PARTNER Users Manual
V800 Series
"NB85E-TP Part Edition"

PARTNER version up

* The latest PARTNER-NB85E-TP can be down-loaded from following web page.

http://www.midas.co.jp/products/download/english/program/partner.htm

Notices

Trademarks

Revision History

Date Y / M / D	Rev	Contents
1999.08.06	1.00	Initial Revision
1999.12.08	1.10	32M-256M specifications are added to Emulation ROM dialog, ROM command size. The unit value and default value are changed in debug information area.
2000.01.13	1.11	Collection of errata (P42)
2000.03.06	1.20	Add"7 Flash Memory support" Support RTE-1000-TP changes in History window, History menu, Trace Address dialog, Set CPU environ dialog, TRG command, ENV command, TD command Add Time Tag Separate Mask data changes in Access Breakpoint dialog, Event point dialog, Trace Address dialog, BH command, EP command and TRG command Correction of errata delete Trace mode address mask
2000.05.10	2.00	The 2nd Revision English version remake
2001.05.20	2.01	Modified download site
2002.01.23	2.10	Support RTE-2000-TP changes Emulation ROM dialog, ROM command add ROM1,ROM2,ROM3,ROM4 command
2002.05.08	2.11	Add Bus Trace The related command of bus trace was added to the initialization command. The local menu in a history window is changed. Bus trace setting dialog is added."4.3 Bus Trace" is added. BTRC command addition
2002.07.25	2.12	Intel type is added to the correspondence device of a Flash Memory.
2003.12.10	2.13	The following items are added "-RTE option" of "1.1 PARTNER-NB85E-TP Configuration" "6.1.3 Making by MULTI2000" "Format 2 of address region specification (FLASH_MEM2)" of "7.2 Configuration"

This page is intentionally left blank.

Table of Contents

This page is intentionally left blank.

Getting Started

This document explains the operation that is dependent on PARTNER-NB85E-TP and has not been explained in "PARTNER Users ManualV800 Series Common Part". Regarding the common operations, refer to the explanations in "PARTNER Users ManualV800 Series Common Part".

The operations of RTE-100-TP, RTE-1000-TP and RTE-2000-TP are common, unless otherwise specified.

Necessary Hardware

* Interface Kit (PCIF, ATIF, 98IF, PCIIF, LAN-BOX)

The interface kit to connect personal computer with RTE-100-TP,RTE-1000-TP or RTE-2000-TP, is required. There are interface kits such as, "PC Card Interface Kit", "PC98 Desk Top PC Interface Kit", "DOS/V Desk Top PC Interface Kit", "10Base-T LAN-BOX" and "PCI Bus Interface Kit".

Necessary Setup

* RTE for Win32

Refer to the install manual, RTE for Win32 to install DLL, set the type of ICE to use and connection port, and confirm the connection.

This page is intentionally left blank.

1 Startup

To startup PARTNER-NB85E-TP, the configuration setup is required. In this chapter, how to set options regarding the startup of PARTNER-NB85E-TP, is explained.

1.1 PARTNER-NB85E-TP Configuration

Set Starting Option

[Starting option] dialog box specifies the PARTNER-NB85E-TP startup options, and changes operation mode and memory usage state. (Fig 1)

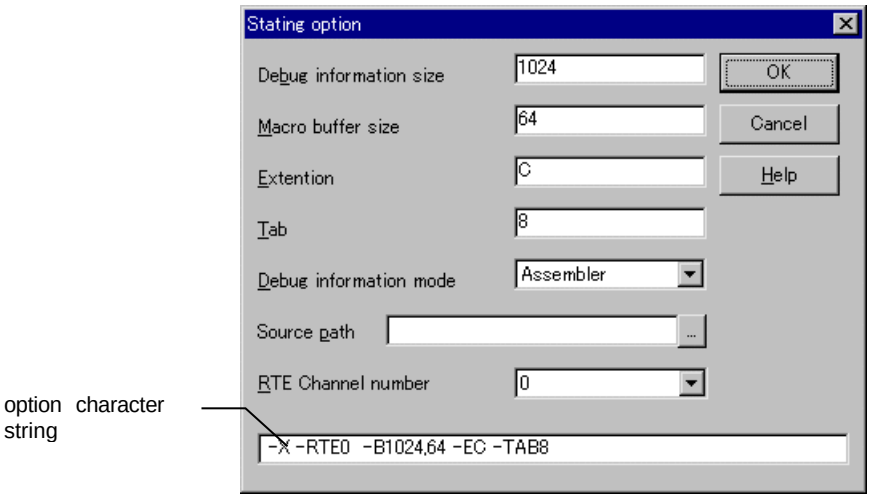


Fig 1 [Starting option] dialog box

In the place of option character string, the contents entered by dialog box are displayed. The startup option is usually specified by [Starting option] dialog box of RPTSETUP. The specified startup option will be written into a project file (RPTNB85ET.KPJ). PARTNER-NB85E-TP startup options are as follows.

-B option

This option specifies the size of debug information and macro region. For debug information region, there are a lot of information, such as, global and local symbol, line number information and attributes of functions and variables.

-B[sizeD][,sizeM]

sizeD	The size of area for debug information registration. If omitted, 1024K byte of area is prepared.
sizeM	The size of area for macro definition information registration. If omitted, 32K byte of area is prepared.

-D option

This option specifies the current directory of PARTNER-NB85E-TP. In the current directory, configuration file (RPTNB85ET.CFG) and project file (RPTNB85ET.KPJ) need to be located.

-D directory

The directory specifies the current directory at PARTNER-NB85E-TP startup. If omitted, it is specified in the order, the directory that specified in [Work folder] of [Property] dialog box, and then the directory where the executable program located that is specified in [command line].

-E option

This option specifies the default file name extension of the source file that is displayed and referred in the Code window.

-E extension

The extension is used as a default extension of source file. As the extension, the part of file name extension eliminated a period, is specified. If this option is omitted, file.C or file.ASM is assumed. Plural extensions can be specified by separated with semicolon (;).

-SD option

This option specifies the directory that is referred in the Code window. Plural directories can be specified by separated with semicolon (;).

-SD directory

The directory is used as a directory for source file. If omitted, the directory specified by -D option is used.

-TAB option

This option specifies the tab size to display source code in Code widow. This capability is convenient to display the file of which tab setting has changed by editor.

-TAB tab_size

The tab is set to the number specified by tab_size. If omitted, the tab size set to 8.

-X option

This option specifies the debug mode for assembler or various kind of C compiler. It is necessary to clarify the programming language system in use, as PARTNER-NB85E-TP supports a lot of the program language. If this option is not specified properly, the debugging capability might not work correctly or some functions might not be available for use. For how to use each programming language system, refer to "6 Debugging with High Level Programming Language" of this document and the documentation of each programming language system.

-X

This option specifies that assembler debug mode (default). If PARTNER-NB85E-TP starts up in this mode, the command related to C language (such as, stack back trace or local variable handling) cannot be used.

-XC

This option specifies Green Hills debug mode. The debug information will be loaded from .OUT file.

-XG

This option specifies exeGCC debug mode. The debug information will be loaded from .OUT file.

-XN

This option specifies NEC C debug mode. The debug information will be loaded from .OUT file.

-XCW

This option specifies Code Warrior C debug mode. The debug information will be loaded from .OUT file.

-RTE option

This option sprcifies the channel number of RTE to use.

-RTE[0|1|2|3]

Please choose 0, when you do not use the channel function of RTE.

@ option

All options other than **-D** option, can be loaded from the file specified with @.

[Example]

Command line input:

C:¥RPTNB85ET¥BIN¥RPTVNB85ET -DC:¥SAMPLE @TEST.INF

Contents of the file (TEST.INF):

-B1024 -SDC:¥SAMPLE¥SOURCE –XC

1.2 Initialize Command

If the configuration of PARTNER-NB85E-TP, other than the contents of configuration file (RPTNB85ET.CFG) needs to be set, following commands are available to initialize them.

For description of each command, refer to "Command Reference" (page 45) or "Hardware User's Manual".

Command to use for initializing

INIT	initialize PARTNER-NB85E-TP and ICE
ENV	set CPU environment, such as input port mask
ROM	set type of emulation ROM and address range (for RTE-100/1000-TP)
ROM1…ROM4	set type of emulation ROM and address range (for RTE-2000-TP)
NC	set none memory cache region
NCD	reset none memory cache region
NROM	set exclusion region for emulation memory assignment
NROMD	reset exclusion region for emulation memory assignment

Though these commands are entered from Command window, it is more convenient to describe them in INIT.MCR file that will be executed automatically at PARTNER-NB85E-TP startup time.

In RTE-2000-TP, there is the following hardware setting commands for Bus Trace. These are through commands (& command). Please refer to "the users manual of an external Bus Trace unit" for details.

BTMULTI, BTCLK, BTCLKQ, BTDLY, BTADDR, BTDATA, BTBE, BTCS, BTSTS, BTETC

Example of initializing by INIT.MCR file

By describing the Initialize commands such as following example, the initialization at each startup time can be done automatically.

init

initialize PARTNER-NB85E-TP and ICE

env !nmi0 !nmi1

set NMI0 and NMI1 port to be masked

rom 100000 20000 1m rom16 bus16

set emulation ROM to 16 bit ROM, 16 bit bus and 1M bit size, and emulation address range to 100000H to 11FFFFH.

* This command is the example of RTE-100-TP/RTE-1000-TP.

nc 20000 1000

specify the area 1000 byte from address 20000 as none memory cache area

L sample.out

load sample.out program to be debug

When using bus trace by RTE-2000-TP, a through command can describe a related command as follows. Please refer to "the users manual of an external Bus Trace unit" for details. These should set up the value recommended by "the users manual of an external Bus Trace unit".

&btmulti !enable	Multiplex is set up
&btclk1 etc 00xxxxxx nor	A bus clock is set up
&btclkq none	Bus clock qualification is set up
&btdly etc 11000000	Delay is set up
&btaddr 32	An address bus is set up
&btdata bus32	A data bus is set up
&btbe bus_etc8	Byte enable is set up

1.3 Error Messages at Startup

The following messages will be displayed when PARTNER-NB85E-TP detects the error at startup time. Restart the system after fixing the cause of errors.

RTE power supply might be off, or some hardware trouble might happen. Check the power supply and cable connection, and execute ChkRTE program.

Other application is using RTE. Or RTE control program has not terminated normally for some trouble. If that is the case, restart Windows.

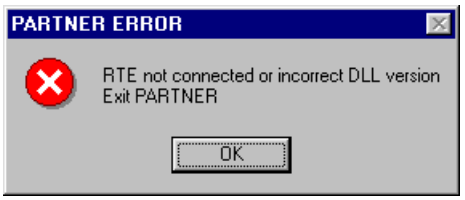
There is no response from RTE during initialization. CPU might not be able to operate, if the target system is connected. Restart the system after checking them and fixing the trouble.

Host interface card or PC card might not be installed. Check that the interface is installed properly and execute ChkRTE program.

The RTE might not be connected, or the power supply of RTE might not be on. Check the power supply and cable connection, then execute ChkRTE program.

The system cannot be initialized. The power supply of RTE might not be on. Check the power supply and connection, and execute ChkRTE program.

For some abnormal termination, RTE control program has not terminated properly. Restart Windows, or forcibly terminate RTE control program.



For some trouble of RTE installation, the necessary DLL cannot be found. Reinstall the system with RTE SetUp CD-ROM, then execute ChkRTE program without fail.

This page is intentionally left blank.

2 Child Window

This chapter explains how to display and operate PARTNER-NB85E-TP child window.

2.1 Register window

Register window displays program register (Fig. 2) and system register (Fig. 3).

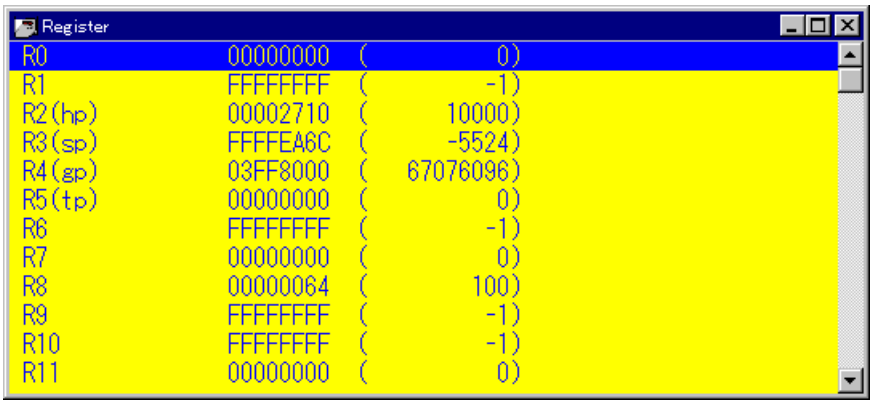


Fig. 2 program register display example

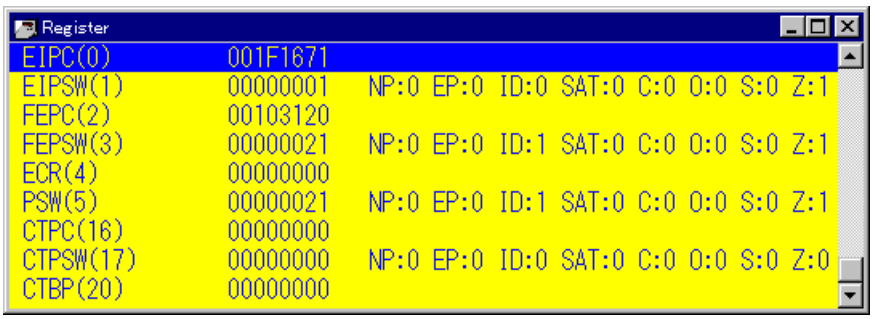


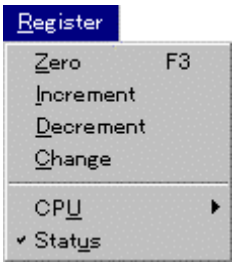
Fig. 3 system register display example

2.1.1 Register window Short-cut Key

In Register window, following short-cut keys are available to execute various kind of commands. These short-cut keys are available if the focus is on the Register window.

Key	Function
F3	initialize the register or flag at cursor position
F6	display the symbol of register value
F7	display the memory contents that is pointed by register value
F9	display the register value in decimal
Enter	pop-up the dialog box to change the register or flag value at cursor position

2.1.2 Local Menu in Register window



[Register] menu has the commands related to controlling the Register window.

Menu Item	Function
Zero	set 0 to the register at cursor position
Increment	increment the register at cursor position
Decrement	decrement the register at cursor position
Change	pop-up the dialog box to change the value of register at cursor position
CPU	display/hide Program register value
Status	display/hide System register value

Program register

Menu Item	Function
Decimal	change the display of register value in decimal
Symbol	change the display of register value symbol
Memory	change the display of memory contents pointed by register value

2.1.3 Mouse Operation in Register window

By double clicking the left-button on register or flag part of Register window, a dialog box will pop-up to change the value.

2.2 Break window

Break window displays the breakpoints that are currently set. (Fig. 4)

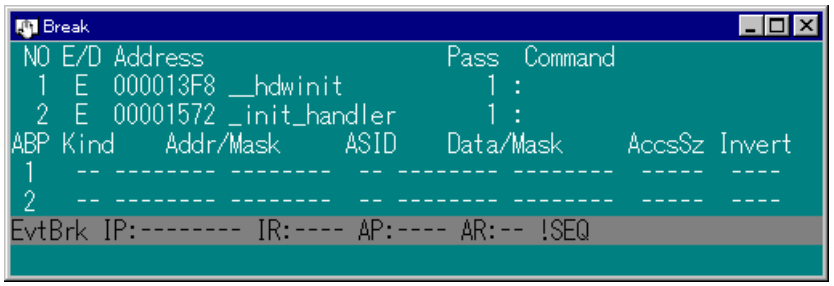


Fig. 4 Break window

2.2.1 Break window Short-cut key

In Break window, following short-cut keys are available to execute various kind of commands. These short-cut keys are available if the focus is on the Register window.

Key	Function
F3	delete all breakpoints
F4	save current breakpoint setting
F6	restore the breakpoint setting saved by F4 key
F9 Enter	enable/disable currently selected breakpoint
INS	display Set Breakpoint dialog box
DEL	delete currently selected breakpoint

2.2.2 Local Menu in Break window

[Break] menu has the commands related to controlling the Register window.

Menu Item	Function
Add	pop-up dialog box to set new breakpoint
Disable/Enable	disable/enable breakpoint at cursor position
Clear	delete the breakpoint at cursor position
Clear All	delete all breakpoints
Load Setting	load breakpoints setting previously saved
Save Setting	save current breakpoints setting
Event break	pop-up dialog box to set access breakpoint (Refer to page 26 "3.1.1 Set Access Break dialog box")
Event	pop-up Set Event dialog box (Refer to page 30"3.1.4 Set Event dialog box")
Event Point	pop-up Set Event Point dialog box (Refer to page 31"3.1.5 Set Event Point dialog box")

2.2.3 Mouse Operation in Break window

By double clicking the left-button on breakpoints in Break window, the breakpoint setting will be toggled between enable and disable.

2.3 History window

The History window displays the contents of real-time trace memory in two types of display mode. They are disassembled listing that analyze trace memory contents (Fig. 5), and hexadecimal dump listing that does not analyze the contents (Fig. 6).

In RTE-100-TP, time tag value is not displayed in disassembled listing.

In RTE-2000-TP, when the bus trace is valid, the contents of the bus trace are also displayed.

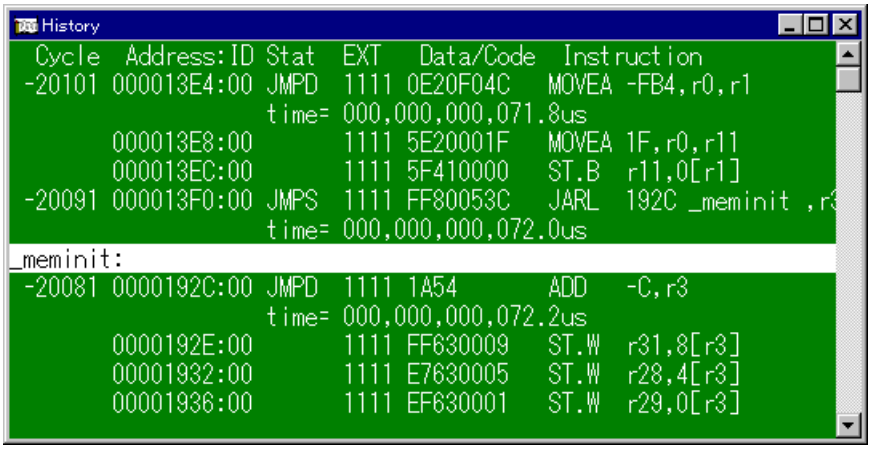


Fig. 5 History window (disassembled listing)

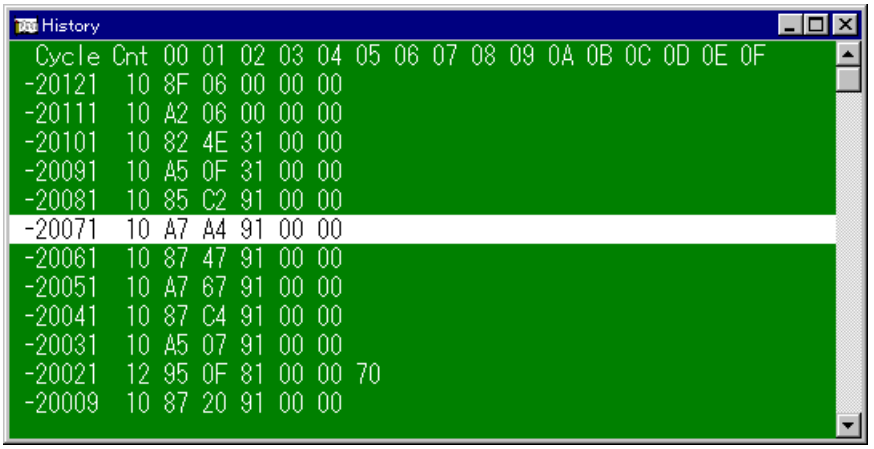


Fig. 6 History window (dump listing)

2.3.1 History window Short-cut key

In History window, following short-cut keys are available to execute various kind of commands. These short-cut keys are available if the focus is on the History window.

Key	Function
F3	change display mode hexadecimal dump/disassembled listing
F6	pop-up the Display Frame Number dialog box
F7	start/stop real-time trace during user program execution

2.3.2 Local Menu in History window

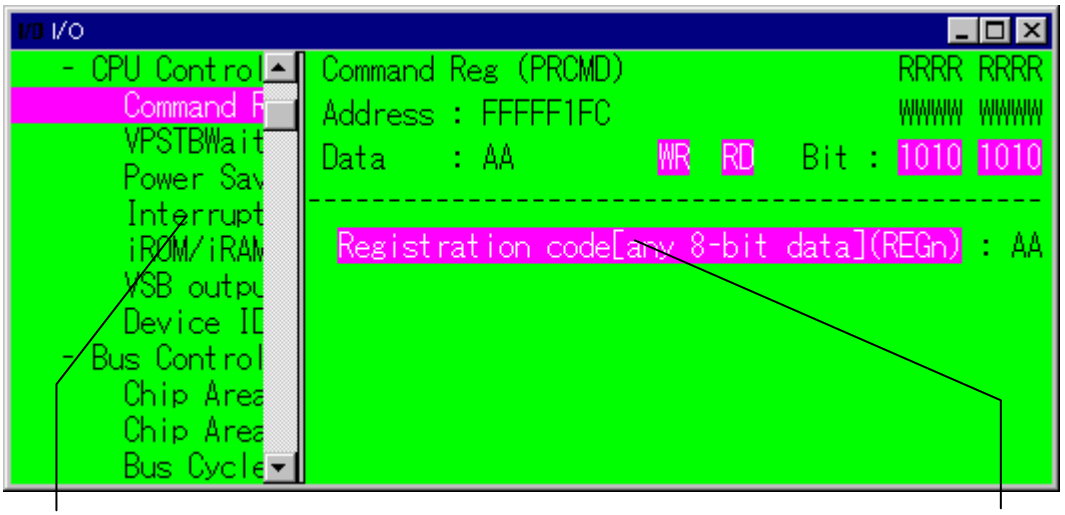
[History] menu has the commands related to controlling the History window.

Menu Item	Function
Dump/Assembler	change History display mode
Timetag:Relative/Timetag:Absolute	change Time Tag display format between relative and absolute time. Not available for RTE-100-TP.
Jump	pop-up Set Display Cycle dialog box
RealTime Trace Go / RealTime Trace Stop	start/stop RealTime trace
Redraw	A display is re-drawn when Bus Trace is effective Not available for RTE-100/1000-TP
Event	pop-up Set Event dialog box (Refer to page 30 "3.1.4 Set Event dialog box")
Event Point	pop-up Set Event Point dialog box (Refer to page 31"3.1.5 Set Event Point dialog box")

Trace Address	pop-up a dialog box to set trigger address and data trace address (Refer to page 29"3.1.3 Set Trace Address dialog box")
Trace Mode	pop-up a dialog box to set trace control condition (Refer to page 27 "3.1.2 Set Trace Mode dialog box")
Bustrace Mode	pop-up a dialog box to set Bus Trace mode (Refer to page 32 "3.1.6 Set Bus Trace Mode dialog box")

2.4 I/O window

The I/O window displays and set the specified internal peripheral I/O register. (Fig. 7)



Item window

Fig. 7 I/O window

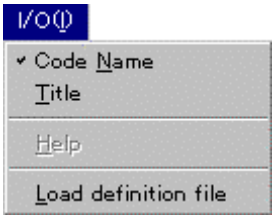
Data window

2.4.1 I/O window Short-cut Key

In I/O window, following short-cut keys are available to execute various kind of commands. These short-cut keys are available if the focus is on the I/O window.

Key	Function
Tab	move cursor between Item and Data window
Enter	in Item window, display/hide hierarchical display in Data window, move to one lower level in the hierarchy
ESC/BS	in Data window, move to one upper level in the hierarchy
[Right Arrow]	in Data window, change the value

2.4.2 Local Menu in I/O window



[I/O] menu has the commands related to controlling the I/O window.

Menu Item	Function
Code Name	display register name
Title	display functional name of register
Help	display the explanation of register
Load definition file	load register information from definition file (RPTxxxxx.IO)

2.4.3 Mouse Operation in I/O window

Change value setting:

By clicking on the data contents to change, pop-up menu will be displayed. Selecting the menu item to change and refer the value.

Display local menu:

Click the right button on the Item window in I/O window.

Copy to clip board:

Click the left mouse button on arbitrary position of the Item window in I/O window, and drag to select the region.

Releasing the button, the selected region will be copied to clipboard.

This page is intentionally left blank.

3 Dialog command

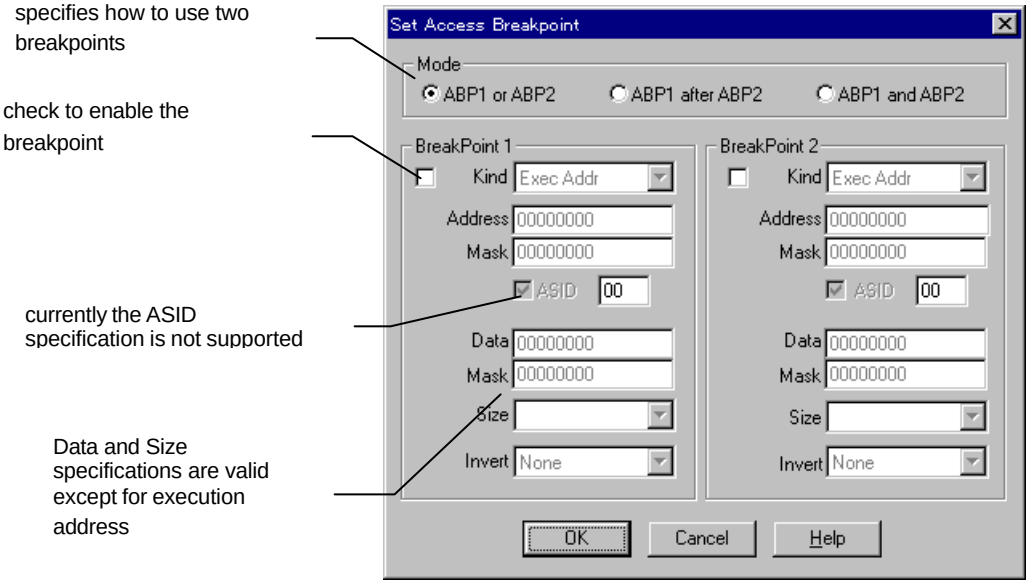
To debug with PARTNER-NB85E-TP, two types of operations, by the dialog box and tool bar and by entering a command from Command window are available. This chapter explains the dialog box that is depend on PARTNER-NB85E-TP.

3.1 Dialog Box

3.1.1 Set Access Break dialog box

[Break]-[Event break]

[Set Access Break] dialog box sets event breakpoints.



There are three operation modes. In AFTER and AND mode, both of the breakpoints needs to be valid.

OR : each breakpoint is used separately

AFTER : break occur, if ABP1 occurs after ABP2

AND : break occur, if the events occur at the same time.

In AND mode, ABP1 and ABP2 should be the event of different type.

Address specifies execution/access address value. Mask sets address mask specification.

Size specifies condition of data access size.

Invert is used if the identical condition of address value and data value s inverted.

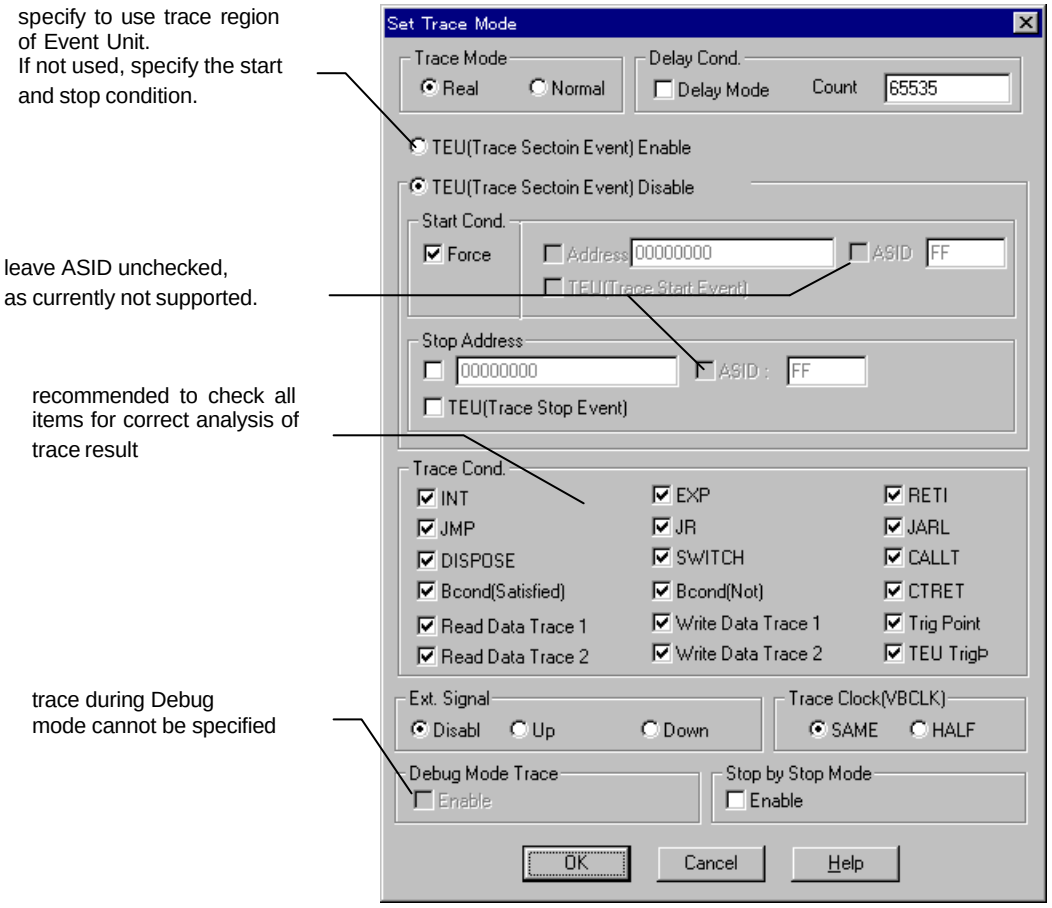
Notes:

Leave the ASID unchecked, for it is reserved for future expansion. If it is not specified, the ASID value will not be compared.

3.1.2 Set Trace Mode dialog box

[History]-[Trace Mode]

[Set Trace Mode] dialog box specifies the Real-time Trace condition.



Trace Mode specifies the mode to acquisition trace data.

Though Normal mode traces execution result in real-time, some data acquisition might fail if trace buffer in CPU becomes full.

All mode will stop the CPU execution temporally to prevent data acquisition failure, if trace buffer in CPU becomes full. Then the CPU execution will be resumed automatically.

The utilization of the trace buffer in these two modes can be altered by Set CPU Environ dialog box. (Refer to page 33, "3.1.7 Set CPU Environ dialog box".)

If Trace Region of Event Unit is used, the data acquisition is done by cause of Trace Region of Event Unit. The trace data is acquainted during the cause of Trace Region of Event Unit is true. In this mode, start and stop condition cannot be specified.

If Region Trace of Event Unit is not used, the data acquisition can be controlled by start and stop condition.
Start condition specifies the condition to start trace data acquisition.
Select the program execution address and trace start condition for the Event Unit.
If the condition is not specified, the data acquisition starts immediately after the CPU execution have started.
To use the start condition of Event Unit, specify the condition by Set Event dialog box and Set Event Point dialog box. (Refer to page 30, "3.1.4 Set Event dialog box" and page 31, "3.1.5 Set Event Point dialog box".)

Stop Cond specifies the condition to stop trace data acquisition.
Select the program execution address and trace stop condition for the Event Unit.
If the Start Cond is specified, trace will start on that condition after trace stop.

Delay Cond specifies the number of cycles (delay count) that trace acquisition the data after the Trigger condition goes true, between the ranges of 0 to 131071 (0x1FFFF).
The Delay mode forcibly stops the trace after acquisition the data of delay count number.
During the Delay mode, Trigger address (Trigger points) will be ignored.
The Trigger is consist of OR condition of Trigger point (Trigger address), external input port and Event Unit.

Trace Cond specifies the type of information to be acquainted as trace data (such as status condition of the program execution address, data trace condition or trigger condition).
If the specified data is not sufficient to analyze the History display data, Trace result might not be displayed properly. Thus, try to specify all information available to avoid this.
The Trigger point and data trace condition is set by Set Trace Address dialog box. (Refer to page 29, "3.1.3 Set Trace Address dialog box".)

The external signal input specifies the condition of external input port that is used as trace trigger.
Trace Clock specifies the divisor rate (divisor of VBCLK) of trace clock (TRCCLK).

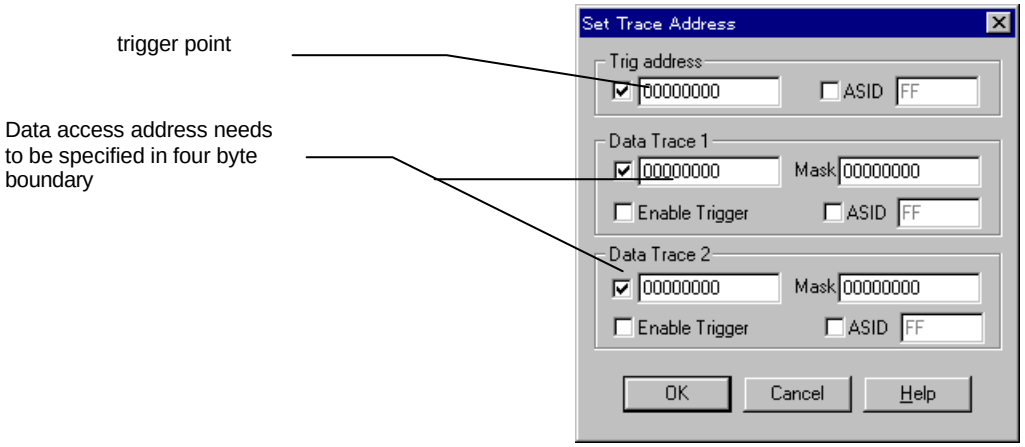
Notes:

Leave the ASID unchecked, for it is reserved for future expansion. If it is not specified, the ASID value will not be compared.

3.1.3 Set Trace Address dialog box

[History]-[Trace Address]

[Set Trace Address] dialog box specifies the address condition that is used for the Real-time Trace.



Data Trace Condition specifies the type of access by the acquisition information in Set Trace dialog box. (Refer to page 27, "3.1.2 Set Trace Mode dialog box".)
The access address mask can be specified only for the bit from 2 to 9.

For model RTE-1000-TP and RTE-2000-TP, the condition can be specified to Trigger point by checking Trigger Enable.
If a trigger matches the both of Data Trace Cond 1 and 2, surely to specify the trigger with Data Cond 1. Unless trigger might not work correctly.

Notes:

Leave the ASID unchecked, for it is reserved for future expansion. If it is not specified, the ASID value will not be compared.

3.1.4 Set Event dialog box

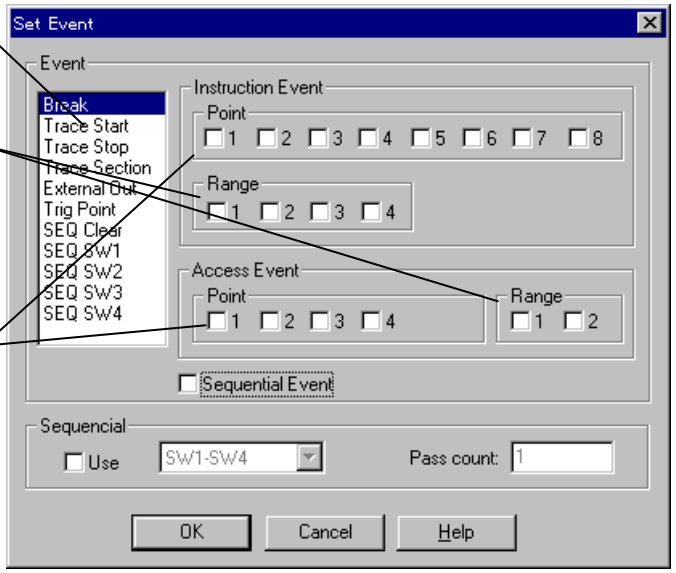
[Break]-[Add] ,[History]-[Event]

[Set Event] dialog box specifies the condition of various types of events.

Select event to be specified.

For Range specification, two event points (2n-1) and 2n are used with AND condition.

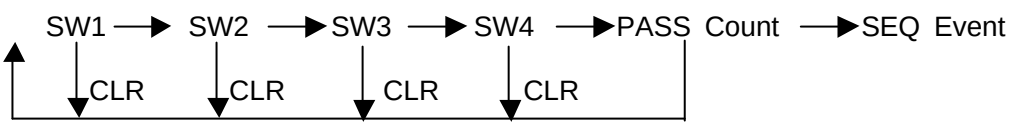
Select points to be detected from instruction and access type events. These points are used as OR condition.



As the range event consists of the AND condition of two event points, thew two point conditions need to be specified.For access type events, the data specification is available only for the condition of point 2n-1.

Event point condition is set by Set Event Point dialog box. (Refer to page 31, "3.1.5 Set Event Point dialog box".)Use SEQ event in Event type specifies the condition of sequential controller.The sequential controller detects that four events occurs in specified sequence.If the ON condition of switch SW1 to SW4 occurs in sequence (SW1->SW2->SW3->SW4), sequential controller events will occur. The pass counter counts up, if the events occurs in the sequence. And if specified number of passes reaches, it will be identical to sequential controller event. By the occurrence of clear factor (SEQ controller CLR) all switches will be cleared.

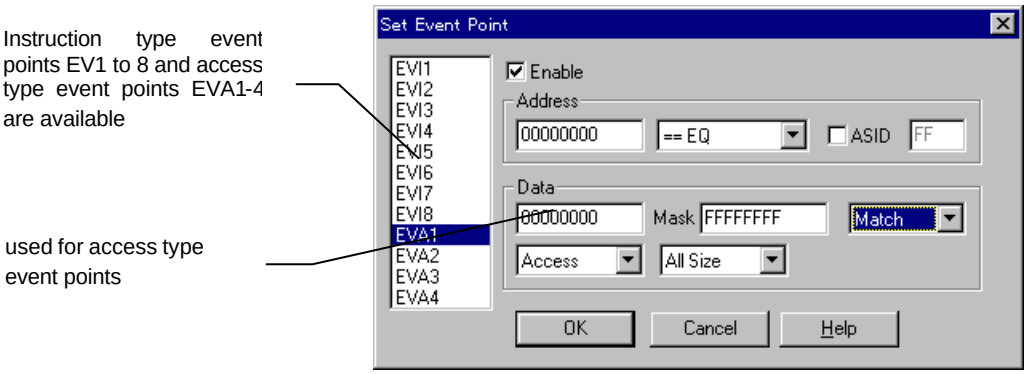
As pass count, the value from 1 to 0xFFF (4095) can be specified. If specified number of switches turn ON in the sequence, it will be counted up. And event occur when it reaches to specified number.



3.1.5 Set Event Point dialog box

[Break]-[Event Point] , [History]-[Event Point]

[Set Event Point] dialog box specifies the Event point condition that is used for event control.



For address condition, address value and comparison condition are specified. For instruction execution type event point, execution address is specified. And for access type event, the address of data access is specified. The comparison conditions are as follows.

specified address == execution/data address : EQ

specified address > execution/data address : LT

specified address < execution/data address : GT

specified address != execution/data address : NE

specified address <= execution/data address : GE

specified address >= execution/data address : LE

If specified as ON, the address will not be compared. (The address condition is always true.)

The detection of these events occurs after the execution for instruction execution type event, and after the access for access type event.

Notes:

Leave the ASID unchecked, for it is reserved for future expansion. If it is not specified, the ASID value will not be compared.

Data Cond specifies data value and next data condition for access type event.

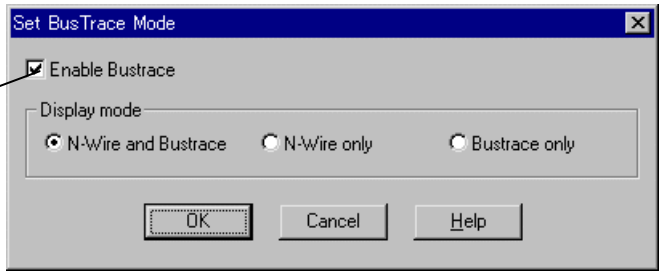
The data value can be specified with mask pattern.The equal or not-equal condition, type of data access and size of data access are specified as data condition.To set the event control that uses event point, refer to "3.1.4 Set Event dialog box", page 30.

3.1.6 Set Bus Trace Mode dialog box

[History]-[Bustrace Mode]

[Set BusTrace Mode] dialog box specifies use of bus trace, and the display conditions of a trace result.

specify to use Bus trace



Display Mode chooses the display form of Bus trace with a history window or TD command.
In "N-Wire & Bustrace", both N-Wire trace (instruction execution trace) and Bus trace result are displayed. Bus trace result is displayed between each cycle of N-Wire trace. It is a series at the time of time tag value.
In "N-Wire only" , Bus trace result is not displayed.
In "Bustrace only", only Bus trace result is displayed.

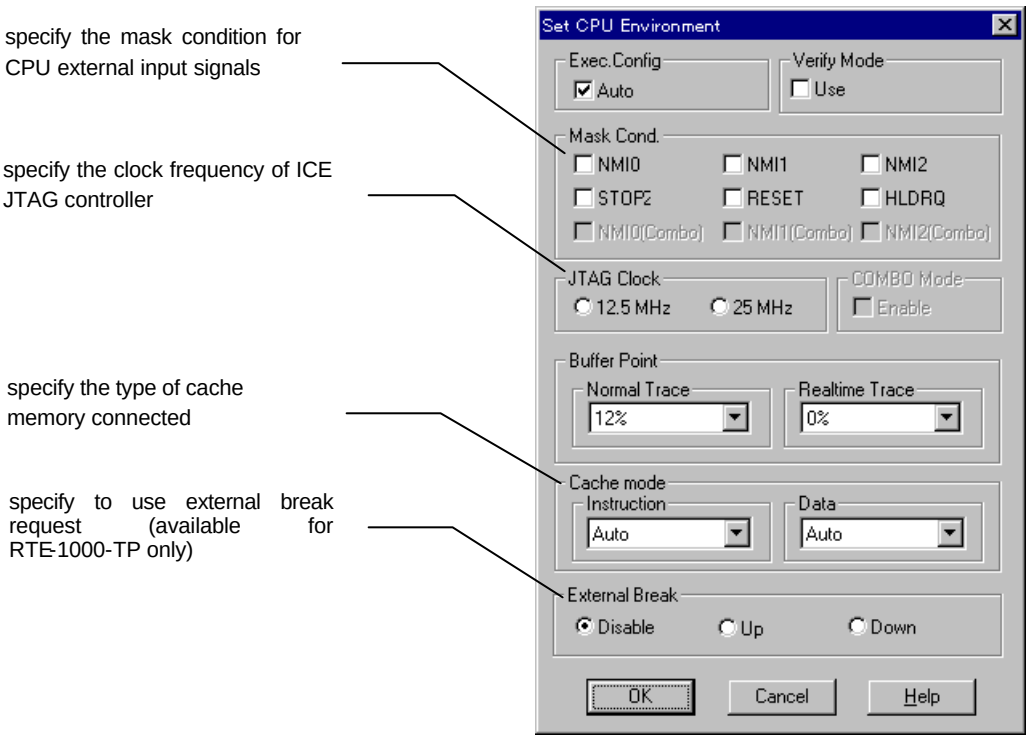
The details of Bus trace are set up by the through command. Please refer to "the Users manual of an external bus trace unit" about those commands.

This dialog is not in RTE-100-TP and RTE-1000-TP.

3.1.7 Set CPU Environ dialog box

[Run]-[Config CPU Environ]

[Set CPU Environ] dialog box specifies the operation condition of CPU.



Execution Cond specifies whether restart automatically or not, when CPU execution break occurs for internal data processing of PARTNER and ICE.

Verify mode specifies whether verify memory contents after memory write for the command or operation associated with memory write. Memory verify will terminate at the address where the first error occurs.

Trace Buffer Utilization specifies the trace buffer utilization ratio for Real-time Trace mode. The meaning of ratio might vary depend on Trace mode.

Normal(real-time) mode:

The utilization ratio at that the trace resumes after trace buffer overflow occurs. The Trace resumes when the utilization ration become smaller than specified percentage. To set to 0 % is recommended.

All mode:

The utilization ratio at that the CPU execution is hold to avoid trace data acquisition failure. The CPU execution stops, when the utilization ration become greater than or equal to specified percentage. The execution resumes, when trace buffer becomes empty. To set to 12 % is recommended.

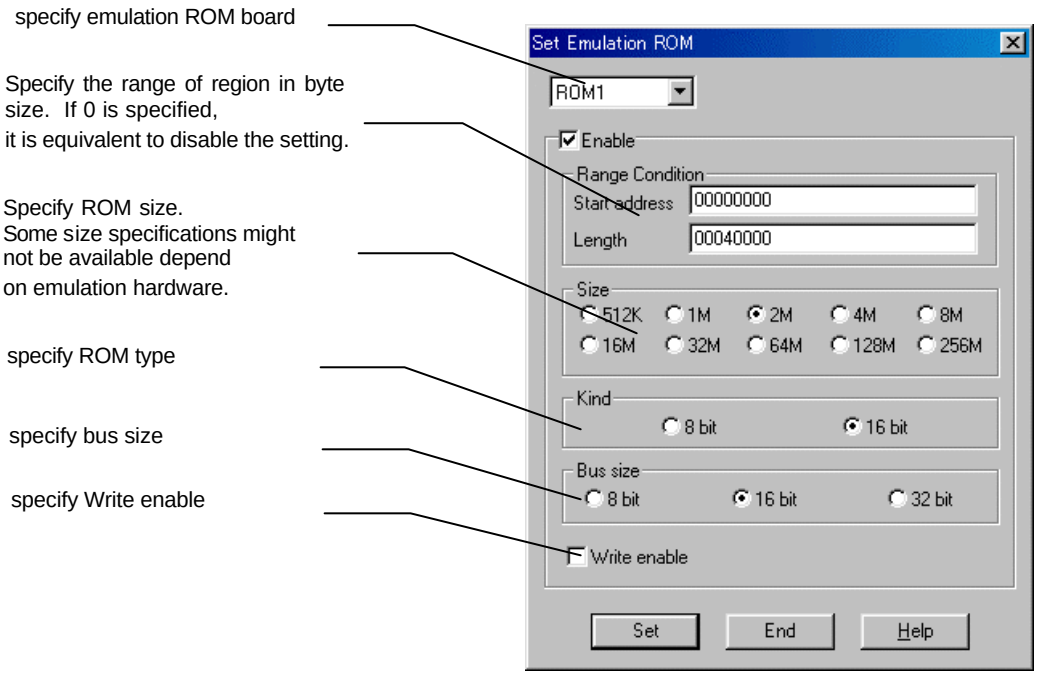
Cache mode:

Specifies the type of cache to be used for each of instruction cache and data cache. Auto can be specified only for an evaluation board that uses NB85E-TEG. In this case, the configuration is automatically identified by checking the contents of CSZ register.

3.1.8 Set Emulation ROM dialog box

[Run]-[Emulation ROM Set]

[Set Emulation ROM] dialog box specifies the area of emulation ROM.



Start Address specifies the lowest address of the ROM to be emulated.

Range Length specifies the size of ROM to be emulated, by the number of byte (with four byte boundary). If 16 bit ROM is used, 8 bit type or 512K size cannot be specified.

Only when a "Set" button is pushed, the contents of specification are set up. A dialog is ended with an "End" button.

In RTE-100-TP and RTE-1000-TP, specification of a ROM board and WriteEnable cannot set up.

In RTE-2000-TP, two or more ROM boards are used by the bus size. Therefore, there is the following relation to a ROM board and a bus size. The bus size which is not in a table cannot be set up.

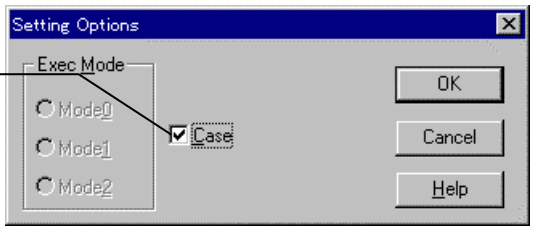
ROM board	Bus size	Slot position	ROM board	Bus size	Slot position
ROM1	8 bit	#3	ROM3	8 bit	#5
	16 bit	#3		16 bit	#5
	32 bit	#3+#4		32 bit	#5+#6
ROM2	8 bit	#4	ROM4	8 bit	#6
	16 bit	#4		16 bit	#6

3.1.9 Config Environ dialog box

[Setup]-[Option Set]

[Config Environ] dialog box specifies various types of configuration for PARTNER-NB85E-TP.

specify the upper/lowercase ignorance for symbols



Notes:

Execution mode is reserved for future expansion, and currently not supported.

If it is not specified, the ASID value will not be compared.

The Execution mode controls user program execution status.

Set to default (Mode 0) as PARTNER-NB85E-TP does not have any special control.

Upper/Lower case specifies case sensitivity for registered symbols such as function names or variable names.

If set a mark (check), upper/lower case will be identified. The case will be ignored, if the mark is not set.

If the mark is set, regarding the operation that handle symbols, a symbol should be specified exactly with case sensitive to the registered one. Unless the symbol cannot be referred.

If the mark is not set, the symbol that matches specified character string ignoring the upper or lower case, will be referred.

4 Real-time Trace

PARTNER-NB85E-TP traces the contents of CPU execution in real-time.

This chapter explains the real-time trace capability.

4.1 Trace Overview

The Real-time Trace writes the contents execution (trace data) from CPU into the trace buffer implemented in ICE in every execution cycle.

The contents of trace buffer can be viewed with history window.

The data acquisition to the trace buffer can be controlled by Trace mode, Trace start condition, trigger condition and stop condition.

The flow of trace data acquisition is shown in Fig. 8 and Fig. 9.

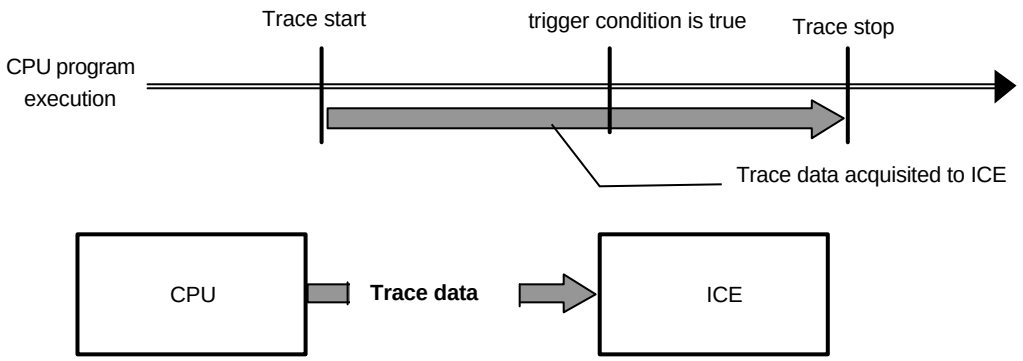


Fig. 8 Flow of trace data acquisition

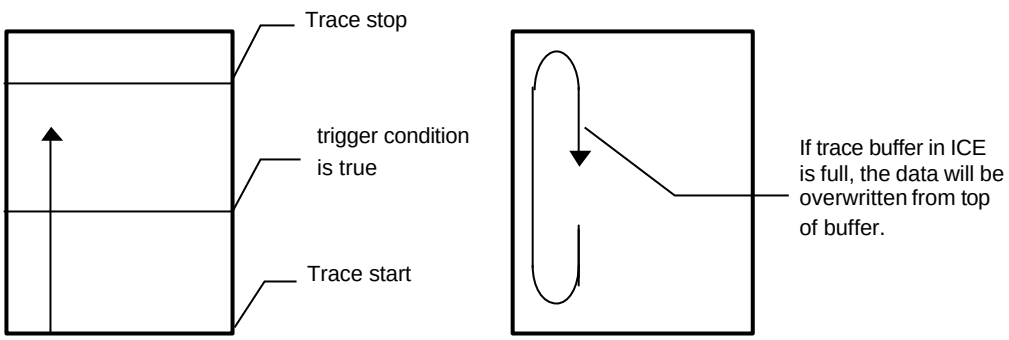


Fig. 9 Trace data inside of ICE

4.1.1 Delay Count

The Delay Count is the number of cycles during that data will be acquisited after trigger condition comes true. (Fig. 10)

The number of cycles might vary depend on the contents of CPU execution, and one cycle is identical with one execution unit.

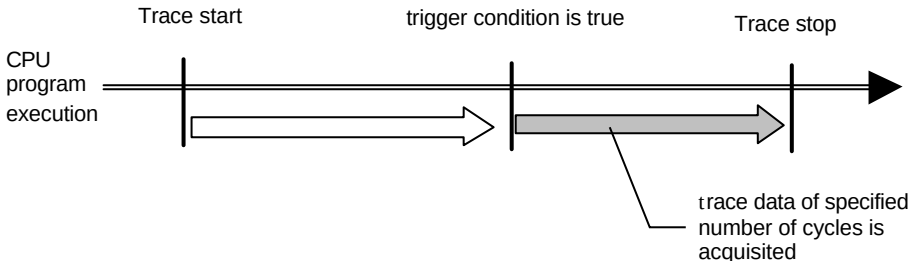


Fig. 10Flow of Delay Count

4.1.2 Normal/All mode

The Normal mode traces the execution contents in real-time.

Thus some data acquisition might fail if trace buffer in CPU becomes full. (Fig. 11)

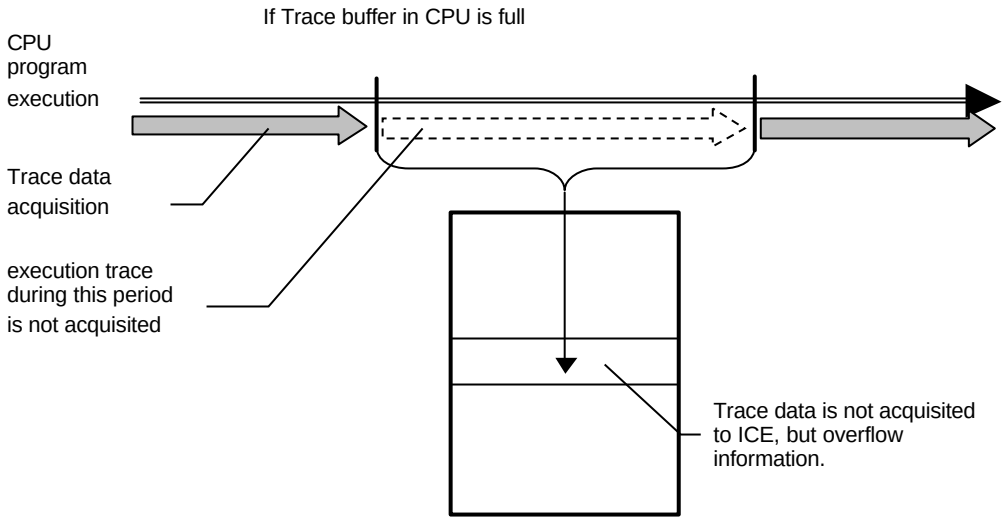


Fig. 11 Normal mode

The all mode is the one considered preventing the data acquisition failure. Thus the CPU execution will stop temporally, if trace buffer in CPU becomes full. Then the CPU execution will be resumed automatically.(Fig. 12)

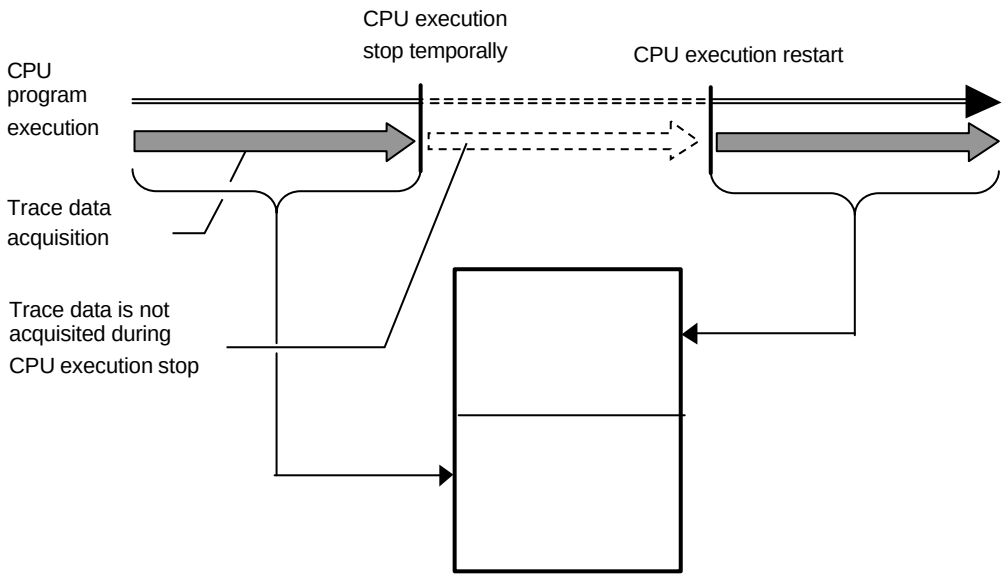


Fig. 12 All mode

4.1.3 Trace Start condition

The Trace Start condition specifies the condition to start trace data acquisition. (Fig. 13)

If the trace region event in Event Unit is used, the Trace start condition cannot be used.

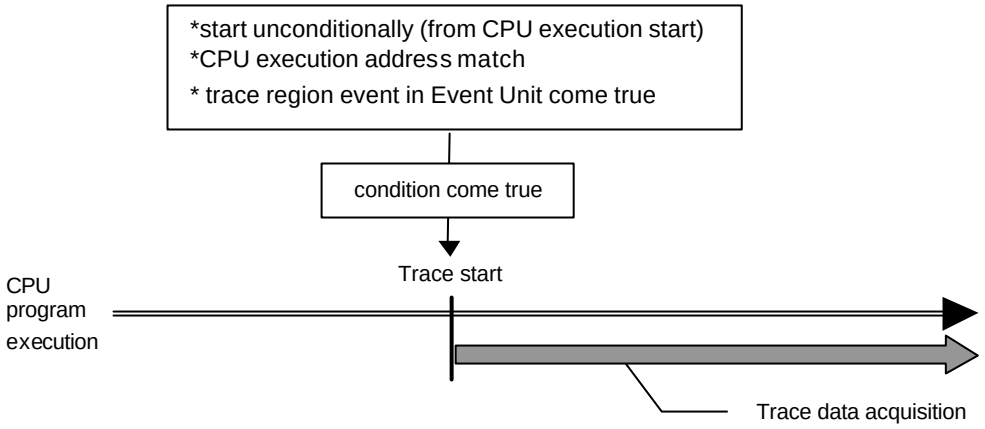


Fig. 13 Trace start

4.1.4 Trigger condition

The Trigger condition is the starting point of the Delay Count.(Fig. 14) With Trigger condition, the execution history before and after the condition comes true, can be viewed.

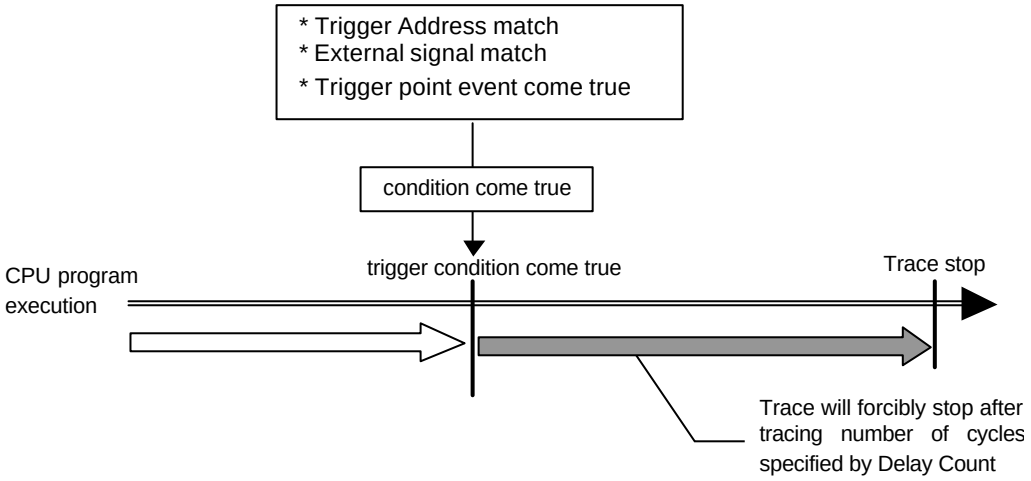


Fig. 14 Trigger condition

4.1.5 Trace Stop condition

The Trace Stop condition is used to temporally stop trace data acquisition.

After stopping, if start condition is specified, the trac e data acquisition will restart by that condition.(Fig. 15) If trace region of Event Unit is used, the start condition cannot be used.

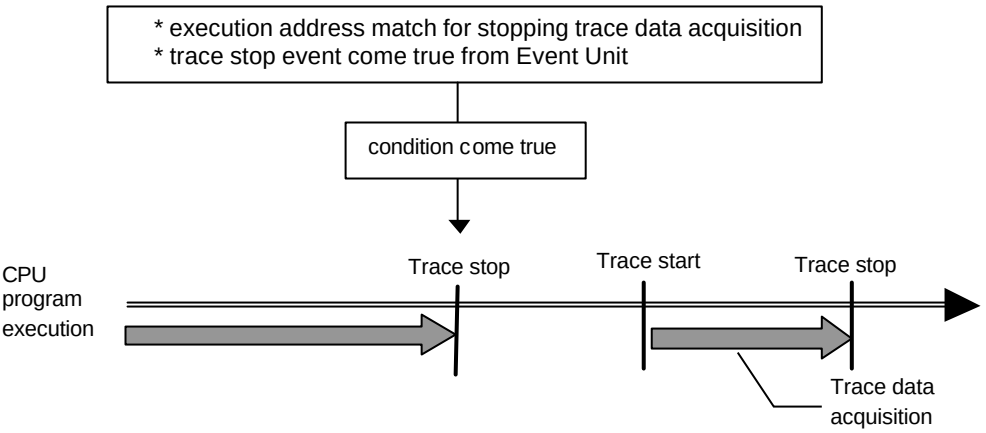


Fig. 15 Trace Stop condition

4.1.6 Trace Quit

If Trace is quitted, trace data will not be acquisited after that. Different from trace stop condition, trace will not start. (Fig. 16)

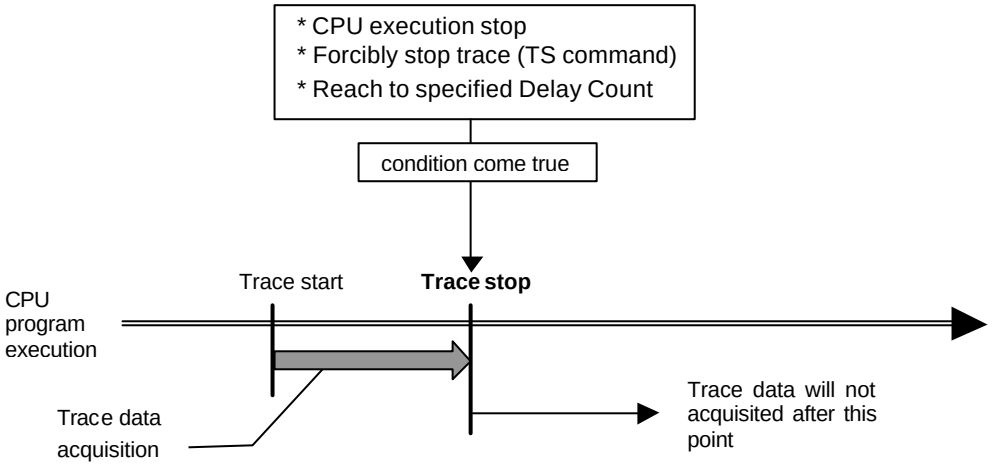


Fig. 16 Trace Quit

4.1.7 Forced Delay mode

The Forced Delay mode stop trace forcibly, if trace data has bee acquisited by specified number of Delay Count (cycles) after trace start. During this mode, the trigger condition will be ignored. (Fig. 17) In this case, trace will start at the start of CPU execution.

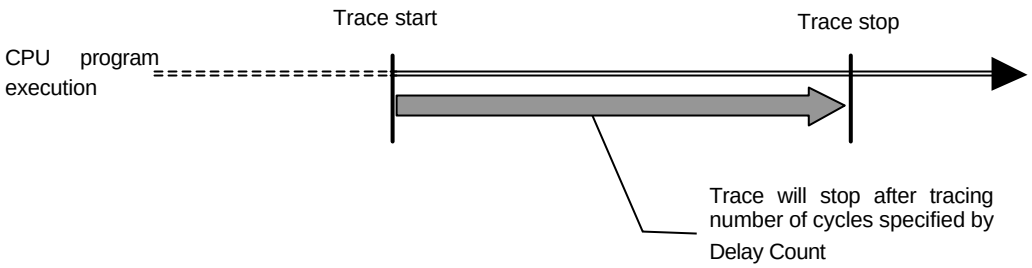


Fig. 17 Forced Delay mode

4.1.8 Trace Region Event

If Set Event (page 30, "3.1.4 Set Event dialog box") is used, the trace data during the region that the event condition is true, will be acquisited. If the condition is not true, the data will not be acquisited. (Fig. 18)

While Trace Region Event is used, the start condition and the stop condition cannot be used.

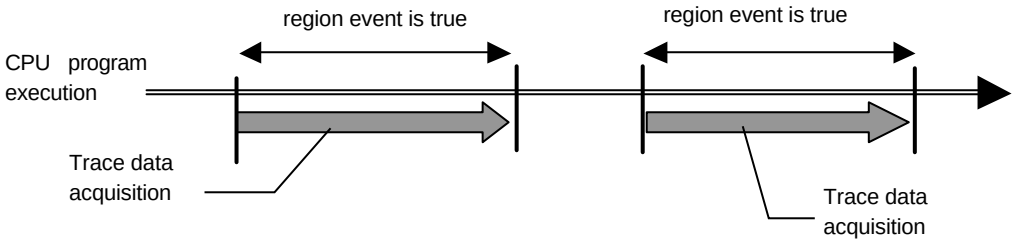


Fig. 18 Using Trace Region Event

4.2 Time Tag

RTE-1000-TP and RTE-2000-TP registers also the data related to time to trace buffer, which shows approximate program execution time. The contents of Time Tag can be viewed by History window.

4.2.1 Time Tag display types

There are two types in Time Tag display, absolute time and relative time display.

Absolute Time Display:

The oldest cycle in trace buffer is marked as time 0, and time will be displayed as the elapsed time from there.

As the internal processing of ICE is not displayed, if the oldest cycle is that one, the cycle of time 0 will not displayed.

Relative Time Display:

The time difference from the last cycle is displayed.

If it is the case that instructions between the cycles is complemented by software, or the Time Tag data is not registered correctly, the Time Tag data will not be displayed.

4.2.2 Notes on Time Tag

PARTNER-NB85E-TP and ICE might implicitly break user program execution and restart it for internal data processing. Thus, if it is the case, these data processing times also will be included to Time Tag elapsed time.

And including above case, if the data before and after break and restart execution is acquisited to trace buffer, the trace data corresponding to internal data processing, will not be displayed. Thus at the trace data around break or execution restart, time display in absolute and relative time format might not be identical.

4.3 Bus Trace

In RTE-2000-TP, bus data is traceable. This is written in the bus trace buffer in ICE with execution trace. These contents can be referred to in a history window. The writing to a bus trace buffer is controllable in the bus trace mode. About the flow of bus trace data writing, it has become like Fig. 19.

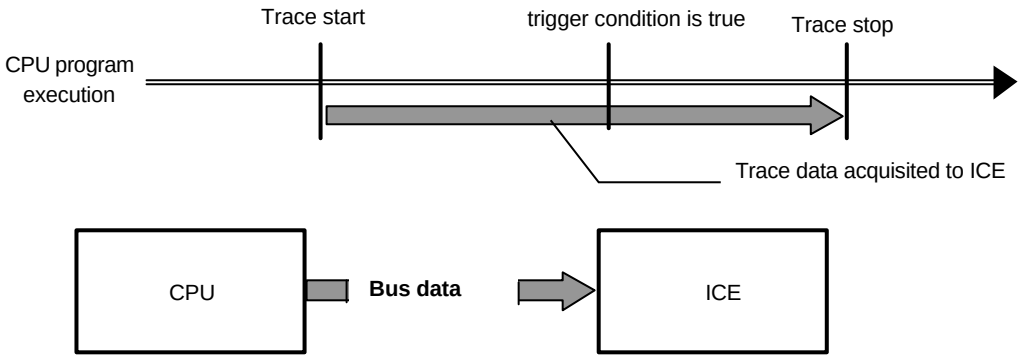


Fig. 19 Flow of Bus trace data acquisition

Refer to "the user manual of an external bus trace unit" for the details of a bus trace setup.

Notes: When using Bustrace, an external bus trace board and the attachment of exclusive use are required.

4.3.1 Notes of a Bus Trace result display

Bus trace result is displayed on the basis of the value of a Time Tag. As follows, Bus trace result is searched and displayed from the Time Tag value which each cycle of execution trace has.

Time Tag value of n cycle <= Bus trace < Time Tag value of n+1 cycle

Bus trace result is not displayed on all cycles.

As for the case of the mixture display with execution trace, each oldest trace data of

execution trace and bus trace is looked for. The cycle after the time tag value of the newer one of either is displayed. Therefore, there is what is not displayed by the execution trace or bus trace of time tag value before it. However, all are displayed when there is few trace data written in ICE (when neither of the trace buffers becomes full).

The result of Bus trace is displayed as the contents set up by the pattern setting command except for address value and data value. When the pattern is not set up, it is displayed in the form of '---'. Please refer to "the user manual of an external bus trace unit" about a pattern setting command.

5 Command Reference

This chapter explains the commands that can be entered into PARTNER-NB85E-TP Command window. For detail contents (such as input examples), please refer to On-line Help.

5.1 Conventions on Command Reference

PARTNER-NB85E-TP commands consist of command name and parameter list. In some cases, parameters can be omitted. These parameters are indicated with braces ([...]). If there are more than two elements to be selected, these elements are described with curly braces ({...}) and (|).

If the parameters are omitted, the initial value of PARTNER-NB85E-TP or the successive parameters used by the command executed last time, will be applied.

5.2 Commands Categorized by their Functions

5.2.1 Hardware Initialization

INIT initialize PARTNER-NB85E-TP and ICE

5.2.2 CPU Reset

RESET reset CPU

5.2.3 Environment command

ENV [[!]AUTO] [[!]VERIFY] [JTAG{12|25|*5M*|*2M*|*1M*|*500K*|*250K*|*100K*}] [R{0|1|2|3}] [N{0|1|2|3}] [[!]NMI{0|1|2}] [*NOEXT*|*NEGA*|*POSI*] [[!]RESET] [[!]HLDRQ] [[!]STOPZ] [INONE|IWAY1|IWAY2|IAUTO] [DNONE|DWAY1|DWAY2|DAUTO]
Set various configuration of CPU, such as input signal mask.
Parameters indicated with italic fonts are available only for RTE-1000-TP and RTE-2000-TP.

ROM [address[range]] [ROM8|ROM16][BUS8|BUS16|BUS32] [512K|1M|2M|4M|8M|16M|32M|64M|128M|256M]
Set type of Emulation ROM and address range.
For RTE-100-TP and RTE-1000-TP.

ROM1…4 [address[range]] [ROM8|ROM16][BUS8|BUS16|BUS32] [512K|1M|2M|4M|8M|16M|32M|64M|128M|256M] [[!]WREN]
Set type of Emulation ROM and address range.
For RTE-2000-TP.

NC [address range] set Uncached Memory Area

NCD list delete Uncached Memory Area

NROM[address range]
set and display emulation memory assignment exclusion area

NROMD list delete emulation memory assignment exclusion area

5.2.4 Program Load

L [file_name] load program to be debugged and debug information

5.2.5 File Read/Write

RD file_name,address
write specified file to specified address

WR file_name,range
write specified memory contents with range to specified file

5.2.6 Program Execution

T [iterations] execute the program with trace (F8)

P [iterations] execute program with step (F10)

G [=execute_address][,break_address][,/W]

G@ [,/W] execute program (F5,F7)
/W: prohibit command during execution

ESC forcibly break the program execution (ESC)

5.2.7 Breakpoint

BP [address[,iterations[,command]]]
set breakpoint at specified address (F9)

BC list clear the break points specified with list

BD list list disable the break points specified with list

BE list list enable the break points specified with list

5.2.8 Access Breakpoint

BH [mode][ABP1 address address_mask [data_mask] [type] [access_size] [invert_condition]],[ABP2 address address_mask [data_mask] [type] [access_size] [invert_condition]]
set access breakpoint condition

BH [!ABP1][!ABP2] delete access breakpoints

BH display access breakpoints currently set

5.2.9 Display Code

V[.][file_name:][line_number]
display specified line of specified file in Code window

V function_name — display source file of specified function in Code window

U [address] — display disassembled listing from specified address in Code window

UPUSH [address] — PUSH current displayed address into address stack (internal stack of eight levels), then display disassembled listing from specified address

UPOP — display disassembled listing from the address UPUSH'ed last time, then POP address stack

UEND — display disassembled listing from the address UPUSH'ed last time

5.2.10 Display/Change Register

R — display register value

_register=expression
change specified register with the value of expression

R register — change register value

5.2.11 Display/Change System Register

STAT — display system register

STAT system_register_name=data
change specified register value with data
EIPC,EIPSW,FEPC,FEPSW,ECR,PSW,.....>

5.2.12 Display/Change Memory

D[type]range[,iterations][,radix_base]
display the memory contents of specified range in specified type format and radix base

E[type] address — change memory contents from specified address with the format specified with type

F[type] range,list — fill memory range in specified format type with the value of list

S[type] range,list — search for the memory patterns specified with list in specified range and format type

C range,address — compare memory contents of specified range with the one of specified address

M range,address — move memory block specified with range to specified address

5.2.13 Input/Output from/to I/O Port

PI[type] address[,/C]
display the memory contents of specified address with specified type format

PO[type] address,data[,/C]
output data to specified address in specified type format

5.2.14 Display/Set Symbols

X[symbol_name] — display symbol_name (all symbols, if not specified)

[.]name=address — register(change) symbol specified name with specified address

5.2.15 Line Assemble

A address — start line assemble from specified address and write directly to memory

5.2.16 Backtrace

K — display backtrace of C function

5.2.17 Real-time Trace

TRC display current Real-time Trace mode

TRC [delay_value][[!]REAL][FORCE|PCADDR address|TEUSEC], [!STOP|STOP address],[[!]DELAY][!TEU][TEUST][TEUSP] [SAME|HALF][[!]NSS] [[!]ALL][[!]EXP][[!]INT][[!]JMP][[!]JR][[!]JARL][[!]RETI][[!]IFD] [[!]IFE][[!]CALLT][[!]SWITCH][[!]DISPOSE][[!]CTRET][[!]RD1][[!]RD2] [[!]WR1][[!]WR2][[!]TRG][[!]TEUTRG][NOEXT|NEGA|POSI]
set Real-time Trace mode

TS stop Real-time Trace

TG resume Real-time Trace

TRG display current address condition of Real-time Trace

TRG [[!]TRG [trigger_address]],[[!]DTADR1 [data_address[address_mask]]],[[!]DTADR2 [data_address[address_mask]]]
set address condition of Real-time Trace

TRG [[!]DTTRG1] [[!]DTTRG2]
set data trace as trigger (available for RTE-1000-TP and RTE-2000-TP only)

TD[U] enter into Real-time Trace display mode. 'U' specifies disassembled listing display mode

Internal command in Real-time Trace display mode

B move display start frame to the top of trace memory

D [frame_address_S][,frame_address_E]
change display mode to dump listing mode, then display

U [frame_address_S][,frame_address_E]
change display mode to disassembled listing mode, then display

T change Time Tag display format between absolute and relative time

(available for RTE-1000-TP and RTE-2000-TP only)

5.2.18 Bus Trace

BTRC display current mode of Bus Trace

BTRC [[!]ENABLE] [BOTH|NWIRE|BUSTRACE]
set Bus Trace mode and Bus Trace display mode.

5.2.19 Event

EV
display current status of each event and sequential controller

EV {BRK|TST1|TST2|TRNG|TOUT|TRGP|SSW1|SSW2|SSW3|SSW4|SCLR}
[[!]IP1][[!]IP2][[!]IP3][[!]IP4][[!]IP5][[!]IP6][[!]IP7][[!]IP8]
[[!]AP1][[!]AP2][[!]AP3][[!]AP4][[!]IR1][[!]IR2][[!]IR3][[!]IR4]
[[!]AR1][[!]AR2][[!]SEQ]
set each event condition

EV [!]FSEQ [{S1|S2|S3|S4}][count]
set condition of sequential controller

EP
display current condition of each event point

EP [!]IP{1|2|3|4|5|6|7|8} [address][{EQ|LT|GT|ON|NE|GE|LE}]
set condition of execution type event point

EP [!]AP{1|2|3|4} address [{EQ|LT|GT|ON|NE|GE|LE}] [data [data_mask]]
[{RD|WR|ACC}] [{/B|/W|/D|/A}] [{MT|UN}]
set condition of access type event point

5.2.20 System Call

SYSC address
set System Call facility ON setting specified address as system call entry point

SYSC OFF
set System Call facility OFF

SYSC
display status of System Call facility

5.2.21 Display Expressions

H expression
display the value of expression in octal, decimal, hexadecimal, ASCII and real number format

H expression1,expression2
display addition and subtraction between expression1 and expression2.

PRINTF format[,parameter]

PF format[,parameter]
formatted display similar to printf() of C function

5.2.22 Display/Change C data type

INS C_expression[,function]
evaluate C expression and display in Inspect window (F6,Ctrl+I)

W? C_expression — register C expression in Watch window (Shift+F7,Ctrl+W)

W[type] address[,range][,radix_base]
register the memory contents specified by address and range in Watch window

Y list — delete watch point specified by list

VAL C_expression[,function]

? C_expression[,function]
evaluate and display C expression

5.2.23 Define character string(evaluate C expression)

DEF string1 string2

DEFINE string1 string2
define character string for mini-preprocessor

DEF — display the contents of character strings definition currently registered

DEF * — disable all character strings definition currently registered

5.2.24 System Control

EXIT — terminate the PARTNER

Q — terminate the PARTNER

HELP — display help

VER — display the version of PARTNER

MAP — display the memory map currently configured

!! — display command line history

! string — search string in command line history

& — move to/terminate through command

& through_command
execute through command

5.2.25 Change Radix Base

N radix_base	set radix base of input to decimal or hexadecimal number

5.2.26 Logging/Batch

> file_name	redirect Command window display and input to file
>> file_name	append logging to specified file
>	stop logging (close log file)
< file_name	redirect Command window input to file. The batch execution can be stopped by ESC key

5.2.27 Set Options

OPTION {ON|OFF}	identify upper and lower case (SHIFT+F10)
	Display Control and Miscellaneous options
CLS	clear Command window
HOME	move Command window cursor to home position
LOCATE X_coordinate,Y_coordinate	move Command window cursor to specified position
LALL	specify display output from macro command
SALL	specify display output suppression from macro command
LIST	specify display output in Command window
NLIST	specify display output suppression in Command window
BEL	ring a bell
TIME	display current time (hour:minute:second)
WAIT	stop macro execution for a while
PROMPT character	change prompt to specified character
*	indicate comment line

5.2.28 Macro Command

{ macro_name	register macro body with specified macro name
DO{ }WHILE expression	macro command similar to do..while statement in C language
FOR{ }	macro command similar to for statement in C language
WHILE{ expression	macro command similar to while statement in C language
REPEAT{ parameter	repeat macro command
BREAK	exit from macro
KILL macro_name	delete macro definition
MLIST [macro_name]	display macro definition
MLIST >file_name	write all macro definition currently registered to file
< file_name	read macro definition from specified file
IF{ expression	flow control command similar to if,elseif,else statement in C language

5.2.29 Set Flash Memory

ZF Stat[,address]	display the condition of flash memory
ZF Erase, address	erase all area of flash memory including specified address
ZF Flush	write uncompleted write data
ZF Clear	discard uncompleted write data

This page is intentionally left blank.

6 Debugging with High Level Programming Language

PARTNER-NB85E-TP supports C language that is available from Green Hills, NEC(CA8x0), GNU(exeGCC) and Code Warrior.

For detail of ROM-ize with these compiler packages, please refer to the manual for them. This chapter explains how to generate the debug information required for PARTNER.

PARTNER-NB85E-TP reads the debug information at the same time when the program to be debugged is loaded. Thus, it is necessary to generate correct debug information from C compiler or assembler to debug properly.

6.1 GreenHills C

PARTNER-NB85E-TP can read ELF format executable file. The file name extension of executable is ".OUT".

6.1.1 How to compile and link

Compiling modules written in C language, add "-g -XDWARF" option at command line. For the module compiled with this option, source level debugging will be available.

If the executable file name is specified by -o option, set the file name extension to ".out".

>ccv850e -g -XDWARF *source_file_name* -o *executable_file_name*.out

source_file_name *source file name*

executable_file_name *executable file name*

If modules written in C language are compiled by builder, add following build options.

:dwarf=true

:debuglevel=plain

:outputname=*executable_file_name*..out

As a consequence of above, the executable file (.out) will be generated in ELF file format.

6.1.2 Example

1) compile

>ccv850e -g -XDWARF -c @*cmpfiles*

cmpfiles list file of source files to be compiled
(to specify plural source file names)

From above command line, object file which has file name extension ".o", will be generated.

2) link

>lx -map=sfile.map -o sfile.out @*chkpro.lx* @*lnkfiles*

chkpro.lx option (link map) file to be passed to linker program

lnkfiles list file that specifies the object files to be linked
(to specify plural object file names)

From above command line, executable file (.out) will be generated in ELF file format.

3) link map example

The link map specifies where each section for the executable file will be allocated.

```
-m 3 -sec
{
        .text          0x00001000     :
        .rodata                       :
        .syscall                      :
        .secinfo                      :
        .fixaddr                      :
        .fixtype                      :
        .data          0x03e00000     :
        .sdabase align(4)             :
        .sdata                        :
        .rosdata                      :
        .sbss                         :
        .bss                          :
        .heap align(8) pad(0x4000)    :
        .stack align(8) pad(0x2000)   :
}
```

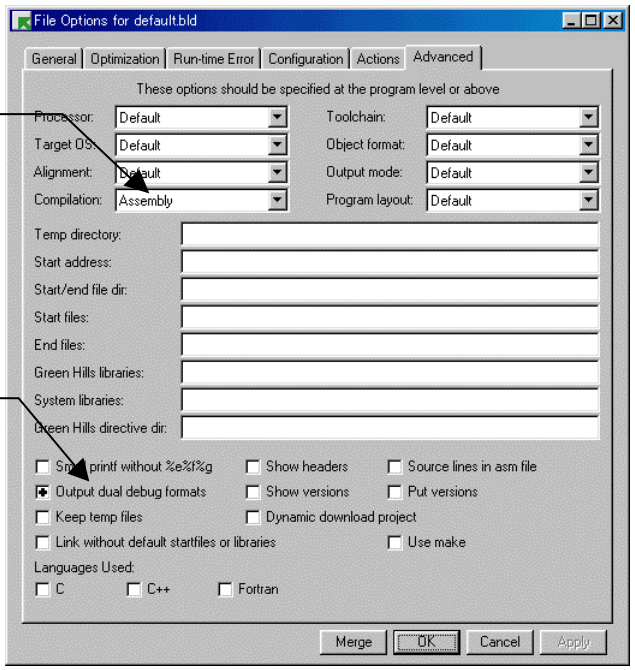
6.1.3 Making by MULTI2000

When you make using MULTI2000, please output debugging information by the next operation.

- Display a "File Option" dialog from the "Project" menu.
- Display an "Advanced" tab.
- Choose the "**Output dual debug formats**".
- Choose the "**Assembly**" of a "Compalation" drops list.

Choose the "Assembly"

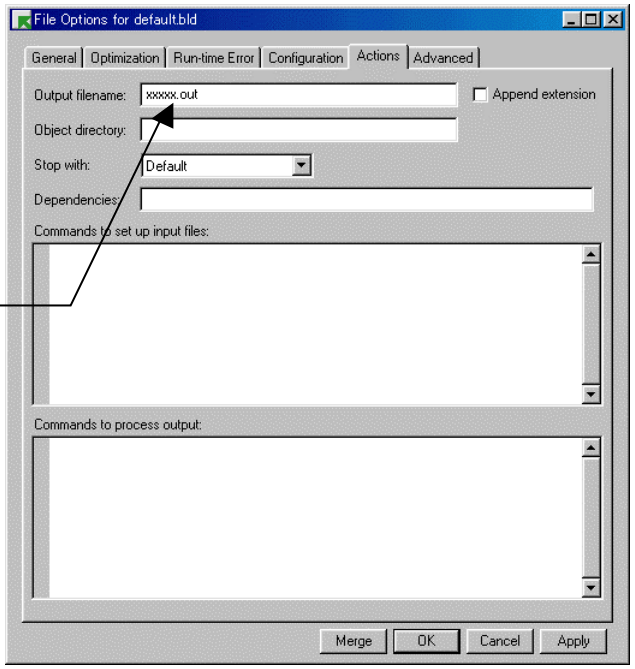
Choose a "Output dual debug formats".



Please set the extension of an output file to "**.out**" by the next operation.

- Display a "Actions" tab.
- The output file name which set the extension to "**.out**" is specified.

The extension of an output file name is set to ".out."



When you use it at a command line, please add "**-g -dual_debug –noobj -o***output file* **.out**" to a parameter.

6.2 GNU C(exeGCC)

PARTNER-NB85E-TP can read GNU C executable file. The file name extension of executable is ".OUT".

6.2.1 How to compile and link

Compiling modules written in C language, add "-g" option at command line. For the module compiled with this option, source level debugging will be available.

If the executable file name is specified by -o option, set the file name extension to ".out".

>gcc -g *source_file_name* -o *executable_file_name*.out -T *command_file_name*

source_file_name	source file name
executable_file_name	executable file name
command_file_name	linker command file

The executable file (.out) will be generated in COFF file format.

As sometimes the options related with optimizing prevent proper source level debugging, compile without these option.

6.2.2 Example

1) compile

>gcc -g –m850e –c *@mpfiles*

cmpfiles	list file of source files to be compiled (to specify plural source file names)

From above command line, object file which has file name extension ".o", will be generated.

2) link

>ld -Map sfile.map –T*chkpro.cmd* -o sfile.out @lnkfiles

chkpro.cmd	option (link map) file to be passed to linker program
lnkfiles	list file that specifies the object files to be linked (to specify plural object file names)

From above command line, executable file (.out) will be generated in COFF file format.

3) link map example

The link map specifies where each section for the executable file will be allocated.

```
SECTIONS
{
     __xfer = 0 ;
     __heep_size = 0x6000 ;
      __stack_init = 0x3e08000 ;
  .text 0x00001000 :
  {
     *(.jmptbl)
     *(.text)
  }
  .call_table ALIGN (4) :
     {
     *(.call_table)
     __ctbp = .;
     *(.call_table_data)
     *(.call_table_text)
    } = 0xff   /* fill gaps with 0xff */
  .rodata ALIGN (4) :
     {
        *(.rodata)
     __erdata  = . ;
     }
  .data ALIGN (4) :
     {
     __data  = . ;
        *(.data)
     __edata  = . ;
     }
  .tdata ALIGN (4) : {
     __ep = .;
     *(.tbyte)
     *(.tcommon_byte)
     *(.tdata)
     *(.tbss)
     *(.tcommon)
     }
  .bss ALIGN (4) :
     {
     __fbss = . ;
        *(.bss)
        *(COMMON)
     __end  = . ;
     __heep = . ;
     }
   .debug                0 : { *(.debug) }
  .debug_srcinfo   0 : { *(.debug_srcinfo) }
  .debug_aranges   0 : { *(.debug_aranges) }
   .debug_pubnames 0 : { *(.debug_pubnames) }
  .debug_sfnames   0 : { *(.debug_sfnames) }
   .line                0 : { *(.line) }
}
```

6.3 NEC C

PARTNER-NB85E-TP can read ELF format executable file. The file name extension of executable is ".OUT".

6.3.1 How to compile and link

Compiling modules written in C language, add "-g" option at command line. For the module compiled with this option, source level debugging will be available.

If the executable file name is specified by -o option, set the file name extension to ".out".

```
>ca850 -c -g source_file_name.c
>ld850 -D link_directive -o executable_file_name.out source_file_name.o -Llib -lc
```

source_file_name	source file name
executable_file_name	executable file name
link_directive	link directive file name

Consequently, an executable file will be generated in ELF format (.OUT).
The executable file (.out) will be generated in COFF file format.

As sometimes the options related with optimizing prevent proper source level debugging, compile without these option.

6.3.2 Example

1) compile and link

For compiling and linking, make tool (VMAKE) in VSH environment is used.

Followings show the example of makefile.

```
NECROOT = C:/nectools

ca850 -g -c sfile.c
ld850 -D dfile -o sfile.out lib¥crtn850.o sfile.o -Llib -lc
```

2) link map example

```
TEXT : !LOAD ?RX V0x00001000 {
 .text                    = $PROGBITS ?AX;
};

DATA : !LOAD ?RW V0x03e00000 {
 .data                    = $PROGBITS ?AW;
 .sdata                   = $PROGBITS ?AWG;
 .sbss                    = $NOBITS ?AWG;
 .bss                     = $NOBITS ?AW;
};
__tp_TEXT @ %TP_SYMBOL;
__gp_DATA @ %GP_SYMBOL &__tp_TEXT;
```

6.4 CodeWarrior C

PARTNER-NB85E-TP can read ELF format executable file. The file name extension of executable is ".OUT".

Notes:

For the debug mode option specified at PARTNER-NB85E-TP start up time, select "-XN" (debug mode for NEC C).

6.4.1 How to compile and link

Compiling modules written in C language will be done by build tool (CodeWarriorIDE).

Creating a project by build tool, specify followings regarding target setup option.

debug format: NEC

optimize level: 0

In addition to them, register several configuration necessary for the project (HeapSize, StackSize, Section Mappings, ...) and necessary modules.

Consequently, an executable file will be generated in ELF format (.OUT).

As sometimes options related with optimizing prevent proper source level debugging, compile files without these options.

6.4.2 Example

1) compile and link

It is done by build tool (CodeWarriorIDE).

2) link map example

The link map specifies where each section for the executable file will be allocated.

```
$segment TEXT 0x00001000
{
        .text
}
$segment DATA 0x03e00000
{
        .data
        .exception
        .exceptlist
        .sdata
        .sbss
        .bss
}
```

7 Flash Memory support

PARTNER-NB85E-TP supports to write flash memory directly without consciousness about it by dialog command (such as E, F, M, L, or RD) or Window command.

ZF command is used to control and display the status of flash memory.

The execution control type dialog command (such as G, T or P) or Window command handle flash memory as ROM.

Thus breakpoints control for flash memory uses hardware break.

7.1 Supported devices

Supported devices for flash memory write need to satisfy following specifications.

1.The product from **AMD** or compatible device and conform to following items.

1) Electrically erasable and programmable device with single power supply.
2) Erasing and writing are controllable by JEDEC standard commands.
3) The sector size is 64 KB except for a boot sector.
4) The structure of bottom boot type sector conforms following specification.
   [16KB x 1 sector] + [8KB x 2 sector] + [32KB x 1 sector] + [64KB x n sector]
5) The structure of top boot type sector conforms following specification.
   [64KB x n sector] + [32KB x 1 sector] + [8KB x 2 sector] + [16KB x 1 sector]
6) The sector protection (block protect) should be applied.
7) To be connected to CPU via 8/16/32 bit data bus.
8) The circuitry should not be particular structure, such as, banked or partially unmapped address space. (Whole address space of the device should be mapped to CPU address space in linear.)

2.The product from **Intel** or compatible device and conform to following items.

1) Electrically erasable and programmable device with single power supply.
2) The sector size is 64/128/256 KB except for a boot sector.
3) The structure of bottom boot type sector conforms following specification.
   [8KB x 8 sector] + [64/128/256KB x n sector]
4) The structure of top boot type sector conforms following specification.
   [64/128/256KB x n sector] + [8KB x 8 sector]
5) The sector protection (block protect) should be applied.
6) To be connected to CPU via 8/16/32 bit data bus.
7) The circuitry should not be particular structure, such as, banked or partially unmapped address space. (Whole address space of the device should be mapped to CPU address space in linear.)

3.The device from which only sector size and sector composition differ in the specification of the above 1 or 2.

For the device that does not conform above, memory write by dialog command or Window command is not supported, but memory read and access from user program is

still possible.

The following table shows the devices that currently satisfy above condition 1 or 2.

[Supported Device]

AMD	Am29F040, Am29F400, Am29F800, Am29LV004, Am29LV400
ST	M29F040, M29F400, M29W008, M29W800
Fujitsu	MBM29F040,MBM29F400,MBM29LV800,MBM29LV008, MBM29LV160
Toshiba	TC58FVT400, TC58FVB400, TC58FVT800, TC58FVT160
INTEL	28F***C3, 28F***w18, 28F***W30, 28F***K**, 28F***J3*
HITACHI	HMB32S4, HMT32S4
MITSUBISHI	M***B33****, M***T33****

7.2 Configuration

To write flash memory by dialog command or Window command, address region for flash memory needs to be specified in configuration file (RPTNB85ET.CFG). There are two kinds (FLASH_MEM, FLASH_MEM2) of forms in specification by sector composition.

Format 1 of address region specification (FLASH_MEM)

This form is the case of the specification of 1 or 2 of "7.1 Supported devices".

Specify the address, type and bus width of the devices that are implemented on target board. A maximum of eight of this format can be specified together with FLASH_MEM2. If plural devices are specified, define them in ascending order.

Format: FLASH_MEM *start_address*, *end_address*, *memory_type*
CPU_bus_width [,*FLASH_bus_width*]

For *start_address* and *end_address*, the range of memory region that the device is mapped to, is specified.

For *memory_type*, one of the followings is specified depending on the device type.

memory type	Device type	Sector Structure
TOP	AMD type	top boot block type
BOTTOM	AMD type	bottom boot block type
NO	AMD type	without boot block
I32W_TOP	INTEL type, with Write Buffer*	top boot block type 32Kword(64KB)/sector
I32W_BOTTOM	INTEL type, with Write Buffer*	bottom boot block type 32Kword(64KB)/sector
I32W_NO	INTEL type, with Write Buffer*	without boot block 32Kword(64KB)/sector
I32_TOP	INTEL type	top boot block type 32Kword(64KB)/sector
I32_BOTTOM	INTEL type	bottom boot block type 32Kword(64KB)/sector
I32_NO	INTEL type	without boot block 32Kword(64KB)/sector
I64_NO	INTEL type	without boot block 64Kword(128KB)/sector
I128_NO	INTEL type	without boot block 128Kword(256KB)/sector

*** Write Buffer** must be controllable by command 41H.

For *CPU_bus_width*, either of 8, 16 or 32 bit is specified as data bus width that connected to CPU. For example, 8bit bus device of two pieces are connected to CPU via 16 bit bus, specify 16.

For *FLASH_bus_width*, either of 8 or 16 bit of bus width is specified as device data bus width. If omitted, the same bus width with CPU is assumed. However, if CPU bus width is 32 bit, 8 or 16 should be specified.

Format 2 of address region specification (FLASH_MEM2)

This form is the case of the specification of 3 of "7.1 Supported devices"

Specify the address, type and bus width of the devices that are implemented on target board. A maximum of eight of this format can be specified together with FLASH_MEM. If plural devices are specified, define them in ascending order.

Format: FLASH_MEM2 *start_address*, *memory_type,sector_info,*
CPU_bus_width [,*FLASH_bus_width*]

For *start_address*, the address of memory region that the device is mapped to, is specified.

For *memory_type*, one of the followings is specified depending on the device type.

memory type	Device type
0	AMD type
1	Intel type
1001	Intel type,with Write buffer*

*** Write Buffer** must be controllable by command 41H.

For *Sector_Info*, the following form describes the sector composition of a device.

size:*count*[,*size*:*count*[,*size*:*count*[,*size*:*count*…..]]]

size:*count* describes the sector specification of a device.

size:*count* is made into one set and it can specify to 128 sets.

size The size of 1 sector is specified(Number of 16s).

count The continuous number of sectors is specified(Number of 10s).

For *CPU_bus_width*, either of 8, 16 or 32 bit is specified as data bus width that connected to CPU. For example, 8bit bus device of two pieces are connected to CPU via 16 bit bus, specify 16.

For *FLASH_bus_width*, either of 8 or 16 bit of bus width is specified as device data bus width. If omitted, the same bus width with CPU is assumed. However, if CPU bus width is 32 bit, 8 or 16 should be specified.

To define this format, there needs to be consistent with MAP format (refer to chapter 3.3 of "V800 series Common Edition"), and needs to satisfy following condition.

1) This format should be described after MAP format.
2) The region specified with this format, should be covered by the region specified with MAP format.
3) The region specified with this format, should be covered by the region specified with single MAP format.

If above conditions are not satisfied, an error message will be displayed at PARTNER-NB85E-TP start up time, and the system will not start up.

Example of configuration file definition

Correct configuration 1-1: specify single block

```
MAP 00000000, FFFFFFFF                # equivalent to default
FLASH_MEM 03000000,037FFFFF, NO, 32, 8
```

Correct configuration 1-2: specify single block

```
MAP 00000000, FFFFFFFF
FLASH_MEM2 00000000,0,2000:8,10000:126,2000:8,16,16
```

Correct configuration 2-1: specify plural blocks

```
MAP 00000000, 003FFFFF
MAP 00400000, 007FFFFF
MAP 00800000, 00FFFFFF
MAP 02000000, 02FFFFFF
MAP 03000000, 037FFFFF
MAP 03C00000, 03FFEFFF
FLASH_MEM 02000000, 027FFFFF, NO, 32, 8
FLASH_MEM 03000000, 037FFFFF, NO, 32, 8
```

Correct configuration 2-2: specify plural blocks

```
MAP 00000000, 003FFFFF
MAP 00400000, 007FFFFF
MAP 00800000, 00FFFFFF
MAP 02000000, 02FFFFFF
MAP 03000000, 037FFFFF
MAP 03C00000, 03FFEFFF
FLASH_MEM2 00000000,0, 10000:128,32,8
FLASH_MEM2 01000000,0, 10000:128,32,8
```

Incorrect configuration 1: out of MAP region

```
MAP 00000000, 033FFFFF
FLASH_MEM 03000000,037FFFFF, NO, 32, 8          <-
```

Incorrect configuration 2: overlap to plural MAP

```
MAP 00000000, 003FFFFF
MAP 00400000, 007FFFFF
MAP 00800000, 00FFFFFF
MAP 02000000, 02FFFFFF
MAP 03000000, 037FFFFF
MAP 03C00000, 03FFEFFF
FLASH_MEM 02000000, 027FFFFF, NO, 32, 8
FLASH_MEM 02800000, 037FFFFF, NO, 32, 8          <-
```

7.3 Memory Write Operation

To write flash memory by in dialog command or Window command, any special operation is not required. The normal operations enable to write without consciousness about flash memory.

However, for the characteristics of flash memory, to change the memory contents, first it needs to be electrically erased by the unit of sector before write operation. The write operation requires several seconds to be completed.

Thus, PARTNER-NB85E-TP has a control buffer inside of debugger to improve a processing speed and eliminate the number of write, and the write operation of flash memory is done by following mechanism.

1) Write request occurs in dialog command or Window command.
2) Read the sector contents that include address of the request into control buffer.
3) Update the contents of control buffer corresponding to changed address.
4) The contents of control buffer will actually be written at following timing.
   - ∗ When user program start to be executed.
   - ∗ At the end of processing accompany with block write.
   - ∗ When the control buffer is explicitly flashed by ZF command.
   - ∗ The other case that write operation is required for internal processing.

When flash operation of control buffer starts, following messages will be displayed in Command window or Status bar.

[Example of messages]

1) Flash ROM [42000000-42003FFF] Sect.Erase. /* erasing a sector */
   Erasing the contents sector address that is displayed.
2) Flash ROM [42000000-42003FFF] Sect.Write. /* writing to a sector */
   Writing the data to displayed sector address.
3) Flash ROM [42000000-420FFFFF] CHIP-Erase.
   /* erasing the whole contents of chip */
   Erasing the whole contents of chip which address is displayed.
4) Flash ROM Device Error! /* fail to read or write */
   Read or write failure occurs. Incorrect configuration file or hardware trouble in circuit or device might be the reason. Write failure also occurs, if sector protection (write protection) is applied.

Technical Supplement on Flash Memory Read/Write Operation

To have better understanding about flash memory read and write operation, this section explains the control buffer prepared in debugger software.

For the control buffer, the same size of memory as flash memory is prepared, and its status is controlled as follows by the unit of sector.

Buffer Status	Meaning
Invalid	whole data in buffer is invalid (buffer is not used): default status
Valid	buffer contains the same data as flash memory
Dirty	buffer data has updated, but not written to flash memory

If read or write operation for flash memory area is requested in dialog command or Window command, the state will transit as follows.

Read request:

If and only if the status of requested sector is Dirty, the data will be read from control buffer. On this request, the state transition does not occur.

Write request:

If the status of requested sector is Invalid, flash memory data will be read to control buffer and the status will transit to Valid.
Then the write data will be written into control buffer, and the status will transit to Dirty.

Flush request:

While the status of the sector is Dirty, the data will be written from buffer into flash memory, then the status will transit to Valid.

Clear request:

The status of requested sector will be unconditionally changed to Invalid. Even if the status of the sector is Dirty, it will not write to flash memory.

Erase request:

The status of requested sector will be unconditionally changed to Invalid. Even if the status of the sector is Dirty, it will not write to flash memory.

7.4 Flash Memory Control command (ZF)

This chapter explains about ZF command, which controls flash memory and displays the status of control buffer.

Status: format ZF Stat[, address]

Displays the status of control buffer that includes specified address. If address is omitted, the status of control buffer for all devices will be displayed.

Erase: format ZF Erase, address

Erases the whole contents of device that includes specified address. The address cannot be omitted.

If Erase is executed, the status of sector in control buffer will be unconditionally changed to Invalid, and the data of Dirty sector will be discarded.

Flush: format ZF Flush

The data of Dirty sector will be written from control buffer into flash memory, and the status of the sector will be changed to Valid.

Clear: format ZF Clear

The status of sector in control buffer will be unconditionally changed to Invalid, and the data of Dirty sector will be discarded.

Appendices

Register Variable

The register variables available in PARTNER-NB85E-TP are as follows.

Register Variable Symbol	register
_r0,,,_r31	R0 register,,,R31register
_pc	PC register
_hp	alias of R2 register
_sp	alias of R3 register
_gp	alias of R4 register
_tp	alias of R5 register
_ep	alias of R30 register
_lp	alias of R31 register

```
>while{ _R0!=_R1        /* compare R0 and R1 register */
? T                     /* execute trace command */
?}                      /* end  of macro definition */
>
```

In above example, T (trace) command will be executed until the value of R0 and R1 got be equal. Even if there is a symbol which name is identical to register's, the register is referred.

Error Messages

Error Messages of PARTNER-NB85E-TP are as follows.

Command error

Specified command cannot be recognized as PARTNER internal command or macro command.

Illegal address

Invalid address is specified in address part of command line or start and end address is oppositely specified. Also occurs when unregistered symbol name is used.

Illegal data

Invalid data or data of out of range is specified in data part of command line.

Illegal syntax

There is an error in command input format, parameter specification or number of parameters.

Verify error occurred

Data cannot be written into memory. This occurs for write operation to the address area where memory is not implemented or ROM area.

Unable to multiply define macro in the macro

Macro definition (registration) needs to be done at PARTNER command line. Macro cannot be defined within a macro command.

Unable to delete macro in the macro

Macro deletion needs to be done at PARTNER command line. Macro cannot be deleted within a macro command.

Conflicted Dialog command

The macro name defined by macro command duplicates PARTNER internal command name. Define it as different macro command name.

Macro buffers full

The macro definition buffer is full or too many macro commands has been defined.

Extend macro buffer size by RPTSETUP program, and then restart PARTNER.

Not completed macro definition {}

The number of curly braces ({}) is not consistent in macro command definition.

Overflow macro nests (max 15)

The maximum macro nesting is 15 levels.

Not Found

Specified number of setting such as breakpoint or watch, cannot be found.

No more break points available (max 15)

More than 15 points of breakpoints have been requested to set. Breakpoints can be set up to 15 points.

Unable to set watch data

More than 16 points of Watch have been requested to set. Watch can be set up to 16 points.

Watch data multiply defined

The content of W command is already registered as Watch.

Illegal file format

The file format to be loaded is invalid. The debug mode specified by RPTSETUP is not match with the file format that is being loaded.

Not found file

The file specified by command line cannot be found.

Unable to open file

Specified file cannot be opened. Confirm that the file really exists. This might occur when too many files have been opened.

Could not create file

This is full or too many files are opened at the same time. Close unnecessary files

Disk full

Specified disk has not enough free area.

Not found debug information

Debug information cannot be found during user program loading by L command.

Illegal debug information

Debug information format is invalid. Confirm the debug mode specified by RPTSETUP and file to be loaded.

Debug information buffers full(Ref. -B option)

The buffer for debug information registration has not enough free area. Expand debug information buffer size by RPTSETUP program.

Unable to set local symbols

Trying to register the global symbol which name is identical with the one of local symbol already registered.

No such variable

C variable specified by ? or VAL command cannot be found.

Illegal expression

C expression specified by ? or VAL command has an error.

Unable to use harmful aftereffect operator

In ? command, operators with side effect (such as =, += or -=) cannot be used. Use VAL command, if these operators are necessary.

Internal Error

This message will be displayed, if trouble occurs in PARTNER internal process. The product is designed so that this error will not usually occur. If this error occurs with enough duplication, contact to sales office.

Unable to use in execution target program

Specified command cannot be used during user program execution. Execute the command after break user program execution.

Cannot execute monitor program

The monitor program to control CPU cannot execute, or terminate properly. Execute Initialize command. If this message is displayed by Initialize command execution, execute Initialize command again after resetting the CPU of target hardware.

Index

D

E

F

G

H

I

L

M

N

P

R

S

T

PARTNER Users Manual
V800 Series
"NB85E-TP Part Edition"

Revision 2.00 12 2003

Midas Lab. Inc.