

To our customers,

---

## Old Company Name in Catalogs and Other Documents

---

On April 1<sup>st</sup>, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: <http://www.renesas.com>

April 1<sup>st</sup>, 2010  
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (<http://www.renesas.com>)

Send any inquiries to <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.

# M3T-PD45 V.1.00

User's Manual

Emulator Debugger for 4500 Series

Active X, Microsoft, MS-DOS, Visual Basic, Visual C++, Windows and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

IBM and AT are registered trademarks of International Business Machines Corporation.

Intel and Pentium are registered trademarks of Intel Corporation.

Adobe, Acrobat, and Acrobat Reader are trademarks of Adobe Systems Incorporated.

All other brand and product names are trademarks, registered trademarks or service marks of their respective holders.

#### **Keep safety first in your circuit designs!**

- Renesas Technology Corporation and Renesas Solutions Corporation put the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

#### **Notes regarding these materials**

- These materials are intended as a reference to assist our customers in the selection of the Renesas Technology product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation, Renesas Solutions Corporation or a third party.
- Renesas Technology Corporation and Renesas Solutions Corporation assume no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation and Renesas Solutions Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation, Renesas Solutions Corporation or an authorized Renesas Technology product distributor for the latest product information before purchasing a product listed herein. The information described here may contain technical inaccuracies or typographical errors. Renesas Technology Corporation and Renesas Solutions Corporation assume no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors. Please also pay attention to information published by Renesas Technology Corporation and Renesas Solutions Corporation by various means, including the Renesas home page (<http://www.renesas.com>).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation and Renesas Solutions Corporation assume no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Renesas Technology semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation, Renesas Solutions Corporation or an authorized Renesas Technology product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Renesas Technology Corporation and Renesas Solutions Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination. Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Renesas Technology Corporation or Renesas Solutions Corporation for further details on these materials or the products contained therein.

For inquiries about the contents of this document or product, fill in the text file the installer generates in the following directory and email to your local distributor.

¥SUPPORT¥Product-name¥SUPPORT.TXT

Renesas Tools Homepage <http://www.renesas.com/en/tools>

In this User's Manual, the emulator debugger "M3T-PD45" is represented as "PD45" respectively. Please replace them with the corresponding one, when you read them.

## **Preface**

PD45 is an emulator debugger for PC4504 system. It controls PC4504 emulator from Windows on a PC, and it helps debugging of application program and target system. This user's manual describes the PD45's features, functions, setting up and operational procedures.

## **Rights to the Program**

The right to use the program is granted according to provisions under a software license agreement. The PD45 program can only be used for the purposes of product development by the user, and cannot be used for any other purposes.

Note also that the information in this manual does not convey any guarantee or license for the use of software.



# Contents

## Introduction 1

|  |          |
|--|----------|
| <b>1.PD45 Introduction</b>               | <b>3</b> |
| <b>2. PD45 Features</b>                  | <b>4</b> |
| 2.1 Multi-Windowing Function .....       | 4        |
| 2.2 Break Functions.....                 | 4        |
| 2.3 Real-Time Trace Function.....        | 4        |
| 2.4 Coverage Measurement Function.....   | 5        |
| 2.5 Time Measurement Function.....       | 5        |
| 2.6 Source Level Debugging Function..... | 5        |
| <b>3. PD45 Input and Output Files</b>    | <b>6</b> |
| 3.1 Input Files .....                    | 6        |
| 3.2 Output Files.....                    | 7        |

## Setup 9

|                         |           |
|-------------------------|-----------|
| <b>1. Setup</b>         | <b>11</b> |
| 1.1 Installation .....  | 11        |
| 1.2 Starting PD45 ..... | 11        |
| 1.3 PD45 Setup.....     | 12        |

## Window function 15

|                                   |           |
|-----------------------------------|-----------|
| <b>1. Window function of PD45</b> | <b>17</b> |
| 1.1 The PD45 Window .....         | 17        |
| 1.2 Program Window .....          | 22        |
| 1.3 Source Window.....            | 25        |
| 1.4 Register Window .....         | 27        |
| 1.5 Memory Window .....           | 28        |
| 1.6 Dump Window.....              | 30        |
| 1.7 ASM Watch Window .....        | 32        |
| 1.8 Script Window .....           | 34        |

|   |    |
|---|----|
| 1.9 Trace Window .....                              | 36 |
| 1.10 Coverage Window .....                          | 41 |
| 1.11 S/W Break Point Setting Dialog Box .....       | 43 |
| 1.12 H/W Break Point Setting Dialog Box .....       | 45 |
| 1.13 Trace Point Setting Dialog Box.....            | 47 |
| 1.14 Time Measurement point Setting Dialog Box..... | 48 |

## Basic operation

49

|  |           |
|--|-----------|
| <b>1. Loading and Displaying the Target Program</b>          | <b>51</b> |
| 1.1 Downloading .....  | 51        |
| 1.2 Uploading.....   | 52        |
| 1.3 Saving Results of Disassembly.....                       | 52        |
| 1.4 Continuing to Display a Selected Program Position .....  | 52        |
| 1.5 Changing the Program Display Position.....               | 53        |
| 1.6 Checking Source Programs in Other Directories.....       | 56        |
| 1.7 Displaying the Results of Disassembling .....            | 57        |
| <b>2. Starting and Stopping Target Program Execution</b>     | <b>58</b> |
| 2.1 Starting and Stopping.....                               | 58        |
| 2.2 Step Execution .....                                     | 60        |
| 2.3 Returning from Current to Calling Routine .....          | 61        |
| 2.4 Program Execution to Specified Location .....            | 61        |
| 2.5 Resetting the Program .....                              | 62        |
| <b>3. Software Breaks</b>                                    | <b>63</b> |
| 3.1 Enabling S/W Breaks.....                                 | 63        |
| 3.2 Opening the S/W Break Point Setting Dialog Box .....     | 64        |
| 3.3 Setting a Break Point .....                              | 64        |
| 3.4 Deleting a Break Point .....                             | 65        |
| 3.5 Temporarily Disabling Break Points .....                 | 66        |
| 3.6 Temporarily Enabling Break Points.....                   | 66        |
| 3.7 Setting a Break Point from Program (Source) Window ..... | 67        |
| 3.8 Setting Breakpoints from the Toolbar.....                | 67        |
| <b>4. Hardware Breaks(1)</b>                                 | <b>68</b> |
| 4.1 Enabling H/W Breaks.....                                 | 68        |
| 4.2 Opening the H/W Break Point Setting Dialog Box.....      | 69        |



|  |           |
|--|-----------|
| 4.3 Setting Address Breakpoints .....                          | 69        |
| 4.4 To set a trigger break point.....                          | 70        |
| 4.5 To combination break point.....                            | 71        |
| <b>5. Real Time Tracing</b>                                    | <b>72</b> |
| 5.1 To check bus data each cycle.....                          | 72        |
| 5.2 To check executed Instructions.....                        | 73        |
| 5.3 To check execution path at the source level.....           | 74        |
| 5.4 Changing the Real-Time Trace Position to be Saved .....    | 75        |
| <b>6. Time Measurement</b>                                     | <b>76</b> |
| 6.1 Checking Program Execution Time.....                       | 76        |
| 6.2 Measuring Execution Time of A Selected Address Range ..... | 77        |
| <b>7. Checking and Setting Register Data</b>                   | <b>79</b> |
| 7.1 Checking the Contents of Registers.....                    | 79        |
| 7.2 Setting the Contents of Registers .....                    | 79        |
| <b>8. Checking and Setting Memory Contents</b>                 | <b>80</b> |
| 8.1 Checking the Value at a Specified Address.....             | 80        |
| 8.2 To reference the contents of contiguous addresses.....     | 83        |
| 8.3 Setting Data at a Specified Address .....                  | 83        |
| <b>9. Script Commands</b>                                      | <b>85</b> |
| 9.1 Executing Script Commands .....                            | 85        |
| 9.2 Logging the Results of Executing Script Commands .....     | 86        |
| 9.3 Executing Script Commands in Batch Mode.....               | 88        |
| <b>10. Exiting PD45</b>  | <b>91</b> |
| 10.1 Exiting PD45.....   | 91        |
| <b>11. Miscellaneous</b>                                       | <b>92</b> |
| 11.1 Line Assemble .....                                       | 92        |
| 11.2 Searching for Character Strings in Target Program.....    | 93        |
| 11.3 Changing Window Proportions.....                          | 93        |
| 11.4 Displaying the Version of PD45.....                       | 94        |

## High-end Debugging

95

|   |           |
|---|-----------|
| <b>1. Hardware Breaks(2)</b>                | <b>97</b> |
| 1.1 To Break Program On Stack Overflow..... | 97        |

1.2 To Break Program On Real-time Trace Event..... 98

1.3 To Break Program Using a Timer ..... 98

**2. Coverage measurement 101**

2.1 Displaying Coverage Measurement Result in Percentage..... 101

2.2 Referencing Coverage Measurement Results at Source Level ..... 102

2.3 Referencing Coverage Measurement Results at Address Level ..... 103

2.4 To initialize the previous results of coverage measurements..... 103

**Reference 105**

**1. Table of Script Commands 107**

1.1 Input Format..... 107

1.2 Tables of Commands..... 108

**2. Writing Script Files 119**

2.1 Structural Elements of a Script File..... 119

2.2 Writing Expressions ..... 122

**3. Error Messages 126**

**Index 135**

# **Introduction**



# 1. PD45 Introduction

The PD45 is an emulator debugger for Windows, which controls the **PC4504** Emulators for 4500 series microcomputer.

## 2. PD45 Features

### 2.1 Multi-Windowing Function

PD45 supports multiple overlapping windows, enabling you to simultaneously view a wide range of data. Each window contains menus and buttons, allowing commands to be executed by clicking with the mouse.

### 2.2 Break Functions

PD45 has the following two break functions:

#### 2.2.1 Software Break

Software breaks allow program execution to be stopped after the command at the specified address. The point at which this break occurs is called the software breakpoint. You can set software breakpoints from the S/W Break Point Setting dialog box. You can also easily set them from the Program Window and Source Window. PD45 supports up to 8 software breakpoints. You can set and read in the software breakpoint file via the S/W Break Point Setting dialog box.

#### 2.2.2 Hardware Break

Hardware break causes the target program to break when an event occurs in the target MCU. The target program can be made to break when execution of a specified address (Fetch) is detected or the rising or falling edge of a signal input from an external trace cable is detected. The former is called an address break and the latter is called a trigger break. PD45 allows you to set two address breakpoints and one trigger breakpoint. A combinatorial condition of these three breakpoints can also be set. In addition to the above breakpoints, you can specify a break for failure such as a stack overflow or a break based on a timer count. Use the H/W Breakpoint Setup dialog box to set these break conditions.

### 2.3 Real-Time Trace Function

PD45 supports a real-time trace function that records the execution history of the target program for 4K cycles. The trace point can be set using the Trace Point Setting dialog box. Trace points can be set using the same conditions as for hardware breakpoints.

## 2.4 Coverage Measurement Function

A coverage measurement function refers to recording the addresses executed (accessed) by the target program (C0 coverage). It allows you to check unexecuted addresses after program execution has stopped. Using this coverage measurement function in a test process, it is possible to check the test items that have been omitted.

## 2.5 Time Measurement Function

A time measurement function refers to measuring the execution time of the target program. It allows you to measure the execution time of a given processing routine.

## 2.6 Source Level Debugging Function

You can display source files and perform source level debugging such as setting breakpoints in the source lines and performing step execution. PD45 allows debugging at the assembly language level.

- You can use the Program Window and Source Window to view source files.
- You can use the ASM Watch Window to view ZXY symbols and bit symbols in assembler source files.
- You can use labels and symbols to specify the addresses of breakpoints.

## 3. PD45 Input and Output Files

### 3.1 Input Files

PD45 processes the following input files.

#### 3.1.1 Intel HEX-format Files

The Intel HEX-format files store machine language data. These files, which are generated by the asm45 absolute assembler, take the attribute ".hex".

#### 3.1.2 Symbol Files

This file contains debug information such as symbol information and line number information. A SYM file can be generated by specifying the "-S" option when executing ASM45(The "-C" option outputs the source line information). The file attribute is ".SYM". Although a SYM file is not necessarily required for PD45 to operate, most of its functions are unavailable to use without a SYM file read in. Therefore, always be sure to read in a SYM file along with a HEX file.

#### 3.1.3 Data Files

A data file contains the information inherent in the target MCU. This data file is included with the product package. The file name is "M345xxxx.DAT" (M345xxxx denotes an MCU name).

The PC4504 emulator allows you to emulate all MCU types in the 4500 series by changing the MCU boards available for the PC4504. The data file is an information file necessary to accommodate differences in MCU specifications when MCU types are changed.

#### 3.1.4 Script File

The script file is for automatic execution of script commands. This file, which is read from the Script Window, takes the attribute ".scr".

#### 3.1.5 Help File

The help file contains help messages for PD45. This file, which is supplied with PD45, takes the attribute ".hlp".



### **3.1.6 Environmental Setup File**

The environmental setup file, which is automatically generated by **PD45** and cannot be directly created or edited by the user, stores information about the environmental setup of **PD45**. The filename is pd45.ini. The environmental setup file is saved to the Windows directory (the directory in which you have installed Windows).

## **3.2 Output Files**

**PD45** outputs the following files.

### **3.2.1 Intel HEX-format Files**

The Intel HEX-format files store machine language data. These files, which are saved using **PD45**'s upload function, take the attribute ".hex". Files saved in the Intel HEX format can be downloaded by **PD45**.

### **3.2.2 Disassemble Files**

Disassemble files store the results of disassembling program memory. These files, which are reference text files, take the attribute ".txt". Disassemble files cannot be reassembled or downloaded.

### **3.2.3 Log File**

The log file is a text file containing the results of executing the script commands. This file, which takes the attribute ".log", contains the results from the logon point to the logoff point.

### **3.2.4 View File**

The view file is a text file that contains the contents of the script window. In **PD45**, the last 1000 lines of the results of executing the script commands are stored in the view buffer. The view file, which takes the attribute ".vew", contains the contents of the view buffer.

[ MEMO ]

# Setup



# 1. Setup

## 1.1 Installation

See the Release Notes provided with the product for how to install **PD45**.

## 1.2 Starting PD45

Follow the instructions below to start **PD45**:

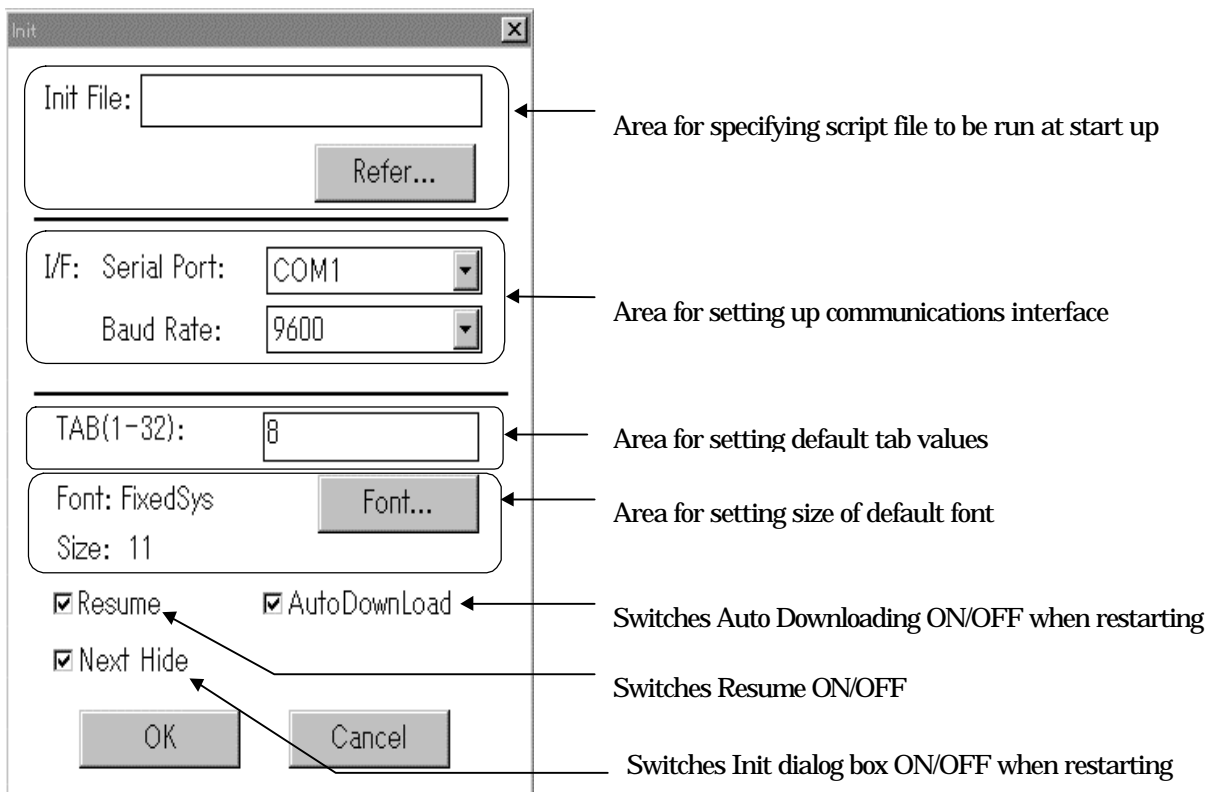
Click the Windows start button, then select  
program (P) -> [RENESAS-TOOLS] ->[**PD45** V.x.xx Release x] ->[**PD45**]

This operation starts **PD45** (the emulator debugger).

## 1.3 PD45 Setup

When you start **PD45**, the Init dialog box is displayed, allowing you to set up the operating environment.

### 1.3.1 Init Dialog Box Screen Configuration



### 1.3.2 Environment Setup

#### 1. Specifying the script file to be executed at startup

Specify a script file if you want to execute script commands at startup. These commands must have been saved as a script file. Click the "Refer" button to open the file selection dialog box, then use the mouse to select the script file to be executed at startup. The selected script file is displayed after InitFile: in the Init dialog box.

## 2. Setting of the communications interface

Specify the communications interface for communication with the emulator.



Specify the communications port in the Port field and the baud rate in the Baud Rate field.

## 3. Setting of default TAB values

Specify the default tab values for the Program Window, Source Window. You can specify TAB values between 1 and 32.

### Note:

You can set the tabs independently in each window. With the target window active, select [Option] -> [TAB] from the menu in the **PD45** Window to open the TAB Setting dialog box. You can now set the tabs for the active window.

## 4. Setting of default font size

Specify the default font for the characters displayed by **PD45**. Click the "Font" button to open the Font dialog box, then specify the font and the font size.

### Note:

You can set the font size independently in each window. With the target window active, select [Option] -> [Font] from the menu in the **PD45** Window to open the Font Setting dialog box. You can now set the font size for the active window.

## 5. Switches of Init dialog box ON/OFF of restarting

Specify whether the Init dialog box should be opened when restarting **PD45**. Check "Next Hide" to stop the Init dialog box being opened when you next start **PD45**. If you want to display the Init dialog box when you next start **PD45**, select [Init] -> [Environ] from the **PD45** Window menu, then uncheck "Next Hide" in the Init dialog box. You can also force the Init dialog box to be displayed when you start up by pressing and holding the Ctrl key when starting **PD45**.

## 6. Switches of Resume ON/OFF

You can specify whether, when you restart **PD45**, the windows are restored to their status when you last quit the program. Check the Resume box to do so.

## 7. Selection of Auto downloading at startup

You can specify whether, when you restart **PD45**, the program automatically fetches the last-read target program. Check the Auto Download box to do so.

[ MEMO ]



# Window function

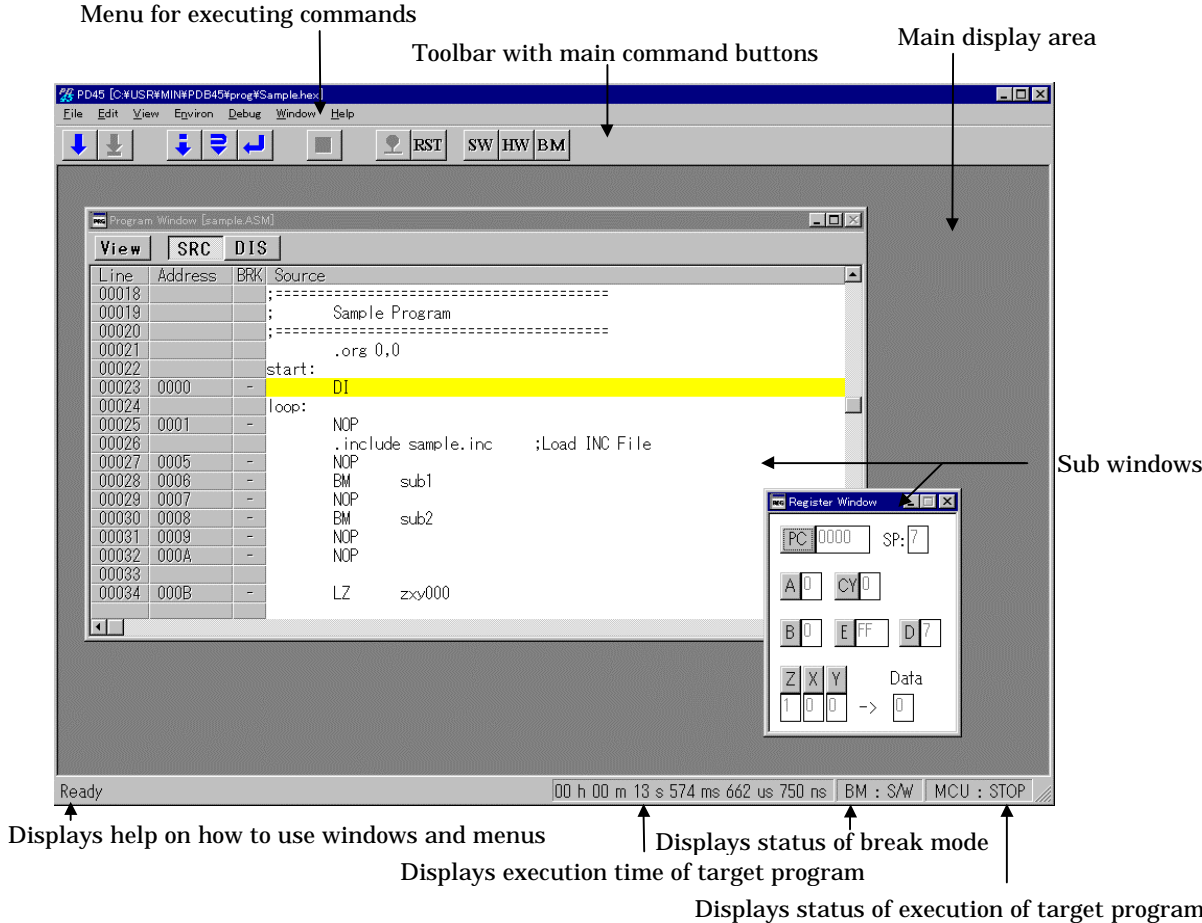


# 1. Window function of PD45

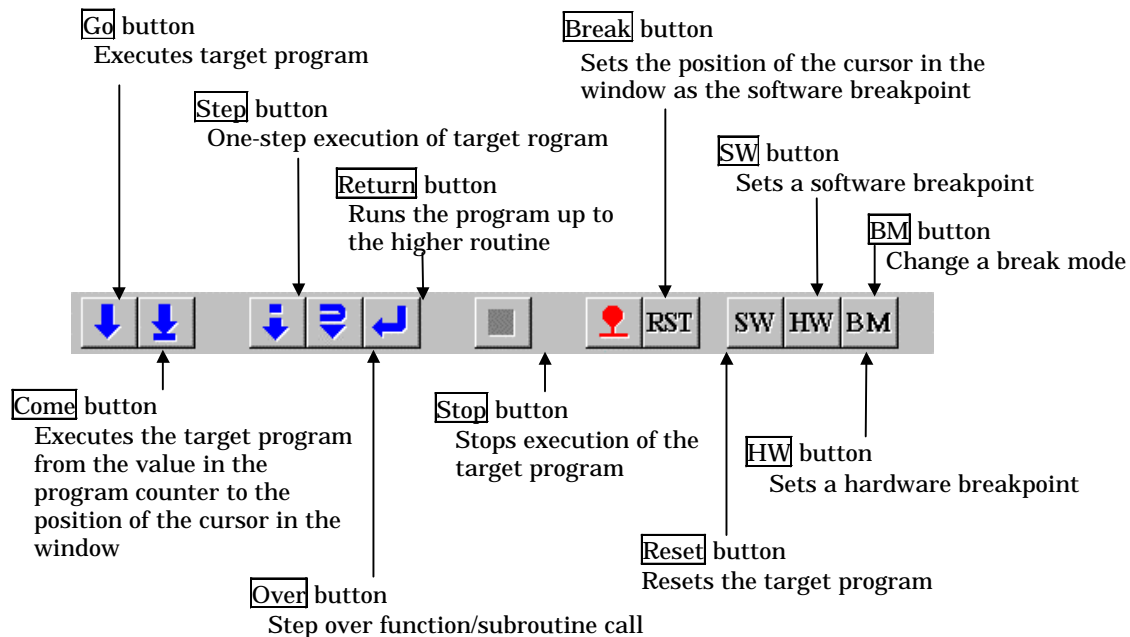
## 1.1 The PD45 Window

The PD45 Window is the main window for PD45. This window displays the main commands on a toolbar. You can click on the buttons on this toolbar to run the target program in normal or one-step mode. The main display area accommodates windows such as the Target Program Window.

### 1.1.1 PD45 Window Screen Configuration

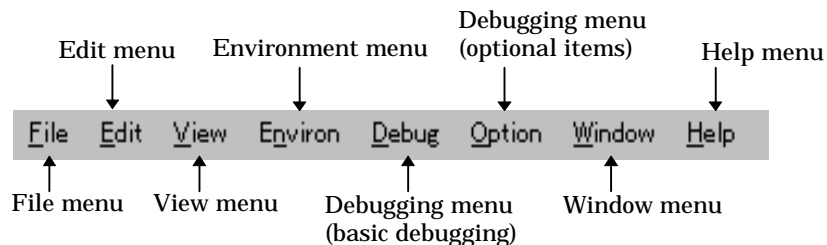


### 1.1.2 PD45 Window Toolbar



### 1.1.3 PD45 Window Menu

The menus in the PD45 Window can be classified as basic menus and extended menus.



#### **Basic Menus and Extended Menus**

The sub-menu items of the optional menu in PD45 automatically change according to which window is active in the main display area of the PD45 window. The optional menus are called extension menus.

In contrast, the items on all menus other than the optional menu remain the same no matter which window is active. These are called basic menus, and provide the items for the basic operation of PD45 and debugging.

Below, we look at the functions of each item on the basic menus. The functions of the items on the extended menus are described under the respective windows.

### 1.1.3.1 File operation

The [File] menu in PD45 contains the items required for file operation such as including files, saving files, and terminating PD45.

| Menu | Menu options    | Function  | Shortcut keys |
|------|-----------------|---|---------------|
| File | Download        | Download target program.                                  | -             |
|      | Load Module...  | Download machine language data and debugging information. | Shift + F.1   |
|      | Memory Image... | Download only machine language data.                      | -             |
|      | Symbol...       | Download only debugging information.                      | -             |
|      | Upload...       | Upload target program.                                    | -             |
|      | Save Disasm...  | Save disassembly result.                                  | -             |
| Exit | Terminate PD45. | -   |               |

### 1.1.3.2 Editing

The [Edit] menu in PD45 contains the items required for editing operation such as character strings copy, paste, and search.

| Menu | Menu options | Function                                       | Shortcut keys |
|------|--------------|--|---------------|
| Edit | Copy         | Copy character strings specified to clipboard. | Ctrl + C      |
|      | Paste        | Paste character strings of clipboard.          | Ctrl + V      |
|      | Find...      | Find character strings.                        | -             |

### 1.1.3.3 Display

The [View] menu in PD45 contains the items required for switching display of the tool bar and status bar.

| Menu | Menu options | Function                                     | Shortcut keys |
|------|--------------|--|---------------|
| View | Tool Bar     | Switch display or non-display of tool bar.   | -             |
|      | Status Bar   | Switch display or non-display of status bar. | -             |

### 1.1.3.4 Setup

The [Environ] menu in PD45 contains the items required for setting up the operating environment.

| Menu    | Menu options | Function                | Shortcut keys |
|---------|--------------|-------------------------|---------------|
| Environ | Init...      | Environment setup       | -             |
|         | Path...      | Source file search path | -             |

### 1.1.3.5 Debugging (Basic)

The [Debug] menu in PD45 contains the items for basic debugging such as starting and stopping and one-step execution of the target program.

| Menu                         | Menu options                             | Function   | Shortcut keys |
|------------------------------|--|--|---------------|
| Debug                        | <u>G</u> o                               | Start target program.                              | -             |
|                              | <u>G</u> o                               | Run from current program counter.                  | F.1           |
|                              | Go <u>O</u> ption...                     | Run from specified address.                        | -             |
|                              | Go <u>M</u> ode                          | Switch over control mode when program has stopped. | -             |
|                              | <u>D</u> i                               | Changed to DI instruction insert mode.             | -             |
|                              | <u>S</u> top                             | Changed to clock stop mode.                        | -             |
|                              | <u>C</u> ome                             | Run to cursor position.                            | F.2           |
|                              | <u>S</u> tep                             | Step execution.                                    | -             |
|                              | <u>S</u> tep                             | Execute one step.                                  | F.3           |
|                              | Step <u>O</u> ption...                   | Execute specified No. of steps.                    | -             |
|                              | <u>O</u> ver                             | Over-step execution.                               | -             |
|                              | <u>O</u> ver                             | Execute one over-step.                             | F.4           |
|                              | Over <u>O</u> ption...                   | Execute specified No. of over-steps.               | -             |
|                              | <u>R</u> eturn                           | Execute until return from current subroutine.      | F.5           |
|                              | <u>B</u> reak Point                      | Set break point.                                   | -             |
| <u>S</u> /W Break Point...   | Open S/W Break Point Setting dialog box. | F.7  |               |
| <u>H</u> /W Break Point...   | Open H/W Break Point Setting dialog box. | Shift + F.7  |               |
| <u>B</u> reak                | Set/cancel software break at cursor.     | -  |               |
| <u>R</u> eset                | Reset target program.                    | F.8  |               |
| <u>S</u> top                 | Stop target program.                     | -  |               |
| <u>B</u> reak <u>M</u> ode   | Change break mode.                       | -  |               |
| <u>T</u> race Point...       | Set trace Point.                         | -  |               |
| <u>M</u> easurement Point... | Set measurement Point.                   | -  |               |

### 1.1.3.6 Debugging (Option)

The extended menus in PD45 contains the items for operating the various PD45 windows. The items on the extended menus differ according to which window is active. The functions of the items on the extended menus are described under the respective windows.

| Menu           | Menu options | Function  | Shortcut keys |
|----------------|--------------|---|---------------|
| <u>O</u> ption |              | (This menu contains the extended menus for the various PD45 windows.) | -             |

### 1.1.3.7 Window Operations

The [Window] menu contains the items for manipulating the display functions of the windows displayed by PD45.

| Menu   | Menu options     | Function                    | Shortcut keys |
|--------|------------------|-----------------------------|---------------|
| Window | Cascade          | Cascade windows.            | -             |
|        | Tile             | Tile windows.               | -             |
|        | Arrange Icon     | Arrange icons.              | -             |
|        | Program Window   | Make Program Window active. | -             |
|        | Source Window    | Open Source Window.         | -             |
|        | Register Window  | Open Register Window.       | -             |
|        | Memory Window    | Open Memory Window.         | -             |
|        | Dump Window      | Open Dump Window.           | -             |
|        | ASM Watch Window | Open ASM Watch Window.      | -             |
|        | Script Window    | Open Script Window.         | -             |
|        | Trace Window     | Open Trace Window.          | -             |
|        | Coverage Window  | Open Coverage Window.       | -             |

### 1.1.3.8 Help

The [Help] menu contains the items for displaying PD45 help messages and the PD45 version No.

| Menu | Menu options | Function                                | Shortcut keys |
|------|--------------|---|---------------|
| Help | Index        | Display help.                           | -             |
|      | About...     | Display version information about PD45. | -             |

## 1.2 Program Window

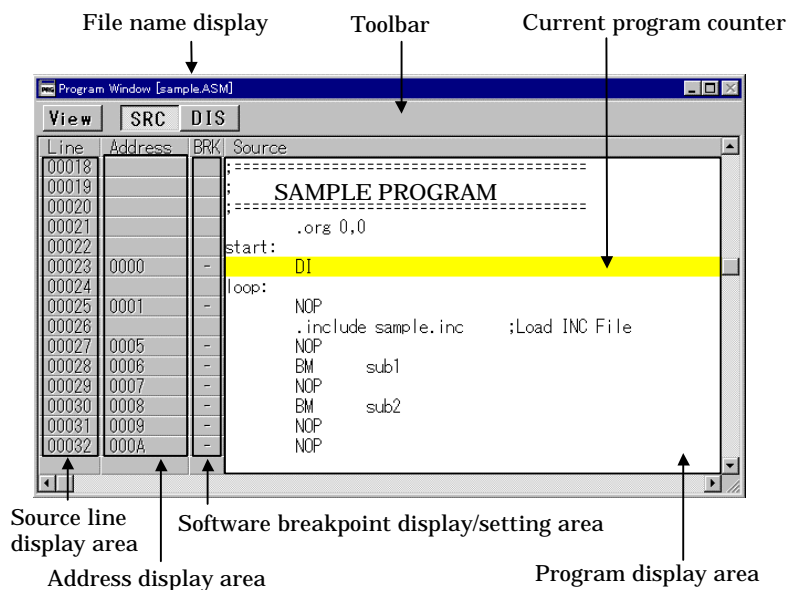
The Program Window displays the machine code at the current program counter. The line at the program counter is highlighted in yellow. The Program Window is automatically opened in the main display area of the PD45 Window when you start PD45. The Program Window can be used for executing the target program up to the cursor position, set or cancel software breakpoints using the mouse, and displaying reverse assembles of the target program, etc. Double-click the software breakpoint display/setting area to set or cancel software breakpoints.

### 1.2.1 Structure of Program Window

The Program Window has two display modes: source and disassemble. The structure of the Program Window is described for each display mode below.

#### 1.2.1.1 Structure of Program Window in Source Display Mode

The source display mode is provided for debugging at the source level. You can check the source files of the target program in source display mode.

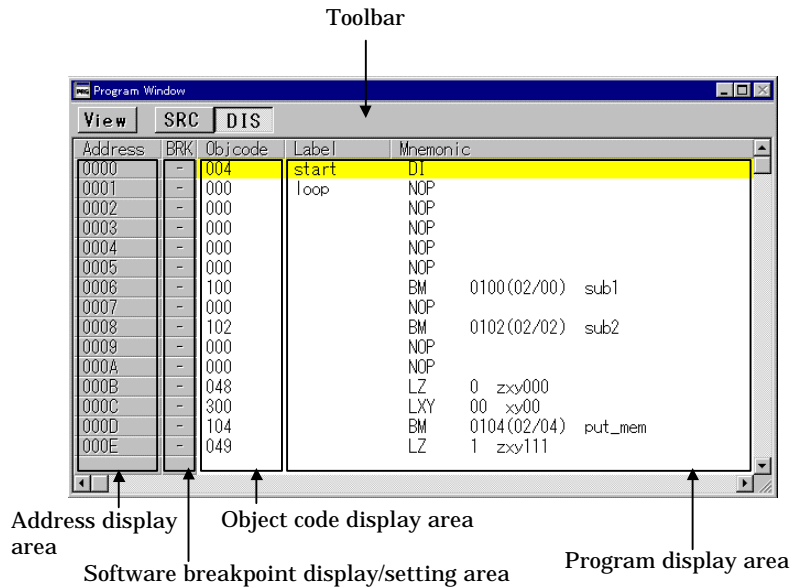


- The line No. display area and address display area can be displayed or hidden by selecting or cancelling [Option] -> [Layout] -> [Line Area] and [Option] -> [Layout] -> [Address Area]. Note that, by default, the address display area is hidden.
- Double-click the line No. display area to change the display source file.
- Double-click the address display area to change the display starting address or the display starting line.



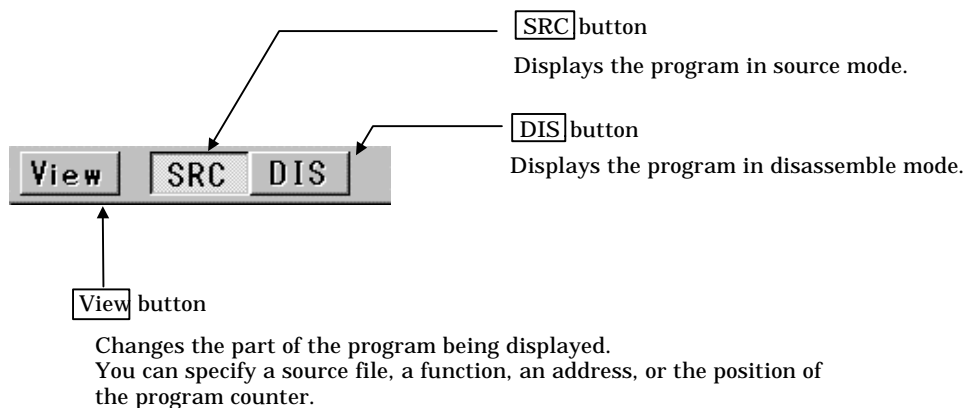
### 1.2.1.2 Structure of Program Window in Disassemble Display Mode

The disassemble display mode is provided for debugging at the instruction level. You can check the results of disassembling the target program in disassemble display mode.



- The address display area and object code display area can be displayed or hidden by selecting or cancelling [Option] -> [Layout] -> [Address Area] and [Option] -> [Layout] -> [Code Area].
- Note that you cannot scroll backwards vertically until you have scrolled forwards. When you scroll forwards, the previous display address is stored in the internal buffer. This address information is used when you scroll backwards. Note that the internal buffer is cleared if you execute a command that changes the first line address.
- If you disassemble an area (data or empty area, etc.) other than the program, the contents of memory are interpreted as instruction code and displayed in disassembled format. In this case, "???" is displayed for undefined instructions and operands.
- Double-click the address display area to change the display starting address.

### 1.2.2 Program Window Toolbar



### 1.2.3 Extended menu in the Program Window

When the Program Window is active in the PD45 main display area, the [Option] menu contains the following items:

| Menu         | Menu options                  | Function  | Shortcut keys |
|--------------|-------------------------------|---|---------------|
| Option       | Font...                       | Change font.                                    | -             |
|              | TAB...                        | Set source file display tabs.                   | -             |
|              | View                          | Change contents of display.                     | -             |
|              | Source...                     | Display from specified source file or function. | -             |
|              | Address...                    | Display from specified address or line No.      | -             |
|              | Program Counter               | Display from current program counter.           | -             |
|              | Mode                          | Switch display mode.                            | -             |
|              | Source mode                   | Switch to source display mode.                  | Ctrl + R      |
|              | Disasm mode                   | Switch to disassemble display mode.             | Ctrl + R      |
|              | Layout                        | Set layout.                                     | -             |
| Line Area    | Turn on/off line No. area.    | -   |               |
| Address Area | Turn on/off address area.     | -   |               |
| Code Area    | Turn on/off object code area. | -   |               |

### 1.3 Source Window

The Source Window is provided for dedicated display of the program being checked. The line at the program counter is highlighted in yellow. In contrast to the Program Window, which follows the program counter, the Source Window is not updated until you specify. Use the Source Window to check what is happening in specific subroutines. You can open up to 10 Source Windows. Other functions are the same as in the Program Window.

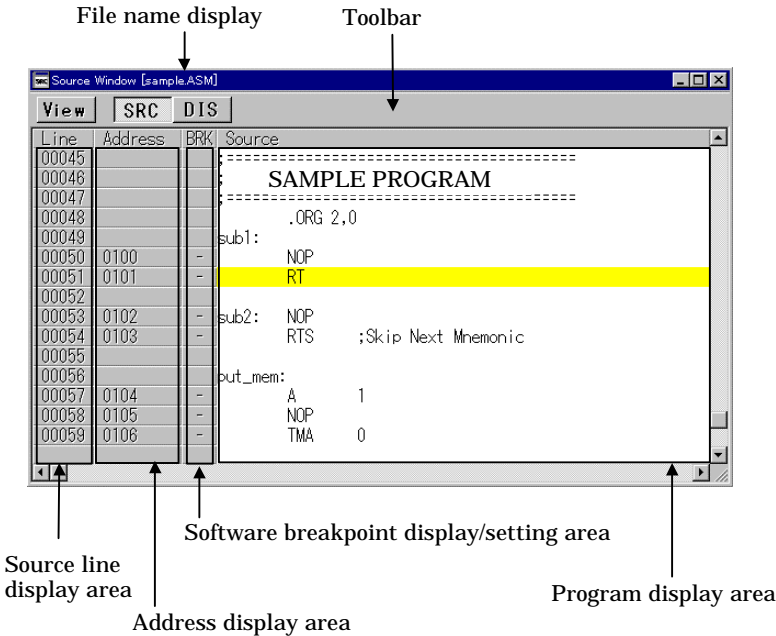
#### 1.3.1 Structure of Source Window

The Source Window has two display modes: source and disassemble. The structure of the Source Window is described for each display mode below.

The Source Window has the same structure as the Program Window. See Section 1.2.1, "Structure of Program Window" under Window Functions for details.

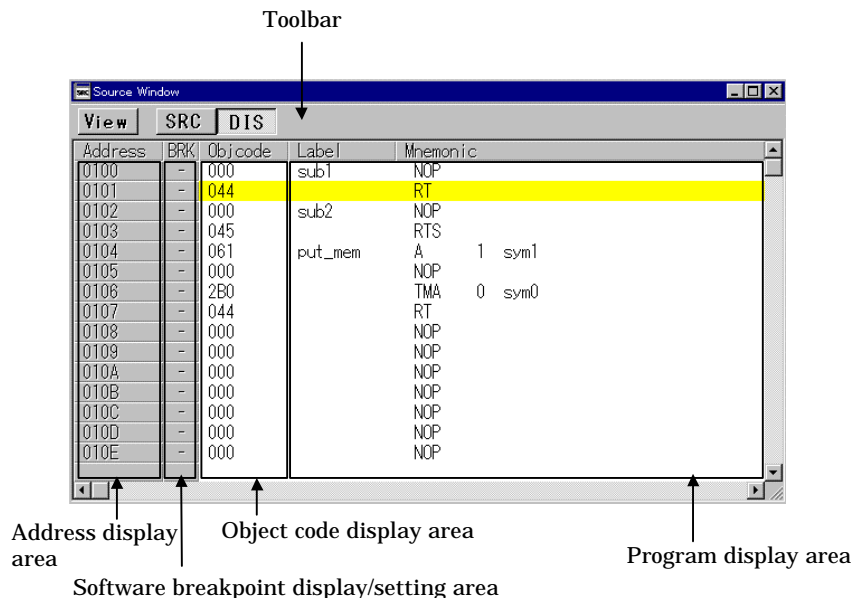
##### 1.3.1.1 Structure of Source Window in Source Display Mode

The source display mode is provided for debugging at the source level. You can check the source files of the target program in source display mode.



### 1.3.1.2 Structure of Source Window in Disassemble Display Mode

The disassemble display mode is provided for debugging at the instruction level. You can check the results of disassembling the target program in disassemble display mode.



### 1.3.2 Source Window Toolbar

The Source Window toolbar is the same as that in the Program Window. See Section 1.2.2, "Program Window Toolbar" under Window Functions for details.

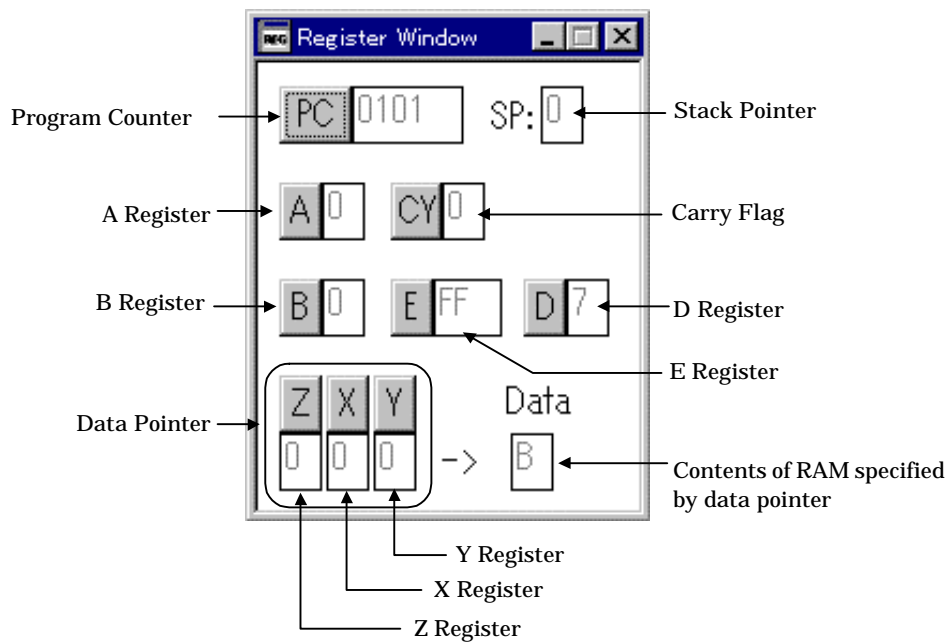
### 1.3.3 Extended Menus in the Source Window

When the Source Window is active in the PD45 main display area, extended menus for the Source Window are allocated to the [Option] menus. The extended menus of the Source Window are identical to those of the Program Window. See Section 1.2.3, "Extended menu in the Program Window" under Window Functions for details.

## 1.4 Register Window

The Register Window shows the contents of the registers. The display is updated after each command is executed. You can click the buttons corresponding to registers in the Register Window to quickly change the values of the registers.

### 1.4.1 Structure of Register Window



### 1.4.2 Extended Menus in the Register Window

The Register Window is not exist the [Option] menu.

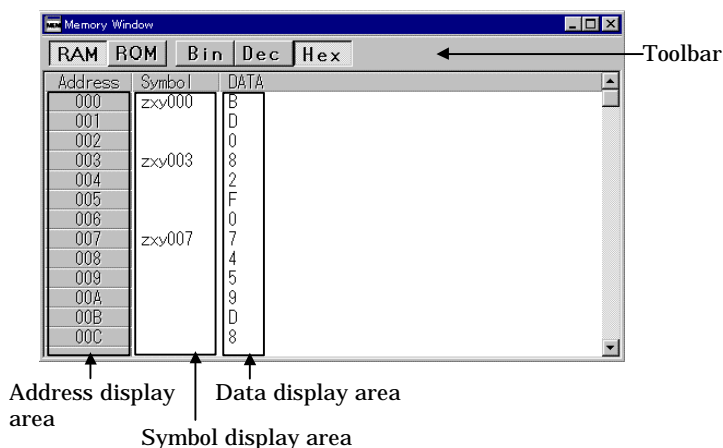
## 1.5 Memory Window

The Memory Window displays the contents of contiguous memory in "address", "symbol", and "data (contents of memory)" formats. The display is updated after each command is executed. Data can be displayed in binary, decimal, and hexadecimal. You can open up to 10 Memory Windows. You can use the Memory Windows to modify the contents of memory, and also to fill and move specified blocks of memory.

### 1.5.1 Structure of Memory Window

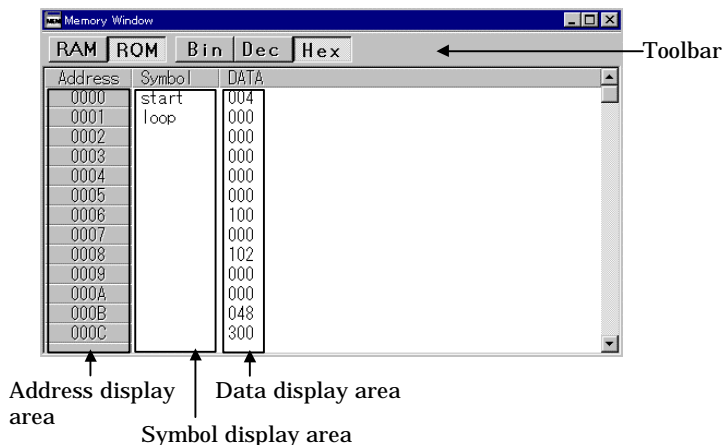
The Memory Window has two display modes: RAM and ROM. The diagrams below show the Memory Window's screen structures in the RAM display and the ROM display modes.

#### 1.5.1.1 Structure of Memory Window in RAM Display Mode



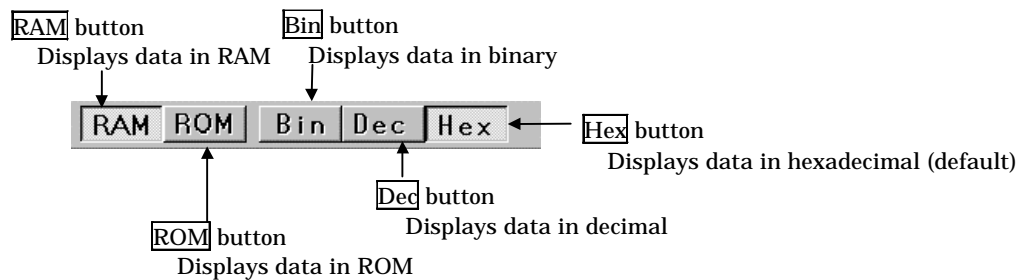
- Double-click the address display area to change the display RAM starting address.
- Double-click a symbol or the memory display area to change the contents of memory.

#### 1.5.1.2 Structure of Memory Window in ROM Display Mode



- Double-click the address display area to change the display ROM starting address.
- Double-click a symbol or the memory display area to change the contents of memory.

### 1.5.2 Memory Window Toolbar



### 1.5.3 Extended Menus in the Memory Window

When the Memory Window is active in the PD45 main display area, the [Option] menu contains the following items:

| Menu    | Menu options                                      | Function   | Shortcut keys |
|---------|---|--|---------------|
| Option  | Font...   | Change font.   | -             |
|         | View  | Change contents of display.                            | -             |
|         | Scroll Area...                                    | Specify scroll range.                                  | -             |
|         | Address...  | Specify display starting address.                      | -             |
|         | Data Pointer                                      | Change display start address to data pointer position. | -             |
|         | Memory Area                                       | Change display memory.                                 | -             |
|         | Ram   | Display in RAM   | -             |
|         | Rom   | Display in ROM   | -             |
|         | Radix   | Specify radix.   | -             |
|         | Bin   | Display in binary.                                     | -             |
|         | Dec   | Display in decimal.                                    | -             |
|         | Hex   | Display in hexadecimal.                                | -             |
|         | Refresh   | Refresh display.                                       | -             |
|         | Debug   | Set memory contents.                                   | -             |
|         | Set...  | Set data at specified address.                         | -             |
| Fill... | Fill specified memory block with data.            | -  |               |
| Move... | Move specified memory block to specified address. | -  |               |

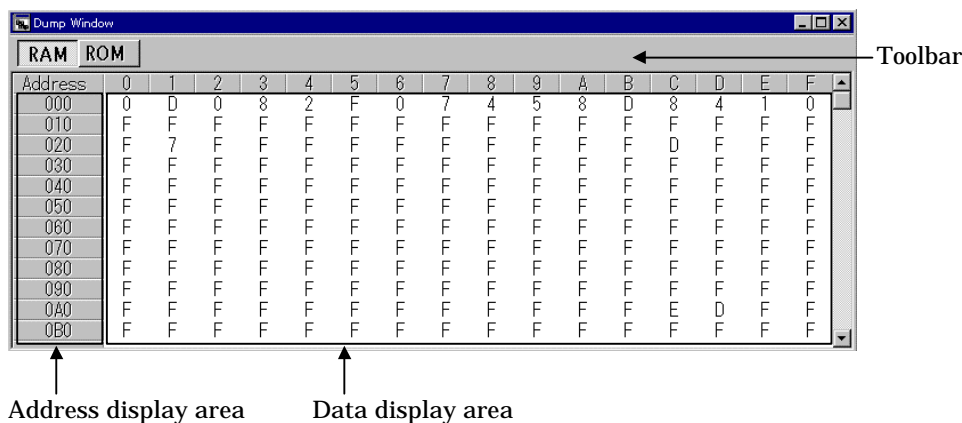
## 1.6 Dump Window

The Dump Window displays the contents of contiguous memory in dump format. The display is updated after each command is executed. You can open up to 10 Dump Windows. You can use the Dump Windows to modify the contents of memory, and also to fill and move specified blocks of memory.

### 1.6.1 Structure of Dump Window

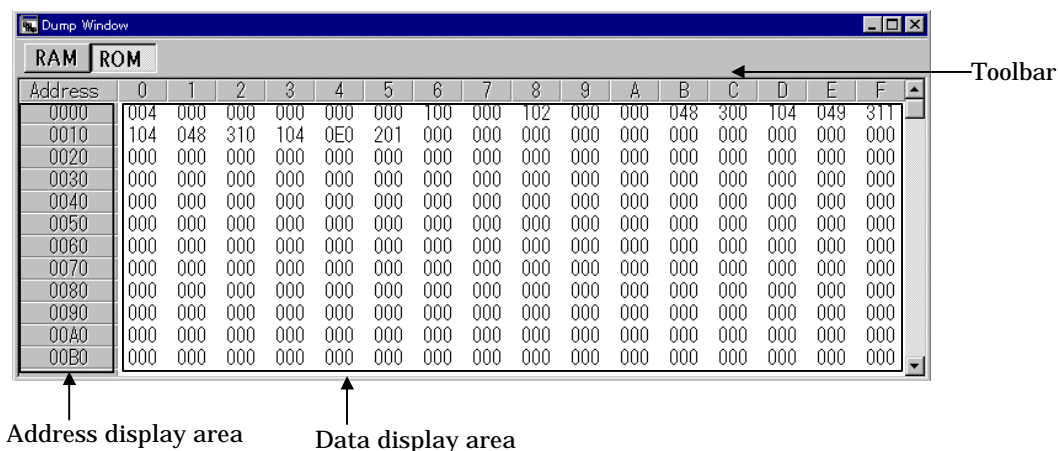
The Dump Window has two display modes: RAM and ROM. The diagrams below show the dump window's screen structures in the RAM display and the ROM display modes.

#### 1.6.1.1 Structure of Memory Window in RAM Display Mode



- Double-click the address display area to change the display RAM starting address.
- Double-click the memory display area to change the contents of memory.

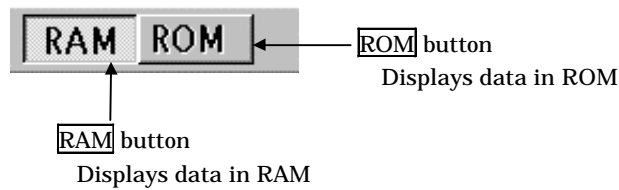
#### 1.6.1.2 Structure of Memory Window in ROM Display Mode



- Double-click the address display area to change the display RAM starting address.
- Double-click the memory display area to change the contents of memory.



## 1.6.2 Dump Window Toolbar



## 1.6.3 Extended Menus in the Dump Window

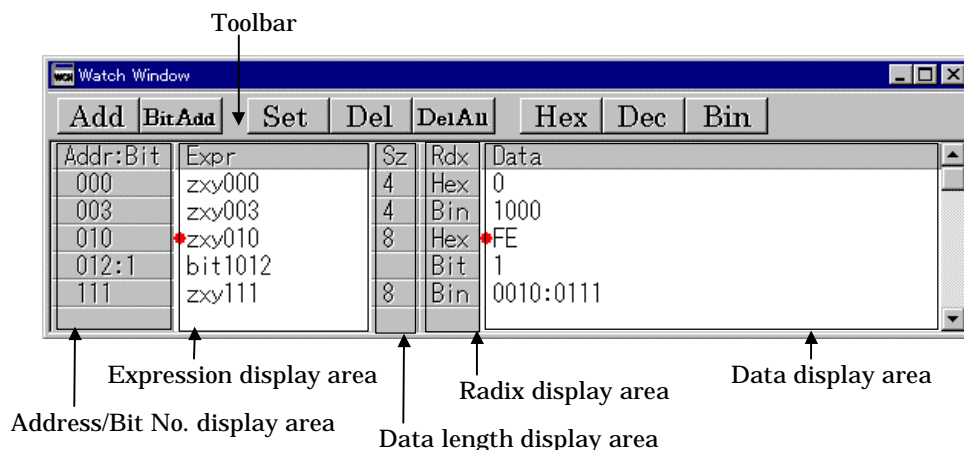
When the Dump Window is active in the PD45 main display area, the [Option] menu contains the following items:

| Menu            | Menu options                                      | Function                               | Shortcut keys |
|-----------------|---|--|---------------|
| Option          | <u>F</u> ont...                                   | Change font.                           | -             |
|                 | <u>V</u> iew                                      | Change contents of display.            | -             |
|                 | <u>S</u> croll Area...                            | Specify scroll range.                  | -             |
|                 | <u>A</u> ddress...                                | Specify display starting address.      | -             |
|                 | <u>M</u> emory Area                               | Change display memory.                 | -             |
|                 | <u>R</u> am                                       | Display in RAM                         | -             |
|                 | <u>R</u> om                                       | Display in ROM                         | -             |
|                 | <u>R</u> efresh                                   | Refresh display.                       | -             |
|                 | <u>D</u> ebug                                     | Set memory contents.                   | -             |
|                 | <u>S</u> et...                                    | Set data at specified address.         | -             |
|                 | <u>F</u> ill...                                   | Fill specified memory block with data. | -             |
| <u>M</u> ove... | Move specified memory block to specified address. | -                                      |               |

## 1.7 ASM Watch Window

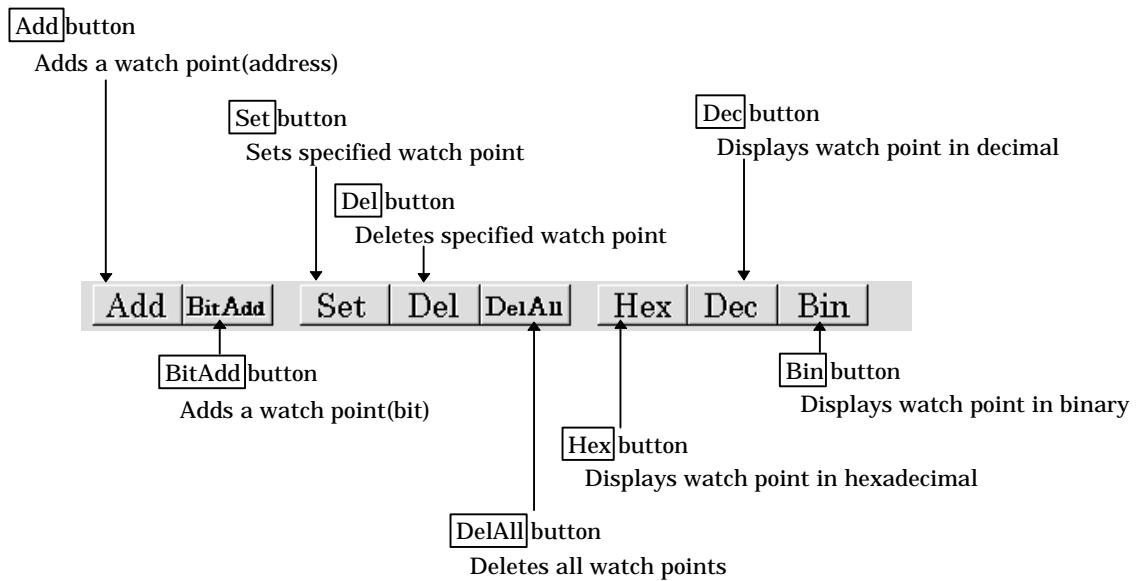
The ASM Watch Window allows you to check the values at any specified RAM address. You can specify the point to watch as an address (symbol or global variable), as an address and bit No., or as a bit symbol. The display is updated after each command is executed.

### 1.7.1 Structure of ASM Watch Window



- The cursor position is indicated by a red mark in the address display area and data display area. Click either area or use the up cursor and down cursor keys to move the cursor.
- Double-click the radix (Rdx) display area to switch the current radix for the data displayed in that area as follows:  
... -> hexadecimal -> decimal -> binary -> hexadecimal -> ...
- Information about set watchpoints is stored in the Init file when you close the ASM Watch Window or exit PD45. When you re-open the ASM Watch Window, the previously set watchpoints are automatically restored.
- In the case of the ASM Watch Window, the addresses of any previously set watchpoints are recalculated when you download a target program and the memory referenced using the new addresses. Thus there is no need to respecify the watchpoint addresses even when they change as a result of changes in the program.
  - When the addresses of inactive watchpoints (indicated by "--<not active>--") are recalculated and the result is a valid address, the watchpoints automatically become active again.

### 1.7.2 ASM Watch Window Toolbar



### 1.7.3 Extended Menus in the ASM Watch Window

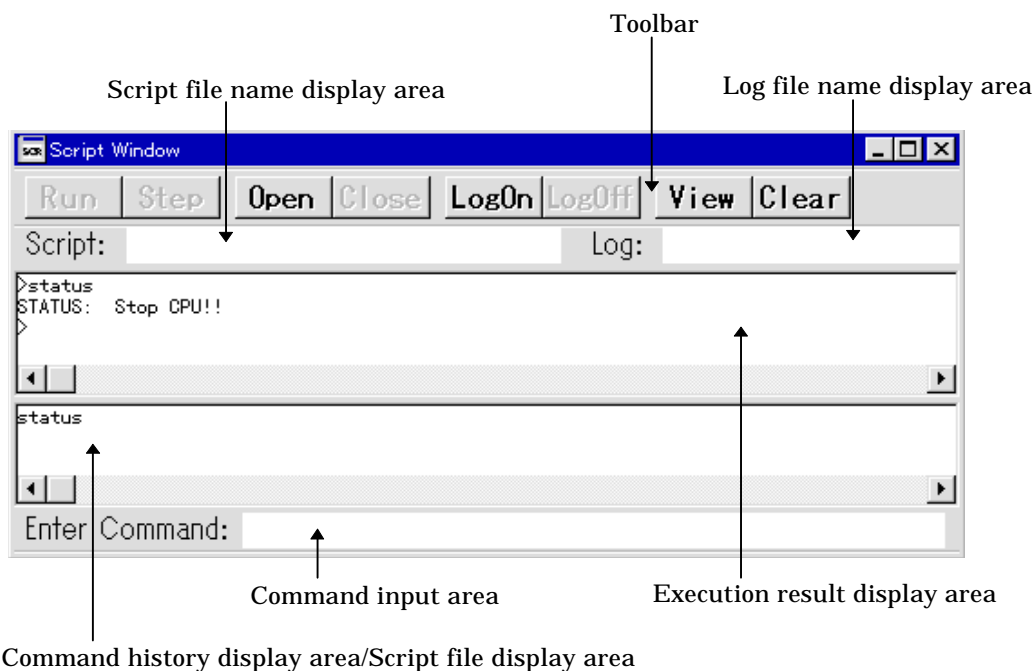
When the ASM Watch Window is active in the PD45 main display area, the [Option] menu contains the following items:

| Menu   | Menu options            | Function  | Shortcut keys |
|--------|-------------------------|---|---------------|
| Option | Font...                 | Change font.  | -             |
|        | Watch                   | Register / delete watch point.                        | -             |
|        | <u>A</u> dd...          | Register watch point.                                 | Ctrl + A      |
|        | <u>B</u> itadd...       | Register bit-level watch point.                       | Ctrl + B      |
|        | <u>S</u> et...          | Set new data to be written to selected watch point.   | Ctrl + S      |
|        | <u>D</u> el             | Delete selected watch point.                          | Ctrl + D      |
|        | <u>D</u> el <u>A</u> ll | Delete all watch points.                              | -             |
|        | Radix                   | Change display radix.                                 | -             |
|        | <u>B</u> in             | Display value at selected watch point in binary.      | Alt + B       |
|        | <u>D</u> ec             | Display value at selected watch point in decimal.     | Alt + D       |
|        | <u>H</u> ex             | Display value at selected watch point in hexadecimal. | Alt + H       |
|        | Layout                  | Set layout.   | -             |
|        | <u>A</u> ddress Area    | Turn on/off address/bit area.                         | -             |
|        | <u>S</u> ize Area       | Turn on/off data size area.                           | -             |

## 1.8 Script Window

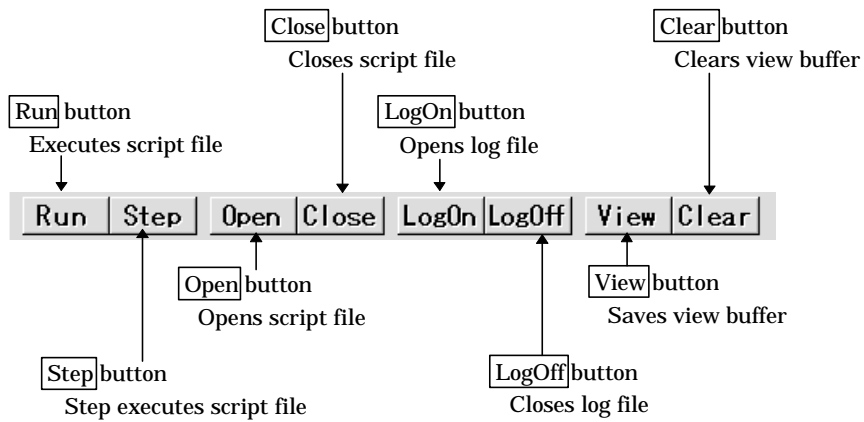
The Script Window displays the execution of text-format script commands and the results of that execution. Script commands can be executed using a script file or interactively. You can also write script commands in the script file so that they are automatically executed. The results of script command execution can also be stored in a previously specified log file.

### 1.8.1 Structure of Script Window



- The Script Window has a view buffer that stores the results of executing the last 1000 lines. The results of execution can therefore be stored in a file (view file) without specifying a log file.
- When a script file is opened, the command history area changes to become the script file display area and displays the contents of the script file. When script files are nested, the contents of the last opened script file are displayed. The script file display area shows the line currently being executed in inverse vide.
- When a script file is open, you can invoke script commands from the command input area provided the script file is not being executed.

## 1.8.2 Script Window Toolbar



## 1.8.3 Extended Menus in the Script Window

When the Script Window is active in the PD45 main display area, the [Option] menu contains the following items:

| Menu   | Menu options | Function                              | Shortcut keys |
|--------|--------------|---------------------------------------|---------------|
| Option | Font...      | Change font.                          | -             |
|        | Script       | Script file operations.               | -             |
|        | Open...      | Open script file.                     | -             |
|        | Run          | Run script file.                      | -             |
|        | Stop         | Stop execution of script file.        | -             |
|        | Step         | One-step execution of script file.    | -             |
|        | Close        | Close script file.                    | -             |
|        | View         | View buffer operations.               | -             |
|        | Save...      | Save view buffer file.                | -             |
|        | Clear        | Clear view buffer.                    | -             |
|        | Log          | Log file operations.                  | -             |
|        | On...        | Open log file (start output to file). | -             |
|        | Off          | Close log file (stop output to file). | -             |

## 1.9 Trace Window

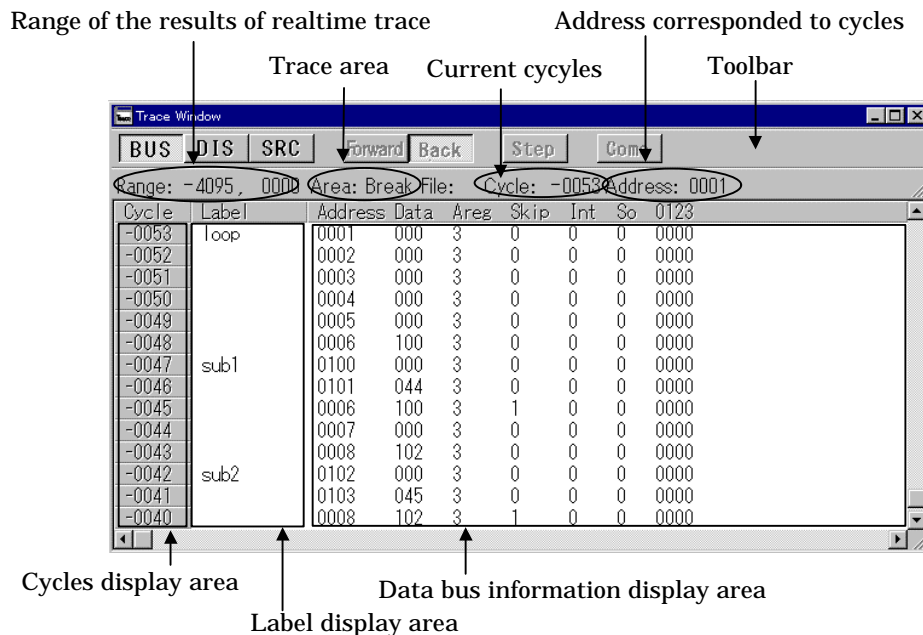
The Trace Window displays the results of the realtime trace function (execution history) of the PC4504 emulator. The Trace Window supports the following three display modes, which are selected using the buttons on the Trace Window toolbar.

- **Bus mode**  
Allows you to check bus data for each cycle. In bus mode, the trace contents are displayed following the execution path. The trace results can therefore be checked by scrolling using the vertical scrollbar.
- **Disassemble mode**  
The executed instructions are displayed in disassembled format. In disassemble mode, the trace contents are displayed following the execution path. The trace results can therefore be checked by scrolling using the vertical scrollbar.
- **Source mode**  
The execution paths can be searched for at the source line level. In source mode, the contents of the executed source file are displayed as-is (in contrast to being displayed following the execution path). The trace results can therefore be checked using the buttons on the toolbar.

The Trace Window displays the results of the trace immediately the realtime trace is completed. Because the realtime trace is incomplete immediately after starting PD45 or immediately after running the target program, the Trace Window remains blank until the trace is completed.

You can use trace points to specify when the trace starts and ends. Trace points are set via the Trace Point setting dialog box or using the Trace Point script command. If you do not specify any trace points (this is the default status), measurements end when execution of the target program breaks. Trace information is then displayed for the 4095 cycles prior to the break.

### 1.9.1 Screen Configuration in Bus Mode



#### Toolbar

In bus mode, only the display mode buttons (<BUS>, <DIS>, and <SRC>) are operative. However, if no debugging information has been downloaded, the <SRC> button is also inoperative.

#### Cycles display area

Displays the number of cycles per line.

- The cycles are displayed in +/- decimals either side of the trace point (the cycle at which the realtime trace starts/ends: cycle 0).
- You can double-click the cycles display area to change the No. of the cycle to be displayed.

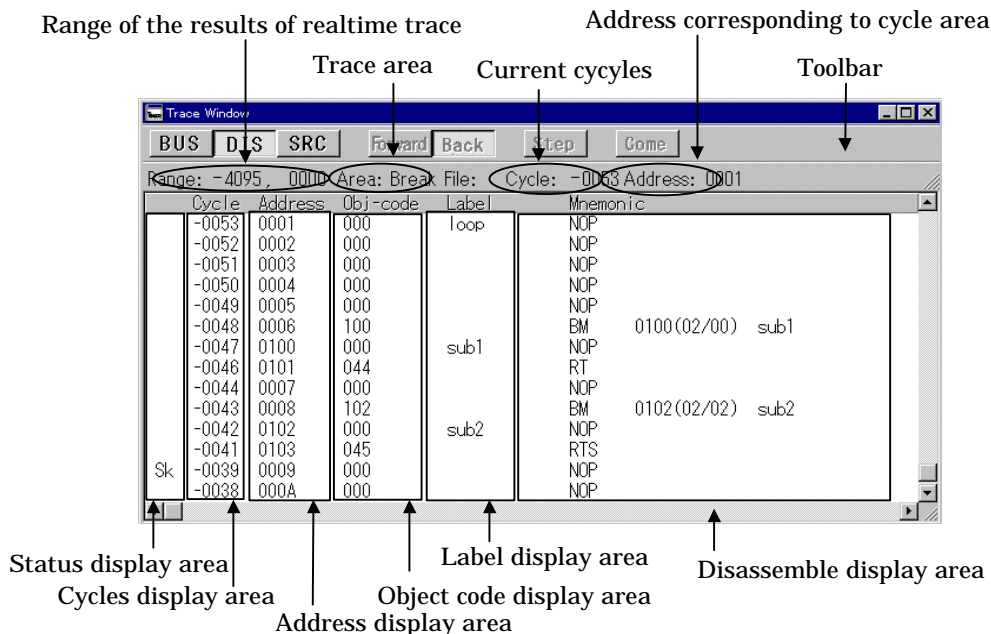
#### Data bus information display area

This area shows the bus information for each cycle. From left to right, the contents are as follows:

- Address  
Show the status of the address bus
- Data  
Show the status of the data bus
- Areg  
Show the content of the Accumulator.
- Skip  
Show the status of the instruction skip. (skipped instruction is indicated by '1')
- Int  
Show the status of the interrupt. (interrupt is indicated by '1')
- So  
Show the status of the stack overflow/underflow. (stack flow is indicated by '1')

- 0123  
Shows the status of the 4-bit external signal (High level = "1", Low level = "0")

## 1.9.2 Screen Configuration in Disassemble Mode



### Toolbar

In disassemble mode, only the display mode buttons (<BUS>, <DIS>, and <SRC>) are operative. However, if no debugging information has been downloaded, the <SRC> button is also inoperative.

### Status display area

Displays the number of status per cycle.

- If a skip occurs, "Sk" is displayed to the left of Cycles display area.
- if a stack overflow or underflow occurs, "So" is displayed to the left of Cycles display area.
- IF an interrupt occurs "++++INTERRUPT" is displayed.

### Cycles display area

Displays the number of cycles per line.

- The cycles are displayed in +/- decimals either side of the trace point (the cycle at which the realtime trace starts/ends: cycle 0).
- You can double-click the cycles display area to change the No. of the cycle to be displayed.

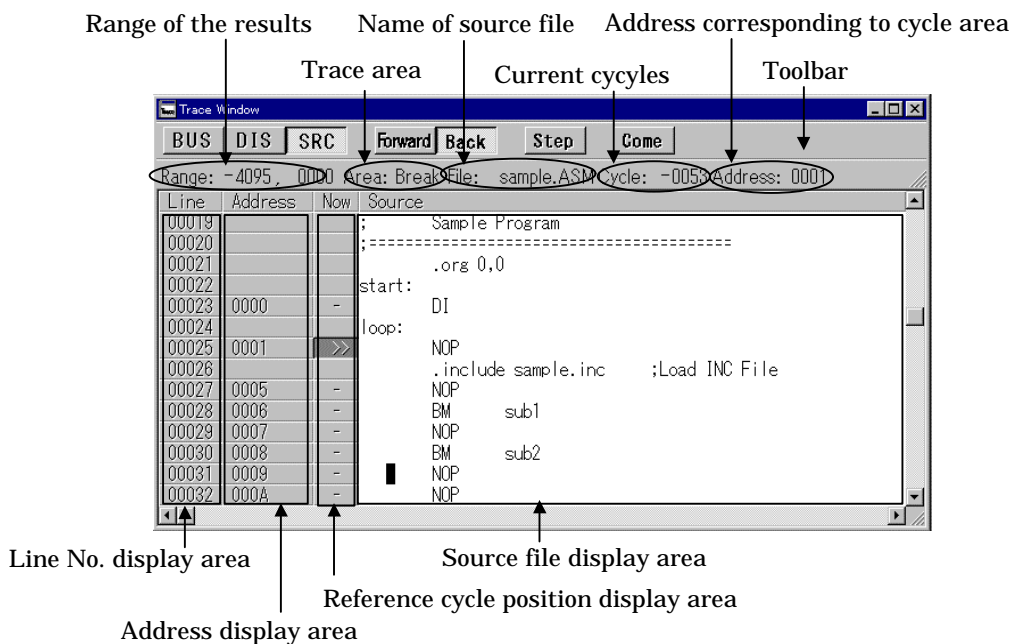
### Address display area

Displays the Address per instruction in hexadecimal format.

- You can search the address being displayed by double-clicking on the address display area.



### 1.9.3 Screen Configuration in Source Mode



#### Toolbar

In source mode, all buttons on the toolbar except the [Come] button are available.

The [Come] button is available when you click on a line in which a Come search can be performed.

#### Line No. display area

Displays the line No. in the source file.

- The line No. display area can be displayed or hidden by selecting or cancelling [Option] -> [Layout] -> [Line Area].
- You can change the source file being displayed by double-clicking on the line area.

#### Address display area

Displays the Address per line in hexadecimal format.

- The address display area can be displayed or hidden by selecting or cancelling [Option] -> [Layout] -> [Address Area]. Note that, by default, the address display area is hidden.
- You can search the address being displayed by double-clicking on the address display area.

#### Reference cycle position display area

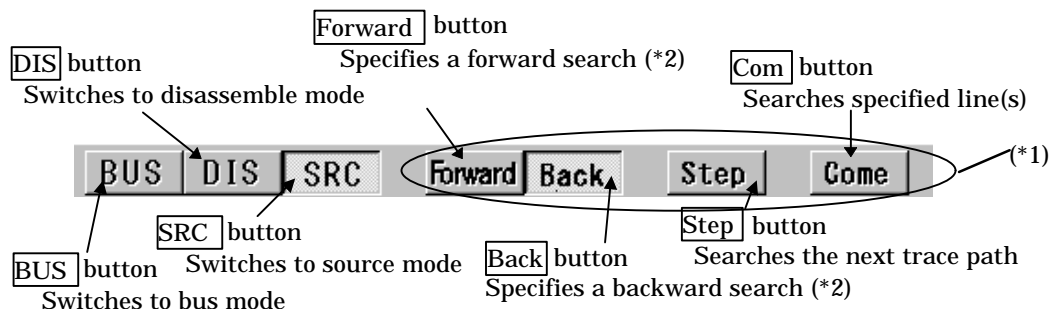
Displays the current position when searching for an address from the trace data.

- Displayed as ">>". This symbol moves when the search is continued. The hyphen (-) indicates a line with address information (Come searches can be performed on such lines).

#### **Note:**

Although the instructions that would otherwise be skipped are traced when searching for execution paths, they are skipped correctly in actual operation.

### 1.9.4 Trace Window Toolbar



\*1 The Forward, Back, Step, and Come buttons are only effective in source display mode.

\*2 Forward searches trace the path in the order in which instructions are executed.

Backward searches trace the path in the reverse order in which instructions were executed.

### 1.9.5 Extended Menus in the Trace Window

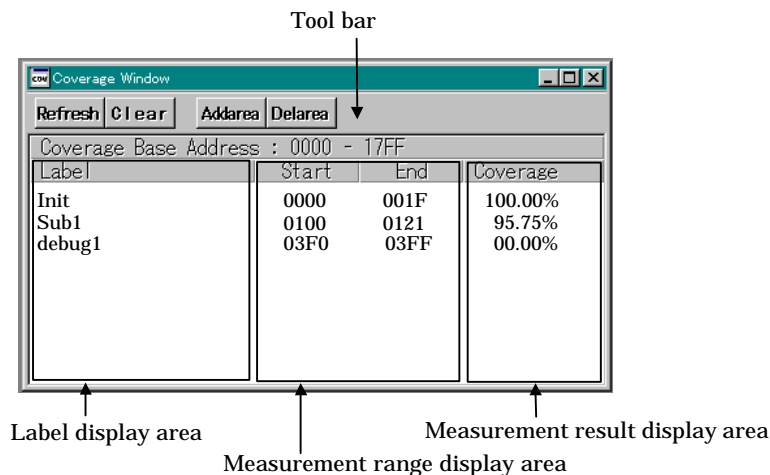
When the Trace Window is active in the PD45 main display area, the [Option] menu contains the following items:

| Menu   | Menu options                         | Function   | Shortcut keys |
|--------|--------------------------------------|--|---------------|
| Option | Font...                              | Change font.   | -             |
|        | TAB...                               | Set tabs for source file display.                    | -             |
|        | View                                 | Change contents of display.                          | -             |
|        | Cycle...                             | Specify cycle.                                       | -             |
|        | Address Search...                    | Search cycle by specifying address.                  | -             |
|        | Source...                            | Change by specifying source file.                    | -             |
|        | Mode                                 | Change display mode.                                 | -             |
|        | Bus                                  | Select bus mode.                                     | -             |
|        | Disasm                               | Select disassemble mode.                             | -             |
|        | Source                               | Select source mode.                                  | -             |
|        | Layout                               | Set layout.  | -             |
|        | Line Area                            | Turn on/off line No. area.                           | -             |
|        | Address Area                         | Turn on/off address area.                            | -             |
|        | Trace                                | Search trace results.                                | -             |
|        | Forward                              | Search forward (in direction of execution).          | -             |
|        | Backward                             | Search backward (in reverse direction of execution). | -             |
| Step   | Search one step (Step search).       | -  |               |
| Come   | Search specified line (Come search). | -  |               |

## 1.10 Coverage Window

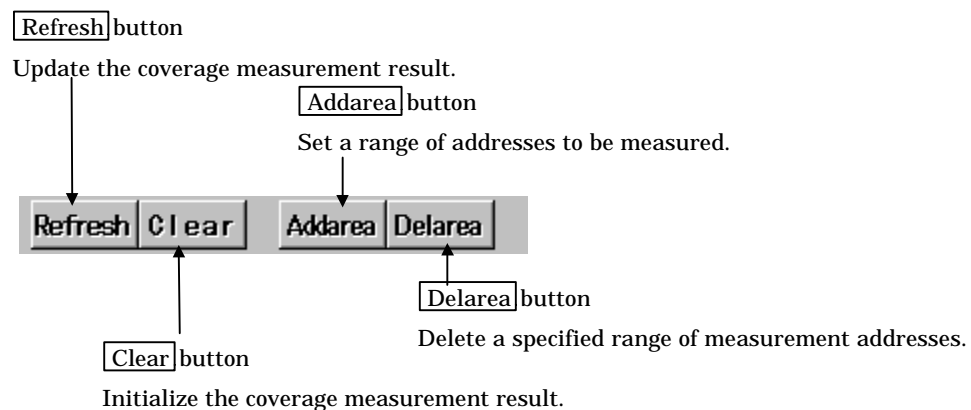
The Coverage Window is used to display the results of measurements performed by the coverage measurement function. This window shows the coverage measurement result of a given address range in percentage.

### 1.10.1 Structure of Script Window



- The coverage measurement result of a given address range is displayed in percentage.
- When you double-click on any desired line, a Coverage Source Window opens that allows you to check each source line to see if they have been executed or not.
- The measurement result display area is changed to '--%' during target program execution. If you want to update the display, press the **Refresh** button on the toolbar.

### 1.10.2 Coverage Window Toolbar



### 1.10.3 Extended Menus in the Coverage Window

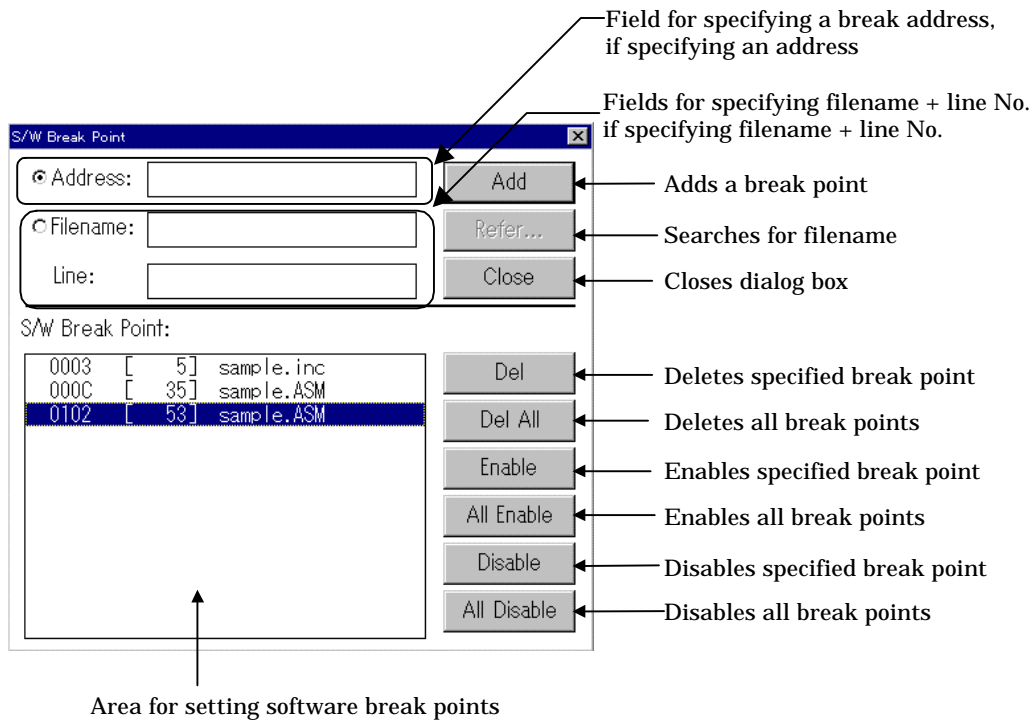
When the Coverage Window is active in the PD45 main display area, the [Option] menu contains the following items:

| Menu   | Menu options | Function   | Shortcut keys |
|--------|--------------|--|---------------|
| Option | Font...      | Change font.                                       | -             |
|        | Refresh      | Update the coverage measurement result.            | -             |
|        | Clear        | Initialize the coverage measurement result.        | -             |
|        | Add...       | Set a range of addresses to be measured.           | -             |
|        | Del          | Delete a specified range of measurement addresses. | -             |
|        | Layout       | Set layout.  | -             |
|        | Address Area | Turn on/off line Address area.                     | -             |

## 1.11 S/W Break Point Setting Dialog Box

The S/W Break Point Setting dialog box allows you to set software break points. Software breaks stop the execution of instructions immediately after the specified break point. You can also enable and disable each of those break points.

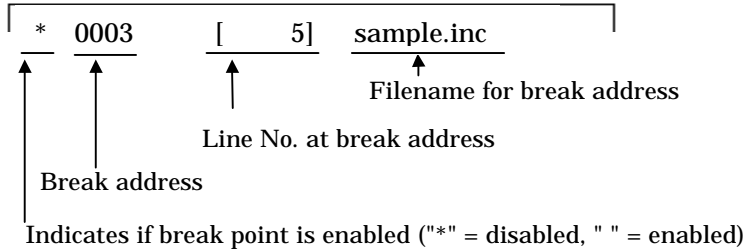
### 1.11.1 Structure of S/W Break Point Setting Dialog Box



- You can set up to 8 software break points in PD45.
- If you have set multiple software breakpoints, program execution stops when any one software break address is encountered (OR conditions).
- You can continue to set software breakpoints until you click the <Close> button to close the S/W BreakPoint setting dialog box.
- You can clear, enable or disable software breakpoints selected by clicking in the software breakpoint display area. You can also enable and disable software breakpoints by double-clicking on them.

### 1.11.2 Entries in List of Software Break Points

S/W Break Point:



- If there is no source line, etc., for the address, only the address is displayed.

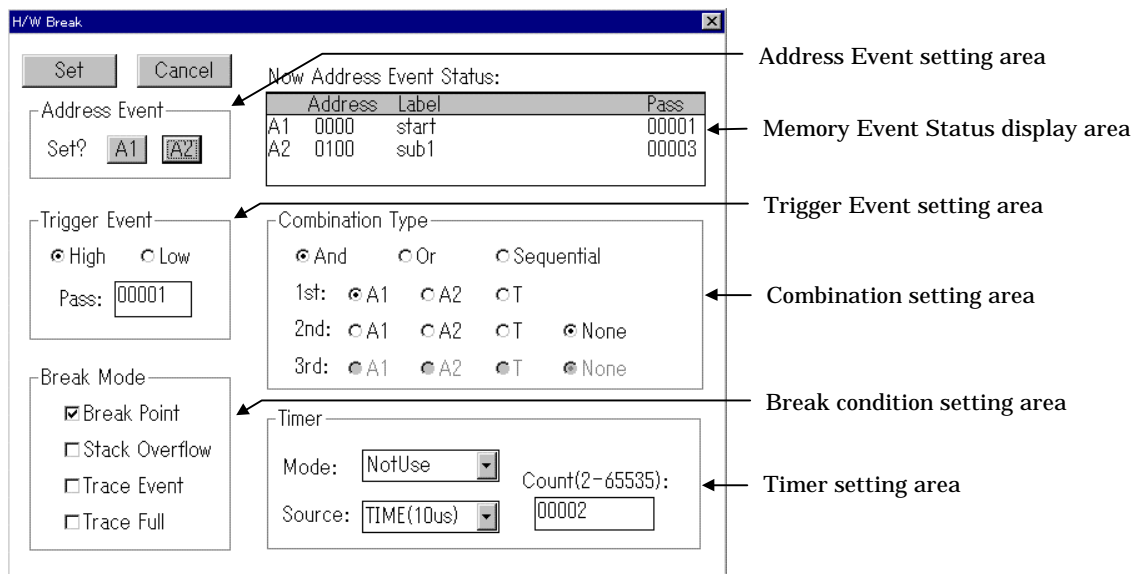
## 1.12 H/W Break Point Setting Dialog Box

The H/W Break Point Setting dialog box allows you to set hardware break points.

### 1.12.1 Structure of H/W Break Point Setting Dialog Box

In H/W Break Point Setting Dialog Box, you can set two address breakpoints with pass counts and one trigger break point. In addition to the above breakpoints, you can specify a break for failure such as a stack overflow or a break based on a timer count. You can also specify combinations of these three break points.

The following shows the screen configuration for the H/W Break Point Setting Dialog Box .



- For the address breakpoint setting area, use one of the three methods available for specifying breakpoint addresses for address break: one address (Point), within an address range (Range In), or outside an address range (Range Out).
- You can specify rising edge(High) or falling edge(Low) as trigger types for trigger breakpoints.
- As combinations, you can specify that all specified break point conditions are satisfied(And), any one is satisfied(Or), or that they are satisfied in the order specified(Sequential).

- For the break condition setting area, there are following four types of break conditions that can be specified. Multiple break conditions can be specified at a time.

| Break Condition | Contents   |
|-----------------|--|
| Break Point     | Break occurs when a break point is reached.            |
| Stack Overflow  | Break occurs when the stack overflows or underflow.    |
| Trace Event     | Break occurs when there is a trace event.              |
| Trace Full      | Break occurs when write to trace memory has completed. |

- For the timer setting area, there are following four modes of timer operation that can be specified.

| Timer Operation | Contents   |
|-----------------|--|
| NotUse          | Select this when no timer is used.   |
| TimeOut         | Trace event occurs when the break point is not reached within specified time.            |
| TimeCount       | Trace event occurs when the specified time has elapsed since the start of the program.   |
| DelayCount      | Trace event occurs when the specified time has elapsed since the break point is reached. |

Furthermore, there are following two types of count sources that can be specified for the timer.

| Count Source | Contents   |
|--------------|--|
| TIME(10us)   | The emulator timer(fixed at 10us) is used as the count source. |
| CYCLE        | The MCU cycle is used as the count source.                     |

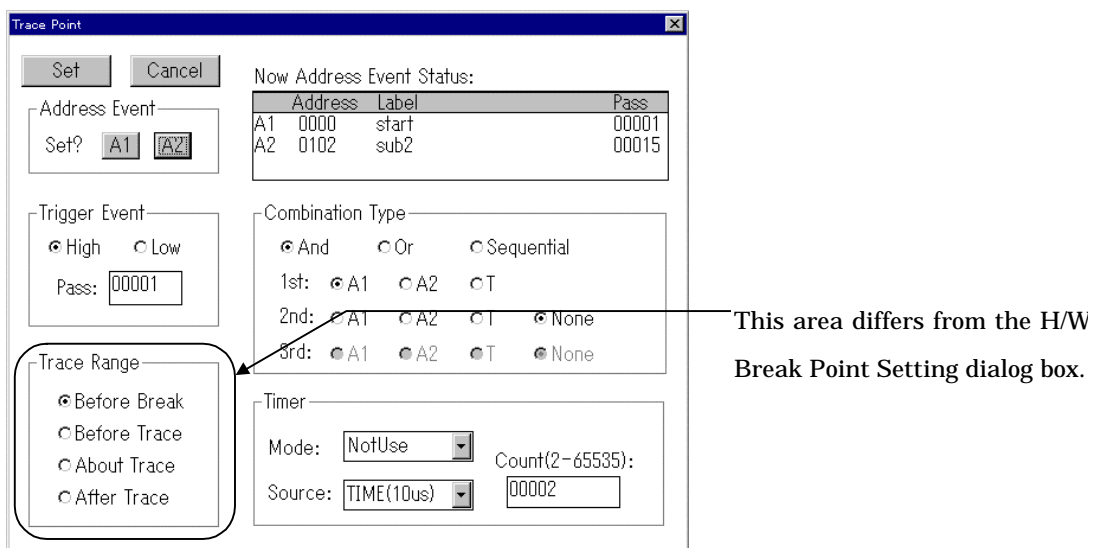


## 1.13 Trace Point Setting Dialog Box

Use the Trace Point Setting dialog box to set trace points. The method for setting trace points is almost identical to setting break points in the H/W Break Point Setting dialog box. See section 1.12.1, “Structure of H/W Break Point Setting Dialog Box” for settings other than those discussed below.

### 1.13.1 Structure of Trace Point Setting Dialog Box

The following describes only those points of the Trace Point Setting Dialog box screen configuration that differ from the H/W Break Point Setting dialog box.



#### 1.13.1.1 Setting the trace area

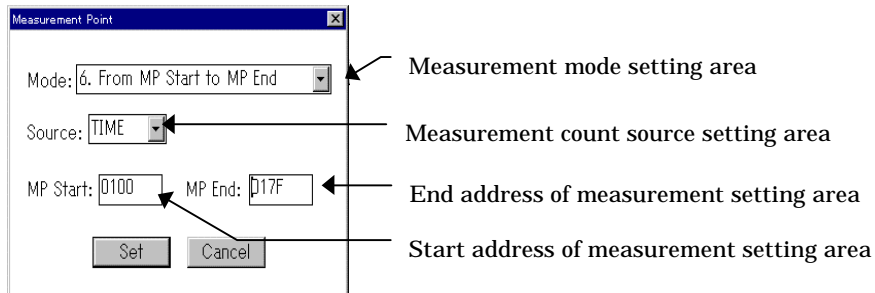
The Trace Point Setting dialog box differs from the H/W Break Point Setting dialog box in that the break condition setting area (Break Mode) in the latter is a trace range setting area (Trace Range) in the former. For the trace range setting area, you can specify a trace range with respect to a trace point. The trace area to be stored in memory varies with the trace range specified here.

| Trace Area   | Contents  |
|--------------|---|
| Before Break | Stores the 4K cycles (-4095 to 0 cycles) to the point at which the target program stops.  |
| Before Trace | Stores the 4K cycles (-4095 to 0 cycles) to the point at which the trace point is passed. |
| About Trace  | Stores the 2K cycles (-2048 to 2047 cycles) either side of the trace point.               |
| After Trace  | Stores the 4K cycles (0 to 4095 cycles) of trace data after the trace point.              |

## 1.14 Time Measurement point Setting Dialog Box

The Time Measurement Point Setting dialog box is used to set a time measurement point.

### 1.14.1 Structure of Time measurement Point Setting Dialog Box



- Following eight modes of time measurements are available to choose from.

| Measurement Mode                | Start Event                | End Event                |
|---------------------------------|----------------------------|--------------------------|
| 1. From Go to Break             | Start of program execution | End of program execution |
| 2. From Go to MP End            | Start of program execution | Measurement point End    |
| 3. Form Go to Trace Event       | Start of program execution | Trace Event              |
| 4. From Trace Event to MP End   | Trace Event                | Measurement point End    |
| 5. From Trace Event to Break    | Trace Event                | End of program execution |
| 6. From MP Start to MP End      | Measurement point Start    | Measurement point End    |
| 7. From MP Start to Trace Event | Measurement point Start    | Trace Event              |
| 8. From MP Start to Break       | Measurement point Start    | End of program execution |

- Following two types of time measurement count sources are available to choose from.

| Count Source | Contents   |
|--------------|--|
| TIME         | Use the PC4504's timer (16 MHz fixed) to measure time. |
| CYCLE        | Use the target MCU cycles to measure time.             |

# Basic Operation



# 1. Loading and Displaying the Target Program

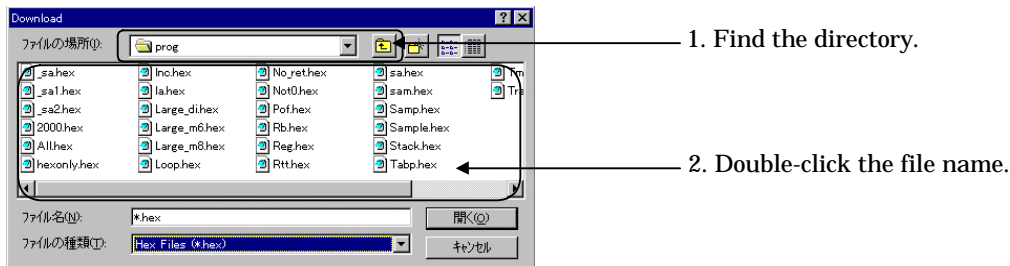
## 1.1 Downloading

To download the target program, select the following from the **PD45** Window:

[File] -> [Download] -> [Load module...]

The File Selection dialog box opens. Select the target program from the File Selection dialog box. The file to be downloaded is in the HEX files have the filename extension ".HEX".

You can also press Shift+F1 to open the File Selection dialog box for downloading files.



Note:

If you download the HEX file but the Program and Source Window does not switch to Source Program mode, the HEX file may not contain source line information. Check the options that you specified when assembling.

### To download only machine language data

To download only machine language data, select the following from the **PD45** Window menu:

[File] -> [Download] -> [Memory Image...]

The File Selection dialog box opens. Select the HEX file, then download the machine language data.

### To download only symbol data

To download only symbol data, select the following from the **PD45** Window menu:

[File] -> [Download] -> [Symbol...]

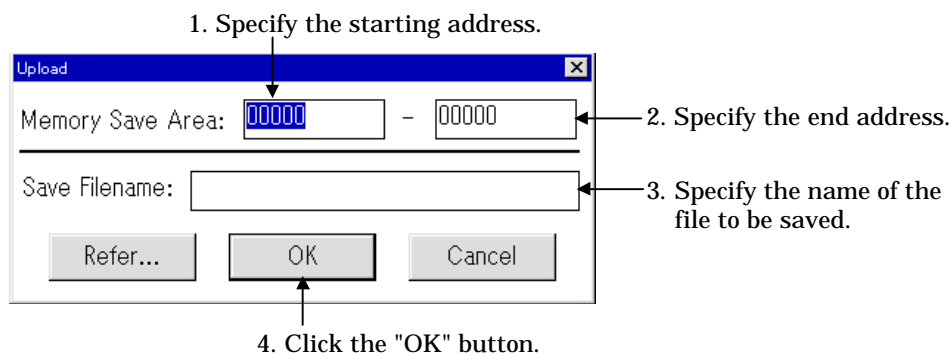
Select SYM file from the file Selection dialog box, then download the symbol data.

## 1.2 Uploading

From the **PD45** Window menu, select

[File]->[Upload...]

to open the Upload dialog box. Enter the upload area and the name of the file to be saved.



Note:

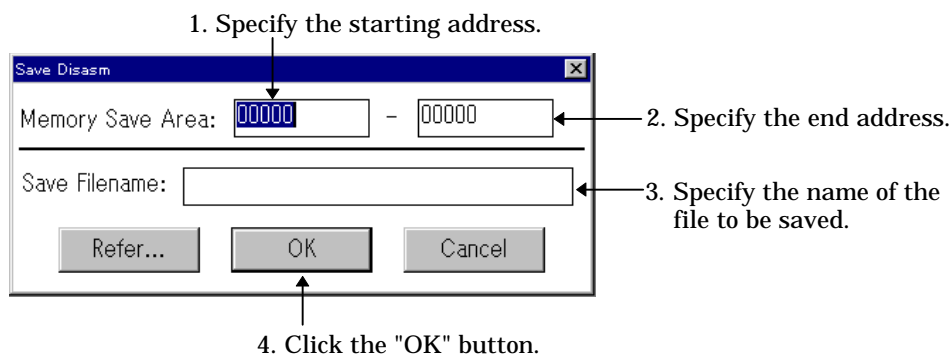
The minimum unit of addresses output to a file is one page. Therefore, if the specified start address is 7E16 and the end address is 10116, data in addresses ranging from 016 to 17E16 is stored in the output file.

## 1.3 Saving Results of Disassembly

From the **PD45** Window menu, select

[File]->[Save Disasm...]

to open the Save Disasm dialog box. Enter the area to be saved and the filename. You can specify any filename and attribute. If you specify an existing filename, that file is overwritten.



## 1.4 Continuing to Display a Selected Program Position

The Program Window always displays the target program at the position of the program counter and you cannot, therefore, choose to always display the same position. You can do so, however, in the Source Window. To open the Source Window, select the following from the **PD45** Window menu:

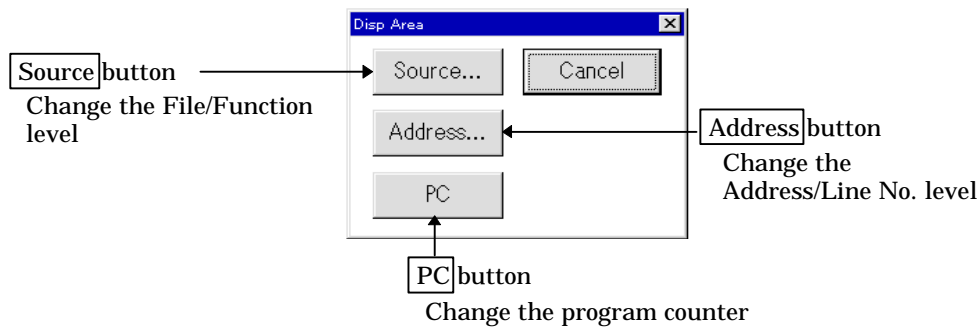
[Window]->[Source Window]

## 1.5 Changing the Program Display Position

You can display the source program in the Program Window and in the Source Window. To change the display position of the Program (or Source) Window, click on the "View" button in the toolbar of the Program (or Source) Window to open the Disp Area dialog box. You can only change the display position in the active window.



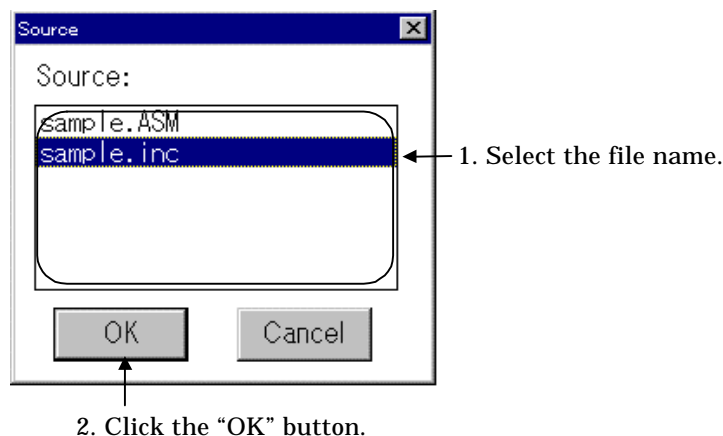
Click the "View" button to open the Disp Area dialog box.



If no debugging information has been read in, no changes can be made at the file level, or at the line No. level. Further, if the Program (or Source) Window is in disassemble display mode, no changes can be made at the line No. level.

### Changing the display position at the file level

Click the "Source" button in the Disp Area dialog box to open the Source dialog box. (The "Source" button cannot be clicked if no debugging information has been read in.) The Source dialog box shows the file structure of the downloaded target program. Click the name of the file to be changed.



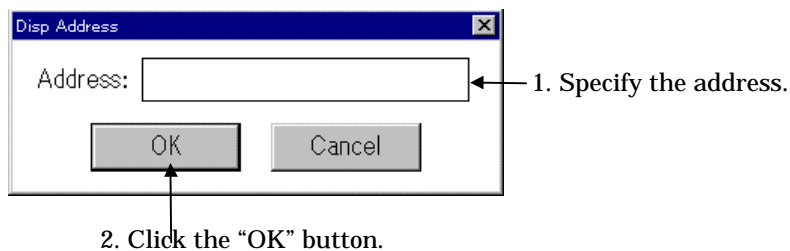
To open the Source dialog box, select the following from the **PD45** Window menu:  
[Option]->[View]->[Source]

You can also double-click the line No. display area of the Program (Source) Window to open the Source dialog box.

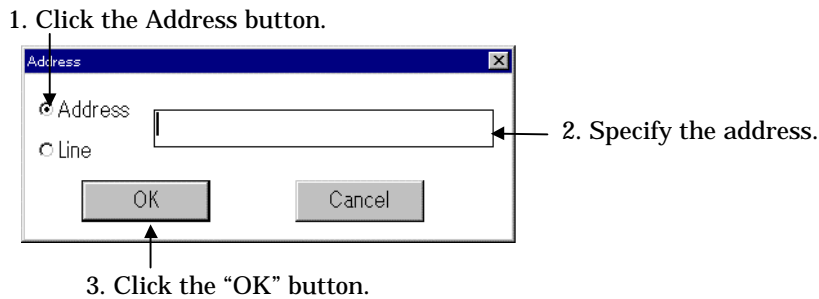
**Changing the display position at the address level**

Click the "Address" button from the Disp Area dialog box. Click the "Address" button to open the Disp Address dialog box. This dialog box is configured differently depending on the window's display mode.

- **Disassemble display mode**  
 If the display mode is the Disassemble mode, the Disp Address dialog box opens. Specify the address you want to be displayed in the input area and click the [OK] button.



- **Source display mode**  
 If the display mode is the Source mode, the Address dialog box opens. After clicking the [Address] button, specify the address you want to be displayed in the input area and click the [OK] button.



To open the Disp Address dialog box, select the following from the **PD45** Window menu:

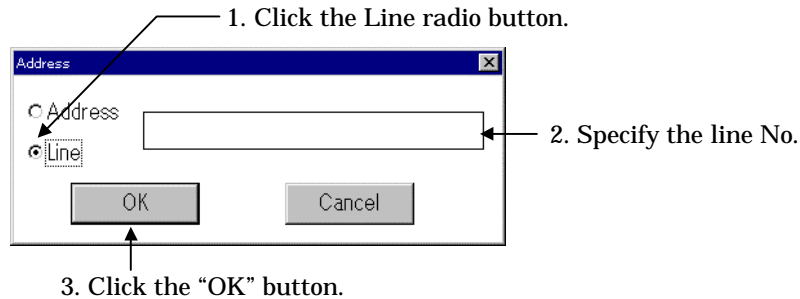
[Option]->[View]->[Address...]

You can also double-click the address display area of the Program (Source) Window to open the Disp Address(Address) dialog box.



### Changing the display position at the line No. level

When the Program (Source) Window is in source display mode, click the "Address" button from the Disp Area dialog box. Click the "Address" button to open the Address dialog box. Click the Line button in the Address dialog box to enter the new line No.



To open the Address dialog box, select the following from the **PD45** Window menu:  
[Option]->[View]->[Address...]

You can also double-click the address display area of the Program (Source) Window to open the Address dialog box.

### Changing the position of the program counter

Click the "PC" button in the Disp Area dialog box. When you click the "PC" button, the display position changes to the position of the program counter. Change the position of the program counter, select the following from the **PD45** Window:

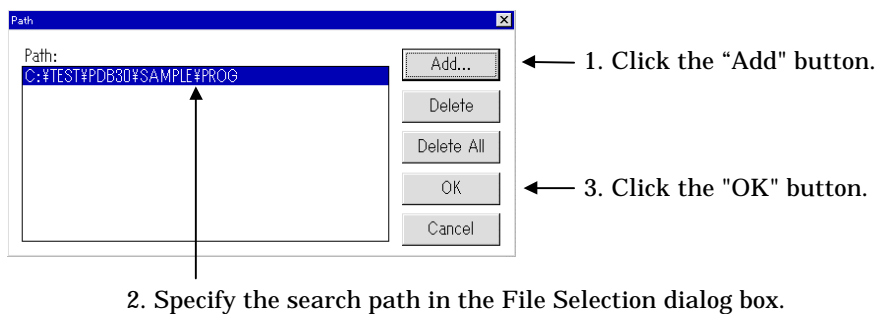
[Option]->[View]->[Program Counter]

## 1.6 Checking Source Programs in Other Directories

You can specify the search path for source files. This specification is valid when the target program's source files do not exist in the current directory. This function allows you to check these source files and set software breakpoints, etc., from the Program (or Source) Window. To open the Path dialog box and set the search path, select the following from the PD45 Window menu:

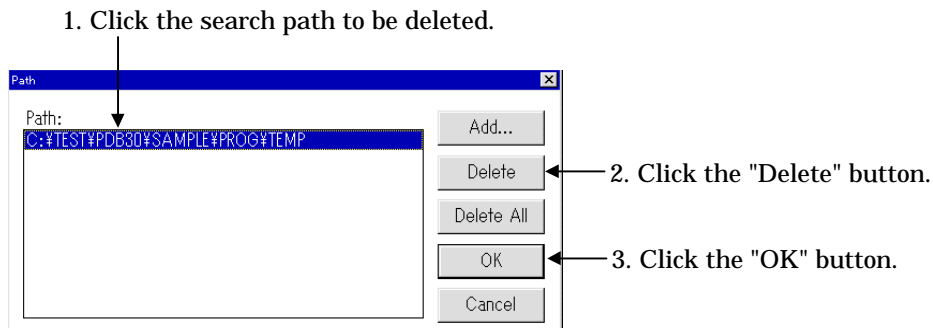
[Environ]->[Path...]

To add a search path, click the "Add" button in the Path dialog box. The File Selection dialog box opens. Use the mouse to select the name of the file to be checked.



### To delete a search path

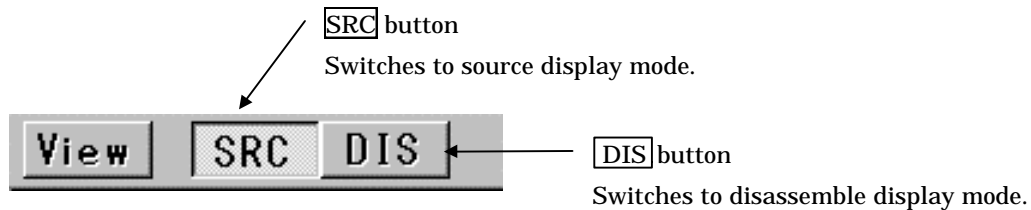
Open the Path dialog box, then carry out the following:



\*Click "Delete All" to delete all search paths.

## 1.7 Displaying the Results of Disassembling

To display the results of disassembling, click the "DIS" button on the Program (or Source) Window toolbar. The display mode can only be changed in the active window.



You can also select the following from the **PD45** Window menu:

[Option]->[Mode]->[Disasm Mode]

If, when the target program stops, the program counter is in an area with no source line data, the Program Window automatically switches to disassemble display mode.

### To revert from disassemble display to source program display mode

Click the "SRC" button on the Program (or Source) Window toolbar.

You can also switch to source program display mode by selecting the following from the **PD45** Window menu:

[Option]->[Mode]->[Source Mode]

Note:

If there is no source line data in the first line in the program display area of the Program (or Source) Window, you cannot switch from disassemble display mode to source display mode (the "SRC" buttons are invalid). In this case, use the vertical scrollbar or Up/Down cursor keys to move the starting address of the program display area to a position where there is source line data.

## 2. Starting and Stopping Target Program Execution

### 2.1 Starting and Stopping

#### To start the target program

Click the "Go" button on the **PD45** Window toolbar. You can also press the "F1" key.



Click the "Go" button.

You can also start the target program by selecting the following from the **PD45** Window menu:

[Debug]->[Go]

The target program runs until it reaches a breakpoint.

#### Free-running the target program

Execute the Go (G) command from the Script Window. Although the Go command basically executes a program with breaks included, the target program can be executed in free-running mode if breakpoints are omitted. For details of using script commands, see Section 9.1, "Executing Script Commands" in the Basic Operation part.

#### Input format(Free Run):

Format 1: Go

Format 2: Go starting\_address

#### Functions:

- When no starting address is specified, runs the target program from the address currently indicated by the program counter (format 1);
- Runs the target program from the specified starting address (format 2);

### To stop the target program

Click the "Stop" button on the **PD45** Window toolbar.



Click the "Stop" button.

You can also stop the target program by selecting the following from the **PD45** Window menu:

[Debug]->[Stop]

Note:

If, when the target program stops, the program counter is in an area with no source line data, the Program Window automatically switches to disassemble display mode.

### To start the target program from a specific address

Select the following from the **PD45** Window menu:

[Debug]->[Go]->[Go Option...]

The Go dialog box opens. Enter the starting address.

### To check if the target program is running

The current execution status is displayed at the right end of the status bar in the **PD45** Window.

MCU : RUN Target program is running.

MCU : STOP Target program is stopped.

### To perform debugging associated with timer operation

The control mode that is assumed when the target program has stopped must be changed to the clock stop mode (STOP). In the clock stop mode, the clock is stopped when the target program stops running. This allows you to perform debugging that is associated with timer operation. Choose the following from the PD45 window menu:

[Debug] -> [Go] -> [Go Mode] -> [Stop]

### To perform debugging associated with clock operation

The control mode that is assumed when the target program has stopped must be changed to the DI instruction insert mode (DI). In the DI instruction insert mode, clock is supplied when the target program stops running. This allows you to debug peripheral equipment that uses the internal clock of an LCD, etc. Choose the following from the PD45 window menu:

[Debug] -> [Go] -> [Go Mode] -> [Di]

## 2.2 Step Execution

Click the "Step" button on the **PD45** Window toolbar. Or press F3.



Click the "Step" button.

When the Program Window is in source program mode, each line of the source program is executed, step by step. When the Program Window is in disassemble mode, each instruction is executed, step by step.

You can also start step execution by selecting the following from the **PD45** Window menu:  
 [Debug]->[Step]->[Step]

### To execute subroutines as single instructions in step mode

Executing subroutines as single instructions in step mode is called as over-step execution. Click the "Over" button on the **PD45** Window toolbar. Or press F4.



Click the "Over" button.

You can also start over-step execution by selecting the following from the **PD45** Window menu:

[Debug]->[Over]->[Over]

Note:

A subroutine that does not return from a subroutine to the current routine (e.g., one that has fallen into an endless loop) cannot be completed of Over-step execution. To suspend Over-step execution in such a case, press the system reset button of the PC4504 emulator.

### To specify the time of steps

Select the following from the **PD45** Window menu:

[Debug]->[Step]->[Step Option...]

(For over-step execution, select [Debug]->[Over]->[Over Option...].)

Specify the time of steps in the displayed Step (or Over) dialog box.

### To stop step execution

Click the "Stop" button on the toolbar. You can also stop step execution by selecting the following from the **PD45** Window menu:

[Debug]->[Stop]

This also applies to over-step execution.

## 2.3 Returning from Current to Calling Routine

Click the "Return" button on the **PD45** Window toolbar (called as Return execution). You can also press F5.



Click the "Return" button.

You can also return to a calling routine by selecting the following from the **PD45** Window menu:

[Debug]->[Return]

### To stop return execution

Click the "Stop" button on the toolbar. You can also stop return execution by selecting the following from the **PD45** Window menu:

[Debug]->[Stop]

## 2.4 Program Execution to Specified Location

To run the target program to the specified position (known as Come Execution), first click the line at which you want the program to stop in the program display area of the Program (Source) Window to specify the cursor position. Note, however, that Come Execution cannot be carried out if the cursor is positioned on a line in which no software breakpoint can be set (comment lines, and data definition lines, etc.).

| Line  | BRK | Source                             |
|-------|-----|------------------------------------|
| 00022 |     | start:                             |
| 00023 | -   | D1                                 |
| 00024 |     | loop:                              |
| 00025 | -   | NOP                                |
| 00026 |     | .include sample.inc ;Load INC File |
| 00027 | -   | NOP                                |
| 00028 | -   | BM sub1                            |
| 00029 | -   | NOP                                |
| 00030 | -   | BM sub2                            |
| 00031 | -   | NOP                                |
| 00032 | -   | NOP                                |
| 00033 |     |                                    |
| 00034 | -   | LZ zxy000                          |

Cursor position set by clicking the mouse.

To start Come execution, click the "Come" button on the **PD45** Window toolbar. You can also press F2.



Click the "Come" button.

You can also start Come execution by selecting the following from the **PD45** Window menu:

[Debug]->[Come]

Note:

In Come execution, the breakpoints specified for software and hardware breaks are ignored.

### To stop come execution

Click the "Stop" button on the toolbar. You can also stop Come execution by selecting the following from the **PD45** Window menu:

[Debug]->[Stop]

## 2.5 Resetting the Program

To reset the target program, click the "Reset" button on the **PD45** Window toolbar. You can also press F8.



Click the "Reset" button.

You can also reset the target program by selecting the following from the **PD45** Window menu:

[Debug]->[Reset]



## 3. Software Breaks

Use a software break to break target program execution at a specified line (address). Use the S/W Break Point Setting dialog box to set a software break point. When you set a software break, execution of the target program stops immediately before the software break point.

- You can set up to 8 software break points.
- If you set multiple software break points, the target program stops as soon as execution reaches any of the software break addresses.

### 3.1 Enabling S/W Breaks

Click the [BM] button on the PD45 window's tool bar to change the break mode to "S/W break." The [BM] button toggles between "S/W break" and "H/W break."



Click the "BM" button.

If the current break mode is "S/W break," you will find that the break mode shown in the right part of the PD45 window's status bar is changed as follows:

BM : S/W

The break mode can also be switched over using the following commands on the PD45 window menu:

[Debug] -> [Break Mode]

### 3.2 Opening the S/W Break Point Setting Dialog Box

Click the "SW" button on the **PD45** Window toolbar. Press function key F7 to open the software breakpoint setting dialog box.



Click the "SW" button.

You can also open the S/W Break Point Setting dialog box by selecting the following from the **PD45** Window menu:

[Debug]->[Break Point]->[S/W Break Point...]

See Section 1.11, "S/W Break Point Setting Dialog Box" in the Introduction for the structure of the S/W Break Point Setting dialog box.

### 3.3 Setting a Break Point

To specify an address as a break point

In the S/W Break Point Setting dialog box, specify an address or label in the Address field, as follows:

1. Click the "Address" button.

2. Specify the address.

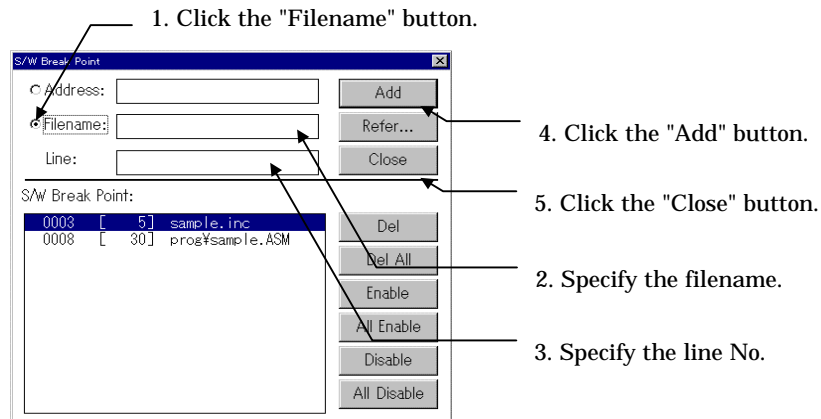
3. Click the "Add" button.

4. Click the "Close" button.

The image shows the 'S/W Break Point' dialog box. It has a title bar with a close button. Below the title bar are three input fields: 'Address:', 'Filename:', and 'Line:'. To the right of these fields are three buttons: 'Add', 'Refer...', and 'Close'. Below the input fields is a list titled 'S/W Break Point:' containing two entries: '0003 [ 5] sample.inc' and '0008 [ 30] prog\$sample.ASM'. To the right of the list are several buttons: 'Del', 'Del All', 'Enable', 'All Enable', 'Disable', and 'All Disable'. Arrows point from the numbered text to the 'Address' field, the 'Add' button, and the 'Close' button.

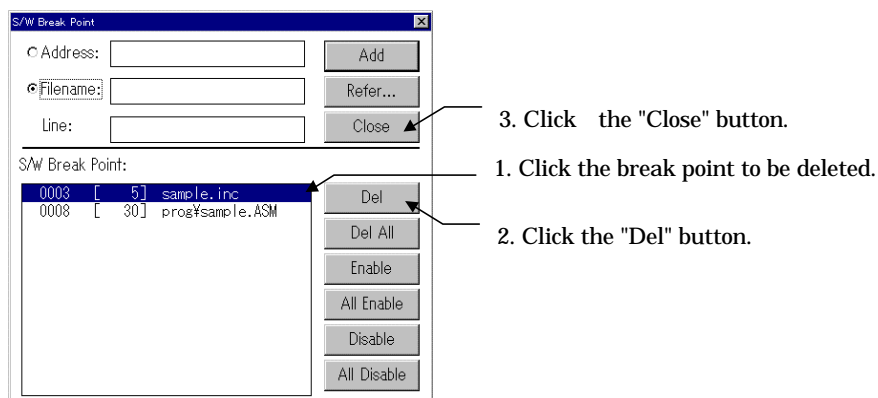
## To specify a line No. as a break point

In the S/W Break Point Setting dialog box, specify a filename and a line No, as follows:



### 3.4 Deleting a Break Point

In the S/W Break Point Setting dialog box, delete the break point as follows:

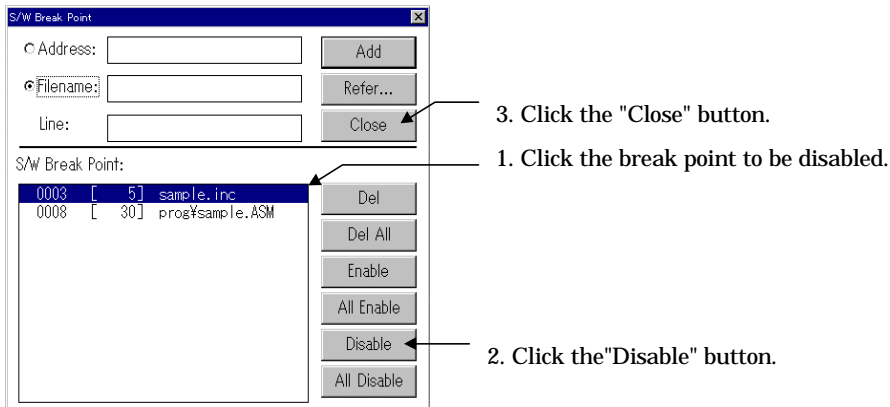


\* Click "Del All" to delete all break points.

You can also click the breakpoint to be deleted, then press the Delete key.

### 3.5 Temporarily Disabling Break Points

In the S/W Break Point Setting dialog box, disable the break point as follows:

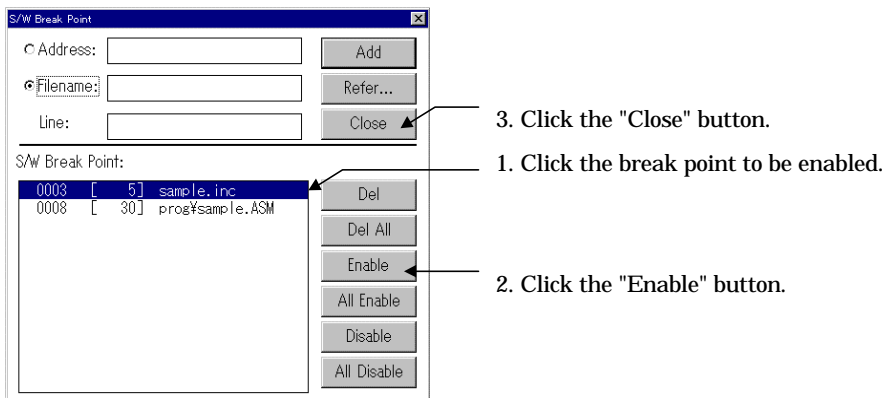


\* Click "All Disable" to disable all break points.

You can also double-click the breakpoint to be disabled. (An asterisk (\*) is displayed).

### 3.6 Temporarily Enabling Break Points

In the S/W Break Point Setting dialog box, enable the break point as follows:



\* Click "All Enable" to enable all break points.

You can also double-click the breakpoint to be enabled. (An asterisk (\*) is deleted).

### 3.7 Setting a Break Point from Program (Source) Window

You can also set break points in the Program or Source Window. To do so, double-click the break point setting display area (BRK column) (indicated by "-") for the line in which you want to set the break.

| Line  | BRK | Source                             |
|-------|-----|------------------------------------|
| 00022 |     | start:                             |
| 00023 | -   | DI                                 |
| 00024 |     | loop:                              |
| 00025 | -   | NOP                                |
| 00026 |     | .include sample.inc ;Load INC File |
| 00027 | -   | NOP                                |
| 00028 | B   | BM sub1                            |
| 00029 | -   | NOP                                |
| 00030 | -   | BM sub2                            |
| 00031 | -   | NOP                                |
| 00032 | -   | NOP                                |
| 00033 |     |                                    |

Double-click here.

Lines in which a software break point have been set are marked by a "B" in place of the "-" in the break point setting display area (BRK column). You can delete the break point by double-clicking again in the BRK column.

### 3.8 Setting Breakpoints from the Toolbar

You can set breakpoints from the toolbar in the **PD45** Window. In the Program (Source) Window, click the program display area of the line in which you want to set the break (lines for which "-" is displayed in the corresponding breakpoint display area).

To set the breakpoint, click the "Break" button in the **PD45** Window toolbar.



Click the "Break" button.

You can also set a breakpoint by selecting the following from the **PD45** Window menu:

[Debug]->[Break Point]->[Break]

The mark on the lines set as software breakpoints changes from "-" to "B" in the breakpoint display area of the window. You can cancel a software breakpoint by again clicking the line indicated by "B" then clicking the "Break" button.

## 4. Hardware Breaks(1)

Use hardware breaks to break target program execution when memory is accessed. There are two types of hardware breaks: an address break where a break occurs when execution of a specified address (Fetch) is detected and a trigger break where a break occurs when the rising or falling edge of a signal input from an external trace cable is detected. Address and trigger breaks can be used in combination.

The method of setting hardware break points differs according to the emulator you are using. See section 1.12.1, "Structure of H/W Break Point Setting Dialog Box" for settings other than those discussed below.

In addition to the above, there are several other hardware breaks available: a break for failure such as a stack overflow or a break based on a timer count. Refer to Section 1, "Hardware Break (2)" in the High-end Debugging Part of this manual.

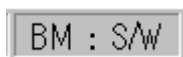
### 4.1 Enabling H/W Breaks

Click the [BM] button on the PD45 window's tool bar to change the break mode to "H/W break." The [BM] button toggles between "S/W break" and "H/W break."



↑  
Click the "BM" button.

If the current break mode is "H/W break," you will find that the break mode shown in the right part of the PD45 window's status bar is changed as follows:



The break mode can also be switched over using the following commands on the PD45 window menu:

[Debug] -> [Break Mode]

## 4.2 Opening the H/W Break Point Setting Dialog Box

Click the "HW" button on the **PD45** Window toolbar. Press function key Shift + F7 to open the hardware breakpoint setting dialog box.



Click the "HW" button.

You can also open the H/W Break Point Setting dialog box by selecting the following from the **PD45** Window menu:

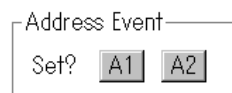
[Debug]->[Break Point]->[H/W Break Point...]

## 4.3 Setting Address Breakpoints

The PC4504 emulator allows you to set two address breakpoints.

### Setting an Address Break Point

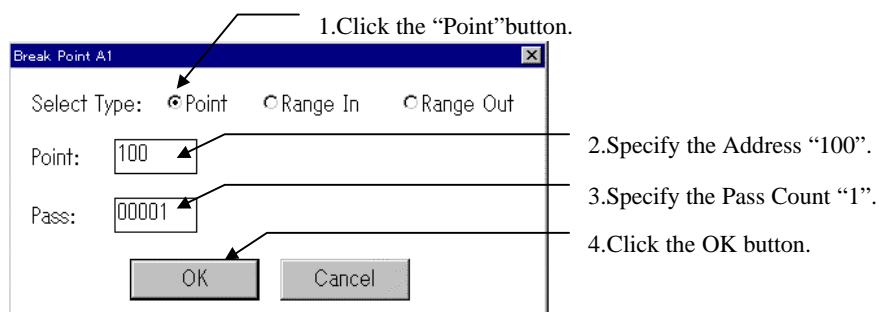
To set an address break point, click on any of buttons "A1" to "A2" in [Address Event] group in the H/W Break Point Setting dialog box to open the Address Break Point Setting dialog box, then set the address break point.



After setting an address breakpoint, click the [OK] button in the H/W Break Point Setting dialog box to enable the hardware break you have set.

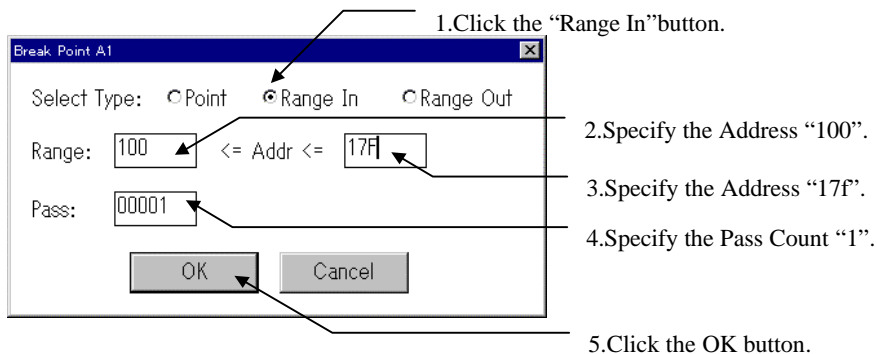
### To break execution on fetching a command at the specified address

The following shows how to set a break point for the third time the command at address "100<sub>16</sub>" is fetched.



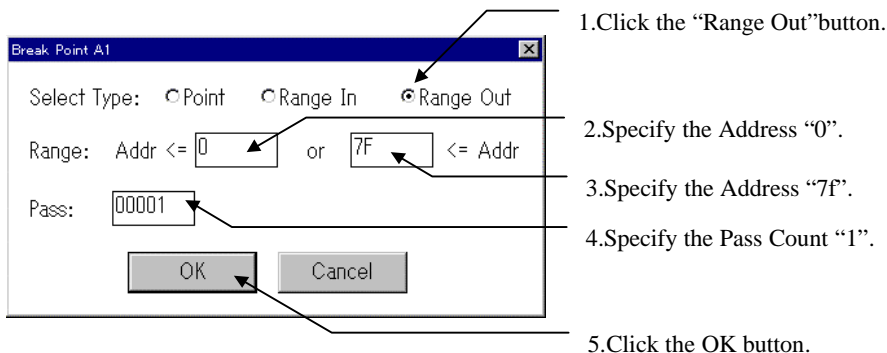
**To set an instruction fetch inside a specified address range as break condition.**

The following shows a setting example where a break occurs when any instruction at addresses 100<sub>16</sub> to 17F<sub>16</sub> is executed (fetched). Use the Address Breakpoint Setting dialog box to enter this setting.



**To set an instruction fetch outside a specified address range as break condition**

The following shows a setting example where a break occurs when any instruction at addresses outside the range of 0<sub>16</sub> to 7F<sub>16</sub> is executed (fetched). Use the Address Breakpoint Setting dialog box to enter this setting.

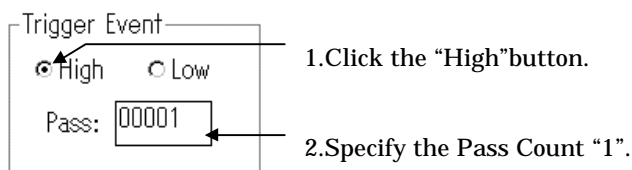


**4.4 To set a trigger break point**

The PC4504 emulator allows you to set one trigger breakpoint. Before using a trigger break, prepare an external trace cable. A trigger signal is taken in from this cable.

**To set a trigger break point**

The following shows a setting example where a break occurs at a rising edge of the input signal (High). Use the H/W Breakpoint Setting dialog box to enter this setting.



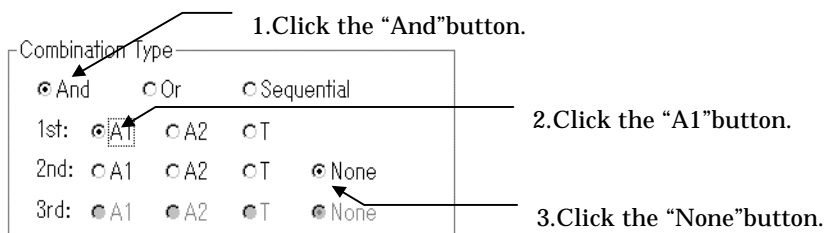


## 4.5 To combination break point

There are three combinatorial conditions that can be specified. These include "And" where a break occurs when all conditions are met, "Or" where a break occurs when one of conditions is met, and "Sequential" where a break occurs when conditions are set in a specified sequence. Use Combination Type in the H/W Break Point Setting dialog box to set these combinatorial conditions.

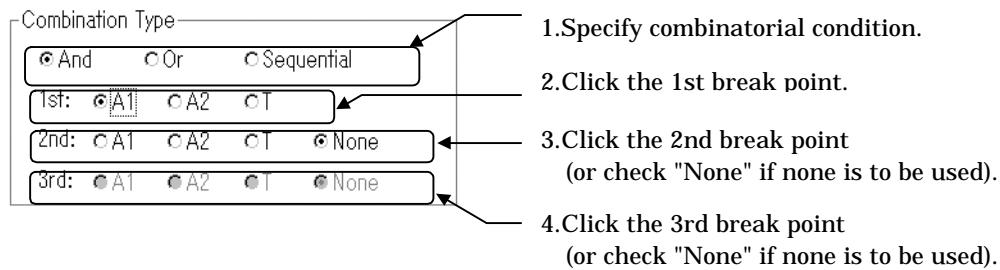
### To enable a break point (one only)

The following shows a setting example where only address breakpoint A1 is enabled.



### To combine break points

You can specify any of three types of combination: when all conditions for the specified break point are satisfied (And), when any one is satisfied (Or), or when they are satisfied in the specified order (Sequential).



## 5. Real Time Tracing

In **PD45**, you can check the trace results using any of three display modes: bus mode in which you can check the bus data each cycle, disassemble mode, in which executed instructions are disassembled for display, and source mode, in which executed instructions are displayed at the source line level. The trace results are shown in the Trace Window. To open the Trace Window, select

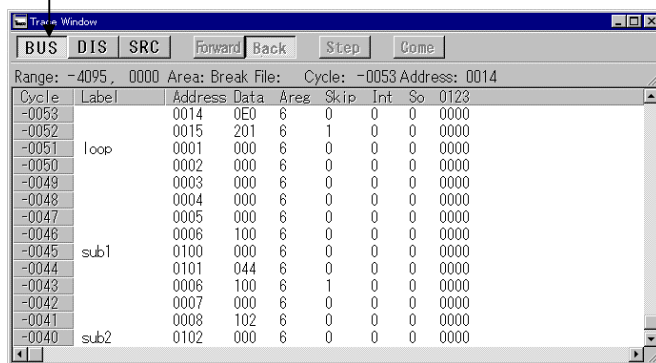
[Trace Window] -> [Window]

in the menu in the **PD45** window.

### 5.1 To check bus data each cycle

Click the "BUS" button on the toolbar in the Trace Window to switch to bus display mode.

Click "BUS" to change to bus mode.



Use the Up and Down cursor keys, the Page Up and Page Down keys, or scroll using the vertical scroll bar to check the bus data outside the displayed cycle. See Section 1.9.1 "Screen Configuration in Bus Mode" in the Window function for details of bus mode.

#### To change the reference position by specifying the trace cycle

Double-click the cycles area in the Trace Window to open the Cycle dialog box. Input the number of cycles. You can also open the Cycle dialog box from the **PD45** Window menu:

[Option]->[View]->[Cycle]

### To search for a specified address

Double-click the address area in the Trace Window to open the Address Search dialog box. Input the address to search. You can also open the Address Search dialog box from the menu in the **PD45** Window:

[Option]->[View]->[Address Search]

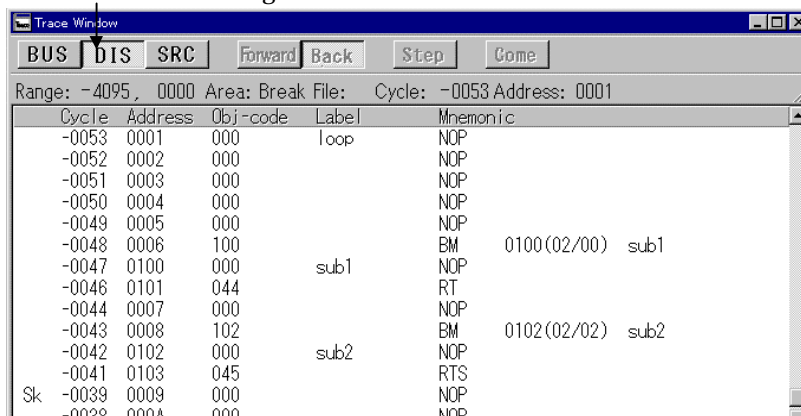
### To check the status of the external signal

Connect the external trace signal cable supplied with the **PC4504** emulator. The status of the external signal is displayed in the "0123" area of the Trace Window.

## 5.2 To check executed Instructions

Click the "DIS" button on the toolbar in the Trace Window to switch to disassemble display mode.

Click "DIS" to change to disassemble mode.



| Cycle | Address | Obj-code | Label | Mnemonic            |
|-------|---------|----------|-------|---------------------|
| -0053 | 0001    | 000      | loop  | NOP                 |
| -0052 | 0002    | 000      |       | NOP                 |
| -0051 | 0003    | 000      |       | NOP                 |
| -0050 | 0004    | 000      |       | NOP                 |
| -0049 | 0005    | 000      |       | NOP                 |
| -0048 | 0006    | 100      |       | BM 0100(02/00) sub1 |
| -0047 | 0100    | 000      | sub1  | NOP                 |
| -0046 | 0101    | 044      |       | RT                  |
| -0044 | 0007    | 000      |       | NOP                 |
| -0043 | 0008    | 102      |       | BM 0102(02/02) sub2 |
| -0042 | 0102    | 000      | sub2  | NOP                 |
| -0041 | 0103    | 045      |       | RTS                 |
| Sk    | -0039   | 0009     |       | NOP                 |
|       | -0038   | 0008     |       | NOP                 |

Use the Up and Down cursor keys, the Page Up and Page Down keys, or scroll using the vertical scrollbar to check the disassemble data outside the displayed area.

### To change the reference position by specifying the trace cycles

Double-click the cycles area in the Trace Window to open the Cycle dialog box. Input the number of cycles. You can also open the Cycle dialog box from the **PD45** Window menu:

[Option]->[View]->[Cycle]

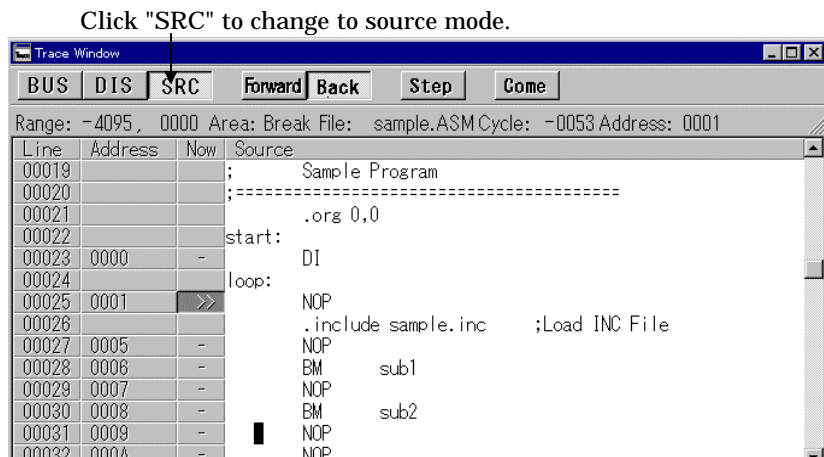
### To search for the cycle in which an operation code was fetched to a specified address

Double-click the address area in the Trace Window to open the Address Search dialog box. Input the address to search. You can also open the Address Search dialog box from the menu in the **PD45** Window:

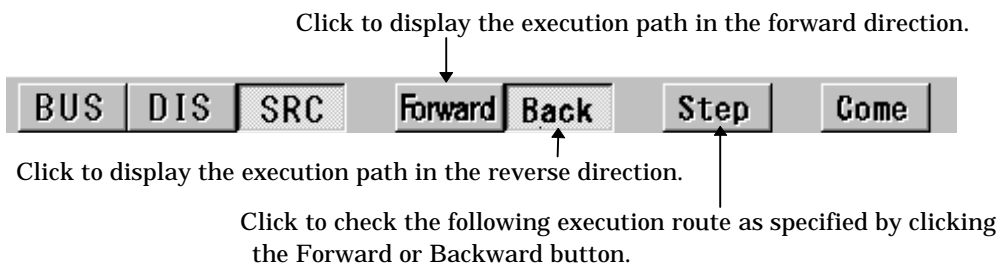
[Option]->[View]->[Address Search]

### 5.3 To check execution path at the source level

Click the "SRC" button on the toolbar in the Trace Window to switch to source display mode.



Click the "Back" button on the toolbar in the Trace Window to check the execution path in reverse direction, then click the "Step" button. Or click the "Forward" button on the toolbar in the Trace Window to check the execution path in the forward direction, then click the "Step" button.



**To change the reference position by specifying the trace cycles**

Double-click the cycles area in the Trace Window to open the Cycle dialog box. Input the number of cycles. You can also open the Cycle dialog box from the PD45 Window menu:

[Option]->[View]->[Cycle]

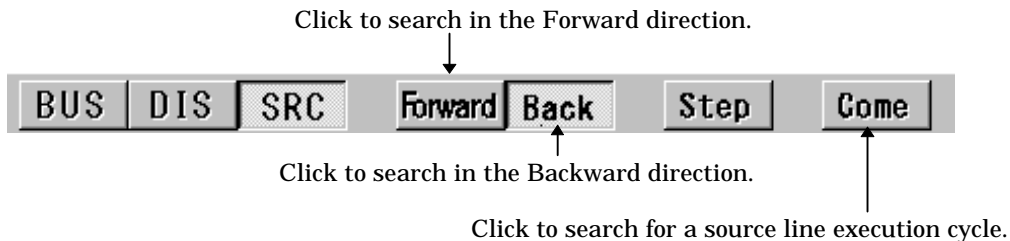
**To search for the cycle in which an operation code was fetched to a specified address**

Double-click the address area in the Trace Window to open the Address Search dialog box. Input the address to search. You can also open the Address Search dialog box from the menu in the PD45 Window:

[Option]->[View]->[Address Search]

To search for the cycle in which an operation code was fetched to a specific source line

Click the mouse on a source line in the source file area in the Trace Window, then click the "Come" button to start the search. The search is performed in the direction currently specified as the execution path ("Forward" for forward searches, and "Back" for backward searches).



## 5.4 Changing the Real-Time Trace Position to be Saved

**PD45** stores the results of real-time traces of target programs. By default, the trace range is the 4K cycles to just before the target program stops (Break). However, **PD45** also allows you to specify the 4K cycles up to satisfaction of the trace point condition (Before), the 16K cycles centered on the satisfaction of the trace point condition (About), or the 4K cycles after the satisfaction of the trace point condition (After). In this case, the trace point is, like the hardware break point, the point at which real-time trace are performed when the system detects that data is written to memory, read from memory, or an instruction is fetched, or detects the rising or falling edge of the signal input from the external trace cable.

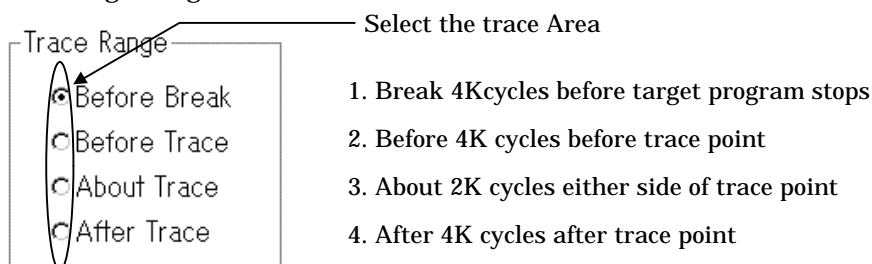
### Setting a Trace Point

To set a trace point, open the Trace Point Setting dialog box by selecting [Trace Point] -> [Debug] from the menu in the **PD45** Window.

For details on how to set trace points, Section 4, "Hardware Break (1) in the basic operation part of this manual and Section 1, "Hardware Break (2)" in the High-end Debugging Part.

### Setting the Trace Area

To change the trace area, change the setting in the "Trace Area" group in the Trace Point Setting dialog box.



# 6. Time Measurement

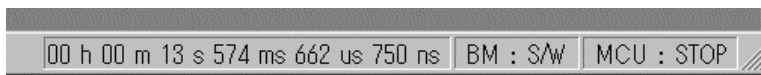
Use the time measurement function to measure the target program's execution time. The measured execution time of the target program is displayed on the status bar of the PD45 window. Time measurement points are set from the Time Measurement Point Setting dialog box.

The default time measurement mode is "1. From Go to Break" in which a duration of time from start of execution till end of execution is measured. The measurement count source is "TIME," i.e., the PC4504's internal timer (16 MHz fixed) is used to measure a time. The Time Measurement Point Setting dialog box is opened by choosing the following commands from the PD45 window menu:

[Debug] -> [Measurement Point...]

## 6.1 Checking Program Execution Time

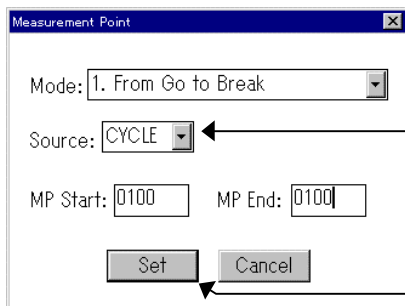
The measured execution time of the target program is displayed on the status bar of the PD45 window. The contents displayed here are updated when the target program has stopped.



Display the target program's execution time (TIME mode).

### To reference execution cycle

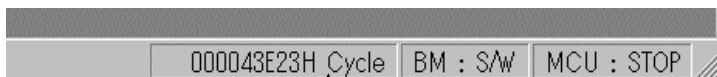
Enter the following settings from the Time Measurement Point Setting dialog box:



1. Change the measurement count source to "CYCLE."

2. Click the [Set] button.

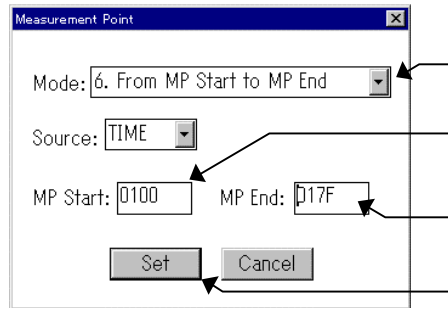
When the target program is executed after making the above settings, the status bar of the PD45 window changes as shown below.



Display the target program's execution time (CYCLE mode).

## 6.2 Measuring Execution Time of A Selected Address Range

Enter the following settings from the Time Measurement Point Setting dialog box:



1. Change the measurement mode to "6. From MP Start to MP End."
2. Specify the measurement start address.
3. Specify the measurement end address.
4. Click the "OK" button.

### To use a debug event as a time measurement point

The PC4504 emulator has the following eight modes of time measurements available. To switch over the time measurement modes, change the Mode area in the Time Measurement Point Setting dialog box to your desired mode.

| No. | Measurement Mode             | Start Event                | End Event                |
|-----|------------------------------|----------------------------|--------------------------|
| 1.  | From Go to Break             | Start of program execution | End of program execution |
| 2.  | From Go to MP End            | Start of program execution | Measurement point End    |
| 3.  | Form Go to Trace Event       | Start of program execution | Trace Event              |
| 4.  | From Trace Event to MP End   | Trace Event                | Measurement point End    |
| 5.  | From Trace Event to Break    | Trace Event                | End of program execution |
| 6.  | From MP Start to MP End      | Measurement point Start    | Measurement point End    |
| 7.  | From MP Start to Trace Event | Measurement point Start    | Trace Event              |
| 8.  | From MP Start to Break       | Measurement point Start    | End of program execution |

If you selected No. 6, 7, or 8 for the measurement mode number, you need to specify a start address of measurement in the MP Start area of the Time Measurement Point Setting dialog box. Or if you selected No. 2, 4, or 6 for the measurement mode number, you need to specify an end address of measurement in the MP End area of the dialog box.

#### Note:

If "End of program execution" is chosen for the end of measurement point, measurement is taken of time until execution of the broken instruction is ended. In all other cases, measurement is taken of time until conditions are met for the end of measurement event.

#### Supplements:

When measuring the interrupt intervals of a program that uses a timer interrupt, follow the measurement procedure described below:

- (1) Change the control mode assumed when the target program has stopped to "STOP," a clock stop mode.
- (2) Change the time measurement mode to "1. From Go to Break."
- (3) After resetting the target program, specify the start address of the timer interrupt routine at a breakpoint and execute the target program with one pass count.

- (4) After resetting the target program, specify the start address of the timer interrupt routine at a breakpoint and execute the target program with two pass counts.
- (5) Subtract the execution time measured in (3) from the execution time measured in (4).

The result obtained in (5) is the interrupt interval.



## 7. Checking and Setting Register Data

### 7.1 Checking the Contents of Registers

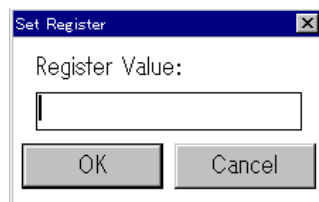
Open the Register Window to check the contents of the registers. The Register Window, which lists the internal registers of the CPU that are common to MCU's, is opened from the **PD45** Window menu by selecting the following:

[Window]->[Register Window]

See Section 1.4, "Register Window" in the Window Functions for details of the Register Window.

### 7.2 Setting the Contents of Registers

To set the contents of the registers, click the register buttons in the Register Window to open the Register dialog box, then input the new value.



#### To change the values of flags

The Set Register dialog box does not open if you clicked the [CY] button of the Register Window. The value of the carry flag changes each time you click the [CY] button.

# 8. Checking and Setting Memory Contents

## 8.1 Checking the Value at a Specified Address

Use the watch function to check the value at a specified address. Check the value at the specified address in the ASM Watch Window. The address is called as the watch point. To open the ASM Watch Window, select the following from the **PD45** Window menu:

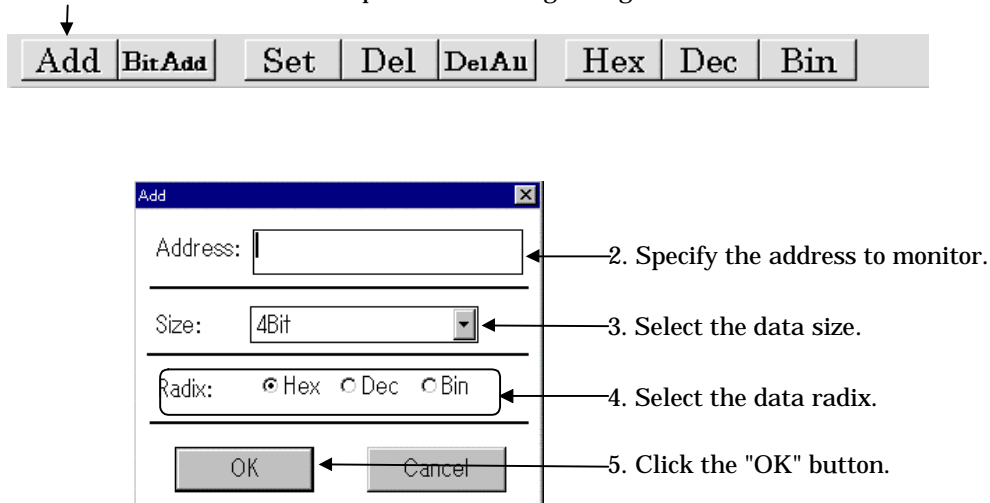
[Window]->[ASM Watch Window]

See Section 1.7, "ASM Watch Window" in the Window Functions for details of the ASM Watch Window.

### To register a watch point

Click the "Add" button in the menu bar in the ASM Watch Window to open the Add dialog box. Input the address to monitor.

1. Click the "Add" button to open the following dialog box.

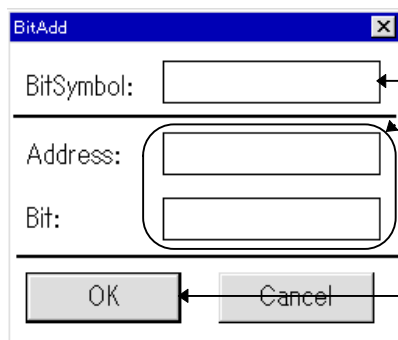


The registered watchpoint is added at the current cursor position in the ASM Watch Window. The cursor position is indicated by a red mark in the address display area and data display area. You can move the cursor by clicking on either area or using the **Up** and **Down** keys.

**To register a specific bit as a watch point**

Click the "BitAdd" button in the menu bar in the ASM Watch Window to open the BitAdd dialog box. Input the bit symbol or the address and bit No. to monitor.

1. Click the "BitAdd" button to open the following dialog box.



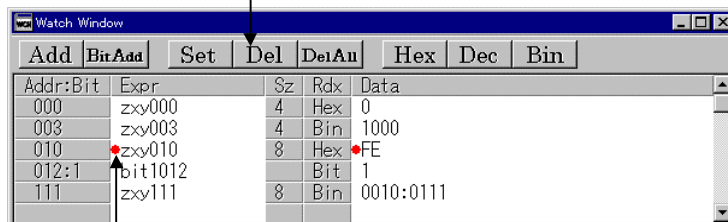
2. Specify the bit symbol. If none has been defined, Specify the address and bit No. in the next fields.

3. Click the "OK" button.

**To change the value at a specified address**

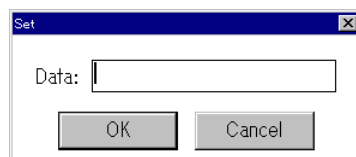
Select the watch point to change in the ASM Watch Window, then click the "Set" button on the toolbar of the ASM Watch Window.

2. Click the "Set" button.



1. Click the watch point to be changed.

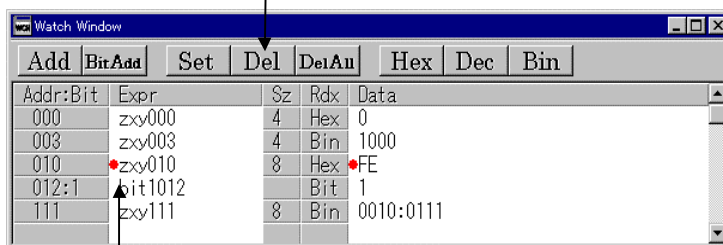
The Set dialog box opens. Enter the new value.



**To delete a watch point**

Select the watch point to be deleted in the ASM Watch Window, then click the "Del" button on the toolbar of the ASM Watch Window.

2. Click the "Del" button.



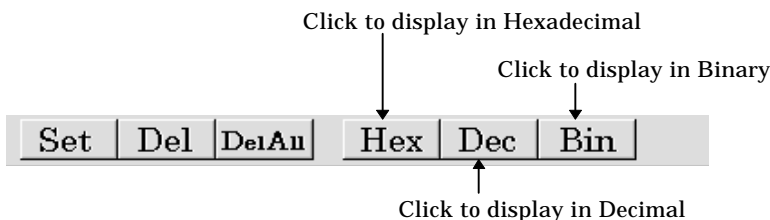
1. Click the watch point to be deleted.

\* To delete all watch points, click the "DelAll" button.

You can also click the watch point to be deleted, then press the Delete key.

**To change the memory display format**

You can change the radix in which data is displayed to select the watch point to be changed in the ASM Watch Window, then clicking "Hex", "Dec", or "Bin" in the toolbar of the ASM Watch Window.



You can also double-click the radix display area in the ASM Watch Window. In this case, the radix changes cyclically in order of the following:

Hexadecimal -> Decimal -> Binary -> Hexadecimal

## 8.2 To reference the contents of contiguous addresses

To reference the contents of contiguous addresses, open the Memory or Dump Window. The Memory Window displays contiguous memory contents in the form of "Address," "Symbol," or "Data (memory contents)." The Dump Window displays contiguous memory contents in dump form. In either case, RAM contents are displayed by default.

To use the Memory Window, select the following from the **PD45** Window menu:

[Window]->[Memory Window]

To use the Dump Window, select the following from the **PD45** Window menu:

[Window]->[Dump Window]

See Section 1.5, "Memory Window" in the Window Functions for details of the Memory Window, and Section 1.6, "Dump Window" in the Window Functions for details of the Dump Window.

### To reference a ROM address

Click the [ROM] button on the Memory or Dump Window tool bar.



Click the ROM button.

### To reference a RAM address

Click the [RAM] button on the Memory or Dump Window tool bar.



Click the RAM button.

## 8.3 Setting Data at a Specified Address

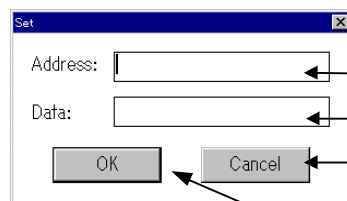
Data can be set at a specified address from the Memory or Dump Window. The type of memory to which data is set depends on the current display mode of the active window. Data is set to RAM memory when the display mode is RAM or ROM memory when the display mode is ROM.

### To change data at a specified address

With the Memory Window or Dump Window active, select the following from the **PD45** Window menu:

[Option]->[Debug]->[Set...]

When the Set dialog box opens, enter the address to be changed and the new data.



1. Specify the address.

2. Specify the new data.

3. Click the "OK" button.

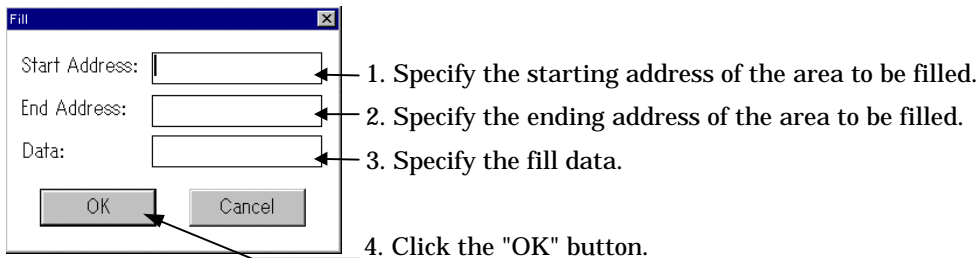
You can also double-click the data display area in the Memory Window or Dump Window to open the Set dialog box.

**To fill a specified area with data**

With the Memory Window or Dump Window active, select the following from the **PD45** Window menu:

[Option]->[Debug]->[Fill...]

When the Fill dialog box opens, enter the area to be filled and the fill data.

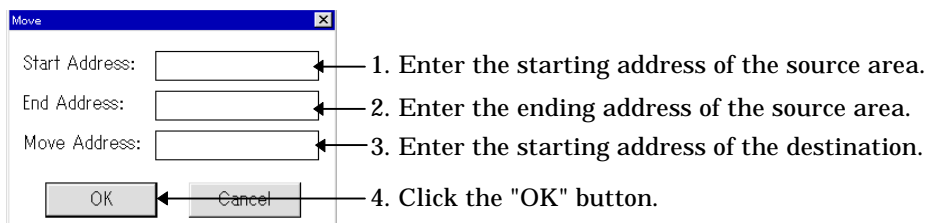


**To move the contents of a specified area to another area**

With the Memory Window or Dump Window active, select the following from the **PD45** Window menu:

[Option]->[Debug]->[Move...]

When the Move dialog box opens, enter the starting and ending addresses of the source area and the starting address of the destination area.



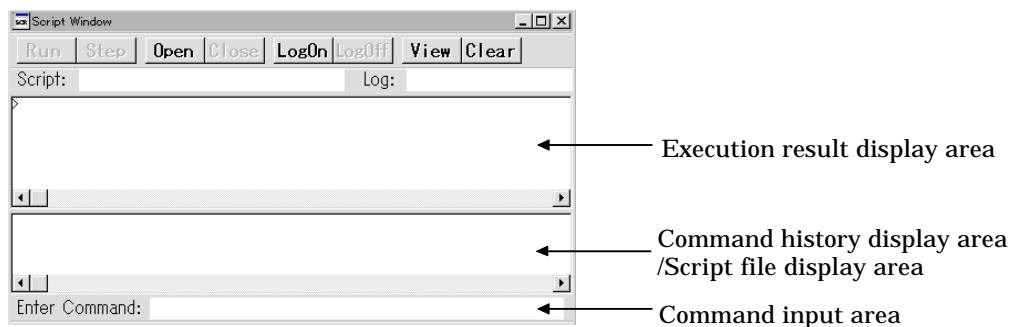
## 9. Script Commands

### 9.1 Executing Script Commands

Script commands are executed from the Script Window. To open the Script Window, select the following from the **PD45** Window menu:

[Window]->[Script Window]

Enter script commands in the Command input area in the Script window. Click the Command input area to locate the cursor in that field, then enter the script command. After executing the command, the results are output to the execution result display area.

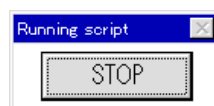


#### To re-execute a previously executed command

The history of command execution is displayed in the command history display area. Double-click the command you want to execute again.

#### To quit a script command that is executing

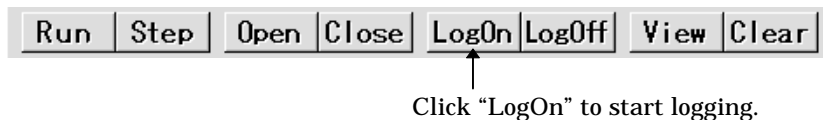
Click the STOP button, which is displayed only while the command is being executed.



## 9.2 Logging the Results of Executing Script Commands

Use the logging function to log the results of executing script commands. Before executing the script command, click the "LogOn" button on the toolbar of the Script Window.

Clicking the "LogOn" button opens the File Selection dialog box. Specify the name of the file to be saved. The default file attribute, if omitted, is ".log".



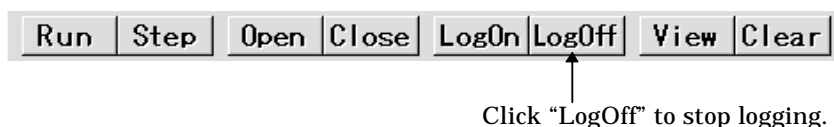
You can also select the following from the **PD45** Window menu (when the Script Window is active):

[Option]->[Log]->[On...]

- If, after starting **PD45**, you open and close a log file, then re-open it, the new data is appended to the data already in that file. However, if you re-open a log file that was created before starting **PD45**, the contents of that file will be overwritten.
- The file list in the File Selection dialog box starts with files with the ".log" attribute. However, you can enter a full filename (including the attribute) directly into the filename input area to open a log file with an attribute other than ".log".
- You can nest log files up to 8 levels.

### To quit logging the results of executing script commands

Click the "LogOff" button on the toolbar of the Script Window.



You can also select the following from the **PD45** Window menu (when the Script Window is active):

[Option]->[Log]->[Off]

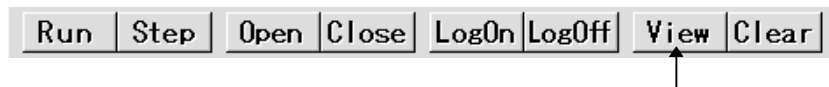
- When log files are nested, output to the current log file stops, but output to the next higher-level log file resumes.



**To save the results of executing script commands after execution**

**PD45** has a View Buffer that stores the results of executing the last 1000 lines of command results. To save the contents of the view buffer, click "View" on the toolbar of the Script Window.

Clicking "View" opens the File Selection dialog box. Specify the name of the file to be saved. The default file attribute, if omitted, is ".viw".



Click "View" to save the contents of the view buffer.

You can also select the following from the **PD45** Window menu (when the Script Window is active):

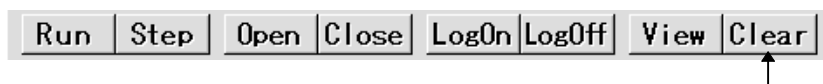
[Option]->[View]->[Save...]

- When you specify an existing filename, the contents of the view buffer are appended to the existing file.
- The file list in the File Selection dialog box starts with files with the ".viw" attribute. However, you can enter a full filename (including the attribute) directly into the filename input area to open a view file with an attribute other than ".viw".

**To clear the results of execution from the screen**

Click the "Clear" button on the menu in the Script Window to clear the contents of the command display.

Note that the contents of the View buffer are cleared at the same time as the command display.



Click the "Clear" button to clear the command display.

You can also select the following from the **PD45** Window menu (when the Script Window is active):

[Option]->[View]->[Clear]

### 9.3 Executing Script Commands in Batch Mode

You can execute script commands in batches. To do so, use an editor to write the commands to be executed in a script file. Script files take the ".scr" attribute.

Script files are opened from the Script Window. Click the "Open" button on the Script Window toolbar.

When the file selection dialog box opens, select the script file to be executed.



Click the "Open" button to open a script file.

You can also select the following from the **PD45** Window menu (when the Script Window is active):

[Option]->[Script]->[Open...]

- The file list in the File Selection dialog box starts with files with the ".scr" attribute. However, you can enter a full filename (including the attribute) directly into the filename input area to open a script file with an attribute other than ".scr".
- You can nest script files up to 5 levels.

When a script file is read in, the command history display in the Script Window changes into the script file display.

To execute the contents of the script file as a batch, click "Run" on the toolbar of the Script Window.

Batch execution starts and the script file is then closed after all the commands have been executed.



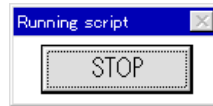
Click the "Run" button to execute all the commands in the script file in batch mode.

You can also select the following from the **PD45** Window menu (when the Script Window is active):

[Option]->[Script]->[Run]

### To stop execution of a script file

Click the STOP button displayed in the Running script dialog box.

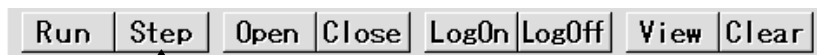


Execution of the script file stops before the next line.

### To independently execute each command in a script file

Click the "Step" button on the Script Window toolbar (for step execution of the script).

A command is executed each time you click the "Step" button.



Click the "Step" button for step execution of the script file.

You can also select the following from the **PD45** Window menu (when the Script Window is active):

[Option]->[Script]->[Step]

### To close a script file

Click the "Close" button on the Script Window toolbar.



Click the "Close" button to close the script file.

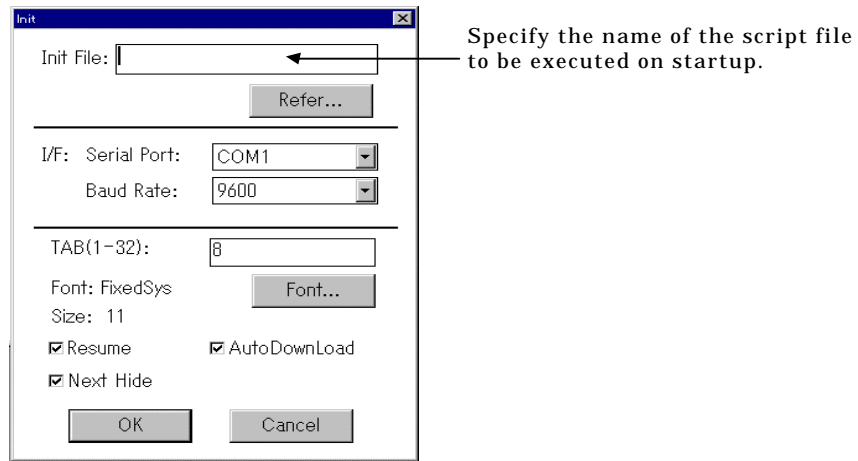
You can also close a script file by selecting the following (when the Script Window is active) from the **PD45** Window menu:

[Option]->[Script]->[Close]

- If the script files are nested, the current script file is closed and the next higher level script file is opened.

**To execute a script file on PD45 startup**

Specify the name of the script file to be executed on startup in the Init dialog box which is displayed when you start PD45.



You can also open the Init dialog box by selecting the following from the PD45 Window menu:

[Environ]->[Init]

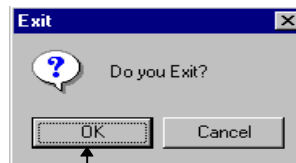
## 10. Exiting PD45

### 10.1 Exiting PD45

Select the following from the **PD45** Window menu:

[File]->[Exit]

A confirmation dialog box is displayed. Click OK to exit.



Click the "OK" button.

# 11. Miscellaneous

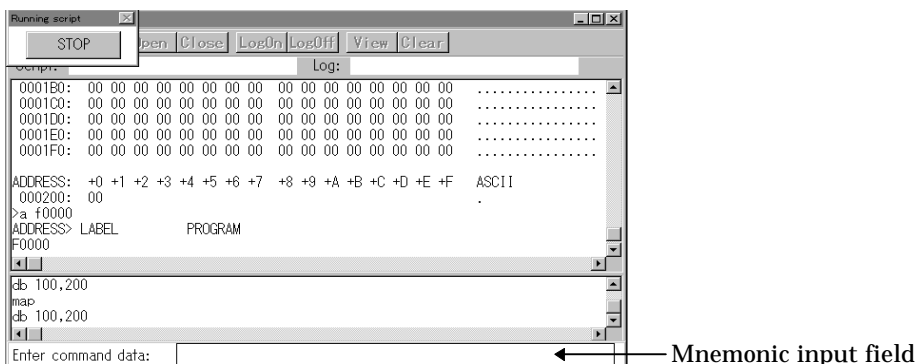
## 11.1 Line Assemble

Execute the assemble command (Assemble(A)) from the Script Window for line assembling. Specify the address for line assembling after the command.

>Assemble assemble-address

On execution of the Assemble command, the command input area of the Script Window changes to a mnemonic input field.

See Section 9.1, "Executing Script Commands" in the Basic Operation for details of how to use script commands. Enter the mnemonic for line assembling. The following shows the Script Window when waiting for a mnemonic to be input.



**To exit the line assemble mode**

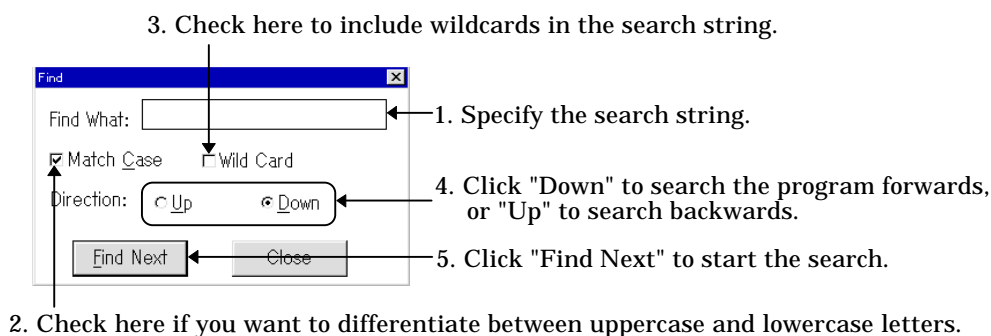
Specify "Enter" in the mnemonic input area. The Script Window changes back to the status in which it waits for a script command to be entered.

## 11.2 Searching for Character Strings in Target Program

You can search for character strings in the target program when the Program Window or Source Window is active. Make sure the target Program or Source Window is active, then select the following from the **PD45** Window menu:

[Edit]->[Find...]

The Find dialog box is displayed. Enter the search string, then click "Find Next" to start the search.



## 11.3 Changing Window Proportions

You can change the proportions of the Program Window, Source Window, Memory Window, and ASM Watch Window using the mouse as described below.

- Program Window and Source Window

When in disassemble display mode, you can change the relative proportions of the object code display area (<Objcode>) and the two parts (<Label> and <Mnemonic>) of the Program display area.

Drag to resize.

| BRK | Objcode | Label | Mnemonic |
|-----|---------|-------|----------|
| -   | 004     | start | DI       |
| -   | 000     | loop  | NOP      |

- Memory Window

You can change the relative proportions of the label display area (<LABEL>) and the memory content display area (<DATA>).

Drag to resize.

| Symbol | DATA |
|--------|------|
| zxy000 | 0    |
|        | D    |
|        | 0    |

- **ASM Watch Window**  
You can change the relative proportion of the expression display area (<Expr>).

Drag to resize.

| Addr:Bit | Expr    | Sz | Rdx | Data       |
|----------|---------|----|-----|------------|
| 000      | zxy000  | 4  | Hex | 0          |
| 010      | •zxy010 | 8  | Bin | •0011:1111 |

- **Trace Window**  
When in bus mode, you can change the relative proportion of the label display area (<LABEL>).

Drag to resize.

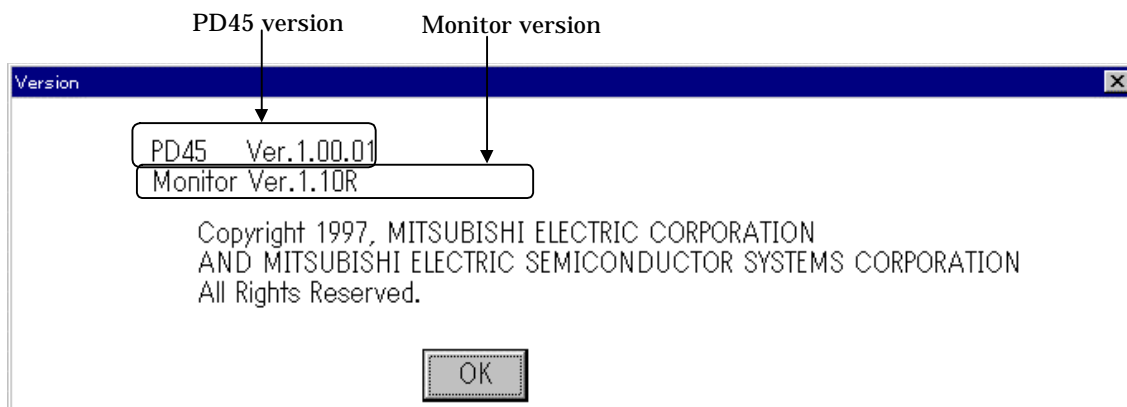
| Cycle | Label | Address | Data |
|-------|-------|---------|------|
| -0050 | sub1  | 0100    | 000  |
| -0049 |       | 0101    | 044  |
| -0048 |       | 0006    | 100  |

## 11.4 Displaying the Version of PD45

Select the following from the **PD45** Window menu:

[Help]->[About...]

The About dialog box, which shows the **PD45** version and monitor version, is displayed.



Press "OK" to close the About dialog box.



# High-end Debugging



## 1. Hardware Breaks(2)

This chapter describes the advanced hardware break functions available with PD45. For details about the structure of the H/W Break Point Setting dialog box, refer to Section 1.12.1, "Screen Structure of H/W Break Point Setting Dialog Box." For details on how to use the basic hardware break functions of PD45, refer to Section 4, "Hardware Breaks (1)" in the Basic Operation Part of this manual.

### 1.1 To Break Program On Stack Overflow

Use the H/W Breakpoint Setting dialog box to enter this setting.

| Address | Label      | Pass  |
|---------|------------|-------|
| A1      | 0000 start | 00001 |
| A2      | 0000 start | 00001 |

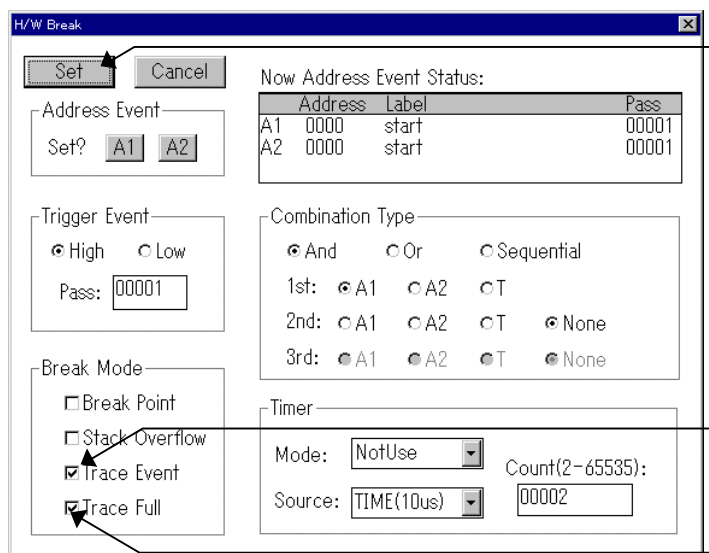
2. Click the "Set" button.

1. Check the "Stack Overflow" Check box.

\*Items in the Break Mode group can be specified simultaneously.

## 1.2 To Break Program On Real-time Trace Event

Two types of real-time trace events are available: establishment of a trace event (condition) and completion of a write to trace memory. Enter the following settings from the H/W Break Point Setting dialog box.



3. Click the "Set" button.

1. Check "Trace Event" if you want the program to break on establishment of a trace event.

2. Check "Trace Full" if you want the program to break on completion of a write to trace memory.

\*Items in the Break Mode group can be specified simultaneously.

## 1.3 To Break Program Using a Timer

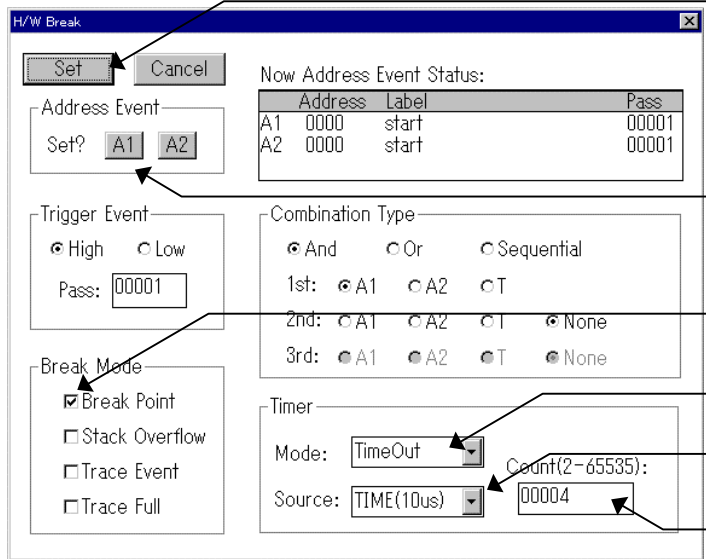
A timer can be used to cause the program to break with respect to an elapse of time. There are following three types of timer-based break conditions:

1. Time out (TimeOut)  
The program breaks when it does not reach a breakpoint within a specified time.
2. Time count (TimeCount)  
The program breaks when a specified time has elapsed after it started executing.
3. Delay count (DelayCount)  
The program breaks when a specified time has elapsed after it reached a breakpoint.

There are two types of count sources with which to measure a time: a timer mode where the emulator's internal timer (10 us, fixed) is used to count and a cycle mode where the MCU cycles are used to count. Choose one of these two modes depending on the specified content of a timer value. Note that a count value can be specified in the range of 2<sub>10</sub> to 65535<sub>10</sub>. The count source times a count value is the timer's specified time (cycles).

**To break program when loop routine period exceeded a specified time**

Use the timer's TimeOut function for this purpose. This method is effective for checking whether a routine that loops at a certain interval has exceeded a specified time for some reason (e.g., interrupt). Use the H/W Break Point Setting dialog box to make this setting. The following shows a setting example where the target program breaks when the routine takes more than 40 us to loop for some reason.



6. Click the "Set" button.

1. Specify the address break point in loop address.

2. Check the "Break Point" check box.

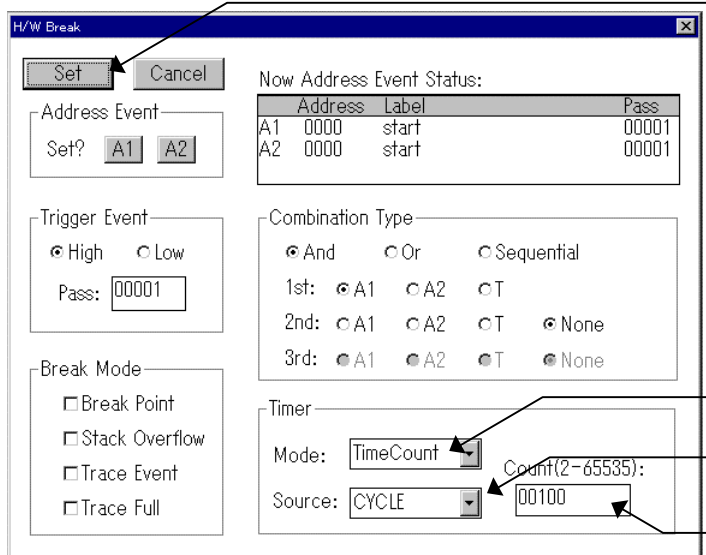
3. Select the mode "TimeOut".

4. Select the source "TIME(10us)".

5. Specify the timer count "4".

**To specify target execution time**

Use the timer's TimeCount function for this purpose. Use the H/W Break Point Setting dialog box to make this setting. The following shows a setting example where the target program is executed for 100 cycles after it started running.



4. Click the "Set" button.

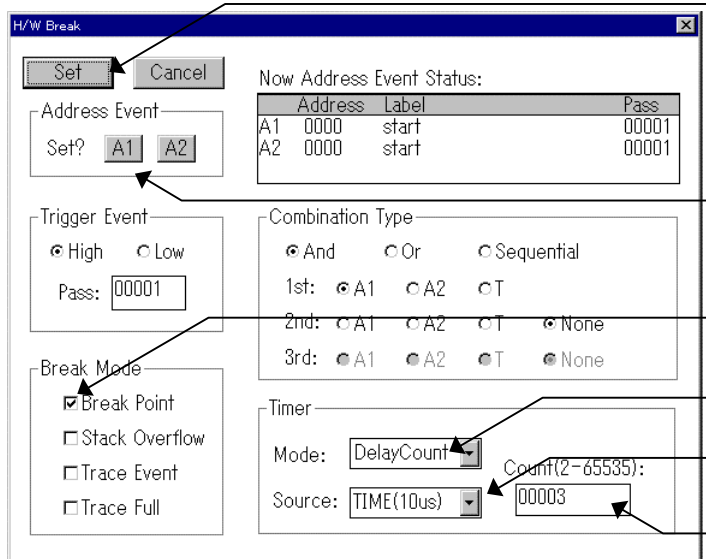
1. Select the mode "TimeCount".

2. Select the source "CYCLE".

3. Specify the timer count "100".

**To break program a specified time after breakpoint**

Use the timer's DelayCount function for this purpose. This method is effective for breaking program execution some time after a breakpoint. Use the H/W Break Point Setting dialog box to make this setting. The following shows a setting example where the target program breaks 30 us after a breakpoint.



6. Click the "Set" button.

1. Specify the address break point.

2. Check the "Break Point" check box.

3. Select the mode "DelayCount".

4. Select the source "TIME(10us)".

5. Specify the timer count "3".

## 2. Coverage measurement

Use the coverage measuring function to check addresses accessed during execution. The coverage measurement results can be referenced from the Coverage Window or using a script command, CoVerage (abbreviated CV).

### 2.1 Displaying Coverage Measurement Result in Percentage

Use the Coverage Window to show coverage measurement results in percentage. To open the Coverage Window, choose the following commands from the PD45 window menu:

[Window] -> [Coverage Window]

Before displaying coverage measurement results in percentage, you must first set a range of addresses to be measured. For details about the Coverage Window, refer to Section 1.10, "Coverage Window" in the Window Function Part of this manual.

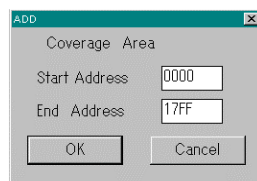
#### To set a measurement address range

Click the "Addarea" button on the Coverage Window tool bar.

1. Click the "Addarea" button.



When you click the "Addarea" button, an Addarea dialog box opens. Input a range of addresses to be measured in this dialog box.



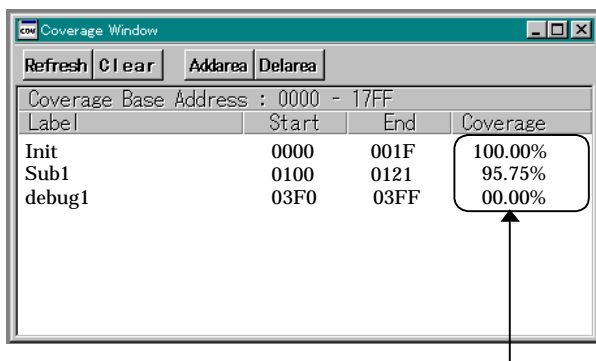
1. Specify the measurement start address.

2. Specify the measurement end address.

3. Click the "OK" button.

**To reference measurement results**

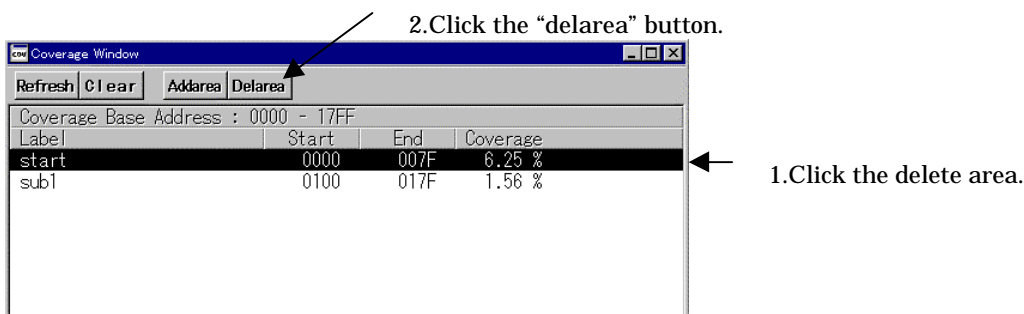
When you chose to show measurement results in percentage, they are displayed in the Coverage Window's measurement result display area.



Measurement results in percentage

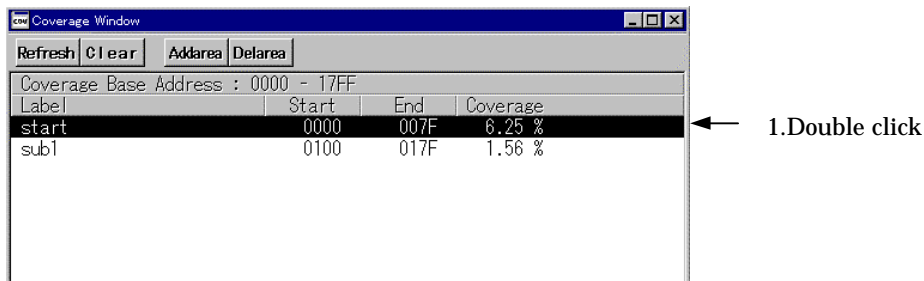
**Deleting a Measurement Address Range in Coverage Measurement Area**

Click on a line of a specified measurement address range in the Coverage Window that you want to be deleted. Then click the "Delarea" button.



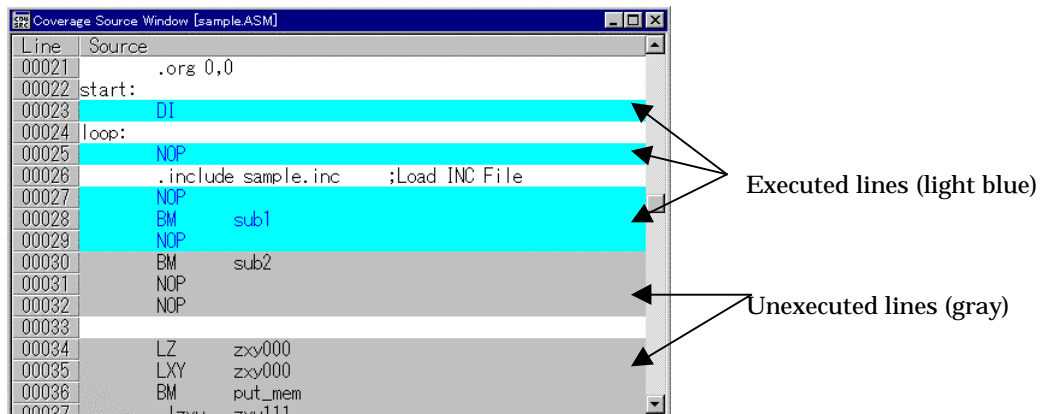
**2.2 Referencing Coverage Measurement Results at Source Level**

Double-click on a line of measurement address range specified in the Coverage Window. The Coverage Source Window will appear.



The lines that have been executed by the target program are displayed in light blue and those not executed are displayed in gray.





## 2.3 Referencing Coverage Measurement Results at Address Level

Use a script command, CoVerge (abbreviated CV), for this purpose. For details on how to use script commands, refer to Section 9.1, "Executing Script Commands" in the Basic Operation Part of this manual. The following shows reference examples where measurement results are referenced for accessed addresses in the address range of 0000<sub>16</sub> to 03FF<sub>16</sub>.

- To reference coverage measurement results one byte at a time in the address range of 0000<sub>16</sub> to 03FF<sub>16</sub>  
**>CoVerge LOCAL, 0000,03FF**
- To reference coverage measurement results four bytes at a time in the address range of 0000<sub>16</sub> to 03FF<sub>16</sub>  
**>CoVerge GLOBAL, 0000, 03FF**

## 2.4 To initialize the previous results of coverage measurements

Click the "Clear" button on the Coverage Window tool bar.

1. Click the "Clear" button.



Coverage measurement results can also be initialized using a CoVerge command. Enter the following to initialize the results of coverage measurements:

**>CoVerge CLEAR**

[ MEMO ]

# Reference



# 1. Table of Script Commands

## 1.1 Input Format

- The format for entering PD45 script commands is as follows:
  1. Insert one or more spaces or tabs between the command and its parameter(s).
  2. You can use uppercase and lowercase letters and numerics for the command.
- Description of symbols used in command format

| Parameter          | Command format  |
|--------------------|---|
| XXXX               | XXXX must be input  |
| [XXXX]             | XXXX is optional  |
| { X1   X2   X3 }   | One of X1, X2, or X3 must be input                          |
| [{ X1   X2   X3 }] | Input can be omitted, or you can input one of X1, X2, or X3 |

## 1.2 Tables of Commands

In the following tables of commands, commands shown with half-tone screening can be executed at run time.

The abbreviated forms of commands are shown in parenthesis after the full command.

See the PD45 on-line help for details of each command. To display on-line help, enter the following from the PD45 Window menu:

[Help] -> [Index]

### 1.2.1 Execution Commands

| Command             | Command format                                    | Summary  |
|---------------------|---|--|
| Go(G)               | Go [StartAddress][,BreakAddress<br>[:Pass Count]] | Execute the target program.                    |
| GoBreak(GB)         | GoBreak [StartAddress]                            | Execute the target program at break table .    |
| GoMode(GM)          | GoMode [{STOP   DI}]                              | Change execute control mode.                   |
| <i>STOP</i>         | STOP  | Stop the target program.                       |
| <i>STATUS</i>       | STATUS  | Display status of target program execution.    |
| Step(T)             | Step [number of steps]                            | Step execution at source level.                |
| StepInstruction     | StepInstruction [number of steps]                 | Step execution at machine language level.      |
| OverStep(TM)        | OverStep [number of steps]                        | Over-step execution at source level.           |
| OverStepInstruction | OverStepInstruction<br>[number of steps]          | Over-step execution at machine language level. |
| Return              | Return  | Return execution at source level.              |
| ReturnInstruction   | ReturnInstruction                                 | Return execution at machine language level.    |
| Reset(Z)            | RESET   | Reset the target program.                      |

### 1.2.2 File Operation Commands

| Command   | Command format                              | Summary   |
|-----------|---|---|
| Load(I)   | Load filename[.hex   .sym]                  | Download HEX/SYM file.                                |
| Upload(O) | Upload StartAddress,EndAddress<br>,Filename | Output contents of specified memory area to hex file. |

### 1.2.3 Register Operation Commands

| Command     | Command format   | Summary  |
|-------------|--|--|
| Register(X) | 1.Register [RegisterName]<br>2.Register RegisterName,Value | 1.Check value of specified register.<br>2.Set value in specified register. |



### 1.2.8 Real-Time Trace Commands

| Command        | Command format   | Summary   |
|----------------|--|---|
| TracePoint(RP) | 1.TracePoint<br>2.TracePoint An, address [:PassCount]<br>3.TracePointArea An, address1,, address2[:PassCount]<br>4.TracePoint T, { H   L }[:PassCount]<br>5.TracePoint {AND   OR   SEQ},{An   T},{An   T},{An   T}}<br>6.TracePoint MODE,{BEFORE   ABOUT   AFTER   BREAK}<br>[, {NO   TO   TC   DC} [, {TIME   CYCLE} [, COUNT]]}] | 1. Reference trace point.<br>2.to4. Set trace point.<br>5. Set combination.<br>6. Set trace condition.<br><br>See page 115 for details. |
| TraceData(RD)  | TraceData [StartCycle] [,EndCycle]   | Display contents of real-time trace memory in bus signal format.  |
| TraceList(RL)  | TraceList [StartCycle] [,end cycle]  | Display the contents of real-time trace memory in disassemble list format.  |
| TraceRange(RR) | TraceRange   | Display the contents of real-time trace range.  |

### 1.2.9 Coverage Measurement Commands

| Command      | Command format   | Summary  |
|--------------|--|--|
| CoVerage(CV) | 1.Coverage LOCAL [,StartAddress, EndAddress]<br>2.Coverage GLOBAL [,StartAddress, EndAddress]<br>3.Coverage TOTAL [,StartAddress, EndAddress]<br>4. Coverage CLEAR | 1.Display results of coverage measurements in 1-byte units.<br>2.Display results of coverage measurements in 4-byte units.<br>3.Display results of coverage measurements as percentages.<br>4.Initialize memory for coverage measurements. |

### 1.2.10 Run-time Display Commands

| Command          | Command format   | Summary   |
|------------------|--|---|
| <i>TIME</i>      | 1. TIME<br>2. TIME { ON   OFF }  | 1. Display settings for run-time display.<br>2. Set whether to display run time in Script Window.   |
| MeasurementPoint | 1.MeasurementPoint<br>2.MeasurementPoint {START   END },address<br>3.MeasurementPoint MODE, ModeNumber<br>4.MeasurementPoint SOURCE, { TIME   CYCLE }<br>5.MeasurementPoint SHOW | 1.Display the measurement point setting.<br>2.Set measurement point address.<br>3.Set measurement range.<br>4.Set measurement count source.<br>5.Display the measurement result.<br>See page 117 for details. |

### 1.2.11 Script/Log-File Commands

| Command       | Command format                         | Summary  |
|---------------|--|--|
| <i>SCRIPT</i> | SCRIPT filename                        | Open script file.  |
| <i>EXIT</i>   | EXIT                                   | Close script file.   |
| <i>WAIT</i>   | WAIT                                   | Wait for command input until target program stops.                               |
| <i>PAUSE</i>  | PAUSE "MESSAGE"                        | Display specified message in Pause dialog box and wait for user to press button. |
| <i>SLEEP</i>  | SLEEP number of seconds                | Wait for command input for specified number of seconds.                          |
| <i>LOGON</i>  | LOGON [ Filename [ .file attribute ] ] | Open log file.   |
| <i>LOGOFF</i> | LOGGOFF                                | Close log file.  |



### 1.2.12 Program Window Control Commands

| Command     | Command format                              | Summary  |
|-------------|---|--|
| <i>PATH</i> | PATH [ search pass [ ;search pass ;.... ] ] | Set source file path.                                      |
| <i>FILE</i> | 1.FILE<br>2.FILE source filename            | 1. List source files.<br>2. Display specified source file. |

### 1.2.13 Utility Commands

| Command             | Command format                               | Summary  |
|---------------------|--|--|
| <i>RADIX</i>        | 1.RADIX<br>2. RADIX { 2   8   10   16 }      | 1. Check radix for input of constants.<br>2. Set radix for input of constants. |
| <i>ALIAS</i>        | 1. ALIAS<br>2. ALIAS new name , command name | 1. Check definition of command alias.<br>2. Define command alias.              |
| <i>UNALIAS</i>      | UNALIAS new name                             | Delete specified alias.  |
| <i>UNALIASALL</i>   | UNALIASALL                                   | Delete all aliases.  |
| <i>VERsion(VER)</i> | VERsion                                      | Display PD45 and monitor versions.   |
| <i>DATE</i>         | DATE   | Display current date and time (yymmdd, day of week, and time).                 |
| <i>ECHO</i>         | ECHO "MESSAGE"                               | Display specified parameter.   |
| <i>QUIT</i>         | QUIT   | Quit PD45.   |
| <i>CD</i>           | 1. CD<br>2. CD directory_name                | 1. Checks the current directory.<br>2. Sets the current directory.             |

## 1.2.14 Supplementation explanation of Script Commands

### HardwareBreak(HB)

Setting and Checking Hardware Breaks

---

#### Input format

- (Format 1) HardwareBreak
- (Format 2) HardwareBreak An, address [:PassCount]
- (Format 3) HardwareBreakArea An, StartAddress, EndAddress [:PassCount]
- (Format 4) HardwareBreak T, { H | L }[:PassCount]
- (Format 5) HardwareBreak {AND|OR|SEQ}, { An | T } [, { An | T }  
[, { An | T }]]
- (Format 6) HardwareBreak MODE, { BP | SO | TE | TF [, { NO | TO | TC | DC }  
[, { TIME | CYCLE } [, COUNT]]}]

The following table shows the values that can be specified for An.

|    |        |
|----|--------|
| An | A1, A2 |
|----|--------|

#### Function

- Hardware breaks allow you to stop target program execution detection of an instruction fetch.
- You can set up to 2 address and one external trigger break.
- To use hardware breaks, use the BreakMode command as shown below to enable the break (Hardware can be abbreviated to H).

#### BreakMode Hardware

#### Setting Break Points

##### Checking hardware break settings

Use format 1. Enter the following to display the contents of the hardware break setting.

#### HardwareBreak

##### Stopping program execution on execution of the instruction at the specified address

Use format 2.

- Enter the following to stop execution of the target program on execution of the instruction at address 100h.

#### HardwareBreak A1, 100

- Enter the following to stop execution of the target program on 5th execution of the instruction at address 312h.

#### HardwareBreak A1, 312:5

##### To stop the program when an instruction in a specified address range is executed

Use formats 3.

- If you want program execution to be stopped when any instruction at addresses from 100h to 17Fh is executed, enter the command as follows:

#### HardwareBreakArea A1, 100, 17F

To stop the program when an instruction outside a specified address range is executed

Use format 3.

- If you want the program to be stopped when any instruction at addresses outside the range of 100h to 17Fh is executed, enter the command as follows:

**HardwareBreakArea A1, 17F, 100**

To stop program execution when the external input signal changes to HIGH or LOW

Use format 4.

- Enter the following to stop program execution when the trigger signal changes to HIGH.

**HardwareBreak T, H**

### Using Multiple Hardware Break Points

The PC4504 allows you to set up to two address break points and one trigger break point. You can also combine these break point conditions to create more complex break points.

To break program execution when all break point conditions are satisfied

Use format 5.

- Enter the following to stop program execution when all conditions for break points A1, A2, and T are satisfied.

**HardwareBreak AND A1,A2,T**

You can specify the break points in any order (For example, A2, A1, T has the same effect as A1, T, A2).

To break program execution when any one of the break point conditions is satisfied

Use format 5.

- Enter the following to stop program execution when any one of break points A1, A2, or T is satisfied.

**HardwareBreak OR, A1,A2,T**

You can specify the break points in any order.

To break program execution when the break point conditions are satisfied in the specified order

Use format 5.

- Enter the following to stop program execution when the conditions for break point T are satisfied followed by those for break point A1.

**HardwareBreak SEQ T,A1**

When you use this format, specify the break point conditions in the order in which they must be satisfied for program execution to be stopped. If you specify **A1, T**, for example, program execution stops if the conditions for A1 are satisfied before those for T.

## Using a Special Event Break

In addition to address and trigger breaks, the PC4504 emulator allows you to use a break for failure (e.g., stack overflow), a trace event break (establishment of a trace event or completion of a write to trace memory), and a timer-based timer break.

To break the program when a stack overflow occurs

Use format 6.

- To stop the program when a stack overflow occurs, enter the command as follows:

**HardwareBreak MODE,SO**

To break the program when a trace event occurs

Use format 6.

- To stop the program when a trace point-passed event is established, enter the command as follows:

**HardwareBreak MODE,TE**

- To stop the program when a write to trace memory is completed, enter the command as follows:

**HardwareBreak MODE,TF**

To break the program when a timer event occurs

Use format 6.

- To stop the program when a routine that loops at a certain interval (35 us) takes more than 40 us to loop for some reason, enter the command as follows:

**HardwareBreak MODE,BP,TO,TIME,4**

- To stop the program 100 cycles after it started running, enter the command as follows:

**HardwareBreak MODE,BP,TC,CYCLE,100**

- To stop the program 50 us after it passed a breakpoint, enter the command as follows:

**HardwareBreak MODE,BP,DC,TIME,5**

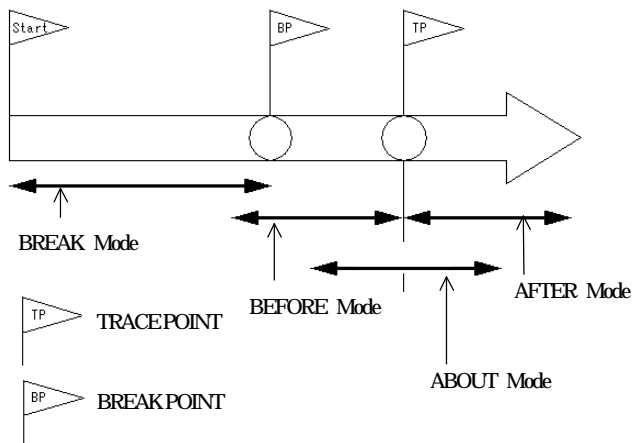


### Setting the area for real-time tracing

Use format 6.

- Enter the following to check the results of the trace up to the set trace point.  
**TracePoint AREA, BEFORE**  
 You can check the 4K cycles before the trace point.
- Enter the following to check the results of the trace either side of the trace point.  
**TracePoint AREA, ABOUT**  
 You can check the 2K cycles before plus the 2K cycles after the trace point.
- Enter the following to check the results of the trace after the trace point.  
**TracePoint AREA, AFTER**  
 You can check the 4K cycles after the trace point.
- Enter the following to check the results up to when the target program stops.  
**TracePoint AREA, BREAK**  
 You can check the 4K cycles up to when the target program stops. This is the default setting when you start PD45.

Figure 1 shows the relationship between the real-time trace area and the trace point.



**Figure 1 Real-time Trace Area**

---

## MeasurementPoint (MP)

### Setting and Checking Measurement Points

---

#### Input format

- (Format 1) MeasurementPoint
- (Format 2) MeasurementPoint { START | END }, Address
- (Format 3) MeasurementPoint MODE, ModeNumber
- (Format 4) MeasurementPoint SOURCE, { TIME | CYCLE }
- (Format 5) MeasurementPoint SHOW

#### Function

- The PC4504 emulator has the following eight modes of time measurements available.

| No. | Start Event                | End Event                |
|-----|----------------------------|--------------------------|
| 1.  | Start of program execution | End of program execution |
| 2.  | Start of program execution | Measurement point End    |
| 3.  | Start of program execution | Trace Event              |
| 4.  | Trace Event                | Measurement point End    |
| 5.  | Trace Event                | End of program execution |
| 6.  | Measurement point Start    | Measurement point End    |
| 7.  | Measurement point Start    | Trace Event              |
| 8.  | Measurement point Start    | End of program execution |

#### Checking measurement condition

Use format 1.

- To reference the measurement conditions of the time measurement function, enter the command as follows:

**MeasurementPoint**

#### Setting measurement points

Use format 2.

- To set the start point of time measurement at address 0100<sub>16</sub>, enter the command as shown below. Note however that the start point of time measurement is valid only when the specified mode of time measurement is 1, 2, or 3.

**MeasurementPoint START, 0**

- To set the end point of time measurement at address 017F<sub>16</sub>, enter the command as shown below. Note however that the end point of time measurement is valid only when the specified mode of time measurement is 2, 4, or 6.

**MeasurementPoint END, 17F**

### **Setting measurement mode**

Use format 3.

The default measurement mode is "1. From start of program execution till end of program execution."

- To set the time measurement mode for "6. From measurement start point till measurement end point", enter the command as follows:

**MeasurementPoint MODE,6**

### **Setting measurement count source**

Use format 4.

The default measurement count source is "TIME" in which the PC4504's internal timer (16 MHz fixed) is used to measure a time

- To set the time measurement count source for "CYCLE" in which the target MCU cycles are used to measure a time, enter the command as follows:

**MeasurementPoint SOURCE,CYCLE**

### **Checking measurement results**

Use format 5.

- To check the time measurement result, enter the command as follows:

**MeasurementPoint SHOW**



## 2. Writing Script Files

PD45 allows you to run script files in a Script Window. The script file contains the controls necessary for automatically executing the script commands.

### 2.1 Structural Elements of a Script File

You can include the following in script files:

- Script commands
- Assign statements
- Conditional statements (if, else, endi)  
Program execution branches to the statement(s) to be executed according to the result of the conditional expression.
- Loop statements (while, endw)  
A block of one or more statements is repeatedly executed according to the expression.
- break statement  
Exits from the innermost loop.
- Comment statements  
You can include comments in a script file. The comment statements are ignored when the script commands are executed.

Specify only one statement on each line of the script file. You cannot specify more than one statement on a line, or write statements that span two or more lines.

#### 2.1.1 Script Commands

You can use the same script commands that you enter in the Script Window. You can also call script files from within other script files (nesting up to 5 levels).

### 2.1.2 Assign Statements

Assign statements define and initialize macro variables and assign values. The following shows the format to be used.

```
% macro-variable = expression
```

- You can use alphanumerics and the underscore ( `_` ) in macro variable names. However, you cannot use a numeric to start a macro variable name.
- You can specify any expression of which the value is an integer between  $0_{16}$  and  $FFFFFFF_{16}$  to be assigned in a macro variable. If you specify a negative number, it is processed as twos complement.
- You can use macro variables within the expression.
- Always precede macro variables with the "%" sign.

### 2.1.3 Conditional Statements (if, endi, else)

In a conditional statement, different statements are executed according to whether the condition is true or false. The following shows the format to be used.

```
if ( expression )  
    statement 1  
else  
    statement 2  
endi
```

- If the expression is true (other than 0), statement 1 is executed. If false, (0), statement 2 is executed.
- You can omit the else statement. If omitted and the expression is false, execution jumps to the line after the endi statement.
- if statements can be nested (up to 32 levels).

### 2.1.4 Loop Statements (while, endw) and Break Statement

In loop statements, execution of a group of statements is repeated while the expression is true. The following shows the format to be used.

```
while ( expression )  
    statement  
endw
```

- If the expression is true, the group of statements is repeated. If false, the loop is exited (and the statement following the endw statement is executed).
- You can nest while statements up to 32 levels.
- Use the break statement to forcibly exit a while loop. If while statements are nested, break exits from the innermost loop.

### 2.1.5 Comment statements

You can include comments in a script file. Use the following format.

|                           |
|---------------------------|
| <b>; character string</b> |
|---------------------------|

- Write the statement after a semicolon (;). You can include only spaces and tabs in front of the semicolon.
- Lines with comment statements are ignored when the script file is executed.

Notes:

- You cannot include comments on the same lines as script commands.
- You can nest script files up to five levels.
- You can nest if statements and while statements up to 32 levels.
- If statements must be paired with endi statements, and while statements with endw statements in each script file.
- Expressions included in script files are evaluated as unsigned types. Therefore, operation cannot be guaranteed if you use negative values for comparison in if or while statements.
- You can specify up to 4096 characters per line. An error occurs if a line exceeds this number of characters.
- When a script file containing illegal commands is automatically executed (when you select [Option] -> [Script] -> [Run] from the Script Window menu after opening a script file, or click the <Run> button in the Script Window), execution of the script file continues even after the error is detected, except when the script line itself cannot be read.  
If an error is detected and the script file continues to be executed, operation after detection of the error cannot be guaranteed. Reliability cannot therefore be placed on the results of execution after an error has been detected.

## 2.2 Writing Expressions

PD45 allows you to use expressions for specifying addresses, data, and number of passes, etc. The following shows example commands using expressions.

```
>Dump TABLE1
>Dump TABLE1+20
```

### 2.2.1 Elements of Expressions

You can use the following elements in expressions:

- Constants
- Symbols and labels
- Macro variables
- Register variables
- Memory variables
- Line Nos.
- Character constants
- Operators

The following describes the respective elements.

### 2.2.2 Constants

You can use binary, octal, decimal, or hexadecimals. The prefix or suffix symbol attached to the numerical value indicates which radix is used.

|          | Hexadecimal     | Decimal | Octal | Binary* <sup>1</sup> |
|----------|-----------------|---------|-------|----------------------|
| Prefix   | 0x,0X           | @       | None  | %                    |
| Suffix   | h,H             | None    | o,O   | b,B                  |
| Examples | 0xAB24<br>AB24h | @1234   | 1234o | %10010<br>10010b     |

\*<sup>1</sup> You can only specify % when the predetermined radix is hexadecimal.

- If you are inputting a radix that matches the predetermined radix, you can omit the symbol that indicates the radix (excluding binary).
- Use the RADIX command to set the predetermined value of a radix. However, in the cases shown below, the radix is fixed regardless of what you specify in a RADIX command.

| Type              | Radix |
|-------------------|-------|
| Address           | Hex   |
| Line No.          | Dec   |
| No. of executions |       |
| No. of passes     |       |
| No. of limits     |       |

### 2.2.3 Symbols and labels

You can include symbols and labels defined in your target program, or symbols and labels defined using the Assemble command.

- You can include alphanumerics, the underscore (`_`), period (`.`), and question mark (`?`) in symbols and labels. However, do not start with a numeric.
- Symbols and labels can consist of up to 255 characters.
- Uppercase and lowercase letters are unique.
- You cannot include the assembler ASM45 pseudo instructions, macro instructions, operation code, or reserved words (`.ORG`, `.IF`, etc.).
- You cannot use strings that start with two periods (`..`) for symbols or labels.

#### **Note 1: Priority levels of labels and symbols**

The conversion of values to labels or symbols, and vice versa, is subject to the following levels of priority:

- Conversion of ROM address values
  1. Labels
  2. Symbols
- Conversion of RAM address values
  1. ZXY symbols
  2. Bit symbols
- Conversion of data values
  1. Symbols
  2. Labels

### 2.2.4 Macro Variables

Macro variables are defined by assign statements in the script file. See Section 2.1, "Structural Elements of a Script File" in the Reference part for details.

Precede variables with `'%'` for use as macro variables.

- You can specify alphanumerics and/or the underbar (`_`) in the variable name following the percent sign (`%`). However, do not start the names with a numeric.
- You cannot use the names of registers as variable names.
- Uppercase and lowercase letters are differentiated in variable names.
- You can define a maximum of 32 macro variables. Once defined, a macro variable remains valid until you quit PD45.

Macro variables are useful for specifying the number of iterations of the while statement.

### 2.2.5 Register variables

Register variables are used for using the values of registers in an expression. Precede the name of the register with '%' to use it as a register variable. Use the following format.

**%register-name**

You can use the following registers.

**PC, A, B, Z, , Y, E, D, SP, CY**

Uppercase and lowercase letters are not unique in register names. You can specify either.

### 2.2.6 Memory variables

Use RAM memory variables to use memory values in expressions. The format is as follows:

**[RAM address].data-size**

- You can specify expressions in RAM addresses (you can also specify memory variables).

Example 1: Referencing the contents of RAM memory at address 100<sub>16</sub>  
[100h]

### 2.2.7 Line Nos.

These are source file line Nos. The format for line Nos. is as follows:

**#line\_no**  
**#line\_no."source file name"**

- Specify line Nos. in decimal.
- You can only specify line Nos. in which software breaks can be set.  
You cannot specify lines in which no assembler instructions have been generated, including comment lines and blank lines.
- If you omit the name of the source file, the line Nos. apply to the source file displayed in the Program Window.
- Include the file attribute in the name of the source file.
- Do not include any spaces between the line No. and name of the source file.

### 2.2.8 Character constants

The specified character or character string is converted into ASCII code and processed as a constant.

- Enclose characters in single quote marks.
- Enclose character strings in double quote marks.
- The character string must consist of one or two characters (16 bits max.).  
If more than two characters are specified, the last two characters of the string are processed. For example, "ABCD" would be processed as "CD", or value 4344<sub>16</sub>.

### 2.2.9 Operators

The table below lists the operators that you can use in expressions.

- The priority of operators is indicated by the level, level 1 being the highest and level 8 the lowest.

If two or more operators have the same level of priority, they are evaluated in order from the left of the expression.

| Operator             | Function  | Priority level |
|----------------------|---|----------------|
| ()                   | Brackets  | Level 1        |
| +, =, ~              | Monadic positive, monadic negative, monadic logical NOT | Level 2        |
| *, /                 | Dyadic multiply, dyadic divide                          | Level 3        |
| +, -                 | Dyadic add, dyadic subtract                             | Level 4        |
| >>, <<               | Right shift, left shift                                 | Level 5        |
| &                    | Dyadic logical AND                                      | Level 6        |
| , ^                  | Dyadic logical OR, dyadic exclusive OR                  | Level 7        |
| <, <=, >, >=, ==, != | Dyadic comparison                                       | Level 8        |

## 3. Error Messages

The following tables list the PD45 error messages.

| No. | Error Message   | Notes and Action  |
|-----|---|---|
| 150 | Can't open more (name) window.                                | The maximum number of the specified window is already open.   |
| 151 | Can't Create (name) window.                                   | Cannot open the specified window. There may not be sufficient memory. Quit other applications or increase memory. |
| 152 | Can't open (name) window, when the target program is running. | Stop the target program, then open the window.  |
| 153 | Value is out of range.  | The specified address exceeds the MCU's maximum address.  |
| 154 | PD45 already running.   |   |

| No. | Error Message   | Notes and Action  |
|-----|---|---|
| 200 | Can't change view mode.                               | The display starting address does not match the first line of the source file, or the specified source file cannot be found.                          |
| 201 | Can't find source file (filename).                    | Specified source file was not found. Use the PATH command, or the [Environ] -> [Path] menu items to specify the directory containing the source file. |
| 202 | Can't find search string (name).                      | The specified search string was not found between the starting position and end.  |
| 203 | Line number of Source File (filename) is over (line). | Because the source file has more lines than can be displayed, the file cannot be displayed in the Source Window. Switch to disassemble display mode.  |



| No. | Error Message                           | Notes and Action   |
|-----|---|--|
| 300 | Illegal endi(filename line).            | There is no if statement paired with the endi statement.               |
| 301 | Illegal endw(filename line).            | There is no while statement paired with the endw statement.            |
| 303 | Script File is already exist.           |  |
| 304 | Can't find endi(filename line).         | There is no endi statement paired with the if statement.               |
| 305 | Line length is overflow(filename line). | The number of characters exceeds the maximum permissible for one line. |
| 306 | Nest level is overflow(filename line).  |  |
| 307 | Can't find Script File(filename).       |  |
| 308 | Can't read Script File(filename).       | Cannot read rest of script file.                                       |
| 309 | Description is illegal(filename line).  |  |
| 310 | Can't find endw(filename line).         | There is no endw statement paired with while statement.                |
| 311 | Nest level is overflow(filename line).  |  |

| No. | Error Message                               | Notes and Action |
|-----|---|------------------|
| 400 | Address value is out range for scroll area. |                  |

| No. | Error Message   | Notes and Action |
|-----|---|------------------|
| 600 | Can't add new watch point because it exceeds limit of watch point number. Max number is(num). |                  |
| 601 | Address value is out of range.  |                  |
| 602 | Data value is out of range.   |                  |
| 603 | Bit value is out of range.  |                  |

| No. | Error Message                    | Notes and Action             |
|-----|----------------------------------|------------------------------|
| 650 | There are no symbol information. | Load module file not loaded. |
| 651 | The expression is too long.      |                              |

| No. | Error Message                     | Notes and Action   |
|-----|-----------------------------------|--|
| 900 | SYMBOL file is illegal.           | Error in format of load module file.   |
| 901 | Loading is canceled.              |  |
| 902 | Can't find SYMBOL file(filename). | No load module file exists.  |
| 903 | Can't get enough memory.          | Insufficient memory. Quit other applications or increase memory.               |
| 904 | Can't open temporary file.        | The temporary file for downloading using the on-demand method could be opened. |

| No.  | Error Message                         | Notes and Action                 |
|------|---------------------------------------|----------------------------------|
| 1001 | Can't find symbol.                    | Specified symbol does not exist. |
| 1002 | Description of expression is illegal. |                                  |
| 1004 | Description is illegal.               | Error in expression.             |
| 1006 | Can't find symbol.                    |                                  |

| No.  | Error Message                                | Notes and Action                                 |
|------|--|--|
| 1100 | Address value is out of range.               | Specified address exceeds MCU's maximum address. |
| 1101 | Description of Assembly language is illegal. |  |
| 1102 | Address value for JUMP is out of range.      |  |
| 1103 | Operand value is out of range.               |  |
| 1104 | Description of expression is illegal.        |  |
| 1107 | Operand value is undefined.                  |  |
| 1108 | Bit-symbol is in expression.                 |  |
| 1109 | Invalid bit-symbol exist.                    |  |
| 1113 | Characters exist in expression.              |  |
| 1115 | Invalid symbol definition.                   |  |
| 1116 | Invalid reserved word exist in operand.      |  |
| 1118 | Reserved word is missing.                    |  |
| 1119 | No space after mnemonic or directive.        |  |
| 1123 | Operand value is not defined.                |  |
| 1124 | Operand size is not appropriate.             |  |
| 1125 | Operand type is not appropriate.             |  |
| 1131 | Size or format specified is not appropriate. |  |
| 1132 | Size specified is missing.                   |  |
| 1133 | String value exist in expression.            |  |
| 1134 | Symbol is missing.                           |  |
| 1135 | Symbol is multiple defined.                  |  |
| 1136 | Symbol is missing.                           |  |
| 1137 | Symbol is multiple defined.                  |  |
| 1138 | Invalid operand exist in instruction.        |  |
| 1139 | Syntax error in expression.                  |  |

| No.  | Error Message                         | Notes and Action |
|------|---------------------------------------|------------------|
| 1153 | Invalid operand exist in instruction. |                  |
| 1154 | Quote is missing.                     |                  |
| 1155 | Right quote is missing.               |                  |
| 1156 | Can't get enough memory.              |                  |

| No.  | Error Message                                     | Notes and Action  |
|------|---|---|
| 1200 | Syntax error.                                     |   |
| 1201 | Command name is wrong.                            |   |
| 1202 | Too many aliases.                                 | Maximum: 256  |
| 1203 | You can register the only command name for alias. |   |
| 1204 | Can't use the command now.                        | You cannot use the specified command while the target program is running. |
| 1208 | Already set hard ware break.                      |   |
| 1209 | This command is not supported now.                |   |
| 1210 | The start address larger than the end address.    |   |
| 1211 | Can't register that token for alias.              |   |
| 1212 | The save file name (string1) is wrong.            | Specified file cannot exist.  |

| No.  | Error Message   | Notes and Action |
|------|---|------------------|
| 1300 | Line number is illegal.   |                  |
| 1302 | The Number of Macro constant is over the limit (limit).             |                  |
| 1303 | Immediate value is out of range.                                    |                  |
| 1304 | Prefix which gives radix of the constant is illegal.                |                  |
| 1305 | Description of indirect reference is illegal.                       |                  |
| 1306 | Can't find end of strings(str).                                     |                  |
| 1307 | Description of expression is illegal.                               |                  |
| 1308 | Macro constant (macro) isn't defined.                               |                  |
| 1309 | Symbol (symbol) isn't defined.                                      |                  |
| 1310 | Immediate value is illegal.   |                  |
| 1311 | Divide by 0.  |                  |
| 1313 | The value is over the maximum value of which can be treated by MCU. |                  |
| 1314 | Register name is using for macro variable name.                     |                  |

| No.  | Error Message  | Notes and Action   |
|------|--|--|
| 1400 | Address value is out of range.                                   | Specified address exceeds the MCU's maximum address.                                 |
| 1401 | Target program is already stopped.                               |  |
| 1402 | The number of break point is over the limit(limit).              |  |
| 1403 | The break point isn't defined at that address.                   |  |
| 1404 | Data value is out of range.                                      |  |
| 1406 | Can't read/write, because there are no memory at that area.      | Do not attempt to reference or write to an address for which no memory is installed. |
| 1407 | Can't get enough memory.   | Insufficient memory. Quit other applications or increase memory.                     |
| 1408 | Register value is out of range.                                  |  |
| 1409 | Can't execute that command, when the target program is running.  |  |
| 1410 | Start address is larger than end address.                        |  |
| 1411 | STOP execution.  |  |
| 1412 | Can't find source lines which include that address.              | There is no source line information at the specified address.                        |
| 1413 | That command has not yet supported.                              |  |
| 1417 | Can't search more on the stack.                                  |  |
| 1418 | Specified times of number is over than 65535.                    |  |
| 1420 | Breakpoint of other type already set.                            | Check the type of breakpoint.  |
| 1428 | The specified register cannot be changed due to CPU restriction. |  |
| 1431 | Stack overflow or underflow has occurred.                        |  |
| 1432 | The time measurement counter has overflowed.                     |  |
| 1433 | The specified register cannot be read due to CPU restriction     |  |
| 1450 | Address value is out of range.                                   |  |
| 1452 | Bit number is out of range.                                      |  |
| 1453 | STOP execution.  |  |
| 1454 | Data value is out of range.                                      |  |
| 1456 | Can't find File (filename).                                      |  |
| 1460 | The time measurement counter has overflowed.                     |  |
| 1462 | Pass count value is out of range.                                |  |
| 1463 | Can't execute that command, when the target program is running.  |  |
| 1464 | Target MCU is reset state.                                       | Reset the target system.   |
| 1465 | Target MCU is unable to reset.                                   | Reset the target system.   |
| 1466 | Target MCU is HOLD state.  |  |
| 1467 | Target MCU is not given clock.                                   |  |
| 1468 | Target MCU is not given power.                                   | Reset target system.   |

| No.  | Error Message                                 | Notes and Action     |
|------|---|----------------------|
| 1473 | Can't find trace data which is able to refer. |                      |
| 1474 | Cycle value is out of range.                  |                      |
| 1475 | Target MCU is not under control.              | Reset target system. |
| 1476 | First data is larger than second data.        |                      |
| 1477 | First address is larger than second address.  |                      |

| No.  | Error Message   | Notes and Action  |
|------|---|---|
| 1500 | Address value is out of range.                            | The specified address is greater than the MCU's maximum address                               |
| 1501 | Bit number is out of range.                               |   |
| 1502 | File (filename) is broken.                                |   |
| 1503 | Can't find File(filename).                                |   |
| 1505 | Illegal character in the strings.                         |   |
| 1507 | Can't find that line number.                              |   |
| 1508 | Multiple definition of symbol/label.                      |   |
| 1509 | There are no code at that line.                           | No machine language has been generated at the address corresponding to the specified line No. |
| 1510 | Can't get enough memory.                                  |   |
| 1513 | Can't find source lines which correspond to that address. |   |
| 1514 | Can't find symbol (symbol).                               |   |
| 1515 | Can't find the scopes which include that address.         |   |
| 1516 | Loading is canceled.                                      |   |
| 1519 | The register name is wrong.                               |   |

| No.  | Error Message                                 | Notes and Action  |
|------|---|---|
| 1704 | The connection with the target isn't created. |   |
| 1705 | Can't connect with the target.                |   |
| 1707 | Time Out ERROR.                               | A time-out error occurred in communication with the target system.                    |
| 1712 | Communication ERROR.                          | Connection to the target system was lost during communication with the target system. |
| 1713 | Communication ERROR.                          | A communications error occurred while sending data to the target system.              |
| 1714 | Communication ERROR.                          | A communications error occurred while receiving data from the target system.          |

| <b>No.</b> | <b>Error Message</b>                 | <b>Notes and Action</b>   |
|------------|--------------------------------------|---|
| 2200       | Can't execute more come instruction. |   |
| 2201       | Can't find strings.                  |   |
| 2202       | Can't execute more step instruction. |   |
| 2203       | Cycle value is out of range.         |   |
| 2204       | Can't find that address.             |   |
| 2205       | Can't find File(filename).           | There is an error in the specified filename or no permission to read. |
| 2206       | Can't find trace data.               |   |

| <b>No.</b> | <b>Error Message</b>                 | <b>Notes and Action</b>   |
|------------|--------------------------------------|---|
| 2400       | Address value is out of range.       |   |
| 2401       | Data value is out of range.          |   |
| 2402       | Start Address value is out of range. | You specified an ending address that is less than the starting address. |
| 2403       | Value is under (num).                | Specify a value of num or more.   |
| 2404       | Data value is out of range.          |   |

| <b>No</b> | <b>Error Message</b>                | <b>Notes and Action</b> |
|-----------|-------------------------------------|-------------------------|
| 2500      | Operation code (ope-code) not found |                         |
| 2501      | File (filename) not found           |                         |

| <b>No.</b> | <b>Error Message</b>                                    | <b>Notes and Action</b>                                     |
|------------|---|---|
| 5200       | Can't execute Come command, without unused break point. | Delete one of the break points, then perform Come execution |

| <b>No.</b> | <b>Error Message</b>   | <b>Notes and Action</b> |
|------------|------------------------|-------------------------|
| 5500       | Value is out of range. |                         |

| <b>No.</b> | <b>Error Message</b>                        | <b>Notes and Action</b>  |
|------------|---|--|
| 5700       | The data value is too large.                |  |
| 5701       | The address area is illegal.                |  |
| 5702       | Address value is out range for scroll area. | The address specified as the scroll range is greater than the MCU's maximum address. |

| <b>No.</b> | <b>Error Message</b>                         | <b>Notes and Action</b> |
|------------|--|-------------------------|
| 5900       | Can't open Script File(filename).            |                         |
| 5901       | Script File (filename) is already opened.    |                         |
| 5902       | Script File is not open.                     |                         |
| 5903       | Can't open Log File (filename).              |                         |
| 5904       | Can't open more Log File.                    |                         |
| 5905       | Can't open Log File.                         |                         |
| 5906       | File (filename) is already log on.           |                         |
| 5907       | Can't open View File (filename) for new/add. |                         |

| <b>No</b> | <b>Error Message</b>                 | <b>Notes and Action</b> |
|-----------|--------------------------------------|-------------------------|
| 10022     | Can't execute more come instruction. |                         |
| 10024     | Can't execute more step instruction. |                         |
| 10025     | Cycle value is out of range.         |                         |
| 10026     | Can't find that address.             |                         |

[ MEMO ]



# Index



## Index

### A

ASM Watch Window..... 5,32, 33,80  
 Address breakpoint..... 45  
 Assign Statement..... 120

### B

Break Functions ..... 4

### C

Character constant ..... 125  
 Comment statement ..... 121  
 Constant ..... 122  
 Coverage Measurement ..... 5,41,101  
 Coverage Window ..... 41

### D

Data file..... 6  
 Disassemble file ..... 7  
 Dump Window ..... 30

### E

Environmental setup file ..... 7  
 Error Messages ..... 126  
 Exit ..... 91  
 else ..... 120  
 endi ..... 120  
 endw ..... 120

### F

Font size ..... 13

### H

H/W Break Point Setting dialog box..... 45  
 Hardware Break ..... 4,97  
 Hardware break point ..... 45  
 Help file ..... 6

### I

Init Dialog Box..... 12,90  
 Installation..... 11  
 Intel HEX-format file ..... 6, 7  
 if ..... 120

### L

Label..... 5,123  
 Line Assemble..... 92  
 Log file..... 7,34  
 Logging function ..... 86

Loop Statement ..... 120

### M

Macro variable..... 123  
 Main command..... 17  
 Memory Window ..... 28  
 Monitor version ..... 94  
 Multi-Windowing Function ..... 4

### O

Operator ..... 125

### P

PD45 Window..... 17  
**PD45** setup..... 12  
**PD45** version..... 94  
 Program Window ..... 13,22,93  
 Program counter ..... 22, 25

### R

Real Time Trace ..... 4,72  
 Real-time Trace Event..... 98  
 Register Window ..... 27,79  
 Register variable ..... 124

### S

S/W Break Point Setting dialog box..... 43  
 Script Command ..... 119  
 Script Window ..... 34, 35,85  
 Script file ..... 6,12,34,88  
 Search ..... 93  
 Setup ..... 11  
 Software Break ..... 4,43  
 Software break point ..... 43  
 Source Level Debugging Function ..... 5  
 Source Window ..... 13,25,93  
 Stack Overflow ..... 97  
 Starting **PD45** ..... 11  
 Symbol ..... 5,123  
 Symbol file..... 6

### T

TAB value ..... 13  
 Table of Script Commands ..... 107  
 Time Measurement..... 5,48  
 Trace Point Setting dialog box ..... 47,75  
 Trace Window..... 36,72  
 Trace point..... 47  
 Trigger break point..... 45

**V**

View buffer ..... 7,34,87  
View file..... 7,34,87

**W**

Writing Script Files ..... 119  
while ..... 120

# M3T-PD45 V.1.00 User's Manual

---

Rev. 1.00  
October 16, 2003  
REJ10J0368-0100Z

COPYRIGHT ©2003 RENESAS TECHNOLOGY CORPORATION  
AND RENESAS SOLUTIONS CORPORATION ALL RIGHTS RESERVED

**M3T-PD45 V.1.00**  
**User's Manual**



**Renesas Electronics Corporation**

1753, Shimonumabe, Nakahara-ku, Kawasaki-shi, Kanagawa 211-8668 Japan

REJ10J0368-0100Z