

Hitachi Microcomputer Support Software

H8S/2655 E7000PC

Graphical User Interface Software

User's Manual

HITACHI

HS2655G7IW1SE

3/15/96

Preface

The H8S/2655 E7000PC graphical user interface software allows Hitachi's in-circuit emulator E7000PC for H8S/2655 series microcomputers to be used for source-level debugging on Windows*¹. This software thus provides effective application program debugging for systems using the H8S/2655-series MCU.

This manual gives an overview and describes the operating instructions of the H8S/2655 E7000PC graphical user interface software.

Read this user's manual and the following manuals, and understand their contents before using the E7000PC GUI.

For details on the H8S/2655 series in-circuit emulator:

E7000 H8S/2655 Series Emulator User's Manual

Description Notes on Using the IBM PC*² Interface Board (HS7000EII01H) Used for the E7000PC Emulator or the Compact Evaluation Board

For details on software development support tools:H8S, H8/300 Series C Compiler User's Manual

H8S, H8/300 Series Cross Assembler User's Manual

H Series Linkage Editor User's Manual

H Series Librarian User's Manual

- Notes:**
- 1. Windows and MS-DOS are registered trademarks of Microsoft Corporation.**
 - 2. IBM PC is a registered trademark of International Business Machines Corporation.**
 - 3. All other company and product names referred to in this manual are trademarks or registered trademarks of corresponding companies.**

Symbols Used in This Manual

The following symbols are used in this manual:

- ___ (underlining): Input
- < >: Information within < > must be specified.
- []: Parameters enclosed by [] can be omitted.
- ...: The entry specified just before this symbol can be repeated.
- |: Select one of the parameters separated by this symbol or select both.
- ||: Select only one of the parameters separated by this symbol.
- Δ: Indicates a space or a tab.
- (key): Press the key on the keyboard.
- (Enter): Press the ENTER key.
- (Ctrl + C): Press the CONTROL and C keys simultaneously.
- (Alt + B): Press the Alt and B keys simultaneously.
- (Alt + Ctrl + A): Press the Alt, CONTROL, and A keys simultaneously.
- [Command]: Indicates a menu command opening a window.
- [Command...]: Indicates a menu command opening a dialog box.
- <Button>: Indicates a button.
- C>: Indicates the MS-DOS* prompt.

Note: MS-DOS is a registered trademark of Microsoft Corporation.

Data values in this manual are expressed as follows:

- Binary: Prefixed by B'
- Octal: Prefixed by Q'
- Decimal: Prefixed by 0 D'
- Hexadecimal: Prefixed by H'

Data without a prefix is in hexadecimal unless otherwise specified.

The display examples shown in this manual are operating results under Microsoft Windows Version 3.1.

Contents

Section 1 Overview.....	1-1
1.1 Functions and Features	1-1
1.2 Configuration	1-2
1.3 Operating Environment	1-3
1.4 File Configuration.....	1-3
1.5 Debug Target Programs	1-4
1.6 Limitations on Debug Target Programs	1-5
1.7 Limitations on Functions	1-6
Section 2 Installation	2-1
2.1 Installation Procedure	2-1
2.2 Confirming Operating Environment.....	2-2
2.3 Initiating Windows	2-3
2.4 Executing Installer Program	2-4
2.5 Modifying System Files	2-9
2.6 Initiating GUI-2600	2-11
Section 3 Graphical Debugging Environment.....	3-1
3.1 Base Window	3-1
3.1.1 Base Window Configuration	3-1
3.1.2 Menu Bar	3-3
3.1.3 Source Information Area	3-4
3.1.4 Source Area	3-4
3.1.5 Tool Bar.....	3-5
3.1.6 Emulation Information Area	3-6
3.1.7 Optimization Status Area	3-7
3.1.8 Command Area	3-8
3.2 Debug Windows	3-10
3.3 Dialog Box.....	3-12
3.4 Message Box.....	3-14
3.5 Edit Function	3-15
3.5.1 Character String Search.....	3-15
3.5.2 Copy to Clipboard	3-16
3.6 Mouse	3-16
3.7 Key Input	3-17
3.8 Input Format	3-18
3.8.1 Expressions.....	3-18
3.8.2 File Names.....	3-20
3.9 Short-Cut Keys	3-21

Section 4	GUI-2600 Operation	4-1
4.1	Emulator Power-On	4-1
4.2	Initiating GUI-2600	4-1
4.3	Initiating Emulator	4-3
4.4	Loading Load Module	4-5
4.4.1	Allocating Emulation Memory	4-5
4.4.2	Loading Program	4-6
4.5	Displaying Source File	4-7
4.6	Setting and Cancelling Breakpoints	4-9
4.6.1	Setting Breakpoints	4-9
4.6.2	Cancelling Breakpoints	4-10
4.7	Setting Execution Start Address and Stack Pointer	4-11
4.8	Executing Programs	4-12
4.8.1	Executing Programs Using Tool Bar	4-12
4.8.2	Executing Programs Using [Go...] Command	4-13
4.8.3	Terminating Programs	4-14
4.9	Displaying Symbol Contents	4-15
4.10	Inputting Emulator Commands	4-16
4.11	Displaying Help Information	4-17
4.11.1	GUI-2600 Operation Help	4-17
4.11.2	Emulator Command Help	4-18
4.12	Calling MS-DOS	4-19
4.13	Terminating GUI-2600	4-20
Section 5	Command Reference	5-1
5.1	Menu Command Functions	5-1
5.1.1	Filing Functions (File Menu)	5-1
5.1.2	Execution Functions (Execution Menu)	5-2
5.1.3	Trace Functions (Trace Menu)	5-3
5.1.4	Debugging Information Display Functions (View Menu)	5-4
5.1.5	Help Functions (Help Menu)	5-6
5.2	Menu Command Reference Format	5-7
5.3	Filing Functions	5-9
5.3.1	Loading Program File	5-9
5.3.2	Loading Other Files	5-15
5.3.3	Loading Directly into the E7000	5-20
5.3.4	Inputting and Executing Command File	5-25
5.3.5	Storing Command Area Contents	5-28
5.3.6	Terminating GUI-2600	5-32
5.4	Execution Functions	5-36
5.4.1	Specifying Emulation Execution Conditions	5-36
5.4.2	Performing Program Emulation	5-40
5.4.3	Setting and Cancelling Breakpoints	5-46

5.4.4	Specifying Hardware Break Condition1	5-49
5.4.5	Specifying Hardware Break Condition2,3,4	5-55
5.4.6	Specifying Hardware Break Condition5,6	5-60
5.5	Trace Functions	5-63
5.5.1	Specifying Trace Acquisition Mode	5-63
5.5.2	Specifying Trace Acquisition Conditions	5-66
5.5.3	Specifying Trace Information Display Conditions	5-74
5.5.4	Displaying Trace Information in Text.....	5-81
5.5.5	Displaying Trace Information in Graph Form.....	5-88
5.6	Debug Information Display Functions	5-101
5.6.1	Specifying Memory Display Range	5-101
5.6.2	Displaying and Modifying Memory Contents.....	5-103
5.6.3	Displaying and Modifying Register Contents	5-108
5.6.4	Setting Watch Points	5-111
5.6.5	Displaying Watch Point Contents	5-114
5.6.6	Displaying Symbol Information	5-116
5.6.7	Displaying and Modifying Symbol Values	5-121
5.6.8	Setting Source Area.....	5-136
5.6.9	Displaying and Selecting a Source File.....	5-138
5.6.10	Disassembling and Executing Programs	5-140
5.6.11	Displaying Function Call Sequence	5-144
5.6.12	Setting LED Display Conditions.....	5-146
5.6.13	Setting Conditions for Analog Output of LED Display Contents.....	5-150
5.6.14	Controlling Command Area	5-153
5.7	Help Functions.....	5-156
5.7.1	Displaying GUI-2600 Operating Help	5-156
5.7.2	Displaying Emulator Commands	5-157
5.7.3	Describing GUI-2600	5-159
Appendix A E7000 Emulator Command List		A-1
Appendix B Error Messages.....		B-1
B.1	Error Message Levels	B-1
B.2	List of Error Messages.....	B-1
B.2.1	Operation Error Messages	B-2
B.2.2	System Error Messages	B-6
Appendix C Command File		C-1
C.1	Command File Description.....	C-1
C.2	Command File Execution at GUI-2600 Initiation	C-1
Appendix D Initial Setting File (gui26.ini).....		D-1

Figures

1-1	Hardware Configuration	1-2
1-2	Debug Target Programs	1-4
2-1	Installation Procedure	2-1
2-2	Example of Windows Initiation Screen	2-3
2-3	File Name Input (INSTALL).....	2-4
2-4	Installation Directory Specification (INSTALL).....	2-5
2-5	E7000PC System Disk Drive Specification (INSTALL)	2-5
2-6	Addition Confirmation Dialog Box (INSTALL).....	2-6
2-7	AUTOEXEC.BAT File Drive Specification (INSTALL)	2-6
2-8	Specification of Memory Address Range for IBM PC Interface Board (INSTALL)	2-7
2-9	Dialog Box for Confirmation of Memory Address Specification Range Rewrite (INSTALL)	2-8
2-10	Installer Program Termination Message Box (INSTALL).....	2-8
2-11	Program Manager Window and GUI-2600 Initiation	2-11
2-12	Start-Up Display	2-12
2-13	Backup File Load Confirmation Message Box	2-13
2-14	Emulator Status Confirmation Message Box.....	2-13
3-1	Base Window Configuration	3-1
3-2	Command List	3-3
3-3	Source Information Area	3-4
3-4	Source Area	3-4
3-5	Tool Bar	3-5
3-6	Emulation Information Area	3-6
3-7	Optimization Status Area.....	3-7
3-8	Command Area	3-8
3-9	Debug Window Example.....	3-11
3-10	Dialog Box.....	3-12
3-11	Operation Confirmation Message Box	3-14
3-12	Error Message Box	3-14
3-13	Edit Menu.....	3-15
3-14	FIND Dialog Box.....	3-15
3-15	Mouse.....	3-16
3-16	Example of a Dialog Box for File Operation.....	3-20
4-1	Display at Windows Initiation	4-1
4-2	Display at GUI-2600 Initiation	4-2
4-3	Emulator Monitor Message	4-3
4-4	Emulator System Message.....	4-4
4-5	Confirmation Message Box for Backup File Load	4-5
4-6	Example of Emulation Memory Allocation with MAP Command	4-5

4-7	LOAD PROGRAM FILE Dialog Box.....	4-6
4-8	Loading Message Box	4-7
4-9	SOURCE DISPLAY Dialog Box	4-8
4-10	SOURCE AREA SETTING Dialog Box.....	4-8
4-11	Example of Setting Breakpoints	4-9
4-12	Example of Breakpoint Cancellation.....	4-10
4-13	Example of Setting Registers.....	4-11
4-14	Program Execution Example Using <CONTINUE> Button.....	4-13
4-15	GO Dialog Box	4-14
4-16	Example of Forcible Program Execution Termination Using <STOP> Button	4-14
4-17	Example of Symbol Contents Display.....	4-15
4-18	Emulator Command Input Format.....	4-16
4-19	GUI-2600 Operation Help Display	4-17
4-20	Emulator Command Help Display.....	4-18
4-21	MS-DOS Call Procedure	4-19
4-22	MS-DOS Application Execution in Windows.....	4-20
4-23	Confirmation Message for Backup File Storing	4-20
5-1	Menu for Filing Functions (File Menu).....	5-1
5-2	Menu for Execution Functions (Execution Menu)	5-2
5-3	Menu for Trace Functions (Trace Menu).....	5-3
5-4	Menu for Debugging Information Display Functions (View Menu).....	5-4
5-5	Menu for Help Functions (Help Menu)	5-6
5-6	Command Description Format.....	5-7
5-7	LOAD PROGRAM FILE Dialog Box.....	5-9
5-8	LOAD INFORMATION Dialog Box (Load Module File)	5-12
5-9	VERIFY INFORMATION Dialog Box (Load Module File)	5-13
5-10	Verification Error Message.....	5-13
5-11	LOAD OTHER FILE Dialog Box	5-15
5-12	LOAD INFORMATION Dialog Box (Other File).....	5-17
5-13	VERIFY INFORMATION Dialog Box (Other File).....	5-18
5-14	Verification Error Message (Other File).....	5-18
5-15	E7000 LOAD Dialog Box	5-20
5-16	LOAD INFORMATION Dialog Box (E7000 LOAD)	5-23
5-17	VERIFY INFORMATION Dialog Box (E7000 LOAD)	5-23
5-18	Verification Error Message.....	5-24
5-19	INPUT FROM Dialog Box.....	5-25
5-20	Stop Confirmation Message Box (INPUT FROM Dialog Box)	5-27
5-21	OUTPUT TO Dialog Box.....	5-28
5-22	File Overwrite Confirmation Message (OUTPUT TO Dialog Box)	5-30
5-23	QUIT Dialog Box	5-32
5-24	BACKUP FILE Dialog Box	5-33
5-25	File Overwrite Confirmation Message (BACKUP FILE Dialog Box)	5-34

5-26	EXECUTION MODE Dialog Box	5-36
5-27	GO Dialog Box	5-40
5-28	BREAK Dialog Box	5-46
5-29	Display Format for Breakpoint Information (BREAK Dialog Box)	5-46
5-30	BREAK CONDITION1 Dialog Box	5-49
5-31	BREAK CONDITION2 Dialog Box	5-55
5-32	BREAK CONDITION5 Dialog Box	5-60
5-33	TRACE MODE Dialog Box	5-63
5-34	TRACE CONDITION Dialog Box	5-66
5-35	PIN/EXTERNAL PROBE CONDITION Dialog Box	5-70
5-36	TRACE DISPLAY SETTING Dialog Box	5-74
5-37	SEARCH CONDITION Dialog Box.....	5-76
5-38	TRACE DISPLAY TEXT Window.....	5-81
5-39	Display Header Format in Instruction Mnemonic Units (TRACE DISPLAY TEXT Window)	5-82
5-40	Display Header Format in Bus-Cycle Units (TRACE DISPLAY TEXT Window)	5-83
5-41	TRACE OUTPUT Dialog Box	5-85
5-42	File Overwrite Confirmation Message (TRACE OUTPUT Dialog Box)	5-87
5-43	TRAGET SELECTION Dialog Box	5-88
5-44	TRACE DISPLAY GRAPH Window.....	5-92
5-45	Trace Information Display for Address Bus Waveform (TRACE DISPLAY GRAPH Window)	5-95
5-46	Trace Information Display for Data Bus Waveform (TRACE DISPLAY GRAPH Window).....	5-96
5-47	Trace Information Display for Signal Waveform (TRACE DISPLAY GRAPH Window).....	5-97
5-48	Trace Information Display for Memory Change (TRACE DISPLAY GRAPH Window).....	5-98
5-49	Trace Information Display for Execution Time (TRACE DISPLAY GRAPH Window).....	5-99
5-50	Zooming Function Example (TRACE DISPLAY GRAPH Window)	5-100
5-51	MEMORY RANGE SETTING Dialog Box	5-101
5-52	MEMORY DUMP Window	5-103
5-53	Example of Modifying Memory Contents.....	5-105
5-54	MEMORY OUTPUT Dialog Box	5-105
5-55	File Overwrite Confirmation Message (MEMORY OUTPUT Dialog Box).....	5-107
5-56	REGISTER Window	5-108
5-57	Example of Modifying Register Contents (REGISTER Window).....	5-110
5-58	WATCH SETTING Dialog Box	5-111
5-59	Watch Point Display Format (WATCH SETTING Dialog Box)	5-111
5-60	WATCH DISPLAY Window	5-114
5-61	Watch Point Contents Display Format (WATCH DISPLAY Window)	5-115

5-62	SYMBOL DISPLAY Window	5-116
5-63	Example of Execution Stop Address Scope.....	5-117
5-64	Display Format of Symbol Information Display Area (SYMBOL DISPLAY Window).....	5-118
5-65	SYMBOL VALUE Window	5-121
5-66	Symbol Value Display Format (SYMBOL VALUE Window).....	5-122
5-67	Symbol Value Modification Procedure (SYMBOL VALUE Window).....	5-129
5-68	Modification Example of a char Pointer Symbol (SYMBOL VALUE Window)..	5-130
5-69	Modification Example of a long-Type Integer Symbol (SYMBOL VALUE Window)	5-131
5-70	Modification Example of a double Floating-Point Symbol (SYMBOL VALUE Window).....	5-132
5-71	Modification Example of a short-Type Two-Dimensional Array Symbol (SYMBOL VALUE Window).....	5-133
5-72	SOURCE AREA SETTING Dialog Box.....	5-136
5-73	SOURCE DISPLAY Dialog Box	5-138
5-74	DISASSEMBLE Window	5-140
5-75	ROUTE Window	5-144
5-76	LED SETTING Dialog Box.....	5-146
5-77	LED Information Display Format (LED SETTING Dialog Box).....	5-146
5-78	LED OUT SETTING Dialog Box	5-150
5-79	LED_OUT Information Display Format (LED OUT SETTING Dialog Box)	5-151
5-80	Command Area Display	5-153
5-81	GUI OPERATING HELP Window.....	5-156
5-82	EMULATOR COMMAND DISPLAY Window	5-157
5-83	EMULATOR COMMAND HELP Window	5-158
5-84	ABOUT H8S/2600 E7000PC GUI Dialog Box.....	5-159
A-1	LOAD Command Syntax.....	A-5
A-2	INPUT Command Syntax	A-6
C-1	Command File Example	C-1
C-2	Command File Execution at GUI-2600 Initiation	C-2
D-1	Initial Setting File (gui26.ini) Contents	D-1

Tables

1-1	GUI-2600 Operating Environment	1-3
1-2	GUI-2600 File Configuration	1-3
1-3	Limitations on Debug Target Programs.....	1-5
1-4	Limitations on Functions	1-6
3-1	Button Functions.....	3-6
3-2	Special Key Codes in Command Area.....	3-9
3-3	Debug Windows.....	3-10
3-4	Buttons in Dialog Box	3-13
3-5	Short-Cut Keys.....	3-21
4-1	Program Execution Buttons in Tool Bar	4-12
5-1	Output Status Messages (OUTPUT TO Dialog Box).....	5-28
5-2	Output File Status Message and Buttons in OUTPUT TO Dialog Box	5-31
5-3	Information Stored in Backup File (QUIT Dialog Box)	5-35
5-4	Address Range Specification (EXECUTION MODE Dialog Box)	5-38
5-5	Emulation Termination Causes (GO Dialog Box).....	5-44
5-6	Hardware Break Condition Specifications According to Bus Width (Break Condition1)	5-53
5-7	Hardware Break Condition Specifications According to Bus Width (Break Condition2,3,4)	5-58
5-8	Hardware Break Condition Specifications According to Bus Width (Break Condition5,6)	5-62
5-9	Trace Modes.....	5-71
5-10	Trace Acquisition Condition Settings According to Bus Width.....	5-72
5-11	Trace Search Condition Settings According to Bus Width	5-80
5-12	[Signal] List in [Trace display - Graph] Command.....	5-91
5-13	Types of Graph	5-94
5-14	Storage Class Display Format (SYMBOL DISPLAY Window).....	5-118
5-15	Symbol Attribute Display Format (SYMBOL DISPLAY Window).....	5-119
5-16	Symbol Type Display Format (SYMBOL VALUE Window)	5-124
5-17	Floating-Point Display Format (SYMBOL VALUE Window).....	5-125
5-18	Symbol Value Display Examples (SYMBOL VALUE Window).....	5-127
5-19	Symbol Display Field Characters (SYMBOL VALUE Window)	5-129
5-20	File Type and Disassembly Display (DISASSEMBLE Window).....	5-142
A-1	E7000 Emulator Commands	A-1
B-1	Operation Error Messages	B-2
B-2	System Error Messages.....	B-6

Section 1 Overview

1.1 Functions and Features

The H8S/2655 E7000PC graphical user interface software (hereinafter referred to as GUI-2600) allows Hitachi's H8S/2655 E7000PC in-circuit emulator (hereinafter referred to as E7000PC emulator) to be operated on Windows, thus enabling the E7000PC emulator to be used for efficient source-level debugging. The GUI-2600 thus provides effective application program debugging for systems using the H8S/2655-series MCU.

The GUI-2600 provides the following functions and features:

- Source-level debugging
 - Displays the source program (assembly language or C) and execution points.
 - Sets and cancels breakpoints from the source program.
 - Displays the values of symbols selected from the source program in a specific window.
 - Executes a program in units of one source program line displayed in a window (single-step execution).
 - Acquires trace information during program execution, and displays the source program contents corresponding to the trace information.
- Multi-windows
 - Provides multi-windows (displays multiple windows at the same time).
 - Provides frequently used commands in a button form, enabling efficient operation by mouse.
- Powerful debugging functions
 - Displays and modifies memory, register, and symbol contents in windows.
 - Displays function call history.
 - Enables easy program and signal analysis by displaying trace information in a graphical form.

1.2 Configuration

Figure 1-1 shows the hardware configuration required for GUI-2600 operation.

The GUI-2600 operates under the control of Windows on a personal computer. The personal computer must be connected to the E7000PC emulator station via an IBM PC interface board to allow the computer and emulator to interact. The E7000PC emulator station must be connected to the user system via the emulator pod.

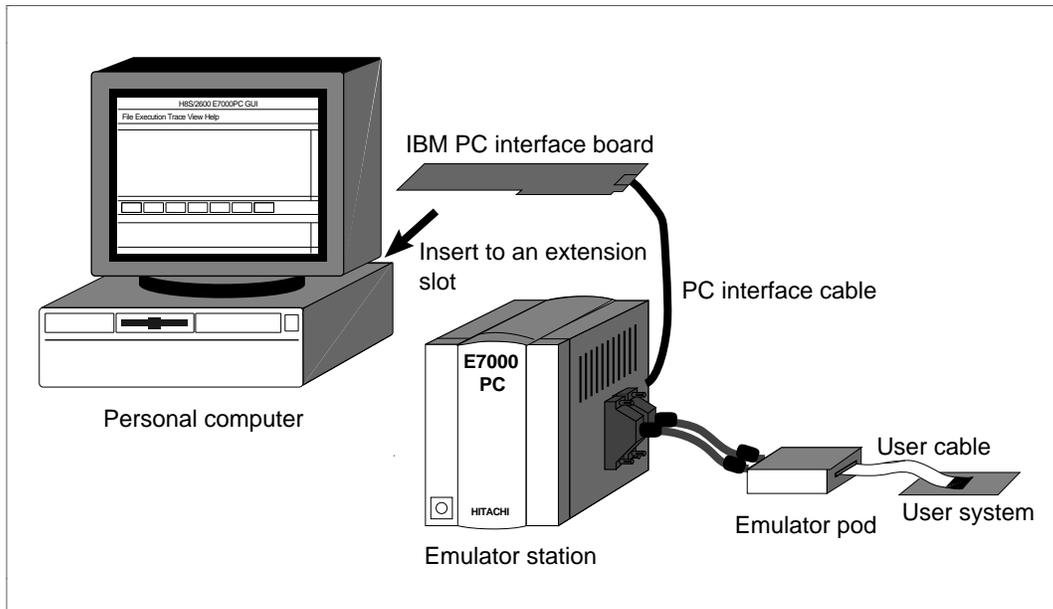


Figure 1-1 Hardware Configuration

1.3 Operating Environment

The GUI-2600 operating environment is shown in table 1-1.

Table 1-1 GUI-2600 Operating Environment

Item	Operating Environment
Personal computer	IBM PC or compatible machine having an 80386 or 80486 CPU, and an AT bus, on which Microsoft Windows Version 3.1 operates
OS	MS-DOS under which Microsoft Windows Version 3.1 operates
Windows	Microsoft Windows Version 3.1
Minimum operating memory	640-kbyte or larger main memory and 8-Mbyte or larger protect memory in 386 enhanced mode
Display	Color or monochrome display which can be connected to the personal computer and can run Windows (VGA or higher is recommended)
Harddisk	2-Mbyte or larger free space available (space for installation)
Mouse	Any mouse which can be connected to the personal computer and can operate under Microsoft Windows Version 3.1

Note: The GUI-2600 may output display contents to a file on the harddisk. If a large amount of data is output, an error message for insufficient memory will be displayed. In this case, make at least 3 Mbytes of free space available on the harddisk.

1.4 File Configuration

The GUI-2600 software is provided on a 1.44-Mbyte floppy disk. The GUI-2600 file configuration is shown in table 1-2.

Table 1-2 GUI-2600 File Configuration

File Name	Contents
GUI26.EXE	GUI-2600 execution file
GUI26OP.HLP	Help file explaining GUI-2600 basic operations
GUI26.INI	File specifying initial values at GUI-2600 initiation
INSTALL.EXE	Program file installing the GUI-2600

1.5 Debug Target Programs

Figure 1-2 shows programs that can be debugged using the GUI-2600. To perform source-level debugging using the GUI-2600, an absolute address type load module file must be created with specifying the debug option with the compiler, assembler, or linkage editor and then it must be loaded to the GUI-2600.

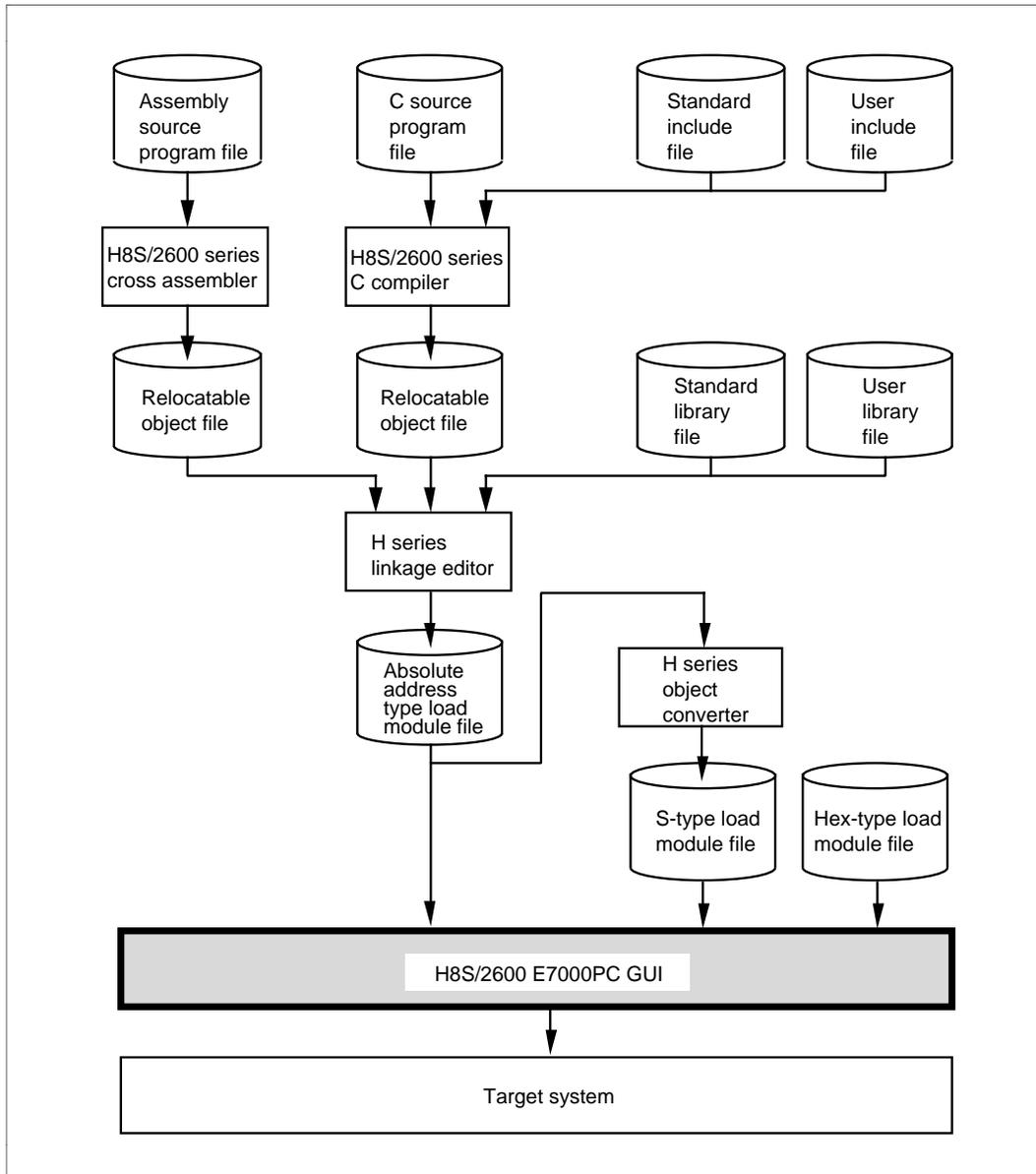


Figure 1-2 Debug Target Programs

1.6 Limitations on Debug Target Programs

The limitations on programs that can be debugged using the GUI-2600 are listed in table 1-3.

Table 1-3 Limitations on Debug Target Programs

Item	Maximum	Remarks
Symbols	16,777,215 symbols	65,535 symbols/unit x 2,339 units (When not limited by memory capacity)
Symbol name length	32 characters	Up to 255 characters can be used for structure symbol names
Units in one segment	2,339 units	
Sections in one segment	4,094 sections	
Functions in one unit	1,364 functions	
Files in one unit	16,383 files	
Lines in one file	32,767 lines	

1.7 Limitations on Functions

The limitations on the GUI-2600 functions are listed in table 1-4.

Table 1-4 Limitations on Functions

Item	Maximum	Remarks
File name length	115 characters	Limited by the MS-DOS file system
Data length	255 characters	
Breakpoints set at one time	255 points	
Waveforms drawn at one time	32 waveforms	
Memory locations changed at one time	256 points	
Watch points	8 points	
Bytes of information displayed for one watch point	256 bytes	
Symbols displayed at one time	16 symbols	For SYMBOL VALUE command
Bytes of symbol contents changed at one time	256 bytes	
Bytes of information displayed for one symbol	20 kbytes	
Nest levels displayed at function call sequence display	64 nests	
Characters input in the command area	248 characters	Limited by the maximum number of command input characters in E7000PC

Section 2 Installation

2.1 Installation Procedure

Install the GUI-2600 using the procedure shown in figure 2-1.

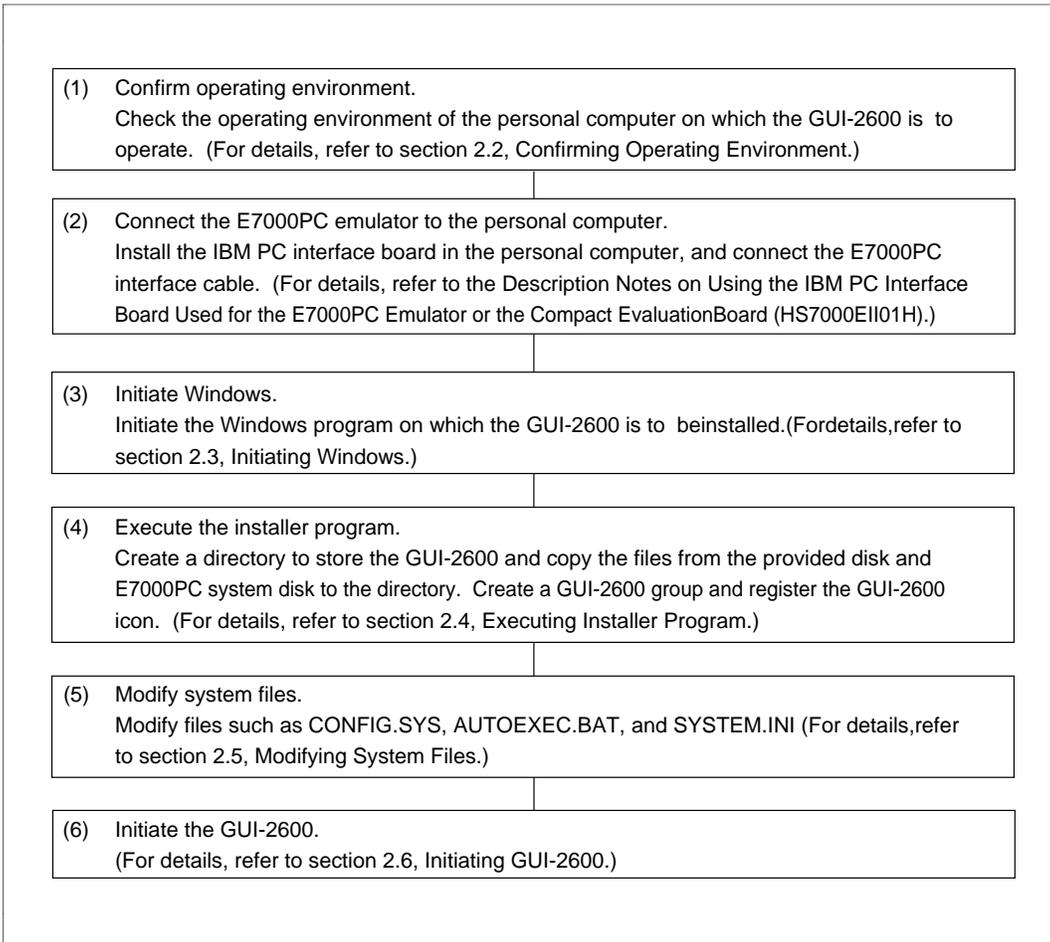


Figure 2-1 Installation Procedure

2.2 Confirming Operating Environment

Check that the personal computer and peripheral devices are correctly connected before installing the GUI-2600. Windows must be correctly installed before the GUI-2600 is installed. Confirm the following before installing the GUI-2600:

- 2 Mbytes or more free space is available on the harddisk to which the GUI-2600 is to be installed.
- Windows has been correctly installed and must be able to be initiated in 386 enhanced mode.

2.3 Initiating Windows

Initiate Windows using the following procedure:

- (1) Initiate MS-DOS from the harddisk.
- (2) Initiate Windows from the harddisk. After Windows has been installed and placed in subdirectory \windows on harddisk drive C, enter as follows:

```
C:\>cd\windows (Enter)  
C:\WINDOWS>win (Enter)
```

If Windows has been installed in another drive or subdirectory, change the current directory to that subdirectory, and then enter win (Enter).

If the subdirectory in which Windows has been installed is registered as a path name, the current directory change command (CD command) need not be executed.

Once Windows is initiated, the display shown in figure 2-2 appears. (The display may be different depending on the Windows environment.)

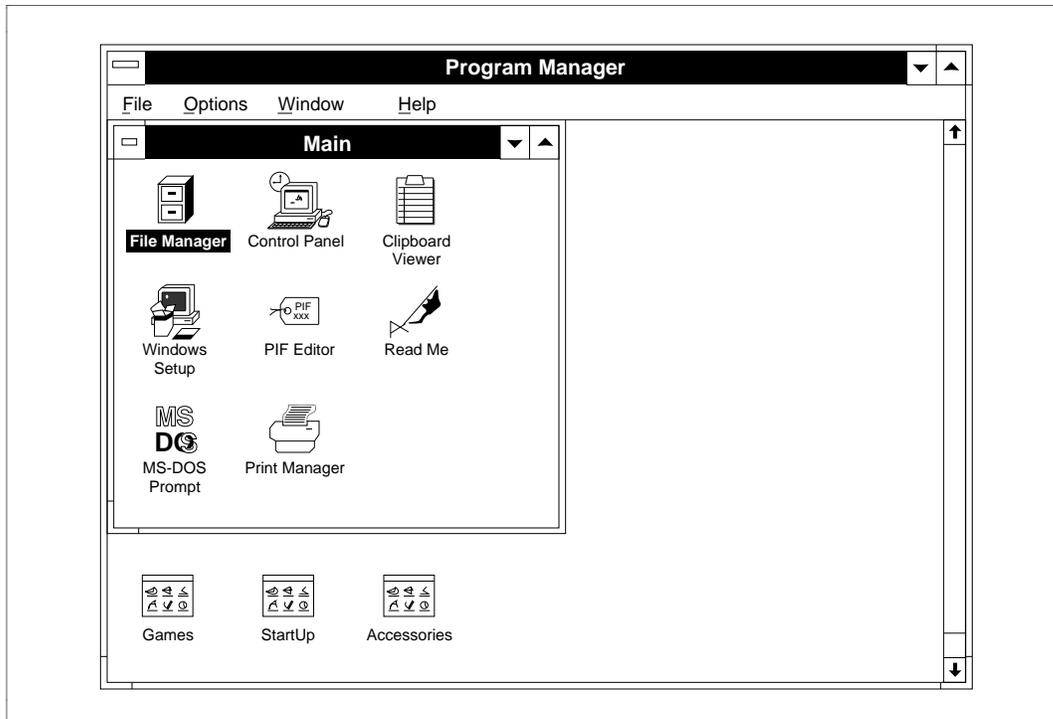


Figure 2-2 Example of Windows Initiation Screen

2.4 Executing Installer Program

The GUI-2600 is installed on Windows by initiating program INSTALL.EXE from the provided disk. Before initiating the INSTALL.EXE, terminate all Windows applications other than the Program Manager.

Initiate the INSTALL.EXE by the following procedure:

- (1) Insert the provided disk into floppy disk drive A.
- (2) Click the [File] menu (icon) on the Program Manager window.
- (3) Select the [Run...] command (specifies a file name and executes it) by clicking on it; the Run dialog box will be displayed.
- (4) Enter the file name of the installer program (a:\INSTALL.EXE) in the [Command Line:] text box as shown in figure 2-3.

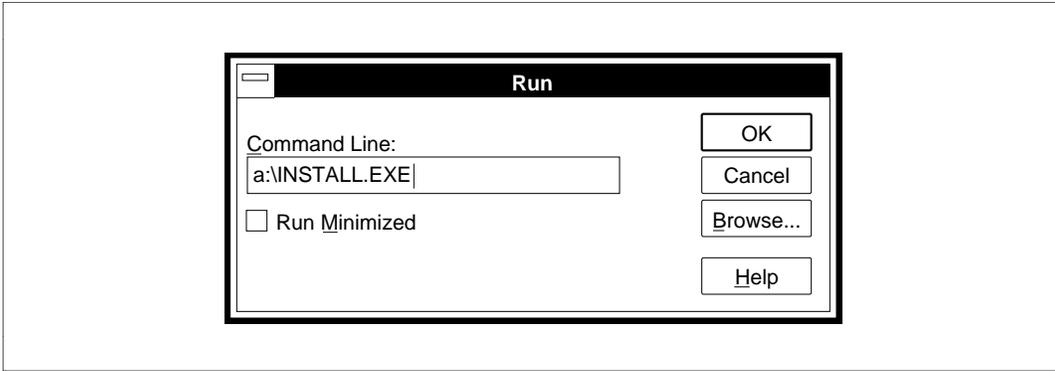


Figure 2-3 File Name Input (INSTALL)

- (5) Click the <OK> button to initiate the installer program.
- (6) When the installer program is initiated, a dialog box is displayed prompting the user to specify the directory in which the GUI-2600 is to be installed, as shown in figure 2-4. The installer program copies the files necessary for GUI-2600 execution to the directory specified in this dialog box. When the files already exist in the specified directory, a message box is displayed to confirm whether the user wants to overwrite the files in the directory.



Figure 2-4 Installation Directory Specification (INSTALL)

- (7) Enter the directory name and click the <Install> button, which starts copying the necessary files for GUI-2600 execution.
- (8) When the files necessary for GUI-2600 execution have been copied, a dialog box is displayed prompting the user to locate the E7000PC system disk, as shown in figure 2-5. Insert the E7000PC system disk into a floppy disk drive and click the <OK> button, which starts copying the E7000PC system files.

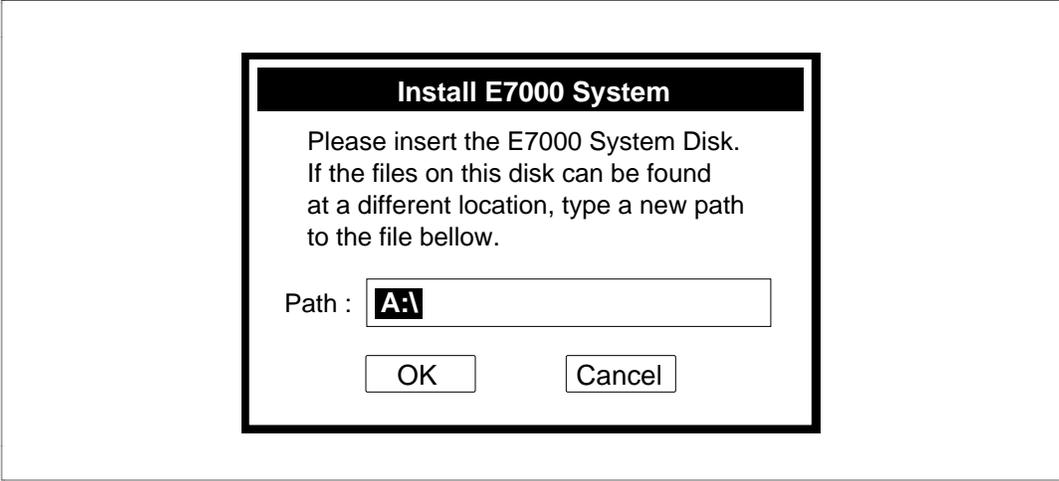


Figure 2-5 E7000PC System Disk Drive Specification (INSTALL)

- (9) When the E7000PC system disk has been copied, a dialog box is displayed to confirm whether the user wants to add to the AUTOEXEC.BAT file the path name of the directory in which the GUI-2600 has been installed and whether the user wants to modify the SYSTEM.INI file, as shown in figure 2-6.

To add the path name to the AUTOEXEC.BAT file or modify the SYSTEM.INI file using the installer program, click the <OK> button. To terminate the installer program, click the <Cancel> button. In this case, change the required file using the procedure described in section 2.5, Modifying System Files.

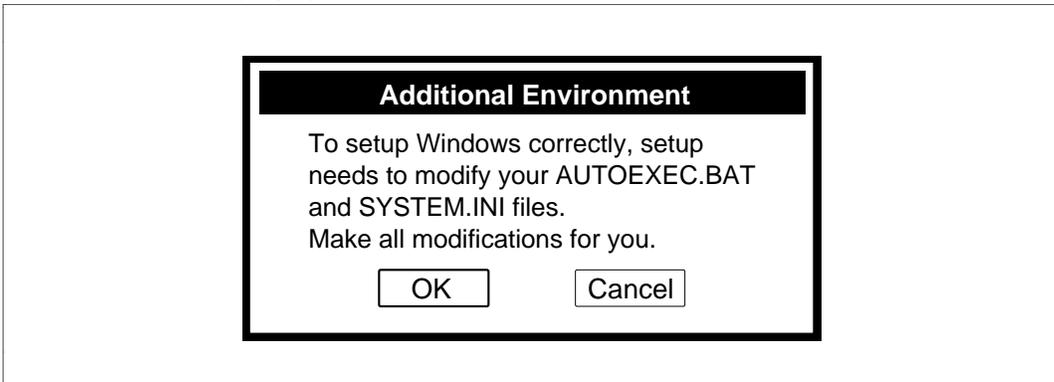


Figure 2-6 Addition Confirmation Dialog Box (INSTALL)

- (10) Click the <OK> button; a dialog box is displayed prompting the user to specify the drive storing the AUTOEXEC.BAT file, as shown in figure 2-7. Click or input from the keyboard the drive name storing the AUTOEXEC.BAT file.

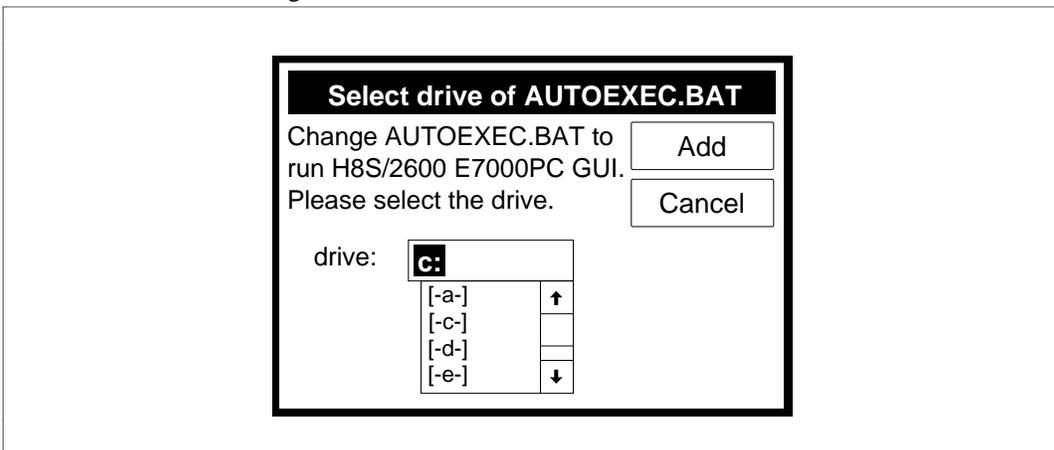


Figure 2-7 AUTOEXEC.BAT File Drive Specification (INSTALL)

- (11) To add to the AUTOEXEC.BAT file, environment variable GUI26PATH and the directory path in which the GUI-2600 has been installed, click the <Add> button. The file before adding the path name is saved as AUTOEXEC.GUI.

To go to the next step without changing the AUTOEXEC.BAT file, click the <Cancel> button. In this case, change the AUTOEXEC.BAT file using the procedure described in section 2.5, Modifying System Files.

- (12) A dialog box is displayed to select the memory address range for the IBM PC interface board, as shown in figure 2-8. Specify the memory address range. For example, select "D000-D3FF" to specify the range from D000:0000 to D3FF:000F.

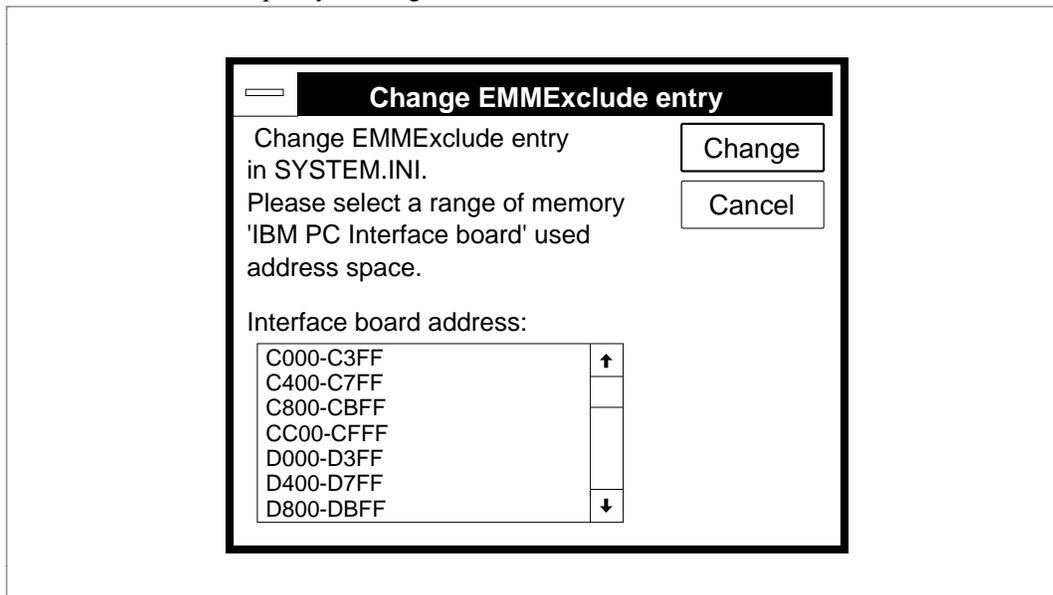


Figure 2-8 Specification of Memory Address Range for IBM PC Interface Board (INSTALL)

- (13) To add to the Windows system file SYSTEM.INI the memory address range for the IBM PC interface board, click the <Change> button. The file before adding the memory address is saved as SYSTEM.GUI. To terminate the installer program without changing the SYSTEM.INI file, click the <Cancel> button. In this case, change the SYSTEM.INI file using the procedure described in section 2.5, Modifying System Files.

(14) To add the memory address range for the IBM PC interface board, the EMMExclude parameter of the SYSTEM.INI file is used. If the memory address range has already been specified, a rewrite confirmation dialog box is displayed, as shown in figure 2-9. Select <Replace> or <Cancel> according to the operating personal computer.

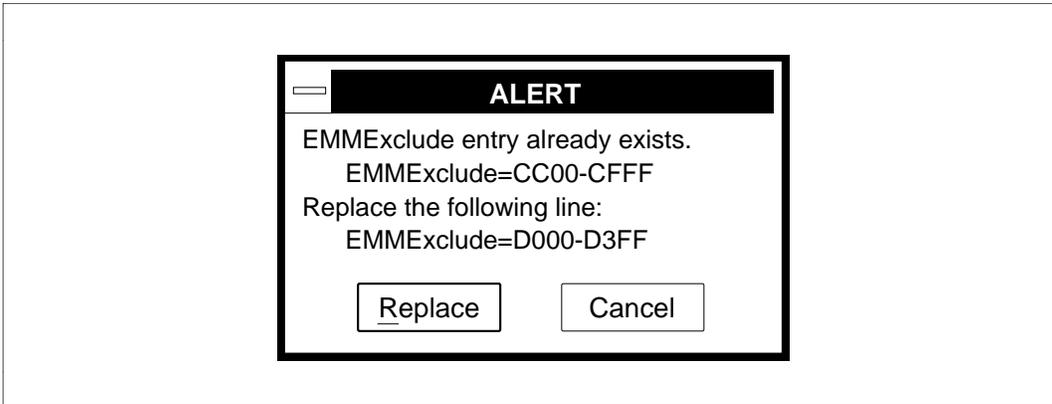


Figure 2-9 Dialog Box for Confirmation of Memory Address Specification Range Rewrite (INSTALL)

(15) The installer program terminates.

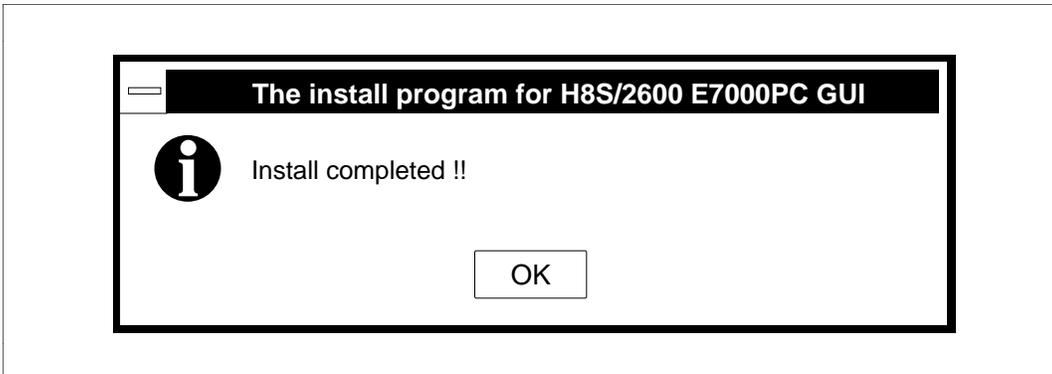


Figure 2-10 Installer Program Termination Message Box (INSTALL)

2.5 Modifying System Files

CAUTION

Back up the CONFIG.SYS, AUTOEXEC.BAT, and SYSTEM.INI files before modifying the files.

(1) Terminate Windows.

Select the [Exit Windows...] command from the [File] menu of the Program Manager to terminate Windows.

(2) Modify CONFIG.SYS.

The GUI-2600 accesses the IBM PC interface board. When a virtual EMS driver installation is specified in the CONFIG.SYS file, the memory addresses for the IBM PC interface board must be set outside the range managed by the virtual EMS driver.

An example of how to set the memory addresses for the IBM PC interface board to D000:0000-D3FF:000F, which is outside the range managed by the virtual EMS driver (EMM386) and how to specify the base address of the EMS page frame as E000:0000, is shown below (enter the underlined part).

```
DEVICE=C:\WINDOWS\EMM386.EXE 1024 RAM x=D000-D3FF frame=E000
```

Note: Add the underlined part.

Specify the memory address range selected in (12) of section 2.4, Executing Installer Program. The memory addresses for the IBM PC interface board can be changed using the DIP switch on the board. For details, refer to the Description Notes on Using the IBM PC Interface Board Used for the E7000PC Emulator or the Compact Evaluation Board.

(3) Modify AUTOEXEC.BAT.

When the AUTOEXEC.BAT file has not been modified by the installer program, set the directory path and environment variable in the AUTOEXEC.BAT file. If the directory name for the GUI-2600 described in section 2.4, Executing Installer Program, is GUI26, modify as follows:

```
PATH C:\;C:WINDOWS;C:\GUI26  
SET GUI26PATH=C:\GUI26  
SET GUI26SRC=C:\PROGRAM\;C:\PROGRAM\SRC\
```

Note: Add the underlined parts.

Environmental variable GUI26SRC specifies the directory containing the source file. When the source file directory at load module file creation and the current source file directory are different, make sure to specify the current source file directory.

(4) Modify SYSTEM.INI

When the memory address range for the IBM PC interface board has not been specified using the installer program, modify the SYSTEM.INI file, which is a Windows initialization file, to specify the memory address range for the IBM PC interface board outside the Windows memory management area. The SYSTEM.INI file exists in the Windows directory. Specify the EMMEExclude in the [386Enh] section in the SYSTEM.INI file. To specify the memory address range for the IBM PC interface board [D000:0000-D3FF:000F] outside the Windows memory management area, modify as follows:

```
[386Enh]  
EMMEExclude = D000-D3FF
```

Note: Add the underlined part.

(5) Initiate MS-DOS.

When MS-DOS file modification is completed, temporarily terminate MS-DOS, then re-initiate it.

2.6 Initiating GUI-2600

Initiate the GUI-2600 using the following procedure:

- (1) Initiate Windows. Immediately after GUI-2600 installation, temporarily terminate Windows and MS-DOS, and then re-initiate them.
- (2) Power on the emulator.
- (3) Using the left button on the mouse, double-click the GUI-2600 icon in the GUI-2600 group window of the Program Manager, as shown in figure 2-11.

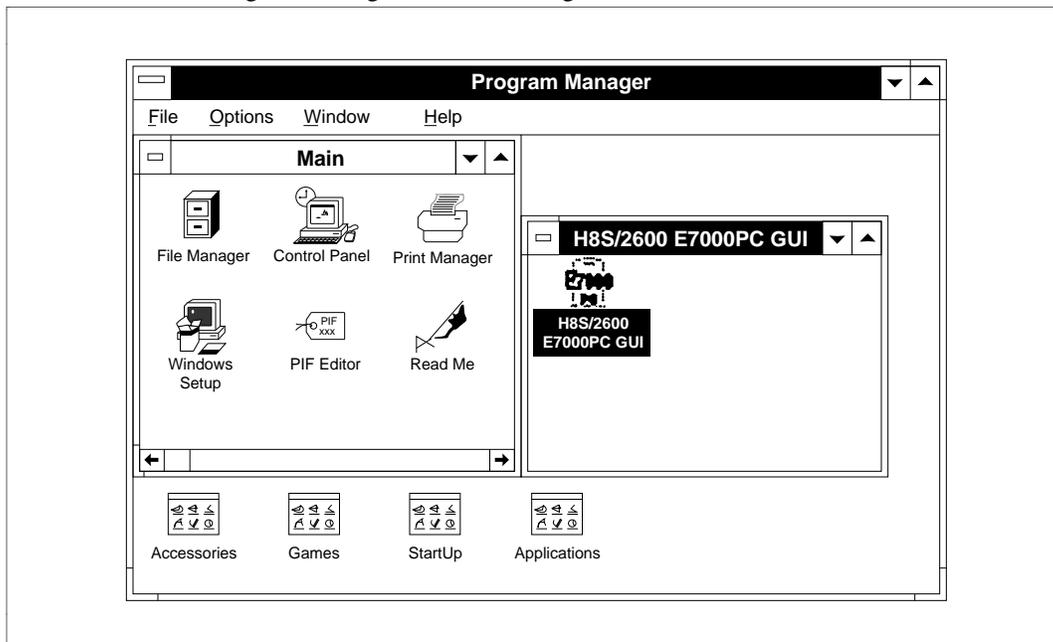


Figure 2-11 Program Manager Window and GUI-2600 Initiation

- (4) When the GUI-2600 is initiated, the display shown in figure 2-12 appears. The E7000PC emulator start-up message is displayed at the bottom, and the system waits for monitor command input. Enter the S command. For other monitor commands, refer to the E7000 H8S/2655 Emulator User's Manual. When initiation is completed, the WARM/COLD start selection request message is displayed. When the (Enter) key is pressed to select the COLD start, the GUI-2600 prompts for emulator command input by displaying a colon (:).

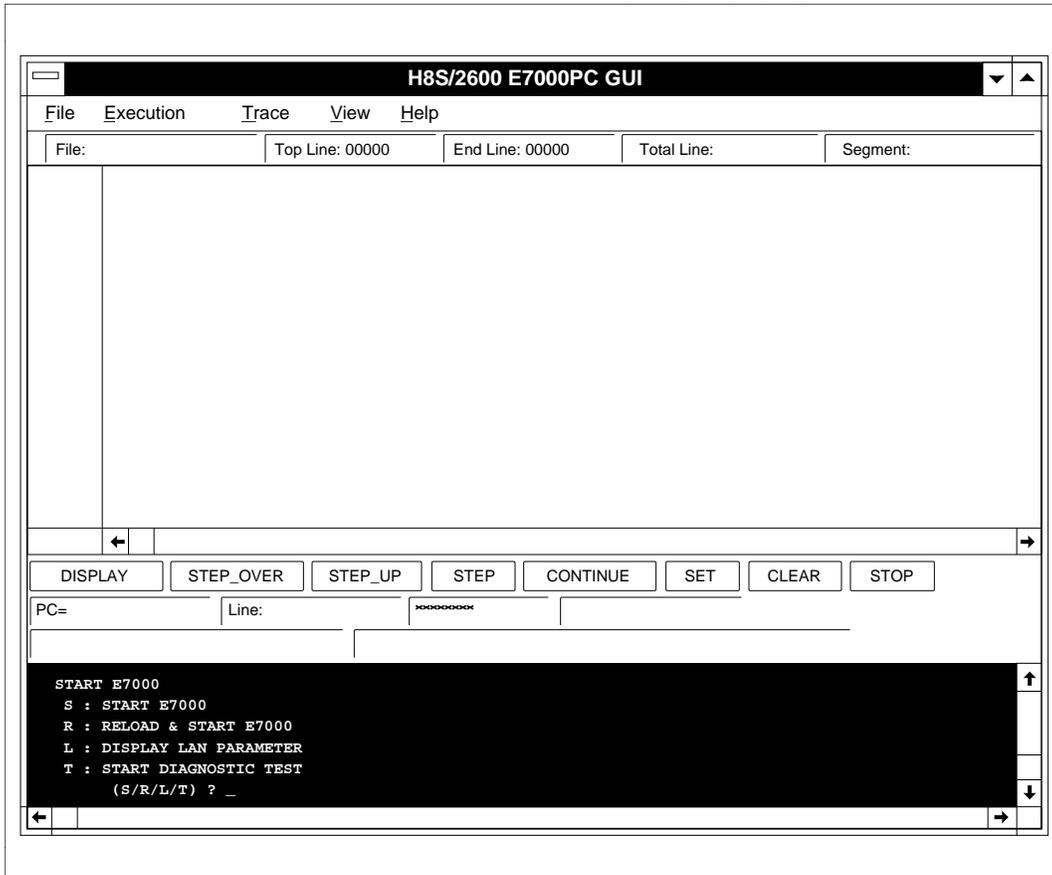


Figure 2-12 Start-Up Display

- (5) When a backup file exists, the message box shown in figure 2-13 is displayed to confirm whether to load the operating environment backup file. To load the operating environment backup file previously saved, click the <Yes> button. To initiate the GUI-2600 without loading the backup file, click the <No> button.

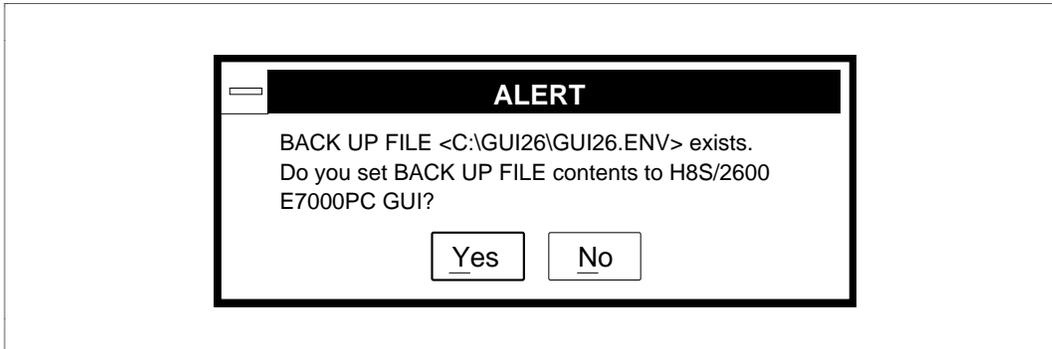


Figure 2-13 Backup File Load Confirmation Message Box

- (6) If the E7000PC emulator power is off, or if the IBM PC interface board or PC interface cable is not correctly connected, the message box shown in figure 2-14 is displayed. In this case, power on the E7000PC emulator and click the <Retry> button, or terminate the GUI-2600 by clicking the <Cancel> button and check the hardware settings.

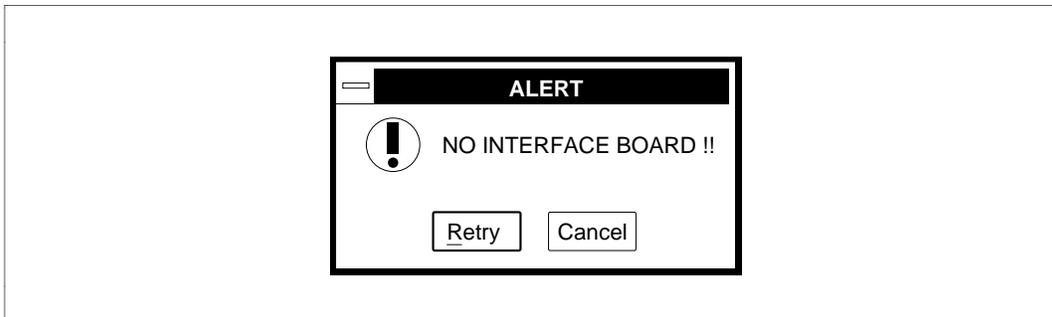


Figure 2-14 Emulator Status Confirmation Message Box

Section 3 Graphical Debugging Environment

This section describes each window and operation of the GUI-2600.

3.1 Base Window

3.1.1 Base Window Configuration

After GUI-2600 initiation, the base window shown in figure 3-1 opens.

The base window provides source-level debugging functions such as displaying the source file and symbol contents, executing programs, and setting and cancelling breakpoints.

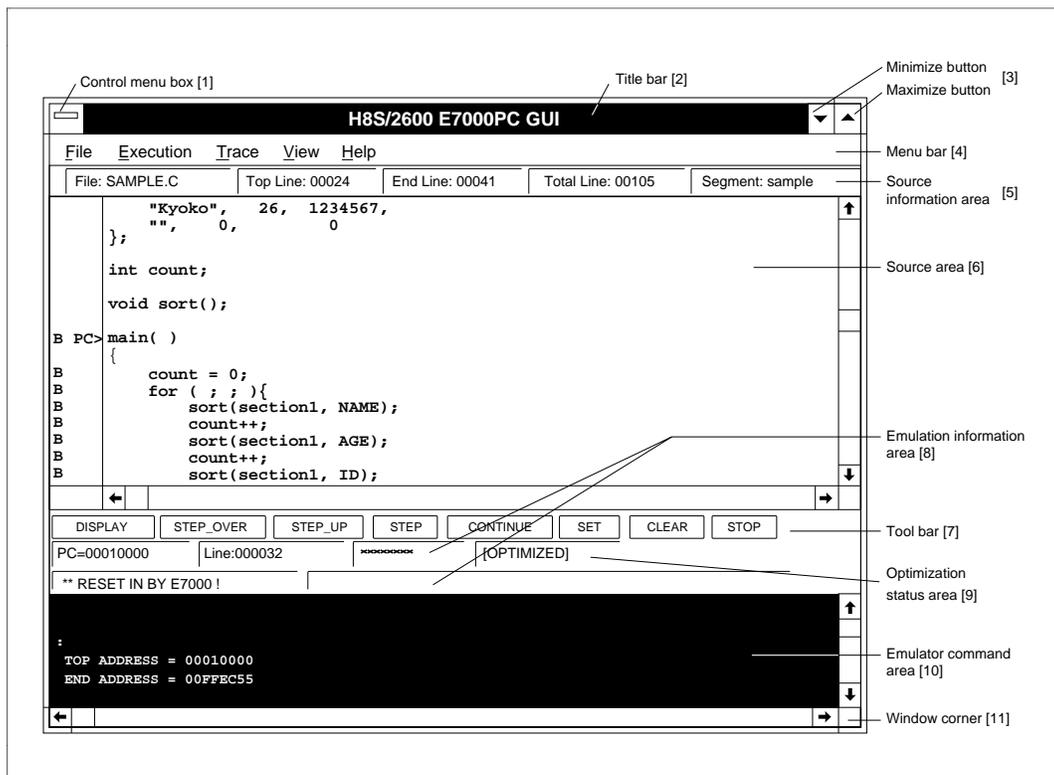


Figure 3-1 Base Window Configuration

The base window consists of the following:

(1) Control menu box

Clicking this box opens the base window control menu.

(2) Title bar

The title of the GUI-2600 program is displayed. Clicking the title bar activates the window. When active, the window can receive key entries and commands.

(3) Minimize button and maximize button

Clicking the minimize button reduces the base window to an icon. Clicking the maximize button enlarges the base window to its maximum display size.

(4) Menu bar

Displays command menus available for use. Clicking a command menu displays a list of commands included in the command menu.

(5) Source information area

Displays information on the source file displayed in the source area.

(6) Source area

Displays the source file being debugged.

(7) Tool bar

Displays buttons for frequently used commands. Clicking a button directly executes the command.

(8) Emulation information area

Displays emulation information during program execution or at termination.

(9) Optimization status area

Displays whether or not the source file displayed in the source area has been optimized.

(10) Emulator command area

Area for inputting emulator commands. The command area size can be changed by the [Command area] command in the View menu.

(11) Window corner

Dragging the corner changes the window size. Changing the window size reduces or enlarges the source area.

Note: Tool bar buttons will be obstructed from view when the window size is reduced.

3.1.2 Menu Bar

The menu bar displays command menus available for use.

Place the mouse pointer on a menu name displayed on the menu bar and click the left button of the mouse; a command list as shown in figure 3-2 is displayed.

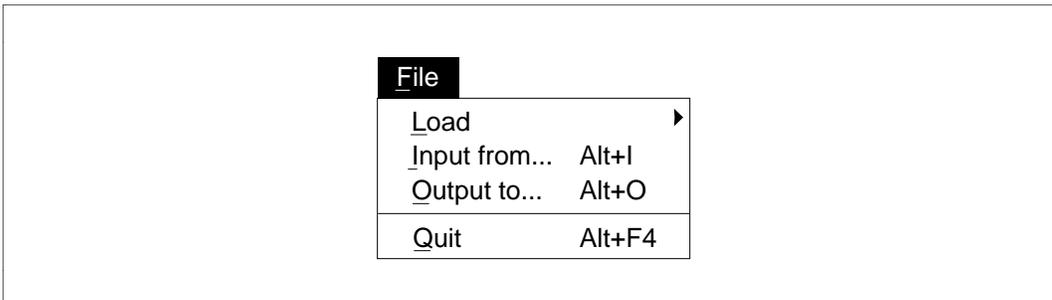


Figure 3-2 Command List

Commands in command lists can be executed by the following procedure:

Click the command to be executed using the left button of the mouse, or drag the mouse until the command name becomes highlighted and release the mouse button.

Symbols used in command lists indicate the following meaning:

- <Command>...: Displays a dialog box.
- <Command>▶ : The command has a subcommand menu.
- <Command>: The command can also be executed by entering only the underlined part of the command name from the keyboard.

When the selected command has already been executed and the window is open, the window is brought to the top of the screen.

The key name, such as (Alt + F4), displayed on the right of each command name in the command list indicates the short-cut keys for the command. Inputting the short-cut keys executes the corresponding command without even having to open the menu bar.

3.1.3 Source Information Area

The source information area displays information on the source program displayed in the source area. Figure 3-3 shows the source information area. This area displays the following information:

[1]	[2]	[3]	[4]	[5]
File: SAMPLE.C	Top Line: 00024	End Line: 00041	Total Line: 00105	Segment: sample

Figure 3-3 Source Information Area

- (1) File: (source file name)
Source program file name
- (2) Top Line: (first line number)
Line number of the first line displayed
- (3) End Line: (last line number)
Line number of the last line displayed
- (4) Total Line: (total number of lines)
Total number of lines in the source program
- (5) Segment: (segment name)
Load module name of the source program

3.1.4 Source Area

The source area displays a section of the source file corresponding to the current program counter (PC). Figure 3-4 shows the source area display.

```
B PC> int ca_gcm(int data1, int data2)
      {
      int rem;

      if (data1 < 0)          /* data1 convert*/
        data1 = 0 - data1;

      if (data2 < 0)          /* data2 convert*/
        data2 = 0 - data2;
      }
```

Figure 3-4 Source Area

When a program is loaded by the [Load - Load program file...] command in the File menu, a section of the source file corresponding to the execution start address of the program is displayed. When program execution stops, for example, at a breakpoint, the section of the source file corresponding to that address is displayed.

The displayed source file can be scrolled using the scroll bar in the source area. The displayed source file can be changed using the [Source - Display...] command in the View menu. In addition, the font, font size, and tab width of the source file can be changed using the [Source - Setting...] command in the View menu.

The PC mark (PC>) indicating the current program counter location and B marks (B) indicating lines where breakpoints can be set are displayed at the left of the source file. After setting the breakpoint, BP marks (BP) are displayed to indicate lines to which breakpoints are set.

By clicking the source program in the source area with the left button of the mouse, a cursor (|) is displayed to indicate the clicked position. The <CONTINUE>, <SET>, and <CLEAR> buttons in the tool bar operate after the cursor is displayed.

Double-clicking a word in the source program highlights and selects the word. Dragging with the left button of the mouse highlights and selects text. The <DISPLAY> button in the tool bar, the character string search function, and function copying the text to the clipboard operate on the selected text.

3.1.5 Tool Bar

The tool bar consists of buttons corresponding to frequently used commands. Clicking a button executes the corresponding command. Figure 3-5 shows the buttons in the tool bar.



Figure 3-5 Tool Bar

Button functions are listed in table 3-1.

Table 3-1 Button Functions

Button	Short-Cut Keys	Function
<DISPLAY>	(Alt + N)	Adds the symbol selected in the source area to the SYMBOL VALUE window.
<STEP_OVER>	(F7)	Executes one program line pointed to by the current program counter (PC) and then stops. When the line includes a function (subroutine) call, program execution stops after executing the function. This button cannot be used in the ROM area.
<STEP_UP>	(F8)	Executes the function including the current program counter (PC), and stops execution when returning to the upper function (calling function). This button cannot be used in the ROM area.
<STEP>	(F9)	Executes one program line pointed to by the current program counter (PC) and stops. When the line includes a function (subroutine) call, program execution stops at the first line of the function.
<CONTINUE>	(Alt + G)	Executes the program from the line pointed to by the current program counter (PC). Execution continues until a line including the cursor is reached. Note that execution stops only when a line marked with B in the source area is clicked.
<SET>	(Alt + B)	Sets a breakpoint to a line including a cursor in the source area. A breakpoint can be specified only to a line marked with B in the source area.
<CLEAR>	(Alt + C)	Cancels the breakpoint set to a line including a cursor in the source area.
<STOP>	(Ctrl + C)	Forcibly stops program execution.

3.1.6 Emulation Information Area

The emulation information area displays emulation information during program execution or at execution termination. Figure 3-6 shows the emulation information area.

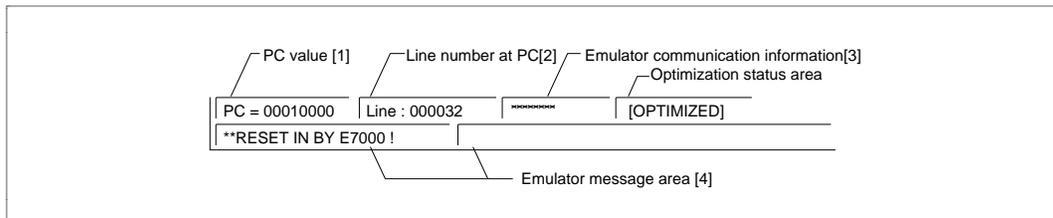


Figure 3-6 Emulation Information Area

The emulation information area displays the following information:

(1) PC = xxxxxxxx

Displays the current program counter value. During program execution, the value is updated to the current program counter value. At execution termination, the program counter value indicates the next address to be executed.

(2) Line:

Displays the line number of the source file corresponding to the program counter value at execution termination.

(3) Emulator communication information

Communication status between the GUI-2600 and the emulator is displayed as follows:

ACCESS: Sending a command to the emulator

PROCESS: Receiving a command from the emulator

*****: Not communicating with the emulator

(4) Emulator message

Displays the cause of program execution termination and execution time. When <Enable> is specified in [Display memory trace data] of the [Go...] command, the contents of the memory address specified in [Trace Memory condition] of the [Trace mode...] command is displayed during emulation.

3.1.7 Optimization Status Area

The optimization status area displays [OPTIMIZED] as shown in figure 3-7 if the program displayed in the source area has been optimized. If not, nothing is displayed.

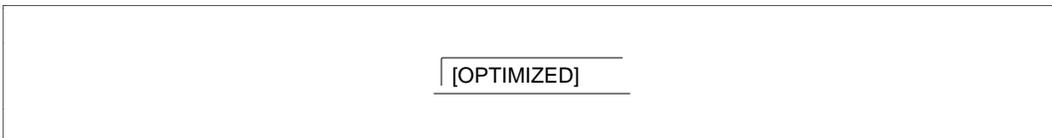


Figure 3-7 Optimization Status Area

3.1.8 Command Area

The command area is used for inputting emulator commands. Figure 3-8 shows the command area.



Figure 3-8 Command Area

Input emulator commands in the command area using the following procedure:

- (1) Click the command area using the left button of the mouse to display the cursor (_).
- (2) When the emulator command input prompt (: or #) is not displayed, press the (Enter) key.
- (3) After the prompt, input emulator commands.

All command execution results are displayed in the command area. Former execution results can be referenced by scrolling the display with the scroll bar in the command area.

The command area window size can be changed by the [Command area] command in the View menu.

Special key codes can be used in the command area as shown in table 3-2.

By using the history function ((Alt + ↑) keys or (Alt + ↓) keys), the previously input command can be displayed, edited, and executed.

For the emulator commands that can be used in the command area, refer to appendix A, E7000 Emulator Command List.

Table 3-2 Special Key Codes in Command Area

Buttons	Function
(Ctrl + C)	Stops command execution or program execution.
(Back space)	Inserts a backspace.
(Delete)	Deletes the character at the cursor.
(Esc)	Cancel the characters on the input line.
(Home)	Moves the cursor to the beginning of the input line.
(End)	Moves the cursor to the last character on the input line.
(Page Up)	Scrolls up by one page.
(Page Down)	Scrolls down by one page.
(↑)	Scrolls up by one line.
(↓)	Scrolls down by one line.
(Alt + ↑)	Displays the previously input commands in reverse order (history function).
(Alt + ↓)	Displays the previously input commands in order (history function).
(←) or (→)	Moves the cursor left or right.
(Space)	Enters parallel mode when pressed during GO command execution.
(Enter)	Executes the command. Enters parallel mode when pressed during GO command execution
(Ctrl + Tab)	Switches the cursor display to the source area (If this key code is input again, the cursor display returns to the command area.)

3.2 Debug Windows

The GUI-2600 includes the debug windows shown in table 3-3 in addition to the base window, which can be opened at the same time during debugging.

Each window opens by a corresponding command. To close a window, click the <Cancel> button in the window or select the [Close] command in the control menu box.

In some of the windows, the contents of the window can be output to a file or specific text in the windows can be searched for.

Table 3-3 Debug Windows

Window	Function	File Output	Text Search
Trace text	Displays trace information as text	Enabled	Enabled
Trace graph	Displays trace information as a graph	Disabled	Disabled
Memory dump	Displays memory contents in dump format and modifies memory contents	Enabled	Enabled
Watch	Displays contents of the watch point	Disabled	Enabled
Register	Displays and modifies register contents	Disabled	Enabled
Symbol display	Displays symbol attributes	Disabled	Enabled
Symbol value display	Displays and modifies symbol values	Disabled	Enabled
Disassembly display	Disassembles and displays a program in the source area	Disabled	Enabled
Function call history display	Displays function call history up to the one including the current program counter value	Disabled	Enabled
Emulator help	Displays a list of emulator commands	Disabled	Enabled
Emulator command help	Displays the input format of emulator commands	Disabled	Enabled
GUI operating help	Displays how to use graphical debugging environment	Disabled	Enabled

Figure 3-9 shows an example of debug window configuration.

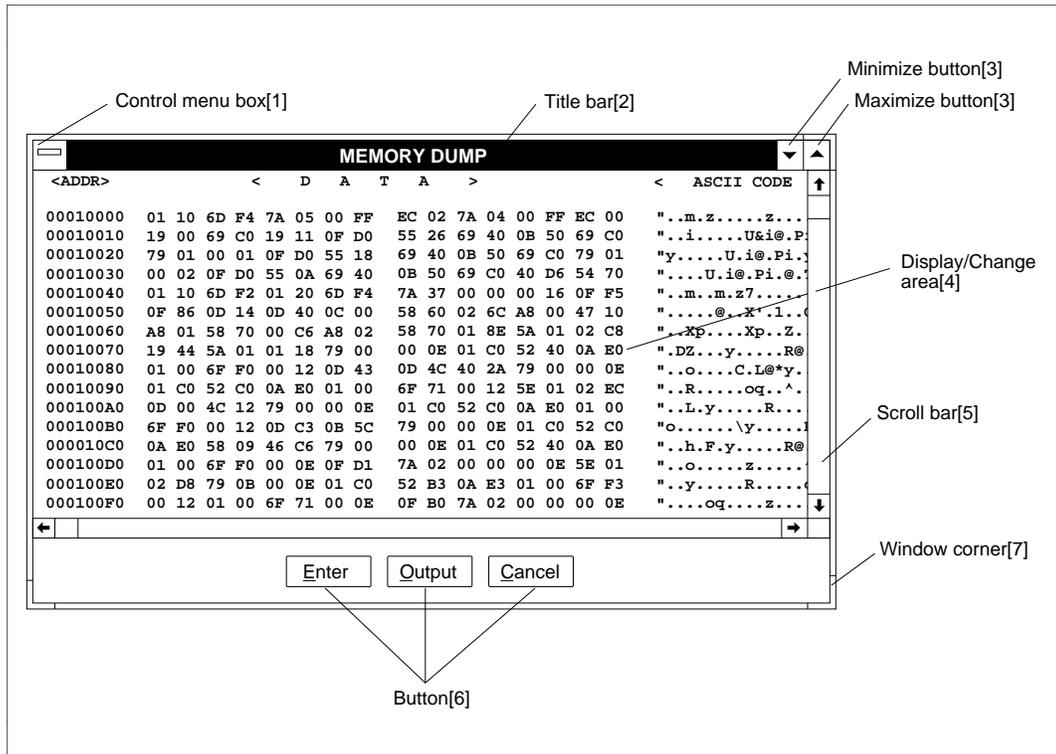


Figure 3-9 Debug Window Example

(1) Control menu box

(2) Title bar

The title of the debug window is displayed. Clicking the title bar activates the window. Once activated, the window can receive key entries and commands.

(3) Minimize button and maximize button

(4) Display/Change area

Displays debugging information. The contents of some windows can be directly changed.

(5) Scroll bar

Scrolls the display/change area.

(6) Button

Include functions such as outputting window contents to a file or closing the window.

(7) Window corner

Dragging a corner changes the window size.

Note: Tool bar buttons will be obstructed from view when the window size is reduced.

3.3 Dialog Box

A dialog box is a window used for inputting parameters necessary for command execution. Figure 3-10 shows a dialog box.

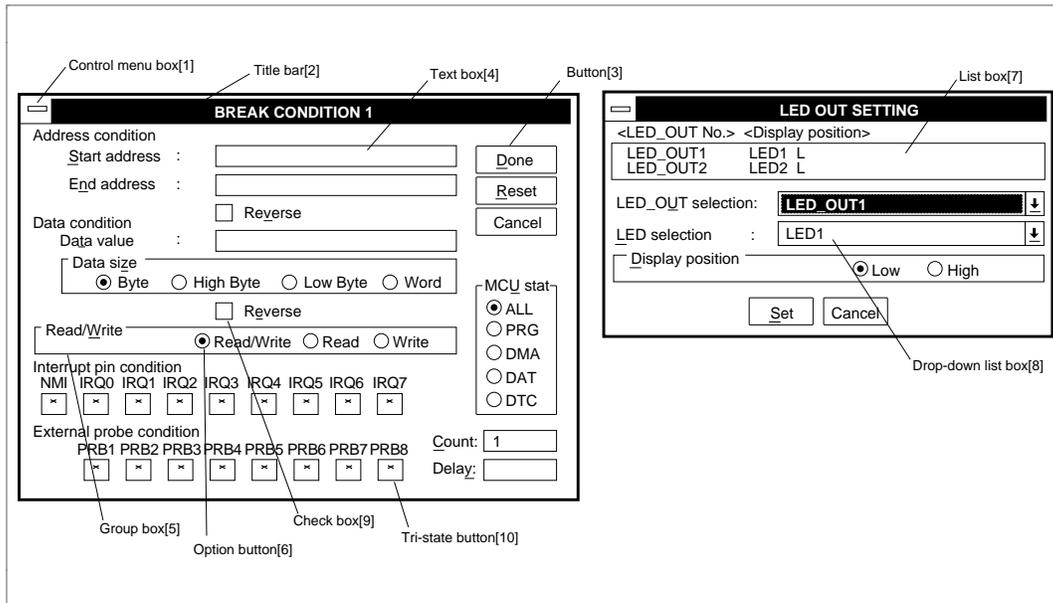


Figure 3-10 Dialog Box

(1) Control menu box

(2) Title bar

The title of the dialog box is displayed. Clicking the title bar activates the dialog box. Once activated, the window can receive key entries and commands.

(3) Button

The buttons used in the dialog box and their functions are listed in table 3-4.

3-12 HITACHI

Table 3-4 Buttons in Dialog Box

Button	Function
<Done>	Activates the input parameters and executes the command
<Cancel>	Cancels the input parameters and closes the dialog box
<Reset>	Resets the parameters to the default values
<Command>	Executes the <Command> command using the input parameters
<Button...>	Opens another dialog box to input more detailed parameters
<Close>	Closes the dialog box

(4) Text box

Receives text from the keyboard such as addresses or symbols. Click the box with the mouse to display the cursor (|), and then enter text from the keyboard.

(5) Group box

Includes items related to each other.

(6) Option button

Exclusively selects one input parameter. Only one button can be selected.

(7) List box

Lists parameters from which required ones can be selected. When the box cannot display all parameters, a scroll bar is displayed. The list can be scrolled using the mouse or keyboard.

(8) Drop-down list box

A list box with an arrow is called a drop-down list box. Clicking the arrow displays the list box.

(9) Check box

Options which can be turned on and off. When selected, \times is displayed in the box. When not selected, nothing is displayed in the box.

(10) Tri-state button

The H8S/2600 control signal, external interrupt signal, and external probe signal levels can be selected from H (high), L (low), and * (don't care). Clicking the button changes the indication on the button and the setting.

3.4 Message Box

The message box is displayed when command operation needs to be confirmed or if an error occurs. The following two types of message boxes will be displayed:

- Operation confirmation message box

The message box shown in figure 3-11 will be displayed when GUI-2600 operations must be selected at GUI-2600 initiation and termination and during command execution. Click the <Yes> or <No> button with the left button of the mouse. The <Yes> button can also be selected by pressing the Y key or (Enter) key, and the <No> button by the N key or (Esc) key.

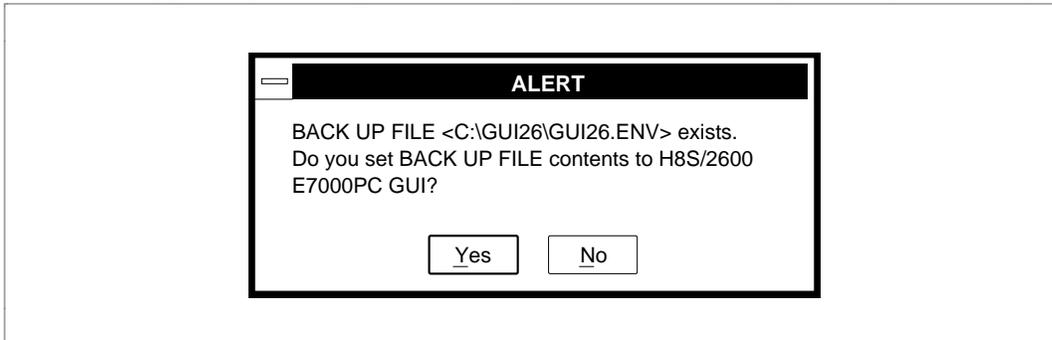


Figure 3-11 Operation Confirmation Message Box

- Error message box

If an error occurs, the error message box shown in figure 3-12 will be displayed. After checking the error contents, click the <OK> button or press the (Enter) key to continue.

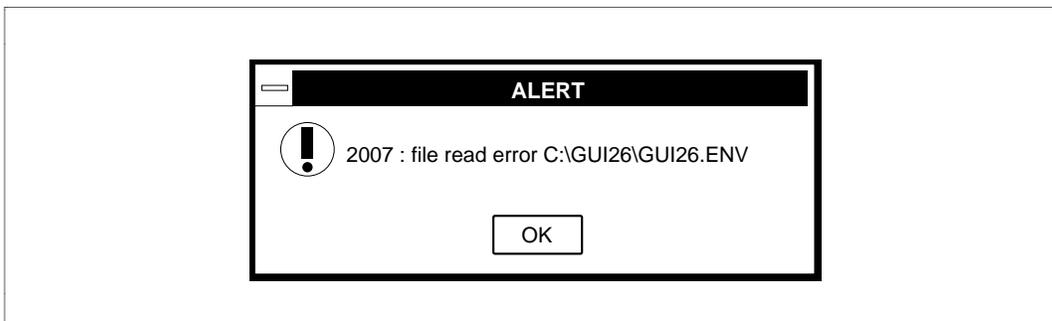


Figure 3-12 Error Message Box

3.5 Edit Function

When a debug window is active, clicking the right button of the mouse displays the edit menu shown in figure 3-13, which enables character strings to be searched and copied to the clipboard.

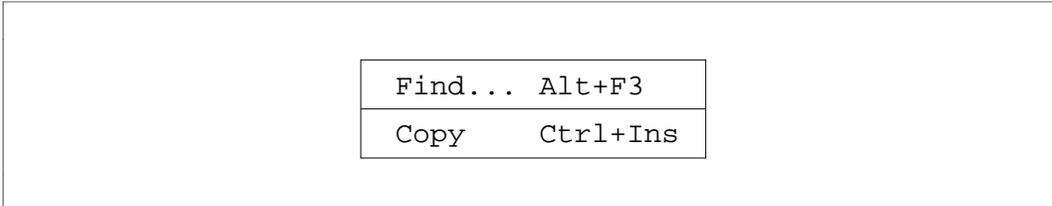


Figure 3-13 Edit Menu

3.5.1 Character String Search

Selecting the [Find...] command from the edit menu opens the dialog box for character string search (hereafter called the FIND dialog box) shown in figure 3-14.

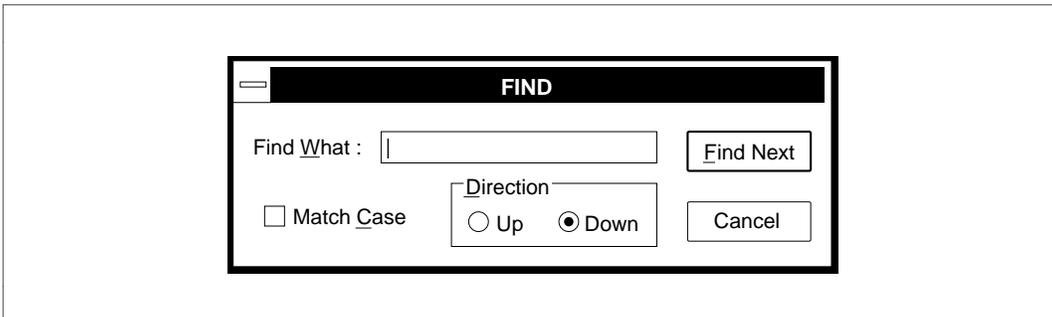


Figure 3-14 FIND Dialog Box

Enter in the [Find What :] text box the character string to be searched for, and click the <Find Next> button; the character string begins to be searched for from the current cursor position. When the character string is found, the found character string is highlighted in the window. To find the next occurrence of the character string, click the <Find Next> button. The search direction can be specified with the [Direction] group box. Selecting <Up> searches text from the cursor position to the top of the text, and <Down> searches text to the end of the text.

For the source area, only the source file currently displayed in the area can be searched.

To terminate the search operation and close the FIND dialog box, click the <Cancel> button.

When a character string is selected in the active window, the character string automatically enters the [Find What :] text box by opening the FIND dialog box. To select a word in the window, double-click the word. To select a character string, drag the mouse to highlight the character string.

Click the <Match Case> check box to search for characters while distinguishing between upper-case and lower-case letters.

3.5.2 Copy to Clipboard

Selecting the [Copy] command from the edit menu copies the selected character string to the clipboard. To select a character string, drag or double-click with the left button of the mouse.

3.6 Mouse

Almost all operations on the GUI-2600 can be performed by the mouse shown in figure 3-15. Values such as addresses, however, must be input from the keyboard.

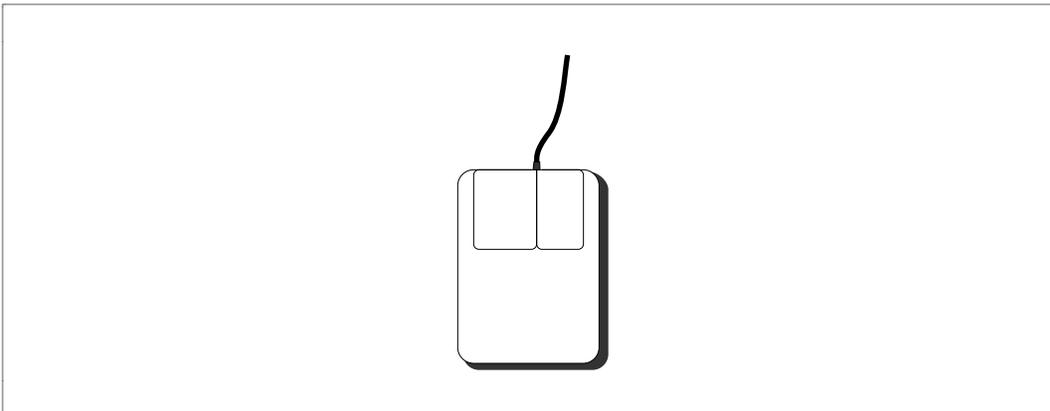


Figure 3-15 Mouse

The left button of the mouse is usually used, but for edit function, the right button is used. For text search, refer to section 3.5.1, Character String Search.

The mouse location is shown as a mouse pointer on the display. The mouse pointer can be moved to another point as needed by moving the mouse. The mouse pointer is an arrow (↑) or an I beam (|) depending on the window.

Basic mouse operations are as follows:

- Click
Press the left button and quickly release it. Click is used to select a window, menu, list, and button.
- Double click
Press the left button twice quickly. Double click is used to activate the item selected in a list box and to select a word in a window.
- Drag
Press and hold the button while moving the mouse. Drag is used to select a character string in a window.

3.7 Key Input

Almost all operations on the GUI-2600 can be performed from the keyboard. To operate using the keyboard, move to the window to be operated on, and then move to the target operation (subwindow or button).

Operate the GUI-2600 with the keyboard as follows:

- (1) Activate the window to be operated on. To activate a window, first enter (Alt + Space) from the keyboard to open the control menu. Then, select the [Switch To...] command (switches applications) using the arrow keys and move to the desired window.

The window can also be switched using the (Ctrl + Esc) keys, (Alt + Tab) keys, or (Alt + Esc) keys.

- (2) When the base window is active, use the following procedure:

Using the (Alt) key in combination with certain key(s), or short-cut keys will open the command menus. The tool bar button can be operated by the short-cut key. The tool bar function included in the Execution and the View menus can also be used as a menu command.

When the base window is active, the cursor can be displayed in the command area or the source area. This cursor can be switched between the command area and the source area by the (Ctrl + Tab) keys.

The (Ctrl + C) key input in the command area stops emulator command execution. The (Ctrl + C) key input outside the command area is invalid. Use the <STOP> button instead.

(3) When a debug window is active, use the following procedure:

Press the (Tab) or (Shift + Tab) keys to move the object of operation among windows and buttons. Wherever the object of operation is located, the (Enter) key performs the same operation as the <Done> button, and the (Esc) key performs the same operation as the <Cancel> button. Pressing the key corresponding to the underlined character of the displayed button name in combination with the (Alt) key moves the object of operation to the button.

(4) When a dialog box is active, use the following procedure:

Press the (Tab) or (Shift + Tab) keys to move the object of operation among subwindows and buttons. Wherever the object of operation is located, the (Enter) key performs the same operation as the <Done> button, and the (Esc) key performs the same operation as the <Cancel> button. Pressing the key corresponding to the underlined character of the displayed button name in combination with the (Alt) key moves the object of operation to the button.

3.8 Input Format

This section describes the input format for windows other than the command area. For input to the command area, refer to the emulator user's manual.

Expressions and file names can be input.

3.8.1 Expressions

Expressions can include numerical constants, symbols, and operators. Operators are + (addition) and - (subtraction). Expressions must not start with an operator.

- Numerical constants

Numerical constants must be in the following format.

S'nnn...nn

S: Radix

B: Binary

Q: Octal

D: Decimal

H: Hexadecimal

When no radix is specified, the value is assumed to be in hexadecimal.

n: Number

Binary: Numerical characters 0 and 1

Octal: Numerical characters 0 to 7

Decimal: Numerical characters 0 to 9

Hexadecimal: Numerical characters 0 to 9 and alphabetical characters A to F and a to f

- Symbols

Symbols in assembler or C source programs can be specified in place of addresses. Symbols are classified into two types: ordinary symbols including label names, variable names, and function names, and line number symbols indicating the line number of the assembler or C source program.

— Ordinary symbol specification

! <unit name>/<symbol name>[/<symbol name>...]

— Line number symbol specification

&<unit name>/<line number>

<unit name> : Name of the object file output by the assembler or C compiler.
The unit name is generally the source file name.

<line number>: Decimal number without radix

<symbol name>: The following items included in the unit:

Variable names (array variable name when an array-type symbol is used)

Function name

Pointer name

Structure name [. member name]

Union name [. member name]

— When specifying an external symbol for <symbol name>, specify the unit name whose external symbol definition has been declared for <unit name>. If the symbol names differ in the external definition and in the external reference, specify the name in the external definition.

— When symbols are nested, specify symbol names after a slash (/).

— To specify a local variable in a function, specify <function name> then /<symbol name>.

— Variable names, structure names, union names, and pointer names can be specified as member names.

— To specify an array-type symbol, specify an array variable name. The address value corresponding to the symbol points to the beginning of the array.

— Symbol names must be expressed by the characters _ and \$, the alphabetical characters A to Z and a to z, and numerical characters 0 to 9.

— Symbol names must start with either _ or \$ or one of the alphabetical characters A to Z and a to z.

— Line numbers must be decimal numbers starting with any number from 1 to 9. They must not start with 0.

— Uppercase and lowercase letters are distinguished.

3.8.2 File Names

File names can be specified up to 127 bytes and must be in the following format:

<drive number>:\<directory path>\<basic file name>. <file extension>

<drive number>: Drive number starting with A

<directory path>: Name of the path to the directory where the file is stored or to be stored

<basic file name>: File name expressed by one to eight characters

<file extension>: File type

Example: .C (C source file)
.SRC (assembly language source file)
.ABS (absolute load module file)
.DBG (debugging information file)

When a file name is input from a dialog box, it can be specified with a relative path name, as shown in figure 3-16, by selecting, from the list box, the current directory from the drive name to the directory path. When a file is to be read, the file name can also be selected from the list box. Wild card characters can be used for file names.

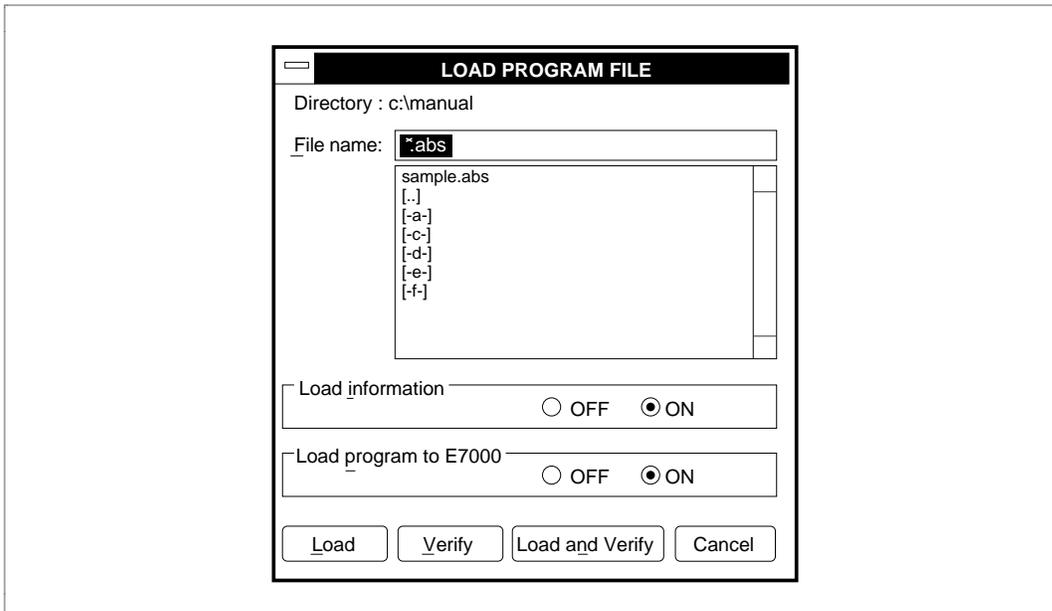


Figure 3-16 Example of a Dialog Box for File Operation

3.9 Short-Cut Keys

The buttons of the commands or on the base window can be operated by short-cut keys. Table 3-5 lists the short-cut keys.

Table 3-5 Short-Cut Keys

Classification	Button/Command Names	Short-Cut Keys
Buttons	<DISPLAY>	(Alt + N)
	<STEP_OVER>	(F7)
	<STEP_UP>	(F8)
	<STEP>	(F9)
	<CONTINUE>	(Alt + G)
	<SET>	(Alt + B)
	<CLEAR>	(Alt + C)
	<STOP>	(Ctrl + C)
FILE menu	Load - Load program file...	(Alt + L)
	Load - Load other file...	(Alt + K)
	Load - E7000 load...	(Alt + J)
	Input from...	(Alt + I)
	Output to...	(Alt + O)
	Quit	(Alt + F4)
EXECUTION menu	Execution mode...	(Ctrl + E)
	Go...	(Alt + Ctrl + G)
	Break...	(Alt + Ctrl + B)
	Break condition1...	(Ctrl + 1)
	Break condition2...	(Ctrl + 2)
	Break condition3...	(Ctrl + 3)
	Break condition4...	(Ctrl + 4)
	Break condition5...	(Ctrl + 5)
Break condition6...	(Ctrl + 6)	

Table 3-5 Short-Cut Keys (cont)

Classification	Button/Command Names	Short-Cut Keys
TRACE menu	Trace mode...	(Alt + Ctrl + T)
	Trace condition...	(Ctrl + T)
	Trace display - Setting...	(Alt + Ctrl + X)
	Trace display - Text...	(Alt + X)
	Trace display - Graph...	(Alt + P)
VIEW menu	Memory - Setting...	(Alt + Ctrl + M)
	Memory - Dump	(Alt + M)
	Watch - Setting...	(Alt + Ctrl + W)
	Watch - Display	(Alt + W)
	Register	(Alt + R)
	Symbol - Display	(Alt + S)
	Symbol - Value	(Alt + Ctrl + N)
	Source - Setting...	(Alt + Ctrl + A)
	Source - Display...	(Alt + A)
	Disassemble	(Alt + D)
	Route	(Alt + U)
	LED setting...	(Alt + Ctrl + Y)
	LED out setting...	(Alt + Y)
	Command area	(Ctrl + F6)
HELP menu	GUI operating help	(F1)
	EMULATOR command display	(Alt + F1)
	About...	None

Section 4 GUI-2600 Operation

This section describes GUI-2600 initiation, basic operations, and termination, assuming that the GUI-2600 has been installed. If it has not been installed yet, do so according to the procedure given in section 2, Installation.

This section mainly describes operations using the mouse. For operations using only the keyboard, refer to section 5, Command Reference.

4.1 Emulator Power-On

Power on the emulator station before initiating the GUI-2600.

4.2 Initiating GUI-2600

After Windows has been initiated, the Program Manager window shown in figure 4-1 is displayed. The group window of the GUI-2600 is in the Program Manager window. (When the group window is not open, place the mouse pointer on the GUI-2600 group icon and double-click it.)

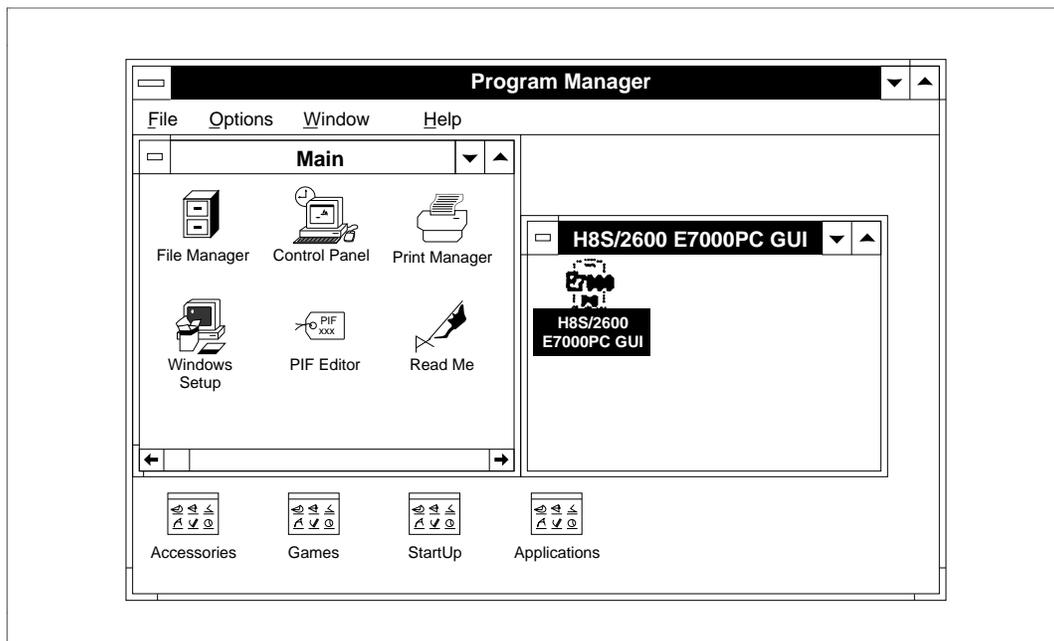


Figure 4-1 Display at Windows Initiation

The application icon named H8S/2600 E7000PC GUI is in the GUI-2600 group window, which is used to initiate the GUI-2600. The GUI-2600 can be initiated using the mouse or the keyboard.

- Using the mouse
 - (1) Place the mouse pointer on the GUI-2600 application icon.
 - (2) Double-click the left button of the mouse.
 - (3) The GUI-2600 base window shown in figure 4-2 appears.
- Using the keyboard
 - (1) Press the (Ctrl + Tab) keys to select the GUI-2600 group window. When selected, the window title is highlighted.
 - (2) Press the (←) or (→) key to select the GUI-2600 application icon. When selected, the icon title is highlighted.
 - (3) Press the (Enter) key.
 - (4) The GUI-2600 base window shown in figure 4-2 appears.

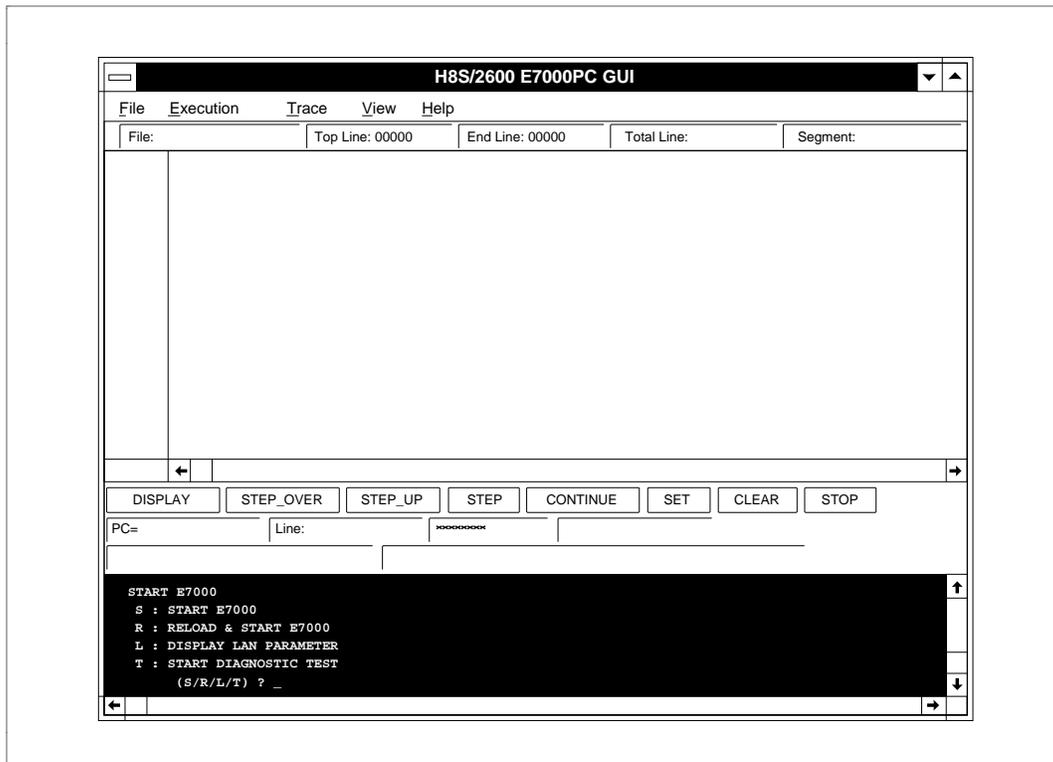


Figure 4-2 Display at GUI-2600 Initiation

4.3 Initiating Emulator

After the GUI-2600 has been initiated, the emulator monitor message shown in figure 4-3 is displayed in the command area, and the GUI-2600 waits for an emulator monitor command input.

```
H8S/2600 E7000PC GUI Vx.x
Copyright (c) Hitachi, Ltd. 1995
Licensed Material of Hitachi, Ltd.

E7000 MONITOR Vx.x
Copyright(C) 1993 Hitachi, Ltd.
Licensed Material of Hitachi, Ltd.

TESTING
  RAM 0123

START E7000
  S:  START E7000
  R:  RELOAD & START E7000
  L:  DISPLAY LAN PARAMETER
  T:  START DIAGNOSTIC TEST
      (S/R/L/T)?_
```

Figure 4-3 Emulator Monitor Message

Enter the (S) key followed by the (Enter) key to display the emulator message shown in figure 4-4 and initiate the emulator.

```
** E7000 SYSTEM LOADING **

H8S/xxxx E7000 (HSxxxxEPDxxSF) Vx.x
Copyright (C) Hitachi, Ltd. 199x
Licensed Material of Hitachi, Ltd.

CONFIGURATION FILE LOADING
HARD WARE REGISTER READ/WRITE CHECK
POD SYSTEM LOADING
EMULATOR POD TEST
** RESET IN BY E7000 !
CLOCK = 8 MHz
MCU NAME=H8S/xxxx OPERATION MODE=7(MD2-0=7) MODE SET=E7000
PIN MODE=128 INTERNAL ROM SIZE=64KB INTERNAL RAM SIZE=4KB
ADC MODE=HI SPEED DMAC MODE=ENABLE REFRESH MODE=ENABLE
SCI MODE=MULTI SCI CHANEL=3CH 16BIT TIMER MODE=TPU0,1,2,3,4,5
REMAINS EMULATION MEMORY S=80000/E=000000
GUARDED AREA = 00000000-00FFFFFF WRITE PROTECT AREA = -

WARM OR COLD START
file name : WARM START
return : COLD START
(file name/return) ?
:_
```

Figure 4-4 Emulator System Message

When a backup file exists, the confirmation message box for backup file load shown in figure 4-5 is displayed. Click the <Yes> button to read the backup file and restore the previous setting information in the GUI-2600. To initiate the GUI-2600 using default settings, click the <No> button. When no backup file exists, this message box is not displayed and the GUI-2600 is initiated using default settings.

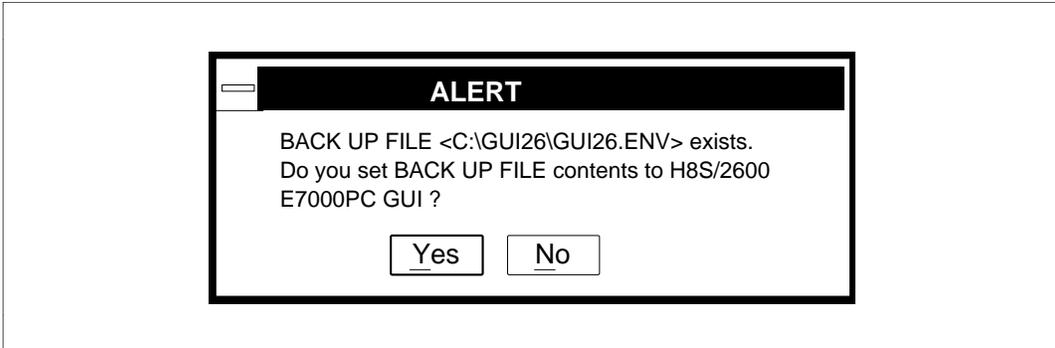


Figure 4-5 Confirmation Message Box for Backup File Load

4.4 Loading Load Module

4.4.1 Allocating Emulation Memory

When the user system does not have memory to load programs, the emulation memory must be allocated with the MAP command of the emulator. The MAP command is entered in the command area. An example of emulation memory allocation with the MAP command is shown in figure 4-6.

- (1) Place the mouse pointer in the command area and click to display the cursor (_).
- (2) If the emulator prompt (:) is not displayed, press the (Enter) key to display it.
- (3) Enter the MAP command.

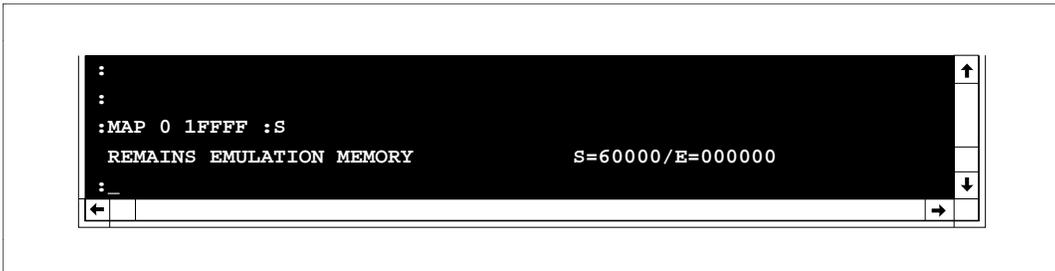


Figure 4-6 Example of Emulation Memory Allocation with MAP Command

4.4.2 Loading Program

Load programs using the following procedure:

(1) Open LOAD PROGRAM FILE dialog box

Select the [Load - Load program file...] command from the File menu. The LOAD PROGRAM FILE dialog box shown in figure 4-7 is displayed.

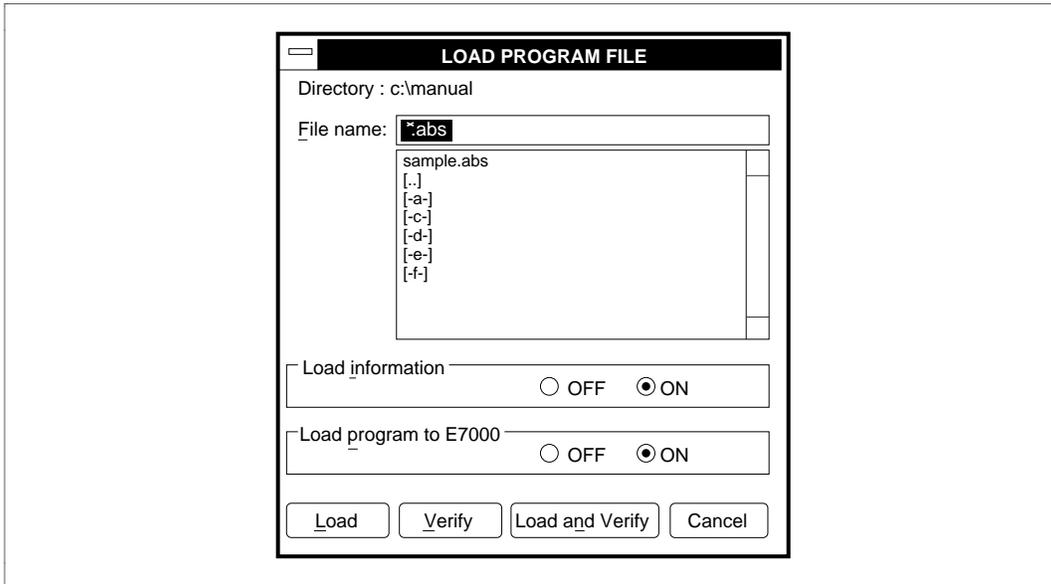


Figure 4-7 LOAD PROGRAM FILE Dialog Box

(2) Set current directory

Select from the list box the current directory where the program is to be loaded. The drive name and directory name must be specified in that order.

Place the mouse pointer on the target drive name or the target directory name in the list box and double-click to display the selected current directory in the directory display box.

(3) Set file name

After the current directory has been selected, file names in the current directory having a file extension of .ABS are displayed in the list box. Place the mouse pointer on the file to be loaded and click to select it.

(4) Start load operation

Click the <Load> button to start loading. The message box shown in figure 4-8 is displayed during program load.

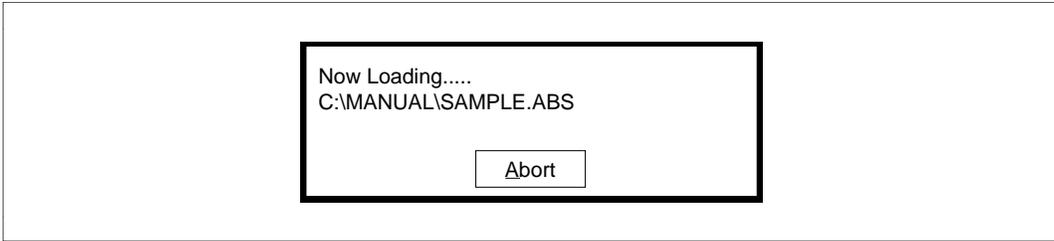


Figure 4-8 Loading Message Box

(5) Load completed

When the load operation is completed, the message box closes, and the section of the source file including the program start address is displayed in the source area. The line corresponding to the program start address is marked with the PC mark (PC>) indicating the location pointed to by the program counter.

Start and end addresses in the load memory are displayed in the command area.

4.5 Displaying Source File

Source files can be displayed in the source area with the [Source - Display...] command in the View menu.

- (1) Select the [Source - Display...] command from the View menu; the SOURCE DISPLAY dialog box shown in figure 4-9 appears.
- (2) Select from [File name :] the file name to be displayed in the source area by placing the mouse pointer on the file name and clicking it.
- (3) Click the <Display> button to display the source file in the source area.

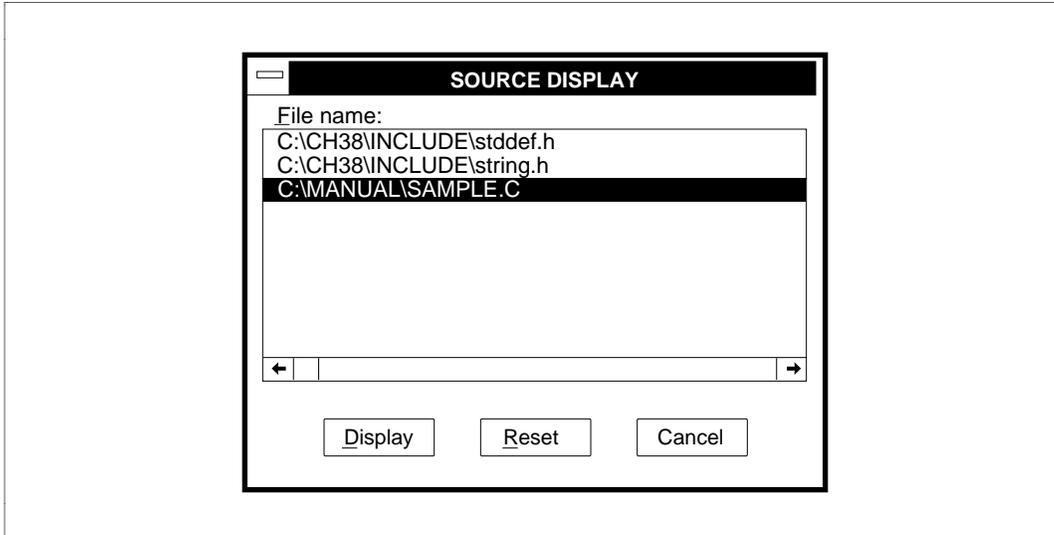


Figure 4-9 SOURCE DISPLAY Dialog Box

In addition, the font, font size, and tab width of the source file displayed in the source area can be specified using the [Source - Setting...] command, as shown in figure 4-10.

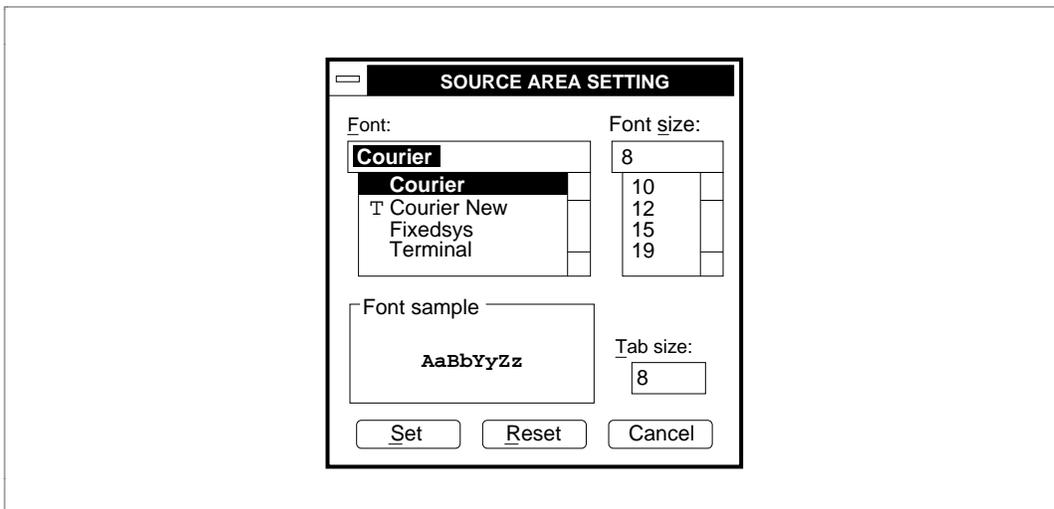


Figure 4-10 SOURCE AREA SETTING Dialog Box

4.6 Setting and Cancelling Breakpoints

4.6.1 Setting Breakpoints

Breakpoints can be set within the source area. They can also be set with the [Break...] command in the Execution menu. An example of breakpoint setting is shown in figure 4-11.

- (1) Scroll the source area to display the line where a breakpoint is to be set, and click the line with the mouse pointer. The cursor (|) will appear on the clicked line. Note that breakpoints can be set only on lines marked with B.
- (2) Click the <SET> button on the tool bar.
- (3) After a breakpoint is set for a selected line, the BP mark appears on the line.

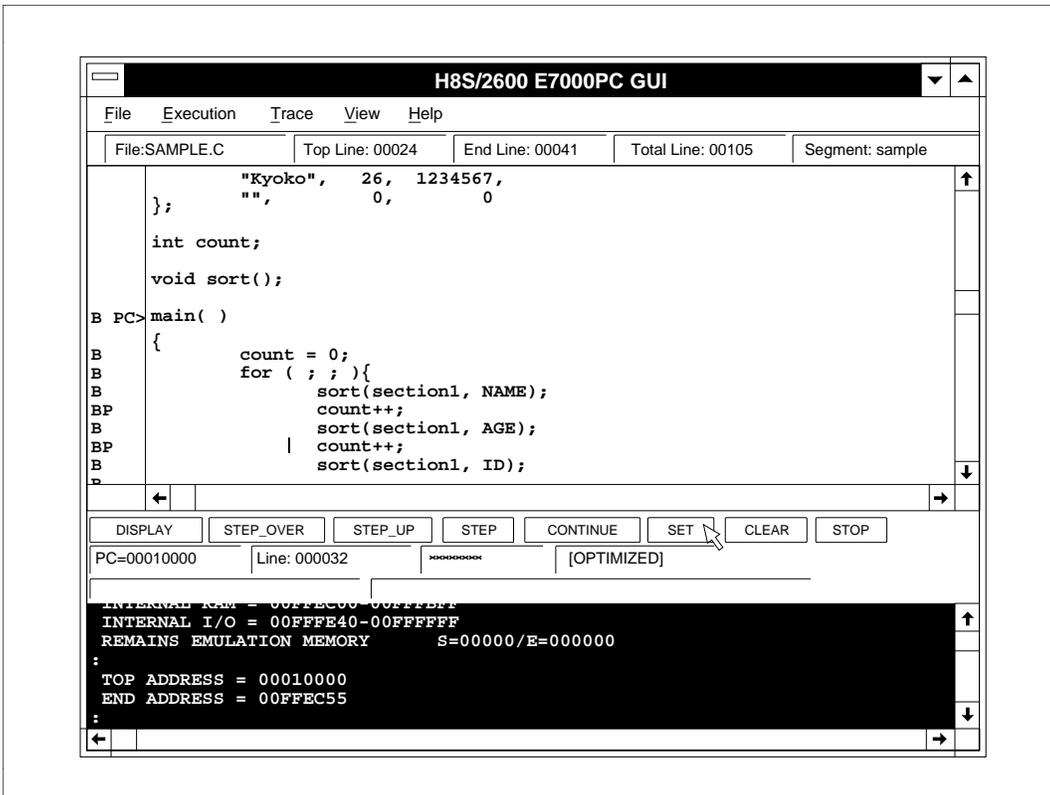


Figure 4-11 Example of Setting Breakpoints

4.6.2 Cancelling Breakpoints

Breakpoints can be cancelled within the source area. They can also be cancelled with the [Break...] command in the Execution menu. An example of breakpoint cancellation is shown in figure 4-12.

- (1) Scroll the source area to display the line for which the breakpoint is to be cancelled, and click the line with the mouse pointer. The cursor (|) appears on the clicked line. Note that breakpoints can be cancelled only on lines marked with BP.
- (2) Click the <CLEAR> button on the tool bar.
- (3) After the breakpoint on the line is cancelled, the BP mark on the line returns to a B mark.

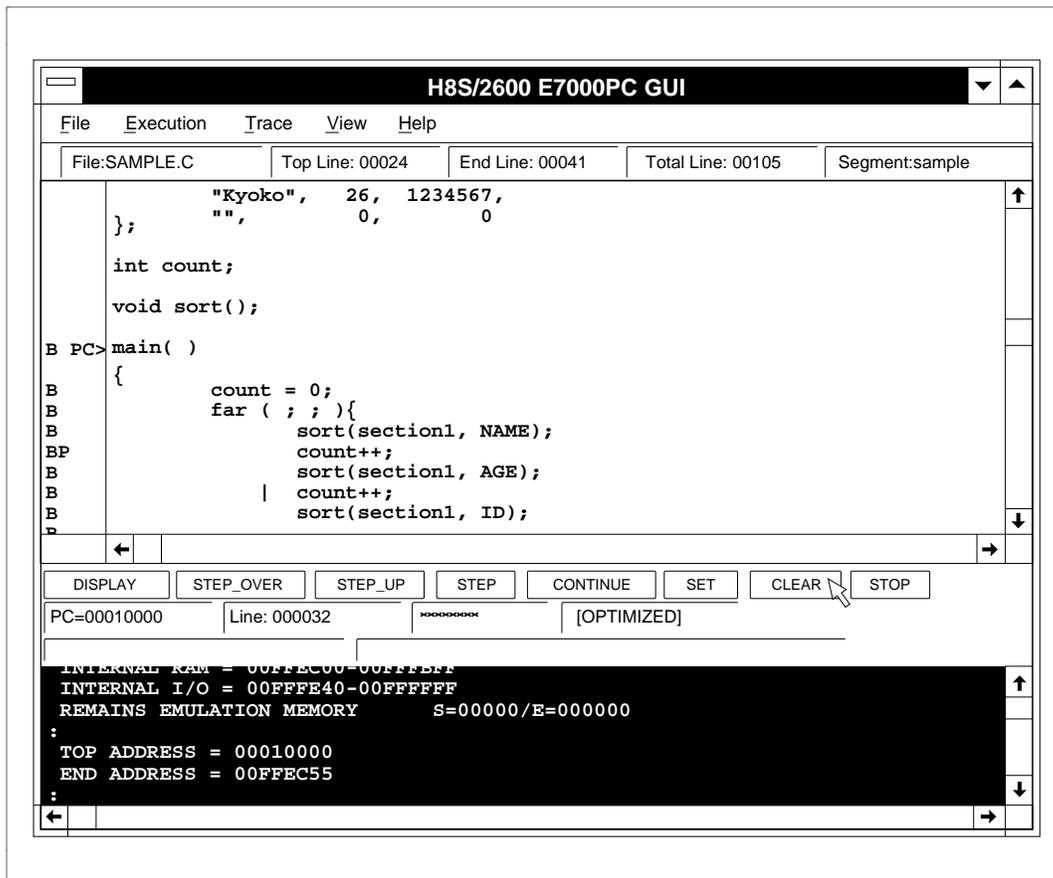


Figure 4-12 Example of Breakpoint Cancellation

4.7 Setting Execution Start Address and Stack Pointer

Before executing the user program, the program counter (PC) and stack pointer (SP) must be set. The PC and SP can be set with the [Register] command in the View menu. They can also be set or modified by entering an emulator command in the command area. An example of register modification is shown in figure 4-13.

- (1) Select the [Register] command in the View menu. The register window shown in figure 4-13 will appear.
- (2) Place the mouse pointer over the hexadecimal data for the PC and click to display the character-insertion cursor (|).
- (3) Enter data from the keyboard.
- (4) Enter data for the SP (ER7) in the same way if necessary.
- (5) Click the <Enter> button to set the modified values in the registers.

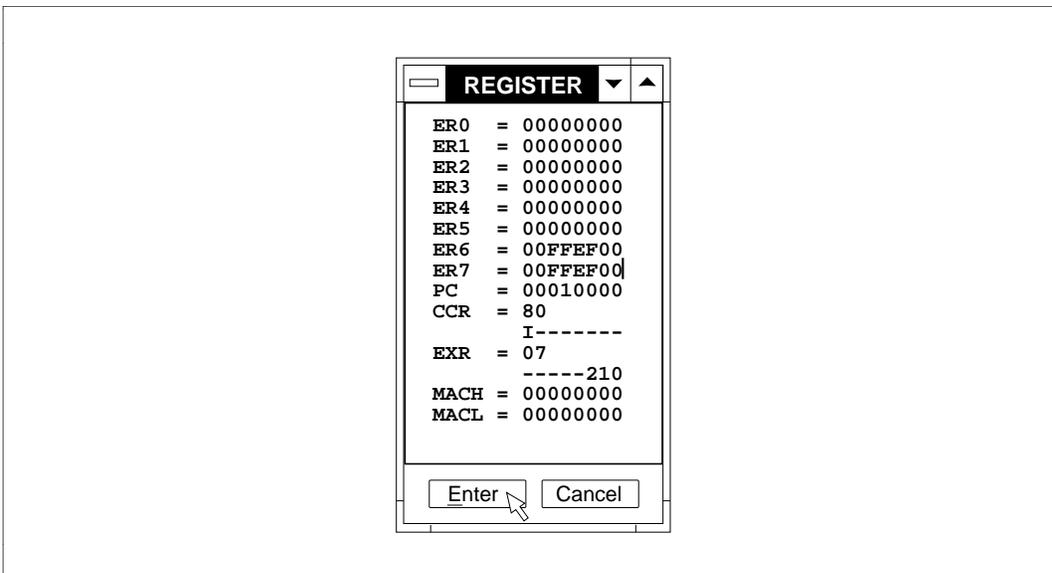


Figure 4-13 Example of Setting Registers

4.8 Executing Programs

Program execution can be started with several procedures. This section describes the procedure using the tool bar and that using the [Go...] command in the Execution menu.

4.8.1 Executing Programs Using Tool Bar

The buttons for program execution on the tool bar are listed in table 4-1.

Table 4-1 Program Execution Buttons in Tool Bar

Button	Short-Cut Keys	Function
STEP_OVER	(F7)	Executes one program line pointed to by the current program counter (PC) and then stops. When the line includes a function (subroutine) call, program execution stops after executing the function. Note that this button cannot be used to execute programs in the ROM area.
STEP_UP	(F8)	Executes the function including the current program counter (PC), and stops execution when returning to the upper function (calling function). This button cannot be used to execute programs in the ROM area.
STEP	(F9)	Executes one program line pointed to by the current program counter (PC) and then stops. When the line includes a function (subroutine) call, program execution stops at the first line of the function.
CONTINUE	(Alt + G)	Executes the program from the line pointed to by the current program counter (PC). Execution continues until a line including the cursor in the source area is reached. Execution stops when a line marked with B in the source area is clicked.

An example of program execution using the <CONTINUE> button is shown in figure 4-14. When program execution stops at the cursor location in the source area, the emulation information area displays the execution stop address, execution stop line number, cause of termination, and execution time.

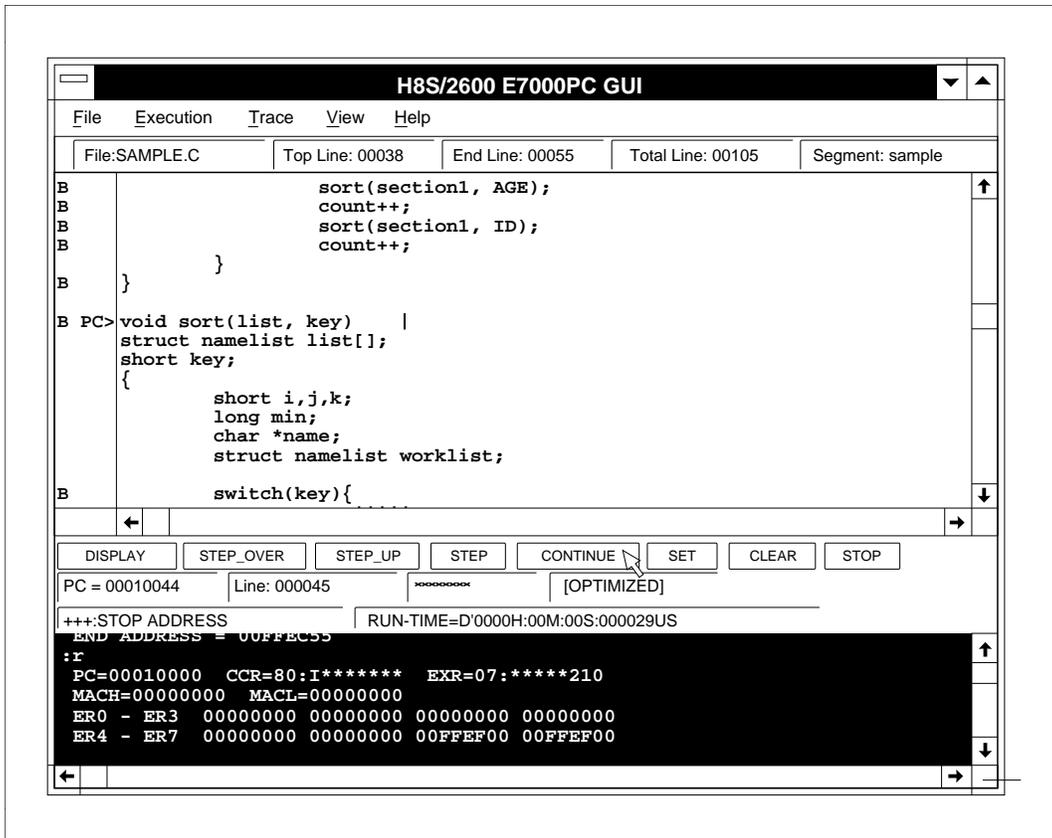


Figure 4-14 Program Execution Example Using <CONTINUE> Button

4.8.2 Executing Programs Using [Go...] Command

Program execution can be started from a specified address with the [Go...] command in the Execution menu. When the [Go...] command is executed, the GO dialog box shown in figure 4-15 opens.

- By selecting <RESET> in the [Start mode] group box and clicking the <Done> button, the emulator outputs a RESET signal to the user system and program execution starts from the reset vector.
- By selecting <Order> in the [Start mode] group box, specifying an address in the [Start address:] text box, and then clicking the <Done> button, program execution starts from the specified address.
- By selecting <PC> in the [Start mode] group box and then clicking the <Done> button, program execution starts from the current PC (program counter) value.

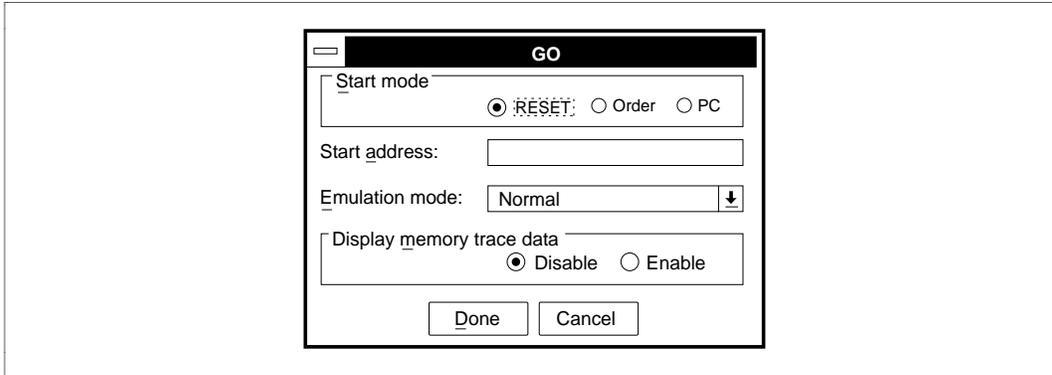


Figure 4-15 GO Dialog Box

4.8.3 Terminating Programs

To forcibly terminate program execution, click the <STOP> button on the tool bar, as shown in figure 4-16. After program execution stops, the emulation information area displays the execution stop address, execution stop line number, cause of termination, and execution time. The section of the source file corresponding to the stop location is displayed in the source area.

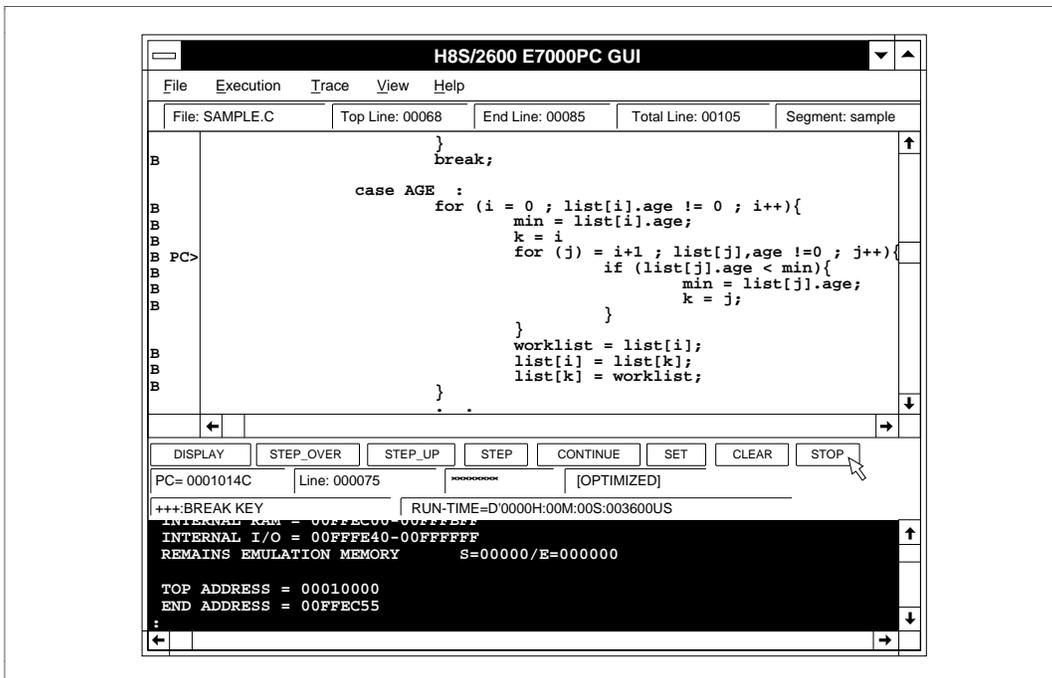


Figure 4-16 Example of Forcible Program Execution Termination Using <STOP> Button

4.9 Displaying Symbol Contents

Symbol contents can be displayed by selecting symbols in the source area using the procedure shown in figure 4-17.

(1) Select symbol

Select a symbol in the source area by placing the mouse pointer on the symbol and double-clicking it. When selected, the symbol is highlighted.

(2) Display symbol contents

Click the <DISPLAY> button on the tool bar to open the SYMBOL VALUE window and display the contents of the selected symbol.

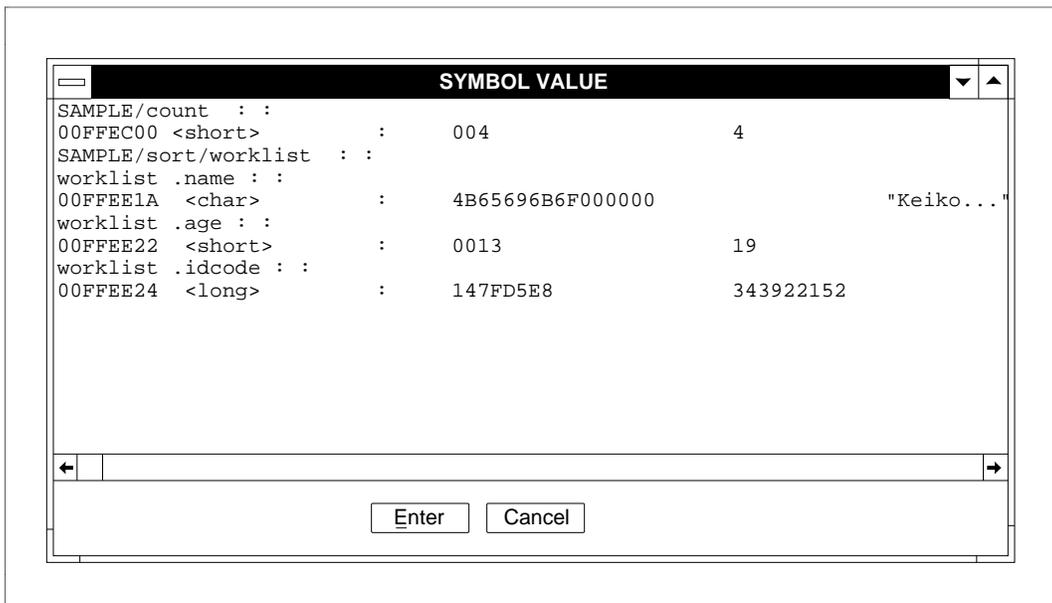


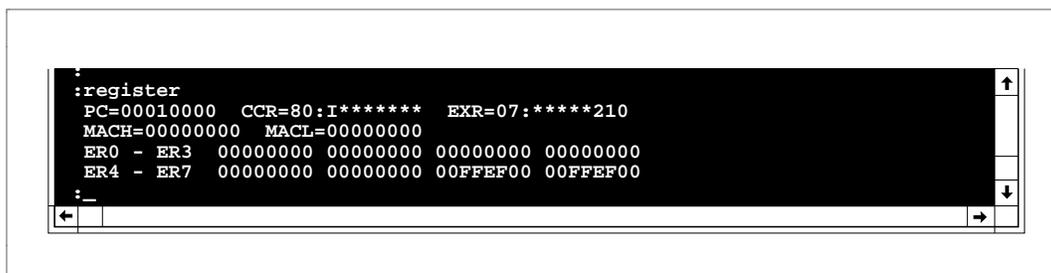
Figure 4-17 Example of Symbol Contents Display

4.10 Inputting Emulator Commands

Emulator commands can be input and executed in the command area of the base window. Some emulator commands, however, cannot be executed using the GUI-2600. For available commands, refer to appendix A, E7000 Emulator Command List.

Emulator commands can be executed using the procedure shown in figure 4-18.

- (1) Place the mouse pointer in the command area and click the left button of the mouse to display the cursor (_).
- (2) If the emulator command prompt (: or #) is not displayed, press the (Enter) key to display it.
- (3) Enter an emulator command and press the (Enter) key.
- (4) The command execution results are displayed in the command area.



```
:register
PC=00010000 CCR=80:I***** EXR=07:*****210
MACH=00000000 MACL=00000000
ER0 - ER3 00000000 00000000 00000000 00000000
ER4 - ER7 00000000 00000000 00FFEF00 00FFEF00
:_
```

Figure 4-18 Emulator Command Input Format

4.11 Displaying Help Information

The help function provides information on GUI-2600 operation and emulator commands.

4.11.1 GUI-2600 Operation Help

The GUI OPERATING HELP window shown in figure 4-19 appears by selecting the [GUI operating help] command from the Help menu. Click <Cancel> to close this window.

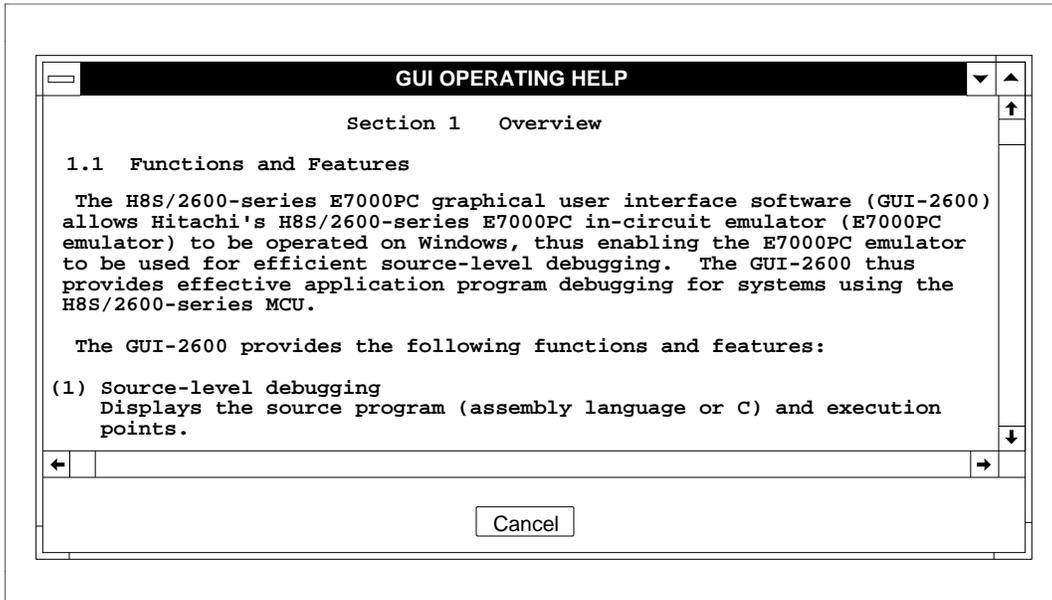


Figure 4-19 GUI-2600 Operation Help Display

4.11.2 Emulator Command Help

The EMULATOR COMMAND DISPLAY window shown in figure 4-20 appears by selecting the [EMULATOR command display] command from the Help menu. The emulator commands are listed in this window. Detailed help information on each command can be displayed using the following procedure:

- (1) Select a command by placing the mouse pointer on the command name displayed in the EMULATOR COMMAND DISPLAY window and double-clicking. When selected, the command name is highlighted.
- (2) Click the <Display> button.
- (3) The EMULATOR COMMAND HELP window opens and detailed help information on the command is displayed.
- (4) Click the <Cancel> button to close the window.

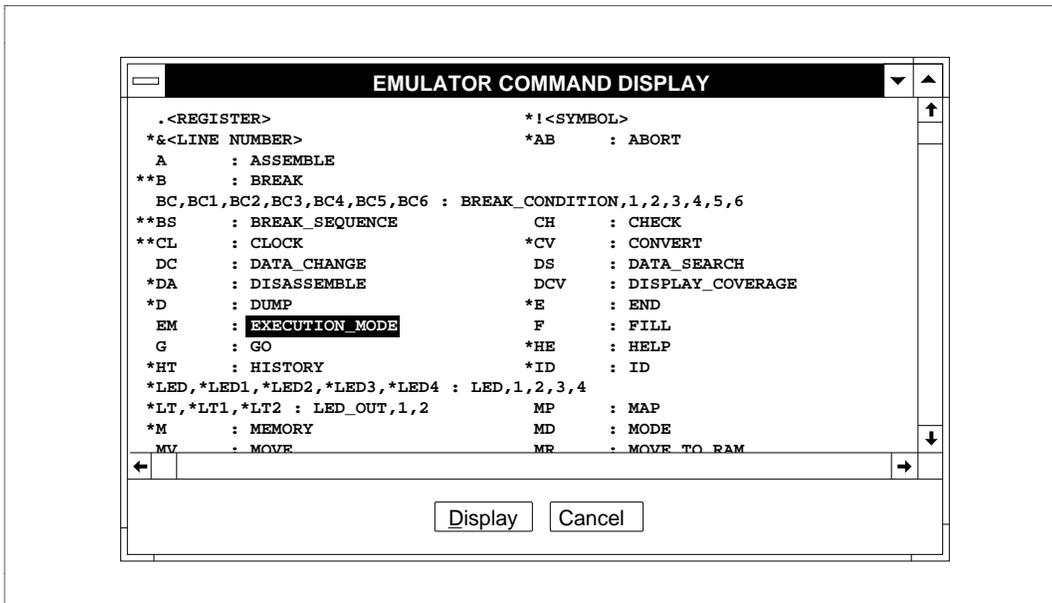


Figure 4-20 Emulator Command Help Display

4.12 Calling MS-DOS

CAUTION

Some MS-DOS commands will damage files when run on Windows.
Read Windows Manual before using MS-DOS commands.

MS-DOS can be called with the [MS-DOS Prompt] in the [Main] group window of the Program Manager when source files need to be edited, compiled, or linked during debugging. Figure 4-21 shows the procedure to call MS-DOS.

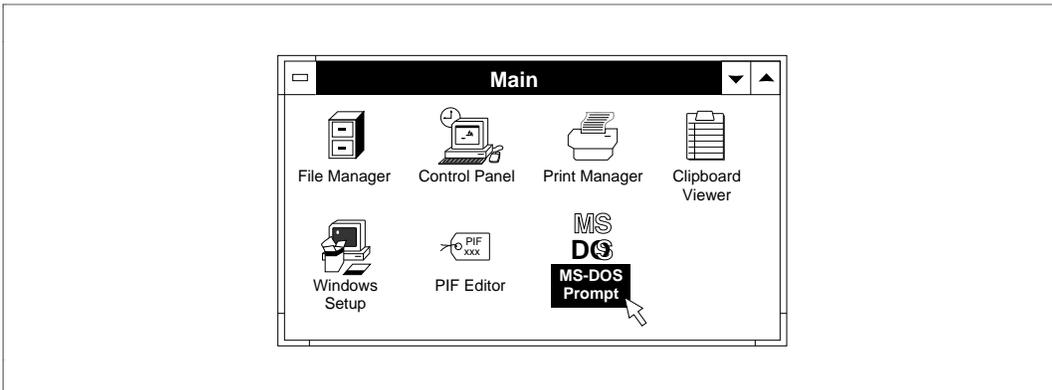


Figure 4-21 MS-DOS Call Procedure

- (1) Double-click the [MS-DOS Prompt] icon in the [Main] group window to switch the display from Windows to MS-DOS full-screen display; the MS-DOS prompt appears.
- (2) Execute an MS-DOS application. Some MS-DOS applications must not run in this state. For details, refer to the documents related to Windows.
- (3) While Windows is operating in 386 enhanced mode, almost all MS-DOS applications can run in a window using the following procedure.
 - (a) Press the (Alt + Enter) keys after executing the MS-DOS prompting program.
 - (b) The contents of the full-screen display appears in a window. In this state, another Windows application can be executed.
 - (c) Press the (Alt + Enter) keys again to switch the window to full-screen display.

(4) Enter exit and press the (Enter) key to return to Windows.

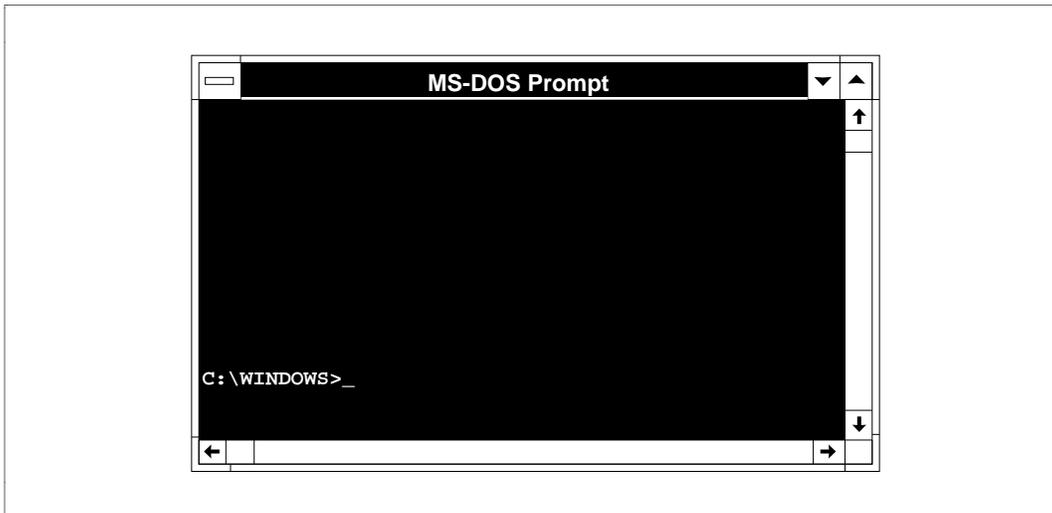


Figure 4-22 MS-DOS Application Execution in Windows

4.13 Terminating GUI-2600

The GUI-2600 is terminated with the [Quit] command in the File menu. If the program debugged is being executed, execution continues even after the GUI-2600 is terminated.

At termination, the confirmation message for storing a backup file shown in figure 4-23 appears. Click <Yes> to store current settings during program execution of the GUI-2600. The backup file can be loaded at GUI-2600 initiation to restore the settings in the GUI-2600. After <Yes> is clicked, a dialog box is displayed to enter the backup file name.

When the <No> button is clicked, the GUI-2600 terminates without saving settings.

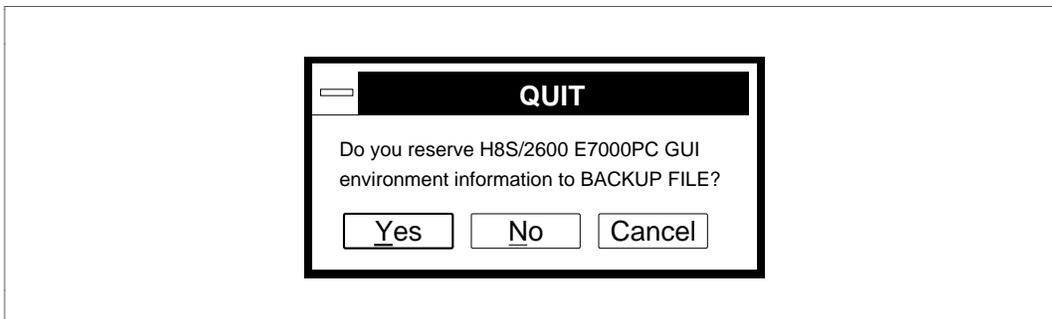


Figure 4-23 Confirmation Message for Backup File Storing

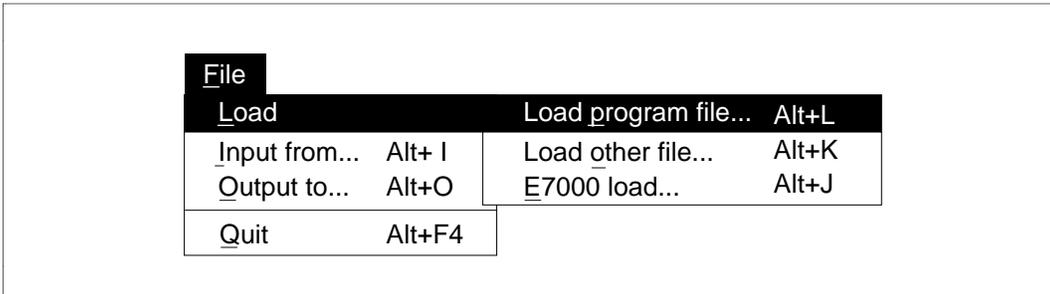
Section 5 Command Reference

This section describes the GUI-2600 menu commands in detail.

5.1 Menu Command Functions

5.1.1 Filing Functions (File Menu)

The menu for filing functions is shown in figure 5-1.



File	
<u>L</u> oad	Load program file... Alt+L
<u>I</u> nterface from...	Alt+I
<u>O</u> utput to...	Alt+O
<u>Q</u> uit	Alt+F4
	Load other file... Alt+K
	E7000 load... Alt+J

Figure 5-1 Menu for Filing Functions (File Menu)

- Loading program file ([Load - Load program file...])
Loads the load module file to be debugged into the GUI-2600 and the E7000PC emulator. Memory contents can be compared with the load module file for verification purposes.
- Loading other files ([Load - Load other file...])
Loads an S-type load module file and a Hex-type load module file into the E7000PC emulator. Memory contents can be compared with the file for verification purposes.
- Loading directly into the E7000 ([Load - E7000 load...])
Loads the load module file to be debugged only into the E7000PC emulator. Symbol information can be loaded into the E7000PC emulator in load module file units. Memory contents can be compared with the load module file for verification.
- Inputting and executing emulator commands ([Input from...])
Inputs a command file and executes emulator commands.
- Storing the contents displayed in the command area ([Output to...])
Stores the contents displayed in the command area.

- Terminating GUI-2600 ([Quit])
Terminates the GUI-2600.

5.1.2 Execution Functions (Execution Menu)

The menu for execution functions is shown in figure 5-2.

Execution	
Execution mode...	Ctrl+E
Go...	Alt+Ctrl+G
Break...	Alt+Ctrl+B
Break condition1...	Ctrl+1
Break condition2...	Ctrl+2
Break condition3...	Ctrl+3
Break condition4...	Ctrl+4
Break condition5...	Ctrl+5
Break condition6...	Ctrl+6
<STEP_OVER>	F7
<STEP_UP>	F8
<STEP>	F9
<CONTINUE>	Alt+G
<SET>	Alt+B
<CLEAR>	Alt+C
<STOP>	Ctrl+C

Figure 5-2 Menu for Execution Functions (Execution Menu)

- Setting program execution conditions ([Execution mode...])
Sets conditions under which program emulation is executed.
- Executing program ([Go...])
Starts program emulation.
- Setting and cancelling breakpoints ([Break...])
Sets and cancels breakpoints.
- Setting hardware break conditions ([Break condition1,2,3,4,5,6...])
Sets hardware break conditions. A maximum of six conditions (1 to 6) can be set.

- Execution buttons
The buttons related to program execution that are located on the tool bar (<STEP_OVER>, <STEP_UP>, <STEP>, <CONTINUE>, <SET>, <CLEAR>, and <STOP>) can be operated from the menu.

5.1.3 Trace Functions (Trace Menu)

The menu for trace functions is shown in figure 5-3.

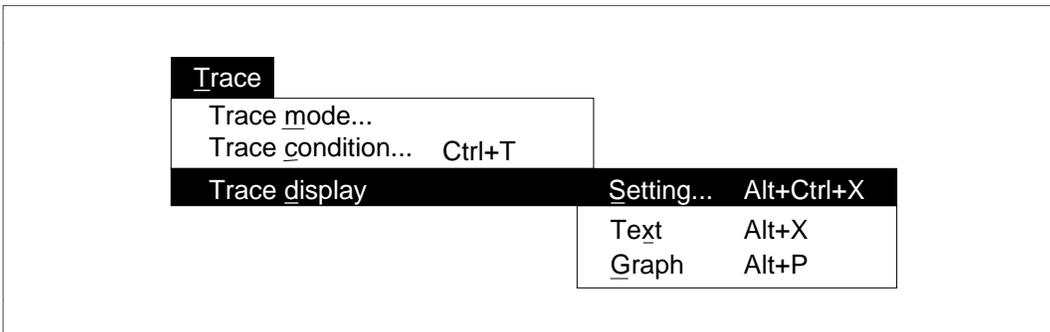


Figure 5-3 Menu for Trace Functions (Trace Menu)

- Setting memory trace conditions ([Trace mode...])
Sets conditions for acquiring trace information on memory. During program emulation, trace information is acquired according to these settings.
- Setting trace information acquisition conditions ([Trace condition...])
Sets conditions for acquiring trace information. During program emulation, trace information is acquired according to these settings.
- Setting trace information display conditions ([Trace display - Setting...])
Sets conditions for text display of trace information ([Trace display - Text]) and graphic display of trace information ([Trace display - Graph]). With this function, only necessary trace information can be displayed.
- Displaying trace information in text form ([Trace display - Text])
Displays trace information in the TRACE DISPLAY TEXT window in text form according to the conditions set with the [Trace display - Setting...] command. The display contents can be output to a file.

- Displaying trace information in graph form ([Trace display - Graph])
Displays trace information in the TRACE DISPLAY GRAPH window in graph form according to the conditions set with the [Trace display - Setting...] command. The graph can be magnified with the zooming function.

5.1.4 Debugging Information Display Functions (View Menu)

The menu for debugging information display functions is shown in figure 5-4.

View				
Memory		▶	Memory	Setting... Alt+Ctrl+M Dump Alt+M
Watch		▶	Watch	Setting... Alt+Ctrl+W Display Alt+W
Register	Alt+R		Symbol	Display Alt+S Value Alt+Ctrl+N
Symbol		▶	Source	Setting... Alt+Ctrl+A Display... Alt+A
Source		▶		
Disassemble	Alt+D			
Route	Alt+U			
LED setting...	Alt+Ctrl+Y			
LED out setting...	Alt+Y			
Command area [small]				
<DISPLAY>	Alt+N			

Figure 5-4 Menu for Debugging Information Display Functions (View Menu)

- Setting memory dump display area ([Memory - Setting...])
Sets memory area whose contents are displayed in dump format and modified with the [Memory - Dump] command.
- Displaying in dump format and modifying memory contents ([Memory - Dump])
Displays memory contents in the MEMORY DUMP window for the area set with the [Memory - Setting...] command. The memory contents can be changed in the MEMORY DUMP window. The display contents can be output to a file.
- Setting watch points ([Watch - Setting...])
Sets watch points to display memory contents during emulation.
- Displaying watch point contents ([Watch - Display])
Displays the contents of the specified watch point in the WATCH DISPLAY window during emulation.

- **Displaying and modifying register contents ([Register])**
Displays register contents in the REGISTER window. The register contents can be changed in the REGISTER window.
- **Displaying symbol information ([Symbol - Display])**
Displays in the SYMBOL DISPLAY window the information on all symbols within the symbol scope determined by the current program counter (PC) value.
- **Displaying and changing symbol contents ([Symbol - Value])**
Displays in the SYMBOL VALUE window the contents of the symbols specified with the [Symbol - Display] command and with the <DISPLAY> button on the tool bar. The symbol contents can be changed in the SYMBOL VALUE window.
- **Setting source area ([Source - Setting...])**
Sets the font, font size, and tab width of the source program displayed in the source area.
- **Listing source file names ([Source - Display])**
Displays a list of the source files making up the loaded load module. A source file can be selected from the list for display in the source area of the base window.
- **Displaying disassembly list ([Disassemble])**
Disassembles and displays in the DISASSEMBLE window the source file displayed in the source area of the base window. Breakpoint setting and step execution can also be performed at an assembly language level.
- **Displaying function call sequence ([Route])**
Displays the function names called up by the program to the function currently pointed to by the program counter (PC) in the order in which they were called.
- **Setting LED display conditions ([LED setting...])**
Sets conditions for displaying the LEDs on the bus monitor board connected to the E7000PC emulator. When no bus monitor board is connected to the emulator, this menu is shaded and cannot be selected.
- **Setting conditions for analog output of LED display contents ([LED out setting...])**
Sets conditions for outputting to the analog output pins the contents displayed by the LEDs on the bus monitor board connected to the E7000PC emulator. When no bus monitor board is connected to the emulator, this menu is shaded and cannot be selected.

- Controlling command area ([Command area])
Changes the size of the command area in the base window to three sizes.
- Symbol display button
Executes the same functions as the <DISPLAY> button on the tool bar from the menu.

5.1.5 Help Functions (Help Menu)

The menu for help functions is shown in figure 5-5.

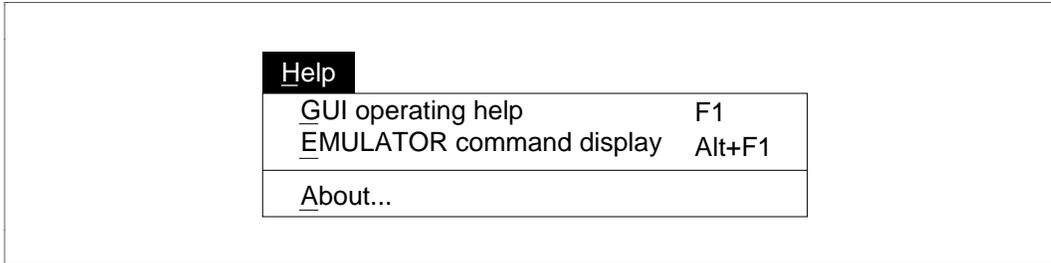


Figure 5-5 Menu for Help Functions (Help Menu)

- Displaying help information on GUI-2600 operations ([GUI operating help])
Displays GUI-2600 operation instructions in the GUI OPERATING HELP window.
- Displaying help information on emulator commands ([EMULATOR command display])
Displays the emulator command list in the EMULATOR COMMAND DISPLAY window.
- Displaying GUI-2600 information ([About...])
Displays information on the GUI-2600, such as the version No.

5.2 Menu Command Reference Format

This section describes menu commands in the format shown in figure 5-6.

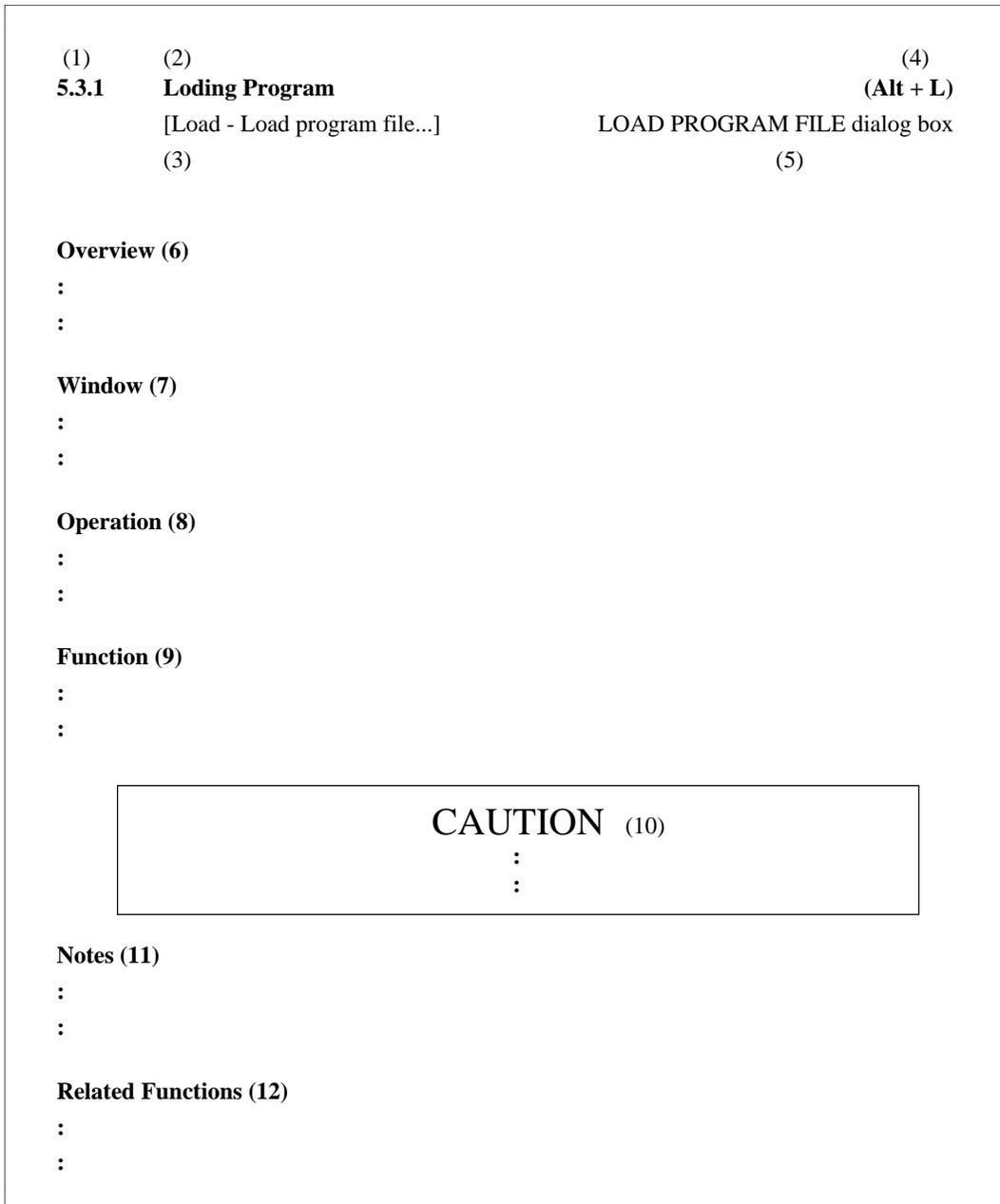


Figure 5-6 Command Description Format

The above item numbers indicate the following:

- (1) Section number
- (2) Command name
- (3) Command name in the menu
- (4) Short-cut keys for command input
- (5) Window name
- (6) Overview of command functions
- (7) Window display
- (8) Operations on the items in the window
- (9) Function of the command
- (10) Caution indicating the case which may result in loss of software
- (11) Exceptional conditions and notes on functions and operations (omitted when nothing is to be noted)
- (12) Related functions or commands

5.3 Filing Functions

5.3.1 Loading Program File

(Alt + L)

[Load - Load program file...]

LOAD PROGRAM FILE dialog box

Overview

Loads a load module file into the GUI-2600 and the E7000PC emulator and compares memory contents with the load module file for verification purposes.

Window

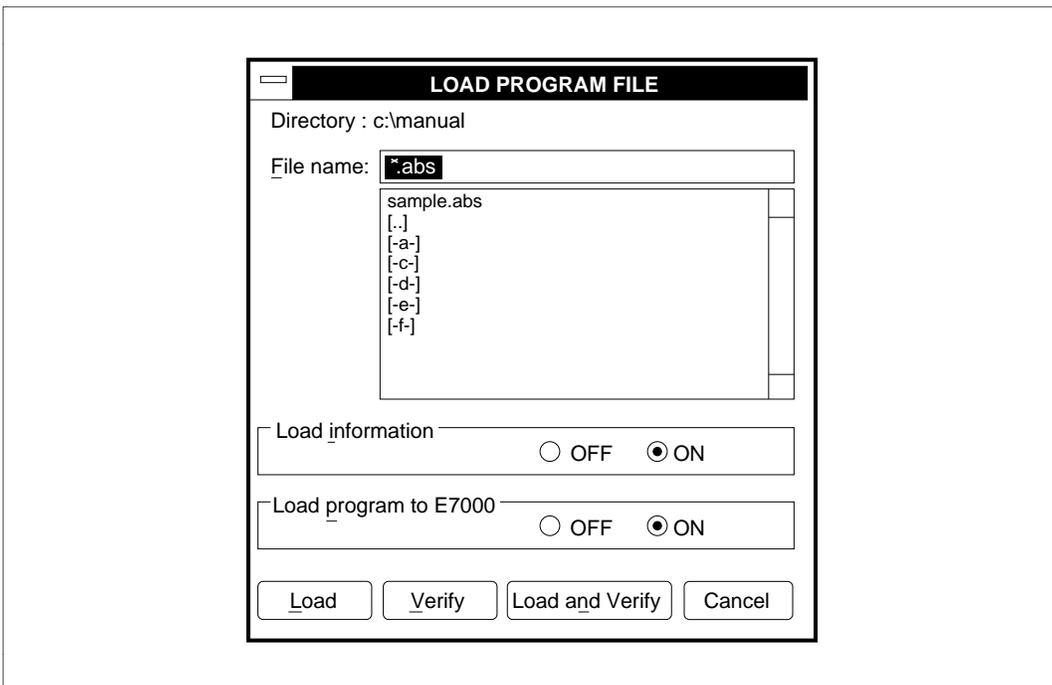


Figure 5-7 LOAD PROGRAM FILE Dialog Box

Operation

- [File name:] text box

Specifies the name of the file to be loaded. The file name can be selected from the list box; the selected file name is displayed in the text box.

Pressing the (Enter) key after specifying a file name or double-clicking the file name selected in the list box starts loading.

The following two types of files can be input:

- Absolute load module files
- Debugging information files

When the LOAD PROGRAM FILE dialog box opens, the text box displays *.abs, and therefore, the list box displays a list of file names having extension .abs in the current directory. A file having extension .abs is an absolute load module file. Debugging information files have extension .dbg.

When the file to be loaded is not in the current directory, select the drive and the directory from the list box to change the current directory, using the following procedure:

- (1) Select the target drive from the list box by double-clicking. Drive names are displayed in the format [-a-]. After selected, the drive name and current directory of the selected drive are displayed in [Directory:]. The current directory contents of the selected drive are displayed in the list box.
 - (2) Select a subdirectory from the list box by double-clicking. Subdirectory names are displayed in the format [directory]. After selected, the subdirectory name is added to [Directory:] and the subdirectory contents are displayed in the list box.
 - (3) Continue selecting subdirectories until the directory including the target file is reached.
- [Directory:]
Displays the current directory. When the drive or subdirectory is changed in [File name:], this display also changes.

- [Load information] group box
Specifies whether debugging information of the load module file is to be loaded into the GUI-2600, as follows:
 - <ON>: Loads debugging information (default at system initiation)
 - <OFF>: Does not load debugging information
- [Load program to E7000] group box
Specifies whether or not the load module file is to be loaded into the E7000PC emulator. When debugging information is to be loaded regardless of this group box setting, the load module is not loaded into the E7000PC emulator.
 - <ON>: Loads load module file (default at system initiation)
 - <OFF>: Does not load load module file
- <Load> button
Clicking the <Load> button starts loading a load module file, closes the LOAD PROGRAM FILE dialog box, and stores the dialog box settings.
- <Verify> button
Clicking the <Verify> button starts verifying memory contents against the load module file, closes the LOAD PROGRAM FILE dialog box, and stores the dialog box settings.
- <Load and Verify> button
Clicking the <Load and Verify> button starts loading the load module file. After loading is completed, the load module file and memory contents are automatically verified. The LOAD PROGRAM FILE dialog box is closed and the dialog box settings are stored.
- <Cancel> button
Clicking the <Cancel> button closes the LOAD PROGRAM FILE dialog box without storing the LOAD PROGRAM FILE dialog box settings. The dialog box settings return to those when the LOAD PROGRAM FILE dialog box was opened.

Function

- Program load
The following two types of files can be input:
 - Absolute load module files
 - Debugging information files

With this command, debugging information for the load module previously loaded will be lost.

When an entry address is set in the file to be loaded, the source file corresponding with the entry address is displayed in the source area after loading. The entry address can be set using the H-series linkage editor.

Whether or not to load debugging information can be specified. [Load information] is used to specify loading of debugging information controlled by the GUI-2600. <ON> should usually be selected. If <OFF> is selected, operations using symbols, such as source-level program debugging or symbol contents display, cannot be performed.

[Load program to E7000] is used to specify whether the load module is to be loaded to the E7000PC emulator. <ON> should usually be selected. Select <OFF> when debugging a ROM program.

When debugging information is to be loaded, the load module is not loaded into the E7000PC emulator regardless of the [Load program to E7000] group box setting. During load, the LOAD INFORMATION dialog box (figure 5-8) automatically opens. When load is completed, the LOAD INFORMATION dialog box automatically closes. Click the <Abort> button in the LOAD INFORMATION dialog box to abort load.

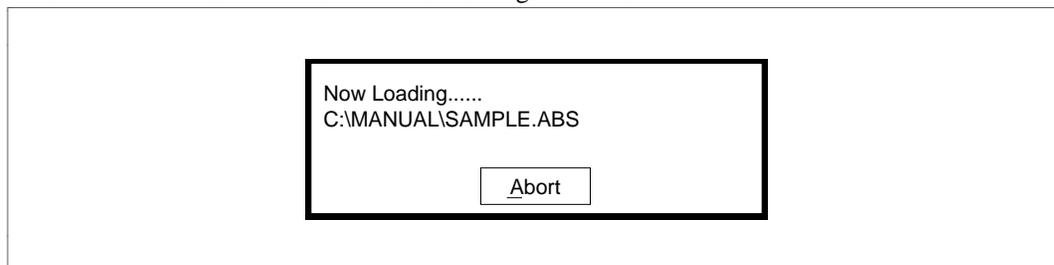


Figure 5-8 LOAD INFORMATION Dialog Box (Load Module File)

- Program verification

Clicking the <Verify> button verifies the memory contents against the file specified in [File name:]. Debugging information for the load module previously loaded is not lost.

During verification, the VERIFY INFORMATION dialog box (figure 5-9) automatically opens.

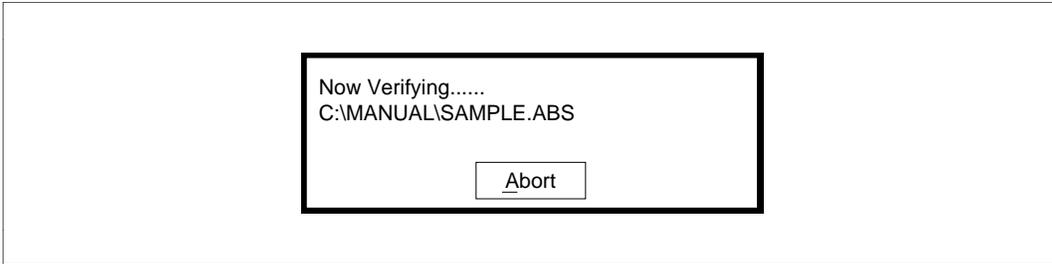


Figure 5-9 VERIFY INFORMATION Dialog Box (Load Module File)

When verification is completed, the VERIFY INFORMATION dialog box automatically closes. Click the <Abort> button in the VERIFY INFORMATION dialog box to abort verification.

If a verification error occurs, the verification results are displayed in the command area. The verification error message displayed in the command area is shown in figure 5-10.

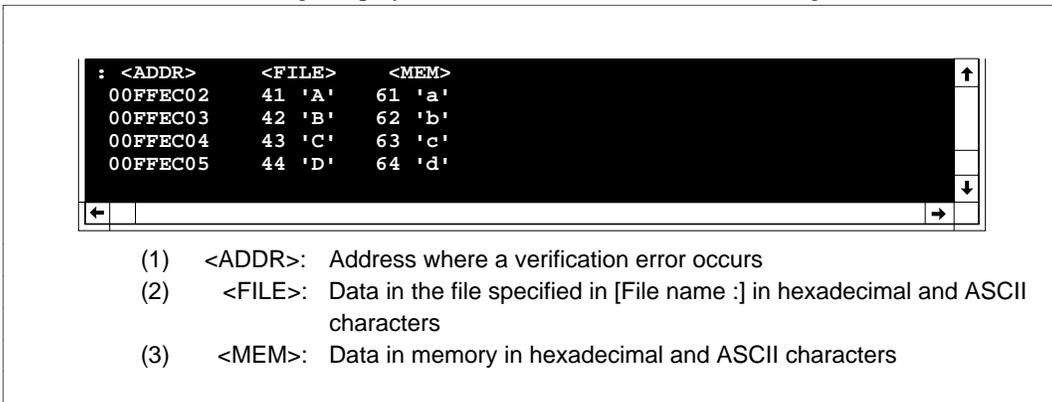


Figure 5-10 Verification Error Message

- Automatic verification after loading
Clicking the <Load and Verify> button starts loading the file specified in the [File name:] text box. When file loading is completed normally, the loaded file and the memory contents are automatically verified.

At this time, if <OFF> is selected in the [Load program to E7000] group box, verification is not performed.

- Debugging ROM programs
To perform source-level debugging of a program stored in ROM on the user system, load the file using the following procedure:

- (1) Create a load module file with debugging information corresponding to the program on the user system.
- (2) Input the name of this file in [File name:].
- (3) Select <ON> in [Load information].
- (4) Select <OFF> in [Load program to E7000].
- (5) When the load module is loaded, the debugging information is loaded only to the GUI-2600 and not to the E7000PC emulator and the user system. Accordingly, the program can be debugged at the source level on the GUI-2600.

Related Functions

Source area and command area

GUI commands: [Load - Load other file...] and [Load - E7000 load...]

5.3.2 Loading Other Files

(Alt + K)

[Load - Load other file...]

LOAD OTHER FILE dialog box

Overview

Loads an S-type load module file and a Hex-type load module file into the E7000PC emulator and compares memory contents with the file for verification purposes.

Window

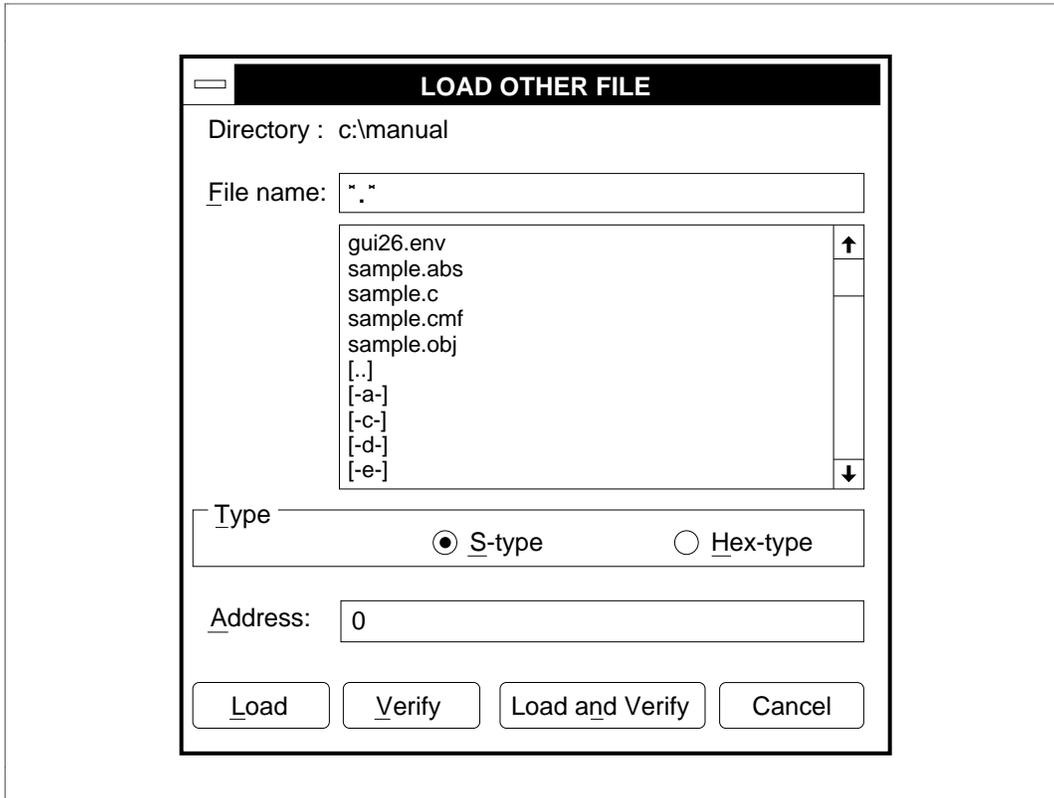


Figure 5-11 LOAD OTHER FILE Dialog Box

Operation

- [File name:] text box
Specifies the name of the file to be loaded. The file name can be selected from the list box; the selected file name is displayed in the text box.

Pressing the (Enter) key after specifying a file name or double-clicking the file name selected in the list box starts loading.

When the LOAD OTHER FILE dialog box opens, the text box displays *.* , and therefore, the list box displays a list of file names in the current directory.

When the file to be loaded is not in the current directory, select the drive and the directory from the list box to change the current directory, using the following procedure:

- (1) Select the target drive from the list box by double-clicking. Drive names are displayed in the format [-a-]. After selected, the drive name and current directory of the selected drive are displayed in [Directory:]. The current directory contents of the selected drive are displayed in the list box.
- (2) Select a subdirectory from the list box by double-clicking. Subdirectory names are displayed in the format [directory]. After selected, the subdirectory name is added to [Directory:] and the subdirectory contents are displayed in the list box.
- (3) Continue selecting subdirectories until the directory including the target file is reached.

- [Directory:]
Displays the current directory. When the drive or subdirectory is changed in [File name:], this display also changes.
- [Type] group box
Selects the format of the load module file to be loaded into the E7000PC emulator.
 - <S-type>: Loads an S-type load module file (default at system initiation)
 - <Hex-type>: Loads a Hex-type load module file
- [Address:] text box
Specifies the offset value from the start address in memory where the specified file is to be loaded.

- <Load> button
Clicking the <Load> button starts loading the file specified in [File name:], closes the LOAD OTHER FILE dialog box, and stores the dialog box settings.
- <Verify> button
Clicking the <Verify> button starts verifying memory contents against the file specified in [File name:], closes the LOAD OTHER FILE dialog box, and stores the dialog box settings.
- <Load and Verify> button
Clicking the <Load and Verify> button starts loading the file specified in [File name:]. After loading is completed, the load module file and memory contents are automatically verified. The LOAD OTHER FILE dialog box is closed and the dialog box settings are stored.
- <Cancel> button
Clicking the <Cancel> button closes the LOAD OTHER FILE dialog box without storing the LOAD OTHER FILE dialog box settings. The settings return to those when the LOAD OTHER FILE dialog box was opened.

Function

- Other file load
Clicking the <Load> button loads the file specified in [File name:] into the E7000PC emulator in the format selected in the [Type] group box. The offset value from the start address is input to the [Address:] text box. With this command, debugging information previously loaded by the load program file will not be lost.

During load, the LOAD INFORMATION dialog box (figure 5-12) automatically opens. When load is completed, the LOAD INFORMATION dialog box automatically closes. Click the <Abort> button in the LOAD INFORMATION dialog box to abort load.

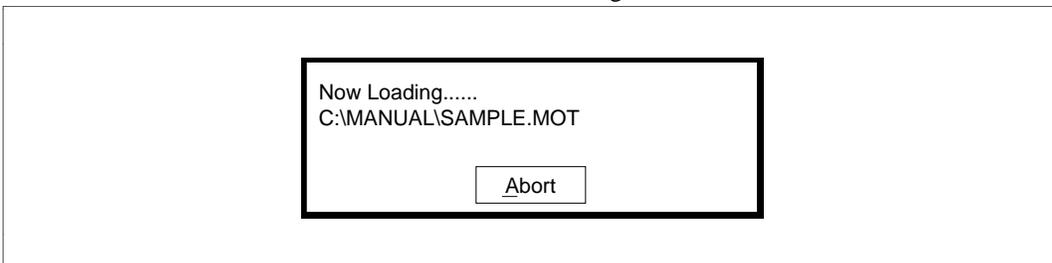


Figure 5-12 LOAD INFORMATION Dialog Box (Other File)

- Other file verification

Clicking the <Verify> button verifies the memory contents against the file specified in [File name:] in the format selected in the [Type] group box. Debugging information for the load module previously loaded is not lost.

During verification, the VERIFY INFORMATION dialog box (figure 5-13) automatically opens.

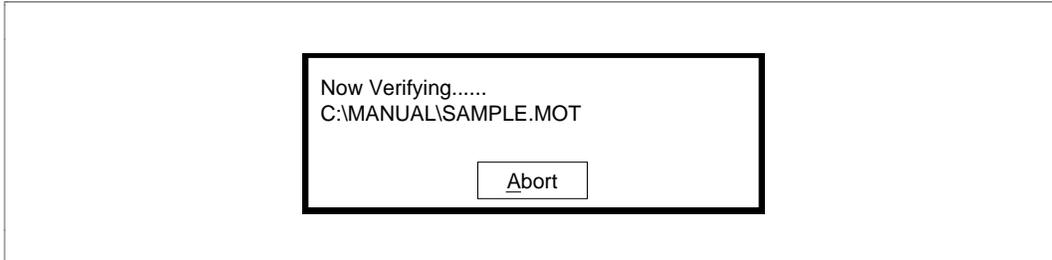


Figure 5-13 VERIFY INFORMATION Dialog Box (Other File)

When verification is completed, the VERIFY INFORMATION dialog box automatically closes. Click the <Abort> button in the VERIFY INFORMATION dialog box to abort verification.

If a verification error occurs, the verification results are displayed in the command area. The verification error message displayed in the command area is shown in figure 5-14.

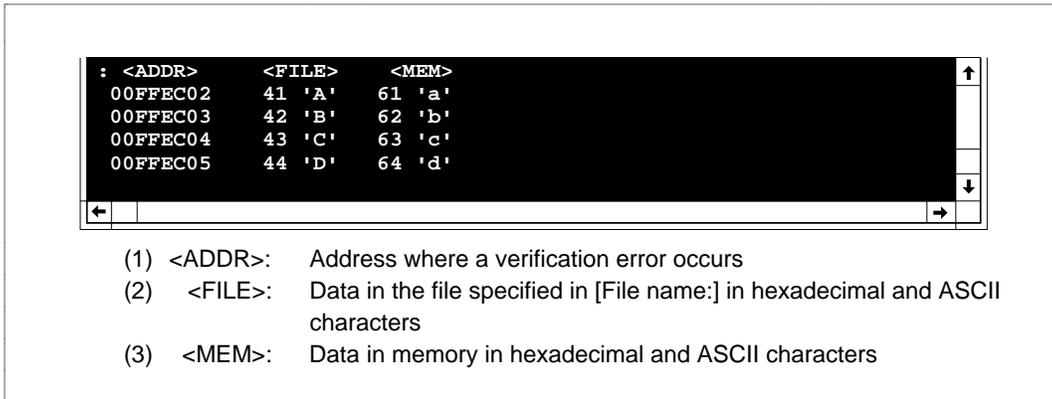


Figure 5-14 Verification Error Message (Other File)

- Automatic verification after loading

Clicking the <Load and Verify> button starts loading the file specified in [File name:] in the format selected in the [Type] group box. When file loading is completed normally, the loaded file and the memory contents are automatically verified.

Related Functions

Command area

GUI command: [Load - Load program file...]

5.3.3 Loading Directly into the E7000

(Alt + J)

[Load - E7000 load...]

E7000 LOAD dialog box

Overview

Loads a load module file to be debugged only into the E7000PC emulator. Symbol information can be registered in the E7000PC emulator in load module file units. Memory contents and the load module file contents can be verified.

Window

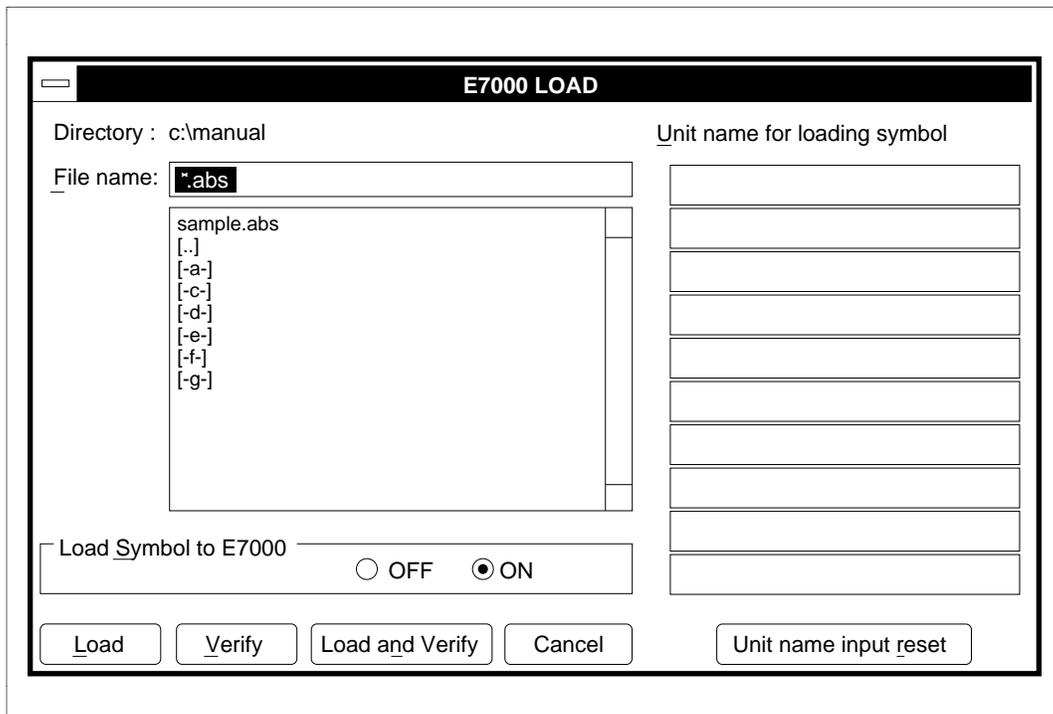


Figure 5-15 E7000 LOAD Dialog Box

Operation

- [File name:] text box

Specifies the name of the file to be loaded. The file name can be selected from the list box; the selected file name is displayed in the text box. The file that can be input is an absolute load module file.

Pressing the (Enter) key after specifying a file name or double-clicking the file name selected in the list box starts loading.

When the E7000 LOAD dialog box opens, the text box displays *.abs, and therefore the list box displays a list of file names having extension .abs in the current directory. A file having extension .abs is an absolute load module file.

When the file to be loaded is not in the current directory, select the drive and the directory from the list box to change the current directory, using the following procedure:

- (1) Select the target drive from the list box by double-clicking. Drive names are displayed in the format [-a-]. After selected, the drive name and the current directory of the selected drive are displayed in [Directory:]. The current directory contents of the selected drive are displayed in the list box.
- (2) Select a subdirectory from the list box by double-clicking. Subdirectory names are displayed in the format [directory]. After selected, the subdirectory name is added to [Directory:] and the subdirectory contents are displayed in the list box.
- (3) Continue selecting subdirectories until the directory including the target file is reached.

- [Directory:]

Displays the current directory. When the drive or subdirectory is changed in [File name:], this display also changes.

- [Load Symbol to E7000] group box

Specifies whether the symbol information is registered in one unit or all units.

<ON>: Registers all units of symbol information (default at system initiation)

<OFF>: Registers one unit of symbol information

- [Unit name for loading symbol] text box

Specifies the unit name of the symbol information to be registered.

- <Load> button
Clicking the <Load> button starts loading a load module file, closes the E7000 LOAD dialog box, and stores the dialog box settings.
- <Verify> button
Clicking the <Verify> button starts verifying memory contents against the load module file, closes the E7000 LOAD dialog box, and stores the dialog box settings.
- <Load and Verify> button
Clicking the <Load and Verify> button starts loading the load module file. After loading is completed, the load module file and the memory contents are automatically verified. The E7000 LOAD dialog box is closed and the dialog box settings are stored.
- <Cancel> button
Clicking the <Cancel> button closes the E7000 LOAD dialog box without storing the E7000 LOAD dialog box settings. The dialog box settings return to those when the E7000 LOAD dialog box was opened.
- <Unit name input reset> button
Clicking the <Unit name input reset> button deletes the contents of the [Unit name for loading symbol] text box.

Function

- Program load
An absolute load module file program can be loaded into the E7000PC emulator.
With this command, debugging information for the load module previously loaded will not be lost.

When loading a load module file, symbol information registration and registered symbol information deletion can be specified.

The [Load Symbol to E7000] group box selects symbol information registration. Selecting <ON> registers all symbol information of the load module to the E7000PC emulator; selecting <OFF> registers only the symbol information in the unit specified in the [Unit name for loading symbol] text box to the E7000PC emulator.

During load, the LOAD INFORMATION dialog box (figure 5-16) automatically opens. When load is completed, the LOAD INFORMATION dialog box automatically closes. Click the <Abort> button in the LOAD INFORMATION dialog box to abort load.

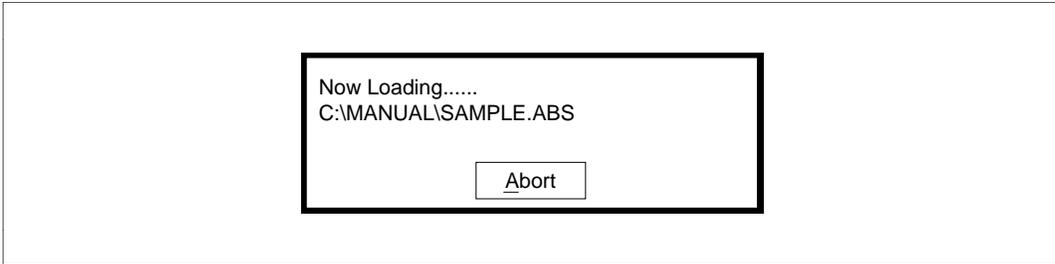


Figure 5-16 LOAD INFORMATION Dialog Box (E7000 LOAD)

- Program verification

Clicking the <Verify> button verifies the memory contents against the file specified in [File name:]. Debugging information for the load module previously loaded is not lost.

During verification, the VERIFY INFORMATION dialog box (figure 5-17) automatically opens.

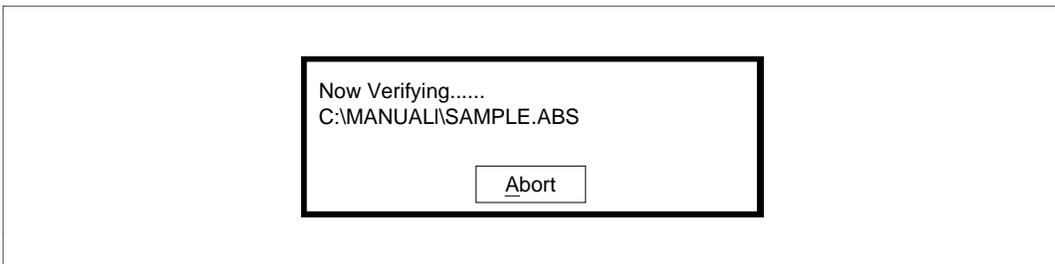


Figure 5-17 VERIFY INFORMATION Dialog Box (E7000 LOAD)

When verification is completed, the VERIFY INFORMATION dialog box automatically closes. Click the <Abort> button in the VERIFY INFORMATION dialog box to abort verification.

If a verification error occurs, the verification results are displayed in the command area. The verification error message displayed in the command area is shown in figure 5-18.

: <ADDR>	<FILE>	<MEM>
00FFEC02	41 'A'	61 'a'
00FFEC03	42 'B'	62 'b'
00FFEC04	43 'C'	63 'c'
00FFEC05	44 'D'	64 'd'

- (1) <ADDR>: Address where a verification error occurs
- (2) <FILE>: Data in the load module file in hexadecimal and ASCII characters
- (3) <MEM>: Data in memory in hexadecimal and ASCII characters

Figure 5-18 Verification Error Message

- Automatic verification after loading
Clicking the <Load and Verify> button starts loading the file specified in [File name:]. When file load is completed normally, the loaded file and the memory contents are automatically verified.

Related Functions

Command area

GUI command: [Load - Load program file...]

5.3.4 Inputting and Executing Command File

(Alt + I)

[Input from...]

INPUT FROM dialog box

Overview

Inputs a command file and automatically executes emulator commands.

Window

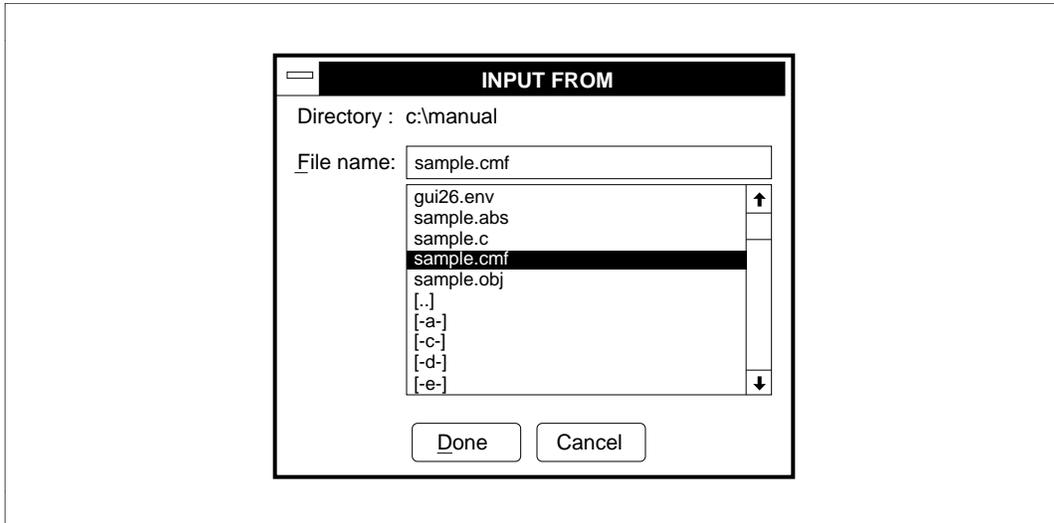


Figure 5-19 INPUT FROM Dialog Box

Operation

- [File name:] text box
Specifies the name of the command file to be input. The file name can be selected from the list box; the selected file name is displayed in the text box.

Pressing the (Enter) key after specifying a file name or double-clicking the file name selected in the list box starts automatic execution.

When the INPUT FROM dialog box opens, the text box displays *.* , and therefore the list box displays all file names in the current directory. When the file to be input is not in the current directory, select the drive and the directory from the list box to change the current directory, using the following procedure:

- (1) Select the target drive from the list box by double-clicking. Drive names are displayed in the format [-a-]. After selected, the drive name and current directory of the selected drive are displayed in [Directory:]. The current directory contents of the selected drive are displayed in the list box.
 - (2) Select a subdirectory from the list box by double-clicking. Subdirectory names are displayed in the format [directory]. After selected, the subdirectory name is added to [Directory:] and the subdirectory contents are displayed in the list box.
 - (3) Continue selecting subdirectories until the directory including the target file is reached.
- [Directory:]
Displays the current directory. When the drive or subdirectory is changed in [File name:], this display also changes.
 - <Done> button
Clicking the <Done> button automatically executes the command file specified in [File name:]. The INPUT FROM dialog box settings are stored in the GUI-2600 and the dialog box is closed. The command execution results are displayed in the command area.
 - <Cancel> button
Clicking the <Cancel> button closes the INPUT FROM dialog box without storing the INPUT FROM dialog box settings. The settings return to those when the INPUT FROM dialog box was opened.

Function

The emulator commands included in the command file specified in [File name:] are automatically executed. For details on command file creation, refer to appendix C, Command File.

Clicking the <Done> button starts emulator command execution. The command execution results are displayed in the command area.

Click the <STOP> button on the tool bar to suspend command execution. The stop confirmation message box shown in figure 5-20 is displayed. Click the <No> button to continue emulator command execution. When the <Yes> button is clicked, emulator command execution terminates.



Figure 5-20 Stop Confirmation Message Box (INPUT FROM Dialog Box)

Related Functions

Command area and <STOP> button

5.3.5 Storing Command Area Contents

(Alt + O)

[Output to...]

OUTPUT TO dialog box

Overview

Outputs to a file the contents input to and displayed in the command area.

Window

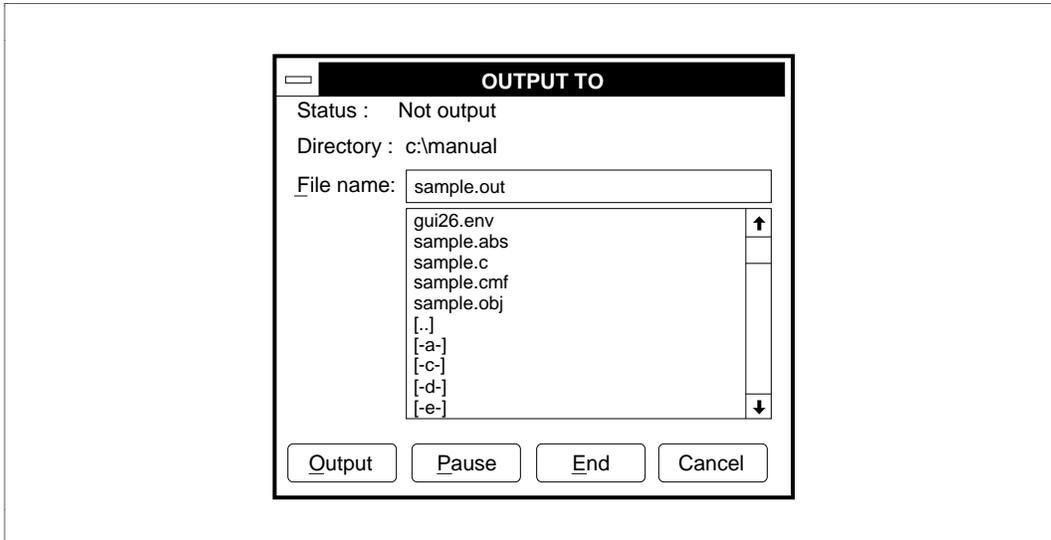


Figure 5-21 OUTPUT TO Dialog Box

Operation

- [Status:] text box
Displays file output status, as shown in table 5-1.

Table 5-1 Output Status Messages (OUTPUT TO Dialog Box)

Message	Description
Not output	No file output
Output	File being output
Pause	File output suspended

- [File name:] text box

Specifies the name of the file for storing the command area contents. The specified file is saved in the current directory, which is displayed in [Directory:].

Pressing the (Enter) key after specifying a file name or double-clicking the file name selected in the list box starts output to the specified file.

To save a file in a different directory, select the drive and the directory from the list box to change the current directory, using the following procedure:

- (1) Select the target drive from the list box by double-clicking. Drive names are displayed in the format [-a-]. After selected, the drive name and current directory of the selected drive are displayed in [Directory:]. The current directory contents of the selected drive are displayed in the list box.
- (2) Select a subdirectory from the list box by double-clicking. Subdirectory names are displayed in the format [directory]. After selected, the subdirectory name is added to [Directory:] and the subdirectory contents are displayed in the list box.
- (3) Continue selecting subdirectories until the directory including the target file is reached.

- [Directory:]

Displays the current directory. When the drive or subdirectory is changed in [File name:], this display also changes.

- <Output> button

Clicking the <Output> button starts outputting command area information to the file specified in [File name:]. When Pause is displayed at [Status:], that is, file output has been temporarily suspended, clicking the <Output> button restarts the output. After this button is clicked, Output is displayed at [Status:].

- <Pause> button

When Output is displayed at [Status:], clicking the <Pause> button suspends outputting to a file and changes display at [Status:] to Pause. Clicking the <Output> button restarts the output.

- <End> button

When Output or Pause is displayed at [Status:], clicking the <End> button terminates outputting to a file and changes file output status to Not output at [Status:].

- <Cancel> button

Clicking the <Cancel> button closes the OUTPUT TO dialog box without storing the OUTPUT TO dialog box settings. The settings return to those when the OUTPUT TO dialog box was opened.

Function

- Output to a file

Clicking the <Output> button outputs the contents input to and displayed in the command area to the file specified in [File name:]. Even when the command area of the base window is closed, the display contents are stored in the file.

Outputting to a file depends on the file output status displayed at [Status:].

— Not output is displayed

Clicking the <Output> button starts outputting command area information to the file specified in [File name:]. When a file having the same name as that specified in [File name:] already exists, the message box shown in figure 5-22 appears. Click <Yes> to start overwriting the file specified in [File name:]. When <No> is clicked, display returns to the OUTPUT TO dialog box.

CAUTION

The existing file will be overwritten when the <Yes> button is clicked in the file overwrite confirmation message box.

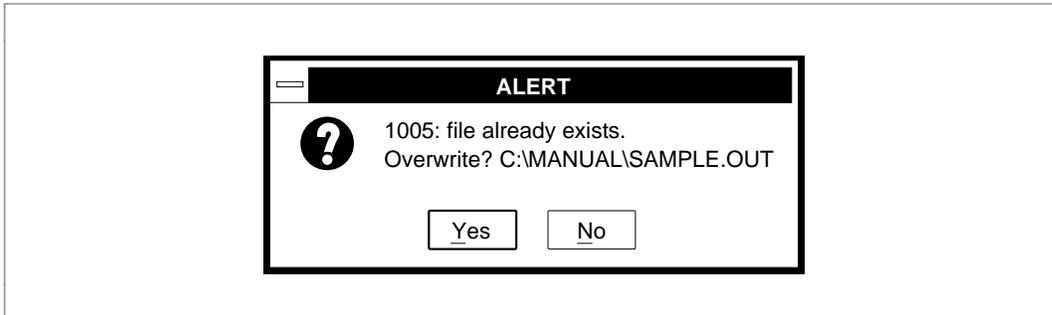


Figure 5-22 File Overwrite Confirmation Message (OUTPUT TO Dialog Box)

— Pause is displayed

Clicking the <Output> button restarts suspended file output.

- Relation between file output status and buttons
Clicking buttons in the OUTPUT TO dialog box changes the file output status displayed at [Status:], as shown in table 5-2.

Table 5-2 Output File Status Message and Buttons in OUTPUT TO Dialog Box

Message	Description	Related Button
Not output	No file output	After initiation or <End> button clicked
Output	File output started or restarted	<Output> button clicked
Pause	File output suspended	<Pause> button clicked

Related Function

Command area

5.3.6 Terminating GUI-2600

(Alt + F4)

[Quit]

QUIT dialog box

Overview

Terminates the GUI-2600 with a prompt asking whether to store debugging settings in a backup file.

Window

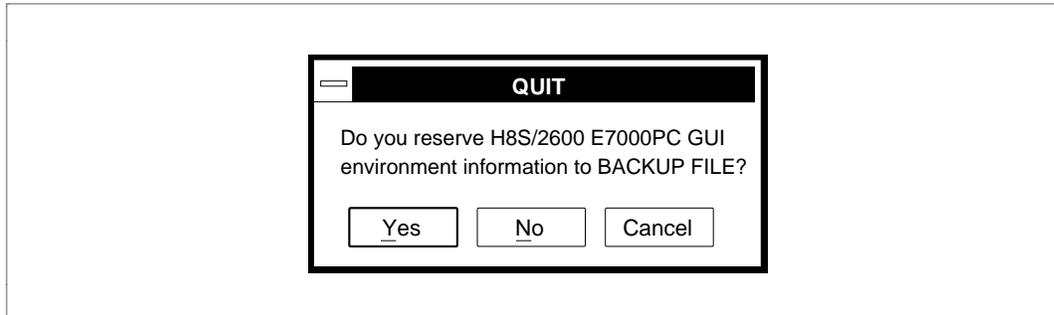


Figure 5-23 QUIT Dialog Box

Operation

- <Yes> button
Clicking the <Yes> button stores debugging settings in a backup file and then terminates the GUI-2600.
- <No> button
Clicking the <No> button terminates the GUI-2600 without storing the settings.
- <Cancel> button
Clicking the <Cancel> button returns to the previous state without terminating the GUI-2600.

Function

- Output to a backup file

Clicking the <Yes> button in the QUIT dialog box displays the BACKUP FILE dialog box for storing a backup file, as shown in figure 5-24.

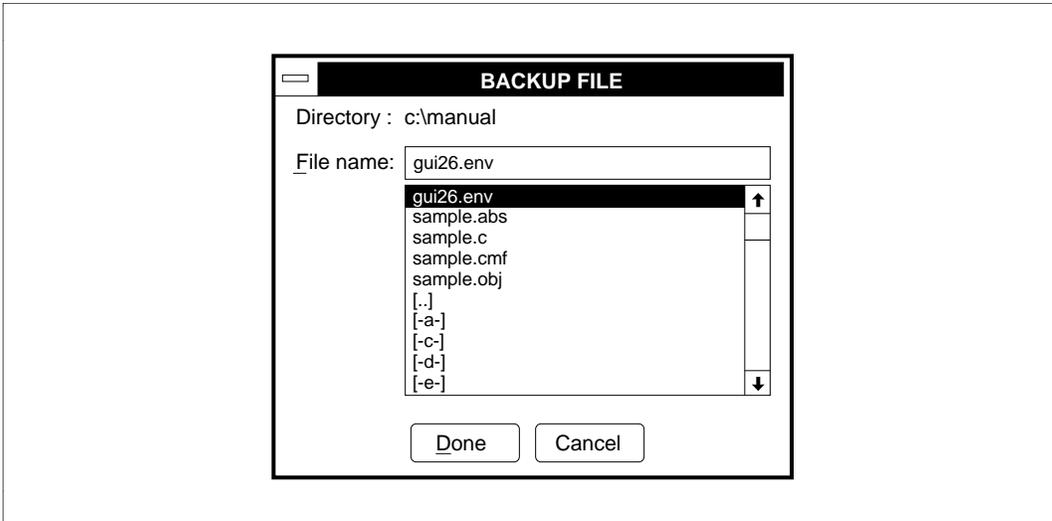


Figure 5-24 BACKUP FILE Dialog Box

— [File name:] text box

Specifies the name of the backup file for storing debugging settings. The specified file is output to the current directory, which is displayed in [Directory:]. Pressing the (Enter) key after specifying a file name or double-clicking the file name selected in the list box starts storing the backup file.

To output a file to a different directory, select the drive and the directory from the list box to change the current directory, using the following procedure:

- (1) Select the target drive from the list box by double-clicking. Drive names are displayed in the format [-a-]. After selected, the drive name and current directory of the selected drive are displayed in [Directory:]. The current directory contents of the selected drive are displayed in the list box.
- (2) Select a subdirectory from the list box by double-clicking. Subdirectory names are displayed in the format [directory]. After selected, the subdirectory name is added to [Directory:] and the subdirectory contents are displayed in the list box.
- (3) Continue selecting subdirectories until the directory including the target file is reached.

— [Directory:]

Displays the current directory. When the drive or subdirectory is changed in [File name:], this display also changes. If a backup file is found when the BACKUP FILE dialog box is open, the subdirectory of the backup file is displayed.

— <Done> button

Clicking the <Done> button starts storing debugging settings to the backup file. When a file having the same name as that specified in [File name:] already exists, the message box shown in figure 5-25 appears. Click <Yes> to start overwriting the backup file specified in [File name:]. When <No> is clicked, the display returns to the QUIT dialog box.

CAUTION

The existing file will be overwritten when the <Yes> button is clicked in the file overwrite confirmation message box.

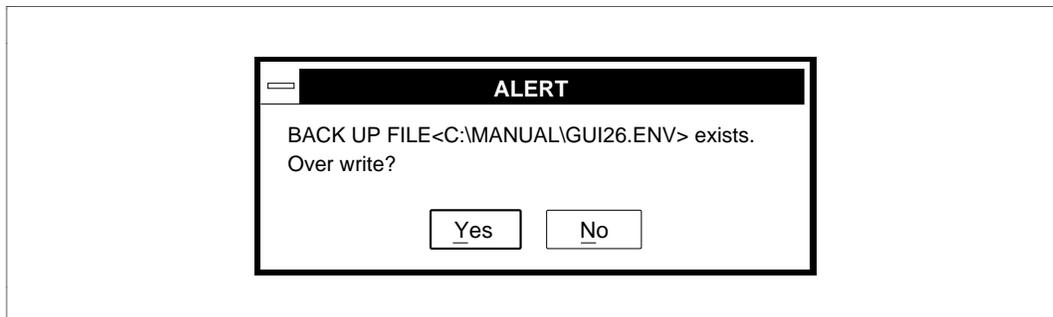


Figure 5-25 File Overwrite Confirmation Message (BACKUP FILE Dialog Box)

— <Cancel> button

Clicking the <Cancel> button closes the BACKUP FILE dialog box and returns to the QUIT dialog box without storing the BACKUP FILE dialog box settings. The settings return to those when the BACKUP FILE dialog box was opened.

- Information stored in the backup file
The information to be stored in the backup file is listed in table 5-3.

Table 5-3 Information Stored in Backup File (QUIT Dialog Box)

Information	Command
Directory and file extension	[Load - Load program file...], [Load - Load other file...], [Load - E7000 load...], [Input from...], [Output to...], [Quit] (file extensions are not stored), [Trace display - Text], [Memory - Dump]
Settings in dialog box	[Load - Load program file...], [Load - Load other file...], [Load - E7000 load...], [Input from...], [Output to...], [Execution mode...]* ¹ , [Go...], [Break condition 1,2,3,4,5,6...]* ¹ , [Trace mode...], [Trace condition...]* ¹ , [Trace display - Setting...], [Memory - Setting...], [Source - Setting...], [LED setting...], [LED out setting...]
Window location and size	All debug windows* ²

Notes: 1. The settings of these commands must be set to the E7000PC emulator when the E7000PC emulator is re-initialized using the <Done> button of each command because they are stored only in the GUI-2600.

2. All debug windows return to the stored size and location when re-opened. Note, however, that the dialog box position cannot be stored.

Related Function

Loading backup file

5.4 Execution Functions

5.4.1 Specifying Emulation Execution Conditions

(Ctrl + E)

[Execution mode...]

EXECUTION MODE dialog box

Overview

Specifies emulation execution conditions.

Window

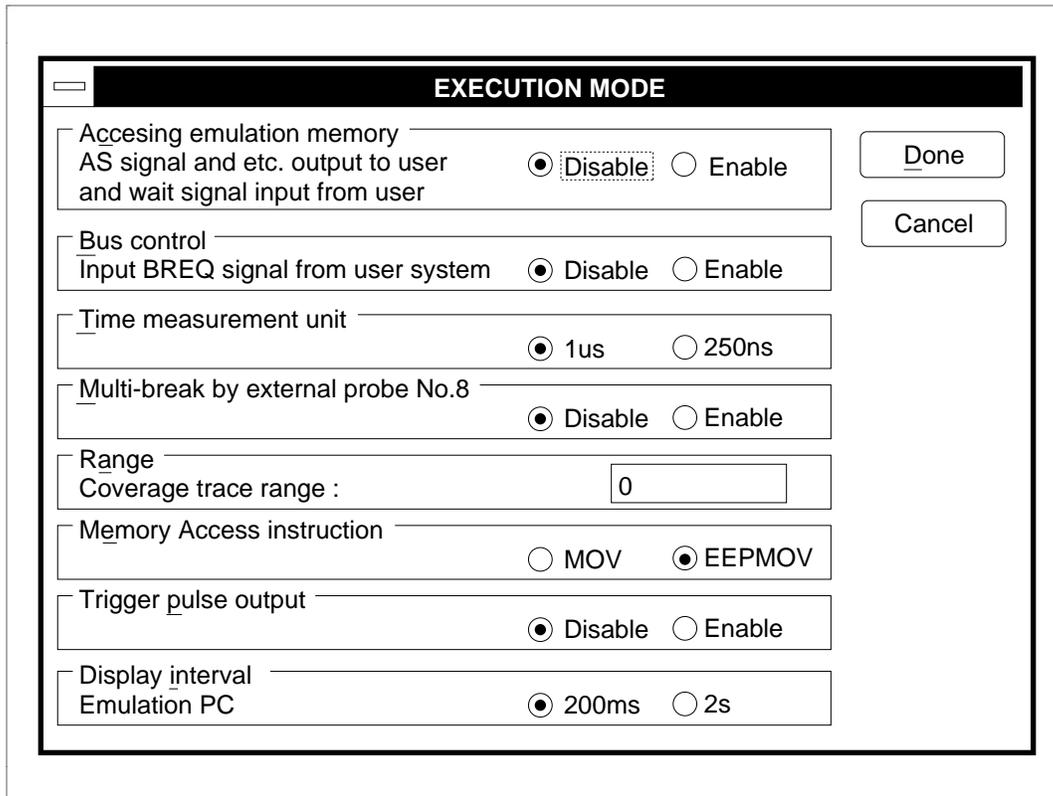


Figure 5-26 EXECUTION MODE Dialog Box

Operation

- [Accessing emulation memory] group box
Specifies if the AS, RD, and WR signals are output to the user system and if the WAIT signal is input from the user system, when emulation memory is accessed as follows:
 - <Disable>: Disables signal input/output (default at system initiation)
 - <Enable>: Enables signal input/output
- [Bus control] group box
Specifies if the bus request signal input (BREQ) is enabled in the user system during emulation as follows:
 - <Disable>: Disables BREQ signal input (default at system initiation)
 - <Enable>: Enables BREQ signal input
- [Time measurement unit] group box
Specifies the time unit to be used when measuring execution time with the [Go...] command and with the PERFORMANCE_ANALYSIS command (emulator command) as follows:
 - <1us>: Measured in 1- μ s units (default at system initiation)
 - <250ns>: Measured in 250-ns units
- [Multi-break by external probe No.8] group box
Specifies if external probe 8 on the emulator pod is used for a multibreak function as follows:
 - <Disable>: Not used for multibreak function (default at system initiation)
 - <Enable>: Used for multibreak function
- [Range]
[Coverage trace range:] text box
Specifies the coverage trace range of the SET_COVERAGE command (emulator command). The default setting is 0.

A value corresponding to one of the address ranges listed in table 5-4 is specified. The address ranges are in 2-Mbyte units.

Table 5-4 Address Range Specification (EXECUTION MODE Dialog Box)

Specification	Address Range
0	H'000000 - H'1FFFFFF
1	H'100000 - H'2FFFFFF
2	H'200000 - H'3FFFFFF
:	:
D	H'D00000 - H'EFFFFFF
E	H'E00000 - H'FFFFFF
F	H'F00000 - H'FFFFFF, H'000000 - H'0FFFFFF

- [Memory Access instruction] group box
Specifies if program loading and saving are performed with the EEPMOV instruction or the MOV instruction as follows:
 - <MOV>: Uses the MOV instruction
 - <EEPMOV>: Uses the EEPMOV instruction (default at system initiation)
- [Trigger pulse output] group box
Specifies whether to continue program execution and output a pulse from the trigger output pin in the emulator pod when the PC break conditions (set by the [Break...] command) or hardware break conditions (set by the [Break condition1,2,3,4,5,6...] command) have been satisfied. A pulse is output from the trigger output probe pin at PC break and from the trigger output pin at hardware break.
 - <Disable>: Terminates program execution instead of outputting a pulse when the break conditions have been satisfied (default at system initiation)
 - <Enable>: Continues program execution and outputs a pulse when the break conditions have been satisfied
- [Display interval] group box
Selects the display interval of the program counter to be displayed in the emulator message area during emulation.
 - <200ms>: Displays the program counter every 200 ms (default at system initiation)
 - <2s>: Displays the program counter every 2 s.
- <Done> button
Clicking the <Done> button sends the EXECUTION MODE dialog box settings to the E7000PC emulator, closes the EXECUTION MODE dialog box, and stores the above settings in the GUI-2600.

- <Cancel> button

Clicking the <Cancel> button closes the EXECUTION MODE dialog box and returns the system to its state before the EXECUTION MODE dialog box was opened without storing the above settings.

Function

Specifies emulation execution conditions.

Related Functions

<STEP> and <STEP_OVER> buttons

GUI commands: [Go...], [Break...], and [Break condition1,2,3,4,5,6...]

Emulator commands: RESET, PERFORMANCE_ANALYSIS, and SET_COVERAGE

5.4.2 Performing Program Emulation

(Alt + Ctrl + G)

[Go...]

GO dialog box

Overview

Performs emulation.

Window

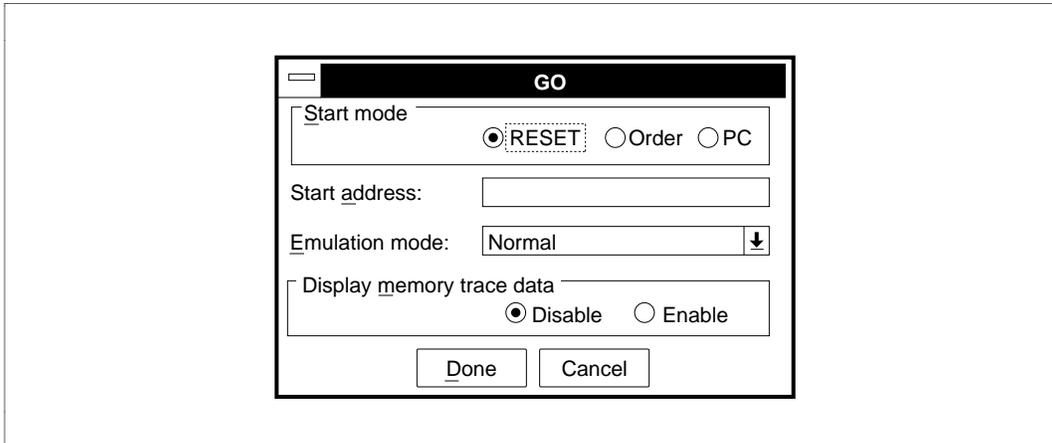


Figure 5-27 GO Dialog Box

Operation

- [Start mode] group box
Specifies the emulation start method as follows:
 - <RESET>: Initiates emulation from the reset vector (default at system initiation)
 - <Order>: Initiates emulation from the address specified in [Start address:]
 - <PC>: Initiates emulation from the current PC
- [Start address:] text box
When <Order> is specified in [Start mode], specifies the emulation start address in hexadecimal or as a symbol. When there is no specification, an error occurs.
- [Emulation mode:] drop-down list box
Specifies the emulation method.
 - <Normal>: Normal mode (default at system initiation)

- <Sequential break 2-1>: Sequential break mode 1
 The program stops only when hardware break conditions set by [Break condition1,2...] commands are satisfied in the sequence of <condition 2> followed by <condition 1>. In this mode, all conditions specified with [Break condition3,4,5,6...] commands are ignored.
- <Sequential break 3-2-1>: Sequential break mode 2
 The program stops only when hardware break conditions set by [Break condition1,2,3...] commands are satisfied in the sequence of <condition 3>, <condition 2>, and <condition 1>. In this mode, all conditions specified with [Break condition4,5,6...] commands are ignored.
- <Sequential break 4-3-2-1>: Sequential break mode 3
 The program stops only when hardware break conditions set by [Break condition1,2,3,4...] commands are satisfied in the sequence of <condition 4>, <condition 3>, <condition 2>, and <condition 1>. In this mode, all conditions specified with [Break condition5,6...] commands are ignored.
- <Interval time measurement>: Time interval measurement mode 1
 The execution time from the point when hardware break conditions set by the [Break condition2...] command are satisfied until those set by the [Break condition1...] command are satisfied is measured. The program stops when a hardware break condition set by the [Break condition1...] command is satisfied.
 In this mode, all conditions specified with the [Break...] and [Break condition3,4,5,6...] menu commands, and the BREAK_SEQUENCE and PERFORMANCE_ANALYSIS emulator commands, are ignored. Note that parallel mode cannot be entered from this mode.
- <Interval time repeat measurement>: Time interval measurement mode 2
 The total execution time from the point when hardware break conditions set by the [Break condition2...] command are satisfied until those set by the [Break condition1...] command are satisfied is measured. Even if these break conditions are satisfied, program execution does not stop. When these conditions are satisfied twice or more, the time is added to the previous measured time.
 In this mode, all conditions specified with the [Break...] and [Break condition3,4,5,6...] menu commands, and the BREAK_SEQUENCE and PERFORMANCE_ANALYSIS emulator commands, are ignored. Note that parallel mode cannot be entered from this mode.
- <Cycle reset 32 μ s>: Cycle reset mode 1
 Continues execution while forcibly inputting the RES signal to the H8S/2655-series MCU every specified time and simultaneously outputting the oscilloscope trigger signal from the trigger output probe pin. In this mode, conditions specified with the [Break...], [Break condition1,2,3,4,5,6...], [Trace condition...] menu commands, and BREAK_SEQUENCE emulator command are ignored. Also, parallel mode cannot be entered from this mode.
 Note that the [Display memory trace data] specification is automatically set to <Disable>.

(These conditions are the same in the following cycle reset modes.)

- <Cycle reset 96 μ s>: Cycle reset mode 2
 - <Cycle reset 512 μ s>: Cycle reset mode 3
 - <Cycle reset 1.024ms>: Cycle reset mode 4
 - <Cycle reset 5.12ms>: Cycle reset mode 5
 - <Cycle reset 10.24ms>: Cycle reset mode 6
 - <Cycle reset 51.2ms>: Cycle reset mode 7
 - <Cycle reset 102.4ms>: Cycle reset mode 8
 - <Cycle reset 512ms>: Cycle reset mode 9
 - <Ignore break>: Break conditions are ignored.
 - Ignores the settings of the [Break...] and [Break condition1,2,3,4,5,6...] commands while executing the program.
- [Display memory trace data] group box
Specifies whether the memory contents at the address specified in [Trace Memory Condition] of the [Trace mode...] command is displayed in the emulation information area during emulation execution, as follows:
 - <Disable>: Disables memory content display (default at system initiation)
 - <Enable>: Enables memory content display
 - <Done> button
Clicking the <Done> button stores the GO dialog box settings, closes the GO dialog box, and starts emulation execution.
 - <Cancel> button
Clicking the <Cancel> button closes the GO dialog box and returns the system to its state before the GO dialog box was opened without storing the above settings.

Function

- Emulation mode
Emulation execution is initiated in the mode specified in [Start mode]. One of the following emulation modes can be selected.
 - Sequential break mode 1
The program stops only when hardware break conditions set by [Break condition1,2...] commands are satisfied in the sequence of <condition 2> followed by <condition 1>.
 - Sequential break mode 2
The program stops only when hardware break conditions set by [Break condition1,2,3...] commands are satisfied in the sequence of <condition 3>, <condition 2>, and <condition 1>.

- Sequential break mode 3

The program stops only when hardware break conditions set by [Break condition1,2,3,4...] commands are satisfied in the sequence of <condition 4>, <condition 3>, <condition 2>, and <condition 1>.
- Time interval measurement mode 1

The execution time from the point when hardware break conditions set by the [Break condition2...] command are satisfied until those set by the [Break condition1...] command are satisfied is measured.
- Time interval measurement mode 2

The total execution time from the point when hardware break conditions set by the [Break condition2...] command are satisfied until those set by the [Break condition1...] command are satisfied is measured.
- Cycle reset modes

A RES signal is forcibly output from the E7000PC emulator to the H8S/2655-series MCU at specified intervals. At the same time, while continuing emulation execution, a trigger signal for an oscilloscope is output from the trigger output probe pin. In this mode, all break conditions and trigger conditions are invalidated.
- Ignoring break conditions

Ignores the settings of the [Break...] and [Break condition1,2,3,4,5,6...] commands while executing the program.
- Parallel mode

Enters parallel mode when the (Space) or (Enter) key is input from the command area during emulation. For details, refer to the E7000PC emulator user's manual.
- Forcible emulation termination

To forcibly terminate emulation, click the <STOP> button or input the (Ctrl + C) keys. In parallel mode, execute the ABORT command (emulator command) in the command area.
- Causes of GO command emulation termination

Table 5-5 lists the messages displayed in the emulation information area when emulation is terminated.

Table 5-5 Emulation Termination Causes (GO Dialog Box)

Display	Termination Cause
BREAK KEY	Forcibly terminated by pressing the <STOP> button or the (Ctrl + C) keys
BREAK POINT xxxxxxxx	Emulation stops at a breakpoint specified with the [Break...] command
STOP ADDRESS	Emulation initiated by the <CONTINUE> button stops at where the cursor indicates
BREAK SEQUENCE	PC break condition specified with the BREAK_SEQUENCE command (emulator command) was satisfied
BREAK CONDITION1	A break condition specified with the [Break condition1...] command was satisfied
BREAK CONDITION2	A break condition specified with the [Break condition2...] command was satisfied
BREAK CONDITION3	A break condition specified with the [Break condition3...] command was satisfied
BREAK CONDITION4	A break condition specified with the [Break condition4...] command was satisfied
BREAK CONDITION5	A break condition specified with the [Break condition5...] command was satisfied
BREAK CONDITION6	A break condition specified with the [Break condition6...] command was satisfied
BREAK CONDITION1,2,3,4,5,6	A break condition specified with a [Break condition1,2,3,4,5,6...] command was satisfied
BREAK CONDITION S1	Sequential break conditions specified with [Break condition1,2...] commands were satisfied
BREAK CONDITION S2	Sequential break conditions specified with [Break condition1,2,3...] commands were satisfied
BREAK CONDITION S3	Sequential break conditions specified with [Break condition1,2,3,4...] commands were satisfied
GUARDED AREA ACCESSED	A guarded area was accessed
WRITE PROTECT	A write-protected area was written to
ILLEGAL INSTRUCTION	A break instruction was executed
NO EXECUTION	The user program was not executed
RESET IN BY E7000	Forcibly terminated with the RES signal output from the E7000PC emulator because an error has occurred in the user system
DMA GUARDED OR WRITE PROTECT	A write-protected area was written to or a guarded memory area was accessed by DMA during [Break...] command processing

Note

If a condition of the [Break condition1,2,3,4,5,6...] command is satisfied during program execution, program execution may not stop immediately; it may stop after executing multiple instructions.

Related Functions

GUI commands: [Execution mode...], [Break...], [Break condition1,2,3,4,5,6...], [Trace mode...], and [Trace condition...]

Emulator commands: BREAK_SEQUENCE and PERFORMANCE_ANALYSIS

5.4.3 Setting and Cancelling Breakpoints

(Alt + Ctrl + B)

[Break...]

BREAK dialog box

Overview

Sets and cancels breakpoints.

Window

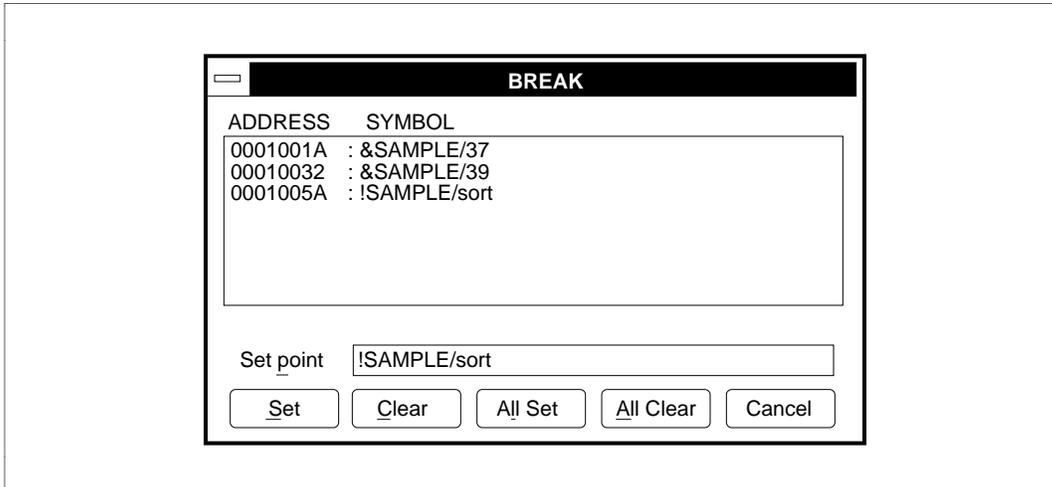


Figure 5-28 BREAK Dialog Box

Operation

- Breakpoint display area

Displays breakpoint settings in the format shown in figure 5-29.

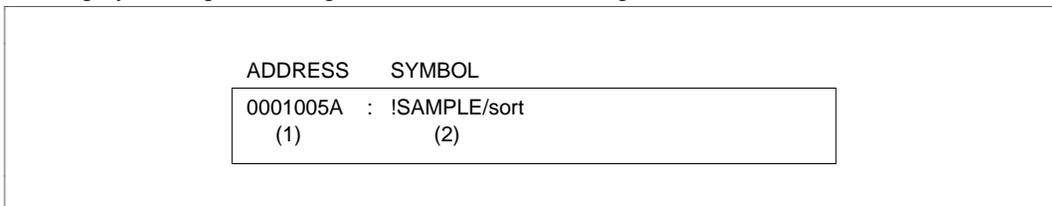


Figure 5-29 Display Format for Breakpoint Information (BREAK Dialog Box)

(1) ADDRESS: Displays the address where a breakpoint is set

(2) SYMBOL: Displays a symbol name, a line number symbol, or an address (when there is no corresponding symbol information)

Displayed breakpoints are set by the [Break...] command or the <SET> button on the tool bar within the base window. The breakpoints set by the BREAK command (emulator command) in the command area are not displayed.

By selecting with the mouse pointer and double-clicking a line including a breakpoint, the information displayed under SYMBOL is automatically set in [Set point:].

- [Set point:] text box
Accepts input of an address in which a breakpoint is to be set or cancelled in hexadecimal or as a symbol.
- <Set> button
Clicking the <Set> button after specifying an address in [Set point:] specifies a breakpoint at that address. The specified breakpoint is displayed in the breakpoint display area.
- <Clear> button
Clicking the <Clear> button cancels the breakpoint set at the address which is input in [Set point:], or the breakpoint at the address selected in the breakpoint display area. The cancelled breakpoint disappears from the breakpoint display area.
- <All Set> button
Clicking the <All Set> button clears all breakpoints set in the E7000PC emulator at once and then sets breakpoints to the addresses shown in the breakpoint display area. Therefore, all breakpoints set by the BREAK command (emulator command) in the command area are cleared.
- <All Clear> button
Clicking the <All Clear> button clears all breakpoints set in the E7000PC emulator. Therefore, all breakpoints displayed in the breakpoint display area and all breakpoints set by the BREAK command (emulator command) in the command area are cleared.
- <Cancel> button
Clicking the <Cancel> button closes the BREAK dialog box.

Function

- Setting

A breakpoint is set at the address input in [Set point:]. A maximum of 255 breakpoints can be set in total. A BP mark is displayed on the corresponding source line.

A breakpoint can be set by clicking the <SET> button on the tool bar or by executing the BREAK command (emulator command) in the command area.

When setting a breakpoint with the <SET> button, select a source line with the mouse pointer and click the <SET> button. Breakpoints set in this way are displayed in the breakpoint display area. However, breakpoints set with the BREAK command (emulator command) are not displayed in the breakpoint display area.

- Cancellation

Breakpoints set at addresses specified in [Set point:] or those selected in the breakpoint display area are cancelled. The BP mark displayed on the source line corresponding to the cancelled breakpoint disappears.

A breakpoint can be cancelled by clicking the <CLEAR> button on the tool bar or by executing the BREAK command (emulator command) in the command area.

When cancelling a breakpoint with the <CLEAR> button, click the source line on which the BP mark is displayed to display the cursor, and then click the <CLEAR> button. The cancelled breakpoint disappears from the breakpoint display area.

Note

Breakpoints set by the [Break...] command become invalid during step execution using the <STEP> or the <STEP_OVER> button.

Related Functions

Source area, and <STEP>, <STEP_OVER>, <SET>, and <CLEAR> buttons

Emulator command: BREAK

5.4.4 Specifying Hardware Break Condition1

(Ctrl + 1)

[Break condition1...]

BREAK CONDITION1 dialog box

Overview

Specifies a hardware break condition (BREAK CONDITION1). When all of the specified conditions are satisfied during emulation started with the [Go...] command, program execution terminates.

Window

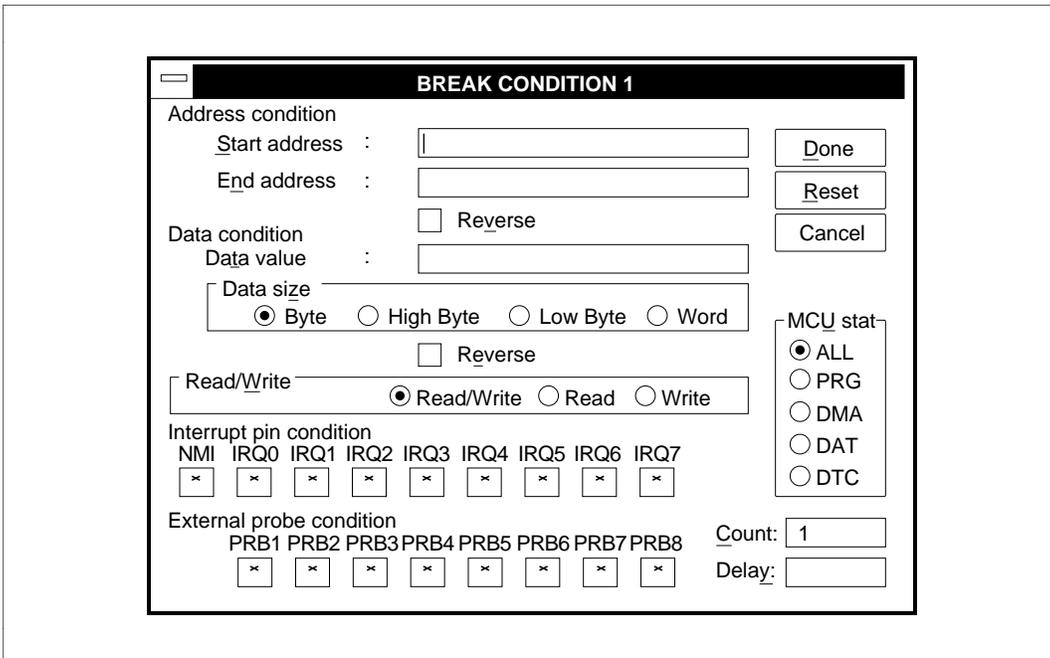


Figure 5-30 BREAK CONDITION1 Dialog Box

Operation

- [Address condition]
 - [Start address:] text box
Specifies an address bus value where a hardware break is to occur. An address bus value can be masked. Specifies the start address when an address range is specified as a condition.
 - [End address:] text box
Specifies the end address of an address range in which a hardware break is to occur.

- [Reverse] check box

Specifies whether the address bus condition is within the address range specified by [Start address:] and [End address:], or is outside this range, as follows:

 - <No check>: Addresses in the address range are hardware break conditions (default at system initiation)
 - <Check>: Addresses outside of the address range are hardware break conditions
- [Data condition]
 - [Data value:] text box

Specifies a data bus value where a hardware break is to occur. A data bus value can be masked.
 - [Data size] group box

Specifies the size of the data bus value specified in [Data value:] as follows:

 - <Byte>: 1-byte data (default at system initiation)
 - <High Byte>: High-order byte of 2-byte data
 - <Low Byte>: Low-order byte of 2-byte data
 - <Word>: 2-byte data
 - [Reverse] check box

Specifies whether the data bus condition is the value specified by [Data value:] or is another value, as follows:

 - <No check>: Value specified in [Data value:] is a data bus condition (default at system initiation)
 - <Check>: Value other than that specified in [Data value:] is a data bus condition
- [Read/Write] group box

Specifies a read or write cycle condition as a hardware break condition as follows:

 - <Read/Write>: Specifies both read and write cycles as hardware break conditions (default at system initiation)
 - <Read>: Specifies a read cycle as a hardware break condition
 - <Write>: Specifies a write cycle as a hardware break condition
- [MCU stat] group box

Specifies an H8S/2655-series bus cycle condition as a hardware break condition as follows:

 - <ALL>: Specifies all bus cycles as hardware break conditions (default at system initiation)
 - <PRG>: Specifies a program fetch cycle as a hardware break condition
 - <DMA>: Specifies a DMA cycle as a hardware break condition
 - <DAT>: Specifies an execution cycle as a hardware break condition
 - <DTC>: Specifies a DTC cycle as a hardware break condition

- [Interrupt pin condition]
Specifies an H8S/2655-series control signal state as a hardware break condition. A condition is satisfied when the specified value and the signal value are the same. Values can be specified for each tri-state button by clicking on the button desired.
 - [NMI] tri-state button
 - <*>: Specifies no condition (default at system initiation)
 - <H>: Specifies NMI signal high level as a hardware break condition
 - <L>: Specifies NMI signal low level as a hardware break condition
 - [IRQ0] to [IRQ7] tri-state button
 - <*>: Specifies no condition (default at system initiation)
 - <H>: Specifies IRQ0-IRQ7 signal high level as a hardware break condition
 - <L>: Specifies IRQ0-IRQ7 signal low level as a hardware break condition
- [External probe condition]
Specifies H8S/2655-series external probe conditions as hardware break conditions. A condition is satisfied when the specified value and the external probe signal value are the same. Values can be specified for each tri-state button by clicking on the button desired.
 - [PRB1] to [PRB8] tri-state button
 - <*>: Specifies no condition (default at system initiation)
 - <H>: Specifies external probe signal high level as a hardware break condition
 - <L>: Specifies external probe signal low level as a hardware break condition
- [Count:] text box
Specifies the number of hardware break conditions that must be encountered before a hardware break occurs. Program execution terminates when the above conditions are satisfied for a number of times equal to that in [Count:]. Data from H'0001 to H'1000 can be specified as [Count:]. The default at system initiation is 1.
- [Delay:] text box
Specifies a delay count starting from the time a hardware break condition is satisfied. When the above conditions have been satisfied, the program will be terminated after the number of bus cycles specified in [Delay:] has been executed. A delay count ranging from H'0001 to H'7FFF can be specified. If nothing is specified, 0 is set.
- <Done> button
Clicking the <Done> button sends the BREAK CONDITION1 dialog box settings to the E7000PC emulator, closes the BREAK CONDITION1 dialog box, and stores the above settings in the GUI-2600.

- <Reset> button
Clicking the <Reset> button cancels the settings in the BREAK CONDITION1 dialog box and the E7000PC emulator, returns the system to a state in which no hardware break conditions are specified, and closes the BREAK CONDITION1 dialog box.
- <Cancel> button
Clicking the <Cancel> button closes the BREAK CONDITION1 dialog box and returns the system to its state before the BREAK CONDITION1 dialog box was opened without storing the above settings.

Function

- Setting
Sets the following hardware break conditions: address ([Address condition]), data ([Data condition]), read/write ([Read/Write]), MCU status ([MCU stat]), external interrupt ([Interrupt pin condition]), external probe ([External probe condition]), number of times ([Count:]), and delay bus cycles ([Delay:]). By combining the conditions with each [Emulation mode:] of the [Go...] command, program emulation can be executed in each emulation mode.
- Specifying conditions for hardware breaks according to bus width
When the address bus value and data bus value match the address and data conditions, respectively, the hardware break condition is satisfied. Hardware break conditions must be specified according to the width of the memory bus accessed by the H8S/2655-series MCU. Table 5-6 shows the hardware break conditions for each access size.

Table 5-6 Hardware Break Condition Specifications According to Bus Width (Break Condition1)

Bus Width	Access Size	Hardware Break Condition Specification Method
16-bit bus area	Word access	Word data is accessed in one bus cycle. [Address condition] must be even addresses and [Data size] must be <Word>. Other specifications are invalid.
	Byte access	Byte data is accessed in one bus cycle. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.
8-bit bus area	Word access	Word data is accessed in two bus cycles. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.
	Byte access	Byte data is accessed in one bus cycle. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.

Note: <Byte>, <High Byte>, or <Low Byte> can be selected as byte data. Note the following to select byte data.

<Byte>: When an even address is specified, the data bus from D15 to D8 is selected, and when an odd address is specified, the data bus from D7 to D0 is selected. When no address is specified or an address range or mask is specified, data bus D15 to D8 is selected.

<High Byte>: When an even address (D15 to D8) is accessed in byte units, the condition is satisfied. The condition cannot be satisfied when an odd address is specified.

<Low Byte>: When an odd address (D7 to D0) is accessed in byte units, the condition is satisfied. The condition cannot be satisfied when an even address is specified.

- Mask specifications

A mask in 1-bit or 4-bit units can be specified for the settings in [Address condition] and [Data condition]. The meaning of a masked bit is that the corresponding bit value is arbitrary when determining whether the condition is satisfied.

— Mask specification for [Address condition]

To implement a mask for [Address condition], specify each digit of the value specified in [Start address:] to be masked at input as an asterisk (*). Note that the mask must be specified for consecutive digits from the lowest digit, and therefore, arbitrary bit positions cannot be masked.

For example, if an address condition is to be satisfied when A31 to A16 bits are 0, specify the mask as H'0000****.

Masking cannot be performed when specifying an address range with [End address:].

— Mask specification for [Data condition]

To implement a mask for [Data condition], specify each digit of the value specified in [Data value:] to be masked at input as an asterisk (*).

For example, if a byte data condition is to be satisfied when the D0 bit is 0, specify the mask as B'*****0.

Related Function

GUI command: [Go...]

5.4.5 Specifying Hardware Break Condition2,3,4

(Ctrl + 2)

(Ctrl + 3) (Ctrl + 4)

[Break Condition2...]

BREAK CONDITION2 dialog box

[Break Condition3...]

BREAK CONDITION3 dialog box

[Break Condition4...]

BREAK CONDITION4 dialog box

Overview

Specifies a hardware break condition (BREAK CONDITION2). When all of the specified conditions are satisfied during emulation started with the [Go...] command, program execution terminates.

BREAK CONDITION3 and BREAK CONDITION4 are specified in the same way.

Window

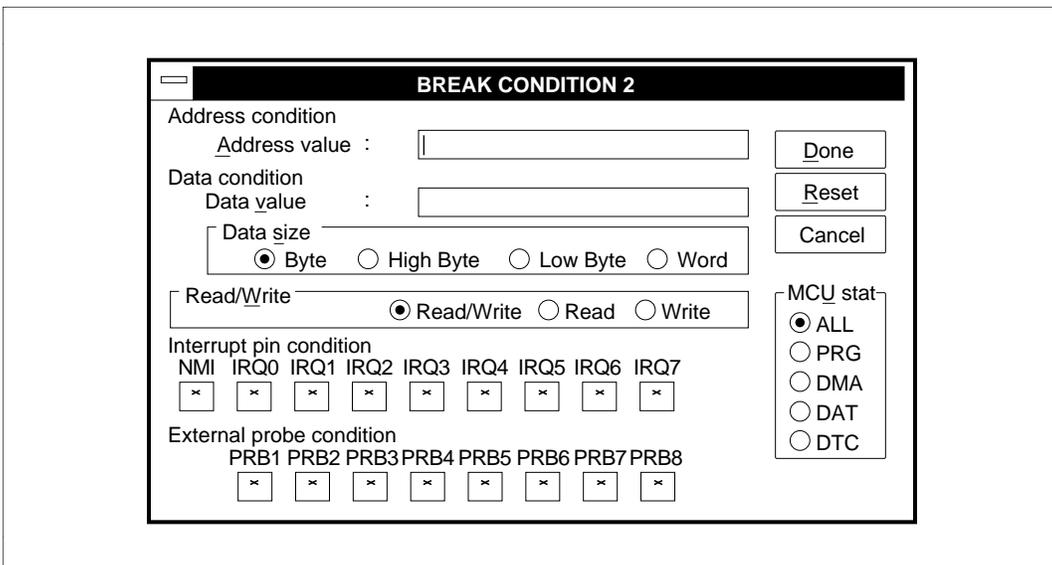


Figure 5-31 BREAK CONDITION2 Dialog Box

Operation

- [Address condition]
[Address value:] text box
Specifies an address bus value where a hardware break is to occur.

- [Data condition]
 - [Data value:] text box

Specifies a data bus value where a hardware break is to occur. A data bus value can be masked.
 - [Data size] group box

Specifies the size of the data bus value specified in [Data value:] as follows:

 - <Byte>: 1-byte data (default at system initiation)
 - <High Byte>: High-order byte of 2-byte data
 - <Low Byte>: Low-order byte of 2-byte data
 - <Word>: 2-byte data

- [Read/Write] group box

Specifies a read or write cycle condition as a hardware break condition as follows:

 - <Read/Write>: Specifies both read and write cycles as hardware break conditions (default at system initiation)
 - <Read>: Specifies a read cycle as a hardware break condition
 - <Write>: Specifies a write cycle as a hardware break condition

- [MCU stat] group box

Specifies an H8S/2655-series bus cycle condition as a hardware break condition as follows.

 - <ALL>: Specifies all bus cycles as hardware break conditions (default at system initiation)
 - <PRG>: Specifies a program fetch cycle as a hardware break condition
 - <DMA>: Specifies a DMA cycle as a hardware break condition
 - <DAT>: Specifies an execution cycle as a hardware break condition
 - <DTC>: Specifies a DTC cycle as a hardware break condition

- [Interrupt pin condition]

Specifies an H8S/2655-series control signal state as a hardware break condition. A condition is satisfied when the specified value and the signal value are the same. Values can be specified for each tri-state button by clicking on the button desired.

 - [NMI] tri-state button
 - <*>: Specifies no condition (default at system initiation)
 - <H>: Specifies NMI signal high level as a hardware break condition
 - <L>: Specifies NMI signal low level as a hardware break condition

— [IRQ0] to [IRQ7] tri-state button

<*>: Specifies no condition (default at system initiation)

<H>: Specifies IRQ0-IRQ7 signal high level as a hardware break condition

<L>: Specifies IRQ0-IRQ7 signal low level as a hardware break condition

- [External probe condition]

Specifies H8S/2655-series external probe conditions as hardware break conditions. A condition is satisfied when the specified value and the external probe signal value are the same. Values can be specified for each tri-state button by clicking on the button desired.

[PRB1] to [PRB8] tri-state button

<*>: Specifies no condition (default at system initiation)

<H>: Specifies external probe signal high level as a hardware break condition

<L>: Specifies external probe signal low level as a hardware break condition

- <Done> button

Clicking the <Done> button sends the BREAK CONDITION2 dialog box settings to the E7000PC emulator, closes the BREAK CONDITION2 dialog box, and stores the above settings in the GUI-2600.

- <Reset> button

Clicking the <Reset> button cancels the settings in the BREAK CONDITION2 dialog box and the E7000PC emulator, returns the system to a state in which no hardware break conditions are specified, and closes the BREAK CONDITION2 dialog box.

- <Cancel> button

Clicking the <Cancel> button closes the BREAK CONDITION2 dialog box and returns the system to its state before the BREAK CONDITION2 dialog box was opened without storing the above settings.

Function

- Setting

Sets the following hardware break conditions: address ([Address condition]), data ([Data condition]), read/write ([Read/Write]), MCU status ([MCU stat]), external interrupt ([Interrupt pin condition]), and external probe ([External probe condition]). By combining the conditions with each [Emulation mode:] of the [Go...] command, program emulation can be executed in each emulation mode.

- Specifying conditions for hardware breaks according to bus width
When the address bus value and data bus value match the address and data conditions, respectively, the hardware break condition is satisfied. Hardware break conditions must be specified according to the width of the memory bus accessed by the H8S/2655-series MCU. Table 5-7 shows the hardware break conditions for each access size.

Table 5-7 Hardware Break Condition Specifications According to Bus Width (Break Condition 2,3,4)

Bus Width	Access Size	Hardware Break Condition Specification Method
16-bit bus area	Word access	Word data is accessed in one bus cycle. [Address condition] must be even addresses and [Data size] must be <Word>. Other specifications are invalid.
	Byte access	Byte data is accessed in one bus cycle. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.
8-bit bus area	Word access	Word data is accessed in two bus cycles. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.
	Byte access	Byte data is accessed in one bus cycle. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.

Note: <Byte>, <High Byte>, or <Low Byte> can be selected as byte data. Note the following to select byte data.

<Byte>: When an even address is specified, the data bus from D15 to D8 is selected, and when an odd address is specified, the data bus from D7 to D0 is selected. When no address is specified or an address range or mask is specified, data bus D15 to D8 is selected.

<High Byte>: When an even address (D15 to D8) is accessed in byte units, the condition is satisfied. The condition cannot be satisfied when an odd address is specified.

<Low Byte>: When an odd address (D7 to D0) is accessed in byte units, the condition is satisfied. The condition cannot be satisfied when an even address is specified.

- Mask specifications
A mask in 4-bit units can be specified for the [Data condition] settings. The meaning of a masked bit is that the corresponding bit value is arbitrary when determining whether the condition is satisfied.

To implement a mask for [Data condition], specify each digit of the value specified in [Data value:] to be masked at input as an asterisk (*).

For example, if a word data condition is to be satisfied when the low-order byte is 0, specify the mask as H'**00.

Related Function

GUI command: [Go...]

5.4.6 Specifying Hardware Break Condition5,6

(Ctrl + 5), (Ctrl + 6)

[Break Condition5...]

BREAK CONDITION5 dialog box

[Break Condition6...]

BREAK CONDITION6 dialog box

Overview

Specifies a hardware break condition (BREAK CONDITION5). When all of the specified conditions are satisfied during emulation started with the [Go...] command, program execution terminates.

BREAK CONDITION6 is specified in the same way.

Window

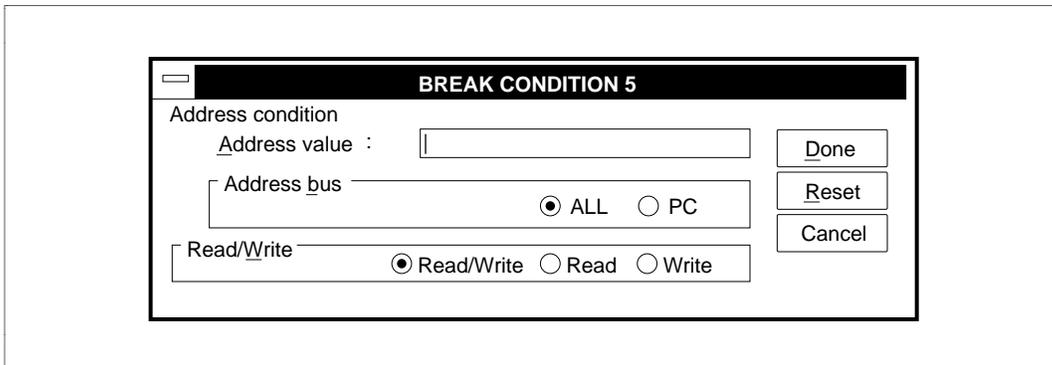


Figure 5-32 BREAK CONDITION5 Dialog Box

Operation

- [Address condition]
 - [Address value:] text box
Specifies an address bus value where a hardware break is to occur.
 - [Address bus] group box
Specifies an address bus condition as a hardware break condition as follows:
 - <ALL>: Specifies when the address bus value matches the specified value as a hardware break condition (default at system initiation)
 - <PC>: Specifies when the instruction at the specified address is executed as a hardware break condition

- [Read/Write] group box
Specifies a read or write cycle condition as a hardware break condition as follows:
 - <Read/Write>: Specifies both read and write cycles as hardware break conditions (default at system initiation)
 - <Read>: Specifies a read cycle as a hardware break condition
 - <Write>: Specifies a write cycle as a hardware break condition
- <Done> button
Clicking the <Done> button sends the BREAK CONDITION5 dialog box settings to the E7000PC emulator, closes the BREAK CONDITION5 dialog box, and stores the above settings in the GUI-2600.
- <Reset> button
Clicking the <Reset> button cancels the settings in the BREAK CONDITION5 dialog box and the E7000PC emulator, returns the system to a state in which no hardware break conditions are specified, and closes the BREAK CONDITION5 dialog box.
- <Cancel> button
Clicking the <Cancel> button closes the BREAK CONDITION5 dialog box and returns the system to its state before the BREAK CONDITION5 dialog box was opened without storing the above settings.

Function

- Setting
Sets the following hardware break conditions: address ([Address condition]) and read/write ([Read/Write]). By combining the conditions with each [Emulation mode:] of the [Go...] command, program emulation can be executed in each emulation mode.
- Specifying conditions for hardware breaks according to bus width
When the address bus value matches the address condition, the hardware break condition is satisfied. Hardware break conditions must be specified according to the width of the memory bus accessed by the H8S/2655-series MCU. Table 5-8 shows the hardware break conditions for each access size.

Table 5-8 Hardware Break Condition Specifications According to Bus Width (Break Condition5,6)

Bus Width	Access Size	Hardware Break Condition Specification Method
16-bit bus area	Word access	Word data is accessed in one bus cycle. [Address condition] must be even addresses. Other specifications are invalid.
	Byte access	Byte data is accessed in one bus cycle. [Address condition] can be an even or odd address.
8-bit bus area	Word access	Word data is accessed in two bus cycles. [Address condition] can be an even or odd address.
	Byte access	Byte data is accessed in one bus cycle. [Address condition] can be an even or odd address.

Note

When <PC> is selected in the [Address bus] group box of [Address condition], the hardware break condition specified in the [Read/Write] group box is ignored.

Related Function

GUI command: [Go...]

5.5 Trace Functions

5.5.1 Specifying Trace Acquisition Mode

(Alt + Ctrl + T)

[Trace mode...]

TRACE MODE dialog box

Overview

Specifies whether or not to display refresh cycles during [Trace display - Text] execution, and sets the memory addresses in which trace information is acquired.

Window

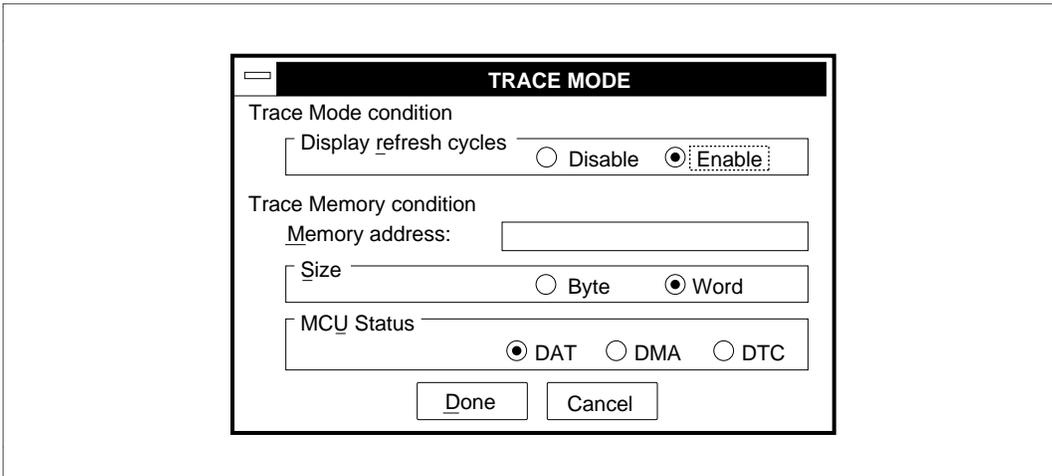


Figure 5-33 TRACE MODE Dialog Box

Operation

- [Trace Mode condition]

[Display refresh cycles] group box

Specifies whether or not to display refresh cycles at [Trace display - Text] command bus cycle display.

<Disable>: Does not display refresh cycles

<Enable>: Displays refresh cycles (default at system initiation)

- [Trace Memory condition]
 - [Memory address:] text box

Specifies the value or symbol for a memory address whose contents are to be traced.

The memory contents of the specified address can be displayed in the emulation information area of the base window during emulation, depending on the [Display memory trace data] specification of the [Go...] command.

Memory contents during emulation can also be displayed in the trace information display by the [Trace display - Text] command.
 - [Size] group box

Specifies the size of the memory data to be traced as follows:

 - <Byte>: 1-byte data
 - <Word>: 2-byte data (default at system initiation)
 - [MCU status] group box

Specifies an H8S/2655-series bus cycle condition as a trace acquisition condition as follows:

 - <DAT>: Specifies an execution cycle as a trace acquisition condition (default at system initiation)
 - <DMA>: Specifies a DMA cycle as a trace acquisition condition
 - <DTC>: Specifies a DTC cycle as a trace acquisition condition
- <Done> button

Clicking the <Done> button sends the TRACE MODE dialog box settings to the E7000PC emulator, closes the TRACE MODE dialog box, and stores the above settings in the GUI-2600.
- <Cancel> button

Clicking the <Cancel> button closes the TRACE MODE dialog box and returns the system to its state before the TRACE MODE dialog box was opened without storing the above settings.

Function

Sets memory conditions whose contents are to be traced.

Memory contents being traced can be displayed in realtime in the emulation information area by selecting <Enable> in the [Display memory trace data] group box in the [Go...] command.

Notes

- (1) This command cannot be set during emulation.
- (2) For memory to be modified within the CPU such as timer counters of the internal I/O modules, the trace memory contents will not be modified until it is accessed by an execution, DMA, or DTC cycle.

Related Functions

GUI commands: [Go...] and [Trace display - Text]

5.5.2 Specifying Trace Acquisition Conditions

(Ctrl + T)

[Trace condition...]

TRACE CONDITION dialog box

Overview

Specifies trace acquisition conditions. Trace information is acquired during emulation execution according to these specifications.

Window

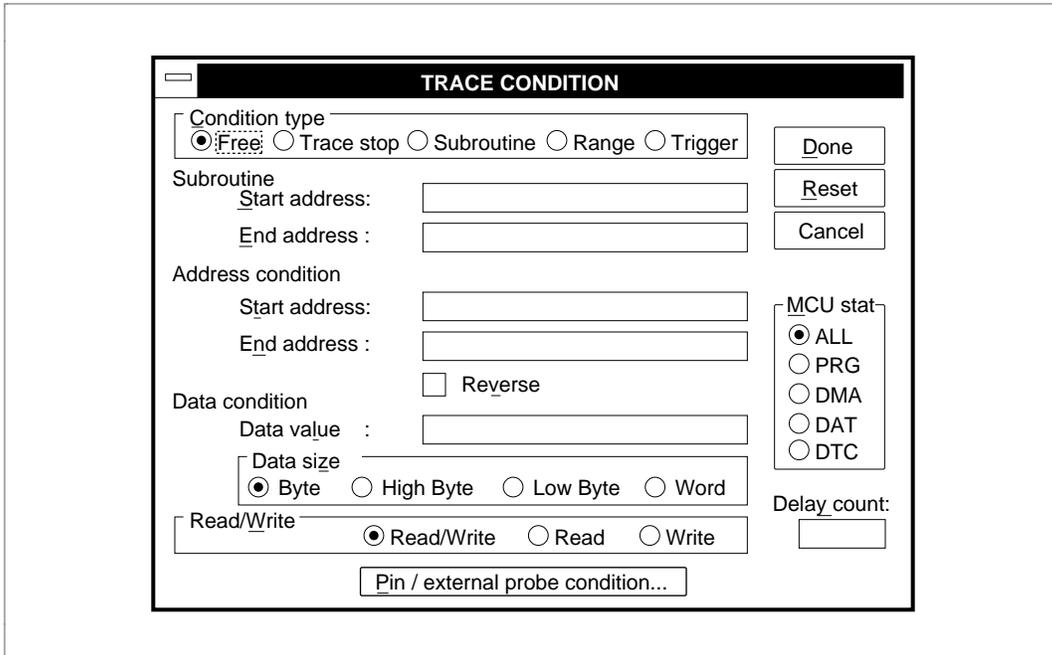


Figure 5-34 TRACE CONDITION Dialog Box

Operation

- [Condition type] group box

Specifies how to acquire trace information during emulation.

<Free>: Acquires trace information during all bus cycles. Trace acquisition conditions are ignored. (Default at system initiation)

<Trace stop>: Acquires trace information during all bus cycles until the specified trace stop condition is satisfied; program execution, however, continues. A trace stop condition is determined by ANDing the following: [Address condition], [Data condition], [Read/Write], [MCU stat], [Delay count:], [Pin condition], and [External probe condition].

<Subroutine>: Acquires trace information for instructions and operand accesses in the range (subroutine) specified by a start address and an end address. A trace acquisition condition is specified by [Subroutine].

<Range>: Acquires trace information during a bus cycle corresponding with the trace acquisition condition. A trace acquisition condition is determined by ANDing [Address condition] and [Read/Write].

<Trigger>: Outputs a low-level pulse from the trigger output pin of the emulator pod during a bus cycle corresponding with the trace acquisition condition. Trace information is acquired during all bus cycles. A trace acquisition condition is determined by ANDing the following: [Address condition], [Data condition], [Read/Write], [MCU stat], [Delay count:], [Pin condition], and [External probe condition].

- [Subroutine]

Specifies an address range determining the subroutine in which trace information is to be acquired, for selecting <Subroutine> in [Condition type].

— [Start address:] text box

Specifies a number or symbol as the start address of the subroutine in which trace information is to be acquired.

— [End address:] text box

Specifies a number or symbol as the end address of the subroutine in which trace information is to be acquired.

If the source program is written in C, the end address can be omitted because the end address of the function is automatically set.

- [Address condition]

Specifies an address bus condition as a trace acquisition condition, for selecting <Trace stop>, <Range>, or <Trigger> in [Condition type].

— [Start address:] text box

Specifies a number or a symbol as an address bus value for trace information acquisition. A mask can be specified for the address bus value. For acquiring trace information in an address range, specify the start address.

— [End address:] text box

Specifies a number or a symbol as the end address of the range in which trace information is to be acquired.

— [Reverse] check box

Specifies whether the address bus condition is within the address range determined by [Start address:] and [End address:], or is outside this range, as follows. This setting is valid only when <Range> is selected in [Condition type].

<No check>: Addresses in the address range are trace acquisition conditions
(default at system initiation)

<Check>: Addresses outside the address range are trace acquisition conditions

- [Data condition] text box

Specifies a data bus condition as a trace acquisition condition, for selecting <Trace stop> or <Trigger> in [Condition type].

- [Data value:] text box

Specifies a number or a symbol as a data bus value for trace information acquisition. A mask can be specified for the data bus value.

- [Data size] group box

Specifies the size of the data bus value set in [Data value:] as follows:

<Byte>: 1-byte data (default at system initiation)

<High Byte>: High-order byte of 2-byte data

<Low Byte>: Low-order byte of 2-byte data

<Word>: 2-byte data

- [Read/Write] group box

Specifies a read or write cycle condition as a trace acquisition condition, for selecting <Trace stop>, <Range>, or <Trigger> in [Condition type] as follows:

<Read/Write>: Specifies both read and write cycles as trace acquisition conditions
(default at system initiation)

<Read>: Specifies a read cycle as a trace acquisition condition

<Write>: Specifies a write cycle as a trace acquisition condition

- [MCU stat] group box

Specifies an H8S/2655-series bus cycle condition as a trace acquisition condition, for selecting <Trace stop> or <Trigger> in [Condition type] as follows:

<ALL>: Specifies all bus cycles as bus cycle conditions (default at system initiation)

<PRG>: Specifies a program fetch cycle as a bus cycle condition

<DMA>: Specifies a DMA cycle as a bus cycle condition

<DAT>: Specifies an execution cycle as a bus cycle condition

<DTC>: Specifies a DTC cycle as a bus cycle condition

- [Delay count:] text box
For selecting <Trace stop> or <Trigger> in [Condition type], specifies a delay count starting from the time a trace acquisition condition determined by ANDing [Address condition], [Data condition], [Read/Write], [MCU stat], [Pin condition], and [External probe condition] is satisfied. A delay count ranging from H'1 to H'7FFF can be specified. If omitted, 0 is assumed.
- <Done> button
Clicking the <Done> button stores the TRACE CONDITION and PIN/EXTERNAL PROBE CONDITION dialog box settings in the GUI-2600, closes the TRACE CONDITION dialog box, and sends the above settings to the E7000PC emulator.
- <Reset> button
Clicking the <Reset> button cancels the settings in the TRACE CONDITION and PIN/EXTERNAL PROBE CONDITION dialog boxes, and the E7000PC emulator settings, returns the system to a state in which no trace acquisition conditions are specified, and closes the TRACE CONDITION dialog box.
- <Cancel> button
Clicking the <Cancel> button closes the TRACE CONDITION dialog box and returns the system to its state when the TRACE CONDITION dialog box was opened without storing the settings in the TRACE CONDITION and PIN/EXTERNAL PROBE CONDITION dialog boxes.
- <Pin/external probe condition...> button
Clicking the <Pin/external probe condition...> button opens the dialog box in which H8S/2655-series control signal states and external probe (in the emulator pod) conditions are specified as trace acquisition conditions.

The specifications are valid only when <Trace stop> or <Trigger> is set in [Condition type]. The dialog box contents are shown in figure 5-35.

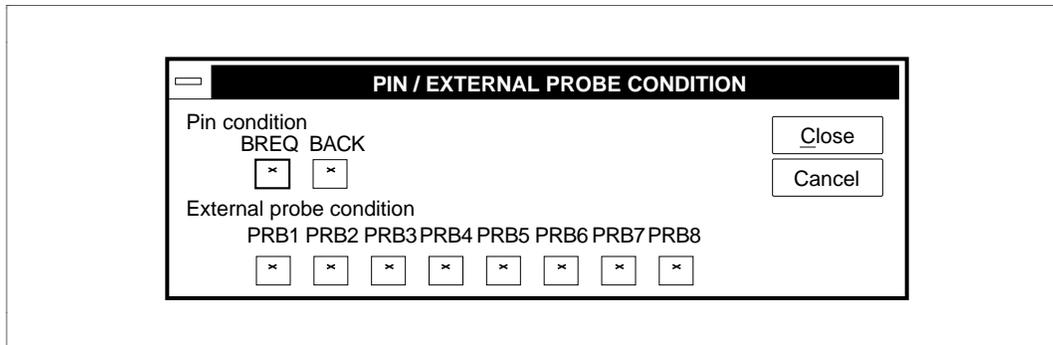


Figure 5-35 PIN/EXTERNAL PROBE CONDITION Dialog Box

- [Pin condition]
Specifies an H8S/2655-series control signal state as a trace acquisition condition. A condition is satisfied when the specified value and the signal value are the same.

Values can be specified for each state button by clicking the button desired.

— [BREQ] state button

<*>: Specifies no condition (default at system initiation)

<L>: Specifies BREQ signal low level as a trace acquisition condition

— [BACK] state button

<*>: Specifies no condition (default at system initiation)

<L>: Specifies BACK signal low level as a trace acquisition condition

- [External probe condition]
Specifies external probe (in the emulator pod) conditions as trace acquisition conditions. A condition is satisfied when the specified value and the external probe signal value are the same. Values can be specified for each tri-state button by clicking the button desired.

[PRB1] to [PRB8] tri-state button

<*>: Specifies no condition (default at system initiation)

<H>: Specifies external probe signal high level as a trace acquisition condition

<L>: Specifies external probe signal low level as a trace acquisition condition

- <Close> button
Clicking the <Close> button closes the PIN/EXTERNAL PROBE CONDITION dialog box.

- <Cancel> button (in PIN/EXTERNAL PROBE CONDITION dialog box)
Clicking the <Cancel> button closes the PIN/EXTERNAL PROBE CONDITION dialog box and returns the system to its state when the PIN/EXTERNAL PROBE CONDITION dialog box was opened without storing the settings in the PIN/EXTERNAL PROBE CONDITION dialog box.

Function

- Trace acquisition condition setting
Sets the conditions for acquiring trace information during program execution initiated by the [Go...] command or the <CONTINUE> button.

The trace acquisition conditions depend on the trace modes listed in table 5-9.

Table 5-9 Trace Modes

Trace Mode	Description
<Free> Free trace	Acquires trace information during all bus cycles with no specified conditions.
<Trace stop> Trace stop	Stops trace acquisition when the specified condition is satisfied, enters command input wait state in parallel mode, and displays a prompt (#) in the command area.
<Subroutine> Subroutine trace	Acquires trace information for instructions and operand accesses in the range (subroutine) determined by a start address and an end address. If the specified subroutine calls another subroutine, trace information during the called subroutine is not acquired.
<Range> Range trace	Acquires trace information during a bus cycle corresponding with the specified address range and condition.
<Trigger> Trigger	Outputs a low-level pulse from the trigger output pin in the emulator pod during a bus cycle corresponding with the specified condition. Acquires trace information during all bus cycles.

- Specifying trace acquisition conditions according to memory bus width
Trace acquisition conditions for <Trace stop> and <Trigger> are specified according to the width of the memory bus accessed by the H8S/2655-series MCU. Table 5-10 shows the trace acquisition conditions for each access size.

Table 5-10 Trace Acquisition Condition Settings According to Bus Width

Bus Width	Access Size	Trace Acquisition Condition Specification Method
16-bit bus area	Word access	Word data is accessed in one bus cycle. [Address condition] must be even addresses and [Data size] must be <Word>. Other specifications are invalid. However, when only [Address condition] is specified, odd addresses are also valid.
	Byte access	Byte data is accessed in one bus cycle. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.
8-bit bus area	Word access	Word data is accessed in two bus cycles. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.
	Byte access	Byte data is accessed in one bus cycle. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.

Note: <Byte>, <High Byte>, or <Low Byte> can be selected as byte data. Note the following to select byte data.

<Byte>: When an even address is specified, the data bus from D15 to D8 is selected, and when an odd address is specified, the data bus from D7 to D0 is selected. When no address is specified or an address range or mask is specified, data bus D15 to D8 is selected.

<High Byte>: When an even address (D15 to D8) is accessed in byte units, the condition is satisfied. The condition cannot be satisfied when an odd address is specified.

<Low Byte>: When an odd address (D7 to D0) is accessed in byte units, the condition is satisfied. The condition cannot be satisfied when an even address is specified.

- Mask specifications

A mask in 1-bit or 4-bit units can be specified for the settings in [Address condition] and [Data condition]. The meaning of a masked bit is that the corresponding bit value is arbitrary when determining whether the condition is satisfied.

- Mask specification for [Address condition]

When <Trace stop> or <Trigger> is specified in [Condition type], specify each digit in [Start address:] to be masked at input as an asterisk (*).

When <Range> is specified in [Condition type], specify each digit in [Start address:] to be masked at input as an asterisk (*). Note that the mask must be specified for consecutive digits from the lowest digit, and therefore, arbitrary bit positions cannot be masked.

For example, if an address range condition is to be satisfied when A16 to A31 bits are 0, specify the mask as H'0000****.

Masking cannot be performed when an address range for trace acquisition is specified in [End address:].

- Mask specification for [Data condition]

To implement a mask for [Data condition] when <Trace stop> or <Trigger> is specified in [Condition type], specify each digit of the value specified in [Data value:] to be masked at input as an asterisk (*).

For example, if a byte data condition is to be satisfied when the D0 bit is 0, specify the mask as B'*****0.

Note

If parallel mode is entered in the command area during emulation, trace information is acquired in the following condition:

- (a) If parallel mode is entered by pressing the (Enter) key or by the satisfaction of a trace acquisition condition (<Trace stop>), trace information will not be acquired in parallel mode. When the E7000PC emulator exits parallel mode, the acquired trace information is cleared and trace acquisition restarts under the conditions specified with this command.
- (b) If parallel mode is entered by pressing the (Space) key, trace information is acquired in parallel mode.

Related Function

GUI command: [Trace display - Setting...]

5.5.3 Specifying Trace Information Display Conditions

(Alt + Ctrl + X)

[Trace display - Setting...]

TRACE DISPLAY SETTING dialog box

Overview

Specifies trace information display conditions. Trace information displayed by the [Trace display - Text] or [Trace display - Graph] command follows these command specifications.

Window

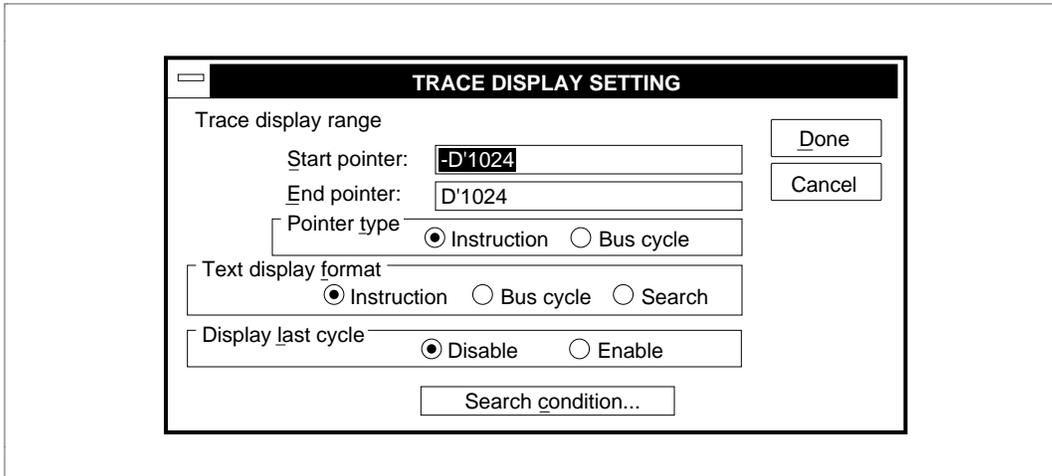


Figure 5-36 TRACE DISPLAY SETTING Dialog Box

Operation

- [Trace display range]
 - [Start pointer:] text box
Specifies a number as the start pointer for the range for which trace information is shown in text form or graph form. When omitted, the value -D'32767 is assumed; the initial value is -D'1024.
 - [End pointer:] text box
Specifies a number as the end pointer for the range for which trace information is shown in text form or graph form. When omitted, the value D'32767 is assumed; the initial value is D'1024.

- [Pointer type] group box

Specifies the pointer format in which trace information is displayed in the TRACE DISPLAY TEXT window as follows:

 - <Instruction>: Displays trace information in instruction pointer format
(default at system initiation)
 - <Bus cycle>: Displays trace information in bus cycle pointer format
- [Text display format] group box

Specifies the format in which trace information is displayed in the TRACE DISPLAY TEXT window. When <Search> is specified, <Bus cycle> is automatically selected in [Pointer type] in [Trace display range].

 - <Instruction>: Displays trace information in instruction mnemonic units
(default at system initiation)
 - <Bus cycle>: Displays trace information in bus cycle units
 - <Search>: Searches for and displays trace information in bus cycle units
- [Display last cycle] group box

Specifies whether trace information is to be displayed for only the last bus cycle or for all the bus cycles, when <Search> is selected in [Text display format], as follows:

 - <Disable>: Displays trace information for the whole range set in [Trace display range]
(default at system initiation)
 - <Enable>: Displays trace information only during the last bus cycle
- <Search condition...> button

Clicking the <Search condition...> button opens the SEARCH CONDITION dialog box in which trace information search conditions are set. These dialog box settings are valid when <Search> is specified in [Text display format]. The SEARCH CONDITION dialog box is shown in figure 5-37.
- <Done> button

Clicking the <Done> button closes the TRACE DISPLAY SETTING dialog box and stores the TRACE DISPLAY SETTING and SEARCH CONDITION dialog box settings.
- <Cancel> button (in TRACE DISPLAY SETTING dialog box)

Clicking the <Cancel> button closes the TRACE DISPLAY SETTING dialog box and returns the system to its state when the TRACE DISPLAY SETTING dialog box was opened without storing the settings in the TRACE DISPLAY SETTING and SEARCH CONDITION dialog boxes.

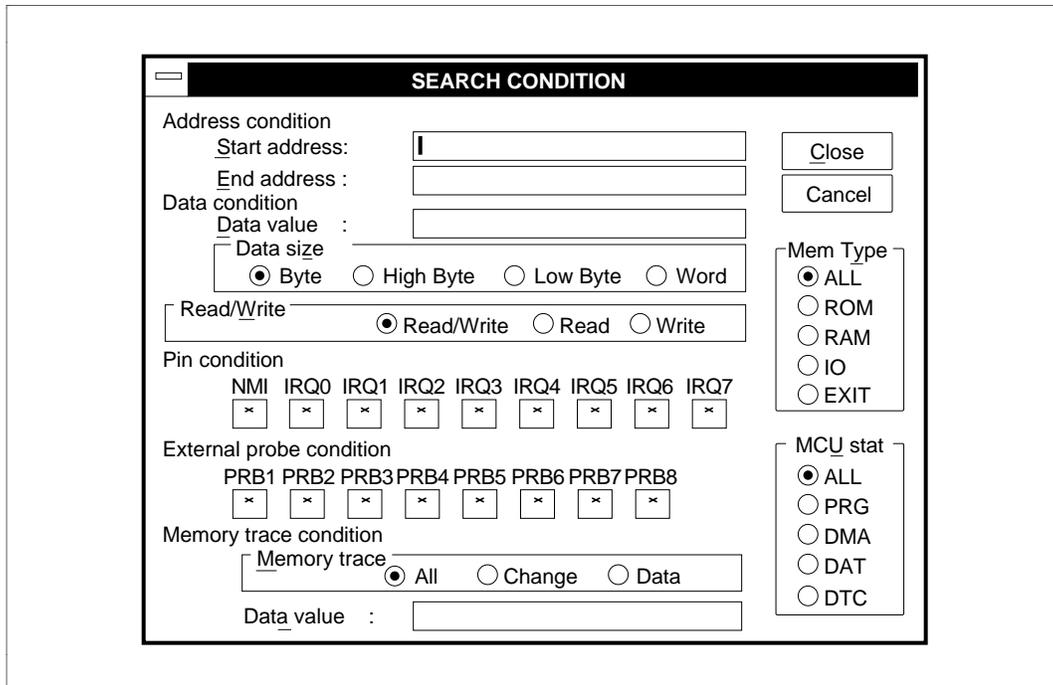


Figure 5-37 SEARCH CONDITION Dialog Box

- [Address condition]
 - Specifies an address bus condition as a trace search condition.
 - [Start address:] text box
 - Specifies a number or a symbol as an address bus value for trace information search. A mask can be specified for the address bus value. For searching trace information in an address range, specifies the start address.
 - [End address:] text box
 - Specifies a number or a symbol as the end address of the range in which trace information is to be searched.

- [Data condition]

Specifies a data bus condition as a trace search condition.

 - [Data value:] text box

Specifies a number or a symbol as a data bus value for trace information search. A mask can be specified for the data bus value.
 - [Data size] group box

Specifies the size of the data bus value set in [Data value:] as follows:

 - <Byte>: 1-byte data (default at system initiation)
 - <High Byte>: High-order byte of 2-byte data
 - <Low Byte>: Low-order byte of 2-byte data
 - <Word>: 2-byte data

- [Read/Write] group box

Specifies a read or write cycle condition as a trace search condition as follows:

 - <Read/Write>: Specifies both read and write cycles as trace search conditions (default at system initiation)
 - <Read>: Specifies a read cycle as a trace search condition
 - <Write>: Specifies a write cycle as a trace search condition

- [Mem Type] group box

Specifies an H8S/2655-series memory type condition as a trace search condition as follows:

 - <ALL>: Specifies all memory types as trace search conditions (default at system initiation)
 - <ROM>: Specifies an internal ROM area access as a trace search condition
 - <RAM>: Specifies an internal RAM area access as a trace search condition
 - <IO>: Specifies an internal I/O area access as a trace search condition
 - <EXT>: Specifies an external memory area access as a trace search condition

- [MCU stat] group box

Specifies an H8S/2655-series bus cycle condition as a trace search condition as follows:

 - <ALL>: Specifies all bus cycles as trace search conditions (default at system initiation)
 - <PRG>: Specifies a program fetch cycle as a trace search condition
 - <DMA>: Specifies a DMA cycle as a trace search condition
 - <DAT>: Specifies an execution cycle as a trace search condition
 - <DTC>: Specifies a DTC cycle as a trace search condition

- [Pin condition]

Specifies an H8S/2655-series control signal state as a trace search condition. A condition is satisfied when the specified value and the signal value are the same.

Values can be specified for each tri-state button by clicking the button desired.

 - [NMI] tri-state button
 - <*>: Specifies no condition (default at system initiation)
 - <H>: Specifies NMI signal high level as a trace search condition
 - <L>: Specifies NMI signal low level as a trace search condition
 - [IRQ0] to [IRQ7] tri-state buttons
 - <*>: Specifies no condition (default at system initiation)
 - <H>: Specifies IRQ0-IRQ7 signal high level as a trace search condition
 - <L>: Specifies IRQ0-IRQ7 signal low level as a trace search condition

- [External probe condition]

Specifies external probe (in the emulator pod) conditions as trace search conditions. A condition is satisfied when the specified value and the external probe signal value are the same.

Values can be specified for each tri-state button by clicking the button desired.

[PRB1] to [PRB8] tri-state button

 - <*>: Specifies no condition (default at system initiation)
 - <H>: Specifies external probe signal high level as a trace search condition
 - <L>: Specifies external probe signal low level as a trace search condition

- [Memory trace condition]

Specifies a trace search condition for traced memory set by [Trace Memory condition] of the [Trace mode...] command.

 - [Memory trace] group box

Specifies a trace search condition for to traced memory contents.

 - <All>: Searches for all trace information for a bus cycle where the specified memory address is accessed (default at system initiation)
 - <Change>: Searches for a bus cycle where traced memory has been modified
 - <Data>: Searches for a bus cycle where data specified by [Data value:] is accessed
 - [Data value:] text box

Specifies data for a trace search condition when <Data> is specified in the [Memory trace] group box. Immediately after system initiation, nothing is specified.

- <Close> button
Clicking the <Close> button stores the above settings and closes the SEARCH CONDITION dialog box.
- <Cancel> button (in SEARCH CONDITION dialog box)
Clicking the <Cancel> button closes the SEARCH CONDITION dialog box and returns the system to its state when the SEARCH CONDITION dialog box was opened without storing the settings in the SEARCH CONDITION dialog box.

Function

- Trace display condition setting
Sets the conditions for displaying trace information. The [Trace display - Text] or [Trace display - Graph] command displays trace information according to these command settings.
- Specifying conditions for trace search according to bus width
Trace search conditions in the SEARCH CONDITION dialog box are specified according to the width of the memory bus accessed by the H8S/2655-series MCU. Table 5-11 shows the trace search conditions for each access size.
- Mask specifications
A mask in 1-bit or 4-bit units can be specified for the settings in [Address condition] and [Data condition]. The meaning of a masked bit is that the corresponding bit value is arbitrary when determining whether the condition is satisfied.
 - Mask specification for [Address condition]
To implement a mask for [Address condition], specify each digit in [Start address:] to be masked at input as an asterisk (*).
Masking cannot be performed when specifying an address range with [End address:].
 - Mask specification for [Data condition]
To implement a mask for [Data condition], specify each digit of the value specified in [Data value:] to be masked at input as an asterisk (*).
For example, if a byte data condition is to be satisfied when the D0 bit is 0, specify the mask as B'*****0.

Table 5-11 Trace Search Condition Settings According to Bus Width

Bus Width	Access Size	Trace Search Condition Specification Method
16-bit bus area	Word access	Word data is accessed in one bus cycle. [Address condition] must be even addresses and [Data size] must be <Word>. Other specifications are invalid.
	Byte access	Byte data is accessed in one bus cycle. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.
8-bit bus area	Word access	Word data is accessed in two bus cycles. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.
	Byte access	Byte data is accessed in one bus cycle. [Data size] must be byte data*. Other specifications are invalid. [Address condition] can be an even or odd address.

Note: <Byte>, <High Byte>, or <Low Byte> can be selected as byte data. Note the following to select byte data.

<Byte>: When an even address is specified, the data bus from D15 to D8 is selected, and when an odd address is specified, the data bus from D7 to D0 is selected. When no address is specified or an address range or mask is specified, data bus D15 to D8 is selected.

<High Byte>: When an even address (D15 to D8) is accessed in byte units, the condition is satisfied. Data in an odd address cannot be searched for.

<Low Byte>: When an odd address (D7 to D0) is accessed in byte units, the condition is satisfied. Data in an even address cannot be searched for.

Related Functions

GUI commands: [Trace mode...], [Trace display - Text], and [Trace display - Graph]

5.5.4 Displaying Trace Information in Text

(Alt + X)

[Trace display - Text]

TRACE DISPLAY TEXT window

Overview

Displays trace information in text according to the specifications in the [Trace display - Setting...] command. The displayed contents can be output to a file.

Window

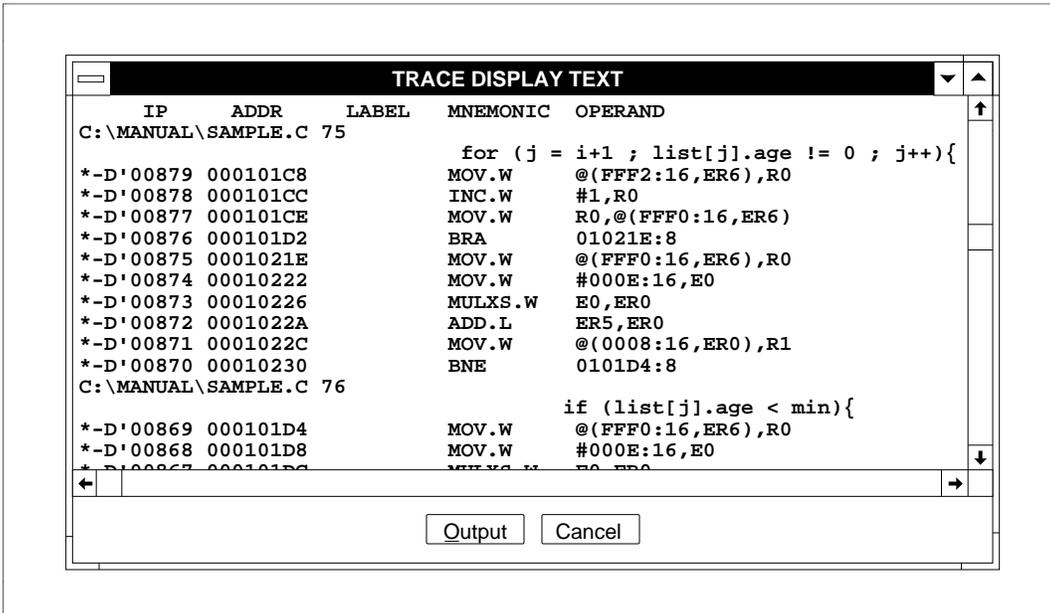


Figure 5-38 TRACE DISPLAY TEXT Window

Operation

- Display header
Indicates the trace information display format with a header.
- Trace display area
In the trace display area, trace information acquired during emulation is displayed. The contents of the source file of the corresponding program is included in the displayed trace information.

- <Output> button
Clicking the <Output> button opens the TRACE OUTPUT dialog box. The contents currently displayed in the trace display area can be output to a file.
- <Cancel> button
Clicking the <Cancel> button closes the TRACE DISPLAY TEXT window.

Function

- Display header format
The display header format depends on the [Text display format] group box specification in the [Trace display - Setting...] command.
— Trace information is displayed in instruction mnemonic units (<Instruction> must be specified in [Text display format])

	IP	ADDR	LABEL	MNEMONIC	OPERAND
*[-]D'	xxxxxx	xxxxxxxxxx	!xxxxxxxxxx	xxxxxxxxxx	xxxxx - xxxxx
	(a)	(b)	(c)	(d)	(e)

Figure 5-39 Display Header Format in Instruction Mnemonic Units (TRACE DISPLAY TEXT Window)

- (a) Instruction pointer (IP)
Relative instruction location (instruction pointer) based on the instruction where a delay condition is satisfied. The instruction pointer normally has a negative value (*-D'xxxxx). However, it will be positive if it is acquired during delay cycles while the delay condition is specified.
The point where a delay condition is satisfied means the starting point of a delay bus cycle which has been specified with [Delay:] in the [Break condition1...] command or [Delay count:] in the [Trace condition...] command. When no value is specified or when program execution terminates due to a different cause, the latest trace information will be shown from D'00000.
- (b) Instruction address (ADDR)
- (c) Label name (LABEL)
- (d) Instruction mnemonic (MNEMONIC)
- (e) Instruction operand (OPERAND)

— Trace information is displayed in bus-cycle units (<Bus cycle> or <Search> must be specified in [Text display format])

	BP	AB	DB	MA	R/W	ST	IRQ	NMI	RA	PROB	CLK	TM
[-]D'	xxxxx	xxxxxxxxxx	xxxx	xxx	x	xxx	xxxxxxxxxx	x	x	xxxxxxxxxx	xx	xxxxxxxxxx=xxxxx
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)

	TOTAL CLOCK NUMBER = xxxxxxxx											(m)

Figure 5-40 Display Header Format in Bus-Cycle Units (TRACE DISPLAY TEXT Window)

(a) Bus cycle pointer (BP)

Relative bus cycle location (bus cycle pointer) based on the bus cycle where a delay condition is satisfied. In bus cycles which prefetch instructions, the instruction mnemonics and instruction addresses are displayed together. When two instructions are executed in one bus cycle, both mnemonics are displayed along with the address of the first instruction. The bus cycle pointer normally has a negative (-D'xxxxx) value. However, it will be positive if it is acquired during delay cycles while the delay condition is specified.

(b) Address bus value (AB)

(c) Data bus value (DB)

2-byte or 1-byte values are displayed depending on the length of the data accessed by the MCU. When no data is on the data bus, ** is displayed.

(d) Memory area type (MA)

Accessed memory areas are displayed in one of the following types.

ROM: Internal ROM area access

RAM: Internal RAM area access

IO: Internal I/O area access

EXT: External memory area access

(e) Read/Write signal type (R/W)

R: Read cycle

W: Write cycle

(f) MCU status (ST)

PRG: Program fetch cycle

REF: Refresh cycle

DAT: Memory or I/O access cycle

DMA: DMA cycle

DTC: DTC cycle

(g) IRQ0-IRQ7 signal levels (IRQ)

The signal levels of IRQ0 to IRQ7 are displayed as follows:

IRQ

x7 x6 x5 x4 x3 x2 x1 x0 (IRQ7 to IRQ0 levels)

xn = 1: High level

= 0: Low level

(h) NMI signal level (NMI)

NMI = 1: High level

= 0: Low level

(i) BREQ and BACK signal levels (RA)

The signal levels of BREQ and BACK are displayed as follows:

RA

x1 x0 (x1: BACK signal, x0: BREQ signal)

xn = 1: High level

= 0: Low level

(j) External probe signal levels (PROB)

The signal levels of PRB1 to PRB8 are displayed as follows:

PROB

x7 x6 x5 x4 x3 x2 x1 x0 (PRB8 to PRB1 levels)

xn = 1: High level

= 0: Low level

(k) Clock cycle (CLK)

The number of clock cycles required from the end of the previous bus cycle to the end of this bus cycle. Up to 255 clock cycles can be displayed. If more than 255 clocks are counted, ** is displayed. Consider the trace acquisition timing of the E7000PC emulator station.

(l) Traced memory contents (TM)

Displays the address specified in [Memory address:] of the [Trace mode...] command and its traced contents.

(m) Total clock cycle (TOTAL CLOCK NUMBER)

The total number of clock cycles described in item (k). If the limit 255 is exceeded in even one display line, ***** is displayed. The total clock cycle is displayed only when <Bus cycle> is selected in [Text display format] of the [Trace display - Setting...] command.

- Source file display

Displays in the trace display area the source file contents corresponding to the address bus value in trace information. The display format depends on which language the corresponding program has been written in.

— C language program

```
<source file name (complete path name)>    <line number>  
<C source file contents>
```

— Assembly program

```
<source file name (complete path name)>
```

- Trace information output

The contents currently displayed in the trace display area can be output to a file. Clicking the <Output> button opens the TRACE OUTPUT dialog box shown in figure 5-41.

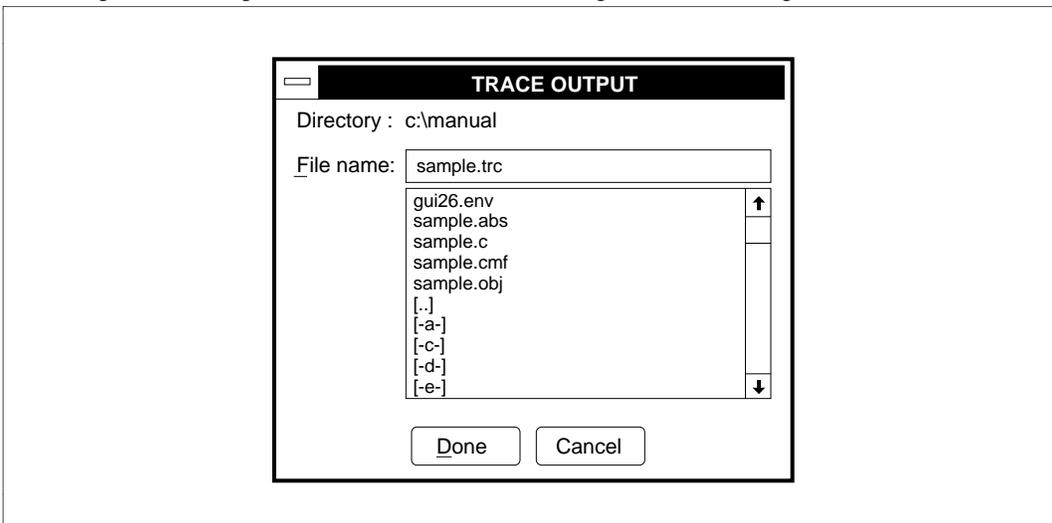


Figure 5-41 TRACE OUTPUT Dialog Box

— [File name:] text box

Specifies the name of the file where the trace information displayed in the trace display area is to be stored. A file with the specified file name is output to the current directory displayed in [Directory:].

Pressing the (Enter) key after inputting or selecting a file name or double-clicking the file name specified in the list box outputs the contents of the trace display area to a file.

The destination directory can be changed by specifying a new current directory in the sequence of drive name and directory name. To change the current directory, use the following procedure.

- (1) Select the target drive name from the list box by double-clicking. Drive names are displayed in the format [-a-]. After selected, the drive name and the current directory of the target drive are displayed in [Directory:]. The current directory contents of the selected drive are displayed in the list box.
- (2) Select a subdirectory from the list box by double-clicking. Subdirectory names are displayed in the format [directory]. After selected, the subdirectory name is added to [Directory:] and the subdirectory contents are displayed in the list box.
- (3) Continue selecting subdirectories until the directory where the trace information area is output is reached.

— [Directory:]

Displays the current directory. When the drive or subdirectory is changed in [File name:], this display also changes.

— <Done> button

Clicking the <Done> button starts storing displayed trace information to the file specified in [File name:] and closes the TRACE OUTPUT dialog box after storing the settings in the GUI-2600. When a file having the same name as that specified in [File name:] already exists, the message box shown in figure 5-42 appears. Click <Yes> to start overwriting the existing file specified in [File name:]. When <No> is clicked, the system returns to the TRACE OUTPUT dialog box.

CAUTION

The existing file will be overwritten when the <Yes> button is clicked in the file overwrite confirmation message box.

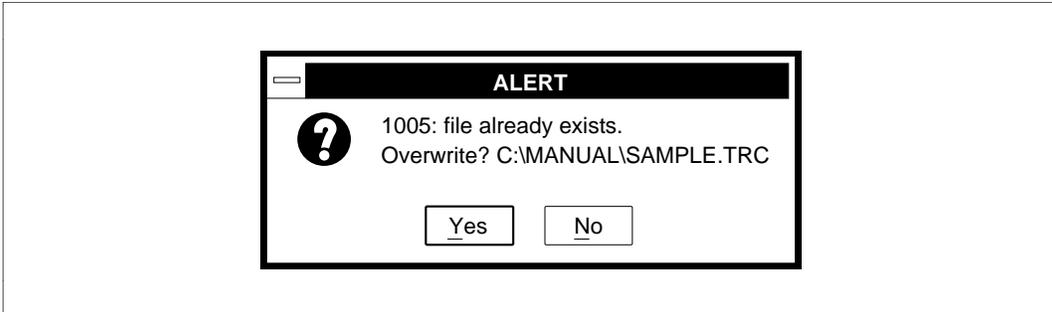


Figure 5-42 File Overwrite Confirmation Message (TRACE OUTPUT Dialog Box)

— <Cancel> button

Clicking the <Cancel> button closes the TRACE OUTPUT dialog box without outputting the settings to a file. It also returns the system to its state when the TRACE OUTPUT dialog box was opened without storing the settings.

- Additional information in bus-cycle unit display

When trace information is displayed in bus-cycle units by selecting <Bus cycle> in [Text display format] of the [Trace display - Setting...] command, the message

```
*** E7000 ***
```

is added to the display to indicate the E7000PC emulator cycle in the last bus cycle in which the user program was terminated. It is also displayed when emulation is temporarily halted by the break conditions set by the BREAK_SEQUENCE command (emulator command) being satisfied, or by the number of breaks set by the BREAK command (emulator command) being encountered.

Note

If <Range> is specified in [Condition type] of the [Trace condition...] command, assembly codes cannot be displayed correctly. They can be used for only reference.

Related Functions

GUI commands: [Trace mode...], [Trace condition...], and [Trace display - Setting...]

Emulator commands: BREAK and BREAK_SEQUENCE

5.5.5 Displaying Trace Information in Graph Form

(Alt + P)

[Trace display - Graph]

TARGET SELECTION dialog box

Overview

Displays trace information as a graph according to the specifications in the [Trace display - Setting...] command.

Window

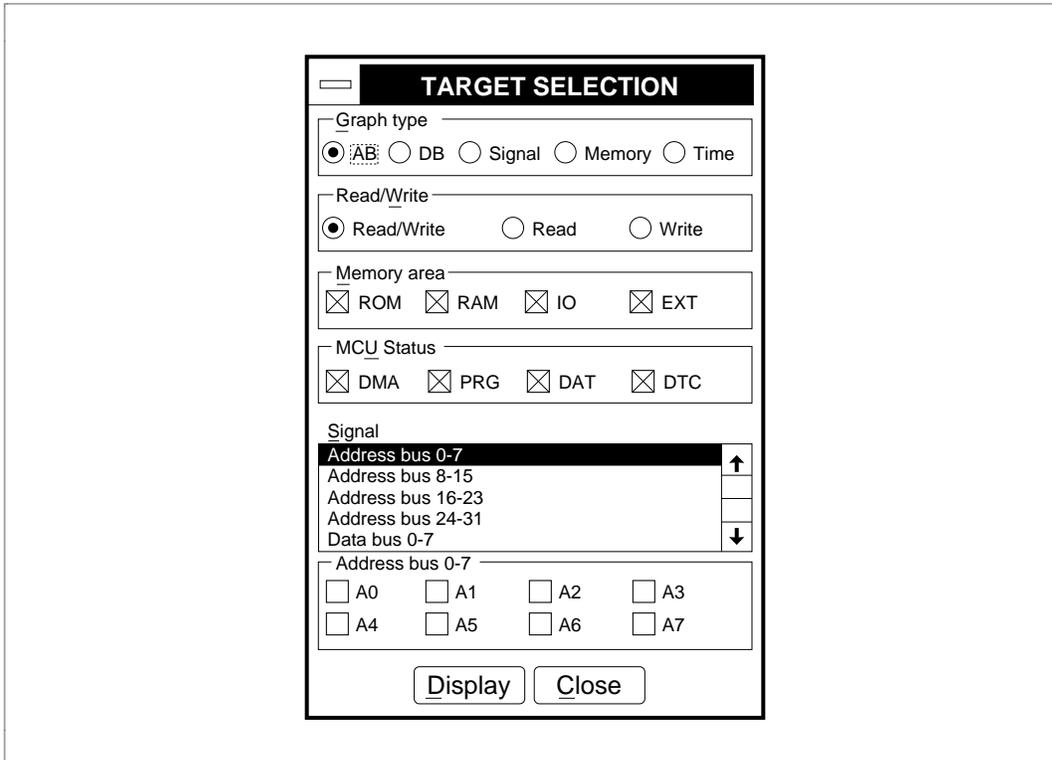


Figure 5-43 TARGET SELECTION Dialog Box

Operation

- [Graph type] group box

Specifies the type of graph to be used in displaying trace information as follows:

<AB>: Change in address bus value in bus cycle units (default at system initiation)

<DB>: Change in data bus value in bus cycle units

<Signal>: Changes in H8S/2655-series control signals and external probe signals in bus cycle units

<Memory>: Change in memory data specified by [Trace Memory condition] in the [Trace mode...] command in bus cycle units

<Time>: Execution time (access time) as total number of clock cycles in address bus cycle units in histogram form

- [Read/Write] group box

Specifies a read or write cycle condition as a trace information display condition as shown below. This specification is valid only when <AB> or <DB> is selected in [Graph type].

<Read/Write>: Displays trace information acquired in read and write cycles as a graph (default at system initiation)

<Read>: Displays trace information acquired in read cycles as a graph

<Write>: Displays trace information acquired in write cycles as a graph

- [Memory area] check box

Specifies memory access conditions as trace information display conditions as shown below. This specification is valid only when <AB> or <DB> is selected in [Graph type].

<ROM>: Displays trace information acquired in internal ROM area access cycles as a graph

<RAM>: Displays trace information acquired in internal RAM area access cycles as a graph

<IO>: Displays trace information acquired in internal I/O area access cycles as a graph

<EXT>: Displays trace information acquired in external memory area access cycles as a graph

The check box works as follows:

<Check>: The access condition is specified as a trace information display condition

<No check>: The access condition is not specified as a trace information display condition

Immediately after system initiation, all check boxes are specified as trace information display conditions.

- [MCU Status] check box

Specifies H8S/2655-series MCU bus cycle conditions as trace information display conditions. This specification is valid only when <AB> or <DB> is selected in [Graph type].

<DMA>: Displays trace information acquired in DMA cycles as a graph

<PRG>: Displays trace information acquired in program fetch cycles as a graph

<DAT>: Displays trace information acquired in I/O access or memory access cycles as a graph

<DTC>: Displays trace information acquired in DTC cycles as a graph

The check box works as follows.

<Check>: The H8S/2655-series bus cycle condition is specified as a trace information display condition

<No check>: The H8S/2655-series bus cycle condition is not specified as a trace information display condition

Immediately after system initiation, all check boxes are specified as trace information display conditions.

- [Signal] list box
Specifies H8S/2655-series control signal and external probe signal conditions as trace information display conditions. A maximum of 32 graphs can be selected and displayed in the graphic display area. This specification is valid only when <Signal> is specified in [Graph type]. Use the following procedure to perform the specification.

(1) Select from the list box a signal group including the required signal and click it. Selectable signal groups in the list box are shown in table 5-12.

Table 5-12 [Signal] List in [Trace display - Graph] Command

Signal Group	Signals
Address bus 0-7	Changes in address bus signal bits 0 to 7 waveforms in bus cycle units
Address bus 8-15	Changes in address bus signal bits 8 to 15 waveforms in bus cycle units
Address bus 16-23	Changes in address bus signal bits 16 to 23 waveforms in bus cycle units
Address bus 24-31	Changes in address bus signal bits 24 to 31 waveforms in bus cycle units
Data bus 0-7	Changes in data bus signal bits 0 to 7 waveforms in bus cycle units
Data bus 8-15	Changes in data bus signal bits 8 to 15 waveforms in bus cycle units
Non maskable interrupt	Change in the NMI signal waveform in bus cycle units
Bus request	Change in the BREQ signal waveform in bus cycle units
Bus acknowledge	Change in the BACK signal waveform in bus cycle units
Interrupt 0-7	Changes in the IRQ0 to IRQ7 signal waveforms in bus cycle units
External probe 1-8	Changes in the external probe 1 to 8 signal waveforms in bus cycle units

- (2) The signals selected in the list box are displayed as check boxes. By checking displayed check boxes, the control signals and external probe signals for drawing trace information as a graph are specified.
- <Check>: The control signal and external probe signal conditions are specified as trace information display conditions
 - <No check>: The control signal and external probe signal conditions are not specified as trace information display conditions
 - <Display> button
 Clicking the <Display> button opens the TRACE DISPLAY GRAPH window and displays trace information as a graph depending on the TARGET SELECTION dialog box settings. Figure 5-44 shows the TRACE DISPLAY GRAPH window.
 - <Close> button
 Clicking the <Close> button stores the TARGET SELECTION dialog box settings and closes only the TARGET SELECTION dialog box.

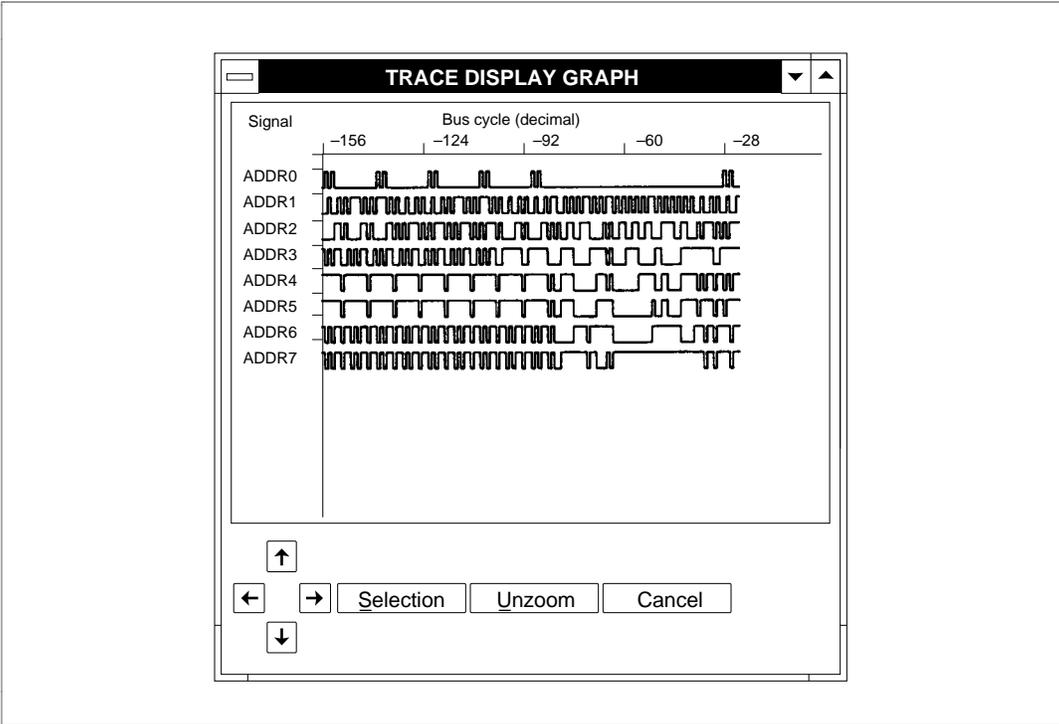


Figure 5-44 TRACE DISPLAY GRAPH Window

- **Graphic display area**
Displays trace information as a graph in the graphic display area. Zooming can be performed on a rectangular area specified by dragging the mouse.

- **Scroll buttons**
Clicking these scroll buttons enables scrolling of the graphic display area in the TRACE DISPLAY GRAPH window.
 - <↑> button
Scrolls the graph display up by 1/4 display.
 - <←> button
Scrolls the graph display left by 1/4 display.
 - <→> button
Scrolls the graph display right by 1/4 display.
 - <↓> button
Scrolls the graph display down by 1/4 display.

- <Selection> button
Clicking the <Selection> button opens the TARGET SELECTION dialog box.

- <Unzoom> button
Clicking the <Unzoom> button returns the display to the display before it was zoomed in by the mouse.

- <Cancel> button
Clicking the <Cancel> button closes the TRACE DISPLAY GRAPH window.

Function

- **Trace information graph display**
The acquired trace information is displayed as a graph in the TRACE DISPLAY GRAPH window. One of the five types of graph listed in table 5-13 can be displayed according to the [Graph type] group box setting in the TARGET SELECTION dialog box.

Table 5-13 Types of Graph

[Graph type]	Vertical Direction	Horizontal Direction
AB	Address bus value	Bus cycle count
DB	Data bus value	Bus cycle count
Signal	Specified signal	Bus cycle count
Memory	Data bus value	Bus cycle count
Time	Address bus value	Clock count

Note: For the bus cycle whose trace information is displayed as ***** E7000 ***** in the TRACE DISPLAY TEXT window, the displayed graph for the bus cycle is invalid.

- Trace information display for an address bus waveform
 When <AB> is specified in [Graph type], trace information acquired for changes in an address bus value which satisfies the settings in [Read/Write], [Memory area], and [MCU status] is displayed in bus cycle units as a graph. An example of trace information display for an address bus waveform is shown in figure 5-45.

In an address bus graph, the vertical axis indicates the address bus value and the horizontal axis indicates the bus cycle count. Clicking the scroll buttons allows the user to scroll the graph display vertically and horizontally.

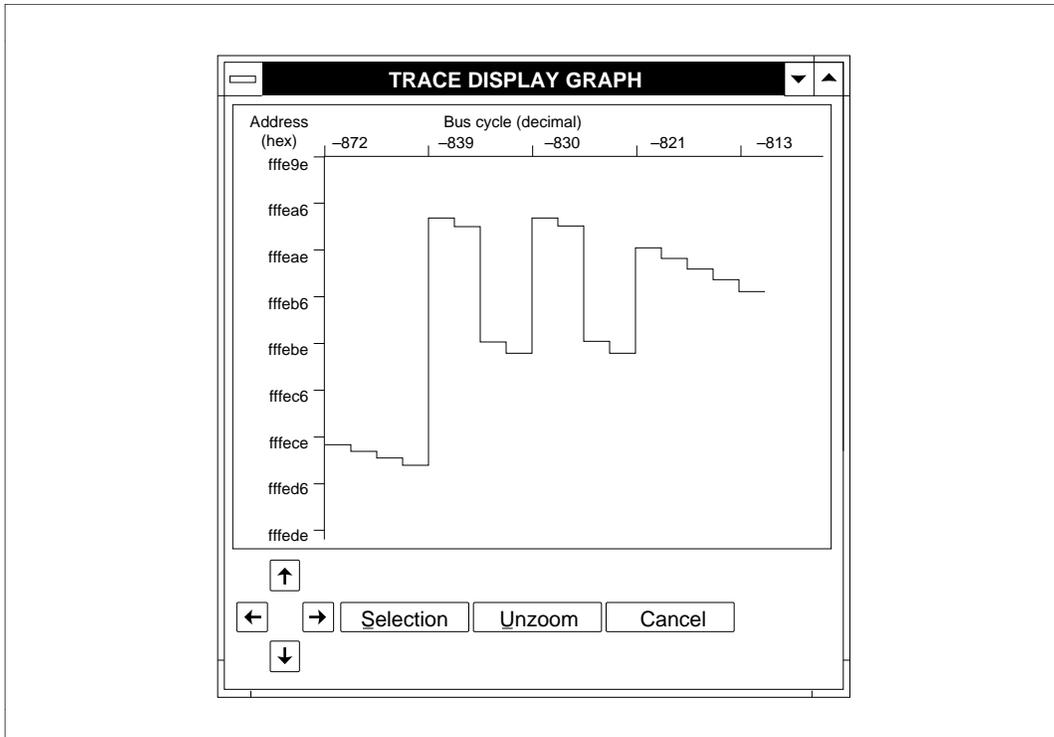


Figure 5-45 Trace Information Display for Address Bus Waveform (TRACE DISPLAY GRAPH Window)

- Trace information display for a data bus waveform
 When <DB> is specified in [Graph type], trace information acquired for changes in a data bus value which satisfies the settings in [Read/Write], [Memory area], and [MCU status] is displayed in bus cycle units as a graph. An example of trace information display for a data bus waveform is shown in figure 5-46.

In a data bus graph, the vertical axis indicates the data bus value and the horizontal axis indicates the bus cycle count. Clicking the scroll buttons allows the user to scroll the graph display vertically and horizontally.

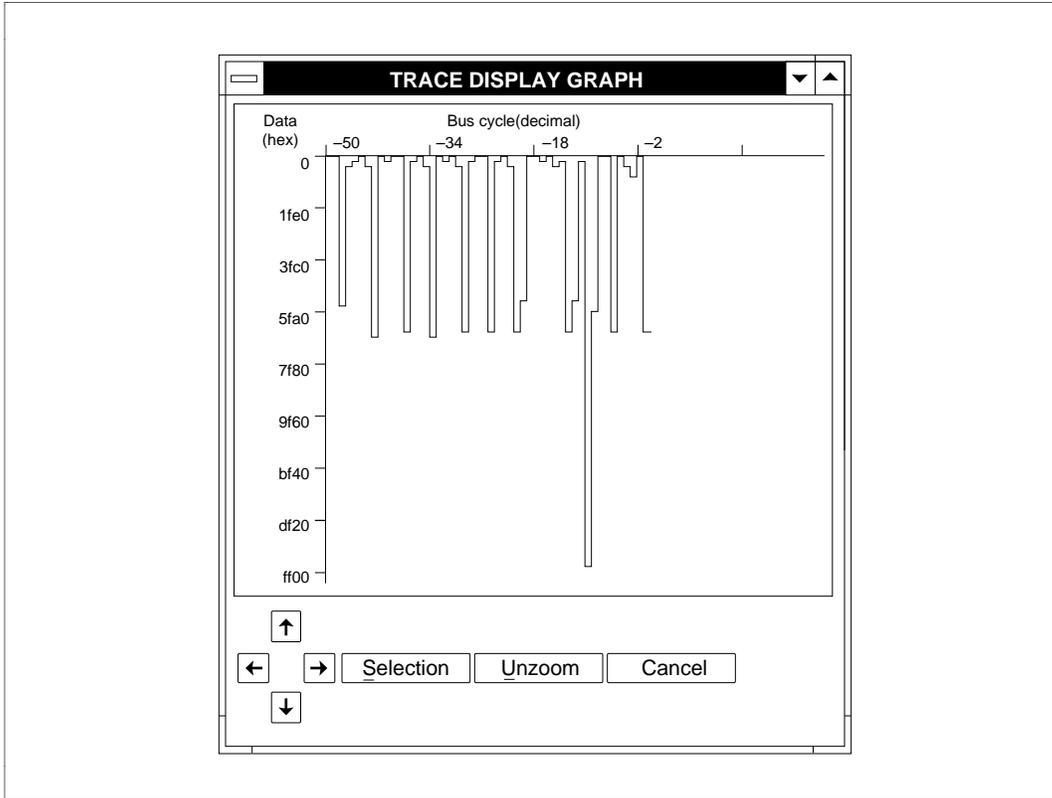


Figure 5-46 Trace Information Display for Data Bus Waveform (TRACE DISPLAY GRAPH Window)

- Trace information display for signal waveforms
 When <Signal> is specified in [Graph type], trace information acquired for changes in the H8S/2655-series control signals and external probe signals is displayed in bus cycle units as a graph. An example of trace information display for signal waveforms is shown in figure 5-47.

In a signal graph, the vertical axis indicates the specified signals and the horizontal axis indicates the bus cycle count. Clicking the scroll buttons allows the user to scroll the graph display vertically and horizontally.

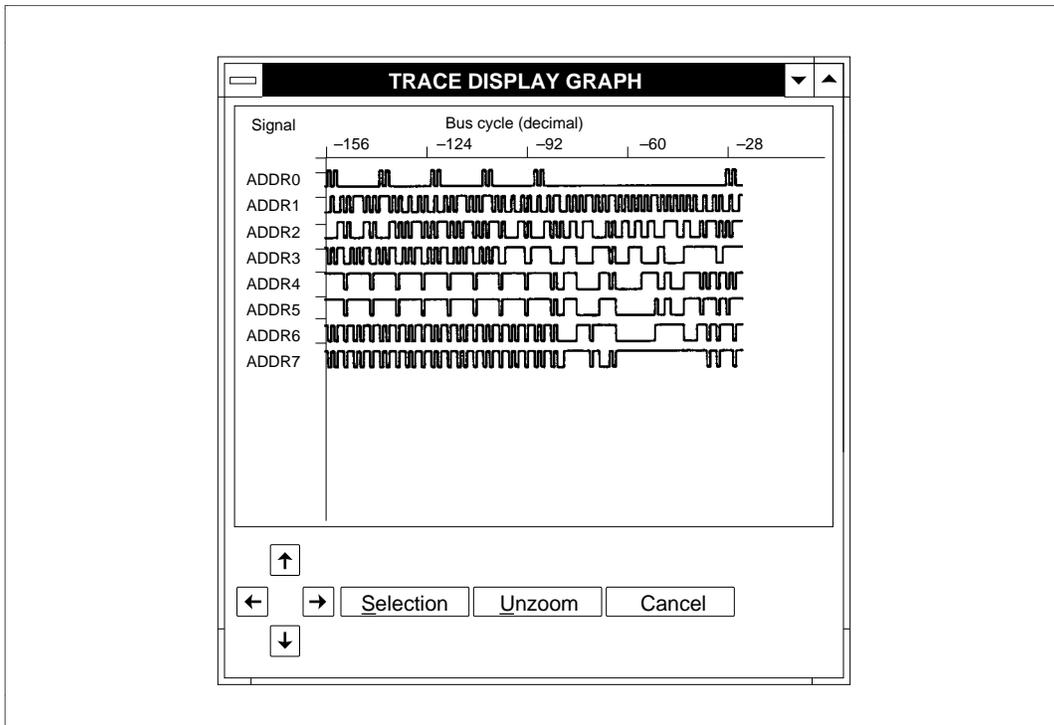


Figure 5-47 Trace Information Display for Signal Waveform (TRACE DISPLAY GRAPH Window)

- Trace information display for memory contents change

When <Memory> is specified in [Graph type], trace information acquired for data changes within the address set in [Trace Memory condition] of the [Trace mode...] command is displayed in bus cycle units as a graph. An example of trace information display for changes in memory is shown in figure 5-48.

In a memory change graph, the vertical axis indicates the data bus value and the horizontal axis indicates the bus cycle count. Clicking the scroll buttons allows the user to scroll the graph display vertically and horizontally.

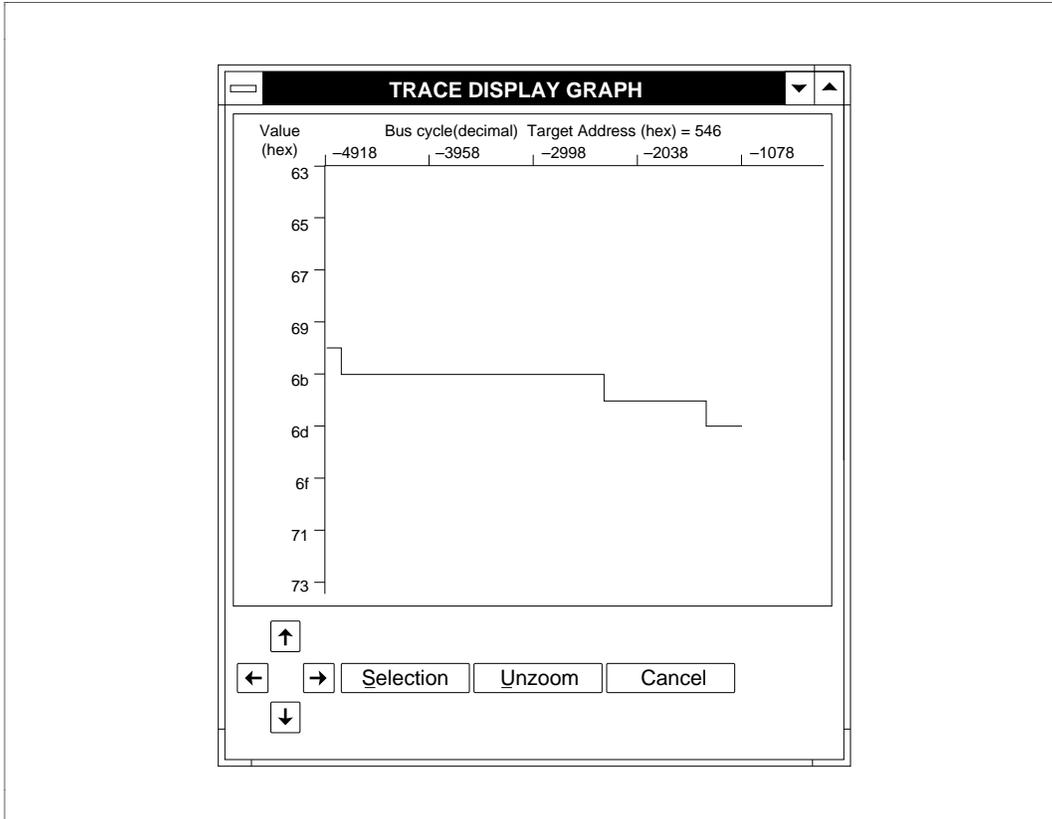


Figure 5-48 Trace Information Display for Memory Change (TRACE DISPLAY GRAPH Window)

- Trace information display for execution time (execution-time histogram)
When <Time> is specified in [Graph type], the total execution time (access time) as a number of clock cycles is shown in a histogram in address bus cycle units. An example of trace information display for execution time is shown in figure 5-49 as a histogram.

In an execution-time histogram, the vertical axis indicates the address bus value and the horizontal axis indicates the clock count. Clicking the scroll buttons allows the user to scroll the graph display vertically and horizontally.

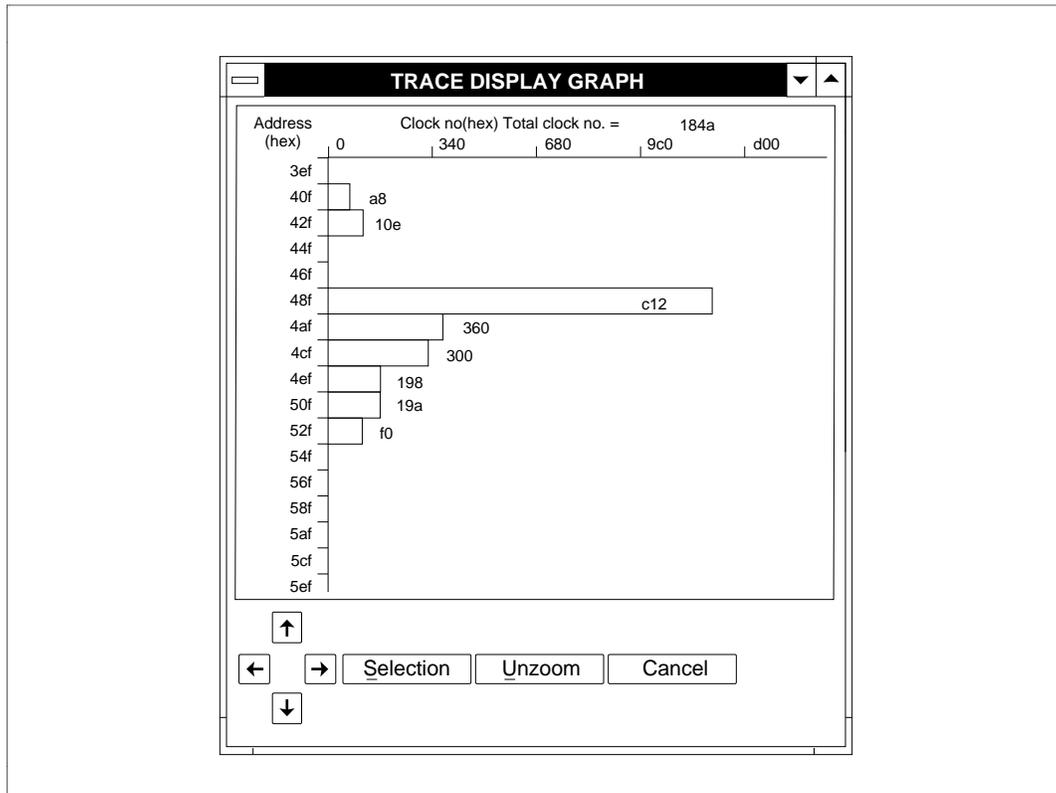


Figure 5-49 Trace Information Display for Execution Time (TRACE DISPLAY GRAPH Window)

- Zooming function
The graphic display area in the TRACE DISPLAY GRAPH window can be enlarged and reduced by dragging a window corner.

To enlarge or reduce the graph itself, use the zooming function. This function is useful for checking trace information displays in detail, since an area on the graph specified by the mouse can be enlarged. An example using the zooming function is shown in figure 5-50. Zooming is performed as follows:

- (1) Move the mouse pointer to the graphic display area and press the left button to set the start point.
- (2) Drag the mouse with the left button pressed so that a zoom display frame (dotted line) appears in the graphic display area.
- (3) Release the left button at the desired end point.
- (4) The part enclosed in the zoom display frame is enlarged.

The specified area can be zoomed up to eight times continuously. Clicking the <Unzoom> button returns the display magnification to its previous value.

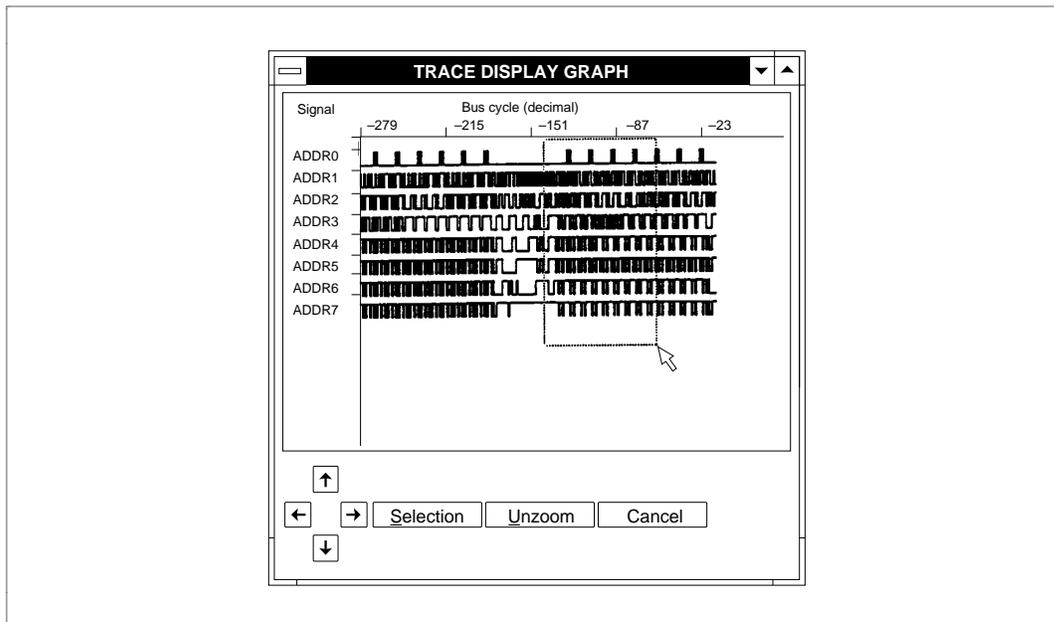


Figure 5-50 Zooming Function Example (TRACE DISPLAY GRAPH Window)

Related Functions

GUI commands: [Trace mode...], [Trace condition...], and [Trace display - Setting...]

5.6 Debug Information Display Functions

5.6.1 Specifying Memory Display Range

(Alt + Ctrl + M)

[Memory - Setting...]

MEMORY RANGE SETTING dialog box

Overview

Specifies the memory range whose contents are displayed with the [Memory - Dump] command.

Window

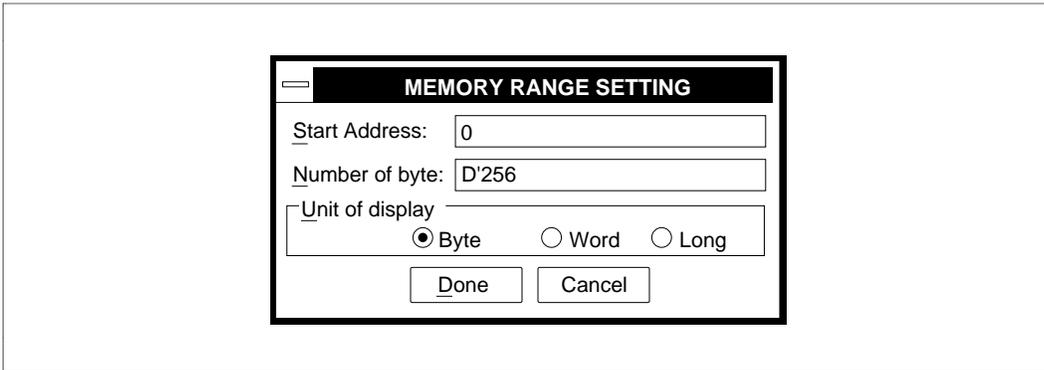


Figure 5-51 MEMORY RANGE SETTING Dialog Box

Operation

- [Start address:] text box
Specifies a number or a symbol for the memory address where display is started. Immediately after system initiation, 0 is specified.
- [Number of byte:] text box
Specifies the size of data to be displayed. Immediately after system initiation, D'256 is specified.
- [Unit of display] group box
Specifies the display unit for memory contents as follows:
 - <Byte>: 1-byte units (default at system initiation)
 - <Word>: 2-byte units
 - <Long word>: 4-byte units

- <Done> button

Clicking the <Done> button opens the MEMORY DUMP window and displays the contents of the memory range specified in the MEMORY RANGE SETTING dialog box. Then stores the MEMORY RANGE SETTING dialog box settings and closes the MEMORY RANGE SETTING dialog box.

- <Cancel> button

Clicking the <Cancel> button closes the MEMORY RANGE SETTING dialog box and returns the system to its state when the MEMORY RANGE SETTING dialog box was opened without storing the above settings.

Function

Specifies the memory range whose contents are displayed with the [Memory - Dump] command.

Clicking the <Done> button opens the MEMORY DUMP window in which memory contents are displayed according to the specifications in [Start address:], [Number of byte:], and [Unit of display].

Related Function

GUI command: [Memory - Dump]

5.6.2 Displaying and Modifying Memory Contents

(Alt + M)

[Memory - Dump]

MEMORY DUMP window

Overview

Displays the memory contents specified by the [Memory - Setting...] command. The memory contents displayed in the window can be modified directly. The displayed memory contents can also be stored in a file.

Window

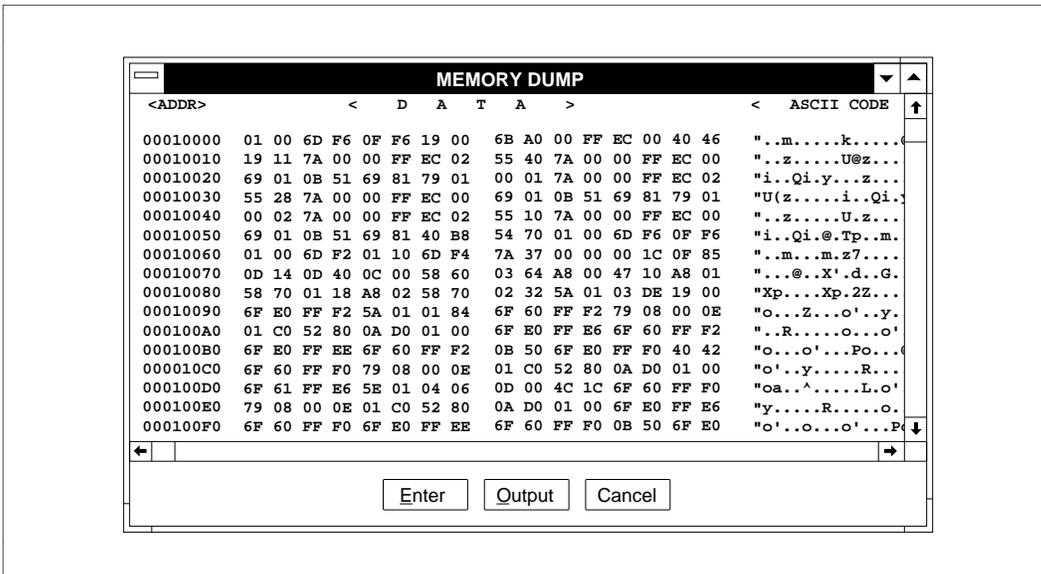


Figure 5-52 MEMORY DUMP Window

Operation

- Memory contents display area
Displays the addresses, data, and corresponding ASCII code of the memory range specified by the [Memory - Setting...] command. The memory contents displayed in the memory contents display area can be modified directly.
- <Enter> button
Clicking the <Enter> button after changing memory data in the memory contents display area actually rewrites data to the E7000PC emulator memory. Note that memory contents are actually rewritten only after the <Enter> button has been clicked.

- <Output> button
Clicking the <Output> button opens the MEMORY OUTPUT dialog box. The data currently displayed in the memory contents display area can be output to a file.
- <Cancel> button
Clicking the <Cancel> button closes the MEMORY DUMP window.

Function

- Memory contents display
Displays contents of the memory range specified by the [Memory - Setting...] command in the memory contents display area. Consequently, when changing the memory contents display area with the [Memory - Setting...] command while the MEMORY DUMP window is opened, the newly specified area will be displayed replacing the previous one.

During emulation execution, the contents displayed in the memory contents display area are automatically modified and re-displayed when emulation is terminated in one of the following ways.

- When the emulation initiated by clicking the <STEP>, <STEP_OVER>, <STEP_UP>, or <CONTINUE> button in the base window is terminated
 - When the emulation initiated by clicking the <Step>, <Step_over>, or <Continue> button in the DISASSEMBLE window is terminated
 - When the emulation initiated by the [Go...] command is terminated
 - When the emulation initiated by the STEP, STEP_OVER, or GO command (emulator commands) entered in the command area is terminated
 - When emulation is forcibly terminated by clicking the <STOP> button in the base window
- Memory contents modification
Memory contents displayed in the memory contents display area can be modified directly, as shown in figure 5-53.
 - After moving the mouse pointer to the address whose data is to be modified, click the left button and produce a cursor (|).
When modifying through only the keyboard, press the (Tab) key and produce a cursor in the memory contents display area. Then move the cursor to the data to be modified with the arrow keys.
 - Enter new data from the keyboard. The cursor will move right by one digit for each new digit entered. The cursor will not move the basic display unit of memory data.
Data can be modified only in the area titled <DATA>.
 - To move the cursor to the next data item to be modified, use the mouse or enter the arrow keys.

- Clicking the <Enter> button actually writes the modified data into the specified memory address on the E7000PC emulator. A maximum of 256 data items can be modified at one time.

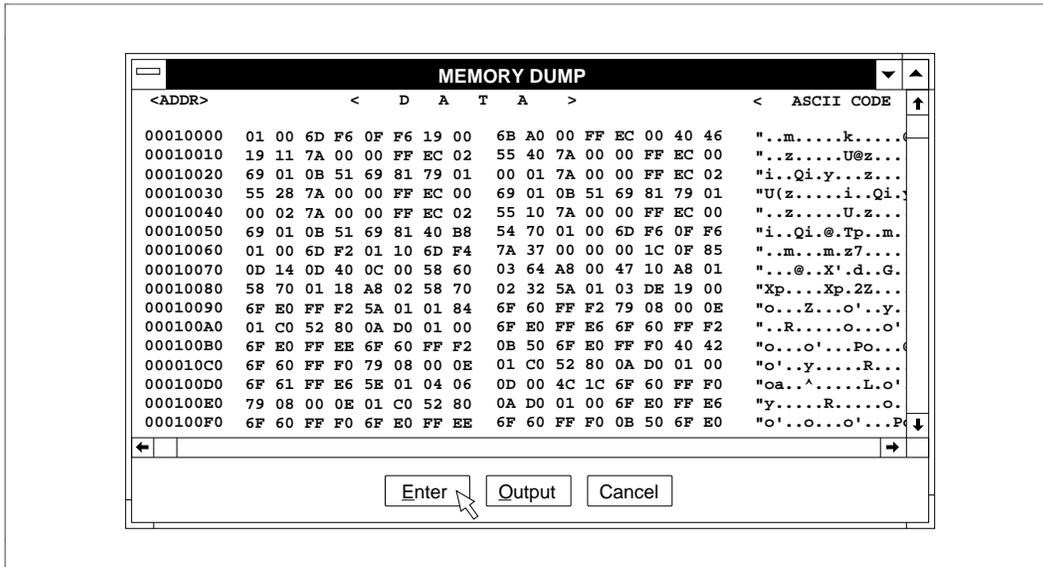


Figure 5-53 Example of Modifying Memory Contents

- Memory contents display area output
The memory contents displayed in the memory contents display area can be output to a file. Clicking the <Output> button displays the MEMORY OUTPUT dialog box shown in figure 5-54.

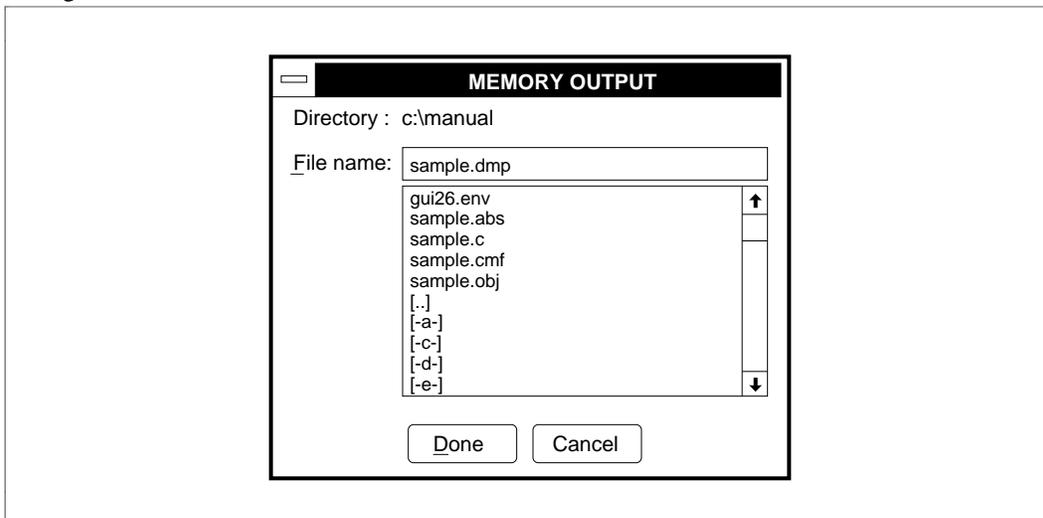


Figure 5-54 MEMORY OUTPUT Dialog Box

— [File name:] text box

Specifies the name of the file where the memory contents displayed in the memory contents display area are to be stored. A file with the specified file name is output to the current directory displayed in [Directory:].

Pressing the (Enter) key after inputting or selecting a file name or double-clicking the file name specified in the list box outputs the contents of the MEMORY DUMP window to a file.

The destination directory can be changed by specifying a new current directory in the sequence of drive name and directory name.

To change the current directory, use the following procedure:

- (1) Select the target drive name from the list box by double-clicking. Drive names are displayed in the format [-a-]. After selected, the drive name and current directory name are displayed in [Directory:]. The current directory contents of the selected drive are displayed in the list box.
- (2) Select a subdirectory name from the list box by double-clicking. The subdirectory name is displayed in the format [directory]. After selected, the subdirectory name is added to [Directory:] and the subdirectory contents are displayed in the list box.
- (3) Continue selecting subdirectories until the directory where the target file is to be output is reached.

— [Directory:]

Displays the current directory. When the drive or subdirectory is changed in [File name:], this display also changes.

— <Done> button

Clicking the <Done> button starts storing displayed memory contents to the file specified in [File name:] and closes the MEMORY OUTPUT dialog box after storing the MEMORY OUTPUT dialog box settings. When a file having the same name as that specified in [File name:] already exists, the message box shown in figure 5-55 appears. Click <Yes> to start overwriting the existing file specified in [File name:]. When <No> is clicked, the system returns to the MEMORY OUTPUT dialog box.

CAUTION

The existing file will be overwritten when the <Yes> button is clicked in the file overwrite confirmation message box.

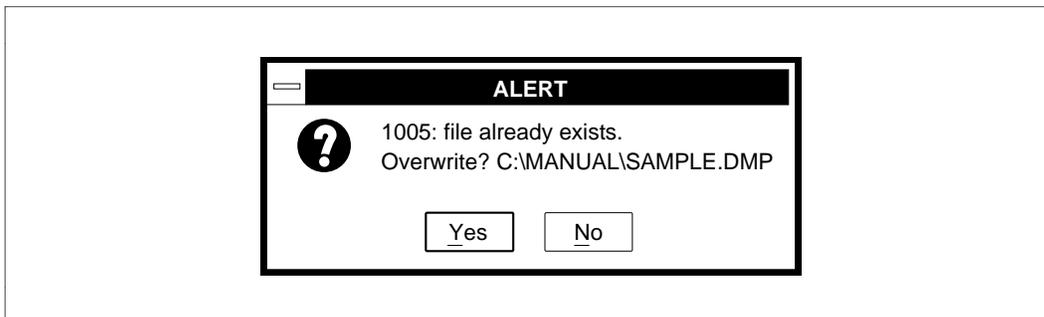


Figure 5-55 File Overwrite Confirmation Message (MEMORY OUTPUT Dialog Box)

— <Cancel> button

Clicking the <Cancel> button closes the MEMORY OUTPUT dialog box without outputting the settings to a file. It also returns the system to its state when the MEMORY OUTPUT dialog box was opened without storing the settings.

Related Functions

<STEP>, <STEP_OVER>, <STEP_UP>, <CONTINUE>, and <STOP> buttons

GUI commands: [Memory - Setting...], [Disassemble], and [Go...]

Emulator commands: STEP, STEP_OVER, and GO

5.6.3 Displaying and Modifying Register Contents

(Alt + R)

[Register]

REGISTER window

Overview

Displays the contents of the H8S/2600-series registers. The registers displayed in the window can be directly modified.

Window

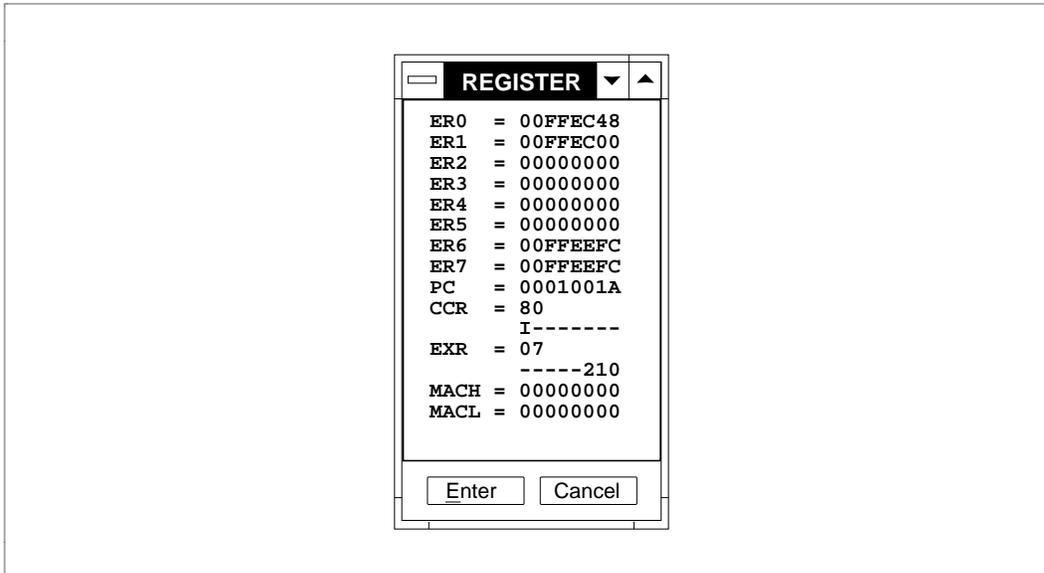


Figure 5-56 REGISTER Window

Operation

- Register display area
Displays the contents of the registers.
- <Enter> button
Clicking the <Enter> button after changing register data in the register display area actually rewrites data to the registers. Note that register data is rewritten only after the <Enter> button has been clicked.
- <Cancel> button
Clicking the <Cancel> button closes the REGISTER window.

Function

- Register data display

Displays register data in the register display area when emulation execution has terminated. During emulation execution, the register contents displayed in the register display area are automatically read and re-displayed when emulation has terminated in one of the following ways.

- When the emulation initiated by clicking the <STEP>, <STEP_OVER>, <STEP_UP>, or <CONTINUE> button in the base window is terminated
- When the emulation initiated by clicking the <Step>, <Step_over>, or <Continue> button in the DISASSEMBLE window is terminated
- When the emulation initiated by the [Go...] command is terminated
- When the emulation initiated by the STEP, STEP_OVER, or GO command (emulator commands) entered in the command area is terminated
- When emulation is forcibly terminated by clicking the <STOP> button in the base window

- Register data modification

Register data displayed in the register display area can be modified directly, as shown in figure 5-57.

- After moving the mouse pointer to the register whose data is to be modified, click the left button and produce a cursor (|).
When modifying through only the keyboard, press the (Tab) key and produce a cursor in the register display area. Then move the cursor to the data to be modified with the arrow keys.
- Enter new data from the keyboard. The cursor will move right by one digit for each new digit entered. The cursor will not move the basic display unit of register data.
- To move the cursor to the next data item to be modified, use the mouse or enter the arrow keys.
- Clicking the <Enter> button actually writes the modified data into the specified register.

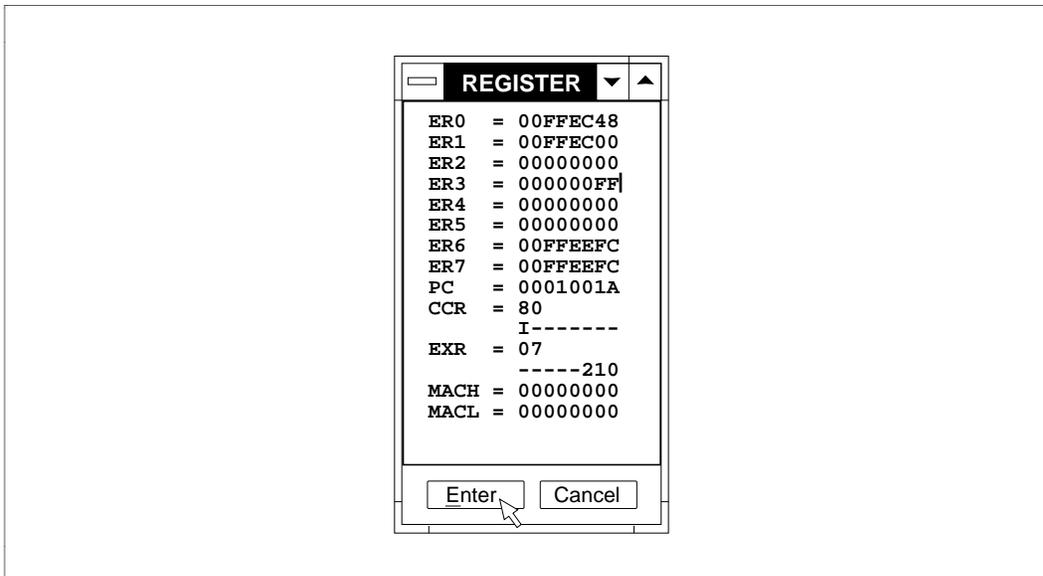


Figure 5-57 Example of Modifying Register Contents (REGISTER Window)

Related Functions

<STEP>, <STEP_OVER>, <STEP_UP>, <CONTINUE>, and <STOP> buttons GUI commands:
[Go...] and [Disassemble]

Emulator commands: STEP, STEP_OVER, and GO

5.6.4 Setting Watch Points

(Alt + Ctrl + W)

[Watch - Setting...]

WATCH SETTING dialog box

Overview

Sets watch points (addresses whose memory contents are displayed in realtime during emulation).

Window

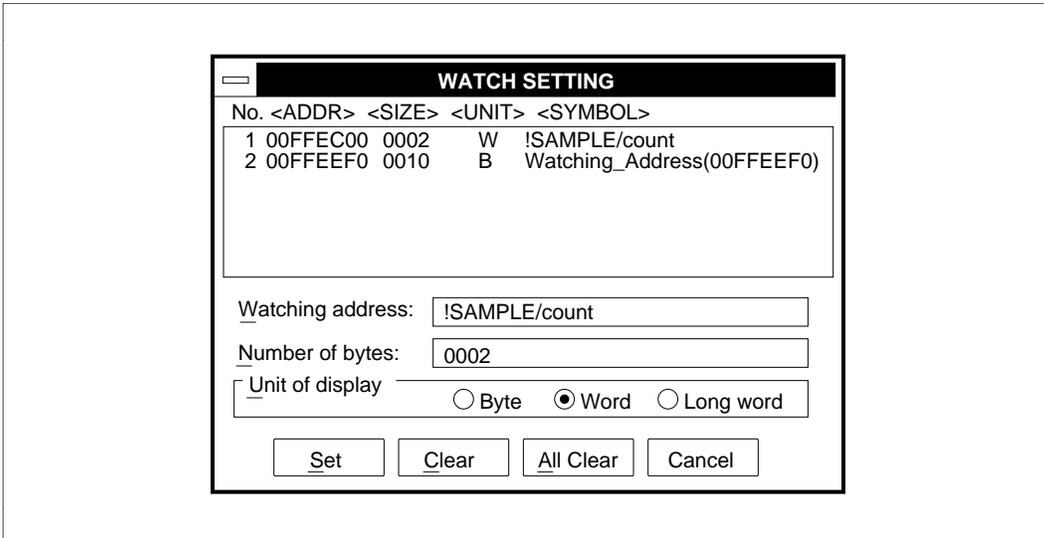


Figure 5-58 WATCH SETTING Dialog Box

Operation

- Watch point display area
Displays watch point settings in the format shown in figure 5-59.

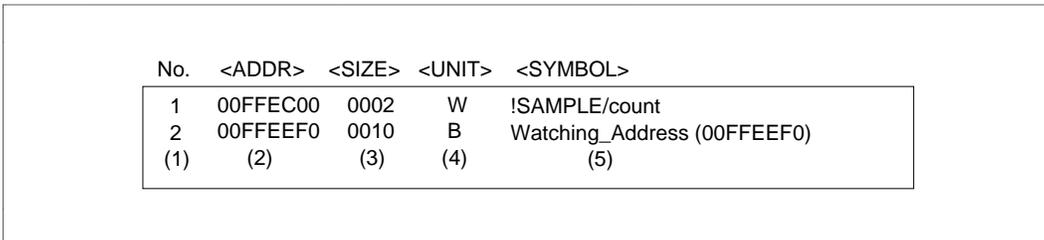


Figure 5-59 Watch Point Display Format (WATCH SETTING Dialog Box)

- (1) No.: Watch point number
- (2) <ADDR>: Address at which a watch point is set
- (3) <SIZE>: Display size of watch point (byte-size in hexadecimal)
- (4) <UNIT>: Display unit of watch point
 - B: 1-byte units
 - W: 2-byte units
 - L: 4-byte units
- (5) <SYMBOL>: Symbol name (If no symbol name is defined for the watch point address, then Watching_Address (address value) is shown.)

When a displayed watch point is selected by the mouse pointer and double-clicked, the contents of the line are set in [Watching address:], [Number of bytes:], and [Unit of display].

- [Watching address:] text box
Specifies a number or a symbol for an address where a watch point is set.
- [Number of bytes:] text box
Specifies the size of the watch point to be displayed. The display size can be specified from one to 256 bytes in 1-byte units. When no size is specified, two bytes is assumed if <Byte> or <Word> is specified in [Unit of display]. Four bytes is assumed if <Long> is specified in [Unit of display].
- [Unit of display] group box
Specifies the unit for displaying the contents of the watch point.
 - <Byte>: 1-byte units (default at system initiation)
 - <Word>: 2-byte units
 - <Long word>: 4-byte units
- <Set> button
Clicking the <Set> button after setting values in [Watching address:], [Number of bytes:], and [Unit of display] sets the specified point as a watch point and additionally displays it in the watch point display area. A maximum of eight watch points can be specified.

- <Clear> button
Clicking the <Clear> button cancels a watch point specified by setting an address in [Watching address:] or by selecting an address in the watch point display area. The cancelled watch point disappears from the watch point display area.
- <All Clear> button
Clicking the <All Clear> button clears all specified watch points. Consequently, all watch points disappear from the watch point display area.
- <Cancel> button
Clicking the <Cancel> button closes the WATCH SETTING dialog box.

Function

- Watch function
The watch function provided for the E7000PC emulator displays the contents of the watch point address executed during emulation in the WATCH DISPLAY window. The watch point display is updated in 1-second units.
- Watch point address
An address in the emulation memory must be specified in [Watching address:]. Otherwise, realtime emulation cannot be performed.
- Watch function termination
When a menu command or an emulator command is executed during emulation, the E7000PC emulator enters parallel mode, thus causing the watch function to be terminated.

Note

A watchpoint cannot be set exceeding the boundary of 256 bytes.

Related Function

GUI command: [Watch - Display]

5.6.5 Displaying Watch Point Contents

(Alt + W)

[Watch - Display]

WATCH DISPLAY window

Overview

Displays the contents of the watch points (addresses whose memory contents are displayed in realtime during emulation).

Window

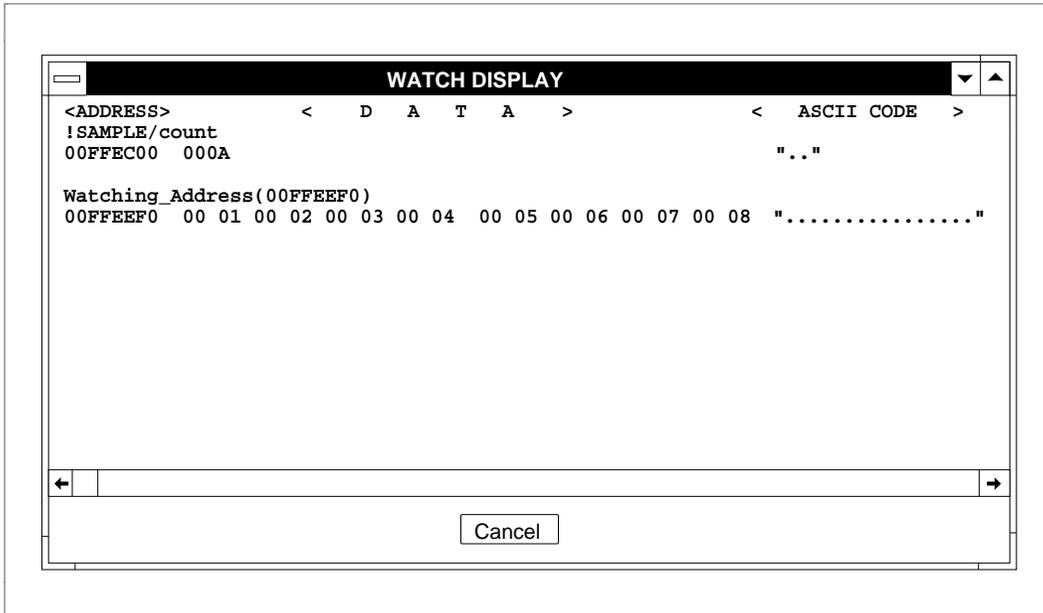


Figure 5-60 WATCH DISPLAY Window

Operation

- Watch point contents display area
According to the [Watch - Setting...] command specifications, displays watch point contents in the format shown in figure 5-61.

<ADDRESS>	< D A T A >	<ASCII CODE>
!SAMPLE/count (1)		
00FFEC00 (2)	000A (3)	".." (4)

Figure 5-61 Watch Point Contents Display Format (WATCH DISPLAY Window)

- (1) <ADDRESS>: Symbol name
Symbol name in the same format as displayed in the watch point display area of the [Watch - Setting...] command
- (2) <ADDRESS>: Watch point address
- (3) <DATA>: Contents of the memory address where the watch point is set (in hexadecimal)
- (4) <ASCII CODE>: Contents of the memory address where the watch point is set (in ASCII code)
- <Cancel> button
Clicking the <Cancel> button closes the WATCH DISPLAY window.

Function

- Watch function
The watch function provided for the E7000PC emulator displays the contents of the watch point address executed during emulation in realtime. The watch point display is updated in 1-second units.
- Watch function termination
When the (Space) key or (Enter) key is pressed in the command area during emulation and the E7000PC emulator enters parallel mode, the watch function terminates.

Related Function

GUI command: [Watch - Setting...]

5.6.6 Displaying Symbol Information

(Alt + S)

[Symbol - Display]

SYMBOL DISPLAY window

Overview

Displays symbol information within the scope of the execution stop address.

Window

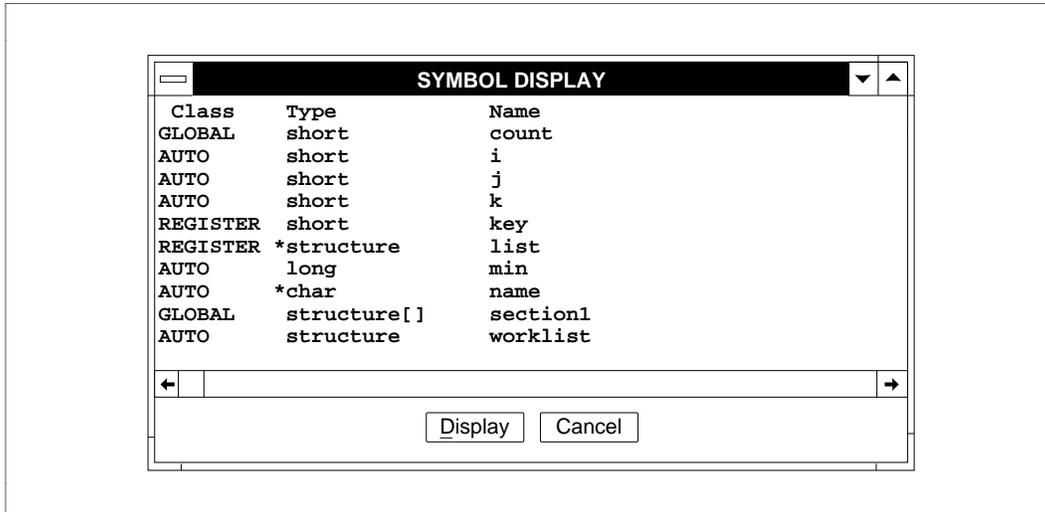


Figure 5-62 SYMBOL DISPLAY Window

Operation

- Symbol information display area

All symbols declared and defined within the scope of the execution stop address are displayed.

The contents of the specified symbol can be displayed by selecting a symbol to be displayed and by clicking the <Display> button.

- <Display> button

Clicking the <Display> button opens the SYMBOL VALUE window and displays that symbol's value in the SYMBOL VALUE window. For details on the SYMBOL VALUE window, refer to section 5.6.7, Displaying and Modifying Symbol Values.

- <Cancel> button
Clicking the <Cancel> button closes the SYMBOL DISPLAY window.

Function

- Scope of the execution stop address
In the symbol information display area, all symbols declared and defined within the scope of the execution stop address are displayed.

The scope of the execution stop address means the range accessible from the function containing the PC when execution stops.

Figure 5-63 shows an example of the execution stop address scope. In this case, the scope covers the area within the main function, and the area outside of functions in the file (unit) containing the main function.

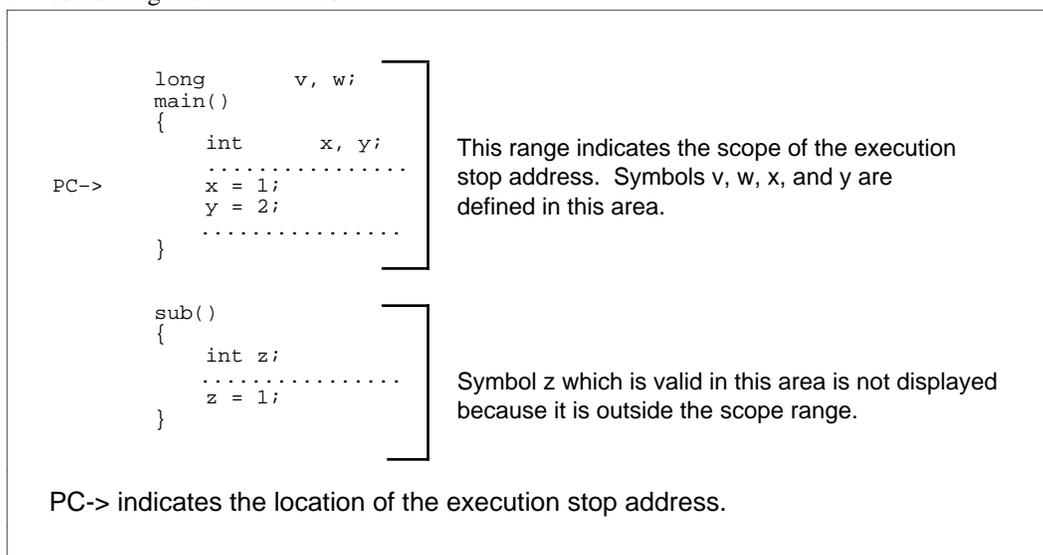


Figure 5-63 Example of Execution Stop Address Scope

When the program is optimized by the C compiler and there are symbols declared with the same name within and outside a function, information on multiple symbols may be displayed depending on the execution stop address.

- Symbol information display area contents

Displays information of symbols in the format shown in figure 5-64.

Class	Type	Name
GLOBAL	short	count
(1)	(2)	(3)

Figure 5-64 Display Format of Symbol Information Display Area (SYMBOL DISPLAY Window)

(1) Class: Storage class to which the symbol is assigned

(2) Type: Symbol attribute

(3) Name: Symbol name

— Class

Displays the storage class to which the symbol is assigned as listed in table 5-14.

Table 5-14 Storage Class Display Format (SYMBOL DISPLAY Window)

Display Format	Storage Class
REGISTER	Register variable
GLOBAL	Externally defined variable
EXTERN	External reference variable
STATIC	Static variable
AUTO	Local variable
(AUTO)	No assignment* ¹
NONE* ²	EQUATE name

Notes: 1. When the program is optimized by the C compiler, it may not be possible to assign local variables and register variables declared within the scope to any storage class. In such a case, the storage class is displayed as (AUTO).

2. No data is displayed as Class for the EQUATE name.

— Type

Displays the symbol attribute in the format shown in table 5-15.

Table 5-15 Symbol Attribute Display Format (SYMBOL DISPLAY Window)

Symbol Attribute	Display Format	Description
Integer	char	Signed 1-byte integer variables
	u_char	Unsigned 1-byte integer variables
	short	Signed 2-byte integer variables (including int)
	u_short	Unsigned 2-byte integer variables (including unsigned int)
	long	Signed 4-byte integer variables
	u_long	Unsigned 4-byte integer variables
Floating point	float	Single-precision floating-point variables
	double	Double-precision floating-point variables (including long double)
Structure	structure	Structure
Union	union	Union
Array	xxxx[]	Array variables are indicated by [] following an integer, floating-point, structure, or union variable Example: struct []
Pointer	*xxx	Pointer variables are indicated by an asterisk (*) placed in front of an integer, floating-point, structure, or union variable Example: *float
Label	label	Labels in the assembly program
EQUATE	EQUATE	Symbols assigned by the .equ directive in the assembly program
Enumeration	short	Enumeration type

— Name

Displays the symbol name.

- Symbol value display

The value of a symbol in the symbol information display area can be displayed by selecting it with the mouse and by clicking the <Display> button to open the SYMBOL VALUE window.

- **Symbol information automatic modification**

The contents of the symbol information display area are automatically modified and displayed when emulation is terminated in one of the following ways. In these cases, the symbol information of all symbols within the scope of the execution stop address is displayed.

- When the emulation initiated by clicking the <STEP>, <STEP_OVER>, <STEP_UP>, or <CONTINUE> button in the base window is terminated
- When the emulation initiated by clicking the <Step>, <Step_over>, or <Continue> button of the DISASSEMBLE window is terminated
- When the emulation initiated by the [Go...] command is terminated
- When the emulation initiated by the STEP, STEP_OVER, or GO command (emulator commands) entered in the command area is terminated
- When emulation is forcibly terminated by clicking the <STOP> button in the base window

Notes

- (1) In regard to symbols declared within the function, the symbol information is always displayed regardless of its declaration position.
- (2) If there are symbols declared with the same name within a function, multiple symbol names may be displayed.
- (3) Labels within a function are not displayed.

Related Functions

<STEP>, <STEP_OVER>, <CONTINUE>, and <STOP> buttons

GUI commands: [Symbol - Value], [Go...], and [Disassemble]

Emulator commands: STEP, STEP_OVER, and GO

5.6.7 Displaying and Modifying Symbol Values

(Alt + Ctrl + N)

[Symbol - Value]

SYMBOL VALUE window

Overview

Displays and directly modifies the symbol values specified by the [Symbol - Display] command or by clicking the <DISPLAY> button on the tool bar.

Window

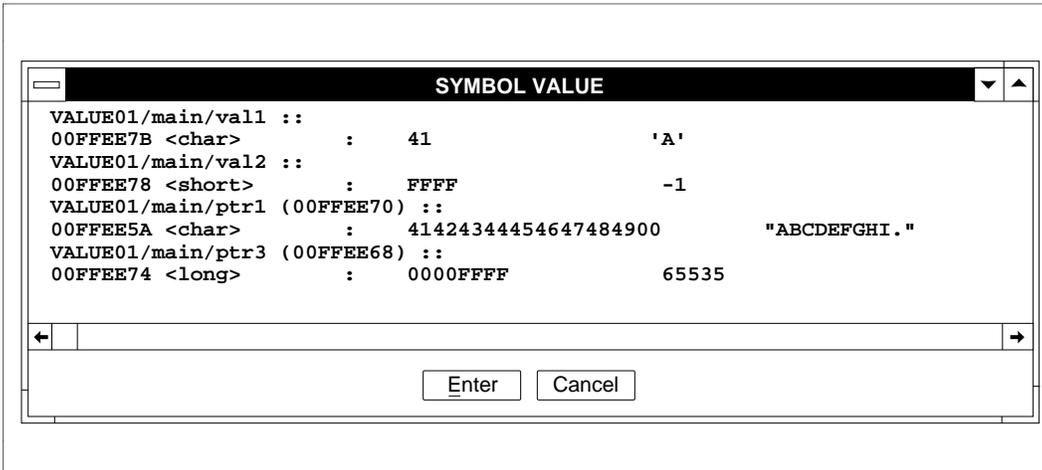


Figure 5-65 SYMBOL VALUE Window

Operation

- Symbol value display area
Displays the symbol names and values or directly modifies the displayed symbol values.
- <Enter> button
Clicking the <Enter> button after modifying symbol values in the symbol value display area actually rewrites data to the E7000PC emulator. Note that symbol values are actually rewritten only after the <Enter> button has been clicked.
- <Cancel> button
Clicking the <Cancel> button closes the SYMBOL VALUE window.

Function

- Symbol value display

Displays a maximum of 16 symbol values specified with the [Symbol - Display] command or the <DISPLAY> button on the tool bar.

- Display format

Figure 5-66 shows the display format of the symbol value display area.

VALUE01/main/ptr2	(00FFEE6C)	::			
(1)	(2)				
00FFEE78	<short>	:	FFFF	-1	
(3)	(4)		(5)	(6)	

Figure 5-66 Symbol Value Display Format (SYMBOL VALUE Window)

(1) Symbol name

Displays the symbol name specified with the [Symbol - Display] command or the <DISPLAY> button on the tool bar.

(2) Pointer address

Displays the address to which a pointer symbol is assigned. This address is displayed within parentheses, only when the symbol is a pointer type.

(3) Symbol address

Displays the address to which a symbol is assigned. For a pointer symbol, the symbol address indicates the address specified by the pointer, and for a register symbol (when the symbol class is register variable), it indicates the register name.

(4) Type

Displays the symbol type enclosed in < >.

(5) Hexadecimal display field

Displays the symbol value in hexadecimal.

(6) Symbol display field

Displays the actual symbol value according to the symbol type.

— Symbol name display format

The symbol name display format is determined depending on the area where the symbol is declared.

- (a) All symbols declared in the assembly source program
 - <unit name>/<variable name>
 - <variable name>
- (b) An externally defined symbol, external reference symbol, or static symbol (declared outside the function) declared in the C source program
 - <unit name>/<variable name>[[<suffix>]...]
 - <variable name>[[<suffix>]...][.<member name>][[<suffix>]...]
- (c) A local symbol, register symbol, or static symbol (declared within the function) declared in the C source program
 - <unit name>/<function name>/<variable name>[[<suffix>]...]
 - <variable name>[[<suffix>]...][.<member name>][[<suffix>]...]

The items are described as follows:

- <unit name>: Unit name (object file name in compile and assembly units)
- <function name>: Function name (only in the C source program)
- <variable name>: Variable name (simple, pointer, array), structure name, union name, label name, or EQUATE name (only a label name and EQUATE name are available in an assembly source program)
- <member name>: Member name of structure or union (only in the C source program)
- <suffix>: Array elements (only in the C source program)

— Symbol type display format

Symbol types are displayed in the symbol value display area in the format shown in table 5-16.

Table 5-16 Symbol Type Display Format (SYMBOL VALUE Window)

Symbol Type	Type Name	Display Size
<char>	char	1 byte
<u_char>	unsigned char	1 byte
<short>	short or int	2 bytes
<u_short>	unsigned short or unsigned int	2 bytes
<long>	long	4 bytes
<u_long>	unsigned long	4 bytes
<float>	float	4 bytes
<double>	double or long double	8 bytes
<char_bit x:y>*	x-bit char	1-byte bit field
<u_char_bit x:y>*	x-bit unsigned char	1-byte bit field
<short_bit x:y>*	x-bit short or x-bit int	2-byte bit field
<u_short_bit x:y>*	x-bit unsigned short or x-bit unsigned int	2-byte bit field
<label>	label	4 bytes
<equate>	EQUATE name	4 bytes

Note: The parameters **x** and **y** indicate the bit field size and offset from the address where the symbol is assigned, respectively.

Example: <u_char_bit 4:0> indicates unsigned char bit-field data whose bit field size is 4 bits and bit offset is 0 bit.

— Symbol display format

The symbol display format is determined depending on the symbol type.

(a) <char>

A <char> symbol is displayed in hexadecimal and ASCII code. If the character cannot be displayed, a period (.) is displayed instead.

For a char pointer symbol, the value up to the null code is displayed in hexadecimal and ASCII code. Hexadecimal values and ASCII codes can display a maximum of 256 bytes.

For a char array symbol, the values for the array elements are displayed in hexadecimal and ASCII code.

```
Example: SAMPLE/main/a ::
          00FFEEFB <char>   :      61           'a'
          SAMPLE/main/p_ascii (00FFEEF6) ::
          00010086 <char>   :      4142434400      "ABCD."
```

(b) Integer type other than <char>

Integer symbols other than <char> are displayed in hexadecimal and decimal.

```
Example: SAMPLE/main/b ::
          00FFEEF2 <long>   :      0000FFFF      65535
```

(c) <float> or <double>

Floating-point data is displayed, as shown in table 5-17.

Table 5-17 Floating-Point Display Format (SYMBOL VALUE Window)

Floating-Point Type	Display Format
Real number (excluding zero)	[-] x.xxxxxxe ± xxx (Signal) (Mantissa) (Exponent) x is a decimal
Zero	0.000000e + 000 or -0.000000e + 000
Positive infinity	Infinity
Negative infinity	-Infinity
Not-a-number (a value not corresponding to a real number or infinity)	NaN (MSB = 0), -NaN (MSB = 1)

```
Example: SAMPLE/main/c ::
          00FFEEEA <double> :      C05ED999999999A -1.234000e+002
```

— Symbol value display examples

Symbol value display examples are summarized in table 5-18.

Table 5-18 Symbol Value Display Examples (SYMBOL VALUE Window)

Symbol Attribute	Declaration Example	Symbol Value Display Example		
Integer	short aaa;	SAMPLE/main/aaa :: 00FFEEE0 <short>	: FFFF	-1
	long bbb;	SAMPLE/main/bbb :: ER5 <long>	: 0000FFFF	65535
Floating-point	double ccc;	SAMPLE/main/ccc : 00FFEEED8 <double> : C05ED9999999999A		-1.234000e+002
Array	char ddd[2];	SAMPLE/main/ddd :: 00FFEEED6 <char>	: 4142	"AB"
	short eee[2];	SAMPLE/main/eee[0] :: 00FFEEED2 <short>	: 0001	1
		SAMPLE/main/eee[1] :: 00FFEEED4 <short>	: 0002	2
Pointer	char *fff;	SAMPLE/main/fff (00FFEECE) :: 00FFEEE6 <char>	: 4142434400	"ABCD."
Structure	struct { short ii1:2; long ii2; }hhh;	SAMPLE/main/hhh :: hhh. ii1 ::		
		00FFEEC4 <short_bit 2: 0>	: 4000	1
		hhh. ii2 :: 00FFEEC6 <long>	: FFFFFFFF	-1
Union	union { char kk1; short kk2; long kk3; }jjj;	SAMPLE/main/jjj :: jjj. kk1 ::		
		00FFEEC0 <char>	: 12	'.'
		jjj. kk2 :: 00FFEEC0 <short>	: 1234	4660
		jjj. kk3 :: 00FFEEC0 <long>	: 12345678	305419896
Label	top:	ASAMPLE/top :: 00FFEC00 <label>	: 00000000	0
EQUATE	end: equ H'FFEF00	ASAMPLE/end :: <equat e>	: 00FFEF00	16772864

- Symbol value automatic modification

The symbol values displayed in the symbol value display area are automatically modified when emulation is terminated in one of the following ways.

- When the emulation initiated by clicking the <STEP>, <STEP_OVER>, <STEP_UP>, or <CONTINUE> button in the base window is terminated
- When the emulation initiated by clicking the <Step>, <Step_over>, or <Continue> button of the DISASSEMBLE window is terminated
- When the emulation initiated by the [Go...] command is terminated
- When the emulation initiated by the STEP, STEP_OVER, or GO command (emulator commands) entered in the command area is terminated
- When emulation is forcibly terminated by clicking the <STOP> button in the base window

- Symbol value modification

Symbol values in the symbol value display area can be modified directly at the hexadecimal display field and symbol display field.

- Modifications

Symbol values can be modified using the procedure shown in figure 5-67.

- (1) After moving the mouse pointer to the hexadecimal display field or symbol display field which is to be modified, click the left button and produce a cursor (|).

When performing modification through only the keyboard, press the (Tab) key and produce a cursor in the symbol value display area. Then move the cursor to the data to be modified with the arrow keys.

- (2) Enter new data from the keyboard. The cursor will move right by one digit for each new digit entered. Data cannot be entered beyond the symbol display size.

- (3) To move the cursor to the next data item to be modified, use the mouse or enter the arrow keys.

- (4) Clicking the <Enter> button actually modifies the symbol values.

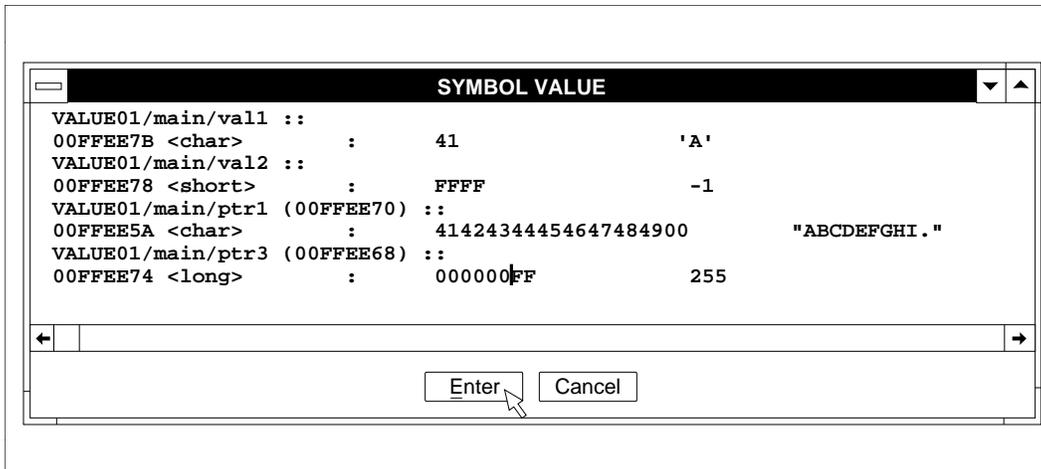


Figure 5-67 Symbol Value Modification Procedure (SYMBOL VALUE Window)

— Usable characters

- (1) Only hexadecimal data can be entered in the hexadecimal display field.

Hexadecimal characters: Numbers (0 to 9) and alphabetical characters (A to F and a to f)

- (2) Table 5-19 lists the characters which can be entered in the symbol display field.

Table 5-19 Symbol Display Field Characters (SYMBOL VALUE Window)

Usable Characters	Description
Numbers (0 to 9)	Used for decimal data
Minus (-)	Changes a value from positive to negative
Plus (+)	Changes a value from negative to positive
Space key	Deletes the decimal character located to the left of the cursor and shifts the characters located to the right of the cursor one digit to the left
All characters	All displayable characters can be entered

— Symbol value modification examples

Example 1: Modifying char pointer symbol xyz from abc to XYZ in the symbol display field

(1) Move the mouse pointer to character a and click the left button. A cursor (|) is displayed.

```
SAMPLE/main/xyz (00FFEE70) ::  
00FFEE5A <char> :      61626300      "|abc."
```

(2) Enter X from the keyboard. a is modified to X and the cursor moves to the next character on the right. At this point, the hexadecimal display field contents are automatically modified.

```
SAMPLE/main/xyz (00FFEE70) ::  
00FFEE5A <char> :      58626300      "X|bc."
```

(3) Enter Y and Z from the keyboard to perform modifications in the same way.

```
SAMPLE/main/xyz (00FFEE70) ::  
00FFEE5A <char> :      58595A00      "XYZ|. "
```

(4) Click the <Enter> button to actually write new data to memory.

Figure 5-68 Modification Example of a char Pointer Symbol (SYMBOL VALUE Window)

Example 2: Modifying long-type integer symbol abc from 120 to -4 in the symbol display field

(1) Move the mouse pointer to digit 1 and click the left button. A cursor (|) is displayed.

```
SAMPLE/main/abc ::  
00FFEE74 <long>      :      00000078          |120
```

(2) Enter a minus (-) from the keyboard. A minus sign is added to the beginning of the value. At this point, the hexadecimal display field contents are automatically modified.

```
SAMPLE/main/abc ::  
00FFEE74 <long>      :      FFFFFFF88          -|120
```

(3) Enter 4 from the keyboard. The value is modified and the cursor moves to the next character on the right.

```
SAMPLE/main/abc ::  
00FFEE74 <long>      :      FFFFFFF5C          -4|20
```

(4) Enter the (Space) key from the keyboard twice to delete 2 and 0.

```
SAMPLE/main/abc ::  
00FFEE74 <long>      :      FFFFFFFFC          -4|
```

(5) Click the <Enter> button to actually write new data to memory.

Figure 5-69 Modification Example of a long-Type Integer Symbol (SYMBOL VALUE Window)

Example 3: Modifying double floating-point symbol fdata from 1.234000e+002 to -1.234500e+005 in the symbol display field

- (1) Move the mouse pointer to digit 1 and click the left button. A cursor (|) is displayed.

```
SAMPLE/main/fdata ::  
00FFFEED8 <double>      :  405ED9999999999A    |1.234000e+002
```

- (2) Enter a minus (-) from the keyboard. A minus sign is added to the beginning of the value. At this point, the hexadecimal display field contents are automatically modified and the <Enter> button can be clicked because shadow display has been cancelled.

```
SAMPLE/main/fdata ::  
00FFFEED8 <double>      :  C05ED9999999999A    -|1.234000e+002
```

- (3) Press the right arrow key (' key) to move the cursor right by five characters.

```
SAMPLE/main/fdata ::  
00FFFEED8 <double>      :  C05ED9999999999A    -1.234|000e+002
```

- (4) Enter 5 from the keyboard. The value is modified and the cursor moves to the next character on the right.

```
SAMPLE/main/fdata ::  
00FFFEED8 <double>      :  C05EDCCCCCCCCCD    -1.2345|00e+002
```

- (5) Press the right arrow key (' key) to move the cursor right by five characters.

```
SAMPLE/main/fdata ::  
00FFFEED8 <double>      :  C05EDCCCCCCCCCD    -1.234500e+00|2
```

- (6) Enter 5 from the keyboard.

```
SAMPLE/main/fdata ::  
00FFFEED8 <double>      :  C0FE23A000000000    -1.234500e+005|
```

- (7) Click the <Enter> button to actually write new data to memory.

Figure 5-70 Modification Example of a double Floating-Point Symbol (SYMBOL VALUE Window)

Example 4: Modifying short-type two-dimensional array symbol dim from 0064 to 0000 in the hexadecimal display field

(1) Move the mouse pointer to digit 6 and click the left button. A cursor (|) is displayed.

```
SAMPLE/main/dim[0][1]  ::  
00FFEECE <short>      :  00|64   100
```

(2) Enter 0 from the keyboard. The value is modified and the cursor moves to the next character on the right. At this point, the symbol value display field contents are automatically modified.

```
SAMPLE/main/dim[0][1]  ::  
00FFEECE <short>      :  000|4    4
```

(3) Enter 0 from the keyboard.

```
SAMPLE/main/dim[0][1]  ::  
00FFEECE <short>      :  0000|   0
```

(4) Click the <Enter> button to actually write new data to memory.

Figure 5-71 Modification Example of a short-Type Two-Dimensional Array Symbol (SYMBOL VALUE Window)

Notes

- (1) If the symbol such as static variable, local variable, and register variable in the symbol value display area exceeds the scope during program execution, the following message,

*** Symbol is not SCOPE

is displayed. A symbol outside the scope during program execution and having the same name can be defined but only the symbol within the scope is displayed.

- (2) If the symbol has not been allocated or if the symbol allocation has been cancelled during program execution, the following message,

*** Symbol is not allocated

is displayed.

- (3) If the symbol in the symbol value display area is to be allocated to an illegal address, the following message
- ```
*** Symbol is ILLEGAL ADDRESS
```
- is displayed.
- (4) Symbol values for the externally defined variable symbol or external reference variable symbol are always displayed regardless of the scope. If there is a symbol having the same name within the scope, the symbol value of the symbol within the scope is displayed.
- (5) If the stack pointer (ER7) is modified during local variable display in the symbol value display area, the displayed symbol value cannot be guaranteed.
- (6) When the program is optimized by the C compiler, the local variables cannot be displayed correctly even if they are assigned to the symbol area. Therefore, care must be taken by referring to assembly instructions before modifying local variables while the program is optimized by the C compiler.
- (7) The following must be noted when specifying the symbol in the C source file optimized by the C compiler.
- If there are symbols declared with the same name inside and outside a function, information on multiple symbols may be displayed depending on the execution stop address.
  - Locations not related to the target symbol must not be referenced or modified.
- (8) The following must be noted when modifying the bit-field symbols.
- In the hexadecimal display field, the symbol value is displayed in hexadecimal using the display size indicated in table 5-16. However, only the bit-field bits within the range assigned to the bit field can be modified.
  - In the symbol display field, the symbol value corresponding to the bit-field size is displayed as decimal data. When modifying this value, decimal data having a value within the specified bit-field size must be entered.
- (9) If a symbol to be assigned requires more than 20,000 bytes, the following message
- ```
1037 : symbol assignment area too large
```
- is displayed instead of the symbol values.

(10) The display format is <pointer> when symbols that have the following attributes are displayed. Contents of the address indicated by the pointer are not displayed.

- A pointer-type symbol for a structure or union member
- A symbol that has pointer and array attributes

(11) A function-type symbol is displayed as a pointer to the function value type for the size from the program start address to the function-type symbol. Note that symbol values cannot be displayed for the void-type function-type symbols.

Related Functions

<DISPLAY>, <STEP>, <STEP_OVER>, <STEP_UP>, <CONTINUE>, and <STOP> buttons

GUI commands: [Symbol - Display], [Go...], and [Disassemble]

Emulator commands: STEP, STEP_OVER, and GO

5.6.8 Setting Source Area

(Alt + Ctrl + A)

[Source - Setting...]

SOURCE AREA SETTING dialog box

Overview

Specifies the font, font size, and tab width of the source program to be displayed in the source area.

Window

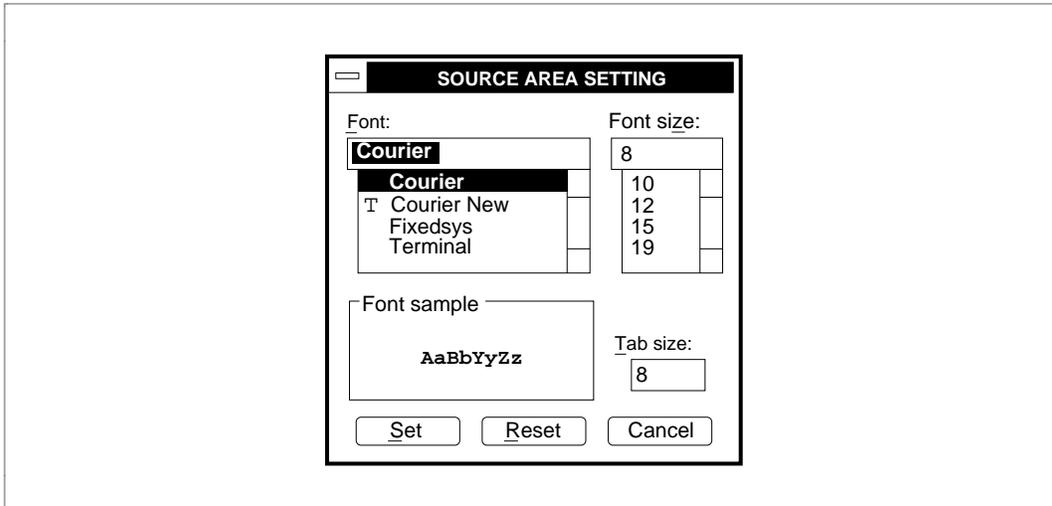


Figure 5-72 SOURCE AREA SETTING Dialog Box

Operation

- [Font:] list box
Selects a source program font to be displayed in the source area from the list box. Only the fixed-pitch fonts installed using the Windows control panel are displayed in the list box. In the GUI-2600, Courier or Fixed System font is recommended. Default at system initiation is Courier.
- [Font size:] list box
Selects the size of the font specified in [Font:] from the list box. In True-type fonts, sizes from 4 to 25 can be entered. Default at system initiation is Courier 8.
- [Font sample]
Displays an example of the font specified by [Font:] and [Font size:].

- [Tab size:] text box
Specifies 1 to 32 as the tab width of a source program to be displayed in the source area. Default at system initiation is 8.
- <Set> button
Clicking the <Set> button redisplay the source area according to the SOURCE AREA SETTING dialog box settings. Then stores the SOURCE AREA SETTING dialog box settings and closes the SOURCE AREA SETTING dialog box.
- <Reset> button
Clicking the <Reset> button redisplay the source area according to the default settings. Then closes the SOURCE AREA SETTING dialog box, and resets the settings in the SOURCE AREA SETTING dialog box.
- <Cancel> button
Clicking the <Cancel> button closes the SOURCE AREA SETTING dialog box and returns the system to its state when the SOURCE AREA SETTING dialog box was opened without storing the above settings.

Function

- Font settings
The font and font size of the source program to be displayed in the source area are specified. Use this function to change the current font settings to a desired font and font size. The modified font settings are valid until they are reset. Even if the source area display contents change according to program execution, the font and font size are not changed.
- Tab width setting
Specifies the tab width of the source program to be displayed in the source area. Use this function to change the current tab setting to a desired tab setting. The modified tab settings are valid until they are reset. Even if the source area display contents change according to program execution, the tab width is not changed.

Related Functions

[Source - Display...] command (GUI command) and source area

5.6.9 Displaying and Selecting a Source File

(Alt + A)

[Source - Display...]

SOURCE DISPLAY dialog box

Overview

Lists all source files included in the load module, selects a source file, and displays the file in the source area of the base window.

Window

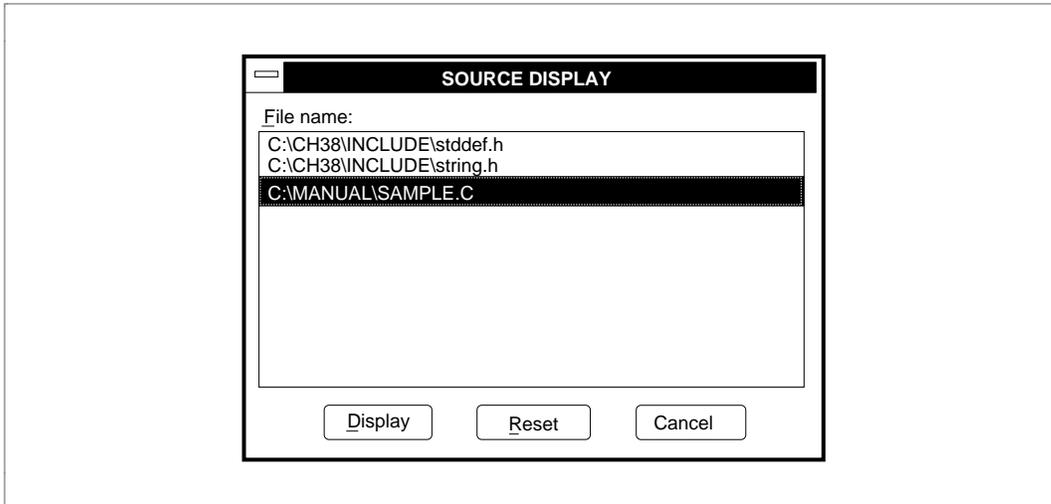


Figure 5-73 SOURCE DISPLAY Dialog Box

Operation

- [File name:] list box
Displays all source file names included in the load module. Each file name is expressed as the complete file name from the route directory.

Clicking a displayed source file with the mouse highlights the name. Click the <Display> button to display the selected source file contents in the source area of the base window.

- <Display> button
Clicking the <Display> button displays in the source area of the base window the contents of the source file selected in the source file name display area.

- <Reset> button
- Clicking the <Reset> button displays the section of the source file pointed to by the current program counter (PC).
- <Cancel> button
Clicking the <Cancel> button closes the SOURCE DISPLAY dialog box.

Function

The source file name display area displays the following type of files:

- C source files
- Assembly source files
- Include files (such as header files)

Related Functions

GUI command: [Source - Setting...]

Source area

5.6.10 Disassembling and Executing Programs

(Alt + D)

[Disassemble]

DISASSEMBLE window

Overview

Disassembles and displays the source file displayed in the source area of the base window, and enables assembly-language level program debugging.

Window

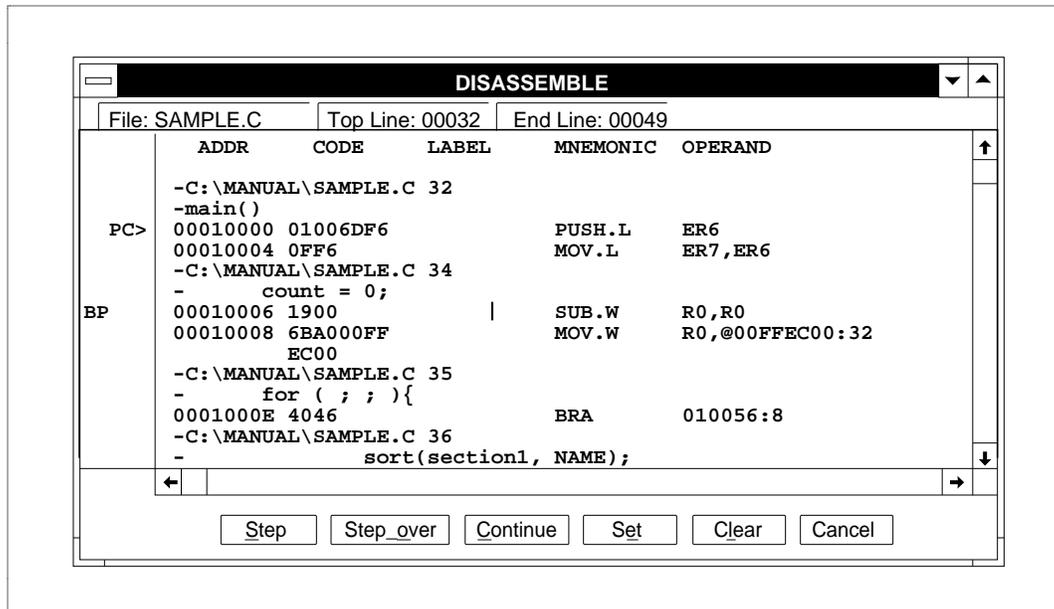


Figure 5-74 DISASSEMBLE Window

Operation

- Source file information area
Displays the contents of the source information area of the base window.
 - [File:]
Displays the name of the file displayed in the source area of the base window.
 - [Top line:]
Displays the first line number displayed in the source area.

— [End line:]

Displays the last line number displayed in the source area.

- Disassemble display area

Displays disassembly results for the section of the source file displayed in the source area of the base window. The headers have the following meanings:

ADDR: Address

CODE: Instruction code

LABEL: Label

MNEMONIC: Instruction mnemonic

OPERAND: Instruction operand

- Program counter display area

Displays the PC mark (PC>) indicating the address pointed to by the current program counter (PC).

- <Step> button

Clicking the <Step> button executes one program line pointed to by the current PC and then stops. When the line includes a subroutine call, program execution stops at the first line of the subroutine.

- <Step_over> button

Clicking the <Step_over> button executes one program line pointed to by the current PC and then stops. When the line includes a subroutine call, program execution stops after executing the subroutine.

- <Continue> button

Clicking the <Continue> button executes the program from the line pointed to by the current PC. Execution continues until a line including the cursor in the source area is reached.

- <Set> button

Clicking the <Set> button sets a breakpoint to an instruction which is pointed to by the cursor.

- <Clear> button

Clicking the <Clear> button clears a breakpoint set at an instruction which is pointed to by the cursor.

- <Cancel> button
Clicking the <Cancel> button closes the DISASSEMBLE window.

Function

- Display of source file disassembly
The information displayed in the disassemble display area depends on the type of the file in the source area, as shown in table 5-20.

Table 5-20 File Type and Disassembly Display (DISASSEMBLE Window)

Source File Type	Disassemble Display Area	Source File Information Area
C source file, Assembly source file	Disassembles and displays the section displayed in the source area	File name, first line number, last line number
None	Disassembles and displays 16 instructions from the current program counter (PC)	None

- Automatic modification of disassemble display area
The contents of the disassemble display area are automatically modified when emulation is terminated in one of the following ways:
 - When the emulation initiated by clicking the <STEP>, <STEP_OVER>, <STEP_UP>, or <CONTINUE> button in the base window is terminated
 - When the emulation initiated by clicking the <Step>, <Step_over>, or <Continue> button in the DISASSEMBLE window is terminated.
 - When the emulation initiated by the [Go...] command is terminated
 - When the emulation initiated by the STEP, STEP_OVER, or GO command (emulator commands) entered in the command area is terminated
 - When emulation is forcibly terminated by clicking the <STOP> button in the base window
- Assembly-language-level debugging function
Enables program debugging at the assembly-language level with the buttons in the DISASSEMBLE window. When a source program is displayed in the source area, the source program in the assembly-language level is displayed in the source area in realtime during assembly-language-level program execution.

If a breakpoint is set in the DISASSEMBLE window, a BP mark is also displayed on the corresponding program line in the source area. If a breakpoint is set to a line which is not the first line of the program using the DISASSEMBLE window, the breakpoint cannot be cancelled with the <CLEAR> button in the base window. The breakpoint must be cleared using the <Clear> button in the DISASSEMBLE window or the [Break...] command.

If multiple breakpoints are specified on a line, a BP mark on the source line is not cleared until all breakpoints specified on that line are cancelled.

Note

When the execution stop address is not in the program address space, the contents of the source area and source information area do not match.

Related Functions

<STEP>, <STEP_OVER>, <STEP_UP>, <CONTINUE>, and <STOP> buttons

GUI command: [Go...]

Emulator commands: STEP, STEP_OVER, and GO

5.6.11 Displaying Function Call Sequence

(Alt + U)

[Route]

ROUTE window

Overview

Displays the function call sequence up to the function pointed to by the current program counter (PC).

Window

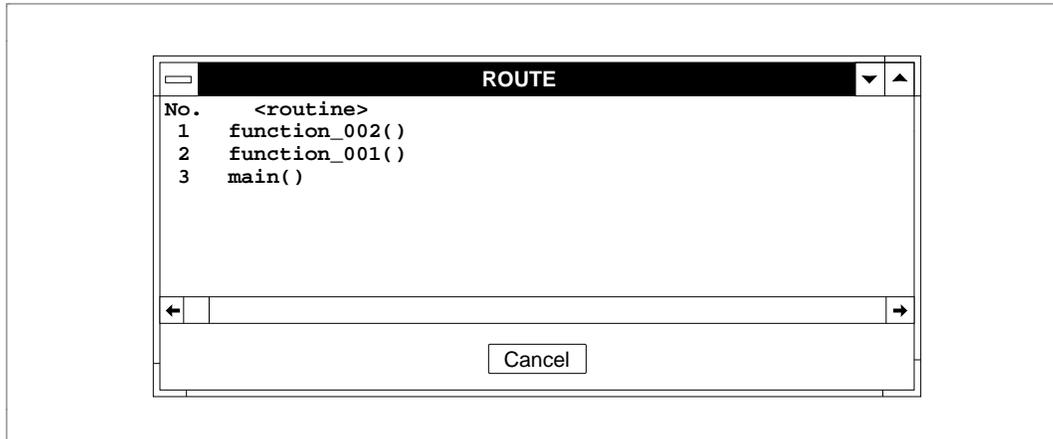


Figure 5-75 ROUTE Window

Operation

- Function call sequence display area
Displays the function call sequence in terms of the number and function name up to the function pointed to by the current program counter (PC). Up to 64 functions can be displayed.
- <Cancel> button
Clicking the <Cancel> button closes the ROUTE window.

Function

- Function call sequence display

Displays the function call sequence from the function pointed to by the current PC to the function at the initial call. The [Route] command is effective only for C programs.

However, if a return address of a function in this sequence does not point to another function, or if a function call in this sequence is written in assembly language, the call sequence is displayed from the function pointed to by the PC up to the last function that can be detected.

- Automatic function call sequence modification

While the ROUTE window is open, information in the function call sequence display area is modified automatically when emulation is terminated in one of the following ways.

- When the emulation initiated by pressing the <STEP>, <STEP_OVER>, <STEP_UP>, or <CONTINUE> button is terminated
- When the emulation initiated by clicking the <Step>, <Step_over>, or <Continue> button in the DISASSEMBLE window is terminated
- When the emulation initiated by the [Go...] command is terminated
- When the emulation initiated by the STEP, STEP_OVER, or GO command (emulator commands) entered in the command area is terminated
- When emulation is forcibly terminated by clicking the <STOP> button in the base window

Notes

- (1) The function call sequence is not displayed correctly if the program counter (PC) or stack pointer (ER7) value is modified after execution is terminated.
- (2) The function call sequence is not displayed correctly if an interrupt function is being called.
- (3) The SYSCR (system control) register address must be set in the initial setting file (gui26.ini) to display the function call sequence. If the SYSCR register address is not set, the function call sequence may not be displayed correctly. For details on setting the SYSCR register address, refer to appendix D, Initial Setting File (gui26.ini).

Related Functions

<STEP>, <STEP_OVER>, <STEP_UP>, <CONTINUE>, and <STOP> buttons

GUI commands: [Go...] and [Disassemble]

Emulator commands: STEP, STEP_OVER, and GO

5.6.12 Setting LED Display Conditions

(Alt + Ctrl + Y)

[LED setting...]

LED SETTING dialog box

Overview

Specifies conditions for displaying memory contents on the LEDs on the optional bus monitor board connected to the E7000PC station. When memory is accessed by a user program under the specified conditions, the data in memory is displayed on the LEDs.

Window

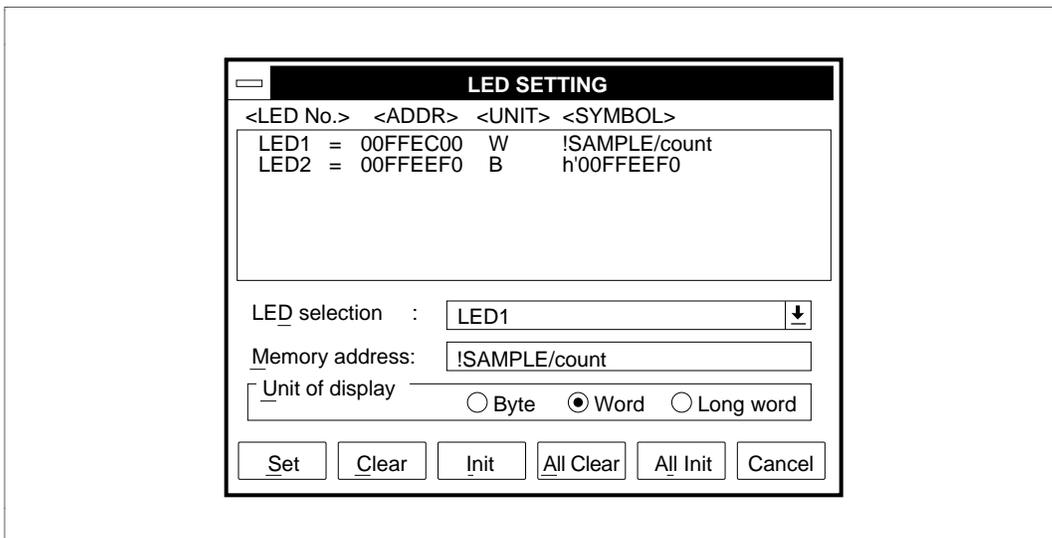


Figure 5-76 LED SETTING Dialog Box

Operation

- LED information display area

Displays conditions for displaying memory contents on each LED as shown in figure 5-77.

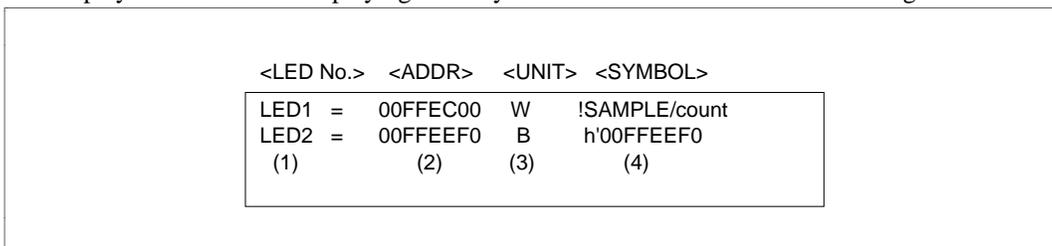


Figure 5-77 LED Information Display Format (LED SETTING Dialog Box)

- (1) <LED No.>: LED number
- (2) <ADDR>: Memory address whose contents are to be displayed on the LED
- (3) <UNIT>: Display data size
 - B: 1-byte units
 - W: 2-byte units
 - L: 4-byte units
- (4) <SYMBOL>: Symbol name of the memory address (if no symbol has been assigned, the memory address will be displayed in hexadecimal)

Clicking a displayed LED information line with the mouse highlights the line.

- [LED selection:] drop-down list box
Specifies the LED number on which memory contents are to be displayed, as follows:
 - <LED1>: Displayed on LED1 (default at system initiation)
 - <LED2>: Displayed on LED2
 - <LED3>: Displayed on LED3
 - <LED4>: Displayed on LED4
- [Memory address:] text box
Specifies a number or a symbol as the memory address whose contents are to be displayed.
- [Unit of display] group box
Specifies the display unit, as follows:
 - <Byte>: Display in 1-byte units (default at system initiation)
 - <Word>: Display in 2-byte units
 - <Long word>: Display in 4-byte units
- <Set> button
Clicking the <Set> button specifies the LED display conditions entered in [LED selection:], [Memory address:], and [Unit of display] and displays the conditions in the LED information display area.
- <Clear> button
Clicking the <Clear> button cancels the conditions for the LED selected in the LED information display area. The LED information display for the selected LED disappears from the LED information display area when the conditions are cancelled.

- <Init> button
Clicking the <Init> button turns off the LED selected in the LED information display area.
- <All Clear> button
Clicking the <All Clear> button cancels the conditions for all LEDs displayed in the LED information display area. The LED information display for all LEDs disappears from the LED information display area when the conditions are cancelled.
- <All Init> button
Clicking the <All Init> button turns off all LEDs.
- <Cancel> button
Clicking the <Cancel> button closes the LED SETTING dialog box.

Function

- LED display condition specifications
Specifies conditions for displaying memory contents on the LEDs on the optional bus monitor board connected to the E7000PC station. When memory is accessed by a user program under the specified conditions, the data in memory is displayed on the LEDs. Up to four addresses can be specified, for each of which 1-byte, 2-byte, or 4-byte data can be displayed.

When specifying <Word> or <Long> in the [Unit of display] group box, specify in the [Memory address:] text box an address that is a multiple of two or four bytes. If an odd address is specified, the [Unit of display] group box automatically changes to <Byte>. If an address indicating a multiple of two bytes is specified when <Long> is specified, the [Unit of display] group box automatically changes to <Word>.

When the conditions are set with the <Set> button, the LEDs do not turn on. Only when memory is accessed by a user program under the specified conditions, the LEDs turn on and are updated. When memory is accessed or changed by commands such as the [Memory - Dump] command, the LEDs do not turn on.

- LED initialization
Clicking the <Init> or <All Init> button turns off the specified LEDs. The LEDs turn on when the specified addresses are accessed again.

- LED display condition cancellation
Clicking the <Clear> or <All Clear> button cancels the specified LED display conditions.
When the LED display conditions are cancelled, the LED information display in the LED information display area disappears.

Related Function

GUI command: [LED out setting...]

5.6.13 Setting Conditions for Analog Output of LED Display Contents

(Alt + Y)

[LED out setting...]

LED OUT SETTING dialog box

Overview

Specifies conditions for outputting the data displayed on LED1 and LED2 on the optional bus monitor board connected to the E7000PC station from the analog output pins (LED_OUT1 and LED_OUT2) on the optional bus monitor board.

Window

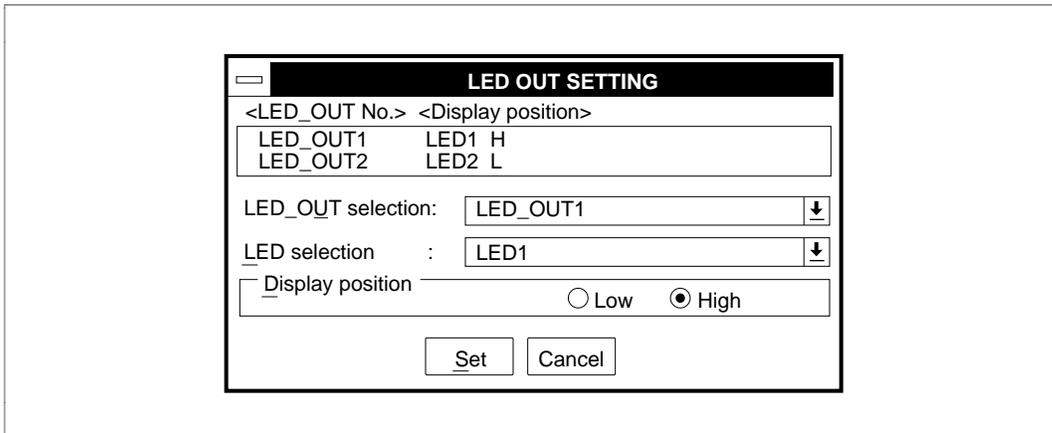


Figure 5-78 LED OUT SETTING Dialog Box

Operation

- LED_OUT information display area
Displays settings for outputting data displayed on the LEDs from the analog output pins on the optional monitor board, as shown in figure 5-79.

<LED_OUT No.>	<Display position>
LED_OUT1	LED1 H
LED_OUT2	LED2 L
(1)	(2)

Figure 5-79 LED_OUT Information Display Format (LED OUT SETTING Dialog Box)

(1) <LED_OUT No.>: Analog output pin number from which data is output

(2) <Display position>: LED number and position of data to be output

H: High-order 16 bits of displayed data

L: Low-order 16 bits of displayed data

- [LED_OUT selection:] drop-down list box
Specifies an analog output pin number to output data, as follows:
<LED_OUT1>: Outputs to the LED_OUT1 pin (default at system initiation)
<LED_OUT2>: Outputs to the LED_OUT2 pin
- [LED selection:] drop-down list box
Specifies an LED number whose display data is to be output from the analog output pin, as follows:
<LED1>: LED1 is to be output (default at system initiation)
<LED2>: LED2 is to be output
- [Display position] group box
Specifies the position of data to be output from the analog output pin, as follows:
<High>: High-order 16 bits of displayed data
<Low>: Low-order 16 bits of displayed data (default at system initiation)
- <Set> button
Clicking the <Set> button specifies the analog output conditions entered in [LED_OUT selection:], [LED selection:], and [Display position] and displays all the analog output conditions in the LED_OUT information display area.
- <Cancel> button
Clicking the <Cancel> button closes the LED OUT SETTING dialog box.

Function

Specifies conditions for outputting data displayed on LED1 or LED2 on the optional bus monitor board connected to the E7000PC station from two analog output pins (LED_OUT1 and LED_OUT2) on the optional bus monitor board. High-order or low-order 16 bits of data displayed on LED1 or LED2 can be selected for output.

If an LED other than those specified with the [LED setting...] command is selected, data output from the analog output pin becomes invalid.

Related Function

GUI command: [LED setting...]

5.6.14 Controlling Command Area

(Ctrl + F6)

[Command area]

Overview

Modifies the command area size of the base window.

Window

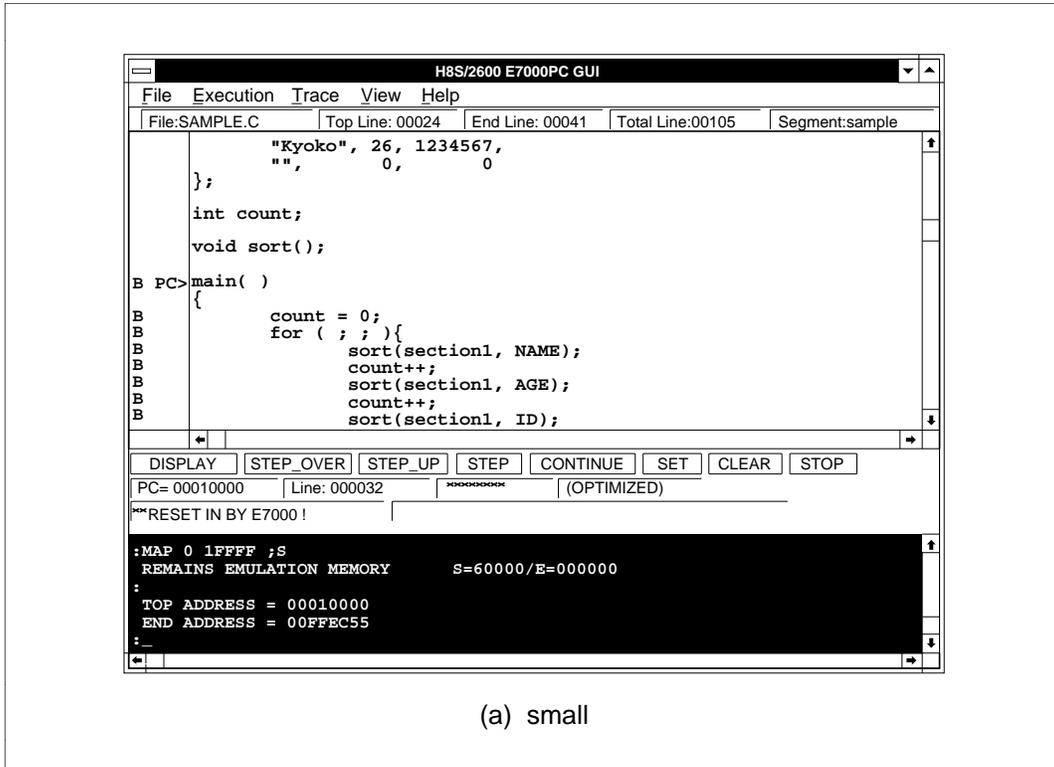


Figure 5-80 Command Area Display

Operation

- Command area size modification

The size of the command area in the base window can be modified to one of the following three sizes: (small), (full), or (none). The command area size cycles in the order of (small), (full), and (none) every time the [Command area] command is executed. The current command area size is displayed on the menu in the command area of the [View] menu.

(a) small

Default size at system initiation. Both the source area and command area are displayed in the base window.

(b)full

The largest sized area. The source area is closed and is not assigned to the base window. This size is effective when using emulator commands.

(c)none

The command area is closed and the source area is enlarged in the base window. This size is effective in maximizing the acquisition of source file information, accomplished by closing the command area while emulator commands are not used.

5.7 Help Functions

5.7.1 Displaying GUI-2600 Operating Help

(F1)

[GUI operating help]

GUI OPERATING HELP Window

Overview

Displays the GUI basic operations.

Window

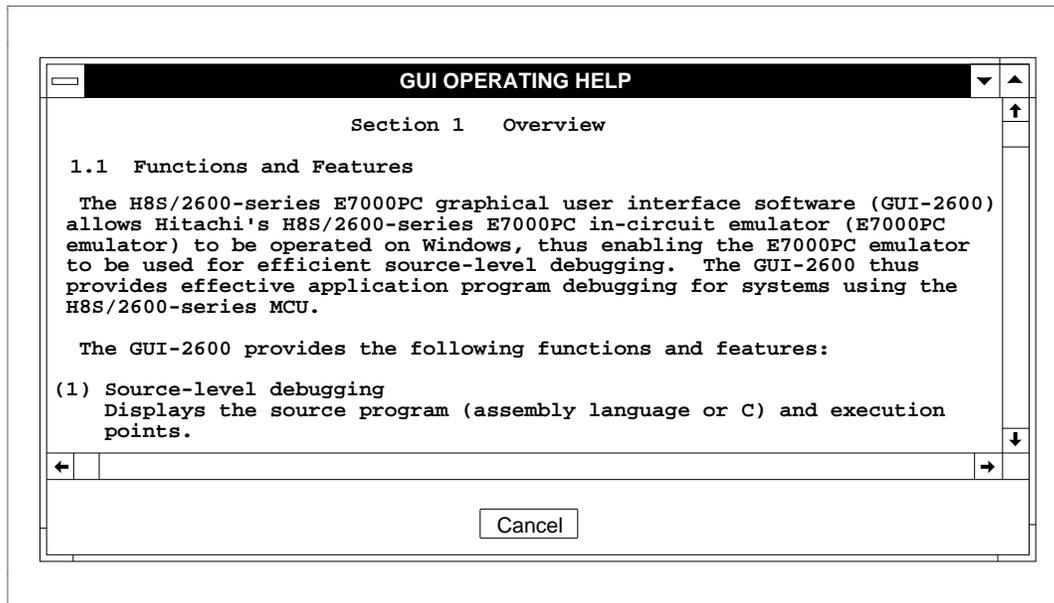


Figure 5-81 GUI OPERATING HELP Window

Operation

- Help display area
Displays the GUI basic operations.
- <Cancel> button
Clicking the <Cancel> button closes the GUI OPERATING HELP window.

5.7.2 Displaying Emulator Commands

(Alt + F1)

[EMULATOR command display] EMULATOR COMMAND DISPLAY Window

Overview

Displays emulator commands. Also displays help information on the specified command.

Window

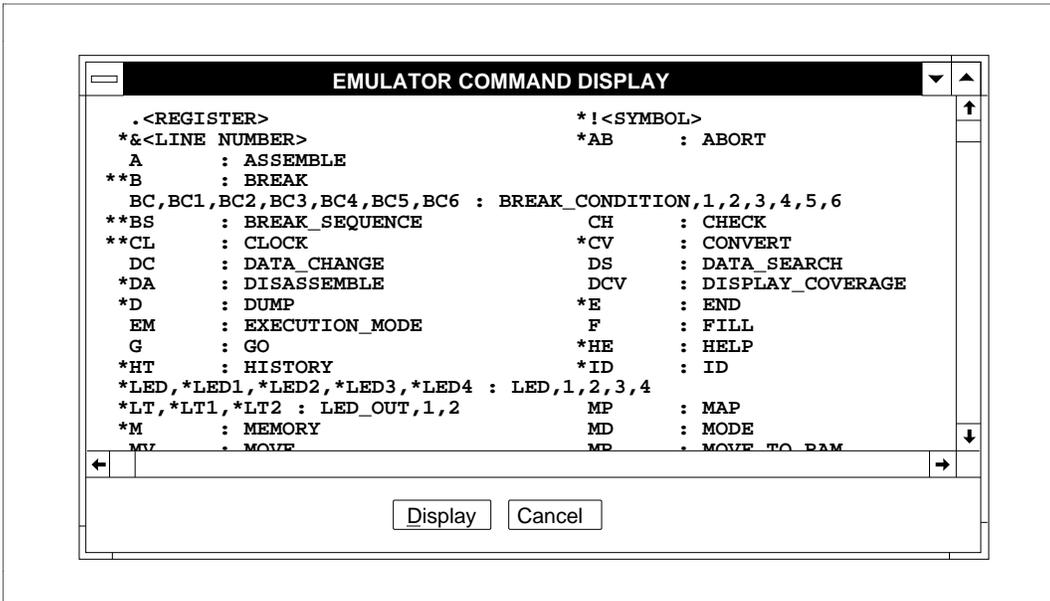


Figure 5-82 EMULATOR COMMAND DISPLAY Window

Operation

- Emulator command display area
Displays E7000PC emulator commands. Double-clicking the name or the abbreviation of a displayed emulator command highlights that command.
- <Display> button
Clicking the <Display> button opens the EMULATOR COMMAND HELP window and displays the command input format of the command name selected in the emulator command display area.
- <Cancel> button
Clicking the <Cancel> button closes the EMULATOR COMMAND DISPLAY window.

Function

- Emulator command display

Both the full command name and abbreviation are displayed for each command. Asterisks located before the abbreviation have the following meanings:

*: Can be executed in parallel mode

** : Can be displayed in parallel mode

No *: Cannot be executed in parallel mode

- Emulator command input format display

Input formats of emulator commands are displayed in the EMULATOR COMMAND HELP window shown in figure 5-83.

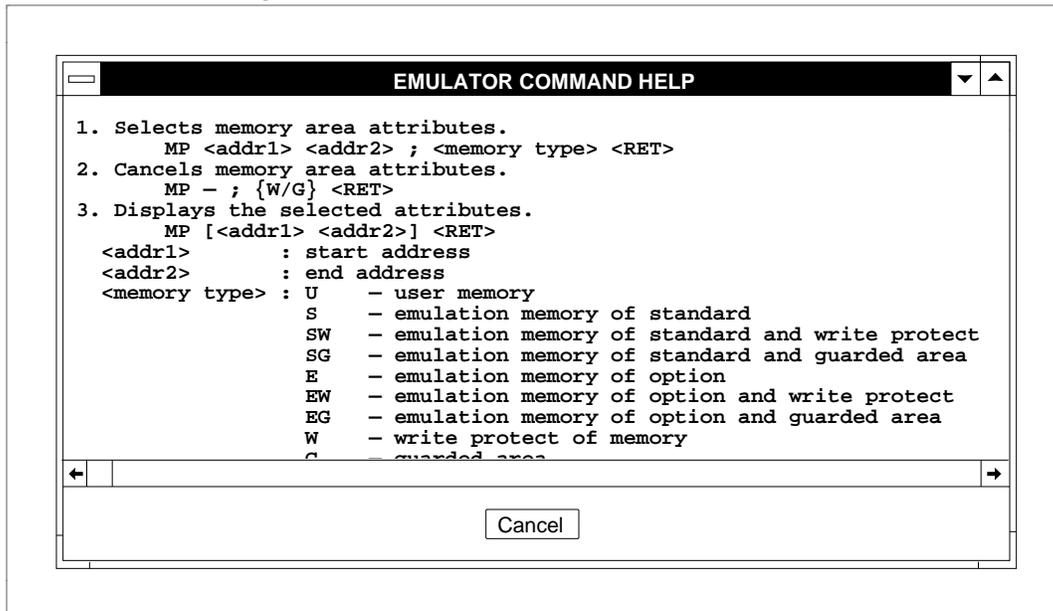


Figure 5-83 EMULATOR COMMAND HELP Window

— Help display area

Displays the input format of the E7000PC emulator command specified in the EMULATOR COMMAND DISPLAY window.

— <Cancel> button

Clicking the <Cancel> button closes the EMULATOR COMMAND HELP window.

5.7.3 Describing GUI-2600

(None)

[About ...]

ABOUT H8S/2600 E7000PC GUI Dialog Box

Overview

Displays information on the GUI-2600.

Window

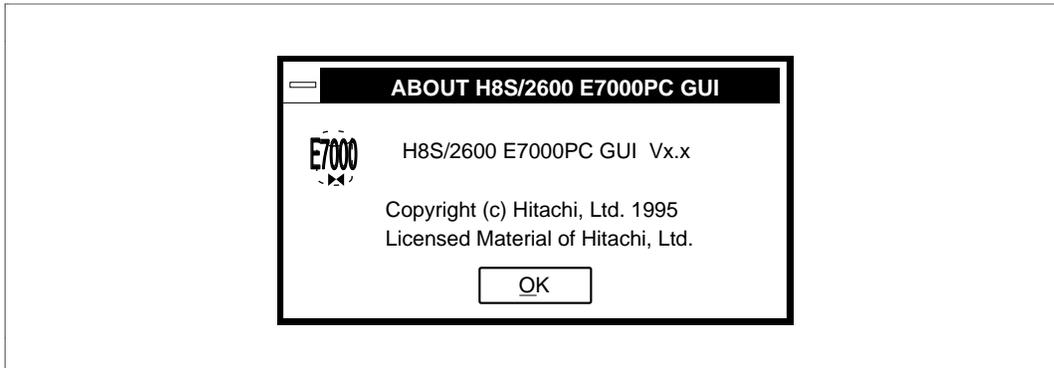


Figure 5-84 ABOUT H8S/2600 E7000PC GUI Dialog Box

Operation

- GUI-2600 information display area
Displays the product name, version, copyright, and license of the GUI-2600.
- <OK> button
Clicking the <OK> button closes the ABOUT H8S/2600 E7000PC GUI dialog box.

Appendix A E7000 Emulator Command List

Table A-1 lists the E7000 emulator commands and shows which commands can be executed in the command area.

Table A-1 E7000 Emulator Commands

Command	Function	Usable/ Unusable in Command Area
.<register>	Modifies and displays register contents	Usable
!<symbol> or &<symbol>	Displays symbol value	Usable
ABORT	Terminates emulation in parallel mode	Usable
ASSEMBLE	Assembles program one line at a time	Usable
BACKGROUND_ INTERRUPT	Sets and displays user interrupts at command input wait state	Usable
BREAK	Sets, displays, and cancels software breakpoints	Usable
BREAK_ CONDITION1,2,3,4,5,6	Sets, displays, and cancels hardware break conditions	Usable
BREAK_ SEQUENCE	Sets, displays, and cancels software breakpoints with pass sequence specification	Usable
CHECK	Tests H8S/2655-series pin status	Usable
CLOCK	Sets and displays clock	Usable
COMMAND_CHAIN	Inputs E7000 commands from a file	Usable in INPUT command
CONVERT	Converts data	Usable
DATA_CHANGE	Replaces memory data	Usable
DATA_SEARCH	Searches for memory data	Usable
DISASSEMBLE	Disassembles and displays memory contents	Usable
DISPLAY_ COVERAGE	Displays coverage trace results	Usable
DUMP	Displays memory contents	Usable
END	Cancels parallel mode	Usable
EXECUTION_ MODE	Specifies and displays execution mode	Usable
FILL	Writes data to memory	Usable
GO	Executes realtime emulation	Usable

Table A-1 E7000 Emulator Commands (cont)

Command	Function	Usable/ Unusable in Command Area
HELP	Displays all commands and command format	Usable
HISTORY	Displays all commands input	Unusable
ID	Displays E7000 system program version	Usable
LED1,2,3,4	Specifies, displays, and cancels memory content display on LEDs	Usable
LED_OUT1,2	Specifies and displays analog output of LED display data	Usable
MAP	Specifies and displays memory attributes	Usable
MEMORY	Displays and modifies memory contents	Usable
MODE	Specifies and displays the H8S/2655-series operating mode	Usable
MOVE	Transfers memory contents	Usable
MOVE_TO_RAM	Moves ROM contents to emulation memory	Usable
PERFORMANCE_ANALYSIS	Specifies, cancels, initializes, and displays performance measurement data	Usable
PRINT	Sets or cancels output device for command result display	Unusable in E7000PC
QUIT	Terminates E7000 system program	Usable
RADIX	Specifies and displays radix for numeric input	Usable
REGISTER	Displays register contents	Usable
RESET	Resets H8S/2655-series MCU	Usable
RESULT	Displays execution results	Usable
SET_COVERAGE	Initializes the coverage trace function	Usable
SHORT_SYMBOL	Defines a short format for a symbol and displays current symbol definition	Unusable
STATUS	Displays E7000 execution status	Usable
STEP	Performs single-step execution	Usable
STEP_INFORMATION	Specifies and displays information during single-step execution	Usable

Table A-1 E7000 Emulator Commands (cont)

Command	Function	Usable/ Unusable in Command Area
STEP_OVER	Performs single-step execution except for subroutines	Usable
SYMBOL	Defines, displays, and deletes symbols	Definition function is unusable
TRACE	Displays trace buffer contents	Usable
TRACE_CONDITION	Specifies, displays, and cancels trace conditions	Usable
TRACE_MEMORY	Specifies, displays, and cancels trace data address	Usable
TRACE_MODE	Specifies and displays trace acquisition mode	Usable
TRACE_SEARCH	Searches for and displays trace information	Usable
HOST	Specifies and displays host computer interface parameters	Unusable in E7000PC
LOAD	Loads program from host computer —Transparent mode and local mode	Usable
SAVE	Saves program in host computer —Transparent mode and local mode	Usable
TERMINAL	Transfers to terminal mode —Transparent mode	Unusable in E7000PC
TRANSFER	Transfers file to and from host computer —Transparent mode and local mode	Unusable in E7000PC
VERIFY	Verifies memory contents against host computer file —Transparent mode and local mode	Usable
INTFC_LOAD	Loads program from host computer —Remote mode	Unusable in E7000PC
INTFC_SAVE	Saves program in host computer —Remote mode	Unusable in E7000PC
INTFC_TRANSFER	Transfers file to and from host computer —Remote mode	Unusable in E7000PC
INTFC_VERIFY	Verifies memory contents against host computer file —Remote mode	Unusable in E7000PC
ASC	Specifies the transferred file type as ASCII	Unusable in E7000PC

Table A-1 E7000 Emulator Commands (cont)

Command	Function	Usable/ Unusable in Command Area
BIN	Specifies the transferred file type as binary	Unusable in E7000PC
BYE	Terminates the FTP interface (Reconnection is performed by the OPEN command)	Unusable in E7000PC
CD	Modifies the file directory name of the FTP server	Unusable in E7000PC
CLOSE	Disconnects the FTP interface(Reconnection is performed by the OPEN command)	Unusable in E7000PC
FTP	Connects via the FTP interface	Unusable in E7000PC
LAN	Displays E7000 IP address	Unusable in E7000PC
LAN_HOST	Specifies, modifies, and displays the IP address of the host computer connected via the FTP interface	Unusable in E7000PC
LAN_LOAD	Loads a program from the host computer via the FTP interface	Unusable in E7000PC
LAN_SAVE	Saves a program in the host computer via the FTP interface	Unusable in E7000PC
LAN_TRANSFER	Transfers a file to and from the host computer connected via the FTP interface	Unusable in E7000PC
LAN_VERIFY	Verifies user system memory contents against the host computer file connected via the FTP interface	Unusable in E7000PC
LS	Displays the host computer directory connected via the FTP interface	Unusable in E7000PC
OPEN	Connects via the FTP interface	Unusable in E7000PC
PWD	Displays the current directory name of the host computer connected via the FTP interface	Unusable in E7000PC
STA	Displays the file type to be transferred	Unusable in E7000PC
LOGOUT	Disconnects from the Telnet	Unusable in E7000PC

CAUTION

If the file specified with the SAVE command already exists,
the existing file will be automatically overwritten.

Notes:

- The following E7000 monitor commands cannot be input from the command area:
 - L (Sets an E7000 IP address)
 - T (Loads and initiates the diagnostic program)
- It is not possible to re-display previously input command parameters by entering <command name>.
- Control codes other than (Ctrl + C) cannot be input from the command area. Therefore, display cannot be controlled (display stop or restart) using control codes.
- With the LOAD command, a GUI-2600 option can be specified in addition to the emulator function. Figure A-1 shows the command syntax including the GUI-2600 option specification.

LOAD [<offset>][;<load module type>]:<file name>[Δ<GUI-2600 option>](Enter)

<GUI-2600 option>: Option usable only in the GUI-2600 when SYSROF (R) is specified as load module type.

A:	Loads debugging information of the load module file to the GUI-2600 and loads the load module file to the E7000 (default).
E:	Does not load debugging information of the load module file to the GUI-2600 and loads the load module file to the E7000.
G:	Loads debugging information of the load module file to the GUI-2600 and does not load the load module file to the E7000.

Figure A-1 LOAD Command Syntax

For details on <offset>, <load module type>, and <file name>, refer to the Emulator User's Manual.

- The command file can be executed from the command area using the INPUT command. Figure A-2 shows the command syntax of the INPUT command.

```
INPUT<file name> (Enter)
```

<file name>: The name of a command file to be executed in the command area.

Figure A-2 INPUT Command Syntax

If the INPUT command is specified within the command file, the INPUT command is ignored and command file execution continues.

Appendix B Error Messages

This section describes error messages displayed in the message box when an error occurs during GUI-2600 operation or E7000PC emulator operation.

B.1 Error Message Levels

GUI-2600 error messages are grouped into two types as follows:

- 1000 – : Operation error messages
Displayed during GUI-2600 operation
- 2000 – : System error messages
Displayed when an error occurs in the system under which the
GUI-2600 is operating

B.2 List of Error Messages

This section describes error messages in detail. Each error message is described in the following format.

Error Message Description Format

Error No.	Error Message	Error Description (P) Solution
-----------	---------------	-----------------------------------

B.2.1 Operation Error Messages

Table B-1 Operation Error Messages

Error No.	Error Message	Error Description and Solution
1001	emulator kind unknown	The E7000PC emulator connected to the GUI-2600 is not designated for the H8S/2655-series MCU. (P) Connect a correct E7000PC emulator to the GUI-2600.
1002	file name not specified	A file name is not specified. (P) Specify a file name.
1003	file name too long	The specified file name is too long. (P) Respecify the file name within the specified length.
1004	file not found	The specified file could not be found (P) Specify a correct file name.
1005	file already exists. Overwrite ? <file name>	The specified file already exists. (P) Specify whether the file specified is to be overwritten or not.
1008	load module target error	The file specified in the LOAD dialog box is not an H8S/2655-series load module. (P) Specify a correct file name.
1009	debug information too many	The number of defined symbols exceeds the limit (16,777,215) that can be defined in the GUI-2600. (P) Reduce the number of defined symbols to within the limit.
1010	source file updated <file name>	Source file modification date of a file displayed in the source area of the base window is later than the load module creation date. (P) Regenerate a load module for the modified source file.
1011	symbol name not specified	No symbol is specified. (P) Specify a symbol.
1013	symbol not found	The specified symbol could not be found. (P) Specify a correct symbol.
1016	address not specified	No address is specified. (P) Specify an address.
1017	value not specified	No data is specified. (P) Specify data.
1019	condition not specified	No condition is specified. (P) Specify conditions.
1021	command line too long	The entered command line is too long. (P) Enter and specify the command line within 248 characters.

Table B-1 Operation Error Messages (cont)

Error No.	Error Message	Error Description and Solution
1022	can not use command in E7000 GUI	The command cannot be entered from the command area. (P) This command must not be entered from the command area.
1023	invalid operation in parallel mode	The operation attempted is not supported during emulation or in parallel mode. (P) The operation cannot be executed during emulation or in parallel mode. Stop E7000PC emulator operation (change the prompt to ":" from the command area).
1024	syntax error	An input or operation error has occurred. (P) Enter commands or other information correctly as specified in the manuals.
1025	invalid value	Invalid data is entered. (P) Enter correct data.
1026	invalid address	An invalid address is entered. (P) Enter a correct address.
1027	invalid token in command	The command syntax is invalid. (P) Enter a command with the correct syntax from the command area.
1029	modified data too much	Memory and symbol modification in the MEMORY DUMP and SYMBOL VALUE windows exceed 256. (P) Click the <Enter> button and start entering again from the 257th data item from which data has not been saved.
1030	can not routing	Function call sequence analysis cannot be performed in the ROUTE window. (P) Check the restrictions on function call sequence analysis described in section 5.6.11, Displaying Function Call Sequence.
1031	can not continue executing	E7000PC emulation operation is aborted due to an error in the E7000PC emulator. (P) Check program execution status.
1034	can not cancel	The window operation cannot be terminated. (P) If this error has occurred in the OUTPUT TO dialog box, click the <End> button to terminate file output.
1035	debug information not included	The load module's debugging information is not loaded in the GUI-2600. Thus, the SOURCE DISPLAY dialog box and ROUTE window cannot be used. (P) Reload the load module including debugging information.

Table B-1 Operation Error Messages (cont)

Error No.	Error Message	Error Description and Solution
1036	too many break point	The number of breakpoints specified in the BREAK dialog box exceeds 255. (P) Cancel a breakpoint already set to set the breakpoint desired.
1037	symbol assignment area too large	The assignment size of a symbol displayed in the SYMBOL VALUE window exceeds 20,000 bytes. (P) Reduce symbol assignment size to 20,000 bytes or less.
1038	invalid operation in sub-command mode	The operation cannot be performed when the prompt (?) is displayed in the command area. (P) Change the E7000PC emulator operating state (change the prompt from "?" to ".:") from the command area.
1043	symbol name already exists	The specified symbol name has already been defined in the SYMBOL VALUE window. (P) To display existing symbol, close the SYMBOL VALUE window once. Re-open the SYMBOL VALUE window and select the symbol.
1046	internal management of E7000 GUI is executed. Please waiting...	GUI-2600 internal processing is being executed. (P) Wait for completion of GUI-2600 internal processing.
1047	effective symbol not found	The specified or input symbol is not defined or the specified symbol assignment area is a register. (P) The symbol cannot be entered or specified. Specify a value or another symbol.
1048	can not enter for emulation executing	A character is entered from the command area during E7000PC emulator execution. (P) To enter a character during E7000PC emulator execution, set the E7000PC emulator to parallel mode via the GUI.
1049	too many display symbol	The number of symbols displayed in the SYMBOL VALUE window exceeds 16 (maximum number). (P) No more symbols can be displayed. Terminate the SYMBOL VALUE window and redefine the symbol.
1051	create file error	The text in the window cannot be output. (P) The disk has no space left or is write-protected. Check the disk.
1052	command file open error	The command file cannot be opened. (P) Specify a correct command file.

Table B-1 Operation Error Messages (cont)

Error No.	Error Message	Error Description and Solution
1053	logging file open error	The logging file cannot be opened. (P) The disk has no space left or is write-protected. Check the disk.
1054	nothing with data	The specified window has no data to output to the file. (P) Specify a window that contains text data.
1055	too many watch point	The number of watch points set in the WATCH SETTING dialog box exceeds eight (maximum number). (P) Cancel some watch points to set new watch points.
1057	double definition	The specified watch point has already been set. [P] Set another watch point in the WATCH SETTING dialog box.
1058	can not step up	The <STEP_UP> button cannot be used when the current program counter (PC) is pointing to at this address. (P) Move the PC to an address where the <STEP_UP> button can be used. The <STEP_UP> button cannot be used for a ROM program.
1059	can not use this command in command file	The INPUT command was entered in a command file. (P) Do not enter the INPUT command in a command file.
1062	can not display this symbol value	The selected symbol cannot be displayed in the SYMBOL VALUE window. (P) Refer to notes in section 5.6.7, Displaying and Modifying Symbol Values, for details on undisplayable symbols.
1063	debug information file unmatched	The debugging information file does not match the load module. (P) Recreate the load module and a matching debugging information file.

B.2.2 System Error Messages

Table B-2 System Error Messages

Error No.	Error Message	Error Description and Solution
2003	can not create socket	A system error has occurred. (P) Check system operation.
2006	file open error <file name>	The specified file cannot be opened. (P) A system error may have occurred. Check system operation.
2007	file read error <file name>	The specified file cannot be read. (P) A system error may have occurred. Check system operation.
2008	file write error <file name>	The specified file cannot be written to a disk. (P) The disk has no space left to write the file or a system error has occurred. Open up enough space for the file or check system operation.
2009	file close error <file name>	The specified file cannot be closed. (P) The disk has no space left to close the file or a system error has occurred. Open up enough space for the file or check system operation.
2010	can not get memory	There is not enough memory space to store symbol information during system operation. (P) Reduce the amount of debugging information for the load module or increase memory space.
2012	internal error	An internal error has occurred. (P) A system error may have occurred. Check system operation.
2013	allocate memory error	There is not enough memory for system operation. (P) Terminate other concurrent application(s) or increase the memory space available for Windows.
2014	source file open error	The source file corresponding to the load module cannot be opened. (P) Either the load module is damaged or the source program has been moved to another directory since the load module was generated. Regenerate the load module.
2015	command file I/O error	An I/O error has occurred during command file input. (P) The command file may be damaged. Check the command file.

Table B-2 System Error Messages (cont)

Error No.	Error Message	Error Description and Solution
2016	logging file I/O error	An I/O error has occurred during logging file output. (P) The disk has no space left or is write-protected. Check the disk.
2017	interface error	An interface error has occurred between the personal computer and the E7000PC emulator. (P) Check the interface.
2018	number of line is system overflow	The number of lines has exceeded the maximum allowable in the system. (P) Lines beyond 32,767 cannot be referenced.

Appendix C Command File

C.1 Command File Description

A file consisting of E7000PC emulator commands is called a command file. E7000PC emulator commands can be automatically executed by specifying a previously created command file with the [Input from...] or INPUT command.

Figure C-1 shows an example of a command file. Command files can be created by inputting commands in the same format as input from the command area.

```
MAP 0 1FFFF ;S
LOAD :SAMPLE.ABS
Y
.PC 00010000
.SP 00FFEEFC
```

Figure C-1 Command File Example

C.2 Command File Execution at GUI-2600 Initiation

If a command file name is specified on the command line at GUI-2600 initiation, the command file can be executed automatically after initiation. If the [Propertis...] (modification of registered contents) command is selected from the [File] menu (icon) in the Program Manager window, the Program Item Propertis dialog box shown in figure C-2 is displayed. In this dialog box, a file name to be executed by the GUI-2600 (gui26.exe), a space, and a command file name (sample.cmf in figure C-2) are entered. After entering the above items, double-clicking the GUI-2600 icon automatically initiates the GUI-2600 and executes the command file. Automatic command file execution starts after the E7000PC emulator system file has been loaded and a prompt is displayed in the command area.

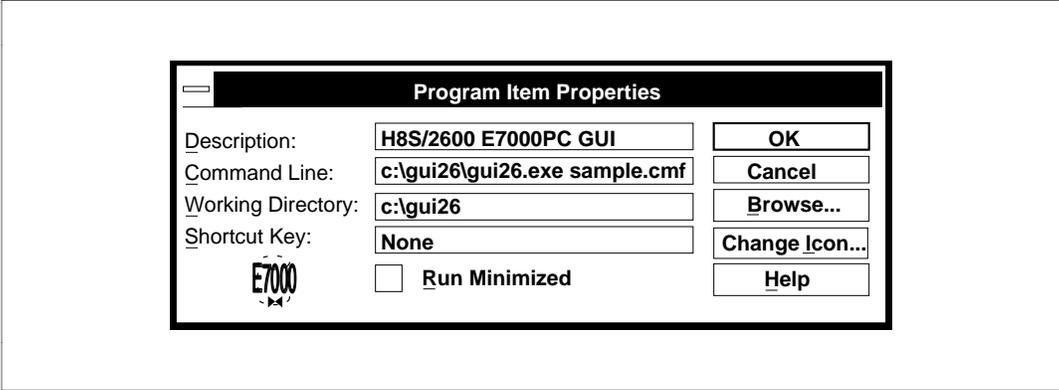


Figure C-2 Command File Execution at GUI-2600 Initiation

Appendix D Initial Setting File (gui26.ini)

The GUI-2600 reads the initial setting file (gui26.ini) at initiation, and operates according to the contents of the file.

By modifying part of the file contents, the GUI-2600 operating environment can be changed.

Figure D-1 shows the contents of the initial setting file (gui26.ini). While the GUI-2600 is operating, part of the contents is automatically modified.

```
[GUI26]
backupfile=
gihelpfile=gui26op.hlp
baseaddr=xxxxxx
SYSCRaddr=xxxxxxxx
```

Figure D-1 Initial Setting File (gui26.ini) Contents

- backupfile=
When a backup file exists, the file name is set by the absolute path name. When the setting information for debugging is stored with the [QUIT...] command, the GUI-2600 automatically sets the file name. The user must not change this setting.
When no backup file exists, no file name is set.
- gihelpfile=gui26op.hlp
The name of the help file (gui26op.hlp), which is displayed with the [GUI operating help] command. The help file must be in the directory where the GUI-2600 is installed. For details of the install directory, refer to section 2.4, Executing Installer Program.

- `baseaddr=xxxxx`

The start address of the memory address range assigned for the IBM PC interface board. If no address is specified, the installer automatically searches for the start address. Therefore, there is usually no need to set the address, but when using two or more IBM PC interface boards, the user must set the address.

Example: When the memory address range for the IBM PC interface board is D000:0000 to D3FF:000F:

```
baseaddr=D0000
```

To directly set the address range, set a memory address range with the `EMMExclude` in the `[386Enh]` section of the `SYSTEM.INI` file. For details, refer to section 2.5, `Modifying System Files`.

- `SYSCRaddr=xxxxxxxx`

The `SYSCR` (system control register) address of the H8S/2655-series MCU is set as eight digits in hexadecimal. After confirming the H8S/2655-series MCU and operating mode to be debugged, set the `SYSCR` address.

Note: If the `SYSCR` address is not set correctly, the `<STEP_UP>` button in the base window and the `ROUTE` window may not operate correctly.

Example: When the `SYSCR` address is 00FFFF39:

```
SYSCRaddr=00FFFF39
```