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.

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	Lot No.				
Tioduct	H8SX/1648 group, H8SX/1650 group H8SX/1651 group, H8SX/1653 group H8SX/1657 group, H8SX/1658R group H8SX/1663 group, H8SX/1668R group	All lots	Reference Document	See below.		
We would lik	te to inform you of some notes on rewriting	the DTC enable	e registers (DTCI	ER) of the above listed [products.	
Specifically, i	if rewriting the DTCER conflicts with generation	ation of a DTC a	ctivation source	interrupt, both DTC act	ivation an	d CPU
nterrupt exc	eption handling may be executed; it has tu	rned out that a c	louble interrupt r	may be caused in some	cases. Th	ne
ollowing give	es details of this problem, which can be pre	evented accordi	ng to the proced	ures in figure 1.		
1. Preventive	e Measures (DTCER Rewriting Method)					
The following	g DTCER rewriting procedures can preven	t a conflict betwe	een rewriting the	DTCER and a DTC act	tivation so	ource
nterrupt fron	n causing execution of both DTC activation	and CPU interr	upt exception ha	andling. The DTCER rev	vriting pro	cedure
depends on	the interrupt control mode: set the I bit in C	CR to "1" in inte	errupt control mo	de 0, and set the interru	ipt mask l	evel to
'7" (I2 to I0 b	oits in EXR = B'111) in interrupt control mo	de 2. This mask	s an interrupt to	the CPU thus preventing	a executio	on of
			o un interrupt to	and of o ands preventing	y executio	
	eption handling by the CPU.				g executio	
	eption handling by the CPU.				y execution	
	eption handling by the CPU. Interrupt Control Mode 0		rupt Control Mc		g executio	
interrupt exc	Interrupt Control Mode 0	Inter	rupt Control Mc	ode 2	g executio	
interrupt exc		Inter		ode 2	geneturi	
interrupt exc	Interrupt Control Mode 0 Copy the CCR register value.	Inter Copy	rupt Control Mo v EXR register v	ode 2 value.	geneturi	
nterrupt exc	Interrupt Control Mode 0 Copy the CCR register value. tet the interrupt mask bit to " 1 ".	Inter Copy	rupt Control Mo v EXR register v ↓ terrupt mask lev	ode 2 /alue. /el to " 7 "		
nterrupt exc	Interrupt Control Mode 0 Copy the CCR register value.	Inter Copy	rupt Control Mo v EXR register v	ode 2 /alue. /el to " 7 "		
nterrupt exc	Interrupt Control Mode 0 Copy the CCR register value. tet the interrupt mask bit to " 1 ".	Inter Copy Set the int (I2 to I	rupt Control Mo v EXR register v ↓ terrupt mask lev	ode 2 value. /el to " 7 ". B'111)		
nterrupt exc	Interrupt Control Mode 0 Copy the CCR register value. t the interrupt mask bit to "1 ". (I bit in CCR = 1)	Inter Copy Set the int (I2 to I	rupt Control Mo v EXR register v v terrupt mask lev 0 bits in EXR =	ode 2 /alue. /el to " 7 ". B'111)	 /lask	
nterrupt exc	Interrupt Control Mode 0 Copy the CCR register value. the interrupt mask bit to "1". (I bit in CCR = 1) Rewrite DTCER.	Inter Copy Set the inf (I2 to I	rupt Control Mo v EXR register v v EXR register v v errupt mask lev 0 bits in EXR = v Rewrite DTCER	ode 2 /alue. /el to " 7 ". B'111) 2.	_	
interrupt exc	Interrupt Control Mode 0 Copy the CCR register value. t the interrupt mask bit to "1 ". (I bit in CCR = 1)	Inter Copy Set the inf (I2 to I	rupt Control Mo v EXR register v v terrupt mask lev 0 bits in EXR =	ode 2 /alue. /el to " 7 ". B'111) 2.	 /lask	
	Interrupt Control Mode 0 Copy the CCR register value. t the interrupt mask bit to " 1 ". (I bit in CCR = 1) Rewrite DTCER. Dummy-read DTCER.	Inter Copy Set the int (I2 to I F Du	rupt Control Mo v EXR register v v errupt mask lev 0 bits in EXR = v Rewrite DTCER v mmy-read DTC	ode 2 /alue. /el to " 7 ". B'111) ER. ER.	 /lask	
	Interrupt Control Mode 0 Copy the CCR register value. the interrupt mask bit to "1". (I bit in CCR = 1) Rewrite DTCER.	Inter Copy Set the int (I2 to I F Du	rupt Control Mo v EXR register v v EXR register v v errupt mask lev 0 bits in EXR = v Rewrite DTCER	<u>value.</u> <u>vel to " 7 ".</u> <u>B'111)</u> <u>R.</u> ER. level to the	 /lask	
	Interrupt Control Mode 0 Copy the CCR register value. et the interrupt mask bit to " 1 ". (I bit in CCR = 1) Rewrite DTCER. Dummy-read DTCER. turn the interrupt mask bit to the previous value.	Inter Copy Set the int (I2 to I F Du	rupt Control Mo v EXR register v v errupt mask lev 0 bits in EXR = v Rewrite DTCER v mmy-read DTC v interrupt mask previous level.	<u>value.</u> <u>vel to " 7 ".</u> <u>B'111)</u> <u>R.</u> ER. level to the	 /lask	
	Interrupt Control Mode 0 Copy the CCR register value. to the interrupt mask bit to " 1 ". (I bit in CCR = 1) Rewrite DTCER. Dummy-read DTCER. turn the interrupt mask bit to the	Inter Copy Set the int (I2 to I F Du	rupt Control Mo v EXR register v v errupt mask lev 0 bits in EXR = v Rewrite DTCER v mmy-read DTC v interrupt mask	<u>value.</u> <u>vel to " 7 ".</u> <u>B'111)</u> <u>R.</u> ER. level to the	 /lask	
	Interrupt Control Mode 0 Copy the CCR register value. Copy the interrupt mask bit to " 1 ". (I bit in CCR = 1) Rewrite DTCER. Dummy-read DTCER. Lummy-read DTCER. Lum the interrupt mask bit to the previous value. Lum the interrupt mask bit to the previous value. Lum the interrupt mask bit to the previous value.	Inter Copy Set the int (12 to 1 I Dur Return the	rupt Control Mo v EXR register v v errupt mask lev 0 bits in EXR = v Rewrite DTCER v mmy-read DTC v mmy-read DTC v interrupt mask previous level. v END	value. /el to " 7 ". B'111) c. BR. level to the	 /lask	
	Interrupt Control Mode 0 Copy the CCR register value. Copy the interrupt mask bit to " 1 ". (I bit in CCR = 1) Rewrite DTCER. Dummy-read DTCER. Lummy-read DTCER. Lum the interrupt mask bit to the previous value. Lum the interrupt mask bit to the previous value. Lum the interrupt mask bit to the previous value.	Inter Copy Set the int (12 to 1 I Dur Return the	rupt Control Mo v EXR register v v errupt mask lev 0 bits in EXR = v Rewrite DTCER v mmy-read DTC v interrupt mask previous level.	value. /el to " 7 ". B'111) c. BR. level to the	 /lask	



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



RENESAS TECHNICAL UPDATE TN-H8*-A386A/E





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 REJ09B0341-0200) H8SX/1668R Group Hardware Manual (revision 1.00 REJ09B0413-0100) H8SX/1663 Group Hardware Manual (revision 1.00 REJ09B0413-0100)

