To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 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 1st, 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.

User’s Manual

ID78K4

Integrated Debugger

Reference

For HP9000 Series 700 (HP-UX), SPARCstation Family (SunOS), and RISC NEWS (NEWS-OS) Ver 1.21 and later

Document No. U11960EJ1V0UM00 (1st edition)
Date Published June 1997 J

Trademarks:

- UNIX is a registered trademark licensed by X/Open Company limited in the US and other countries.
- X Window System is a trademark of Massachusetts Institute Technology.
- OSF/Motif is a trademark of Open Software Foundation, Inc.
- Ethernet is a trademark of Xerox Corporation.
- SPARCstation is a trademark of SPARC International, Inc. in the United States, licensed exclusively through Sun Microsystems, Inc. SunOS is a trademark of Sun Microsystems, Inc.
- Other company names and products which may appear in this manual are trademarks or registered trademarks of the relevant companies.

The export of this product from Japan is prohibited without governmental license. To export or re-export this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 800-366-9782
Fax: 800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.1.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 253-8311
Fax: 250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.
Sao Paulo-SP, Brasil
Tel: 011-889-1680
Fax: 011-889-1689

J96. 8

Preface

Conventional debuggers are used by entering commands directly. The ID78K4 integrated debugger, on the other hand, runs under X Window to provide a friendly, easy-to-use GUI (Graphical User Interface). Its operation is mouse-based and intuitive, so that the user does not have to refer to a manual. Also, frequently used commands are represented as buttons, allowing their use simply by clicking the corresponding button.

Purpose

This manual is intended to explain the capabilities of the integrated debugger. It assumes that the reader is reasonably familiar with the operation of in-circuit emulators and X Window.

Operating environment

Host machine

Sun:	SUN4
HP:	HP9000/700
Sony:	NEWS

Operating system

UNIX + X Window

How to read this manual

- Organization

  Use this manual as a reference manual for the integrated debugger.

Table of Contents

Chapter 1 Overview

1.1 Overview of ID78K4

ID78K4 is cross development software for efficiently debugging programs developed for the NEC 78K4 series of single-chip microcomputers and target boards.

To run ID78K4, the host machine (running UNIX as its OS) for debugging, connected to a local area network, must exchange data with the in-circuit emulator.

ID78K4 uses an X Window-based mouse-based graphical user interface as its user interface to provide an environment that offers better operability.

1.2 Overview of Functions

ID78K4 runs on a host machine that uses UNIX as its OS. It communicates with the in-circuit emulator to provide an optimum environment for debugging programs developed for the 78K4 series and target boards.

1.2.1 Functions and Features

The functions and features of ID78K4 are explained below.

(1) GUI

ID78K4 runs under an X Window System environment, allowing the user to debug programs simply by manipulating the mouse. The required buttons and menus are displayed in the corresponding windows, allowing easy switching from the currently displayed information to other related information.

(2) Debugging at the C source level

The following operations can be performed effectively by specifying function names and line numbers at the source code level: referencing/setting variables and structures, displaying programs, and setting breakpoints.

(3) Debugging at the assembly language level

The following operations can be performed effectively by specifying labels and addresses at the assembly language level: referencing/setting symbols and register values, displaying programs, and setting breakpoints.

(4) Available in-circuit emulator functions

The fine event setting functions of the in-circuit emulator can be used to set breakpoints and trace programs.

(5) Debugging in a network environment

ID78K4 on the host machine, which is connected to a network (Ethernet), can be used to perform remote debugging for a target system that is directly connected to that network.

(6) Tracing (monitoring) (automatic display update at execution stop)

The values in the currently displayed windows (view windows and view/setting windows) are automatically updated when execution of the program to be debugged terminates.

(7) Saving and restoring the debugging status

The status at any point during debugging can be saved. Subsequently, if necessary, that saved status can be restored.

(8) Displaying source text from a function

By selecting a function from a function list, the source text containing that function is displayed.

(9) Changing key binding

A keyboard accelerator function is added to allow the user to change the key binding by using a resource file.

1.3 Operating Environment

1.3.1 Hardware Environment

Host machine

Sun:	SPARCstation
Sony:	NEWS (RISC)
HP:	HP9000/700

In-circuit emulator

- IE-784000-R

1.3.2 Software Environment

OS

Sun:	SunOS
NEWS:	NEWS-OS
HP:	HP-UX

In addition to the above OS, the following software packages are required:

- X Window System X11R5
- Window manager mwm

Chapter 2 Installing and Starting the Debugger

2.1 Initializing the IE

Before the in-circuit emulator (IE) can be used for debugging, the host name, port name, and other items must be set. Always set these items. The settings related to network information are written to the EEPROM mounted on the IE-78000-R-SV3. Once these items have been set, they remain set even after power-off. Set these items as described in the manual provided with the IE network board.

2.2 Installation

For an explanation of how to install the debugger, refer to the description in the manual provided with the media. Upon the successful completion of installation, the following directories are generated:

./Setup:	Directory which contains files for creating debugger start scripts
./lib:	Directory which contains the debugger main section
./bin:	Directory which contains start scripts
./hlp:	Directory which contains help files

Caution Device files are required to run ID78K4. Install the device files in the installed directory in the same way. The device files will exist under the directory named "./dev".

2.3 Initializing the Debugger

First, edit id78k4userdef in the Setup directory. Edit the contents of id78k4userdef as described in Table 2-1.

Table 2-1. Contents of id78k4userdef

Description	Contents
DEFAULT_CHIP=EDIT_HERE_CHIPNAME	Specify the main target chip to be used for "EDIT_HERE_CHIPNAME." (Specify the chip number, excluding the two high-order digits (78). Example: For the 784026, specify 4026. For the 784038, specify 4038.)
TIME=1	Specify whether to cause a timeout when an excessively long time is required to connect the debugger host machine to an IE host due to network congestion. Specify 1 when a timeout is to be caused. Otherwise, specify 0.
HOSTMAX=10	Specify, as a decimal number, the maximum number of IEs to be specified in this script. This specification can be omitted. By default, a value of 10 is assumed.
HOST0=EDIT_HERE_IEHOST EDIT_HERE_IEPORT	Specify the name of an IE to be connected for "EDIT_HERE_IEHOST" and the port number of the IE for "EDIT_HERE_IEPORT." The port number is handled as a hexadecimal number. When specifying more than one IE, replace 0 of HOST0 with 1, 2, 3, ..., or n sequentially. Specify the IE name and port number in the same way. The numbers indicate the connection priority, with lower numbers indicating the higher priority.

Example: Specifications: The chip name is 4026, a timeout is to be caused, the maximum number of hosts is 8, and eight IE hosts are specified.

```
DEFAULT_CHIP=4026
TIME=1
HOSTMAX=8
HOST0=necie01 1000
HOST1=necie02 2000
HOST2=necie03 1000
HOST3=123.22.39.21 1000
HOST4=123.22.39.22 1000
HOST5=necie99 2000
HOST6=necie100 3000
HOST7=necie200 1000
```

After the completion of editing, create a script command for starting ID78K4.

```
% ./id78k4install
```

When this script is executed, a starting script named id78k4 is created in ../bin.

This starting script is used as the command for starting ID78K4.

2.4 Starting and Exiting

2.4.1 Starting the Debugger

Execute the starting script.

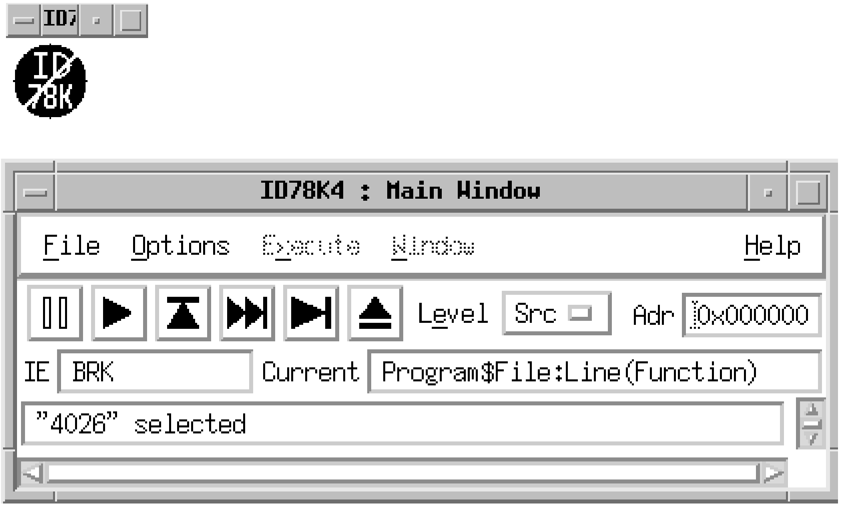
```
% id78k4
```

When this script is executed, the following message appears:

```
NEC 78K/4 series Integrated Debugger ID78K4
        Copyright(C) NEC Corporation 1996
```

Then, as shown in Figure 2-1, the emergency button appears and the Main window opens.

Figure 2-1. Window Displayed When the Debugger Starts



Level Src

Adr 0x000000

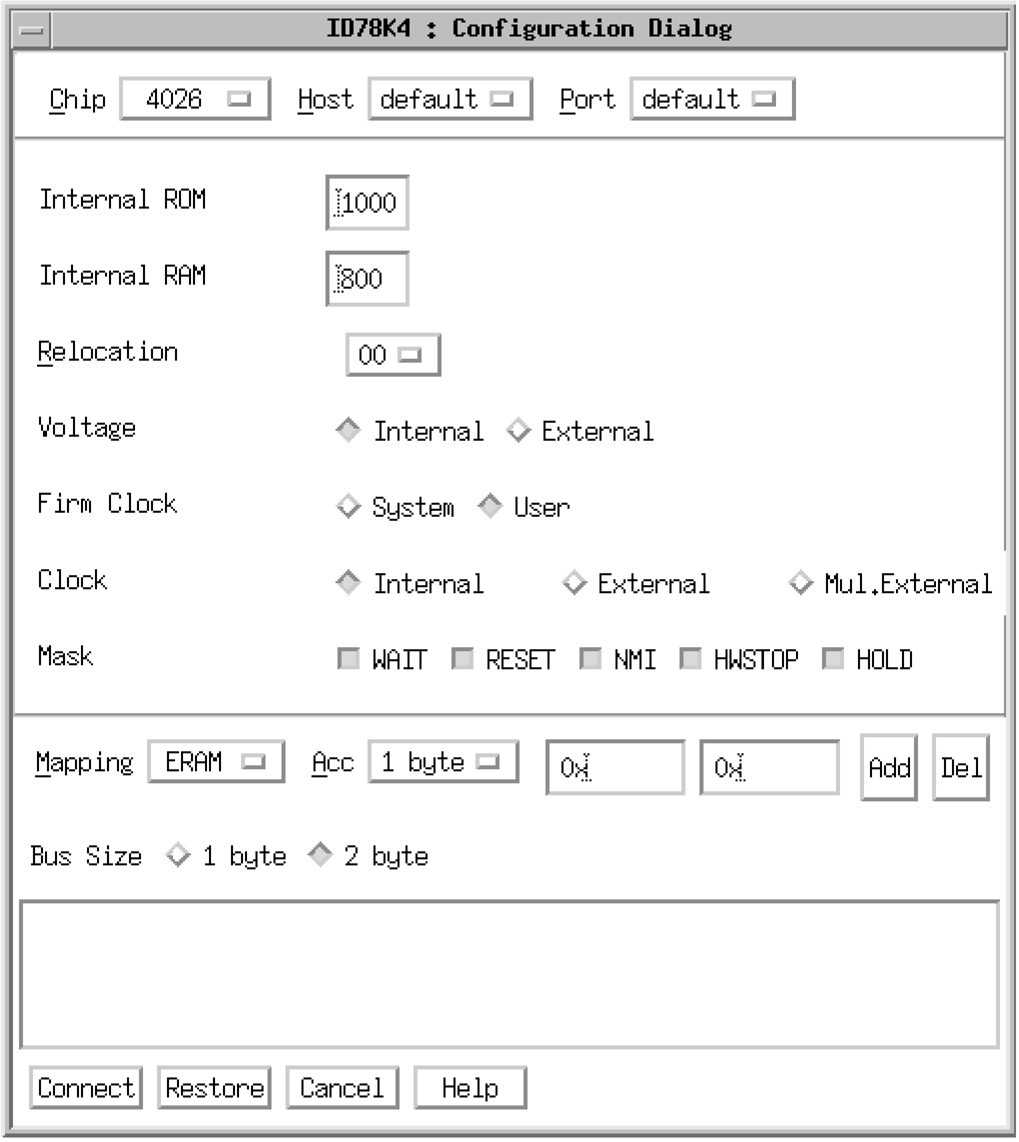
IE BRK

Current Program$File:Line(Function)

"4026" selected

Then, the Configuration dialog box opens to enable the connection of the debugger host machine to an IE.

Figure 2-2. Configuration Dialog Box



ID78K4 : Configuration Dialog

Chip 4026   Host default   Port default

Internal ROM 1000

Internal RAM 800

Relocation 00

Voltage ● Internal ○ External

Firm Clock ○ System ● User

Clock ● Internal ○ External ○ Mul.External

Mask □ WAIT □ RESET □ NMI □ HWSTOP □ HOLD

Mapping ERAM   Acc 1 byte   0x   0x

Bus Size ○ 1 byte ● 2 byte

As the configuration information, the chip specified in the created starting script is set by default. When the debugger is connected with default set for Host and Port, it automatically attempts to sequentially connect the IE hosts specified in the created starting script. The first target IE host is HOST0. If the IE specified for HOST0 is already in use, the debugger attempts to connect the IE specified for HOST1. When the IE specified for HOST1 is also in use, the debugger attempts to connect the IE specified for HOST2.

The Host option button can be clicked to display the option menu. Using this menu, the user can select an IE host from among those specified in the created starting script. When an IE host is selected from the menu, the port displayed for Port is also changed automatically. When an IE host is specified for connection and the specified IE host is already in use, an error is returned.

Like the Host option button, the Port option button can be clicked to display the option menu. Using this menu, the user can select a port from among those which appear in the created starting script. When a port is specified from the menu, the debugger automatically attempts to connect the IE hosts for which the specified port is set. The connection order is based on the connection priority specified by n of HOSTn.

An example is given below:

Example

```
DEFAULT_CHIP=4026
TIME=1
HOSTMAX=8
HOST0=necie01 1000
HOST1=necie02 2000
HOST2=necie03 1000
```

When the debugger is connected with 1000 specified for Port in the created starting script listed above

First, the debugger attempts to connect necie01. When necie01 is already in use, the debugger searches for an IE with port 1000. In this case, the debugger attempts to connect necie03.

When multiple users share multiple IE hosts on a network, the same port number can be set for the IE hosts with the same chip, allowing the debugger to successfully connect an available IE host without having to consider the IE host names.

2.4.2 Starting Script Options

The following options are available for the starting script.

To change a setting in the starting script, start the debugger with the required options specified.

-chip chip-name: Changes the specified chip. Specify a chip name. Specify the chip number, excluding the two high-order digits (78). (Example: For the 784026, specify 4026. For the 784038, specify 4038.)

-iehost IE-host-name: Changes the default IE host name to that of the specified IE host.

-ieport port: Changes the default port to the specified port.

-project file-name: Specifies and starts a project file when the debugger is started.
A project file contains debug information. It is manipulated using the Main window.

-insmode: Starts the debugger in default instruction debug mode.

-raiseonce: Starts the debugger in that mode in which the dialog boxes are always displayed as inactive windows. The debugger can be started with this option specified to conceal the dialog boxes.

-version: Displays the version of ID78K4.

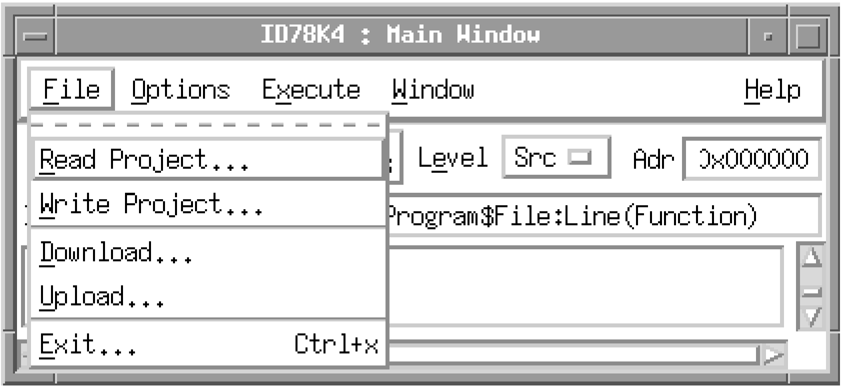
-help: Displays an explanation of the starting script options.

In addition to the above options, those options supported by the X tool kit (such as -fg and -bg) can also be specified.

2.4.3 Exiting the Debugger

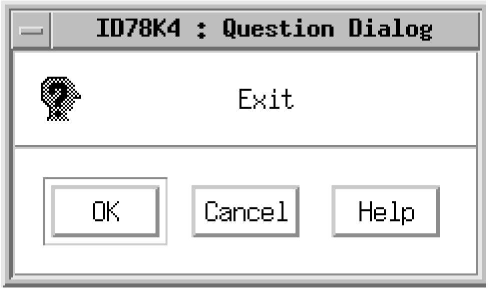
In the main window, select File, then Exit.

Figure 2-3. Selecting File in the Main Window



A dialog box will appear, requesting confirmation. Click the OK button to exit the debugger.

Figure 2-4. Exiting Confirmation Dialog Box



2.4.4 Handling Debugger Errors

The debugger creates two processes, xid78k4 and xid78k4e, at startup.

If a debugger error occurs, delete these processes (if they still exist), reset the IE, then restart the debugger.

Similarly, to forcibly stop the debugger during operation, delete these processes, reset the IE, then restart the debugger.

Chapter 3 Explanation of Terms

3.1 Overview

This chapter explains the following terms, used in the explanation of ID78K4.

(1) Source level debugging

(2) Instruction level debugging

(3) Delimiter

(4) File

(5) Current file

(6) Function

(7) Current function

(8) Structure

(9) Variable expression

(10) Common integer constant

(11) Stack frame number

(12) Line

(13) Register name

(14) Character constant and string constant

(15) List

3.2 Source Level Debugging

Source level debugging is the debugging of C source programs and structured assembler source programs.

At this level, the user can use identifiers such as source file names, line numbers, function names, and variable names in source programs.

To enable debugging at this level, the debug option (-g) must be specified when the target program is compiled.

3.3 Instruction Level Debugging

Instruction level debugging is the debugging of machine code.

At this level, symbols, addresses, and registers are used.

3.4 Delimiter

Files, functions, variables, and lines, as explained in Section 3.5 and after, can be qualified using the corresponding qualifier listed in Table 3-1.

Table 3-1. Delimiters

Qualification	Symbol	Example
Qualifies a file with a program.	$	program$file
Qualifies a function or variable with a file.	#	file#func file#variable
Qualifies a line with a file.	:	file:38
Qualifies a variable with a function.	!	func!variable

3.5 File

ID78K4 supports the following file types:

- Source text files
- Load module files
- Hexadecimal files
- Debugging environment files
- Binary files

The file specification formats are as follows:

a. file

b. path/file

path indicates a path name; file indicates a file name.

When no path name is specified (a), a file in the current directory is assumed. When a source file is specified and the path to the source file (source path) is specified, the source path then the current directory are searched for the file.

File specification example

test.c	Source file "test.c" in the current program or on the specified path

3.6 Current File

The current file is the source file which contains the instruction at the current breakpoint.

When lines and functions in the current file are specified by commands, the specification of the file name can be omitted.

3.7 Function

In this manual, a function refers to those functions constituting a C source program.

The function specification format is as follows:

a. _func

b. file#_func

func indicates a function name in a source program, while file indicates a file name.

In format a, a corresponding function is retrieved from the current file of the current program, starting with the effective static functions, then the global functions.

In format b, a corresponding function is retrieved from the specified file of the current program, starting with the effective static functions, then the global functions.

Function specification examples

_main	Function "main" in the current file
test.c#_calc_data	Static function "calc_data" in the "test.c" file

3.8 Current Function

The current function is that function which contains the instruction at the current breakpoint.

When a local variable in the current function is to be accessed, the specification of the function name can be omitted.

3.9 Structure

In C, both structures and unions are generically referred to as structures.

The term structure is used when a structure or a union variable is used without explicitly specifying a member.

3.10 Variable Expression

Expressions conforming to the C expression specification that do not contain operators, functions, macros, and real operations are defined as variable expressions that can be specified in commands.

The C scope rules apply to the scope (effective range of definition/declaration) of variable expressions for the current function.

The variable expression specification formats are as follows:

a. variable-expression Retrieved from the current function according to the scope rules.

b. file#variable-expression Retrieved from static variables in file.

c. func@variable-expression Retrieved from the specified function (func) according to the scope rules.

d. stack-frame-number@variable-expression
Retrieved from the function specified by the stack frame number according to the scope rules.

Variable expressions are expressions conforming to the C expression specifications that consist of other than operators ++, --, sizeof, cast, &&, ||, ?:, =, +=, -=, *=, /=, %=, &=, ^=, |=, <<=, >>=, and ",", functions, macros, and real operations.

3.11 Common Integer Constant

ID78K4 supports integer constants in both C and assembly language. Such integer constants are generically referred to as common integer constants.

The specification format of common integer constants is as follows (Backus-Naur form (BNF)):

```
<common-integer-constant>  ::= <C-integer-constant>
                               <ASM-integer-constant>
```

<C-integer-constant> indicates an integer constant in C, while <ASM-integer-constant> indicates an integer constant in assembly language.

For integers, the following formats are also available, regardless of whether they are <C-integer-constant> or <ASM-integer-constant> ("+" indicates that at least one character in the immediately preceding brackets appears, while "*" indicates that at least zero appears).

Hexadecimal integer	0[xX][0-9a-fA-F]+ or [0-9a-fA-F]+[hH]
Decimal integer	(0)[dD][0-9]+ or [0-9]+[dD]
Octal integer	0[0-7]* or [0-7]+[oO]
Binary integer	[01]+[bB]

3.12 Stack Frame Number

A stack frame number is a decimal integer beginning from 1. It indicates a level in a stack that corresponds to a local variable in a function.

Stack frame number 1 corresponds to the current function.

3.13 Line

A line is specified to identify a specific line in a source file.

The line specification formats are as follows:

a.	line-number	Line number in the current file
b.	file:line-number	Line number in the specified file

Specify a line number with an integer constant beginning from 1.

3.14 Register Name

A register name is specified with either its absolute name or functional name.

3.15 Character Constant and String Constant

A character constant is specified by enclosing an ASCII character with single quotation marks ('). A string constant is specified by enclosing an ASCII string consisting of more than 0 characters with double quotation marks (").

The following C escape characters can also be specified as ASCII characters:

\a	Bell
\n	Line feed
\t	Horizontal tab
\b	Backspace
\r	Carriage return
\f	Form feed
\v	Vertical tab
\"	" itself
\'	' itself
\?	? itself
\\	Backslash
\ddd	Octal specification character
\xhh	Hexadecimal specification character

3.16 List

The data specification formats that are supported by ID78K4 are generically referred to as lists.

The list specification formats are as follows (Backus-Naur form (BNF)):

```
<list>  ::= <character-constant>
          | <string-constant>
          | <common-integer-constant>
          | <list><character-constant>
          | <list><string-constant>
          | <list><common-integer-constant>
```

Chapter 4 Explanation of Debugger Functions

4.1 System Operating Modes

The system operating mode indicates the current operating state of the system, i.e., whether the “user program execution (emulation) functions” and the “analyzer functions” are activated.

4.1.1 Operating Mode Types

There are three system operating modes. Commands are restricted according to the current system operating mode.

Break mode

The execution of both “user program execution (emulation) functions” and “analyzer functions” is stopped.

Emulation mode

The “user program execution (emulation) functions” are activated but execution of the “trace functions” is stopped. This mode is used when the user does not want the execution of a user program to be stopped. In this mode analyzer functions other than the tracer, such as coverage and timer measurement, are activated.

Trace mode

Both the “user program execution (emulation) functions” and “analyzer functions” are activated.

4.1.2 System Operating Mode

The current system operating mode is displayed in the status bar of the main window.

System operating mode	CPU	Tracer
Break mode	Stopped	Stopped
Emulation mode	Activated	Stopped
Trace mode	Activated	Activated

4.1.3 System Operating State

The relationship between an emulation CPU and the analyzer functions is shown in the following figure. The relationship shown here is just one example.

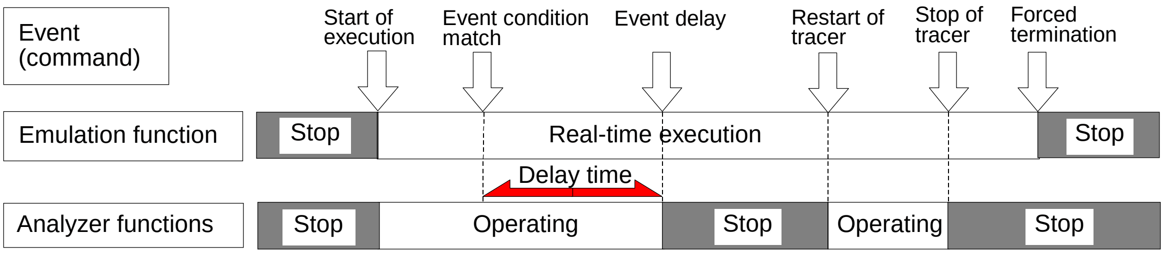


Figure 4-1. Example of System Operating State

4.2 Using the Basic Functions

4.2.1 Clock Selection Function

The clock selection function is used to specify the clock source to be supplied to an emulation CPU (target device). Either of the following three clock sources can be selected:

Fixed clock inside the in-circuit emulator (25 MHz) (Internal)
User-established clock (External)
Multiplied user-established clock (Multiple Ext)

The desired clock source can be selected when the debugger is first started or from within the Configuration dialog box.

There are two ways of specifying a user-established clock to supply any clock:

a. Create a clock on the target system and supply the clock to an emulation CPU via the emulation probe.
b. Install a transmitter in the clock socket on the break board in the main section of the in-circuit emulator and supply the clock to an emulation CPU.

When the clock source is changed, the emulation CPU is reset.

4.2.2 Mapping Functions

There are five mapping functions, which allow the following settings for address areas other than internal ROM, RAM, and SFRs. These settings can be made when the debugger is first started or from within the Configuration dialog box.

Internal ROM

The memory area specified as internal ROM represents the internal ROM of the target device.
With this type of mapping, the target device can access the memory of the in-circuit emulator.
When the target device writes data into this memory area, a write protect break is generated.

Internal RAM

The memory area specified as internal RAM represents the internal RAM of the target device.

With this type of mapping, the target device can access the memory of the in-circuit emulator.

User area mapping (Target)

The memory area specified for user area mapping allows access to the memory of the target system.

With this type of mapping, the target device can access the memory of the target system.

IE alternate ROM (Emulation ROM)

The memory area specified for IE alternate ROM represents the additional ROM of the target device.

With this type of mapping, the target device can access the memory of the in-circuit emulator.

When the target device writes data into this memory area, a write protect break is generated.

IE alternate RAM (Emulation RAM)

The memory area specified for IE alternate RAM represents the additional RAM of the target device.

With this type of mapping, the target device can access the memory of the in-circuit emulator.

4.2.3 Reset Functions

The reset functions are used to reset the entire in-circuit emulator system or the emulation device only.

Reset of emulation device only (CPU)

Reset of entire system (CPU & BODY)

Reset of symbol information only (SYMBOL_ALL)

These functions can be specified from within the Selection dialog box.

4.2.4 Load Function

The load function is used to load specified files such as the debugging environment, object, load module, and symbol files individually.

There are two types of files to be loaded: View files for screen reference and information files for updating information in the debugger.

A view file stores screen information that was current when the file was saved. When a view file is loaded, a reference window opens.

The following table lists the types of view files.

File	Window	Explanation
Variable view file (file name: XXXXXXXX.var)	Variables window	Stores variable information.
Local variable view file (file name: XXXXXXXX.loc)	Local Variables window	Stores local variable information.
Stack trace view file (file name: XXXXXXXX.stk)	Stack Trace window	Stores stack trace information.
Register view file (file name: XXXXXXXX.reg)	Registers window	Stores register information.
SFR view file (file name: XXXXXXXX.Sfr)	SFR window	Stores SFR information.
Memory view file (file name: XXXXXXXX.mem)	Memory window	Stores memory information.
Coverage view file (file name: XXXXXXXX.cov)	Coverage window	Stores coverage information.
Assemble view file (file name: XXXXXXXX.dis)	Disassemble window	Stores assemble information.
Trace view file (file name: XXXXXXXX.trv)	Trace View window	Stores trace information.
Timer view file (file name: XXXXXXXX.tim)	Timer window	Stores timer information.

The following table lists the types of information files.

File	Window	Explanation
Object file (file name: XXXXXXXX.hex)	Download dialog box	Stores the object code (in Intel standard hexadecimal format) of a user program.
Symbol table file (file name: XXXXXXXX.sym)	Download dialog box	Stores the symbols defined by the user on a source for a user program.
Load module file (file name: XXXXXXXX.lnk)	Download dialog box	Stores the object code, symbols, and source information of a user program.
Project file (file name: XXXXXXXX.prj)	Main window	Stores a debugging environment.
Event setting file (file name: XXXXXXXX.evt)	Event Manager window	Stores event setting information.

4.2.5 Emulation Execution Functions

The emulation execution functions are used to start “user program execution (emulation)” by an emulation CPU and the analyzers. The functions are classified according to the emulation execution mode, as follows:

Real-time execution functions:

- Go (▶ button) Performs real-time execution.
  The program breaks upon the occurrence of a break event.
- Go NoBreak Performs real-time execution.
  The program ignores a break event and continues execution.
- CPU Reset & Go Resets the emulation CPU then performs real-time execution.

Non-real-time execution functions:

- Return (⤒ button) Performs step execution until control returns to the calling function.
- Next (⏭ button) If source level is selected
  Performs Next step execution at the source level.
  If instruction level is selected
  Performs Next step execution at the instruction level.
- Step (▶| button) If source level is selected
  Performs step execution at the source level.
  If instruction level is selected
  Performs step execution at the instruction level.
- Slowmotion Performs step execution continuously.

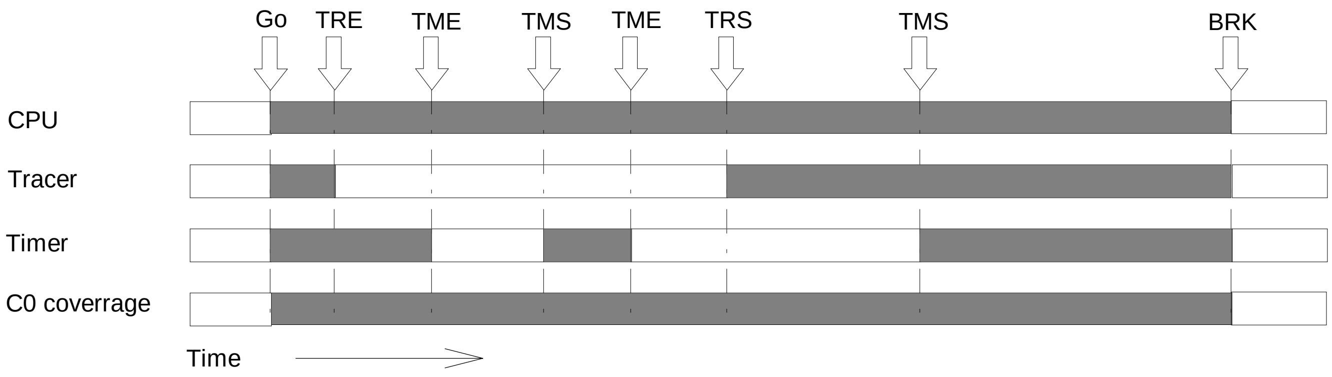
Real-time execution functions

There are two real-time execution functions: "Go" which executes a user program until the occurrence of a break event and "Go NoBreak" which executes a user program continuously, ignoring any break event that may occur.

Go command

During real-time execution by the Go command, the user program is executed, starting from the specified address. The user program stops upon the occurrence of a break event. Each analyzer becomes operable when the program is executed, and is executed and stopped by events.

The relationship between the CPU, tracer, timer, and coverage during real-time execution by the Go command is shown in the following figure.



□ : Stopped ▨ : Executing

TRS: Trace start event, TRE: Trace stop event, TMS: Timer start event, TME: Timer stop event
Go: Go command, BRK: Break event

Figure 4-2. Example of System Operating State (Go)

Go NoBreak command

During real-time execution by the Go NoBreak command, the user program is executed, starting from the specified address. The user program stops upon the issue of a stop command. Each analyzer becomes operable when the program is executed, and is executed and stopped by events.

CPU Reset & Go command

During CPU Reset & Go execution,

(1) The emulation CPU is reset.
(2) The program is executed with a reset vector.

The emulation CPU is reset before the program is executed. Subsequently, the operation is the same as that of the Go command.

Non-real-time execution functions

The non-real-time execution functions are roughly divided into “Step” which performs step execution; “Return” which performs step execution until control returns to the calling function; “Next” which performs Next step execution; and “Slowmotion” which performs continuous step execution.

Step command

During step execution by the Step command,

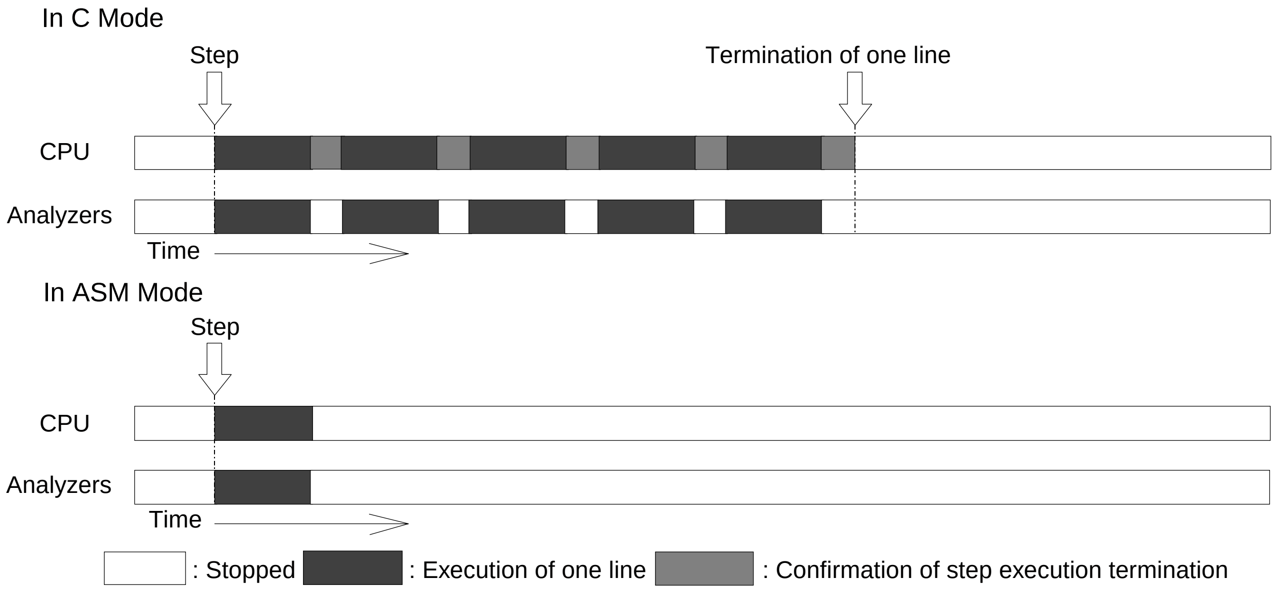
- In C Mode,

  Step execution is performed for one line starting from a specified source line.
- In ASM Mode,

  One instruction is executed starting from a specified address.

After execution, each window is updated.

The relationship between the CPU and the analyzers during step execution by the Step command is shown in the following figure.



Step: Step command, Termination of one line: Termination of execution of an instruction on one line

Figure 4-3. Example of System Operating State (Step)

Return command

During step execution by the Return command, step execution is performed until control returns to the calling function.

The concept of step execution by the Return command is shown in the following figure.

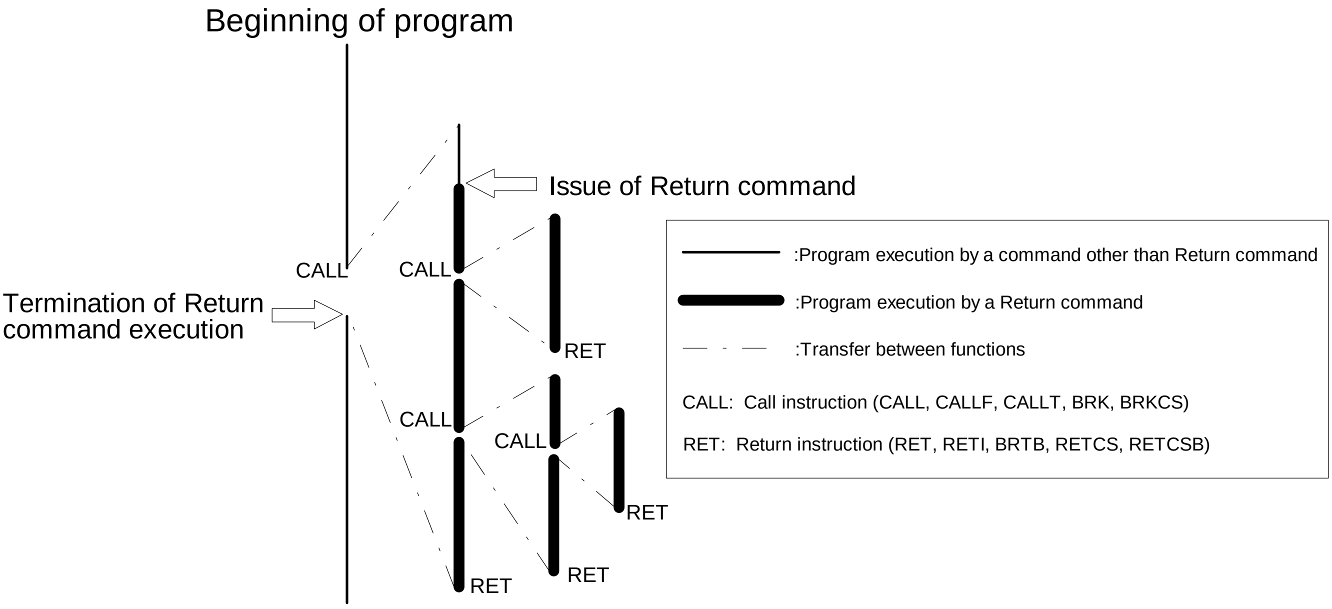

Beginning of program
Issue of Return command
Termination of Return command execution
CALL
CALL
CALL
CALL
RET
RET
RET
RET
:Program execution by a command other than Return command
:Program execution by a Return command
:Transfer between functions
CALL: Call instruction (CALL, CALLF, CALLT, BRK, BRKCS)
RET: Return instruction (RET, RETI, BRTB, RETCS, RETCSB)

Figure 4-4. Concept of Return Command Execution

Next command

Next step execution by the Next command differs depending on whether a call statement or a statement other than a call statement is executed. Depending on the debug mode, the following can be used as a call instruction:

- In source mode,
  Line calling a function
- In instruction mode,
  CALL, CALLF, CALLT, BRK, and BRKCS instructions

The operation of the Next command is explained below:

- If a call statement is executed,
  Next step execution is performed for the destination subroutine of the call statement. (Step execution is performed until the nesting depth for Next step execution is the same as that for the call statement execution. At this time, trace data is not written to the tracer.)
- If a statement other than a call statement is executed,
  the same processing as that resulting from execution of the Step command is performed.

The concept of Next step execution by the Next command is as shown in the figure below.

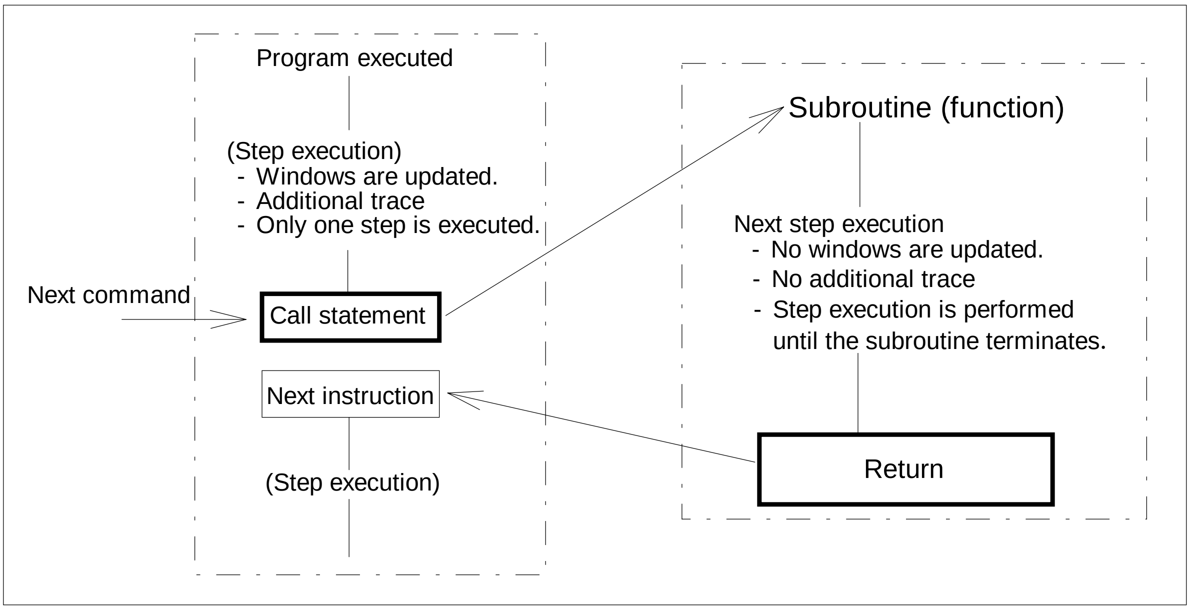


Figure 4-5. Concept of Next Step Execution

Slowmotion command

During step execution by the Slowmotion command,

(1) Starting from a specified address, step execution is performed line-by-line if the debug mode is source mode, or instruction-by-instruction if it is instruction mode.

(2) Each window is updated.

(3) (1) and (2) are repeated until the Stop command is issued.

4.2.6 Break Functions

The break functions are used to stop "user program execution (emulation)" by an emulation CPU and the "analyzers (tracer, timer, and coverage)".

There are basically six break functions:

- Event detection break
- Software break
- Temporary break
- Break caused by a condition being satisfied during step execution
- Forcible break
- Fail-safe break

The table below lists the relationship between these "break functions" and the "emulation execution functions".

	Event detection break	Temporary break	Software break	Break due to a step condition being satisfied	Forcible break	Fail-safe break
Real-time execution by Go command	○	×	○	×	○	○
Real-time execution by Go NoBreak command	×	×	×	×	○	○
Real-time execution by CPU Reset & Go command	○	×	○	×	○	○
Non-real-time execution by Step command	×	×	×	○	○	○
Non-real-time execution by Return command	×	×	×	○	○	○
Non-real-time execution by Next command	×	×	×	○	○	○
Non-real-time execution by Slowmotion command	×	×	×	×	○	○

○: Applied ×: Not applied

Event detection break

Event detection break enables the stopping of the execution of a user program upon the detection of a specified event condition.

This break is effective for the Go and CPU Reset & Go commands.

A break event must be set as an event detection condition by using the Event Set dialog box, Event Manager, or Break dialog box.

Temporary break

Temporary break enables the stopping of the execution of a user program upon the detection of a specified address. Once execution of the user program stops, the temporary break point is cleared.

Software break

Software break enables the stopping of the execution of a user program upon the detection of a specified address. Compared with event detection break or temporary break, which uses one hardware resource for each event condition, software break allows breakpoints to be set at multiple addresses for one event condition.

Since software break allows breakpoints to be set in multiple addresses with a single event detector, it performs the following processing before and after execution of a user program:

Immediately before execution:

(1) The instruction at the location where a software break is set is changed to a CALLT instruction immediately before execution of a user program.

(2) In the event detector, a break point is set in the vector address of the above CALLT instruction.

After a break:

(1) Correction of the analyzers (tracer and coverage) used when the CALLT instruction was executed

(2) The CALLT instruction is restored to the original instruction.

Since a software break uses a CALLT instruction as stated above, a vector table must be left open for the CALLT instruction. These settings are made from within the Extended Option dialog box.

The analyzers are corrected after the break. Therefore, the result of executing the CALLT instruction may not be displayed correctly in the tracer.

Break caused by a condition being satisfied during step execution

Break caused by a condition being satisfied during step execution enables the execution of a program to be stopped when the termination condition for a command (Step, Next, Return, and Slowmotion) is satisfied. The processing time is longer than that for real-time execution because execution, stop, and condition confirmation is performed for each instruction.

Forcible break

Forcible break enables the execution of a user program to be stopped forcibly. This break is effective for all commands used to execute a program.

There are two types of forcible break:

1. **Stop command**

   Forcibly stops the execution of a user program.

2. **Reset command**

   Forcibly stops the execution of a user program, then resets the devices.

The Stop command is useful for temporarily stopping a program. The Reset command is useful for executing a program from the beginning.

Fail-safe break

Fail-safe break enables the execution of a user program to be stopped forcibly if the program attempts to perform execution that is prohibited, using the memory and registers.

There are four types of fail-safe break:

1. **Non-map break**

   Generated when an attempt is made to access a non-mapping area.

2. **Write-protect break**

   Generated when an attempt is made to write to write-protected memory such as ROM.

3. **SFR illegal access break**

   Generated when an attempt is made to gain illegal access to the SFR area.

4. **Stack overflow break**

   Generated when an overflow to a specified stack area occurs.

A fail-safe break can be generated by either of two causes: an error in the user program or an environment setting error in the debugger.

Caution: If a program is written up to the vicinity of the boundary between the mapping and non-mapping areas, a non-map break may be generated.

More specifically, a non-map break may be generated if:

Maximum address in a mapping area - 5 ≤ Program address ≤ Maximum address in a mapping area

Example: For a mapping area of 0x00000 to 0x03FFF and a non-mapping area of 0x04000 and above, a non-map break may be generated if a program is written to addresses 0x3FFA to 0x3FFF.

If a non-map break is generated, it is related to prior fetch and a queue buffer.

4.2.7 Trace Functions

The trace functions allow a user to write data such as external sense clip values and data obtained by memory access during the execution of a user program to "trace memory" in real-time mode. By opening the Trace View window, the data written to trace memory can be referenced to check the progress of the target program.

The main functions of trace execution and view are as follows. Trace conditions can be set in the Trace dialog box. Trace data view can be specified in the Trace View window.

Trace operations:

- Operation during real-time execution
- Operation during step execution
- Operation during Next step execution

Trace condition setting function (Trace dialog box)

- Trace mode specification
- Point trace setting
- Section trace setting

Trace data view, format, and retrieval condition setting

- Trace data view specification
- Trace data retrieval condition setting

Relationship between trace execution and trace memory

Trace is divided into trace blocks according to the corresponding period:

<1> Block from real-time execution to a break caused by the occurrence of an event
<2> Block from emulation execution until the generation of a fail-safe break
<3> Block from emulation execution until application of a forcible break
<4> Block of step execution groups which are consecutive as viewed from the program

Trace memory is a ring buffer having 32K frames. If trace exceeds 32K frames, the oldest frame is overwritten with the newest trace data.

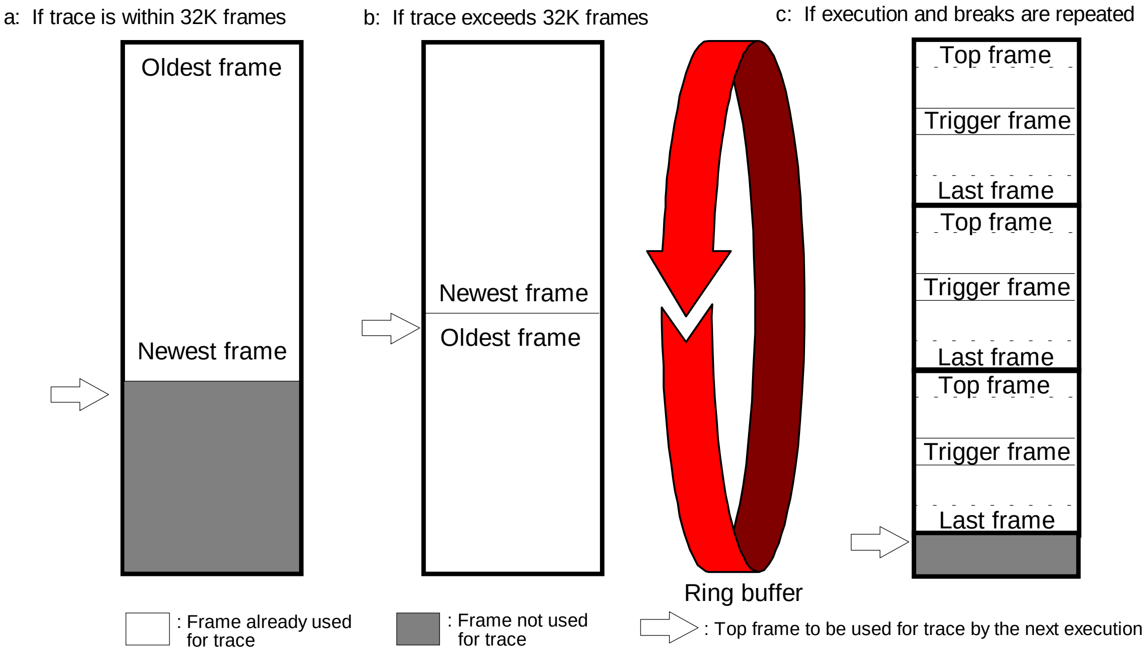

a: If trace is within 32K frames
Oldest frame
Newest frame
b: If trace exceeds 32K frames
Newest frame
Oldest frame
Ring buffer
c: If execution and breaks are repeated
Top frame
Trigger frame
Last frame
Top frame
Trigger frame
Last frame
Top frame
Trigger frame
Last frame
: Frame already used for trace
: Frame not used for trace
: Top frame to be used for trace by the next execution

Figure 4-6. Concept of Trace Memory

Trace operation

The operation of the tracer varies with the execution mode:

Operation during real-time execution

The tracer starts tracing with real-time execution specified and stops tracing when the event condition (including a delay condition), specified as a break condition from within the Trace dialog box, is satisfied.

Operation during step execution

The tracer operates upon the execution of each step, trace information for each step being added to the tracer every time a step is executed.

Operation during Next step execution

If an instruction to be executed is other than a call instruction (CALL, CALLF, CALLT, BRK, or BRKCS)

The tracer performs the same processing as that performed during step execution.

If the instruction to be executed is a call instruction (CALL, CALLF, CALLT, BRK, or BRKCS)

The tracer traces the call instruction only.

The tracer does not trace the function to be called.

The tracer resumes tracing after control is returned by the return instruction (RET, RETI, BRTB, RETCS, or RETCSB) from the function

Trace condition setting function (Trace dialog box)

The following specifications enable the user to specify the conditions for tracing. If these specifications are omitted, “all trace” is assumed, i.e., trace information is recorded for each instruction of a user program.

Trace mode specification

Specifies whether to execute all trace or conditional trace. There are two types of conditional trace: Point trace and section trace. The trace mode specified with this setting becomes valid.

Point trace setting

Qualified trace involves tracing only when a previously specified address is executed or accessed. The condition to be specified can be created in the Event dialog box.

Section trace setting

In section trace, tracing begins when a previously specified trace start condition is satisfied and stops when a stop condition is satisfied. In other words, it is a range-specified trace. The condition to be specified can be created in the Event dialog box.

4.2.8 Snapshot Function

The snapshot function is used to interrupt real-time execution and output specified information to trace memory as snap data upon the detection of a specified snap event during real-time execution. Once the output of information to trace memory has been completed, the function resumes real-time execution.

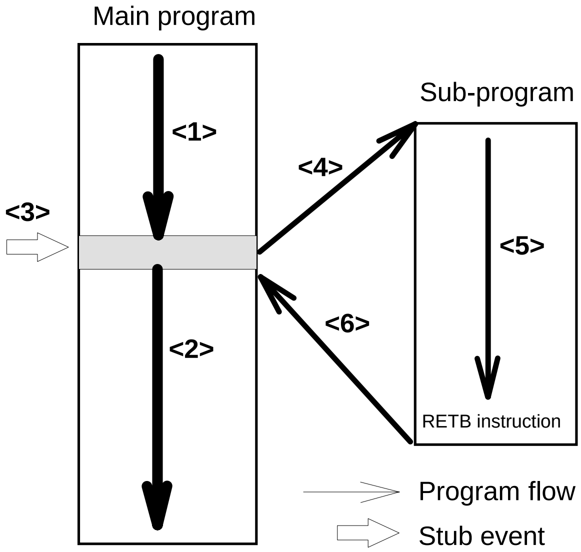
The following can be specified as snap data:

- Data in a general-purpose register
- Data in an SFR
- Data in memory

Snap data output can be specified for a maximum of sixteen points.

4.2.9 Stub Function

The stub function is used to insert a simple program into the user program currently being checked. When a stub event is generated, the user program stops and another user-supplied program is executed. A RETB instruction must be specified at the end of the sub-program to be executed when a stub event is generated. Failing to specify this instruction can cause a malfunction.



The stub function enables a user to easily insert a program.

When the stub function is not used
<1> is executed.
<2> is executed.

When the stub function is used
<1> is executed.
A stub event is generated at **<3>** and the main program breaks.
In **<4>**, IE pushes the current address into the stack and overwrites the PC with the entry address of the sub-program.
<5> is executed.
In **<6>**, control is returned to the main program by execution of the RETB instruction which appears at the end of the sub-program.
<2> is executed.

Figure 4-7. Concept of Stub Function

4.2.10 Event Setting and Detection Functions

The event setting and detection functions are used to set the conditions for stopping "user program execution" by an emulation CPU and for starting and stopping "trace operation" and "timer measurement" by the analyzers.

The following event condition setting and detection functions are provided:

Event detection condition setting functions

Bus event condition setting function

Execution event condition setting function

Event condition link setting function

Event detection condition integration functions

Break event setting

Trace event setting

Timer event setting

Event condition setting function

The event condition setting function is used to set, in the event condition register, the conditions for stopping "user program execution" by an emulation CPU and for starting and stopping "trace operation" and "timer measurement" by the analyzers. The event detection condition specified by the event detection condition setting function (Event Set dialog box and Event Link dialog box) does not become effective until set in the event mode register by the event detection condition integration function (Event Manager window, Break dialog box, Trace dialog box, Snap Shot dialog box, Stub dialog box, and Timer dialog box).

Three functions are provided for setting the event detection conditions:

Bus event condition setting function

When the user program accesses specified memory or data is input to an external sense clip, it is possible to set this as an event detection condition in the bus event condition register.

(a) Bus event condition register

Using the Event Set dialog box, up to seven conditions can be set in the bus event condition register (BRA).

(b) Event conditions

The following items can be set as event detection conditions:

Item	Status	Explanation
Address	Address	Address (address range)
	Mask	Address mask
Status	Fetch	Program fetch
	Program Read	Read by a program
	Program Write	Write by a program
	Program R/W	Read/write by a program
	Macro Read	Read by a macro service
	Macro Write	Write by a macro service
	Macro Read/Write	Read/write by a macro service
	Program/Macro Read	All read
	Program/Macro Write	All write
	Program/Macro R/W	All read/write
Data	Data	Data value
	Mask	Data mask value
External	External	External sense data value
	Mask	External sense data mask value
Pass count	Pass count	Pass count value

Caution: When Fetch is selected as a Status condition, an event is generated with the execution of a Fetch operation. That is, an event is generated prior to the fetched program being executed. To generate an event with the actual execution of the fetched program, use the execution event condition setting function.

Execution event condition setting function

When the user program attempts to execute the instruction at a specified address and data is simultaneously input to an external sense clip, it is possible to set this as an event detection condition in the execution event condition register.

(a) Execution event condition register

Using the Event Set dialog box, up to three conditions can be set in the execution event detection register (BRS).

(b) Event conditions

The following items can be set as event detection conditions:

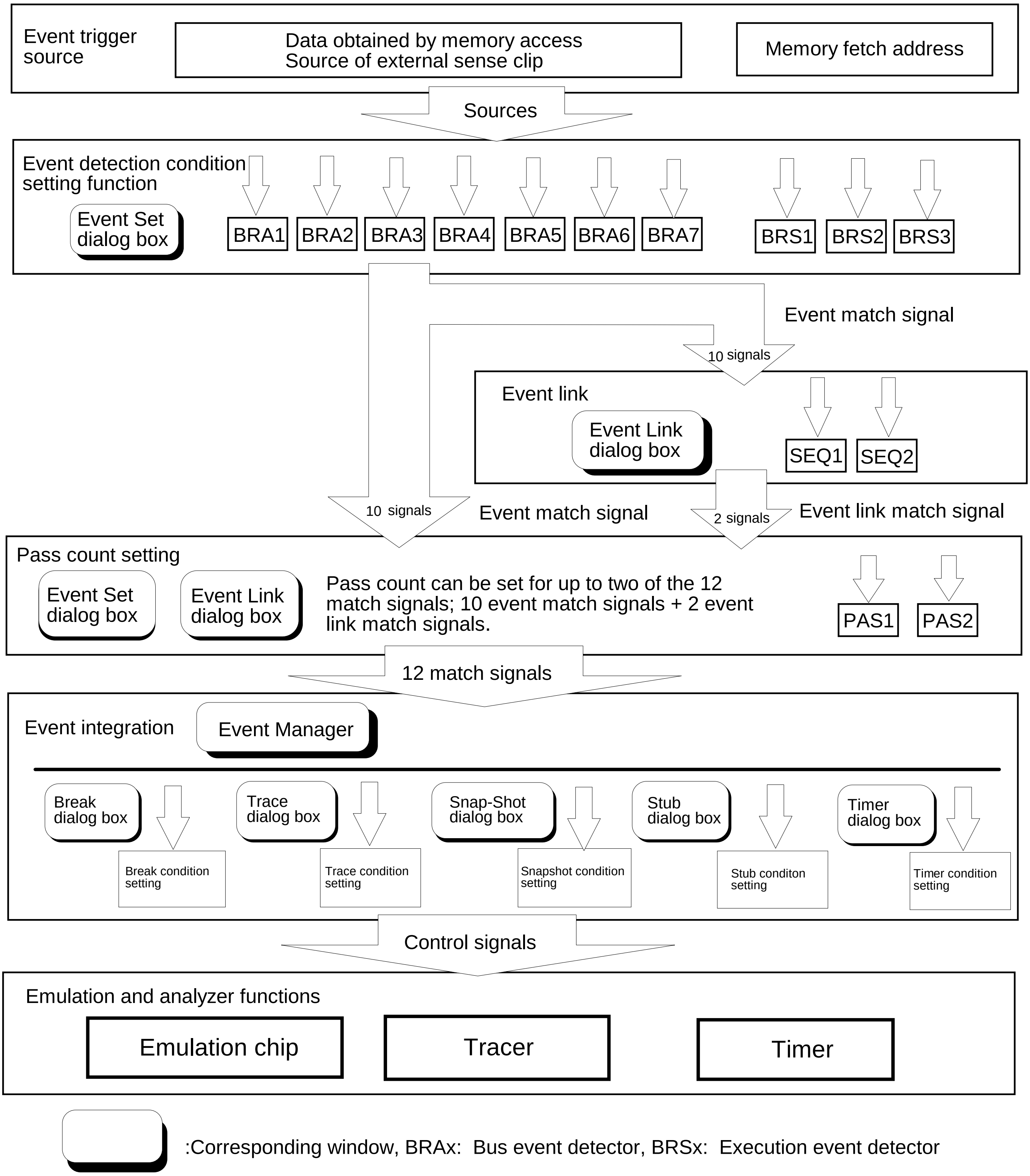
Item	Status	Explanation
Address	Address	Address (address range)
	Mask	Address mask
Status	Run	Program execution
Data	Data	Data value
	Mask	Data mask value
External	External	External sense data value
	Mask	External sense data mask value
Pass count	Pass count	Pass count value

Event condition link setting function

In the Event Link dialog box, an event link condition can be registered for the event condition registered in the Event Set dialog box.

Concept of event detection

The figure below illustrates the concept of event detection starting from event condition setting.



SEQx: Event link detector, PASx: Pass counter

Figure 4-8. Concept of Event Detection

4.2.11 Register Operation Functions

The register operation functions are used to view and modify the general-purpose registers and SFRs. The main functions are explained below.

(1) General-purpose register operation function (Register window)

This function is used to view and modify the control registers and general-purpose registers.

• Control registers:	PC, SP, PSW
• General-purpose registers:	RP0, RP1, RP2, RP3, RG4, RG5, RG6, RG7, AX, BC, VVP, UUP TDE, WHL

PSW can also be viewed and modified using the PSW flag names given below:

• PSW flag names:	UF, RSB, S, Z, RSS, AC, IE, P/V, CY

(2) Special function register operation function (SFR window)

This function is used to view and modify the special function registers (SFRs).

Bit operation is possible for SFRs.

4.2.12 Memory Operation Functions

The memory operation functions are used to view and modify memory using mnemonic codes, hexadecimal codes, and ASCII characters. These functions can be used in the Assemble and Memory windows.

4.2.13 Save Function

The save function is used to store, to a file, the object code and debugging environment on the in-circuit emulator, via the disk device connected to the host machine.

4.2.14 Time Measurement Function

This function is used to measure the total execution time from the start of execution to a break and the time from one event to another.

Time can be measured at up to three locations. The following table lists the specifications of the timers.

Item	Explanation
Number of timers	Three independent timers
Maximum measurable time	Approximately 14 minutes and 18 seconds
Minimum resolution	200 ns

4.2.15 Real-Time RAM Sampling

The contents of memory space that can be read can be displayed and updated in real time, even during user program execution. This function is called real-time RAM sampling, and such a memory space is called a real-time RAM space. The contents of this space can also be displayed in the Realtime RAM Tracer window. The Location command maps the real-time RAM space, as follows:

Location command	Real-time RAM space
Location 00H	0x00fd00-0x00fe00
Location 0fH	0x0ffd00-0x0ffe00

4.2.16 Window Linkage Function

The widow linkage function is used to link any windows. When the current position is moved in one of the linked windows, the current position in the other windows is also moved accordingly. By this function, the following windows can be mutually linked:

- Source window
- Memory window
- Coverage window
- Disassemble window

The above windows can also be linked from the Trace View window. The Trace View window, however, cannot be linked from any of the above windows.

4.2.17 Source Debugging

ID78K4 can be used to debug not only object programs but also source programs. The debugging of a source program is referred to as source debugging.

The debugging of a source program offers the following advantages over the debugging of an object program:

- Debugging can be carried out while viewing the source, created by the editor in C or structured assembler.
- Breakpoints can be set and step execution can be performed for the source.

To set a breakpoint, for example, the actual address of the breakpoint must normally be set. In source debugging, on the other hand, a breakpoint can be set by specifying the position in the source program where the breakpoint is desired, using the mouse. Also, in source debugging, the line of the source program that is currently being executed is indicated by a '>' during step execution. This allows the user to comprehend the operation of the program more accurately.

Source debugging is particularly useful when debugging a program written in C or structured assembler.

Note the following when performing source debugging:

(1) Before assembling or compiling a source program, an appropriate option must be specified so that the object program includes source debugging information.

Type of source program to be debugged	Necessary action
C program	Specify the -G option before compiling.
Structured assembler program	Specify the -GS option before structured assembling.
Assembler program	Specify the -GA option before assembling.
Link	Specify the -G option before linking.

(2) Information on the path of the source program must be specified in the Source Path dialog box.

(3) To perform source debugging, the load module file created by the linker must always be loaded. Source debugging is not possible when the object file created by the object converter is loaded.

Chapter 5 Functions of the Debug Windows

5.1 Basic Window Operations

The basic operations in a debugging process using the graphical user interface (GUI) are specified in noun + verb format. In other words, the user first selects a noun, namely the object to be debugged (such as a variable, line, or task), then selects a verb, namely the desired function button (such as Write, OK, or Cancel).

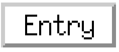
The following explains the objects that the user can manipulate when using ID78K4.

(1) Mouse

Most mouse-based operations with ID78K4 are done using the left mouse button. In the following explanations, therefore, all references to a mouse button refer to the left mouse button, unless otherwise specified. (When the user changes the file to set left-handed mode, however, all references refer to the right mouse button.)
There are four basic mouse operations:

Click: Pressing the mouse button then releasing it.
Double-click: Pressing the mouse button twice, in rapid succession.
Drag: Moving the mouse while holding down the mouse button.
Drag & drop: Moving the mouse while holding down the mouse button, then releasing the button to reposition a desired object (normally, using the center mouse button).

(2) Function button



A function button is a rectangular, sculptured button on which a character string or bit map is displayed. Clicking a function button instigates the corresponding action.

(3) Toggle button

A toggle button is a square button on which nothing is displayed. Instead, the meaning of a toggle button is displayed to the right of the button. A toggle button can either be selected or deselected. Clicking a toggle button selects the corresponding item. Clicking it again deselects the item.

(4) Radio button

◆

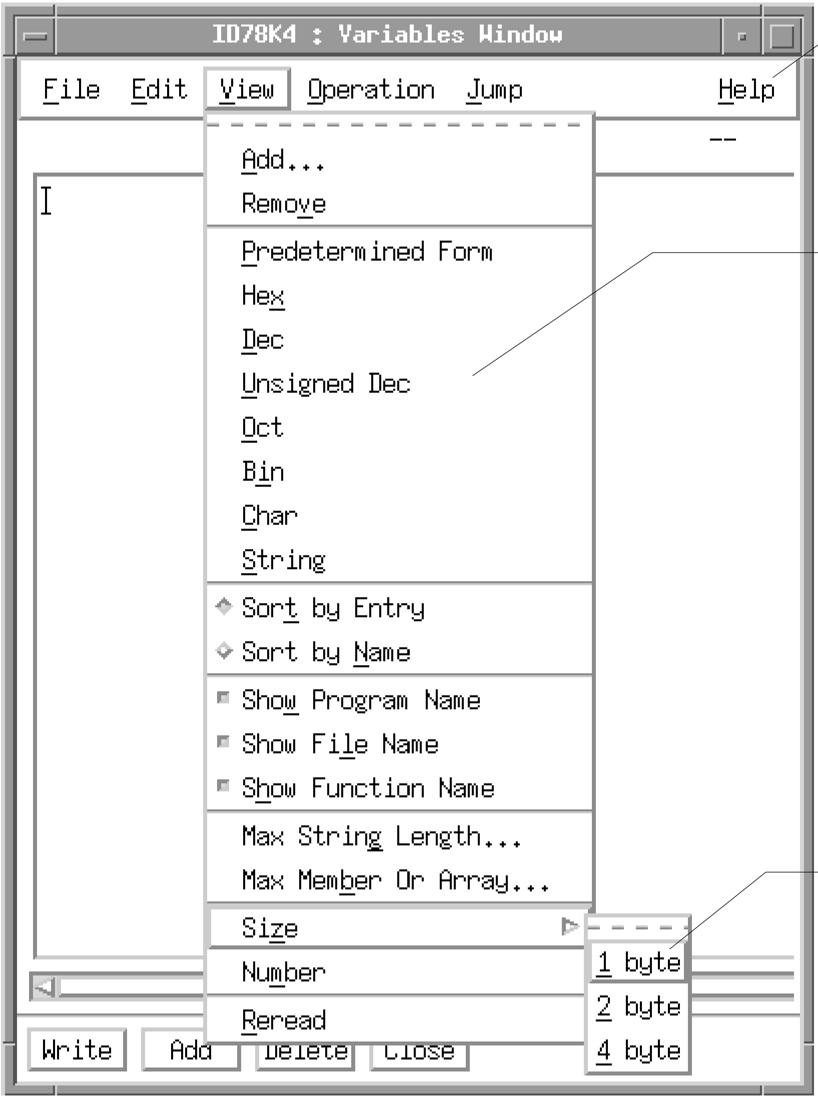
A radio button is a rhombic button on which nothing is displayed. Instead, the meaning of a radio button is displayed to the right of the button. Radio buttons are always displayed as a group of two or more, from which only one can be selected.

A radio button can be either selected or deselected. Clicking a radio button selects the corresponding item. Clicking it again deselects the item.

(5) Menu bar

A menu bar is a rectangle on which the titles of available pull-down menus are displayed. Clicking a title within the menu bar displays the corresponding pull-down menu. The user can also enter, from the keyboard, the alphabetic character corresponding to a title while holding down the META key. This displays the corresponding pull-down menu.

Figure 5-1. Pull-Down Menu



(5) Menu bar

(6) Pull-down menu

(7) Cascade menu

(6) Pull-down menu

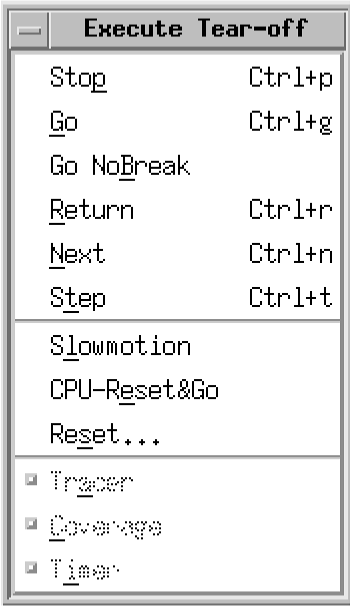
Clicking a "title" within a menu bar displays the corresponding pull-down menu. The user can click the dotted line at the top of a pull-down menu to "tear off" the menu items (display them separately from the window). (Figure 5-2 shows an example of a tear-off window.)

Clicking an item among those listed in the window instigates the corresponding action. The user can also enter, from the keyboard, the alphabetic character corresponding to the menu to instigate the action corresponding to the item. When an accelerator, such as "Ctrl+a", is displayed to the right of the menu, the user can enter the accelerator from the keyboard, without displaying the menu, to instigate the corresponding action.

For pull-down menus, menu items are indicated as follows, such that the effect of selecting a menu item can easily be recognized.

- "Item"
  When this item is selected, the indicated action is instigated.
- "Item..."
  When this item is selected, the corresponding dialog box appears.
- "Item ∇"
  When this item is selected, the corresponding cascade menu appears.

Figure 5-2. Tear-Off Window



(7) Cascade menu

Clicking an "item ∇" in a pull-down menu displays the corresponding cascade menu. Clicking an item among those listed in the window instigates the corresponding action.

The user can also enter, from the keyboard, the alphabetic character corresponding to the menu to instigate the corresponding action.

(8) Option button

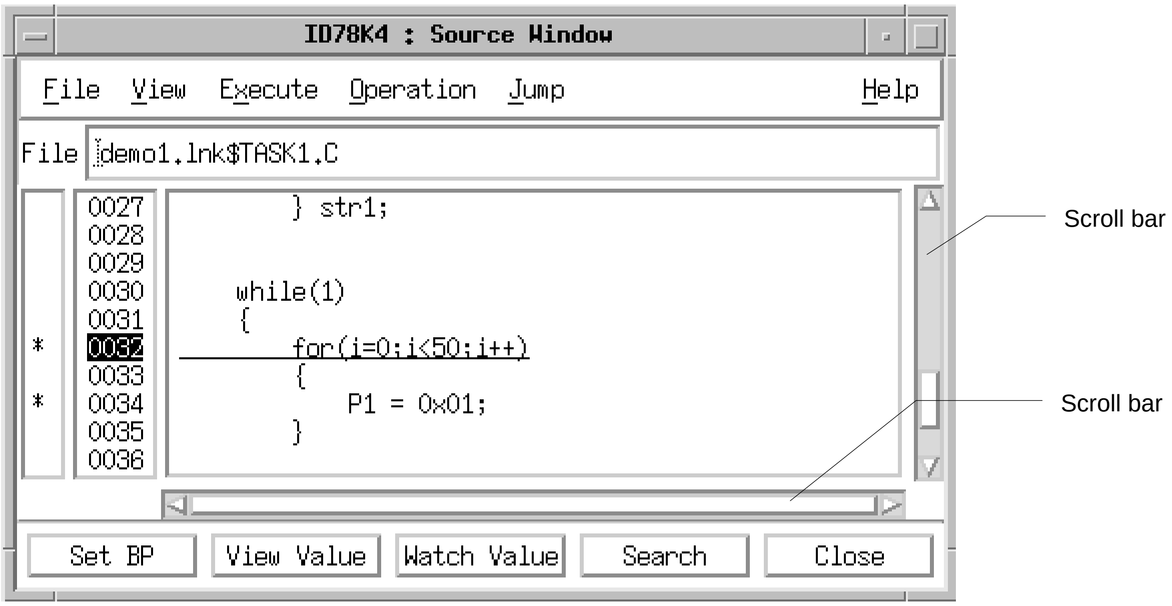
An option button is a rectangular button on which a character string is displayed. Clicking an option button displays the corresponding option menu.

The character string displayed on an option button corresponds to the item selected from the corresponding option menu.

(9) Option menu

An option menu has the same function as a pull-down menu, with the exception that it is displayed by clicking the corresponding option button.

(10) Scroll bar



File demo1.lnk$TASK1.C

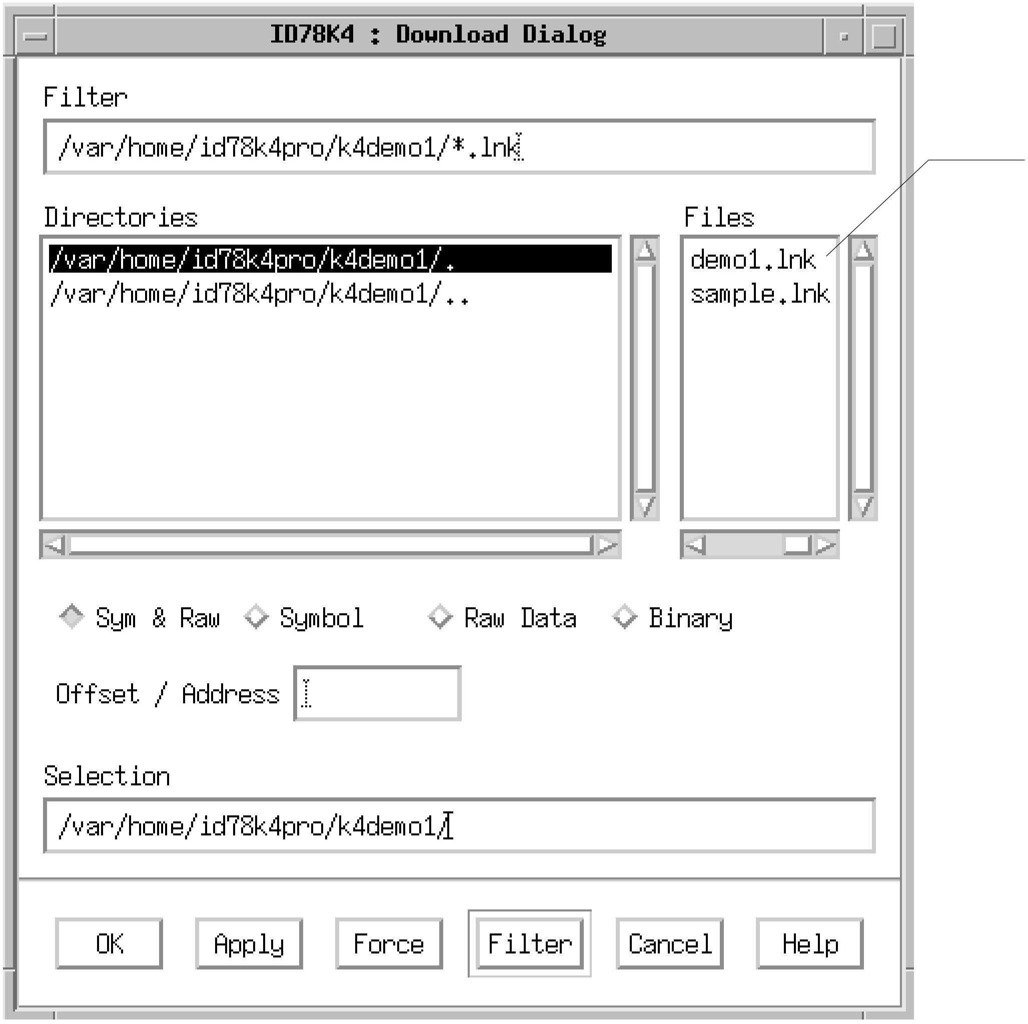
```
        } str1;

    while(1)
    {
*       for(i=0;i<50;i++)
        {
*           P1 = 0x01;
        }
```

A scroll bar consists of a slider and scroll buttons. The slider in a scroll bar indicates the position of the displayed portion relative to the entire scrollable information. Dragging the slider within the scroll bar scrolls the screen.

Clicking the scroll arrow (scroll button) at each end of a scroll bar scrolls the screen by one line or column in the direction of the arrow.

(11) Selection list



Filter: /var/home/id78k4pro/k4demo1/*.lnk

Directories:
- /var/home/id78k4pro/k4demo1/.
- /var/home/id78k4pro/k4demo1/..

Files:
- demo1.lnk
- sample.lnk

● Sym & Raw ○ Symbol ○ Raw Data ○ Binary

Selection: /var/home/id78k4pro/k4demo1/

Selection list

A selection list is an area in which a list of selectable data is displayed. Clicking a displayed data item selects it. When all the data items constituting a list cannot be displayed within the window at one time, the user can manipulate a scroll bar to view them.

5.2 Key-Based Window Operations

With a window-based interface, the user executes each function by manipulating the mouse. The user can also execute each function from the keyboard.

Table 5-1 lists the functions of those keys used for most of the keyboard-based operations performed with windows. For the functions of other keys, see the "OSF/MotifTM User's Guide Revision 1.2."

Table 5-1. Key Functions

Key	Function
Tab or Ctrl + Tab	Moves the cursor to the next area.
Shift + Tab or Ctrl + Shift + Tab	Moves the cursor to the previous area.
Arrow key	Moves the cursor within the current area.
Space key	Selects the item.
Return	Executes the buttons selected when the window is displayed.
META key + alphabetic character corresponding to a title within a menu bar	Displays the corresponding pull-down menu.
Alphabetic character corresponding to a pull-down or cascade menu	Executes the corresponding item in the menu.
Accelerator to the right of a pull-down menu	Executes the corresponding item without displaying the menu.

5.3 Key-Based Dialog Box Operations

Table 5-2 lists the functions of those keys used for entering text in dialog boxes.

Table 5-2. Key Operations (Dialog Boxes)

Key	Function
Ctlr+a	Moves the cursor to the beginning of the line.
Ctlr+b	Moves the cursor one character position to the right.
Ctlr+d	Deletes the character at the cursor position.
Ctlr+e	Moves the cursor to the end of the line.
Ctlr+f	Moves the cursor one character position to the left.
Ctlr+h	Deletes the character to the left of the cursor.
Ctlr+k	Deletes the character string starting from the cursor position to the end of the line
Ctlr+m	Decision

5.4 Active State and Hold State

While a window is in the active state, the contents of the window change as execution of the target program progresses. While a window is in the hold state, its contents do not change (remain as is) regardless of whether the target program is being executed.

With ID78K4, a window whose contents change as the target program is being executed (view window and view/setting window) can be used in either the active state or the hold state.

With ID78K4, only one window of each type can be set to the active state. When a window is switched from the hold state to the active state, the window which was in the active state is automatically switched to the hold state.

The background of a window in the hold state is displayed in reverse video.

5.5 Mark Point Area

A mark point area is used to display executable lines and break positions. The Source window and Disassemble window each feature a mark point area.

An asterisk (*) is displayed for each executable line (only for the Source window).

By default, "B" is displayed for each break position.

To change the mark, make the necessary settings using the resource file (XId78k4.ad) before starting ID78K4.

An example of changing the mark is shown below. In this example, "B" is changed to "H".

Resource file (XId78k4.ad) before change

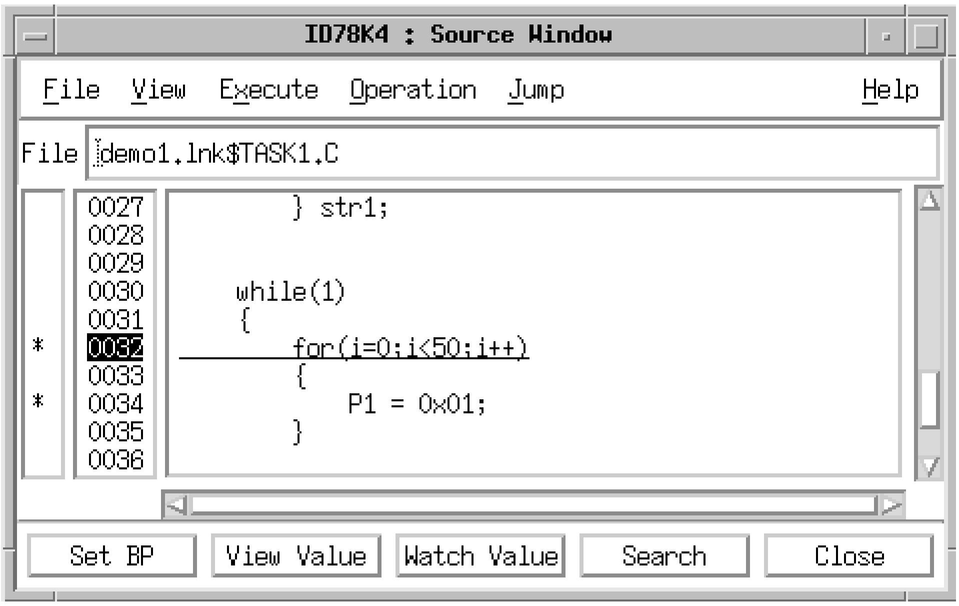
```
:
XId78k4*MarkBreakHard: B
:
```

Resource file (XId78k4.ad) after change

```
:
XId78k4*MarkBreakHard: H
:
```

Figure 5-1 shows the mark point area in the Source window.

Figure 5-1. Mark Point Area



demo1.lnk$TASK1.C

```
        } str1;


    while(1)
    {
*       for(i=0;i<50;i++)
        {
*           P1 = 0x01;
        }

```

The mark point area displays the break positions set using the Break dialog box. The user can click a desired line in the mark point area to set a break.

Setting/resetting a break:

In the mark point area, clicking a line sets a break. Clicking the same line again clears the break. In the Source window, the user can click only those lines against which "*" appears.

When a break is set, "B" is displayed in the mark point area by default.

Each time a break is set in the mark point area, the related windows (Break dialog box and Disassemble window) are modified accordingly.

5.6 Indication of Execution Positions

In the Source window and Disassemble window, when execution of the target program stops, the current PC line indicating the execution stop position is underlined.

In a window that is in the active state, the execution stop position is underlined when displayed in the window. When the execution stop position is not displayed in the window, it is underlined after the user scrolls the screen to the execution stop position.

In those windows that are in the hold state, the execution stop position is not underlined.

5.7 Errors/Warnings

5.7.1 GUI Operational Errors/Warnings

All GUI operational errors are handled as warnings.

If a warning occurs, the system beeps or displays the Warning dialog box.

5.7.2 Errors/Warnings Generated by the Debugger

If an error or warning occurs, the Error dialog box or Warning dialog box appears.

Chapter 6 Debug Windows

6.1 Window Types and Configurations

ID78K4 supports four types of windows and three types of dialog boxes.

Dialog boxes are displayed only temporarily to enable a specified operation. Both windows and dialog boxes can be modified by using the Window Manager. In principle, however, while windows can be reduced to an icon, dialog boxes cannot. Dialog boxes can be closed by pressing the ESC key.

The configuration of each window and dialog box is described below.

6.1.1 Execution Window

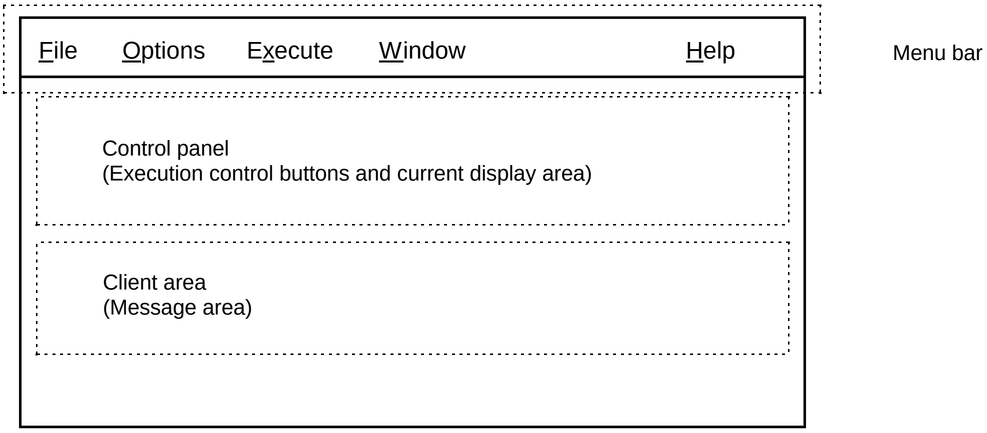
The execution window is used to control or record the execution of the target program. It also displays the current position and can be used to open other windows.

The execution window is explained below:

- Main window

Figure 6-1 shows the configuration of the execution window (top, middle, and bottom of the window).

Figure 6-1. Configuration of the Execution Window



(a) Top of window

Menu bar

(b) Middle of window

Control panel which consists of execution control buttons and current display area

(c) Bottom of window

Client area which consists of a message area

6.1.2 Auxiliary Window

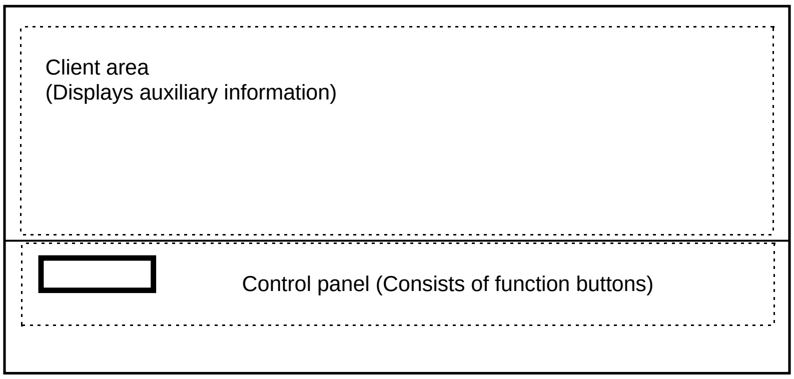
The auxiliary window is used to open other debugger windows or execute debugging commands.

The auxiliary window is explained below:

- Windows Box window

Figure 6-2 shows the configuration of the auxiliary window (top and bottom of the window).

Figure 6-2. Configuration of the Auxiliary Window


Client area
(Displays auxiliary information)
Control panel (Consists of function buttons)

(a) Top of window

Client area which displays auxiliary information

(b) Bottom of window

Control panel which contains function buttons

6.1.3 View Windows

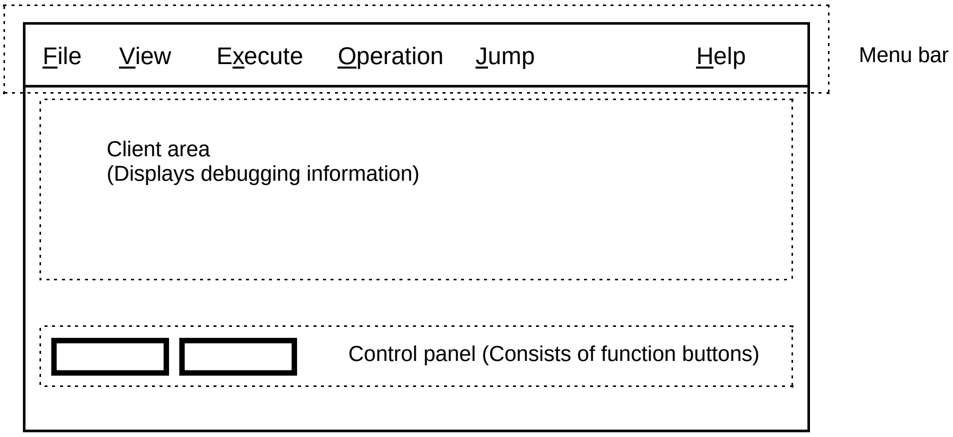
The view windows are used to display debugging information.

The view windows are explained below:

- Source window
- Stack Trace window
- Realtime RAM Tracer window
- Trace View window
- Timer window

Figure 6-3 shows the configuration of a view window (top, middle, and bottom of the window).

Figure 6-3. Configuration of a View Window


File
View
Execute
Operation
Jump
Help
Menu bar
Client area
(Displays debugging information)
Control panel (Consists of function buttons)

(a) Top of window

Menu bar

(b) Middle of window

Client area which displays debugging information

(c) Bottom of window

Control panel which consists of function buttons

6.1.4 View/Setting Windows

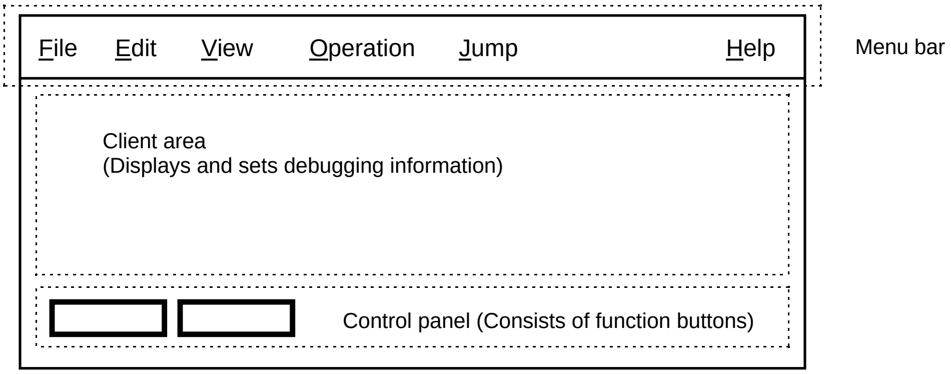
The view/setting windows are used to display debugging information and set values.

The view/setting windows are explained below:

- Variables window
- Local Variables window
- Registers window
- SFR window
- Memory window
- Coverage window
- Disassemble window
- Event Manager window

Figure 6-4 shows the configuration of a view/setting window (top, middle, and bottom of the window).

Figure 6-4. Configuration of a View/Setting Window


File
Edit
View
Operation
Jump
Help
Menu bar
Client area
(Displays and sets debugging information)
Control panel (Consists of function buttons)

(a) Top of window

Menu bar

(b) Middle of window

Client area which displays debugging information and provides a setting area

(c) Bottom of window

Control panel which consists of function buttons

6.1.5 Selection Dialog Boxes

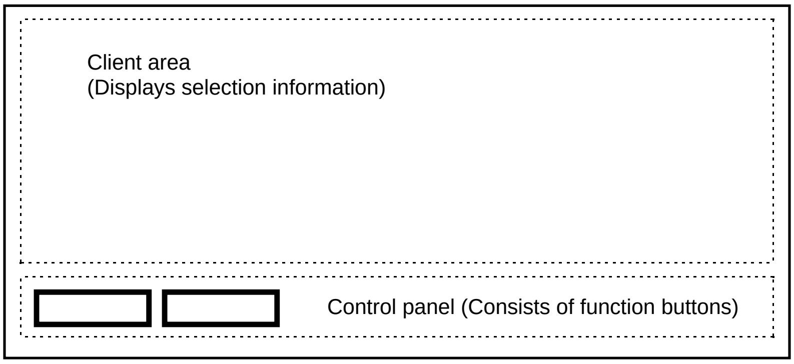
The selection dialog boxes are used to select a debugging environment or file.

The selection dialog boxes are explained below:

- Configuration dialog box
- Option dialog box
- Download dialog box
- Upload dialog box
- File Selection dialog box
- Selection dialog box
- Items Selection dialog box
- Source File Select dialog box
- Result dialog box
- Verify dialog box
- Coverage items selection dialog box
- Event dialog box
- Break dialog box
- Trace dialog box
- Trace Search dialog box
- Input dialog box
- Snap Shot dialog box
- Stub dialog box
- Timer dialog box

Figure 6-5 shows the configuration of a selection dialog box (top and bottom of the dialog box).

Figure 6-5. Configuration of a Selection Dialog Box


Client area
(Displays selection information)
Control panel (Consists of function buttons)

(a) Top of dialog box

Client area which displays selection information

(b) Bottom of dialog box

Control panel which consists of function buttons

6.1.6 Confirmation Dialog Boxes

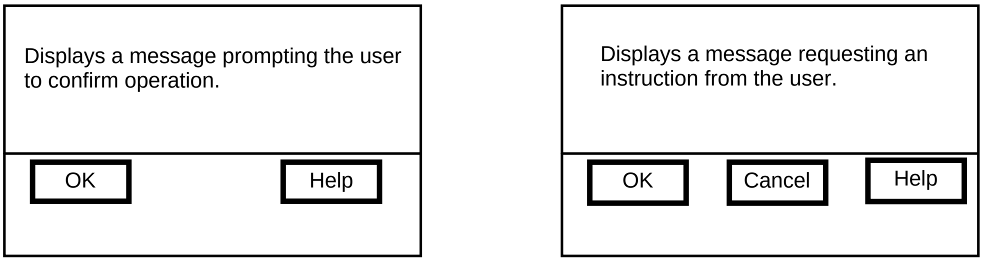
The confirmation dialog boxes are used to display messages (help information, errors, warnings, and termination confirmation) and prompt the user to confirm operation.

The confirmation dialog boxes are explained below:

- Question dialog box
- Message dialog box
- Verify result dialog box
- Coverage result dialog box
- Error dialog box
- Warning dialog
- Working dialog box
- Help dialog box

Figure 6-6 shows the configuration of a confirmation dialog box (top and bottom of the dialog box):

Figure 6-6. Configuration of a Confirmation Dialog Box



Displays a message prompting the user to confirm operation.

Displays a message requesting an instruction from the user.

(a) Top of window

Displays a message which prompts the user to confirm the operation or request an instruction from the user.

(b) Bottom of window

Control panel which consists of function buttons

6.1.7 Auxiliary Dialog Boxes

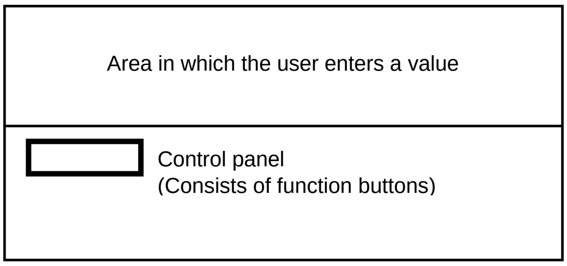
The auxiliary dialog boxes are used to enter a value by using the keyboard, to open a window or perform operation with a window.

The auxiliary dialog boxes are explained below:

- Add Variable dialog box
- Memory dialog box
- Prompt dialog box

Figure 6-7 shows the configuration of an auxiliary dialog box (top and bottom of the dialog box):

Figure 6-7. Configuration of an Auxiliary Dialog Box


Area in which the user enters a value
Control panel
(Consists of function buttons)

(a) Top of window

Area in which the user enters a value

(b) Bottom of window

Control panel which consists of function buttons

6.2 Debug Windows

Debug windows are listed on the subsequent pages.

Window list (in the order in which they appear in this manual)

Window name	Outline	Page
Trace Search dialog box	Searches for trace data.	260
Timer window	Displays the results of timer measurement.	263
Windows Box window	Displays buttons used to open windows.	267
Error dialog box	Displays an error message.	270
Warning dialog box	Displays a warning message.	270
Help dialog box	Displays help information.	271

Window list (in alphabetical order)

Window name	Outline	Page
Add Variable dialog box	Adds a variable to be monitored.	141
Break dialog box	Sets, registers, cancels, or deletes a break condition.	229
Configuration dialog box	Displays/sets the target environment.	87
Coverage Items Selection dialog box	Sets the coverage efficiency conditions.	197
Coverage Result dialog box	Displays the coverage efficiency in the result dialog box.	195
Coverage window	Displays/initializes the results of coverage measurement.	188
Disassemble window	Displays the disassembled text of a program.	201
Download dialog box	Selects and downloads a load module file.	96
Error dialog box	Displays an error message.	270
Event Manager window	Displays events.	213
Event dialog box	Sets/modifies events.	222
File Selection dialog box	Selects a file to be read or written.	104
Help dialog box	Displays help information.	271
Input dialog box	Specifies the address range for coverage memory clear.	199
Items Selection dialog box	Selects registers and SFRs to be displayed.	111
Local Variables window	Displays/sets the values of the local variables in the current scope.	144
Main window	Appears immediately after the debugger is started.	74
Mark dialog box	Displays/sets a mark point.	129
Memory dialog box	Compares, copies, fills, and searches the contents of memory.	179
Memory window	Displays/sets the contents of memory.	170
Message dialog box	Temporarily displays the debugger version or the value of a variable.	131
Option dialog box	Sets options.	94
Prompt dialog box	Prompts the user to enter an address or variable from the keyboard.	115
Question dialog box	Prompts the user to confirm debugger termination.	113
Realtime RAM Tracer window	Displays the contents of RAM in real time.	184
Registers window	Displays/sets the values of registers.	155
Result dialog box	Displays the results of memory comparison.	182

Window name	Outline	Page
SFR window	Displays the values of SFRs.	163
Selection dialog box	Specifies initialization of the target or debugger.	108
Snap Shot dialog box	Sets a snapshot.	242
Source File Select dialog box	Selects a source file to be displayed in the Source window.	117
Source window	Displays a source text.	120
Stack Trace window	Displays the contents of the stack.	151
Stub dialog box	Sets a stub.	246
Timer dialog box	Sets a timer.	249
Timer window	Displays the results of timer measurement.	263
Trace dialog box	Sets, registers, cancels, or deletes a trace condition.	233
Trace Search dialog box	Searches for trace data.	260
Trace View window	Displays the results of trace.	252
Upload dialog box	Selects and uploads a file.	100
Variables window	Sets/modifies the value of a specified variable.	133
Warning dialog box	Displays a warning message.	270
Windows Box window	Displays buttons used to open windows.	267

6.3 Details of Debug Windows

This section describes the functions of each window.

window-name	**window-type**

Outline

Provides a brief outline of the window.

Window

Displays the configuration of the window as it appears on the screen.

Description

Describes the functions and usage of the window.

Buttons

Describes each function button displayed in the window.

If an item must be selected before a button can be clicked, it is described at the left end of an arrow, preceding the description of that button.

Menu bar

Lists the menus that can be pulled down from the menu bar, and describes the functions of each menu item.

Notes

Describes notes on using the window, if any.

Errors

Describes errors which may occur while using the window, if any.

Main window

Execution window

Outline

The Main window is automatically opened immediately after the user has activated the debugger and completed the initial settings. The window remains displayed until the debugger is terminated. Other windows are opened from within this window.

The execution of a user program is controlled by using this window at the following two levels:

(1) Source level

When a program is debugged in units of source text lines.

(2) Instruction level

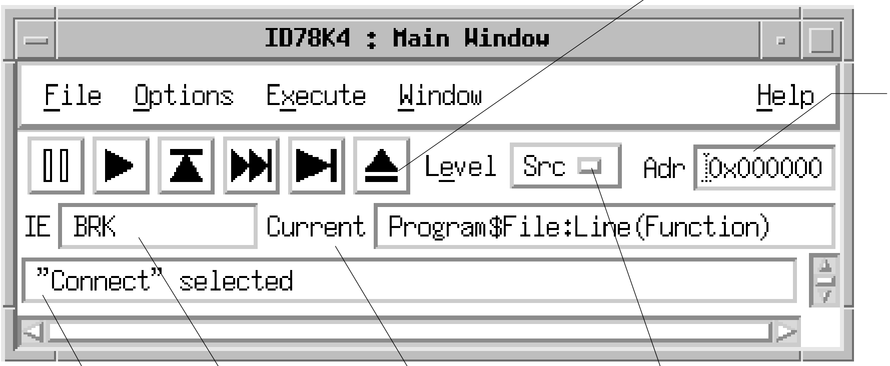
When a program is debugged in units of instructions.

When the debugger is started, source level debugging is selected by default. If it is started with the -insmode option specified, however, the instruction level is selected.

The display of the Main window can be switched between "small view" and "large view." In both sizes, the same display is provided. In large view, however, the same buttons as those in the Windows Box window are displayed. When the debugger is started, small view is selected by default. The display size can be switched using the Options menu in the menu bar.

Window

(1) Small view


ID78K4 : Main Window
File
Options
Execute
Window
Help
Level
Src
Adr
IE
Current
(1) Execution control buttons
Current address
(2) Debug mode selection option button
(4) Current position display area
(3) IE status display area
(5) Message display area

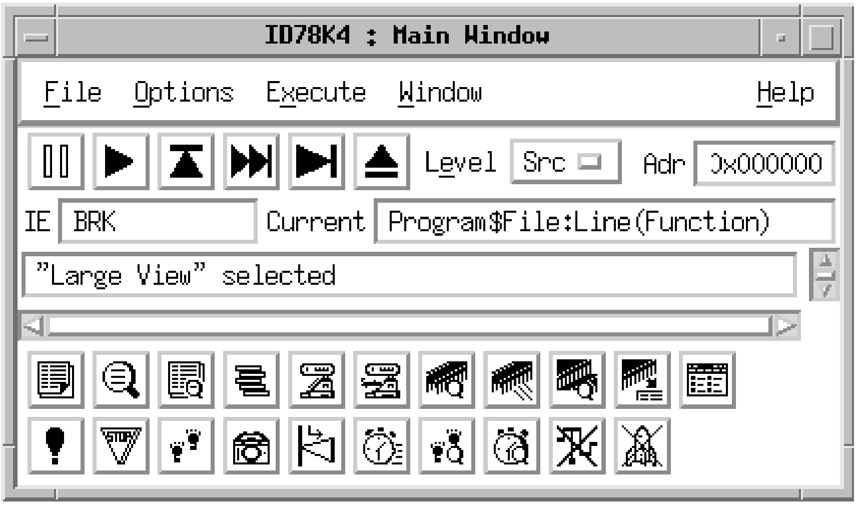
Adr: 0x000000

IE: BRK

Current: Program$File:Line(Function)

"Connect" selected

(2) Large view



Level Src

Adr 0x000000

IE BRK

Current Program$File:Line(Function)

"Large View" selected

Description

The Main window consists of the following components:

Menu bar

Top of client area: Execution control buttons

Debug mode selection button

Message area (displays the current position)

Middle of client area: Message area (displays the window operation)

The message area which displays the operation performed with the window has scroll bars along its right and bottom edges.

Bottom of client area: Same buttons as those in the Windows Box window (displayed only for large view)

The function of each component is described below.

(1) Execution control buttons

The execution control buttons allow the user to execute frequently used commands simply by clicking the corresponding button. On each button, the corresponding command is indicated graphically, as an icon. Commands assigned to the buttons can also be executed from the menu bar. Any buttons that are currently disabled are displayed in reverse video. During the execution of a program, for example, buttons other than the stop button are displayed in reverse video, so that clicking them has no effect.

⏸ Stops the user program.

▶ Executes the user program.
Double-clicking this button displays the Prompt dialog box. By specifying a line or function (or an address expression if debugging is being performed at the instruction level) in the Prompt dialog box, the user can set a temporary breakpoint (quick break) at the specified position, after which the target program is executed.

⤒ Executes the program step by step, until execution returns to the calling function.

⏭ Performs Next step execution of the program.
Double-clicking this button displays the Prompt dialog box. By specifying the number of lines (or the number of instructions if debugging is being performed at the instruction level) in the Prompt dialog box, the specified number of steps are executed.
For source level debugging, the program is executed in units of lines. For instruction level debugging, the program is executed in units of instructions.
If a program text area which contains no debug information is encountered during step execution, the program is stopped at that point.

⇥ Executes a program step by step.
Double-clicking this button displays the Prompt dialog box. By specifying the number of lines (or the number of instructions if debugging is being performed at the instruction level) in the Prompt dialog box, the specified number of steps are executed.
For source level debugging, the program is executed in units of lines. For instruction level debugging, the program is executed in units of instructions.

⏏ Initializes the emulation CPU.

(2) Debug mode selection option button

Clicking the Level option button, in the example above, displays an option menu from which either Src (source level debugging) or Ins (instruction level debugging) can be selected.

Once this selection has been made, the option button is displayed as Src or Ins, accordingly.

Src: Debugs a C source program. To use this mode, specify a debug option (-g) when compiling the program.

Ins: Debugs a machine language or assembly language source program.

(3) IE status display area

This area displays the current status of the in-circuit emulator, using the following symbols.

BRK: During a break

TRC: During tracer execution, or emulation

TIM: During timer execution, or emulation

RUN: During emulation

COV: During coverage execution

(4) Current position display area

Adr: Displays the current address (current PC value).

Current: Displays the program name, file name, line number, and function name corresponding to the current position.

"program-name $ file-name: line-number (function-name)"

Displays "-$-:0(-)" if no corresponding file is found.

(5) Message display area

This area displays a message indicating the operation which has been performed, or a warning message.

Buttons

The function buttons of the Main window are displayed only in large view mode. These buttons are identical to those in the Windows Box window.

Opens the Source window (p.120).

Opens the Variables window (p.133).

Opens the Local Variables window (p.144).

Opens the Stack Trace window (p.151).

Opens the Registers window (p.155).

Opens the SFR window (p.163).

Opens the Memory window (p.170).

Opens the Realtime RAM Tracer window (p.184).

Opens the Coverage window (p.188).

Opens the Disassemble window (p.201).

Opens the Event Manager window (p.213).

Opens the Event dialog box (p.222).

Opens the Break dialog box (p.229).

Opens the Trace dialog box (p.233).

Opens the Snap Shot dialog box (p.242).

Opens the Stub dialog box (p.246).

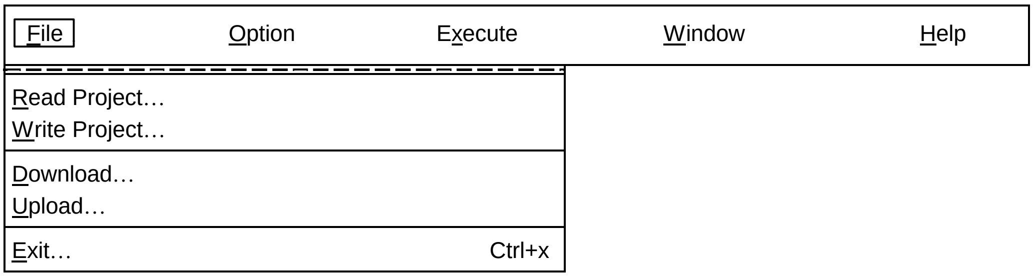
Opens the Timer dialog box (p.249).

Opens the Trace View window (p.252).

Opens the Timer window (p.263).

Menu bar

(1) File



Read Project… Displays the File Selection dialog box to read a debugging environment from a project file. Specify “prj” as the file extension.

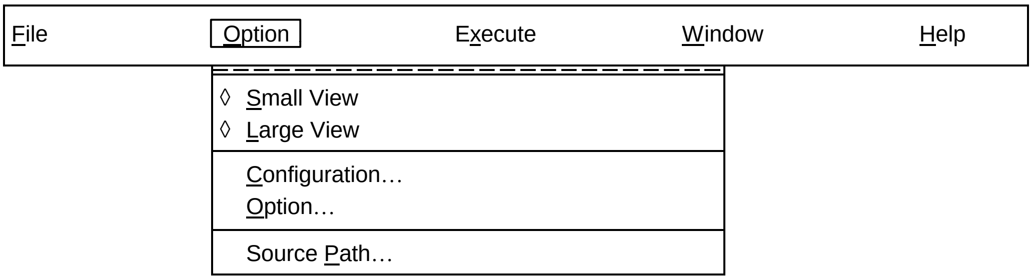
Write Project… Displays the File Selection dialog box to write the current debugging environment to a project file. Specify “prj” as the file extension.

Download… Displays the Download dialog box (p.96) to download a program. The directory to which the file has been downloaded is added to the source paths.

Upload… Displays the Upload dialog box (p.100) to upload a program.

Exit… Displays the Question dialog box (p.113) to terminate the debugger.

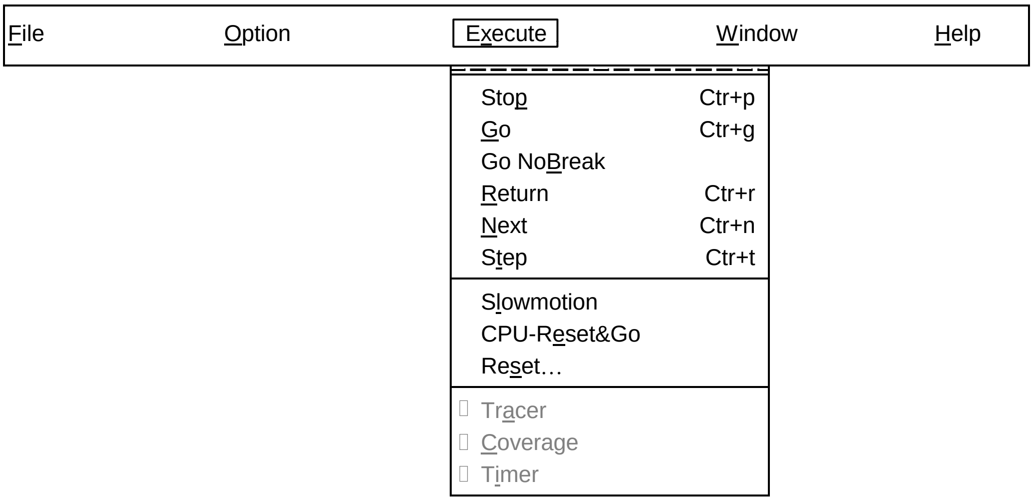
(2) Options



The first two items of this menu are configured as radio buttons.

Small View	Selects "small view" for the Main window.
Large View	Selects “large view” for the Main window.
Configuration…	Opens the Configuration dialog box (p.87) to set the debugging environment.
Option…	Opens the Option dialog box (p.94) to set the debugging options.
Source Path…	Opens the Prompt dialog box (p.115) to set the source paths.

(3) Execute



Stop Stops program execution: Executing this command has the same effect as clicking the ❚❚ button.

Go Starts program execution: Executing this command has the same effect as clicking the ▶ button.

Go NoBreak Executes the program by ignoring all the set breakpoints.

Return Executes the program step by step, until the execution returns to the calling function: Executing this command has the same effect as clicking the ⤒ button.

Next Performs Next step execution of the program: Executing this command has the same effect as clicking the ⏭ button.

Step Executes the program step by step: Executing this command has the same effect as clicking the ⏯ button.

Slowmotion Repeatedly performs step execution.

CPU-Reset&Go Resets the emulation CPU, then starts program execution.

Reset… Resets both the emulation CPU and debugger. The Selection dialog box (p.108) is displayed.

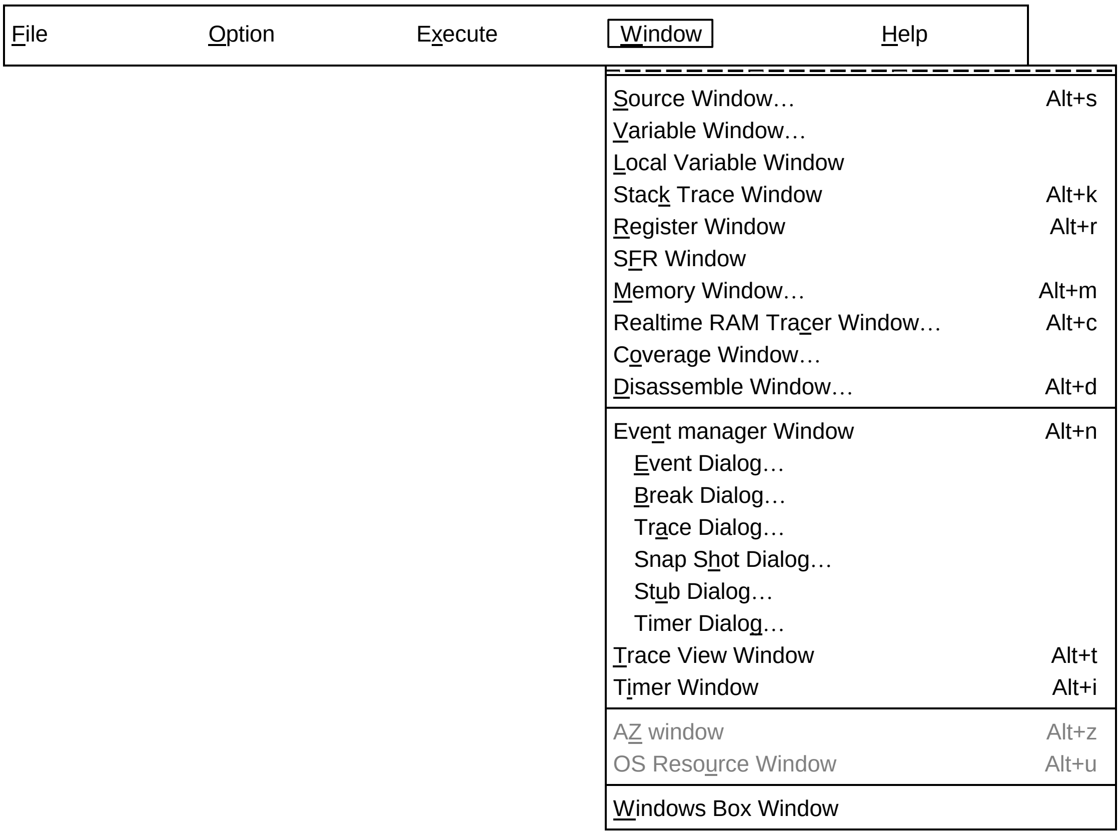
The following three items are configured as toggle buttons. These items cannot be selected in the break state. These items can be selected only during program execution.

□Tracer Enables/disables the tracer. The tracer is enabled while the button for this item is depressed. This item can be selected only during program execution with “Go NoBreak" specified.

□Coverage Enables/disables coverage measurement. Coverage measurement is enabled while the button for this item is depressed.

□Timer Enables/disables timer measurement. Timer measurement is enabled while the button for this item is depressed.

(4) Window



Source Window… Opens the Source window (p.120).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Variable Window…

Opens the Variables window (p.133).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Local Variable Window

Opens the Local Variables window (p.144).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Stack Trace Window

Opens the Stack Trace window (p.151).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Register Window Opens the Registers window (p.155).

Executing this command has the same effect as clicking the button in the Windows Box Window.

SFR Window Opens the SFR window (p.163).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Memory Window…

Opens the Memory window (p.170).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Realtime RAM Tracer Window…

Opens the Realtime RAM Tracer window (p.184).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Coverage Window…

Opens the Coverage window (p.188).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Disassemble Window…

Opens the Disassemble window (p.201).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Event Manager Window

Opens the Event Manager window (p.213).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Event Dialog… Opens the Event dialog box (p.222).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Break Dialog… Opens the Break dialog box (p.229).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Trace Dialog… Opens the Trace dialog box (p.233).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Snap Shot Dialog…

Opens the Snap Shot dialog box (p.242).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Stub Dialog… Opens the Stub dialog box (p.246).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Timer Dialog… Opens the Timer dialog box (p.249).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Trace View Window

Opens the Trace View window (p.252).

Executing this command has the same effect as clicking the button in the Windows Box Window.

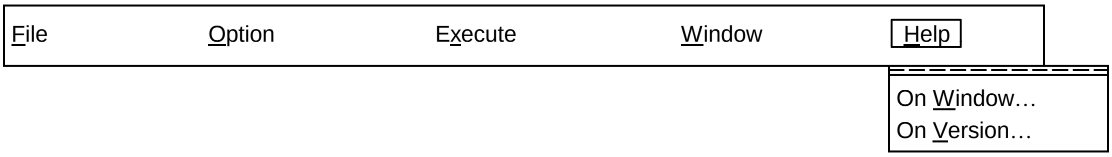
Timer Window Opens the Timer window (p.263).

Executing this command has the same effect as clicking the button in the Windows Box Window.

Windows Box Window

Opens the Windows Box window (p.267).

(5) Help



On Window…	Opens the Help dialog box (p.271), which displays help messages for the Main window.
On Version…	Displays the debugger version.

Notes

- Execution control at the source level cannot be applied if the program has been compiled without the debugging option (-g) specified, or if the load module has been loaded into the debugger with only raw data specified.
- During step execution, any break event which may occur is ignored, step execution taking precedence.
- While the emulation CPU is operating, only the ⏸ and ⏏ buttons are enabled in this window.

Configuration dialog box

Selection dialog box

Outline

The Configuration dialog box is used to display and set the operating environment for the in-circuit emulator.

This dialog box is opened in either of the following ways:

(1) Automatically opened when the debugger is started

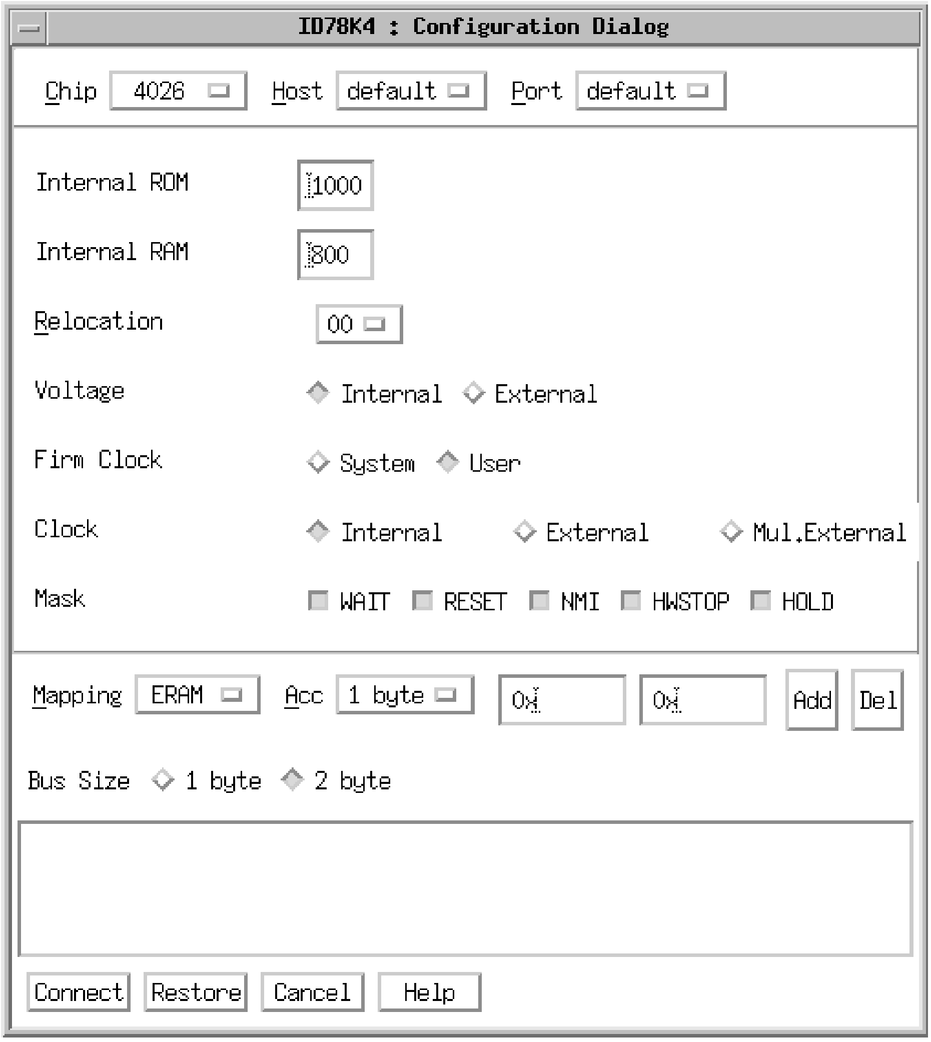
(2) In the Main window

Select Options → Configuration from the menu bar.

The CPU must be reset to make the modified mapping settings effective. If the mapping settings are modified, therefore, applying the set configuration closes all windows other than the Main window.

If the debugger is reset using the Main window (with CPU & BODY specified), the configuration information is initialized. In this case, re-specify the configuration information.

Window



ID78K4 : Configuration Dialog

Chip 4026 Host default Port default

Internal ROM 1000

Internal RAM 800

Relocation 00

Voltage ● Internal ○ External

Firm Clock ○ System ● User

Clock ● Internal ○ External ○ Mul.External

Mask □ WAIT □ RESET □ NMI □ HWSTOP □ HOLD

Mapping ERAM Acc 1 byte 0x 0x

Bus Size ○ 1 byte ● 2 byte

Function

The client area provides areas for setting the following items, which constitute the operating environment:

- Chip
- IE
- Port
- Internal ROM size
- Internal RAM size
- Relocation
- Operating power supply
- Firmware clock
- Clock
- Masking
- Bus sizing
- Mapping

The configuration procedure is described below.

(1) Selecting the chip

Select the chip to be used, by using an option button. By default, the chip specified upon starting the debugger is displayed. To use a different chip, click the option button, then select the chip from the displayed menu. Only those devices for which corresponding device files are stored in the device file directory can be selected.

(2) Selecting the IE

Select the in-circuit emulator (IE) to be used, by using an option button. Click the option button, then select the IE from the displayed menu. Only those IEs described for HOST in the setting file (= id78k4userdef) during installation can be selected. Selecting the IE also changes the port number to the corresponding number.

(3) Selecting the port

Select the port number of the IE to be used, by using an option button. Click the option button, then select the port number from the displayed menu. Only those port numbers described for HOST in the setting file (= id78k4userdef) during installation can be selected. Selecting a port number changes the IE selection to “default”, so that the IE having the selected port number will be connected automatically. Connection to IEs is attempted in the order in which they are described for HOST in the setting file (id78k4userdef).

(4) Setting the internal ROM size

Set the size of the internal ROM. By default, the internal ROM size of the selected chip is displayed.

(5) Setting the internal RAM size

Set the size of the internal RAM. By default, the internal ROM size of the selected chip is displayed.

(6) Setting relocation

Select the relocation address for the internal RAM area, by using an option button. By default, 00 is displayed. To change the address, click the option button, then select the address from the menu. Either 00 or 0F can be selected. If the downloaded program executes a location instruction based on a relocation address other than that set here, a relocation break occurs.

Downloading a load module file automatically sets the relocation address.

(7) Selecting the operating power supply

Select either Internal or External by using the radio buttons. By default, Internal is selected if the debugger is started without a target connected.

Internal: Uses the internal power supply of the IE as the operating power supply. The operating voltage is fixed to 5 V.

External: Uses the power supply of the target as the operating power supply. The operating voltage can be changed within the range specified for the device.

(8) Selecting the firmware clock

Select either System or User by using the radio buttons. By default, User is selected. Select the operating clock of the alternate software to be run while the IE is in the break state.

System: Operation at 1/2 divided clock of the main clock

User: Operation at the division ratio currently set with the main clock

(9) Selecting the clock

Select Internal, External, or Mul.External by using the radio buttons. By default, Internal is selected.

Internal: Uses the clock of the IE as the CPU clock.

External: Uses the clock of the target as the CPU clock.

Mul.External: Multiplies the clock of the target and uses the result as the CPU clock.

(10) Setting masking

Set masking for signals sent from the target, by using toggle buttons.

Masking can be set for each of WAIT, RESET, NMI, HWSTOP, and HOLD. To set masking for a signal, click the toggle button for that signal. The masked signals are not input to the IE. (Set masking only when the target operation is unstable during debugging.)

(11) Setting bus sizing

Select 1 byte or 2 bytes by using the radio buttons.

(12) Mapping

Specify the attribute, bus size, and address range for memory mapping. When the dialog box is selected, ERAM is selected as the attribute. Clicking the option button displays a menu containing the following attributes:

- ERAM: Maps the specified area as an ERAM area.
- EROM: Maps the specified area as an EROM area.
- TMem: Maps the specified area as a target area.
- Clear: Maps the specified area as a clear area.
- Stack: Maps the specified area as a stack area.

For the bus size, select 1 byte, 2 bytes, 4 bytes, or 8 bytes by using an option button. (The supported bus sizes vary with the chip.)

An attempt to map the specified area as a target area will result in an error if the target system is not connected or if the power to the target system is not turned on.

Enter the address range to be mapped, from the keyboard. Specify the start address at the left and the end address at the right. If the specified range does not match the mapping units, the minimum range including the specified addresses will become the mapping range. The mapping units vary with the addresses to be mapped, as follows:

Mapping address	Mapping units
0x000000 - 0x00ffff	2K bytes
0x010000 - 0x0fffff	64K bytes
0x100000 - 0xffffff	1M byte

Specified address ranges to be mapped are listed in the window at the bottom of the dialog box. Double-clicking an address range selects the address range. The selected address range can be added or deleted.

The following mapping attributes are displayed:

- ilram: Internal RAM
- clr: Mapping clear
- tmem: Target memory
- erom: Emulation ROM
- eram: Emulation RAM
- stk: Stack area

Note that, if the mapping settings are modified, applying the set configuration resets the CPU.

To set a mapping address range, specify the address range, then press the return key or click the Add button, located on the right of the dialog box.

To delete an address range, select the address range, then click the Del button.

Buttons

Connect — Applies the configuration information as specified in the dialog box, then connects to the IE. If the Configuration dialog box is opened with the IE already connected, this button is displayed as Set . Clicking the Set button re-sets the configuration information.

Restore — Restores the configuration information to that corresponding to the current environment.

Cancel — Closes the Configuration dialog box without any processing.

Help — Displays the Help dialog box (p.271), which provides help messages.

Notes

- When the use of the target clock or target memory has been specified, turn on the power to the target before applying the set configuration information.
- This dialog box is disabled while the emulation CPU is operating.

Errors

- If any invalid address is specified for mapping, the system beeps and displays the Warning dialog box (p.270).

Option dialog box

Selection dialog box

Outline

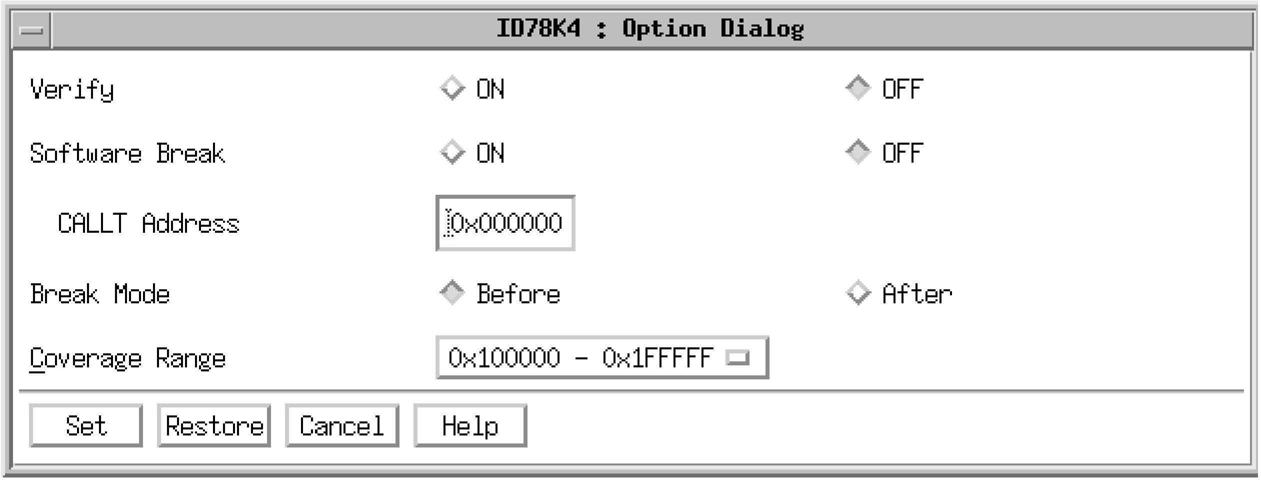
The Option dialog box is used to set the debugger options.

This dialog box is opened as follows:

- In the Main window, select Options from the menu bar.

  → Select Option... from the pull-down menu.

Window



ID78K4 : Option Dialog

Verify ◇ ON ◆ OFF

Software Break ◇ ON ◆ OFF

CALLT Address 0x000000

Break Mode ◆ Before ◇ After

Coverage Range 0x100000 - 0x1FFFFF

Description

This dialog box consists of the following components for setting options:

- Verify
- Software Break
- Break Mode
- Coverage Range

The function of each component is described below.

(1) Verify

Specify whether verification is to be performed when an operation such as a write into IE memory has been performed. Select the desired radio button.

ON: Performs verification.

OFF: Does not perform verification.

(2) Software Break

Specify whether software break is to be enabled. If it is enabled, the vector address of the CALLT instruction to be freed to the debugger must be specified. Select the desired radio button.

ON: Enables software break.

OFF: Disables software break.

CALLT Address: Specifies the vector address of the CALLT instruction.

(3) Break Mode

Specify whether the break is to be a before- or after-execution break. Select the desired radio button.

Before: Before-execution break.

After: After-execution break.

(4) Coverage range

Set the coverage range. Click the option button, then select the desired range from the option menu.

Buttons

Set Sets specified options.

Restore Restores the dialog box to the previous state.

Cancel Closes the Option dialog box without performing any processing.

Help Displays the Help dialog box (p.271).

Download dialog box

Selection dialog box

Outline

The Download dialog box is used to download a file to the target machine by specifying the name of the file.

The following files can be downloaded:

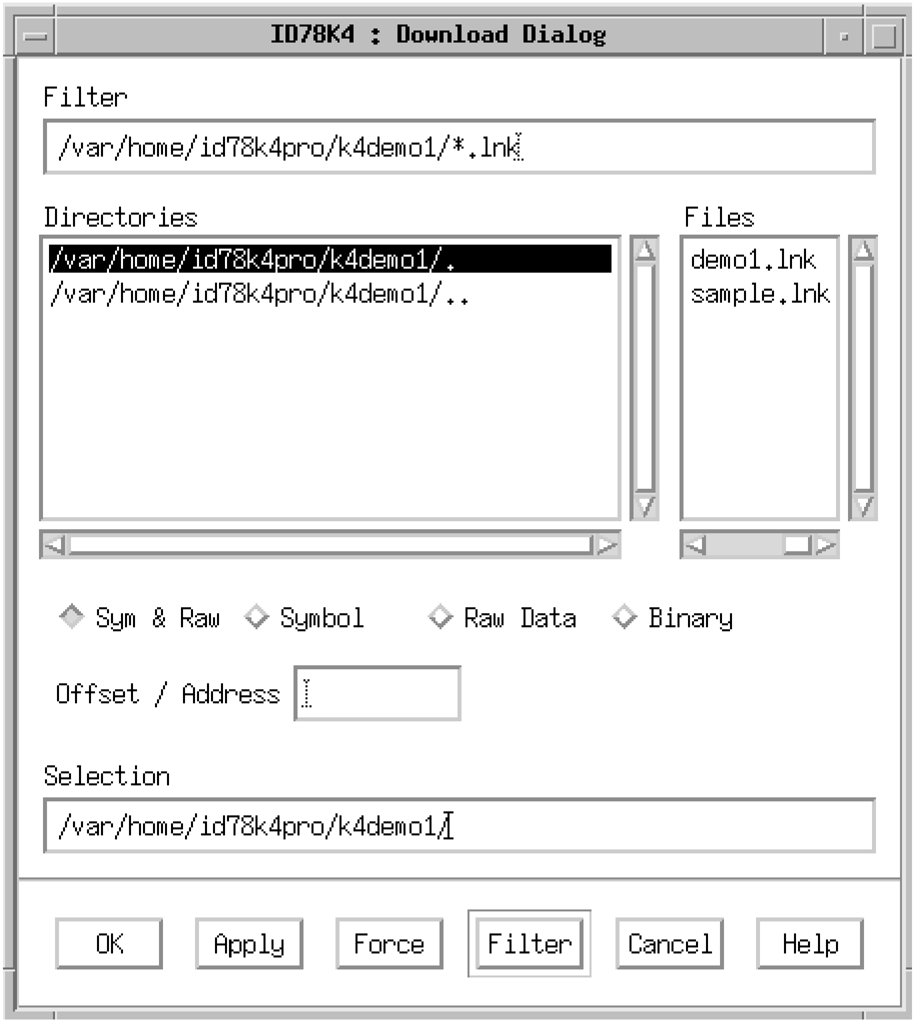
- Load module files, created using a compiler, assembler, and linker corresponding to the 78K4 series.
- Intel hexadecimal format files
- Motorola hexadecimal format files
- Tektronix hexadecimal format files
- Binary data files

This dialog box is opened as follows:

- In the Main window, select File from the menu bar.

  → Select Download from the pull-down menu.

Window



ID78K4 : Download Dialog

Filter

/var/home/id78k4pro/k4demo1/*.lnk

Directories

/var/home/id78k4pro/k4demo1/.
/var/home/id78k4pro/k4demo1/..

Files

demo1.lnk
sample.lnk

◆ Sym & Raw ◇ Symbol ◇ Raw Data ◇ Binary

Offset / Address

Selection

/var/home/id78k4pro/k4demo1/

Description

This dialog box consists of the following components:

Top of client area:	Area for displaying the filter for the file names to be listed. The current directory is displayed by default.
Middle of client area:	Lists the directories and files to be searched for the load module. Each list has scroll bars along its right and bottom edges. In the directory list, the current directory and current directory /.. are displayed by default.
Bottom of client area:	Area for setting the download conditions, area for displaying the selected file, and functional buttons. In the area for displaying the selected file, the current directory is displayed by default.

The download procedure is described below.

(1) Specifying the load directory

Specify the directory which contains the file to be downloaded, in any of the following ways:

- Enter the directory in the filter area from the keyboard, then press the return key.
- Double-click the directory in the directory list.
- Click the directory in the directory list, then click the Filter button.
- Click the directory in the directory list, then press the return key.

Once a directory has been specified, the files in that directory are listed.

(2) Specifying the load file

Specify the file to be downloaded, in either of the following ways:

- Click the file to be downloaded in the file list.
- Enter the file name in the file selection display area from the keyboard.

The selected file name is displayed in reverse video in the list and appears in the file selection display area. Clicking another file in the list deselects the selected file.

Only a single file can be selected.

(3) Setting the download conditions

Specify the download conditions for the selected file by using radio buttons and specifying an offset or address. Select one of the following three radio buttons according to the information to be downloaded. By default, both the symbol information and object body are downloaded.

• Sym & Raw:	Symbol information and object body
• Symbol:	Symbol information only
• Raw Data:	Object body only

To load symbol information (Sym & Raw or Symbol), enter a hexadecimal offset from the keyboard. The default offset is 0. To prevent the loading of symbol information (Raw Data or Binary), enter a hexadecimal address from the keyboard.

(4) Specifying the offset or address

Specify the offset or address.

(5) Executing download

Click the OK button, or double-click the file name in the file list. The selected file is downloaded according to the set download conditions.

If downloading of the file fails, the Error dialog box appears. The Error dialog box also appears if a load module has already been download.

To newly download a file when load module has already been downloaded, click the Force button.

Buttons

Select a file →

OK	Downloads the selected file according to the set conditions. Upon the completion of downloading, the Download dialog box is closed.
Apply	Downloads the selected file under the set conditions. Clicking this button does not close the Download dialog box.
Force	Newly downloads a file when a load module has already been downloaded. Clicking this button does not close the Download dialog box.
Filter	Displays the directories and files contained in the directory specified in the filter display area or directory list. The files are listed as specified in the filter display area.
Cancel	Closes the Download dialog box.
Help	Displays the Help dialog box (p.271).

Notes

- If a file in other than ID78K4 output file format is specified, specifying “symbol information only” for the load module conditions is ignored, resulting in both the symbol information and object body being loaded.
- This window cannot be opened while the CPU is performing emulation.

Errors

- If the OK button is clicked when no load module is selected, the system beeps.

Upload dialog box

Selection dialog box

Outline

The Upload dialog box is used to upload an object from target memory to the host machine.

The following files can be uploaded:

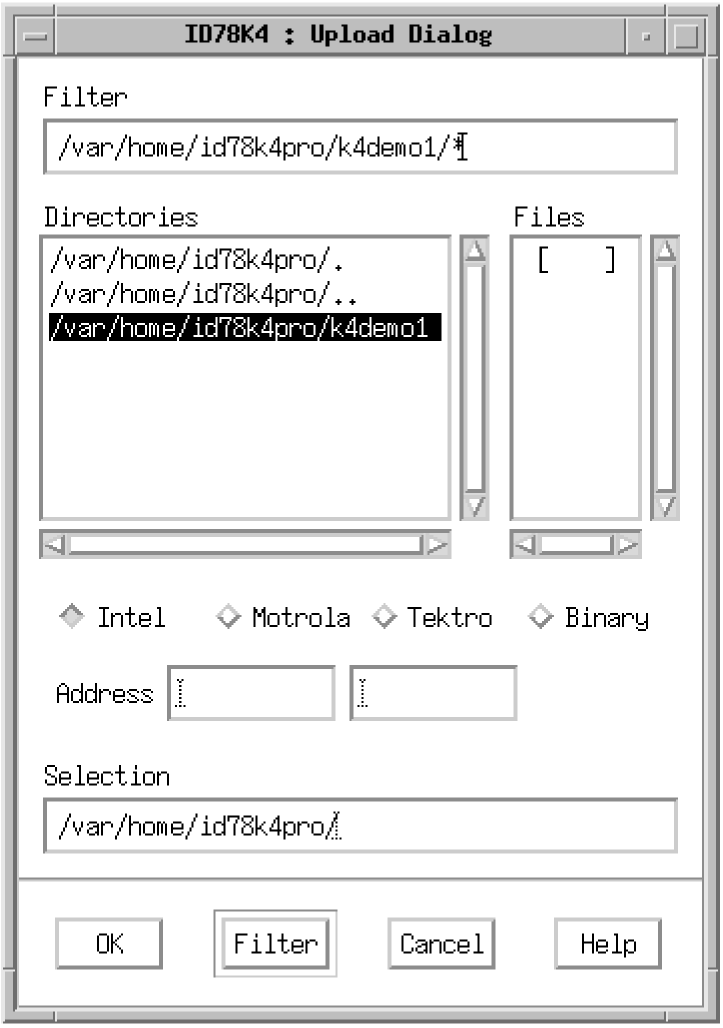
- Intel hexadecimal format files
- Motorola hexadecimal format files
- Tektronix hexadecimal format files
- Binary data files

This dialog box is opened as follows:

- In the Main window, select File from the menu bar.

  → Select Upload from the pull-down menu.

Window



ID78K4 : Upload Dialog

Filter: /var/home/id78k4pro/k4demo1/*

Directories:
/var/home/id78k4pro/.
/var/home/id78k4pro/..
/var/home/id78k4pro/k4demo1

Files: [   ]

● Intel ○ Motrola ○ Tektro ○ Binary

Address:

Selection: /var/home/id78k4pro/

Description

This dialog box consists of the following components:

Top of client area: Area for displaying the filter for the name of the file used to save the object file to be uploaded.

The current directory is displayed by default.

Middle of client area: Lists of the directories and files to be searched for the file that is to be used to save the object file.

Each list has scroll bars along its right and bottom edges.

In the directory list, the current directory and current directory /.. are displayed by default.

Bottom of client area: Area for selecting the format of the object file to be uploaded and area for displaying the file selected to save the object file.

In the area for displaying the selected file, the current directory is displayed by default.

The upload procedure is described below.

(1) Specifying the upload directory

Specify the directory used to save the file to be uploaded, using any of the following operations:

- Enter the directory in the filter area from the keyboard, then press the return key.
- Double-click the directory in the directory list.
- Click the directory in the directory list, then press the return key.

Once a directory has been specified, the files in that directory are listed.

(2) Specifying the upload file

Specify the name of the file on the host machine used to save the object file to be uploaded, in either of the following ways:

- Click the desired file name in the file list.
- Enter the file name in the file selection display area from the keyboard.

The selected file name is displayed in reverse video in the list and appears in the file selection display area. Clicking another file in the list deselects the selected file.

Only a single file can be selected.

(3) Specifying the format of the file to be uploaded

Specify the format of the object file to be uploaded. Select one of the following four radio buttons according to the desired format. By default, the file is uploaded in Intel hexadecimal format.

- Intel: Intel hexadecimal format
- Motorola: Motorola hexadecimal format
- Tektro: Tektronix hexadecimal format
- Binary: Binary format

(4) Specifying the address range

From the keyboard, enter the range of addresses in target memory that are to be uploaded.

(5) Executing upload

Click the OK button, or double-click the file name in the file list. The object file in the specified address range is loaded from the target machine to the host machine, and saved with the specified file name.

If uploading of the file fails, the Error dialog box appears.

Buttons

Select
an address/file →

OK Uploads the object file in the specified address range from the target machine to the host machine, and saves it with the specified file name.

Filter Displays the directories and files contained in the directory specified in the filter display area or directory list. The files are listed as specified in the filter display area.

Cancel Closes the Upload dialog box.

Help Displays the Help dialog box (p.271).

Notes

- This window cannot be opened while the CPU is performing emulation.

Errors

- If the OK button is clicked while no address range is specified, the system beeps.
- If the start address is greater than the end address, the system beeps.

File Selection dialog box

Selection dialog box

Outline

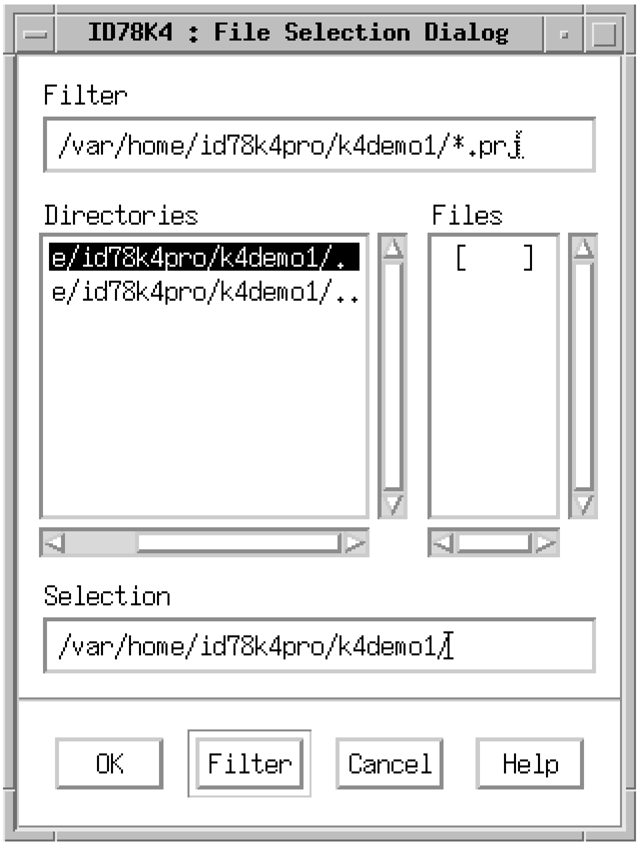
The File Selection dialog box is used to select a file to be accessed.

The File Selection dialog box enables the user to:

(1) Access a project file

(2) Access a view file

Window



Filter: /var/home/id78k4pro/k4demo1/*.prj

Directories:
e/id78k4pro/k4demo1/.
e/id78k4pro/k4demo1/..

Files: [ ]

Selection: /var/home/id78k4pro/k4demo1/

Description

This dialog box consists of the following components:

Top of client area: Area for displaying the filter for the name of the file to be accessed.
The current directory is displayed by default.

Middle of client area: Lists the directories and files to be searched for the file to be accessed.
Each list has scroll bars along its right and bottom edges.
In the directory list, the current directory and current directory /.. are displayed by default.

Bottom of client area: Area for displaying the selected file.
The current directory is displayed by default.

The directory and file manipulation procedures are described below.

(1) Specifying the directory

Specify the directory which contains the file to be accessed, using any of the following operations:

- Using the keyboard, enter the directory in the filter area, then press the return key.
- Double-click the directory in the directory list.
- Click the directory in the directory list, then press the return key.

Once a directory has been specified, the files contained in that directory are listed on the right.

(2) Specifying the file

Specify the file to be accessed, in either of the following ways:

- Click the desired file name in the file list.
- Using the keyboard, enter the file name in the file selection display area.

The selected file name is displayed in reverse video in the list and appears in the file selection display area. Clicking another file in the list deselects the selected file.

Only a single file can be selected.

Files are initially filtered with the following extensions:

(a) Project files

.prj extension

(b) View files

Files have different extensions depending on the window state which they contain, as follows:

- Files containing the state of the Variables window: .var
- Files containing the state of the Local Variables window: .loc
- Files containing the state of the Stack Trace window: .stk
- Files containing the state of the Registers window: .reg
- Files containing the state of the SFR window: .sfr
- Files containing the state of the Memory window: .mem
- Files containing the state of the Coverage window: .cov
- Files containing the state of the Disassemble window: .dis
- Files containing the state of the Event Manager window: .evt
- Files containing the state of the Trace View window: .trv
- Files containing the state of the Timer window: .tim

The function of each component is described below.

(1) If a project file is specified

If Write Project... is selected, the debugging environment is written into the specified file. If Read Project... is selected, the debugging environment previously saved to the file is read and reproduced.

For writing, the sizes and positions of the displayed windows are also saved. For windows that can be in either the active or hold state, however, only the active-state windows are subject to writing.

The reading of a project file is followed by resetting (of the emulation CPU and the debugger); therefore, all windows other than the Main window are closed.

(2) If a view file is specified

If Write View File... is selected, the contents of the window used to call this dialog box are written into the specified file.

If Read View File... is selected, the contents of the file are read and reproduced.

For reading, the contents of the file are reproduced in the window used to call the dialog box (overwriting the previous contents).

Buttons

File →

OK	Executes the function of the File Selection dialog box. Closes the File Selection dialog box upon the completion of execution.

Directory →

Filter	Displays the directories and files contained in the directory specified in the filter display area or directory list. The files are listed as specified in the filter display area.
Cancel	Closes the File Selection dialog box.
Help	Displays the Help dialog box (p.271).

Notes

- The reading of a project file is followed by resetting (of CPU & BODY); therefore, all windows other than the Main and Command windows are closed.
- When a project file is created, the raw data for load modules is not saved. To save the raw data for load modules, use the Upload dialog box.

  When a project file is read, it need not be loaded because it is automatically downloaded under the same conditions as those under which it was previously downloaded.
- The symbol information when a project file is created is saved by saving the position of the project file when it is read. If, therefore, the path of the project file when it is read differs from its path when created, the symbol information cannot be loaded.
- This window cannot be opened while the CPU is performing emulation.

Error

- If the OK button is clicked when no file is selected, the system beeps.

Selection dialog box

Selection dialog box

Outline

The Selection dialog box is used to select from among multiple items.

While this dialog box is displayed, the other windows of the debugger cannot be accessed. This dialog box enables the user to:

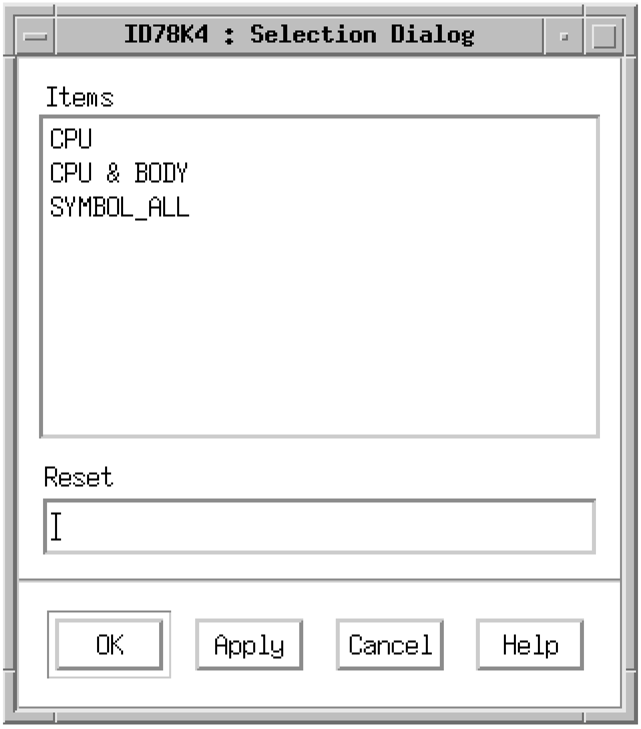
(1) Initialize the debugger

This window is opened as follows:

- In the Main window, select Execute.

  → Select Reset... from the pull-down menu.

Window



Description

This dialog box consists of the following components:

Top of client area: Lists the selectable items.

Bottom of client area: Area for displaying the selected item.

The function of each component is described below.

(1) Initializing the debugger

The debugger and the emulation CPU can be initialized.

Select one of the three items as the object of initialization:

(a) CPU

Only the emulation CPU is initialized.

Only a reset signal is sent to the emulation CPU; the program can be rerun by setting the registers again.

(b) CPU & BODY

Both the emulation CPU and the debugger are initialized. Thus, all windows other than the Main window are closed.

The entire environment retained by the debugger is initialized. To rerun the program, therefore, the environment must be set and the program must be loaded.

The following are initialized:

- Memory mapping (internal ROM size and stk area)
- Source path
- Events
- External sense clip mode
- Integrated events (breaks)
- Trace mode
- Trace display format
- Snapshots
- Target CPU (such as PC value and SFRs)
- Coverage information
- Symbol information

(c) SYMBOL_ALL

Only symbol information is initialized. Thus, any window displaying symbol information is closed. The program can be rerun merely by loading the symbol information.

Buttons

OK	Executes the function of the Selection dialog box. Closes the Selection dialog box upon the completion of execution.
Apply	Has the same function as the OK button; however, the Selection dialog box remains open even upon the completion of execution.
Cancel	Closes the Selection dialog box without performing any processing.
Help	Displays the Help dialog box (p.271).

Notes

- The emulation CPU is initialized by sending a reset signal to the emulation CPU. Thus, the program can be rerun by setting the registers again.
- When the debugger is initialized, the entire environment retained by the debugger is initialized. To rerun the program, therefore, the environment must be set and the program must be loaded.
- This window cannot be opened while the CPU is performing emulation.

Items Selection dialog box

Selection dialog box

Outline

The Items Selection dialog box is used to select or deselect items.

While this dialog box is displayed, the other windows of the debugger cannot be accessed. This dialog box enables the user to:

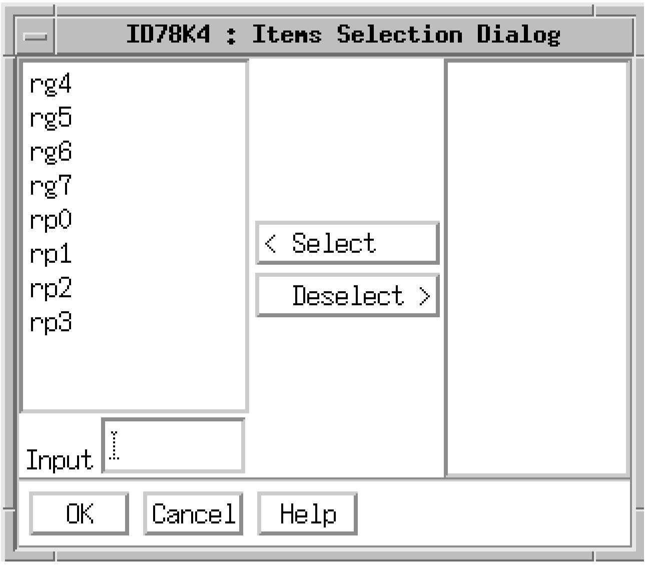
(1) Select the register to be displayed in the Registers window

(2) Select the SFR to be displayed in the SFR window

This window can be opened in any of the following ways:

- In the Registers window, select View.
  → Select Select General Registers… from the pull-down menu.
- In the Registers window, select View.
  → Select Select Banks... from the pull-down menu.
- In the Registers window, select View.
  → Select Select System Registers... from the pull-down menu.
- In the SFR window, select View.
  → Select Select SFRs... from the pull-down menu.

Window



Description

This dialog box consists of the following components:

Client area Left:	Lists the selected items.
Client area Right:	Lists those items which have not been selected.
Bottom of client area:	Register condition setting area (displayed only when register conditions are set)
Button area	

The function of each component is described below.

(1) Selecting the register to be displayed in the Registers window

The left part of the client area lists the registers that are currently selected for display. The right part lists those registers that have not been selected for display. To select a register, click the desired register. Multiple registers can be specified by dragging.

(2) Selecting the SFR to be displayed in the SFR window

The left part of the client area displays the SFRs that are currently selected for display. The right part lists the SFRs that have not been selected for display. To select an SFR, click the desired SFR. Multiple SFRs can be specified by dragging.

Buttons

OK	Executes the function of the Items Selection dialog box. Closes the Items Selection dialog box upon the completion of execution.
Cancel	Closes the Items Selection dialog box without performing any processing.
Help	Displays the Help dialog box (p.271).

Question dialog box

Confirmation dialog box

Outline

The Question dialog box displays a message requesting a response.

While this dialog box is displayed, the other windows of the debugger cannot be accessed.

The Question dialog box enables the user to:

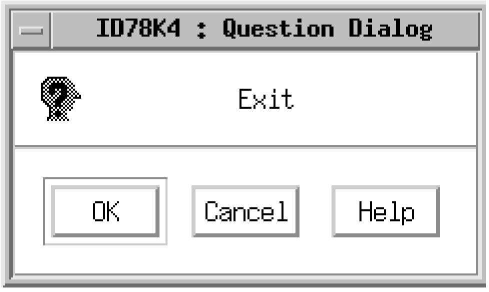
(1) Initialize the debugger

This dialog box can be opened as follows:

- In the Main window, select File.

  → Select Exit... from the pull-down menu.

Window



Description

This dialog box contains the following component:

Client area: Displays a message requesting a response.

The function of the component is described below.

(1) Exiting from the debugger

Confirm existing from the debugger.

When the debugger is exited, all its windows are closed.

Buttons

OK	Executes the function of the Question dialog box.
Cancel	Closes the Question dialog box without performing any processing.
Help	Displays the Help dialog box (p.271).

Prompt dialog box

Auxiliary dialog box

Outline

The Prompt dialog box is used to enter, from the keyboard, addresses and variable names.

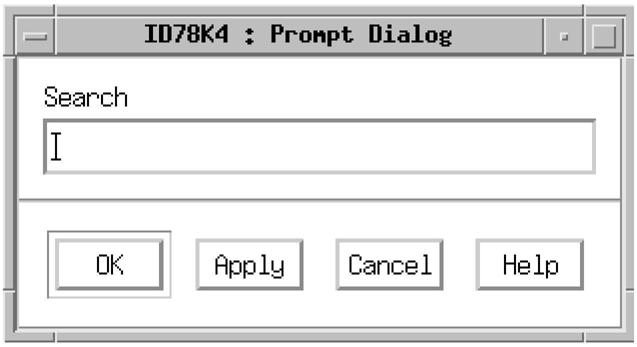
The Prompt dialog box enables the user to:

(1) Specify the start display address in the Memory window, the Realtime RAM Tracer window, and the Disassemble window

(2) Search for a list in the Source window and the Disassemble window

(3) Print the contents of a window

Window



Description

In the client area, an area for entering data from the keyboard is provided. If an address expression, variable name, or search list has been selected in a window, it is displayed.

The function of each component is described below.

(1) If an address has been specified

A memory dump or the results of disassembly are displayed, starting from the specified address. Only a hexadecimal number or address expression can be specified as an address.

(2) If a search list has been specified

The system searches for the specified list in the forward direction, starting from the position in the window where a prompt appears. The list is displayed in reverse video. To continue the search, select View from the Source or Disassemble window and select Search Forward or Search Backward from the pull-down menu.

(3) Printing the contents of a window

Specify a command for printing. The contents of the window from which the dialog box was called are printed. The default command is lpr %s. %s is automatically replaced with the name of the temporary file used to store the contents.

Buttons

OK Executes the function of the Prompt dialog box. Closes the Prompt dialog box upon the completion of execution.

Apply Executes the function of the Prompt dialog box. The Prompt dialog box remains open even upon the completion of execution.

Cancel Closes the Prompt dialog box.

Help Displays the Help dialog box (p.271).

Notes

- None

Error

- If the OK button is clicked while nothing is specified in the client area, the system beeps.

Source File Select dialog box

Selection dialog box

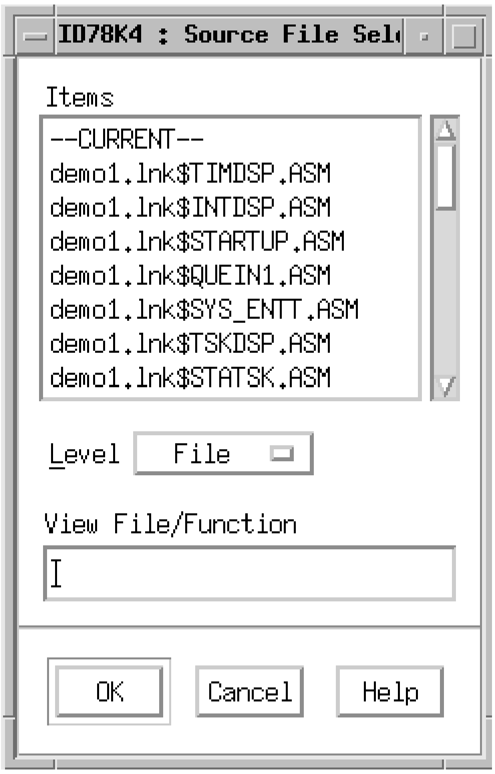
Outline

The Source File Select dialog box displays the names of the source files or functions contained in the downloaded load module, enabling the selection of the file to be displayed in the Source window.

This dialog box can be opened in any of the following ways:

- In the Main window, click Window on the menu bar.
  → Click Source Window... on the pull-down menu.
- In the Windows Box window, select 📄.
- In the Source window, click File on the menu bar.
  → Click Open on the pull-down menu.

Window



```
--CURRENT--
demo1.lnk$TIMDSP.ASM
demo1.lnk$INTDSP.ASM
demo1.lnk$STARTUP.ASM
demo1.lnk$QUEIN1.ASM
demo1.lnk$SYS_ENTT.ASM
demo1.lnk$TSKDSP.ASM
demo1.lnk$STATSK.ASM
```

Description

This dialog box consists of the following components:

Top of client area: Lists the names of the source files or functions contained in the downloaded load module. The option button located in the middle of the client area is used to specify whether source file or function names are to be listed.

Middle of client area: Option button used to specify the level of the list displayed at the top of the client area. If File is selected, the top of the client area lists the names of the source files contained in the downloaded load module.

If Function is selected, the names of the functions contained in the downloaded load module are listed.

This selection is displayed if the program being debugged has been compiled without the debug option (-g).

File is selected by default.

Bottom of client area: Area for displaying the name of the selected source file or function.

The source file manipulation procedure is described below.

(1) Selecting the level

Click the File ▭ button next to Level. A menu for switching between File and Function appears. Select either one. By default, File is selected.

(2) Selecting a source file or function

Select a source file or function in either of the following ways:

- Click the desired file or function name in the list.
- Enter, from the keyboard, the desired file or function name in the source file/function selection display area. The selected file or function name is displayed in reverse video in the list and appears in the source file/function selection display area. Clicking another file or function in the list deselects the selected file or function.

If symbol information corresponding to the current position exists, --CURRENT-- is displayed beside the source file or function name.

(3) Displaying the Source window

After selecting a source file or function, click the OK button. By default, the system searches for the selected source file or function, starting from the directory containing the downloaded load module. If, however, the path of the directory containing source files has been set in the Main window, the system searches that directory first, then the directory containing the downloaded load module. If, therefore, the source file to be debugged is in a directory other than that containing the downloaded load module, the path of that directory must be set in the Main window (p.74) in advance. Once the system has located a source file or function, the Source window appears.

If --CURRENT-- is selected, the Source window appears, starting from the current position.

Buttons

File/function name →

OK	Displays the specified file or function in the Source window and closes the Source File Select dialog box.
Cancel	Closes the Source File Select dialog box..
Help	Displays the Help dialog box (p.271).

Errors

- If the OK button is clicked when no source file or function name has been specified, the system beeps.
- If the selected file cannot be read, the Error dialog box appears.

Source window

View window

Outline

The Source window displays source text. It also allows the user to set breakpoints, display variable values, and search for character strings.

This window can be opened in any of the following ways:

- In the Main window, select Window from the menu bar.

  → Select Source Window... from the pull-down menu.

  → In the Source File Select dialog box, select a source file or function name.
  → Click the OK button.
- In the Windows Box window, select 🗎 .

  → In the Source File Select dialog box, select a source file or function name.
  → Click the OK button.
- In the Memory window, select an address.

  → Click Jump on the menu bar.

  → Click Source window on the pull-down menu.
- In the Disassemble window, select an address.

  → Click Jump on the menu bar.

  → Click Source window on the pull-down menu.
- In the Trace View window, select an address.

  → Click Jump on the menu bar.

  → Click Source window on the pull-down menu.
- In the Event Manager window, select an address.

  → Click Jump on the menu bar.

  → Click Source window on the pull-down menu.
- In the Coverage window, select an address.

  → Click Jump on the menu bar.

  → Click Source window on the pull-down menu.
- In the Stack Trace window, select an address.

  → Click Jump on the menu bar.

  → Click Source window on the pull-down menu.

- In the Registers window, select an address.

  → Click Jump on the menu bar.

  → Click Source window on the pull-down menu.

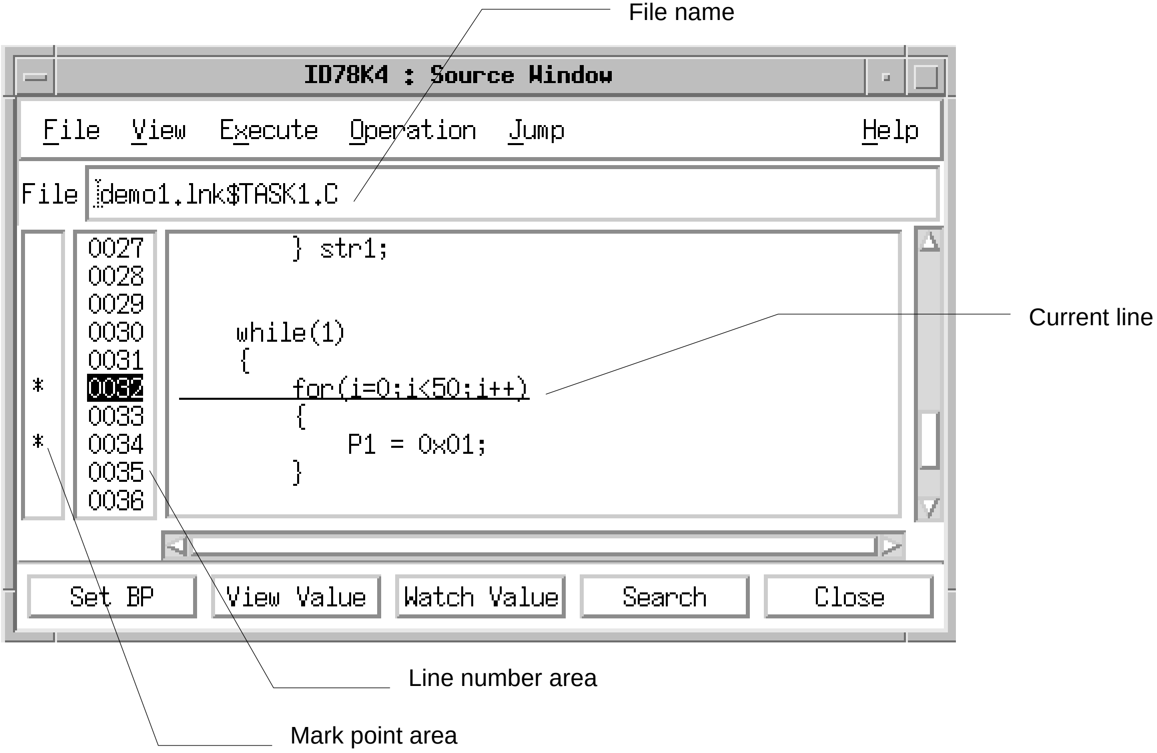
- In the SFR window, select an address.

  → Click Jump on the menu bar.

  → Click Source window on the pull-down menu.

Resetting the debugger by using the Main window (with CPU & BODY specified) closes this window.

Window



File demo1.lnk$TASK1.C

```
   0027         } str1;
   0028
   0029
   0030     while(1)
   0031     {
*  0032         for(i=0;i<50;i++)
   0033         {
*  0034             P1 = 0x01;
   0035         }
   0036
```

Description

This window consists of the following components:

Menu bar	
Top of client area:	Displays a file name.
Middle of client area:	Displays the contents of the file. A mark point area and a line number area are located on the left. Scroll bars are provided along the right and bottom edges.

In the part displaying a file name, a file name can be entered from the keyboard. This allows the user to switch from one file to another.

In the line number area, the line numbers corresponding to source text are displayed.

In the mark point area, an asterisk * is displayed to indicate an executable line. If a breakpoint is set, a breakpoint mark appears (b for a software breakpoint and B for a hardware breakpoint). The mark point area is two characters wide; breakpoint marks, described above, appear in the left position. In the right position, one of the following symbols may appear depending on the event that has been set:

T:	Trace
S:	Snapshot
I:	Timer
U:	Stub
&:	Two or more of the above events are set on the same line.

The current PC is indicated by an underline in the source text.

A line or variable required for using a function button or the menu bar can be selected as follows:

(1) Selecting a line

To select a line, click the desired line number (number in the line number area). Clicking a blank area or another line in the area deselects the selected line.

(2) Selecting a variable

To select a variable, either click the desired variable position (string in the text) (note that character strings consisting of alphanumeric characters and underlines, that begin with either an alphabetic character or an underline, are regarded as being variables) or use the mouse to drag the desired variable name. The selected variable is displayed in reverse video.

Clicking a blank area or another variable in the area deselects the selected variable.

The procedures for setting and canceling breakpoints in the Source window are described below.

A breakpoint can be set in either of the following ways.

(1) Using the mark point area

A breakpoint can be set by clicking the desired point in the mark point area. Clicking the same point again cancels the breakpoint. Breakpoints can be set only on those lines for which * appears in the mark point area.

(2) Select the desired line number and click the Set BP button.

For details of operation (2), see Buttons, described below.

Buttons

Line →

Set BP Displays a breakpoint mark in the mark point area, next to the selected line number in the line number area, and sets a breakpoint. If a line for which a breakpoint has already been set is selected, the breakpoint mark is erased and the breakpoint deleted.

Variable →

View Value Causes the Message dialog box (p.131) to appear, which contains the value of the specified variable.

Variable →

Watch Value Causes the Variables window (p.133) to appear, which contains the value of the specified variable.

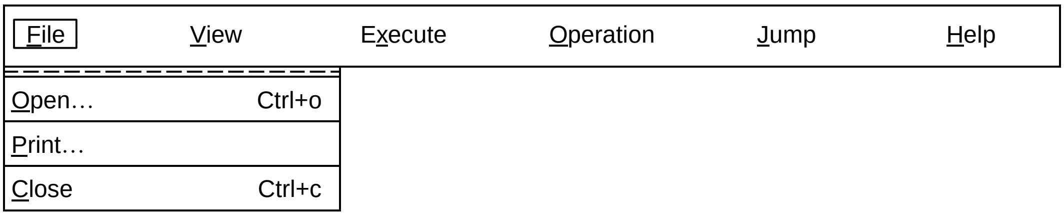
(List) →

Search Causes the Prompt dialog box (p.115) to appear, enabling the user to enter, from the keyboard, the name of the list to be retrieved. If a list has been selected in the Source window, the selected list is displayed in the Prompt dialog box, as retrieval data. To perform search, select View from the Source window and select Search Forward or Search Backward from the pull-down menu. The search starts from the current cursor position.

Close Closes the Source window.

Menu bar

(1) File

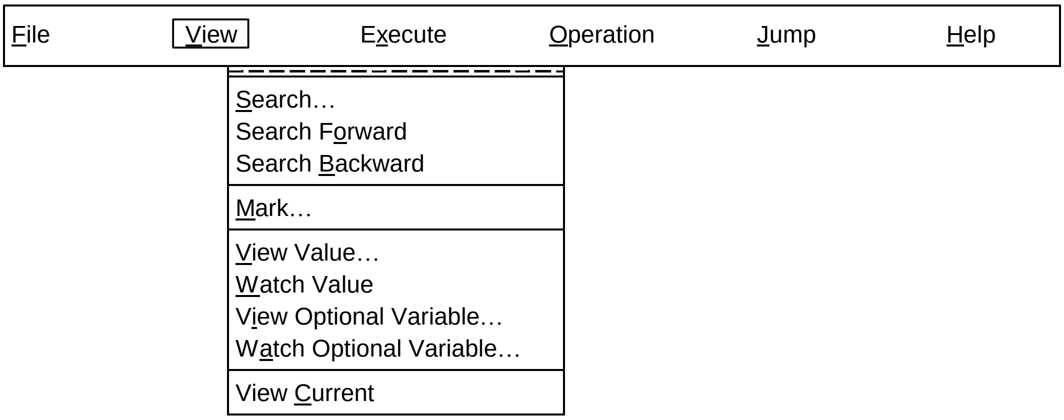


Open… Causes the Source File Select dialog box (p.104) to appear, enabling the user to display other source text.

Print… Causes the Prompt dialog box (p.115) to appear, enabling the user to specify a print command. The contents of the Source window can be printed.

Close... Closes the Source window. This has the same function as the Close button.

(2) View



(List) →

Search… Causes the Prompt dialog box (p.115) to appear, enabling the user to search for a list. This has the same function as the Search button.

Search Forward Performs search in the forward direction under the same conditions as those for the previous search. If a list is retrieved, the retrieved list is displayed in reverse video. If no list is retrieved, the system beeps.

Search Backward Performs search in the reverse direction under the same conditions as those for the previous search. If a list is retrieved, the retrieved list is displayed in reverse video. If no list is retrieved, the system beeps.

Line number →

Mark — Saves the selected line number as a mark point, and displays the Mark dialog box (p.126) containing the mark point that has just been saved. By selecting a mark point in the Mark dialog box, the system jumps to that mark point. If no line number has been selected, only the Mark dialog box is displayed.

Variable →

View Value… — Causes the Message dialog box (p.131) to appear, containing a variable value. It has the same function as the View Value button.

Variable →

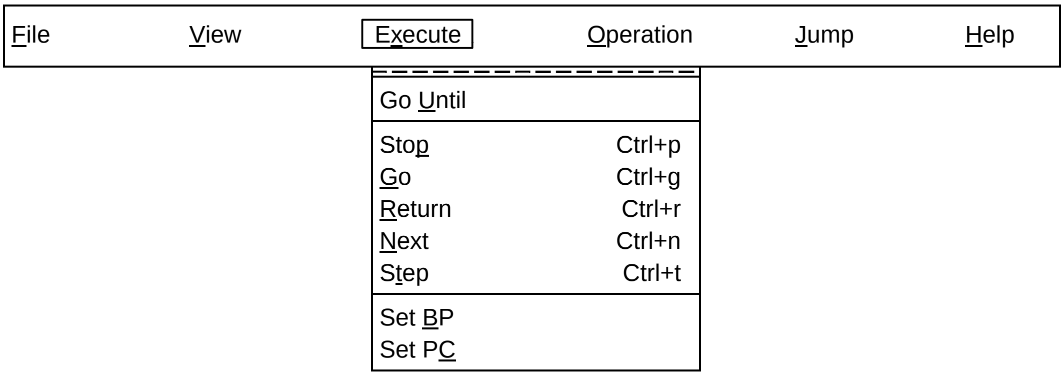
Watch Value — Causes the Variables window (p.133) to appear, displaying a variable value. It has the same function as the Watch Value button.

View Optional Variable…

Causes the Prompt dialog box (p.115) to appear, enabling the user to enter the name of the variable whose value is to be displayed. If a variable has been selected in the source text, it is displayed by default.

View Current — Displays the current source file.

(3) Execute



Line number →

Go Until — Executes the target program up to the specified line. The target program cannot be executed if a line number has not been selected by clicking.

Stop — Stops the target program. This has the same function as the ⏸ button in the Main window.

Go — Executes the target program. This has the same function as the ▶ button in the Main window.

Return — Executes the target program until execution returns to the calling function. This has the same function as the ⤒ button in the Main window.

Next Performs Next step execution of the program. This has the same function as the ▶▶| button in the Main window.

Step Executes the target program step by step. This has the same function as the ▶| button in the Main window.

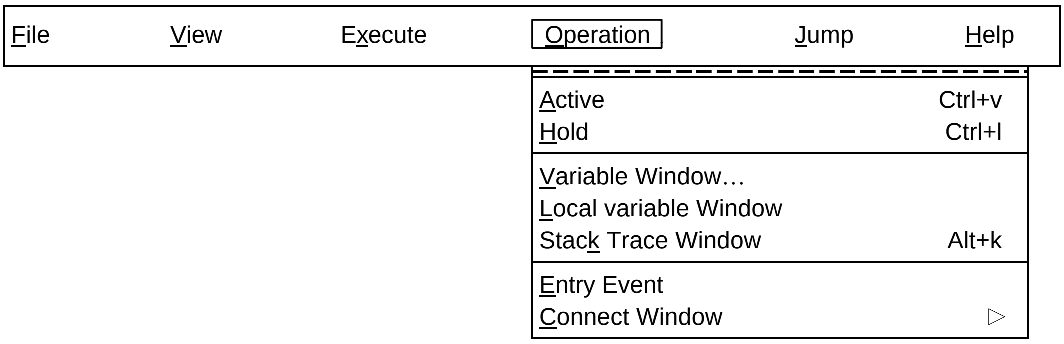
Line number →

Set BP Displays a breakpoint mark in the mark point area, next to the selected line number in the line number area, and sets a breakpoint. If a line for which a breakpoint has already been set is selected, the breakpoint mark is erased and the breakpoint deleted. This has the same function as the Set BP button.

Line number →

Set PC Sets the program counter to the specified line.

(4) Operation



Active Switches the window from the hold state to the active state.

Hold Switches the window from the active state to the hold state.

Variable →

Variable window… Causes the Prompt dialog box (p.115) to appear, enabling the user to enter the name of the variable to be monitored. If a variable has been selected in the source text, it is displayed in the Variables window.

Local variable →

Local Variable Window

Displays the Local Variables window (p.144).

Stack Trace Window

Displays the Stack Trace window (p.151).

Line number →

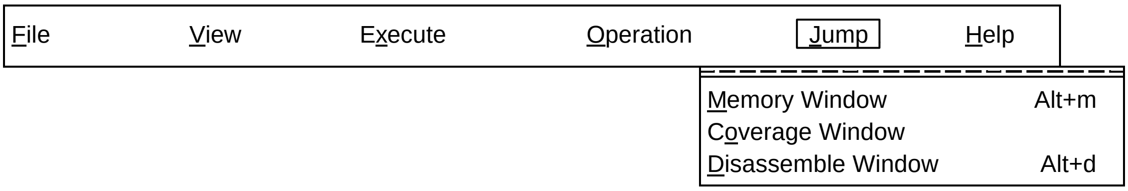
Entry Event Causes an event to be registered on the selected line.

Connect Window Displays a submenu for connecting windows.

In the submenu, the following windows can be selected using toggle buttons. Select the window to be connected.

- Memory window
- Coverage window
- Disassemble window

(5) Jump



Line number →

Memory Window

Displays the Memory window (p.170) starting from the address of the specified line. If the active-state Memory window is already displayed, the contents of the window are overwritten.

Line number →

Coverage Window

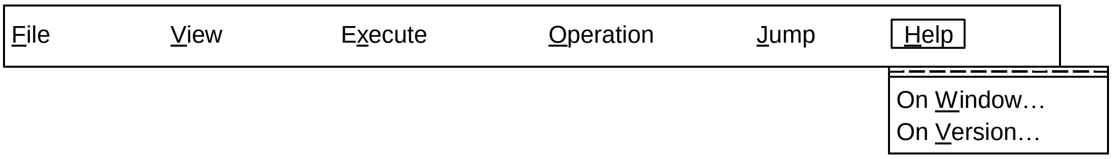
Displays the Coverage window (p.188), starting from the address corresponding to the specified line number.

Line number→

Disassemble Window

Displays the Disassemble window (p.201), starting from the address of the specified line. If the active-state Disassemble window is already displayed, the contents of the window are overwritten.

(6) Help



On Window… Displays the Help dialog box (p.271).

On Version… Displays the Message dialog box (p.131), which indicates the version of ID78K4.

Notes

- If the program was compiled with the debug option (-g) not specified, or if only raw data was specified when the load module was loaded into the debugger, a source list is not displayed.
- This window cannot be opened while the CPU is performing emulation.

Errors

- If an attempt is made to display another window using the Jump menu with no line number specified, the system beeps.

Mark dialog box

Selection dialog box

Outline

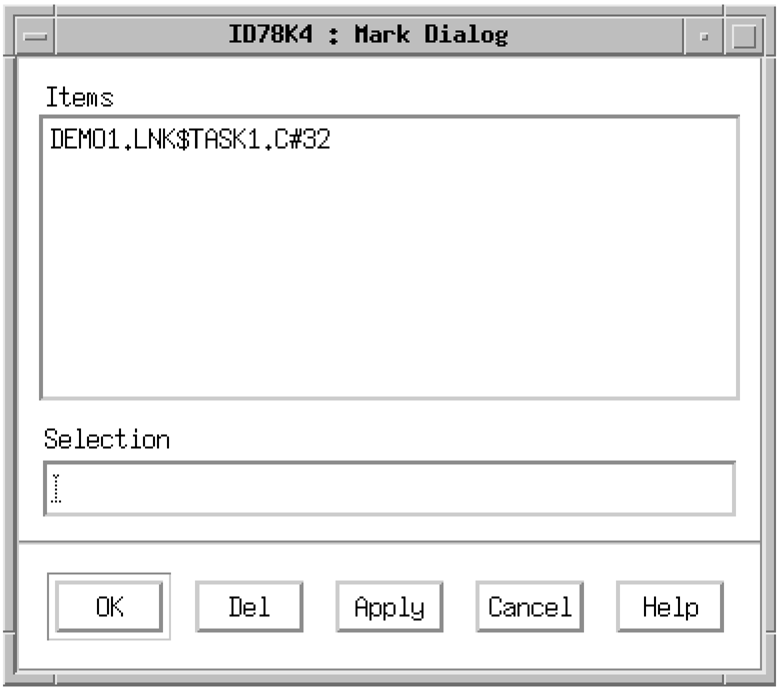
The Mark dialog box is used to manage and select the mark points selected in the Source window (p.120).

While this dialog box is displayed, the other windows of the debugger cannot be accessed.

This window can be opened in the following way:

- In the Source window, select a line, then select View from the menu bar.
  → Select Mark... from the pull-down menu.

Window



DEMO1.LNK$TASK1.C#32

Description

This dialog box consists of the following components:

Top of client area: Lists the selectable items.

Bottom of client area: Area for displaying the selected item.

Button area

The function of each component is described below.

Select the mark point to which a jump is to be performed, from the selectable items at the top of the area.

To select it, click it with the mouse.

The selected mark point is displayed in reverse video.
With the mark point displayed in reverse video, click either OK or Apply to jump to the mark point.

To delete the selected mark point from the dialog box, click Del.

Buttons

OK	Executes the function of the Mark dialog box. Closes the Mark dialog box upon the completion of execution.
Apply	Has the same function as the OK button, except that the Mark dialog box remains open even upon the completion of execution.
Cancel	Closes the Mark dialog box without performing any processing.
Help	Displays the Help dialog box (p.271).

Message dialog box

Confirmation dialog box

Outline

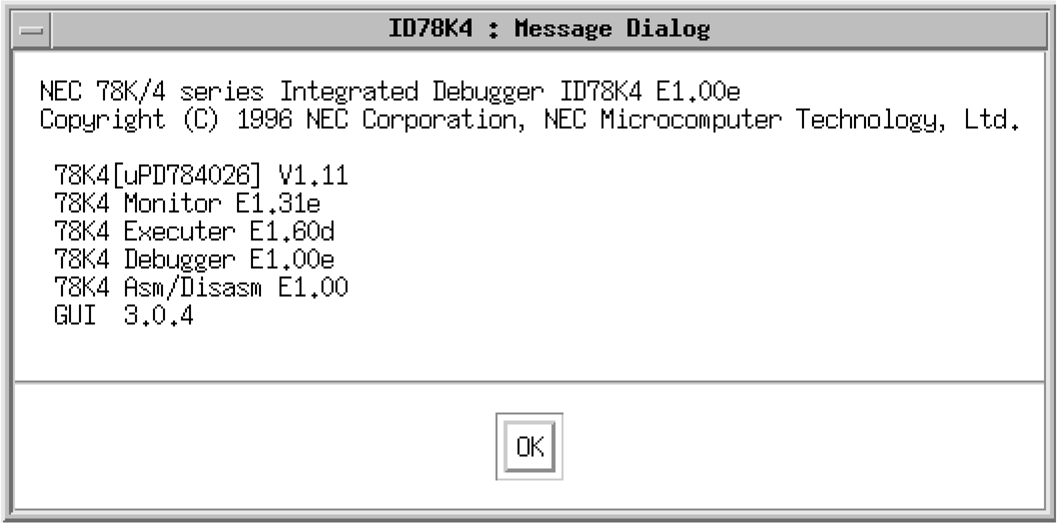
The Message dialog box displays the version of ID78K4 and temporarily displays the value of a variable. The Message dialog box enables the user to:

(1) Display the version of ID78K4

(2) Temporarily display the value of a variable

While this dialog box is displayed, the other windows of the debugger cannot be accessed.

Window



```
NEC 78K/4 series Integrated Debugger ID78K4 E1.00e
Copyright (C) 1996 NEC Corporation, NEC Microcomputer Technology, Ltd.

 78K4[uPD784026] V1.11
 78K4 Monitor E1.31e
 78K4 Executer E1.60d
 78K4 Debugger E1.00e
 78K4 Asm/Disasm E1.00
 GUI  3.0.4
```

Description

This dialog box contains the following component:

Client area: Displays messages.

The function of this component is described below.

(1) Displaying the version of ID78K4

To display the version of ID78K4, select Help and click On Version on the pull-down menu.

(2) Temporarily displaying the value of a variable

To display the value of a variable, select the desired variable in the Source window and click the View Value button in the Source window.

Button

OK	Closes the Message dialog box.

Variables window

View/setting window

Outline

The Variables window is used to display or modify the value of the variable specified in the Source window (p.112) or the Prompt dialog box (p.107).

This window can be opened in any of the following ways:

- In the Main window, select Window from the menu bar.

  → Select Variable Window... from the pull-down menu.

  → Enter the variable to be displayed in the Prompt dialog box.
- In the Windows Box window, click 🔍 .

  → Specify a variable in the Prompt dialog box.
  → Click the OK button in the Prompt dialog box.
- In the Source window, select a variable.
  → Click the Watch Value button.
- In the Source window, select a variable.

  → Click View on the menu bar.

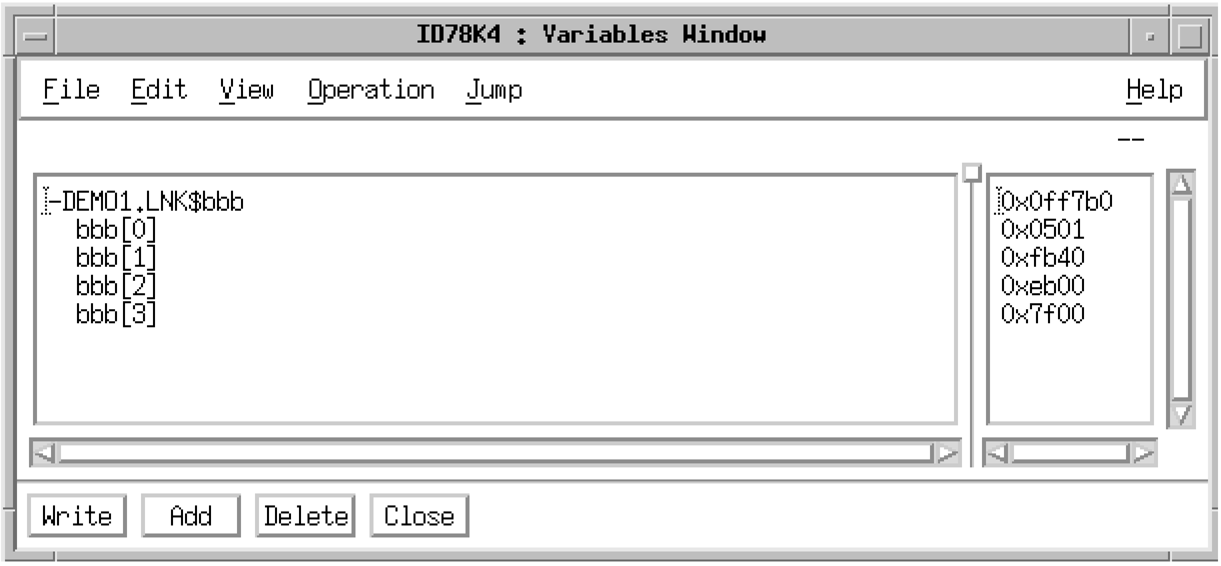
  → Click Watch Value on the pull-down menu.
- In the Source window, select a variable.

  → Click Operation on the menu bar.

  → Click Variable Window... on the pull-down menu.

Resetting the debugger by using the Main window (with CPU & BODY specified) closes this window.

Window



-DEMO1.LNK$bbb 0x0ff7b0
bbb[0] 0x0501
bbb[1] 0xfb40
bbb[2] 0xeb00
bbb[3] 0x7f00

Description

This window consists of the following components:

Menu bar

Client area Left: Displays variable names.

Client area Right: Displays variable values.

Button area

The following markings are displayed at the upper right corner of the client area to indicate whether modified values have already been set in the target.

--: Indicates that no values have been modified or that all the modified values have already been set in the target.

**: Indicates that a value is being modified or that a modified value has not yet been set in the target.

The value of a variable is displayed together with the names of the program, file, and function to which the variable belongs, as described below.

- The name of a structure or array is preceded by a + or -. A + indicates that the variable represents an address. Double-clicking the variable displays its members or elements individually, indented, with the + being replaced with a "-."
- For an array, all elements of the array are displayed according to the type of the array variable.
- For a structure, all members of the structure are displayed according to the types of the member variables. If a structure is defined within the structure, the tag and variable names of the internal structure are also displayed.
- For an enumeration variable, its member names are displayed.
- For a string pointer variable, a string is displayed, because the variable is regarded as being a pointer to a string.
- For a pointer variable other than character pointer variables, the address stored in the variable is displayed in hexadecimal.
- For a character variable, a character is displayed. A period is displayed to indicate a character code that cannot be displayed.
- For a variable other than the above, a hexadecimal number is displayed.

A variable is displayed whenever it is requested. If, however, the same variable is requested again, it is not displayed. Even if the same variable as that requested exists in a hold-state window, that variable is ignored.

Values are updated upon the completion of execution. To save the values, therefore, place the window in the hold state. To place the window in the hold state, click Operation on the menu bar and click the appropriate radio button in the pull-down menu.

A variable is set as follows.

When modifying the value of a variable, note the variable type. If a string constant is specified as a set value, the string is set at the address pointed to by the variable if the variable is a string pointer. \0 is appended to the string.

(1) Selecting a variable

To select a variable, click the desired variable name or value.

Clicking the variable name again deselects the variable.

(2) Setting a variable name

After selecting a variable value as described above, directly modify the value from the keyboard, then click the Write button.

Buttons

Write Writes the modified value into target memory.

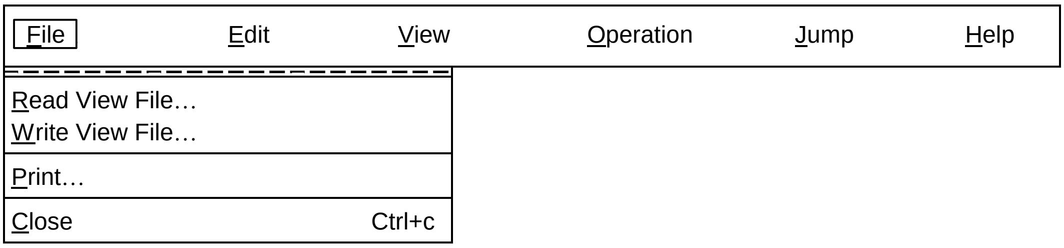
Add Displays the Prompt dialog box (p.115), enabling the specification of a new variable.

Variable → Delete Deletes the specified variable from the Variables window.

Close Closes the Variables window.

Menu bar

(1) File



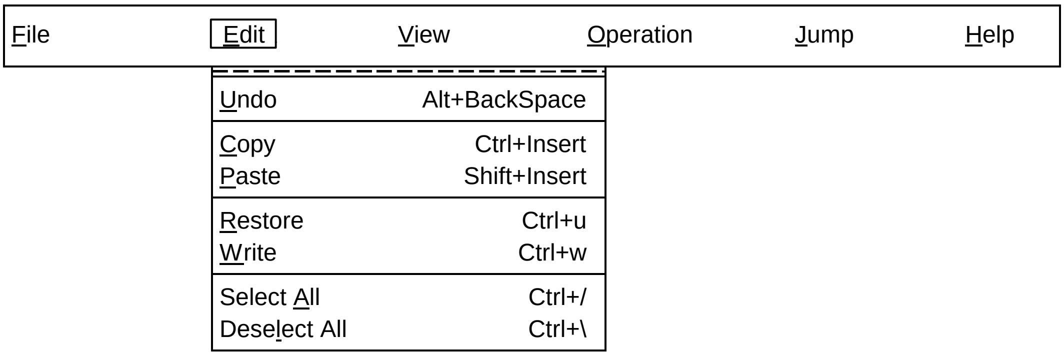
Read View File… Displays the File Selection dialog box (p.104), enabling the specification of the file containing the contents of this window. The contents of the file are read for reproduction. The file to be specified must have the var extension.

Write View File… Displays the File Selection dialog box (p.104), enabling the specification of the file into which the contents of this window are to be saved. The contents of the window are written into the file. The file to be specified must have the var extension.

Print… Displays the Prompt dialog box (p.115), enabling specification of a print command. The contents of the Variables window can be printed.

Close Closes the Variables window. This has the same function as the Close button.

(2) Edit



Undo — Undoes the previous edit operation.

String →

Copy — Copies the string, selected with the mouse, to the clipboard buffer.

Paste — Pastes the contents of the clipboard buffer at the text cursor position. The position must have already been selected by double-clicking. Even if the display mode has been changed, the contents of the clipboard buffer are automatically changed according to the current display mode. A string that has been copied in another window cannot be pasted.

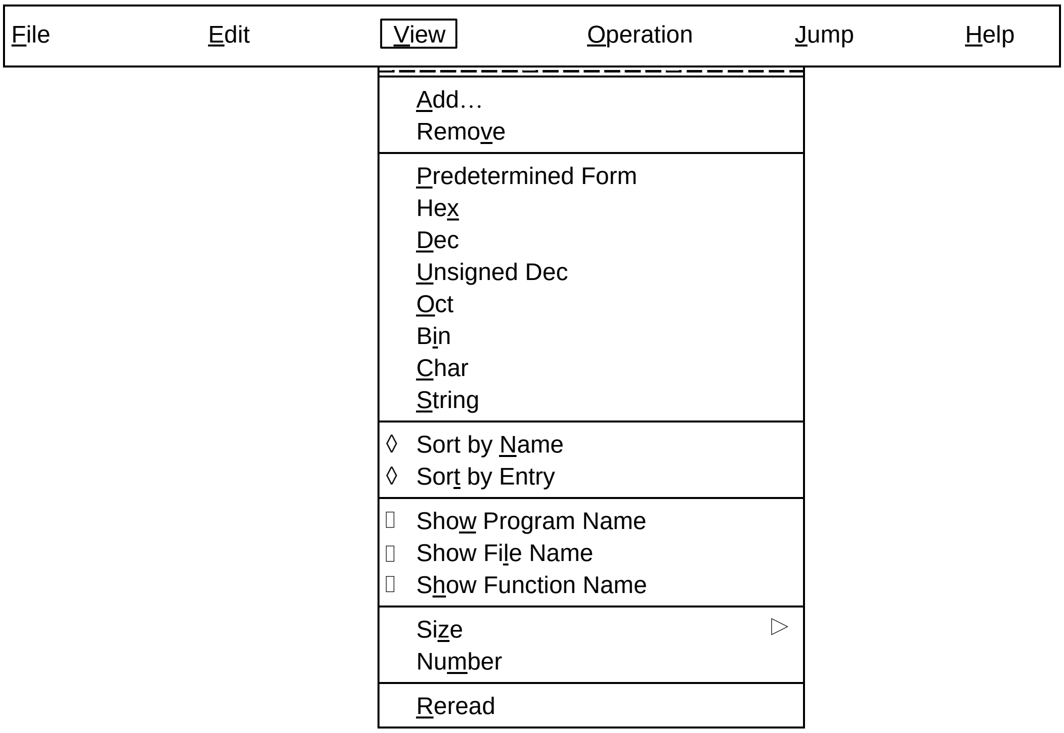
Restore — Abandons the modified value and restores the previous value.

Write — Writes the modified value into the target memory. This has the same function as the Write button.

Select All — Selects all displayed variable names.

Deselect All — Deselects all selected variable names.

(3) View



Add… Displays the Prompt dialog box (p.115), enabling the specification of the variable to be displayed. This has the same function as the Add button.

Variable →

Remove Removes the specified variable from the Variables window. This has the same function as the Delete button.

Predetermined Form

Displays variable values in predetermined format.

Hex Displays variable values in hexadecimal. The values are preceded by 0x.

Dec Displays variable values as signed decimal numbers.

Unsigned Dec Displays variable values as unsigned decimal numbers.

Oct Displays variable values in octal. The values are preceded by 0.

Bin Displays variable values in binary. The values are preceded by 0b.

Char Displays variable values as characters. Characters are enclosed in single quotation marks (').

String Displays the characters, starting from the address pointed to by each variable value to a NULL character (\0), as a string. The character string is enclosed in double quotation marks ("). The maximum number of characters in a string can be specified using Max String Length...

The following two items are provided as radio buttons on the menu:

Sort by Name Sorts variables into alphabetical order.

Sort by Entry Sorts variables into the order in which they were entered.

The following three items are provided as toggle buttons on the menu. Each button is selected by default.

Show Program Name

Displays program names in addition to variable names.

Show File Name Displays file names in addition to variable names.

Show Function Name

Displays function names in addition to variables.

Size Displays the following submenu, enabling the specification of the assembler variable size.

1 byte Reads assembler variables in byte units.

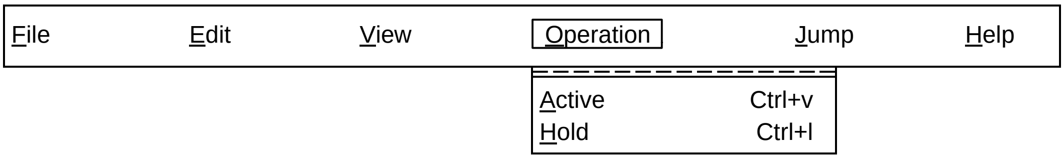
2 byte Reads assembler variables in word units.

4 byte Reads assembler variables in double word units.

Number Displays the Prompt dialog box, enabling the specification of the assembler variable read size and the number of assembler variables.

Reread Rereads and displays the values of the displayed variables.

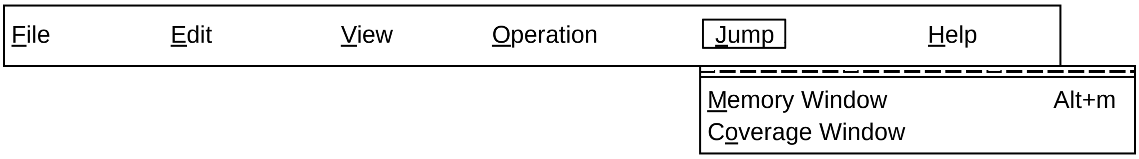
(4) Operation



Active Switches the window from the hold state to the active state.

Hold Switches the window from the active state to the hold state.

(5) Jump



Variable value →

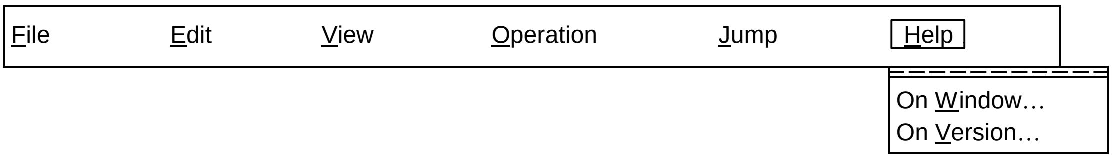
Memory Window Displays the Memory window (p.170), starting from the address of the specified variable value. If the active-state Memory window is already displayed, the contents of the window are overwritten.

Variable value →

Coverage Window

Displays the Coverage window (p.188) with the address of the selected variable.

(6) Help



On Window... Displays the Help dialog box (p.271).

On Version... Displays the Message dialog box (p.131), which shows the version of ID78K4.

Notes

- If the program was compiled with the debug option (-g) not specified, or if only raw data was specified when the load module was loaded into the debugger, a source list is not displayed.
- If a string is specified as a variable and a string longer than the area pointed to by the variable value is specified as a set value, data outside the area may be destroyed.
- When the window is closed and then displayed again, all the previous variable values are cleared.

Errors

- If data other than a variable name is specified, the Warning dialog box appears.

Add Variable dialog box

Selection dialog box

Outline

The Add Variable dialog box is used to specify the variable to be displayed in the Variables window and set necessary information.

This window can be opened in either of the following ways:

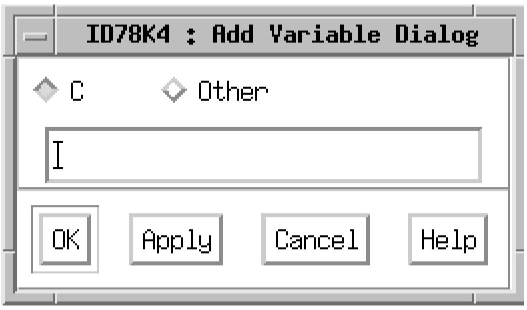
- When the active-state Variables window is not displayed, select Variable Window... from the Main window.
- In the Variables window, select View from the menu bar.

  → Select Add... from the pull-down menu.

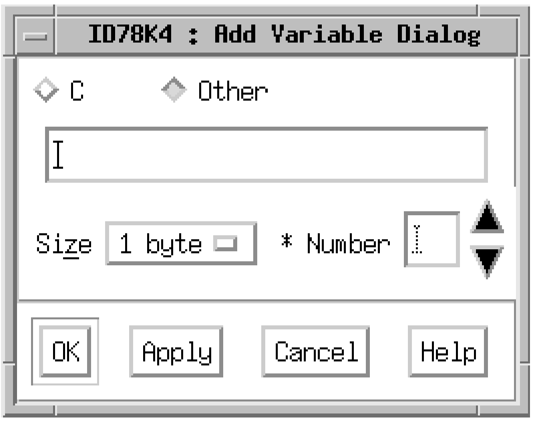
  → Enter the variable to be displayed in the Prompt dialog box.

Window

(1) If C is selected (default)



(2) If Other is selected



Description

This window consists of the following components:

Top of client area:	Area for selecting a language.
Middle of client area:	Area for specifying the variable to be added and additional information area.
Bottom of client area:	Function button area.

In the area for selecting a language, select the language in which the variable to be added is coded (C or assembler).

(1) If C is selected

To specify an external variable, specify the name of the variable in the area for specifying the variable to be added.

Format: "program-name"$"file-name"#"variable-name"

(program-name and file-name are optional.)

To specify a local variable, specify the information for the variable to be added, in the following format:

Format: "function-name"!"local-variable-name"

(function-name is optional.)

In the additional information area, additional information can be confirmed or canceled using the function button area.

(2) If Other is selected (assembler)

Specify a label as a variable name, as well as a size and number.

Format: "program-name"$"file-name"#"label"

(program-name and file-name are optional.)

Size is provided as option buttons, from which one of 1 byte, 2 byte, and 4 byte can be selected. In the Number field, enter the necessary number.

Buttons

OK	Adds, to the Variables window (p.130), the variable specified in the area for specifying the variable to be added, then closes the Add Variable dialog box.
Apply	Adds, to the Variable window, the variable specified in the area for specifying the variable to be added.
Cancel	Closes the Add Variable dialog box.
Help	Displays the help message for the Add Variable dialog box in the Help dialog box (p.271).

Errors

- If a variable name that does not exist is specified, or if a local variable is specified incorrectly, the Error dialog box (p.270) appears.

Local Variables window

View/setting window

Outline

The Local Variables window is used to display and modify the local variables in the current function.

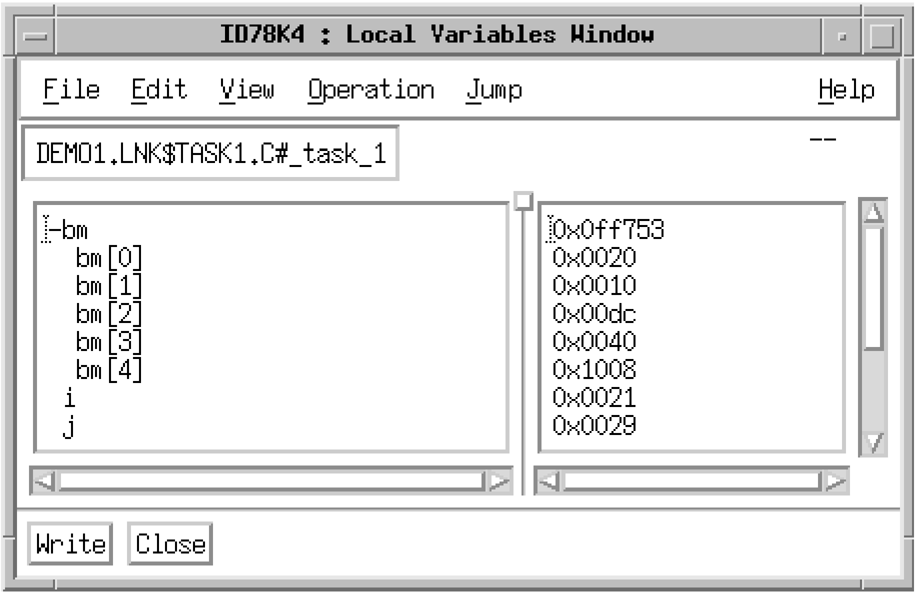
All the local variables in the current function are displayed.

This window can be opened in any of the following ways:

- In the Main window, select Window from the menu bar.
  → Select Local Variable Window from the pull-down menu.
- In the Windows Box window, click .
- In the Source window, select Operation from the menu bar.
  → Select Local Variable Window from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



DEMO1.LNK$TASK1.C#_task_1

```
-bm          0x0ff753
  bm[0]      0x0020
  bm[1]      0x0010
  bm[2]      0x00dc
  bm[3]      0x0040
  bm[4]      0x1008
 i           0x0021
 j           0x0029
```

Description

This window consists of the following components:

Menu bar

Top of client area:	Displays the current program and function names.
Client area Left:	Displays the names of all local variables contained in the current function.
Client area Right:	Displays the values of the local variables.

Button area

The following markings are displayed in the upper right corner of the client area to indicate whether modified values have already been set in the target.

--: Indicates that no values have been modified or that all the modified values have already been set in the target.

**: Indicates that a value is being modified or that a modified value has not yet been set in the target.

The value of a local variable is displayed according to the variable type, as described below.

- The name of a structure or array is preceded by a + or -. A + indicates that the variable represents an address. Double-clicking the variable displays its members or elements individually, indented, with the + being replaced with a "-."
- For an array, all elements of the array are displayed according to the type of the array variable.
- For a structure, all members of the structure are displayed according to the types of the member variables. If a structure is defined within the structure, the tag and variable names of the internal structure are also displayed.
- For an enumeration variable, its member names are displayed.
- For a string pointer variable, a string is displayed, because the variable is regarded as being a pointer to a string.
- For a pointer variable other than character pointer variables, the address stored in the variable is displayed in hexadecimal.
- For a character variable, a character is displayed. A period is displayed to indicate a character code that cannot be displayed.
- For a variable other than those above, a hexadecimal number is displayed.

Each time execution stops, the local variables in the function assumed to be current at that time are displayed.

To save variable values, therefore, place the window in the hold state. To place the window in the hold state, click Operation on the menu bar then click the appropriate radio button in the pull-down menu.

A local variable can be set as follows.

When modifying the value of a variable, note the variable type. If a string constant is specified as a set value, the string is set at the address pointed to by the variable if the variable is a string pointer. \0 is appended to the string.

(1) Selecting a local variable

To select a variable, click the desired variable name or value. The selected variable name is displayed in reverse video.

Clicking the variable name again deselects the variable.

(2) Setting a local variable name

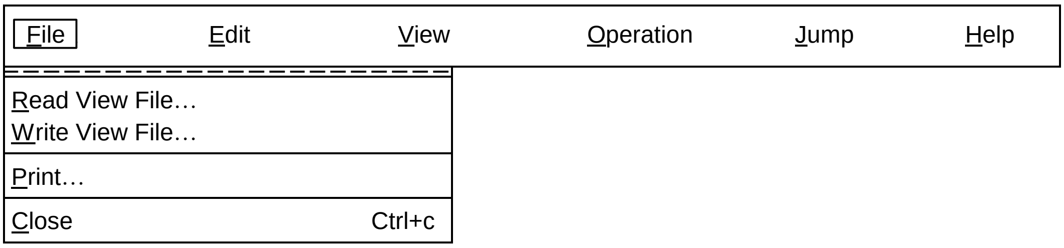
After selecting a local variable as described above, directly modify the value from the keyboard, then click the Write button.

Buttons

Write	Writes the modified value into the target memory.
Close	Closes the Local Variables window..

Menu bar

(1) File



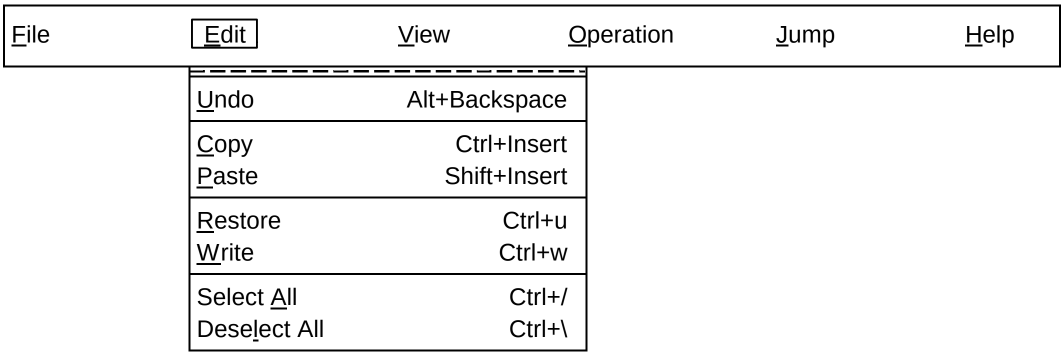
Read View File… Displays the File Selection dialog box (p.104), enabling the specification of the file containing the contents of this window. The contents of the file are read for reproduction. The file to be specified must have the loc extension.

Write View File… Displays the File Selection dialog box (p.104), enabling the specification of the file into which the contents of this window are to be saved. The contents of the window are written into the file. The file to be specified must have the loc extension.

Print… Displays the Prompt dialog box (p.115), enabling the specification of a print command. The contents of the Variables window can be printed.

Close Closes the Local Variables window. This has the same function as the Close button.

(2) Edit



Undo — Undoes the previous edit operation.

String →

Copy — Copies the string, selected with the mouse, into the clipboard buffer.

Paste — Pastes the contents of the clipboard buffer into the text cursor position. The position must have already been selected by double-clicking. Even if the display mode has been changed, the contents of the clipboard buffer are automatically changed according to the current display mode. A string that has been copied in another window cannot be pasted.

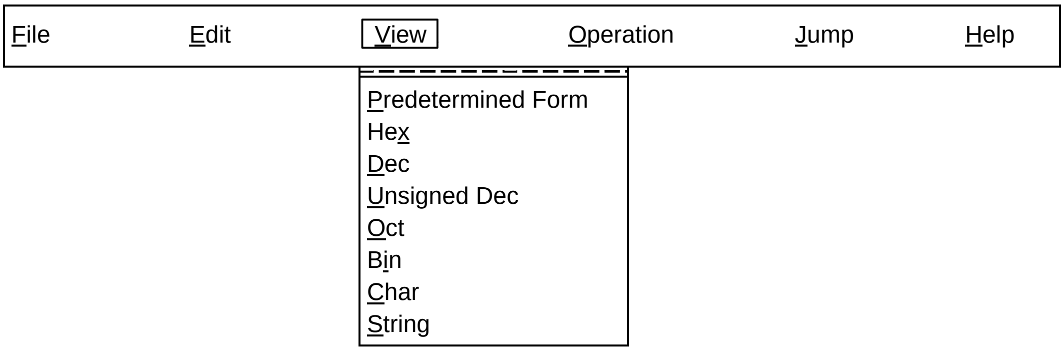
Restore — Abandons the modified value and restores the previous value.

Write — This item can be selected only when a value is modified. It writes the modified value into the target memory. This has the same function as the Write button.

Select All — Selects all displayed variable names.

Deselect All — Deselects all selected variable names.

(3) View



Predetermined Form

Displays variable values in predetermined format.

Hex Displays variable values in hexadecimal.

Dec Displays variable values as signed decimal numbers.

Unsigned Dec Displays variable values as unsigned decimal numbers.

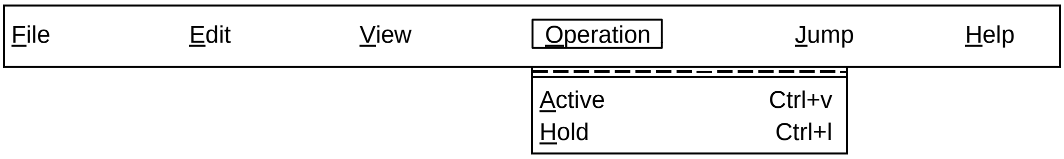
Oct Displays variable values in octal.

Bin Displays variable values in binary.

Char Displays variable values as characters. Characters are enclosed in single quotation marks (').

String Displays the characters, starting from the address pointed to by each variable value to a NULL character (\0), as a string. The character string is enclosed in double quotation marks ("). The maximum number of characters in a string can be specified using Max String Length...

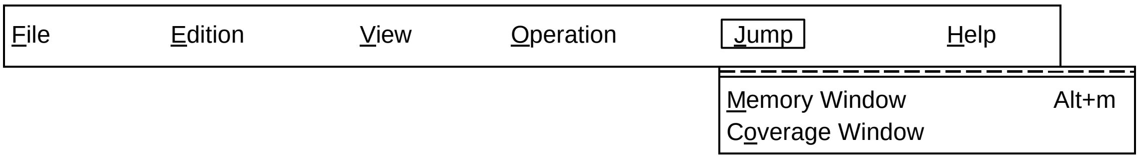
(4) Operation



Active Switches the window from the hold state to the active state.

Hold Switches the window from the active state to the hold state.

(5) Jump



Variable value →

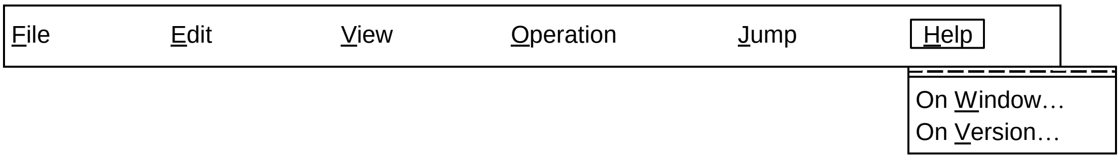
Memory Window Displays the Memory window (p.170), starting from the specified variable value. If the active-state Memory window is already displayed, the contents of the window are overwritten.

Variable value →

Coverage Window

Displays the Coverage window (p.188) with the address of the selected variable.

(6) Help



On Window… Displays the Help dialog box (p.271).

On Version… Displays the Message dialog box (p.131), which shows the version of ID78K4.

Notes

- If the program was compiled with the debug option (-g) not specified, or if only raw data was specified when the load module was loaded into the debugger, a source list is not displayed.
- If a string is specified as a variable and a string longer than the area pointed to by the variable value is specified as a set value, data outside the area may be destroyed.

# Stack Trace window	View window

Outline

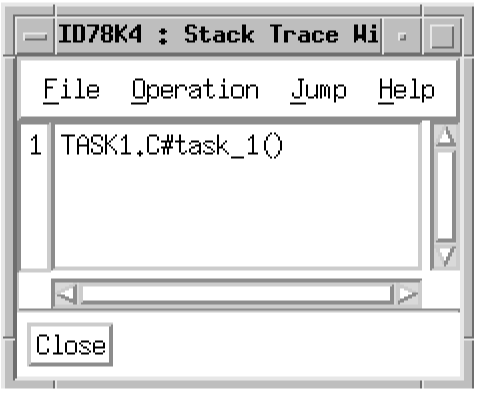
The Stack Trace window displays the contents of a stack. It can also display the source text and disassemble results corresponding to the contents of the stack.

This window can be opened in any of the following ways:

- In the Main window, select Window from the menu bar.
  → Click Stack Trace Window from the pull-down menu.
- In the Windows Box window, click ☰ .
- In the Source window, click a function.
  → Select Operation from the menu bar.
  → Click Stack Trace Window from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



1 TASK1.C#task_1()

Description

This window consists of the following components:

Menu bar

Client area Left:	Displays a stack frame number.
Client area Right:	Displays the contents of the stack, from the current function to the bottom of the stack. Scroll bars are always provided on the right and bottom edges.

Button area

Stack contents are displayed in function-name (parameter) [program-name$file-name:line] format, from the deepest to the shallowest stack nest. If a portion having no symbol information is traced, the [program-name$file-name:line] portion is replaced with [address] because the stop position is an address, not a line number.

A stack frame required for using the menu bar can be selected as follows:

(1) Selecting a stack frame

To select a stack frame, click the stack frame number in the left part of the client area.

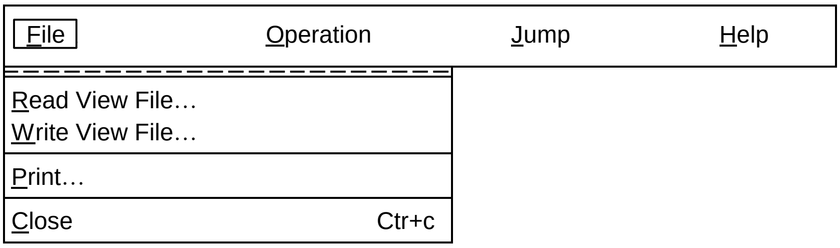
The selected stack frame number is displayed in reverse video. Clicking another frame number or the function button deselects the selected frame.

Button

Close	Closes the Stack Trace window.

Menu bar

(1) File



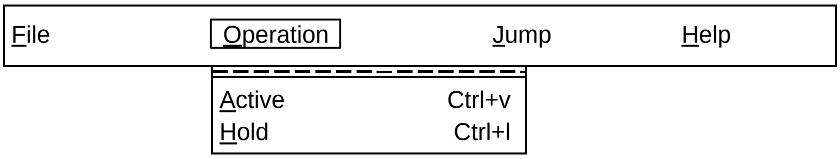
Read View File… Displays the File Selection dialog box (p.104), enabling the specification of the file containing the contents of this window. The contents of the file are read for reproduction. The file to be specified must have the stk extension.

Write View File… Displays the File Selection dialog box (p.104), enabling the specification of the file into which the contents of this window are to be saved. The contents of the window are written into the file. The file to be specified must have the stk extension.

Print… Displays the Prompt dialog box (p.115), enabling the specification of a print command. The contents of the Stack Trace window can be printed.

Close Closes the Stack Trace window.
This has the same function as the Close button.

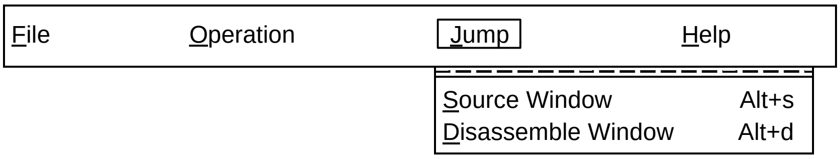
(2) Operation



Active Switches the window from the hold state to the active state.

Hold Switches the window from the active state to the hold state.

(3) Jump



Stack frame number →

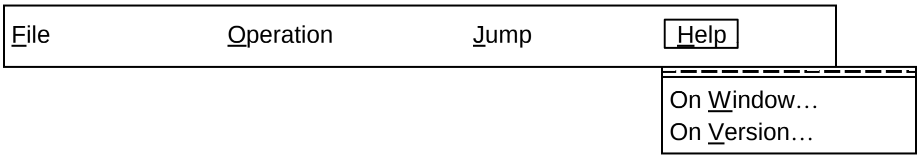
Source Window Displays the Source window (p.120) containing the function corresponding to the specified stack contents. If the active-state Source window is already displayed, the contents of the window are overwritten.

Stack frame number →

Disassemble Window

Displays the Disassemble window (p.201) starting from the address corresponding to the specified stack contents. If the active-state Disassemble window is already displayed, the contents of the window are overwritten.

(4) Help



On Window…	Displays the Help dialog box (p.271).
On Version…	Displays the Message dialog box (p.131), which shows the version of ID78K4.

Notes

- If only raw data was specified when the load module was loaded into the debugger, the contents of a stack are not displayed.
- This window cannot be opened while the CPU is performing emulation.

Registers window

View/setting window

Outline

The Registers window is used to display and modify register values.

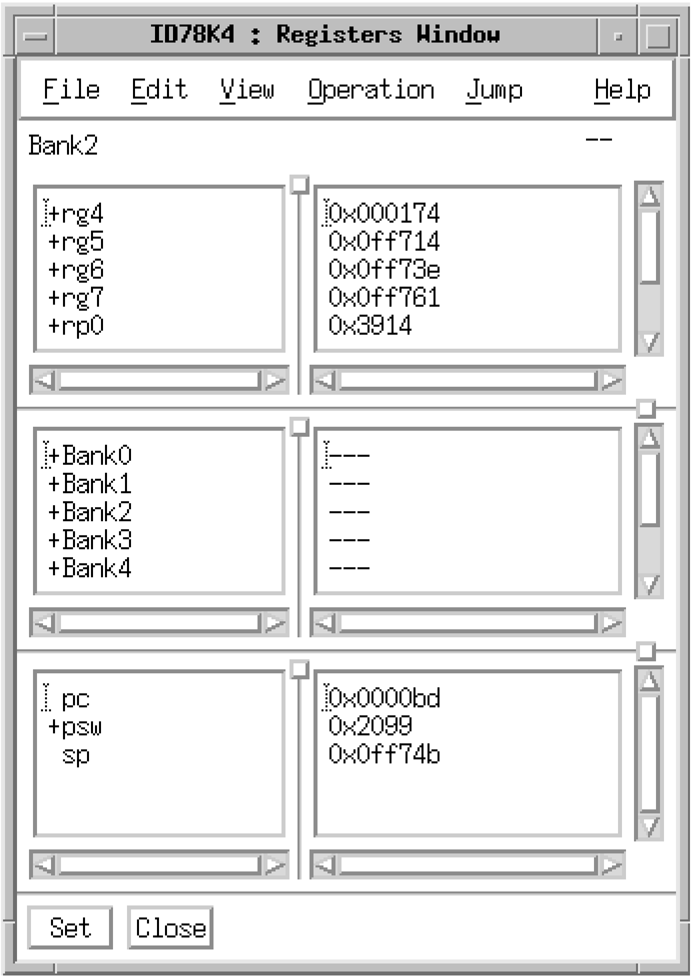
This window can be opened in either of the following ways:

- In the Main window, select Window from the menu bar.

  → Click Register Window on the pull-down menu.
- In the Windows Box window, click .

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



Bank2 --

+rg4	0x000174
+rg5	0x0ff714
+rg6	0x0ff73e
+rg7	0x0ff761
+rp0	0x3914

+Bank0	---
+Bank1	---
+Bank2	---
+Bank3	---
+Bank4	---

pc	0x0000bd
+psw	0x2099
sp	0x0ff74b

Description

This window consists of the following components:

Menu bar	
Top of client area Left:	Displays general-purpose register names.
Top of client area Right:	Displays general-purpose register values.
Middle of client area Left:	Displays the general-purpose registers in each bank.
Middle of client area Right:	Displays the general-purpose register values in each bank.
Bottom of client area Left:	Displays system register names.
Bottom of client area Right:	Displays system register values.
Button area	

The following markings are displayed in the upper right corner of the client area to indicate whether modified values have already been set in the target.

--: Indicates that no values have been modified or that all the modified values have already been set in the target.

**: Indicates that a value is being modified or that a modified value has not yet been set in the target.

A register required for using a function button or the menu bar can be selected and set as follows:

(1) Selecting a register

To select a register, click the name or value of the register. The selected register name is displayed in reverse video. Clicking the selected register name again deselects it.

By double-clicking a register name, the register display mode can be changed from word to byte. The default mode is word.

(2) Setting a register value

After selecting a register value as described above, change the value from the keyboard, then click the Set button.

(3) Displaying/setting a general-purpose register

A general-purpose register name is preceded by a + or -. If a + is displayed, the register is displayed in word units. In this state, double-clicking the register name causes the register to be displayed in byte units. At this time, the + is replaced with a -. Double-clicking a register name preceded by a - causes the register to be displayed in word units.

(4) Displaying/setting the general-purpose registers in each bank

A bank name is preceded by a + or -. If a + displayed, no general-purpose register values are displayed. In this state, double-clicking the bank name causes the values of the general-purpose registers in the bank to be displayed. At this time, the + is replaced with a -. Double-clicking a bank name preceded by a - suppresses the display of the general-purpose register values.

(5) Displaying/setting fields

The name of a register having meaningful fields, such as PSW, is preceded by a + or -. Double-clicking a register name preceded by a + displays each of the fields, indented, with the + being replaced with a -. Double-clicking a register name preceded by a - suppresses the display of the fields, replacing the - with a +. A field can be selected and set in the same way as a register.

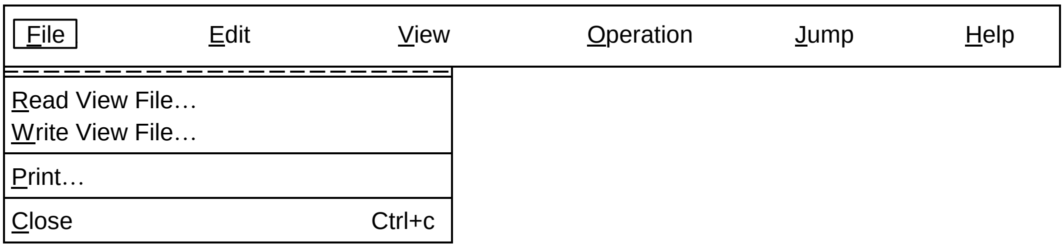
Buttons

Set   Writes the modified value into the target memory.

Close   Closes the Registers window.

Menu bar

(1) File



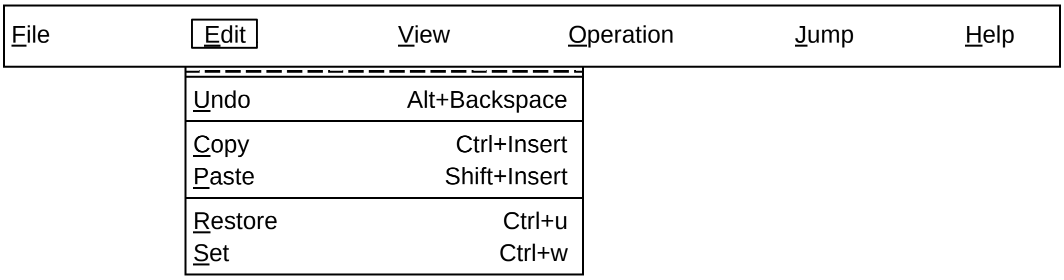
Read View File… Displays the File Selection dialog box (p.104), enabling the specification of the file containing the contents of this window. The contents of the file are read for reproduction. The file to be specified must have the reg extension.

Write View File… Displays the File Selection dialog box (p.104), enabling the specification of the file into which the contents of this window are to be saved. The contents of the window are written into the file. The file to be specified must have the reg extension.

Print… Displays the Prompt dialog box (p.115), enabling the specification of a print command. The contents of the Variables window can be printed.

Close Closes the Registers window. This has the same function as the Close button.

(2) Edit



Undo — Undoes the previous edit operation.

String →

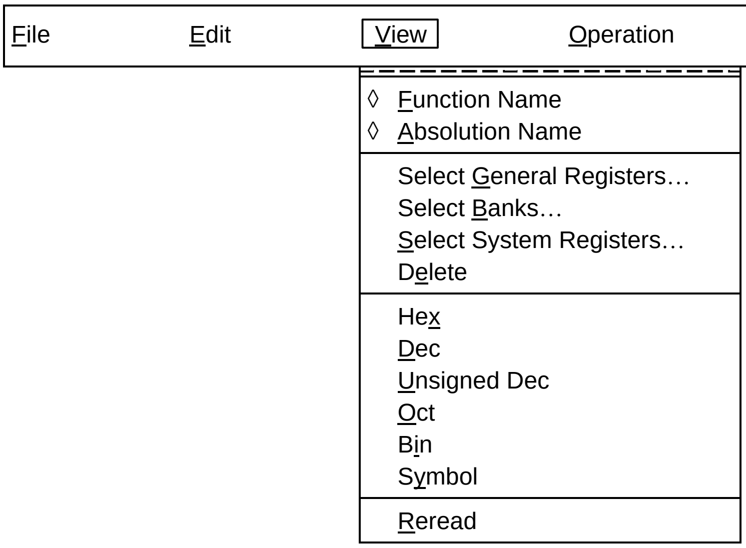
Copy — Copies the string, selected with the mouse, to the clipboard buffer.

Paste — Pastes the contents of the clipboard buffer at the text cursor position. The position must have already been selected by double-clicking. Even if the display mode has been changed, the contents of the clipboard buffer are automatically changed according to the current display mode. A string that has been copied in another window cannot be pasted.

Restore — Abandons the modified value and restores the previous value.

Set — Writes the modified value into the target memory. This has the same function as the Set button.

(3) View



The following two items are provided as radio buttons on the menu:

Function Name Displays register names as functional names.

Absolution Name Displays register names as absolute names.

Select General Registers…

Displays the Items Selection dialog box (p.111), enabling the selection of the general-purpose register to be displayed in the Registers window.

Select Banks… Displays the Items Selection dialog box (p.111), enabling the selection of the bank for which general-purpose registers are to be displayed in the Registers window.

Select System Registers…

Displays the Items Selection dialog box (p.111), enabling the selection of the system register to be displayed in the Registers window.

Delete Deletes a selected register from the Registers window.

Hex Displays variable values in hexadecimal.

Dec Displays variable values as signed decimal numbers.

Unsigned Dec Displays variable values as unsigned decimal numbers.

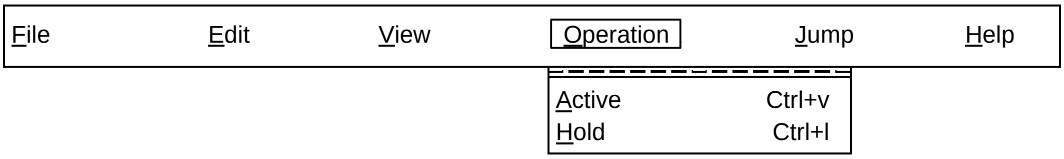
Oct Displays variable values in octal.

Bin Displays variable values in binary.

Symbol Displays variable values as symbols + offsets.

Reread Rereads the values of the displayed registers.

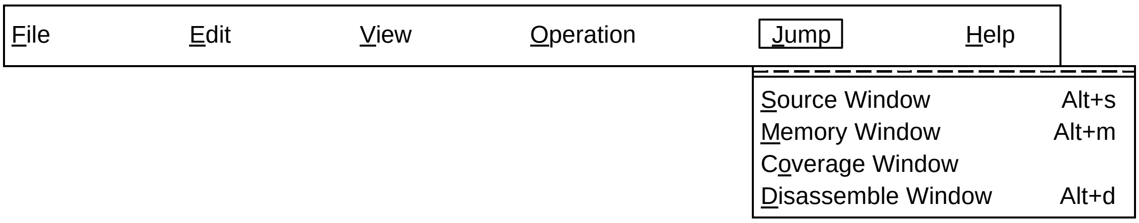
(4) Operation



Active Switches the window from the hold state to the active state.

Hold Switches the window from the active state to the hold state.

(5) Jump



Select a register →

Source Window Displays the Source window (p.120) if source text corresponding to the specified register value exists. If the active-state Source window is already displayed, the contents of the window are overwritten.

Select a register →

Memory Window Displays the Memory window (p.170) starting from the specified register value. If the active-state Memory window is already displayed, the contents of the window are overwritten.

Select a register →

Coverage Window

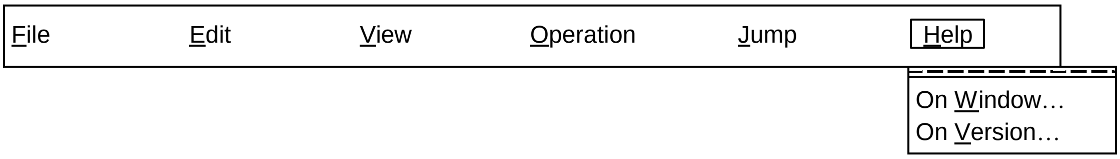
Displays the Coverage window (p.188) starting from the specified register value. If the active-state Memory window is already displayed, the contents of the window are overwritten.

Select a register →

Disassemble Window

Displays the Disassemble window (p.201) starting from the specified register value. If the active-state Disassemble window is already displayed, the contents of the window are overwritten.

(6) Help



On Window…	Displays the Help dialog box (p.271).

On Version…	Displays the Message dialog box (p.131), which shows the version of ID78K4.

Notes

- This window cannot be opened while the CPU is performing emulation.

SFR window

View/setting window

Outline

The SFR window is used to display and modify SFR values.

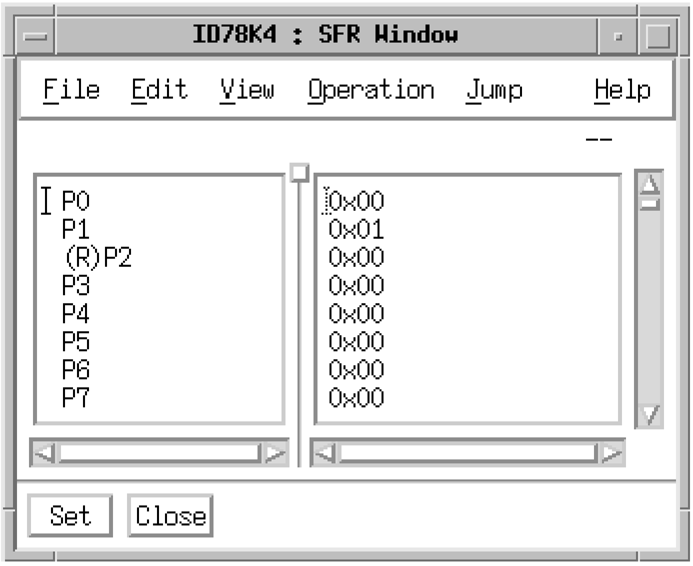
This window can be opened in either of the following ways:

- In the Main window, select Window from the menu bar.

  → Click SFR Window from the pull-down menu.
- In the Windows Box window, click .

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



```
P0      0x00
P1      0x01
(R)P2   0x00
P3      0x00
P4      0x00
P5      0x00
P6      0x00
P7      0x00
```

Description

This window consists of the following components:

Menu bar
Client area Left: Displays SFR names.
Client area Right: Displays SFR values.
Button area

The following markings are displayed at the right corner of the client area to indicate whether modified values have already been set in the target.

--: Indicates that no values have been modified or that all the modified values have already been set in the target.

**: Indicates that a value is being modified or that a modified value has not yet been set in the target.

An SFR required for using a function button or the menu bar can be selected and set as follows:

(1) Selecting an SFR

To select an SFR, click the name or value of the SFR. The selected SFR name is displayed in reverse video. Clicking the selected SFR name again deselects it.

(2) Setting an SFR value

After selecting an SFR value as described above, change the value from the keyboard, then click the Set button.

(3) Displaying/setting fields

The name of an SFR having meaningful fields is preceded by a + or -. Double-clicking an SFR name preceded by a + displays each of the fields, indented, with the + being replaced with a -. Double-clicking an SFR name preceded by a - suppresses the display of the fields, replacing the - with a +. A field can be selected and set in the same way as a register.

(4) Displaying the attribute

The names of SFRs that are read only, write only, and whose values change when they are read are preceded by the following symbols, enclosed in ():

R: Read only SFRs

W: Write only SFRs

V: SFRs whose values change when they are forcibly read

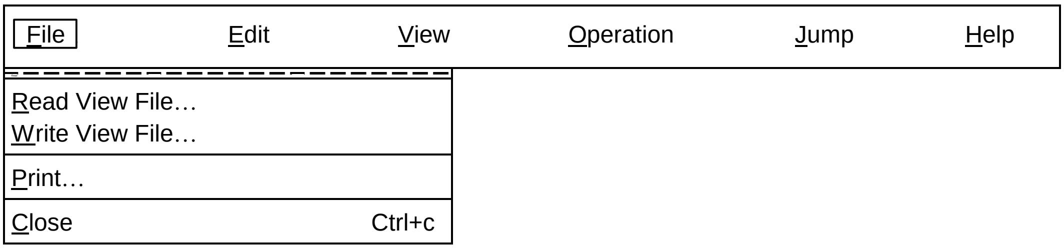
Buttons

Set Writes the modified value into the target memory.

Close Closes the SFR window.

Menu bar

(1) File



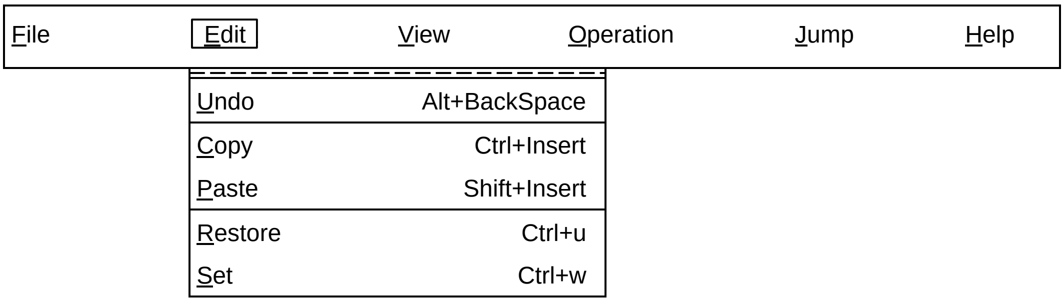
Read View File… Displays the File Selection dialog box (p.104), enabling the specification of the file containing the contents of this window. The contents of the file are read for reproduction. The file to be specified must have the sfr extension.

Write View File… Displays the File Selection dialog box (p.104), enabling specification of the file into which the contents of this window are to be saved. The contents of the window are written into the file. The file to be specified must have an sfr extension.

Print… Displays the Prompt dialog box (p.115), enabling the specification of a print command. The contents of the SFR window can be printed.

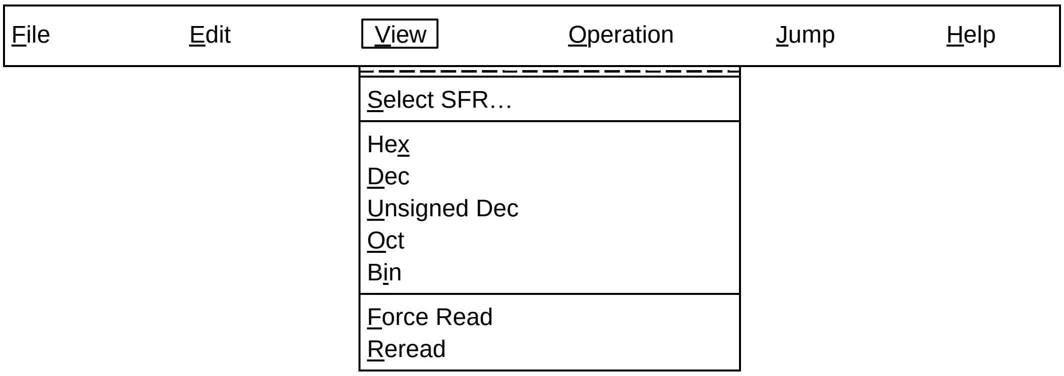
Close Closes the SFR window. This has the same function as the Close button.

(2) Edit



Undo	Undoes the previous edit operation.
String →	
Copy	Copies the string, selected with the mouse, to the clipboard buffer.
Paste	Pastes the contents of the clipboard buffer to the text cursor position. The position must have already been selected by double-clicking. Even if the display mode has been changed, the contents of the clipboard buffer are automatically changed according to the current display mode. A string that has been copied in another window cannot be pasted.
Restore	Abandons the modified value and restores the previous value.
Set	Writes the modified value into the target memory. This has the same function as the Set button.

(3) View



Select SFR… Displays the Items Selection dialog box (p.111), enabling the selection of the SFR to be displayed in the SFR window.

Hex Displays variable values in hexadecimal.

Dec Displays variable values as signed decimal numbers.

Unsigned Dec Displays variable values as unsigned decimal numbers.

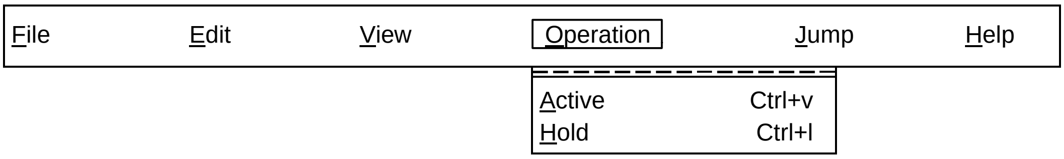
Oct Displays variable values in octal.

Bin Displays variable values in binary.

Force Read Forcibly reads the selected SFRs.

Reread Rereads the values of the displayed SFRs.

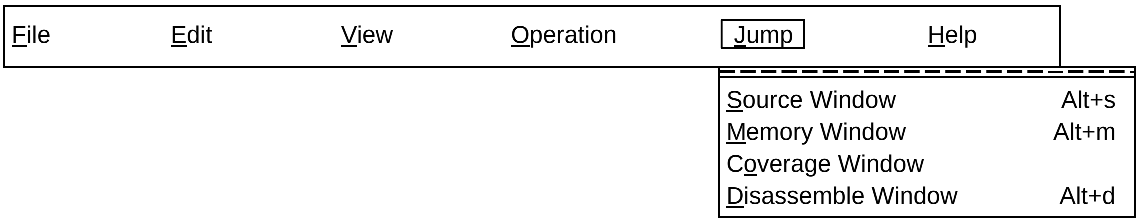
(4) Operation



Active Switches the window from the hold state to the active state.

Hold Switches the window from the active state to the hold state.

(5) Jump



Select an SFR →

Source Window — Jumps to the Source window (p.120) corresponding to the specified SFR. If an SFR has not been selected, the system beeps. If multiple SFRs are selected, the first SFR on the selected list is assumed.

Select an SFR →

Memory Window — Jumps to the Memory window (p.170) corresponding to the specified SFR. If an SFR has not been selected, the system beeps. If multiple SFRs are selected, the first SFR on the selected list is assumed.

Select an SFR →

Coverage Window

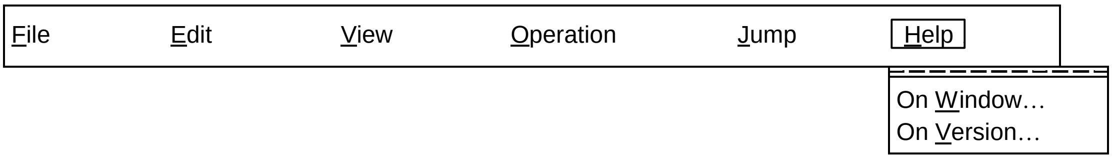
Jumps to the Coverage window (p.188) corresponding to the specified SFR. If an SFR has not been selected, the system beeps. If multiple SFRs are selected, the first SFR on the selected list is assumed.

Select an SFR →

Disassemble Window

Jumps to the Disassemble window (p.201) corresponding to the specified SFR. If an SFR has not been selected, the system beeps. If multiple SFRs are selected, the first SFR on the selected list is assumed.

(6) Help



On Window… — Displays the Help dialog box (p.271).

On Version… — Displays the Message dialog box (p.131), which shows the version of ID78K4.

Notes

- This window cannot be opened while the CPU is performing emulation.

Memory window

View/setting window

Outline

The Memory window is used to display and modify the contents of memory. This window can also be used to search memory for specified contents.

This window can be opened by any of the following operations:

- In the Main window, select Window from the menu bar.

  → Select Memory Window from the pull-down menu.

  → Specify an address in the Prompt dialog box.
  → Click the OK button in the Prompt dialog box.

- In the Windows Box window, click 🔍.

  → Specify an address in the Prompt dialog box.
  → Click the OK button in the Prompt dialog box.

- In the Source window, select a line number.

  → Select Jump from the menu bar.

  → Select Memory Window from the pull-down menu.

- In the Variables window, select a variable.

  → Select Jump from the menu bar.

  → Select Memory Window from the pull-down menu.

- In the Local Variables window, select a local variable.

  → Select Jump from the menu bar.

  → Select Memory Window from the pull-down menu.

- In the Registers window, select a register.

  → Select Jump from the menu bar.

  → Select Memory Window from the pull-down menu.

- In the SFR window, select an SFR.

  → Select Jump from the menu bar.

  → Select Memory Window from the pull-down menu.

- In the Coverage window, select an address.

  → Select Jump from the menu bar.

  → Select Memory Window from the pull-down menu.

- In the Disassemble window, select an address.

  → Select Jump from the menu bar.

  → Select Memory Window from the pull-down menu.

- In the Event Manager window, select an event.

  → Select Jump from the menu bar.

  → Select Memory Window from the pull-down menu.

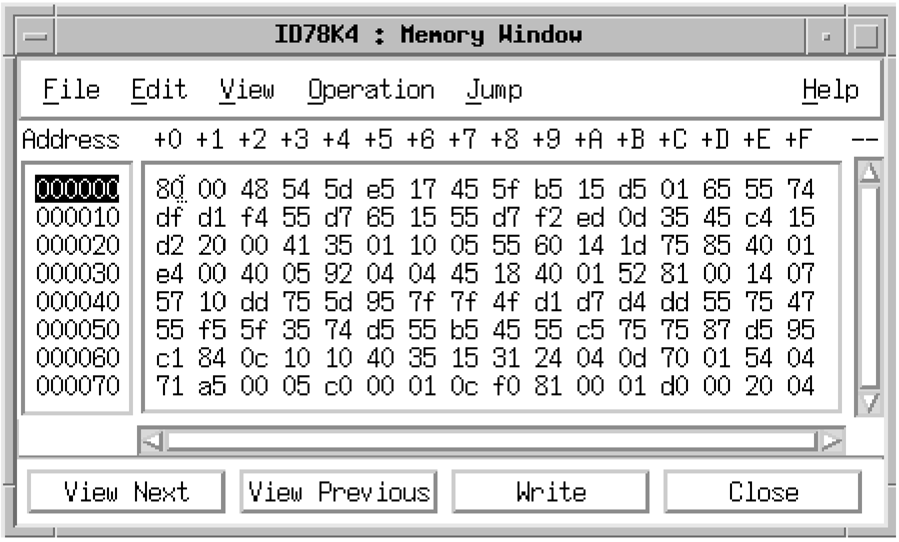
- In the Trace View window,

  → Select Jump from the menu bar.

  → Select Memory Window from the pull-down menu.

Resetting the debugger by using the Main window (with CPU & BODY specified) closes this window.

Window



```
Address  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F  --
000000   80 00 48 54 5d e5 17 45 5f b5 15 d5 01 65 55 74
000010   df d1 f4 55 d7 65 15 55 d7 f2 ed 0d 35 45 c4 15
000020   d2 20 00 41 35 01 10 05 55 60 14 1d 75 85 40 01
000030   e4 00 40 05 92 04 04 45 18 40 01 52 81 00 14 07
000040   57 10 dd 75 5d 95 7f 7f 4f d1 d7 d4 dd 55 75 47
000050   55 f5 5f 35 74 d5 55 b5 45 55 c5 75 75 87 d5 95
000060   c1 84 0c 10 10 40 35 15 31 24 04 0d 70 01 54 04
000070   71 a5 00 05 c0 00 01 0c f0 81 00 01 d0 00 20 04
```

Description

This dialog box consists of the following components:

Menu bar	
Top of client area:	Displays character strings indicating address offsets, as a header.
Middle of client area Left:	Displays addresses starting from that specified in the Prompt dialog box.
Middle of client area Right:	Displays the contents of memory corresponding to the addresses. This area has scroll bars on the right and bottom.
Button area	

The following markings are displayed at the upper right corner of the client area to indicate whether modified values have already been set in the target:

--: Indicates that no values have been modified or all the modified values have already been set in the target.

**: Indicates that any value is being modified or any modified value has not yet been set in the target.

An address or memory value required for using a function button or the menu bar can be selected as follows:

(1) Selecting an address

To select an address, click it in the left part of the client area. The selected address is displayed in reverse video. Clicking a blank area or another address deselects the selected address.

(2) Selecting a memory value

To select a memory value (string in the text), click it in the right part of the client area. The selected memory value is displayed in reverse video. Clicking a blank area or another memory value deselects the selected memory value.

To modify the contents of memory, select the memory value to be modified, then enter a new value by using the keyboard. Clicking the Write button writes the new value, which is reflected to each relevant window.

Buttons

View Next — Scrolls the contents of memory by (the number of displayed lines - 1) lines. When the end of the read results is being displayed, clicking this button reads the next contents of memory.

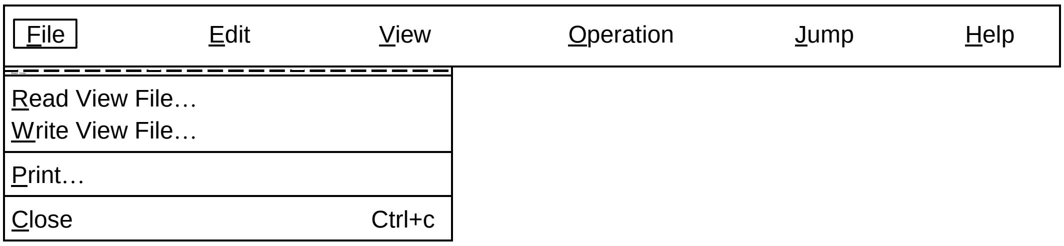
View Previous — Scrolls the contents of memory backwards by (the number of displayed lines - 1) lines.

Write — Writes the input values to the target memory.

Close — Closes the Memory window.

Menu bar

(1) File



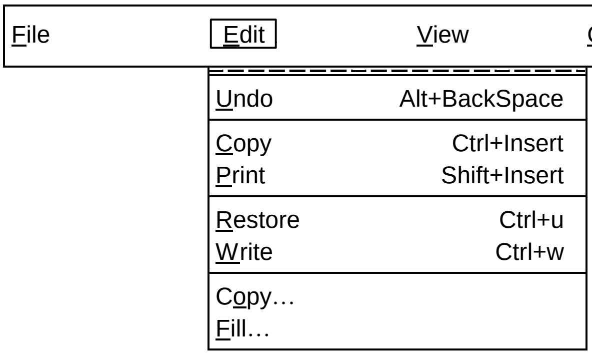
Read View File... Displays the File Selection dialog box (p.104) to specify the file in which the window contents are saved. Reads and reproduces the contents of the file. Specify "mem" as the file extension.

Write View File... Displays the File Selection dialog box (p.104) to specify the file into which the window contents will be saved. Writes the contents of the window into the file. Specify "mem" as the file extension.

Print... Displays the Prompt dialog box (p.115) to specify the print command. Prints the contents of the Variables window.

Close Closes the Variables window. Has the same function as the Close button.

(2) Edit



Undo Undoes the previous edit operation.

Character string →

Copy Copies the character string, selected with the mouse, into the clipboard buffer.

Paste Pastes the contents of the clipboard at the text cursor position. The paste position must be selected in advance by double-clicking. When pasting, even if the view form is modified, the contents of the clipboard are automatically pasted in the current view form. Character strings copied from other windows cannot, however, be pasted.

Restore Restores the modified value.

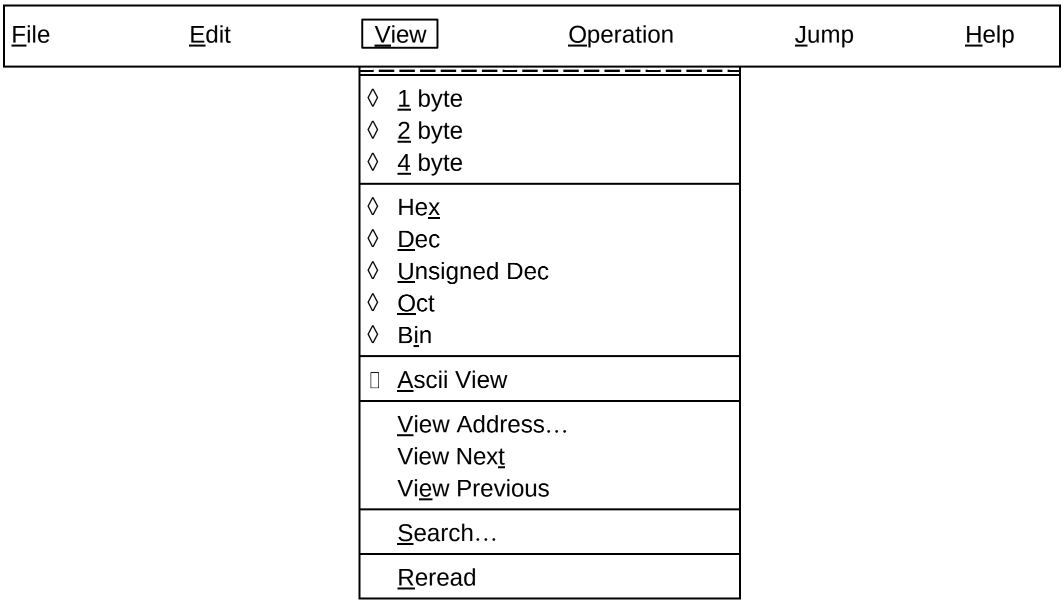
Write Writes the modified value into the target. Has the same function as the Write button.

Address →

Copy… Displays the Memory dialog box (p.179) to specify the memory address range to be replaced. When an address is selected, the address is displayed as the transfer source address of memory copy in the dialog box.

Fill… Displays the Memory dialog box (p.179) to specify the memory file address range and the fill list. When the address is selected, the address is displayed as the start address of memory file in the dialog box.

(3) View



In the menu, the following three items are configured as radio buttons.

1 byte	Displays the contents of memory in one-byte units.
2 byte	Displays the contents of memory in word units.
4 byte	Displays the contents of memory in double-word units.

The following five items are configured as radio buttons in the menu.

Hex	Displays the variable as a hexadecimal value.
Dec	Displays the variable as a signed decimal value.
Unsigned Dec	Displays the variable as an unsigned decimal value.
Oct	Displays the variable as an octal number.
Bin	Displays the variable as a binary.

In the menu, the following items are configured as toggle buttons.

Ascii View — Adds the Ascii display. Can be selected only when the display unit is one byte and the display format is hexadecimal.

View Address… — Displays the Prompt dialog box (p.115) to specify the address and redisplay the Memory window (p.170).

View Next — Scrolls the contents of memory down by one less than the number of displayed lines. When the end of the read results is displayed, clicking this button reads the next contents of memory. Has the same function as the View Next button.

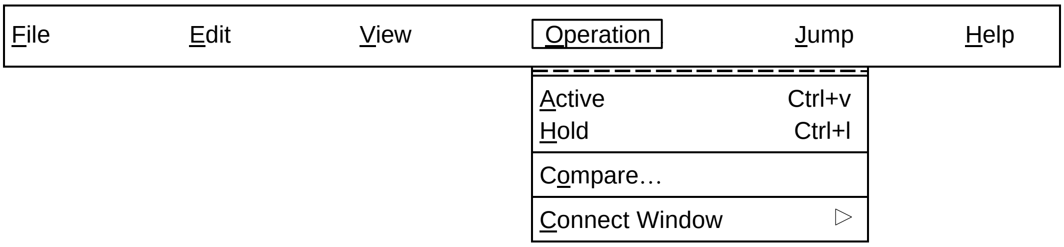
View Previous — Scrolls the contents of memory up by one less than the number of displayed lines. Has the same function as the View Previous button.

Address →

Search… — Displays the Memory dialog box (p.179) to specify the range of the address to be searched. When the address is selected, the address is displayed as the start address for memory search in the dialog box.

Reread — Rereads from the address for window display and displays the data.

(4) Operation



Active — Switches the window from hold status to active status.

Hold — Switches the window from active status to hold status.

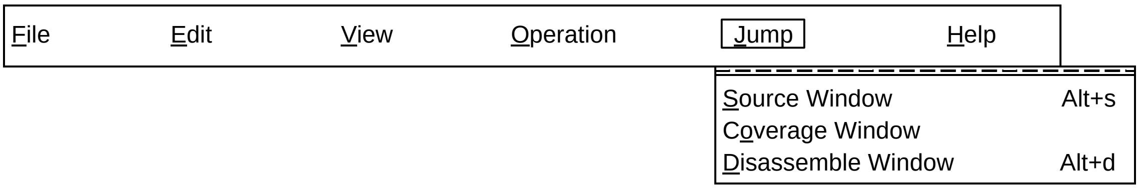
Address →

Compare… — Displays the Memory dialog box (p.179) to specify the memory address to be compared. When an address is selected, the address is displayed as the address from which memory comparison will start in the dialog box.

Connect Window — Displays the submenu to connect windows. In the submenu, the following windows can be selected using the toggle buttons. Select the windows to be connected.

- Source window
- Coverage window
- Disassemble window

(5) Jump



Select an address →

Source Window Displays the corresponding Source window (p.120) when source text that matches the specified address value exists. When the active Source window is already displayed, however, the contents of the displayed window are rewritten.

Select an address →

Coverage Window

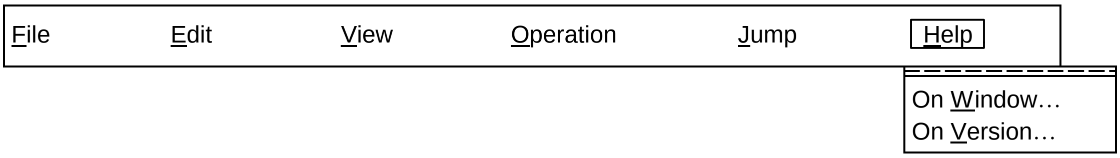
Displays the Coverage window (p.188), starting from the specified address value. When the active Memory window is already displayed, however, the contents of the displayed window are rewritten.

Select an address →

Disassemble Window

Displays the Disassemble window (p.201), starting from the specified address value. When the active Memory window is already displayed, however, the contents of the displayed window are rewritten.

(6) Help



On Window… Displays the Help dialog box (p.271).

On Version… Displays the Message dialog box (p.131) that indicates the ID78K4 version.

Notes

- When the range of compared addresses is large, depending on the amount of memory installed in the system that activated the debugger, the processing may take an abnormally long time. The range to be compared should not exceed 1 kilobyte.
- This window cannot operate while the emulation CPU is operating.

Errors

- If an attempt is made to display another window displayed from the Jump menu without first selecting an address, the system beeps.

Memory dialog box

Auxiliary dialog box

Outline

The Memory dialog box is used to manipulate the memory in the Memory window. The Memory dialog box is displayed in the following conditions:

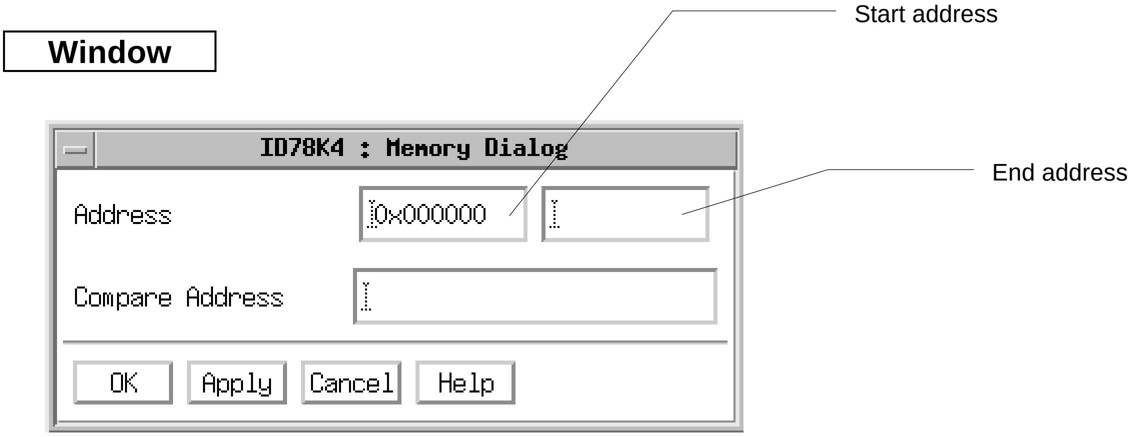
(1) When the contents of memory are copied

(2) When the contents of memory are padded with the list

(3) When the contents of memory are searched

(4) When the contents of memory are compared

Window



Description

The window consists of the following components.

Top of client area: Area for specifying the address range

Bottom of client area: Area for specifying the address or list

The address or list to be specified is input from the keyboard. When an address has been selected in the Memory window, that address is displayed as the start address. The address expression can be specified as the address.

Each function is described below.

(1) When the contents of memory are copied

Copies the contents of memory from the specified address area to the specified copy destination address. The contents of the Memory window (p.170) are modified accordingly.

(2) When the contents of memory are filled with the list

Fills the specified address area with the specified list.

The contents of the Memory window (p.170) are modified accordingly.

(3) When the contents of memory are searched

Searches the specified address range for the specified list.

There are two search methods: Search Forward and Search Backward. After the search list has been specified, forward search is performed by clicking the Search Forward button, while backward search is performed by clicking the Search Backward button.

An item located by the search is displayed in the Memory window (p.170) in reverse video.

(4) When the contents of memory are compared

Compares the contents of memory in the specified address range with the contents of memory in the specified compare destination address range.

The results are displayed in the Result dialog box (p.182). As a result of the comparison, however, if the contents of the memory ranges are found to be the same, the Result dialog box is not displayed. When the range of compared addresses is large, depending on the amount of memory installed in the system that activated the debugger, the debugger may terminate abnormally. The specified range should not exceed 1 kilobyte.

Buttons

OK — Executes the function of the Memory dialog box. Closes the Memory dialog box after execution.

Apply — Executes the function of the Memory dialog box. Leaves the Memory dialog box open even after execution

Cancel — Closes the Memory dialog box without any processing.

Help — Displays the Help dialog box (p.271).

Notes

- The fill list is always sorted by byte size. When specifying numerics other than character constants and string constants in a fill list, therefore, specify them in one-byte units.
- The total data size set in the fill list must be an integer multiple of the memory access size.
- When specifying character constants and string constants in the fill list, search list, escape with a backslash (\) to prevent the shell from interpreting single quotes (') and double quotes (").
- This window cannot operate while the emulation CPU is operating.

Errors

- If the OK button is clicked while no load module is selected, the system beeps.

Result dialog box

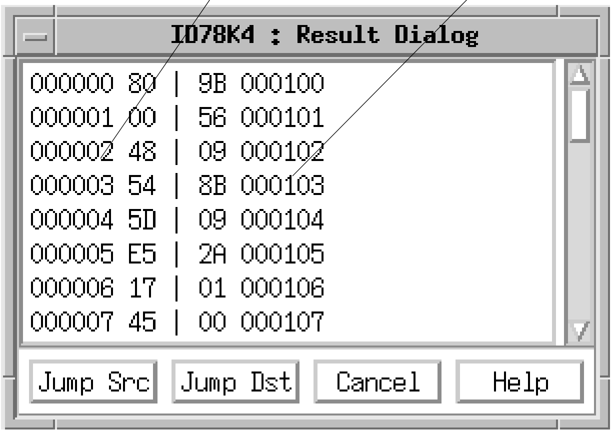
Selection dialog box

Outline

The Result dialog box is used to display the results of comparing the contents of memory. Addresses that differ are searched and displayed in the Memory window.

If, however, as a result of comparison, the contents of memories are the same, the Result dialog box is not displayed.

Window



Comparison source

Comparison destination

```
000000 80 | 9B 000100
000001 00 | 56 000101
000002 48 | 09 000102
000003 54 | 8B 000103
000004 5D | 09 000104
000005 E5 | 2A 000105
000006 17 | 01 000106
000007 45 | 00 000107
```

Description

The Result dialog box consists of the following components.

Client area Left: Displays a comparison source address that was found to differ as a result of comparing the contents of memories.

Client area Right: Displays a comparison destination address that was found to differ as a result of comparing the contents of memories.

Middle of client area: Displays the contents of the memory corresponding to the addresses.

To select an address, click the line to be compared. The selected line is displayed in reverse video.

After the line to be compared has been selected, clicking the Jump Src button redisplays the Memory window (p.170) starting from the source address of the specified line and displays, in reverse video, the contents of memory that corresponds to the source address.

Clicking the Jump Dst button redisplays the Memory window (p.170) starting from the destination address of the specified line and displays, in reverse video, the contents of memory that corresponds to the destination address.

This dialog box remains displayed until the Cancel button is clicked.

Buttons

Compared line →

Jump Src — Redisplays the Memory window (p.170) starting from the source address of the specified line and displays, in reverse video, the contents of memory that corresponds to the source address.

Compared line →

Jump Dst — Redisplays the Memory window (p.170) starting from the destination address of the specified line and displays, in reverse video, the contents of memory that corresponds to the destination address.

Cancel — Closes the Result dialog box.

Help — Displays the Help dialog box (p.271).

Notes

- None

Errors

- If the Jump Src or Jump Dst button is clicked without specifying an address, the system beeps.

Realtime RAM Tracer window

View/setting window

Outline

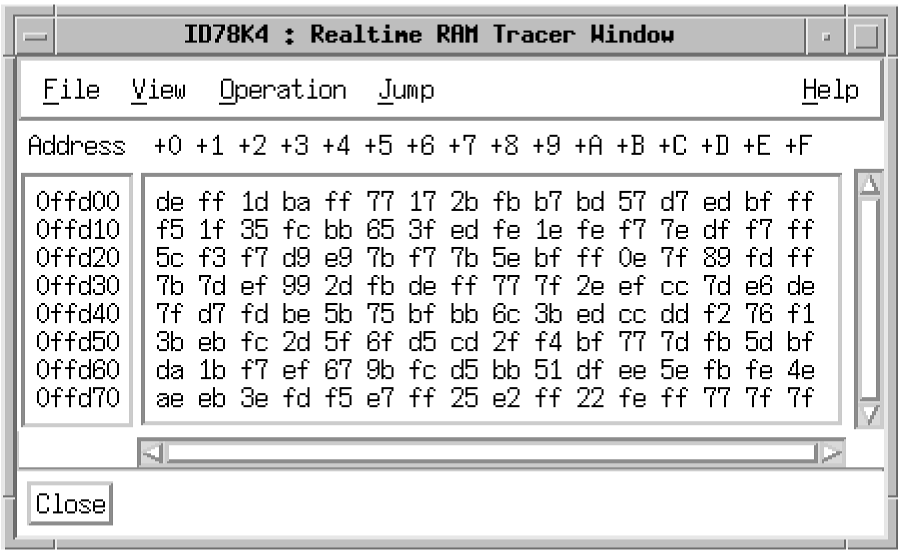
The Realtime RAM Tracer window is used to display and update the specified internal RAM area in real time.

This window can be displayed in any of the following ways.

- In the Main window, select Window from the menu bar.
  → Select Realtime RAM Tracer Window from the pull-down menu.
  → Specify an address in the Prompt dialog box.
  → Click the OK button in the Prompt dialog box.
- Select from the Windows Box window.
  → Specify an address in the Prompt dialog box.
  → Click the OK button in the Prompt dialog box.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



```
Address  +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +A +B +C +D +E +F
0ffd00   de ff 1d ba ff 77 17 2b fb b7 bd 57 d7 ed bf ff
0ffd10   f5 1f 35 fc bb 65 3f ed fe 1e fe f7 7e df f7 ff
0ffd20   5c f3 f7 d9 e9 7b f7 7b 5e bf ff 0e 7f 89 fd ff
0ffd30   7b 7d ef 99 2d fb de ff 77 7f 2e ef cc 7d e6 de
0ffd40   7f d7 fd be 5b 75 bf bb 6c 3b ed cc dd f2 76 f1
0ffd50   3b eb fc 2d 5f 6f d5 cd 2f f4 bf 77 7d fb 5d bf
0ffd60   da 1b f7 ef 67 9b fc d5 bb 51 df ee 5e fb fe 4e
0ffd70   ae eb 3e fd f5 e7 ff 25 e2 ff 22 fe ff 77 7f 7f
```

Description

The Realtime RAM Tracer window consists of the following components.

Menu bar

Top of client area: Displays, as a header, character strings indicating address offsets.

Middle of client area Left: Displays addresses starting from that specified in the Prompt dialog box.

Middle of client area Right: Displays the contents of memory corresponding to the addresses.
This area has scroll bars along its right and bottom edges.

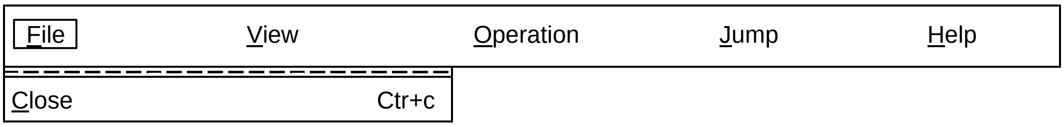
Button area

Button

Close Closes the Realtime RAM Tracer window.

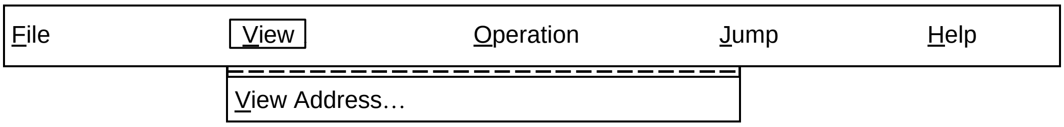
Menu bar

(1) File



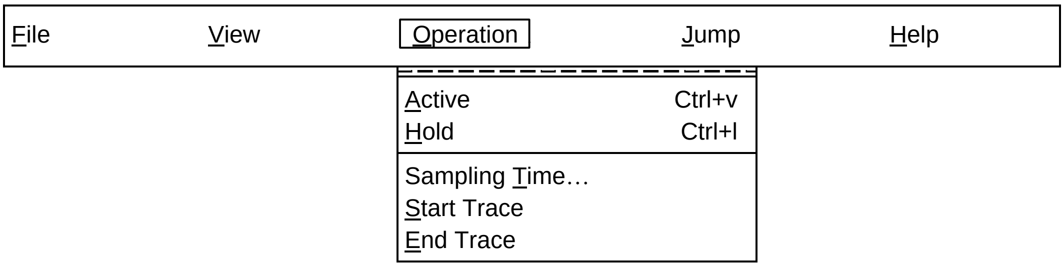
Close Closes the Realtime RAM Tracer window. Has the same function as the Close button.

(2) View



View Address… Displays the Prompt dialog box (p.115) to specify the address from which the Realtime RAM Trace window data is displayed.

(3) Operation



Active Switches the window from hold status to active status.

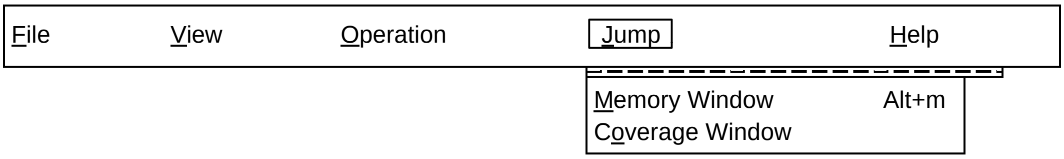
Hold Switches the window from active status to hold status.

Sampling Time… Displays the Prompt dialog box to set the sampling time. Specify any value, in ms units. The default is every 500 ms. The minimum value varies with the host and window status. But, as far as possible, sampling is performed at the specified time intervals.

Start Trace Starts real-time RAM tracing. Note, however, that real-time RAM tracing is not executed unless the program is running.

End Trace Terminates real-time RAM tracing.

(4) Jump



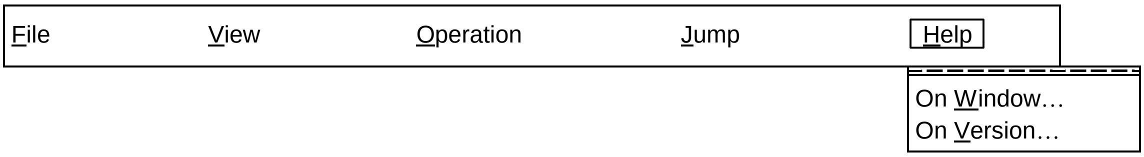
Address →

Memory Window Displays the Memory window (p.170), starting from the specified address. When the active Memory window is already displayed, however, the contents of the displayed, however, window are rewritten.

Address →

Coverage Window Displays the Coverage window (p.188), starting from the specified address value. When the active Memory window is already displayed, however, the contents of the displayed window are rewritten.

(5) Help



On Window… Displays the Help dialog box (p.271).

On Version… Displays the Message dialog box (p.131) that indicates the ID78K4 version.

Errors

- If an attempt is made to display another window from the Jump menu without first selecting an address, the system beeps.
- If real-time RAM tracing is started when it is not running, the system beeps and displays a warning message on the screen.
- If real-time RAM tracing is terminated when it is not running, the system beeps and displays a warning message on the screen.

Coverage window

View/setting window

Outline

The Coverage window is used to display the coverage results and perform initialization. This window can also be used to search the memory for specified contents.

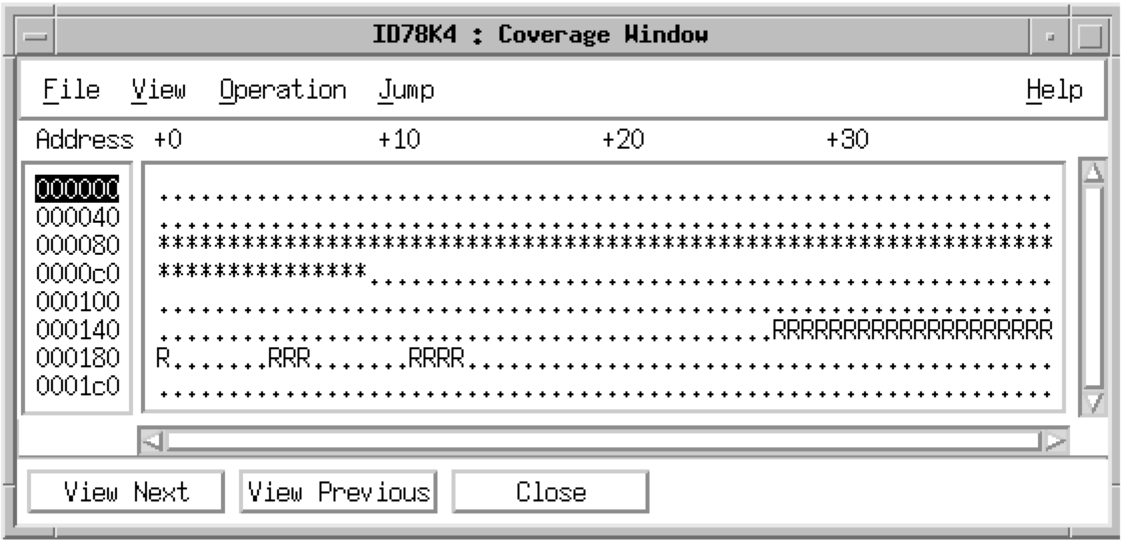
This window can be displayed in any of the following ways.

- In the Main window, select Window from the menu bar.
  → Select Coverage Window from the pull-down menu.
  → Specify an address in the Prompt dialog box.
  → Click the OK button in the Prompt dialog box.
- Select from the Windows Box window.
  → Specify an address in the Prompt dialog box.
  → Click the OK button in the Prompt dialog box.
- In the Source window, select a line number.
  → Select Jump from the menu bar.
  → Select Coverage Window from the pull-down menu.
- In the Variables window, select a variable.
  → Select Jump from the menu bar.
  → Select Coverage Window from the pull-down menu.
- In the Local Variables window, select a local variable.
  → Select Jump from the menu bar.
  → Select Coverage Window from the pull-down menu.
- In the Registers window, select a register.
  → Select Jump from the menu bar.
  → Select Coverage Window from the pull-down menu.
- In the SFR window, select an SFR.
  → Select Jump from the menu bar.
  → Select Coverage Window from the pull-down menu.
- In the Memory window, select an address.
  → Select Jump from the menu bar.
  → Select Coverage Window from the pull-down menu.

- In the Disassemble window, select an address.
  → Select Jump from the menu bar.
  → Select Coverage Window from the pull-down menu.
- In the Event Manager window, select an event.
  → Select Jump from the menu bar.
  → Select Coverage Window from the pull-down menu.
- In the Trace View window,
  → Select Jump from the menu bar.
  → Select Coverage Window from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



```
Address +0              +10             +20             +30
000000  ................................................................
000040  ................................................................
000080  ****************************************************************
0000c0  ***************.................................................
000100  ................................................................
000140  ............................................RRRRRRRRRRRRRRRRRRRR
000180  R.......RRR.......RRRR..........................................
0001c0  ................................................................
```

Description

The Coverage window consists of the following components.

Menu bar		
Top of client area:		Displays, as a header, character strings indicating address offsets.
Middle of client area	Left:	Displays addresses starting from that specified in the Prompt dialog box.
Middle of client area	Right:	Displays the contents of memory corresponding to the addresses. This area has scroll bars along its the right and bottom edges.
Button area		

The following symbols are displayed to indicate coverage data.

Symbol	Default display	When -64 is specified
.	No R/W/X	Areas for which 64-byte access is not applied
*	Only X	All 64 bytes changed to X
R	Only R	All 64 bytes changed to R
$	X & R	All 64 bytes changed to X or R
W	Only W	All 64 bytes changed to W
#	W & W	All 64 bytes changed to X or W
A	R & W	All 64 bytes changed to R or W
%	R & W &X	All 64 bytes changed to R, W, or X

R: Read W: Write X: Fetch

Buttons

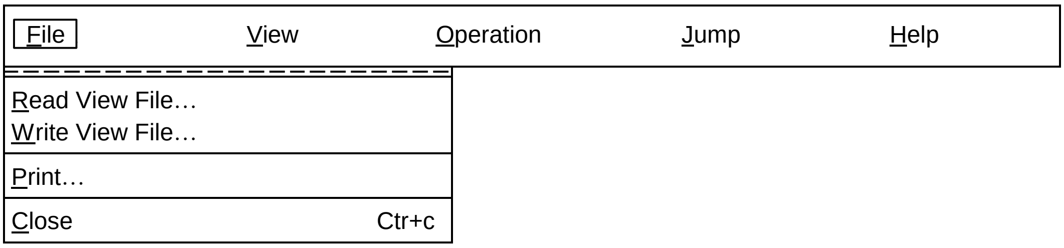
View Next — Scrolls the contents of memory down by one less than the number of displayed lines. When the end of the read results is displayed, clicking this button reads the next contents of memory.

View Previous — Scrolls the contents of memory up by one less than the number of displayed lines.

Close — Closes the Coverage window.

Menu bar

(1) File



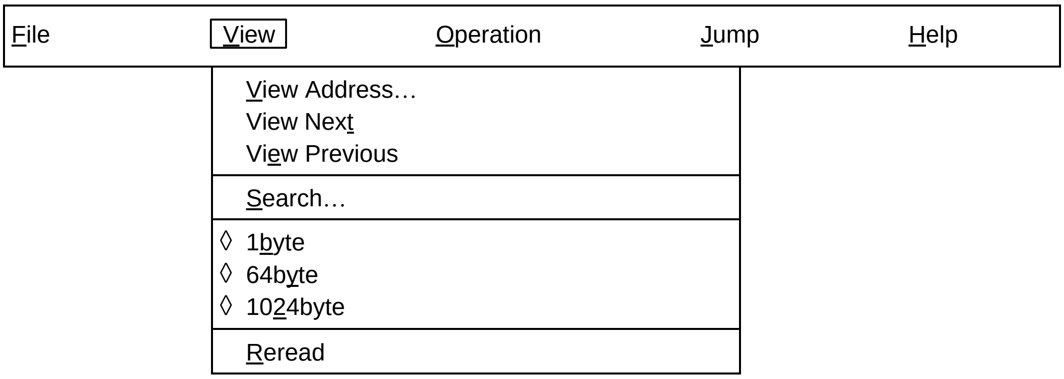
Read View File… Displays the File Selection dialog box (p.104) to specify the file in which the window contents are saved. Reads and reproduces the contents of the file. Specify "cov" as the file extension.

Write View File… Displays the File Selection dialog box (p.104) to specify the file to which the window contents will be saved. Writes the contents of the window in the file. Specify "cov" as the file extension.

Print… Displays the Prompt dialog box (p.115) to specify the print command. Prints the contents of the Coverage window.

Close Closes the Coverage window. Has the same function as the Close button.

(2) View



View Address… Displays the Prompt dialog box (p.115) to specify the address from which the coverage data display starts.

View Next Scrolls the contents of memory down by one less than the number of displayed lines. When the end of the read results is being displayed, clicking this button reads the next contents of memory. Has the same function as the View Next button.

View Previous Scrolls the contents of memory up by one less than the number of displayed lines. Has the same function as the View Previous button.

Character string →

Search… Searches for the specified character string. When the character string is found, the cursor indicates its position.

The following three items are configured as radio buttons in the menu.

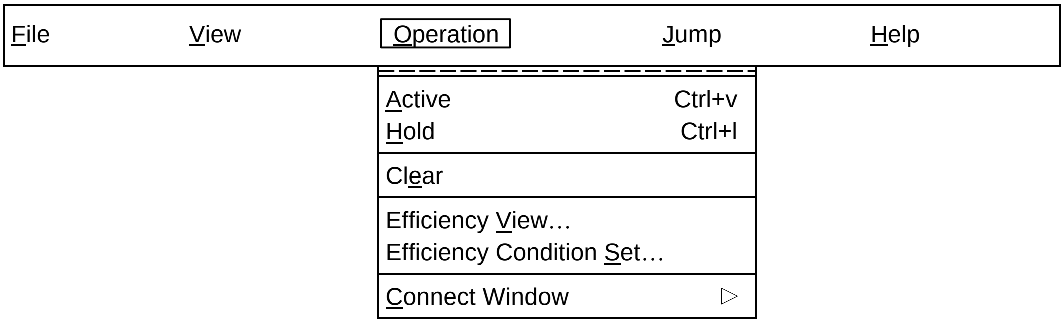
1byte Reads coverage data in one-byte units.

64byte Reads coverage data in 64-byte units.

1024byte Reads coverage data in 1024-byte units.

Reread Rereads from the address for window display and displays the data.

(3) Operation



Active Switches the window from hold status to active status.

Hold Switches the window from active status to hold status.

Clear Clears the coverage memory. Displays the Input dialog box (p.199) to specify the range to be cleared.

Efficiency View… Displays the coverage efficiency in the Coverage result dialog box (p.195) with the condition specified by Efficiency Condition Set.

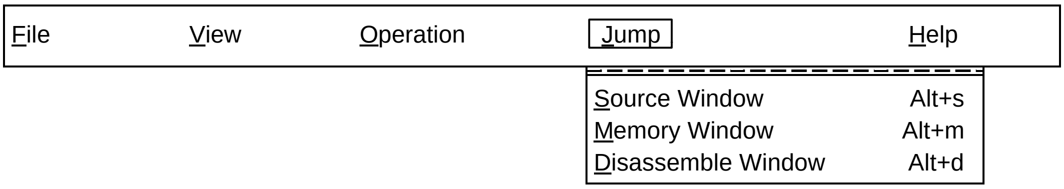
Efficiency Condition Set…

Sets the condition to be displayed by Efficiency View. Displays the Coverage items selection dialog box (p.197) to set the condition.

Connect window Displays the submenu to connect windows. In the submenu, the following window can be selected by clicking the toggle button. Select the windows to be connected.

- Source window
- Memory window
- Disassemble window

(4) Jump



Address →

Source Window — Displays the corresponding Source window (p.120) when source text that matches the specified address exists. When the active Source window is already displayed, however, the contents of the displayed window are rewritten.

Address →

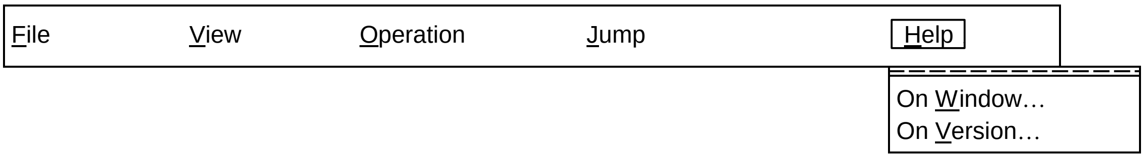
Memory Window — Displays the Memory window (p.170), starting from the specified address. When the active Memory window is already displayed, however, the contents of the displayed window are rewritten.

Address →

Disassemble Window

Displays the Disassemble window (p.201), starting from the specified address. When the active Memory window is already displayed, however, the contents of the displayed window are rewritten.

(6) Help



On Window… — Displays the Help dialog box (p.271).

On Version… — Displays the Message dialog box (p.131) that indicates the ID78K4 version.

Errors

- If an attempt is made to display another window from the Jump menu without first selecting an address, the system beeps.

Coverage Result dialog box	Confirmation dialog box

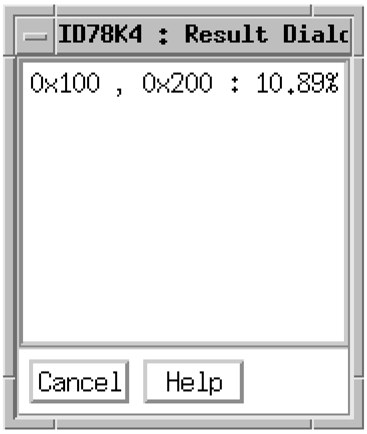
Outline

The Coverage result dialog box is used to display the coverage efficiency. The Coverage result dialog box is displayed in any of the following ways.

- In the Coverage window, select Efficiency View from Operation.

Note, however, that the dialog box is not displayed unless the efficiency condition has already been set.

Window



0x100 , 0x200 : 10.89%

Description

When Efficiency Condition is selected from Operation after Efficiency Condition Set has been specified in the Coverage window and the condition is set, the data is displayed.

The data is displayed as follows, in the order of the conditions specified in the client area.

0x80,0x100 : 61.24%

"0x80, 0x100", displayed here, is the address range set in the Coverage items selection dialog box. "61.24%" indicates the coverage efficiency in the specified address range.

Buttons

Cancel	Closes the dialog box.
Help	Displays the Help dialog box (p.271).

Notes

- Other windows cannot be accessed while this dialog box is open.

Coverage Items Selection dialog box

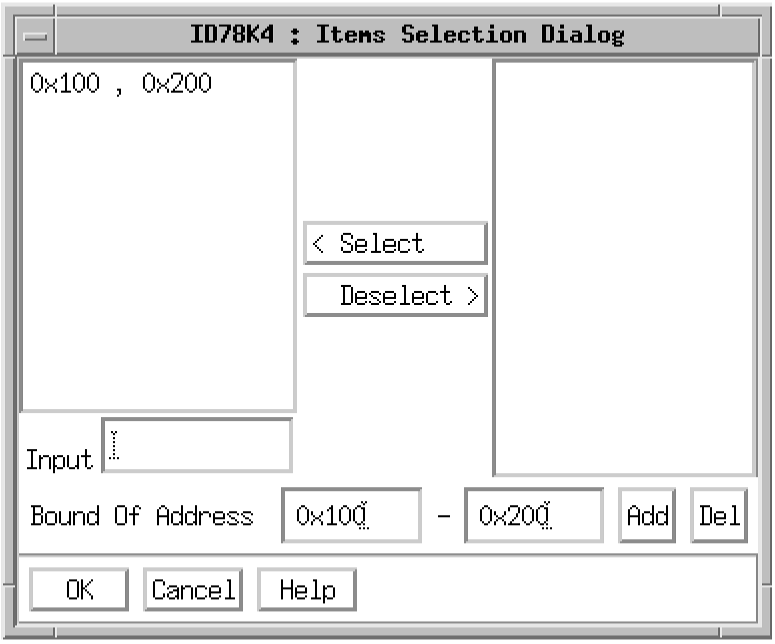
Selection dialog box

Outline

The Coverage items selection dialog box is used to set the condition that enhances the Coverage efficiency. The coverage items selection dialog box is displayed in any of the following ways.

- In the Coverage window, select Efficiency Condition Set from Operation.

Window



Description

The Coverage items selection dialog box consists of the following components.

Top of client area:	Area for selecting the coverage efficiency condition.
Bottom of client area:	Function button area.

Specifying a start and end address within the address range and clicking the Add button adds the condition to the coverage condition selection area. Clicking the Select or Deselect button when the added condition is clicked and displayed in reverse video either validates or deselects the set condition.

Buttons

The following four buttons are provided in the area for selecting the coverage efficiency condition.

Select	Validates the specified condition.
Deselect	Invalidates the specified condition.
Add	Adds the selected condition.
Del	Deletes the selected condition.

The following three buttons are provided in the button area.

OK	Closes the Coverage items selection dialog box after validating the currently displayed condition.
Cancel	Unconditionally closes the Coverage items selection dialog box.
Help	Displays the Help dialog box (p.271).

Input dialog box

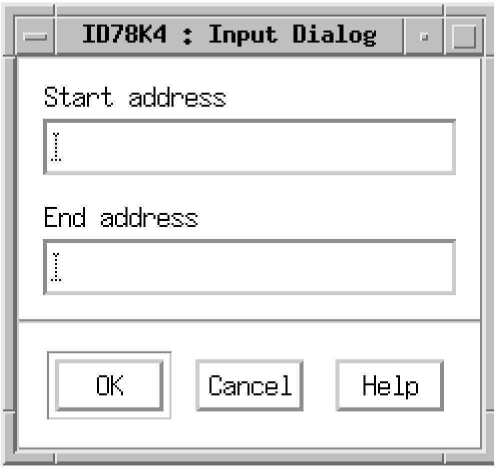
Selection dialog box

Outline

The Input dialog box is used to specify the range of coverage memory addresses to be cleared. The Input dialog box is displayed in any of the following ways.

- In the Coverage window, select Clear from Operation.

Window



Description

The Input dialog box consists of the following components.

Top of client area: Area for specifying the coverage memory range to be cleared.

Function button area

Specifying the start and end addresses of the range to be cleared in the coverage memory clear specification range and clicking the OK button clears the specified range of coverage memory.

Buttons

OK	Validates the currently displayed clear range and clears the coverage memory.
Cancel	Unconditionally closes the Input dialog box.
Help	Displays the Help dialog box (p.271).

Disassemble window

View/setting window

Outline

This Disassemble window is used to display the disassemble text. This window can also be used to execute line assemble, set a software or hardware break, and search for a character string.

This window can be opened in any of the following ways:

- In the Main window, select Window from the menu bar.

  → Select Disassemble window from the pull-down menu.

  → Specify an address in the Prompt dialog box.
  → Click the OK button in the Prompt dialog box.

- In the Windows Box window, click .

  → Specify an address in the Prompt dialog box.
  → Click the OK button.

- In the Source window, select a line number.

  → Select Jump from the menu bar.

  → Select Disassemble window from the pull-down menu.

- In the Stack Trace window, select a stack value.

  → Select Jump from the menu bar.

  → Select Disassemble window from the pull-down menu.

- In the Registers window, select a register.

  → Select Jump from the menu bar.

  → Select Disassemble window from the pull-down menu.

- In the SFR window, select an SFR.

  → Select Jump from the menu bar.

  → Select Disassemble window from the pull-down menu.

- In the Memory window, select an address.

  → Select Jump from the menu bar.

  → Select Disassemble window from the pull-down menu.

- In the Coverage window, select an address.

  → Select Jump from the menu bar.

  → Select Disassemble window from the pull-down menu.

- In the Event Manager window, select an event.

  → Select Jump from the menu bar.

  → Select Disassemble window from the pull-down menu.

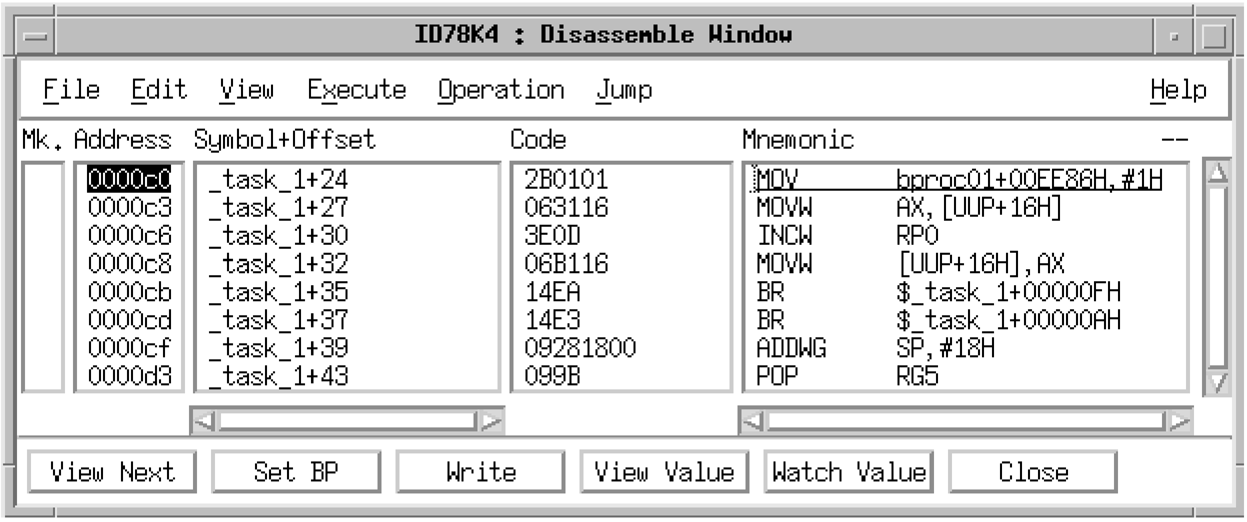
- In the Trace View window, select an address.

  → Select Jump from the menu bar.

  → Select Disassemble window from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



```
0000c0  _task_1+24  2B0101    MOV     bproc01+00EE86H,#1H
0000c3  _task_1+27  063116    MOVW    AX,[UUP+16H]
0000c6  _task_1+30  3E0D      INCW    RP0
0000c8  _task_1+32  06B116    MOVW    [UUP+16H],AX
0000cb  _task_1+35  14EA      BR      $_task_1+00000FH
0000cd  _task_1+37  14E3      BR      $_task_1+00000AH
0000cf  _task_1+39  09281800  ADDWG   SP,#18H
0000d3  _task_1+43  099B      POP     RG5
```

Description

This dialog box consists of the following components:

Menu bar

Client area: Displays the results of disassembly, from the address specified in the Prompt dialog box (p.115), in the following order: address, symbol+offset, code, and mnemonic. This area has a mark point area on the left. This area always has scroll bars along the right and bottom edges of Symbol+Offset and Mnemonic.

Button area

The following markings are displayed at the upper right corner of the client area to indicate whether modified values have already been set in the target:

--: Indicates that no values have been modified, or that all the modified values have already been set in the target.

**: Indicates that a value is being modified or that a modified value has not yet been set in the target.

In the mark point area, the display of * indicates an executable line. When a breakpoint is set, the breakpoint mark (b for a software break and B for a hardware break) is displayed. The mark point area is two characters wide. The above breakpoint symbols are displayed on the left. Any of the following symbols is displayed on the right according to the set event.

T: Trace

S: Snapshot

I: Timer

U: Stub

&: Displayed when two or more of the above events are set on the same line.

The current PC is indicated by an underline.

An address or mnemonic required to use a function button or the menu bar can be selected as follows:

(1) Selecting an address

To select an address, click it in the address area. The selected address is displayed in reverse video.

(2) Selecting a mnemonic

To select a mnemonic, double-click it in the mnemonic area.

Clicking a blank area or another mnemonic deselects the selected mnemonic.

To modify a mnemonic, select it and modify it in the Disassemble window. The mnemonic is modified instruction by instruction. Each time an instruction is modified, write it into the target memory by clicking the Write button or selecting Write from the edit menu. The new value is set in the memory window. If the size of a new mnemonic is greater than that of the old one, it can be set by selecting Force Write from the edit menu. The next mnemonic is destroyed, however. To fill the remainder of the destroyed mnemonic with the nop instruction, select Fill Up With NOP from the operation menu.

Setting or releasing a break in the Disassemble window is explained below.

A break is set in either of the following two ways:

(1) Set the break in the mark point area.

A breakpoint can be set by clicking any point in the mark point area. Clicking the same point again releases the breakpoint.

(2) Set the breakpoint by clicking the Set BP button.

For an explanation of method (2), see Buttons.

Buttons

View Next — Scrolls the disassemble result by the number of view lines - 1.

If the last part of the previously read result is being displayed, the next disassemble result is read from the target.

Address →

Set BP — To set a breakpoint, displays the breakpoint mark in the mark point area corresponding to the selected line of the address area. If, however, a line for which a breakpoint has already been set is selected, this button erases the breakpoint mark to delete the breakpoint mark.

Write — Writes the modified mnemonic into the target memory (line assembly is executed).

Symbol →

View Value — Displays the value of the selected symbol in the Message dialog box.

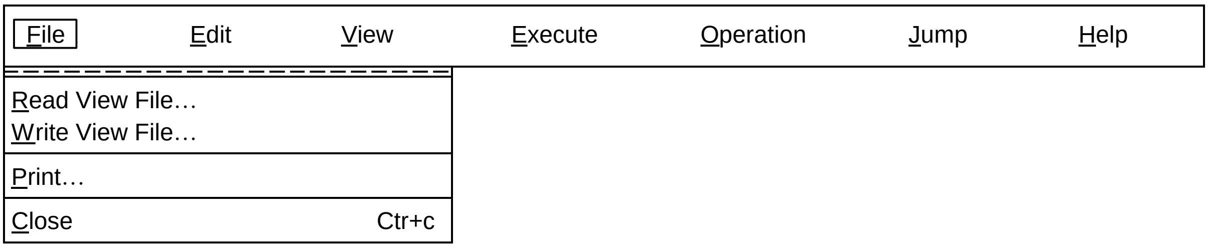
Symbol →

Watch Value — Adds the value of the selected symbol to the Variables window.

Close — Closes the Disassemble window.

Menu bar

(1) File



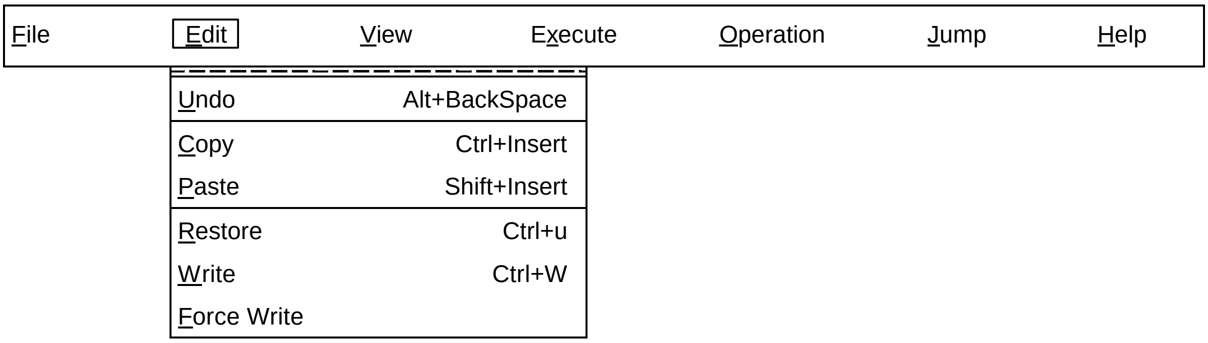
Read View File… Displays the File Selection dialog box (p.104) to enable specification of the file storing the contents of this window. The contents of the file are read and redisplayed. Specify dis as the file extension.

Write View File… Displays the File Selection dialog box (p.104) to enable the specification of the file storing the contents of this window. The contents of the window are written into the file. Specify dis as the file extension.

Print… Displays the Prompt dialog (p.115) box to enable specification of the print command. The contents of the Variables window are printed.

Close Closes the Disassemble window. This has the same function as the Close button.

(2) Edit



Undo — Undoes the previous edit operation.

Character string →

Copy — Copies the character string, selected with the mouse, onto the clipboard buffer.

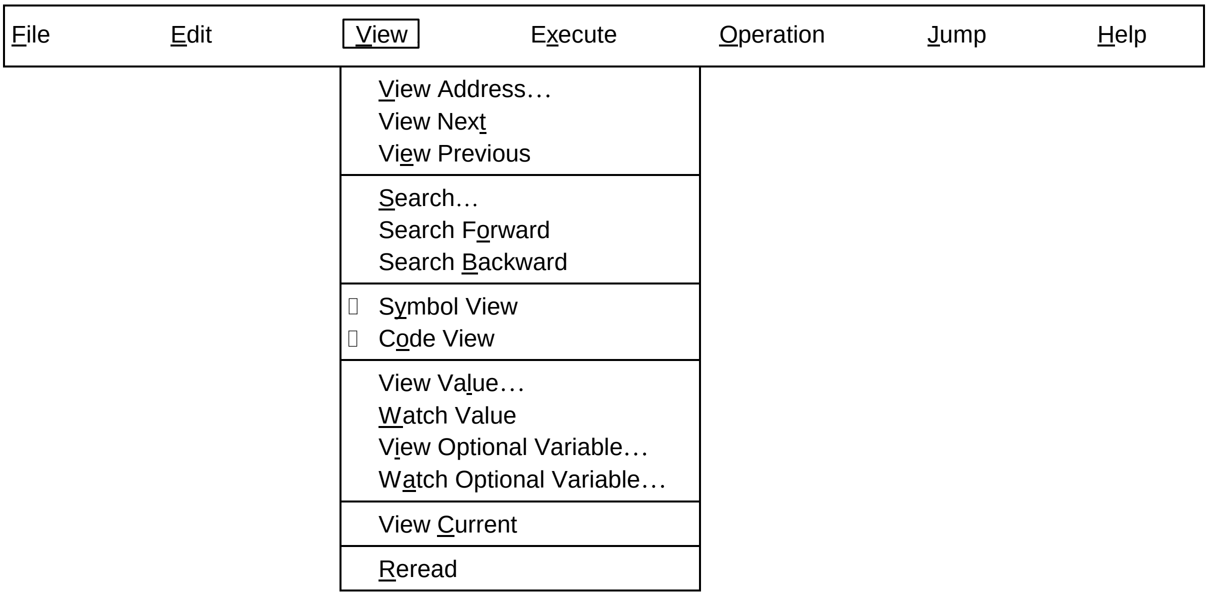
Paste — Pastes the contents of the clipboard buffer into the text cursor position. The pasting position must have already been selected by double-clicking. Even if the view mode is changed at pasting, the contents of the clipboard buffer are automatically changed to the view mode currently being displayed. A character string copied in another window cannot be pasted.

Restore — Restores a modified mnemonic to its original status.

Write — Writes a modified mnemonic in the target memory (line assemble is executed). This function is the same as that of the Write button.

Force Write — Even if the size of a new mnemonic exceeds that of the old mnemonic, forcibly sets the new mnemonic.

(3) View



View address… Displays the Prompt dialog box (p.115) to enable redisplay of the Memory window by specifying an address.

View Next Scrolls the disassemble result by the number of view lines - 1.

If the last part of the result which was read is being displayed, the next disassemble result is read from the target. This function is the same as that of the View Next button.

View Previous Displays the disassemble result by returning to 1/2 of the view address range.

List →

Search… Displays the Prompt dialog box (p.115) to enable specification of the list for which a search is to be made in the mnemonic area.

Search Forward Searches forwards through the list under the same conditions as the previous search. When a search is made for a list, it is displayed in reverse video. If a search is not made for a list, the system beeps.

Search Backward Searches backwards through the list under the same conditions as the previous search. When a search is made for a list, it is displayed in reverse video. If a search is not made for a list, the system beeps.

The following items are provided as toggle buttons on the menu:

Symbol View Determines whether the Symbol+Offset area is to be displayed. The default is the display of the Symbol+Offset area.

Code View Determines whether the Code area is to be displayed. The default is the display of the Code area.

View Value… Displays the value of the selected symbol in the Message dialog box. This has the same function as the View Value button.

Watch Value Adds the value of the selected symbol to the Variables window. This has the same function as the Watch Value button.

View Optional Variable…

Displays the Add Variable dialog box to enable the entry of the symbol name whose value is to be displayed. If a symbol has been selected, it is displayed in the Add Variable dialog box by default.

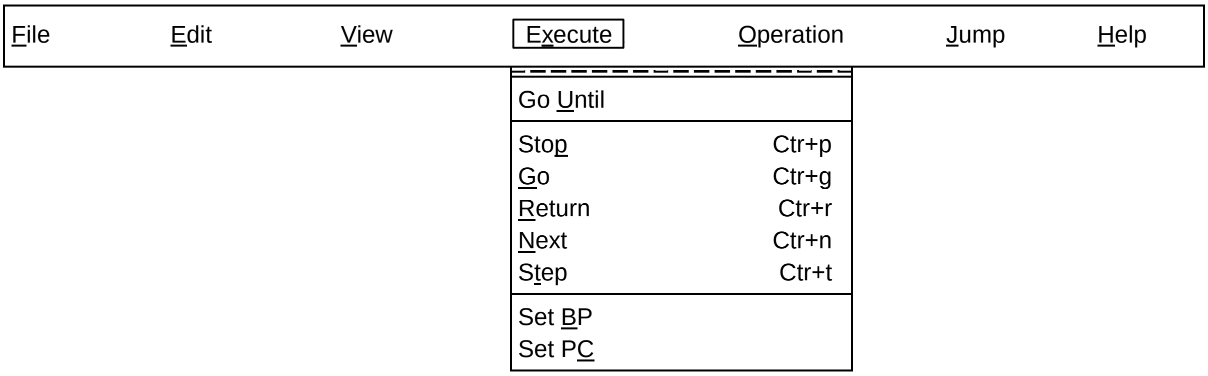
Watch Optional Variable…

Displays the Add Variable dialog box to enable entry of the symbol name whose value is to be monitored. When a symbol has been selected, it is displayed in the Add Variable dialog box by default.

View Current Displays the result of disassembly, starting from the current address.

Reread To redisplay the window, rereads it starting from the address in which it was displayed.

(4) Execute



Address →

Go Until Executes the target program up to the specified line.

Stop Stops the target program. This has the same function as the ⏸ button of the Main window.

Go Executes the target program. This has the same function as that of the ▶ button of the Main window.

Return Executes the target program until control returns to the calling function. This operation is the same as that of the ⏏ button of the Main window.

Next Executes the next step. This has the same function as the ⏭ button of the Main window.

Step Executes a single step. This has the same function as the ▶| button of the Main window.

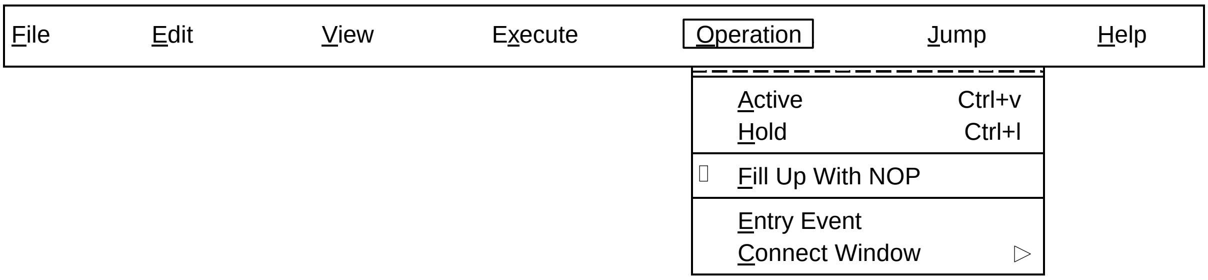
Address →

Set BP Displays the breakpoint mark in the mark point area corresponding to the selected line in the address area. If, however, a line for which the break point has already been set is selected, the breakpoint mark will be erased and the breakpoint deleted. This has the same function as the Set BP button.

Address →

Set PC Moves the program counter to the specified address.

(5) Operation



Active Switches the window status from hold to active.

Hold Switches the window status from active to hold.

The following items are provided as radio buttons on the menu:

Fill Up With NOP When a mnemonic has been rewritten by forcible setting, fills the excess area with the NOP instruction.

Address →

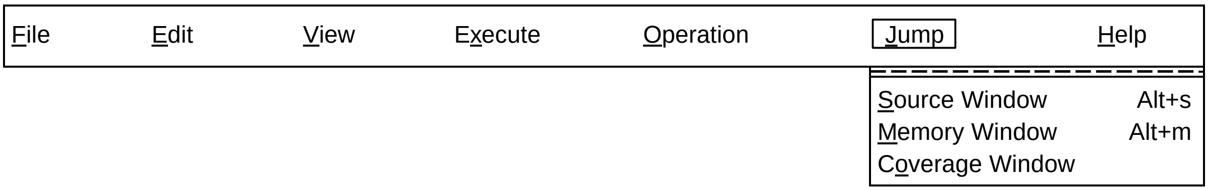
Entry Event Registers an event for the selected address.

Connect Window Displays the submenu for connecting windows.

In the submenu, the windows listed below can be selected by using the toggle button. Select the window to be connected.

- Source window
- Memory window
- Coverage window

(6) Jump



Address →

Source Window If a source text corresponding to the specified address exists, displays the associated Source window (p.120). If, however, the Source window in the active status is being displayed, rewrites the view contents of the Source window.

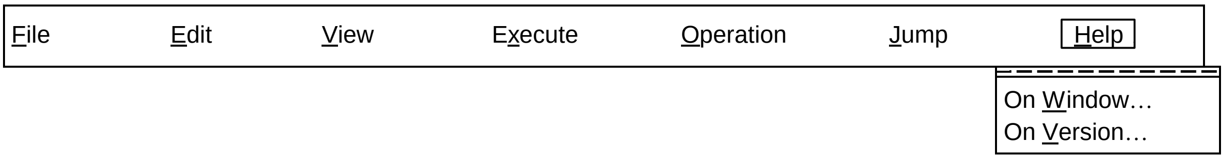
Address →

Memory Window Displays the Memory window (p.170), starting from the specified address. If, however, a Memory window in the active status is being displayed, rewrites the view contents of the Memory window.

Address →

Coverage Window Displays the Coverage window (p.188), starting from the specified address. If, however, the Coverage window in the active status is being displayed, rewrites the contents of the Coverage window.

(7) Help



On Window… Displays the Help dialog box (p.271).

On Version… Displays the Message dialog box (p.131) indicating the ID78K4 version.

Notes

- If a mnemonic changed by forcible setting in modify mode is longer than the original mnemonic at the specified address, the next mnemonic may be destroyed. If it is shorter than the original mnemonic, the next mnemonic is assumed to be invalid.
- A breakpoint can be set only within the same load module.
- This window cannot be opened while the CPU is performing emulation.

Errors

- The system beeps if an attempt is made to display another window from the Jump menu without first selecting a line number.

Event Manager window

View/setting window

Outline

This Event Manager window is used to display the event, break, and trace setting statuses.

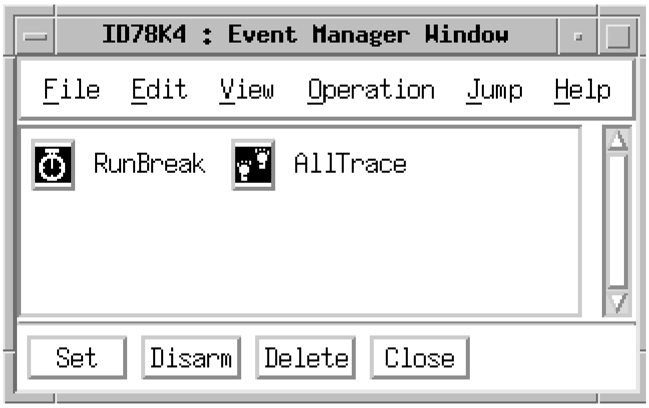
This window can be opened in any of the following ways:

- In the Main window, select Window from the menu bar.

  → Click the Event Manager window from the pull-down menu.
- In the Windows Box window, click ▦.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



Description

This window consists of the following components:

Client area: Displays the icons corresponding to the registered events, breaks, and traces.
In detail mode, the contents of each event are displayed.

The icon view area shows the type and name of an icon.

The following icon names are used regardless of their types: for events, Event0, Event1…; for breaks, Break0, Break1…; for traces, Trace0, Trace1…; for snapshots, SnapShot0, SnapShot1…; for stubs, Stub0, Stub1…; for timers, Timer0, Timer1…; and for breaks set from the Source window, MakeEvent0, MakeEvent1…

The icon types listed below are used. The icons for breaks, traces, snapshots, stubs, and timers are displayed in reverse video when set in the in-circuit emulator.

Event for which Fetch was specified as the access condition

Event for which an operation other than Fetch (Read, Write, Read&Write) was specified as the access condition

Event for which an operation other than Fetch was specified as the access condition and for which the address range was specified

Software break event

Hardware break event

Trace event

Snapshot event

Stub event

Timer event

An icon required to use a function button or menu bar can be selected as follows:

(1) Selecting an address

Select an icon by clicking the icon itself. The selected icon is marked with a frame and the icon name is displayed in reverse video. The selection is released by clicking the icon again.

After an icon has been selected, events are generated by dragging & dropping for the Event dialog box, Break dialog box, Trace dialog box, Snap Shot dialog box, Stub dialog box, and Timer dialog box.

Buttons

Select an icon →

Set — Sets the icon which is currently selected.

Select an icon →

Disarm — Releases the icon which is currently selected.

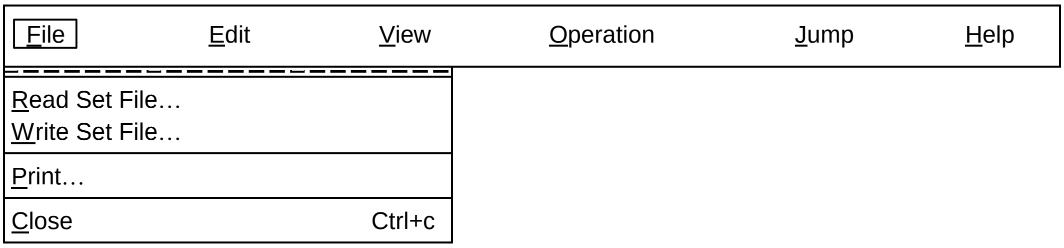
Select an icon →

Delete — Deletes the icon. If the selected icon is an event, it is also deleted from the in-circuit emulator setting. If the event is being used by the break trace, a Message dialog box prompting confirmation is displayed. The deleted event is also deleted from the break trace.

Close — Closes the Event Manager window.

Menu bar

(1) File



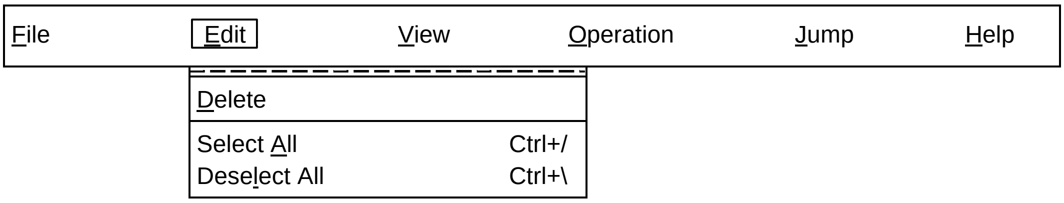
Read Set File… Displays the File Selection dialog box (p.104) to specify the file to be used to save the contents of this window. The contents of the file are read and redisplayed. Specify evt as the file extension.

Write Set File… Displays the File Selection dialog box (p.104) to specify the file to be used to store the contents of this window. The contents of the window are written into the file. Specify evt as the file extension.

Print… Displays the Prompt dialog box (p.115) to enable specification of the print command. The contents of the window are printed.

Close Closes the window. This has the same function as the Close button.

(2) Edit



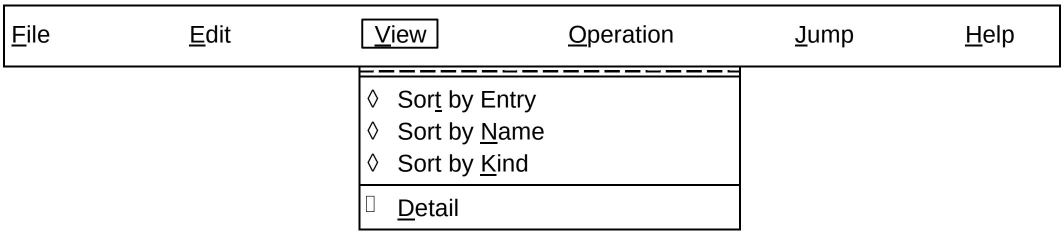
Select an icon →

Delete — Deletes the icon. If the selected icon is an event, it is deleted from the in-circuit emulator setting. If the event is being used by break trace, a Message dialog box prompting confirmation is displayed. The deleted event is also deleted from the break trace.
This function is the same as that of the Delete button.

Select All — Selects all icons in the Event Manager window.

Deselect All — Deselects all icons in the Event Manager window.

(3) View



The following three items are configured as radio buttons on the menu:

Sort by Entry	Displays icons by rearranging them into the order in which they were registered.
Sort by Name	Displays icons by rearranging them into alphabetical order.
Sort by Kind	Displays icons by rearranging them by type.

The following item is provided as a toggle button on the menu:

Detail Displays the icon and its registered contents. The registered contents are displayed with the symbols shown below. A condition set after a symbol listed below is displayed.

For the event condition:

[S]: Status condition
[A]: Address condition
[M]: Address mask condition
[d]: Data condition
[E]: External sense data condition
[F]: Pass count condition

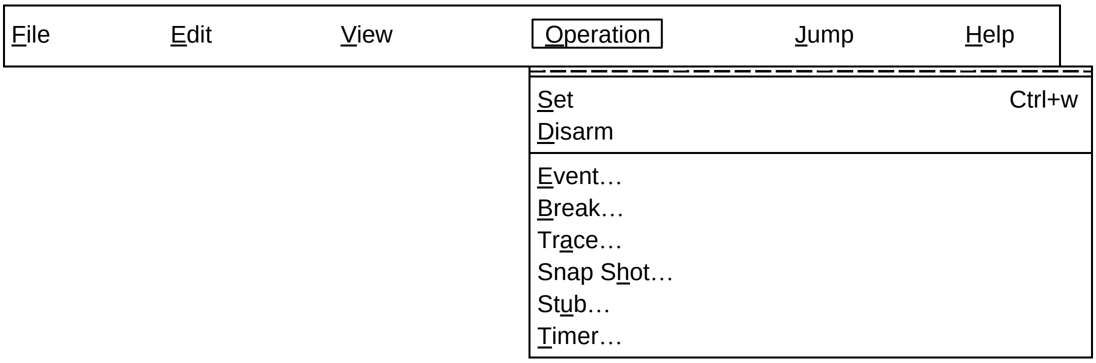
For the event link condition:

[P1]: Event link condition of first step
[P2]: Event link condition of second step
[P3]: Event link condition of third step
[P4]: Event link condition of fourth step
[D]: Disable condition

For the break, trace, timer, and stub conditions:

[B]:	Break condition
[SS]:	Sequential trace start condition
[SE]:	Sequential trace end condition
[Q]:	Qualify trace condition
[S]:	Timer start condition
[E]:	Timer end condition
[Sn]:	Snap condition
[Su]:	Stub condition
[A]:	Jump destination address upon stub event occurrence

(4) Operation



Select an icon →

Set — Sets the selected icon.

Select an icon →

Disarm — Releases the selected icon.

Event… — Displays details of the specified icon or displays the Event dialog box (p.222) to register a new event.

Break… — Displays details of the specified icon, or displays the Break dialog box (p.229) to register a new break event.

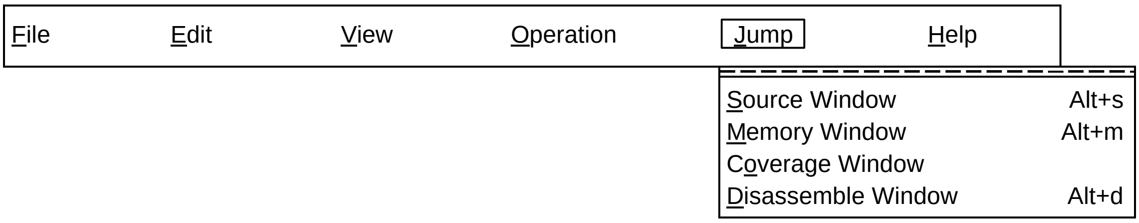
Trace… — Displays details of the specified icon, or displays the Trace dialog box (p.233) to register a new trace event.

Snap Shot… — Displays details of the specified icon, or displays the Snap Shot dialog box (p.242) to register a new snapshot event.

Stub… — Displays details of the specified icon, or displays the Stub dialog box (p.246) to register a new stub event.

Timer… — Displays details of the specified icon, or displays the Timer dialog box (p.249) to register a new timer event.

(5) Jump



Select an icon →

Source Window Displays the Source window (p.120). If, however, the active Source window is being displayed, the contents of this window are rewritten. The display starts from the event-set source line for the selected icon.

Select an icon →

Memory Window Displays the Memory window (p.170). If, however, the active Memory window is being displayed, the contents of the window are rewritten. The display starts from the event-set address for the selected icon.

Select an icon →

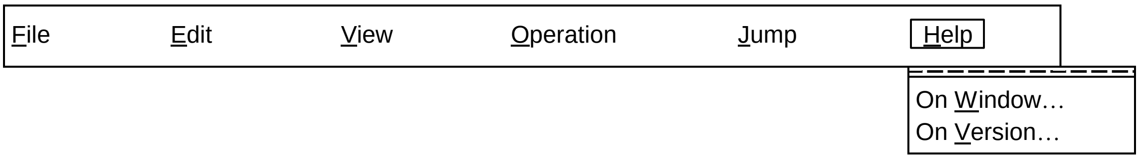
Coverage Window Displays the Coverage window (p.188). If, however, the active Coverage window is being displayed, the contents of the window are rewritten. The display starts from the event-set address for the selected icon.

Select an icon →

Disassemble Window

Displays the Disassemble window (p.201). If, however, the active Memory window is being displayed, the contents of the window are rewritten. The display starts from the event-set address of the selected icon.

(6) Help



On Window… Displays the Help dialog box (p.271).

On Version… Displays the Message dialog box (p.131) indicating the ID78K4 version.

Errors

- The system beeps if an attempt is made to display another window from the jump menu without first selecting an icon.
- The system beeps if an attempt is made to display another window from the jump menu with two or more icons selected.

Event dialog box

Selection dialog box

Outline

This Event dialog box is used to set or modify an event. A set event is registered in the Event Manager window (p.213).

This window is opened in any of the following ways:

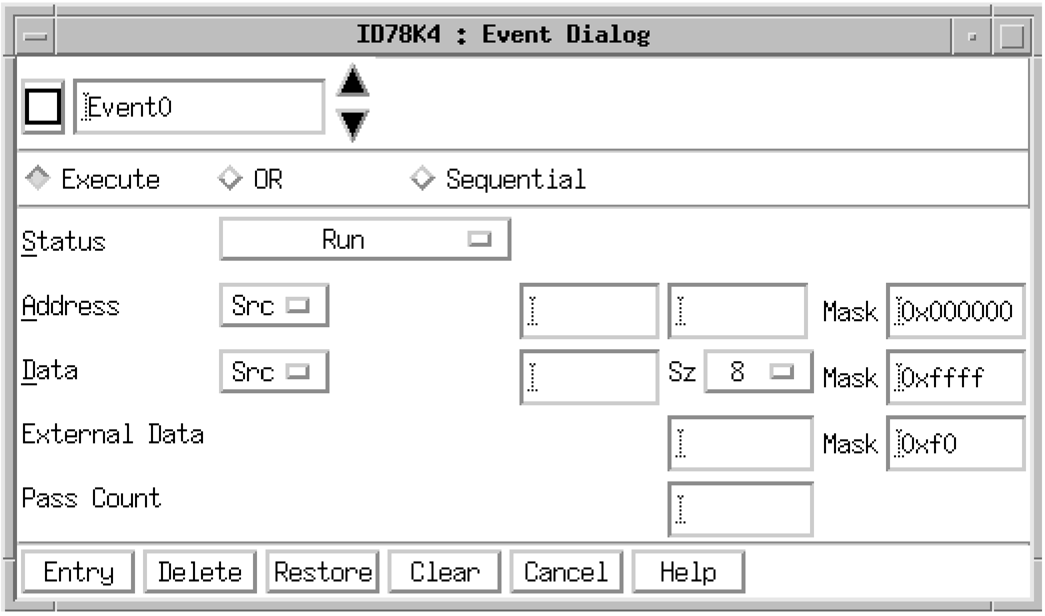
- In the Main window, select Window from the menu bar.

  → Click Event Dialog from the pull-down menu.
- In the Windows Box window, click 💡.
- Select Operation from the Event Manager window.

  → Click Event Dialog from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



ID78K4 : Event Dialog

□ Event0

◆ Execute ◇ OR ◇ Sequential

Status: Run

Address: Src Mask 0x000000

Data: Src Sz 8 Mask 0xffff

External Data Mask 0xf0

Pass Count

Description

This window consists of the following components:

Top of the client area:	Displays an icon indicating the event name, and the scroll buttons.
Middle of the client area Top:	Event type selection area
Middle of the client area Bottom:	Event display and edit area
Bottom of the client area:	Function button area

Each condition is set as follows:

(1) Setting an event name

When a new event is set, the icon indicating the event name displays the event name having the smallest value that is not currently being used. To modify a previously set event, move the icon in the Event Manager window (p.213) to the Event dialog box by dragging and dropping or change the event name by right-clicking. In this case, the order of the event names to be displayed conforms to the setting made with the view menu of the Event Manager window. An event name can also be modified by entering a new name from the keyboard.

(2) Setting an event type

Specify the event type to be set, by clicking the corresponding radio button.

The following event types can be specified:

<1> Execution event

<2> OR event

<3> Sequential event

To set a new event, select **<1>**.

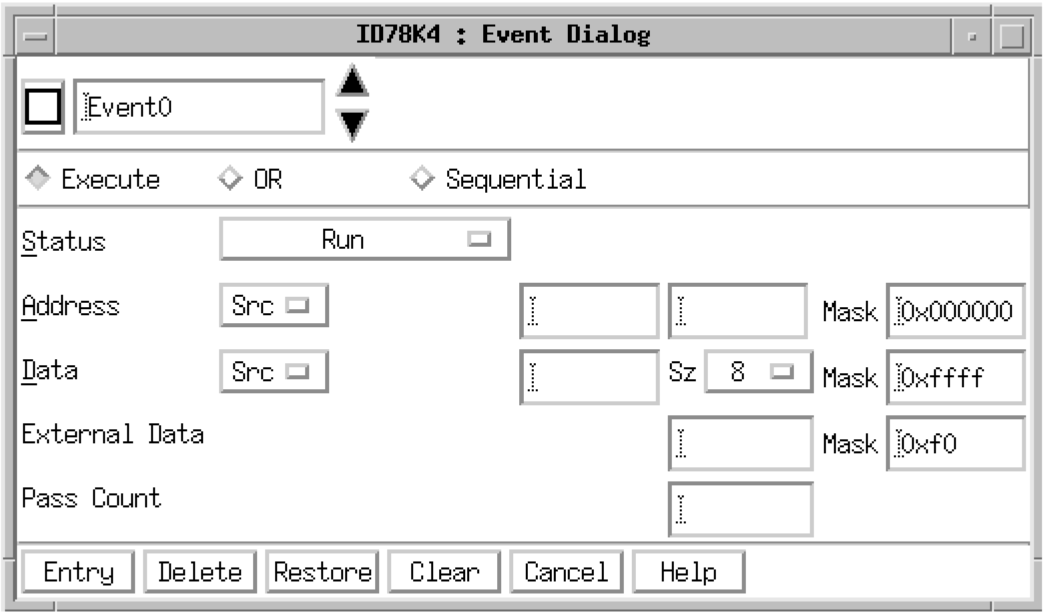
To create an event by combining previously set events, select **<2>** and **<3>**.

(3) Registering an event

The method for registering an event depends on the event type selected in (2). The registration method is explained below.

<1> Registering an execution/data access event

(Screen contents when an execution/data access event is registered)



ID78K4 : Event Dialog

□ Event0

◆ Execute ◇ OR ◇ Sequential

Status: Run

Address: Src Mask 0x000000

Data: Src Sz 8 Mask 0xffff

External Data: Mask 0xf0

Pass Count

a) Status

A status is selected by clicking an option button. An execution or access event is set at the same time as the status to be selected. The contents of the status conditions are listed below.

Status	Event type	Explanation
Run	Execution event	Program execution
Fetch	Access event	Program fetch (including prefetch)
Program Read(1)		Program data read
Program Write(2)		Program data write
Program R/W(3)		Program data read/write
MacrORead(4)		Data read in the macro service
Macro Write(5)		Data write in the macro service
MacrOR/W(6)		Data read/write in the macro service
Data Read(7)		Data read
Data Write(8)		Data write
Data R/W(9)		Data read/write
VECT		Vector address fetch
All		All conditions are valid.

b) Address

Select a level and specify an address condition.

A level is selected by clicking an option button. Select a level from Src and Ins.

After selecting a level, specify the address condition. For the condition specifying the address range, the left side is the start address condition and the right side is the end address condition. If no range is specified, specify only the left side. To set the mask condition, code it in the mask area.

The following address conditions can be specified:

Source level	Instruction level
C source line Variable name Function name Structure Assembler source line	Address expression Symbol name Label

For details of the specification method, see Chapter 3.

c) Data

Select a level and specify an address condition.

A level is selected by clicking the option button. Select a level from Src and Ins.

Set a data condition for an address condition. Specify the access size of the specified data by clicking the option button.

To specify the mask for the data condition, set it in the next mask setting area.

d) External Data

Set a data condition for data being received from the probe.

To specify the mask for the data condition, set it in the next mask setting area.

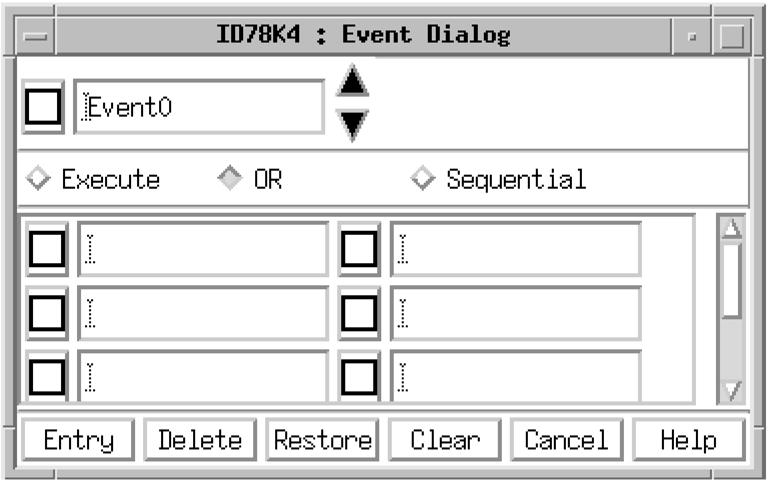
e) Pass Count

Specify a pass count. If no value is specified, 0 is assumed.
Clicking the Clear button clears the specified condition.

The specified condition is registered by clicking the Entry button.

<2> Registering the OR event

(Screen contents when the OR event is registered)



Specify the conditions for registering an OR event, using two or more events:

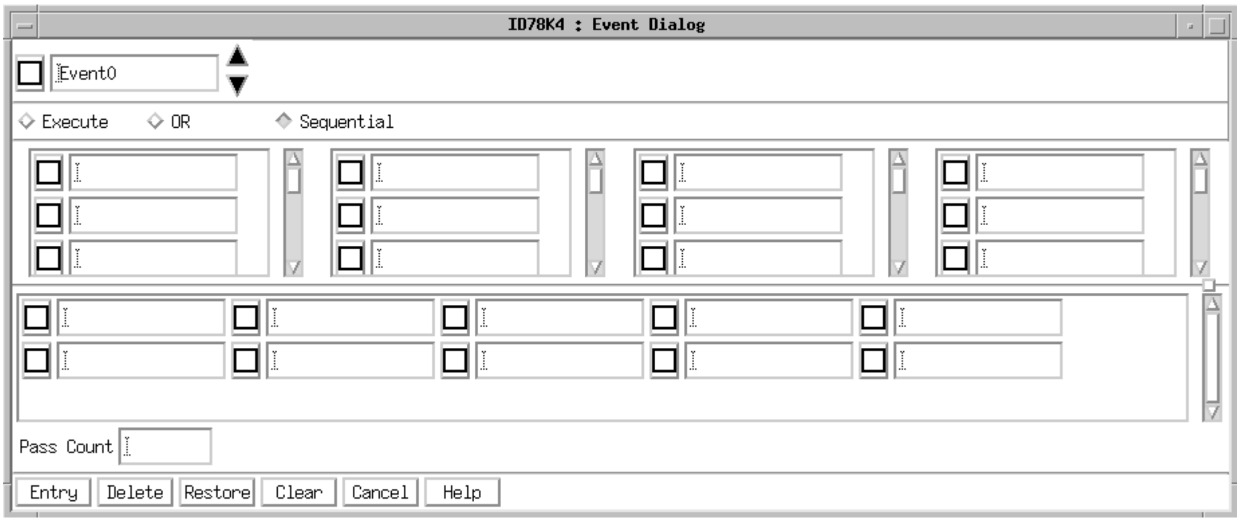
- Select the icon corresponding to the event from the Event Manager window by dragging and dropping. (Two or more event icons can be selected.)
- Click the event display edit area and enter the event name from the keyboard.

For the OR event, up to three execution events and up to seven access events can be defined.
The specified condition is registered by clicking the Entry button.

When an OR event is registered, the icon corresponding to the event is registered in the Event Manager window. If any one of the events specified for the OR event has been established, then the OR event will also have been established.

<3> Setting the sequential event

(Screen contents when a sequential event is set)



As a sequential event, specify the setting of the disable and sequential conditions by using two or more events. Specify the sequential conditions in the upper half and the disable conditions in the lower half. Specify the setting in any of the following ways:

- Select the icon of the event from the Event Manager window by dragging and dropping. (Two or more event icons can be selected at one time.)
- Click the event display edit area and enter the event name from the keyboard.

The specified condition is registered by clicking the Entry button.

When a sequential event is set, the icon corresponding to the event is registered in the Event Manager window. A sequential event is assumed to have been established when the event conditions are satisfied in the order in which they are specified. For a sequential event, up to three execution events and seven access events can be defined. A sequential event is established with the conditions in up to four steps. Specify the detection conditions in sequence, starting from the left. Two or more conditions can be set for each step. When any one of the set conditions has been satisfied, the conditions in the relevant step will also have been satisfied. When the conditions in all the steps are satisfied, the event will have been established. For the disable condition specification, if the event condition specified here is satisfied, the event condition for invalidating event establishment can also be specified.

Buttons

Entry

Modifies the Event Manager window based on the edited contents.

The event used when the OR or sequential condition was set cannot be registered by modifying its name and contents as long as an event generated by using it exists.

Delete

Deletes the displayed event from the Event manager window. The event used when the OR or sequential condition was set cannot be registered by modifying its name and contents as long as an event generated by using it exists.

Restore

Restores the contents of the event being displayed to the status existing before editing.

Clear

Clears the display of the Event dialog box. The icon returns to its initial status, or is cleared.

Cancel

Closes the Event dialog box.

Help

Displays the Help dialog box (p.271).

Notes

- As the event condition, up to three execution events and seven access events can be set.
- Any event cannot be set by the source file in any of the following cases: a program compiled without specifying the debug option (-g) is used, and only raw data is specified when a load module is loaded into the debugger.
- If the address range is specified for the address condition, no address mask value can be specified.
- This window cannot be opened when the CPU is performing emulation.

Errors

- The system beeps if the number of events exceeds the limit.
- The system beeps if an attempt is made to register an event that has already been registered.

Break dialog box

Selection dialog box

Outline

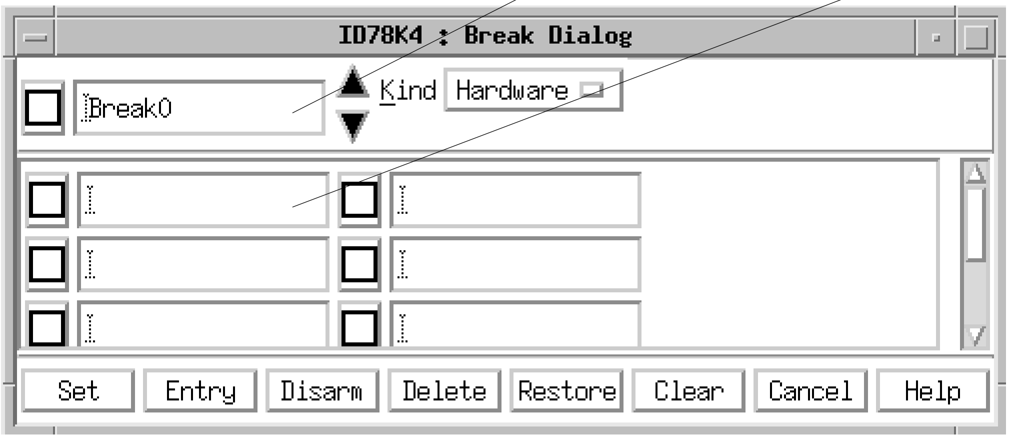
The Break dialog box is used to set a break. The set and registered breaks are managed as icons within the Event Manager window (p.213).

This window is displayed in any of the following ways:

- In the Main window, select Window from the menu bar.
  → Click Break Dialog... from the pull-down menu.
- In the Windows Box window, click ▽.
- Select Operation from the Event Manager window.
  → Click Break Dialog... from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this dialog box.

Window



Break name setting area

Break condition setting area

ID78K4 : Break Dialog

□ Break0

Kind Hardware

Description

This window consists of the following components:

Top of client area:	Displays the break name icon, scroll buttons, and option buttons.
Top of client area (middle):	Break event display/editing area
Bottom of client area (middle):	Break event condition display/editing area
Bottom of client area:	Function button area

The method for setting each condition is explained below.

(1) Setting a break name (break name setting area)

When a new break condition is set, the name of the break having the smallest unused value is with the break name icon. To modify the set break condition, move, by dragging & dropping, the icon in the Event Manager window (p.213) to the break name setting area. Or, use the button to the right of the icon to change the break name to another. In this case, the order of the break name to be displayed complies with the View menu in the Event Manager window (p.213). The break name can also be modified directly from the keyboard.

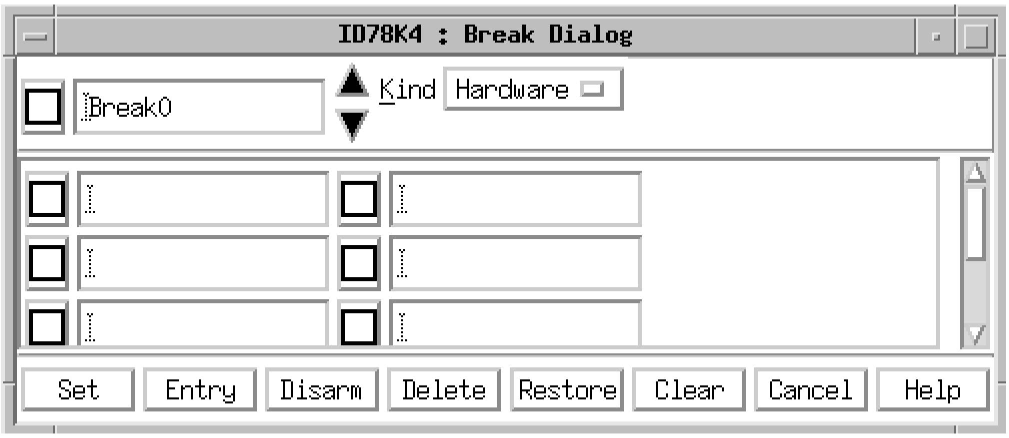
(2) Selecting a type

Two option buttons are provided for making this selection. Or, select hardware or software from the option menu. The default is the status in which hardware break was selected.

(3) Setting break conditions (break condition setting area)

<1> Setting a hardware break

(Hardware break setting screen)



ID78K4 : Break Dialog

Break0

Kind: Hardware

Specify break conditions with events, in either of the following two ways:

- Move, by dragging & dropping, the event icon from the Event Manager window (p.213) to the break condition setting area. (Two or more break conditions can be selected at a time.)
- Click the break condition setting area, then enter an event name from the keyboard.

Clicking the Clear button clears the specified condition.

The specified condition is set by clicking the Set button.

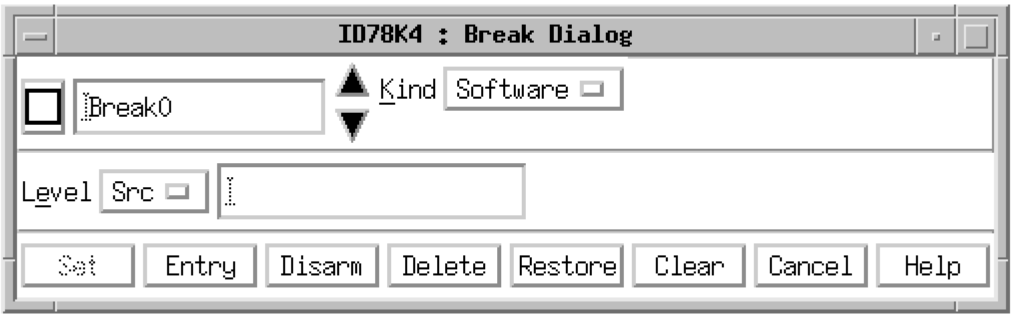
Setting the break event causes the event icon to be registered in the Event Manager window (p.213). When one of the events specified as the break conditions is established, the debugger executes a break.

By selecting an icon from the break condition setting area and clicking the Delete button, the event can be deleted from the break conditions.

For a hardware break, up to ten events can be set as the break condition.

<2> Setting software break

(Software break setting screen)



ID78K4 : Break Dialog

□ Break0

Kind Software

Level Src

Use the option buttons to set a break level. Select Src or Ins from the option menu.

After selecting the break level, specify an address condition.

The following can be specified as the address condition.

Source level	Instruction level
C source line Variable name Function name Structure Assembler source line	Address expression Symbol name Label

For a detailed description of the address conditions, see Chapter 3.

The specified address condition is cleared by clicking the Clear button.

The specified address condition is set by clicking the Set button.

When a break event is set, the event icon is registered in the Event Manager window (p.213). To validate a software break, set Software Break to ON in the Option dialog box (p.94) and specify the vector address of the jump destination in advance.

Buttons

Set — Sets the specified break according to the displayed conditions. Once a break has been set successfully, the break icon is registered in the Event Manager window (p.213).

Entry — Registers the break icon in the Event Manager window according to the displayed conditions.

Disarm — Disarms the displayed break if it has already been set.

Delete — Deletes the displayed break from the Event Manager window. A registered break is deleted when the setting is disarmed.

Restore — Restores the displayed break to the status existing before editing.

Clear — Clears the display of the break dialog box and restores the initial status.

Cancel — Closes the break dialog box.

Help — Displays the Help dialog box (p.271).

Notes

- If the break conditions to be added to an existing break conflict with other break conditions, the Error dialog box is displayed. Should this occur, first delete the set break, then set a new break.
- Up to ten events can be set as break conditions.

Errors

- If an attempt is made to re-register an already-registered break condition, the system beeps.
- If an attempt is made to re-set an already-set break condition, the system beeps.

Trace dialog box

Selection dialog box

Outline

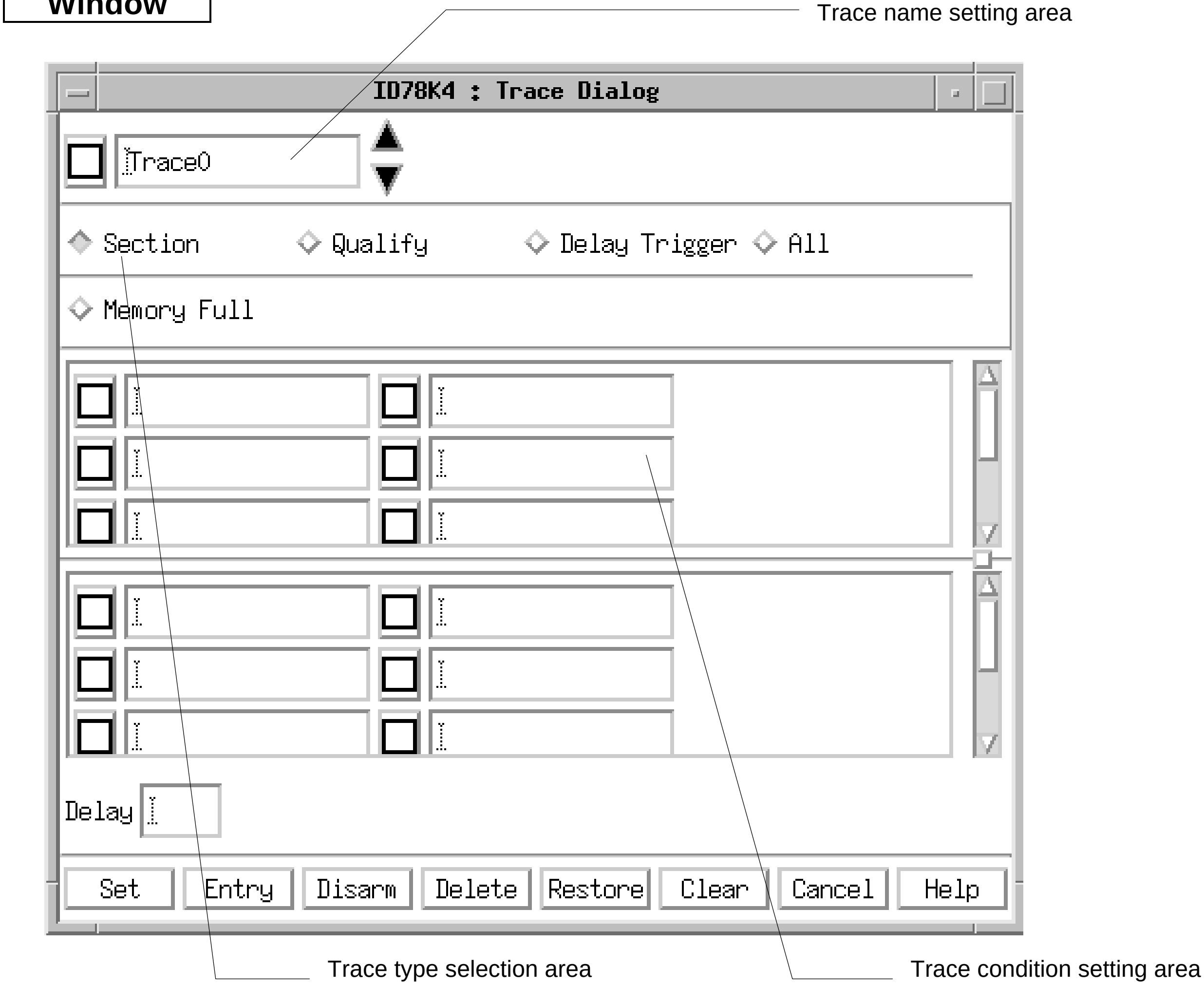
The trace dialog box is used to set a trace. The set and registered traces are managed as icons within the Event Manager window (p.213).

This dialog box is displayed in any of the following ways:

- In the Main window, select Window from the menu bar.
  → Click Trace Dialog... from the pull-down menu.
- In the Windows Box window, click .
- Select Operation from the Event Manager window.
  → Click Break Dialog... from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this dialog box.

Window



Description

This dialog box consists of the following components:

Top of client area:	Displays the trace icon and scroll buttons.
Middle of client area:	Trace type selection area
Bottom of client area:	Trace display/editing area
Button area	

The method of setting each condition is explained below.

(1) Setting a trace name (trace name setting area)

When a new trace condition is set, the name of the trace having the smallest unused value is displayed with the trace name icon. To modify the set trace condition, move, by dragging & dropping, the icon in the Event Manager window to the trace name setting area. Or, use the button to the right of the icon to switch the trace name to another. In this case, the order of the trace name to be displayed complies with the View menu in the Event Manager window. The trace name can also be modified directly from the keyboard.

(2) Selecting a trace type (trace type selection area)

Select a trace type. The setting screen changes depending on the selected type.

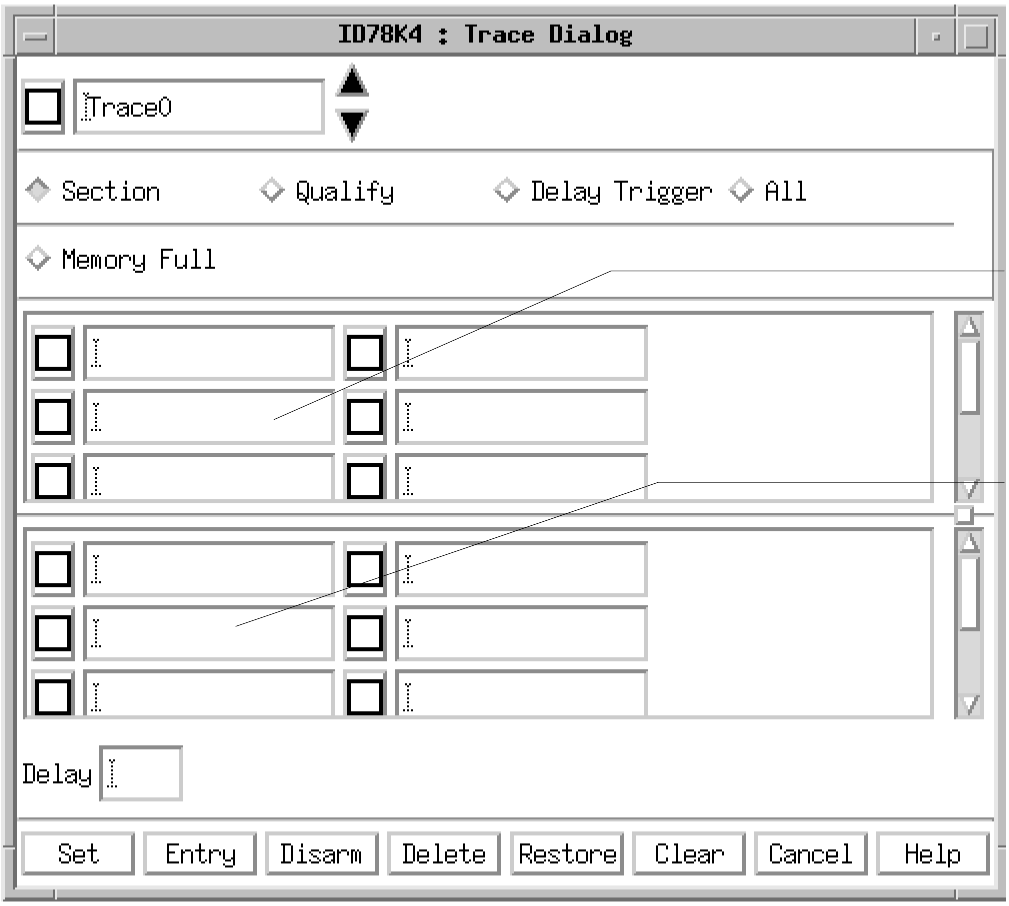
Select Section, Qualify, Delay Trigger, or All as the trace type.

When Memory Full is selected, the operation to be performed once the trace memory is full must be specified.

Setting trace conditions

<1> Section

(Section trace setting screen)



Trace start condition setting area

Trace end condition setting area

Selecting Section displays the section trace setting screen. Set the trace conditions as follows:

Setting the trace start conditions:

Specify trace start conditions, with events, in the trace start condition setting area, using either of the following two methods:

- Select, by dragging & dropping, the event icon in the Event Manager window. (Two or more trace start conditions can be selected at a time.)
- Click the trace start condition setting area, then enter an event name from the keyboard.

Clicking the Clear button clears the specified condition.

The specified condition is set by clicking the Set button. When the break event is set, the trace icon is registered in the Event Manager window. Upon the occurrence of any of the events specified as trace start conditions, the debugger executes trace. If only one event is specified, the debugger assumes that only a single trace start condition has been specified.

By selecting an icon from the trace start condition setting area and clicking the Delete button, an event can be deleted from the trace conditions.

Setting the trace end conditions:

Specify trace end conditions, using events, in the trace end condition setting area using events. The method for specifying trace end conditions is the same as that for specifying trace start conditions.

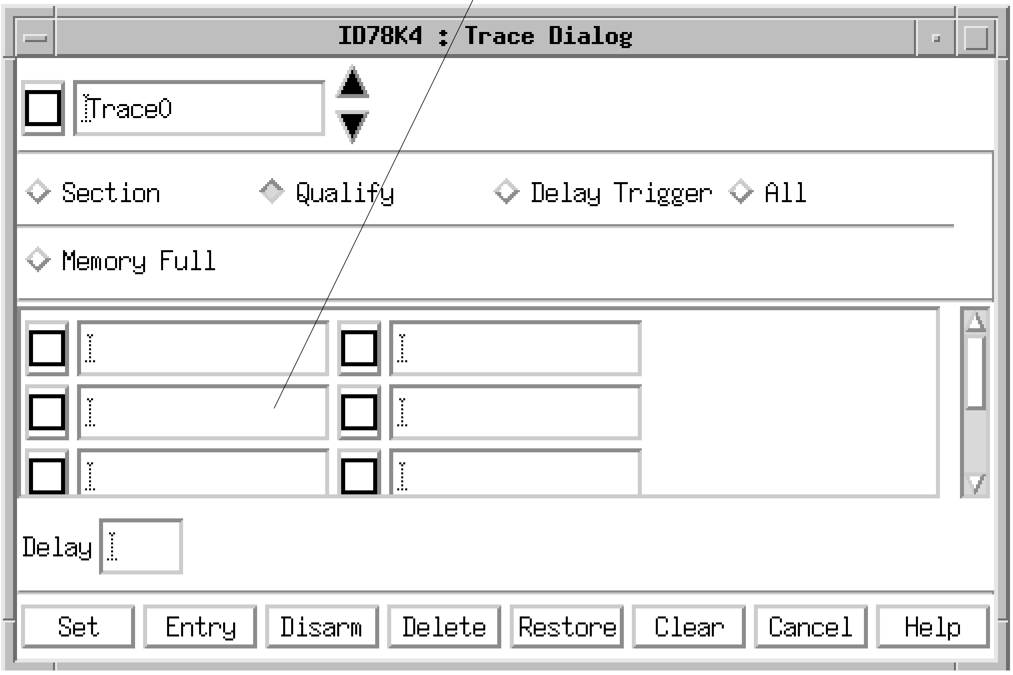
Specifying Delay:

Once the specified trace end conditions have been satisfied, the number of extra frames to be acquired can be specified as Delay.

<2> Qualify

(Qualify trace setting screen)

Qualify trace condition setting area



Selecting Qualify displays the qualify trace setting screen. Set the qualify trace conditions as follows:

Setting the qualify trace conditions:

Specify the qualify trace conditions, with events, in the qualify trace condition setting area, in either of the following two ways:

- Select, by dragging & dropping, the event icon in the Event Manager window. (Two or more qualify trace conditions can be selected at a time.)
- Click the qualify trace condition setting area, then enter an event name from the keyboard.

Clicking the Clear button clears the specified condition.

The specified condition is set by clicking the Set button. When a break event is set, the trace icon is registered in the Event Manager window.

Upon the occurrence of any of the specified events, the debugger executes trace.
By selecting an icon from the qualify trace condition setting area and clicking the Delete button, an event can be deleted from the trace conditions.

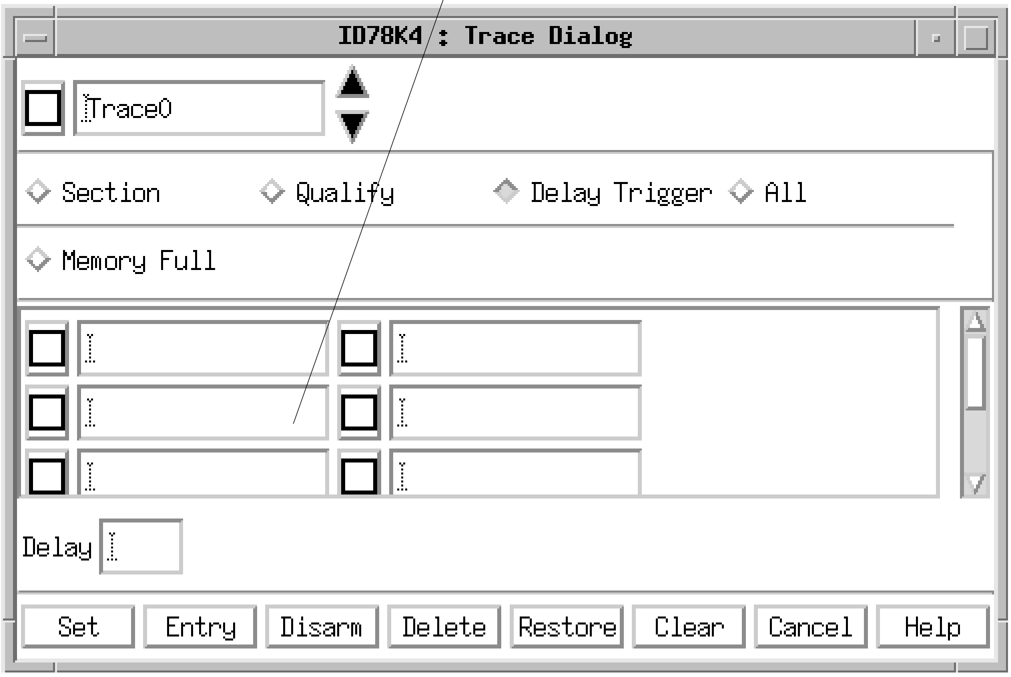
Specifying Delay:

Once the specified qualify trace conditions have been satisfied, the number of extra frames to be acquired can be specified as Delay.

<3> Delay Trigger

(Delay trigger setting screen)

Trace end condition setting area



Selecting Delay Trigger displays the delay trigger setting screen. This screen is used to specify only the section trace end conditions. Set the trace end conditions as follows:

Setting the trace end conditions:

Specify trace end conditions, with events, in the trace end condition setting area, using either of the following two methods:

- Select, by dragging & dropping, the event icon from the Event Manager window. (Two or more trace end conditions can be selected at a time.)
- Click the trace end condition setting area, then enter an event name from the keyboard.

Clicking the Clear button clears the specified condition.

A specified condition is set by clicking the Set button. When a break event is set, the trace icon is registered in the Event Manager window.

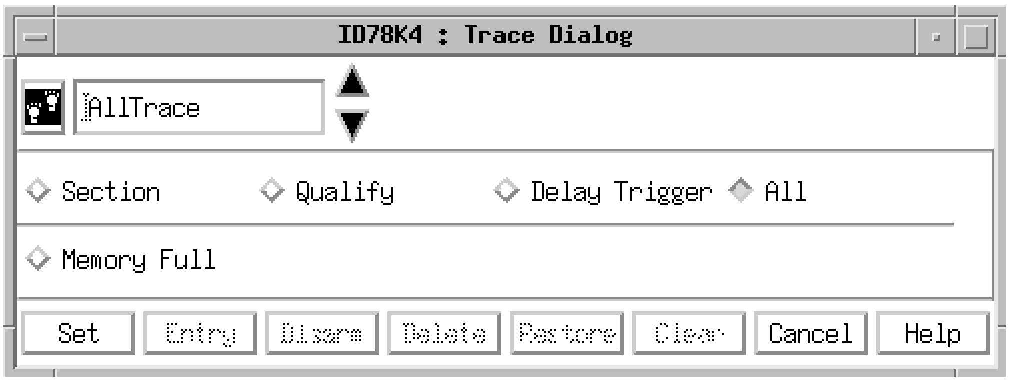
Each time any of the specified events occurs, the debugger executes trace.
By selecting an icon from the trace end condition setting area and clicking the Delete button, an event can be deleted from the trace end conditions.

Specifying Delay:

Once the specified trace end conditions have been satisfied, the number of extra frames to be acquired can be specified as Delay.

<4> All

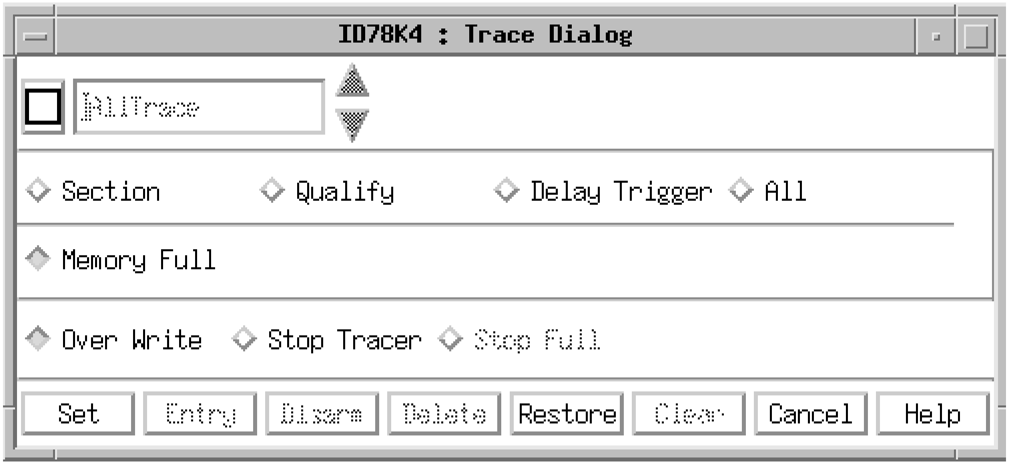
(All trace setting screen)



Selecting All from the trace dialog box displays the all trace setting screen.
Clicking the Set button sets all trace mode.

<5> Memory Full trace

(Memory full trace screen)



When trace memory becomes full, select whether the trace memory is to be overwritten or whether the tracer is to be stopped. The selected item is set by clicking the Set button.

Buttons

Set	Sets the specified trace according to the displayed conditions. Once trace has been set successfully, the trace icon is registered in the Event Manager window.
Entry	Registers the trace icon according to the displayed conditions.
Disarm	Disarms the displayed trace if it has already been set.
Delete	Deletes the displayed trace from the Event Manager window. A registered trace is deleted when the setting is disarmed.
Restore	Restores the displayed trace to the status existing before editing.
Clear	Clears the display of the Trace dialog box and restores the initial status.
Cancel	Closes the Trace dialog box.
Help	Displays the Help dialog Box (p.271).

Notes

- In the data access system, up to seven events can be used as trace conditions. In the execution system, up to three events can be used. That is, a total of ten events can be used. Of these ten events, two can be set within the valid address range. However, those events set as sequential conditions within the valid address range cannot be used.

Errors

- If an attempt is made to re-register an already-registered trace condition, the system beeps.
- If an attempt is made to re-set an already-set trace condition, the system beeps.

Snap Shot dialog box

Selection dialog box

Outline

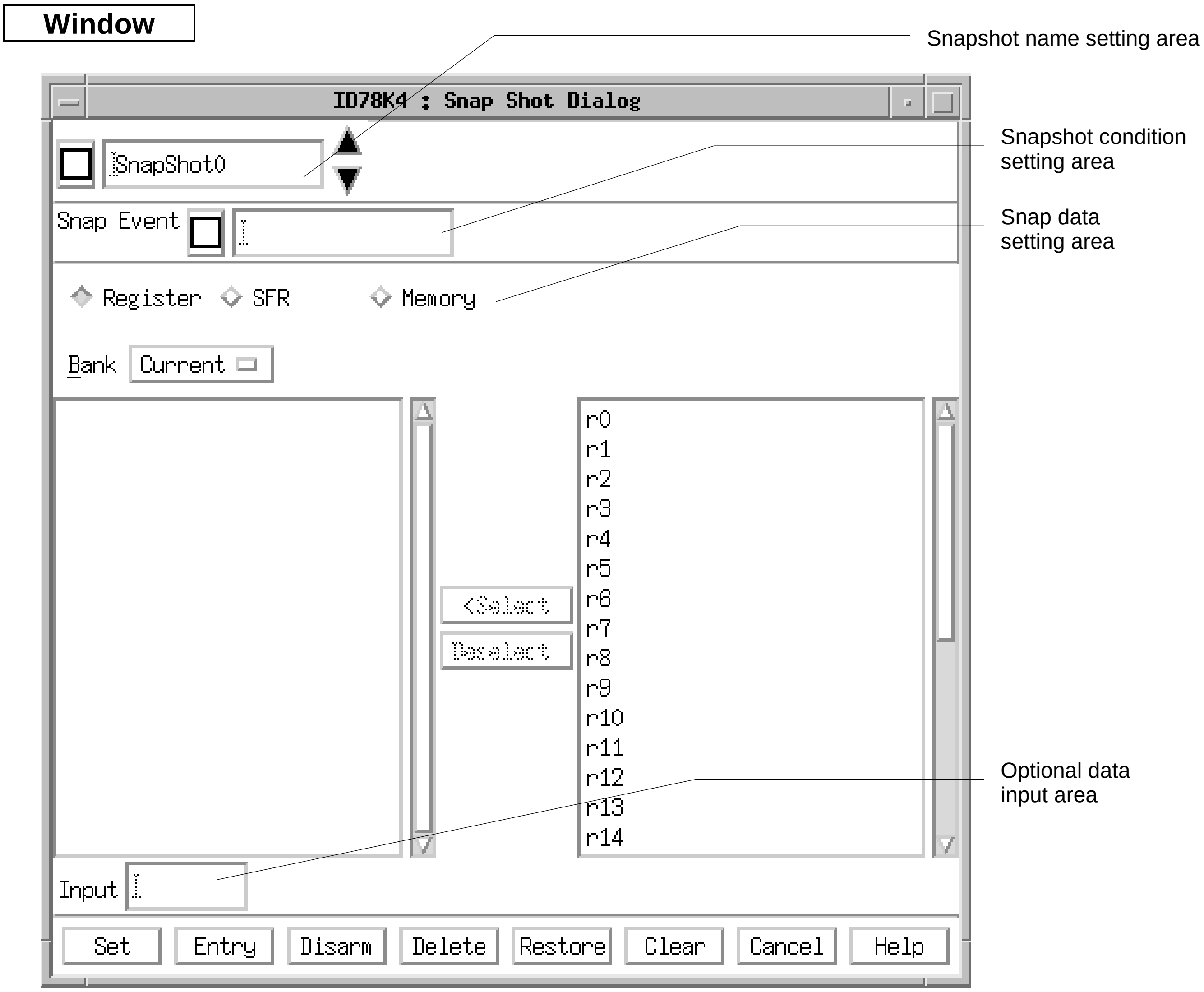
The Snap Shot dialog box is used to set a snapshot. The set and registered snapshots are managed as icons within the Event Manager window (p.213).

This dialog box is displayed in any of the following ways:

- In the Main window, select Window from the menu bar.
  → Click Snap Shot Dialog... from the pull-down menu.
- In the Windows Box window, click 📷.
- Select Operation from the Event Manager window.
  → Click Snap Shot Dialog... from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this dialog box.

Window



Description

This dialog box consists of the following components:

Top of client area:	Displays the snapshot event name icon and scroll buttons.
Middle of client area Top:	Snapshot event display/editing area
Middle of client area Bottom:	Snapshot target display/editing area
Button area	

The method of setting each condition is explained below.

(1) Setting a snapshot name (snapshot name setting area)

When a new snapshot condition is set, the name of the snapshot having the smallest unused value is displayed with the snapshot name icon. To modify the set snapshot condition, move, by dragging & dropping, the icon in the Event Manager window to the snapshot name setting area. Or, use the button to the right of the icon to switch the snapshot name to another. In this case, the order of the snapshot name to be displayed complies with the View menu in the Event Manager window. The snapshot name can also be modified directly from the keyboard.

(2) Setting the snapshot conditions (snapshot condition setting area)

Specify snapshot conditions, with events, in the snapshot condition setting area, using either of the following two methods:

- Select, by dragging & dropping, the event icon from the Event Manager window. (Two or more snapshot conditions can be selected at a time.)
- Click the snapshot condition setting area, then enter an event name from the keyboard.

(3) Setting the snap data (snap data setting area)

Only one type (Register, SFR, or Memory) can be set as snap data.

Up to 16 snap data items can be set.

<1> Selecting a register

To set a register as snap data, select the Register radio button.

Use an option button to specify a register bank from the option menu. When Current is selected, the bank that is set when the snapshot conditions are established is selected.

Registers are listed in the nonselection area to the right of the selection area. Click the register to be set. The register is displayed in reverse video. It is registered in the selection area by clicking the <Select button. Up to 16 registers can be registered in the selection area. To deselect a register in the selection area, click that register, then click the Deselect button.

A register can also be selected or deselected by writing it in the optional data input area and clicking the <Select or Deselect button.

<2> Selecting an SFR

To set an SFR as snap data, select the SFR radio button.

SFRs are listed in the nonselection area to the right of the selection area. Click the SFR to be set. The SFR is displayed in reverse video. It is registered in the selection area by clicking the <Select button. Up to 16 SFRs can be registered in the selection area. To deselect an SFR in the selection area, click that SFR, then click the Deselect button.

An SFR can also be selected or deselected by writing it in the optional data input area and clicking the <Select or Deselect button.

<3> Selecting memory

To set memory as snap data. select the Memory radio button.

Specify the start address to the left of the memory address area and the end address to the right.

Select a size from the access size area.

Buttons

Set Sets the specified snapshot according to the displayed conditions and registers it in the Event Manager window.

Entry Registers the snapshot icon according to the displayed conditions.

Disarm Disarms the displayed snapshot if it has already been set.

Delete Deletes the displayed snapshot from the Event Manager window. A registered snapshot is deleted when the setting is disarmed.

Restore Restores the displayed snapshot to the status existing before editing.

Clear Clears the displayed conditions.

Cancel Closes the Snap Shot dialog box.

Help Displays the Help dialog box (p.271) indicating the Snap Shot dialog box.

Notes

- This dialog box cannot be opened while the CPU is performing emulation.

Errors

- If an attempt is made to re-register an already-registered snapshot condition, the system beeps.
- If an attempt is made to re-set an already-set snapshot condition, the system beeps.
- If an attempt is made to select more than 16 items of snap data, the system beeps.

Stub dialog box

Selection dialog box

Outline

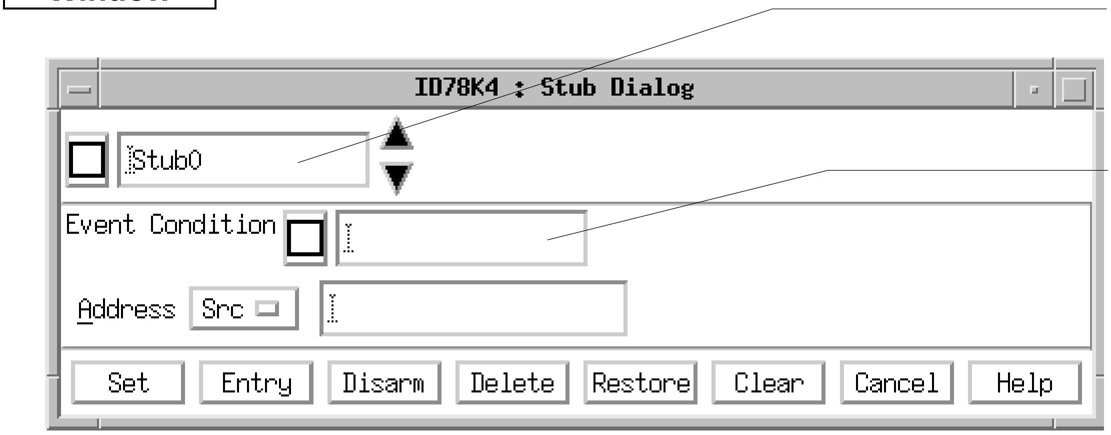
The Stub dialog box is used to set a stub. The set and registered stubs are managed as icons within the Event Manager window (p.213).

This dialog box is displayed in any of the following ways:

- In the Main window, select Window from the menu bar.
  → Click Stub Dialog... from the pull-down menu.
- In the Windows Box window, click .
- Select Operation from the Event Manager window.
  → Click Stub from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this dialog box.

Window



Stub0

Stub name setting area

Stub condition setting area

Description

This dialog box consists of the following components:

Top of client area:	Displays the stub name icon and scroll buttons.
Middle of client area:	Stub condition display/editing area
Bottom of client area:	Function button area

The method of setting each condition is explained below.

(1) Setting a stub name (stub name setting area)

When a new stub condition is set, the name of the stub having the smallest unused value is displayed with the stub name icon. To modify the set stub condition, move, by dragging & dropping, the icon in the Event Manager window (p.213) to the stub name setting area. Or, use the button to the right of the icon to switch the stub name to another. In this case, the order of the stub name to be displayed complies with the View menu in the Event Manager window (p.213). The stub name can also be modified directly from the keyboard.

(2) Setting the stub conditions (stub condition setting area)

Set an event condition in the event condition area. The event condition is set by moving, by dragging & dropping, the icon in the Event Manager window (p.213) to the event condition area, or by entering it directly from the keyboard.

(3) Setting an effective address (stub condition setting area)

Specify the address to be executed when the specified condition is established in the address area.

The following can be specified as the address condition.

Source level	Instruction level
C source line Variable name Function name Structure Assembler source line	Address expression Symbol name Label

For a detailed explanation of the address conditions, see Chapter 3.

The specified address condition is set by clicking the Set button.

When the stub is set, the event icon is registered in the Event Manager window.

Buttons

Set: Sets the specified stub according to the displayed conditions. Once a stub has been set successfully, the stub icon is registered in the Event Manager window (p.213).

Entry: Registers the stub according to the displayed conditions.

Disarm: Disarms the displayed stub if it has already been set.

Delete: Deletes the displayed stub from the Event Manager window. When an event is registered, this stub is deleted when the setting has been disarmed.

Restore: Restores the displayed stub to the status existing before editing.

Clear: Clears the display of the Stub dialog box and restores the initial status.

Cancel: Closes the Stub dialog box.

Help: Displays the Help dialog box (p.271).

Timer dialog box

Selection dialog box

Outline

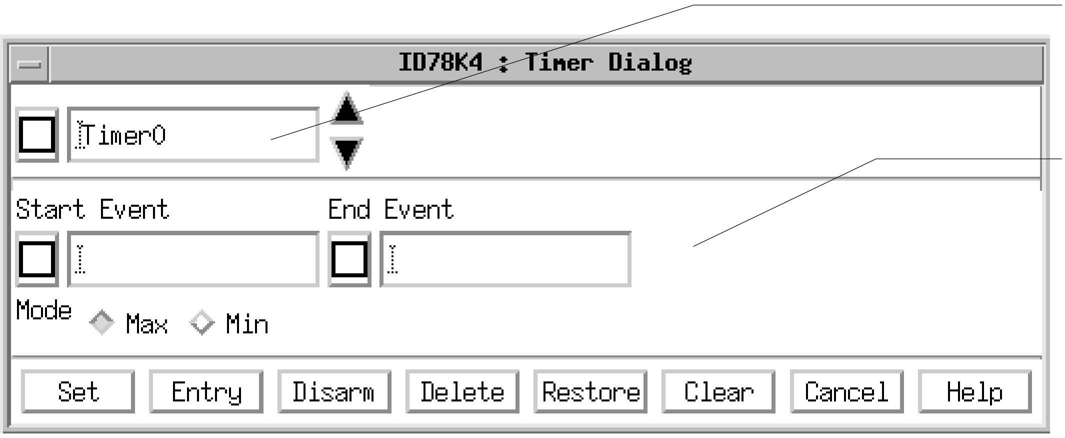
The Timer dialog box is used to set the conditions for measuring the execution time. The set and registered events are managed as icons within the Event Manager window (p.213). The measurement result is displayed in the Timer window.

This dialog box is displayed in any of the following ways:

- In the Main window, select Window from the menu bar.
  → Click Timer Dialog... from the pull-down menu.
- In the Windows Box window, click ⏱.
- Select Operation from the Event Manager window.
  → Click Timer from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this dialog box.

Window



Timer name setting area

Timer condition setting area

Description

This dialog box consists of the following components:

Top of client area:	Displays the timer name icon and scroll buttons.
Middle of client area:	Timer display/editing area
Bottom of client area:	Function button area

The method of setting each condition is explained below.

(1) Setting a timer name (timer name setting area)

When a new timer condition is set, the name of the timer having the smallest unused value is displayed with the timer name icon. To modify the set timer condition, move, by dragging & dropping, the icon in the Event Manager window (p.213) to the timer name setting area. Or, use the button to the right of the icon to switch the timer name to another. In this case, the order of the timer name to be displayed complies with the View menu in the Event Manager window (p.213). The timer name can also be modified directly from the keyboard.

(2) Setting the measurement conditions (timer condition setting area)

Set a start event condition in the start event condition area and an end event condition in the end event condition area. The start and end event conditions are set by moving, by dragging & dropping, the icon on the Event Manager window (p.213) to the associated condition area or by entering them directly from the keyboard.

(3) Setting a measurement mode (timer condition setting area)

Select Max or Min as the measurement mode.

Selecting Max displays the maximum data. Selecting Min displays the minimum data. The maximum data and minimum data are selected from that data for which timer measurement has been performed several times.

The specified timer condition is set by clicking the Set button.

When the timer event is set, the event icon is registered in the Event Manager window.

Buttons

Set	Sets the specified timer according to the displayed conditions. Once a timer has been set successfully, the timer icon is registered in the Event Manager window (p.213).
Entry	Registers the timer according to the displayed conditions.
Disarm	Disarms the timer event when the displayed timer has already been set.
Delete	Deletes the displayed timer from the Event Manager window. When the event is registered, this timer is deleted when the setting is disarmed.
Restore	Restores the displayed timer condition to the status existing before editing.
Clear	Clears the display of the Timer dialog box and restores the initial status.
Cancel	Closes the Timer dialog box.
Help	Displays the Help dialog box (p.271).

Trace View window

View window

Outline

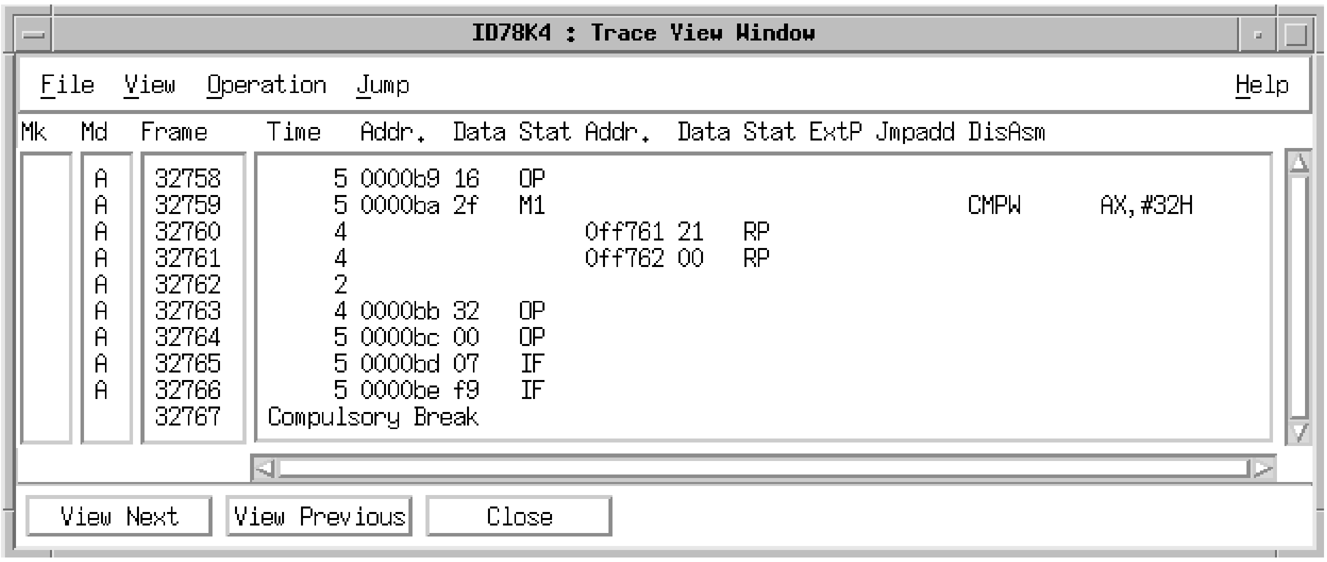
The Trace View window is used to displays the trace data written into trace memory. This window can also display the source text corresponding to trace data, memory dump, and disassemble information.

This window is displayed in any of the following ways:

- In the Main window, select Window from the menu bar.
  → Click Trace View Window... from the pull-down menu.
- In the Windows Box window, click .

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



```
Mk   Md   Frame    Time   Addr.   Data Stat Addr.   Data Stat ExtP Jmpadd DisAsm
      A   32758        5 0000b9 16   OP
      A   32759        5 0000ba 2f   M1                                 CMPW      AX,#32H
      A   32760        4                    0ff761 21   RP
      A   32761        4                    0ff762 00   RP
      A   32762        2
      A   32763        4 0000bb 32   OP
      A   32764        5 0000bc 00   OP
      A   32765        5 0000bd 07   IF
      A   32766        5 0000be f9   IF
          32767   Compulsory Break
```

Description

This window consists of the following components:

Menu bar	
Top of client area:	Label area
Bottom of client area:	Trace display area
Button area	

The items displayed in the trace display area are explained below.

(1) Mk

Displays the setting status of events when trace data is executed, using the following symbols:

Left of mark area:

B: Hardware break

b: Software break

Right of mark area:

T: Trace

S: Snapshot

I: Timer

U: Stub

&: When several of the above events are set on the same line

(2) Md

Displays the trace mode to be used, using the following characters:

A: All traces or section trace

Q: Qualify trace

S: Single step trace

M: Machine cycle trace

E: Event cycle trace

I: Instruction cycle trace

(3) Frame

Displays frame numbers (0 to 32767).

If the same frame is displayed on two lines, → is displayed in the second line.

An invalid frame is displayed in -.

(4) Time

Displays the time required to execute the frame, in ms.

(5) Addr.

Displays fetch addresses.

(6) Data

Displays fetch data.

(7) Stat

Displays the following fetch status information:

M1:	M1 fetch
BRM1:	M1 fetch after branch
OP:	OP fetch
IF:	Invalid fetch
SNAP:	Snap display

(8) Addr.

Displays data access addresses.

(9) Data

Displays access data.

(10) Stat

Displays the following access status information:

R-P:	Read of data read/write
-WP:	Write of data read/write
RP:	Data read
WP:	Data write

(11) ExtP

Displays external logic probe data.

(12) Jmpadd

Displays jump addresses.

(13) DisAsm

Displays disassemble information.

Buttons

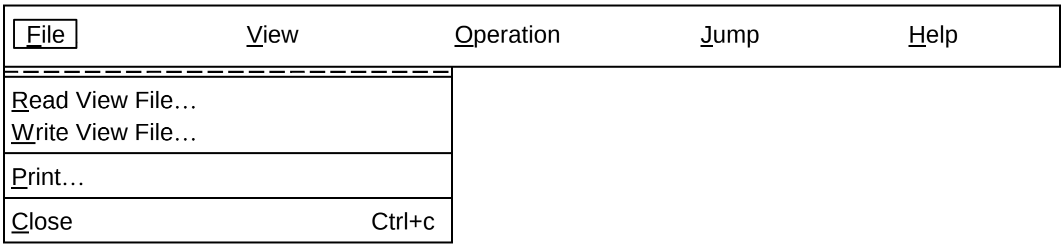
View Next — Moves to the next frame.

View Previous — Moves back to the previous frame.

Close — Closes the Trace View window.

Menu bar

(1) File



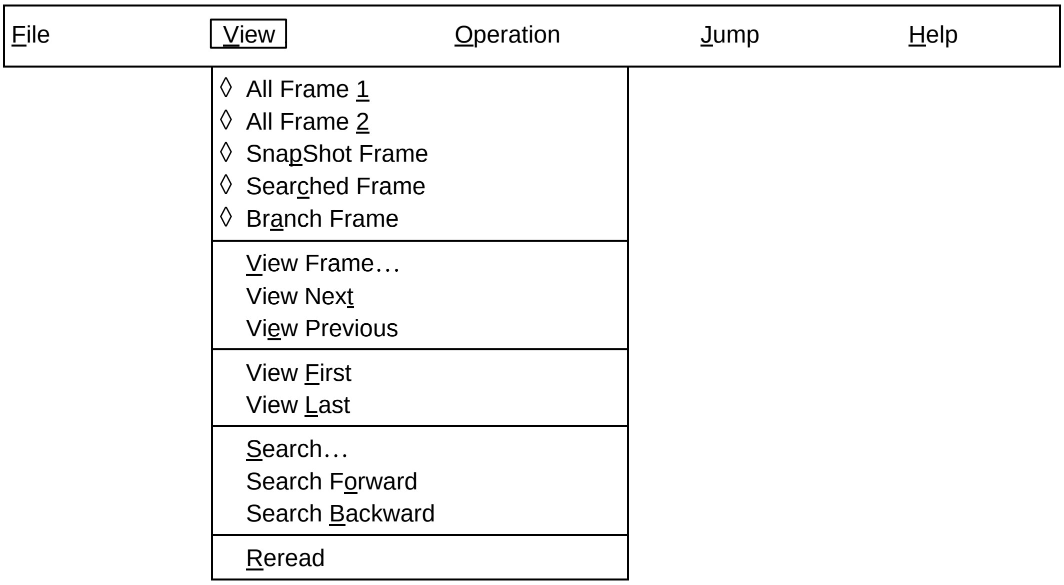
Read View File... Displays the File Selection dialog box (p.104) from which the file containing the contents of this window is specified. Specifying a file from the File Selection dialog box reads (reproduces) the contents of this window from the file. Specify trv as the file extension.

Write View File… Displays the File Selection dialog box from which the file that stores the contents of this window is to be specified. Specifying a file from the File Selection dialog box (p.104) writes the contents of this window into the file. Specify trv as the file extension.

Print… Displays the Prompt dialog box (p.115) from which the print command is specified. Specifying the print command from this dialog box prints the contents of the Trace View window.

Close… Closes the Trace View window. This has the same function as the Close button.

(2) View



The following five items are provided as radio buttons on the menu.

All Frame 1 Displays all trace data except search marking information.

All Frame 2 Displays all trace data including search marking information.

SnapShot Frame Displays the snapshot frame and trace block frame.

Searched Frame Displays the frames that satisfy the specified search conditions and the trace block frame. This item is valid only when trace search is executed.

Branch Frame Displays the branch and trace block frames for the program.

View Frame Displays the Prompt dialog box from which the frame number to be displayed is to be specified. Specify the frame number to be displayed.

View Next Displays the data immediately subsequent to the currently displayed data. If the number of internally allocated trace data items is less than the number of subsequent blocks that can be displayed, rereads all the trace data. This has the same function as the View Next button.

View Previous Displays the immediately preceding data. If the number of internally allocated trace data items is less than the number of previous blocks that can be displayed, rereads all trace data. This has the same function as the View Previous button.

View First Displays the foremost (first data) of the currently allocated trace data.

View Last Displays the endmost (last data) of the currently allocated trace data.

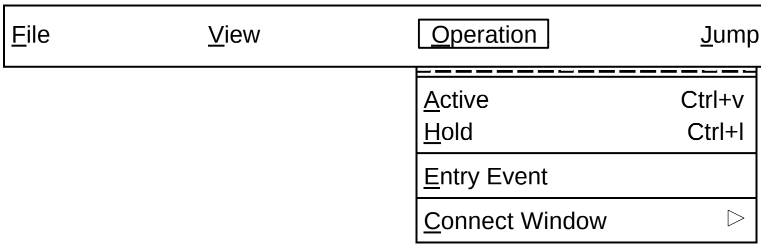
Search… Displays the Trace Search dialog box for trace qualify search. Trace data is redisplayed, starting from the searched point.

Search Forward Searches for the next candidate according to the previous search conditions.

Search Backward Searches for the previous candidate according to the previous search conditions.

Reread Rereads and displays trace data, starting from the address displayed in the window.

(3) Operation



Active Switches the window from the hold state to the active state.

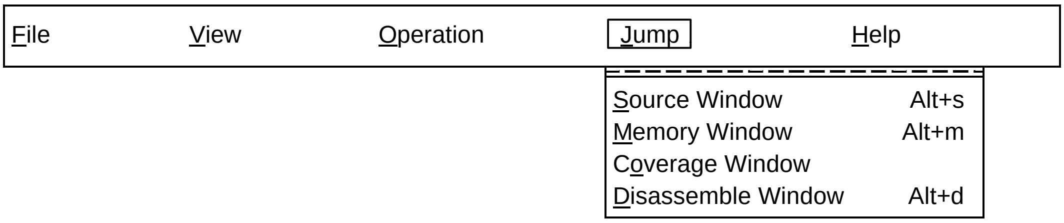
Hold Switches the window from the active state to the hold state.

Entry Event Registers the address of the selected trace line in the event.

Connect Window Displays the submenu for window connection. This submenu enables the following windows to be selected by clicking a toggle button. Select the window to be connected from the submenu.

- Source window
- Memory window
- Coverage window
- Disassemble window

(4) Jump



Address →

Source Window When source text corresponding to the address selected in the trace data area exists, displays the Source window (p.120) corresponding to the source text. If an active-state Source window is already displayed, however, the contents of the window are rewritten.

Address →

Memory Window Displays the Memory window (p.170) corresponding to the address selected in the trace data area. If an active-state Memory window is already displayed, however, the contents of the window are rewritten.

Address →

Coverage Window

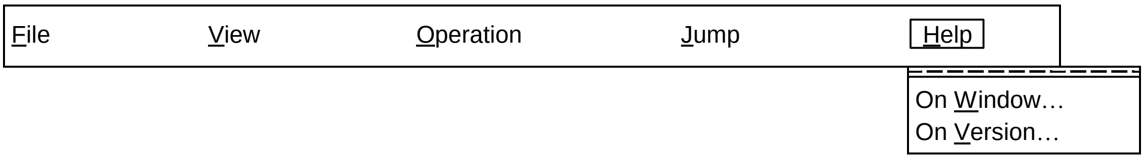
Displays the Coverage window (p.188) corresponding to the address selected in the trace data area. If an active-state Coverage window is already displayed, however, the contents of the window are rewritten.

Address →

Disassemble Window

Displays the Disassemble window (p.201) corresponding to the address selected in the trace data area. If an active-state Disassemble window is already displayed, however, the contents of the window are rewritten.

(5) Help



On Window… Displays the Help dialog box (p.271).

On Version… Displays the Message dialog box (p.131) indicating the ID78K4 version.

Errors

- If an attempt is made to display other windows from the jump menu without any address selected, the system beeps.

Trace Search dialog box

Selection dialog box

Outline

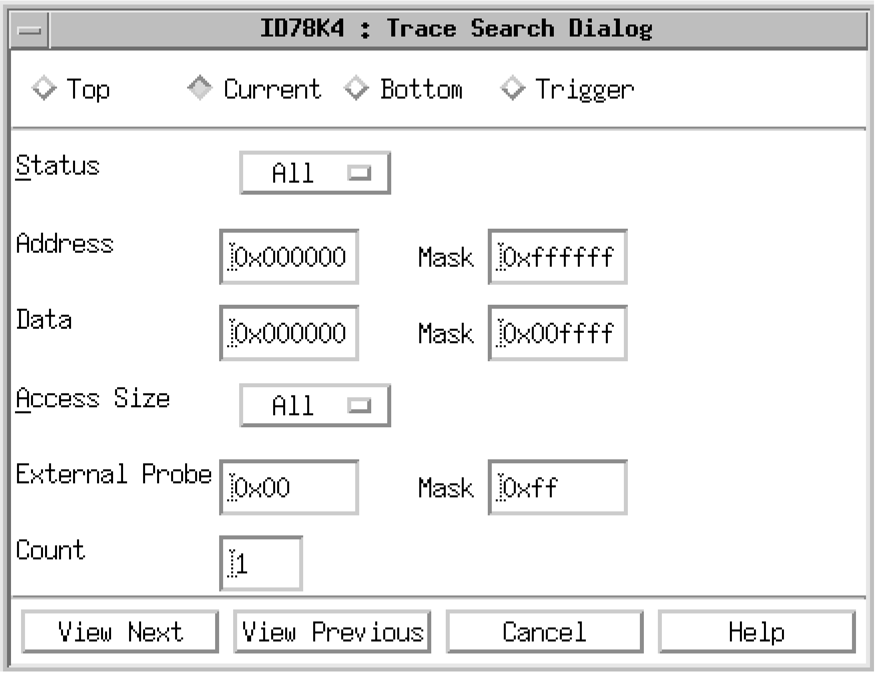
The Trace Search dialog box is used to search for trace data and display the frame corresponding to the searched data.

This dialog box is displayed in the following way:

- In the Trace View window, select View from the menu bar.

  → Click Search... from the pull-down menu.

Window



ID78K4 : Trace Search Dialog

◇ Top ◆ Current ◇ Bottom ◇ Trigger

Status: All

Address: 0x000000 Mask: 0xffffff

Data: 0x000000 Mask: 0x00ffff

Access Size: All

External Probe: 0x00 Mask: 0xff

Count: 1

Description

This window consists of the following components:

Top of client area: Displays radio buttons for specifying the search start position, and radio buttons for specifying the search method.

Middle of client area: Search contents setting area

Bottom of client area: Displays a search count condition and function buttons.

The method of setting search conditions is explained below.

(1) Search position setting area

Set a trace buffer search start position in this area. Select Top, Current, Bottom, or Trigger from this area. Click the toggle switch at the position to be set.

(2) Search condition display specification area

<1> Selecting a status

Option buttons are provided for status selection. Specify a search status condition from the option menu. The following statuses can be specified.

- All
- BRM1
- M1
- VECT
- RW(1)
- R(2)
- W(3)
- RWP(4)
- RP(5)
- WP(6)
- RWM(7)
- RM(8)
- WM(9)
- OP

<2> Address condition setting area

Specify an address condition and a mask value in this area. The following can be specified as the address condition.

Source level	Instruction level
C source line	Address expression
Variable name	Symbol name
Function name	Label
Structure	
Assembler source line	

For a detailed explanation of the address conditions, see Chapter 3.

<3> Data condition setting area

Specify a data condition and a mask value in this area.

<4> Access size setting area

Option buttons are provided for access size setting. Select an access size from the pull-down menu.

<5> External probe condition setting area

Specify external probe conditions in this area.

<6> Search count specification area

Specify a search count in this area.

Buttons

View Next Searches for the next candidate according to the set search condition.

View Previous Searches for the previous candidate according to the set search condition.

Cancel Closes the Trace Search dialog box.

Help Displays the Help dialog box (p.271).

Timer window

View/setting window

Outline

The Timer window is used to display the timer measurement result.

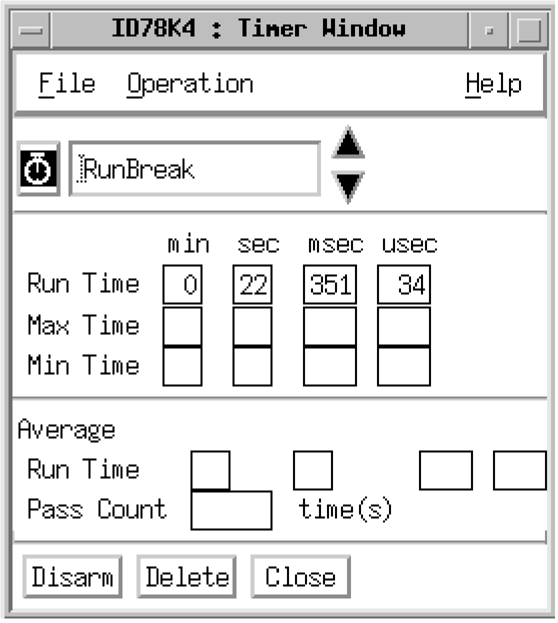
This window is displayed in any of the following ways:

- In the Main window, select Window from the menu bar.

  → Click Timer Window from the pull-down menu.
- In the Windows Box window, click ⏱.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



RunBreak

	min	sec	msec	usec
Run Time	0	22	351	34
Max Time				
Min Time				

Description

This window consists of the following components:

Menu bar

Client area: Timer display area

Button area

The execution time measurement result is displayed.

(1) Timer name display area

Set the timer event, set in advance in the Timer dialog box, in the timer name display area. Move, by dragging & dropping, the icon from the Event Manager window (p.213) to the timer name setting area. Or, use the button to the right of the icon to switch the timer name to another. In this case, the order of the timer name to be displayed complies with the View menu in the Event Manager window (p.213). The timer name can also be modified directly from the keyboard.

(2) Timer display area

This area displays the timer measurement result.

Buttons

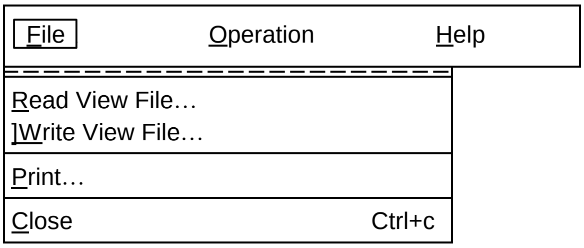
Disarm Disarms the setting of the current timer event.

Delete Deletes the current timer event.

Close Closes the Timer window.

Menu bar

(1) File



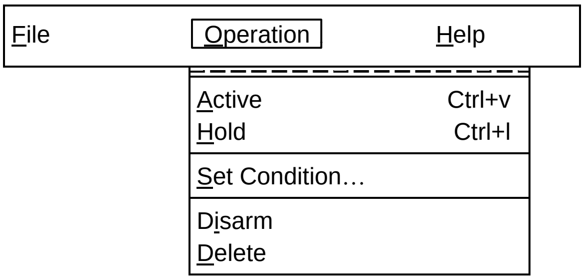
Read View File… Displays the File Selection dialog box (p.104) from which the file containing the contents of this window is specified. Specifying a file from the File Selection dialog box reads (reproduces) the contents of this window from the file. Specify tim as the file extension.

Write View File… Displays the File Selection dialog box (p.104) from which the file that stores the contents of this window is to be specified. Specifying a file from the File Selection dialog box writes the contents of this window into the file. Specify tim as the file extension.

Print… Displays the Prompt dialog box (p.115) from which the print command is to be specified. Specifying the print command from this dialog box prints the contents of the Timer window.

Close Closes the Timer window. This has the same function as the Close button.

(2) Operation



Active Switches the window from the hold state to the active state.

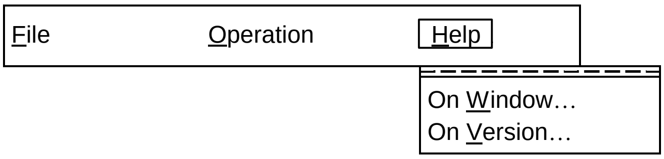
Hold Switches the window from the active state to the hold state.

Set Condition… Displays the Timer dialog box for setting timer conditions.

Disarm Disarms the displayed timer conditions.

Delete Deletes the displayed timer conditions.

(3) Help



On Window… Displays the Help dialog box (p.271).

On Version… Displays the Message dialog box (p.131) indicating the ID78K4 version.

Windows Box window

Auxiliary window

Outline

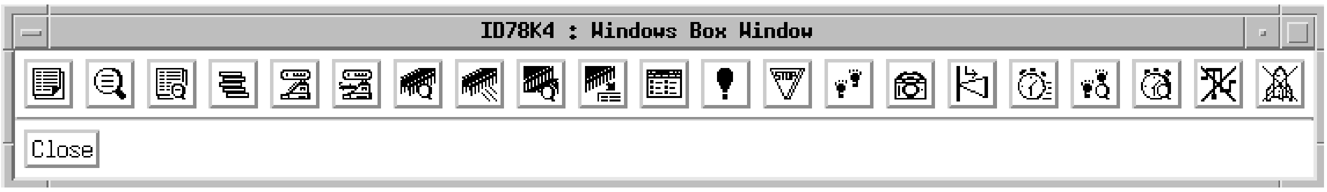
The Windows Box window is provided with function buttons for displaying windows and dialog boxes. These buttons correspond to the pull-down menus displayed by clicking Window on the menu bar of the Main window.

This window is displayed as follows:

- In the Main window, select Window from the menu bar.
  → Click Windows Box Window from the pull-down menu.

Resetting the debugger from the Main window (with CPU & BODY specified) closes this window.

Window



Description

Details of each function button are given below.

Buttons

Opens the Source window (p.120).

Opens the Variables window (p.133).

Opens the Local Variables window (p.144).

Opens the Stack Trace window (p.151).

Opens the Registers window (p.155).

Opens the SFR window (p.163).

Opens the Memory window (p.170).

Opens the Realtime RAM Tracer window (p.184).

Opens the Coverage window (p.188).

Opens the Disassemble window (p.201).

Opens the Event Manager window (p.213).

Opens the Event dialog box (p.222).

Opens the Break dialog box (p.229).

Opens the Trace dialog box (p.233).

Opens the Snap Shot dialog box (p.242).

Opens the Stub dialog box (p.246).

Opens the Timer dialog box (p.249).

Opens the Trace View window (p.252).

Opens the Timer window (p.263).

Error dialog box or Warning dialog box

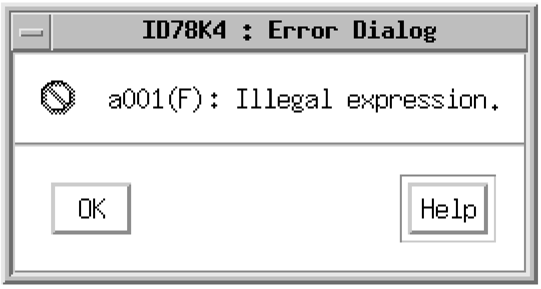
Confirmation dialog box

Outline

The Error dialog box or Warning dialog box is used to display error or warning message.

While this dialog boxes is displayed, all other windows are disabled.

Window



a001(F): Illegal expression.

Description

This dialog box consists of the following component:

Client area: Displays error and warning messages.

Buttons

OK	Closes this dialog box.
Help	Displays the Help dialog box (p.271) that provides help messages for rectifying errors.

Help dialog box

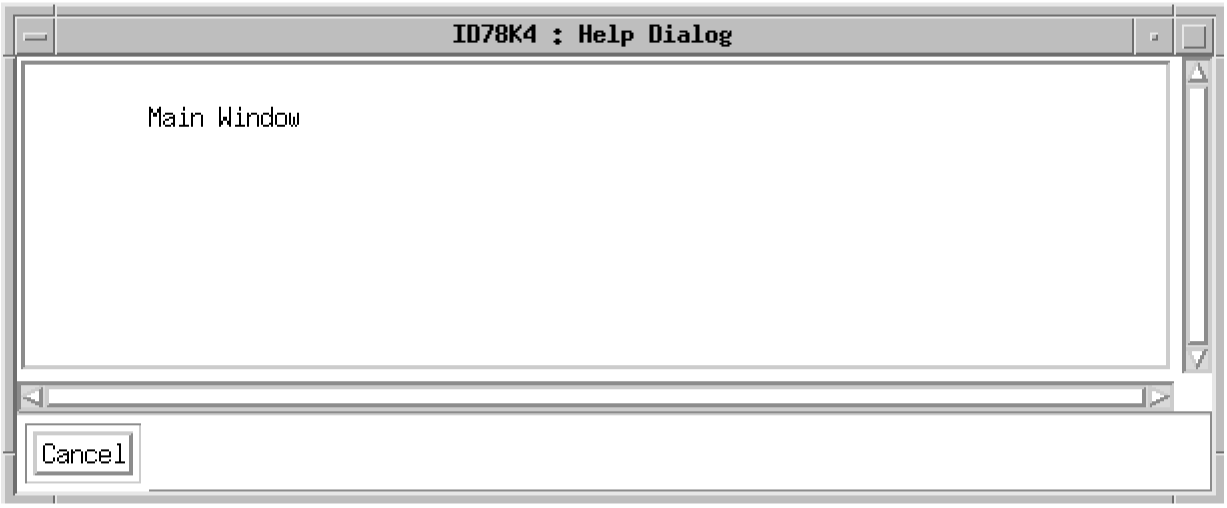
Confirmation dialog box

Outline

The Help dialog box is used to display the explanations of windows and dialog boxes.

While this dialog box is displayed, all other windows are disabled.

Window



Main Window

Description

This dialog box consists of the following component:

Client area: Displays help messages.

Button

Cancel	Closes the Help dialog box.

[MEMO]

NEC

Facsimile Message

From:

Name

Company

Tel. FAX

Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-889-1689

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Corporation
Semiconductor Solution Engineering Division
Technical Information Support Dept.
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: ____________________

Document number: ____________________ Page number: __________

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❏	❏	❏	❏
Technical Accuracy	❏	❏	❏	❏
Organization	❏	❏	❏	❏