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](http://www.renesas.com)

---

April 1st, 2010
Renesas Electronics Corporation

---

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

There are the restriction which is described in following document when the HI7700/4 is used on SH3-DSP or SH4AL-DSP. If the problem may occur, please apply the workaround.

Attached : HI7700 -4_LIMIT_050331_E
" A Note on Using the HI7700/4 (Mar.31.2005)"

We plan to fix or improve this problems in the next version (V.1.03 Release 02). The V.1.03 Release 02 will be released at the middle of April 2005.
A Note on Using the HI7700/4 (Mar.31.2005)

1. CPU exception might occur in the kernel when you uses SH3-DSP or SH4AL-DSP

1.1 Description

The kernel might execute DSP instructions in the non-DSP mode (the DSP bit of the SR register in the CPU is 0). In this case, the CPU generates exception.

1.2 Versions Concerned

V.1.03 Release 01 or before

1.3 Conditions

This problem occurs if the following conditions are all satisfied:

Condition = (1) && (2) && (3) && (4) && (5) && (6) && (7.1) || (7.2)

1. The CPU core of the microcomputer that you use is SH3-DSP or SH4AL-DSP.
2. Either of the following kernel libraries for DSP have been linked.
   - dsp_knl_big.lib, dsp_knl_little.lib, sh4al_dsp_knl_big.lib, sh4al_dsp_knl_little.lib
   - dspbty_big.lib, dspbty_little.lib, sh4al_dspbtby_big.lib, sh4al_dspbtby_little.lib
   - dsp_expand_big.lib, dsp_expand_little.lib, sh4al_dsp_expand_big.lib, sh4al_dsp_expand_little.lib
3. The [CFG_DSP] is selected in the configurator.
4. The task-exception processing function (def_tex service call) is selected in the configurator.
5. The TA_COP0 attribute is used.
6. The iras_tex service call is used in the non-task context, such as an interrupt handler or time-event handler.
7. An interrupt handler or CPU-exception exits in the non-DSP mode (the DSP bit of the SR register in the CPU is 0).

Note, it is the following cases that these handlers become non-DSP mode.

a) The DSP bit of “SR at initiation”, which is specified at the handler definition, is 0. And The handler itself does not set DSP bit of SR register.

b) The DSP bit of “SR at initiation”, which is specified at the handler definition, is 1. And The handler itself clear DSP bit of SR register.

(7.1) The chg_ims or ichg_ims service call is called with specifying SR_IMS00(=0) for the parameter in the non-DSP mode (the DSP bit of the SR register in the CPU is 0).

Note, in the specification of the HI7700/4, only task and task-exception processing routine are permitted to specify SR_IMS00. It is the following cases that task and task-exception processing routine become non-DSP mode.

a) Task or task-exception processing routine itself clear DSP bit of SR register.

1.4 Workaround

- Workaround for (7.1)
  - Specify the DSP bit of “SR at initiation” to 1. And the handler itself must not clear DSP bit of SR register.
- Workaround for (7.2)
  - Task or task-exception processing routine must not clear DSP bit of SR register.

1.5 Schedule of Fixing the Problem

We plan to fix this problem in V.1.03 Release 02.
2. Initializing DSR register for DSP operation

2.1 Description

As described to Section 4 of the user's manual, the initial value of the DSR register of various programs (such as task, interrupt handler, etc.) is undefined. The DSR must be initialized before DSP operation by application program. For details, please refer to the hardware manual of the microcomputer that you use. While using a microcomputer that has the SH3 DSP as the CPU core, the DSR register must be initialized before performing a DSP operation. Otherwise, the program may not work as expected. Usually, it recommends initializing to 0.

The initialization of DSR register is required for the following each programs. It has to initialize DSR register only once before DSP operation.

- Each tasks
- Each task-exception processing routine
- Each interrupt handlers
- Each CPU exception handlers (including TRAPA)
- Each cyclic handlers
- Each alarm handlers
- Overrun handler
- Each initialization routine

The example of initializing DSR register for task is shown as follows.

```c
#include "itron.h"
#include "kernel.h"
#pragma inline_asm(SetDSR)
static void SetDSR(UW dsr)
{
    lds r4, dsr
}

task(VP_INT exinf)
{
    SetDSR(0); // Initialize DSR register

    // DSP operations
}
```

2.2 Versions Concerned

V.1.03 Release 01 or before
2.3 Schedule of Improvement

In V.1.03 Release 02, we will improve that the kernel initializes DSR register of a task and task-exception processing routine with TA_COP0 attribute to 0. Note, the initial value of other programs will be not changed.

The initial value of DSR register of each programs is shown as follows.

<table>
<thead>
<tr>
<th>Program</th>
<th>V.1.03 Release 01 or before</th>
<th>V.1.03 Release 02</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task without TA_COP0 attribute</td>
<td>Undefined</td>
<td>Undefined</td>
</tr>
<tr>
<td>Task with TA_COP0 attribute</td>
<td>Undefined</td>
<td>0</td>
</tr>
<tr>
<td>Task-exception processing routine without TA_COP0 attribute</td>
<td>Undefined</td>
<td>Undefined</td>
</tr>
<tr>
<td>Task-exception processing routine with TA_COP0 attribute</td>
<td>Undefined</td>
<td>0</td>
</tr>
<tr>
<td>Interrupt handler</td>
<td>Undefined</td>
<td>Undefined</td>
</tr>
<tr>
<td>CPU exception handler</td>
<td>Undefined</td>
<td>Undefined</td>
</tr>
<tr>
<td>Time-event handler</td>
<td>Undefined</td>
<td>Undefined</td>
</tr>
<tr>
<td>Initialization routine</td>
<td>Undefined</td>
<td>Undefined</td>
</tr>
</tbody>
</table>