Outline

When using the E20 emulator, E1 emulator, E2 emulator, or E2 emulator Lite (hereafter E2 Lite), note the following points.

1. When using the large trace capacity or coverage function of the E20 emulator for the RX family
2. When using the large trace capacity or coverage function of the E20 emulator and the internal trace function of E20, E1, E2, and E2 Lite for the RX family

1. When Using the Large Trace Capacity or Coverage Function of the E20 Emulator for the RX Family

1.1 Applicable Product
➢ E20 emulator

1.2 Applicable MCUs
➢ RX family: RX64M, RX65N, RX651, and RX71M groups

1.3 Details

When the large trace capacity or coverage function of the E20 emulator is used for the RX family, a user program runs out of control under certain conditions. Note that this phenomenon occurs only during debugging with the emulator, and does not occur during operation of the chip itself.

1.4 Conditions

If both conditions (1) and (2) are met, this problem may occur depending on the condition of the trace buffer embedded in MCU.

(1) Either of the following functions is used.

   (1-1) Large trace capacity
       For e² studio: The trace output mode is set to "trace priority".
       For CS+: The external trace output is set to "trace output priority".

   (1-2) Coverage function
       For e² studio: The acquisition mode is set to "trace priority".
       For CS+: The coverage measurement method is set to "coverage measurement priority".
(2) "RTSD instruction or RTFI instruction" is placed immediately after the "BSR instruction or JSR instruction".

Example:

```
_func_A:
    :                  _func_B:
    BSR   _func_B    ; (1)         RTS
    RTSD  #20, R8-R9  ; (2)
```

The RTSD instruction (2) is placed immediately after the BSR instruction (1).

In C programs, when call for another function is coded at the end of a function (as shown in the following example), an instruction sequence (as shown in the preceding example) may be generated depending on the compiler type and settings.

Example:

```
void func_A(void) {
    :               void func_B(void) {
    func_B();          :               
}                        }
```

The specific instruction format of (2) in which this phenomenon occurs is as follows.

<table>
<thead>
<tr>
<th>Instruction format of (2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTSD src, dest-dest2</td>
</tr>
<tr>
<td>RTFI</td>
</tr>
</tbody>
</table>

When this phenomenon occurs, the branch destination for the execution of (2) becomes an address that is different from the normal address, and the program runs out of control.
1.5 Workaround

Do either of the following methods to change the settings so as not to satisfy the conditions of section 1.4 (1):

- When using the large trace capacity
  
  For e² studio: The trace output mode is set to "trace priority". --> Change the setting to "CPU execution priority".
  
  For CS+: The external trace output is set to "trace output priority". --> Change the setting to "CPU execution priority".

- When using the coverage function
  
  For e² studio: The acquisition mode is set to "trace priority". --> Change the setting to "CPU execution priority".
  
  For CS+: The coverage measurement method is set to "coverage measurement priority". --> Change the setting to "CPU execution priority".

1.6 Schedule for Fixing the Problem

This problem will be fixed in the next version of e² studio and CS+.

2. When Using the Large Trace Capacity or Coverage Function of the E20 Emulator and the Internal Trace Function of E20, E1, E2, and E2 Lite for the RX Family

2.1 Applicable Products

- E20 emulator, E1 emulator, E2 emulator, and E2 emulator Lite

2.2 Applicable MCUs

- RX family: MCU with RXv2 core and RXv3 core *

  * See below for details.
  

2.3 Details

When the large trace capacity or coverage function of the E20 emulator and the internal trace function of E20, E1, E2, and E2 Lite are used for the RX family, if specific conditions are satisfied, trace information or coverage information cannot be correctly acquired. Note that this phenomenon occurs only during debugging with the emulator, and does not occur during operation of the chip itself.
2.4 **Conditions**

When the trace function or coverage function is enabled, if the time of branch instruction execution to branch to a smaller address overlaps with the time of interrupt acceptance, this phenomenon occurs.

Example: When branching to address (1) that is smaller than (3) in the following branch instruction of (3) and at the same time, accepting an interrupt:

```
LOOP:                  ; (1)
    :                  
    SUB r3, r4
    BNE LABEL1        ; (2) (the first branch instruction after (1))
    :                  
    BNE LOOP          ; (3)
    :                  
LABEL1:                
```

When this phenomenon occurs, the operation will be as follows:

<table>
<thead>
<tr>
<th>Function</th>
<th>Operation when this phenomenon occurs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace acquisition</td>
<td>The trace information of (1) to (2) (the first branch instruction after (1)) which was not actually executed is displayed additionally before the trace information of the interrupt handler.</td>
</tr>
<tr>
<td>Coverage acquisition</td>
<td>It is displayed that all instructions from (1) to (3) have been executed (covered) regardless of the actual execution status.</td>
</tr>
</tbody>
</table>

For trace acquisition, if the same instruction sequence is output to the trace result before and after the interrupt processing execution, it can be determined that this phenomenon may have occurred.

For coverage acquisition, there is no way to determine whether this phenomenon has occurred.

2.5 **Workaround**

There is no workaround for this problem.

2.6 **Schedule for Fixing the Problem**

We do not plan to fix the problem.
Revision History

<table>
<thead>
<tr>
<th>Rev.</th>
<th>Date</th>
<th>Page</th>
<th>Description</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>Feb. 16, 2018</td>
<td>-</td>
<td>First edition issued</td>
<td></td>
</tr>
<tr>
<td>1.01</td>
<td>Mar. 25, 2021</td>
<td>3</td>
<td>Applicable MCUs</td>
<td></td>
</tr>
</tbody>
</table>

TOYOSU FORESIA, 3-2-24 Toyosu, Koto-ku, Tokyo 135-0061 Japan
Renesas Electronics Corporation

■Inquiry
https://www.renesas.com/contact/

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.
The past news contents have been based on information at the time of publication. Now changed or invalid information may be included.
The URLs in the Tool News also may be subject to change or become invalid without prior notice.

All trademarks and registered trademarks are the property of their respective owners.