How to change interrupt registers with Holly

H/W#1R&D Dept March 10th, 1999

There is a set procedure for changing interrupt registers with Holly. If you do not follow this procedure, it may happen that processing jumps to the interrupt routine even though nothing is set on the INTEVT register or that an interrupt already cleared is applied again by mistake.

Procedure Case1: general

When changing the Holly interrupt registers, follow steps (1)- (4) as shown below.

(1) Disable processing of other interrupts by the CPU. (There are two ways of doing this as follows.)

(1.a) Use SR. IMASK to raise the priority of Holly interrupts above that of other interrupts.
(1.b) Set SR. BL to 1.

(2) Change the Holly interrupt registers.

(3) Read the changed Holly registers twice.

(4) Undo the change made in step (1).

Procedure Case2: Clearing interrupt from external devices

Interrupt controlled by Holly includes the ones issued by external devices. The external devices that can issue interrupts are GD-ROM, AICA, MODEM, and G2 extended device. Related registers are ISTEXT, IM2EXT, IML4EXT, and IML6EXT. Special steps are required to clear those interrupts. Steps (1) and (4) are the same as procedure case1.

(1) Disable processing of other interrupts by the CPU. (There are two ways of doing this as follows.)

(1.a) Use SR. IMASK to raise the priority of Holly interrupts above that of other interrupts.
(1.b) Set SR. BL to 1.

(2) Use interrupt control registers of external devices to clear interrupt. If more than one interrupt are needed to be cleared, execute them here at once.

(2.5) By reading ISTEXT register (external interrupt status), confirm if interrupts have been cleared.

(3) Read the ISTEXT register twice.

(4) Undo the change made in step (1).

Supplement

(supplement 1) The foregoing procedure is required when an interrupt that was formerly present disappears. In other cases, no error occurs even if the procedure is not followed.

<Necessary> When you wish to clear the status of an unmasked (valid) interrupt.
<Unnecessary> When you wish to clear the status of a masked (invalid) interrupt.
<Necessary> When you wish to mask an interrupt (so that it no longer occurs).
<Unnecessary> When you wish to unmask an interrupt (so that it occurs).

(supplement 2) While in the interrupt processing routine, steps (1) and (4) above are ordinarily not required because SR. BL =1 (unless it has been changed intentionally). Nonetheless, the procedure is necessary even within interrupt processing routines if SR. BL is set to 0 and multiple interrupts are enabled. Steps (1) – (4) must be followed when outside of interrupt routines. (Of course, steps (1) and (4) are not needed if either of conditions (1.a) or (1.b) is satisfied.)

(supplement 3) Step (3) above is a dummy read included to maintain the sequence of steps (2) and (4). When steps 2 (an external CPU access) and 4 (an internal CPU access) are executed in immediate succession, the sequence in which the steps are performed may be reversed so that step 4 is done first.

(supplement 4) The value of INTEVT register will be set 0 if setting is failed.

(supplement 5) Dummy read mentioned above (3) must be executed until interrupts can be accepted. However, two read after each change is not required. Two dummy read after all the changes are done is enough so that interrupt can be accepted. No problems if there is any other accesses.

Reference

(Reference 1) The Holly interrupt registers are as follows.

ISTNRM (A05F6900) normal interrupt status
ISTERR (A05F6908) error interrupt status
IML2NRM (A05F6910) Level2 normal interrupt mask control
IML2EXT (A05F6914) Level2 external interrupt mask control
IML2ERR (A05F6918) Level2 error interrupt mask control
IML4NRM (A05F6920) Level4 normal interrupt mask control
IML4EXT (A05F6924) Level4 external interrupt mask control
IML4ERR (A05F6928) Level4 error interrupt mask control
IML6NRM (A05F6930) Level6 normal interrupt mask control
IML6EXT (A05F6934) Level6 external interrupt mask control
IML6ERR (A05F6938) Level6 error interrupt mask control

(Reference 2) This procedure must also be followed when processing interrupts from peripheral modules built into the CPU. Refer to the Hardware Manual (19. 2. 3) Only one dummy read is necessary in this case.

(Reference 3) Sequence of events leading to errors (for reference)

a) Within the CPU there is a time lapse between the recognition of an interrupt and the setting of INTEVT. Normally, events occur in this order:

the interrupt is issued
⇩
the CPU detects the interrupt and alters its internal state
⇩
the CPU prepares to handle the interrupt and sets INTEVT (for the issued interrupt)
⇩
processing of the interrupt begins
⇩
the interrupt is cleared

However, depending on the timing at which the interrupt is cleared, there may be no interrupt to refer to upon setting INTEVT, so that INTEVT is not properly set even though an interrupt has occurred. The sequence of events is as follows.

the interrupt is issued
⇩
the CPU detects the interrupt and alters its internal state
⇩
the interrupt is cleared
⇩

the CPU prepares to handle the interrupt and sets INTEVT (for the issued interrupt) <there is no interrupt at this point>

⇩

processing of the interrupt begins

The above procedure prevents interrupts from being cleared between the time they are detected and when INTEVT is set.

b) Also, when ending up an interrupt routine like below:

```
MOV.L   R0,@R1 ;        write to clear interrupt
        RTE
        NOP
```

When programs get out of an interrupt routine, it may give occasion to perform the processing again. It happens such case that the interrupt is not cleared yet or that it is cleared within the holly but not within the CPU. The following prevents it by securing time for the CPU state to be interrupt cleared.

```
        MOV.L   R0,@R1 ;        write to clear interrupt
        MOV.L   @R1,R0 ;        dummy read1
        MOV.L   @R1,R0 ;        dummy read2
        RTE
        NOP
```