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.

Date: Jun.11.2008

RENESAS TECHNICAL UPDATE

Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan
Renesas Technology Corp.

Product Category	MPU&MCU		Document No.	TN-H8*-A386A/E	Rev.	1.00
Title	Notes on Rewriting DTC Enable Registers (DTCER)		Information Category	Technical Notification		
Applicable Product	H8SX/1622 group, H8SX/1638 group H8SX/1648 group, H8SX/1650 group H8SX/1651 group, H8SX/1653 group H8SX/1657 group, H8SX/1658R group H8SX/1663 group, H8SX/1668R group	Lot No. All lots	Reference Document	See below.		

We would like to inform you of some notes on rewriting the DTC enable registers (DTCER) of the above listed products. Specifically, if rewriting the DTCER conflicts with generation of a DTC activation source interrupt, both DTC activation and CPU interrupt exception handling may be executed; it has turned out that a double interrupt may be caused in some cases. The following gives details of this problem, which can be prevented according to the procedures in figure 1.

1. Preventive Measures (DTCER Rewriting Method)

The following DTCER rewriting procedures can prevent a conflict between rewriting the DTCER and a DTC activation source interrupt from causing execution of both DTC activation and CPU interrupt exception handling. The DTCER rewriting procedure depends on the interrupt control mode: set the I bit in CCR to "1" in interrupt control mode 0, and set the interrupt mask level to "7" (I2 to I0 bits in EXR = B'111) in interrupt control mode 2. This masks an interrupt to the CPU thus preventing execution of interrupt exception handling by the CPU.

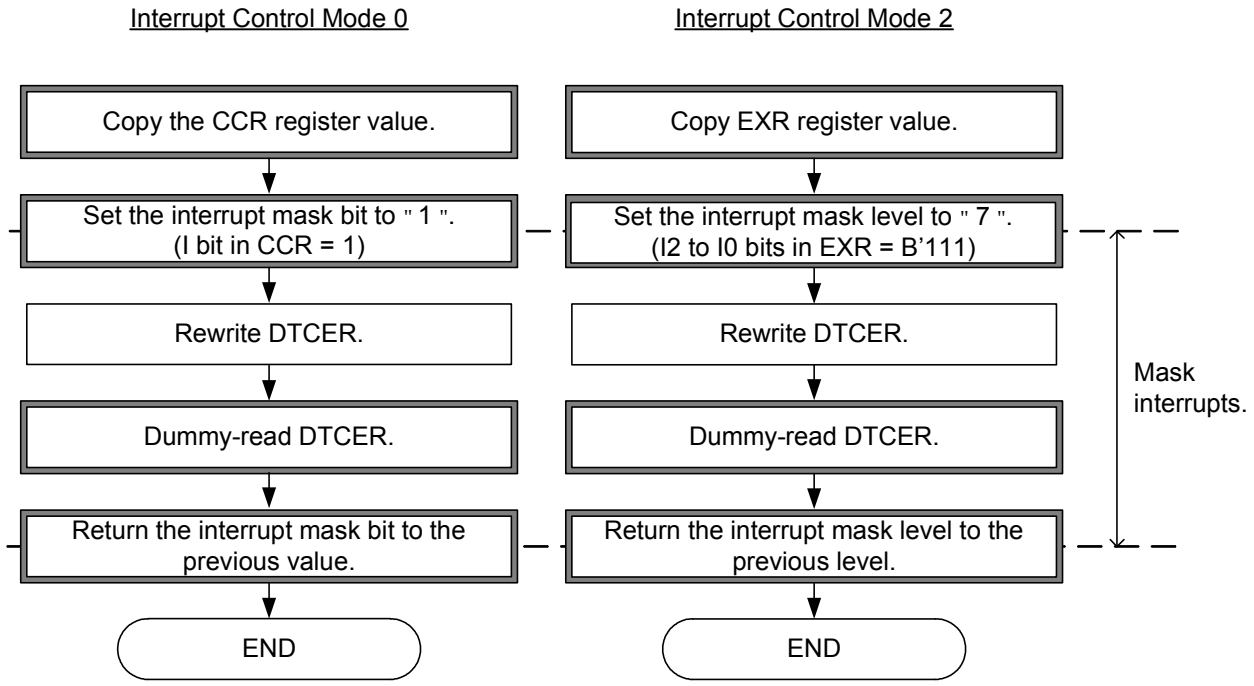

Figure 1 Recommended DTCER Rewriting Procedures

2. Conditions on Which both DTC Activation and CPU Interrupt Exception Handling are Executed and Specific Executed Operations

If the above preventive measure is not taken, both DTC activation and CPU interrupt exception handling may be executed when rewriting the DTC enable registers (DTCER) conflicts with generation of a DTC activation source interrupt.

If the above preventive measure is taken, CPU interrupt exception handling is not executed.

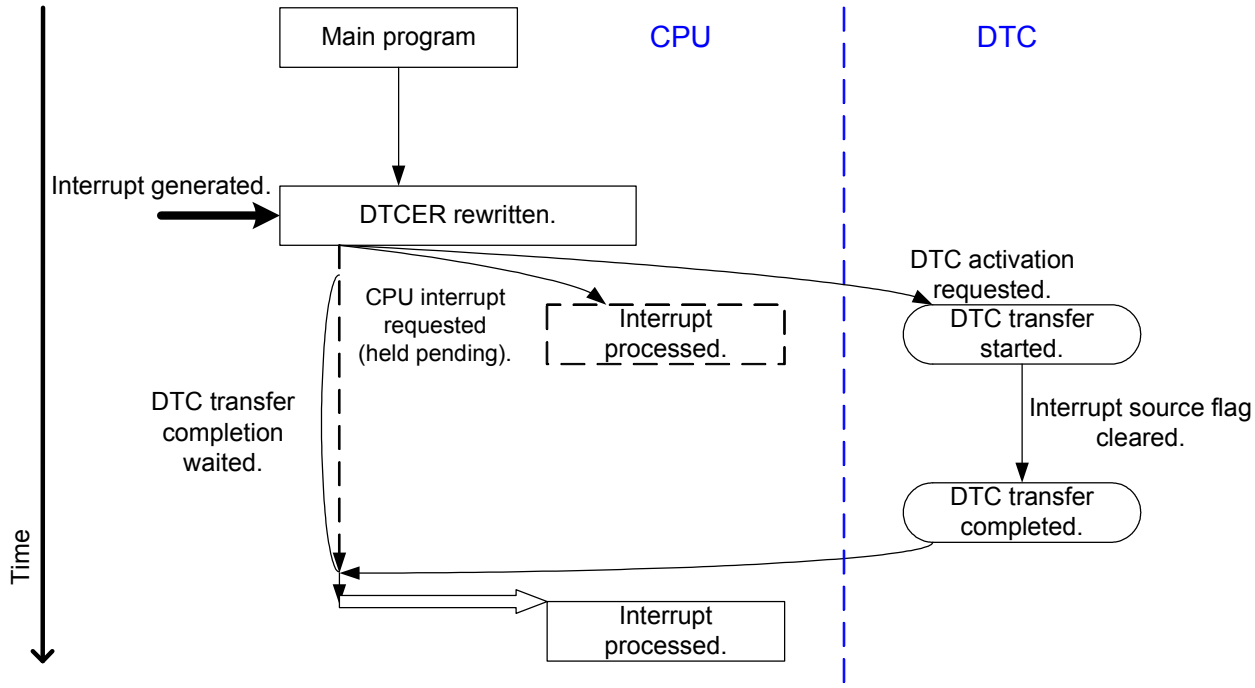

Figure 2 Execution of Both DTC Activation and CPU Interrupt Exception Handling

3. Conditions on Which a Double Interrupt is Caused and Executed Operations

• Conditions

— Interrupt control mode 2 is set (INTM1 bit in INTCR = 1).

— Interrupts A and B are enabled (interrupts A and B can be any interrupts).

— While interrupt A is processed, the DTC transfer enable register (DTCER) for interrupt B is rewritten.

— While interrupt A is processed, interrupt B is generated.

When all the above conditions are satisfied and rewriting the DTC enable register conflicts with generation of a DTC activation source interrupt B, double exception handling of interrupt A may be caused.

• Operations

As described above, when all the above listed conditions are satisfied and rewriting the DTCER for interrupt B conflicts with generation of interrupt B, double exception handling of interrupt A may be generated.

When such a conflict occurs, both a DTC transfer request and a CPU interrupt request may be generated. When both the requests are generated and the DTC transfer request priority is higher than the CPU interrupt request priority, the DTC transfer request is processed first, holding the CPU interrupt request pending.

After the DTC transfer is completed, the interrupt B source flag is cleared by the DTC. Due to this, exception handling of interrupt A, whose interrupt source flag has not been cleared, is erroneously executed again instead (double processing of interrupt A).

Figure 3 shows a double processing example of interrupt A and figure 4 shows an operation example when a preventive measure is taken.

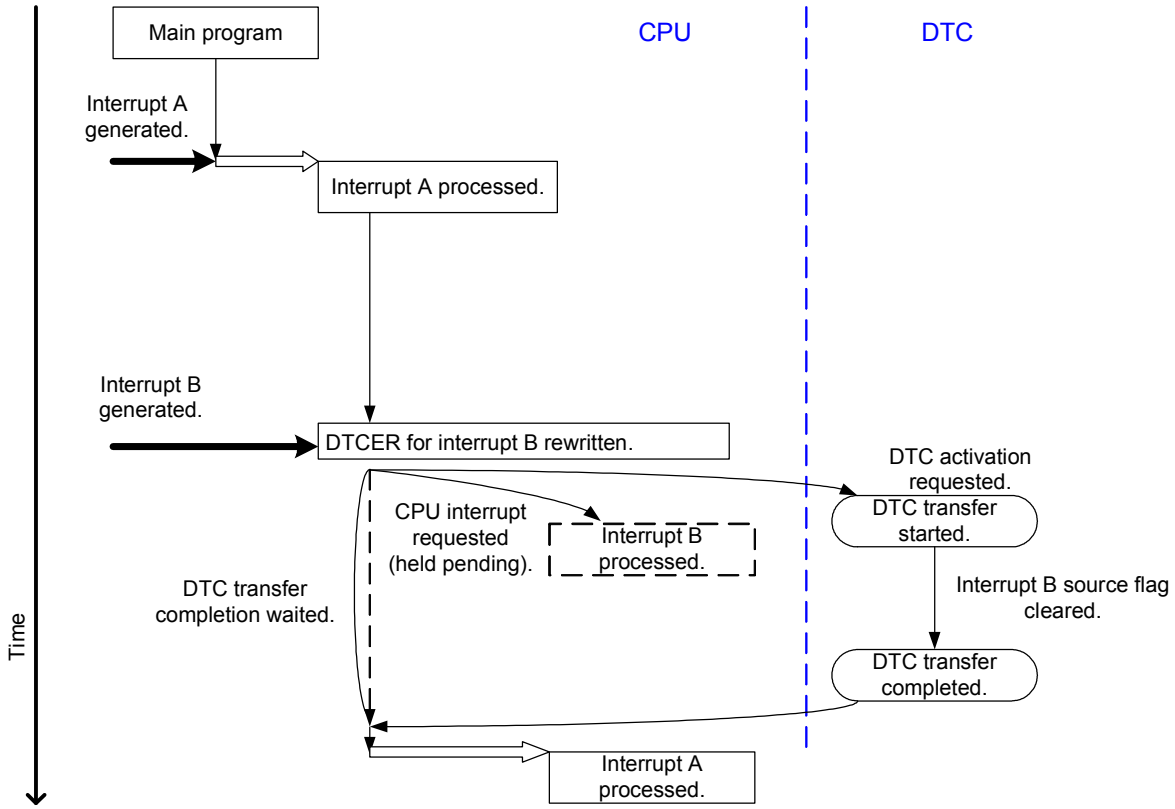

Figure 3 Double Processing Example of Interrupt A

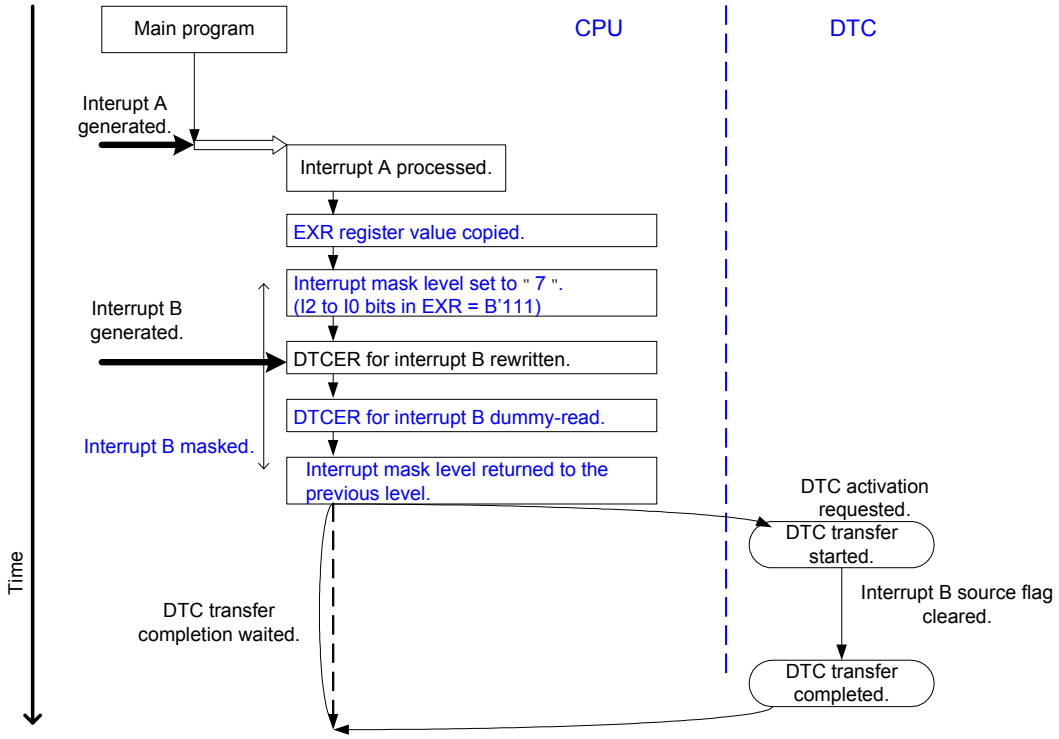

Figure 4 Operation Example when a Preventive Measure is Taken

Reference Document

H8SX/1622 Group Hardware Manual (revision 1.00 REJ09B0414-0100)

H8SX/1638 Group Hardware Manual (revision 1.00 REJ09B0364-0100)

H8SX/1648 Group Hardware Manual (revision 1.00 REJ09B0365-0100)

H8SX/1650 Group Hardware Manual (revision 2.00 REJ09B0311-0200)

H8SX/1651 Group Hardware Manual (revision 2.00 REJ09B0248-0200)

H8SX/1653 Group Hardware Manual (revision 1.00 REJ09B0219-0100)

H8SX/1657 Group Hardware Manual (revision 2.00 REJ09B0341-0200)

H8SX/1658R Group Hardware Manual (revision 1.00 REJ09B0413-0100)

H8SX/1663 Group Hardware Manual (revision 1.00 REJ09B0294-0100)

H8SX/1668R Group Hardware Manual (revision 1.00 REJ09B0412-0100)