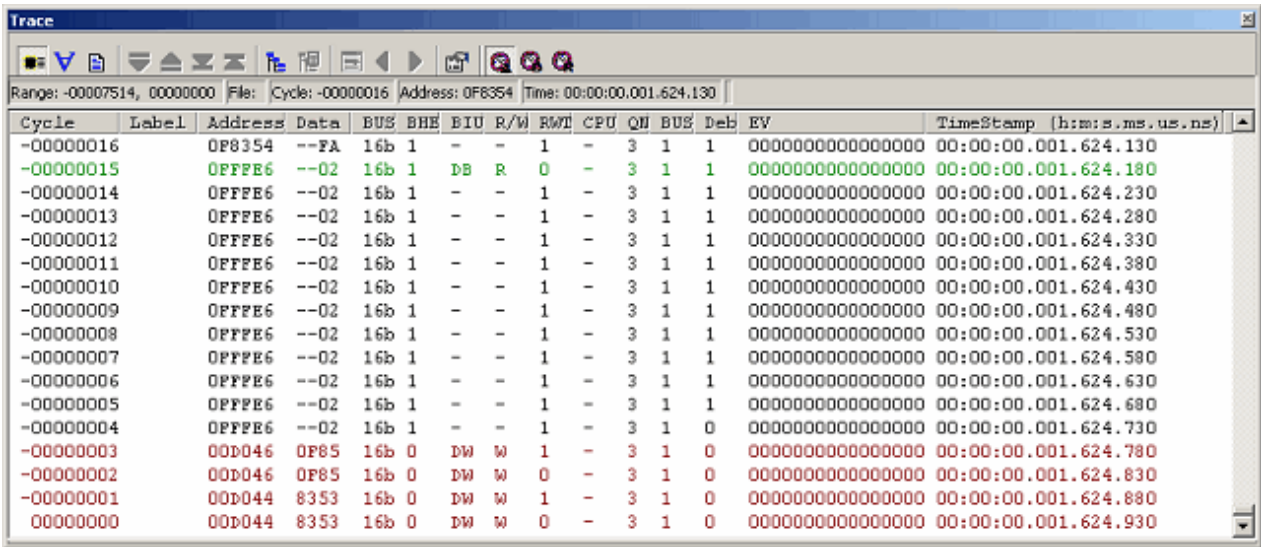


图 3.31 [Trace] (跟踪) 窗口 (空闲跟踪)

- (4) 可以混合显示总线、反汇编和 / 或源。从上下文菜单选择 [Display Mode] (显示模式) → DIS, 可以在总线和反汇编的混合模式下显示跟踪信息。

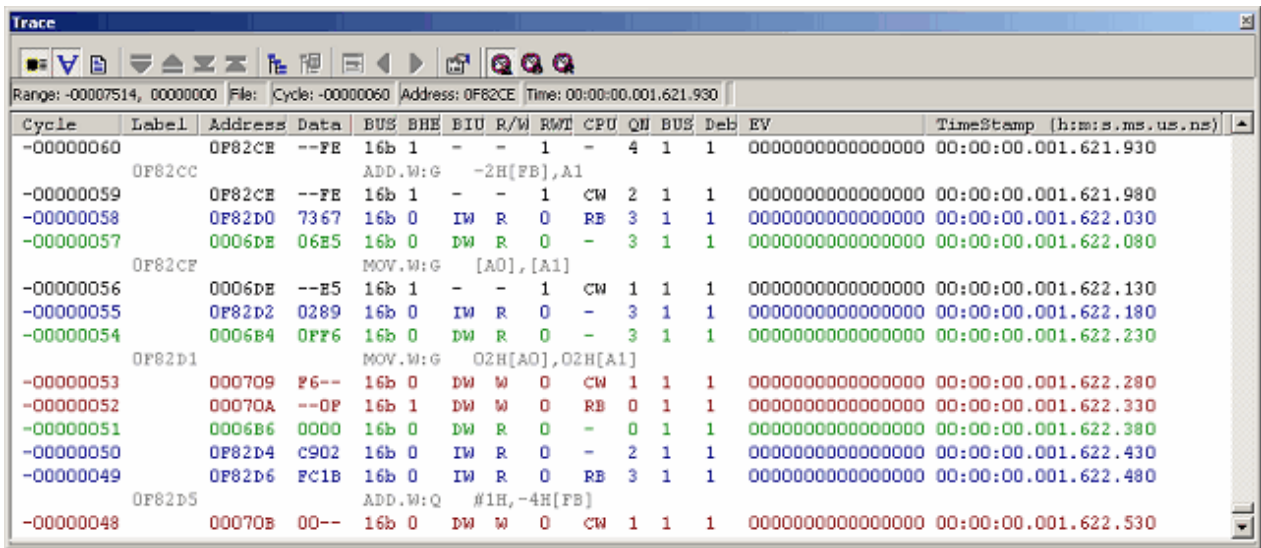


图 3.32 [Trace] (跟踪) 窗口 (总线和反汇编混合显示)

(5) 此外，从上下文菜单选择 [Display Mode]（显示模式）→ [SRC]，可以在总线、反汇编和源混合模式下显示跟踪信息。

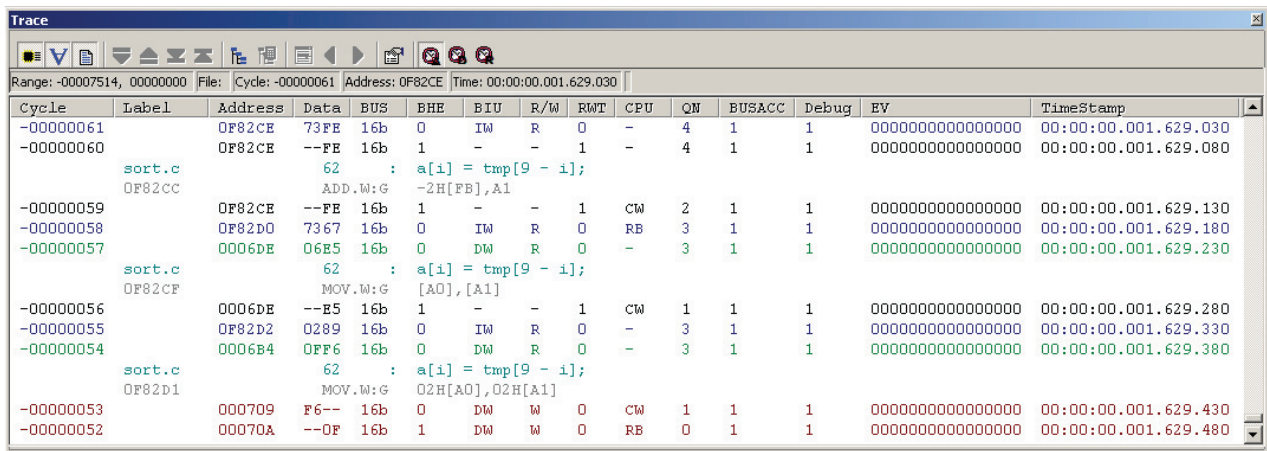


图 3.33 [Trace]（跟踪）窗口（总线、反汇编和源混合显示）

3.17.2 显示由 Fill around TP（围绕 TP 填充）获取的跟踪信息

点和延迟工具在遇到跟踪点之后的指定数量的周期处停止获取跟踪信息。此工具允许用户在不中断用户程序的情况下利用跟踪信息了解程序流。

- (1) 如果设置了任何断点条件，请全部清除这些条件。
- (2) 在 [Trace conditions]（跟踪条件）对话框中选择 [Fill around TP]（围绕 TP 填充）作为跟踪模式。在 [Delay Value (Cycles)]（延迟值（周期数））列中，指定 [4M]。（将从遇到跟踪点处开始最多获取 4M 周期的跟踪信息。）

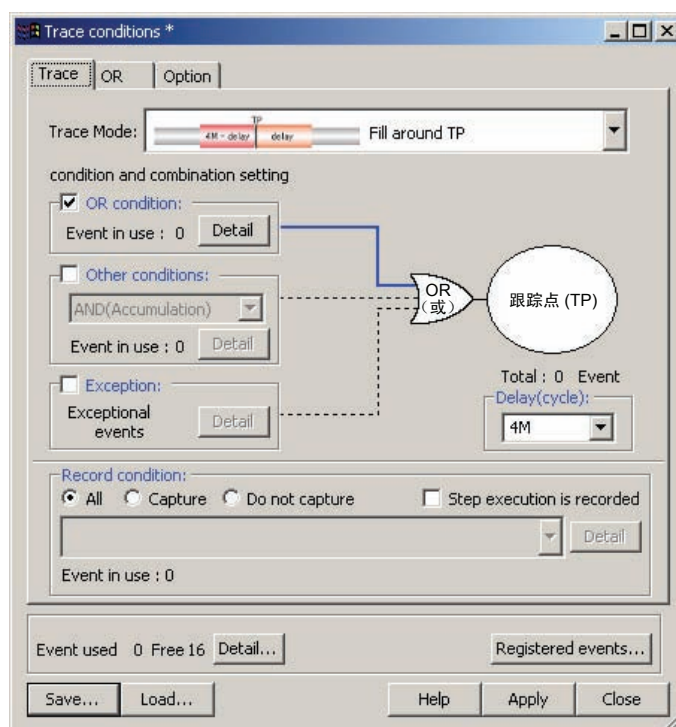


图 3.34 [Trace conditions]（跟踪条件）对话框（[Fill around TP]（围绕 TP 填充））

- (3) 接下来，在调试器开始获取跟踪信息处设置一个跟踪点。打开 [Trace conditions]（跟踪条件）对话框的 [OR] 页。在 [Editor]（编辑器）窗口中选择 main 函数，并将其拖放到 [OR] 页。单击 [Apply]（应用）按钮，然后单击 [Close]（关闭）按钮。这样，调试器将从执行 main 函数时开始获取跟踪信息。

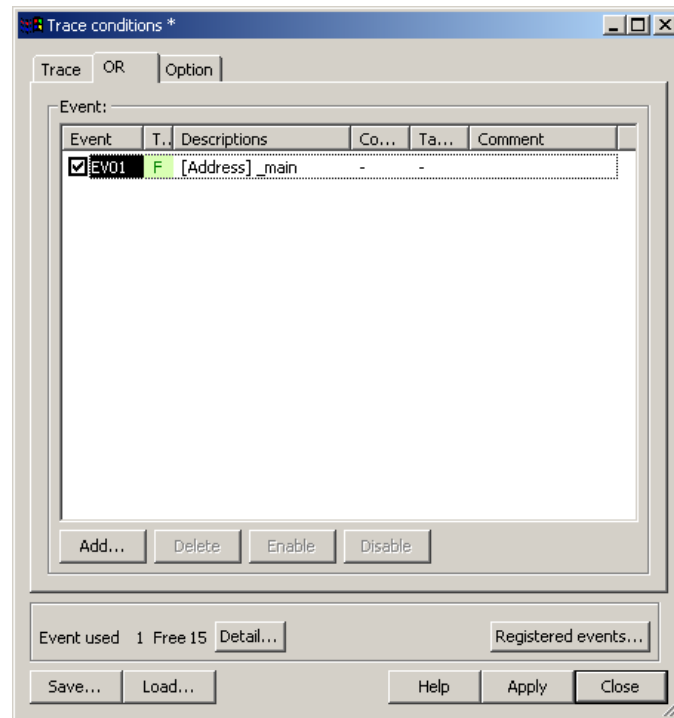


图 3.35 [Trace conditions]（跟踪条件）对话框（[OR] 页）

- (4) 从 [Debug]（调试）菜单选择 [Reset Go]（复位执行）。到达跟踪点之后很短时间内，[Trace]（跟踪）窗口将显示如下所示的跟踪内容。

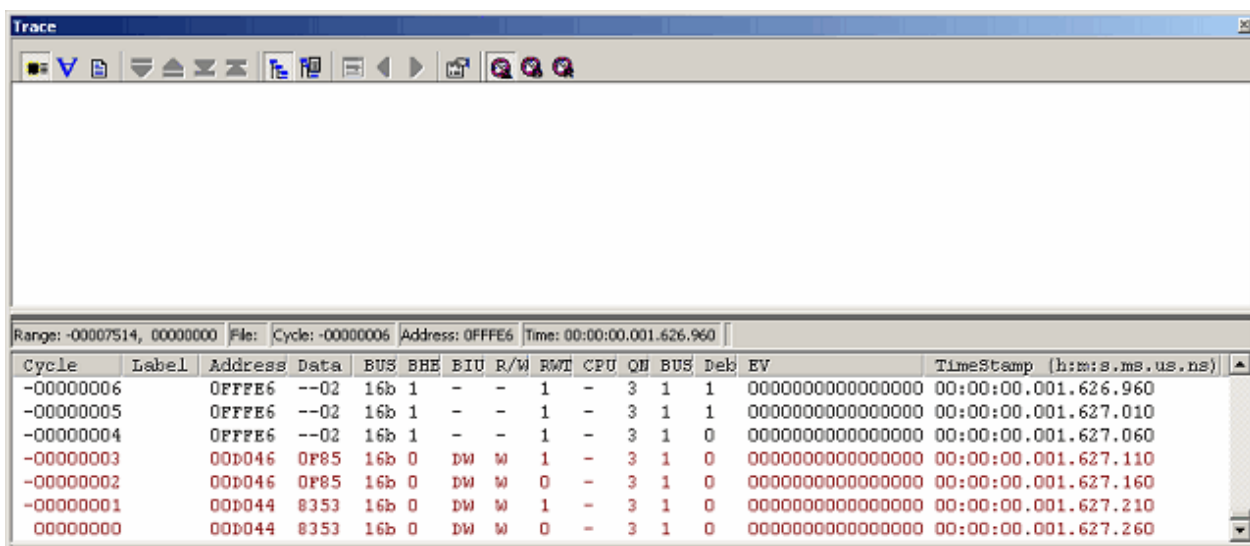
Cycle	Label	Address	Data	BUS	BHE	BIU	R/W	RWT	CPU	QM	BUS	Deb	EV	TimeStamp (h:m:s.ms.us.ns)
-00000001		00071E	--0F	16b	1	-	-	1	-	4	1	1	0000000000000000	00:00:00.001.276.690
00000000		00071E	--0F	16b	1	-	-	1	CW	2	1	1	0000000000000001	00:00:00.001.276.740
00000001		0F82E0	05F5	16b	0	IW	R	0	CB	3	1	1	0000000000000000	00:00:00.001.276.790
00000002		0F82E0	--F5	16b	1	-	-	1	RB	2	1	1	0000000000000000	00:00:00.001.276.840
	0F82E0					JSR.W			_tutoria					
00000003		0F82E2	FE00	16b	0	IW	R	0	CB	3	1	1	0000000000000000	00:00:00.001.276.890
00000004		0F82E2	--00	16b	1	-	-	1	RW	1	1	1	0000000000000000	00:00:00.001.276.940
00000005		0F82E2	--00	16b	1	-	-	1	QC	0	1	1	0000000000000000	00:00:00.001.276.990
00000006	_tutor	0F82E6	F27C	16b	0	IW	R	0	-	2	1	1	0000000000000000	00:00:00.001.277.040
00000007		0F82E8	7b32	16b	0	IW	R	0	-	4	1	1	0000000000000000	00:00:00.001.277.090
00000008		000719	E3--	16b	0	DW	W	0	-	4	1	1	0000000000000000	00:00:00.001.277.140
00000009		00071A	--82	16b	1	DW	W	0	-	4	1	1	0000000000000000	00:00:00.001.277.190
00000010		00071B	0F--	16b	0	DB	W	0	-	4	1	1	0000000000000000	00:00:00.001.277.240
00000011		00071B	0F--	16b	0	-	-	1	-	4	1	1	0000000000000000	00:00:00.001.277.290
	0F82E6					_tutorial ENTER			#32H					
00000012		00071B	0F--	16b	0	-	-	1	CW	2	1	1	0000000000000000	00:00:00.001.277.340
00000013		0F82EA	00E2	16b	0	IW	R	0	-	4	1	1	0000000000000000	00:00:00.001.277.390

图 3.36 [Trace]（跟踪）窗口（[Fill around TP]（围绕 TP 填充））

3.17.3 显示函数执行历史记录

从获取的跟踪信息中可以显示函数执行历史记录。

- (1) 清除所有断点条件。在 [Trace]（跟踪）窗口中的任意位置单击鼠标右键，从显示的上下文菜单中选择 [Acquisition]（获取）。将显示 [Trace conditions]（跟踪条件）对话框。将跟踪模式切换到 [Fill until stop]（填充直至停止）并单击 [Apply]（应用）按钮。然后单击 [Close]（关闭）按钮。
- (2) 在 tutorial 函数中写入 p_sam -> s0=a[0]; 的行处设置一个软件断点。
- (3) 从 [Debug]（调试）菜单选择 [Reset Go]（复位执行）。断点将会中断处理，从开始到断点的跟踪信息将显示在 [Trace]（跟踪）窗口中。
- (4) 在 [Trace]（跟踪）窗口中的任意位置单击鼠标右键，从显示的上下文菜单中选择 [Function Execution History]（函数执行历史记录）→ [Function Execution History]（函数执行历史记录）。



Cycle	Label	Address	Data	BUS	BHE	BIU	R/W	RWT	CPU	QB	BUS	Dek	EV	TimeStamp (h:m:s.ms.us.ns)
-00000006		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.626.960
-00000005		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.010
-00000004		0FFFE6	--02	16b	1	-	-	1	-	3	1	0	0000000000000000	00:00:00.001.627.060
-00000003		00D046	0F85	16b	0	DW	W	1	-	3	1	0	0000000000000000	00:00:00.001.627.110
-00000002		00D046	0F85	16b	0	DW	W	0	-	3	1	0	0000000000000000	00:00:00.001.627.160
-00000001		00D044	0353	16b	0	DW	W	1	-	3	1	0	0000000000000000	00:00:00.001.627.210
00000000		00D044	0353	16b	0	DW	W	0	-	3	1	0	0000000000000000	00:00:00.001.627.260

图 3.37 [Trace]（跟踪）窗口（函数执行历史记录 — 分析前）

- (5) 在显示的函数执行历史记录窗口中的任意位置单击鼠标右键，从上下文菜单中选择 [Analyze Execution History]（分析执行历史记录）。函数执行历史记录将显示在 [Trace]（跟踪）窗口的上窗格中。



图 3.38 [Trace]（跟踪）窗口（函数执行历史记录 - 分析后）

- (6) 在显示的函数执行历史记录中双击任意函数，对应于该函数的跟踪信息将显示在 [Trace]（跟踪）窗口的下窗格中。

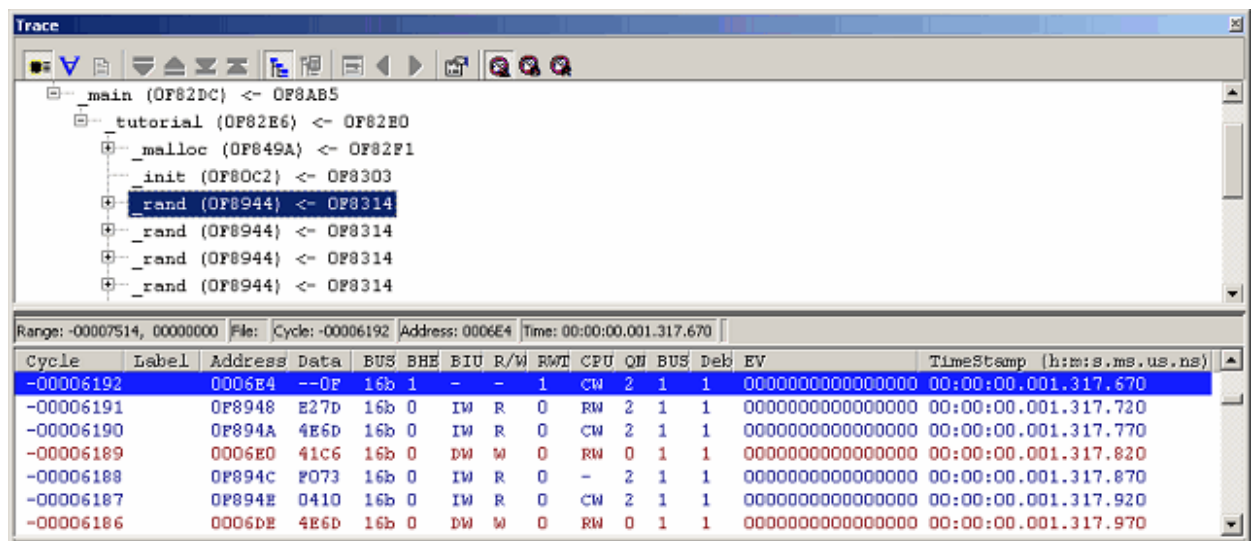



图 3.39 [Trace]（跟踪）窗口（函数执行历史记录）

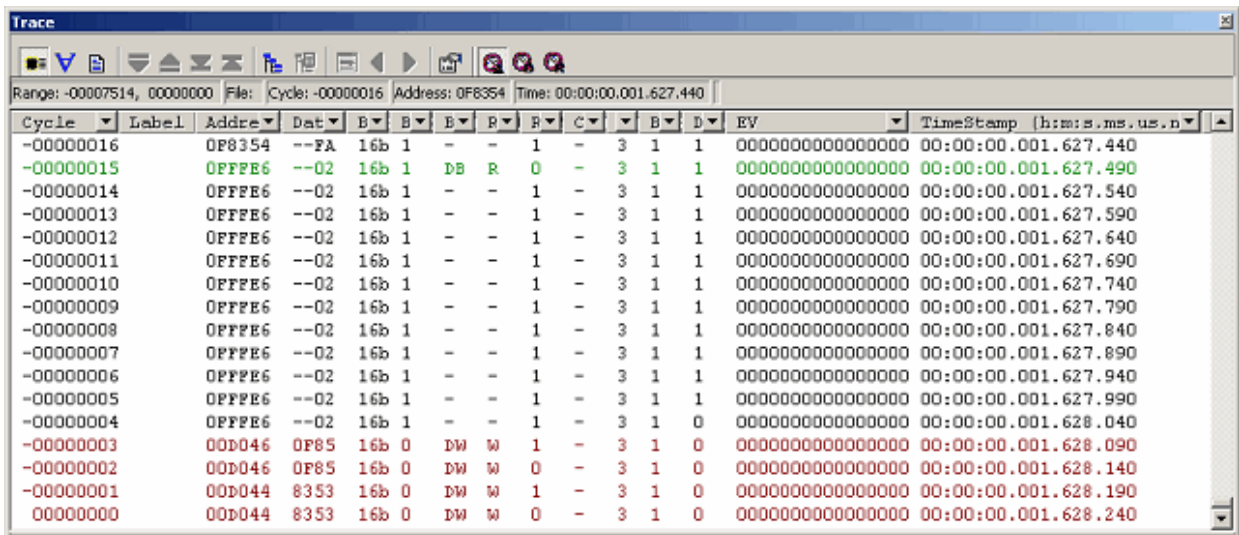
3.17.4 筛选工具

使用筛选工具可从获取的跟踪信息中仅提取必要的周期。

筛选工具通过筛选软件中由硬件获取的跟踪信息来完成此操作。


与获取跟踪信息前将获取条件设置为“捕捉 / 不捕捉条件”不同，此工具允许用户无数次更改获取的跟踪信息的筛选器设置，无需重新执行。因此，可以轻松地提取所需的信息。

- (1) 清除所有断点条件。在 [Trace]（跟踪）窗口中的任意位置单击鼠标右键，从显示的上下文菜单中选择 [Acquisition]（获取）。将显示 [Trace conditions]（跟踪条件）对话框。查看选中的跟踪模式是否为 [Fill until stop]（填充直至停止）。单击 [Close]（关闭）按钮。
- (2) 在 tutorial 函数中写入 p-sam s0=a[0]; 的行处设置一个软件断点。
- (3) 从 [Debug]（调试）菜单选择 [Reset Go]（复位执行）。断点将会中断处理，从开始到断点的跟踪信息将显示在 [Trace]（跟踪）窗口中。
- (4) 从 [Trace]（跟踪）窗口的上下文菜单中选择 [Auto Filter]（自动筛选）。筛选可应用于的列将用  按钮来标记。



Cycle	Label	Address	Data	B	B	B	P	R	C	B	D	EV	TimeStamp (h:m:s.ms.us.n)
-00000016	0F8354	--FA	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.440
-00000015	0FFFE6	--02	16b	1	DB	R	0	-	3	1	1	0000000000000000	00:00:00.001.627.490
-00000014	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.540
-00000013	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.590
-00000012	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.640
-00000011	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.690
-00000010	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.740
-00000009	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.790
-00000008	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.840
-00000007	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.890
-00000006	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.940
-00000005	0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.627.990
-00000004	0FFFE6	--02	16b	1	-	-	1	-	3	1	0	0000000000000000	00:00:00.001.628.040
-00000003	00D046	0F85	16b	0	DW	W	1	-	3	1	0	0000000000000000	00:00:00.001.628.090
-00000002	00D046	0F85	16b	0	DW	W	0	-	3	1	0	0000000000000000	00:00:00.001.628.140
-00000001	00D044	8353	16b	0	DW	W	1	-	3	1	0	0000000000000000	00:00:00.001.628.190
00000000	00D044	8353	16b	0	DW	W	0	-	3	1	0	0000000000000000	00:00:00.001.628.240

图 3.40 [Trace]（跟踪）窗口（[Auto Filter]（自动筛选））

(5) 在 [R/W] 列中单击  按钮并从上下文菜单中选择 [R]。

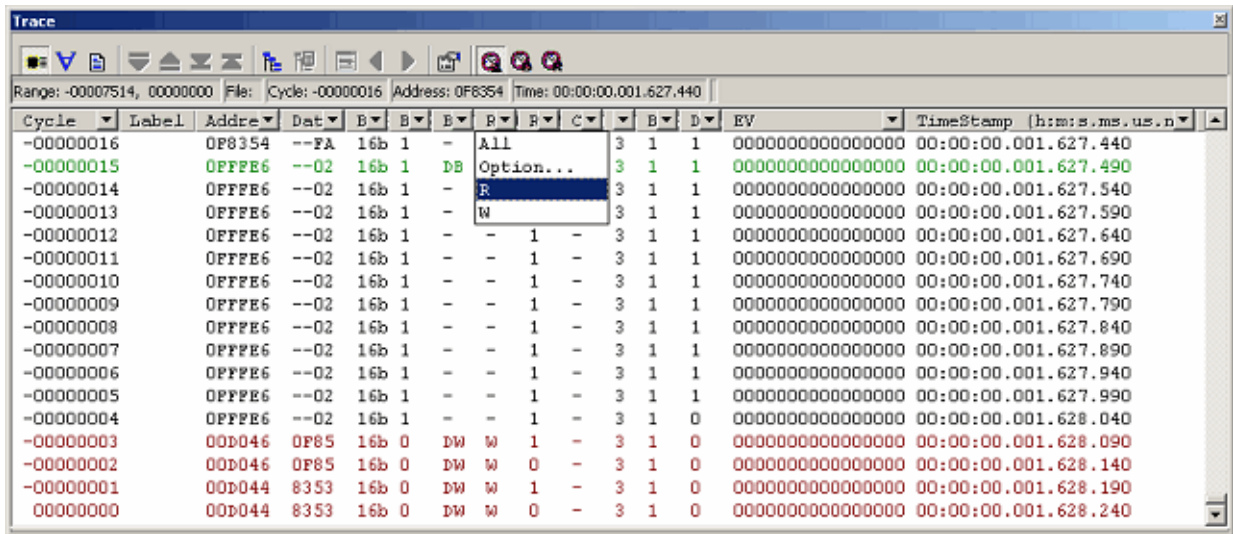


图 3.41 [Trace] (跟踪) 窗口 ([Auto Filter] (自动筛选))

(6) 这样，在 [R/W] 列就只显示 R 的跟踪信息。

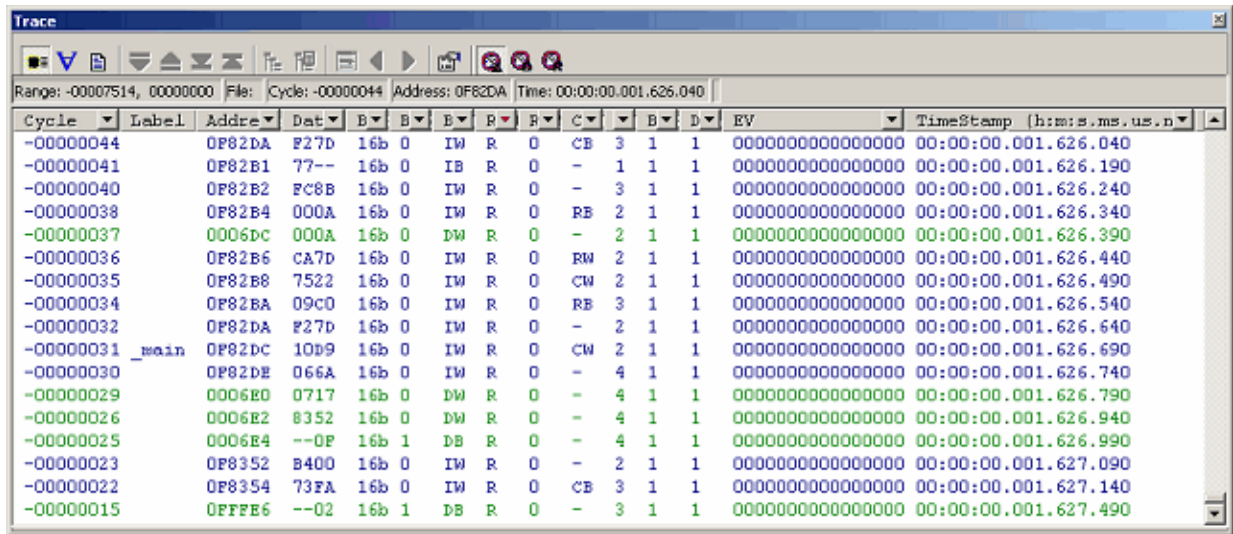


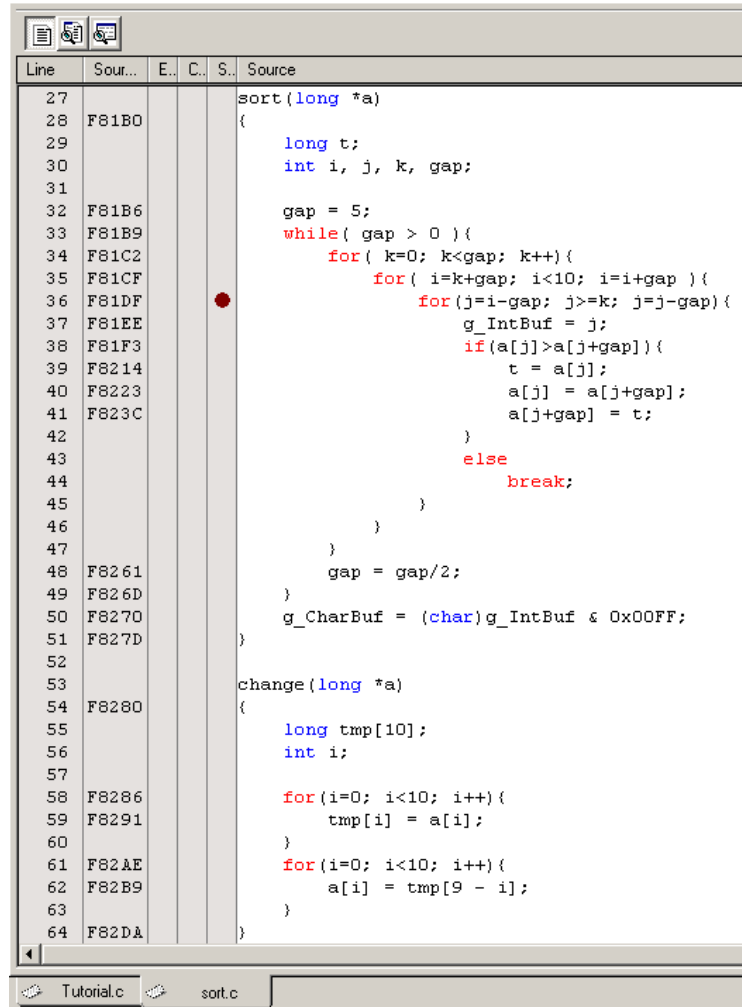
图 3.42 [Trace] (跟踪) 窗口 ([Auto Filter] (自动筛选))

- 注意：
1. 筛选器功能不会影响跟踪存储器，因此其内容保持不变。
 2. 选中的跟踪模式是 [Fill until stop] (填充直至停止)、[Fill until full] (填充至满) 或 [Fill around TP] (围绕 TP 填充) 时可以使用筛选器。

3.18 堆栈跟踪工具

使用堆栈信息，可以显示哪个函数是当前 PC 所在处的函数的调用方。

通过双击 sort 函数在 [S/W Breakpoints]（软件断点）列中对应的行，可以在 sort 函数任意行中设置软件断点。

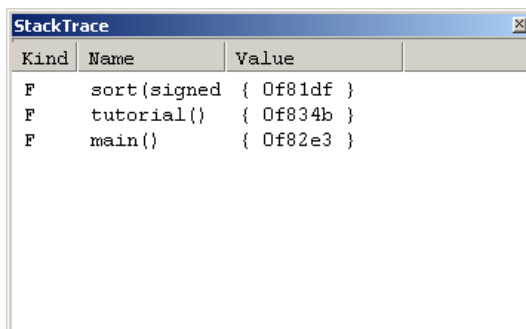


Line	Sour...	E...	C...	S...	Source
27					sort(long *a)
28	F81B0				{
29					long t;
30					int i, j, k, gap;
31					
32	F81B6				gap = 5;
33	F81B9				while(gap > 0){
34	F81C2				for(k=0; k<gap; k++){
35	F81CF				for(i=k+gap; i<10; i=i+gap){
36	F81DF				for(j=i-gap; j>=k; j=j-gap){
37	F81EE				g_IntBuf = j;
38	F81F3				if(a[j]>a[j+gap]){
39	F8214				t = a[j];
40	F8223				a[j] = a[j+gap];
41	F823C				a[j+gap] = t;
42					}
43					else
44					break;
45					}
46					}
47					}
48	F8261				gap = gap/2;
49	F826D				}
50	F8270				g_CharBuf = (char)g_IntBuf & 0x00FF;
51	F827D				}
52					
53					change(long *a)
54	F8280				{
55					long tmp[10];
56					int i;
57					
58	F8286				for(i=0; i<10; i++){
59	F8291				tmp[i] = a[i];
60					}
61	F82AE				for(i=0; i<10; i++){
62	F82B9				a[i] = tmp[9 - i];
63					}
64	F82DA				}

图 3.43 [Editor]（编辑器）窗口（设置软件断点）

从 [Debug]（调试）菜单选择 [Reset Go]（复位执行）。

中断后，从 [View]（视图）菜单选择 [Code]（代码）→ [Stack Trace]（堆栈跟踪）以打开 [Stack Trace]（堆栈跟踪）窗口。



Kind	Name	Value
F	sort(signed	{ 0f81df }
F	tutorial()	{ 0f834b }
F	main()	{ 0f82e3 }

图 3.44 [Stack Trace]（堆栈跟踪）窗口

用户将看到当前 PC 存在于 sort() 函数内部，并且 sort() 函数是从 tutorial() 函数调用的。

通过再次双击 sort 函数在 [S/W Breakpoints]（软件断点）列中对应的行，可以清除在 sort 函数的某行处设置的软件断点。

3.19 后续内容

在此教程中，向用户介绍了 E100 仿真器的一些功能以及如何使用 HEW。

使用 E100 仿真器提供的仿真工具，用户可以执行高级调试。一旦调试准确分离和确定了导致硬件或软件问题发生的条件，用户就可以有效地检查这些问题了。

第 4 章 准备调试

4.1 启动 HEW

按照下述步骤启动 HEW。

- (1) 将主机、E100 仿真器和用户系统连接起来。然后打开 E100 仿真器和用户系统的电源。
- (2) 从 [Start] (中文系统: [开始]) 菜单的 [Programs] (中文系统: [程序]) 中选择 [Renesas] (瑞萨) → [High-performance Embedded Workshop] (高性能嵌入式工作区) → [High-performance Embedded Workshop] (高性能嵌入式工作区)。

此时会出现如下所示的 [Welcome!] (欢迎!) 对话框。

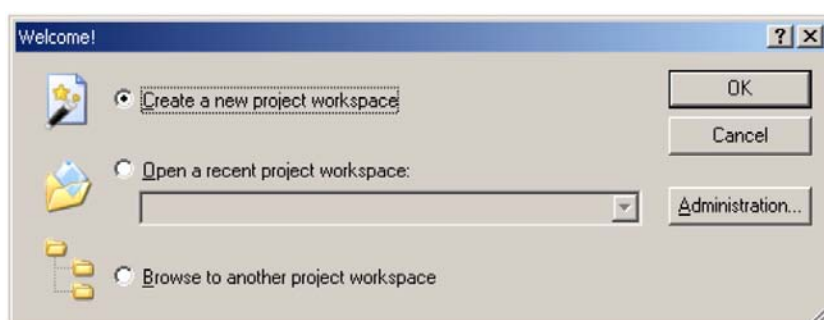


图 4.1 [Welcome!] (欢迎!) 对话框

从下面选择一个启动方法。

- [Create a new project workspace] (建立新的工程工作空间)
- [Open a recently used project workspace] (打开最近的工程工作空间)
如果使用现有的工作空间, 请选择此选项。
将显示所打开工作空间的历史记录。
- [Browse to another project workspace] (浏览到另一个工程工作空间)
如果使用现有的工作空间, 请选择此选项。
如果已打开的工作空间没有历史记录, 可选择此选项。

4.2 建立新的工作空间（不使用工具链）

建立新工程工作空间的步骤因是否使用工具链而异。

E100 仿真器中不包含工具链。在已安装 C/C++ 编译器套件的环境中可以使用工具链。

按照下述步骤建立新的工作空间。

- (1) 在 [Welcome!]（欢迎!）对话框中，选择标题为“Create a new project workspace”（建立新的工程工作空间）的单选按钮，然后单击 [OK]（确定）按钮。

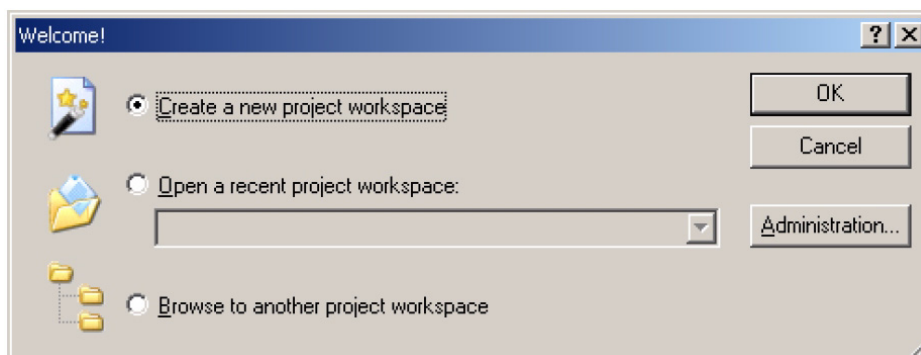


图 4.2 [Welcome!]（欢迎!）对话框

- (2) 此时将启动工程生成器 (Project Generator)。

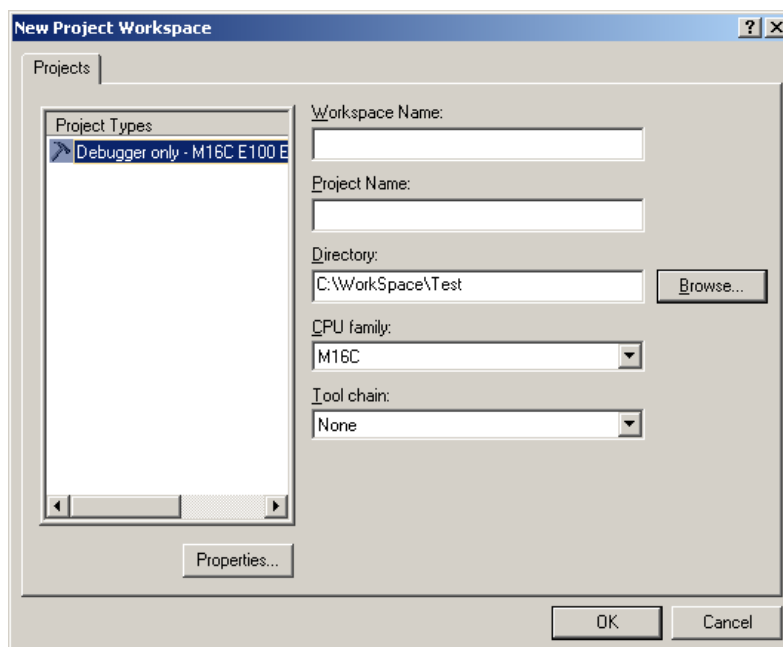


图 4.3 [New Project Workspace]（新建工程工作空间）对话框

Workspace Name: (工作空间名称 :)	在此处输入工作空间名称。
Project Name: (工程名称 :)	在此处输入工程名称。如果工程名称可以与工作空间名称相同, 则无需输入。
Directory: (目录 :)	输入要在其中建立工作空间的目录, 也可以单击 [Browse] (浏览) 按钮并从随后显示的列表选择工作空间目录。
CPU family: (CPU 族 :)	选择所使用 MCU 的 CPU 族。

其他列表框用于设置工具链。如果未安装工具链, 则该框中会显示特定于 CPU 族的信息。单击 [OK] (确定) 按钮。

(3) 选择调试器目标。

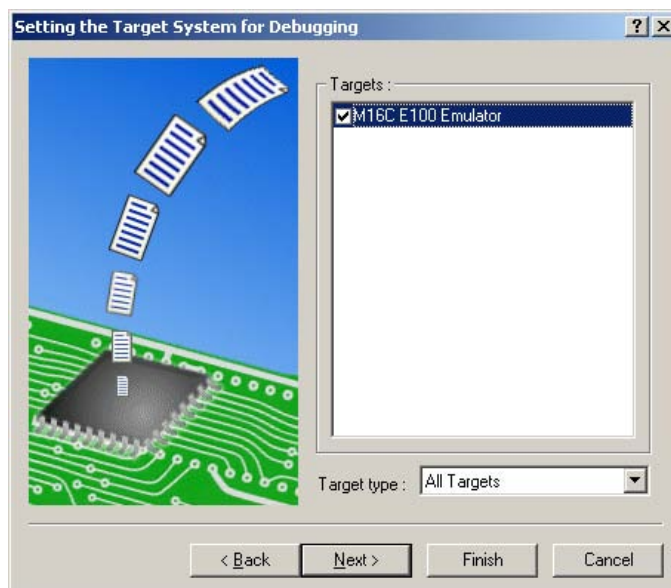


图 4.4 [Setting the Target System for Debugging] (设置调试的目标系统) 对话框

在使用的目标平台所对应的复选框中放置复选标记来选中该平台, 然后单击 [Next] (下一步) 按钮。

(4) 设置配置名称。配置是指保存 HEW 状态（而非保存仿真器）的文件。

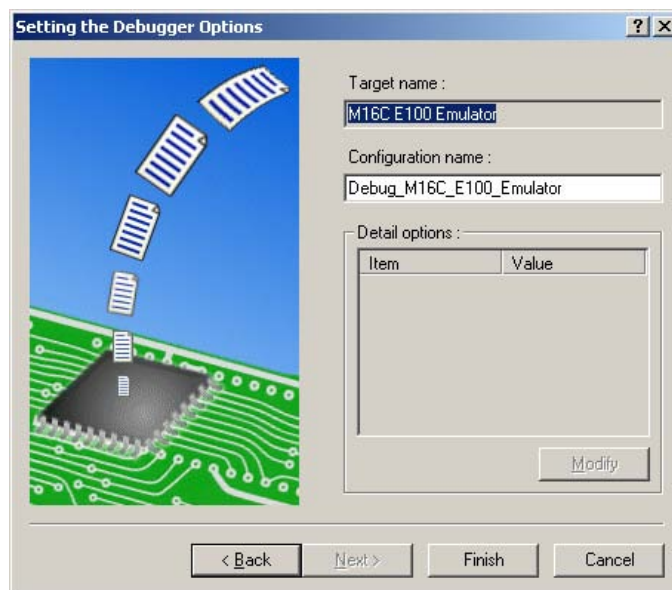


图 4.5 [Setting the Debugger Options]（设置调试程序选项）对话框

如果选择了两个或多个目标平台，请单击 [Next]（下一步）按钮，然后为每个选定的目标平台设置配置名称。

完成配置名称设置后，也就完成了与仿真器相关的设置。

单击 [Finish]（完成）按钮，将显示 [Summary]（汇总）对话框。单击其中的 [OK]（确定）按钮将启动 HEW。

(5) 启动 HEW 后，请连接 E100 仿真器。

4.3 建立新的工作空间（使用工具链）

按照下述步骤建立新的工作空间。

- (1) 在 [Welcome!]（欢迎!）对话框中，选择标题为“Create a new project workspace”（建立新的工程工作空间）的单选按钮，然后单击 [OK]（确定）按钮。

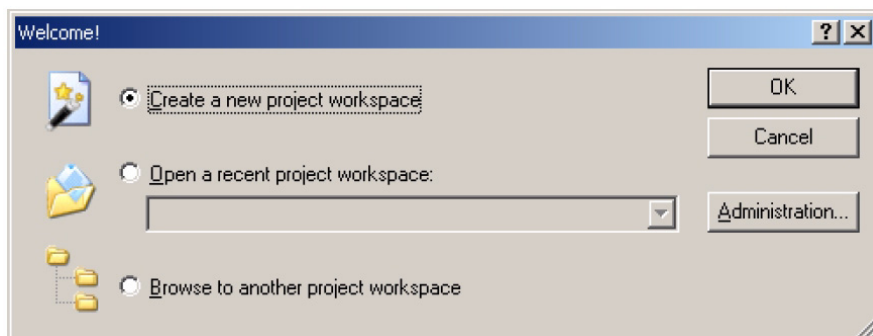


图 4.6 [Welcome!]（欢迎!）对话框

- (2) 此时将启动工程生成器 (Project Generator)。

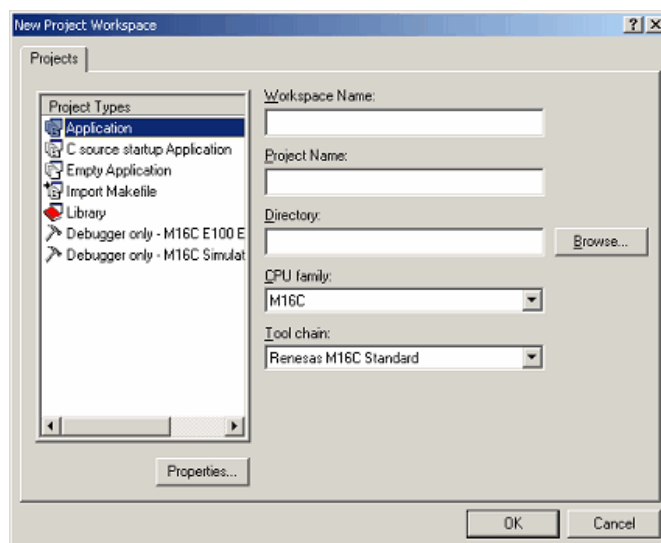


图 4.7 [New Project Workspace]（新建工程工作空间）对话框

Workspace Name: (工作空间名称:)	在此处输入工作空间名称。
Project Name: (工程名称:)	在此处输入工程名称。如果工程名称可以与工作空间名称相同，则无需输入。
Directory: (目录:)	输入要在其中建立工作空间的目录，也可以单击 [Browse]（浏览）按钮并从随后显示的列表选择工作空间目录。
CPU family: (CPU 族:)	选择所使用 MCU 的 CPU 族。
Toolchain: (工具链:)	要使用工具链，请在此处选择适当的工具链。如果不使用工具链，请选择 [None]（无）。

其他列表框用于设置工具链。如果未安装工具链，则该框中会显示特定于 CPU 族的信息。单击 [OK] (确定) 按钮。

(3) 设置 CPU 和用于工具链的选项，然后进行其他必要的设置。

(4) 选择调试器目标。

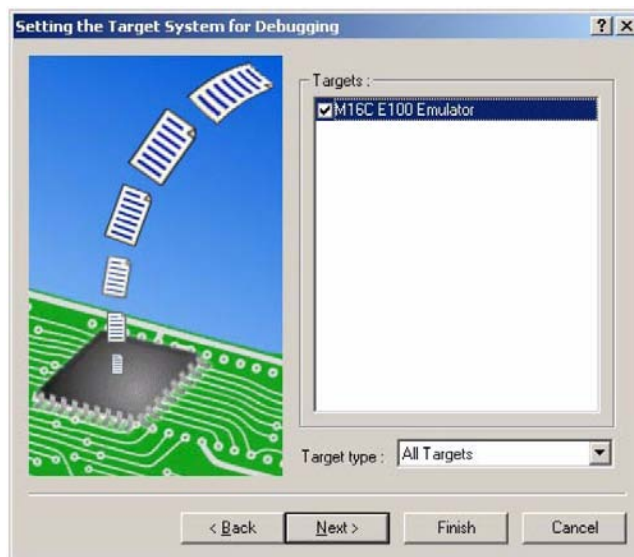


图 4.8 [Setting the Target System for Debugging] (设置调试的目标系统) 对话框

在使用的目标平台所对应的复选框中放置复选标记来选中该平台，然后单击 [Next] (下一步) 按钮。

(5) 设置配置名称。

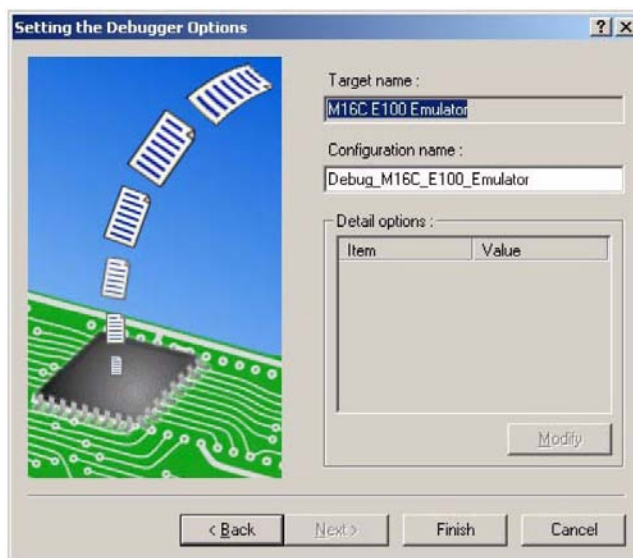


图 4.9 [Setting the Debugger Options] (设置调试程序选项) 对话框

如果选择了两个或多个目标平台，请单击 [Next] (下一步) 按钮，然后为每个选定的目标平台设置配置名称。完成配置名称设置后，也就完成了与仿真器相关的设置。单击 [Finish] (完成) 按钮，将显示 [Summary] (汇总) 对话框。单击其中的 [OK] (确定) 按钮将启动 HEW。

(6) 启动 HEW 后，请连接 E100 仿真器。

4.4 打开现有工作空间

请按照下述步骤打开现有工作空间。

- (1) 在 [Welcome!] (欢迎!) 对话框中, 选择标题为 “Browse to another project workspace” (浏览到另一个工程工作空间) 的单选按钮, 然后单击 [OK] (确定) 按钮。

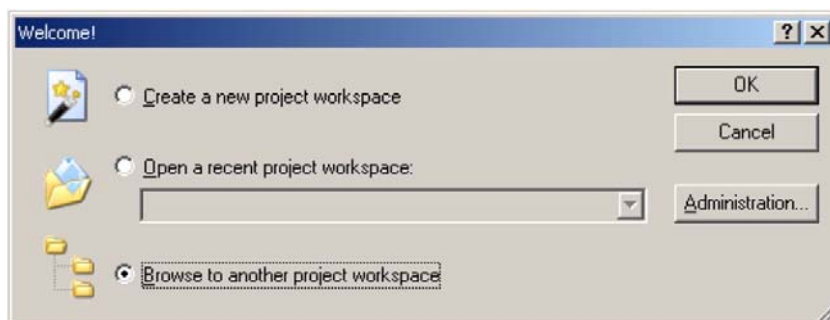


图 4.10 [Welcome!] (欢迎!) 对话框

- (2) 此时将出现如下所示的 [Open Workspace] (打开工作空间) 对话框。

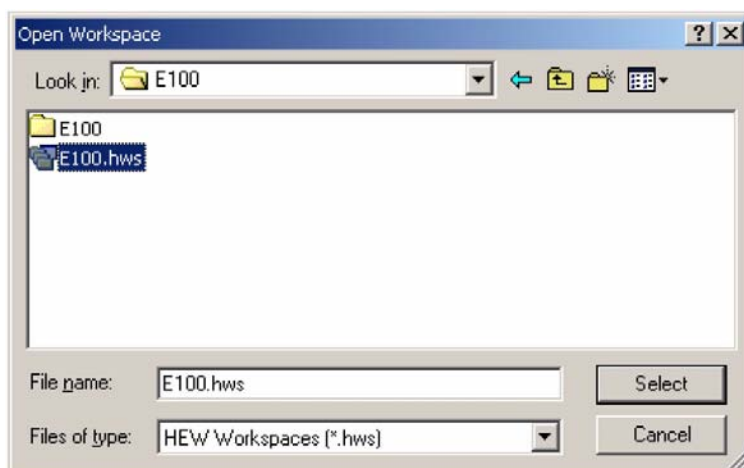


图 4.11 [Open Workspace] (打开工作空间) 对话框

指定在其中建立工作空间的目录, 选择工作空间文件 (扩展名为 “.hws”), 然后单击 [Select] (选择) 按钮。

- (3) 将启动 HEW, 同时还原选定工作空间保存时的状态。如果选定工作空间的保存状态是处于连接到仿真器的状态, 则工作空间会自动连接到仿真器。如果选定工作空间的保存状态是处于未连接到仿真器的状态, 且您要连接到仿真器, 请参阅 4.5 连接仿真器。

4.5 连接仿真器

4.5.1 连接仿真器

连接仿真器可以使用以下几种方法。

(1) 连接前启动时设置仿真器

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。在此对话框中，可以注册下载模块和要自动执行的命令链。完成填写 [Debug Settings] (调试设置) 对话框后，将连接仿真器。


(2) 加载会话文件

切换到具有已注册仿真器使用设置的会话文件，有助于轻松连接仿真器。

4.5.2 重新连接仿真器

如果仿真器断开了连接，可以按照如下所述步骤之一重新连接仿真器。

(1) 从 [Debug] (调试) 菜单选择 [Connect] (连接)。

(2) 单击 [Connect] (连接) 工具栏按钮 []。

(3) 在 [Command Line] (命令行) 窗口中输入连接命令。

4.6 断开仿真器连接

4.6.1 断开仿真器连接

要断开处于活动状态的仿真器连接，可以按照如下所述步骤之一断开仿真器连接。

(1) 从 [Debug] (调试) 菜单选择 [Disconnect] (断开连接)。

(2) 单击 [Disconnect] (断开连接) 工具按钮 []。

(3) 在 [Command Line] (命令行) 窗口中输入断开连接命令。

4.7 退出 HEW

从 [File] (文件) 菜单中选择 [Exit] (退出)，可以关闭 HEW 本身。

关闭 HEW 之前，会显示一个消息框，询问是否要保存会话。要保存会话，请单击 [Yes] (是) 按钮。

4.8 设置调试

注册下载模块，设置命令行批文件的自动执行以及设置下载选项等。

4.8.1 指定下载模块

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。

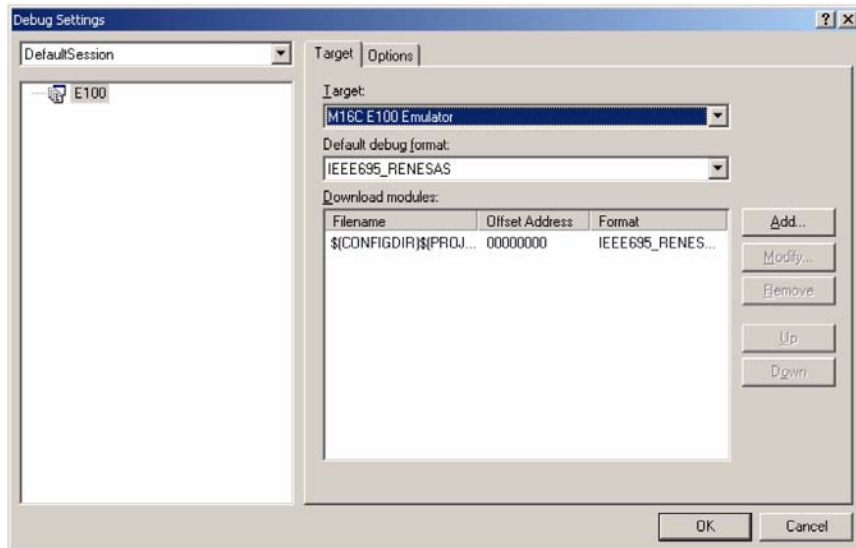


图 4.12 [Debug Settings]（调试设置）对话框

在 [Target]（目标）下拉式列表框中选择要连接的产品名称。

在 [Default Debug Format]（默认调试格式）下拉式列表框中选择要下载的加载模块的格式。然后在 [Download Modules]（下载模块）列表框中注册与选定格式对应的加载模块。

注意： 此时，尚未下载任何程序。有关如何下载的详细信息，请参阅 5.2 下载程序。

4.8.2 设置命令行批文件的自动执行

单击对话框的 [Options]（选项）选项卡。

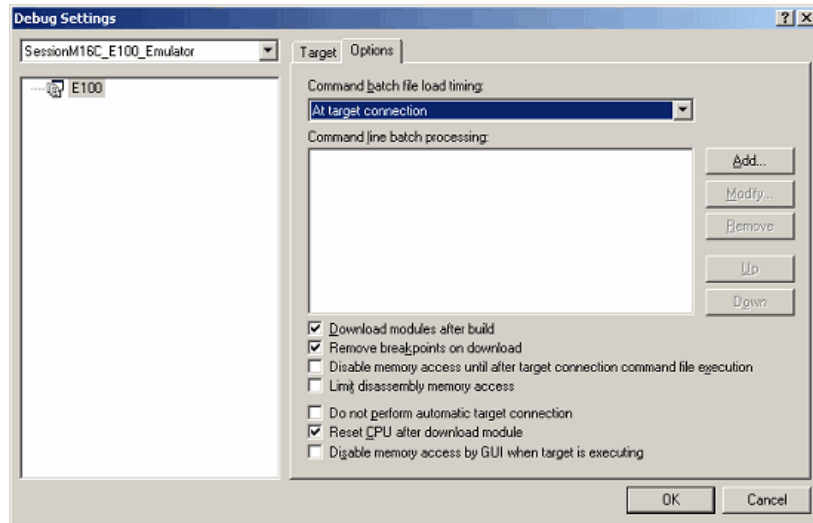


图 4.13 [Debug Settings]（调试设置）对话框

在这里注册在指定时间自动执行的命令链。

从以下四个选项中选择所需的时间：

- [When the emulator is connected]（在连接仿真器时）
- [Immediately before download]（紧接下载之前）
- [Immediately after download]（紧接下载之后）
- [Immediately after reset]（紧接复位之后）

在 [Command batch file load timing]（命令批文件加载时间）下拉式列表框中选择执行命令链的时间。

第 5 章 调试功能

E100 仿真器支持下表中所列的功能。

表 5.1 调试功能列表 (1)

项目编号	项目		说明
1	软件断点		4,096 点
2	事件	事件点的数量	有效点的最大数量: 16
		事件的内容	已执行的地址检测
			数据存取信息
			中断生成 / 退出检测
			外部触发器检测
		任务 ID	可针对每个事件单独设置
事件发生的次数	最多 255 次		
3	异常检测		存取保护的违规操作
			从未初始化的存储器读取
			堆栈存取的违规操作
			性能上溢
			实时分析上溢
			跟踪存储器上溢
			任务堆栈存取的违规操作
			操作系统调度
4	硬件断点	硬件断点	事件组合 OR、AND (累加)、AND (同时)、子例程、顺序和状态转换
		异常检测	
	延迟	最多 65,535 个总线周期	

表 5.1 调试功能列表 (2)

项目编号	项目		说明		
5	跟踪	跟踪大小		最多 4M 个周期	
		跟踪模式	Fill until stop (填充直至停止)		继续收集, 直到程序停止运行
			Fill until full (填充至满)		当跟踪存储器充满时, 停止收集跟踪
			Fill around TP (围绕 TP 填充)		从到达跟踪点时开始, 在延迟几个延迟周期后停止收集跟踪
			Repeat fill until stop (重复填充直至停止)		在跟踪点前后收集总共 512 个周期
			Repeat fill until full (重复填充至满)		在跟踪点前后收集总共 512 个周期
		跟踪点	事件组合		OR、AND (累加)、AND (同时)、子例程、顺序和状态转换
			异常检测		请参见编号 3 的项目
		延迟		最多 4M 个总线周期	
		跟踪捕捉 / 不捕捉		按事件捕捉 / 不捕捉 - 两个事件之间 - 某一事件过程中 - 子例程中发生的某一事件过程中	
		数据存取指令提取			
6	性能	测量内容		在最多 8 个段和通过计数中, 测量最长、最短和平均的执行时间	
				超时和计数上溢检测	
		解析度		10 ns 到 1.6 us	
		测量模式	事件组合	两个事件之间、事件期间以及两个事件之间禁止中断的范围	
7	RAM 监控		512 字节 x 32 块 - 显示上次执行的读 / 写存取 - 带有省略初始化的检测功能		
8	分析		128 KB x 8 块 (1 MB 空间)		
			累计时间和通过计数上溢检测		

表 5.1 调试功能列表 (3)

项目编号	项目	说明
9	覆盖	C0 级别代码覆盖 256 KB x 8 块 (2 MB 空间)
		C0+C1 级别代码覆盖 128 KB x 8 块 (1 MB 空间)
		地址范围和源文件指定
		数据覆盖 64 KB x 8 块 (512 KB 空间) 地址范围、段指定以及任务堆栈

5.1 设置仿真环境

连接仿真器时，将显示 [Device setting]（器件设置）和 [Configuration properties]（配置属性）对话框。在对话框中选择与仿真器相关的常规选项。请注意，只能在启动时对要调试的目标 MCU 等选项设置一次。

5.1.1 启动时设置仿真器

仿真器启动时，会显示以下三个对话框。

(1) [Device setting]（器件设置）对话框

使用此对话框可选择目标 MCU 和建立通信。

启动仿真器后，可通过从 [Setup]（设置）菜单选择 [Emulator]（仿真器）[Device setting]（器件设置）来重新显示此对话框。但在这种情况下，要注意在启动仿真器后进行的设置更改不会立即反映出来，而要在重新连接仿真器时才会设置为初始值。

(2) [Configuration properties]（配置属性）对话框

此对话框在 [Device setting]（器件设置）对话框之后显示。使用此对话框可以进行与仿真器和调试功能相关的设置。

启动仿真器后，只能通过从 [Setup]（设置）菜单选择 [Emulator]（仿真器）[System]（系统）来重新显示此对话框。启动后，可以更改此对话框中一些选项的设置。可更改的选项显示为活动状态，而不可更改的选项则为不可用状态（显示为灰色），只显示其中的已设置内容。

(3) [Connecting]（连接）对话框

此对话框显示引导处理的进度。

5.1.2 设置目标 MCU

(1) 选择目标 MCU

在 [Device setting]（器件设置）对话框的 [Device]（器件）页上，指定要仿真的目标 MCU。有关详细信息，请参阅每种产品附带提供的硬件手册。

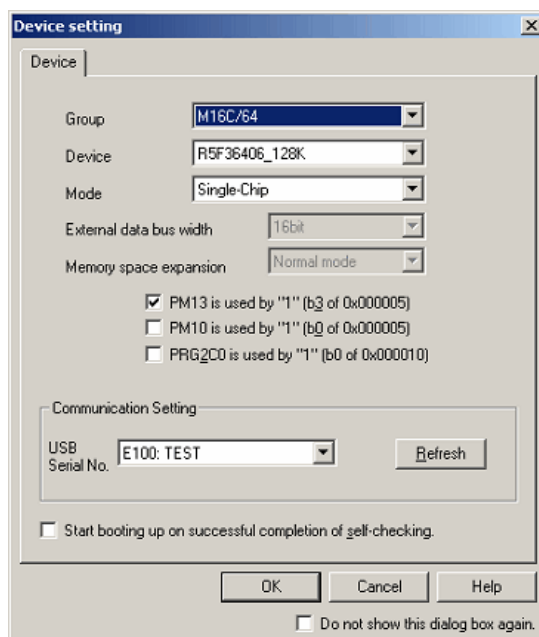


图 5.1 [Device setting]（器件设置）对话框（[Device]（器件）页）

连接仿真器后，将不能更改在此页中设置的目标 MCU。要更改目标 MCU，需要断开仿真器连接，然后重新连接。

(2) 选择操作模式

请选择以下选项之一：

[Single-Chip]（单片）模式、[Memory expansion]（存储器扩展）模式和 [Microprocessor]（微处理器）模式

注意：选项因用户选择的目标 MCU 而异。

(3) 选择外部数据总线宽度

当用户所选的运行模式为“Memory expansion”（存储器扩展）模式或“Microprocessor”（微处理器）模式时，可以设置此选项。请选择以下选项之一：

[8 bits]（8 位）和 [16 bits]（16 位）（初始值）

(4) 选择存储器扩展空间

如果用户所选的运行模式是“Memory expansion”（存储器扩展）模式或“Microprocessor”（微处理器）模式，就可以设置此选项。

请选择以下选项之一：

[Normal Mode]（普通模式）（初始值）和 [4MB Mode]（4MB 模式）

(5) PM13（0x000005 的 b3）设置为 1

要更改 CS2 区域的设置，请指定 PM13（处理器模式寄存器 1 的第 3 位）设置。在 PM13 设置为 1 的情况下使用用户程序时，请选中此复选框。

(6) PM10（0x000005 的 b0）设置为 1

要扩展内部保留区，请指定 PM10（处理器模式寄存器 1 的零位）设置。在 PM10 设置为 1 的情况下使用用户程序时，请选中此复选框。

(7) PRG2C0（0x000010 的 b0）和 PM10（0x000005 的 b0）设置为 1

要允许或禁止程序 ROM 2 区域，请指定 PRG2C0（程序 2 区域控制寄存器的零位）设置。将 PRG2C0 设置为 1 的情况下使用用户程序时，请选中此复选框。

(8) 设置通信

选择系统要通过 USB 连接到的另一方。USB 序列号列表框显示连接 USB 的仿真器的单元标识信息的列表。单击 [Refresh]（刷新）按钮可以刷新单元标识信息。

(9) 执行自我诊断

通过选中 [Start booting up on successful completion of self checking]（在成功完成自检后开始引导）复选框并单击 [OK]（确定）按钮，可在已设置的通信条件下，连接用户系统和仿真器来执行硬件自我诊断。在诊断完成时会显示结果。如果诊断正常终止，系统会继续执行引导处理。如果诊断因错误终止，系统会中断引导处理。

5.1.3 设置系统

在 [Configuration Properties]（配置属性）对话框的 [System]（系统）页上，设置整个仿真器系统。启动时，此对话框在 [Device setting]（器件设置）对话框之后显示。尽管启动后可重新显示此对话框，但将无法更改其中一些设置。这些设置只能在启动时更改。

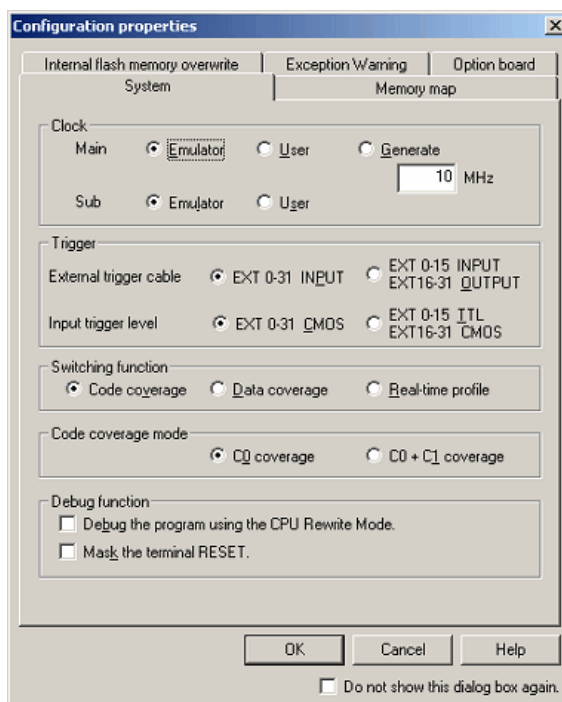


图 5.2 [Configuration properties]（配置属性）对话框（[System]（系统）页）

(1) 选择运行时钟

在 [System]（系统）页的 [Clock]（时钟）部分，选择为主时钟和副时钟提供的时钟源。

可以从以下三个选项中选择主时钟：[Emulator]（仿真器）、[User]（用户）和 [Generate]（生成）。（默认情况下选择 [Emulator]（仿真器））。

如果主时钟是从内部源提供的，请选择 [Emulator]（仿真器）；如果主时钟是从外部源提供的，请选择 [User]（用户）。要使用用户定义的时钟，请选择 [Generate]（生成），并在频率输入文本框中设置时钟频率。

可以在 1.0 到 99.9 MHz 范围内设置时钟频率，最小增减幅度为 0.1 MHz。[Generate]（生成）的时钟频率只能在启动时设置一次。

只有支持副时钟时才会显示副时钟的选项。可从 [Emulator]（仿真器）或 [User]（用户）选择副时钟。（默认情况下选择 [Emulator]（仿真器））。

注意： [Generate]（生成）的频率精确度为 $\pm 5\%$ 。请确保执行最终评估时，所用谐振器或振荡器的频率与板上安装实际目标板的频率一致。

(2) 选择外部触发器电缆的方向

对于外部触发器电缆，请选择 EXT 引脚 16~31 指向输入还是输出。EXT 引脚 0~15 固定用于输入。
请从以下选项中选择此选项：

- EXT 0~31 INPUT（初始值）
- EXT 0~15 INPUT，EXT 16~31 指向 OUTPUT

此选项的设置只在启动时才会反映出来。如果启动后在对话框中设置此选项，设置的内容不会生效。

(3) 选择触发器输入电平

对于触发器输入电平，请选择 CMOS 电平或 TTL 电平。请从以下选项中选择此选项：

- EXT 0~31 选择为 CMOS（初始值）
- EXT 0~15 选择为 TTL，EXT 16~31 选择为 CMOS

(4) 选择切换功能

代码覆盖、数据覆盖和实时分析功能不能同时使用。请从它们之中选择一项功能。

初始状态下选择代码覆盖。

此选项的设置启动后也可以更改。

如果选择了代码覆盖功能，则在 [Code coverage mode]（代码覆盖模式）下选定的覆盖级别下执行测量。

(5) 选择代码覆盖模式。

选择代码覆盖模式。

C0：指令覆盖率

C0 + C1：指令覆盖率 + 分支覆盖率

在使用 C0 级别覆盖时，最多可以测量 2 MB。在使用 C0 + C1 级别覆盖时，最多可以测量 1 MB。

初始值为 C0 覆盖。

此选项的设置只在启动时才会反映出来。

只有在 [Switching function]（切换功能）中选择了 [Code coverage]（代码覆盖）时，此选项才可用。

如果使用代码覆盖功能，请在启动时在此选项下选择模式。

(6) 启用 [Debug the program using the CPU Rewrite Mode]（使用 CPU 重写模式调试程序）

选择是否要使用 CPU 重写模式来调试程序。

(7) 屏蔽目标系统的 RESET 引脚输入

选择是否屏蔽目标系统中 RESET 引脚的输入信号。

5.1.4 创建存储器映像

在 [Configuration Properties]（配置属性）对话框的 [Memory map]（存储器映像）页上，设置借用存储器分配。可以指定 4 个区域。（以 4KB 为单位）

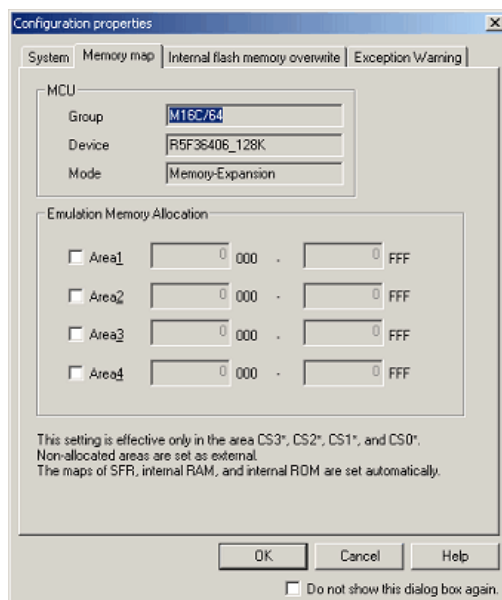


图 5.3 [Configuration properties]（配置属性）对话框（[Memory map]（存储器映射）页）

MCU 分组框显示在 [Device setting]（器件设置）对话框中选择的器件。在此页上不能进行更改。如果选择了单片模式，不会显示 [Memory map]（存储器映射）页。

(1) 分配仿真存储器

最多可以为 4 个区域分配仿真存储器。

选中要使用的区域的复选框，并输入起始地址和结束地址。

地址可以 4KB 为单位设置。因此，起始地址和结束地址的低 12 位是固定的。

5.1.5 设置闪存 ROM 盖写

在 [Configuration properties]（配置属性）对话框的 [Internal flash memory overwrite]（内部闪存盖写）页上，逐块设置闪存 ROM 块的盖写。

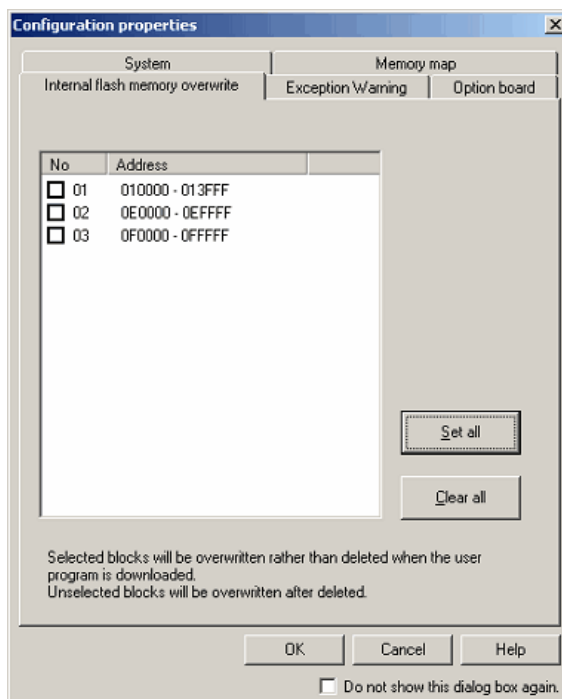


图 5.4 [Configuration properties]（配置属性）对话框（[Internal flash memory overwrite]（内部闪存盖写）页）

与所选目标 MCU 相匹配的逐块设置自动显示在列表中。

通过在相应复选框中放置复选标记而选择的块将在下载程序时被盖写（合并），而不会被擦除。

5.1.6 设置异常事件的警告

在 [Configuration properties]（配置属性）对话框的 [Exception Warning]（异常警告）页上，设置是否在 [Status]（状态）窗口和状态栏提示框中显示异常事件的警告。

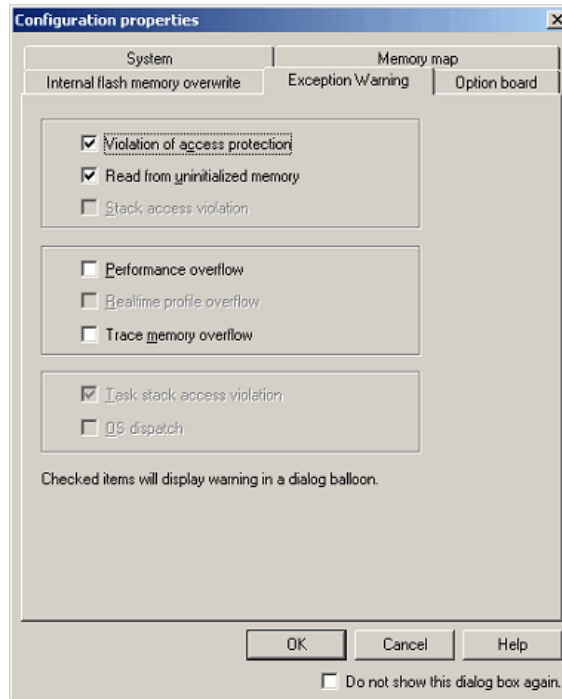


图 5.5 [Configuration properties]（配置属性）对话框（[Exception Warning]（异常警告）页）

[Violation of access protection]（存取保护的违规操作）和 [Read from uninitialized memory]（从未初始化的存储器读取）的初始设置有效。

在下载包括操作系统的加载模块时，[Task stack access violation]（任务堆栈存取的违规操作）的初始设置也有效。

未选择其他项目。

如果取消选择该复选框，则 [Status]（状态）窗口中的项目将显示为“-”。

5.1.7 设置可选板

在 [Configuration Properties]（配置属性）对话框的 [Option board]（可选板）页上，选择可选板。

5.1.8 显示引导处理的进度

通过检查 [Connecting]（连接）对话框，可以确认引导处理的进度。

引导处理开始时，[Connecting]（连接）对话框会不断显示进度信息，直到引导处理完成。

在显示 [Device setting]（器件设置）和 [Configuration Properties]（配置属性）对话框时，不能对此对话框执行操作。

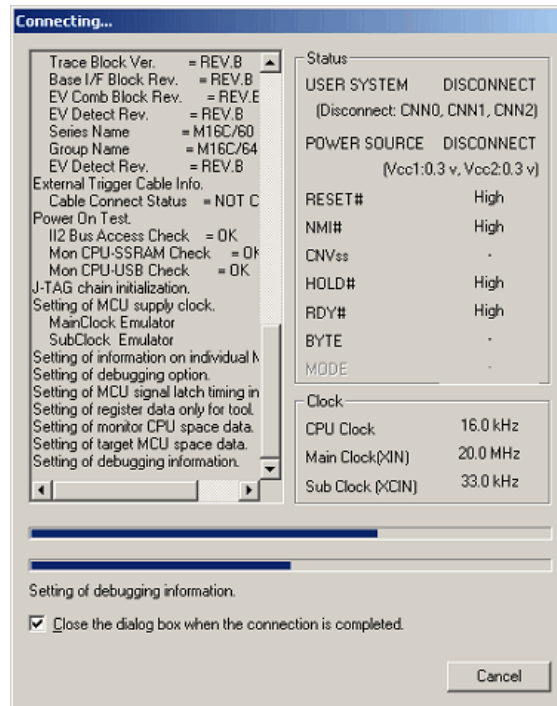


图 5.6 [Connecting]（连接）对话框

(1) 显示处理的历史记录

对话框左侧的历史记录显示区域显示已完成处理的历史记录。

此处显示的内容记录在故障报告中。要检查故障报告的内容，请从 [Help]（帮助）菜单中选择 [Technical Support]（技术支持）[Create Bug Report]（建立程序错误报告）。

(2) 显示引脚状态

在仿真器的启动过程完成后，会显示引脚状态。

如果在 [Device setting]（器件设置）对话框中设置的内容与此处显示的引脚状态不匹配，则会在历史记录显示区中显示警告信息。

(3) 显示时钟

在仿真器的启动过程完成后，会显示时钟。

此处只显示实际运行的时钟。

- (4) 通过进度条显示进度
上方的进度条显示整个引导处理的进度。
下方的进度条显示每个单独处理的进度。
当前执行的处理的内容显示在进度条下方。
- (5) 中断连接
单击 [Cancel]（取消）按钮将中断引导处理。

5.2 下载程序

5.2.1 下载程序

下载要调试的加载模块。

要下载程序，请从 [Debug]（调试）菜单中选择 [Download]（下载），并从随后显示的列表中选择所需的加载模块，或者在 [Workspace]（工作空间）窗口的 [Download modules]（下载模块）中右键单击一个加载模块，然后从弹出式菜单或上下文菜单中选择 [Download]（下载）。

注意：必须在高性能嵌入式工作区（High-performance Embedded Workshop，简称 HEW）中将程序注册为加载模块，才能下载程序。有关如何注册的详细信息，请参阅 4.8 设置调试。

5.2.2 显示源代码

按照下述步骤显示源代码。

- 在 [Workspace]（工作空间）窗口中双击源文件。
- 右键单击源文件，并从上下文菜单中选择 [Open]（打开）。

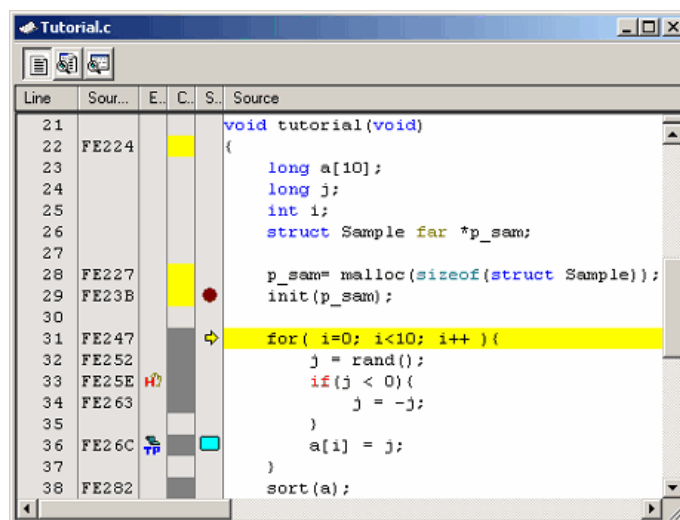




图 5.7 [Editor]（编辑器）窗口

此窗口的左侧显示行信息，包括以下内容：

- (1) [Line]（行）列
显示与源文件中的行相对应的行编号。
- (2) [Source Address]（源地址）列
程序下载后，此列显示与当前源文件中的行相对应的地址。在用户确定要在何处设置 PC 值或断点时，此功能将是非常便利的。
- (3) [Event]（事件）列
此列显示以下内容：

表 5.2 [Event]（事件）列列表

	已设置硬件断点
	已设置跟踪点（取条件）




通过双击事件列可以插入硬件断点。

如果设置了取条件，则会显示取条件。

[Hardware break]（硬件断点）、[Trace conditions]（跟踪条件）或 [Performance Analysis Conditions]（性能分析条件）对话框标题栏上的标题后的 [*] 表示正在编辑某些设置。如果正在进行编辑工作，则不能通过 [Editor]（编辑器）窗口的 [event]（事件）列更改设置。

- (4) [Code Coverage]（代码覆盖）列
以图形方式显示 CO 代码覆盖信息。
- (5) [S/W Breakpoints]（软件断点）列
此列显示以下内容：

表 5.3 [Software Breakpoint]（软件断点）列列表

	已设置书签
	已设置软件断点
	PC 位置

5.2.3 关闭所有源文件中的列

1. 从 [Editor]（编辑器）窗口
 - A. 右键单击 [Editor]（编辑器）窗口，并从上下文菜单选择 [Define Column Format]（定义列格式）。
 - B. 将显示 [Global Editor Column States]（全局编辑器列状态）对话框。

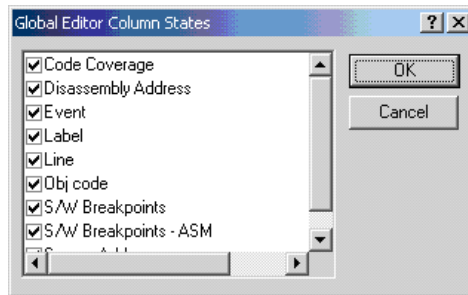


图 5.8 [Global Editor Column States]（全局编辑器列状态）对话框

- C. 对于要关闭的列，请取消选中其复选框。单击 [OK]（确定）按钮，已设置的新列设置将生效。

5.2.4 关闭一个源文件中的列

1. 从 [Editor]（编辑器）窗口
 - A. 右键单击 [Editor]（编辑器）窗口，并从上下文菜单选择 [Columns]（列）。
 - B. 此时将显示层叠的菜单项。当前已启用列的相应名称左侧带有一个复选标记。

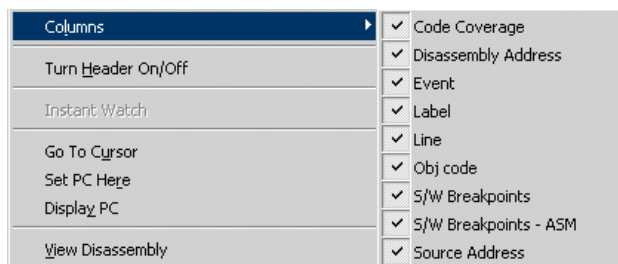


图 5.9 弹出菜单窗口


- C. 单击列名可以交替允许或禁止列。

5.2.5 显示汇编语言代码

在源文件打开时，鼠标右键单击 [Editor]（编辑器）窗口，然后从上下文菜单中选择 [View Disassembly]（查看反汇编）。此时将显示 [Disassembly]（反汇编）窗口。

[Disassembly]（反汇编）窗口中显示的起始地址是与 [Editor]（编辑器）窗口中的光标位置相对应的那个地址。

还可以使用 [Editor]（编辑器）窗口中的 [Disassembly View]（反汇编视图）按钮来显示反汇编代码。如果源文件不存在，可使用以下方法之一显示反汇编代码。

- 单击 [Disassembly]（反汇编）工具栏按钮 。
- 从 [View]（视图）菜单中选择 [Disassembly]（反汇编）。
- 使用“Ctrl + D”快捷键。

这种情况下，将在当前 PC 位置打开 [Disassembly]（反汇编）窗口。

还可支持将“Mixed”（复合）模式显示作为一个选项，在这种模式下，将显示以该地址开始的所有源代码行。要在复合模式中显示反汇编代码，请单击 [Show in Mixed Mode]（在复合模式下显示）按钮。

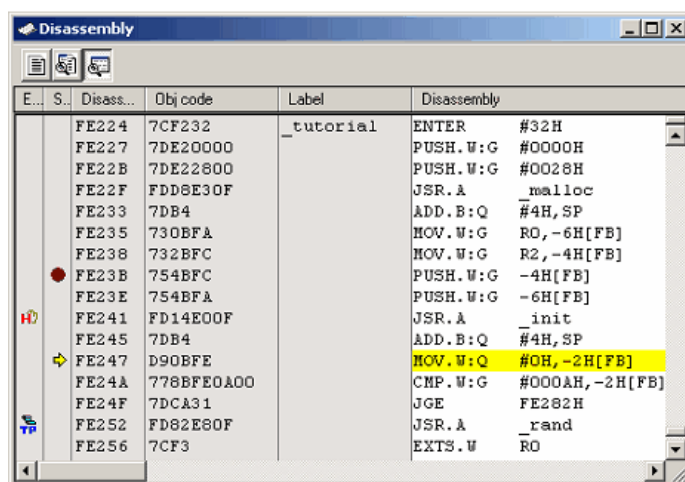




图 5.10 [Disassembly]（反汇编）窗口

此窗口的左侧显示行信息，包括以下内容：

- (1) [Event]（事件）列
此列显示以下内容：

表 5.4 [Event]（事件）列列表

	已设置硬件断点。
	已设置跟踪点（取条件）。


通过双击事件列可以插入硬件断点。

如果设置了取条件，则会显示取条件。

(2) [S/W Breakpoints - ASM] (软件断点 - ASM) 列

此列显示以下内容：

表 5.5 [Software Breakpoint - ASM] (软件断点 - ASM) 列列表

	已设置软件断点。
	PC 位置

(3) [Disassembly Address] (反汇编地址) 列

显示反汇编地址。双击此处将弹出 [Address Specification] (指定地址) 对话框。在此对话框中，输入开始反汇编显示的起始地址。

(4) [Obj code] (对象代码) 列

显示对象代码。

(5) [Label] (标签)

显示标签。若未下载模块，则此列不可用。

5.2.6 更正汇编语言代码

在 [Disassembly]（反汇编）窗口中双击要更正的指令，或者从上下文菜单中选择 [Edit]（编辑），将显示一个标记为“Assembler”（汇编）的对话框。使用此对话框更正汇编语言。



图 5.11 [Assembler]（汇编器）对话框

该对话框显示所选指令的地址、指令码和助记符。

在 [Mnemonic]（助记符）编辑框中输入新指令（或编辑旧指令）。完成后，按下 Enter 键。存储器内容将被新指令码盖写，并且指针移至下一条指令。

单击 [OK]（确定）按钮可使用新指令码盖写存储器内容，然后关闭对话框。

注意： 汇编语言代码从当前存储器内容显示。当用户更正存储器内容时，新的汇编语言代码将在 [Disassembly]（反汇编）窗口和 [Assembler]（汇编器）对话框中显示。但是，在 [Editor]（编辑器）窗口中显示的源文件保持不变。当源文件包括汇编器语言时也同样如此。

5.3 实时显示存储器内容

5.3.1 实时显示存储器内容

要在用户程序运行时监控存储器内容，请使用 [RAM monitor]（RAM 监控）窗口。

使用 RAM 监控功能，可以实时记录和检查所分配存储区域中的存储器内容和存取状态，而不会妨碍用户程序的运行。

[RAM monitor]（RAM 监控）窗口以不同颜色显示存取状态（读、写、未初始化或未检查）。

(1) 分配 RAM 监控区域

提供 16 KB 的 RAM 监控区域。

此 RAM 监控区域可以分配到选定的相邻地址或分开的 32 块（每块 512 字节）。

如果使用初始设置，则最多可将从内部 RAM 的起始地址开始的 16 KB 区域分配为 RAM 监控区域。

(2) 监控显示

存取状态以不同背景颜色显示，取决于存取属性，如下所列。（背景颜色可以自定义。）

读写存取显示最近进行的存取。

可通过从上下文菜单中选择 [Show Error Detection]（显示错误检测）来显示错误检测。在此情况下，不显示读写存取。

表 5.6 存取属性和背景颜色

	存取属性		背景颜色
1	读存取		绿色
2	写存取		红色
3	检测到错误时	未初始化存储器（对尚未进行写存取的区域进行了读存取）	黄色
4		未检查存储器（已进行写存取的区域尚未进行读存取）	天蓝色
5	无存取		白色

注意：[RAM Monitor]（RAM 监控）窗口中显示的内容是从总线存取获取的数据。因此，当直接从外部 I/O 盖写存储器时，通过用户程序以外的途径存取存储器进行的更改不会反映在显示的存储器内容中。

(3) 忽略初始化的检测功能

如果对尚未进行写存取的区域进行读存取，则此功能会假定此种情况是“忽略初始化的”并输出错误。

要显示忽略初始化的检测，请从上下文菜单中选择 [Show Error Detection]（显示错误检测）。

未初始化存储器以黄色显示。

“忽略初始化”检测可用作一个异常事件，由硬件断点或跟踪点的条件组成。（请参阅检测异常事件。）

(4) 未检查存储器检测功能

如果进行过写存取的区域从未进行一次读存取，则此功能会假定此种情况是“未检查的”并输出错误。

要显示未检查存储器检测，请从上下文菜单中选择 [Show Error Detection]（显示错误检测）。

未检查存储器以天蓝色显示。

5.3.2 设置 RAM 监控更新间隔

从 [RAM Monitor] (RAM 监控) 窗口的上下文菜单中选择 [Update Interval Setting] (更新间隔设置)。此时将显示如下所示的 [Update Interval Setting] (更新间隔设置) 对话框。

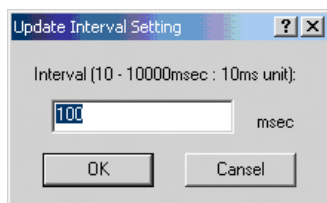


图 5.12 [Update Interval Setting] (更新间隔设置) 对话框

可为每个窗口单独指定更新间隔。
初始值为 100 ms。

5.3.3 清除 RAM 监控存取历史记录

从 [RAM Monitor] (RAM 监控) 窗口的上下文菜单中选择 [Access Data Clear] (存取数据清除)。对 RAM 监控区域进行的所有存取的历史记录将被清除。

注意：如果在用户程序运行时执行此功能，则用户程序的实时功能可能由于发生存储器转储而丢失。


5.3.4 清除 RAM 监控错误检测数据

从 [RAM Monitor] (RAM 监控) 窗口的上下文菜单中选择 [Error Detection Data Clear] (错误检测数据清除)。RAM 监控区域的所有未初始化存储器和未引用存储器的已检测数据将被清除。

5.4 显示当前状态

5.4.1 显示仿真器状态

要了解仿真器的当前状态，请显示 [Status]（状态）窗口。

要打开 [Status]（状态）窗口，请从 [View]（视图）菜单中选择 [CPU] [Status]（状态），或单击 [View Status]（视图状态）工具栏按钮 。

在程序执行过程中，此窗口不会更新显示的状态。

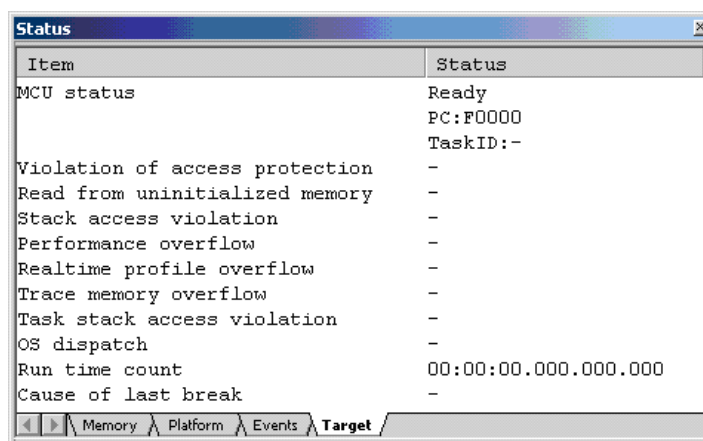


图 5.13 [Status]（状态）窗口

[Status]（状态）窗口包含以下四页。

表 5.7 [Status]（状态）窗口的页列表

页名称	描述
[Memory]（存储器）	显示与存储器资源相关的信息。
[Platform]（平台）	显示与仿真器和调试相关的信息。
[Events]（事件）	显示与事件相关的信息。
[Target]（目标）	显示与目标 MCU 相关的信息。

5.4.2 在状态栏中显示仿真器状态

可在状态栏中显示仿真器的状态。

右键单击状态栏可显示这些项目。请在状态栏中检查要显示的项目。

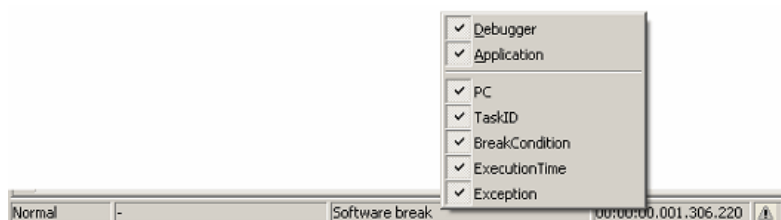


图 5.14 状态栏

表 5.8 状态栏中显示的仿真器状态的项目列表

项目	描述
[PC]	PC 值 执行过程中: PC 值 中断期间: [Normal] (正常)
[Task ID] (任务 ID)	任务 ID、任务项目标签
[BreakCondition] (断点条件)	用户程序的暂停原因
[ExecutionTime] (执行时间)	时间计量结果
[Exception] (异常)	异常事件的状态

(1) 当多个暂停原因发生时。

单击显示暂停原因的状态栏区域时, 会显示一个提示框。

可在提示框中检查当前发生的暂停原因。

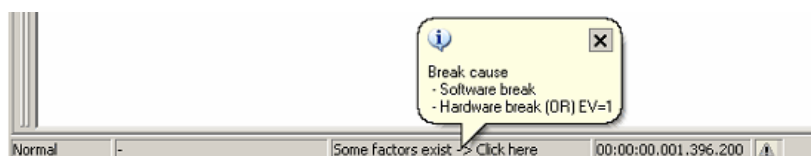


图 5.15 暂停原因发生时显示的暂停原因的示例

(2) 当异常事件发生时。

当异常事件发生时, 会在状态栏提示框中显示警告。

不会显示在 [Configuration properties] (配置属性) 对话框的 [Exception Warning] (异常警告) 页上没选中的异常事件。

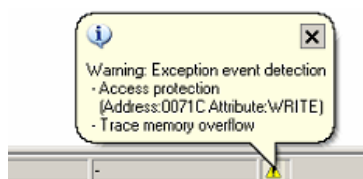



图 5.16 异常事件发生时显示的警告的示例

5.5 定期读出和显示仿真器状态

5.5.1 定期读出和显示仿真器信息

要了解用户程序运行或保持空闲时不断变化的仿真器信息，请使用 [Extended Monitor]（扩展监控器）窗口。

扩展监控器功能仅监控从用户系统或 MCU 输出的信号，而不影响用户程序的执行。

要打开 [Extended Monitor]（扩展监控器）窗口，请从 [View]（视图）菜单中选择 [CPU] [Extended Monitor]（扩展监控器），或单击 [Extended Monitor]（扩展监控器）工具栏按钮 。

用户程序执行过程中，显示的项目更新间隔约为 1,000 ms，而在中断期间，则更新间隔约为 5,000 ms。

注意：只有当用户程序正在执行时，才能够测量“CPU 时钟”。

Item	Value
User System Connection	DISCONNECT (Disconnect: CNN0, CNN1, CNN2)
User System Power Source	DISCONNECT (Vcc1:0.6 v, Vcc2:0.4 v)
User System RESET#	High
User System NMI#	High
User System CNVss	-
User System HOLD#	High
User System RDY#	High
User System BYTE	-
User System MODE	Not support
CPU Clock	Can not measure
Main Clock(XIN)	Emulator 20.0 MHz
Sub Clock(XCIN)	Emulator 33.0 kHz

图 5.17 [Extended Monitor]（扩展监控器）窗口

5.5.2 选择要显示的项目

从 [Extended Monitor] (扩展监控器) 窗口的上下文菜单中选择 [Properties] (属性), 将显示 [Extended Monitor Configuration] (扩展监控器配置) 对话框。

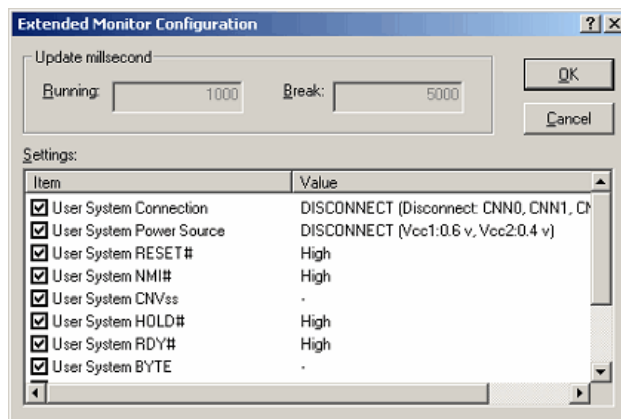


图 5.18 [Extended Monitor Configuration] (扩展监控器配置) 对话框

通过此对话框, 可以设置要在 [Extended Monitor] (扩展监控器) 窗口中显示的每个项目。

5.6 使用软件断点

5.6.1 使用软件断点

软件断点使用 **BRK** 指令重写指定地址的指令码, 产生 **BRK** 中断, 从而导致用户程序停止运行。从这种意义上说, 它是一个执行前的断点功能。

可以设置 4096 个断点。

如果设置了多个软件断点, 则程序会在到达其中任何一个断点时中断。

(1) 在软件断点中断时

在运行创建的程序时, 如果到达已设置为软件断点的地址, 则在 [Output] (输出) 窗口的 [Debug] (调试) 页上显示信息 “Software Breakpoint” (软件断点), 程序在该处停止运行。此时, [Editor] (编辑器) 或 [Disassembly] (反汇编) 窗口将会更新, 在 [S/W Breakpoints] (软件断点) 列中, 程序停止的位置会标记一个箭头 [➡]。

注意: 断点发生时, 程序会在执行设置了软件断点的行或指令之前停止。如果在程序在该软件断点停止后选择 [Go] (执行) 或 [Step] (单步), 则程序会从标记箭头的那一行再启动。

5.6.2 添加 / 移除软件断点

按照以下方法之一添加或移除软件断点。

- 从 [Editor]（编辑器）或 [Disassembly]（反汇编）窗口
- 从 [Breakpoints]（断点）对话框（只能移除断点）
- 从命令行

1. 从 [Editor]（编辑器）或 [Disassembly]（反汇编）窗口

- A. 检查确认当前打开的 [Editor]（编辑器）或 [Disassembly]（反汇编）窗口包括了要设置软件断点的位置。
- B. 在 [S/W Breakpoints]（软件断点）列中，双击要让程序停止的那一行。

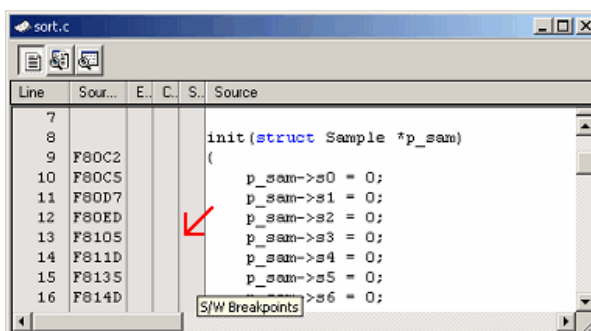


图 5.19 [Editor]（编辑器）窗口

或者可以使用以下所述的方法来设置断点。从上下文菜单中选择 [Toggle Breakpoint]（切换断点），或按键盘上的 F9 键。

- C. 如果设置了软件断点，则在 [Editor]（编辑器）或 [Disassembly]（反汇编）窗口的 [S/W Breakpoints]（软件断点）列中的相应位置会显示一个红色圆圈 [●]。

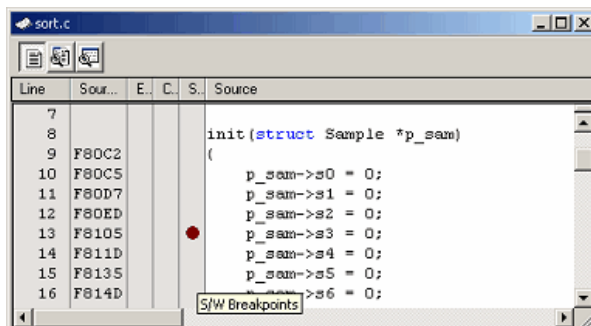


图 5.20 [Editor]（编辑器）窗口

再双击一次可以移除断点。

5.6.3 允许 / 禁止软件断点

按照以下方法之一允许或禁止软件断点。

- 从 [Editor]（编辑器）或 [Disassembly]（反汇编）窗口
- 从 [Breakpoints]（断点）对话框
- 从命令行

1. 从 [Editor]（编辑器）或 [Disassembly]（反汇编）窗口

- A. 将光标放在软件断点所在的行中，然后从上下文菜单中选择 [Enable/Disable Breakpoint]（允许 / 禁止断点）。或同时按 **Ctrl** 和 **F9** 键。

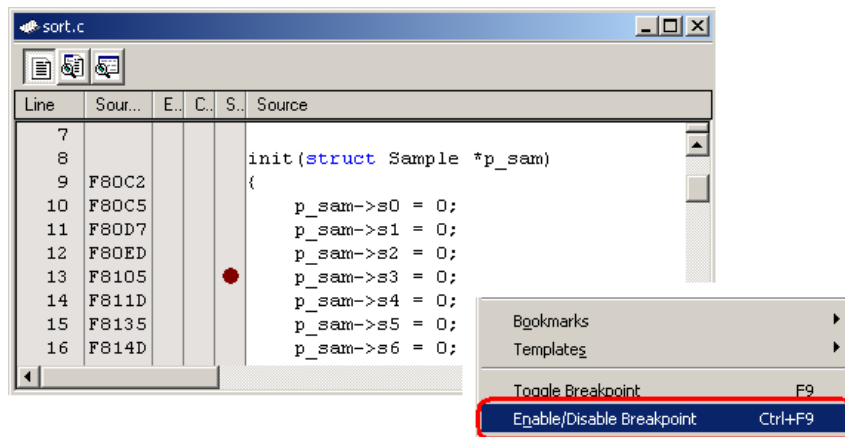


图 5.21 [Editor]（编辑器）窗口和弹出菜单

- B. 交替允许或禁止软件断点。

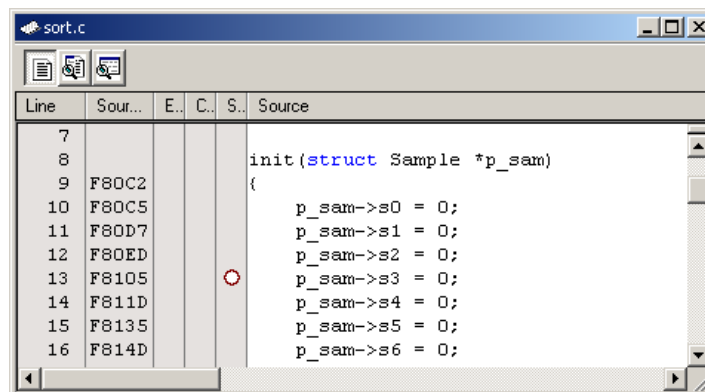


图 5.22 [Editor]（编辑器）窗口

2. 从 [Breakpoints] (断点) 对话框

- A. 从 [Edit] (编辑) 菜单中选择 [Source Breakpoints] (源断点), 显示 [Breakpoints] (断点) 对话框。在此对话框中, 可以交替允许或禁止当前设置的某个断点, 也可以移除该断点。

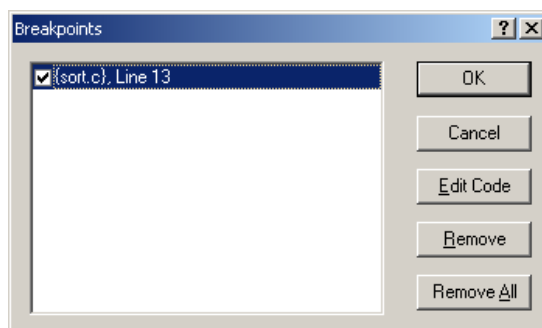


图 5.23 [Breakpoints] (断点) 对话框

5.7 使用事件

5.7.1 使用事件

事件指在程序执行过程中发生的一些现象的组合。

E100 仿真器允使用已设置的事件作为中断, 跟踪或性能功能的条件。

最多可以同时 在 16 个点设置事件。

这 16 个点可以在用户需要的任何位置。

可以注册所创建的事件, 以备今后重复使用。

(1) 事件的类型

有以下类型的事件。

表 5.9 事件类型列表

取指令	在 CPU 执行指定地址的指令时, 会检测到此类型事件。 此类型事件不是在指令预取的周期中检测到的, 而是在 CPU 执行的周期中检测到的。
数据存取	在指定条件下, 对指定地址或指定地址范围进行存取时, 会检测到此类型事件。
中断	会检测中断生成和中断终止。
触发器输入	如果从外部触发器信号输入电缆馈送的信号处在指定状态下, 会检测到此类型事件。

(2) 事件组合

可以组合使用两个或更多事件，指定以下组合条件之一。

表 5.10 事件组合列表

OR	只要发生任一指定事件，就满足条件。
AND（累加）	发生所有指定事件时，即满足条件，与时间轴无关。
AND（同时）	同时发生所有指定事件时，则满足条件。
子例程	当指定事件在指定地址范围内发生时，则满足条件。
顺序	当指定事件按照指定顺序发生时，则满足条件。
状态转换	当事件在状态转换图中指定的条件下发生时，则满足条件。

5.7.2 添加事件

按照以下方法之一添加事件。

- 创建新事件
- 通过从另一个窗口拖放添加
- 从命令行添加

1. 创建新事件

[从任何设置对话框创建事件]

- A. 单击 [Add]（添加）按钮，或选择要输入的那一行并双击。

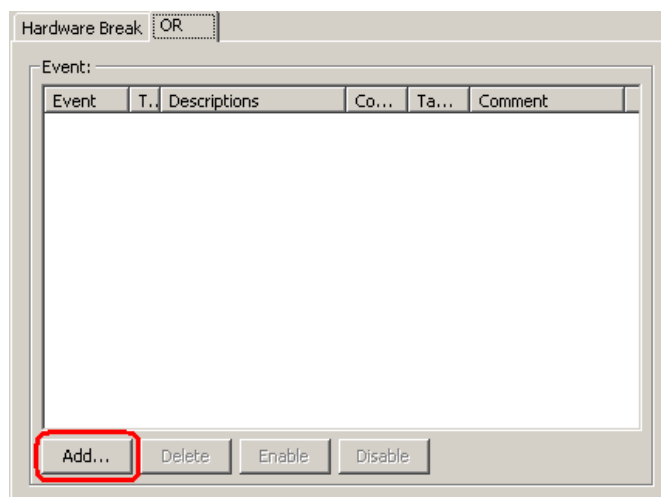


图 5.24 [Hardware Break]（硬件断点）对话框

- B. 此时将显示如下所示的 [Event]（事件）对话框。在此对话框中设置详细的事件条件，然后单击 [OK]（确定）按钮。

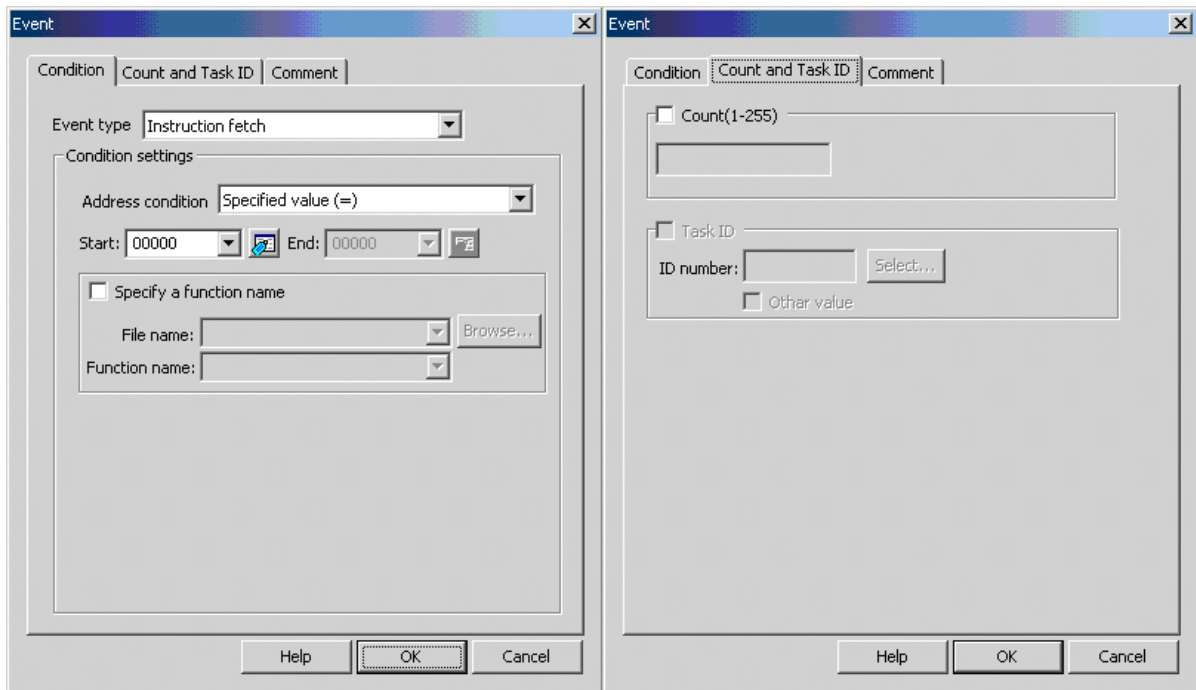


图 5.25 [Event]（事件）对话框

C. 将在指定位置添加事件。

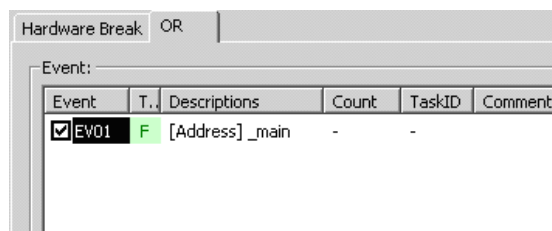


图 5.26 [Hardware Break]（硬件断点）对话框

D. 在创建事件时，如果事件超过 16 个点，则会显示错误。如果创建的事件超过 16 个点，则所添加的事件不会生效。

[从 [Registered Events] (已注册事件) 对话框添加事件]

- A. 单击 [Registered Events] (已注册事件) 对话框中的 [Add] (添加) 按钮。

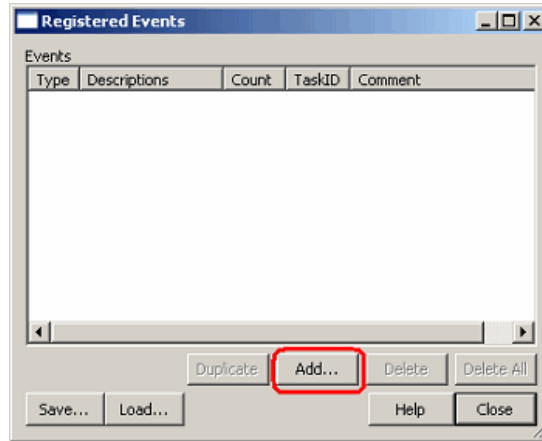


图 5.27 [Registered Events] (已注册事件) 对话框

- B. 此时将显示如下所示的 [Event] (事件) 对话框。在此对话框中设置详细的事件条件。如果需要，请输入注释。然后单击 [OK] (确定) 按钮。

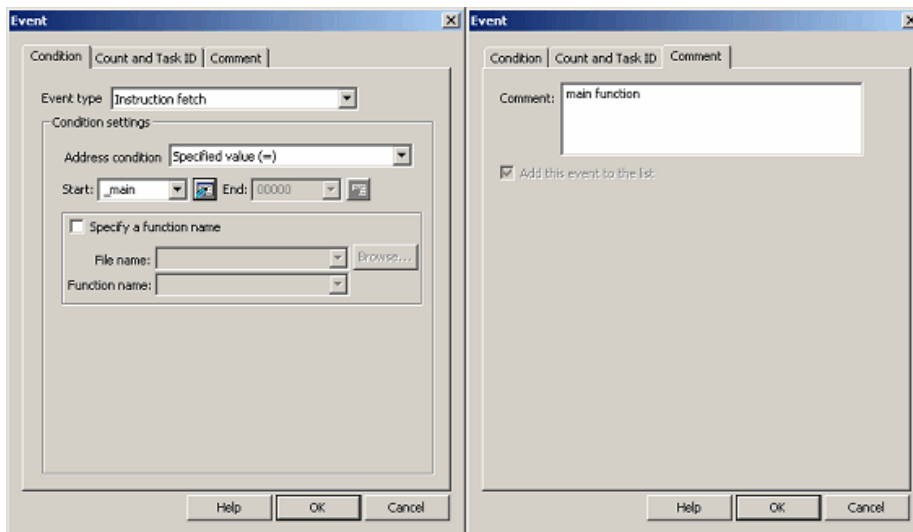


图 5.28 [Event] (事件) 对话框

- C. 事件将添加到已注册事件列表中。

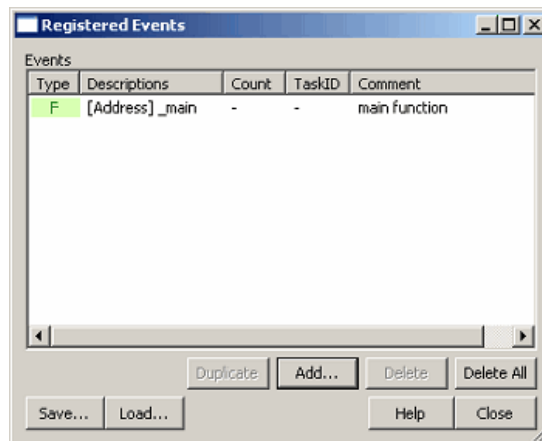


图 5.29 [Registered Events]（已注册事件）对话框

2. 从 [Editor]（编辑器）窗口的 [event]（事件）列添加事件
[在添加硬件断点]

- A. 双击或右键单击 [Editor]（编辑器）窗口的 [event]（事件）列中的任何位置，从显示的弹出菜单中选择 [HW Break Point]（硬件断点）。
可以将对该地址的取操作作为一个条件 => 取指令条件，来设置硬件断点。

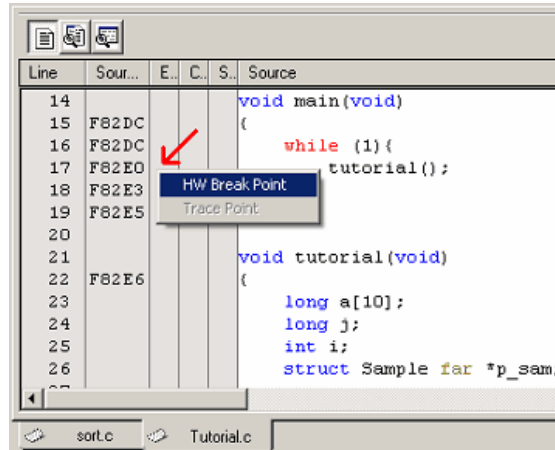


图 5.30 [Editor]（编辑器）窗口

- B. 如果事件计数还有空间，则用户从 [Editor]（编辑器）窗口添加的事件会作为 OR 条件添加到其他事件。如果没有空间，则显示错误信息。

注意：如果正在 [Hardware Break]（硬件断点）对话框中进行编辑工作，则不能从 [Editor]（编辑器）窗口的 [event]（事件）列设置硬件断点。

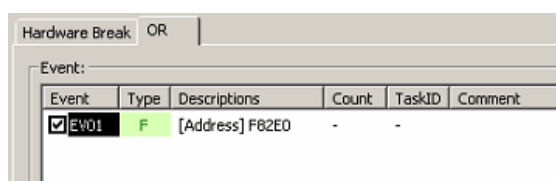


图 5.31 [Hardware Break] (硬件断点) 对话框

[添加跟踪点]

双击或右键单击 [Editor] (编辑器) 窗口的 [event] (事件) 列中的任何位置, 从显示的弹出菜单中选择 [Trace Point] (跟踪点)。

可以将对该地址的取操作作为一个条件 => 取指令条件, 来设置跟踪点

在 [Editor] (编辑器) 窗口的 [Event] (事件) 列中双击取指令事件, 以删除该事件。

注意: 在以下情况下, 不能从 [Editor] (编辑器) 窗口的 [Event] (事件) 列设置跟踪点。

- 编辑 [Trace conditions] (跟踪条件) 对话框中的内容时
- 在选择 [Fill until stop] (填充直至停止) 或 [Fill until full] (填充至满) 跟踪模式时

3. 通过拖放添加事件

[在 [Editor] (编辑器) 窗口中拖放变量和函数名称]

- A. 将变量名称拖放到 [Event] (事件) 列中, 可以将对该变量的存取作为一个条件 => 数据存取条件, 来设置事件。

此时, 变量的大小会自动设置为数据存取事件的条件。

只有大小为 1 字节或 2 字节的全局或静态变量才能注册为事件。函数中的静态变量不能注册为事件。

- B. 将函数名称拖放到 [Event] (事件) 列中, 可以将对该函数起始地址的取指令作为条件来设置事件。

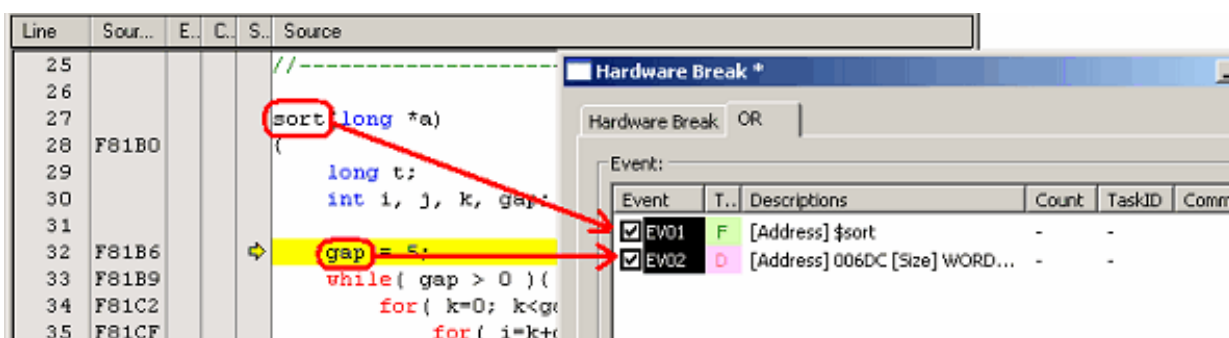


图 5.32 [Editor] (编辑器) 窗口和 [Hardware Break] (硬件断点) 对话框

[在 [Memory] (存储器) 窗口中拖放地址范围]

在 [Memory] (存储器) 窗口中选择存储器内容, 并将其拖放到 [Event] (事件) 列中。通过这种方式, 可以将所选存储器内容的地址范围作为一个条件 => 数据存取条件, 来设置数据存取事件。

[在 [Label] (标签) 窗口中拖放标签]

可以将对该标签的取操作作为一个条件 => 取指令条件, 来设置事件

5.7.3 移除事件

按照以下方法之一移除事件。

[从任何设置对话框移除事件]

1. 要移除一个点，请在事件设置区中选择要移除的行，然后单击 [Delete]（删除）按钮（可以使用 Ctrl + Del 键代替单击 [Delete]（删除）按钮）。
所选事件将从事件设置区中移除。

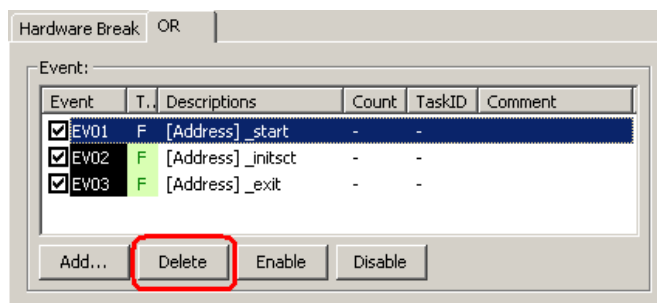


图 5.33 [Hardware Break]（硬件断点）对话框

2. 要移除多个事件，请按住 Shift 或 Ctrl 键，同时在事件设置区中选择要移除的行，然后单击 [Delete]（删除）按钮（可以使用 Ctrl + Del 键代替单击 [Delete]（删除）按钮）。
所选事件将从事件设置区中移除。

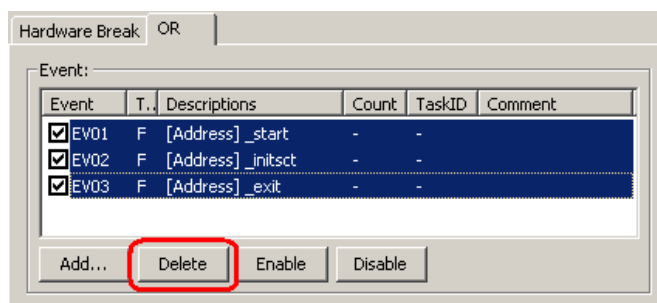


图 5.34 [Hardware Break]（硬件断点）对话框

[从 [Registered Events] (已注册事件) 对话框删除事件]

要移除一个点，请在 [Registered Events] (已注册事件) 对话框中选择要移除的行，然后单击 [Delete] (删除) 按钮 (可以使用 Ctrl + Del 键代替单击 [Delete] (删除) 按钮)。

所选事件将从已注册事件列表中移除。

要删除所有事件，请单击 [Delete All] (全部删除) 按钮。

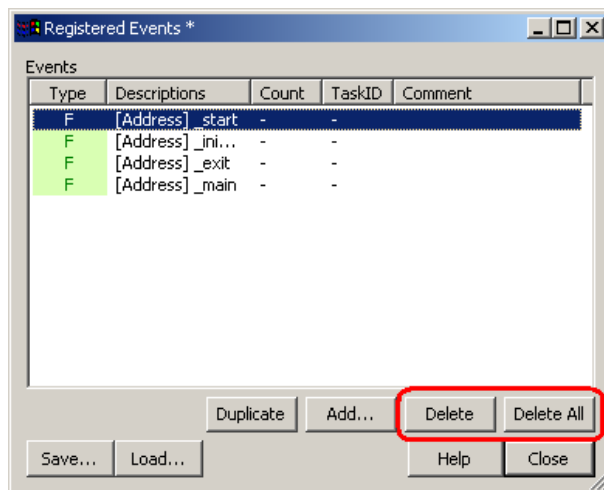


图 5.35 [Registered Events] (已注册事件) 对话框

5.7.4 注册事件

“注册事件”指将事件放到已注册事件列表中。已注册事件可在以后重复使用。按照以下方法之一注册事件。最多可以注册 256 个事件。

1. 注册事件

[从 [Event]（事件）对话框创建事件]

- A. 显示 [Event]（事件）对话框的 [Comments]（注释）页，并选中“Add this event to the list”（将此事件添加到列表中）复选框。然后单击 [OK]（确定）按钮。

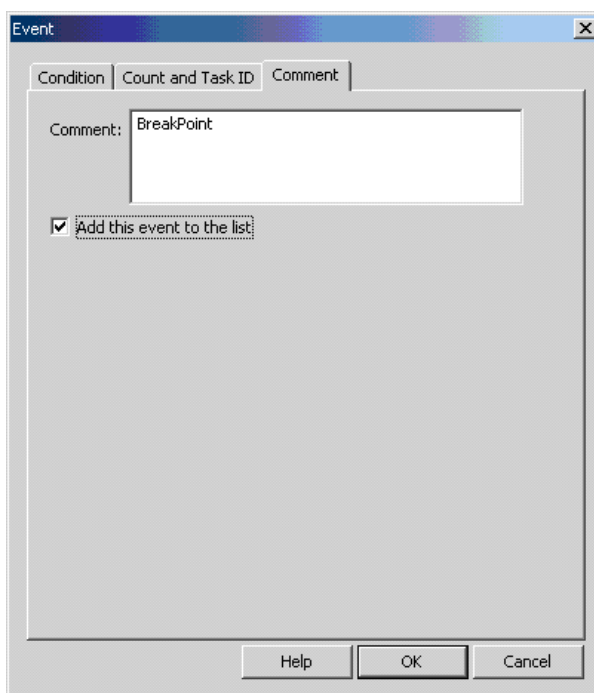


图 5.36 [Event]（事件）对话框

- B. 事件在 [Registered Events]（已注册事件）对话框中注册的同时被添加到指定位置。

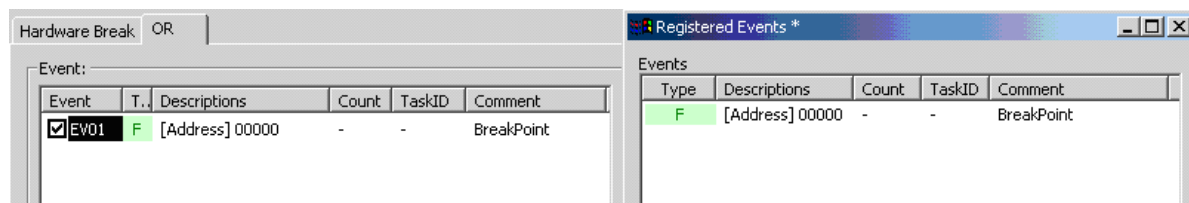


图 5.37 [Hardware Break]（硬件断点）对话框和 [Registered Events]（已注册事件）对话框

[通过拖放注册事件]

可通过将已创建的事件拖放到列表中，在 [Registered Events]（已注册事件）对话框中注册该事件。

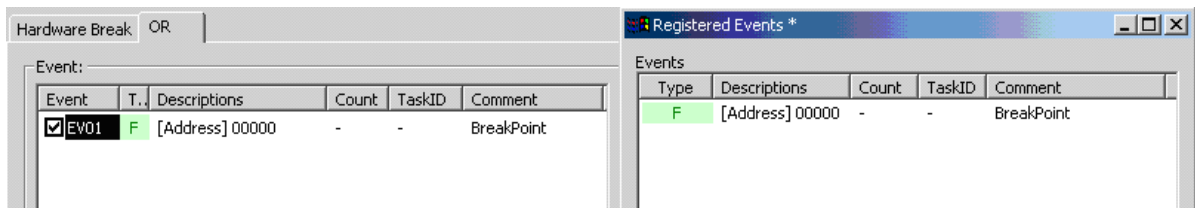


图 5.38 [Hardware Break]（硬件断点）对话框和 [Registered Events]（已注册事件）对话框

[从 [Registered Events]（已注册事件）对话框注册事件]

单击 [Add]（添加）按钮创建事件。在此处创建的事件将添加到 [Registered Events]（已注册事件）中。

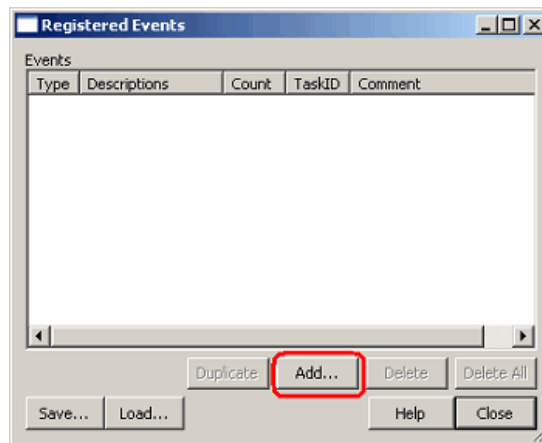


图 5.39 [Registered Events]（已注册事件）对话框

2. 附加注释

根据需要向已注册事件添加注释。检查 [Registered Events]（已注册事件）对话框，了解已注册的内容和注释。

5.7.5 每次输入事件或重复使用事件

可以通过以下两种方法设置任何相关函数中的事件。

一种方法是每次在相应的设置对话框中创建事件。

另一种方法是从已注册事件列表中选择要使用的一个条件，然后将其拖放到要在其中设置事件的条件区域。

前一种方法在此处称为“每次输入事件”，后一种方法则称为“重复使用事件”。

[每次输入事件]

这是只使用一次的情况。创建的事件“未注册过。”

在事件被使用后（即更改或删除），其设置将不再存在。

通过双击 [Editor]（编辑器）窗口的 [Event]（事件）列创建的事件就是每次输入的事件。

[重复使用事件]

在 [Registered Events]（已注册事件）列表中注册的任何事件都可以重复使用，方法是将其拖放到任何相关函数的条件设置区中。

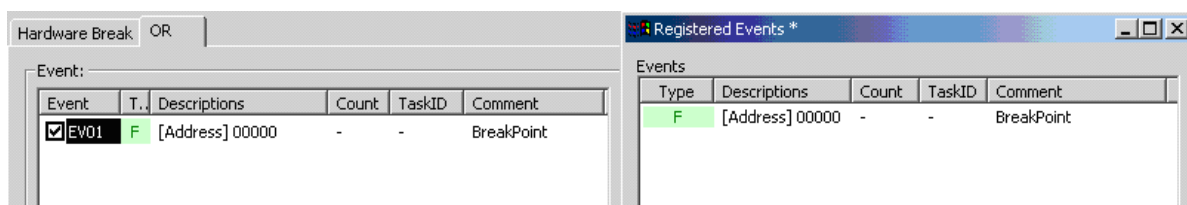


图 5.40 事件重复使用示意图

(1) 拖放到多个函数中

可以将 [Registered Events]（已注册事件）列表中的一个事件拖放到多个函数中。

如果事件内容在拖放之后被更改，用户所做的更改不会反映在已注册事件列表端。

(2) 在已注册事件列表中注册重复事件

即使多个事件设置了相同内容，也可以在列表中相互重叠注册。

5.7.6 应用事件

要在创建事件之后允许该事件的设置，请单击 [Apply]（应用）按钮。在单击 [Apply]（应用）按钮之前，所设置的内容不会生效。

[Hardware break]（硬件断点）、[Trace conditions]（跟踪条件）或 [Performance Analysis Conditions]（性能分析条件）对话框标题栏上的标题后的 [*] 表示正在编辑某些设置。如果正在进行编辑工作，则不能通过 [Editor]（编辑器）窗口的 [Event]（事件）列或命令行更改设置。

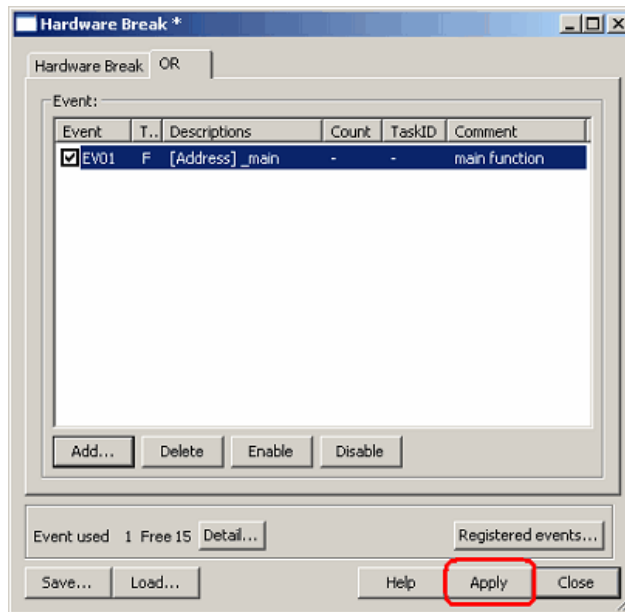


图 5.41 应用设置

5.8 设置硬件断点条件

5.8.1 设置硬件断点条件

如果设置了硬件断点，则在检测到所设置的事件或现象后（即遇到硬件断点），会让用户程序停止运行指定数量的周期。最多可将 16 个事件指定为硬件断点条件。

5.8.2 设置硬件断点

(1) 设置硬件断点

对于硬件断点，可以设置 OR 条件、其他条件（包括 AND（累加）、AND（同时）、子例程、顺序或状态转换）和异常事件检测。

OR 条件、其他条件和异常事件检测既可同时设置，也可一次只设置一个。

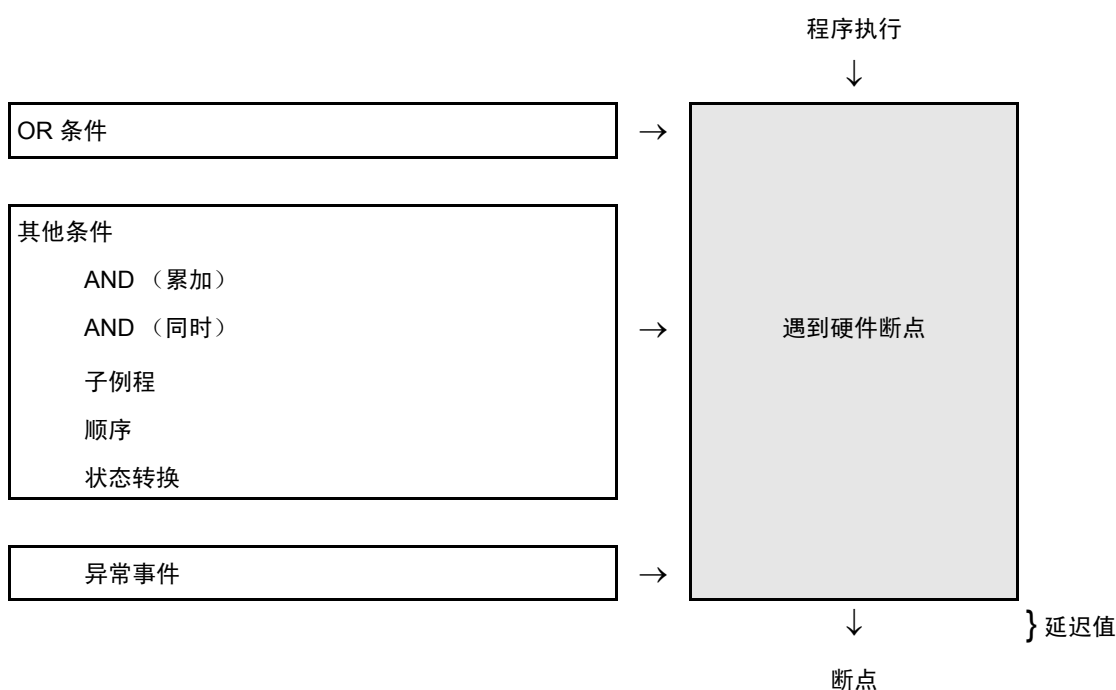


图 5.42 硬件断点概图

(2) 设置 OR 条件

可以选择允许或禁止 OR 条件。默认情况下，允许 OR 条件。

要禁止 OR 条件，请取消选中“OR Condition”（OR 条件）左侧的复选框。

如果在禁止 OR 条件时，通过双击 [Editor]（编辑器）窗口添加了事件，则会自动允许 OR 条件。

如果在禁止 OR 条件时重新允许 OR 条件，则以前设置的事件会恢复，同时其 [OR condition]（OR 条件）复选框被选中。

但是，如果已重新允许某个事件的 OR 条件，但事件数超过了 16 个事件的最大值，则该事件恢复时不会选中它的 [OR condition]（OR 条件）复选框（禁止）。

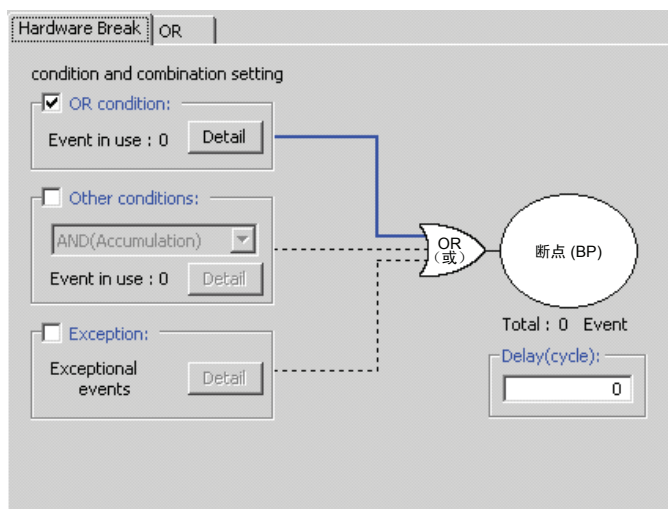


图 5.43 [Hardware Break]（硬件断点）对话框

表 5.11 OR 条件

	类型	描述
1	OR 条件	发生任一所设置的事件时，就会遇到断点。

(3) 设置其他条件

可以选择五个可用选项中的一个：[AND (Accumulation)]（AND（累加））、[AND (Simultaneous)]（AND（同时））、[Subroutine]（子例程）、[Sequential]（顺序）和 [State Transition]（状态转换）。要设置任何条件，请选中“Other Conditions”（其他条件）左侧的复选框。默认情况下，其他条件被禁止（“Other Conditions”（其他条件）左侧的复选框没有选中）。

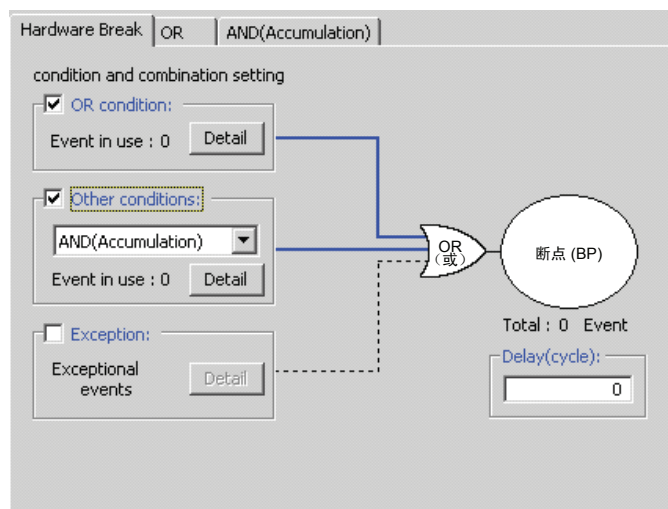


图 5.44 [Hardware Break]（硬件断点）对话框

表 5.12 其他条件

	类型	描述
1	AND (Accumulation) (AND (累加))	发生设置的所有事件时, 就会遇到断点, 与时间轴无关。
2	AND (Simultaneous) (AND (同时))	同时发生设置的所有事件时, 就会遇到断点。
3	Subroutine (子例程)	当指定事件在指定地址范围内 (子例程、函数) 发生时, 就会遇到断点。
4	Sequential (顺序)	6 个步骤 (正向) + 复位点 当设置的事件按照指定顺序发生时, 就会遇到断点。
5	State Transition (状态转换)	3 个步骤、9 条路径 + 复位点 当设置的事件按照指定顺序发生时, 就会遇到断点。

可以使用 Ctrl + Del 键删除每个条件的列表中显示的事件。

(4) 异常事件检测

指定是否要将以下异常事件的检测用作断点。

- 存取保护的违规操作
- 从未初始化的存储器读取
- 堆栈存取的违规操作
- 性能上溢
- 实时分析上溢
- 跟踪存储器上溢
- 任务堆栈存取的违规操作
- 操作系统调度

(5) 指定延迟值

在遇到断点后, 程序会暂停指定数量的周期。可以在 0 到 65,535 个总线周期范围内设置断点延迟值。
(初始值 = 0)

5.8.3 保存 / 加载硬件断点的设置内容

(1) 保存硬件断点设置

单击 [Hardware Break] (硬件断点) 对话框的 [Save] (保存) 按钮。将显示 [Save] (保存) 对话框。指定要保存断点设置的文件的名称。文件扩展名为 .hev。如果省略扩展名, 将自动附加扩展名 .hev。

(2) 加载硬件断点的设置内容

单击 [Hardware Break] (硬件断点) 对话框的 [Load] (加载) 按钮。将显示 [Load] (加载) 对话框。指定要加载的文件的名称。

加载文件时, 此前使用的硬件断点设置将被放弃, 并使用加载的设置来复位硬件断点。

单击 [Hardware Break] (硬件断点) 对话框的 [Apply] (应用) 按钮, 确认已加载的硬件断点设置。

5.9 查看跟踪信息

5.9.1 查看跟踪信息

跟踪是一种功能，用于在用户程序执行过程中获取每个周期的总线信息，并将该信息存储在跟踪存储器中。使用跟踪功能，用户可以跟踪应用程序执行流程，并且检查发生问题的点。

使用 E100 仿真器可以获取最多 4M 个总线周期。

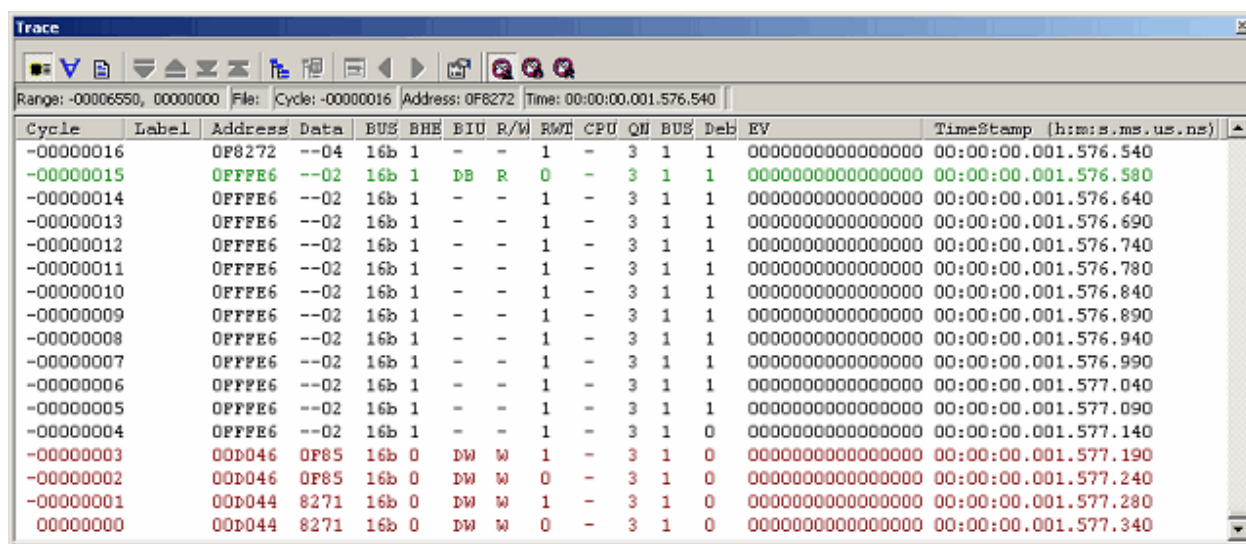
程序执行停止时（由于异常中断、强制暂停或断点），则在程序停止时存储在跟踪存储器中的内容会显示为跟踪结果，即使在尚未遇到跟踪点的情况下也是如此。

5.9.2 获取跟踪信息

按照 E100 仿真器的运行方式，在没有设置跟踪信息获取条件时，它默认情况下会跟踪所有总线周期，无条件地获取跟踪信息。（跟踪模式 = [Fill until stop]（填充直至停止））

在空闲模式下，在用户程序开始运行的同时，仿真器开始跟踪总线周期以获取跟踪信息，而当用户程序停止时，仿真器也停止跟踪。

所获取的跟踪信息在 [Trace]（跟踪）窗口中显示。



Cycle	Label	Address	Data	BUS	BHE	BIU	R/W	RWT	CPU	QV	BUS	Deb	EY	TimeStamp (h:m:s.ms.us.ns)
-00000016		0F8272	--04	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.576.540
-00000015		0FFFE6	--02	16b	1	DB	R	0	-	3	1	1	0000000000000000	00:00:00.001.576.580
-00000014		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.576.640
-00000013		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.576.690
-00000012		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.576.740
-00000011		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.576.780
-00000010		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.576.840
-00000009		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.576.890
-00000008		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.576.940
-00000007		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.576.990
-00000006		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.577.040
-00000005		0FFFE6	--02	16b	1	-	-	1	-	3	1	1	0000000000000000	00:00:00.001.577.090
-00000004		0FFFE6	--02	16b	1	-	-	1	-	3	1	0	0000000000000000	00:00:00.001.577.140
-00000003		00D046	0F85	16b	0	DW	W	1	-	3	1	0	0000000000000000	00:00:00.001.577.190
-00000002		00D046	0F85	16b	0	DW	W	0	-	3	1	0	0000000000000000	00:00:00.001.577.240
-00000001		00D044	8271	16b	0	DW	W	1	-	3	1	0	0000000000000000	00:00:00.001.577.280
00000000		00D044	8271	16b	0	DW	W	0	-	3	1	0	0000000000000000	00:00:00.001.577.340

图 5.45 [Trace]（跟踪）窗口

此时将显示以下信息项目。（适用于总线显示。）

表 5.13 显示项目 (1)

列	描述
Cycle (周期)	跟踪存储器中存储的周期数。获取的最后一个周期的编号为 0，而更早的周期则分配更小的编号（如 -1、-2 等），按顺序重新跟踪过去的周期。如果设置了延迟计数，则满足跟踪停止条件的周期的编号为 0，而对于满足条件之前执行的周期（延迟期间的周期），则按顺序分配更大的编号（如 +1、+2 等），直至获取的最后一个周期。
Label (标签)	与地址相对应的标签（仅在设置了标签时显示）。
Address (地址)	地址总线的地址。 注意：在使用 4M 模式的情况下，当存取存储体 0~6 时，地址 b31 将为“1b”。当 b31 为“1b”时，b30~28 显示在正在存取的存储体。
Data (数据)	数据总线的数据。以十六进制显示。
BUS (总线)	显示外部数据总线宽度，当总线为 8 位宽度时，表示为“8b”，当总线为 16 位宽度时，则表示为“16b”。
BHE	显示 BHE（允许高位字节）信号的状态（1 或 0）。如果此信号为“0”，则表示 CPU 在存取奇地址。
BIU	显示 BIU（总线接口单元）与存储器和 I/O 之间的状态。 - 无变化 DMA 从非 CPU 源请求的数据存取，例如 DMA INT INTACK 序列启动 IB 从 CPU 请求的指令码读（以字节为单位） DB 从 CPU 请求的数据存取（以字节为单位） IW 从 CPU 请求的指令码读（以字为单位） DW 从 CPU 请求的数据存取（以字为单位）
R/W	显示数据总线状态。读状态时，表示为“R”。写状态时，表示为“W”。没有进行存取时，表示为“-”。
RWT	指示总线周期的有效位置的信号。有效时，该信号为“0”。当此信号为“0”时，地址、数据和 BIU 行有效。
CPU	显示 CPU 和 BIU（总线接口单元）之间的状态。 - 无变化 CB 操作码读（以字节为单位） RB 操作数读（以字节为单位） QC 指令队列缓冲器清除 CW 操作码读（以字为单位） RW 操作数读（以字为单位）
QN	显示存储在指令队列缓冲器中的字节数。显示的字节数在 0 到 4 范围内。 注意：使用软件断点停止用户程序时，软件断点发生后的下一个周期中的 QN（存储在指令队列缓冲器中的字节数）会显示不正确。
BUSACC	用户程序执行过程中，当调试器操作执行存储器存取时，会在仿真器占用 MCU 总线期间显示“0”。 注意：执行存储器存取的过程中，用户程序会暂停。
Debug (调试)	用户程序执行过程中，当调试器操作执行存储器存取时，会在仿真器占用 MCU 总线期间显示“0”。 注意：执行存储器存取的过程中，用户程序会暂停。

表 5.13 显示项目 (2)

列	描述
EV	所设置事件发生时的事件编号。 要显示 EV 列，需要从 [Trace]（跟踪）窗口菜单显示的 [Trace conditions]（跟踪条件）对话框的 [Option]（选项）页中选择 [EV number]（EV 编号）。
TID	任务 ID（使用 RTOS 时）。 显示示例：任务 ID（任务条目标签）以 1 (_Task1) 格式显示。 要显示 [Task ID]（任务 ID）列，需要从 [Trace]（跟踪）窗口菜单显示的 [Trace conditions]（跟踪条件）对话框的 [Option]（选项）页中选择 [Task ID]（任务 ID）。
EXT	显示从外部触发器电缆馈送的信号，当信号为高时，表示为“1”，当信号为低时，表示为“0”。 要显示 EXT 列，需要从 [Trace]（跟踪）窗口菜单显示的 [Trace conditions]（跟踪条件）对话框的 [Option]（选项）页中选择 [External trigger]（外部触发器）。
TimeStamp (时间戳)	显示自从目标程序启动以来所经过的时间。 每次用户程序开始运行时，时间戳开始从 0 计数。 注意：如果计数器上溢，时间会显示不正确。

可以隐藏 [Trace]（跟踪）窗口中不需要的列。要隐藏某一列，请右键单击标题列，并从上下文菜单中选择要隐藏的列。

5.9.3 设置跟踪信息获取条件

跟踪缓冲区的大小受到限制，因此当缓冲区充满时，将用新数据盖写旧的跟踪数据，从最旧的数据开始按顺序进行。

通过设置跟踪信息获取条件，可以只获取有用的跟踪信息，从而有效地利用跟踪缓冲区。

要设置跟踪信息获取条件，请使用从 [Trace]（跟踪）窗口的上下文菜单中选择 [Acquisition]（获取）时显示的 [Trace conditions]（跟踪条件）对话框。

(1) 设置跟踪模式

首先选择一种跟踪模式。

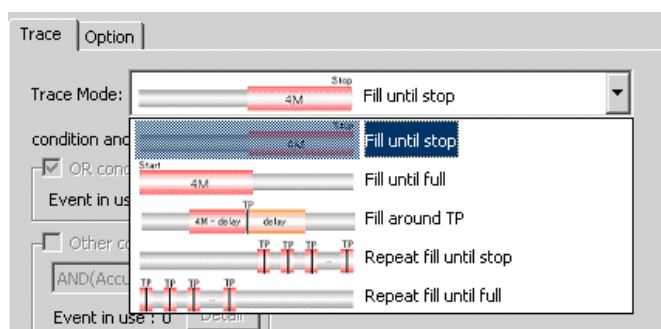


图 5.46 [Trace conditions]（跟踪条件）对话框

(2) 设置跟踪点

如果选择 [Fill around TP]（围绕 TP 填充）、[Repeat fill until stop]（重复填充直至停止）或 [Repeat fill until full]（重复填充至满）作为跟踪模式，请设置跟踪点。

对于跟踪点，可以根据条件和异常事件设置事件。

此外，对于“Fill around TP”（围绕 TP 填充）模式，可以设置一个延迟值。

(3) 设置捕捉 / 不捕捉

如果选择的跟踪模式为“Fill until stop”（填充直至停止）、“Fill until full”（填充至满）或“Fill around TP”（围绕 TP 填充），可在 [Record condition]（记录条件）分组框中指定捕捉 / 不捕捉。

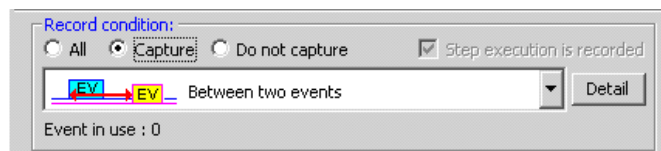


图 5.47 [Record condition]（记录条件）分组框

可以选择只提取事件指定的跟踪信息的必需部分，或者删除不需要的部分。

(4) 记录单步执行

如果选择的跟踪模式是“Fill until stop”（填充直至停止），则可记录单步执行。要记录单步执行，请在 [Recording condition]（记录条件）分组框中选中 [Step execution is recorded]（记录单步执行）复选框。

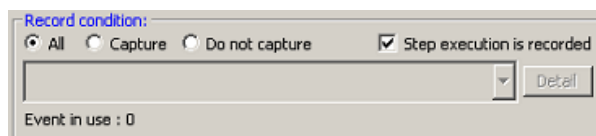


图 5.48 记录单步执行

可记录的分步执行模式为“Step In”（跳入）、“Step Over”（跳过）或“Step Out”（跳出）。

(5) 设置跟踪获取方法

使用 [Trace conditions]（跟踪条件）对话框的 [Options]（选项）页，设置与整个跟踪相关的获取方法。默认情况下，选择 [External Trigger]（外部触发器）进行跟踪获取。

5.9.4 设置跟踪模式

(1) 设置跟踪模式

可以使用以下五种跟踪模式。

表 5.14 跟踪模式

	停止模式	描述
1	Fill until stop (填充直至停止)	在程序停止前一直进行跟踪获取。
2	Fill until full (填充至满)	跟踪存储器填满时跟踪获取停止。
3	Fill around TP (围绕 TP 填充)	在遇到跟踪点之后, 跟踪获取会停止指定数量的周期。可以指定延迟值, 其范围的上限为跟踪容量的最大值。
4	Repeat fill until stop (重复填充直至停止)	每次遇到跟踪点时, 对该跟踪点前后的总共 512 个周期 * 进行获取, 在程序停止运行之前会一直通过这种方式进行获取。
5	Repeat fill until full (重复填充至满)	每次遇到跟踪点时, 对该跟踪点前后的总共 512 个周期 * 进行获取。在跟踪存储器填满之前会一直通过这种方式进行获取。

注意: 以总共 512 个周期为单位进行记录, 包括遇到跟踪点时所在的行处的一个周期、跟踪点前的 255 个周期、跟踪点后的 256 个周期。

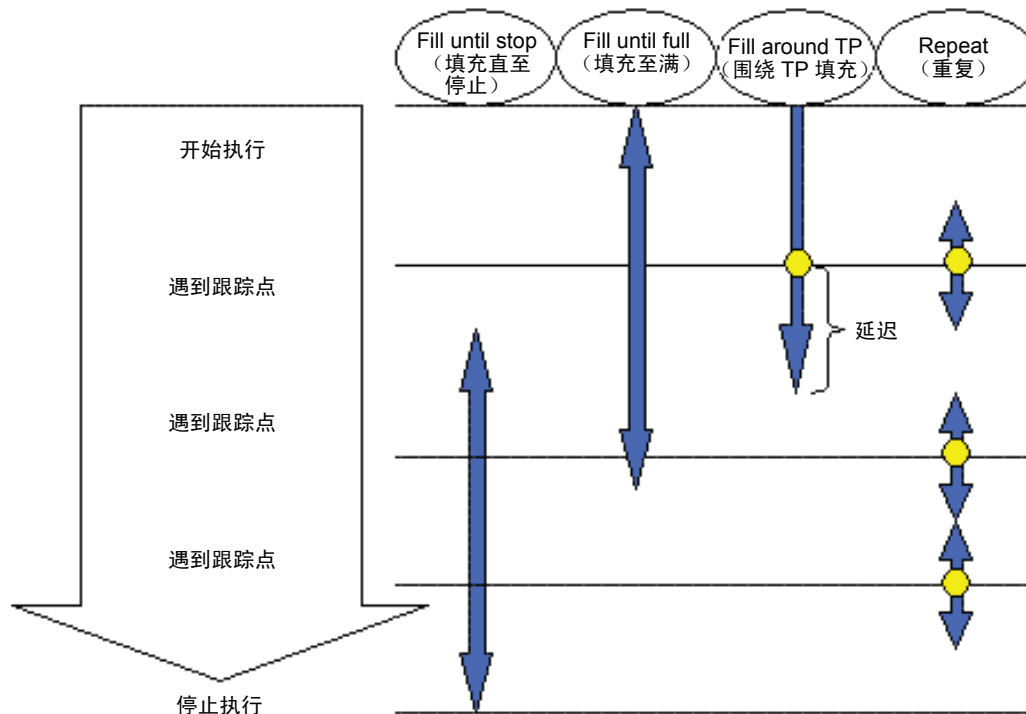


图 5.49 跟踪模式之间的差异

可指定条件因跟踪模式而异，下表对此进行了总结。

(1) Fill until stop（填充直至停止）

跟踪存储器最多可以容纳 4M 个总线周期。当缓冲器填充时，所获取的跟踪信息的新数据将盖写最旧的数据。通过此种方式，仿真器可持续获取跟踪信息。

表 5.15 可指定条件：Fill until stop（填充直至停止）

跟踪点设置	指定延迟	捕捉 / 不捕捉条件设置	单步执行记录
-	-	可以	可以

(2) Fill until full（填充至满）

如果仿真器主部件的跟踪存储器在跟踪获取过程中上溢，仿真器将停止获取跟踪信息。

表 5.16 可指定条件：Fill until full（填充至满）

跟踪点设置	指定延迟	捕捉 / 不捕捉条件设置	单步执行记录
-	-	可以	-

(3) Fill around TP（围绕 TP 填充）

在遇到跟踪点后，跟踪获取暂停指定延迟数量的周期。在此模式下，用户程序继续运行，只暂停跟踪获取。最多可以使用 16 个事件点设置复杂条件。延迟值可以选择为 0、1M、2M、3M 或 4M 个周期。

表 5.17 可指定条件：Fill around TP（围绕 TP 填充）

跟踪点设置	指定延迟	捕捉 / 不捕捉条件设置	单步执行记录
可以	可以	可以	-

(4) Repeat fill until stop（重复填充直至停止）

每次遇到跟踪点时，对该跟踪点前后的总共 512 个周期进行获取，并且通过此种方式持续进行获取。获取将持续进行，将在遇到断点或强制停止时暂停。可在 [Trace]（跟踪）窗口中查看遇到的跟踪点的位置。

表 5.18 可指定条件：Repeat fill until stop（重复填充直至停止）

跟踪点设置	指定延迟	捕捉 / 不捕捉条件设置	单步执行记录
可以	-	-	-

(5) Repeat fill until full（重复填充至满）

每次遇到跟踪点时，对该跟踪点前后的总共 512 个周期进行获取，并且通过此种方式持续进行获取。当跟踪存储器上溢时，获取暂停。可在 [Trace]（跟踪）窗口中查看遇到的跟踪点的位置。

表 5.19 可指定条件：Repeat fill until full（重复填充至满）

跟踪点设置	指定延迟	捕捉 / 不捕捉条件设置	单步执行记录
可以	-	-	-

注意：在“Repeat fill until stop”（重复填充直至停止）或“Repeat fill until full”（重复填充至满）模式下，如果在连续周期中遇到跟踪点，则只将第一个周期作为跟踪点以黄色突出显示。

5.9.5 设置跟踪点

(1) 设置跟踪点

对于跟踪点，可以设置 OR 条件、其他条件（包括 AND（累加）、AND（同时）、子例程、顺序或状态转换）和异常事件检测。

OR 条件、其他条件和异常事件检测既可同时设置，也可一次只设置一个。

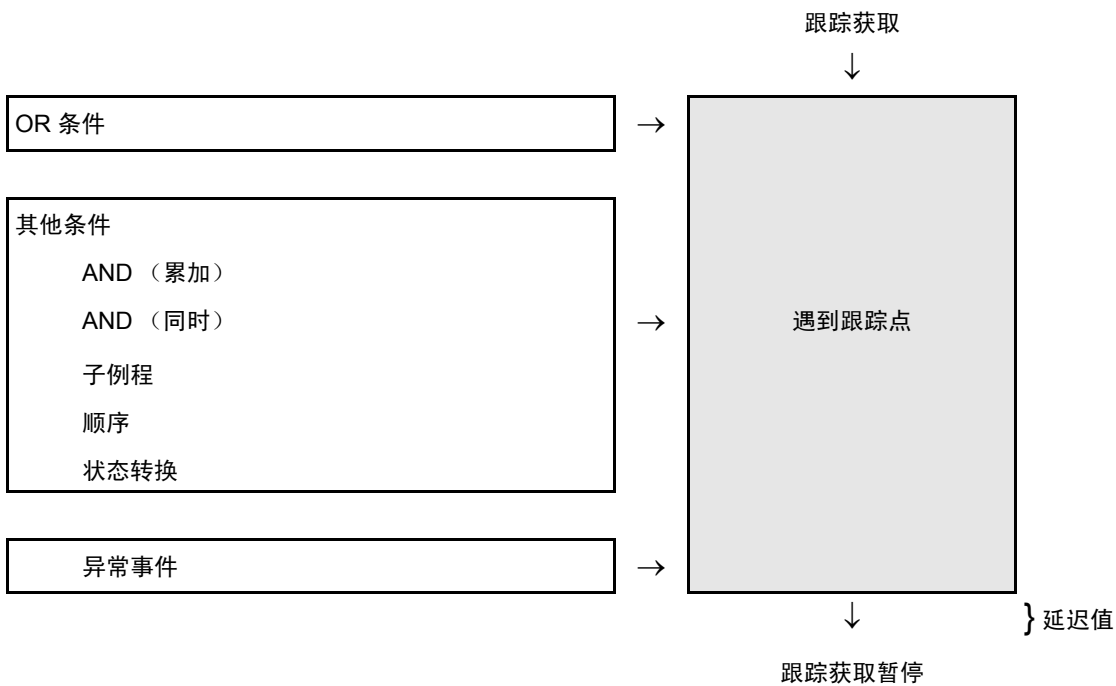


图 5.50 跟踪点概图

(2) OR 条件

可以选择允许或禁止 OR 条件。默认情况下，允许 OR 条件。

如果在禁止 OR 条件时重新允许 OR 条件，则以前设置的事件会恢复，同时其 [OR condition]（OR 条件）复选框被选中。但是，如果已重新允许某个事件的 OR 条件，但事件数超过了 16 个事件的最大值，则该事件恢复时不会选中它的 [OR condition]（OR 条件）复选框（禁止）。

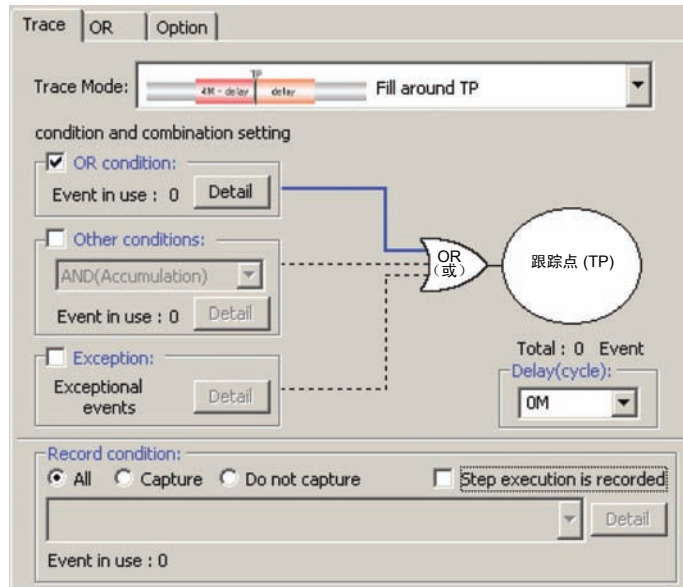


图 5.51 [Trace conditions]（跟踪条件）对话框

表 5.20 OR 条件

	类型	描述
1	OR 条件	如果发生任一所设置事件，就会遇到跟踪点。

(3) 其他条件

可以选择五个可用选项中的一个：[AND (Accumulation)]（AND（累加））、[AND (Simultaneous)]（AND（同时））、[Subroutine]（子例程）、[Sequential]（顺序）和 [State Transition]（状态转换）。要设置任何条件，请选中“Other Conditions”（其他条件）左侧的复选框。

默认情况下，其他条件被禁止（“Other Conditions”（其他条件）左侧的复选框没有选中）。

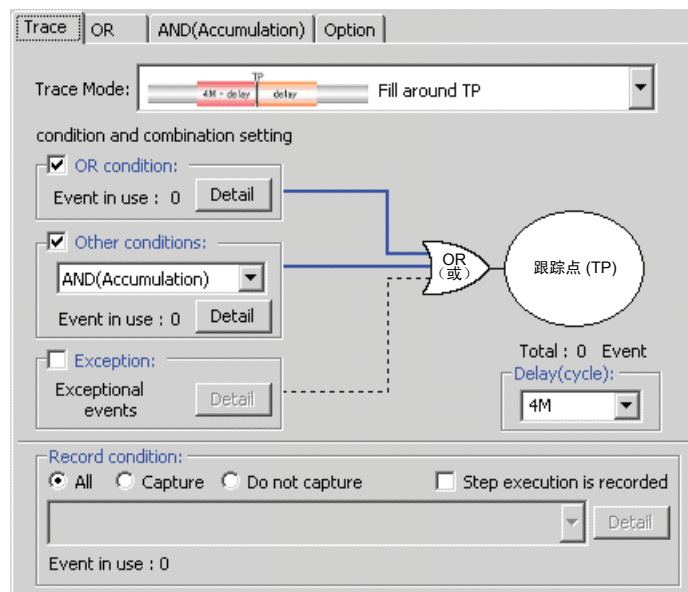


图 5.52 [Trace conditions] (跟踪条件) 对话框

表 5.21 其他条件

	类型	描述
1	AND (Accumulation) (AND (累加))	如果发生设置的所有事件，就会遇到跟踪点，与时间轴无关。
2	AND (Simultaneous) (AND (同时))	同时发生设置的所有事件时，就会遇到跟踪点。
3	Subroutine (子例程)	当指定事件在指定地址范围内 (子例程或函数) 发生时，就会遇到跟踪点。
4	Sequential (顺序)	6 个步骤 (正向) + 复位点 当设置的事件按照指定顺序发生时，就会遇到跟踪点。
5	[State Transition] (状态转换)	3 个步骤、9 条路径 + 复位点 当设置的事件按照指定顺序发生时，就会遇到跟踪点。

(4) 异常事件检测

指定是否要将以下异常事件的检测用作跟踪点。

- 存取保护的违规操作
- 从未初始化的存储器读取
- 堆栈存取的违规操作
- 性能上溢
- 实时分析上溢
- 任务堆栈存取的违规操作
- 操作系统调度

(5) 指定延迟值

在遇到跟踪点后，程序会暂停指定数量的延迟周期。

可从 0M、1M、2M、3M 或 4M 个总线周期中选择跟踪点延迟值。（初始值为 0M 周期。）

在延迟值设置列中选择所需的延迟值。

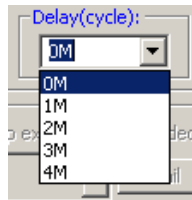


图 5.53 [Trace conditions]（跟踪条件）对话框

5.9.6 设置捕捉 / 不捕捉条件

如果选择的跟踪模式为“Fill until stop”（填充直至停止）、“Fill until full”（填充至满）或“Fill around TP”（围绕 TP 填充），可以指定捕捉 / 不捕捉条件。

可以选择只提取事件指定的跟踪信息的必需部分，或者删除不需要的部分。

- (1) 捕捉 / 不捕捉条件
有以下类型的条件。

表 5.22 捕捉 / 不捕捉条件

	类型	描述	
提取		Between two events (在两个事件之间)	提取的周期从发生开始事件时开始，在最后结束事件发生前结束（不提取结束事件发生的周期）。
		Duration of an event (事件期间)	只提取发生指定事件的周期。
		Duration of an event occurring in a subroutine (子例程中发生的事件期间)	只提取在指定地址范围内（子例程或函数）发生指定事件的周期。
		Instruction accessing specific data (存取特定数据的指令)	检测到存取特定数据的指令。
删除		Between two events (在两个事件之间)	提取的周期从发生开始事件时开始，在倒数第二个结束事件发生时结束（不提取结束事件发生的周期）。
		Duration of an event (事件期间)	只删除发生指定事件的周期。
		Duration of an event occurring in a subroutine (子例程中发生的事件期间)	只删除在指定地址范围内（子例程或函数）发生指定事件的周期。

从 [Trace conditions]（跟踪条件）对话框的 [Record condition]（记录条件）分组框中选择 [Extract]（提取）或 [Delete]（删除）时，将显示一个列表框，请从中选择要设置的条件。

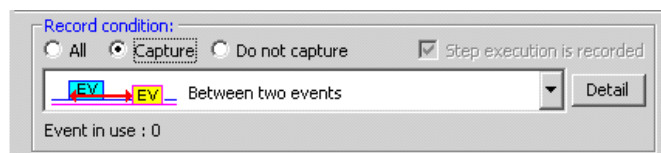


图 5.54 [Record condition]（记录条件）分组框

然后单击 [Detail]（详细信息）按钮。此时将显示一个页，可在其中设置事件。

- 注意：
1. 在指定提取或删除条件时，从 [Trace]（跟踪）窗口的 [Display Mode]（显示模式）中不能选择 [DIS]（反汇编显示）和 [SRC]（源显示）。
 2. 在使用提取或删除的数据存取事件时，请确保为存取类型指定 MCU 总线。

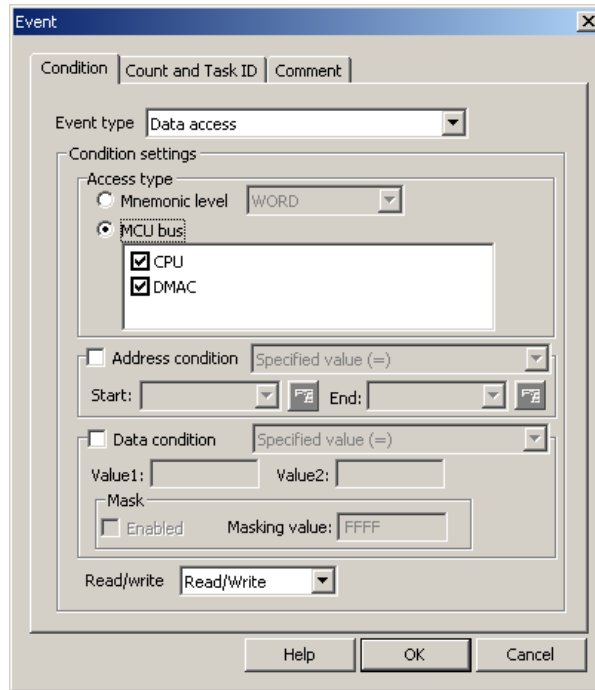


图 5.55 [Event]（事件）对话框

5.9.7 选择跟踪获取的内容

选择要捕捉到跟踪存储器中的跟踪信息内容。使用 [Trace conditions]（跟踪条件）对话框的 [Options]（选项）页进行此设置。

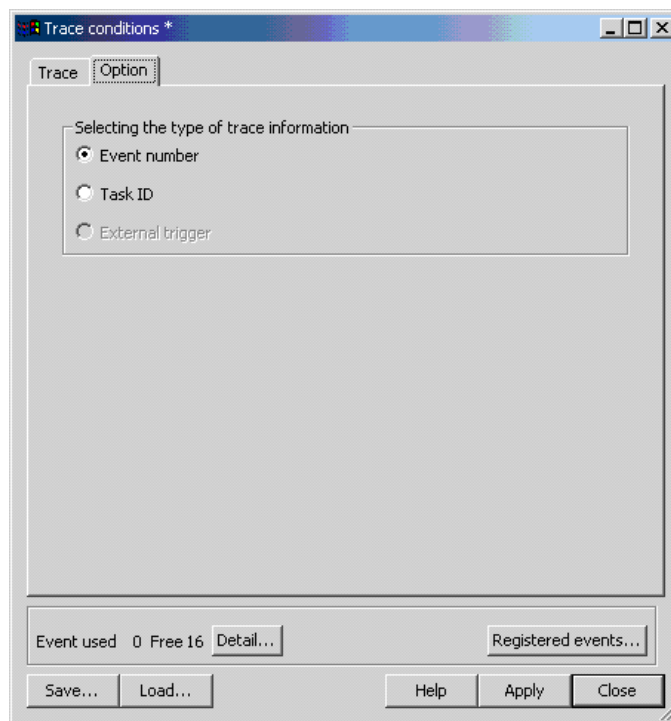


图 5.56 [Trace conditions]（跟踪条件）对话框

从三个可用选项中选择要获取的信号：[Event Number]（事件编号）、[Task ID]（任务 ID）或 [External trigger]（外部触发器）。默认情况下，选择 [Event Number]（事件编号）。

注意： 在通过运行实时操作系统程序来执行的跟踪获取中，如果要显示跟踪执行的历史记录，请确保选择 [Task ID]（任务 ID）。

5.9.8 显示跟踪结果

要检查跟踪结果，请查看 [Trace]（跟踪）窗口。跟踪结果可在以下一种显示模式下显示。
使用 [Trace]（跟踪）窗口的上下文菜单中的 [Display Modes]（显示模式）可以切换这些显示模式。
有五种跟踪结果显示模式：总线显示、反汇编显示、源显示和混合显示。

(1) 总线显示模式

在上下文菜单中，选择 [Display Modes]（显示模式）[BUS]（总线）。显示跟踪的每个周期的总线信息。（默认显示模式）

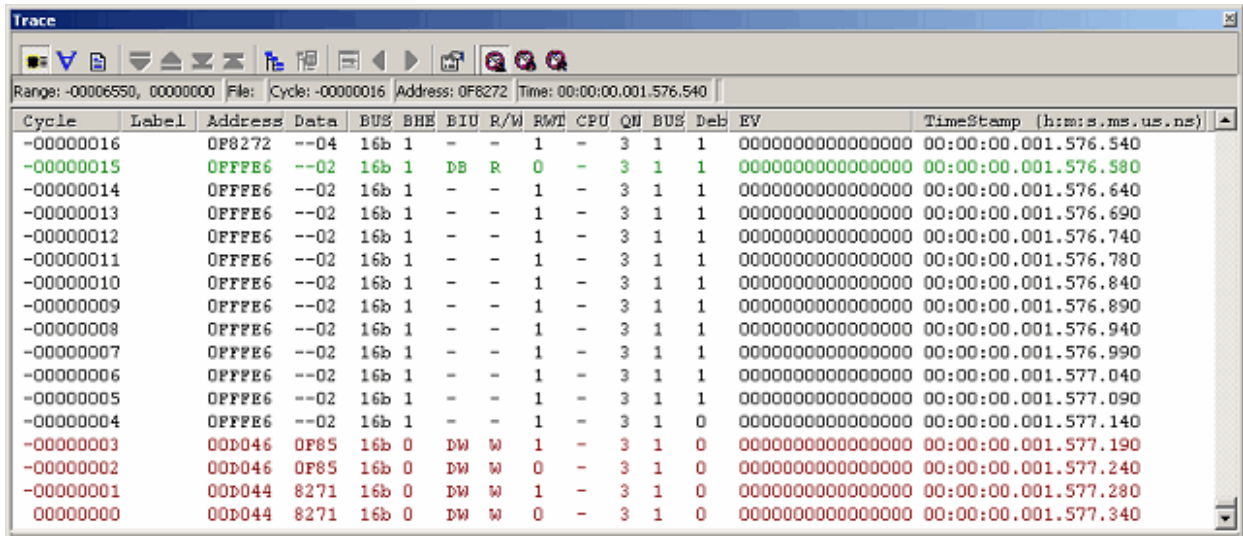


图 5.57 [Trace]（跟踪）窗口

(2) 反汇编显示模式

从上下文菜单中选择 [Display Modes]（显示模式）[DIS]。在此显示模式下，可以检查执行的机器语言指令。

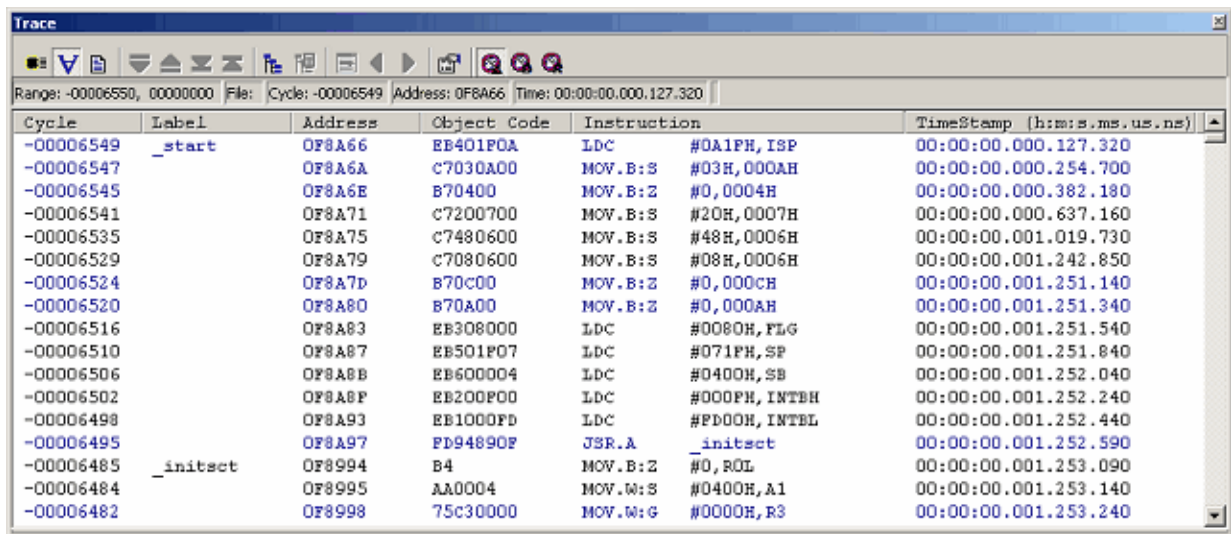


图 5.58 [Trace]（跟踪）窗口

(3) 源显示模式

从上下文菜单中选择 [Display Modes]（显示模式）[SRC]。该显示模式用于检查源程序的执行路径。通过从当前跟踪周期向前或向后逐步执行跟踪数据内的源可以验证执行路径。

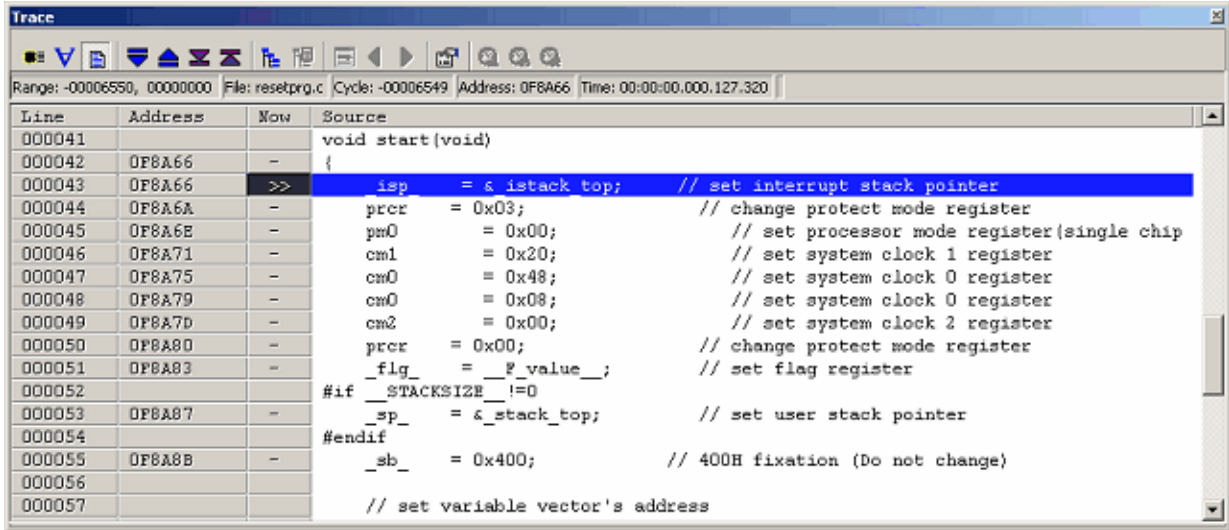


图 5.59 源显示屏幕

(4) 混合显示模式

该显示模式提供总线、反汇编或源显示的混合显示。

从上下文菜单中选择 [Display Modes]（显示模式）[BUS] 之后，请选择 [Display Modes]（显示模式）[DIS]。这样，即可生成总线和反汇编的混合显示。

同样的方式，可以生成总线和源、反汇编和源或总线、反汇编和源这三种混合显示。

要在查看总线和反汇编混合显示后恢复为只显示总线，请重新从上下文菜单中选择 [Display Modes]（显示模式）[DIS]。

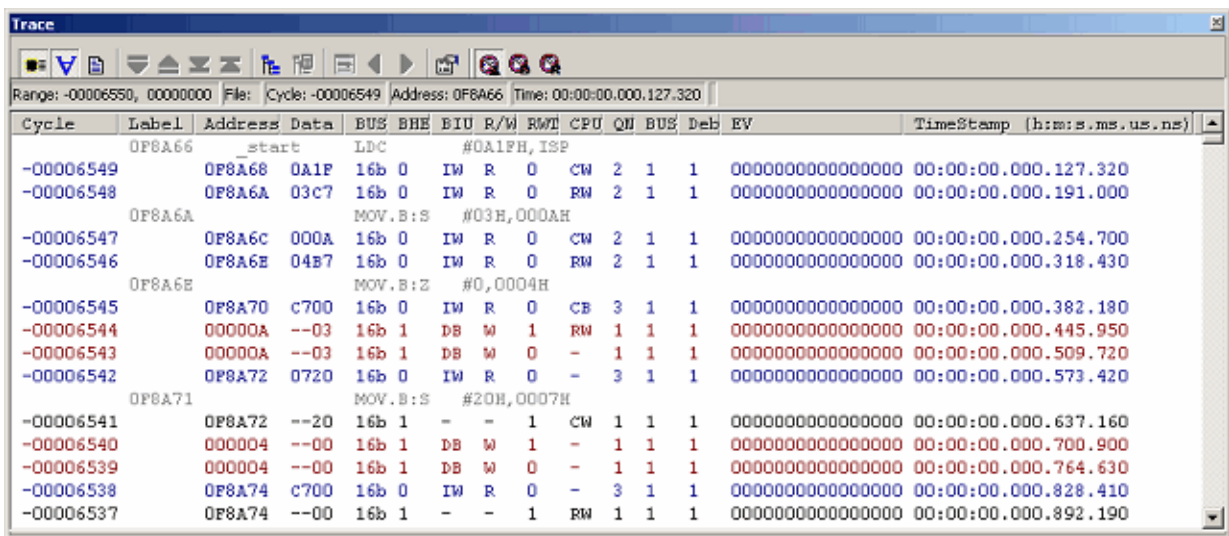


图 5.60 [Trace]（跟踪）窗口

5.9.9 过滤跟踪信息

使用过滤器功能从获取的跟踪信息中仅提取必要的记录。过滤器功能在软件中过滤硬件获取的跟踪信息。此功能允许为获取的跟踪信息随时更改过滤器设置，这与“捕捉 / 不捕捉条件”功能不同（该功能需要在获取跟踪信息前设置获取条件）。

因此，可以轻松地提取必要的信息，显著加快数据分析。

过滤器功能不会影响跟踪存储器，因此其中的内容保持不变。

如果所选跟踪模式是“Fill until stop”（填充直至停止）、“Fill until full”（填充至满）或“Fill around TP”（围绕 TP 填充），并且选中的显示模式是总线或反汇编模式，则可以使用过滤器。

(1) 自动过滤功能

要使用过滤功能，请从 [Trace]（跟踪）窗口的上下文菜单中选择 [Auto Filter]（自动过滤）。打开 [Auto Filter]（自动过滤）后，[Trace]（跟踪）窗口的每一列都用一个自动过滤箭头 [▼] 标识。单击任一箭头 [▼] 从出现的下拉列表中选择必要的条件。这样，就可以轻松地过滤记录从而得到满足条件的记录了。在下拉列表中选择 [Option]（选项）会弹出 [Option]（选项）对话框。在此对话框中，可以设置详细信息条件。

[Address]（地址）和 [Data]（数据）等一些列没有其他固有项，只在下拉列表提供 [Option]（选项）。选择 [All]（全部）将返回未过滤状态。

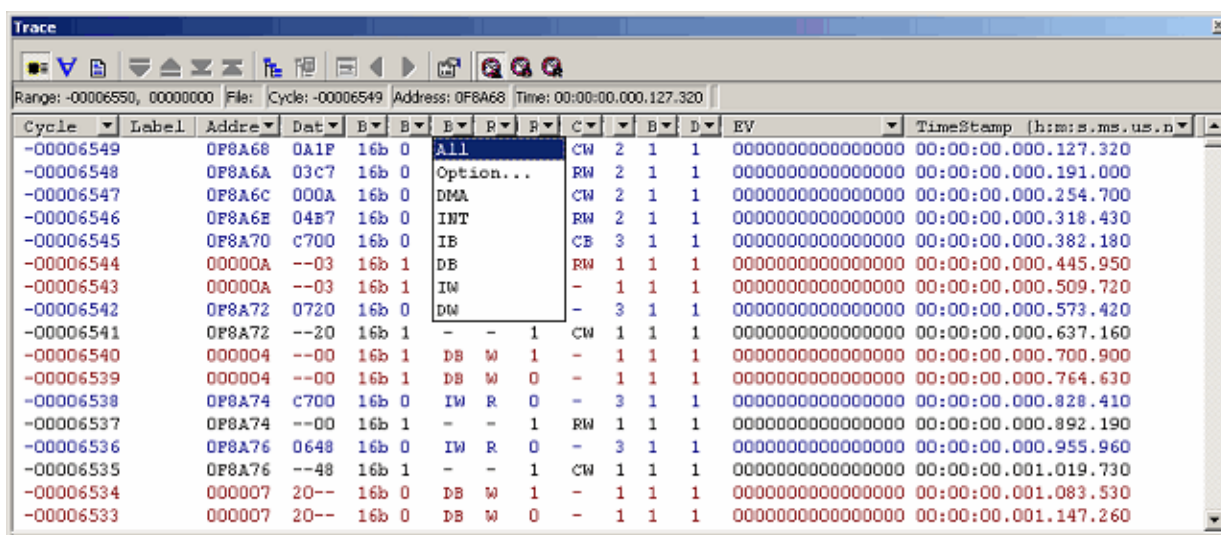


图 5.61 [Trace]（跟踪）窗口

如果在总线显示模式下过滤记录后，切换到仅反汇编显示或仅源显示模式，将取消选择 [Auto Filter]（自动过滤）。同样，如果在反汇编显示模式下过滤记录后，切换到仅总线显示或仅源显示模式，将取消选择 [Auto Filter]（自动过滤）。

如果在 [Option]（选项）对话框中可以指定多个项，则这些项可以作为用于过滤的 OR 条件。

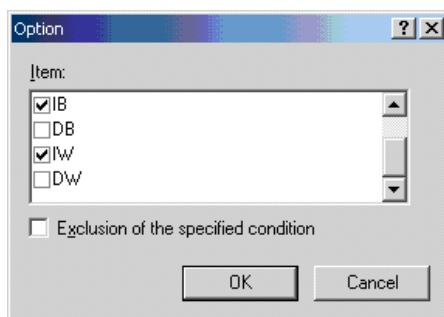



图 5.62 [Option]（选项）对话框

5.9.10 搜索跟踪记录

可以为特定跟踪记录搜索获取的跟踪信息。

要搜索跟踪记录，请使用 [Find]（查找）对话框。要打开该对话框，请从 [Trace]（跟踪）窗口的上下文菜单中选择 [Find]（查找）[Find]（查找）或者单击工具栏上的 [Find]（查找）按钮 。

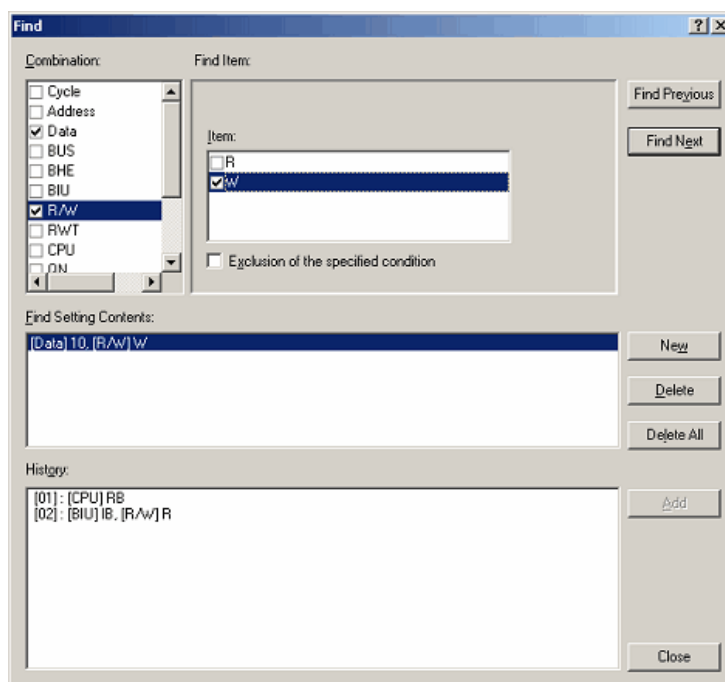


图 5.63 [Find]（查找）对话框

在 [Combination]（组合）列中选择要搜索的条件并选中对应的复选框。

在 [Find Item]（查找项）列中，可以选择与所选条件对应的项。

如果在组合列中选定了多个条件，则为每个条件都设置项。设置的项作为多个 AND 条件搜索。

用户设置的项显示在 [Find Setting Contents]（查找设置内容）中。

设置搜索条件后，单击 [Find Previous]（查找上一个）或 [Find Next]（查找下一个）按钮开始搜索。从用户在 [Trace]（跟踪）窗口中单击的行（此行用蓝色突出显示）正向搜索或反向搜索跟踪记录。

通过搜索找到匹配的跟踪记录后，将突出显示 [Trace]（跟踪）窗口中相应的行。如果找不到匹配的跟踪记录，则显示一个消息对话框。

成功找到跟踪记录的实例后，请从上下文菜单中选择 [Find Previous]（查找上一个）或 [Find Next]（查找下一个）。此时将搜索跟踪记录的下一实例。

(1) 搜索历史记录

曾搜索过的条件作为 [History]（历史记录）列中的历史记录，而高性能嵌入式工作区（High-performance Embedded Workshop，简称 HEW）保持活动状态。

下次执行搜索时，从该历史记录中选择要搜索的行并单击 [Add]（添加）按钮。这样，即可再次使用该条件搜索跟踪信息。

搜索历史记录最多可包含 10 条上次执行搜索的历史记录。

(2) OR 搜索

可以将两个或多个搜索条件作为 OR 条件来执行搜索。

要设置 OR 条件，应先设置第一个条件（在 [search content setting]（搜索内容设置）列的第一行显示）再单击 [New]（新建）按钮。


然后输入第二个条件。此时，第二个条件添加到 [search content setting]（搜索内容设置）列的第二行。

在这种情况下，[search content setting]（搜索内容设置）列中的第一行和第二行的条件可以用作所执行搜索的 OR 条件。

最多可以设置 16 个条件（16 行）。

注意：在 [search content setting]（搜索内容设置）列中的同一行设置的条件构成 AND 条件。

5.9.11 将跟踪信息保存到文件

要将跟踪信息保存到文件，请从上下文菜单中选择 [File]（文件）[Save]（保存）或者单击工具栏上的 [Save]（保存）按钮 。[Trace]（跟踪）窗口中显示的跟踪信息以二进制或文本格式保存。

(1) 保存为二进制格式

要将跟踪信息保存为二进制格式，请先从上下文菜单中选择 [File]（文件）[Save]（保存），在所显示的对话框中的 [Save As Type]（保存类型）列表框中选择 [Trace Data File:Memory Image (*.rtt)]（跟踪数据文件：存储器图像 (*.rtt)）。


保存为二进制格式后，也保存了所有周期。该文件类型可以加载到 [Trace]（跟踪）窗口。

(2) 保存为文本格式

要将跟踪信息保存为文本格式，请先从上下文菜单中选择 [File]（文件）[Save]（保存），在所显示的对话框中的 [Save As Type]（保存类型）列表框中选择 [Text Files:Save Only (*.txt)]（文本文件：仅保存 (*.txt)）。

保存为文本格式后，可以指定要保存的周期范围。该文件类型只能保存而无法加载到 [Trace]（跟踪）窗口。


5.9.12 从文件加载跟踪信息

要从文件加载跟踪信息，请从上下文菜单中选择 [File]（文件）[Load]（加载）或者单击工具栏上的 [Load]（加载）按钮 。指定一个保存为二进制格式的跟踪信息文件。将盖写当前跟踪结果。

加载保存为二进制格式的文件之前，请切换到保存跟踪信息时的跟踪模式。从 [Trace]（跟踪）窗口的上下文菜单中选择 [Acquisition]（获取），在所显示的 [Trace conditions]（跟踪条件）对话框中执行切换操作。


如果当前跟踪模式与保存跟踪信息时的模式不同，会导致错误。保存为文本格式的跟踪信息文件无法加载到 [Trace]（跟踪）窗口。

5.9.13 临时停止跟踪信息获取

要在用户程序执行期间临时停止获取跟踪信息，请从 [Trace]（跟踪）窗口的上下文菜单中选择 [Trace]（跟踪）[Stop]（停止）或者单击工具栏上的 [Stop]（停止）按钮 。

此时将中断跟踪获取，同时更新跟踪显示。如果只想停止获取跟踪信息和在不停止程序执行的情况下检查跟踪信息，可以使用此功能。


5.9.14 重新启动跟踪信息获取

如果在用户程序执行期间临时停止获取跟踪信息之后要重新启动获取跟踪信息，请从 [Trace]（跟踪）窗口的上下文菜单中选择 [Trace]（跟踪）[Restart]（重新启动）或者单击工具栏上的 [Restart]（重新启动）按钮 。


5.9.15 切换时间戳显示

在 [Trace]（跟踪）窗口中显示的时间戳可以切换到 [Absolute time]（绝对时间）、[Differential time]（差异时间）或 [Relative time]（相对时间）。在初始状态下，时间戳以绝对时间显示。

(1) [Absolute time]（绝对时间）

从上下文菜单中选择 [Time]（时间）[Absolute Time]（绝对时间）或者单击工具栏上的 [Absolute Time]（绝对时间）按钮 。启动程序执行后，时间戳将显示为绝对时间。


(2) [Differential time]（差异时间）

从上下文菜单中选择 [Time]（时间）[Differences]（差异时间）或者单击工具栏上的 [Differences]（差异时间）按钮 。时间戳将显示为与前一个周期的差异时间。


(3) [Relative time]（相对时间）

从上下文菜单中选择 [Time]（时间）[Relative Time]（相对时间）或者单击工具栏上的 [Relative Time]（相对时间）按钮 。时间戳将显示为与指定周期的相对时间。

5.9.16 显示函数执行历史记录

要从获取的跟踪信息中显示函数执行的历史记录，请从上下文菜单中选择 [Function Execution History]（函数执行历史记录）[Function Execution History]（函数执行历史记录）或者单击工具栏上的 [Function Execution History]（函数执行历史记录）按钮 。

此时将显示窗口的上窗格。（最初，该窗口是空白的。）

从上下文菜单中选择 [Analyze Execution History]（分析执行历史记录）或者单击工具栏上的 [Analyze Execution History]（分析执行历史记录）按钮  后，仿真器将从跟踪结果的结尾开始分析执行历史记录并在树状结构中显示结果。

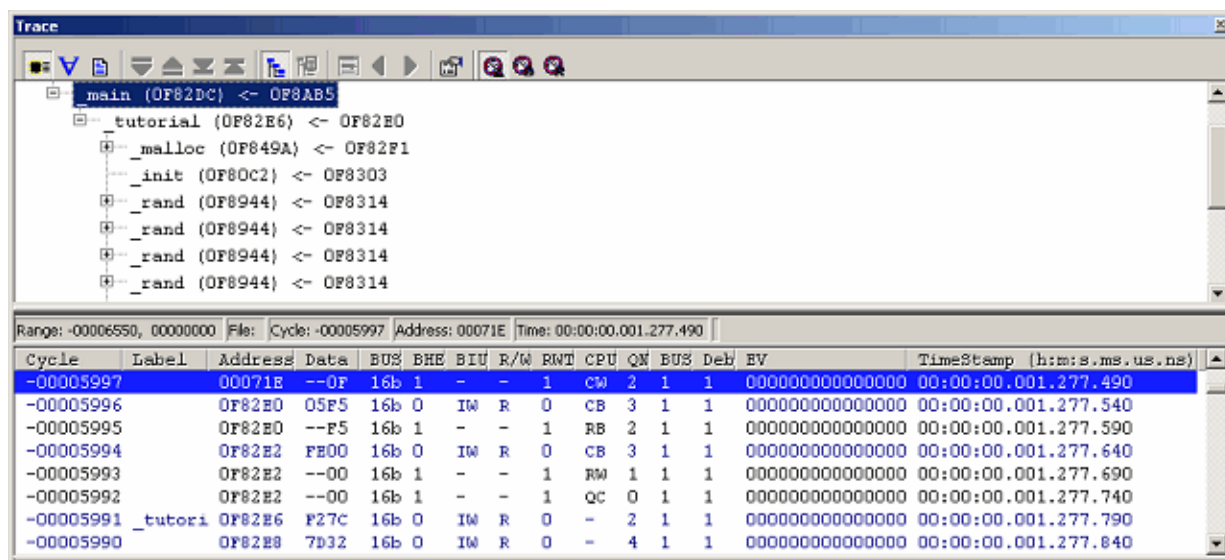


图 5.64 [Trace]（跟踪）窗口

窗口的下窗格显示跟踪结果，这些结果以调用上窗格所选函数的周期开头。


窗口的下窗格可以显示反汇编模式、源模式或混合模式的跟踪结果。

- 注意：
1. 如果指定了跟踪提取或者删除条件，则无法显示函数执行历史记录。
 2. 如果指定了重复（空闲）或者重复（满载）模式，则无法显示函数执行历史记录。


5.9.17 显示任务执行历史记录

只在调试实时操作系统程序时才显示任务执行历史记录。

此外，要显示任务执行的历史记录，需先从 [Trace]（跟踪）窗口的上下文菜单中选择 [Acquisition]（获取），在此时显示的 [Trace conditions]（跟踪条件）对话框的 [Options]（选项）页中选择 [Task ID]（任务 ID）。

要从获取的跟踪信息中显示函数执行的历史记录，请从上下文菜单中选择 [Show Function Execution History]（显示函数执行历史记录）或者单击工具栏上的 [Show Function Execution History]（显示函数执行历史记录）按钮 。

此时将显示窗口的上窗格。（最初，该窗口是空白的。）

在上窗格中右键单击或者单击工具栏上的 [Analyze Execution History]（分析执行历史记录）按钮 ，在此时显示的上下文菜单中选择 [Analyze Execution History]（分析执行历史记录）后，仿真器将显示任务执行的历史记录。

显示任务执行的历史记录时，请注意从任务内部调用的函数不在树状结构中显示。只显示执行函数的顺序。

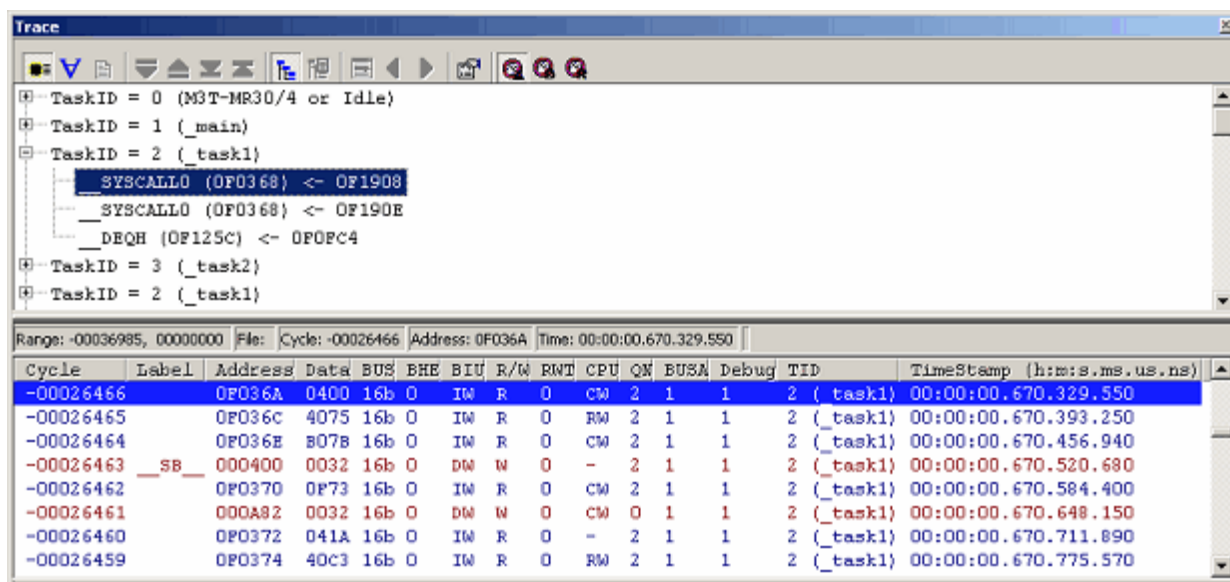


图 5.65 [Trace]（跟踪）窗口

窗口的下窗格显示跟踪结果，该结果以调用上窗格所选任务的周期开头。

窗口的下窗格可以显示反汇编模式、源模式或混合模式的跟踪结果。

- 注意：
1. 注意如果指定了跟踪提取或者删除条件，则无法显示任务执行历史记录。
 2. 如果指定了重复（空闲）或者重复（满载）模式，则无法显示任务执行历史记录。

5.10 测量性能

5.10.1 测量性能


性能函数测量用户程序的每个指定段（最多 8 个段）中的最长、最短、平均和总计执行时间以及通过计数，然后用百分比和图形以数字方式显示相对于总数执行时间的的时间比率 (Go-Break)。

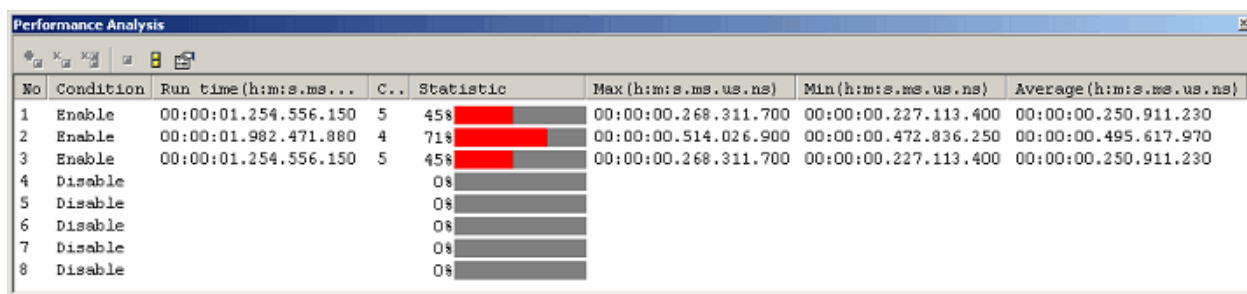
由于性能函数使用仿真器的性能测量电路来测量执行时间，所以不会妨碍用户程序的执行。

在程序执行期间不能处理性能测量条件。

5.10.2 显示性能测量的结果

测量结果显示在 [Performance Analysis]（性能分析）窗口中。

要打开 [Performance Analysis]（性能分析）窗口，请从 [View]（视图）菜单中选择 [Performance]（性能）[Performance Analysis]（性能分析），或者单击 [Performance Analysis]（性能分析）工具栏按钮 。



No	Condition	Run time (h:m:s.ms...)	C.	Statistic	Max (h:m:s.ms.us.ns)	Min (h:m:s.ms.us.ns)	Average (h:m:s.ms.us.ns)
1	Enable	00:00:01.254.556.150	5	45%	00:00:00.268.311.700	00:00:00.227.113.400	00:00:00.250.911.230
2	Enable	00:00:01.982.471.880	4	71%	00:00:00.514.026.900	00:00:00.472.836.250	00:00:00.495.617.970
3	Enable	00:00:01.254.556.150	5	45%	00:00:00.268.311.700	00:00:00.227.113.400	00:00:00.250.911.230
4	Disable			0%			
5	Disable			0%			
6	Disable			0%			
7	Disable			0%			
8	Disable			0%			

图 5.66 [Performance Analysis]（性能分析）窗口

[Performance Analysis]（性能分析）窗口用百分比和图形以数字的方式显示符合在即将开始执行的程序中所设置条件的执行时间比率。

可以隐藏此窗口中不必要的列。

要隐藏任一列，请在标头列右键单击并从上下文菜单中选择要隐藏的列。

要重新显示任一隐藏的列，请从上下文菜单中重新选择该列。

下面列出了此窗口中显示的内容。

表 5.23 列和内容

列	描述
No (编号)	分配给在 [Performance Analysis Conditions] (性能分析条件) 对话框中设置的 1~8 测量段的数字。 在上下文菜单上单击 [Settings] (设置) 打开 [Performance Analysis Conditions] (性能分析条件) 对话框。
Condition (条件)	在 [Performance Analysis Conditions] (性能分析条件) 对话框中设置测量条件时表示为 [Enable] (允许)。 否则, 表示为 [Disable] (禁用)。
Run time (h:m:s.ms.us.ns) (运行时间 (h:m:s.ms.us.ns))	累计执行时间。显示测量的累计执行时间。
Count (计数)	显示测量的次数。
Statistic (统计信息)	显示累计执行时间相对于 Go-Break 执行时间的比率。 [比率计算公式] (累计执行时间 / Go-Break 累计执行时间) * 100
Max (h:m:s.ms.us.ns) (最长 (h:m:s.ms.us.ns))	每次执行测量的最长执行时间
Min (h:m:s.ms.us.ns) (最短 (h:m:s.ms.us.ns))	每次执行测量的最短执行时间
Average (h:m:s.ms.us.ns) (平均 (h:m:s.ms.us.ns))	每次执行测量的平均执行时间

5.10.3 设置性能测量条件

在 [Performance]（性能）窗口中，选择要在其中设置条件的段编号行，并从上下文菜单中选择 [Set]（设置）。将显示 [Performance Analysis Conditions]（性能分析条件）对话框。

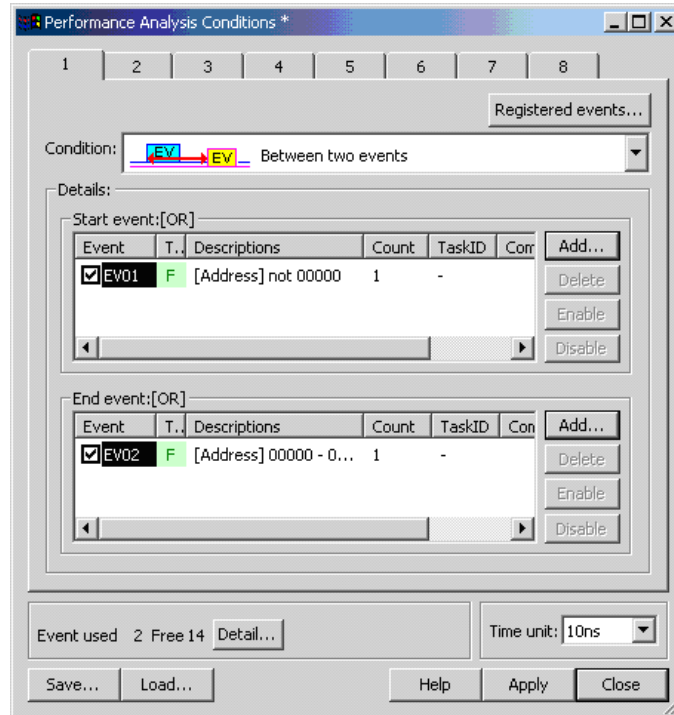


图 5.67 [Performance Analysis Conditions]（性能分析条件）对话框

(1) 设置测量条件

可以从以下四种模式中选择测量条件。为一个段选择一种测量条件。使用事件设置段。事件计数固定为 1。即使将事件计数设置为 1 以外的数字，也将其视为 1。

表 5.24 测量条件模式 (1)


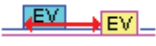
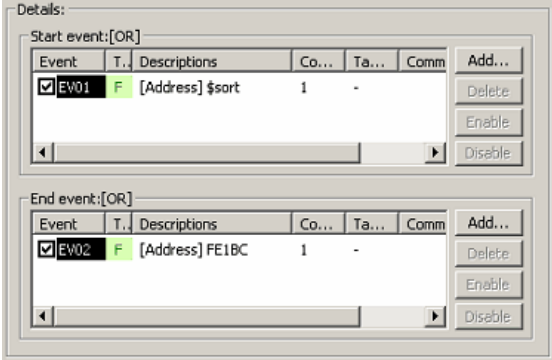

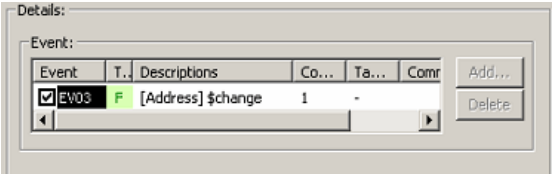

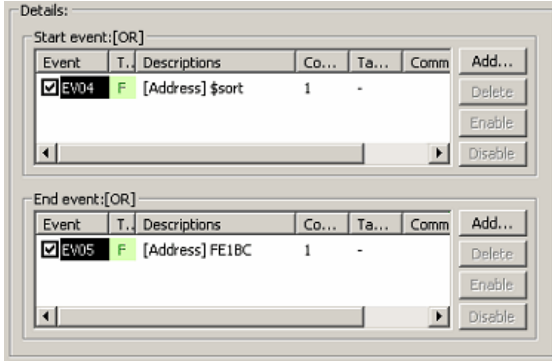
	<p>[Disabled] (禁用) 不测量。</p>
	<p>[Between two events] (在两个事件之间)</p> <div style="text-align: center;">  </div> <p style="text-align: center;">图 5.68 在两个事件之间</p> <p>从开始事件发生到结束事件发生时进行测量。 具体地说，在开始事件和结束事件设置的范围内测量执行时间和执行计数。时间测量从开始事件发生时开始，到结束事件发生时中止。每次开始事件和结束事件在设置范围内成对发生时，执行计数都会增加 1。 [Start event:] (开始事件:) 可以设置一个或多个事件。 [End event:] (结束事件:) 可以设置一个或多个事件。</p>
	<p>[Event cycle counting] (事件循环计数)</p> <div style="text-align: center;">  </div> <p style="text-align: center;">图 5.69 事件循环计数</p> <p>在事件发生期间进行测量。 即对事件发生周期和执行计数进行测量。将一个事件发生到下一个事件发生经过的时间作为一个测量实例进行测量。每次事件发生时，执行计数都增加 1。 [Event:] (事件:) 只能设置一个事件点。</p>

表 5.24 测量条件模式 (2)

	<p>[Interrupt-disabled range between two events] (两个事件之间的禁止中断范围)</p> <div style="text-align: center;">  </div> <p style="text-align: center;">图 5.70 两个事件之间的禁止中断范围</p> <p>从开始事件发生到结束事件发生时，对禁止中断段进行测量。 具体地说，在开始事件和结束事件设置的范围内，测量禁止中断时间和禁止中断计数。在禁止中断的同时开始测量时间，在重新启用中断的同时中止测量时间。每次禁止中断时，计数都增加 1。 [Start event:] (开始事件：) 可以设置一个或多个事件。 [End event:] (结束事件：) 可以设置一个或多个事件。</p>
---	---

- 注意： 1. 要测量函数的执行时间（函数的最长、最短或平均执行时间），请使用 [Between two events]（在两个事件之间）。
2. 对函数的起始地址设置一个取操作作为开始事件，对函数的出口（在此写入返回语句）设置一个取操作作为结束事件。如果有多个出口，则为每个出口设置取条件作为结束事件。

(2) 选择测量单位

该设置同样适用于全部 8 个段。可以从以下选项中选择测量单位：

[10 ns]（10 纳秒）、[20 ns]（20 纳秒）（初始值）、[40 ns]（40 纳秒）、[80 ns]（80 纳秒），[160 ns]（160 纳秒）和 [1.6 μs]（1.6 微秒）

最长测量时间因设置的测量单位而异。

5.10.4 开始性能测量

运行用户程序后，根据设置的性能测量条件会自动开始性能测量。

暂停用户程序后，测量结果显示在 [Performance Analysis]（性能分析）窗口中。

如果用户程序暂停后不改变测量条件重新运行，这种情况下，测量时间会添加到以前的测量值中。

要重新执行测量，请在运行程序前清除测量结果。

5.10.5 清除性能测量条件

在 [Performance Analysis]（性能分析）窗口中选择要清除的测量条件，然后从上下文菜单中选择 [Set]（设置）显示 [Performance Analysis Conditions]（性能分析条件）对话框。在 [Performance Analysis Conditions]（性能分析条件）对话框中，禁用要清除的条件。



图 5.71 [Performance Analysis Conditions]（性能分析条件）对话框

5.10.6 清除性能测量结果

在 [Performance Analysis]（性能分析）窗口中选择要清除的段，然后从上下文菜单中选择 [Clear Data]（清除数据）。将清除所选段的测量结果。要清除所有测量结果，请从上下文菜单中选择 [Clear All Data]（清除所有数据）。

5.10.7 关于性能的最长测量时间

(1) 最长测量时间

用于性能测量的定时器由一个 40 位计数器组成。

最长测量时间因选择的测量单位而异。

要选择测量单位，请使用 [Performance Analysis Conditions]（性能分析条件）对话框中的 [Measurement Unit]（测量单位）列表框。

下表列出了可测量的最长时间。

表 5.25 可测量的最长时间

编号	解析度	可测量的最长时间
1	10ns	约 3 小时 03 分钟 15 秒
2	20ns	约 6 小时 06 分钟 30 秒
3	40ns	约 12 小时 13 分钟 00 秒
4	80ns	约 24 小时 26 分钟 00 秒
5	160ns	约 48 小时 52 分钟 01 秒
6	1.6 μ s	约 488 小时 40 分钟 18 秒

注意： 请注意性能测量会导致 ± 1 的解析度误差（如果解析度 = 20 ns、 ± 20 ns）。

(2) 最大测量计数

使用 32 位计数器测量执行计数。最多可以对 4,294,967,295 的计数执行测量。

5.11 测量代码覆盖

5.11.1 测量代码覆盖

代码覆盖是一种功能，用于指示测试的“领悟”程度，例如，“对软件代码需要测试到什么程度（通过）。”

指令执行信息在 C/C++ 和汇编器级别显示。

此功能从程序收集指令执行信息，而不会引起程序中断。因此，不会影响用户系统的实时功能。

覆盖结果在中断时更新。

E100 仿真器支持 C0：指令覆盖率和 C1：分支覆盖率。


表 5.26 代码覆盖定义

C0: 指令覆盖率	代码内的所有语句至少执行一次。
C1: 分支覆盖率	代码内的所有分支至少执行一次。

E100 仿真器在使用 C0 + C1 级别覆盖时最大提供 2 MB 代码覆盖存储器，使用 C1 级别覆盖时最大提供 1 MB 代码覆盖存储器。

使用初始设置时，会按此顺序为 ROM 和 RAM 区域的地址自动分配代码覆盖存储器。

5.11.2 打开 [Code Coverage]（代码覆盖）窗口

从 [View]（视图）菜单中选择 [Code]（代码）[Code Coverage]（代码覆盖）或者单击 [Code Coverage]（代码覆盖）工具栏按钮 。

[Code Coverage]（代码覆盖）窗口最初显示时为空白状态。

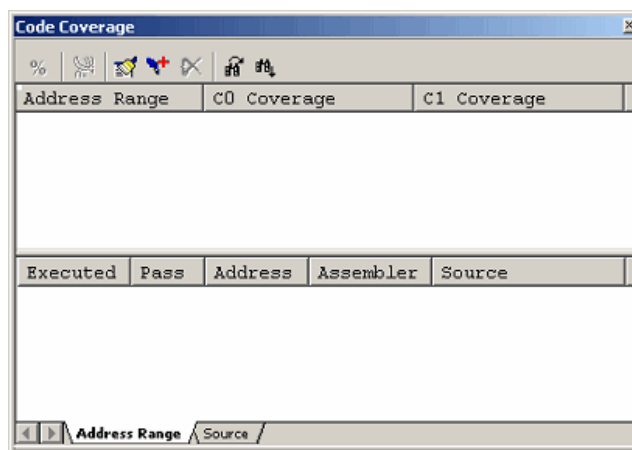


图 5.72 [Code Coverage]（代码覆盖）窗口

(1) 测量方法

[Code Coverage]（代码覆盖）窗口包括两页。

表 5.27 [Code Coverage]（代码覆盖）窗口包括的页

页名称	描述
[Address Range]（地址范围）页	对所有地址范围执行测量。
[Source]（源）页	对指定的源文件执行测量。

各页允许注册多个范围。

最多可同时打开 [Code Coverage]（代码覆盖）窗口的两个实例。

5.11.3 分配代码覆盖存储器（硬件资源）

(1) 存储器分配

必须将代码覆盖存储器分配给要对其进行测量的地址后，才能测量代码覆盖。只能从分配了存储器的地址范围获取覆盖数据。

要分配代码覆盖存储器，请使用 [Coverage Memory Allocation]（覆盖存储器分配）对话框。

要打开该对话框，请从 [Code Coverage]（代码覆盖）窗口的上下文菜单中选择 [Hardware Settings]（硬件设置）。

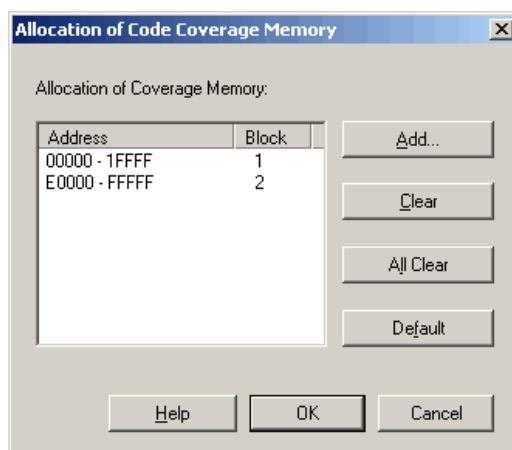


图 5.73 [Allocation of Code Coverage Memory]（分配代码覆盖存储器）对话框

如果使用 C0 级别覆盖和 C1 级别覆盖，可以将每个以 256 KB 范围开头的 1~8 块（最大 2 MB）中任意一块以及每个以 128 KB 范围开头的 1~8 块（最小 1 MB）中任意一块指定为代码覆盖测量区域。

可以设置连续块、非连续块或一个块。

初始设置时，会自动为 ROM 和 RAM 区域中的地址分配覆盖存储器。

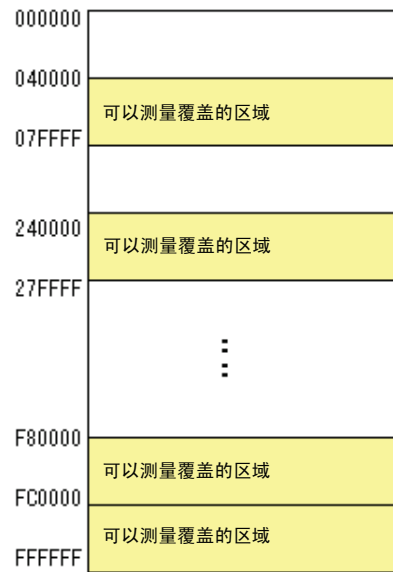


图 5.74 覆盖存储器分配示意图

(2) 更改存储器分配

如果覆盖存储器分配发生更改，则覆盖数据将从覆盖存储器检索到仅覆盖缓冲器中，此数据可从更改前的地址获取。

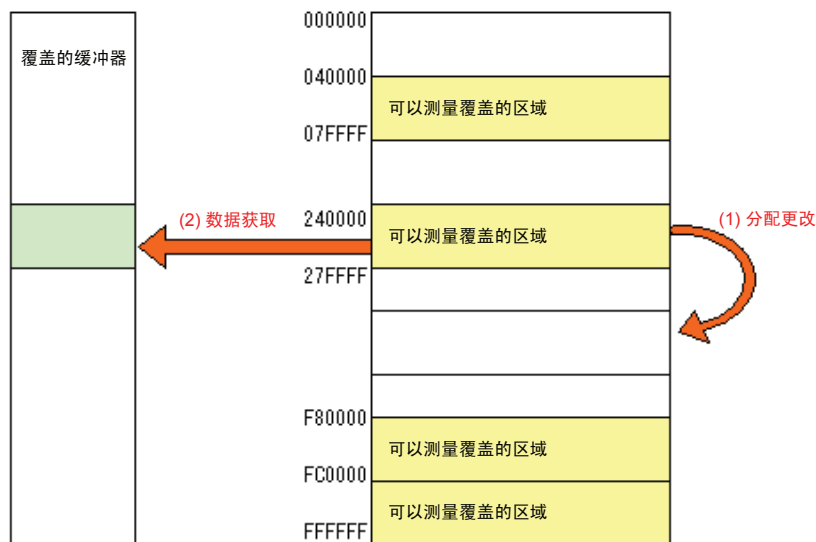


图 5.75 覆盖存储器分配更改示意图

数据将一直在仅覆盖缓冲器中累积，直到用户将其清除。但是，对于未分配覆盖存储器的区域将不更新数据。

[Code Coverage]（代码覆盖）窗口中显示的覆盖信息包含仅覆盖缓冲器的内容。

5.11.4 测量地址范围

[Address Range] (地址范围) 页显示仿真器从用户指定的地址范围收集的代码覆盖信息 (C0 覆盖和 C1 覆盖)。

可以注册多个地址范围。

可以指定超过 2 MB 的地址范围, 甚至可以指定未分配覆盖存储器的区域。但是, 对于未分配覆盖存储器的区域将不更新数据。

不更新数据的区域显示为灰色。

如下例所示。

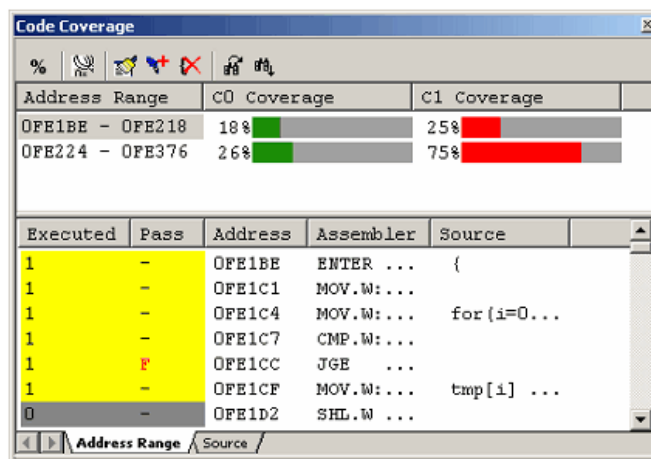


图 5.76 [Code Coverage] (代码覆盖) 窗口 (指定地址)

一条分割线将 [Code Coverage] (代码覆盖) 窗口垂直平均分为两部分。

上面的区域显示要测量的地址范围、C0 覆盖和 C1 覆盖。

表 5.28 [Code Coverage] (代码覆盖) 窗口上面区域中显示的内容

Address Range (地址范围)	测量其中覆盖的地址范围
C0 Coverage (C0 覆盖)	按百分比和图形显示 C0 覆盖
C1 Coverage (C1 覆盖)	按百分比和图形显示 C1 覆盖

下面区域显示上面区域中选定的地址范围的详细信息。(汇编器级别)。

表 5.29 [Code Coverage]（代码覆盖）窗口下面区域中显示的内容

Executed（已执行）	1: 执行的指令 0: 未执行的指令
Pass（通过）	显示条件分支指令的执行条件 T: 条件符合且程序已分支 F: 条件不符合且程序未分支 T/F: 条件符合和条件不符合
Address（地址）	指令地址
Assembler（汇编器）	反汇编显示
Source（源）	C/C++ 或汇编器源

获取的覆盖信息将一直在存储器中累积，直到用户将其清除。

如果双击 [Address Range]（地址范围）页中显示的 [Assembler]（汇编器）代码，将在 [Editor]（编辑器）窗口中显示相应的源代码。

请确保下面列出的情况下不显示源代码。

- 汇编器行对应的源文件不存在。
- 汇编器行对应的源行不存在。
- 不包括调试信息的区域，此区域中的汇编器行是一个库。

5.11.5 添加地址范围

按照下述步骤添加地址范围。

1. 从 [Code Coverage]（代码覆盖）窗口的 [Address Range]（地址范围）页执行以下操作：
 - A. 在 [Address Range]（地址范围）页的上面区域右键单击并从上下文菜单中选择 [Add Measurement Ranges]（添加测量范围）。

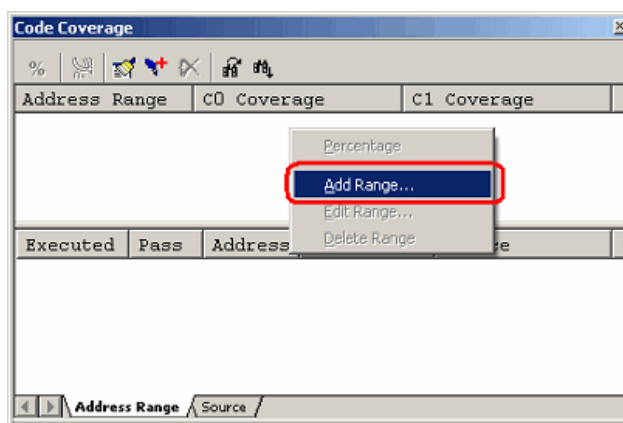


图 5.77 [Code Coverage]（代码覆盖）窗口

- B. 在显示的 [Add Address Range]（添加地址范围）对话框中，输入地址范围。

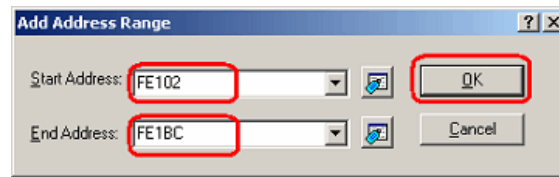


图 5.78 [Add Address Range] (添加地址范围) 对话框

C. 添加的地址范围将显示在 [Code Coverage] (代码覆盖) 窗口的上面区域。

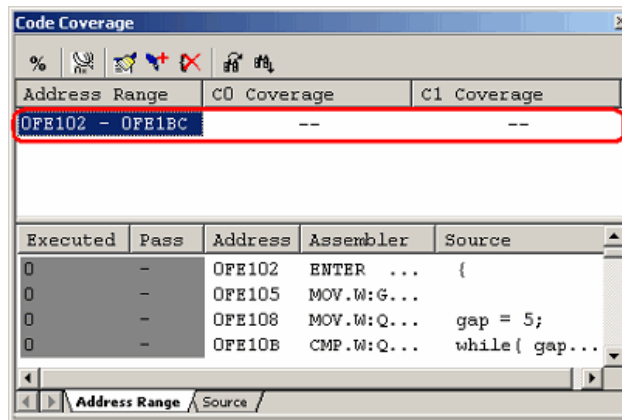


图 5.79 [Code Coverage] (代码覆盖) 窗口

5.11.6 更改地址范围

按照下述步骤更改地址范围。

1. 从 [Code Coverage]（代码覆盖）窗口的 [Address Range]（地址范围）页执行以下操作：
 - A. 在 [Address Range]（地址范围）页中选择要更改的地址范围，保持其处于选中状态，并从上下文菜单选择 [Edit Range]（编辑范围）。

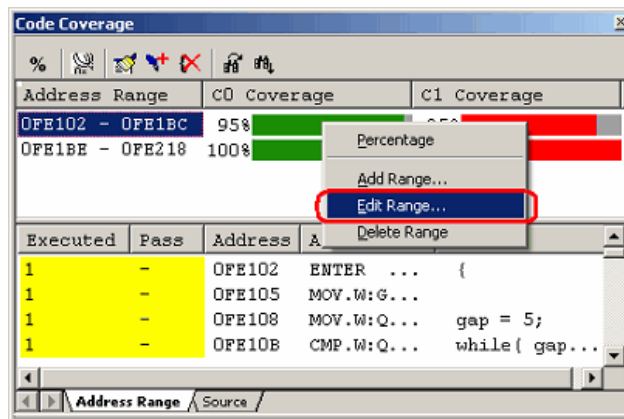


图 5.80 [Code Coverage]（代码覆盖）窗口

- B. 在显示的 [Edit Address Range]（编辑地址范围）对话框中，更改地址范围。

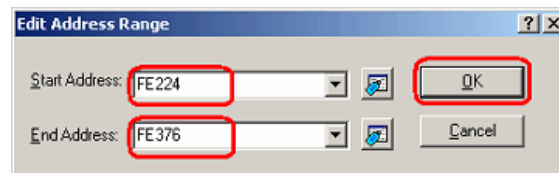


图 5.81 [Edit Address Range]（编辑地址范围）对话框

- C. 更改后的地址范围将显示在 [Code Coverage]（代码覆盖）窗口的上面区域。

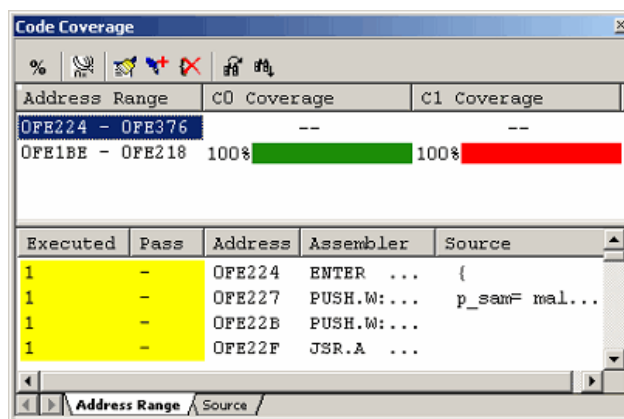


图 5.82 [Code Coverage]（代码覆盖）窗口

5.11.7 移除地址范围

按照下述步骤移除地址范围。

1. 从 [Code Coverage]（代码覆盖）窗口的 [Address Range]（地址范围）页执行以下操作：
 - A. 在 [Address Range]（地址范围）页中选择要移除的地址范围，保持其处于选中状态，并从上下文菜单选择 [Delete Range]（删除范围）。

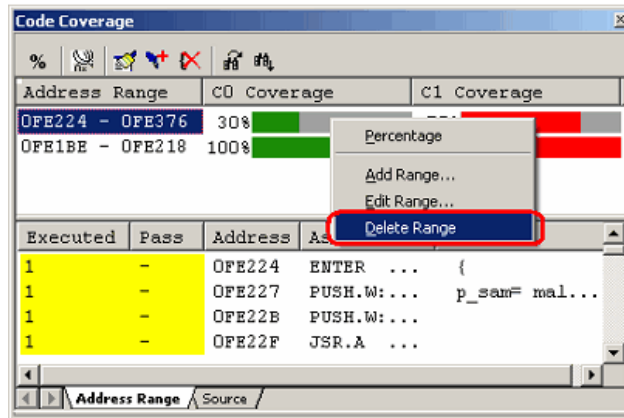


图 5.83 [Code Coverage]（代码覆盖）窗口

- B. 此时将显示一个对话框，要求用户进行确认。
选择保存或不保存覆盖数据。要保存数据，请指定文件名，然后单击 [OK]（确定）按钮。如果不保存数据，只需单击 [OK]（确定）按钮。

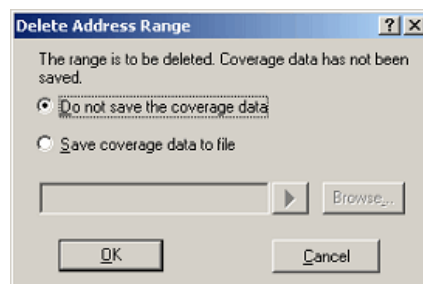


图 5.84 [Delete Address Range]（删除地址范围）对话框

C. 将删除选定的地址范围。

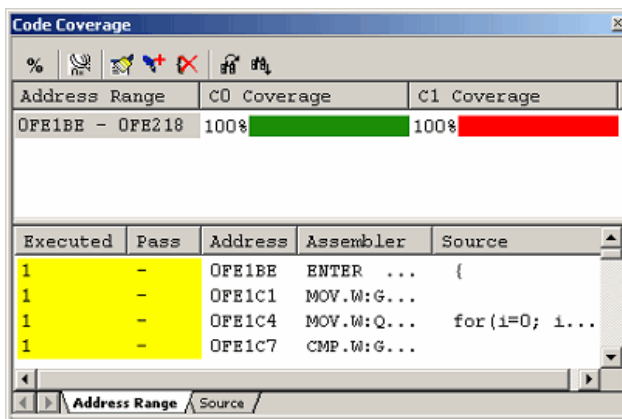


图 5.85 [Code Coverage] (代码覆盖) 窗口

5.11.8 测量源文件

[Source]（源）页显示仿真器从用户指定的源文件收集的代码覆盖信息（C0 覆盖和 C1 覆盖）。

可以注册多个源文件。

可以指定大小超过 2 MB 的源文件，甚至可以指定包括未分配覆盖存储器区域的文件。

但是，对于未分配覆盖存储器的部分将不更新数据。

不更新数据的地址行显示为灰色。

如下例所示。

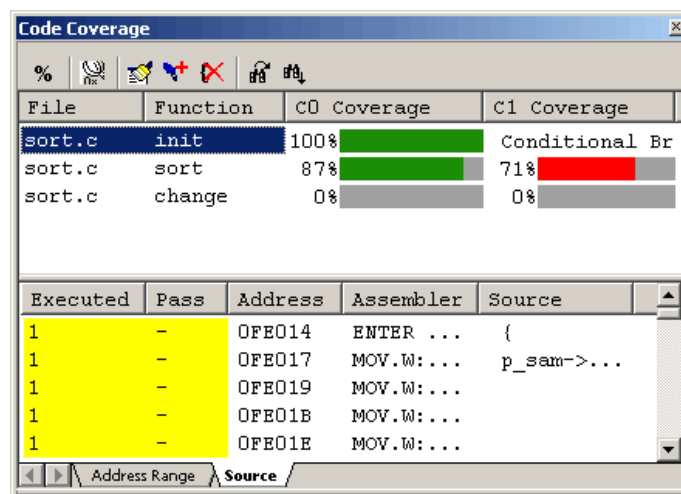


图 5.86 [Code Coverage]（代码覆盖）窗口（指定源文件）

一条分割线将 [Code Coverage]（代码覆盖）窗口垂直平均分为两部分。

上面的区域显示要测量的地址范围（文件和函数名称）、C0 覆盖和 C1 覆盖。

表 5.30 [Code Coverage]（代码覆盖）窗口上面区域中显示的内容

File（文件）	文件名
Function（函数）	函数名称
C0 Coverage（C0 覆盖）	按百分比和图形显示 C0 覆盖
C1 Coverage（C1 覆盖）	按百分比和图形显示 C1 覆盖

下面区域显示上面区域中选定的地址范围的详细信息（汇编器级别）。

表 5.31 [Code Coverage]（代码覆盖）窗口下面区域中显示的内容

Executed（已执行）	1: 执行的指令 0: 未执行的指令
Pass（通过）	显示条件分支指令的执行条件 T: 条件符合且程序已分支 F: 条件不符合且程序未分支 T/F: 条件符合和条件不符合
Address（地址）	指令地址
Assembler（汇编器）	反汇编显示
Source（源）	C/C++ 或汇编器源

获取的覆盖信息将一直在存储器中累积，直到用户将其清除。

5.11.9 添加源文件

按照下述步骤添加源文件。

1. 从 [Code Coverage]（代码覆盖）窗口的 [Source]（源）页执行以下操作：

A. 右键单击 [Source]（源）页的上面区域并从上下文菜单中选择 [Add Range]（添加范围）。

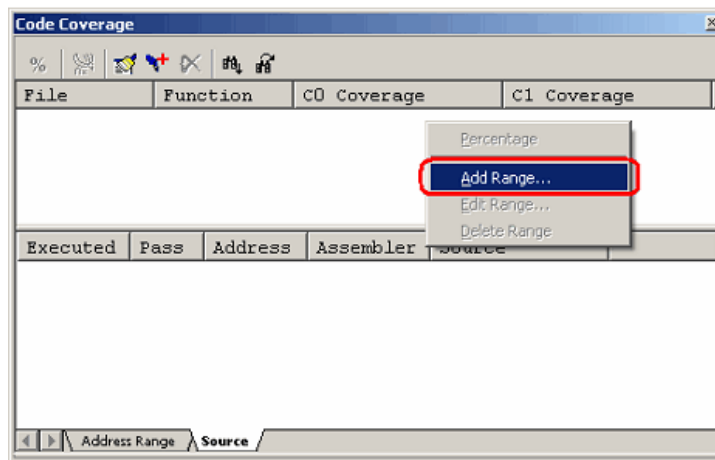


图 5.87 [Code Coverage]（代码覆盖）窗口

B. 在显示的 [Add Source File]（添加源文件）对话框中，输入文件名。

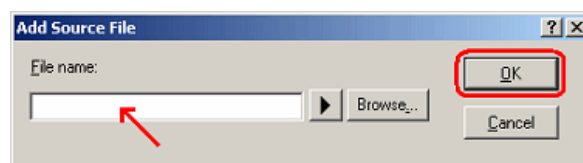


图 5.88 [Add Source File]（添加源文件）对话框

C. 添加的源文件和其中包含的函数名称将显示在 [Code Coverage]（代码覆盖）窗口的上面区域。

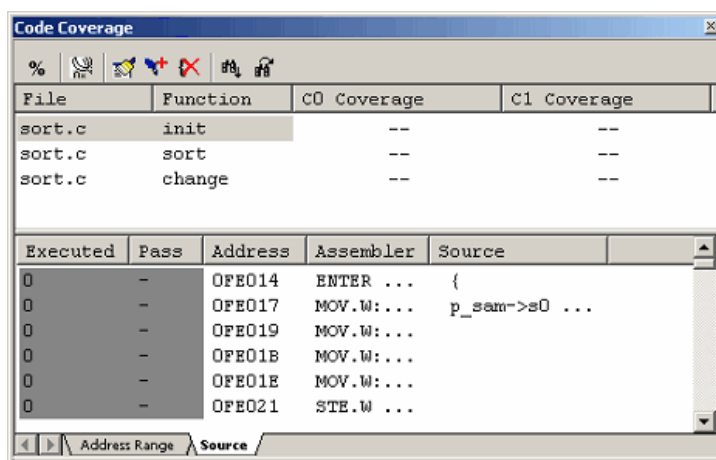


图 5.89 [Code Coverage]（代码覆盖）窗口

5.11.10 移除源文件

通过以下方法删除源文件。

1. 从 [Code Coverage]（代码覆盖）窗口的 [Source]（源）页执行以下操作：
 - A. 在 [Source]（源）页的上面区域选择要移除的函数，保持其处于选中状态，并从上下文菜单选择 [Delete Range]（删除范围）。

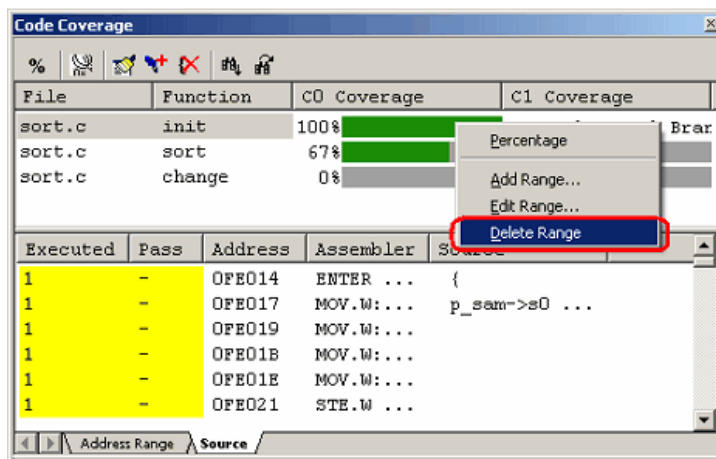


图 5.90 [Code Coverage]（代码覆盖）窗口

- B. 此时将显示一个对话框，提示用户进行确认。选择保存或不保存覆盖数据。要保存数据，请指定文件名，然后单击 [OK]（确定）按钮。如果不保存数据，只需单击 [OK]（确定）按钮。

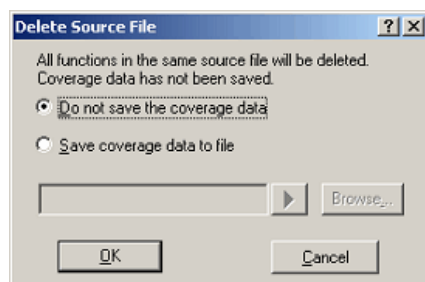


图 5.91 [Delete Source File]（删除源文件）对话框

- C. 将移除选定源文件中的所有函数。

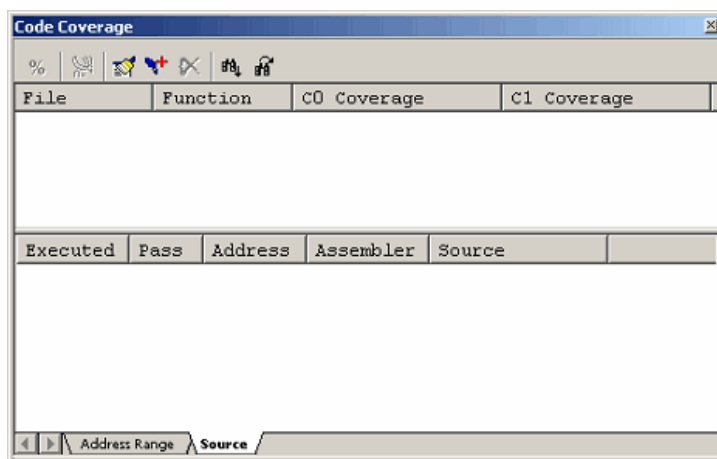


图 5.92 [Code Coverage]（代码覆盖）窗口

5.11.11 显示百分比和图形

程序停止后，在 [Code Coverage]（代码覆盖）窗口的上面区域右键单击并从上下文菜单中选择 [Percentage]（百分比）。仿真器将开始计算每个地址范围的 C0：指令覆盖率和 C1：分支覆盖率。

计算完成后，通过百分比值和图形在上面区域中显示覆盖信息。

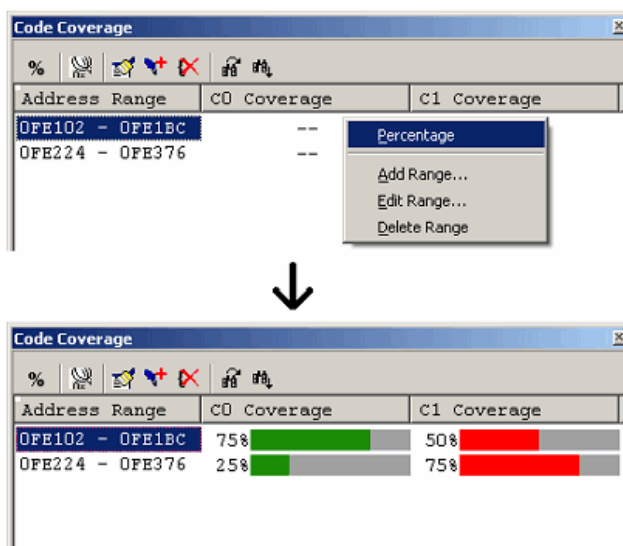


图 5.93 [Code Coverage]（代码覆盖）窗口

5.11.12 使用 Sort 函数

单击 [Code Coverage]（代码覆盖）窗口中上面区域的标题列，可以对覆盖数据进行排序。

(1) 单击 [File]（文件）列

可以按文件名排序数据。每行一个数据，相同的文件按函数名称排序。

示例：

File	Function	C0 Coverage
file1.cpp	func1	40% ■■■■
file1.cpp	func2	10% ■
file1.cpp	func3	80% ■■■■■■■■
file1.cpp	func4	70% ■■■■■■■■
file2.cpp	func1	20% ■■
file2.cpp	func2	60% ■■■■■■
file2.cpp	func3	90% ■■■■■■■■
file3.cpp	func1	0%
file3.cpp	func2	30% ■■■
file3.cpp	func3	10% ■

- (2) 单击 [C0 Coverage] (C0 覆盖) 列

可以按覆盖率排序数据。

首次单击时，数据按递减百分比排序。

再次单击时，数据按递增百分比排序。

示例：

File	Function	C0 Coverage
file2.cpp	func3	90% ■■■■■■■■■■
file1.cpp	func3	80% ■■■■■■■■
file1.cpp	func4	70% ■■■■■■■
file2.cpp	func2	60% ■■■■■■
file1.cpp	func1	40% ■■■■
file3.cpp	func2	30% ■■■
file2.cpp	func1	20% ■■
file1.cpp	func2	10% ■
file3.cpp	func3	10% ■
file3.cpp	func1	0%


- (3) 按顺序单击 [C0 Coverage] (C0 覆盖) 和 [File] (文件) 列

每个文件按递减覆盖分别排序数据。

示例：

File	Function	C0 Coverage
file1.cpp	func3	80% ■■■■■■■■
file1.cpp	func4	70% ■■■■■■■
file1.cpp	func1	40% ■■■■
file1.cpp	func2	10% ■
file2.cpp	func3	90% ■■■■■■■■■■
file2.cpp	func2	60% ■■■■■■
file2.cpp	func1	20% ■■
file3.cpp	func2	30% ■■■
file3.cpp	func3	10% ■
file3.cpp	func1	0%

5.11.13 搜索未执行的行

搜索未执行的行的选定地址范围或函数。单击工具栏中的 [Find]（查找）按钮  后，会显示如下所示的 [Find]（查找）对话框。

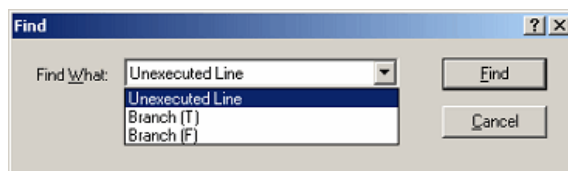


图 5.94 [Find]（查找）对话框

提供了以下三个搜索选项。

表 5.32 搜索选项

[Unexecuted Line] (未执行的行)	还没有执行的指令
[Branch (T)] (分支 (T))	仅 TRUE 方传递的分支指令。
[Branch (F)] (分支 (F))	仅 FALSE 方传递的分支指令。

单击 [Search]（搜索）按钮开始搜索。

如果找到匹配的指令，将突出显示该指令行。

如果找不到匹配的指令，将显示一条消息。

5.11.14 清除代码覆盖信息

(1) 清除代码覆盖信息的指定范围

从上下文菜单中选择 [Clear Coverage Range]（清除覆盖范围），将显示如下所示的 [Clear Address Range]（清除地址范围）对话框。

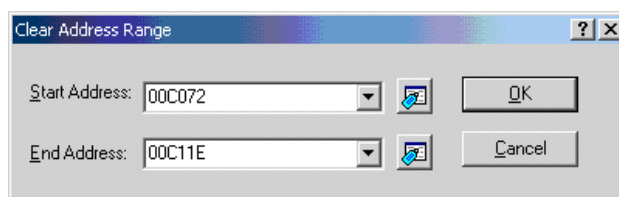


图 5.95 [Clear Address Range]（清除地址范围）对话框

指定要清除范围的起始地址和结束地址。

单击 [OK]（确定）按钮，将清除指定的范围。

(2) 清除所有代码覆盖信息

从上下文菜单中选择 [Clear the Entire Coverage]（清除整个覆盖），将清除所有代码覆盖信息。

5.11.15 更新覆盖信息

将 [Code Coverage]（代码覆盖）窗口的内容更新为最新内容。从 [Code Coverage]（代码覆盖）窗口的上下文菜单中选择 [Refresh]（刷新）。

如果禁止更新覆盖信息，则在程序中断时将不自动更新这些信息。因此，需要手动更新才能查看最新的信息。

5.11.16 禁止信息更新

有时不自动更新 [Code Coverage]（代码覆盖）窗口的内容，例如，用户系统停止运行时。从 [Code Coverage]（代码覆盖）窗口的上下文菜单中选择 [Lock Refresh]（锁定刷新）。

5.11.17 将代码覆盖信息保存到文件

将当前选定页的代码覆盖信息保存到文件。

从 [Code Coverage]（代码覆盖）窗口的上下文菜单中选择 [Save Data]（保存数据），将显示如下所示的 [Save Coverage Data]（保存覆盖数据）对话框。

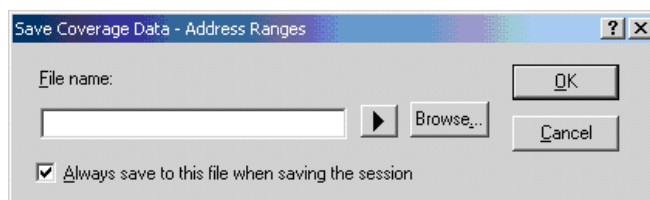


图 5.96 [Save Coverage Data]（保存覆盖数据）对话框

输入要在其中保存信息的文件的名称。

如果省略文件扩展名，将自动附加扩展名 “.cov”。

如果指定现有的文件名，将盖写该文件。

5.11.18 从文件加载代码覆盖信息

加载代码覆盖信息文件。

从 [Code Coverage]（代码覆盖）窗口的上下文菜单中选择 [Load Data]（加载数据），将显示如下所示的 [Load Coverage Data]（加载覆盖数据）对话框。

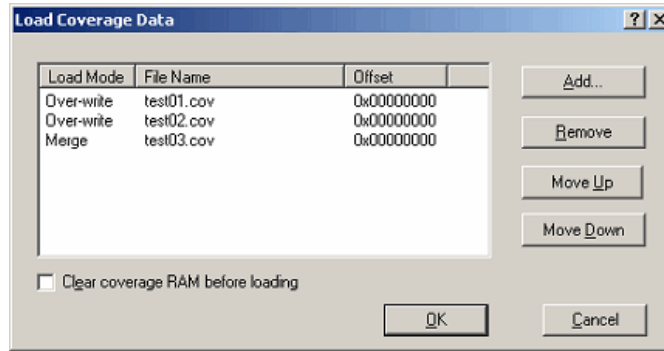


图 5.97 [Load Coverage Data]（加载覆盖数据）对话框

单击 [Add]（添加）按钮，将显示如下所示的 [Add Coverage Files]（添加覆盖文件）对话框。

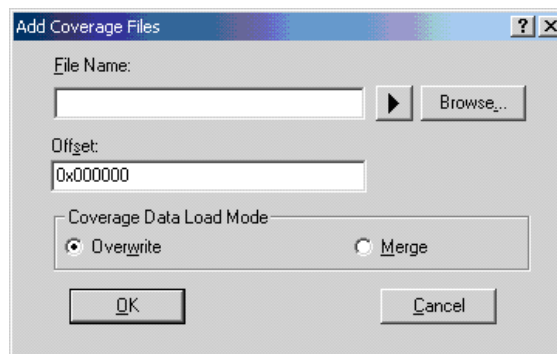


图 5.98 [Add Coverage Files]（添加覆盖文件）对话框

使用此对话框输入要加载的覆盖信息文件。可以为加载的每个文件指定加载模式和偏移量。

只能加载扩展名为 “.cov” 的文件。如果输入任何其他文件扩展名，将输出一条错误消息。

在 [Load Coverage Data]（加载覆盖数据）对话框中会列出所添加的文件。

将以文件列出顺序进行加载。可根据需要使用 [Up]（向上）或 [Down]（向下）按钮来更改顺序。

注意： 如果加载的覆盖信息文件是源文件指定类型，则加载时无法指定偏移量。

5.11.19 关于覆盖信息文件加载模式

覆盖信息文件加载模式示意图显示如下。

(1) 选择 [Overwrite]（盖写）时

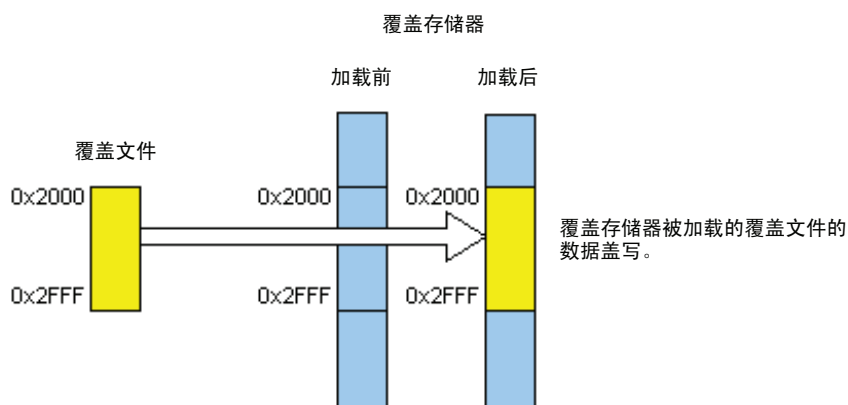


图 5.99 选择 [Overwrite]（盖写）时的加载模式示意图

(2) 选择 [Merge]（合并）时

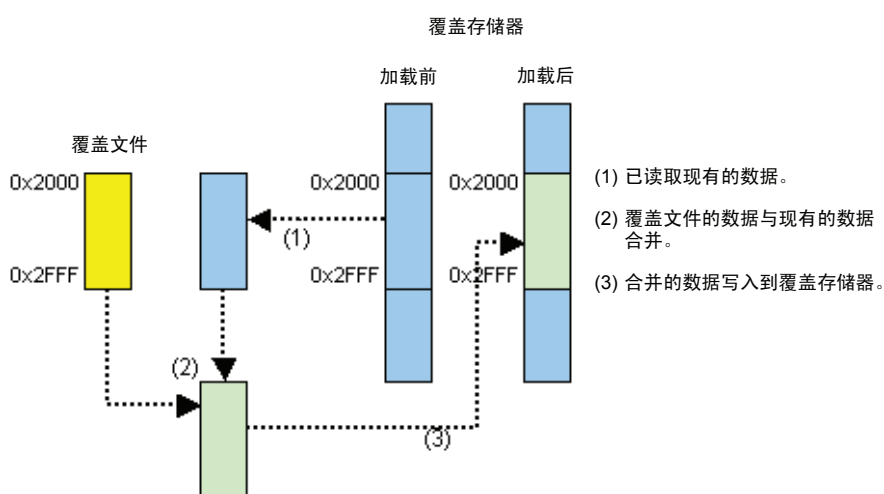


图 5.100 选择 [Merge]（合并）时的加载模式示意图

(3) 合并模式的示例应用程序

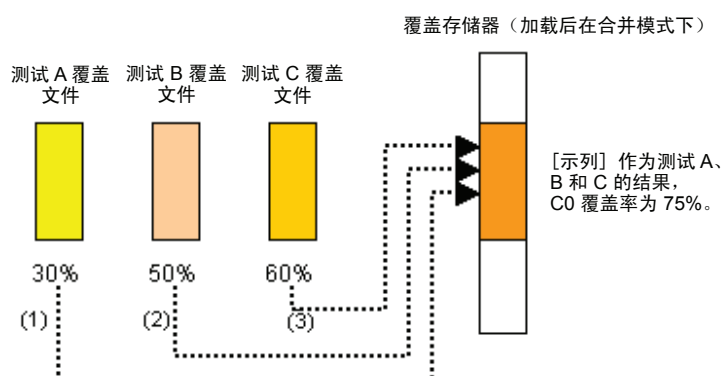


图 5.101 合并模式应用程序的示意图

[步骤]

- (1) 打开 [Load Coverage Data]（加载覆盖数据）对话框。
要开始合并，请选中带有“Clear coverage RAM before loading”（加载前清除覆盖 RAM）的复选框。
- (2) 在合并模式下添加测试 A 的覆盖文件。
- (3) 在合并模式下添加测试 B 的覆盖文件。
- (4) 在合并模式下添加测试 C 的覆盖文件。
- (5) 单击 [OK]（确定）按钮。

执行上述步骤后，就完成了对三个文件的合并。

通过在 [Code Coverage]（代码覆盖）窗口重新计算百分比就可以查看这些测试作为一个整体的覆盖率（以百分比表示）。

此外，还可以将合并的数据保存到一个文件，以便可以将这些数据作为一个文件进行管理。

5.11.20 在 [Editor]（编辑器）窗口中显示代码覆盖结果

在“source mode”（源模式）显示 [Editor]（编辑器）窗口时，覆盖结果显示在其代码覆盖列中。与已执行指令所在源行相对应的位置以黄色突出显示。

如果在 [Code Coverage]（代码覆盖）窗口中更改了与覆盖相关的设置，则相应代码覆盖列的显示也会进行更新。

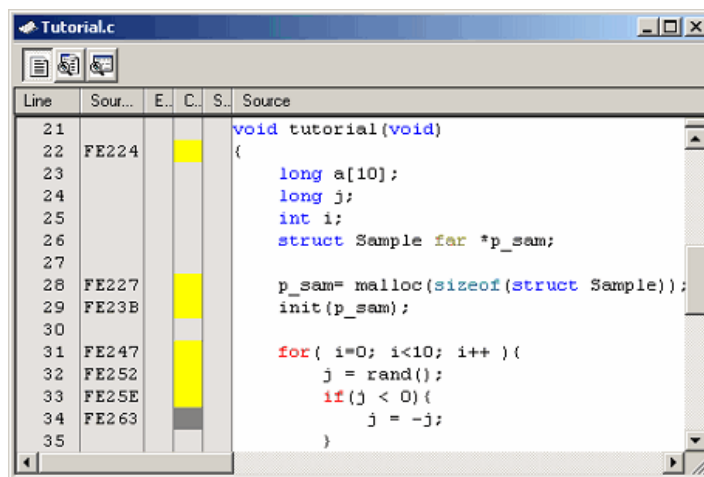


图 5.102 代码覆盖结果示例

5.12 测量数据覆盖

5.12.1 测量数据覆盖

E100 仿真器具有自己的代码覆盖，数据覆盖和实时分析功能，且不能同时使用。

要使用数据覆盖功能，请在 [Configuration Properties]（配置属性）对话框的 [System]（系统）页上的 [Exclusive Functions]（专用功能）部分中选择 [Data Coverage]（数据覆盖）。

数据覆盖是一种功能，用于指示对数据区域“what kinds of accesses have been made”（已进行了哪种存取）。


此功能收集每字节的存取信息，而不会引起程序中断。因此，不会影响用户系统的实时功能。

覆盖结果在中断时更新。

E100 仿真器具有 512 KB 的数据覆盖存储器。

初始设置时，会自动为 ROM 和 RAM 区域的地址按此顺序分配数据覆盖存储器。

5.12.2 打开 [Data Coverage]（数据覆盖）窗口

从 [View]（视图）菜单中选择 [Code]（代码）[Data Coverage]（数据覆盖）或者单击 [Data Coverage]（数据覆盖）工具栏按钮 。

[Data Coverage]（数据覆盖）窗口最初显示为空白状态。

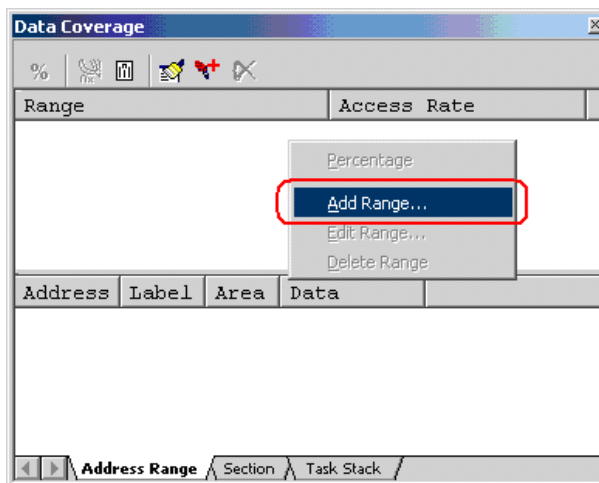


图 5.103 [Data Coverage]（数据覆盖）窗口

(1) 测量方法

[Data Coverage]（数据覆盖）窗口包括以下三个页。

表 5.33 [Data Coverage]（数据覆盖）窗口中的页

页名称	描述
[Address Range] (地址范围) 页	对所有地址范围执行测量。
[Section] (段) 页	对指定的段执行测量。
[Task Stack] (任务堆栈) 页	对所有任务堆栈区进行测量。

各页允许注册多个范围。

[Task Stack]（任务堆栈）页仅支持自动注册。

最多可同时打开 [Data Coverage]（数据覆盖）窗口的三个实例。

5.12.3 分配数据覆盖存储器（硬件资源）

(1) 存储器分配

必须将数据覆盖存储器分配给要对其进行测量的地址后，才能测量数据覆盖。只能从分配了存储器的地址范围获取覆盖数据。

要分配数据覆盖存储器，请使用 [Allocation of Data Coverage Memory]（数据覆盖存储器分配）对话框。要打开该对话框，请从 [Data Coverage]（数据覆盖）窗口的上下文菜单中选择 [Hardware Settings]（硬件设置）。

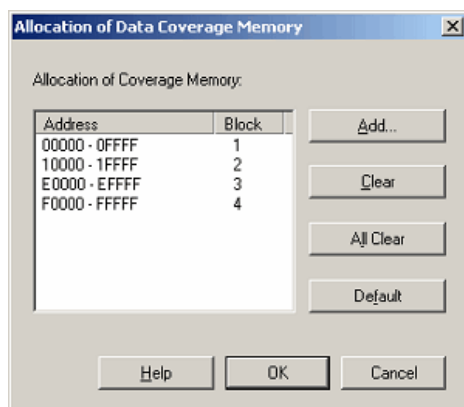


图 5.104 [Allocation of Data Coverage Memory]（数据覆盖存储器分配）对话框

仿真器允许将每个以 64 KB 范围开头的 1~8 块（最大 512 KB）中的任意一块指定为数据覆盖测量区域。

可以设置连续块、非连续块或一个块。

初始设置时，会自动为 ROM 和 RAM 区域中的地址分配覆盖存储器。

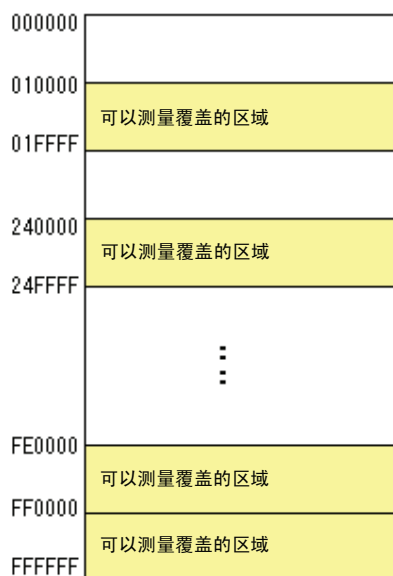


图 5.105 数据覆盖存储器分配示意图

(2) 更改存储器分配

如果覆盖存储器分配发生改变，则会将从更改前的地址获取的覆盖数据从覆盖存储器检索到仅覆盖缓冲器中。

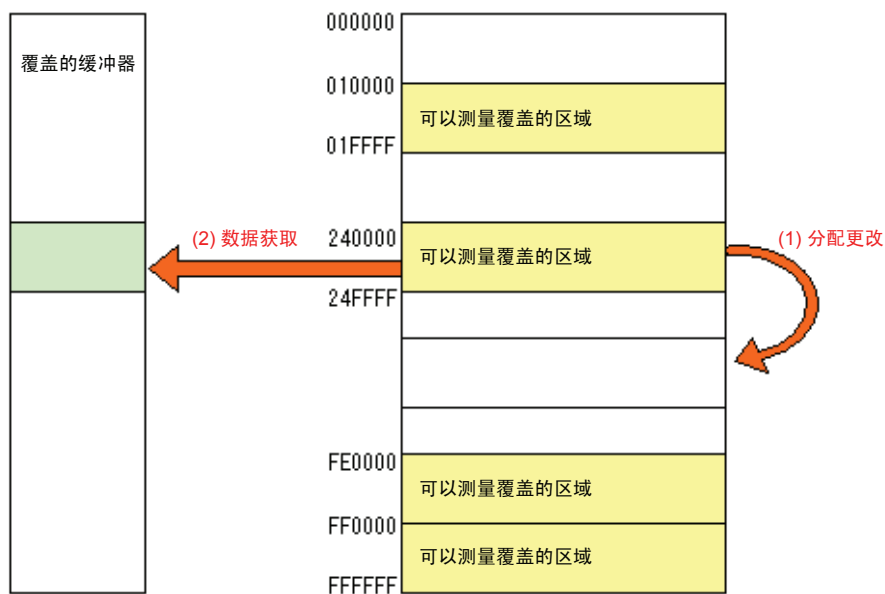


图 5.106 数据覆盖存储器分配更改示意图

数据将一直在仅覆盖缓冲器中累积，直到用户将其清除。

但是，对于未分配覆盖存储器的区域将不更新数据。

[Data Coverage]（数据覆盖）窗口中显示的覆盖信息包含仅覆盖缓冲器的内容。

5.12.4 测量地址范围

E100 仿真器显示从用户指定的地址范围收集的存取信息。

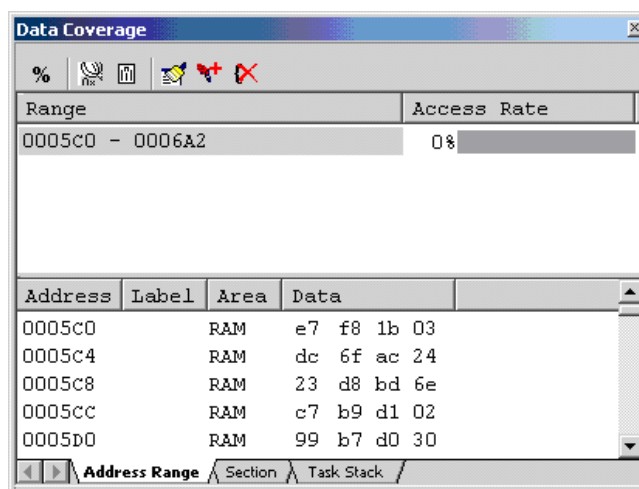


图 5.107 [Data Coverage]（数据覆盖）窗口（指定地址）

一条分割线将 [Data Coverage]（数据覆盖）窗口垂直平均分为两部分。

上面的区域显示要测量的地址范围和存取率。

表 5.34 [Data Coverage]（数据覆盖）窗口上面区域中的内容

[Range]（范围）	测量其中覆盖的地址范围
[Access Rate]（存取率）	按百分比和图形显示存取率

下面区域显示上面区域中选定的地址范围的详细信息。

表 5.35 [Data Coverage]（数据覆盖）窗口下面区域中的内容

[Address]（地址）	地址值
[Label]（标签）	标签名称
[Area]（区）	存储区（FlashROM、RAM、SFR） 对于未使用的区此列为空。
[Data]（数据）	存储器数据 存取的数据的背景显示为紫色。

如果地址位于覆盖存储器分配的区域外部，则地址行以灰色显示。尽管保留了这些地址的现有覆盖信息，但是并不更新程序执行所做的覆盖信息。

获取的覆盖信息将一直在存储器中累积，直到用户将其清除。

5.12.5 添加地址范围

按照下述步骤添加地址范围。

1. 从 [Data Coverage]（数据覆盖）窗口的 [Address Range]（地址范围）页执行以下操作：
 - A. 右键单击 [Address Range]（地址范围）页的上面区域并从上下文菜单中选择 [Add Range]（添加范围）。

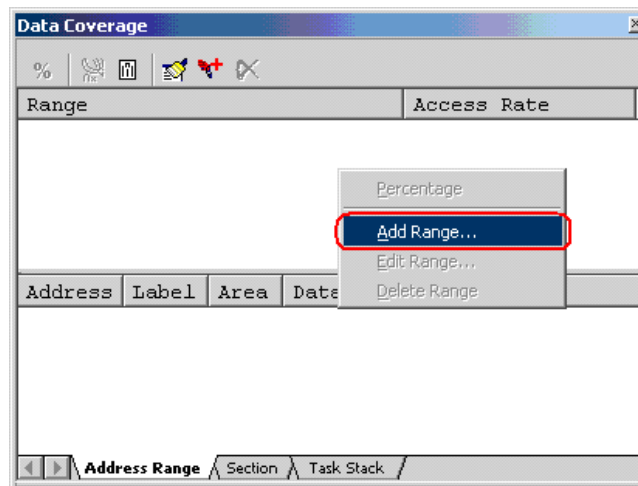


图 5.108 [Data Coverage]（数据覆盖）窗口

- B. 在显示的 [Add Address Range]（添加地址范围）对话框中，输入地址范围。

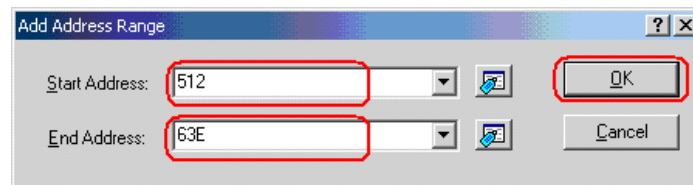


图 5.109 [Add Address Range]（添加地址范围）对话框

C. 添加的地址范围将显示在 [Data Coverage]（数据覆盖）窗口的上面区域。

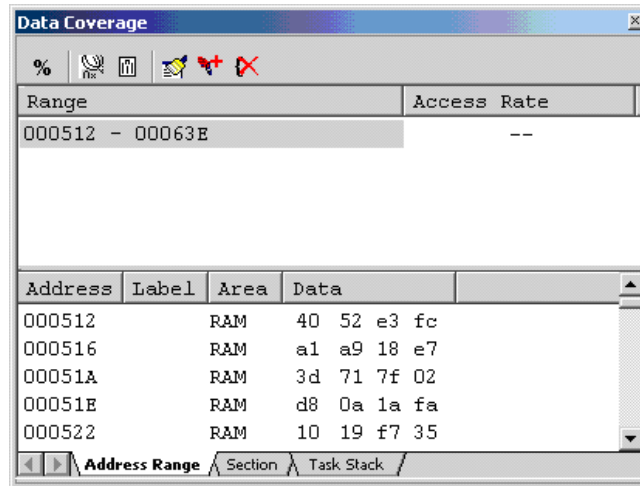


图 5.110 [Data Coverage]（数据覆盖）窗口

5.12.6 更改地址范围

按照下述步骤更改地址范围。

1. 从 [Data Coverage]（数据覆盖）窗口的 [Address Range]（地址范围）页执行以下操作：
 - A. 在 [Address Range]（地址范围）页中选择要更改的地址范围，保持其处于选中状态，并从上下文菜单选择 [Edit Range]（编辑范围）。

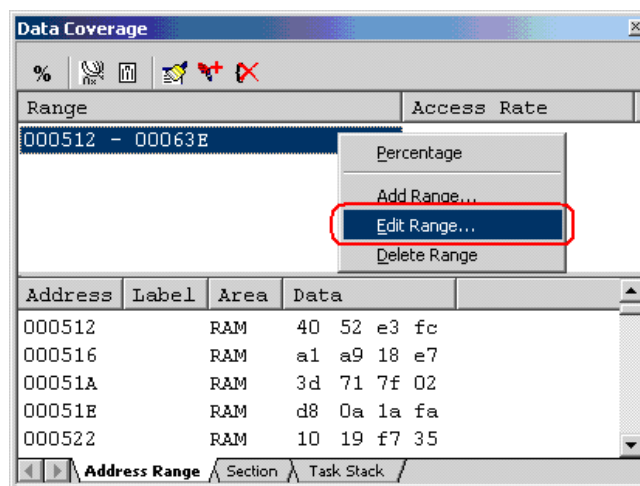


图 5.111 [Data Coverage]（数据覆盖）窗口

B. 在显示的 [Edit Address Range]（编辑地址范围）对话框中，更改地址范围。

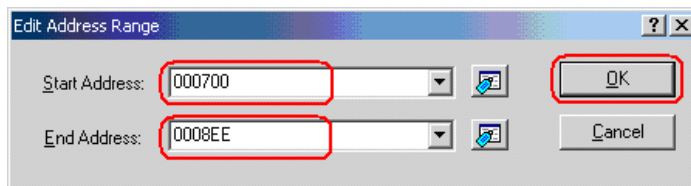


图 5.112 [Edit Address Range]（编辑地址范围）对话框

C. 更改后的地址范围将显示在 [Data Coverage]（数据覆盖）窗口的上面区域。

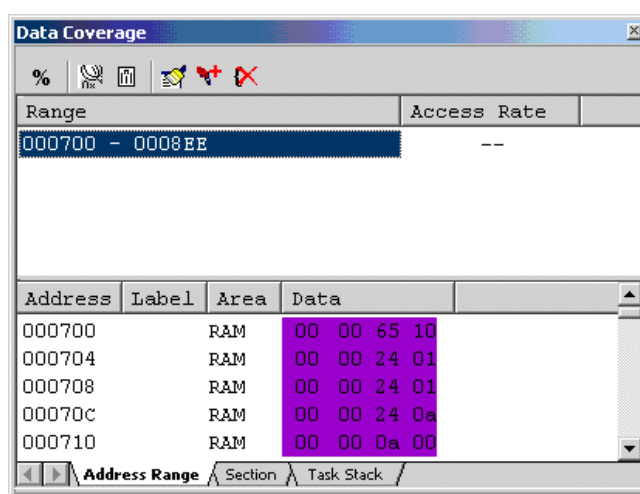


图 5.113 [Data Coverage]（数据覆盖）窗口

5.12.7 移除地址范围

按照下述步骤移除地址范围。

1. 从 [Data Coverage]（数据覆盖）窗口的 [Address Range]（地址范围）页执行以下操作：
 - A. 在 [Address Range]（地址范围）页中选择要移除的地址范围，保持其处于选中状态，并从上下文菜单选择 [Delete Range]（删除范围）。

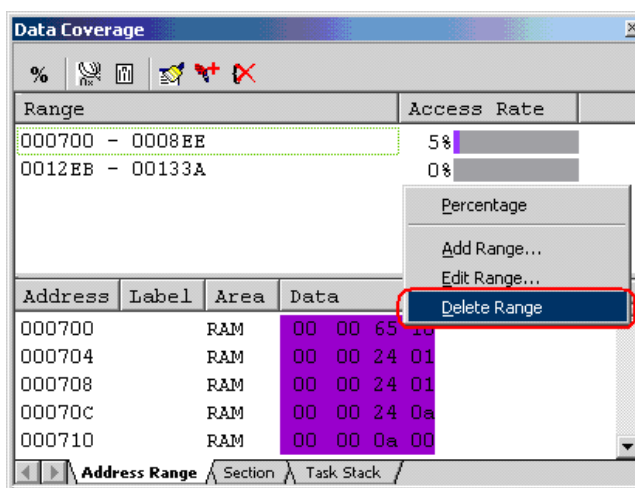


图 5.114 [Data Coverage]（数据覆盖）窗口

- B. 此时将显示一个对话框，提示用户进行确认。选择保存或不保存覆盖数据。要保存，请单击 [Yes]（是）按钮。如果不保存，请单击 [No]（否）按钮。

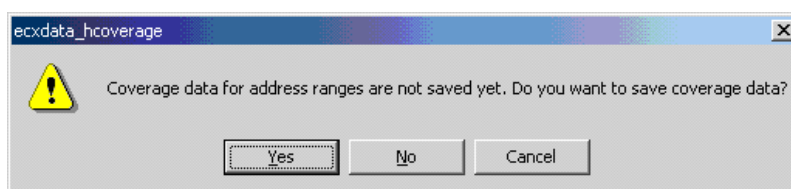


图 5.115 [Confirmation of Edit Address Range]（确认编辑地址范围）对话框

C. 将删除选定的地址范围。

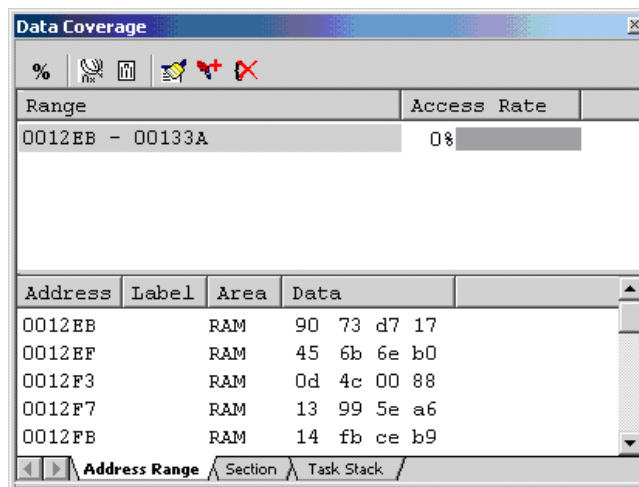


图 5.116 [Data Coverage] (数据覆盖) 窗口

5.12.8 测量段

E100 仿真器显示从用户指定的段收集的存取信息。

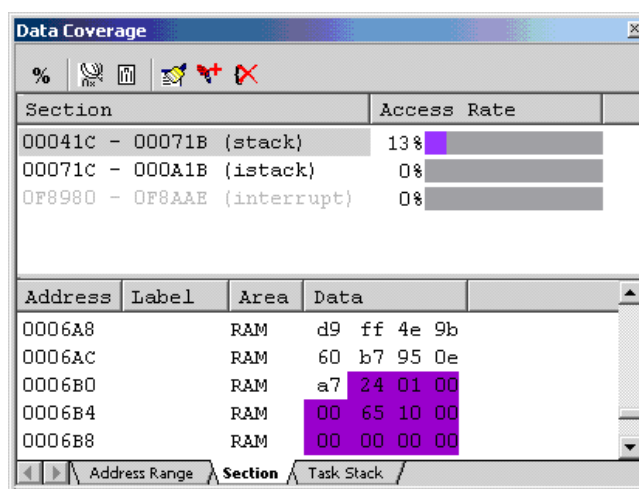


图 5.117 [Data Coverage] (数据覆盖) 窗口 (指定段名称)

一条分割线将 [Data Coverage] (数据覆盖) 窗口垂直平均分为两部分。
上面的区域显示要测量的地址范围 (段名称) 和存取率。

表 5.36 [Data Coverage] (数据覆盖) 窗口上面区域中的内容

[Section] (段)	测量其中覆盖的地址范围 (段)
[Access Rate] (存取率)	按百分比和图形显示存取率

下面区域显示上面区域中选定的地址范围的详细信息。

表 5.37 [Data Coverage] (数据覆盖) 窗口下面区域中的内容

Address (地址)	地址值
Label (标签)	标签名称
Area (区)	存储区 (FlashROM、RAM、SFR) 对于未使用的区此列为空。
Data (数据)	存储器数据 存取的数据的背景显示为紫色。

如果地址位于覆盖存储器分配的区域外部, 则地址行以灰色显示。尽管保留了这些地址的现有覆盖信息, 并不更新程序执行所做的覆盖信息。

获取的覆盖信息将一直在存储器中累积, 直到用户将其清除。

5.12.9 添加段

按照下述步骤添加地段。

1. 从 [Data Coverage]（数据覆盖）窗口的 [Section]（段）页执行以下操作：
 - A. 右键单击 [Section]（段）页的上面区域并从上下文菜单中选择 [Add Range]（添加范围）。

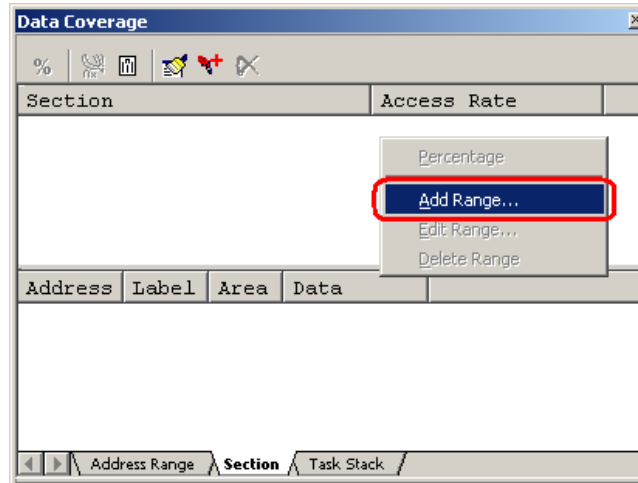


图 5.118 [Data Coverage]（数据覆盖）窗口

- B. 在显示的 [Add A Section]（添加段）对话框中，输入段名称。



图 5.119 [Add A Section]（添加段）对话框

- C. 添加的地址范围（段名称）将显示在 [Data Coverage]（数据覆盖）窗口的上面区域。

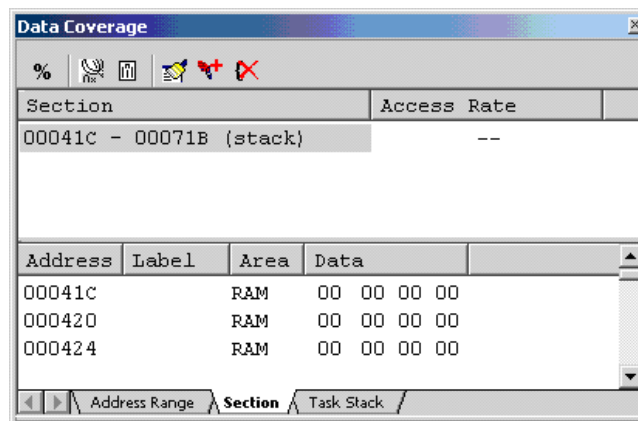


图 5.120 [Data Coverage]（数据覆盖）窗口

5.12.10 移除段

按照下述步骤移除段。

1. 从 [Data Coverage]（数据覆盖）窗口的 [Section]（段）页执行以下操作：
 - A. 在 [Section]（段）页中选择要移除的段名称，保持其处于选中状态，并从上下文菜单选择 [Delete Range]（删除范围）。

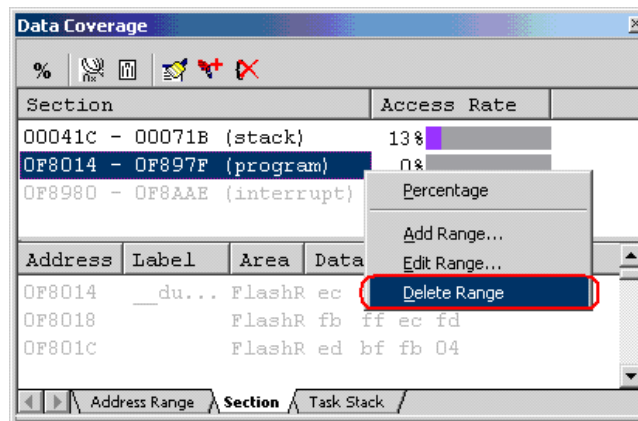


图 5.121 [Data Coverage]（数据覆盖）窗口

- B. 此时将显示一个对话框，提示用户进行确认。选择保存或不保存覆盖数据。要保存，请单击 [Yes]（是）按钮并指定文件名。如果不保存，请单击 [No]（否）按钮。

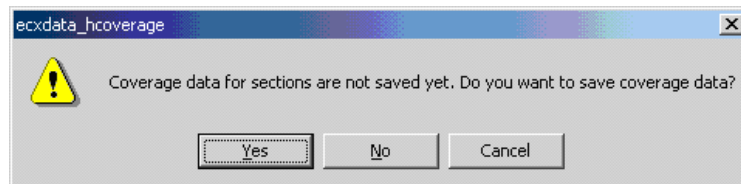


图 5.122 [Confirmation of Removing Section]（确认移除段）对话框

C. 将删除选定的段名称。

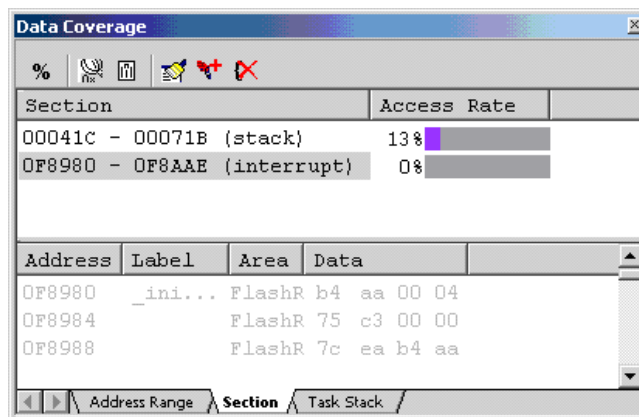


图 5.123 [Data Coverage] (数据覆盖) 窗口

5.12.11 测量任务堆栈

[Task Stack]（任务堆栈）页显示从任务堆栈收集的存取信息。

任务堆栈是自动注册的。

不能添加、删除或更改任何任务。

例如，如果任务根据用户程序的更改发生了更改，则窗口将自动更新。

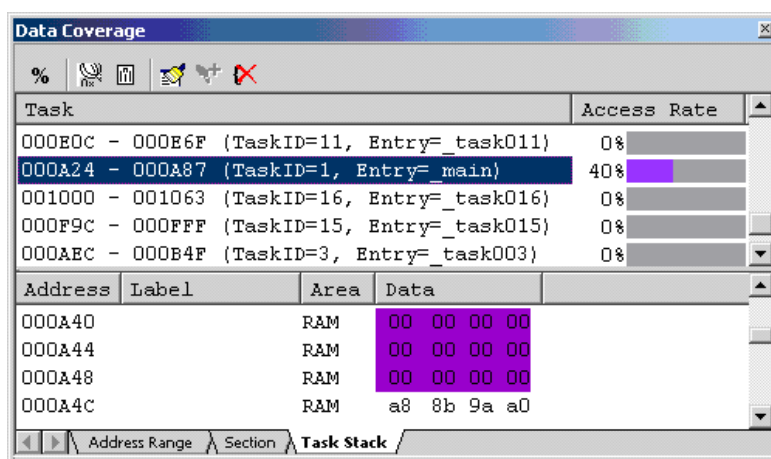


图 5.124 [Data Coverage]（数据覆盖）窗口（指定任务堆栈）

一条分割线将 [Data Coverage]（数据覆盖）窗口垂直平均分为两部分。

上面的区域显示自动注册的任务堆栈和存取率。

表 5.38 [Data Coverage]（数据覆盖）窗口上面区域中的内容

[Task]（任务）	任务堆栈（任务 ID、任务条目标签）
[Access Rate]（存取率）	按百分比和图形显示存取率

下面区域显示上面区域中选定的任务堆栈的详细信息。

表 5.39 [Data Coverage]（数据覆盖）窗口上面区域中的内容

[Address]（地址）	地址值
[Label]（标签）	标签名称
[Area]（区）	存储区（FlashROM、RAM、SFR） 对于未使用的区此列为空。
[Data]（数据）	存储器数据 存取的数据的背景显示为紫色。

如果地址位于覆盖存储器分配的区域外部，则地址行以灰色显示。尽管保留了这些地址的现有覆盖信息，并不更新程序执行所做的覆盖信息。

获取的覆盖信息将一直在存储器中累积，直到用户将其清除。

5.12.12 清除 [Data Coverage]（数据覆盖）信息

(1) 清除指定的数据覆盖信息范围

从 [Address Range]（地址范围）或 [Section]（段）页的上下文菜单选择 [Clear Coverage Range]（清除覆盖范围）。此时将显示如下所示的 [Clear Coverage Range]（清除覆盖范围）对话框。

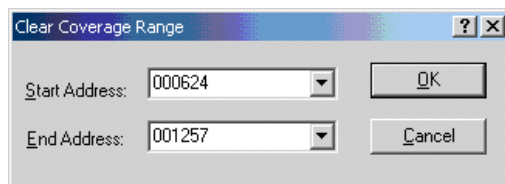


图 5.125 [Clear Coverage Range]（清除覆盖范围）对话框

指定要清除范围的起始地址和结束地址。单击 [OK]（确定）按钮，将清除指定的范围。

(2) 清除所有数据覆盖信息

从上下文菜单中选择 [Clear the Entire Coverage]（清除整个覆盖），将清除所有数据覆盖信息。

5.12.13 更新覆盖信息

将 [Data Coverage]（数据覆盖）窗口的内容更新为最新内容。

从 [Data Coverage]（数据覆盖）窗口的上下文菜单中选择 [Refresh]（刷新）。

如果禁止更新覆盖信息，则在程序中断时将不自动更新这些信息。因此，需要手动更新才能查看最新的信息。

5.12.14 禁止信息更新

有时 [Data Coverage]（数据覆盖）窗口的内容不进行更新，例如，在用户系统停止运行时。

请从 [Data Coverage]（数据覆盖）窗口的上下文菜单中选择 [Lock Refresh]（锁定刷新）。

5.12.15 将数据覆盖信息保存至文件

将当前选定页的数据覆盖信息保存至文件。

从 [Data Coverage]（数据覆盖）窗口的上下文菜单中选择 [Save Data]（保存数据），将显示如下所示的 [Save Data]（保存数据）对话框。

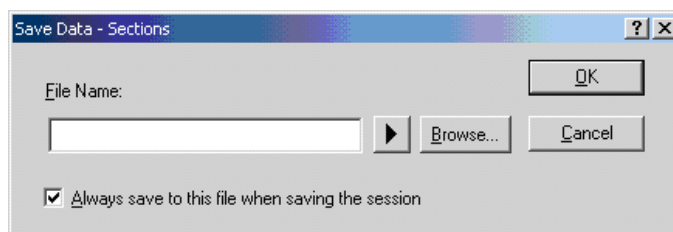


图 5.126 [Save Data]（保存数据）对话框

输入要在其中保存信息的文件的名称。

如果省略文件扩展名，将自动附加扩展名 “.cdv”。

如果指定现有的文件名，将盖写该文件。

5.12.16 从文件加载数据覆盖信息

加载数据覆盖信息文件。

从 [Data Coverage]（数据覆盖）窗口的上下文菜单中选择 [Load Data]（加载数据），将显示如下所示的 [Load Coverage Data]（加载覆盖数据）对话框。

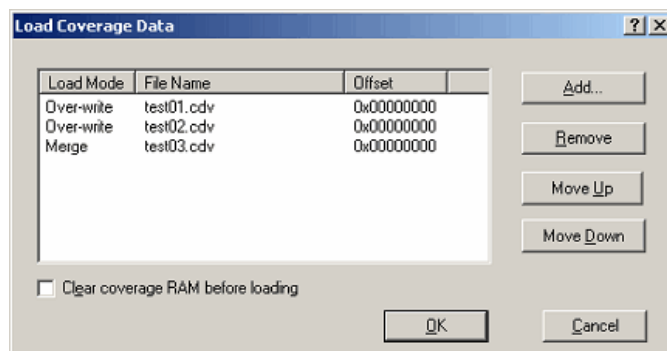


图 5.127 [Load Coverage Data]（加载覆盖数据）对话框

单击 [Add]（添加）按钮，将显示如下所示的 [Add coverage data file]（添加覆盖数据文件）对话框。

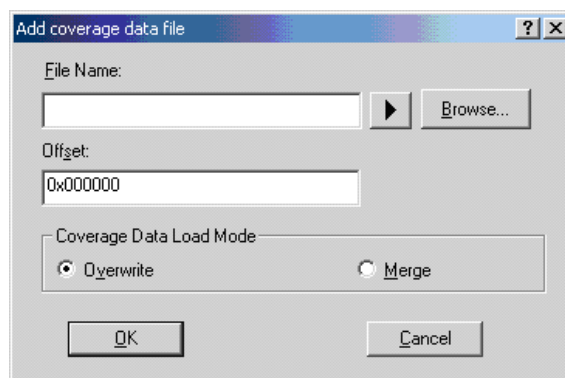


图 5.128 [Add coverage data file]（添加覆盖数据文件）对话框

使用此对话框输入要加载的覆盖信息文件。

可以为加载的每个文件指定加载模式和偏移量。

只能加载扩展名为 “.cdv” 的文件。如果输入任何其他文件扩展名，将输出一条错误消息。

在 [Load Coverage Data]（加载覆盖数据）对话框中会列出所添加的文件。

将以文件列出顺序进行加载。

可根据需要使用 [Up]（向上）或 [Down]（向下）按钮来更改顺序。

5.13 查看实时分析信息

5.13.1 查看实时分析信息

E100 仿真器具有自己的代码覆盖，数据覆盖和实时分析功能，且不能同时使用。

要使用实时分析功能，请在 [Configuration Properties]（配置属性）对话框的 [System]（系统）页上的 [Switching Function]（切换功能）部分中选择 [Realtime Profile]（实时分析）。

实时分析是一种功能，用来测量为分析范围中的地址所分配区域的执行性能，一次执行一个函数或任务。它可以帮助在应用程序中查找性能降级的位置和原因。

可以在不妨碍用户程序执行的情况下进行测量。

测量结果会在程序中断时进行更新。

(1) 函数分析

一次测量一个函数的执行性能。

[Realtime profile]（实时分析）窗口显示函数名称、函数的起始地址、函数大小、函数计数和函数的累积执行时间以及执行速率和平均执行时间。

E100 仿真器的函数分析在其函数执行时间的累积显示中不包括子例程的执行时间。

注意：功能分析遵循以下限制：

(a) 关于要测量的区域

E100 仿真器最多可获取 8 块区域中（每块单位为 128 KB）所有函数的分析信息。

设置的每个块可以由连续或非连续地址区组成。

不能设置位于块地址范围外的函数。在这种情况下，函数或任务以灰色显示。

(b) 对函数数目的限制

最多可以对 $8K - 1 (= 8,191)$ 个函数执行测量。

如果测量的函数数目超过 $8K - 1 (= 8,191)$ ，则其他函数将排除在该测量主题之外。在这种情况下，这些函数以灰色显示。（函数名称、地址和函数大小都以灰色显示。）

(c) 内联扩展

[Realtime Profile]（实时分析）窗口中不显示编译器内联扩展进行优化的函数。

(d) 递归函数

尽管可以正确测量递归函数的执行时间，但是它们仅执行一次。

(e) 测量范围内 Go 执行起始地址和中断地址与可测量范围之间的关系

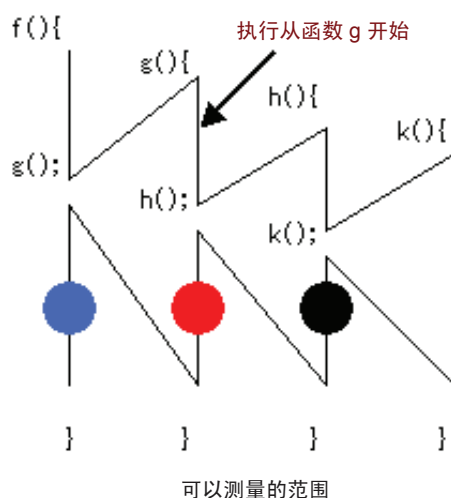


图 5.129 可测量范围

程序在黑点位置 [●] 中断时可测量的范围：函数 h 和 k 的执行时间和执行计数

程序在红点位置 [●] 中断时可测量的范围：函数 h 和 k 的执行时间和执行计数

程序在蓝点位置 [●] 中断时可测量的范围：函数 h 和 k 的执行时间和执行计数

对于函数 g，可以测量其执行部分中的执行时间和计数。

因此，上述关系是中断地址和可测量范围之间的关系。

即使程序返回到高阶函数，也无法测量程序执行开始时的函数的执行计数。

(f) 函数测量

要正确测量函数，需要在进入函数后 100 ns 或更长时间内测量该函数。否则，测量的执行时间和计数可能不正确。

(g) 调试信息选项

要获得函数的执行时间和执行计数，需要指定一个包含要进行测量的函数的源文件或指定一个在编译过程中将调试信息输出到库中的选项。如果不指定 [Debug information]（调试信息）选项，则无法测量函数的执行时间和执行计数。

(h) 最长执行时间和最短执行时间

使用实时分析无法测量函数的最长和最短执行时间。要测量函数的最长和最短执行时间，请使用 [Performance Analysis]（性能分析）窗口。

(2) 任务分析

一次测量一个任务的执行性能。

[Realtime profile]（实时分析）窗口显示任务 ID、计数和累积执行时间以及执行速率和平均执行时间。

5.13.2 设置实时分析测量模式

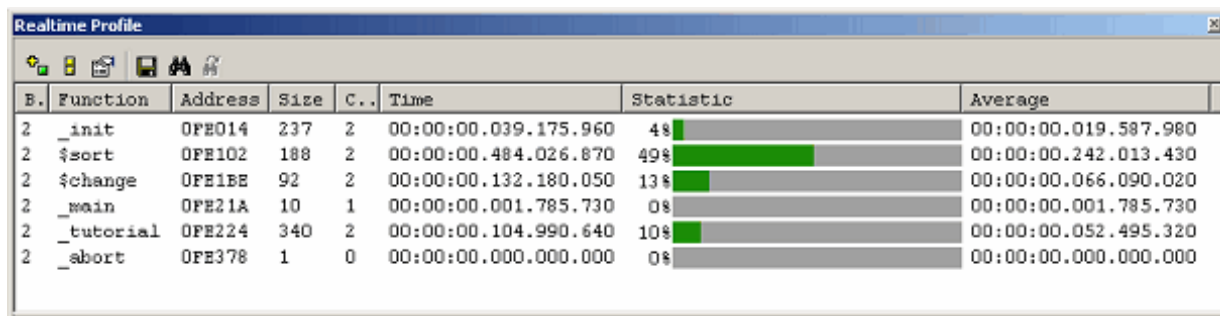
右键单击当前窗口将显示上下文菜单，从中选择 [Set Range]（设置范围）。

将显示 [Realtime Profile Setting]（实时分析设置）对话框。在此对话框的 [Profile Mode]（分析模式）列表框中，可以选择“function profile”（函数分析）或“Task profile”（任务分析）。

如果分析模式发生改变，将清除所有测量结果。

5.13.3 测量函数分析

一次测量一个函数的执行性能。



B.	Function	Address	Size	C..	Time	Statistic	Average
2	_init	0FE014	237	2	00:00:00.039.175.960	4%	00:00:00.019.587.980
2	\$_sort	0FE102	188	2	00:00:00.484.026.870	49%	00:00:00.242.013.430
2	\$_change	0FE1BE	92	2	00:00:00.132.180.050	13%	00:00:00.066.090.020
2	_main	0FE21A	10	1	00:00:00.001.785.730	0%	00:00:00.001.785.730
2	_tutorial	0FE224	340	2	00:00:00.104.990.640	10%	00:00:00.052.495.320
2	_abort	0FE378	1	0	00:00:00.000.000.000	0%	00:00:00.000.000.000

图 5.130 [Realtime Profile]（实时分析）窗口（函数分析）

下面显示每列的详细信息。

表 5.40 有关每列的详细信息

[Block]（块）	块编号
[Function]（函数）	函数名称
[Address]（地址）	函数的起始地址
[Size]（大小）	函数大小
[Count]（计数）	调用函数的次数
[Time]（时间）	函数执行的累积时间 时间戳按照如下所示格式显示。 小时:分钟:秒.毫秒.微秒.纳秒
[Statistic]（统计信息）	函数时间与 Go-Break 执行时间的比率
[Average]（平均）	每次执行测量的平均执行时间

如果函数位于分析存储器分配的区域外部，则地址行以灰色显示。

获取的分析测量结果将一直在存储器中累积，直到用户将其清除。

5.13.4 设置函数分析测量范围

右键单击当前窗口将显示上下文菜单，从中选择 [Set Range]（设置范围）。
将显示 [Realtime Profile Setting]（实时分析设置）对话框。在此对话框中，设置分析测量范围。
[“Function”（函数）模式]

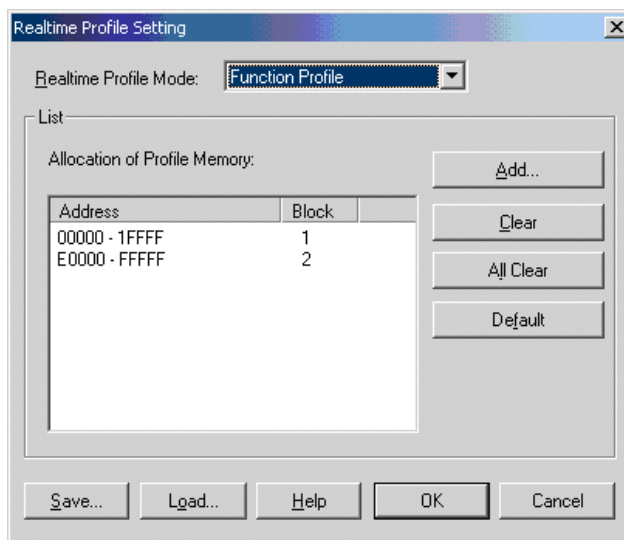


图 5.131 [Realtime Profile Setting]（实时分析设置）对话框

(1) 存储器分配

必须将分析存储器分配给要对其进行测量的地址后，才能测量函数分析。只能从分配了存储器的地址范围获取分析数据。

仿真器允许将每个以 128 KB 范围开头的 1~8 块（最大 1 KB）中的任意一块指定为分析测量区域。可以设置连续块、非连续块或一个块。

初始设置时，会自动为 ROM 和 RAM 区域中的地址分配分析存储器。

(2) 自动函数检测

将分析存储器分配给地址后，E100 仿真器会自动检测该地址范围包含的函数并在窗口中注册这些函数。

5.13.5 保存函数分析测量范围

保存当前的任务模式和函数分析测量范围（存储器分配状态）。

单击 [Realtime Profile Setting]（实时分析设置）对话框的 [Save]（保存）按钮，将显示 [Save As]（另存为）对话框。

输入要在其中保存函数分析测量范围的文件的名称。

如果省略文件扩展名，将自动附加扩展名 “.rpf”。

如果指定现有文件名，则会显示一条消息，要求确认是否要盖写该文件。

5.13.6 加载函数分析测量范围

加载函数分析测量范围。

单击 [Realtime Profile Setting]（实时分析设置）对话框的 [Load]（加载）按钮，将显示 [Open]（打开）对话框。

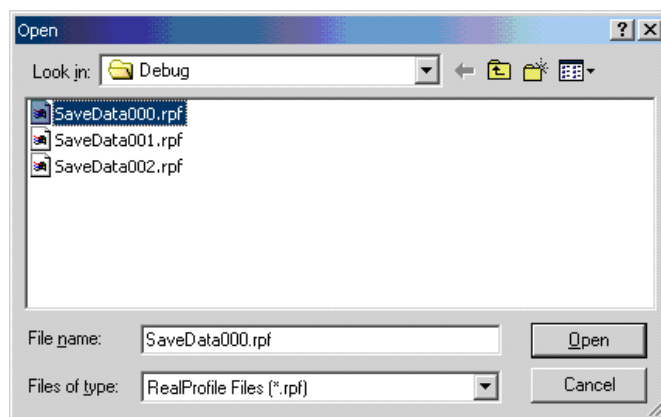


图 5.132 [Open]（打开）对话框

输入要加载的文件的名称。

只能加载扩展名为 “.rpf” 的文件。如果输入任何其他文件扩展名，将输出一条错误消息。

文件加载完毕后，将更新 [Realtime Profile Setting]（实时分析设置）对话框中的列表。

如果加载的文件中包含任务分析信息，则 [Realtime Profile Setting]（实时分析设置）对话框中的模式将切换到任务模式。

5.13.7 测量任务分析

一次测量一个任务的执行性能。

Block	Task ID	Count	Time	Statistic	Average
1	0 (M3T-MR30/4 or Idle)	1	00:00:00.577.565.090	16%	00:00:00.577.565.090
1	1 (_main)	1	00:00:00.084.891.850	2%	00:00:00.084.891.850
1	2 (_task1)	38	00:00:00.972.308.480	28%	00:00:00.025.587.060
1	3 (_task2)	25	00:00:00.541.265.350	15%	00:00:00.021.650.610

图 5.133 [Realtime Profile]（实时分析）对话框（任务分析）

下面显示每列的详细信息。

表 5.41 有关每列的详细信息

[Block] (块)	块编号
[Task ID] (任务 ID)	任务 ID, 条目地址
[Count] (计数)	调用任务的次数
[Time] (时间)	任务执行的累积时间 时间戳按照如下所示格式显示。 小时:分钟:秒.毫秒.微秒.纳秒
[Statistic] (统计信息)	任务时间与 Go-Break 执行时间的比率
[Average] (平均)	每次执行测量的平均执行时间

禁用的任务以灰色显示。

获取的分析测量结果将一直在存储器中累积，直到用户将其清除。

5.13.8 设置任务分析测量范围

右键单击当前窗口将显示上下文菜单，从中选择 [Set Range] (设置范围)。

将显示 [Realtime Profile Setting] (实时分析设置) 对话框。在此对话框中，设置分析测量范围。

[“Task” (任务) 模式]

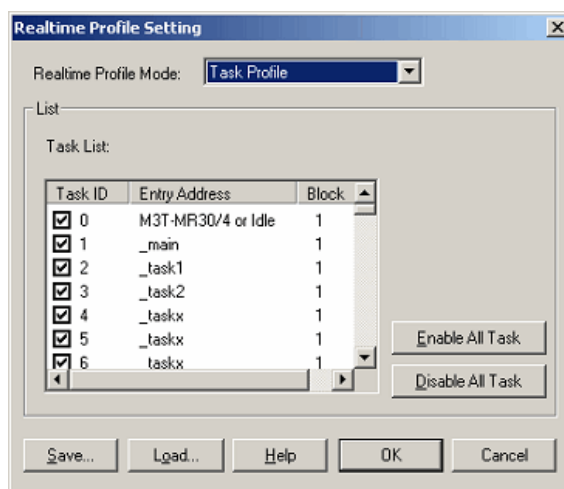


图 5.134 [Realtime Profile Setting] (实时分析设置) 对话框

(1) 自动任务检测

如果下载的加载模块包含操作系统，则 E100 仿真器将自动检测任务列表。

(2) 选择任务

选中要测量的任务 ID 的复选框。（默认情况下，选中所有复选框。）

将自动为选定的任务分配块数 (1~8)。

注意：如果缺少测量块，则块编号变为空，也就无法再注册任何任务 ID。在这种情况下，请取消选中不必要任务 ID 的复选框。

5.13.9 保存任务分析测量任务

保存当前的任务模式和测量任务（任务 ID 和允许的 / 禁止的状态）。

单击 [Realtime Profile Setting]（实时分析设置）对话框的 [Save]（保存）按钮，将显示 [Save As]（另存为）对话框。

输入要在其中保存任务分析测量任务的文件的名称。

如果省略文件扩展名，将自动附加扩展名 “.rpf”。

如果指定现有文件名，则会显示一条消息，要求确认是否要盖写该文件。

5.13.10 加载任务分析测量任务

加载任务分析测量任务。

单击 [Realtime Profile Setting]（实时分析设置）对话框的 [Load]（加载）按钮，将显示 [Open]（打开）对话框。

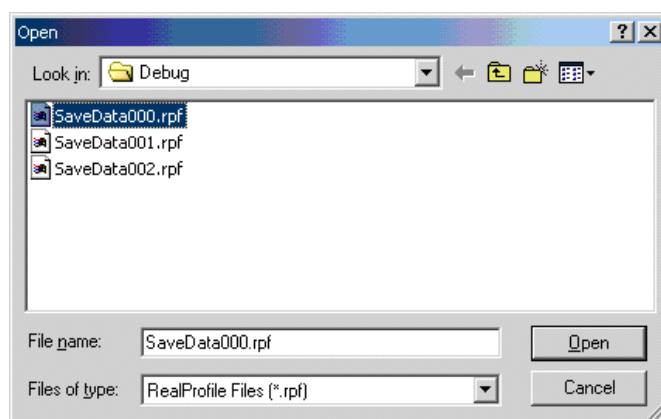


图 5.135 [Open]（打开）对话框

输入要加载的文件的名称。

只能加载扩展名为 “.rpf” 的文件。如果输入任何其他文件扩展名，将输出一条错误消息。

文件加载完成后，将更新 [Realtime Profile Setting]（实时分析设置）对话框中的列表（任务列表）。

如果任何加载的任务 ID 不存在，尽管它们曾经显示在 [Realtime Profile Setting]（实时分析设置）对话框中的列表（任务列表）中，在单击 [OK]（确定）按钮时也只有现有的任务 ID 注册为测量任务。

重新打开 [Realtime Profile Setting]（实时分析设置）对话框可以检查当前注册的测量任务。

如果加载的文件中包含函数分析信息，则 [Realtime Profile Setting]（实时分析设置）对话框中的模式将切换到函数模式。

5.13.11 清除实时分析测量结果

从 [Realtime Profile]（实时分析）窗口的上下文菜单选择 [Clear]（清除），将清除所有测量结果。

除非选择 [Clear]（清除），否则测量结果将一直在存储器中累积。

5.13.12 保存实时分析测量结果

以文本格式保存当前实时分析测量结果。

从 [Realtime Profile]（实时分析）窗口的上下文菜单中选择 [Save To File]（保存到文件），将显示 [Save As]（另保存）对话框。

输入要在其中保存测量结果的文件的名称。

如果省略文件扩展名，将自动附加扩展名 “.txt”。

如果指定现有文件名，则会显示一条消息，要求确认是否要盖写该文件。

5.13.13 设置测量单位

右键单击当前窗口将显示上下文菜单，从中选择 [Properties]（属性）。

将显示 [Properties]（属性）对话框。

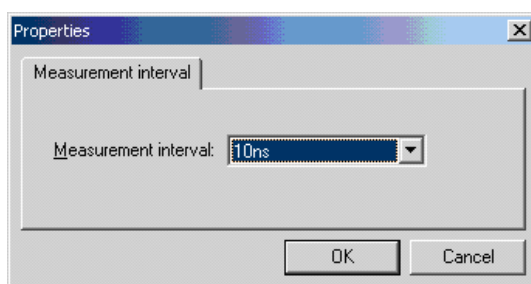


图 5.136 [Properties]（属性）对话框

可以从以下选项中选择测量单位：

[10 ns]（10 纳秒）、[20 ns]（20 纳秒）、[40 ns]（40 纳秒）、[80 ns]（80 纳秒），[160 ns]（160 纳秒）和 [1.6 μ s]（1.6 微秒）

注意： 如果当前设置的测量单位发生了更改，则迄今为止累积的测量结果都会被清除。

5.13.14 实时分析的最长测量时间

(1) 最长测量时间

用于性能测量的定时器由一个 40 位计数器组成。最长测量时间因选择的测量单位而异。

要选择测量单位，请在 [Properties]（属性）对话框的 [Measurement Interval]（测量间隔）下拉式列表中指定它。

下面列出了可测量的最长时间。

表 5.42 最长测量时间

编号	解析度	最长测量时间
1	10ns	约 3 小时 03 分钟 15 秒
2	20ns	约 6 小时 06 分钟 30 秒
3	40ns	约 12 小时 13 分钟 00 秒
4	80ns	约 24 小时 26 分钟 00 秒
5	160ns	约 48 小时 52 分钟 01 秒
6	1.6us	约 488 小时 40 分钟 18 秒

注意：只要输入函数，性能测量就会发生导致 \pm “2 resolution + 100 ns”（2 个解析度 + 100 ns）的误差（如果解析度 = 20 ns、 \pm 140 ns）。如果解析度为 20 ns 且输入函数 10 次，则产生 \pm 1400 ns 误差。

(2) 最大测量计数

使用 16 位计数器测量实时分析的执行计数。执行测量的最大计数可以为 65,535。

5.14 检测异常事件

5.14.1 检测异常事件

E100 仿真器允许检测用户程序执行过程中发生的各种异常事件。

异常事件包括用户程序的不正常行为以及涉及任何函数的测量计数器的上溢等。可以将指定异常事件的检测设置为断点或跟踪点的条件。

(1) 异常事件

E100 仿真器可检测下列异常事件。

- 存取保护的违规操作：尝试指定存取属性之外的存取时将检测到错误。
- 从未初始化的存储器读取：读存取未初始化的区域（未写存取的）时将检测到错误。
- 堆栈存取的违规操作：检测到堆栈寄存器的值超出了堆栈区域。
- 性能上溢：检测到在程序的指定段的时间测量期间，某些段超出了最长测量时间或最大测量计数。
- 实时分析上溢：检测到在分析测量期间，某些函数（或任务）超出了最长测量时间或最大测量计数。
- 跟踪存储器上溢：检测到跟踪存储器上溢。
- 任务堆栈存取的违规操作：检测到曾尝试从另一个任务写入相关任务堆栈。
- 操作系统调度：检测到发生任务调度。

5.14.2 检测存取保护的违规操作

这是一种用于检测存取保护的违规操作的功能，可以检测向 ROM 区写入数据或对未使用的区进行存取（如读、写或指令执行）以及输出错误等。

(1) 存取属性

对于任何区域都可以在字单元中指定以下属性。

Read/Write（读 / 写）：可进行读 / 写存取

Read Only（只读）：只可读

Write Only（只写）：只可写

Disable（禁止）：禁止存取

Disable (OS)（禁止（操作系统））：禁止来自操作系统之外的所有存取（如果下载的程序包含操作系统，则自动分配此属性）。

(2) 受保护的区域

可以对整个存储器空间的任何区域进行存取保护。

仿真器启动时，默认情况下会为整个区域分配 Read/Write（读 / 写）存取属性。

(3) 设置保护的方法

有以下两种指定方法：

- 根据下载模块中的段信息自动设置
- 分别为各个区域指定存取属性

(4) 检测方法

由仿真器的内部资源（块 1~16）检测存取保护的违规操作。

块是由仿真器的互斥算法自动分配的。

注意： 由于仿真器的内部资源有限，因此并不能对所有块执行存取保护。在这种情况下，可使用“removing blocks”（移除块）减少使用的块数量，然后再设置保护。

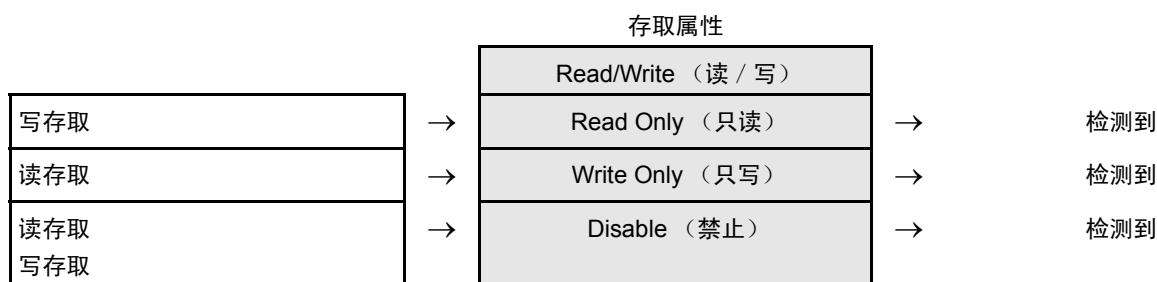


图 5.137 检测方法的 NG 模式

(5) 检测到存取保护的违规操作时执行的操作

可以设置以下操作：

- 显示警告
通过在 [Configuration properties]（配置属性）对话框的 [Exception Warning]（异常警告）页面上选中 [Access Protect Violation]（存取保护的违规操作）复选框，可以在 [Status]（状态）窗口和状态栏提示框中显示警告。
- 将存取保护的违规操作检测设置为硬件断点条件
- 将存取保护的违规操作检测设置为跟踪点条件

5.14.3 设置存取保护区

按照下述步骤设置存取保护区。

1. 从 [Hardware Break] (硬件断点) 对话框执行以下操作:
 - A. 选中 [Hardware Break] (硬件断点) 页上的 [Exception] (异常) 复选框, 然后单击 [Detail] (详细信息) 按钮。

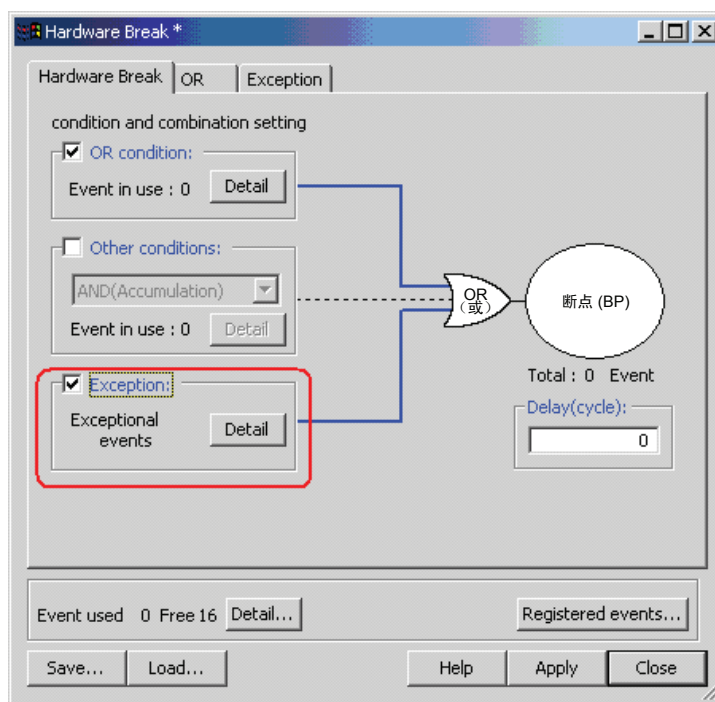


图 5.138 [Hardware Break] (硬件断点) 对话框

- B. 此时将出现如下所示的 [Exception]（异常）页面。单击 [Violation of access protection]（存取保护的违规操作）复选框右边的 [Detail]（详细信息）按钮。

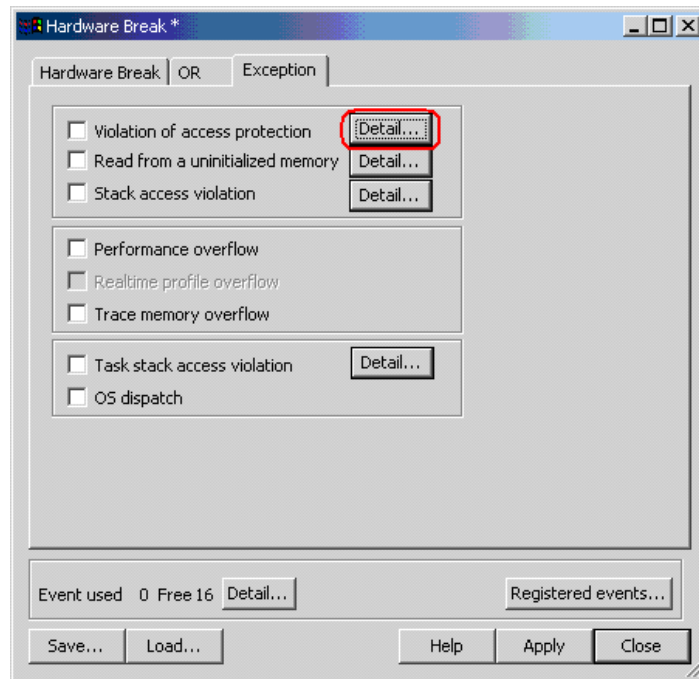


图 5.139 [Hardware Break]（硬件断点）对话框

- C. 此时将显示如下所示的 [Violation of access protection]（存取保护的违规操作）对话框。要根据下载程序时下载模块中的段信息来自动设置存取属性，请选中标有“Automatically set address areas at downloading”（下载时自动设置地址区）的复选框。

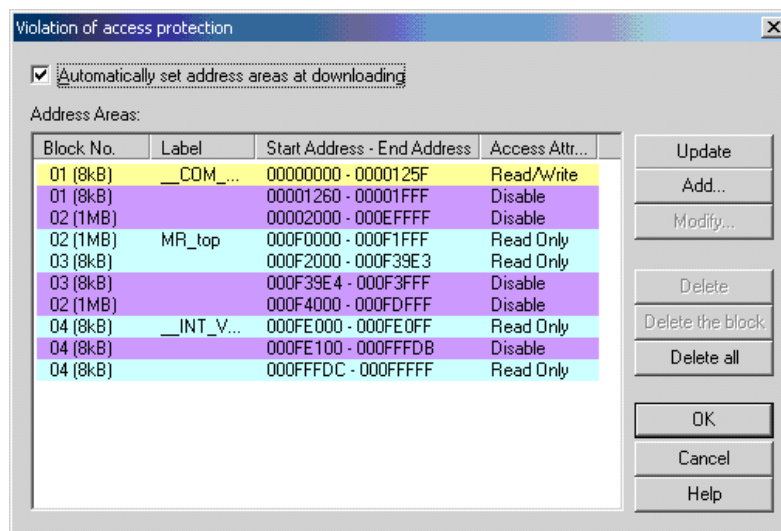


图 5.140 [Violation of access protection]（存取保护的违规操作）对话框

- D. 单击 [Update]（更新）按钮，存取属性将根据下载模块中的段信息进行更新。
- E. 要手动添加存取属性，请单击 [Add]（添加）按钮。此时将显示如下所示的 [Access protection condition]（存取保护条件）对话框。指定任意地址范围和存取属性。

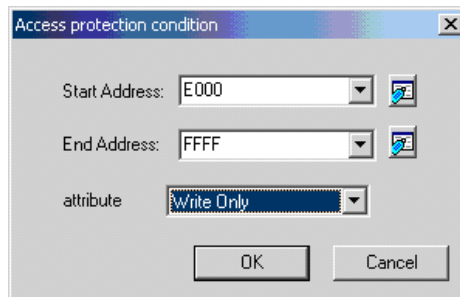


图 5.141 [Access protection condition]（存取保护条件）对话框

- F. 添加的保护区域将显示在 [Violation of access protection]（存取保护的违规操作）对话框的 [Address Areas]（地址区）列表中。

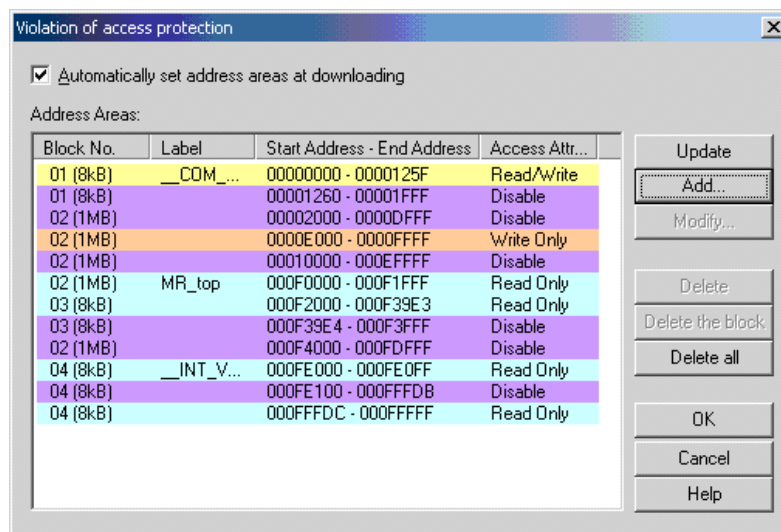


图 5.142 [Violation of access protection]（存取保护的违规操作）对话框

2. 从 [Trace conditions]（跟踪条件）对话框执行以下操作：

- A. 在 [Trace]（跟踪）页中的 [Trace Mode]（跟踪模式）下拉式列表中，选择 [Fill around TP]（围绕 TP 填充）。选中 [Exception]（异常）复选框，然后单击 [Detail]（详细信息）按钮。

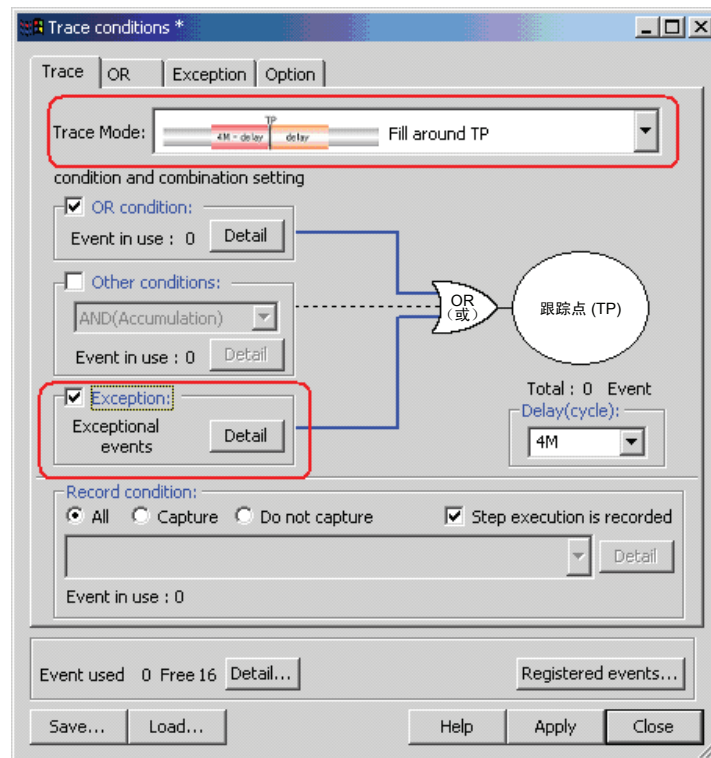


图 5.143 [Trace conditions]（跟踪条件）对话框

- B. 此时将出现如下所示的 [Exception]（异常）页面。单击 [Violation of access protection]（存取保护的违规操作）复选框右边的 [Detail]（详细信息）按钮。

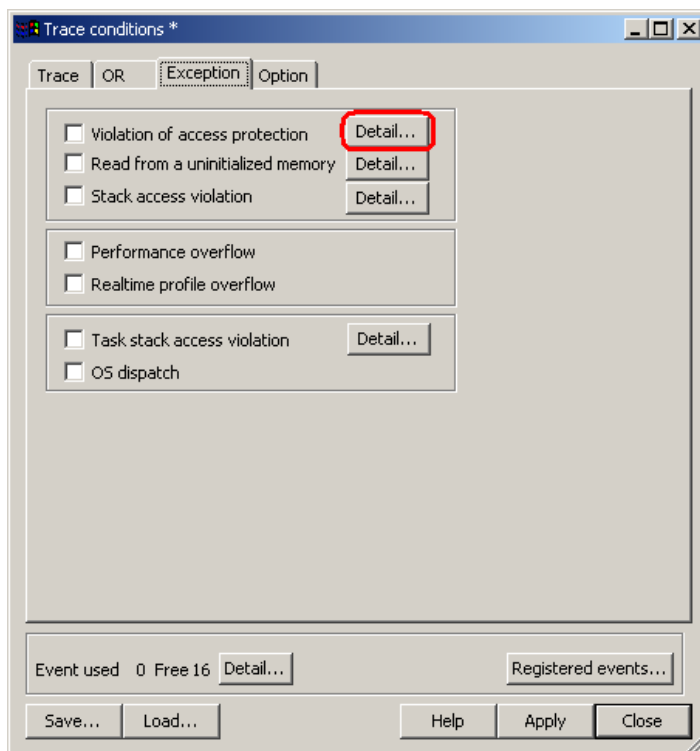


图 5.144 [Trace conditions]（跟踪条件）对话框

- C. 此时将显示 [Violation of access protection]（存取保护的违规操作）对话框。其他步骤与从 [Hardware Break]（硬件断点）对话框打开该对话框时相同。

5.14.4 检测忽略初始化

这是一种功能，用于确定读存取和写存取的历史记录都不存在时，在执行写存取之前执行读存取的情况为“initialization omitted”（忽略初始化的）并输出错误。

在仿真器中，可将块 0~31（最大 16 KB）指定为忽略初始化的检测区域。

(1) 检测方法

由 RAM 监控功能来检测忽略初始化。

为给定的地址范围分配 RAM 监控区域并在该区域启用错误检测。

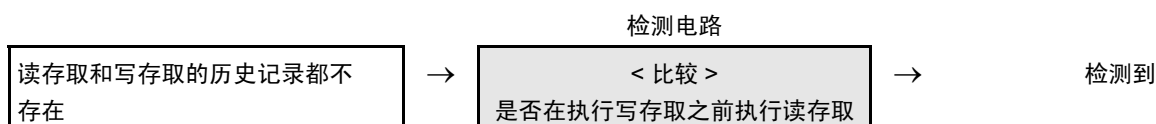


图 5.145 忽略初始化概图

(2) 检测到忽略初始化时执行的操作

可以设置以下操作：

— 显示警告

通过在 [Configuration properties]（配置属性）对话框的 [Exception Warning]（异常警告）页面上选中 [Read from uninitialized memory]（从未初始化的存储器读取）复选框，可以在 [Status]（状态）窗口和状态栏提示框中显示警告。

[RAM Monitor]（RAM 监控）窗口中的颜色显示

— 将忽略初始化检测设置为硬件断点条件

— 将忽略初始化检测设置为跟踪点条件

5.14.5 检测性能上溢

这是一种功能，用于检测性能函数测量的时间或计数已超出最长测量时间或最大测量计数并输出一个错误。

性能测量中的超时和计数上溢（计数超出限制）事例统称为性能上溢。

(1) 检测到性能上溢时执行的操作

可以设置以下操作：

— 显示警告

警告显示在 [Performance]（性能）窗口中。

对程序段中出现超时或计数上溢现象的结果显示行，使用字符串“Time out”（超时）或“Count out”（计数上溢）进行标记。

通过在 [Configuration properties]（配置属性）对话框的 [Exception Warning]（异常警告）页面上选中 [Performance Overflow]（性能上溢）复选框，可以在 [Status]（状态）窗口和状态栏提示框中显示警告。

— 将性能上溢检测设置为硬件断点条件

— 将性能上溢检测设置为跟踪点条件

5.14.6 检测实时分析上溢:

这是一种功能，用于检测实时分析功能测量的时间或计数是否已超出最长测量时间或最大测量计数并输出一个错误。

实时分析中的超时和计数上溢（计数超出限制）事例统称为实时分析上溢。

(1) 检测到实时分析上溢时执行的操作

可以设置以下操作：

- 显示警告

警告显示在 [Realtime Profile]（实时分析）窗口中。

对于任务中出现超时或计数上溢现象的函数或结果显示行，使用字符串“overflow”（上溢）进行标记。

通过在 [Configuration properties]（配置属性）对话框的 [Exception Warning]（异常警告）页面上选中 [Realtime Profile Overflow]（实时分析上溢）复选框，可以在 [Status]（状态）窗口和状态栏提示框中显示警告。

- 将实时分析上溢检测设置为硬件断点条件
- 将实时分析上溢检测设置为跟踪点条件

5.14.7 检测跟踪存储器上溢

这是一种功能，用于检测跟踪存储器容量（4MB 循环）上溢并输出一个错误。

(1) 检测到跟踪存储器上溢时执行的操作

可以设置以下操作：

- 显示警告

通过在 [Configuration properties]（配置属性）对话框的 [Exception Warning]（异常警告）页面上选中 [Trace memory overflow]（跟踪存储器上溢）复选框，可以在 [Status]（状态）窗口和状态栏提示框中显示警告。

- 将跟踪存储器上溢检测设置为硬件断点条件

5.14.8 检测任务堆栈存取的违规操作:

如果下载的加载模块包含操作系统，会启用此功能。它检测是否已从另一个任务向相关任务堆栈尝试写入。

(1) 启动时的初始设置

启动时，会选中带有“Automatically set address areas at downloading”（下载时自动设置地址区）的复选框（用复选标记进行标记）。但是，由于地址信息不存在，所以该功能直到应用程序下载后才起作用。

(2) 检测到任务堆栈存取的违规操作时执行的操作

可以设置以下操作：

- 显示警告

通过在 [Configuration Properties]（配置属性）对话框的 [Exception Warning]（异常警告）页面上选中 [Task stack access violation]（任务堆栈存取的违规操作）复选框，可以在 [Status]（状态）窗口和状态栏提示框中显示警告。

- 将任务堆栈存取的违规操作检测设置为硬件断点条件
- 将任务堆栈存取的违规操作检测设置为跟踪点条件

5.14.9 设置任务堆栈区域

按照下述步骤设置任务堆栈区域。

1. 从 [Hardware Break] (硬件断点) 对话框执行以下操作:
 - A. 选中 [Hardware Break] (硬件断点) 页上的 [Exception] (异常) 复选框, 然后单击 [Detail] (详细信息) 按钮。

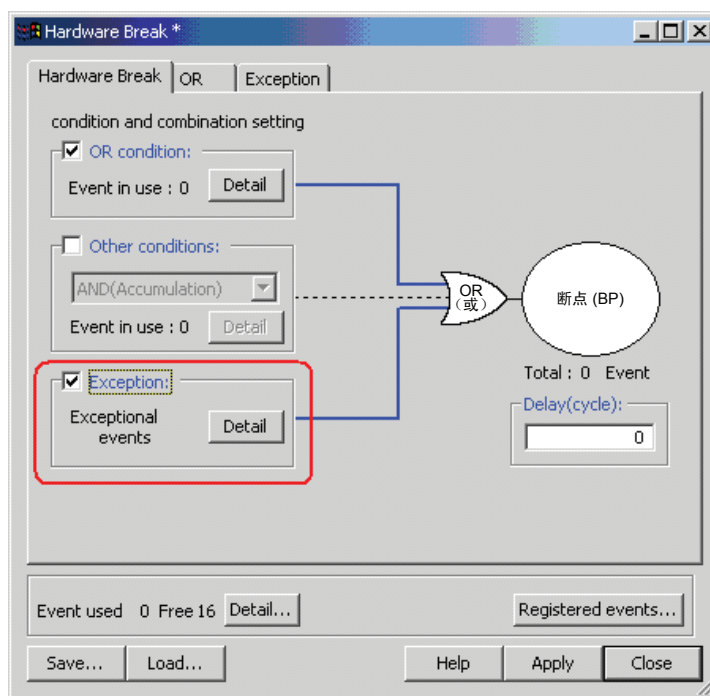


图 5.146 [Hardware Break] (硬件断点) 对话框

- B. 此时将出现如下所示的 [Exception]（异常）页面。单击 [Task stack access violation]（任务堆栈存取违规操作）复选框右边的 [Detail]（详细信息）按钮。

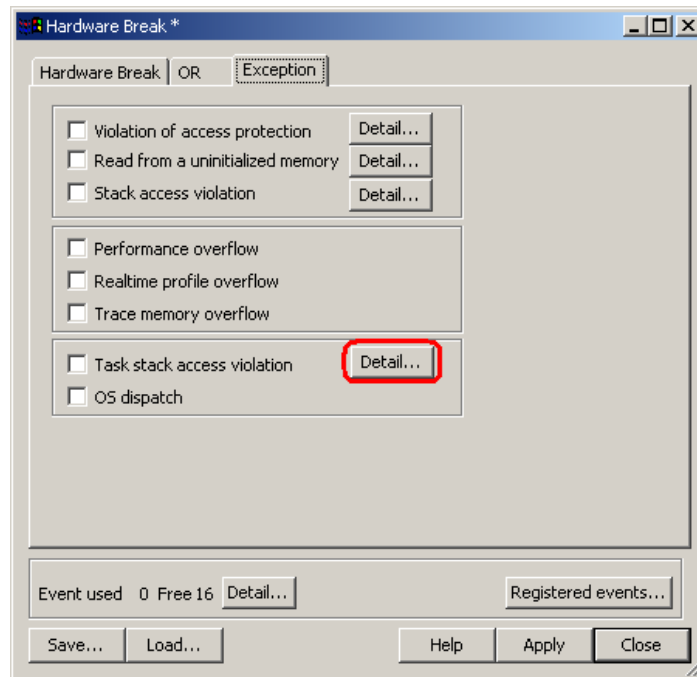


图 5.147 [Hardware Break]（硬件断点）对话框

- C. 此时将显示如下所示的 [Violation of task stack access]（任务堆栈存取的违规操作）对话框。要在下载程序时自动设置任务堆栈范围，请选中带有“Automatically set address areas at downloading”（下载时自动设置地址区）的复选框。

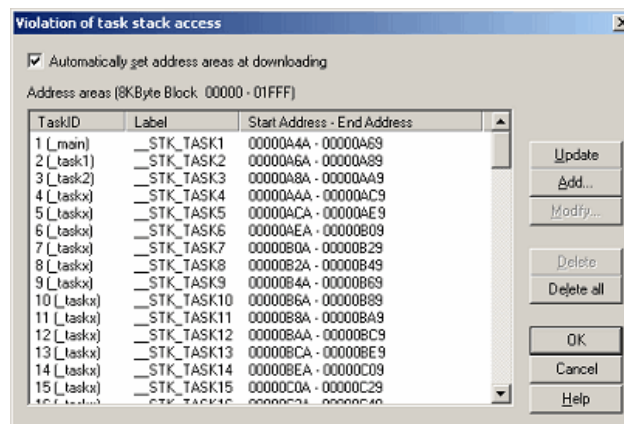


图 5.148 [Violation of task stack access]（任务堆栈存取的违规操作）对话框

- D. 单击 [Update]（更新）按钮，将自动设置任务堆栈范围。
- E. 要手动添加任务堆栈范围，请单击 [Add]（添加）按钮。此时将显示如下所示的 [Task stack access condition]（任务堆栈存取条件）对话框。指定任务堆栈的任意任务 ID 和地址范围。

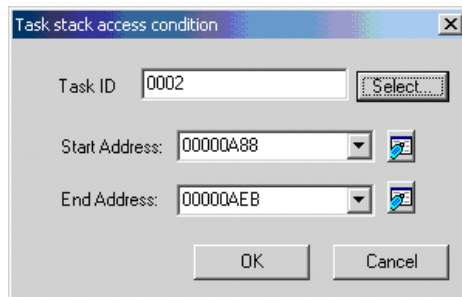


图 5.149 [Task stack access condition]（任务堆栈存取条件）对话框

- F. 添加的任务堆栈范围将显示在 [Violation of task stack access]（任务堆栈存取的违规操作）对话框的 [Address Areas]（地址区）列表中。
2. 从 [Trace conditions]（跟踪条件）对话框执行以下操作：

- A. 在 [Trace]（跟踪）页中的 [Trace Mode]（跟踪模式）下拉式列表中，选择 [Fill around TP]（围绕 TP 填充）。选中 [Exception]（异常）复选框，然后单击 [Detail]（详细信息）按钮。

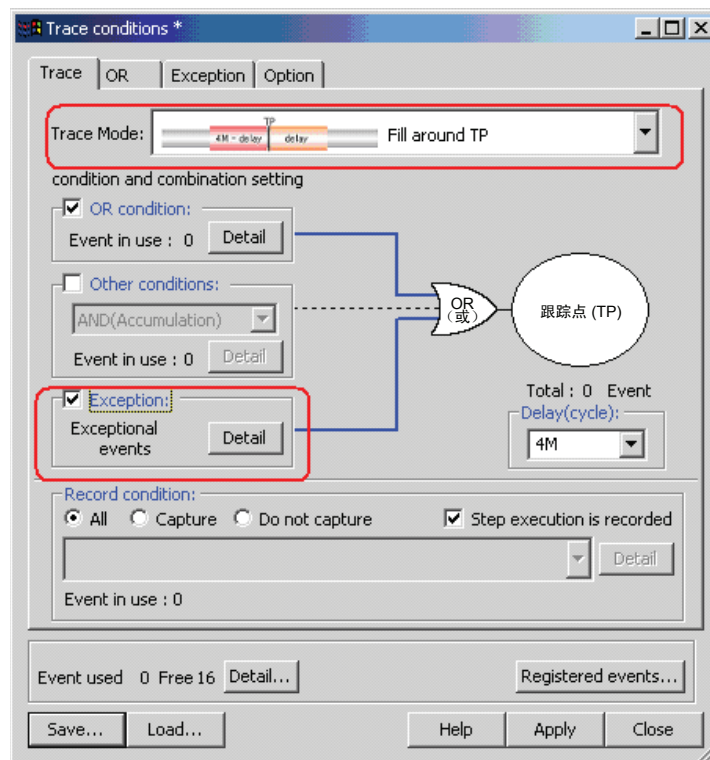


图 5.150 [Trace conditions]（跟踪条件）对话框

- B. 此时将出现如下所示的 [Exception]（异常）页面。单击 [Task stack access violation]（任务堆栈存取违规操作）复选框右边的 [Detail]（详细信息）按钮。

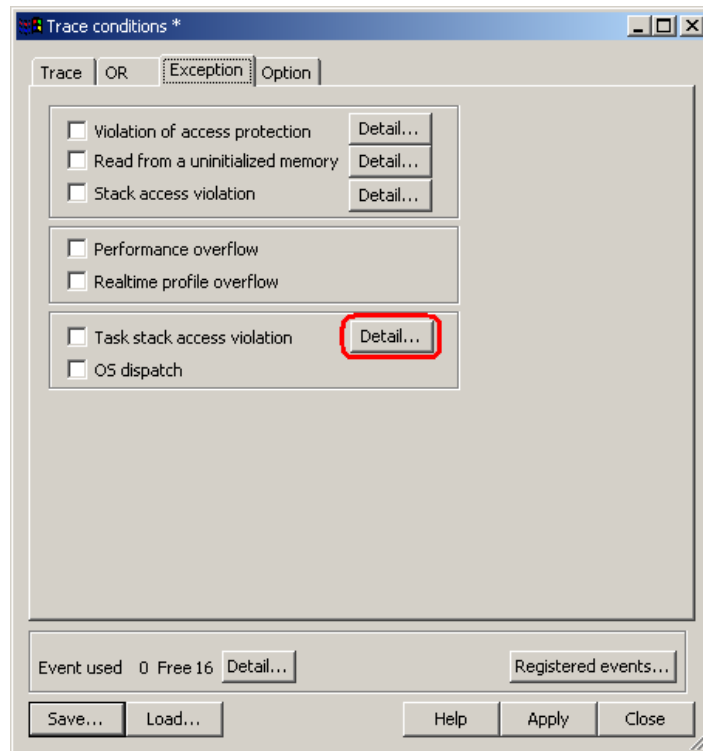


图 5.151 [Trace conditions]（跟踪条件）对话框

- C. 此时将显示 [Violation of task stack access]（任务堆栈存取的违规操作）对话框。其他步骤与从 [Hardware Break]（硬件断点）对话框打开该对话框时相同。

5.14.10 检测操作系统调度

如果下载的加载模块包含操作系统，则功能有效。它可检测是否发生任务调度。

(1) 检测到操作系统调度时执行的操作

可以设置以下操作：

- 显示警告

通过在 [Configuration properties]（配置属性）对话框的 [Exception Warning]（异常警告）页面上选中 [OS dispatch]（操作系统调度）复选框，可以在 [Status]（状态）窗口和状态栏提示框中显示警告。

- 将操作系统调度检测设置为硬件断点条件
- 将操作系统调度检测设置为跟踪点条件

5.15 使用启动 / 停止功能

仿真器在程序执行启动之前和暂停之后执行指定的例行用户程序。此功能可用于控制用户系统，使之与用户程序的执行和暂停同步。

5.15.1 打开 [Start/Stop Function Setting]（启动 / 停止功能设置）对话框

在 [Start/Stop function setting]（启动 / 停止功能设置）对话框中，可指定在用户程序开始执行之前和用户程序暂停执行之后执行的例行程序。

要打开 [Start/Stop function setting]（启动 / 停止功能设置）对话框，请从菜单中选择 [Setup]（设置）→ [Emulator]（仿真器）→ [Start/Stop function settin...]（启动 / 停止功能设置...）。

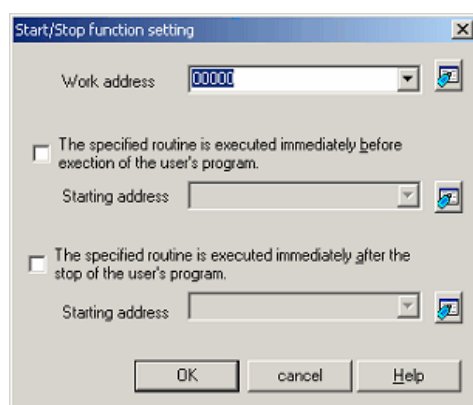


图 5.152 [Start/Stop function setting]（启动 / 停止功能设置）对话框

5.15.2 指定要执行的例行程序

可以分别指定在用户程序开始执行之前和用户程序停止执行之后执行的例行程序。

选中 [The specified routine is executed immediately before execution of the user's program]（在执行用户程序之前执行指定的例行程序）复选框后，在此复选框下面的 [Starting address]（起始地址）组合框中指定的例行程序会在开始用户程序执行之前执行。

选中 [The specified routine is executed immediately after the stop of the user's program]（在用户程序停止之后执行指定的例行程序。）复选框后，在此复选框下面的 [Starting address]（起始地址）组合框中指定的例行程序会在用户程序停止执行之后执行。

5.15.3 启动 / 停止功能的限制

启动 / 停止功能遵循以下限制：

- 启动 / 停止功能正在使用时，请勿使用下面列出的调试功能。
 - (a) 存储器设置和下载到指定例行程序的程序区。
 - (b) 在指定例行程序的程序区进行断点设置。
- 执行指定的例行程序期间，在仿真器端控制下使用中断堆栈所指向的 4 字节值。
- 指定例行程序中使用的通用寄存器和标志受下面的限制：

表 5.43 对寄存器和标志的限制

寄存器 / 标志的名称	限制
ISP 寄存器	指定例行程序结束后，必须将此寄存器的值还原为指定例行程序开始时它所具有的一个值。
U 标志	指定例行程序结束后，必须将此标志的值始终设置为 0。
I 标志	在指定例行程序执行期间不允许中断。

- 执行指定的例行程序时，下面列出的调试功能无效。
 - (a) 跟踪功能
 - (b) 与断点有关的功能
 - (c) RAM 监控功能
- 执行指定的例行程序时，始终禁止非可屏蔽中断。
- 下表显示在执行指定例行程序后用户程序开始运行时 MCU 将处于哪种状态。

表 5.44 用户程序开始时 MCU 状态

MCU 资源	状态
MCU 通用寄存器	这些寄存器处于用户程序上次停止时它们所处的状态，或者这些寄存器是用户在寄存器窗口设置的 MCU 寄存器。不会反映执行指定例行程序后更改的寄存器的内容。
MCU 空间中的存储器	将反映在执行指定例行程序后尝试的存储器存取。
MCU 外围功能	执行指定例行程序后，将继续 MCU 外围功能执行的操作。

5.15.4 对指定例行程序中所写语句的限制

指定例行程序中所写语句受下述限制。

- 如果指定例行程序需要使用某个堆栈，则一定要始终使用该用户堆栈。
- 要终止指定例行程序的进程，请编写一个返回子例程的指令。
- 确保指定例行程序执行的进程的一个会话在 10 ms 内终止。例如，如果时钟被关闭且在指定例行程序中保持为不活动，则仿真器可能变为无法控制程序执行。
- 在指定例行程序开始运行时存储在寄存器中的值将不确定。请确保在指定例行程序中初始化寄存器的值。

第 6 章 故障排除（针对错误采取操作）

6.1 故障解决流程图

图 6.1 显示了从仿真器电源激活到仿真调试器启动这一过程中的故障解决流程图。请在用户系统断开连接后查看此流程图。有关最新常见问题，请访问瑞萨工具主页。

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

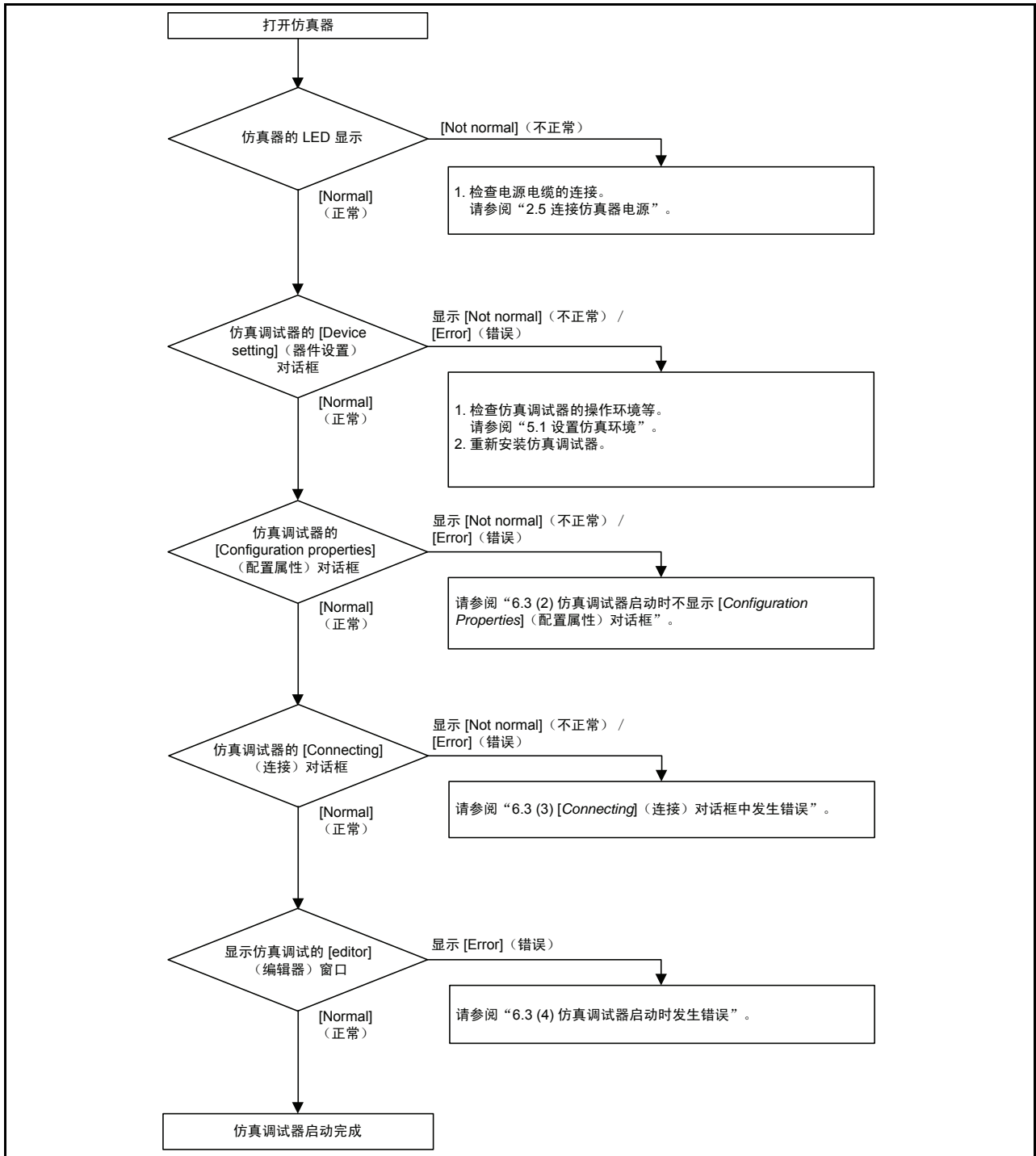


图 6.1 故障解决流程图

6.2 自检错误

如果自检中出现错误，请检查以下内容。

- (1) 重新检查 E100 仿真器主部件与 MCU 单元的连接。
- (2) 重新下载正确的固件。
- (3) 检查调试器软件的自检错误日志，并参考其中的说明。（请参阅图 6.2）

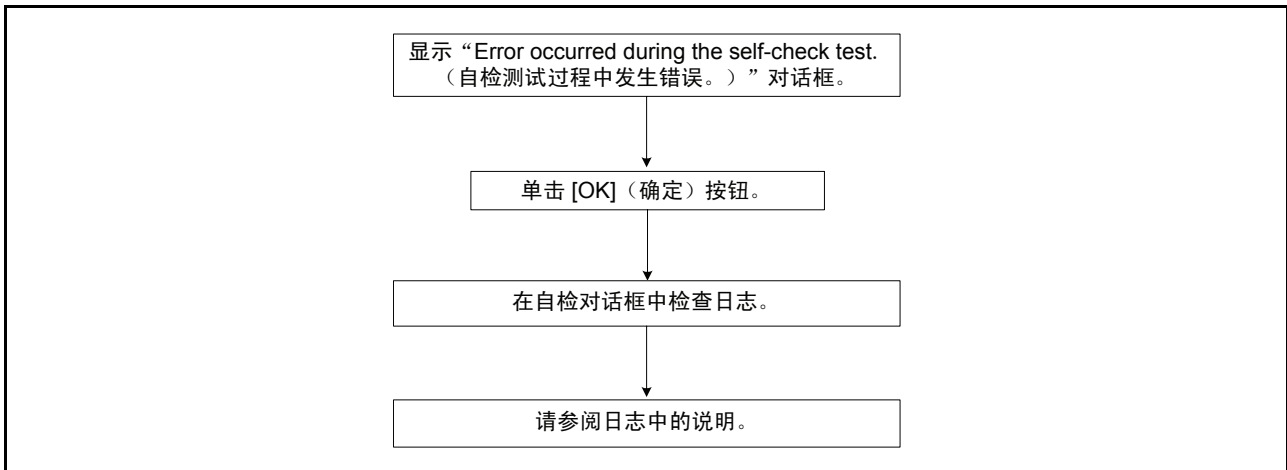


图 6.2 自检错误检查流程图

重要事项

关于自检的注意事项：

- 请确保在未连接到转换器板和用户系统的情况下执行自检。
- 如果自检结果不正常（目标状态错误除外），则产品可能已损坏。请联系当地的经销商。

6.3 调试器启动时的错误

(1) E100 的 LED 显示不正常

表 6.1 LED 显示不正常时的错误检查点

错误	与用户系统的连接	检查点
SAFE LED（安全指示灯）保持点亮。	-	检查是否已连接电源电缆。 请参阅第 11 页上的 2.4 连接主机。
SAFE LED（安全指示灯）不点亮。	-	重新检查 E100 与本产品的连接。 请参阅第 10 页上的 2.3 将 MCU 单元连接到 E100 仿真器主部件或断开它们之间的连接。
目标状态 POWER LED（电源指示灯）不点亮。	已连接	检查是否已正确向用户系统供电（Vcc 和 GND），且用户系统是否已正确接地。
目标状态 RESET LED（复位指示灯）不熄灭。	已连接	1. 检查是否拔起了用户系统的复位引脚。 2. 在没有用户系统的情况下使用仿真器时，查看是否已断开转换器板与仿真器的连接。

(2) 仿真调试器启动时不显示 [Configuration Properties]（配置属性）对话框

表 6.2 调试器启动时的错误检查点 1

错误	检查点
发生通信错误。 数据未发送到目标。	检查所有仿真调试器设置和接口电缆设置。 请参阅第 51 页上的第 4 章“准备调试”。

(3) [Connecting]（连接）对话框中发生的错误

表 6.3 调试器启动时的错误检查点 (1)

错误	检查点
无法正确构建用户系统。	1. 下载正确的固件。 请参阅第 51 页上的第 4 章“准备调试”。 2. 重新检查 E100 与本产品的连接。 请参阅第 10 页上的 2.3 将 MCU 单元连接到 E100 仿真器主部件或断开它们之间的连接。
仿真器的版本与目标中的固件版本不同。	下载正确的固件。 请参阅第 51 页上的第 4 章“准备调试”。
目标 MCU 处于复位状态。	1. 检查是否拔起了用户系统的复位引脚。 2. 检查用户系统的复位引脚是否已从“L”改为“H”级。
无法复位目标 MCU。	1. 如果用户系统的复位电路有一个看门狗定时器，请禁用该看门狗定时器。 2. 检查是否已正确向用户系统供电，且用户系统是否已正确接地。
目标处于“HOLD”（保持）状态。	MCU 处于停止模式或等待模式下。复位 MCU 或者使用中断取消该模式。 请参阅 MCU 规格。

表 6.3 调试器启动时的错误检查点 (2)

错误	检查点
目标时钟停止。	如果该时钟是从外部振荡器提供的，请检查用户系统中的振荡器电路是否振荡正常。
目标 MCU 接收不到供电。	检查是否已正确向用户系统供电，且用户系统是否已正确接地。

(4) 仿真调试器启动时发生错误

表 6.4 调试器启动时的错误检查点

错误	检查点
无法控制目标 MCU。	<ol style="list-style-type: none"> 1. 检查用户系统上安装的 NQPACK 等是否焊接良好。 2. 检查连接器是否正确安装到用户系统。

6.4 如何请求支持

检查完第 6 章“故障排除（针对错误采取操作）”中的各项后，填写从以下 URL 下载的文本文件，然后将信息发送给当地的经销商。

<http://tool-support.renesas.com/eng/toolnews/registration/support.txt>

便于快速回复，请指定以下信息：

(1) 操作环境

- 工作电压：_____ [V]
- 工作频率：_____ [MHz]
- 向 MCU 提供的时钟： 内部振荡器 / 外部振荡器

(2) 情况

- 仿真调试器启动 / 不启动
- 在自检中检测到 / 未检测到错误
- 错误频率：始终 / 经常 ()

(3) 问题

第 7 章 硬件规格

本章介绍本产品的规格。

7.1 目标 MCU 规格

表 7.1 列出可使用本产品调试的目标 MCU 的规格。

表 7.1 R0E530640MCU00 的目标 MCU 的规格

项目	描述
适用 MCU	M16C/60 系列 M16C/64 组 MCU，带有 512 KB 或更低容量 ROM
评估 MCU	R5F650MNFG-EVA ROM 大小：8KB+16KB+512KB，RAM 大小：31KB
适用 MCU 模式	单片模式、存储器扩展模式、微处理器模式
最大 ROM/RAM 容量	1. 内部闪存 ROM：536 KB 0E000h~0FFFFh：数据闪存 10000h~13FFFh：程序 1 ROM 80000h~FFFFFFh：程序 2 ROM 2. 内部 RAM：31 KB 00400h~043FFh
电源电压	Vcc1=Vcc2：2.7 ~ 5.5V
工作电压 / 频率	电源电压：2.7 ~ 5.5V 25MHz（带 PLL）

7.2 实际 MCU 与仿真器的不同点

下面显示了实际 MCU 与仿真器的不同点。当使用本产品调试 MCU 时，一定要注意以下事项。

重要事项

关于实际 MCU 与仿真器之间的不同点的注意事项：

- 下面列出了仿真器系统操作与实际 MCU 操作之间的不同点。
 - (1) 复位条件
将启动时间（0.2 Vcc 到 0.8 Vcc）设置为 1 μ s 或更少。
 - (2) 上电时 MCU 内部资源数据的初始值
 - (3) 复位解除后的中断堆栈指针 (ISP)
 - (4) 内部存储器（ROM 和 RAM）的容量
本产品的评估 MCU 的 RAM 为 31 KB (00400h~07FFFh)，闪存 ROM 为 8 KB (0E000h~0FFFFh)、16 KB (10000h~13FFFh) 和 512 KB (80000h~FFFFFFh)。
 - (5) 振荡器电路
在引脚 X_{IN} 和 X_{OUT} 之间连有振荡器的振荡器电路中，由于在评估 MCU 与用户系统之间使用了转换器板，因此不发生振荡。对于引脚 X_{CIN} 和 X_{COU}T，也是如此。
 - (6) A/D 转换
由于在评估 MCU 和用户系统之间有转换器板和其他设备，因此 A/D 转换器的特性与实际 MCU 的特性不同。

重要事项

关于 RESET# 输入的注意事项:

- 仅当执行用户程序时（仅当 E100 上部面板上的“RUN”（运行）状态 LED 点亮时），才接受向用户系统的引脚 RESET# 进行低电平输入。

关于电压检测电路的注意事项:

- 由于评估 ECU 与用户系统之间有节距转换器板等，因此本产品与实际 MCU 有所不同。应当使用实际 MCU 来执行电压检测电路的最终评估（电压下降检测中断、电压下降检测复位等）。

关于可屏蔽中断的注意事项:

- 即使未在执行用户程序（包括正在执行运行时调试的情况），评估 MCU 也会执行调试控制程序。因此，定时器和组件不会停止运行。如果在未执行用户程序时（包括正在执行运行时调试的情况）请求可屏蔽中断，由于仿真器禁止中断，将无法接受可屏蔽中断请求。开始执行用户程序后，会立即接受中断请求。
- 请注意，未执行用户程序时（包括执行运行时调试的情况）不会接受外围 I/O 中断。

关于 DMA 传送的注意事项:

- 对于本产品，用户程序通过循环程序停止到一个特定地址。因此，如果在用户程序停止时定时器或其他源生成了 DMA 请求，则会执行 DMA 传送。但是，请注意这样一个事实：用户程序停止时可能无法正确执行 DMA 传送。另外请注意，即使用户程序已停止，下面的寄存器也会改为生成 DMA 传送（如上文所述）。
 (1) DMA0 传送计数寄存器 TCR0 (2) DMA1 传送计数寄存器 TCR1
 (3) DMA2 传送计数寄存器 TCR2 (4) DMA3 传送计数寄存器 TCR3

关于最终评估的注意事项:

- 确保使用评估 MCU 来评估系统。在开始大批量生产之前，使用 CS（商业样本）版本 MCU 来评估系统并做出最终确认。

7.3 连接图

7.3.1 R0E530640MCU00 的连接图

图 7.1 显示了 R0E530640MCU00 的连接图。此连接图主要显示要连接到用户系统的电路。将忽略未连接到用户系统（例如仿真器的控制系统）的电路。

表 7.2 显示了本产品的 IC 电特性（供参考）。

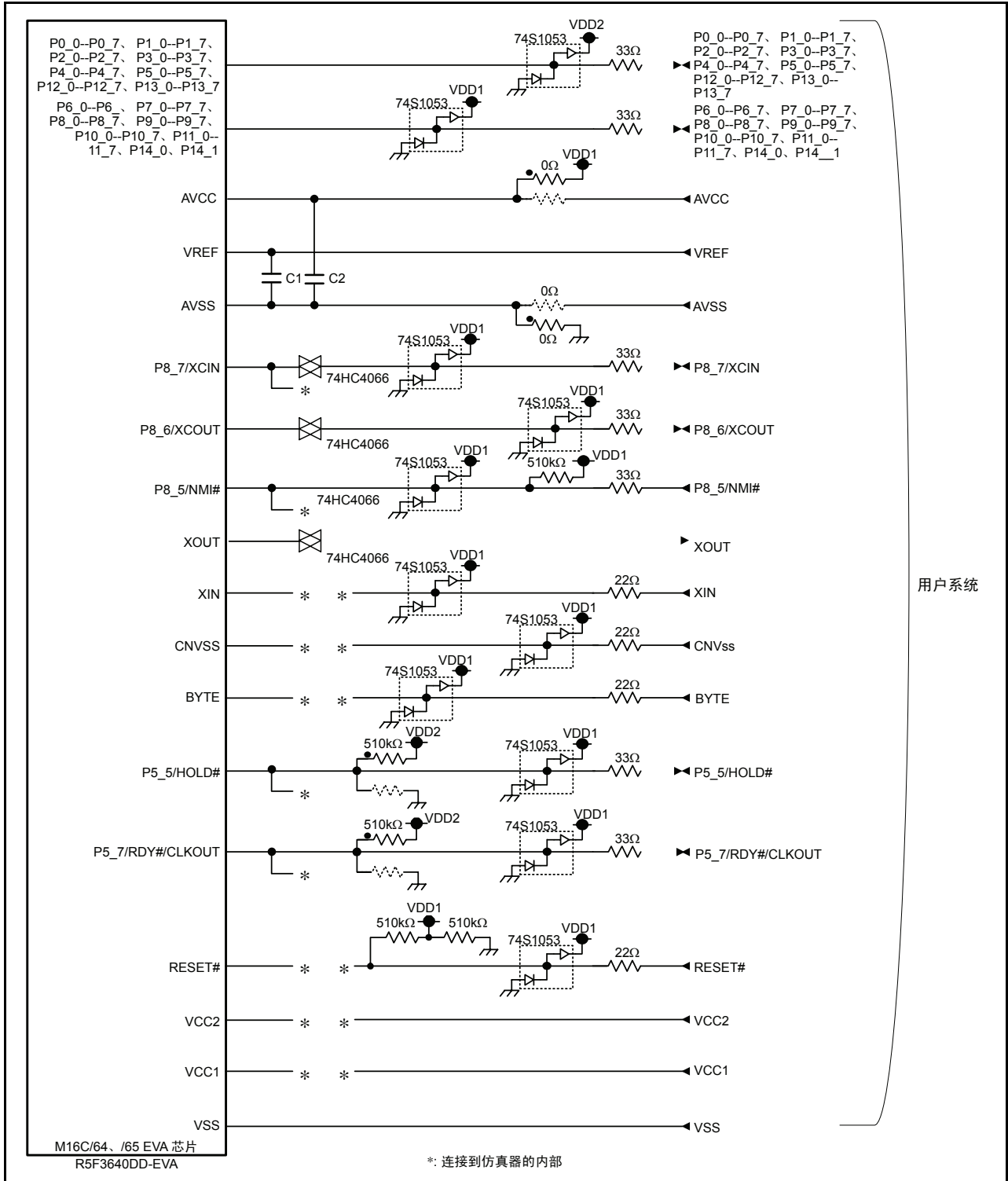


图 7.1 连接图

表 7.2 74HC4066 的电特性

符号	项目	情况	标准值			单位
			最小值	标准值	最大值	
		Vcc				
R _{ON}	ON 阻抗	4.5V	-	96	170	Ω
Δ R _{ON}	ON 阻抗差	4.5V	-	10	-	
I _{OFF}	漏电电流 (关闭)	12.0V	-	-	± 100	nA
I _{Iz}	漏电电流 (开启, 输出: 开路)	12.0V	-	-	± 100	

7.4 外部尺寸

7.4.1 E100 仿真器的外部尺寸

图 7.2 显示了 E100 仿真器的外部尺寸。

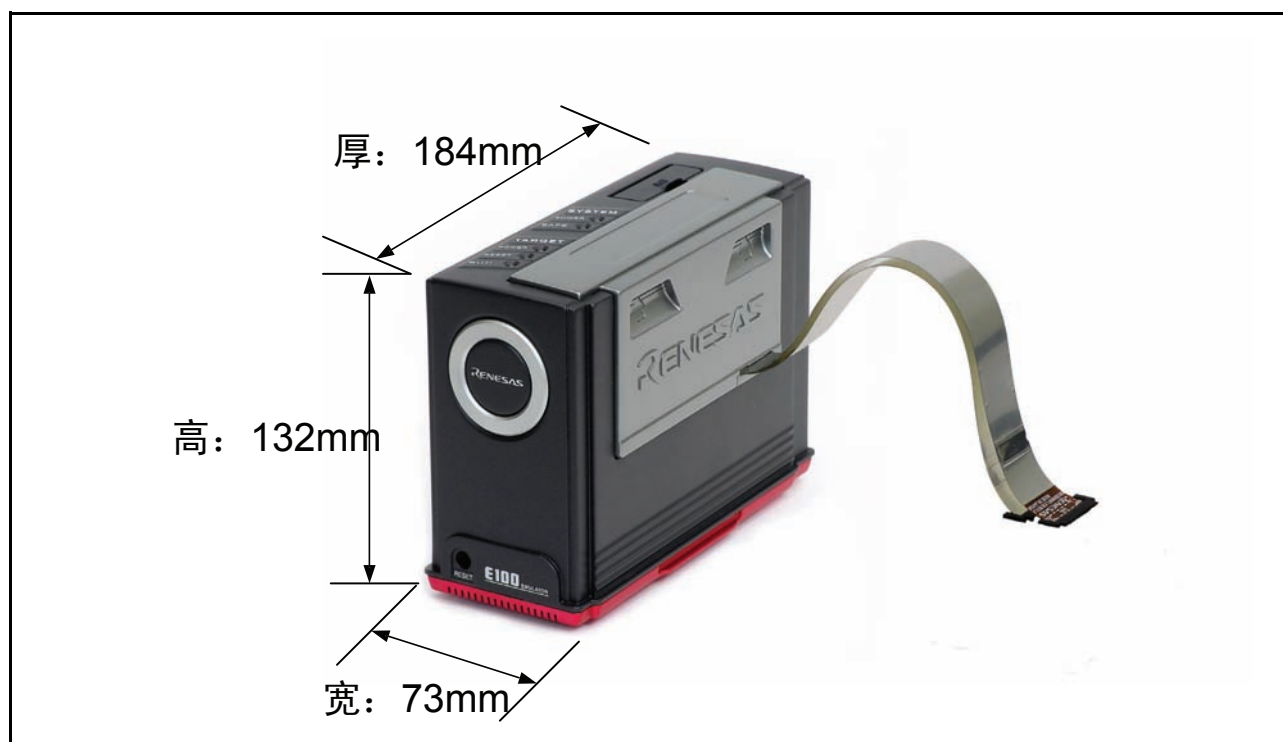


图 7.2 E100 仿真器的外部尺寸

7.4.2 转换器板 R0E0100TNPJ00 的外部尺寸

图 7.3 显示了 100 引脚 0.65mm 节距 QFP 的转换器板 R0E0100TNPJ00 的外部尺寸和焊脚尺寸示例。

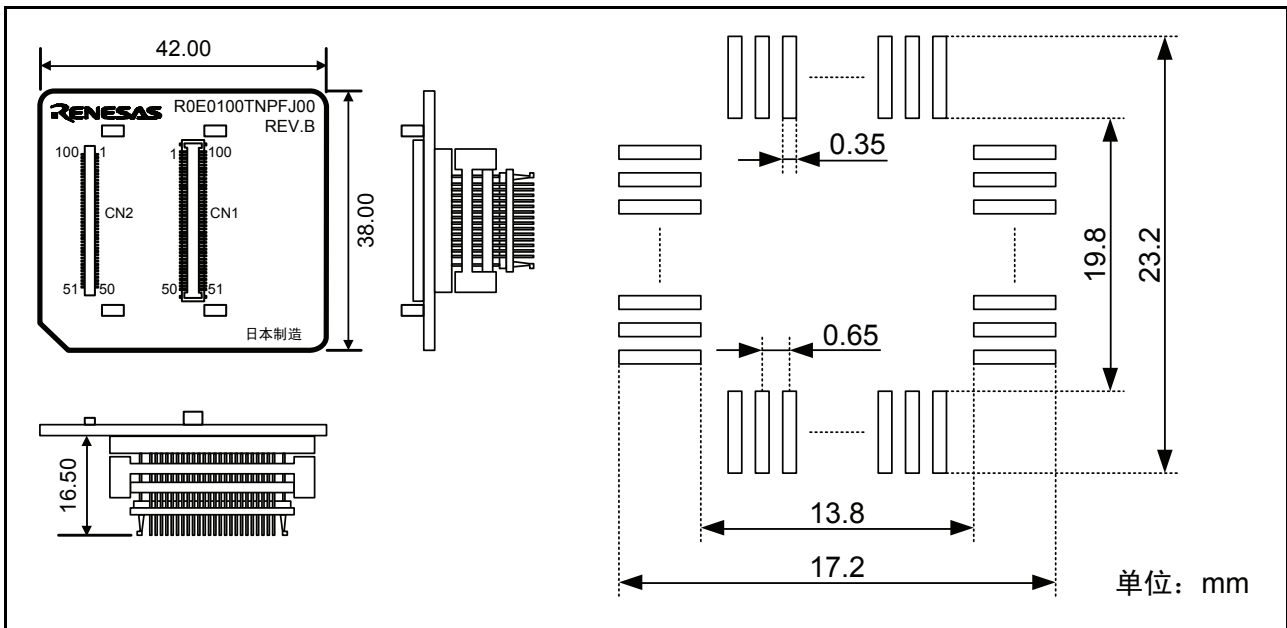


图 7.3 R0E0100TNPJ00 的外部尺寸和焊脚尺寸示例

7.4.3 转换器板 R0E0100TNPFK00 的外部尺寸

图 7.4 显示了 100 引脚 0.5mm 节距 LQFP 的转换器板 R0E0100TNPFK00 的外部尺寸和焊脚尺寸示例。

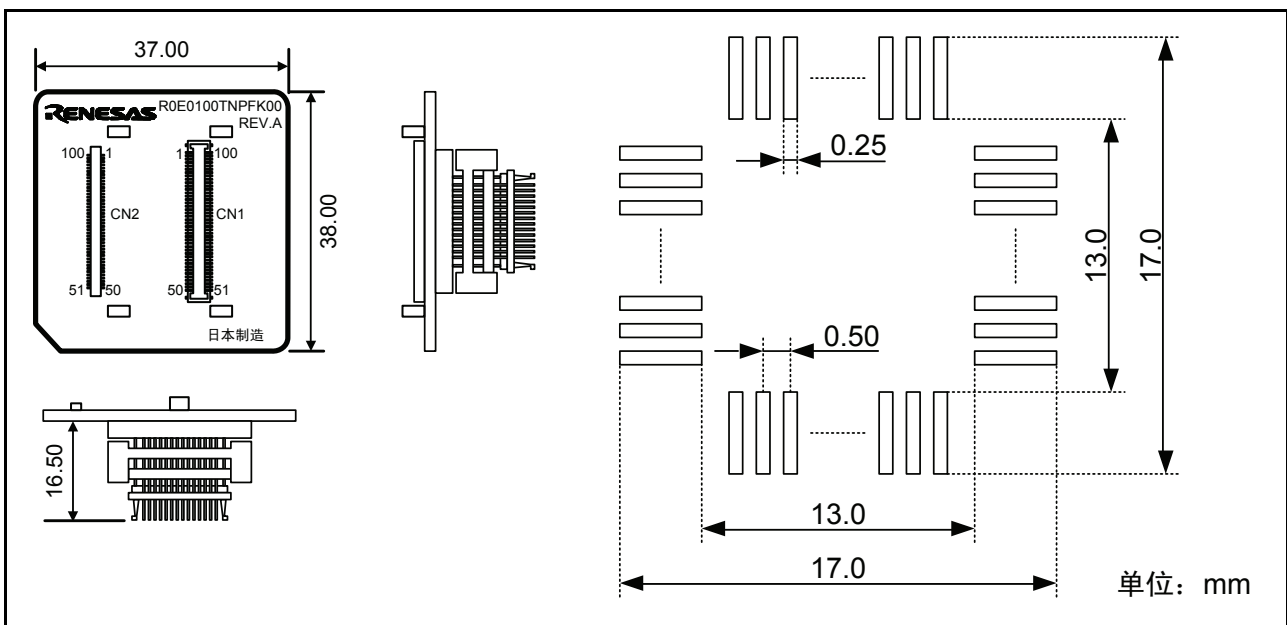


图 7.4 R0E0100TNPFK00 的外部尺寸和焊脚尺寸示例

7.5 关于使用本产品的注意事项

下面列出了关于使用本产品的注意事项。当使用仿真器调试 MCU 时，请注意以下事项。

重要事项

关于仿真调试器版本的注意事项：

- 确保将本产品与以下仿真调试器一起使用。
- M16C R8C E100 仿真调试器 V.1.00 Release 00 或更高版本

关于下载固件的注意事项：

- 第一次使用本产品之前，需要下载专用固件（E100 闪存中安装的仿真器控制软件）。如果需要在调试器启动时下载，将显示一条消息。请根据该消息下载固件。
- 下载固件过程中不要关闭电源。如果关闭了电源，本产品将无法启动。如果电源意外关闭，请重新下载固件。
- 请在未连接用户系统时下载固件。

关于自检的注意事项：

- 如果自检结果不正常（用户系统错误除外），则产品可能已损坏。那么，请联系当地经销商。
- 请在未连接用户系统的情况下运行自检。

关于退出仿真调试器的注意事项：

- 要重新启动仿真调试器，请始终先关闭仿真器电源，然后再将其打开。

关于 MCU 状态显示的注意事项：

- 用户在仿真调试器的 [MCU Setting]（MCU 设置）对话框的 [MCU] 选项卡中查看的 MCU 状态显示了用户系统的引脚级别。请确保根据所使用的模式指定正确的引脚级别。

关于处理器模式寄存器 0 的注意事项：

- 不要将 BCLK 输出禁止位 (PM07) 设置为 “1b”（不输出）。如果进行了这样的设置，评估 MCU 中的内部时钟将停止，仿真器将无法正常工作。

关于向 MCU 提供的时钟的注意事项：

- 向评估 MCU 提供的时钟是在仿真调试器的 [Init]（初始化）对话框的 [Emulator]（仿真器）选项卡中选定的。
 - (1) 选择 “Emulator”（仿真器）时：
提供由 MCU 单元上的振荡器电路板生成的时钟。无论用户系统时钟的状态和用户程序执行的状态如何，都会持续提供该时钟。
 - (2) 选择 “User”（用户）时：
提供由用户系统中振荡器生成的时钟。它取决于用户系统的振荡状态（开 / 关）。
 - (3) 选择 “Generated”（已生成）时：
提供由 E100 中的专用电路生成的时钟。无论用户系统时钟的状态和用户程序执行的状态如何，都会持续提供该时钟。

重要事项

关于停止和等待模式的注意事项：

- 不要单步执行转移到停止或等待模式的指令。这可能导致通信错误。

关于看门狗功能的注意事项：

- 如果用户系统的复位电路有一个看门狗定时器，请在使用仿真器时禁用它。

关于保护寄存器的注意事项：

- 保护寄存器 PRCR 的第 2 位 (PRC2) 用于写入端口 P9 方向寄存器和 SI/Oi 控制寄存器，使用以下步骤更改该位时，不会取消保护。
 - (1) 单步执行将 PRC2 设置为 “1” 的指令。
 - (2) 在将 PRC2 设置为 “1” 的指令与设置端口 P9 方向寄存器或 SI/Oi 控制寄存器的点之间设置一个断点。
 - (3) 通过 [Memory] (存储器) 窗口或 [Command Line] (命令行) 窗口将 PRC2 设置为 “1”。

关于访问禁止区域的注意事项：

- 不能使用内部保留区域。将忽略这些区域的写入信号，不会对读取的值进行定义。

关于中断的注意事项：

- 在仿真调试器的程序窗口中显示断点的区域中，显示了以下断点。
 - (1) 软件断点
这是一个用于生成 BRK 中断的调试函数，方法是将指定地址处的指令更改为 BRK 指令 (00h)，以便在系统执行指定地址处的指令之前立即中断程序。将不执行预置地址处的指令。
 - (2) 硬件断点
这是一个用于中断程序的调试函数，方法是：将对指定地址处的指令执行进行的检测设置为中断事件。该程序将在执行指定地址处的指令之后中断。
 - (3) 异常事件
这是一个调试函数，用于通过用户程序的异常操作或每个函数的测量计数器的上溢等停止程序。

关于软件断点的注意事项：

- BRK 指令仅可用于仿真器。不能在用户程序中使用它。由于 BRK 指令中断向量由仿真器系统使用，因此读取的数据与期望值不同。
- 在用户程序执行过程中，在 MCU 的内部 ROM 区域中既不能设置也不能取消软件断点，但在 MCU 的内部 RAM 中可以设置或取消它。

重要事项

关于用户系统的电源的注意事项：

- 引脚 Vcc1 和 Vcc2 连接到用户系统是为了观察电压。因此，用户系统的电源不是从仿真器提供的。请将系统设计成单独为用户系统供电。
- 用户系统的电压应当如下所示。

$$2.7\text{ V} \leq V_{cc1} = V_{cc2} \leq 5.5\text{ V}$$

关于 MCU 的内部闪存 ROM 的注意事项：

- 由于 MCU 的内部闪存 ROM 的写入 / 擦除数有限，因此必须在其服务期结束时将其更换掉。
- 如果在下载程序时经常出现以下错误，请更换 MCU 板。
 - (1) “Flash ROM erase error occurred ERROR (16258)”（发生闪存 ROM 擦除错误，错误编号 16258）
 - (2) “Flash ROM verify error occurred ERROR (16259)”（发生闪存 ROM 验证错误，错误编号 16259）

在 CPU 重写模式下进行调试时的注意事项：

- 在 CPU 重写模式下调试 M16C/60 系列 MCU 时，不要更改闪存的块 0 区域 (FF000h~FFFFh)。否则，仿真器将变为无法控制。
- 如果在仿真调试器的 [Configuration properties]（控制属性）对话框的 [System]（系统）选项卡中选中 “Debug the program using CPU Rewrite Mode”（使用 CPU 重写模式调试程序），则不能使用以下功能。
 - (1) 在内部 ROM 区域中设置软件断点
 - (2) 在内部 ROM 区域中执行 COME
- 在 CPU 重写模式和擦除挂起模式下，不要停止程序。并且，不要单步执行转移到 CPU 重写模式或擦除挂起模式的指令。在 CPU 重写模式和擦除挂起模式下，仿真器将变为无法控制。
- 要在执行 CPU 重写后引用数据，请在重写控制程序区域之外的位置停止该程序，并使用 [Memory]（存储器）窗口等。
- 由于以下中断向量由仿真器系统使用，因此读取的数据与期望值不同。
 - 单步执行 (FFFECh~FFFEFh)
- 由于无法调试用户引导函数，因此不要进入用户引导模式。

关于访问地址 00000h 和 00001h 的注意事项：

- 对于 M16C/60 系列 MCU，当生成可屏蔽中断时，会读出存储在地址 00000h 和 00001h 中的中断数据（中断编号和中断请求级别）。同样，在读出地址 00000h 或 00001h 时，会清除中断请求位。因此，执行地址 00000h 或 00001h 读出指令时，或者因程序失控而读出地址 00000h 或 00001h 时，由于已经清除了启用的最高优先级中断系数的请求位，该地址将出现故障：收到中断请求时不执行中断。
 对于此故障，在生成到地址 00000h 或 00001h 的读出（不包括中断）时，将显示扩展监控器窗口。此时，请检查用户程序。其中可能有错误的访问。

重要事项

存储器空间扩展功能（4M 字节模式）：

- 使用存储器空间扩展功能（4M 字节模式）时，评估 MCU 访问的存储器因每项设置而有所不同。请参考下表。

使用存储器空间扩展功能（4M 字节模式）时评估 MCU 的访问区域

处理器模式	PM13*1	OFS*2	目标 MCU 的访问区域	存储体 0~5	存储体 6	存储体 7
存储器扩展模式	1	0	40000h-7FFFFh	EXT*3	EXT	MAP*4
		1	40000h-7FFFFh	EXT	EXT	MAP
	0	0	40000h-7FFFFh			
			80000h-BFFFFh	EXT	EXT	MAP
		1	40000h-7FFFFh	EXT	EXT	MAP
			80000h-BFFFFh	EXT	EXT	---
微处理器模式	---	0	40000h-7FFFFh	EXT	EXT	MAP
			80000h-BFFFFh	EXT	EXT	---
			C0000h-FFFFFh	---	---	MAP
		1	40000h-7FFFFh	EXT	EXT	MAP
			80000h-BFFFFh	EXT	EXT	---

*1: 显示地址 00005h 的第 3 位

*2: 显示地址 0000Bh 的第 2 位

*3: 显示用户系统上的存储器存取

*4: MAP 根据 [Configuration properties]（配置属性）对话框的 [Memory map]（存储器映像）页上的 [Emulation Memory Allocation]（仿真存储器分配）设置显示区域访问。

第 8 章 维修和保证

本章介绍如何执行维修、保修信息、维修提供和申请维修的过程。

8.1 用户注册

在购买我们的产品时，一定要注册为用户。有关用户注册的信息，请参考本用户手册的**用户注册**（第 6 页）。

8.2 维修

- (1) 如果在仿真系统的任何设备上有灰尘堆积，请用干的软布将其擦去。不要使用稀释剂或其他溶剂，这些化学品可能导致设备表面涂层剥离。
- (2) 如果长时间不使用本产品，为安全起见，请将电源电缆与电源断开。

8.3 保证

在遵守本用户手册“重要事项”和“安全事项”规定且正确使用的前提下，如果本产品在购买后一年内发生故障，我们将免费为您维修或替换。不过请注意，如果产品的故障是下列原因之一所造成，我们将在额外收费的情况下维修或以新产品替换：

- 产品的误用、滥用或者在其他异常条件下使用
- 私自修理、改造、维修或者其他行为
- 用户系统不完善或者误使用
- 火灾、地震或者其他意外事故

发生上述情况时，请联系当地的经销商。如果产品是租用的，请联系租赁公司或所有者。

8.4 维修提供

- (1) 超出保修范围的维修

如果产品在购买一年后损坏，将不提供保修。

- (2) 超出保修范围的更换

如果产品在下列任何情况下发生故障，将通过更换整个产品（而不是维修）来解决该故障，或者建议您购买一个新产品，具体取决于故障的严重程度。

- 机械部分出现故障或破损
- 涂层表面部分出现裂纹、剥离或生锈
- 塑料部分出现裂纹或破损
- 使用不正确或未经授权的维修或修改导致的故障或损坏
- 由于过电压、过电流或电源供电不足导致的电路严重损坏
- 印刷电路板损坏或电路烧坏
- 修理费用高于更换费用的众多故障
- 无法定位或无法识别的故障

(3) 修理期满

某种产品型号停产一年后，可能无法对该型号的产品进行维修。

(4) 产品送往修理的运费

将产品返回给我们进行维修的运输费将由用户自己支付。

8.5 如何提出维修申请

填写本产品附带的维修申请表，将其与本产品一起送到当地的经销商处进行维修。请务必在修理申请单上填写尽可能详细的信息，以便于维修。

**注意**

关于运输产品的注意事项：



- 产品送往修理时，请使用购买本产品时附带的包装箱和缓冲材料，并明示谨慎处理，要求以精密装置对待。如果产品包装不全，则可能在运输过程中损坏。如果以塑料袋来包装产品，请务必使用本产品附送的导电聚乙烯塑料袋（通常为蓝色袋）。如果使用其他包，可能会因静电而使产品受损。

修订记录	R0E530640MCU00 用户手册
------	---------------------

Rev.	发行日	修订内容	
		页	修订处
1.00	2008.09.08	—	初版发行

**M16C/64 组的 E100 仿真器 MCU 单元
用户手册
R0E530640MCU00**

Publication Date: Rev1.00, Sep. 08, 2008

Published by: Sales Strategic Planning Div.
Renesas Technology Corp.

Edited by: Microcomputer Tool Development Department
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan



RENESAS SALES OFFICES

<http://www.renesas.com>

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

Renesas Technology America, Inc.

450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited

Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.

Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.

7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.

10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.

1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.

Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology 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: <603> 7955-9390, Fax: <603> 7955-9510



R0E530640MCU00



瑞萨电子株式会社

RCJ10J0082-0100