To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: “Standard”, “High Quality”, and “Specific”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as “Specific” without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as “Specific” or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is “Standard” unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   “Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   “High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   “Specific”: Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

RENESAS

Customer Notification

IE-703288-G1-EM1™

In-Circuit-Emulator

Operating Precautions

Target Device

V850ES/SG2
V850ES/SJ2

Global Document No. U18078EE5V0IF00 (5th edition)
Document No. TPS-HE-B-2844
Date Published September 2005

DISCLAIMER

The related documents in this customer notification may include preliminary versions. However, preliminary versions may not have been marked as such.

The information in this customer notification is current as of its date of publication. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC's data sheets or data books, etc., for the most up-to-date specifications of NEC PRODUCT(S). Not all PRODUCT(S) and/or types are available in every country. Please check with an NEC sales representative for availability and additional information.

No part of this customer notification may be copied or reproduced in any form or by any means without prior written consent of NEC. NEC assumes no responsibility for any errors that may appear in this customer notification. NEC does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC PRODUCT(S) listed in this customer notification or any other liability arising from the use of such PRODUCT(S).

No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC or others. Descriptions of circuits, software and other related information in this customer notification are provided for illustrative purposes of PRODUCT(S) operation and/or application examples only. The incorporation of these circuits, software and information in the design of customer's equipment shall be done under the full responsibility of customer. NEC assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.

While wherever feasible, NEC endeavors to enhance the quality, reliability and safe operation of PRODUCT(S) the customer agree and acknowledge that the possibility of defects and/or erroneous thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects and/or errors in PRODUCT(S) the customer must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.

The customer agrees to indemnify NEC against and hold NEC harmless from any and all consequences of any and all claims, suits, actions or demands asserted against NEC made by a third party for damages caused by one or more of the items listed in the enclosed table of content of this customer notification for PRODUCT(S) supplied after the date of publication.

Applicable Law:

The law of the Federal Republic of Germany applies to all information provided by NEC to the Customer under this Operating Precaution document without the possibility of recourse to the Conflicts Law or the law of 5th July 1989 relating to the UN Convention on Contracts for the International Sale of Goods (the Vienna CISG agreement).

Düsseldorf is the court of jurisdiction for all legal disputes arising directly or indirectly from this information. NEC is also entitled to make a claim against the Customer at his general court of jurisdiction.

If the supplied goods/information are subject to German, European and/or North American export controls, the Customer shall comply with the relevant export control regulations in the event that the goods are exported and/or re-exported. If deliveries are exported without payment of duty at the request of the Customer, the Customer accepts liability for any subsequent customs administration claims with respect to NEC.

Notes: 1. "NEC" as used in this statement means NEC Corporation and also includes its direct or indirect owned or controlled subsidiaries.

2. "PRODUCT(S)" means 'NEC semiconductor products' (NEC semiconductor products means any semiconductor product developed or manufactured by or for NEC) and/or 'TOOLS' (TOOLS means 'hardware and/or software development tools' for NEC semiconductor products' developed, manufactured and supplied by 'NEC' and/or 'hardware and/or software development tools' supplied by NEC but developed and/or manufactured by independent 3rd Party vendors worldwide as their own product or on contract from NEC)

(A) Table of Operating Precautions 5

(B) Description of Operating Precautions. 7

(C) Valid Specification 24

(D) Revision History 25

(A) Table of Operating Precautions

No.	Outline		IE-703288-G1-EM1			
		Rev.	1.02	1.03	1.04	1.16
		Control-Code[Note]	C	D	E	F
1	ROM correction function cannot be emulated (Direction of use)		✗	✗	✗	✗
2	Use-prohibited area (Direction of use)		✗	✗	✗	✗
3	Emulator memory settings (Direction of use)		✗	✗	✗	✗
4	Accuracy of ADC and DAC (Technical limitation)		✗	✓	✓	✓
5	Watchdog timer during break (Direction of use)		✗	✗	✗	✗
6	Timer M during break (Direction of use)		✗	✗	✗	✗
7	Timer M compare interrupt (Specification change notice)		✗	✗	✗	✗
8	Access of UAnRX register during break (Specification change notice)		✗	✗	✗	✗
9	Access of CBnRX register during break (Specification change notice)		✗	✗	✗	✗
10	Access of C0RGPT register during break (Specification change notice)		✗	✗	✗	✗
11	Access of C0TGPT register during break (Specification change notice)		✗	✗	✗	✗
12	Access of C0GNCTRL register during break (Specification change notice)		✗	✗	✗	✗
13	SLD instruction precaution (Specification change notice)		✗	✗	✗	✗
14	aFCAN transmission / reception (Technical limitation)		✗	✗	✓	✓
15	TMPn / TMQn external event counter function (Direction of use)		✗	✗	✗	✗
16	TMPn / TMQn capture operation (Direction of use)		✗	✗	✗	✗

No.	Outline		IE-703288-G1-EM1			
		Rev.	1.02	1.03	1.04	1.16
		Control-Code[Note]	C	D	E	F
17	aFCAN: Rx limitation (Technical limitation)		✗	✗	✗	✓
18	RESF register after WDTM2 reset (Technical limitation)		✗	✗	✗	✓
19	Operating frequency (Direction of use)		✗	✗	✗	✓

✓ Not applicable

✗ Applicable

Note: The Control Code is indicated by the letter appearing at the 2nd position from the left in the serial number of the product.

(B) Description of Operating Precautions

No. 1	ROM correction function cannot be emulated (Direction of use)
	Details The ROM correction function cannot be emulated. *Workaround* There is no workaround.

No. 2	Restriction on use-prohibited area (Direction of use)

Details

A fail-safe break is not generated if program execution or an access to a use prohibited memory area of the device is attempted.

Workaround

A break can be generated when the program is executed or a memory access occurs by setting a break in the debugger under the following conditions:

Detailed description of restriction on use-prohibited area				
Device:	D703260(Y), D703270(Y), D703280(Y)	D70(F)3261(Y), D70(F)3271(Y), D70(F)3281(Y), D70(F)3264(Y), D70(F)3274(Y), D70(F)3284(Y)	D703262(Y), D703272(Y), D703282(Y), D703265(Y), D703275(Y), D703285(Y), D703287(Y)	D70(F)3263(Y), D70(F)3273(Y), D70(F)3283(Y), D70(F)3266(Y), D70(F)3276(Y), D70(F)3286(Y), D70(F)3288(Y)
Use-prohibited area in which a fail safe break does not occur:	(1): 0x3FF8000 to 0x3FF8FFF	(2): 0x60000 to 0x7FFFF (3): 0x3FF0000 to 0x3FF6FFF	(4):0x3FF0000 to 0x3FF4FFF	(5):0xA0000 to 0xFFFFF (6):0x3FF0000 to 0x3FF2FFF
Execution access break (used to replace fail save break):	• Event: Execution • Address: Area (1) (2 execution events used for above conditions)	• Event: Execution • Address:Area (2) • Event: Execution • Address:Area (3) (4 execution events used for above conditions)	• Event: Execution • Address: Area (4) (2 execution events used for above conditions)	• Event: Execution • Address:Area (5) • Event: Execution • Address:Area (6) (4 execution events used for above conditions)
R/W access break (used to replace fail save break):	• Event: R/W • Access size: No Condition • Address: Area (1) (2 execution events used for above conditions)	• Event: R/W • Access size: No Condition • Address: Area (3) (2 access events used for above conditions)	• Event: R/W • Access size: No Condition • Address: Area (4) (2 access events used for above conditions)	• Event: R/W • Access size: No Condition • Address: Area (6) (2 access events used for above conditions)

No. 3	Emulator memory settings (Direction of use)

Details

When the debugger connects to the emulator the memory settings of the devicefile are not automatically set for several devices. This applies for devices with internal memory sizes that do not match to the below list.
The size of internal memory of the emulator can only be set to the following values:
Internal ROM:
32 KBytes, 64 KBytes, 128 KBytes, 256 KBytes, 512 KBytes or 1 MByte.
Internal RAM:
4 KBytes, 12 KBytes, 28 KBytes or 60 KBytes.

Greenhills Multi:
For devices which have different memory sizes than listed above, the size of internal ROM or RAM memory is set to the next smaller size of the above list.
E. g.: for devices that contain 384/32 KBytes of internal ROM/RAM memory, 256/28 KBytes is set (512/28 KBytes is set for devices containing 640/48 KBytes of internal ROM/RAM).

IAR Embedded workbench:
When connecting the debugger to the emulator an error message is shown if the ROM size selected in the devicefile does not fit the possible settings of the emulator (see list above).
The RAM size is automatically set to the next larger acceptable option.

Workaround

Change the settings when the connection between debugger and emulator is established.

Greenhills Multi:
Set the desired size of internal ROM/RAM memory in the .rc file which is executed when the debugger connects to the emulator using the target command “CPU [R= | A=]”.
Set the internal memory to the next larger size of the above list (e. g. select 512 KBytes for a devices that contains 384 KBytes of internal ROM).
The actual settings for internal memory can be checked in the target window using the “CPU” command.

IAR Embedded workbench:
Set the internal memory size in the configuration window for “Hardware Settings” which opens automatically when connecting to the emulator for the first time. The “Hardware Settings“ window can also be invoked under the “Emulator | Hardware Setup” drop down menu to alter or check the actual settings. Set the internal memory to the next larger size of the above list (e. g. select 512 KBytes for a devices that contains 384 KBytes of internal ROM).

Take care that the memory borders of the actual device are not exceeded since the acutal memory size provided by the emulator may be larger than the memory size of the emulated device.

No. 4	Accuracy of A/D converter and D/A converter (Technical limitation)

Details

The accuracy of the A/D converter does not meet the specification.

Workaround

There is no workaround. The accuracy has been improved with emulation boards of control code D or later (the conventional error is approx. 8%).

No. 5	Watchdog timer during break (Direction of use)

Details

When both of the following conditions (a) and (b) are fulfilled simultanously and a break occurs, the watchdog timer does not stop and will cause a reset or non maskable interrupt. If a reset occurs, the debugger hangs up.

Conditions that need to be fulfilled so that the above behaviour occurs:
(a) The main clock or subclock is selected as the clock source of the watchdog timer and
(b) The ring oscillator is stopped (RSTOP flag = 1).

Workaround

As a workaround to prevent the above behaviour do not stop the ring oscillator clock.

No. 6	Timer M during break (Direction of use)

Details

When a break occurs while the following conditions (a) and (b) are both fulfilled, timer M does not stop even if the peripheral break function has been set to 'break'.

(a) INTWT, Ring oscillator clock (fR/8) or subclock is selected as the clock source for timer M.
(b) The main clock is stopped by setting the MCK flag.

(Note: The peripheral break function is not supported by the debugger ID850 V2.51.)

Workaround

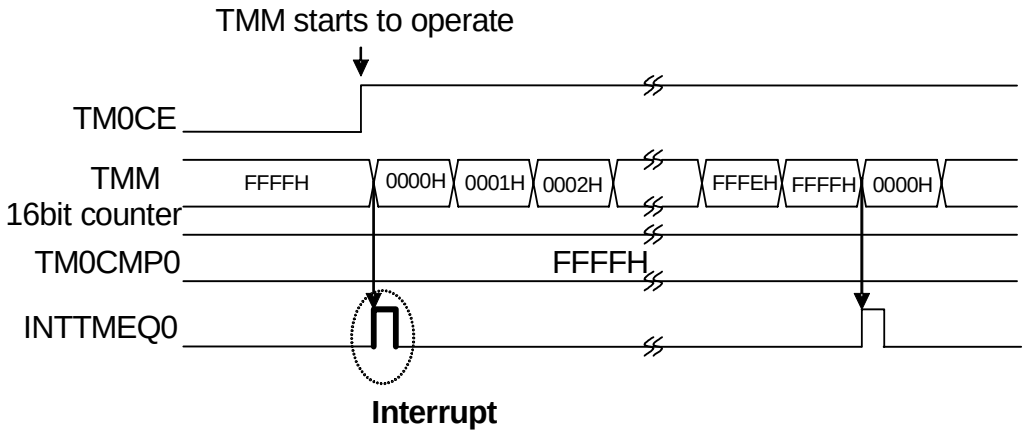
Implement one of the below workarounds to stop timer M during a break using the peripheral break function:

(a) Use the main clock (fXX, fXX/2, fXX/4, fXX/64, fXX/512) as the source clock for timer M.
(b) Do not stop the main clock oscillation.

No. 7	Timer M compare interrupt (Specification change notice)

Details

An unexpected interrupt occurs after activation of timer M when the compare register TM0CMP0 contains the value 0xFFFF.



Workaround

Do not set TM0CMP0 to 0xFFFF.

No. 8	Access of UAnRX register during break (Specification change notice)

Details

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 for the 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 the 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 the next time.

Note: Including breaks by the RAM monitor function or DMM function. However the realtime RAM monitor function does not cause this behaviour since it does not set breaks.

Remark: An overrun error also occurs when the UART receives data multiple times during a break (This complies with the specification of the emulator).

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.

No. 9	Access of CBnRX register during break (Specification change notice)

Details

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 register (CBnRX).
(b) If a DMA transfer from the CSIBn receive data register (CBnRX) is performed during a break**NOTE**. As a result the communication stops or the DMA controller stops.

Note: Including breaks by the RAM monitor function or DMM function. However the realtime RAM monitor function does not cause this behaviour since 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.

No. 10	Access of C0RGPT register during break (Specification change notice)

Details

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 again.

(a) If a software break occurs immediately after reading the CAN0 module receive history list register (C0RGPT)
(b) If a DMA transfer from the CAN0 module receive history list register (C0RGPT) is performed during a break**NOTE**.

Note: Including breaks by the RAM monitor function or DMM function. However the realtime RAM monitor function does not cause this behaviour since it does not set breaks.

Workaround

(a) Set a hardware break when setting a break immediately after reading the C0RGPT register.
(b) There is no workaround.

No. 11	Access of C0TGPT register during break (Specification change notice)

Details

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 again.

(a) If a software break occurs immediately after reading the CAN0 module transmit history list register (C0TGPT).
(b) If a DMA transfer from the CAN0 module transmit history list register (C0TGPT) is performed during a break**NOTE**.

Note: Including breaks by the RAM monitor function or DMM function. However the realtime RAM monitor function does not cause this behaviour since it does not set breaks.

Workaround

(a) Set a hardware break when setting a break immediately after reading the C0TGPT register.
(b) There is no workaround.

No. 12	Access of C0GNCTRL register during a break (Specification change notice)

Details

When a register access is performed in the following sequence, an unexpected forcible shutdown may occur after the sequence is complete.

Sequence :
(1) The EFSD bit of the CAN0 module control register (C0GMCTRL) is set.
(2) The I/O register**NOTE** is accessed.
(3) The GOM bit of the CAN0 mode control register (C0GMCTRL) is cleared.

Note: I/O register access except for clearing the GOM bit of the C0GMCTRL register

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

No. 13	SLD instruction precaution (Specification change notice) (Specification change notice)

Details

If a conflict occurs between the decode operation of the instruction (<2> in the examples mentioned below) immediately before the **sld** instruction (<3> in the examples) following a special instruction (<1> in the examples) and an interrupt request before execution of the special instruction is complete, the execution result of the special instruction may not be stored in a register as expected.
This situation may only occur when the same register is used as the destination register of the special instruction and the **sld** instruction, and when the register value is referenced by the instruction followed by the **sld** instruction.

Conditions under which the conflict occurs:

The situation may occur when all the following conditions (1) to (3) are satisfied.

(1) Either condition (I) or (II) is satisfied

Condition (I):
The same register is used as the destination register of a special instruction (see below) and the subsequent **sld** instruction and as the source register (reg1) of an instruction shown below followed by the **sld** instruction (See Example 1).

mov **reg1**,reg2	not **reg1**,reg2	satsubr **reg1**,reg2	satsub **reg1**,reg2
satadd **reg1**,reg2	or **reg1**,reg2	xor **reg1**,reg2	and **reg1**,reg2
tst **reg1**,reg2	subr **reg1**,reg2	sub **reg1**,reg2	add **reg1**,reg2
cmp **reg1**,reg2	mulh **reg1**,reg2		

Condition (II):
The same register is used as the destination register of a special instruction (see below) and the subsequent **sld** instruction and as the source register (reg2) of an instruction shown below followed by the **sld** instruction (See Examples 2 and 3).

not reg1,**reg2**	satsubr reg1,**reg2**	satsub reg1,**reg2**	satadd reg1,**reg2**
satadd imm5,**reg2**	or reg1,**reg2**	xor reg1,**reg2**	and reg1,**reg2**
tst reg1,**reg2**	subr reg1,**reg2**	sub reg1,**reg2**	add reg1,**reg2**
add imm5,**reg2**	cmp reg1,**reg2**	cmp imm5,**reg2**	shr imm5,**reg2**
sar imm5,**reg2**	shl imm5,**reg2**		

Special instruction:

- **ld** instruction: ld.b, ld.h, ld.w, ld.bu, ld.hu
- **sld** instruction: sld.b, sld.h, sld.w, sld.bu, sld.hu
- Multiply instruction: mul, mulh, mulhi, mulu

(2) When the execution result of the special instruction (see above) has not been stored in the destination register before execution of the instruction (instruction of condition (I) or (II)) immediately before the **sld** instruction starts in the CPU pipeline.

No. 13	SLD instruction precaution (Specification change notice) (Specification change notice)

(cont.)

(3) When the decode operation of the instruction (instruction of condition (I) or (II)) immediately before the **sld** instruction and interrupt request servicing conflict.

Examples of instruction sequences that may cause the conflict:

Example 1:

```
<1>  ld.w   [r11], r10
                ⋮
<2>  mov    r10, r28
<3>  sld.w  0x28, r10
```

This situation occurs when the decode operation of the **mov** instruction (<2>) immediately before the **sld** instruction (<3>) and interrupt request servicing conflict before the execution of the special instruction **ld** (<1>) is complete.

Example 2:

```
<1>  ld.w   [r11], r10
                ⋮
<2>  cmp    imm5, r10
<3>  sld.w  0x28, r10
<4>  bz label
```

This situation occurs when the decode operation of **cmp** (<2>) immediately before the **sld** instruction (<3>) and interrupt request servicing conflict before execution of the special instruction **ld** (<1>) is complete. As a result, the compare result of the **cmp** instruction becomes undefined, which may cause an unexpected operation of the branch instruction **bz** (<4>).

Example 3:

```
<1>  ld.w   [r11], r10
                ⋮
<2>  add    imm5, r10
<3>  sld.w  0x28, r10
<4>  setf   c, r16
```

This situation occurs when the decode operation of the **add** instruction (<2>) immediately before the **sld** instruction (<3>) and interrupt request servicing conflict before execution of the special instruction **ld** (<1>) is complete. As a result, the result of the **add** instruction and the depending status flags become undefined, which may cause an unexpected operation of the **setf** instruction (<4>).

Workaround

(1) Do not use the **sld** instruction (e. g. by avoiding code optimization that makes use of **sld**).

(2) If a code sequence as described above is used (a **sld** instruction following an instruction that can be executed in parallel), insert a **nop** instruction before the **sld** instruction.

(3) If a code sequence as described above is used (a **sld** instruction following an instruction that can be executed in parallel), exchange the order of the previous two instructions as long as the program algorithm is not disturbed:

No. 13	SLD instruction precaution (Specification change notice) (Specification change notice)

(cont.)

Example:
1. (before implementing workaround)
ld.w [r11], r10
...
add r11, r12
mov r10, r28
sld.w 0x28, r10

2. (after implementing workaround)
ld.w [r11], r10
...
mov r10, r28
add r11, r12
sld.w 0x28, r10

(4) When assembler code is used:
Avoid the crictical code sequences as described above.

No. 14	aFCAN transmission / reception (Technical limitation)

Details

The aFCAN macro will under certain timing conditions accompanied by a particular configuration of the message buffers and a specific operational usage not operate as expected. Different behaviors have to be considered. The description of the particular unexpected behavior is organized by the buffer Type: TX and RX, the frame format (extended or standard identifier), and the buffer number.

TX-Buffer Behavior

This section describes all unexpected behaviors linked to the configuration of transmit buffers. Configurations not listed are of no concern and can be used without restrictions.

- Buffer #0 or Buffer #0 and #1 are configured as TX-buffer

When using message buffer #0 as a TX-buffer, the message requested for transmission of this buffer may not be sent at the next possible timing. This behavior is caused when the internal scan for new transmission requests reached buffer #0 and at the same time the transmission of a previously sent message ends. Instead of sending the message object from buffer #0, the aFCAN attempts to send the contents of message buffer #1. In case the TRQ of buffer #1 is set, the message from buffer #1 is sent followed by the transmission of the message in buffer #0. This resembles an inner priority inversion. In case the TRQ of buffer #1 is not set or buffer #1 is not a TX-buffer, the contents of buffer #0 are sent whenever any of the other TRQ-bits in the aFCAN are set or cleared, or when a receive operation is started; i.e. when the next bus activity occurs. This behavior is valid for both frame types, extended or standard identifier format.

- Buffer #1 - #31 are configured as TX-buffer

There are two configurations that lead to the same, unexpected behavior. In the first configuration Buffer #1 through #31 are set up as normal TX-buffers, and in the second configuration buffer #0 through #7 are operated in ABT-mode (automatic block transmission) and the remaining buffers (#8 - 31) are configured as normal TX-buffers. The unexpected behavior occurs as well if only a subset of buffers are configured as normal TX-buffers. When using any message buffer #n in the range of buffer #1 through #31 with extended identifier, an inner priority inversion can be encountered. In that case the message from buffer #n+1 is sent in advance of the message in buffer #n even though the priority of the identifier in buffer #n is higher. This behavior is seen when the internal transmit search algorithms of the aFCAN processing buffer #n meets the start of a transmission, the end of a frame on the bus (RX or TX) that needs to end with an error frame, or the event of transmission request by the CPU or by the ABTmode. In case the ABT-mode is active, the unexpected behavior does only apply for messages in buffer #8 through #31. This behavior applies only if extended identifiers are in use. Applications exclusively using standard Identifiers do not suffer any limitation of this kind.

No. 14	aFCAN transmission / reception (contd.)

RX-Buffer Behavior

There are several configurations for the receive buffers that can cause different unexpected behaviors.

- Buffer #0 is configured as RX-buffer with EXT ID handling

There are two kinds of unexpected behavior in this configuration. The first one requires that a newly received message would normally be received in buffer #0. When in this case a transmission request (TRQ) by the host processor is submitted at the time where buffer #0 is scanned by the internal RX-Search algorithm for newly received messages, the storage of the message for buffer #0 will not occur. The message will be stored nowhere.
In a second scenario, the newly received message would under normal conditions not be stored in buffer #0. When in this case a TRQ for buffer #i by the host processor is submitted, which again happens at the same time where the acceptance filtering for the newly received message is active, the message can be stored in buffer #0. The unexpected storage of the message in buffer #0 does only occur when the identifier bits ID15-0 of the received message coincidentally match the values in MCONF/MDLC of the TX-buffer #i. The behavior applies only for extended format frames. Standard format frames can be used without suffering this limitation.

- Buffer #1 - #31 are configured as RX-buffer with EXT ID handling

When a newly received message does not match the acceptance filter criteria for buffer #n (n = 1...31), the following timing can lead to an unexpected behavior. When additionally to that condition a transmission request is set for buffer #i at the time where buffer #n is scanned by the internal RX-Search algorithm, the storage of the message can be performed in buffer #n+1 although this buffer does not match with the acceptance filter criteria for the received message. The unexpected storage of the message in buffer #n+1 does only occur when the identifier bits ID15-0 of the received message coincidentally match the values in MCONF/MDLC of the Txbuffer #i. This unexpected behavior is also possible if only a subset of buffers in the range of #1 through #31 are configured as RX-buffers. The behavior applies only for extended format frames. Standard format frames can be used without suffering this limitation.

Workaround

The application needs to abstain from the usage of message buffer #0 or configure buffer #0 as a receive buffer with standard identifier handling.
The ABT-mode is not affected from this limitation. Thus buffer #0 can be configured as a transmit buffer in that mode.
Do not use extended identifiers for transmit and receive objects. If the application needs to process extended identifiers, the host processor must not issue a transmission request anytime the CAN-bus is busy. For this case the application can monitor the bus status and issue a transmission request right after the bus idle state was detected. Then the TRQ will be submitted in an uncritical point of time. This is even valid when the TRQ is submitted during the first 19 bits of the extended identifier.

No. 15	TMPn / TMQn external event counter function (Direction of use)

Details

When the external event counter mode is used and the compare register of timer TMP or TMQ is set to 0x0000 an interrupt occurs after the overflow of the timer.

Workaround

Avoid setting 0x0000 to the timer compare registers of TMP or TMQ.

No. 16	TMPn / TMQn capture operation (Direction of use)

Details

When the pulse width measurement mode or the free-running mode are used the initial value of the capture register is 0xFFFF when the count clock of the TMPn / TMQn is lower than the sampling clock of the capture trigger input after the timer is enabled (TPnCE = 1, TQnCE = 1).

Workaround

There is no workaround.

No. 17	aFCAN: Rx limitation (Technical limitation)

Details

RX Limitation

The aFCAN macro may store an incoming message although this message was interrupted by a bus error frame. Thus, the incomplete reception causes that a message buffer is updated with old or incorrect data or that the message is even stored at an incorrect location.

This unexpected behaviour affords that the bus error occurs in a certain relation to the currently present message on the bus. The critical time window starts at the sample point of the LSB of the DLC-field and lasts for the duration of an internal process in the aFCAN macro (RX-search). This time window usually lasts for a few bit times only. The actual length depends on the clock supply for the AFCAN, the CPU accesses during this period, the baud rate and the number of message buffers of the particular AFCAN macro.
In this time window the RX-search evaluates the received identifier of the current message. When the bus error is detected within this window and when the RX-search has just scanned buffer #n for reception and found it is matching, the message will unexpectedly be treated as a received message. As the time window is limited as described above, only a stuff bit error occurring right in this window can cause this behaviour.
There are two types of unexpected behavior for the RX limitation depending on the presence of pending transmission request (TRQ) for any other message buffer.

1. Behaviour at pending TRQ (TRQi = 1)

When the host processor has already submitted a transmit request (TRQ) for at least one buffer, the unexpected reception of the message will take place into the message buffer found by internal RX-search. This is the correct location to store the message i.e. the acceptance filter criteria are correctly fulfilled. However the data part will be updated with the contents of the shift register of the CAN protocol core. As this register is immediately stopped at detection of the bus error, the data provided to the message buffer can not be interpreted by the host processor.

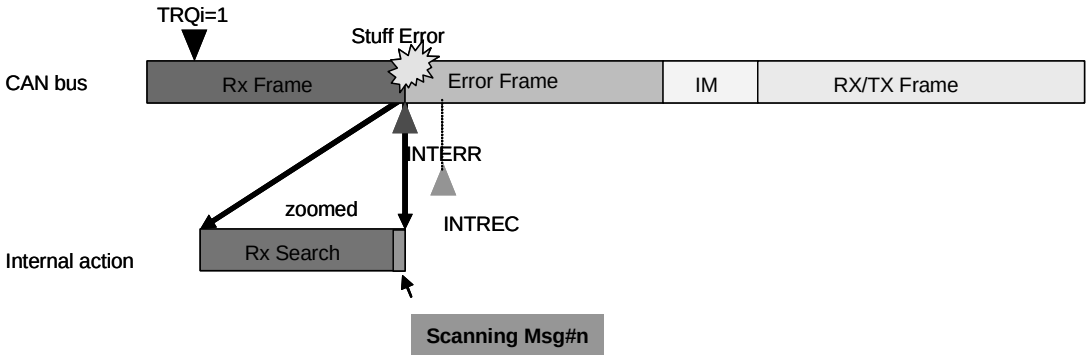


Figure 1: Behavior at pending TRQ

As during a regular reception, the RX-interrupt (if enabled) is generated and the application processes the message object.

No. 17	aFCAN: Rx limitation (Technical limitation)

2. Behavior without pending TRQ (TRQi = 0)
In case the host processor has not submitted a transmit request (TRQ) for any buffer before the detection of the bus error but submits TRQ = 1 after that point in time (see figure below) before the re-transmission of the message interrupted by the stuff bit error started, the unexpected reception of the message will take place.

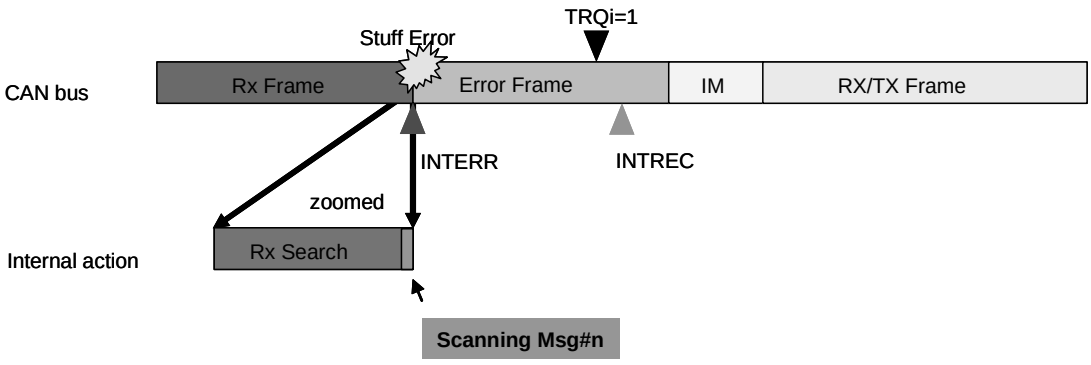


Figure 2: Behavior without pending TRQ

The unexpected storage of the message is issued in the particular message buffer that matches the acceptance filter criteria at the time where the bus error is detected (as described in 1.) or the message buffer #0 is overwritten independently of its configuration.

Impact on application
In typical applications the RX-limitation will lead to transiently incorrect data. In the vast majority of cases the message interrupted by a bus error is repeated by the transmitter right away. Then the application receives correct data shortly after the unexpected reception.
In scenarios where the message buffer #0 is overwritten, the impact for the application depends on the usage of that buffer. If it is configured as a receive buffer, the application receives a message at an unexpected location and will interpret the data to belong to the identifier originally programmed for that buffer. The message buffer #0 needs to be re-configured in order to receive the originally intended message object again.
In case of a transmit message buffer the unexpected storage may falsify a transmit object; i.e. when the unexpected behavior occurs after preparation of the message data but before the actual start of transmission. This scenario is even less likely than the scenario described in 1, which itself has a low probability. However the transmission of a falsified message can lead to repetitive transmission attempts when the original provider of that message (identifier) tries to send its message at the same time. Then the messages most likely will differ in their data part and a bit error is detected. This repetition resumes until one of the nodes enters error passive or bus off state. Then the situation is resolved as all pending TRQ are send with delay or are cancelled (in case of bus off state).

No. 17	aFCAN: Rx limitation (Technical limitation)

Workaround

NEC will update the affected products.
NEC does not recommend a S/W workaround as first choice as it is fairly complex. On the one hand it is based on the control of submitting transmission requests only when the bus is idle. On the other hand a less complex algorithm can be used which does not prevent the unexpected reception but detects it safely and discards the unexpected reception in the CAN S/W driver. Any of these algorithms require that message buffer #0 is not used or that a 'dummy' TRQ in an unused buffer is set. This prevents behaviors as described in 2.

No. 18	RESF register after WDTM2 reset (Technical limitation)

Details

The WDT2RF bit of the RESF register is not set to 1 when a reset by the watchdog timer occurs.

Workaround

There is no workaround.

No. 19	Operating frequency (Direction of use)

Details

The maximum operating frequency of IE-703288-G1-EM1 is 20 MHz for emulators up to version “E”. From version “F” onwards the maximum operating frequency is 32 MHz. For operation at frequencies over 20 MHz the emulator IE-V850ES-G1 version “F” or higher must be used.

Workaround

There is no workaround.

(C) Valid Specification

Item	Date pulished	Document No.	Document Title
1	March, 2003	SUD-DT-02-0101-2-E	IE-703288-G1-EM1 (Preliminary User's Manual)
2	July, 2005	U16541EJ4V0UD00	V850ES/SG2 Hardware (User's Manual)
3	June, 2005	U16603EJ4V0UD00	V850ES/SJ2 Hardware (User's Manual)
4	May, 2005	U17340EJ1V0UD00	V850ES/SJ2H Hardware (Preliminary User's Manual)
5	August, 2005	U17500EJ1V0UD00	V850ES/SJ3 Hardware (Preliminary User's Manual)
6	April, 2004	U15943EJ3V0UM00	V850ES Architecture (User's Manual)

(D) Revision History

Item	Date pulished	Document No.	Comment
1	April 28, 2003	TPS-HE-B-2840	First release
2	August 11, 2003	TPS-HE-B-2841	Added item 4, Control Code ’D’
3	April 7, 2003	TPS-HE-B-2842	Added items 4 to 14, Control Code ’E’
4	August, 2004	TPS-HE-B-2843	Added items 15 to 17
5	September, 2005	TPS-HE-B-2844	Added items 18, 19, Control Code ’F’