RENESAS Tool News

RENESAS TOOL NEWS on June 1, 2008: 080601/tn2

Notes on Using the Real-Time OSes HI7000/4, HI7700/4, and HI7750/4, Which Are Used for the SuperH MCU Family

Please take note of the following problems in using the real-time OSes HI7000/4, HI7700/4, and HI7750/4, which are used for the SuperH MCU family:

1. With clearing an event flag waited for by two or more tasks
2. With issuing the irel_mpl service call with or without rel_mpl when CFG_NEWMPL selected

1. Problem with Clearing an Event Flag Waited for by Two or More Tasks

1.1 Products and Versions Concerned

(1) HI7000/4 V.2.02 Release 03 and earlier versions
(for the SH-1-, SH-2-, SH2-DSP-, SH-2A-, SH2A-FPU-Cored devices)
(2) HI7700/4 V.2.03 Release 01 and earlier versions
(for the SH-3-, SH3-DSP-, SH4AL-DSP-cored devices)
(3) HI7750/4 V.2.02 Release 03 and earlier versions
(for the SH-4- and SH-4A-cored devices)

1.2 Description

Any of the two or more tasks that are waiting for an event flag (consisting of 32 bits) to be set may not be released from their WAITING states even if the conditions for canceling these WAITING states are satisfied.
However, the unreleased tasks will exit from their WAITING states if other conditions for canceling them that are independent of the conditions described in Section 1.3 below are fulfilled.

1.3 Conditions

If the following conditions are all satisfied, the tasks that would be released from their WAITING states by the issuance of the set_flg or iset_flg service call in (3) below will not be done so in some cases:

(1) In the program exists an event flag having the TA_WMUL attribute (allows two or more tasks to enter the WAITING states). This flag is hereafter called F.

(2) Two or more tasks are waiting for F to be set to the value represented by the bit pattern that satisfies the conditions for canceling their WAITING states.

(3) The set_flg or iset_flg service call is issued to set F to the value represented by the bit pattern that satisfy the conditions for canceling the WAITING states of any of tasks in (2).

(4) While the kernel is handling set_flg or iset_flg in (3), an interrupt is requested.

(5) The interrupt in (4) invokes the interrupt handler or time-event handler, which performs any of the following processing:

a. Issues iset_flg to set F to the value represented by the bit pattern in (2). Here F has the TA_CLR attribute (clears all the F's bits to 0s if the WAITING states are canceled) as well as TA_WMUL.

b. Issues the ipol_flg service call that takes F as a parameter and then ends it properly. Here F has the TA_CLR attribute as well as TA_WMUL.

c. Issues the iclr_flg service call to clear the bits in F that are included in those set in (3) and satisfy the conditions for canceling the WAITING states of the tasks in (2).

1.4 Workaround

Before and after issuing set_flg or iset_flg in Condition (3), change the level of the interrupt mask to that of the kernel interrupt mask as follows:

(1) If set_flg issued

```
#include <machine.h>
int old_imask;
old_imask = get_imask();
chg_ims(15);  /* Interrupt mask level changed to kernel interrupt
                 mask level (15 in this example) */
iset_flg(...);   /* As context is interpreted as non-task one while
```

```
                interrupt is masked in product specifications,
                set_flg is changed to iset_flg */
  ichg_ims((IMASK)old_imask);  /* Interrupt mask level resumed */
```

(2) If iset_flg issued

```
  #include <machine.h>
  int old_imask;
  old_imask = get_imask();
  set_imask(15);  /* Interrupt mask level changed to kernel interrupt
              mask level (15 in this example) */
  iset_flg(...);
  set_imask(old_imask);  /* Interrupt mask level resumed */
```

2. Problem with Issuing the irel_mpl Service Call with or without rel_mpl

When CFG_NEWMPL Selected

2.1 Products and Versions Concerned

(1) HI7000/4 V.2.01 Release 00through V.2.02 Release 03
(for the SH-1-, SH-2-, SH2-DSP-, SH-2A-, SH2A-FPU-cored devices)

(2) HI7700/4 V.2.01 Release 00 through V.2.03 Release 01
(for the SH-3-, SH3-DSP-, SH4AL-DSP-cored devices)

(3) HI7750/4 V.2.01 Release 00 through V.2.02 Release 03
(for the SH-4- and SH-4A-cored devices)

2.2 Description

If the irel_mpl service call has been issued with or without rel_mpl, contradictions will arise in the kernel controlling data, and your system may not operate properly. Note, however, that this problem does not occur if only rel_mpl is issued with irel_mpl not used since this does not satisfy Condition (4) below.

2.3 Conditions

This problem may occur if the following conditions are all satisfied:

(1) In the the Modification of Variable-Size Memory Pool Information dialog box, the CFG_NEWMPL check box is checked when the GUI configurator used.

(2) While any tasks are waiting for the variable-size memory pool (hereafter called M) to offer the memory blocks they require, the application program issues the irel_mpl service call with or without rel_mpl.

(3) While the kernel is handling the service call in (2),

an interrupt is requested.

(4) The interrupt in (3) invokes the interrupt handler or time-event handler, which issues irel_mpl.

(5) The issuance of irel_mpl with or without irel_mpl in (2) and (4) makes the maximum size of the unoccupied continuous areas in M larger than the size of the memory block required by the task in front of the queue for memory blocks to be offered by M; that is, the condition for canceling the WAITING state of the task in front of the queue is satisfied.

2.4 Workaround

Before and after issuing irel_mpl with or without rel_mpl in Condition (2), change the level of the interrupt mask to that of the kernel interrupt mask as follows:

(1) If rel_mpl issued

```
#include <machine.h>
int old_imask;
old_imask = get_imask();
chg_ims(15);  /* Interrupt mask level changed to kernel interrupt
                 mask level (15 in this example) */
irel_mpl(...);  /* As context is interpreted as non-task one while
                   interrupt is masked in product specifications,
                   rel_mpl is changed to irel_mpl */
ichg_ims((IMASK)old_imask);  /* Interrupt mask level resumed */
```

(2) If irel_mpl issued

```
#include <machine.h>
int old_imask;
old_imask = get_imask();
set_imask(15);  /* Interrupt mask level changed to kernel interrupt
                   mask level (15 in this example) */
irel_mpl(...);
set_imask(old_imask);  /* Interrupt mask level resumed */
```

3. Schedule of Fixing the Problems

These problems have been resolved in the following latest versions:

- HI7000/4 V.2.02 Release 04
- HI7700/4 V.2.03 Release 02
- HI7750/4 V.2.02 Release 04

They will be opened on the download site at

HI7000/4: http://www.renesas.com/hi7000_4_download

HI7700/4: http://www.renesas.com/hi7700_4_download

HI7750/4: http://www.renesas.com/hi7750_4_download
from June 5 on. So update yours to any of them you want.
Free-of-charge online update is available. For details see RENESAS TOOL NEWS Document No. 080601/tn5, "Five Real-Time OSes for the SuperH MCU Family Revised," on the Web page at
http://tool-support.renesas.com/eng/toolnews/080601/tn5.htm

Note, however, that the update from V.1 to the latest version is not allowed in every product. So if you are using any of the V.1 products, you are encouraged to purchase its latest version.
