Microcomputer Technical Information

CP(K), O

QB-V850EIA4 In-Circuit Emulator for V850E/IA3, V850E/IA4, V850ES/IK1 Usage Restrictions		Document No.	ZBG-CD-05-0081	1/3
		Date issued	October 28, 2005	
		Issued by	Development Tool Group Multipurpose Microcomputer Systems Division 4th Systems Operations Unit NEC Electronics Corporation	
Related documents	QB-V850EIA4 User's Manual: U17167E	Notification classification	√	Usage restriction
				Upgrade
				Document modification
				Other notification

1. Target product

Product	Control Code[Note]	Remark
QB-V850EIA4-xxx-yyy	A, B, C	xxx and yyy are arbitrary codes.

Note For the identification method for the control code, see the Operating Precautions supplied with the product.

2. New restrictions

Restrictions No. 11 and No. 12 have been added. See the attachment for details.

No. 11 Illegal break occurs during program execution in internal RAM (2)

No. 12 Address is not retained during external bus access

The erroneous description on restriction No. 4 has been corrected. After correction, regard one of the restriction conditions as a permanent restriction.

- Corrected item: No. 4 Bug in program execution and DMA transfer in internal RAM
- Before correction:

[Description]

When a DMA transfer for the internal RAM and a bit manipulation instruction (SET1, CLR1, or NOT1) allocated in the internal RAM or a data access instruction for a misaligned address are executed simultaneously, the CPU may deadlock due to conflict between the internal bus operations. At this time, only a reset can be acknowledged. (An NMI or interrupt cannot be acknowledged.)

[Workaround]

Implement any of the following workarounds.

- Do not perform a DMA transfer for the internal RAM when an instruction allocated in the internal RAM is being executed.
- Do not execute an instruction allocated in the internal RAM when a DMA transfer for the internal RAM is being performed.

This bug has been corrected in control code B or later.

- After correction:

[Description]

When a DMA transfer for the internal RAM and an instruction of (1) or (2) described below are executed simultaneously, the CPU may deadlock due to conflict between the internal bus operations. At this time, only a reset can be acknowledged. (An NMI or interrupt cannot be acknowledged.)

(1) A bit manipulation instruction (SET1, CLR1, or NOT1) allocated in the internal RAM

(2) A data access instruction for a misaligned address allocated in the internal RAM

[Workaround]

Implement either of the following workarounds.

(1) For a bit manipulation instruction (SET1, CLR1, or NOT1) allocated in the internal RAM

- Do not perform a DMA transfer for the internal RAM when a bit manipulation instruction allocated in the internal RAM is being executed.
- Do not execute a bit manipulation instruction allocated in the internal RAM when a DMA transfer for the internal RAM is being performed.

This restriction has been corrected in control code B or later.

(2) For a data access instruction for a misaligned address allocated in the internal RAM

- Do not perform a DMA transfer for the internal RAM when a data access instruction for a misaligned address allocated in the internal RAM is being executed.
- Do not execute a data access instruction for a misaligned address allocated in the internal RAM when a DMA transfer for the internal RAM is being performed.

Please regard this item as a permanent restriction.

3. Workarounds

See the attachment for details on workarounds for restrictions added in this edition.

4. Modification schedule

No. 11 This item has been corrected in control code C or later.

No. 12 This item is not planned for correction.

5. List of restrictions

See the attachment for details.

6. Revision history

QB-V850EIA4 In-Circuit Emulator for V850E/IA3, V850E/IA4, V850ES/IK1 Usage Restrictions

Document Number	Issued on	Description
ZBG-CD-04-0040	August 16, 2004	Newly created.
ZBG-CD-05-0046	May 20, 2005	Addition of restriction No. 10
ZBG-CD-05-0081	October 28, 2005	Addition of restrictions No. 11 and No. 12 Correction of erroneous description in No. 4

Notes on Using QB-V850EIA4

This document describes restrictions applicable only to the emulator and restrictions that are planned for correction in the emulator.

Refer to the following documents for the restrictions in the target device.

- User’s manual of target device
- Restrictions notification document for target device

Also refer to the user’s manual of the emulator for cautions on using the emulator.

1. Product Version

Control Code[Note]	Remark
A	–
B	Version in which bugs No. 1 to No. 4 and No. 6 to No. 9 in control code A are corrected
C	Version in which bugs No. 10 and No. 11 in control code B is corrected

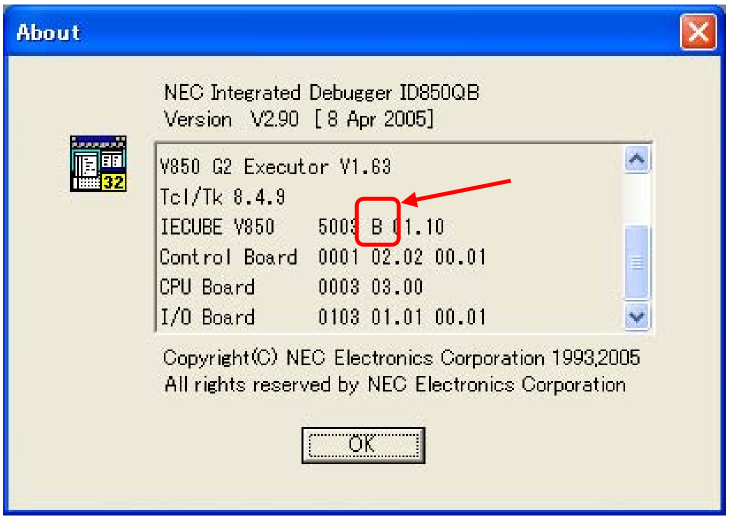
Note The “control code” is the second digit from the left in the 10-digit serial number printed on the sticker attached to the bottom side of IECUBE (if it has not been upgraded).

If the product has been upgraded, the control code can also be checked with the following methods while the debugger is running.

- When using ID850QB

  Click the [Help] menu and then click the About submenu to display the About dialog box.

  “X” in “IECUBE V850 **** X **.**” is the control code.

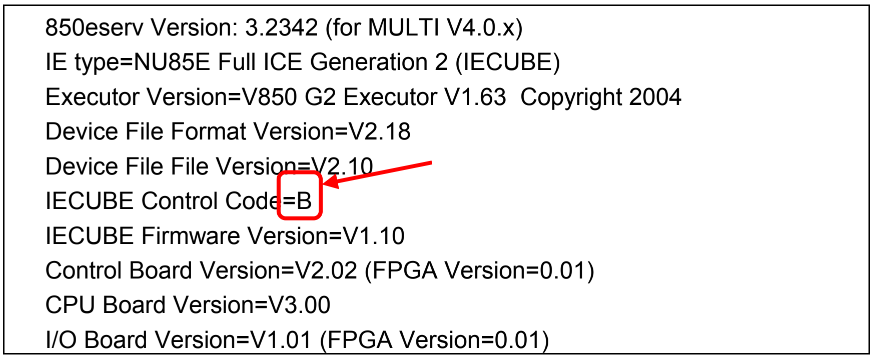


```
NEC Integrated Debugger ID850QB
Version  V2.90  [ 8 Apr 2005]

V850 G2 Executor V1.63
Tcl/Tk 8.4.9
IECUBE V850     5003 B 01.10
Control Board   0001 02.02 00.01
CPU Board       0003 03.00
I/O Board       0103 01.01 00.01

Copyright(C) NEC Electronics Corporation 1993,2005
All rights reserved by NEC Electronics Corporation
```

- When using Green Hills Software^{TM} (GHS)’s debugger MULTI$^{®}$

  Execute the version command of 850eserv.

  “X” in “IECUBE Control Code=X” is the control code.



```
850eserv Version: 3.2342 (for MULTI V4.0.x)
IE type=NU85E Full ICE Generation 2 (IECUBE)
Executor Version=V850 G2 Executor V1.63  Copyright 2004
Device File Format Version=V2.18
Device File File Version=V2.10
IECUBE Control Code=B
IECUBE Firmware Version=V1.10
Control Board Version=V2.02 (FPGA Version=0.01)
CPU Board Version=V3.00
I/O Board Version=V1.01 (FPGA Version=0.01)
```

2. Product History

No.	Bugs and Changes/Additions to Specifications		Control Code		
			A	B	C
1	Bug in accessing UAnRX register during break		×	√	√
2	Bug in accessing CBnRX register during		×	√	√
3	Bug related to DMA transfer forcible termination		×	√	√
4	Bug in program execution and DMA transfer in internal RAM	(1) Bit manipulation instruction	×	√	√
		(2) Access for misaligned address	Permanent restriction		
5	Illegal break occurs during program execution in internal RAM (1)		Permanent restriction		
6	Restriction on reset input during a break		×	√	√
7	Emulator hangs up upon internal reset		×	√	√
8	Restriction related to dead time of motor control function		×	√	√
9	Restriction on timer M (TMM)		×	√	√
10	Bugs in A/D conversion function during a break		×	×	√
11	Fail-safe break function does not operate		×	×	√
12	Illegal break occurs during program execution in internal RAM (2)		Permanent restriction		

×: Applicable, √: Not applicable or already corrected

3. Details of Bugs and Added Specifications

No. 1 Bug in accessing UAnRX register during break

[Description]

An overrun error occurs under the following conditions (a) to (c).

(a) If a break occurs after reading the UART receive buffer register (UAnRX) and the UAnRX register is displayed in the I/O register window of the debugger, an overrun error occurs when UART reception is performed next time.

(b) If a software break occurs immediately after reading the UART receive buffer register (UAnRX), an overrun error occurs when UART reception is performed next time regardless of whether or not the UAnRX register is displayed in the I/O register window.

(c) If a DMA transfer from the UART receive buffer register (UAnRX) is performed during a break[Note], an overrun error occurs when UART reception is performed next time.

Note Including breaks by the RAM monitor function or DMM function. However, the real-time RAM monitor function does not cause this bug because it does not set breaks.

Remarks 1. An overrun error also occurs when UART receives data multiple times during a break. This is an emulator specification.

2. n = 0 or 1

[Workaround]

(a) Do not display the UAnRX register in the I/O register window.

(b) Set a hardware break when setting a break immediately after reading the UAnRX register.

(c) There is no workaround.

This bug has been corrected in QB-V850EIA4 control code B or later and the device files with the following versions.

Target Device	Device File Name (Package Name)	Device File Version (Package Version)
V850E/IA4	DF703186	V2.00 or later
V850E/IA3		
V850ES/IK1	DF703329	V1.00 or later

In the corrected versions, the value of the UART receive buffer register (UAnRX) is indicated by an asterisk (*) in the I/O register window. To refer to the value, point to the UAnRX register in the I/O register window, and select “Compulsion Read” from the right-click menu.

Remark n = 0 or 1

No. 2 Bug in accessing CBnRX register during break

[Description]

When the CSIBn receive data register (CBnRX) is read, it usually starts the next reception operation. Under the following conditions (a) and (b), however, the next reception operation is not started even if CBnRX is read.

(a) If a software break occurs immediately after reading the CSIBn receive data register (CBnRX).

(b) If a DMA transfer from the CSIBn receive data register (CBnRX) is performed during a break[Note].

As a result, communication stops or the DMA controller stops.

Note Including breaks by the RAM monitor function or DMM function. However, the real-time RAM monitor function does not cause this bug because it does not set breaks.

Remark n = 0 or 1 in the V850E/IA4 and V850E/IA3, n = 0 in the V850ES/IK1

[Workaround]

(a) Set a hardware break when setting a break immediately after reading the CBnRX register.

(b) There is no workaround.

This bug has been corrected in QB-V850EIA4 control code B or later and the device files with the following versions.

Target Device	Device File Name (Package Name)	Device File Version (Package Version)
V850E/IA4	DF703186	V2.00 or later
V850E/IA3		
V850ES/IK1	DF703329	V1.00 or later

In the corrected versions, the value of the CSIBn receive data register (CBnRX) is indicated by an asterisk (*) in the I/O register window. To refer to the value, point to the CBnRX register in the I/O register window, and select “Compulsion Read” from the right-click menu.

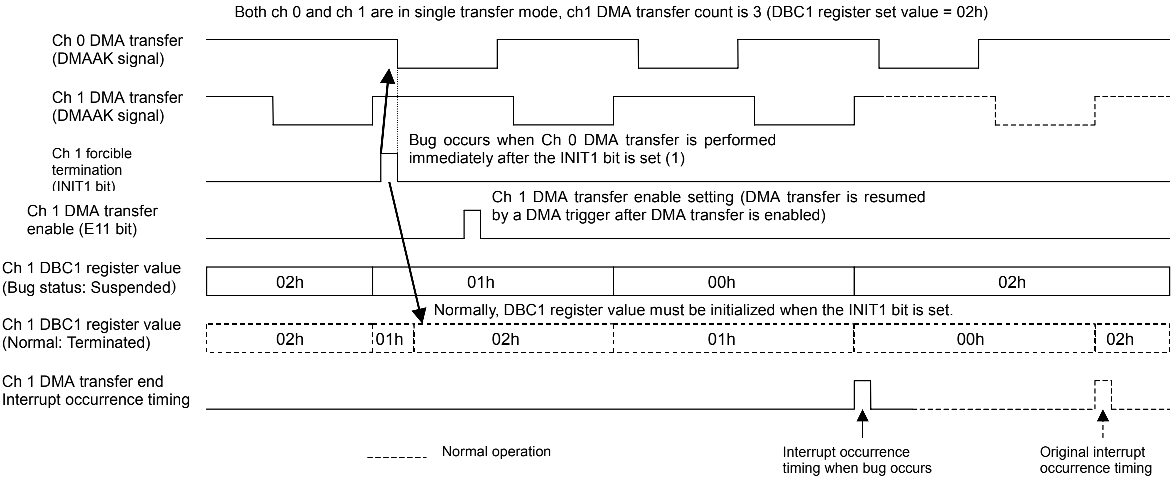
Remark n = 0 or 1 in the V850E/IA4 and V850E/IA3, n = 0 in the V850ES/IK1

No. 3 Bug related to DMA transfer forcible termination

[Description]

When terminating a DMA transfer by setting the INITn bit of the DCHCn register, the transfer may not be terminated, but just suspended, even though the INITn bit is set (1). As a result, when the DMA transfer of a channel that should have been terminated is resumed, the DMA transfer will terminate after an unexpected number of transfers are completed and a DMA transfer completion interrupt may occur (n = 0 to 3). This bug occurs if a DMA transfer is executed immediately after a forcible termination is set (by setting the INITn bit) (see the figure below).

This bug does not depend on the number of transfer channels, transfer type (2-cycle or flyby), transfer target (between memory and memory, memory and I/O; including internal resources), transfer mode (single, single-step, or block), or trigger (external request, interrupt from internal peripheral I/O, or software), and can occur with any combination of the above elements that can be set under the specifications. In addition, another channel may affect the occurrence of this bug.


Both ch 0 and ch 1 are in single transfer mode, ch1 DMA transfer count is 3 (DBC1 register set value = 02h)
Ch 0 DMA transfer (DMAAK signal)
Ch 1 DMA transfer (DMAAK signal)
Ch 1 forcible termination (INIT1 bit)
Bug occurs when Ch 0 DMA transfer is performed immediately after the INIT1 bit is set (1)
Ch 1 DMA transfer enable (E11 bit)
Ch 1 DMA transfer enable setting (DMA transfer is resumed by a DMA trigger after DMA transfer is enabled)
Ch 1 DBC1 register value (Bug status: Suspended)
02h
01h
00h
02h
Normally, DBC1 register value must be initialized when the INIT1 bit is set.
Ch 1 DBC1 register value (Normal: Terminated)
02h
01h
02h
01h
00h
02h
Ch 1 DMA transfer end Interrupt occurrence timing
Normal operation
Interrupt occurrence timing when bug occurs
Original interrupt occurrence timing

The following registers are buffer registers with a 2-stage FIFO configuration of master and slave. If these registers are overwritten during a DMA transfer or in the DMA-suspended status, the value is written to the master register, and reflected in the slave register when the DMA transfer of the overwritten channel is terminated.

The “initialization” in the above figure means that the contents of the master register are reflected in the slave register.

2-stage FIFO configuration registers (n = 0 to 3):

- DMA source address register (DSAnH, DSAnL)
- DMA destination address register (DDAnH, DDAnL)
- DMA transfer count register (DBCn)

This bug applies only when using the V850E/IA4 and V850E/IA3; it does not apply when using the V850ES/IK1 (because the DMA function is not provided in the V850ES/IK1).

[Workaround]

This bug can be avoided by implementing one of the following procedures using the software.

(1) Stop all the transfers from DMA channels temporarily (n = 0 to 3)

The following measure is effective if the following condition is satisfied.

- Except for the following workaround processing, the program does not assume that the TCn bit of the DCHCn register is 1. (Since the TCn bit of the DCHCn register is cleared (0) when it is read, execution of the following procedure (b) under <5> clears this bit.)

[Procedure to avoid bug]

<1> Disable interrupts (DI state).

<2> Read the DMA restart register (DRST) and transfer the ENn bit of each channel to a general-purpose register (value A).

<3> Write 00H to the DMA restart register (DRST) twice[Note].

By executing twice[Note], the DMA transfer is definitely stopped before proceeding to <4>.

<4> Set (1) the INITn bit of the DCHCn register of the channel that should be terminated forcibly.

<5> Perform the following operations for value A read in <2>. (Value B)

(a) Clear (0) the bit of the channel that should be terminated forcibly

(b) If the TCn and ENn bits of the channel that is not terminated forcibly are 1 (AND makes 1), clear (0) the bit of the channel.

<6> Write value B in <5> to the DRST register.

<7> Enable interrupts (EI state).

Note Execute three times if the transfer target (transfer source or transfer destination) is the internal RAM.

Remark Be sure to execute <5> to prevent the ENn bit from being set illegally for channels that are terminated normally during the period of <2> and<3>.

(2) Repeat setting the INITn bit until the forcible DMA transfer termination is correctly performed (n = 0 to 3)

[Procedure to avoid bug]

<1> Copy the initial transfer count of the channel that should be terminated forcibly to a general-purpose register.

<2> Set (1) the INITn bit of the DCHCn register of the channel that should be terminated forcibly.

<3> Read the value of the DMA transfer count register (DBCn) of the channel that should be terminated forcibly and compare the value with the one copied in <1>. If the values do not match, repeat <2> and <3>.

Remarks 1. When the DBCn register is read in procedure <3>, the remaining transfer count will be read if the DMA is stopped due to this bug. If the forcible DMA termination is performed correctly, the initial transfer count will be read.

2. Note that it may take some time for forcible termination to take effect if this workaround is implemented in an application in which DMA transfers other than for channels subject to forcible termination are frequently performed.

This bug has been corrected in control code B or later.

No. 4 Bug in program execution and DMA transfer in internal RAM

[Description]

When a DMA transfer for the internal RAM and an instruction of (1) or (2) described below are executed simultaneously, the CPU may deadlock due to conflict between the internal bus operations. At this time, only a reset can be acknowledged. (An NMI or interrupt cannot be acknowledged.)

(1) A bit manipulation instruction (SET1, CLR1, or NOT1) allocated in the internal RAM

(2) A data access instruction for a misaligned address allocated in the internal RAM

This bug applies only when using the V850E/IA4 and V850E/IA3; it does not apply when using the V850ES/IK1 (because the DMA function is not provided in the V850ES/IK1).

[Workaround]

Implement either of the following workarounds.

(1) For a bit manipulation instruction (SET1, CLR1, or NOT1) allocated in the internal RAM

- Do not perform a DMA transfer for the internal RAM when a bit manipulation instruction allocated in the internal RAM is being executed.
- Do not execute a bit manipulation instruction allocated in the internal RAM when a DMA transfer for the internal RAM is being performed.

This restriction has been corrected in control code B or later.

(2) For a data access instruction for a misaligned address allocated in the internal RAM

- Do not perform a DMA transfer for the internal RAM when a data access instruction for a misaligned address allocated in the internal RAM is being executed.
- Do not execute a data access instruction for a misaligned address allocated in the internal RAM when a DMA transfer for the internal RAM is being performed.

Please regard this item as a permanent restriction.

No. 5 Illegal break occurs during program execution in internal RAM (1)

[Description]

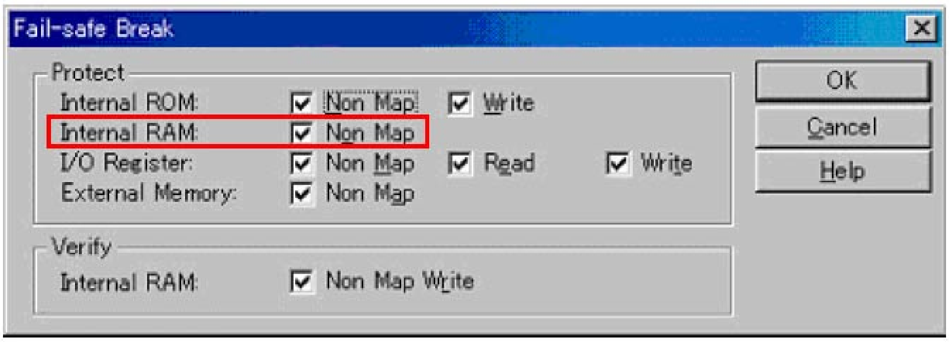
An illegal break may occur when a peripheral I/O register is accessed during program execution in the internal RAM.

[Workaround]

Cancel the fail-safe break setting for the internal RAM in the debugger.

- When using ID850QB

Click the [Detail] button in the Fail-safe Break field in the Configuration window and clear the check box for “Internal RAM”.



Fail-safe Break

Protect

Internal ROM: ☑ Non Map ☑ Write

Internal RAM: ☑ Non Map

I/O Register: ☑ Non Map ☑ Read ☑ Write

External Memory: ☑ Non Map

Verify

Internal RAM: ☑ Non Map Write

- When using MULTI

Cancel the fail-safe break for “ramgrd” and “ramgrdv” using the *Target flsf* command.

Please regard this item as a permanent restriction.

No. 6 Restriction on reset input during a break

[Description]

The QB-V850EIA4 may hang up if a break occurs when the RESET pin is active (low level).

[Workaround]

Mask the RESET pin using the pin mask function of the debugger.

This restriction has been corrected in control code B or later.

No. 7 Emulator hangs up upon internal reset

[Description]

The emulator may hang up when a reset is generated by the watchdog timer.

[Workaround]

Implement either of the following workarounds.

(1) Stop the watchdog timer after reset.

(2) Mask the RESET pin using the pin mask function of the debugger.

This restriction has been corrected in control code B or later.

No. 8 Restriction related to dead time of motor control function

[Description]

When a conflict occurs between the timing at which a match occurs in the TMQn capture/compare register (TQnCCRm), which sets the duty of timer output by the motor control function, and the timing at which a match occurs in the TMQn dead time compare register (TQnDTC), timer output (TOQnTn, TOQnBm) is inverted at the match timing in the TMQ0 capture/compare register.

It is also inverted when a conflict occurs between the timing at which a compare match occurs in TQnCCRm and the timing at which a match occurs in the TMQn dead time compare register (TQnDTC), due to overwriting the TMQn capture/compare register (TQnCCRm) using the reload function.

(n = 0 or 1 and m = 1 to 3 when using V850E/IA4)
(n = 0 and m = 1 to 3 when using V850E/IA3)
(n = 1 and m = 1 to 3 when using V850ES/IK1)

[Workaround]

Set an odd number to the TMQn dead time compare register (TQnDTC) and an even number to the TMQn capture/compare register (TQnCCRm). When using 0% PWM output, set an odd number to TMQn capture/compare register 0 (TQnCCR0) for setting the cycle, and the TQnCCR0 register value + 1 (even number) to the TQnCCRm register.
(n = 0 or 1 and m = 1 to 3 when using V850E/IA4)
(n = 0 and m = 1 to 3 when using V850E/IA3)
(n = 1 and m = 1 to 3 when using V850ES/IK1)

This restriction has been corrected in control code B or later.

No. 9 Restriction on timer M (TMM)

[Description]

When TMM compare register 0 (TMM0CMP0) is set to FFFFH and a count operation is enabled (TM0CE = 1), a compare match interrupt (INTTM0EQ0) occurs when the count operation starts.

[Workaround]

Do not set TMM compare register 0 (TMM0CMP0) to FFFFH. Set a value in a range from 0H to FFFEH.

This restriction has been corrected in control code B or later.

No. 10 Bugs in A/D conversion function during a break

[Description]

(1) A/D conversion does not start if any one of the following conditions <a> to <c> is satisfied in peripheral break mode (mode in which peripheral functions are stopped during a break). In addition, no interrupt requests are generated upon completion of the A/D conversion.

<a> A break occurs from when an A/D conversion start trigger is generated[Note 1] until the execution of two instructions ends[Note 2].

Example: In software trigger mode

```
* set1 0x7, ADA0M0  ⎫
* nop               ⎬ A/D conversion does not start if a break occurs
* nop               ⎭ during this period.
* nop               } If a break occurs after this point, A/D conversion starts normally.
                      (Caution must still be exercised for the bugs described in (2) and
                      (3).)
```

<b> If execution is started using an A/D conversion start instruction in software trigger mode, and a software break or a break before execution is set to the instruction.

Example:

```
B set1 0x7, ADA0M0   ← A/D conversion does not start if an attempt is made to
* nop                  start A/D conversion using the instruction in this line.
```

<c> A break occurs while an A/D conversion operation is stopped, and an attempt is made to start A/D conversion during this break[Note 3].

(2) If a break occurs[Note 2] during A/D conversion in peripheral break mode, a write is performed[Note 5] on an A/D-related register[Note 4], and the A/D conversion is re-executed, then conversion is performed once or twice with the values before the writing. (If the break occurs in normal conversion operation mode, A/D conversion may be performed twice with the values before the writing.) After this conversion is completed, A/D conversion starts with the values after the writing. Consequently, an invalid A/D conversion result is obtained and it seems as though invalid interrupts occur once or twice for the operation. (Normally, re-conversion is performed immediately after re-execution with values newly set to the A/D-related register.)

(3) If a break occurs during A/D conversion in peripheral break mode, the A/D conversion result immediately after re-execution is invalid. Moreover, if a break occurs during A/D conversion in high-speed conversion mode, and the ADA0CE bit is cleared and re-set during the break, then the result of the subsequent one A/D conversion operation is invalid.

Notes **1.** Starting conversion by DMA transfer, external trigger, and timer trigger are included in this condition, in addition to starting conversion triggered by instruction execution.

2. Includes the following break sources.

- Step execution
- Fail-safe break
- RAM monitoring (real-time RAM monitoring does not apply)
- DMM
- Change of event while the program is running

Among these sources, RAM monitoring, DMM, and a change of event while the program is running is implemented through an instantaneous break, so the actual break point cannot be specified, and thus the A/D conversion unexpectedly becomes invalid.

3. DMA transfer, external trigger, and timer trigger are included in this condition, in addition to a write access to the ADA0CE bit in the IO register window.

4. A/D-related registers: ADA0M0, ADA0M1, ADA0M2, ADA0S, ADA0PFT, and ADA0PFM

5. Cases such that the write setting is applied in the IO register window, or through DMA transfer.

[Workaround]

Do not set peripheral break mode if you want to avoid this bug entirely, or observe all of the following.

- Do not set breaks between the A/D conversion start trigger and the end of A/D conversion.
- Do not perform step execution of an A/D conversion start instruction in software trigger mode.
- Do not perform write accesses to A/D-related registers during a break.
- Disable the RAM monitoring function.
- Do not use DMM.
- Do not change events while the program is running.

This restriction has been corrected in control code C or later.

In control code C or later, the A/D conversion function does not stop during a break even if the peripheral break mode is set.

No. 11 Fail-safe break function does not operate

[Description]

The fail-safe break function does not operate for the following operations.

- Peripheral I/O register
  - Read or write to a peripheral I/O register that does not exist
  - Write to a read-only peripheral I/O register
  - Read to a write-only peripheral I/O register
- External memory area
  - Fetch from or access to an unmapped area
  - Write to ROM area

[Workaround]

There is no workaround.

This restriction has been corrected in control code C or later.

No. 12 Illegal break occurs during program execution in internal RAM (2)

[Description]

A non-map break occurs if all of the following conditions are satisfied, even if the program itself is correct.

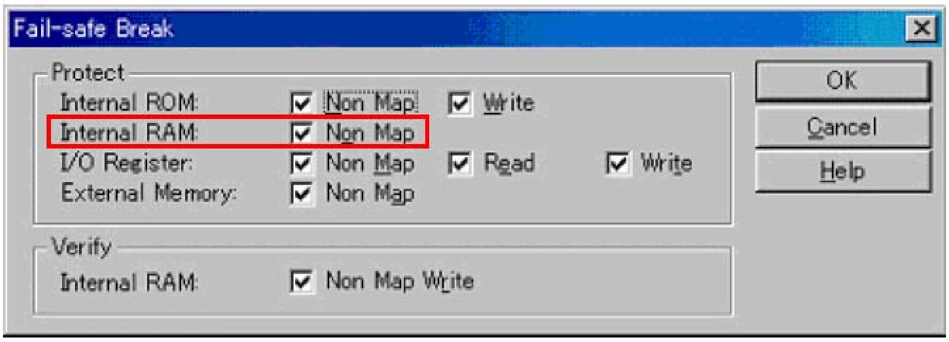
- A program is executed in the internal RAM area.
- Data access for the internal RAM area is performed twice in succession.
- An execution branches to the internal ROM area using a JR or JARL instruction immediately after the above successive data access, or one NOP instruction after the above successive data access.

[Workaround]

Implement either of the following workarounds.

- Cancel the fail-safe break setting for the internal RAM in the debugger.
  - When using ID850QB

    Click the [Detail] button in the Fail-safe Break field in the Configuration window and clear the check in the check box for "Internal RAM".



Fail-safe Break

Protect

Internal ROM: ☑ Non Map ☑ Write

Internal RAM: ☑ Non Map

I/O Register: ☑ Non Map ☑ Read ☑ Write

External Memory: ☑ Non Map

Verify

Internal RAM: ☑ Non Map Write

- When using MULTI

Cancel the fail-safe break for “ramgrd” and “ramgrdv” using the Target flsf command.

- Insert two or more NOP instructions between the successive data access for the internal RAM area and an instruction to branch to the internal ROM area.

Please regard this item as a permanent restriction.

4. Cautions

4.1 General Cautions for Handling This Product

(a) Circumstances not covered by product guarantee

- If the product was disassembled, altered, or repaired by the customer
- If it was dropped, broken, or given another strong shock
- Use at overvoltage, use outside guaranteed temperature range, storing outside guaranteed temperature range
- If the power was turned on while the AC adapter, USB interface cable, or target system connection was in an unsatisfactory state
- If the AC adapter cable, USB interface cable, emulation probe, or the like was bent or pulled excessively
- If an AC adapter other than the one supplied with the product is used
- If the product got wet
- If the product and target system were connected while a potential difference existed between the GND of the product and the GND of the target system
- If a connector or cable was removed while the power was being supplied to the product
- If an excessive load was placed on a connector or socket
- If a metal part of the power switch, cooling fan, or another such part comes in contact with an electrostatic charge

(b) Safety precautions

- If used for a long time, the product may become hot (50°C to 60°C). Be careful of low temperature burns and other dangers due to the product becoming hot.
- Be careful of electrical shock. There is a danger of electrical shock if the product is used as described above in **(a) Circumstances not covered by product guarantee**.

4.2 Cautions on Extension Probe

- When using the extension probe, there is a restriction on the maximum operating frequency at which a high-speed signal such as a clock or external bus can be propagated. (See the table below.)

  In the QB-V850EIA4, the extension probe can be used at the maximum operating frequency because its target device (V850E/IA4, V850E/IA3, or V850ES/IK1) does not have the external bus function. Note, however, that a clock signal of 32 MHz or higher cannot be propagated to the extension probe.

Use of Clock Signal (CLKOUT, BUSCLK, SDCLK, etc.)	Use of External Bus	Upper Limit of Frequency When Using Extension Probe
Used	Used	32 MHz
	Not used	
Not used	Used	64 MHz
	Not used	80 MHz

- An impedance of approx. 50 Ω is applied to the extension probe.

- The signal level decreases by approx. 0.1 V when it passes through the extension probe. This degrades the precision of analog signal propagation upon A/D conversion, etc.

- A delay of approx. 5 ns (propagation delay) occurs when a signal passes through the extension probe.

  Consequently, it may be necessary to set data waits or address waits when using the external bus.

- Be sure to connect IECUBE and the target to the GND line of the extension probe when using the extension probe; otherwise the level of the propagated signal may degraded.

5. Optional Functions

The following functions can be added to the QB-V850EIA4 (control code C or later). This chapter explains the functional outline and specifications of the optional functions, and how to obtain them.

- Coverage measurement function
- TimeMachine^{TM} function

The support status of each optional function differs depending on the debugger used. The following table lists the support statuses as of October 2005. If you have any questions regarding the support status, consult an NEC Electronics sales representative or distributor.

Function	Support Status	
	ID850QB	MULTI
Coverage measurement function	Supported in V2.90, V3.10 and later	Support under consideration
TimeMachine function	Not supported	Supported in 850eserv2 V1.000 and later

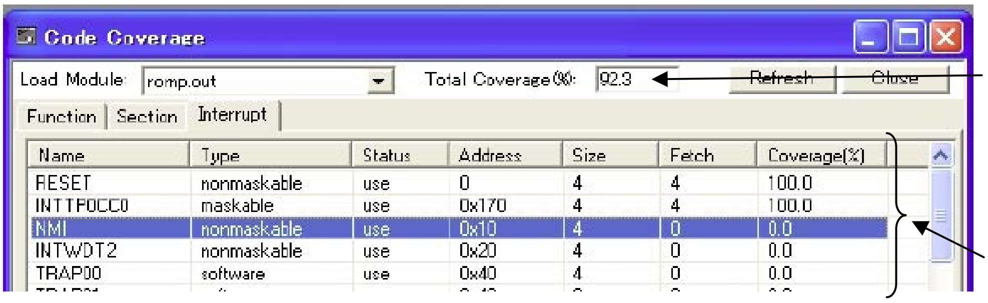
5.1 Coverage Measurement Function

This section explains the functional outline of the coverage measurement function and differences in specifications that occur after the addition of this function.

5.1.1 Functional outline

The coverage measurement function is used to measure the percentage of the executed code in a load module, section, or other such area. After the addition of this function, the Code Coverage window will be added and the Source and Assemble windows will be modified in the debugger ID850QB, as follows.

- Code Coverage window



Load Module: romp.out

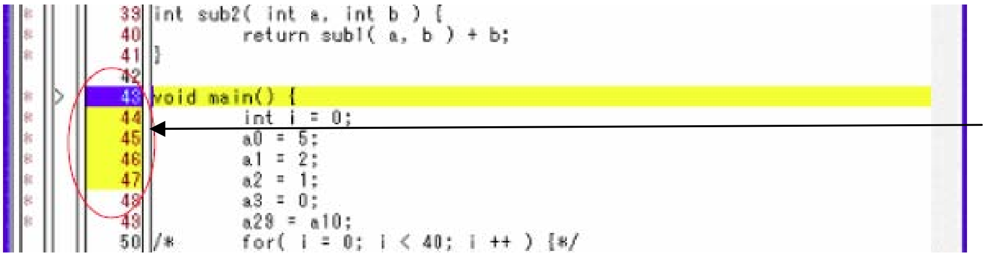
Total Coverage(%): 92.3

Name	Type	Status	Address	Size	Fetch	Coverage(%)
RESET	nonmaskable	use	0	4	4	100.0
INTTP0CC0	maskable	use	0x170	4	4	100.0
NMI	nonmaskable	use	0x10	4	0	0.0
INTWDT2	nonmaskable	use	0x20	4	0	0.0
TRAP00	software	use	0x40	4	0	0.0

Displays the coverage (%) of the executed code for total code in a load module.

Displays the coverage (%) of the executed code separately by functions, sections and vectors.

- Source window and Assemble window



```
int sub2( int a, int b ) {
        return sub1( a, b ) + b;
}

void main() {
        int i = 0;
        a0 = 5;
        a1 = 2;
        a2 = 1;
        a3 = 0;
        a28 = a10;
/*      for( i = 0; i < 40; i ++ ) {*/
```

The executed lines are highlighted.

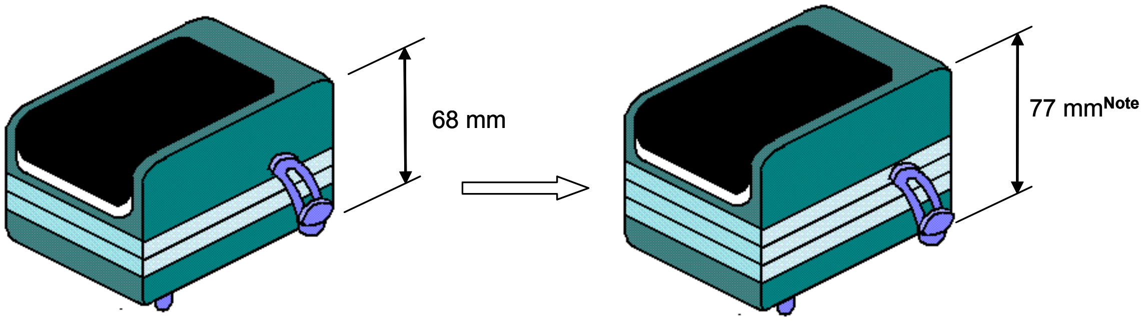
Refer to the user's manual for the debugger for details on use of the coverage measurement function.

5.1.2 Differences from hardware specifications

After addition of the coverage measurement function, differences from the hardware specifications described in the QB-V850EIA4 User's Manual (U17167E) are as follows.

- External dimensions

  The height increases by 9 mm.



Note When the rear spacer is adjusted to the lowest height (107 mm max.)

- Weight

  The weight increases by approximately 70 g.

5.1.3 Differences from system specifications

After addition of the coverage measurement function, replace the following description with the item "Coverage function" in "Table 1-2 QB-V850EIA4 System Specifications" that is described in the QB-V850EIA4 user's manual (U17167E) for reference.

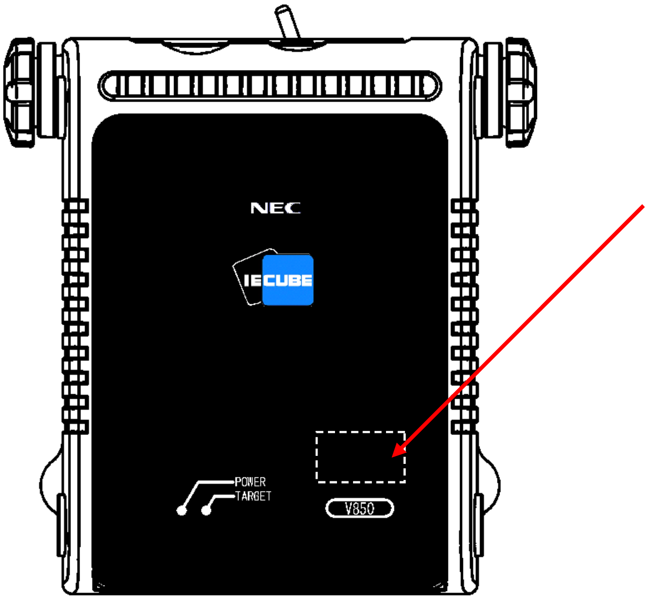
Item		Specifications
Coverage function		Detection of execution or pass
	Measured range	Internal ROM space + arbitrary 1 MB space

5.2 TimeMachine Function

This function is supported by the Green Hills Software (GHS) debugger. For details on the functional outline and specifications, consult a GHS tool distributor.

5.3 Changes to Top Side of Product Consequent to Addition of Optional Functions

After the addition of the optional functions, the following stickers will be attached to the top of IECUBE. The addition of the optional functions can be confirmed through the presence of these stickers.



IECUBE Top View

A sticker is attached to this position according to the function added, as follows.

For the coverage measuring function:

COV

For the TimeMachine function:

STP

5.4 How to Add Optional Functions

To add the optional functions, the option board corresponding to each function, as listed in the following, must be mounted.

Function	Option Board Required for Adding Function
Coverage measurement function	Coverage memory board[Note 1]
TimeMachine function	SuperTrace™ probe board[Notes 1, 2]

Notes 1. Either the coverage memory board or the SuperTrace probe board can be added, but not both.

2. To use the TimeMachine function, the SuperTrace Probe (Green Hills Software (GHS)) must be mounted in IECUBE, in addition to the SuperTrace probe board.

The following two methods have been provided for mounting the option boards.
For more information on ordering, price and schedule, consult an NEC Electronics sales representative or distributor.

- New purchase

  By adding one of the following suffixes at the end of the ordering code, you can purchase IECUBE with the corresponding option board mounted.

  -C: Coverage memory board mounted
  -S: SuperTrace Probe board mounted

  Part number example: QB-V850EIA4-ZZZ-S

- System upgrade

  Using this method, the option board can be mounted in your IECUBE.