Microcontroller Technical Information

CP(K), O

QB-V850ESFX3 In-Circuit Emulator for V850ES/FE3, V850ES/FF3, V850ES/FG3, V850ES/FJ3, V850ES/FK3 Usage Restrictions		Document No.	ZBG-CD-07-0004		1/2
		Date issued	January 17, 2007		
		Issued by	Development Tool Group Multipurpose Microcomputer Systems Division 4th Systems Operations Unit NEC Electronics Corporation		
Related documents	QB-V850ESFX3 User's Manual: ZUD-CD-06-0062	Notification classification	√	Usage restriction	
				Upgrade	
				Document modification	
				Other notification	

1. Affected product

Product	Outline	Control Code[Note]
QB-V850ESFX3	In-circuit emulator for V850ES/FE3, V850ES/FF3, V850ES/FG3, V850ES/FJ3, V850ES/FK3	A, B

2. New restriction

Restrictions No. 13 and No. 14 have been added. See the attachment for details.

3. Workaround

See the attachment for details.

4. Modification schedule

Products in which No. 14 is corrected are scheduled for release as follows.

Newly shipped products: Shipments as of February 2007 (control code: C)

Upgrade for already shipped products: Available from February 14, 2007
(Upgrade by sending the product to NEC Electronics)

* Note that this schedule is subject to change without notice. For the detailed release schedule of modified products, contact an NEC Electronics sales representative.

4. List of restrictions

See the attachment.

Note The "control code" is the second digit from the left in the 10-digit serial number.
If the product has been upgraded, the control code can be checked in the About dialog box in the ID850-QB.
"X" in version information "IECUBE V850 **** X F/W: V*. **" is the control code.

5. Document revision history

QB-V850ESFX3 In-Circuit Emulator for V850ES/FE3, FF3, FG3, FJ3, FK3 - Usage Restrictions

Document Number	Issued on	Description
ZBG-CD-07-0004	January 17, 2007	Newly created. Addition of new bugs (No. 13 and No. 14)

Notes on Using QB-V850ESFX3

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.

Programming to the V850ES/FE3, V850ES/FF3, V850ES/FG3, V850ES/FJ3 or V850ES/FK3 using the flash programmer PG-FPL, supplied with the QB-V850ESFX3, has been supported in GUI Ver. 1.50 and later. Programming using the QB-MINI2 (MINICUBE2) has been supported in QB-Programmer Ver. 2.00 and later and F/W Ver. 4.00 and later.

1. Product Version

Control Code[Note]	Remark
A	–
B	–
C	–

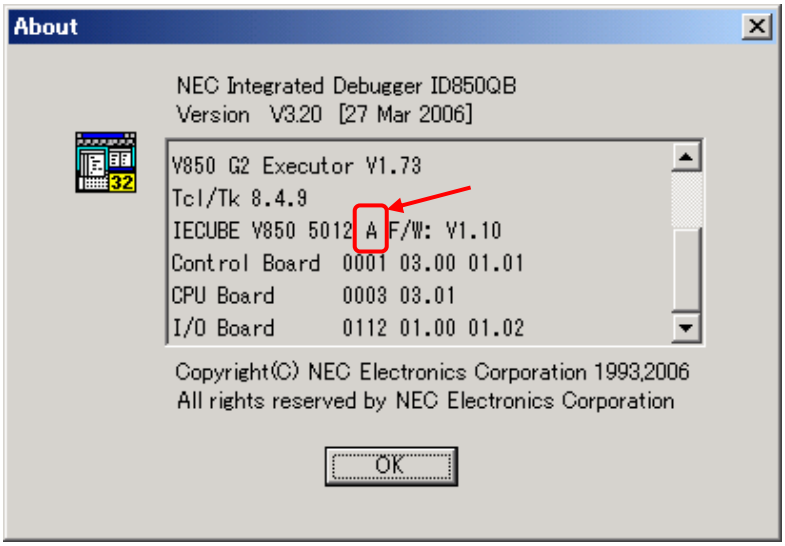
Note The "control code" is the second digit from the left in the 10-digit serial number printed on the sticker attached to the bottom side of IECUBE (if it has not been upgraded).

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 F/W: V*.**" is the control code.



- When using Green Hills Software™ (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=A
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 CnRGPT register during break (n = 0 to 3)	Permanent restriction		
2	Bug in accessing CnTGPT register during break (n = 0 to 3)	Permanent restriction		
3	Bug in accessing CnGNCTRL register during break (n = 0 to 3)	Permanent restriction		
4	Bug in program execution and DMA transfer in internal RAM	Permanent restriction		
5	Emulator hangs up while downloading data or setting software break	Avoidable by debugger upgrade		
6	Data loss occurs when external RAM is connected	Avoidable by debugger upgrade		
7	Restriction on emulation of POC circuit and clock monitor	Avoidable by debugger upgrade		
8	Illegal break occurs during program execution in internal RAM (1)	Permanent restriction		
9	Restriction on reset input during a break	Permanent restriction		
10	Bug that occurs upon entering and releasing STOP mode when RESET pin is masked	Permanent restriction		
11	Illegal break occurs during program execution in internal RAM (2)	Permanent restriction		
12	Bug in LOCK bit value of LOCKR register	×	○	○
13	Bug in accessing CBnRX register during break (n = 0 to 2)	Permanent restriction		
14	Bug in subsystem clock oscillator	×	×	○

×: Bug applicable, ○: Bug not applicable or already corrected

3. Details of Bugs and Added Specifications

No. 1 Bug in accessing CnRGPT register during break (n = 0 to 3)

[Description]

Under the following conditions (a) and (b), the read pointer (RGPT) that should be incremented is not incremented, and the same data as previously read is read.

(a) If a software break occurs immediately after reading the CANn module receive history list register (CnRGPT).

(b) If a DMA transfer from the CANn module receive history list register (CnRGPT) is performed during a break[Note].

Note Including breaks by the RAM monitor function, DMM function, step-wise execution, fail-safe break, or breaks due to event change while the program is running. The real-time RAM monitor function does not cause this bug because it does not set breaks.

[Workaround]

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

(b) There is no workaround.

Please regard items (a) and (b) as permanent restrictions.

No. 2 Bug in accessing CnTGPT register during break (n = 0 to 3)

[Description]

Under the following conditions (a) and (b), the read pointer (TGPT) that should be incremented is not incremented, and the same data as previously transmitted is transmitted.

(a) If a software break occurs immediately after reading the CANn module transmit history list register (CnTGPT).

(b) If a DMA transfer from the CANn module transmit history list register (CnTGPT) is performed during a break[Note].

Note Including breaks by the RAM monitor function, DMM function, step-wise execution, fail-safe break, or breaks due to event change while the program is running. The real-time RAM monitor function does not cause this bug because it does not set breaks.

[Workaround]

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

(b) There is no workaround.

Please regard items (a) and (b) as permanent restrictions.

No. 3 Bug in accessing CnGNCTRL register during break (n = 0 to 3)

[Description]

When a register access is performed in the following sequence, a forcible shutdown that should not take place normally may occur after the sequence is complete.

[Sequence for bug occurrence]

(1) The EFSD bit of the CANn module control register (CnGMCTRL) is set.

(2) The I/O register[Note] is accessed.

(3) The GOM bit of the CANn module control register (CnGMCTRL) is cleared.

Note I/O register access except for clearing the GOM bit of the CnGMCTRL register

Conditions under which a forcible shutdown takes place are shown below.

(a) If a break occurs immediately after the I/O register access in (2) occurs

(b) If a break by the RAM monitor function or DMM function occurs immediately after the I/O register access in (2) occurs

(c) Stepwise execution is performed for the I/O register access in (2)

[Workaround]

Be sure to set the EFSD bit and clear the GOM bit successively when executing a forcible shutdown.

Do not perform register access in the above sequence when not performing a forcible shutdown.

Please regard this item as a permanent restriction.

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

[Description]

When a data access instruction for a misaligned address allocated in the internal RAM and DMA transfer for the internal RAM 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 either of the following workarounds.

(1) 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.

(2) 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 Emulator hangs up while downloading data or setting software break

[Description]

The emulator may hang up if the WAIT pin or HLDRQ pin is at the active level while data is being downloaded to the internal ROM area or a software break is set to the internal ROM area.

[Workaround]

If the WAIT and HLDRQ pins are not used, mask the WAIT and HLDRQ pins using the pin mask function of the debugger.

If the WAIT and HLDRQ pins are used, do not make these pins active while data is being downloaded to the internal ROM area or a software break is set to the internal ROM area.

This restriction can be avoided by upgrading the debugger to the following version.

- When using ID850QB: Use Ver. 2.92 or later.
- When using MULTI: Use in combination with EXEC Ver. 1.57 or later.

No. 6 Data loss occurs when external RAM is connected

[Description]

When there is external RAM in the target system and the bus control pins are active, the data in the CS0 area (0x100000 to 0x1FFFFF) in the external RAM may be overwritten by downloading a program to the internal ROM area or by setting a software break in the internal ROM.

[Workaround]

Initialize the external RAM value by running the downloaded program.

Do not use a software break for the internal ROM space; use a hardware break instead.

This restriction can be avoided by upgrading the debugger to the following version.

- When using ID850QB: Use Ver. 2.92 or later.
- When using MULTI: Use in combination with EXEC Ver. 1.57 or later.

No. 7 Restriction on emulation of POC circuit and clock monitor

[Description]

The POC circuit and the clock monitor cannot be emulated.

[Workaround]

There is no workaround.

This restriction will be corrected by upgrading the debugger and device file.

No. 8 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”.

- 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. 9 Restriction on reset input during a break

[Description]

The QB-V850ESFX3 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.

Please regard this item as a permanent restriction.

No. 10 Bug that occurs upon entering and releasing STOP mode when RESET pin is masked

[Description]

When the RESET pin is masked using the pin mask function of the debugger and watchdog timer 2 is used in reset mode, the CPU’s operating clock is switched to the internal oscillation clock after STOP mode is released, depending on the timing for entering and releasing STOP mode (one of (1) to (3) in the following table). After the clock is switched to the internal oscillation clock, the CPU continues the operation with the internal oscillation clock until the CPU reset button on the debugger is pressed.

No.	Operating Clock for Watchdog Timer 2	Timing at Which This Bug Occurs
(1)	Main clock	STOP mode is entered during the period from when a reset of watchdog timer 2 occurs until the reset is released[Note]
(2)	Internal oscillation clock	STOP mode is entered during the period from when a reset of watchdog timer 2 occurs until the reset is released[Note]
(3)		The internal oscillation clock is stopped during the period from when a reset of watchdog timer 2 occurs until the reset is released[Note], and then STOP mode is entered

Note The period in which watchdog timer 2 generates a reset signal while the reset signal of watchdog timer 2 is masked as a result of masking RESET using the pin mask function of the debugger.

[Workaround]

Do not use watchdog timer 2.

To generate a reset of watchdog timer 2, do not mask the RESET pin using the pin mask function of the debugger.

Please regard this item as a permanent restriction.

No. 11 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.

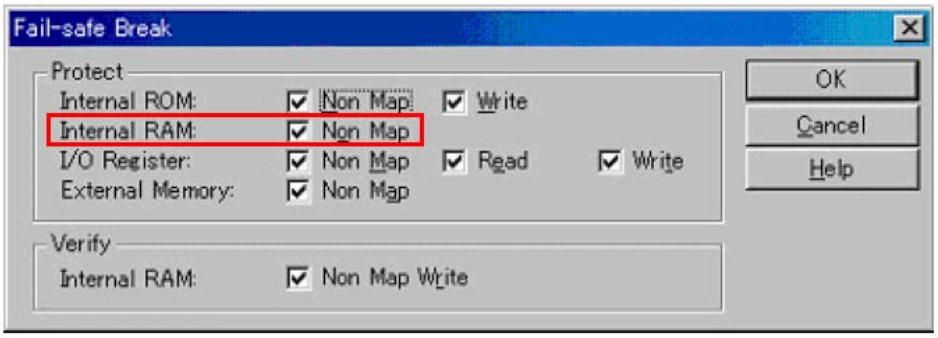
[Workaround]

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 box for “Internal RAM”.



  - 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.

No. 12 Bug in LOCK bit value of LOCKR register

[Description]

The LOCK bit may not be cleared after the PLLON bit is set to 1 even if the PLL lockup time timer overflows.

[Workaround]

There is no workaround.

This item has been corrected in control code B and later.

No. 13 Bug in accessing CBnRX register during break (n = 0 to 2)

[Description]

When the CSIBn receive data register (CBnRX) is read, the next reception operation starts in the normal operation. Under the following conditions (a) and (b), however, the next reception operation is not started even if CBnRX is read. As a result, communication stops or the DMA controller stops.

(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].

Note Including breaks by the RAM monitor function, DMM function, step-wise execution, fail-safe break, or breaks due to event change while the program is running. The real-time RAM monitor function does not cause this bug because it does not set breaks.

[Workaround]

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

(b) There is no workaround.

Please regard items (a) and (b) as permanent restrictions.

No. 14 Bug in subsystem clock oscillator

[Description]

The frequency is not halved when an externally connected RC resonator is used as the subsystem clock source.

[Workaround]

There is no workaround.

This bug has been corrected in control code C and later.